INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

MESTRADO EM ENGENHARIA MECÂNICA




DESIGN FOR MANUFACTURE USING MACHINING FEATURES ON CNC MACHINING CENTERS

PEDRO MANUEL DE OLIVEIRA FESTA

novembro de 2021

POLITÉCNICO
DO PORTO

Instituto Superior de Engenharia do Porto

DESIGN FOR MANUFACTURE USING MACHINING FEATURES ON CNC MACHINING CENTERS

Pedro Manuel de Oliveira Festa

2021

ISEP – School of Engineering, Polytechnic of Porto
Department of Mechanical Engineering

POLITÉCNICO
DO PORTO

DESIGN FOR MANUFACTURE USING MACHINING FEATURES ON CNC MACHINING CENTERS

Pedro Manuel de Oliveira Festa
1160895

Dissertation presented to ISEP – School of Engineering, Polytechnic of Porto to fulfill the requirements necessary to obtain a Master's degree in Mechanical Engineering, carried out under the guidance of Doctor Francisco José Gomes da Silva, from the department of Mechanical Engineering, ISEP – School of Engineering, Polytechnique of Porto and co-supervision of Doctor George Christopher Vosniakos, on framework of Erasmus, from the School of Mechanical Engineering, Laboratory of Manufacturing Technology, National Technical University of Athens.

2021
ISEP – School of Engineering, Polytechnic of Porto
Department of Mechanical Engineering

JURY

President
Isabel Cristina Silva Barros Rodrigues Mendes Pinto, PhD
Associate Professor, Department of Mathematics, ISEP
Supervisor
Francisco José Gomes da Silva, PhD
Associate Professor w/ Habilitation, Department of Mechanical Engineering, ISEP
Second supervisor
George Christopher Vosniakos, PhD
Professor, Manufacturing division, NTUA
Examiner
Abílio Manuel Pinho de Jesus, PhD
Associate Professor, Department of Mechanical Engineering, FEUP

ACKNOWLEDGEMENTS

Firstly, I would like to say thank you to my supervisor and professor at ISEP, Doctor Francisco José Gomes da Silva for his dedication and support since the beginning. For all the help provided over the last months.

Secondly, I would like to thank my supervisor at NTUA, Doctor George Christopher Vosniakos, for the opportunity to do this work. For all the meetings, support and dedication that led to the presentation of this work.

To my friends that were present.

Lastly, to my parents and sister who gave me support and motivation.

KEYWORDS

Machining; Feature-based design; Feature-based manufacturing; Milling; CAD-CAM

ABSTRACT

Computer-Aided Design (CAD) and Computer-Aided Manufacturing (CAM) systems have become more and more needed and useful in the machining processes environment. In order to achieve competitive advantage, companies adopted new manufacturing methods. As a consequence, and in machining processes context, the interaction of CAD and CAM has growth over the years in order to increase the production efficiency, as well as to reduce costs and time.

The development of this work started with an extensive literature review. In that review, the author found that only a few articles approached the interaction or integration of CAD and CAM systems. Moreover, the authors that studied this interaction focused on systems for turning parts. Thus, there is a gap in the literature related to the integration and automation of these systems when applied to milling parts. Therefore, the purpose of this dissertation is to enable the interaction of these systems in order to provide a completely automated process since the design stage until the machining stage.

Finally, the process' implementation showed that the developed algorithm was able to satisfy the initial requirements of this work, i.e., when given a set of initial parameters, the program drew the required geometry, and then generated the required G-code, such that this code can be sent to the CAM software to machine the workpiece, thereby obtaining the final product.

PALAVRAS CHAVE

Maquinagem; Feature-based design; Feature-based manufacturing; Fresagem; CAD-CAM

RESUMO

Os sistemas Computer-Aided Design *(CAD) and* Computer-Aided Manufacturing*(CAM) estão, cada vez mais, a ser mais necessários e úteis no contexto da maquinagem. De modo a conseguir vantagem competitiva, as empresas têm adotado novos métodos de produção. Consequentemente, no contexto da indústria da maquinagem, a interação entre CAD e CAM tem crescido nos últimos anos, de modo a permitir uma maior eficácia na produção, assim como também redução de tempo e custo.*

O desenvolvimento deste trabalho começou com uma extensa revisão da literatura. Nesta revisão, o autor apercebeu-se que apenas alguns artigos se debruçaram sobre a interação ou integração dos sistemas CAD e CAM. Para além disso, os autores desses artigos focaram-se em sistemas para torneamento. Assim, constata-se que existe um espaço livre na literatura no que diz respeito à integração destes sistemas quando aplicados à fresagem. Por isso, o objetivo desta dissertação é permitir a interação dos dois sistemas referidos, de forma a promover um processo completamente automático desde o design até à maquinagem.

Por fim, a implementação do processo mostrou que o algoritmo desenvolvido alcançou os objetivos iniciais do trabalho, ou seja, baseando-se apenas nos parâmetros fornecidos, o programa desenhou as geometrias necessárias, sendo depois capaz de gerar o código G respetivo, para que este possa ser transferido para o centro de maquinagem, de modo a que o material possa ser maquinado, dando origem ao produto final.

LIST OF SYMBOLS AND ABBREVIATIONS

List of abbreviations

CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
CNC	Computerized Numeric Control
NC	Numeric Control
VBA	Visual Basic for Applications

List of units

mm	Millimeters

FIGURES INDEX

FIGURE 1 - EVOLUTION OF MACHINING SYSTEMS AND MACHINE TOOLS (JURKOVIC & JURKOVIC, 2008) . 8
FIGURE 2 - EVOLUTION OF MACHINING PROCESSES AND TOLERANCES (OTOBONI ET AL., 2013) 8
FIGURE 3 - PORTUGAL'S METALWORK INDUSTRY. ADAPTED FROM (CARDOSO & QUELHAS, 2020) 9
FIGURE 4 - WIREFRAME OBJECT (HOFFMANN, 1989) 10
FIGURE 5 - COLLABORATIVE CAD ENVIRONMENT. ADAPTED FROM LEE *ET AL.* (2010) 11
FIGURE 6 - ARCHITECTURE OF A SYSTEM THAT INTEGRATES CAD AND CAM (ALTINTAS ET AL., 2014) 13
FIGURE 7 - MILLING OPERATIONS (LEE ET AL., 2020) 13
FIGURE 8 - DRILLING STAGES (SURESH KUMAR ET AL., 2020) 14
FIGURE 9 - WORKPIECE WITH FOUR DIFFERENT FEATURES (XU, 2009) 15
FIGURE 10 – FEATURE-BASED DESIGN AREAS OF RESEARCH (JONG *ET AL.*, 2009) 16
FIGURE 11 - DESIGN AND MANUFACTURING (MACHINING) FEATURES (ZHENG ET AL., 2012) 19
FIGURE 12 - CATEGORIES OF MACHINING FEATURE (LIU ET AL., 2017) 20
FIGURE 13 - MANUFACTURING FEATURES (BESHARATI-FOUMANI ET AL., 2020) 21
FIGURE 14 - DIFFERENT MACHINING FEATURES – (A) CLOSED MACHINING FEATURES; (B) – SEMI-CLOSED MACHINING FEATURES; (C) – OPEN MACHINING FEATURES (JI ET AL., 2018) 21
FIGURE 15 – POCKET TEST PERFORMED BY DITTRICH *ET AL.* (2019) 25
FIGURE 16 - SELF-OPTIMIZATION MACHINING SYSTEM (MÖHRING *ET AL.,* 2020) 26
FIGURE 17 - METHOD FOR SELF-OPTIMIZING ADAPTION OF PROCESS PARAMETERS (DENKENA ET AL., 2021) 26
FIGURE 18 - HOLE FEATURE 30
FIGURE 19 - HOLE FEATURE'S SKETCH 31
FIGURE 20 - SECTION VIEW OF THE HOLE FEATURE 32
FIGURE 21 - POCKET FEATURE 33
FIGURE 22 - SLOT FEATURE 34
FIGURE 23 - EXAMPLE OF A "FOR" LOOP 36
FIGURE 24 - ZIG ZAG STRATEGY 37
FIGURE 25 - "FOR" CYCLE FOR CALCULATING THE DEPTH 37
FIGURE 26 - CONTOUR STRATEGY 38
FIGURE 27 - HOLE FEATURE'S USER INTERFACE 39
FIGURE 28 - POCKET FEATURE'S USER INTERFACE 40
FIGURE 29 - POCKET 40
FIGURE 30 - SLOT'S USER INTERFACE 41
FIGURE 31 - SLOT COORDINATE POINTS 41
FIGURE 32 - WORKPIECE 42
FIGURE 33 - CNC MILLING MACHINE 43
FIGURE 34 - TOOL EXAMPLE 43
FIGURE 35 - POCKETS IN THE PROPOSED OBJECT 44
FIGURE 36 - HOLES IN THE PROPOSED GEOMETRY 45
FIGURE 37 - SLOTS IN THE PROPOSED GEOMETRY 46

FIGURE 38 - DETAIL OF THE OVERLAPPED SLOTS 47
FIGURE 39 - DESIGNED WORKPIECE 47
FIGURE 40 - FIRST SIMULATION FRAME 49
FIGURE 41 – SECOND SIMULATION FRAME 49
FIGURE 42 – FINAL PRODUCT 50
FIGURE 43 - ANOTHER PERSPECTIVE OF THE FINAL PRODUCT 50

TABLES INDEX

TABLE 1 - FEATURE-BASED DESIGN STUDIES ... 17
TABLE 2 - FEATURE-BASED MANUFACTURING STUDIES ... 22
TABLE 3 - BITS OF G-CODE FROM DIFFERENT FEATURES ... 48

INDEX

1 INTRODUCTION 3

1.1 Contextualization 3

1.2 Main Goals 3

1.3 Motivation and Methodology 3

1.4 Framework 4

2 LITERATURE REVIEW 7

2.1 Machining 7

2.1.1 Evolution of machining systems 7

2.1.2 The relevance of machining in the metalwork context 9

2.1.3 Economical aspect of machining 9

2.2 Computer-Aided Design 10

2.3 Computer-Aided Manufacturing 12

2.3.1 Different CAM strategies – Milling and Drilling - for prismatic parts 13

2.4 Feature-based design 14

2.5 Feature-based Manufacturing 18

2.6 Constraint modelling 23

2.7 Automated machining operations 24

3 PRACTICAL WORK 29

3.1 Algorithm development 29

3.2 Algorithm programming 30

3.2.1 Library of features 30

3.2.2 Toolpath creation 35

3.3 User Interface 39

3.4 Results and critical analysis 42

3.4.1 Indicative CNC programming case 42

3.4.2 Interactive Process Planning 44

3.4.3 Resulting G-code 48

3.4.4 Discussion 51

4 CONCLUSIONS AND PROPOSALS OF FUTURE WORKS 55

4.1 Conclusions..55

4.2 Proposals of future works...55

REFERENCES AND OTHER SOURCES OF INFORMATION 59

ANNEXES.. 71

Annex 1...71

Annex 2...96

INTRODUCTION

1.1 Contextualization

1.2 Main Goals

1.3 Motivation and Methodology

1.4 Framework

1 INTRODUCTION

1.1 Contextualization

The present dissertation was developed in the context of the master's degree in Mechanical Engineering in ISEP, in the framework of Erasmus in School of Mechanical Engineering, Laboratory of Manufacturing Technology, National Technical University of Athens.

In the past years, manufacturing organizations have been facing a lot of challenges due to countless factors, such as economic crisis. Consequently, these companies have been developing new methods in order to live through the different challenges and improve their performance. Another key aspect is cost and time reduction.

New methods are essentially based on the use of technology. In fact, computers are heavily involved in product development. In the mid of 19th century, Computer-Aided Design (CAD) was firstly introduced, providing new and better ways of designing the products. Later, Computer-Aided Manufacturing (CAM) was created. The evolution of these different technologies over the years allowed their partial integration. As a result, manufacturing companies tried to take the most out of these two computer tools. The feature-based systems concept emerged from that. They allow a more automated process with less human errors, and, in addition, they save time and allow a cost reduction.

1.2 Main Goals

The present work has as main goal the development of an algorithm applied to design and machining by milling process, using features. To do so, a few objectives have been defined:

- Characterize different common and basic features that can have different uses in order to allow different geometry creations;
- Development of macros that, based on given parameters, automatically design the features in the CAD software;
- Implement a way to automatically generate the G-code of the designed features in the already created algorithm;
- Perform a simulation of the work developed in order to analyze it and seek for future possible improvements.

1.3 Motivation and Methodology

This dissertation development started with a deep understanding of the problematic. For that, an extensive literature review was done. In fact, it started by looking at scientific articles related to machining in order to see the existing work and the evolution of that theme.

In fact, many authors focused on machining automated operations as a key concept in the present days, due to the fact that they can provide better products and allow time and cost saving. Thus, it was found that in the recent years feature-based systems started to appear, giving both designers and manufacturers new ways of product development. When taking a look at feature-based specific literature, there are countless case studies (further mentioned in this dissertation) related to the different areas of machining. However, few of them integrate CAD and CAM systems. Furthermore, the small amount of articles that refer to the integration of both systems are focused on features design for turning manufacturing. Hence, a gap in the literature related to design and manufacturing of products by milling was found. To address this issue, the purpose of this thesis is to develop an algorithm that can enable design and manufacturing by only using features that will further be machined in a milling center.

On the scope of this literature review, an algorithm was written to address those issues. After that, it was tested with a geometry created by the author. Lastly, the automatically G-code created was tested in a simulator.

1.4 Framework

The way that this thesis is structured allows the reader to follow a guiding line throughout the work. Thus, the thesis is divided into four chapters.

In the first chapter, an introduction to the subject is made. Moreover, this chapter includes the main goals related to the project developed, the motivation to do this work as well as the methodology used. Lastly, it includes the thesis structure.

Afterwards, the second chapter is dedicated to the literature review. Here, the different topics needed to support the development of this thesis are shown.

The third chapter is where all developed work is presented and described. In this section, the code written in the algorithm is detailed. The different features created are explained in depth as well as the code lines related to the G-code process. Then, a demonstration of the work described is shown, both in design and manufacturing parts.

Lastly, the fourth chapter presents the conclusions and suggestions for further work.

LITERATURE REVIEW

2.1 Machining

2.2 Computer-Aided Design

2.3 Feature-based design

2.4 Feature-based Manufacturing

2.5 Constraint modelling

2.6 Computer-Aided Manufacturing

2.7 Automated machining operations

2 LITERATURE REVIEW

2.1 Machining

Machining can be described, according to Youssef & El-Hofy (2008), as the process that allows the removal of material from the workpiece in order to obtain a finished product. The products obtained by this manufacturing process can have different shapes and sizes as well as different surface qualities. In addition, machining parameters are a key point to obtain the desired result (Bissoonauth & Namazi, 2019). Thus, machining is one of the most used manufacturing processes. For example, the aeronautics or automobile industries have high dependence on this manufacturing process. This can be explained by the fact that these products can have low surface roughness or other characteristics specified by the customer at lower cost when comparing with other processes. In addition, some products can only be made using machining (Costa et al., 2018; Gouveia et al., 2016).

2.1.1 Evolution of machining systems

The concept of machining dates back to early 1700s. Due to the growth of demand the manufacturers have to constantly adapt and innovate, therefore this concept has been evolving over the years. Furthermore, nowadays, companies are implementing more and more systems that in one hand, achieve high quality products and, in other hand, allow cost reduction. As a result of this paradigm intelligent machining systems are more common in the recent years (Jing et al., 2020).

According to Jurkovic *et al.* (2008), the first CNC machines appeared in late 1900s. Moreover, the 20th century was marked by the beginning of the implementation of automated machining systems. In Figure 1, it is possible to see the evolution of machining systems over the years. The author also included in this figure the evolution of machine tools.

Other authors such as Federal *et al.* (2013), analyzed the evolution of machining systems in another perspective. In this paper, they analyzed the machining accuracy. In other words, it is possible to acknowledge the evolution of the level of tolerances over the years (Figure 2). The demand for high quality surfaces on ductile materials was triggered by the need of alternatives to machine special components.

With the natural evolution over the years, there are other types of machining that have appeared, such as cryogenic machining. Shokrani *et al.* (2013) produced a review article on this theme. When a super cold liquified gas is used as a coolant in material cutting operations, it is called cryogenic machining. This technique improves the tool life, and it also contributes to the surface finish.

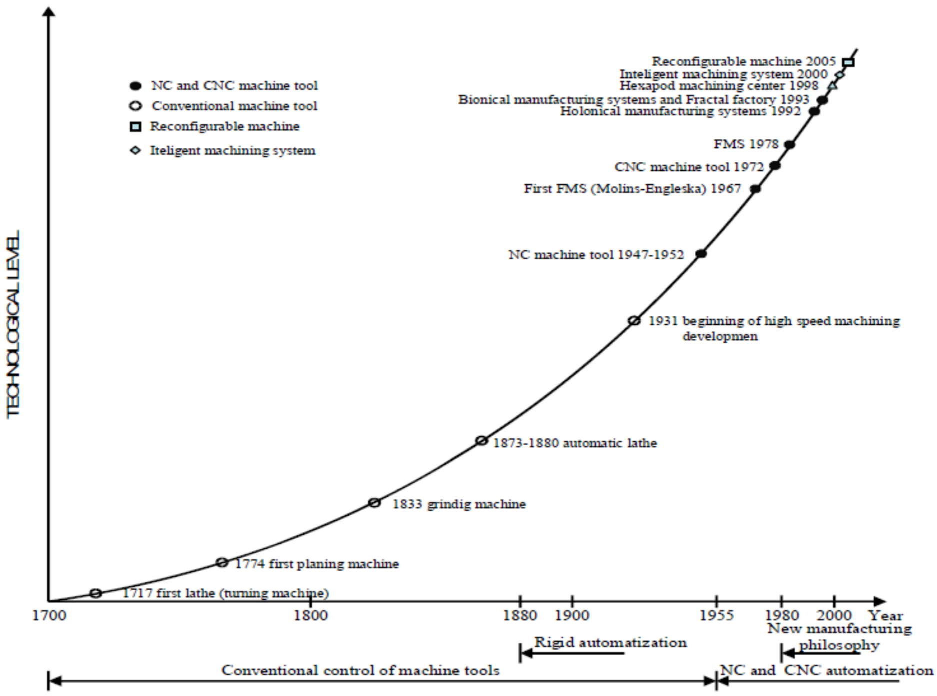

Figure 1 - Evolution of machining systems and machine tools (Jurkovic & Jurkovic, 2008)

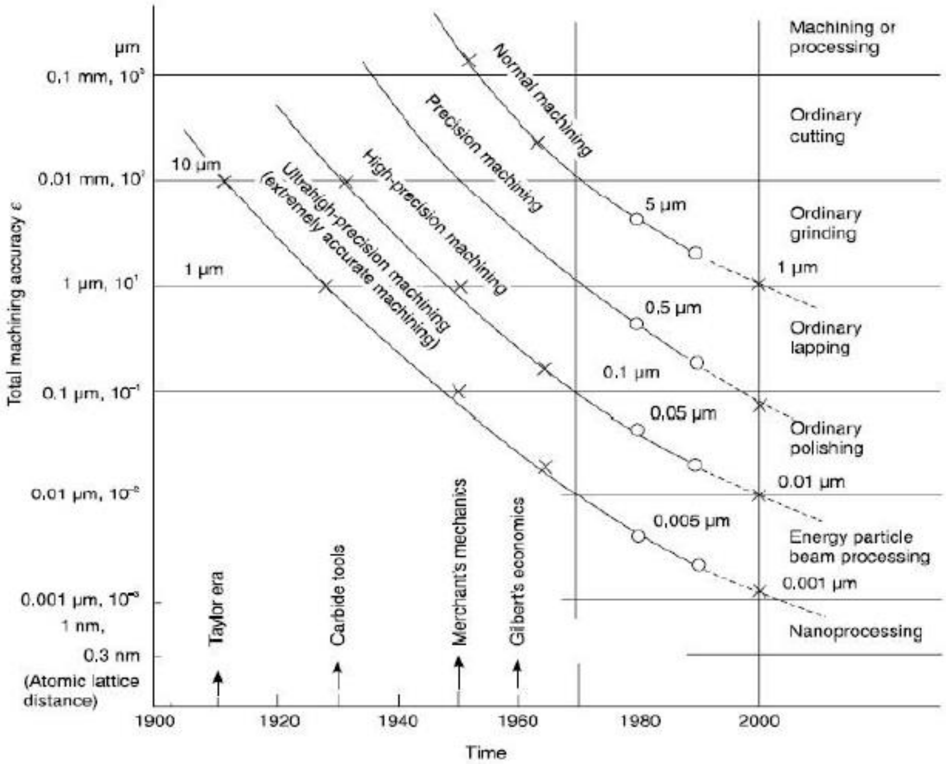

Figure 2 - Evolution of machining processes and tolerances (Otoboni et al., 2013)

2.1.2 The relevance of machining in the metalwork context

Metalwork consists in the transformation of the different metals in final products. The fabrication of the final product can occur by different processes (Brinksmeier et al., 2015). Machining process is still one of the most used in this industry due to the advantages explained in the previous sub-section.

Taking the metalwork industry in Portugal as an example and looking at Figure 3, it can be seen that this industry has been growing in the last few years. When comparing 2015 to 2018 (last year for which there is information), all of the indicators represented have shown an increase (Cardoso & Quelhas, 2020). Furthermore, metalwork industry contributes more to Portuguese exports than any other industry. Moreover, in the last few years big international companies have been investing in this sector in Portugal. Therefore, it is clear that metalwork has a huge role in the manufacturing industry. As a consequence, and since machining is one of the most used processes in metalwork, it has a lot of impact and potential in Portugal's economy (Cardoso & Quelhas, 2018).

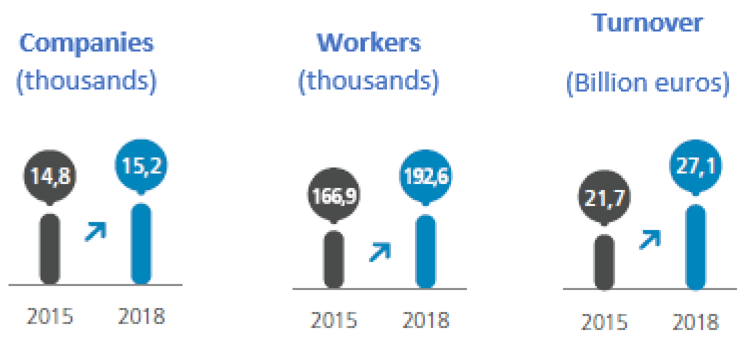

Figure 3 - Portugal's metalwork industry. Adapted from (Cardoso & Quelhas, 2020)

2.1.3 Economical aspect of machining

As previously stated, the demand for high quality products is increasing. Nevertheless, traditional machining systems continue to demand extensive and time-consuming operations. Consequently, there is the need to adopt more efficient and cost-effective machining systems (Panda et al., 2019). Moreover, cutting conditions have an important role in increasing the economics of machining. These conditions have an impact, for instance, in efficiency (Kumar et al., 2021). In order to obtain the best possible economic outcome from machining, several researchers, studied different approaches. Usubamatov *et al.* (2014) created a new mathematical model that allows the calculation of the optimal cutting speed of single and multi-tooling process and, therefore, the maximum productivity rate of a machine tool. This model takes into account some machining parameters, such as machining time (Yingjie & Liling, 2015). In another report, He *et al.* (2017) proposed a method to optimize machining parameters considering economic objectives. The authors used this method in three different case studies and found that the combination of the following machining parameters, mid-high spindle rotation speed, high feed rate, small depth-of-cut and small cutting width contribute to the problem identified. Seeing that CNC machining is one of the most used manufacturing processes nowadays, *Li et al.* (2017) investigated the role of energy consumption on this process and how it can be improved in order to lead to a cost reduction of the final product. Likewise, they concluded that machining parameters such as cutting time or setup time have a huge impact on this matter.

2.2 Computer-Aided Design

Computer-aided design (CAD) has been evolving rapidly as a result of the need of designing products with complex geometry (Kamrani et al., 2015). Industries such as aeronautic or automobile had a huge role on the CAD development. It all started around 1960 with the development of the first computer-based product modelling software. At that time, the system was used mainly for drafts. The geometric information in these systems can be described, among others, by wireframe models (Zeid, 2004). Later, the development in the technology ended up in transforming the purpose of this computer aided tool from drafting to design. As a result, CAD as we know today started to be used (Lyu et al., 2017). Moreover, the wireframe models disappeared. This was an important goal due to the fact that in these wireframe models there were only represented edges and vertices of the objects which resulted in ambiguities when looking at the different representations (Hoffmann, 1989). Taking Figure 4 as an example, in the middle of the polygon there is a square hole. However, it is impossible to determine the direction of the hole.

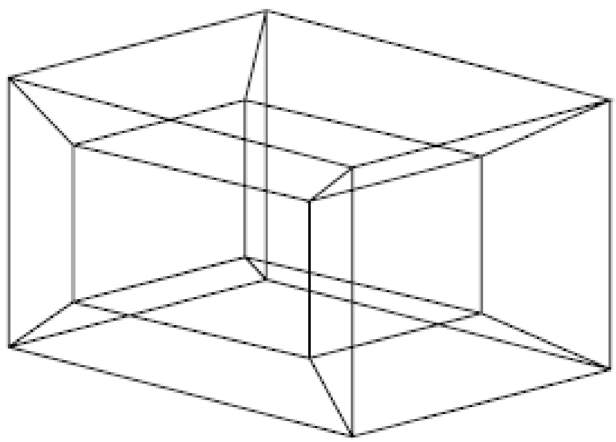

Figure 4 - Wireframe object (Hoffmann, 1989)

Nevertheless, these wireframe objects had small storage requirements and that can explain the use of these objects in the 1960s (Hoffmann, 1989).

The contemporary CAD software allows the users to do multiple operations. In addition to creating complex shapes, it is also possible to combine dozens of parts into one assembly (Li et al., 2020). The possibility of creating such shapes implies in its turn the need of new manufacturing technologies or the adaptation of the existing ones (Ramos et al., 2003). On top of that, the software enables the user to easier implement changes on the original design, while at the same time, providing time and cost savings (Pelliccia et al., 2021).

Furthermore, CAD has a collaborative concept that was first used in the design of complex products. This concept allows changes implementations in the same model by different users, as shown in Figure 5 (Cheng et al., 2016; Lee et al., 2010).

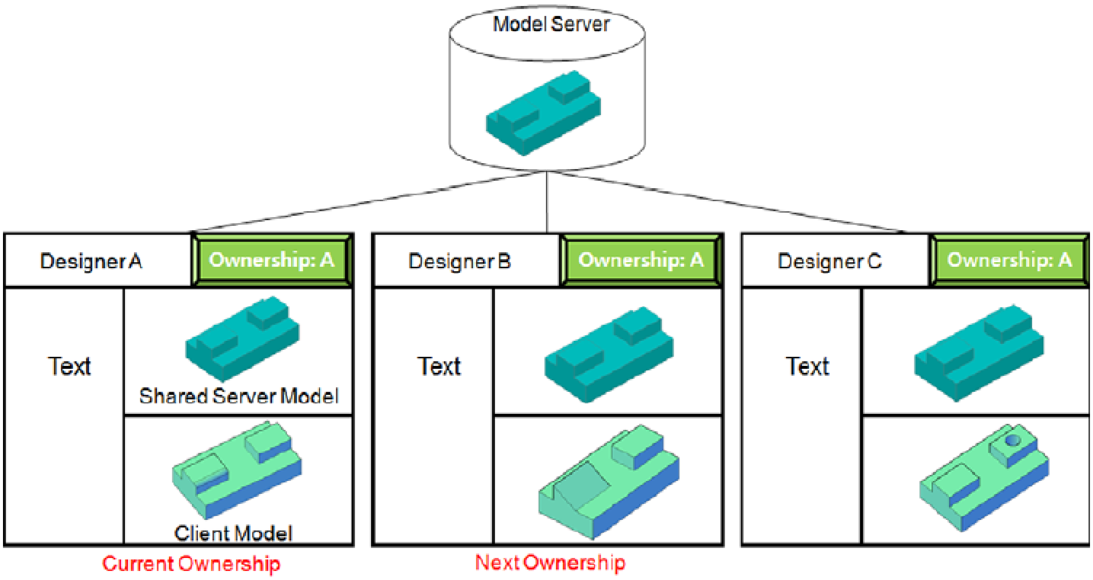

Figure 5 - Collaborative CAD environment. Adapted from Lee *et al.* (2010).

2.3 Computer-Aided Manufacturing

Computer-Aided Manufacturing (CAM) comprehends the use of computer software and the final goal is to control machine tools and machinery in a manufacturing process (Giannuzzi et al., 2020). CAM was created in the 1950s, on that time called Automatically Programmed Tool, with the objective of helping in the process of controlling NC (Numeric Control) machines (Chang et al., 1998; Chen, 2021). Therefore, it can be said that the emerging of NC, initiated the CAM that we know today.

In fact, CAM is still growing. Looking at the data, it is expected that NC market can grow up to a $100billion industry by 2025 (Sousa et al., 2020). Hence, CAM will also grow. As said, CAM has an important role on manufacturing processes, namely in machining processes. To manufacture these products there are three main stages (Michalik et al., 2012):

- Roughing: As the starting point to the process, there is a raw stock from where the material will be removed (in this stage, the material is removed roughly);
- Semi-finishing: After that, this stage removes a small amount of material to facilitate the next step;
- Finishing: Lastly, a slow pass across the material is done using low feed rates. After that, the product is completed with the desired final shape.

Many authors approach the integration of CAD and CAM systems. According to Venu & Komma (2017), this integration can reduce human errors and, in consequence, lead times. For this reason, this integration is a path to industrial success. In fact, nowadays, this integration is standardized. An example of this integration can be seen on a paper written by Altintas *et al.* (2014) where they approach this topic. As shown in Figure 6, the model is designed in a CAD software. Posteriorly, the machining operations needed to manufacture the part are created with the help of CAM software. In addition, this two first stages are done on a virtual environment. The next stages are done on a real-world environment. The G-Code and the workpiece with the desired material are put into a machining center that will create the final product accordingly to all the specifications previously given.

With the evolution of the different computer tools, these programs have included a simulator feature. This has a high relevance due to the fact that these processes are very expensive. Using the simulation before the manufacture of the real products allows a great cost reduction (Majerik & Jambor, 2015).

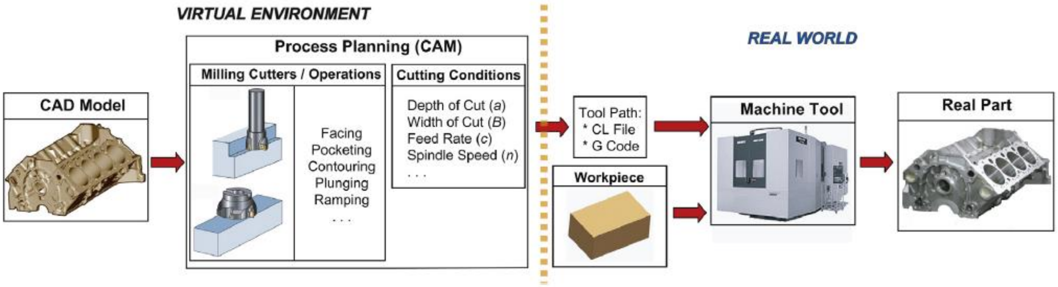

Figure 6 - Architecture of a system that integrates CAD and CAM (Altintas et al., 2014)

The most used machine tools are 3-axis CNC. However, nowadays, 5-axis CNC machine tools are becoming more popular in the industry due to the fact that these ones can machine the same workpiece in less time and can hit larger material remove rates. Moreover, 5-axis CNC machine tools can generate better cutting tool paths, increasing the overall quality of a finished product (Fountas et al., 2019).

2.3.1 Different CAM strategies – Milling and Drilling - for prismatic parts

2.3.1.1 Milling

The milling process is one of the most used machining processes in the world. In fact, this is a versatile process and allows the manufacturing of products with complex geometry by eliminating material from the workpiece with the help of rotating cutters (Lee et al., 2020). Thus, milling strategies and tools are always in development to fulfill the market needs (Perez et al., 2013). As shown in Figure 7, there are two basic milling operations: end milling and face milling. The first operation is usually required for profiling works and the second operation is used to make flat surfaces of a workpiece. Several authors have done research on this theme. For example, Chen *et al.* (2018) developed an algorithm that automatically determines critical milling parameters such as depth or tool orientation in order to improve the quality of the product and have better finishes on critical features. Moreover, Yue *et al.* (2021) stated that clearance helix angle, feed per tooth, and cutting depth are essential parameters.

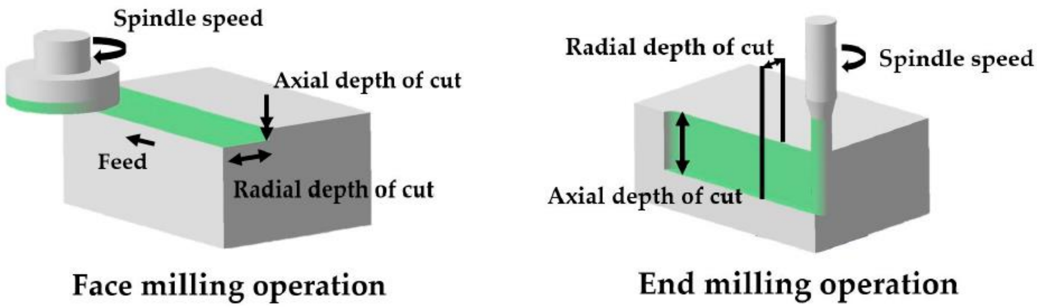

Figure 7 - Milling operations (Lee et al., 2020)

2.3.1.2 Drilling

Drilling is a machining operation destinated to produce holes in the workpiece, by using a drill. This operation has an important role in the overall manufacturing time, due to the fact that usually it is responsible for 25% of the time consumed during the production. Moreover, drilling is often observed as the last operation done in a manufacturing process. Hence, it has a relatively high economic importance (Li et al., 2007; López De Lacalle et al., 2000). As shown by Figure 8 there are three drilling stages. Firstly, the drill point contacts with the workpiece (A). Then, the drill reaches the bottom of the surface, accordingly to the specified length of the hole (B). Lastly, the drill cutting edge is disengaged (C) (Suresh Kumar et al., 2020).

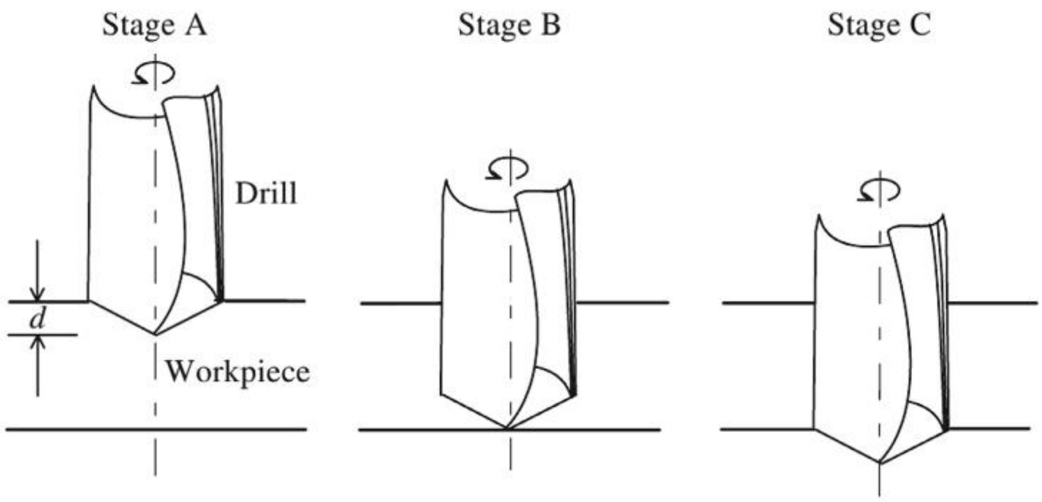

Figure 8 - Drilling stages (Suresh Kumar et al., 2020)

2.4 Feature-based design

As a starting point to this section, it is important to understand the concept of feature. However, there are different definitions. The high number of definitions can be explained by the constant development of this theme over the years (Lei Li et al., 2020). Below, there are some of the definitions:

- "A specific geometric configuration formed on the surface, edge or corner of a workpiece" providing the possibility of changing the design of the aforementioned workpiece (Roller, 1989);
- Chu & Gadh (1996) classify a feature based on how much tools are needed to manufacture by machining that feature;

- On their turn, Yang *et al.* (2008) define feature as a way to storage information about the design or performance about a part or an assembly where the specific feature is included;
- Similarly, Z. Cheng & Ma (2017), consider that a feature represents the geometry of a part or assembly;
- On the other hand, features can carry information to link the design process with the manufacturing process (Dipper et al., 2011).

Furthermore, features can reflect the designer’s intent when designing the 3D model. Thus, feature-based design (FBD) is a core technology for solid modeling (Deja & Siemiatkowski, 2013). In fact, FBD uses shape features as design primitives (Zhang et al., 2017). As a matter of fact, FBD enables smoother part-process integration owing to the fact that this technique provides diversified information about the product (Patil & Pande, 2002). Figure 9 shows a common process when using FBD; a 3D part designed in a CAD software constituted by four features: two holes (with different diameter and length), one slot and one pocket. When designing by feature, this is, using FBD, the designer uses some of the features available in a certain feature library. It starts with a blank workpiece, and after that, the user will insert the desired features with the specified dimensions in order to achieve the final design of the product that will be later machined.

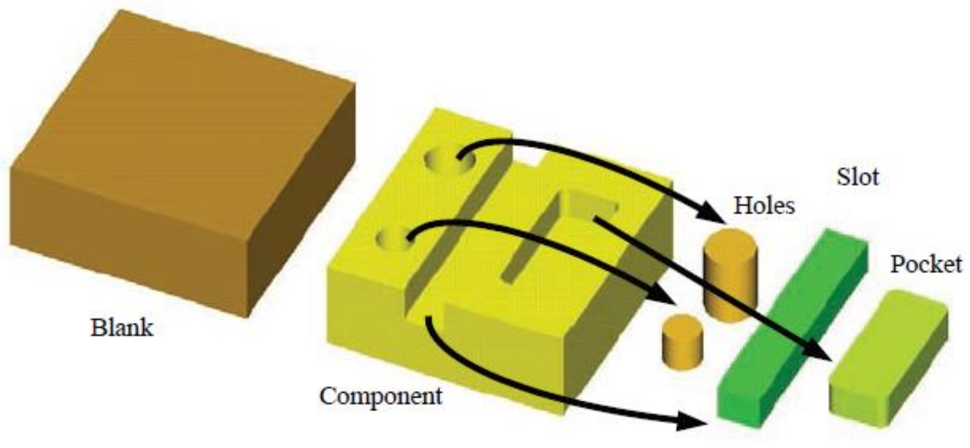

Figure 9 - Workpiece with four different features (Xu, 2009)

FBD has different areas of research (Shahin, 2008) and some of them are represented in Figure 10. The development of strategies' techniques has a lot of relevance in FBD. According to Duan *et al.* (1993), there are three major strategies:

- Concurrent design;
- Design for assembly;
- Parametric design

The first strategy suggests that design and manufacturing are done concurrently. In other words, the process planning of the product does not follow a path with one step at a time, but it does simultaneous steps. Thus, the features used in this strategy are selected considering their machinability. As a consequence, manufacturers can reduce production costs, lead time and, in addition, can increase the automation of the process (Duan et al., 1993; Jong et al., 2009).

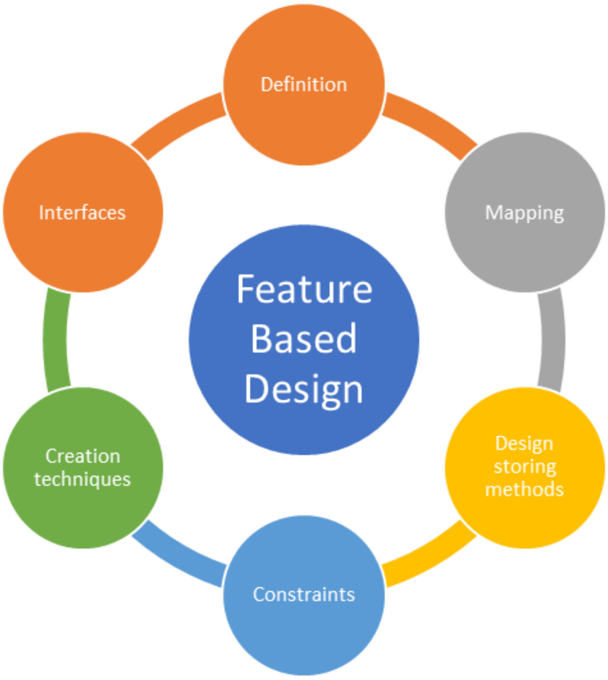

Figure 10 – Feature-based design areas of research (Jong *et al.*, 2009)

With regard to design for assembly, this strategy provides guidelines when designing the part in order to provide an easier and better assembly; thus, a better product. To do this, some information is needed. Starting with component position and orientation, as well as mating features and operations and lastly, component or feature geometry (Molloy et al., 1993). As for parametric design, this methodology aggregates features into parentships based on similarities in design, dimensions or attributes. Hence, the models can be updated based on its parameters, which makes it easier for designers that do not need to manually change each feature when modifying a part or assembly (Z. Cheng & Ma, 2017; Ma et al., 2009). On top of that, this strategy is the most used in the modern CAD systems (X. Li et al., 2012).

Later on this chapter, area of research that focus on constraints will be analyzed in depth.

FBD also has an important role in prismatic machined parts. These prismatic parts constituted by, for example, slots, steps or holes features are widely used in manufacturing context. Several authors have done research on this area and other ones proposed new approaches to FBD for prismatic parts. Patil & Pande (2002) developed an intelligent environment for the FBD and process planning of prismatic parts with the objective of it being produced on CNC machining centers. This intelligent environment comprises two primary modules: feature-based modeler and intelligent process planner. The first module provides the graphical environment for intelligent representation of the part and the second one processes the feature information from the FBM to map into the corresponding machining operations. After testing it, the authors stated that with this methodology it is possible to have a rapid product development since the change from design to manufacture can be done quickly. On their turn, Sunil *et al.* (2010) presented an algorithm that recognizes the interacting machining prismatic features from CAD models. As expected, this approach was able to recognize simple holes as well as stepped holes (such as counter bore holes). They concluded that the algorithm performed well; hence, after recognizing the features, it can be used for an integration with process planning applications such as CAPP (Computer Aided Process Planning). Similarly, Cardone *et al.* (2006), proposed an algorithm for identifying prismatic machined parts, in a database, that are similar to a given part to be machined. According to the authors, the main goal of this methodology is to estimate the cost of machining the new part based on the cost of previously machined parts archived in the database. On the other hand, Pedagopu & Herano (2019) discussed a framework that integrates efficient setup sequencing for machining of 2.5 dimension prismatic parts. The authors considered every hole as a feature. Their goal is to apply this algorithm in automated process planning of these parts. In the same way, Leo Kumar *et al.* (Leo Kumar *et al.*, 2015), developed a method to extract the part feature information from a FBD related to micro prismatic parts. In addition, they integrated feature extraction and decision making on their database. The authors verified that, when using this method, complex feature extraction can be avoided.

Table 1 shows relevant studies in the scope of FBD.

Table 1 - Feature-based design studies

References	Work Description
(Li *et al.*, 2017)	This paper proposed a design system of full parametric association modeling based on geometric features for standard cam. In fact, standard cam consists in holder, slider, and driver. After testing it in a process of stamping dies for automotive panels, the authors concluded that the model allows to improve the quality design, while reducing costs.
(Wang *et al.*, 2012)	This work aimed to reduce costs of new aircraft product development, as well as to reduce lead-time. With that in mind, the authors presented an agent-based collaborative design framework to facilitate the collaboration of feature-based aircraft structural parts design and analysis tools, including cost estimation. For that, FBD, feature-based cost estimation are used and described. The authors concluded

	that, when using this approach, there are improvements on the efficiency and quality of parts designed.
(Qian & Dutta, 2004)	In this paper, the goal was to create a FBD framework to facilitate heterogenous object design. As a starting point, the authors studied the relation between features and material features in these objects. After that, they proposed three synthetized material features and two enabling methods to achieve the goal. It is a constructive design process based on a set of user pre-defined heterogeneous features.
(Samanta & Koc, 2005)	Similarly, Samanta & Koc (2005) proposed a method to design heterogeneous objects. In this study, the authors focused on the features that dictate the material composition of the part. They conclude that by changing the geometric and material features of the object is possible to create variant heterogeneous objects. Thus, this method allows to create complex heterogeneous parts that can have a huge impact in different applications.
(Lee & Lee, 2012)	This study approaches a new FBD that can provide multiresolution of dynamic changing CAD models. In other words, this multiresolution framework suppresses detailed features, providing simplified shapes of parts. As a conclusion, the authors stated that using this methodology it is possible for the designers to carry their task more efficiently due to the fact of fast rendering, for example.
(Jong *et al.*, 2011)	The authors of this paper proposed an integration of feature-based mould design with CAD. This framework organizes similar features by groups while also using existing features to create a detailed cavity design. As a consequence, time savings can go up to 59%, due to the fact that mould design is easier with this method. That is, time is saved, due to the fact that, there is no need of doing the redundant work again between conceptual and detailed mould design.

2.5 Feature-based Manufacturing

The 3D CAD technology is no longer applied only to one stage of the process. In contrast to the early years of CAD, when it was only applied to the design stage, nowadays this 3D software also integrates the process planning stage and the manufacturing stage (J. Liu et al., 2017). The latter will be analyzed on this section.

Feature-based manufacturing (FBM) is utilized for adaptative and distributed manufacturing. In other words, countless manufacturing services, such as cloud manufacturing or internet of things, can power FBM. The use of these services enables smart decisions to be made based on updated and real-time information to check if the requirements are being met (Adamson et al., 2019). Numeric control (NC) is also another important aspect of FBM. In fact, NC contains product geometry, working steps and machining parameters data that are necessary for manufacturing the desired product. Moreover, NC machining provides effective and efficient machining in the context of FBM (Laguionie et al., 2011; W. Wang et al., 2014).

As shown in Figure 11, the core of FBM is based on design features and manufacturing features, namely machining features (Zheng et al., 2012).

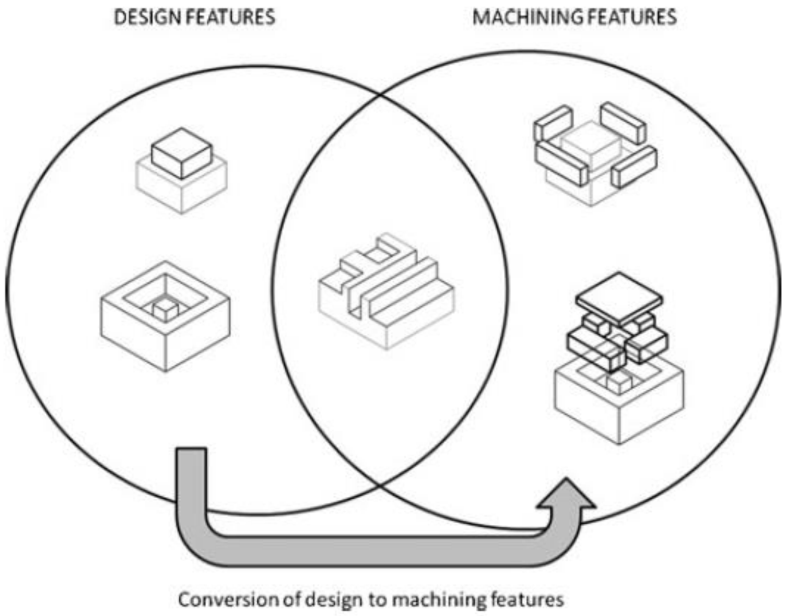

Figure 11 - Design and manufacturing (machining) features (Zheng et al., 2012)

Design features were described in the previous sub-section but to fully understand the concept of FBM is important to realize the definition of manufacturing feature. Due to the fact that the development chapter of this thesis is focused on machining, machining features will have more relevance in the following exposition.

Similarly to the design features, there are several definitions of machining features. Firstly, at the most general approach, machining feature is a feature created by a machining process (Sanfilippo, 2018). Chu *et al.* (2012) reports that machining features can be divided into two categories: geometry-based and manufacturing-based. On the other hand, Liu *et al.* (2017) states that machining features can be classified into three categories (as shown in Figure 12): depression feature, protrusion feature and transition feature.

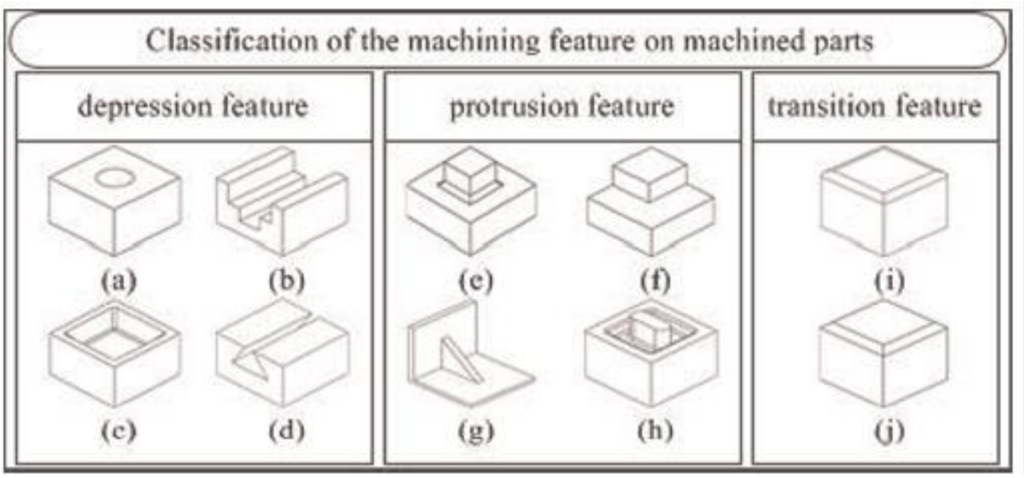

Figure 12 - Categories of machining feature (Liu et al., 2017)

According to the author the different categories can be summarize as following:

- Depression feature – Is formed by using the cut operation in the 3D design software. This is the type of machining feature that will be utilized in this thesis;
- Protrusion feature – Formed by removing the surrounding material;
- Transition feature – Transition area between two surfaces. Examples of this features are fillets and chamfers.

However, some authors, such as Wang *et al.* (1993), do not recognize protrusion feature as a manufacturing/machining feature but as a design feature. This classification is based on the fact that this feature, as previously referred, can only be created by removing surrounding material.

Furthermore, there are various manufacturing features. Figure 13 summarizes all the features (Besharati-Foumani et al., 2020). On their turn Ji *et al.* (2018), tackled the different machining features as closed, semi-closed or open features, as shown in Figure 14. The work developed in this thesis will focus on slots, holes and pockets (open and closed) features.

Lastly, it is relevant to mention an important aspect of FBM, feature recognition. When performing feature recognition, three tasks are done: definition of the features, sorting and extracting features from the model and storage those features. Furthermore, the purpose is to search for stored information and compare that with the features contained in the part (Babic et al., 2008; Dodok et al., 2017). Recognizing features is a key point to trigger the technology to identify a machining process (Komatsu & Nakamoto, 2021).

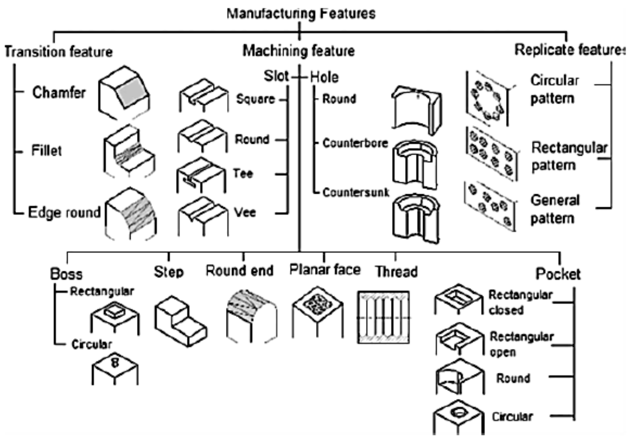

Figure 13 - Manufacturing features (Besharati-Foumani et al., 2020)

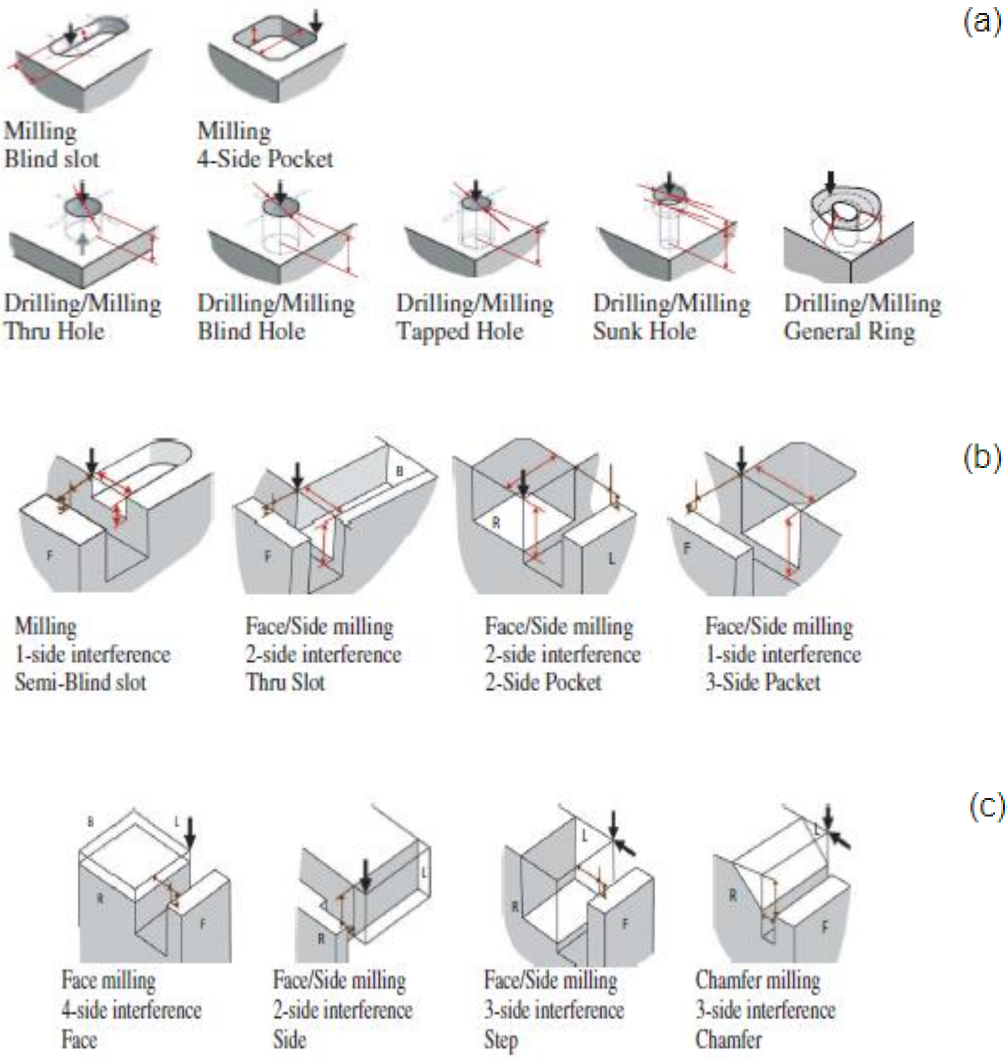

Figure 14 - Different machining features – (a) Closed machining features; (b) – Semi-closed machining features; (c) – open machining features (Ji et al., 2018)

Table 2 summarizes research articles whose subject is FBM.

Table 2 - Feature-based manufacturing studies

References	Work description
(Volkmann *et al*., 2020)	The authors of this study proposed a structure based on Open Platform Communications Unified Architecture (OPC UA). The approach was tested with a drilling process and milling of a pocket. In addition, a manufacturing code was stored in a robot. After that, the manufacturing parameters were determined and transferred to that robot with the help of CAD. As a result of this approach, the authors verified that there is a considerable time saving. Moreover, the manufacture of small batches with economic profit might become possible in the near future.
(Nagarajan & Reddy, 2010)	The authors proposed a methodology based on a STEP-based platform-independent for design and manufacturing feature recognition. The system relies on the access direction of the features. The necessary volumes to be removed as well as the access directions for the machining process are determined by this manufacturing feature recognition. In conclusion, they stated that the created system can handle situations where machining is done after other manufacturing processes. Furthermore, intersecting features are recognized which did not happen with previous studies.
(Han *et al.,* 2001)	Similarly, this study was based on manufacturing feature recognition. The goal of the authors was to create a framework that can integrate CAD and CAM, something that is not fully accomplished. In addition, this integration can impact the final cost of the machining process.
(Adamson *et al.,* 2017)	This paper focused on the development of a framework of FBM for adaptative equipment control and resource task machining in collaborative manufacturing environments. To begin with, the authors stated that one of the key pillars are the manufacturing features. They affirmed that the required control for these environments can be created at run-time. Although time saved when

	generating control instructions is hard to calculate, it is possible to say that time is significant.
(Jong *et al.,* 2014)	The aim of this study was to create a methodology to convert part design into manufacturing features. They used a feature recognition strategy and showed that planning time can be reduced by 87%.
(Deja & Siemiatkowski, 2013)	This paper approaches a framework to deal with dynamic operation conditions of manufacturing facilities. This framework was tested on a machining process of a complex prismatic part. As a conclusion, the authors stated that the methodology used can identify the most effective machining process design, taking into account the real-time production conditions.

2.6 Constraint modelling

When connecting the design stage to the manufacturing stage is important to have in mind that FBD has constraints (Shi et al., 2018). Constraints can be expressed as a concept that, in order to enable the final product, allow the combination of the values of the features. According to Ovtcharova & Jasnoch (1994), there are two types of constraints: violable and non-violable.

- Violable constraints: These constraints are often related to the design process. Since the design is changeable, they can be changed accordingly with the design.
- Non-viable constraints: On the contrary, these constraints cannot be violated, on account of elementary laws. Eventually, this would lead to an impossible situation.

In addition to constraints related to design, there are assembly constraints (Z. Li et al., 2020) as well as some related to machining operations (Battaïa et al., 2017). According to Frisch *et al* (2005), the modelling of constraints, or constraint modelling, is needed to solve problems related to this concept. Furthermore, constraint-based models open the possibility of addition of design semantics during the different steps of product design (Bodein et al., 2014). In fact, several authors used constraint modelling on their research. Kardos *et al.* (2020), proposed a constraint model to solve assembly planning on a generic feature-based representation of a product. This framework has feedback from both macro and micro-level planning. After testing it on different industries, the authors found the model useful and efficient. Yin & Ma (2012) discussed a method for modelling parametric constraints among different features. On their turn, Liu *et al.* (2017) presented a federate optimization method on five-axis parametric tool paths with preset multi-constraints. As a conclusion, reasonable results related to planning results and computing efficiency were shown.

2.7 Automated machining operations

The availability of machining processes is constantly increasing as a result of the growth of global competition. To face that and to be more competitive, companies are adapting their machining systems. As a matter of fact, their goal is to automate the countless machining operations needed. Therefore, the optimization of machining operations brings high quality products with less costs. Furthermore, the production time is shortened (Núñez et al., 2011; Tolouei-Rad, 2011).

Over the last few years, several methodologies and frameworks to automate machining operations and processes have been proposed to achieve what is exposed in the last paragraph. To begin with, Shojaeipour (2010) implemented a digital camera on the Z-axis of a CNC machine with the objective of taking images of the workpiece. These images, posteriorly, trigger the CNC machine tool to, automatically, decide the best tool path to follow during the machining operation. On their turn, Date et al. (2009) studied a way to automate the NC code generation of a machining operation that can be done on a 3-axis mill-turn center. Initially, they proposed a system that can determine how much diameter can be reduced from the workpiece by only using a turning operation. After that, the process plan for this operation is automatically done. Then, the process planning for roughing profile is done automatically. Lastly, the system checks if the tool will interfere with the workpiece during the machining operation. The authors acknowledge that the proposed methodology allows rapid generation and correctness of process planning as well as efficiency in the process. Villanueva et al. (2021) described a fully automated robotic cell for the machining of cross-laminated timber. They verified that the system has a good efficiency and utilization rates. In the meantime, Shipulin et al. (2017) designed an automated solution for selecting abrasive tools for machining, considering the high speed of machining processes. They observed that the high speed of machining is enhanced while reducing production preparation. In other paper, Zahid et al. (2014) discussed a method to optimize roughing operations in CNC machining. This method consists in enabling the automation of the generation of process planning. This system can be integrated with CAD to enhance a fully automatic process, achieving the goal of rapid manufacturing. The article submitted by He et al. (2018) includes a model to automate a machining line with the goal of minimizing the machining cycle time on that line. Lastly, Mei et al. (2021) developed an automated machining system to, among others, helically mill fastener holes of an aircraft. The use of this system demonstrated a reduction in the number of errors. Moreover, the system can fulfill the demand.

Several authors approached the automation and optimization of toolpaths that highly benefits the companies. These automations are being more used in the framework of Industry 4.0. Dittrich *et al.* (2019) developed a self-optimizing process planning approach for 5-axis milling. This system allows an automatic compensation for tool deflection. By using a machine learning strategy, the created model acquires knowledge about machining tools and material removal processes. This approach was tested in two pockets, as it can be seen in Figure 15. When using this machine learning method, the resulting shape error can be predicted. Therefore, the toolpath can be adapted automatically. Moreover, the authors realized that this model could predict shape deviations.

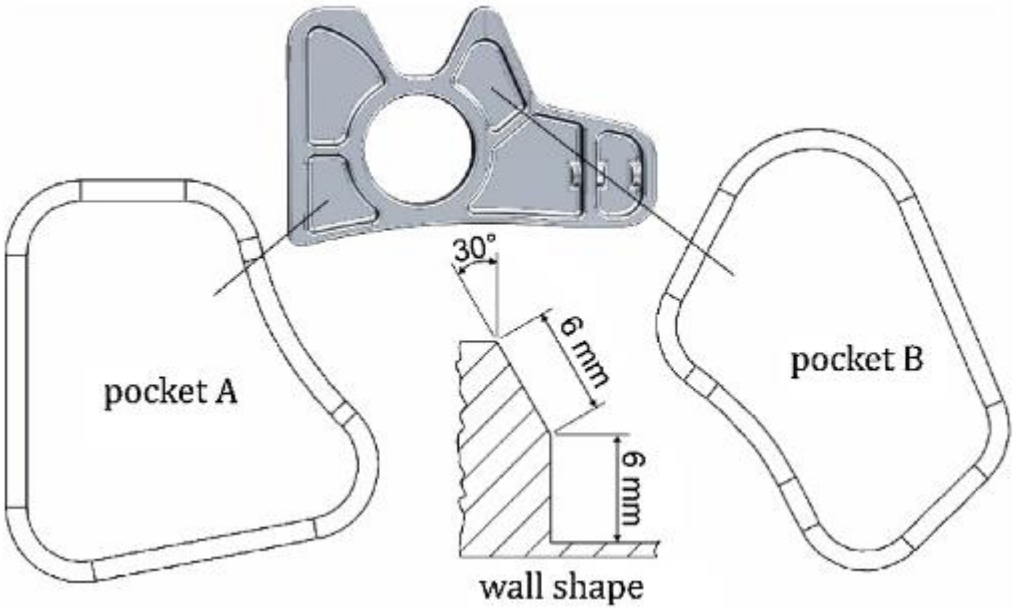

Figure 15 – Pocket test performed by Dittrich *et al.* (2019)

Similarly, Möhring *et al.* (2020) discussed the self-optimization machining systems. This idea is represented in the Figure 16. These systems provide *a priori* information and soft-sensory data that could not be obtained otherwise. The authors concluded that self-optimization machining systems provide real-time monitoring, model estimation and machining toolpaths optimization. Moreover, it allows the measure and improvement of parts produced in a virtual machine environment. Lastly, using machine learning, it is possible to provide a wealth information that can support data-driven prediction.

The self-optimization of machining process plans reduces ramp-up times (Denkena et al., 2021). An approach to automate the production and post-processing, as well as to fulfill the high surface requirements was taken by the authors, as shown in f. For this purpose, a digital process chain and a learning roughness model based were developed. Although the polishing process itself is characterized by several influencing parameters, a prediction accuracy of 83% could be achieved. Thus, an iterative procedure allows to adapt the polishing process in a self-optimizing manner.

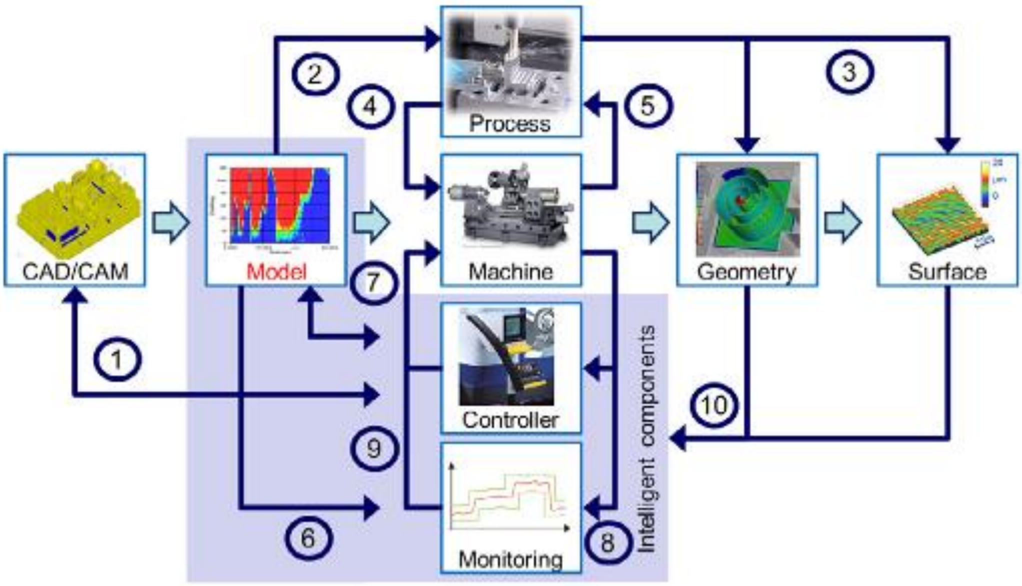

Figure 16 - Self-optimization machining system (Möhring *et al.,* 2020)

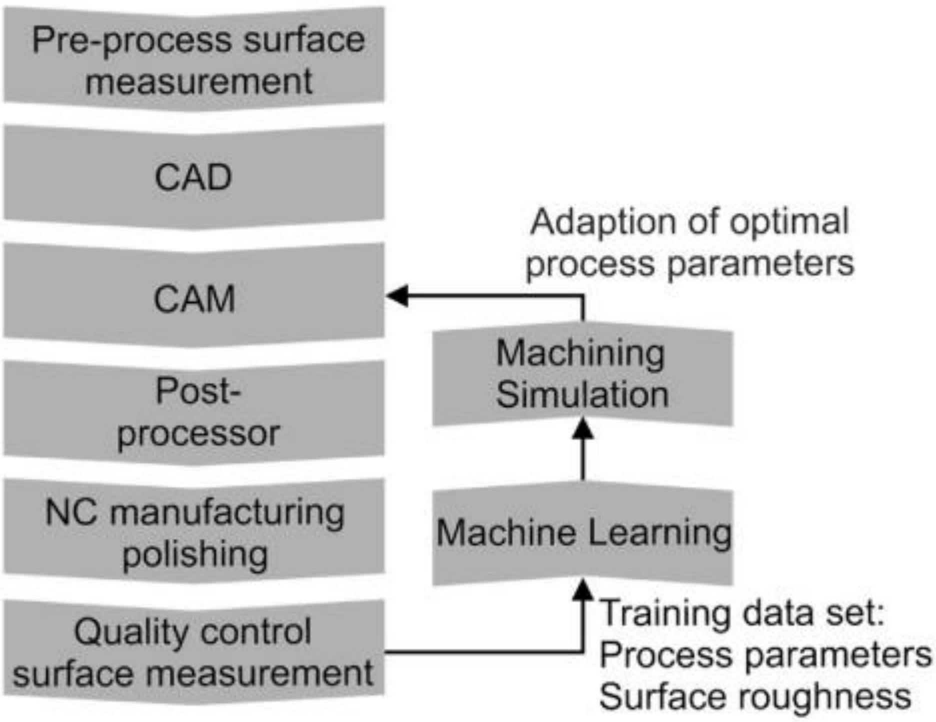

Figure 17 - Method for self-optimizing adaption of process parameters (Denkena et al., 2021)

PRACTICAL WORK

3.1 Algorithm development

3.2 Algorithm programming

3.3 User Interface

3.4 Results and critical analysis

3 PRACTICAL WORK

This section presents the development of this dissertation. Based on what was analyzed in the literature review, it is possible to conclude that several authors have done research and developed methodologies regarding FBD and FBM and have also made efforts to automate the different machining operations. Moreover, some of them investigated the benefits of integrating CAD and CAM software. Nevertheless, there is a gap on this research. Such gap owes to the fact that there is not a system developed that does automatically every step starting in the design stage until the manufacturing stage. Thus, this dissertation will try to fulfill this gap.

3.1 Algorithm development

A machined part can have different geometries. As a starting point to the machining process, a workpiece is needed. Then, the stage of removing material begins. This material removal depends on the coordinates written in the G-Code, which, in turn, depends on the designer's intention. However, sometimes the designer perspective cannot be reproduced in the final product, due to the fact that the process has several stages. As a consequence, the machine operator may not type the exactly coordinates needed to machine the workpiece.

Hence, the algorithm proposed will have various components. Firstly, it is important to note that each stage depends on the conclusion of the previous one. As per normal in standard product development, the first stage of the product development is focused on the design. Therefore, the algorithm development starts at this stage. In order to fully automate the program, the designer will not draw the product on the 3D CAD program. Instead, the designer will add features that together will form the final part. To accomplish the first goal of the program, a library of features is created. The program will have limited features that together can provide several types of creations. Furthermore, the designer will only be able to use the created features. These features represent the volume of material that the cutting tool removes when cutting on a machining center. At this point, it is necessary to choose the features that will be developed and will integrate the features' library. The features chosen are:

- Hole;
- Slot;
- Pocket.

They will be further analyzed on this thesis.

Afterwards, it is necessary to give the tools needed to the designer in order to choose the different features available and place them on a desired place in the workpiece. Moreover, an important and obligatory aspect is that the designer can choose all the measures related to the selected machining feature. These features will subtract volume from the original workpiece. As a result, a process plan is constructed interactively by the user.

The program also defines, based on the initially given dimensions, the cutting tools required for the machining process. Lastly, the toolpaths are defined in a parametric way in G-code.

3.2 Algorithm programming

This section describes in detail how the algorithm was developed and, in addition, how it works. The present work was developed in Visual Basic for Applications (VBA), which is integrated with the 3D CAD program SolidWorks.

3.2.1 Library of features

The first step of the algorithm consists in creating the features that will be available on the library of features. As shown in Figure 18, the first feature programed is a hole.

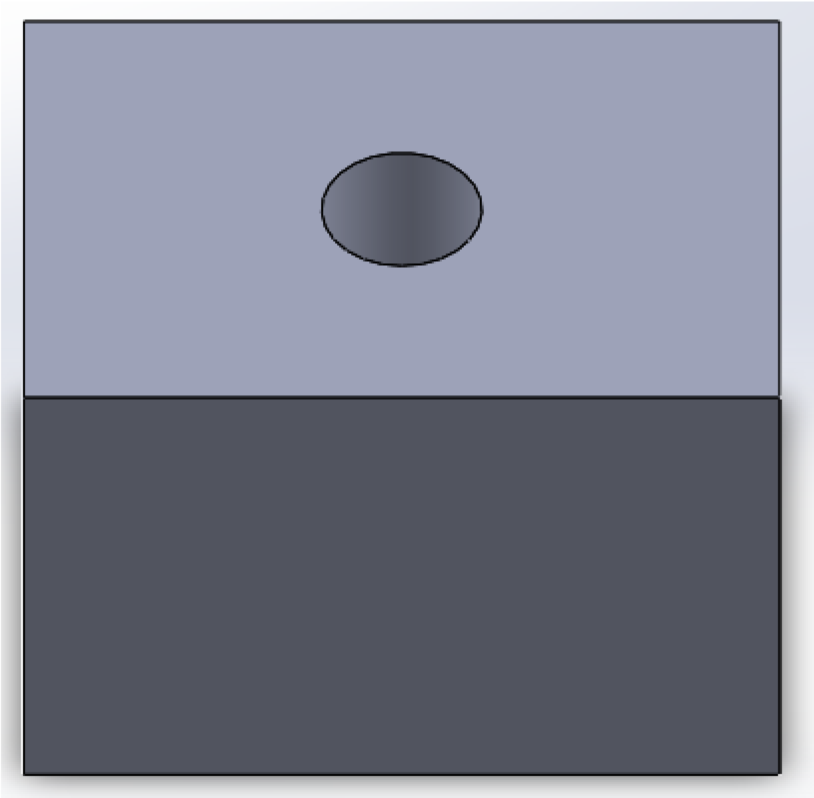

Figure 18 - Hole feature

Firstly, the program checks if the user has a SolidWorks part open. After confirmation that there is a part open, the program will proceed. Afterwards, an interface for the user to enter the input needed for the program will appear. This user interface will be analyzed in depth on the subchapter 3.3. To locate the hole in the workpiece, the center coordinates will be asked. Note that the program assumes that all of this will be done only in the XY plan, due to the fact that the simulator will be using a 3-axis CNC machine. However, due to the VBA functions, the Z coordinate is also needed. Moreover, the hole diameter and depth are obviously needed.

Using the SelectByRay function, the algorithm can open a sketch, followed by a point on that sketch and that will enable the start of the hole design.

As a side note, a hole is not only a cylinder. The bottom part of the hole is a cone. Therefore, this feature will be designed using vertical lines, as shown in Figure 19.

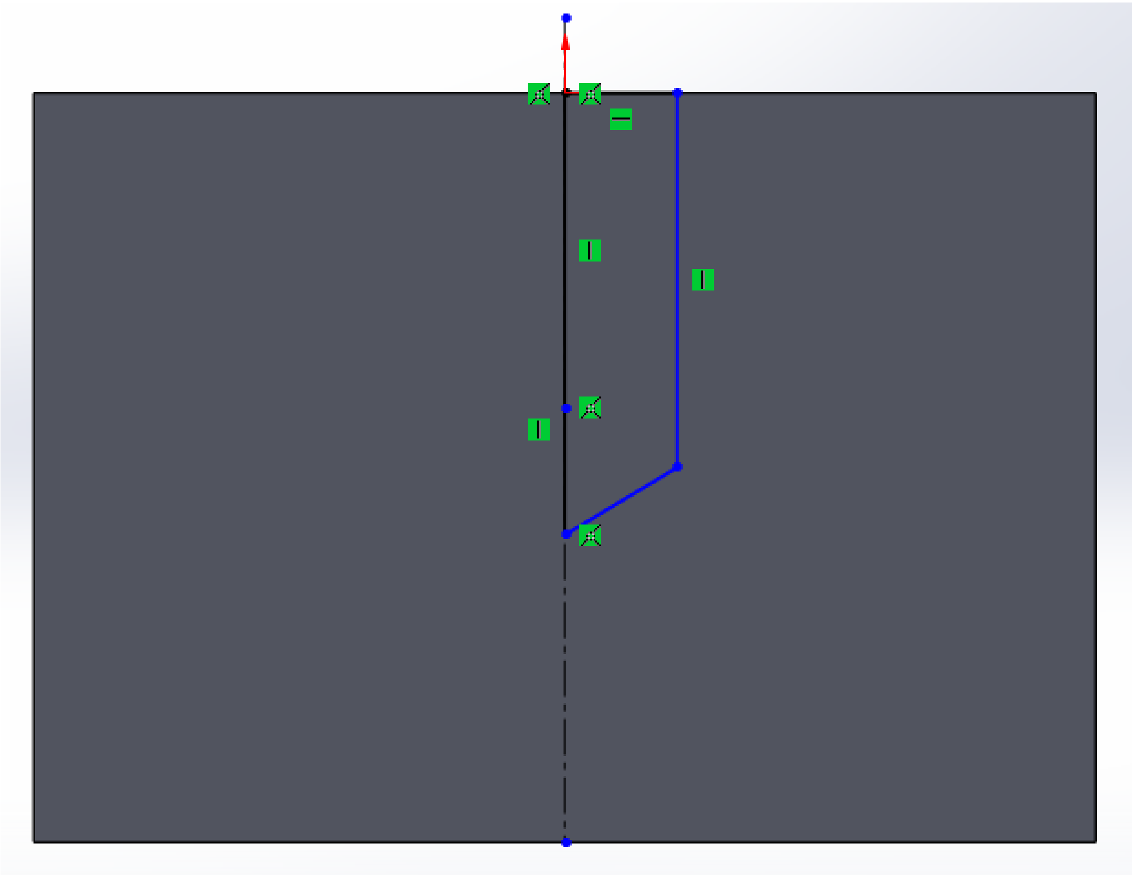

Figure 19 - Hole feature's sketch

To start with, a centerline with the help of CreateCenterLine function is drawn with 5 mm more than the hole depth. Then, all the other lines are drawn using the CreateLine function. The horizontal line has the length of the hole radius. Employing mathematical functions, the other lines will be created. The angle between the right vertical line and the diagonal one is 118 °. This value corresponds to the standard value.

Finished the sketch stage, the FeatureRevolve2 function is used to proceed with the feature cut. The centerline drawn previously is used as the axis of revolution of the hole. In Figure 20, a section view of the hole feature can be seen.

Some constraints were introduced, and they are verified as the program runs. If the hole length is greater than 8 times the diameter then, the user will be warned that the desired dimensions define a deep hole. However, a deep hole can only be done with the use of a special machine. Consequently, the program stops if that condition is not fulfilled. Another constraint implemented is the relation between the hole depth and the workpiece depth. If the first is greater than the second, then the program will automatically stop, and the hole will not be done.

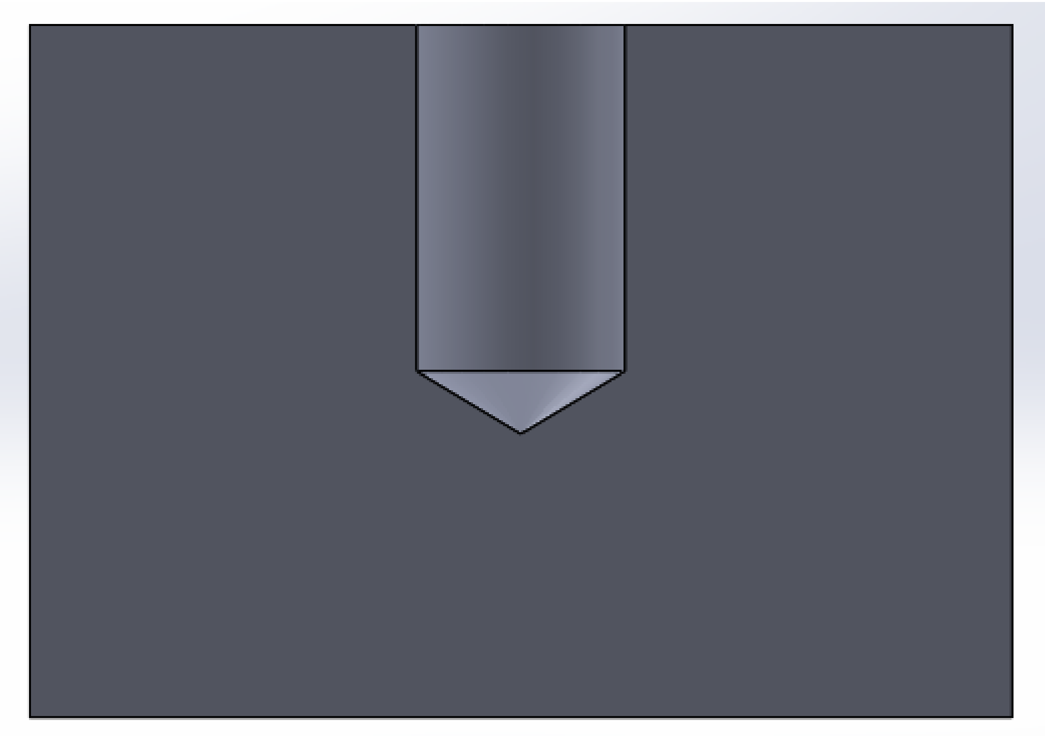

Figure 20 - Section view of the hole feature

The next feature added to the library of features is a pocket, as shown in Figure 21.

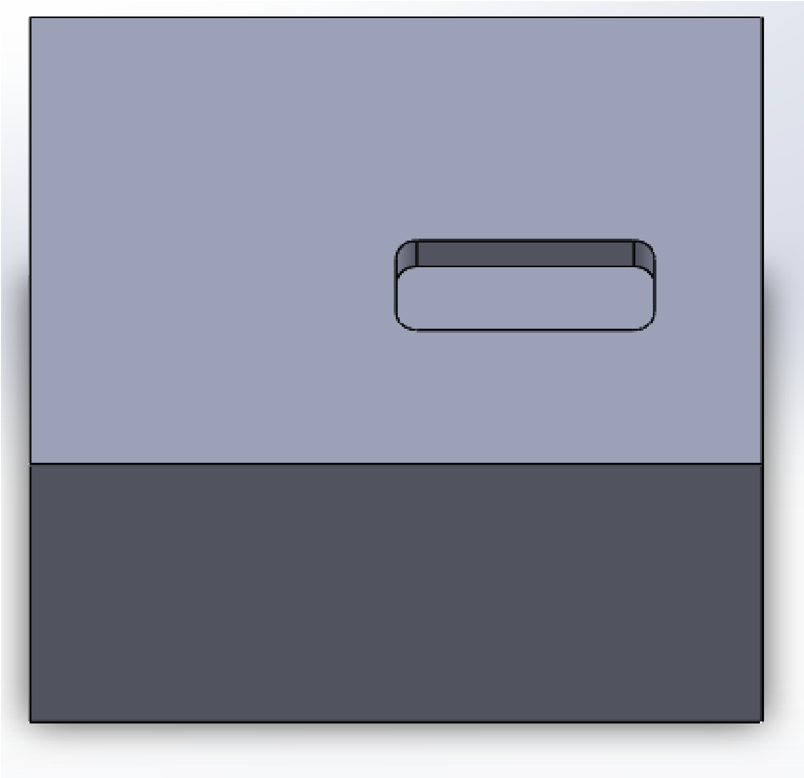

Figure 21 - Pocket feature

In a similar way to the hole feature, pocket feature's algorithm starts by checking whether there is a SolidWorks part open. Afterwards, the user is asked to insert the coordinates. In the case of this feature, the coordinates needed correspond to a corner of the pocket. Then, the other measures needed to draw and cut extrude the pocket, namely, length, width and depth have to be introduced. In addition, the radius is required, due to the fact that in the CAD software it is necessary to draw fillets. Thus, the final product and the design stage will have the same shape. Otherwise, a pocket with square corners would be represented in the design software and a pocket with curved radii corners would be done in the CAM software. This detail is a consequence of all of the exiting cutting tools having a cylindrical shape. Therefore, a radius will be created when cutting a pocket.

On account of using the SelectByID2 function, the algorithm will automatically open a sketch on XY plane based on the coordinates given by the designer. Then, based on the length and width, and utilizing the CreateLine function, four lines (two horizontal and two vertical) are drawn forming a rectangle. Besides that, the CreateFillet function is used four times, each time to create a curved corner. Lastly, FeatureCut4 function will perform the extrude cut, hence finalizing this feature, that can be seen in the workpiece.

Once again, constraints are taken into account. Here, the program will check if the radius utilized to do the fillet in each corner is two times greater than the length, two times greater than the width or two times greater than the depth of the pocket. As long as none of these conditions is assured, the program will continue.

Although is just one feature, it can provide different shapes to the workpiece. This feature can be used as:

- Closed pocket
    - Rectangular pocket with length higher than the width;
    - Rectangular pocket with width higher than the length;
    - Square pocket;
- Open pocket.

Thus, as synthesized in this list, there are various types of pockets available when running the program, thus, enabling different designs.

Lastly, as expressed in Figure 22, a slot feature is added to the library of features.

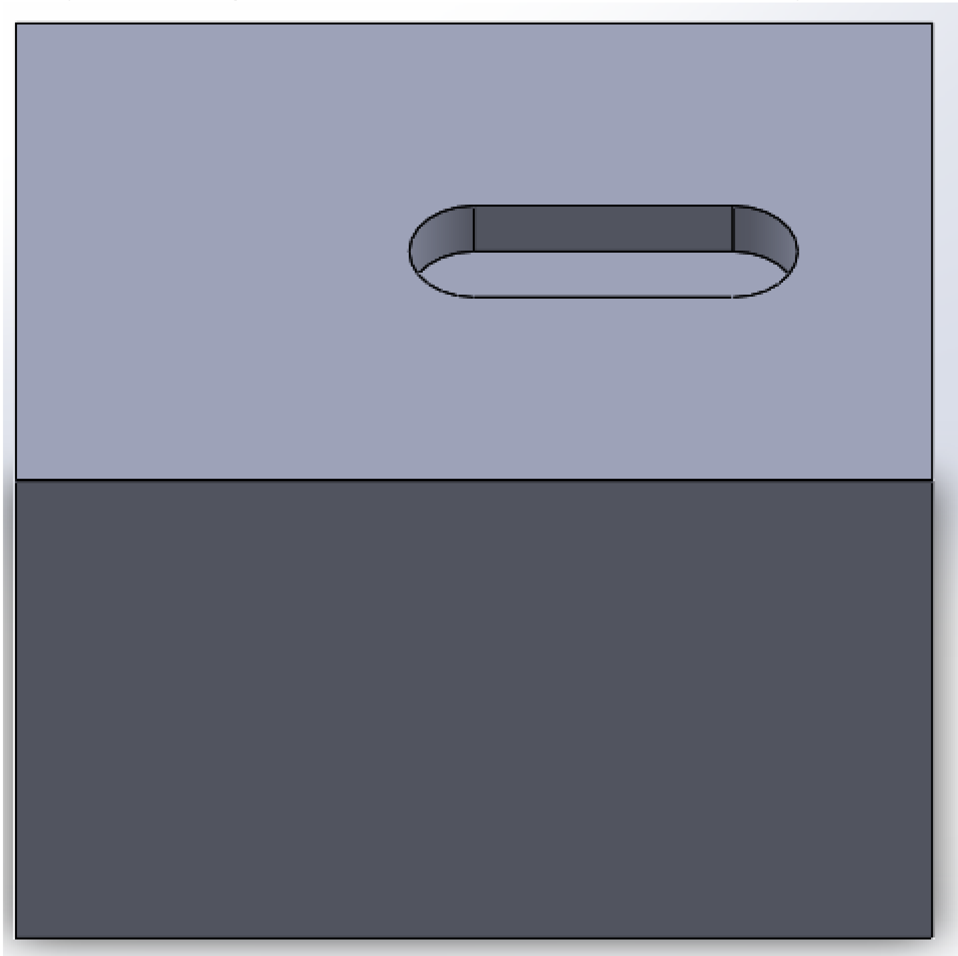

Figure 22 - Slot feature

Following the method used in the other features, here the program will also start by checking if there is a SolidWorks part file open to run the macro. To apply this feature to the workpiece, the designer will have to insert six coordinates. These coordinates are two points with three coordinates for each one. The explanation of the coordinates needed will be further analyzed in the subchapter 3.3. The width and depth measures are also needed to the algorithm. In contrast to the other features, here there is not the need of using several CreateLine functions, due to the fact that the VBA has a specific function for slots. Thus, the CreateSketchSlot function is used. Here, the six coordinates and the width are needed. After the sketch stage done, the extrude cut is done. For that, the depth input will be used in the known FeatureCut4 function.

On top of that, and as seen in the previous feature, this one can also provide different designs to include in the workpiece. Namely, the slot can be used in a vertical or horizonal way, as well as in the diagonal. The use of different numbers in the extensive parameters needed can also provide alternative types of design for the final product done by machining.

3.2.2 Toolpath creation

When the user chooses a feature, the G-code will automatically be created. Thus, on each feature code, there are lines destinated to the generation of the toolpath. When programming, it was considered that per default all of the text files with the G-code would be saved on the user's desktop. However, that definition can be changed easily.

After that, there is a table created using arrays, included in each feature's macro, that has all of the tools available to do the machining operations needed. In this case, the tools presented in the table correspond to the tools created on the simulator software that will be described later on. In the above-mentioned table, for each tool there are six parameters specified. Firstly, one of the parameters is the tool ID that is a very important setting, since it will be specified directly in the G-code; hence, it will be used by the CNC machine. Secondly, the stock material in which the tool can be used. In this dissertation only one material will be considered. Subsequently, the recommended feed and recommend speed for each tool is also specified in the table. The last three columns refer to the tool properties, namely, tool diameter, tool cutting length and tool material. Following the main goal of this dissertation, in a fully automatic system, the user will not select the tool needed for a specified machining operation manually. Thus, some lines were written to address this situation. Two "For" loops were created, so the program will check the user input and will match with the tools presented on the table, choosing the most suitable for the operation needed. Below, in Figure 23, there is an example of those "for" loops.

```
For i = 1 To 6
    If DrillTable(i, 4) = (Diameter * 1000) Then
        If DrillTable(i, 1) = "Steel" And DrillTable(i, 6) = "Cobalt" Then
            ofile.writeline "T0" & DrillTable(i, 0) & "0" & DrillTable(i, 0)
            DrillId = DrillTable(i, 0)
        End If
    End If
Next i
```

Figure 23 - Example of a "For" loop

After the tool selection stage completed, and with the file created, it is time for the algorithm to start writing the code on that file. For that, the coordinates stored in the algorithm “memory” will be used.

The first things needed to be print in G-code, in all programs, are the initial processes that are needed for a CNC machine tool to run. These basic procedures include the movement of the machine to a point in the workpiece, enabling the tool coolant, selecting the tool, as well as providing the suitable cutting speed and cutting feed.

Applying to this dissertation context, the first thing to be printed is the G21 and G90 cycle. Respectively, they mean that the program will be using the metric system as well as absolute coordinates. After that, the cutting tool needed is selected. As previously seen, the tool chosen is related to the user input. To perform this action, the code M06 is needed. Afterwards, the program will indicate the zero point. Here, there are two ways to do that. Both cycles, G92 and G54, can be used. In the first one, the zero-point coordinates are specified in the code. On the other hand, when utilizing G54, there is no need, in fact it is impossible, to write the coordinates after the command. The CNC machine will automatically use the zero point already given to the machine. Moreover, and given that the program will run on the XY plane, the G17 cycle has to be used. The last command, relative to a basic procedure, that is required, is the coolant activation. That is done by using M08.

At this stage, the common code ends, and the code developed for each feature starts to be different.

Starting with the hole feature, the program will print in the text file a sequence of commands that will enable the drill. Starting with the G00, the tool will be rapidly moved to the X and Y position that represent the hole center. However, the Z coordinate will be above the plane. Then, there will be a rapid movement of the tool only in the Z coordinate to a plane created in the code based on the given coordinates. Afterwards, the drilling starts, using the G81 cycle. Here, five parameters need to be specified. The X and Y center coordinate, the Z coordinate that represents the bottom of the hole, as well as the R coordinate (the Z plane where the drill starts), and the recommended feed rate. Note that, this feed rate is taken from the initial table. After that, there will be another G00 cycle, where the tool will rapidly go back to the last position. As a result, the hole is done, and the program can stop. For that, the M09 cycle will be used to stop the coolant. Moreover, the M05 will shut the spindle speed and the M30 will stop the program.

As the hole feature program is completed, it is time to move on to the pocket feature program. As said before, the beginning of the program is the same to all features; hence, the explanation of this code will start on the stage of the material removal. As shown in Figure 24, the material will be removed following a zig-zag strategy.

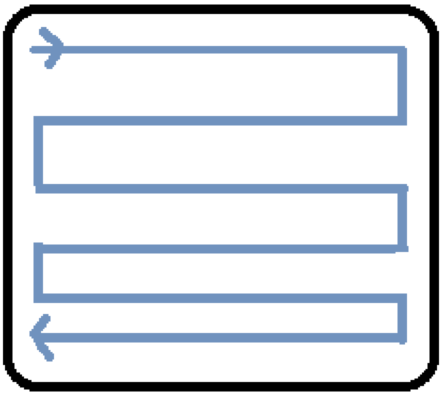

Figure 24 - Zig zag strategy

Here, the main cycles used will be the G00 and G01. Thus, as a starting point, the G00 cycle will be used to rapidly move the cutting tool to the corner of the pocket. When the tool is on the desired plan, once again according to the user input, the G01 cycle will be used. A linear cut from the starting point until the corner will be done. This situation will continue until the cut reaches the final wall of the pocket. This is verified by using a “For” cycle. This cycle will constantly compare the cutting tool coordinates with the user input and will tell the program when to stop. Another important aspect is the cutting depth. Note that, in case of open pockets, the “For” cycle will understand and an error will not be shown. If the depth is too high, the cutting tool will not cut all in a single time. To address that issue, there is another cycle “For” created (Figure 25) that will check the depth given by the designer and calculate how many times the cutting tool will perform its action and how many stock material will cut per pass.

```
If Depth Mod 2 = 0 And Depth < 12 Then
    dpc = 2
Else
    If Depth Mod 2 = 0 And Depth > 12 Then
        dpc = 4
    Else
        If Depth Mod 2 <> 0 And Depth > 1 Then
            dpc = 3
        Else
           dpc = 1
        End If
    End If
End If
```

Figure 25 - "For" cycle for calculating the depth

Lastly, when the roughing part is completed, the CNC machine, based on the code automatically programed, will do the finishing pass around the walls of the pocket. The program ends as previously stated.

Finally, there is the slot feature G-code. As shown in Figure 26, the strategy will be different. Here, the contour strategy will be used.

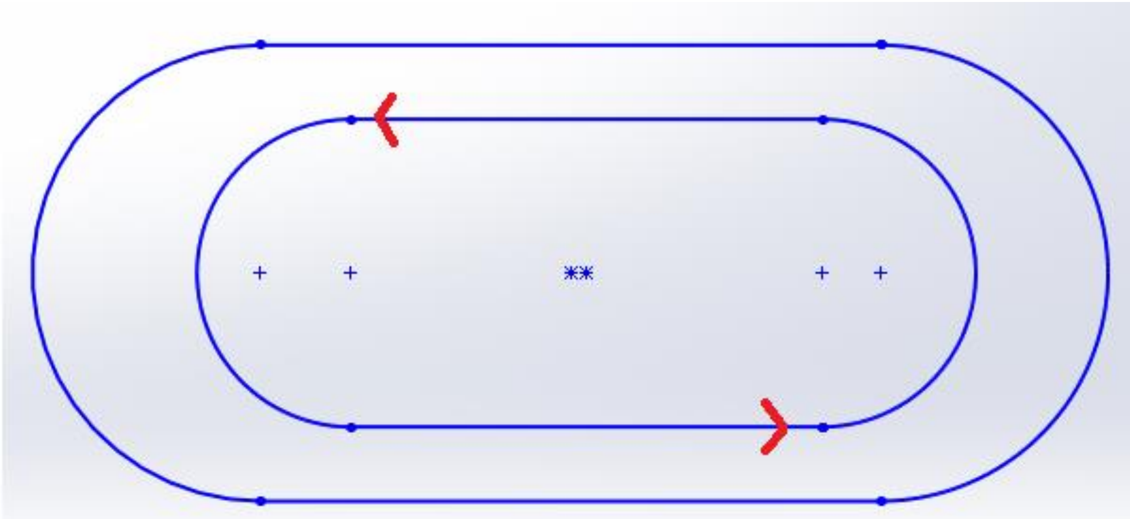

Figure 26 - Contour strategy

With this method, one of the “For” cycles used in the previous feature will not be needed. Hence, the code is easier to understand. The G00, G01, G02 and G03 are the cycles needed to perform the slot machining operation, starting by moving the cutting tool rapidly to the correct position, with the G00 cycle. Then, the G01, G02 and G03 cycles will be used to form a path similar to the highlighted in Figure 26. As in the last feature, a “For” cycle is needed to check the depth specified by the user. Otherwise, the program could not determine the end of the pocket. Then, an “If” cycle inside the “for” cycle is used to check the constraints relatively to the wall. As expected, the code will end as the other ones.

The code from the design stage to the toolpath stage of all features is described in annex 1.

3.3 User Interface

A key aspect of this work is the user interface. Here, all the input needed is provided by the designer. After that, the program will do everything automatically. Figure 27 shows the hole feature's user interface.

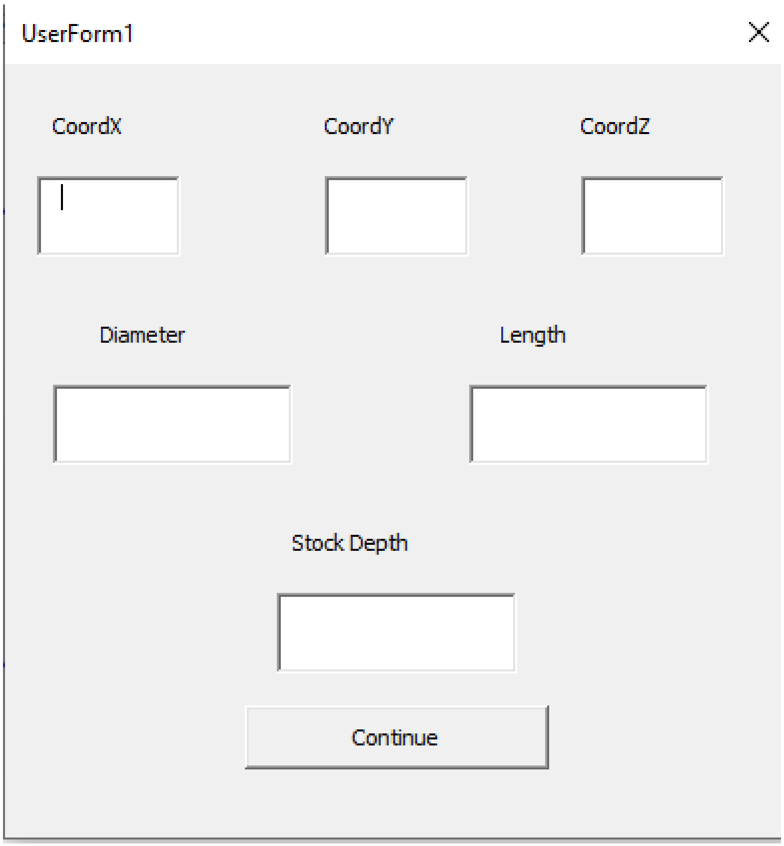

Figure 27 - Hole feature's user interface

The user will be asked to enter 6 different parameters. The CoordX, CoordY and CoordZ parameters, represent the X, Y and Z coordinates of the center of the hole (top part). As indicated by the name, the diameter and length of the hole are also asked. Lastly, the user should also indicate the stock depth. This parameter is related with the depth constraint. Thus, the program will be able to compare the stock depth with the hole depth. All the measure parameters are given in millimeters. After clicking in the continue button, the program will run and a file text with the G-code will appear on the user's desktop.

Looking at Figure 28 the pocket user interface can be seen.

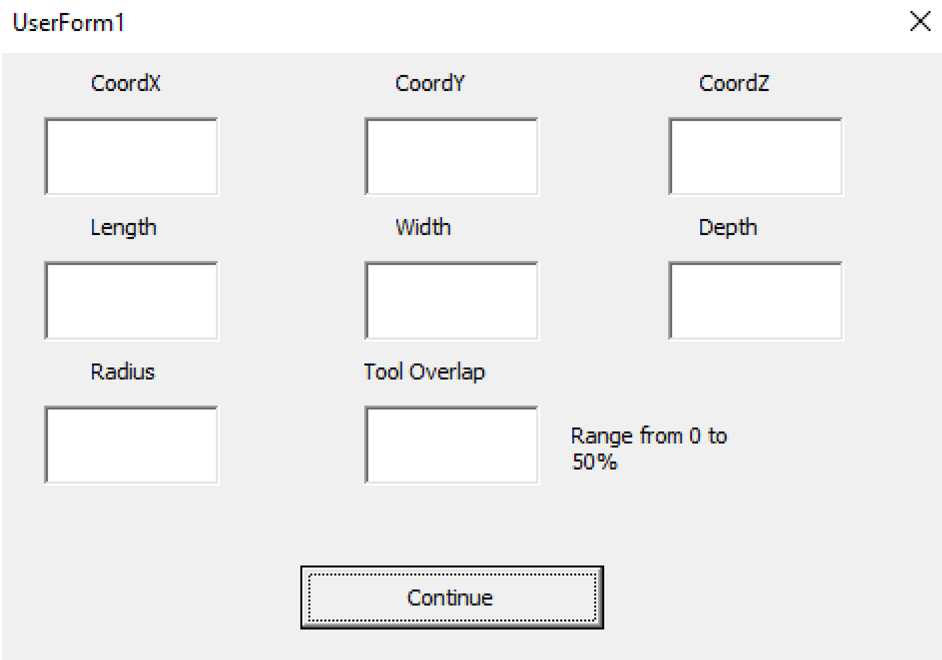

Figure 28 - Pocket feature's user interface

As can be seen, there are more parameters here than the existing ones in the hole feature. The CoordX, CoordY and CoordZ mean, once again, the X, Y and Z coordinates. Unlike the previous feature, these coordinates are not from the center of the feature. In order to explain the values needed for this feature, a pocket was drawn, as shown in Figure 29. Here, the yellow arrow points to the top right corner of the pocket. The different coordinates asked in the user form define this corner. Every pocket built using this macro will start in this corner. The length, width and depth are also asked. Also, in Figure 29, the length is represented by the blue line and the width by the green line. Moreover, in order to design the pocket in the CAD software, the radius parameter is required, to draw the fillets in each corner. Thus, the user form created requires the designer to specify this dimension when drawing this feature. Finally, the tool overlap parameter is required to enable the program to calculate the percentage of overlapping when using the cutting tool in a zig-zag strategy.

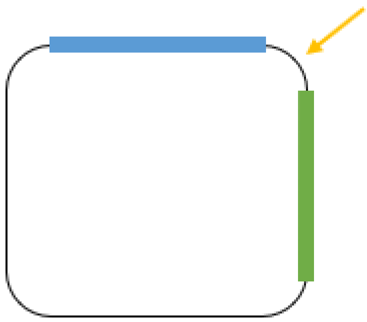

Figure 29 - Pocket

Lastly, the slot’s user interface is presented, as shown in Figure 30.

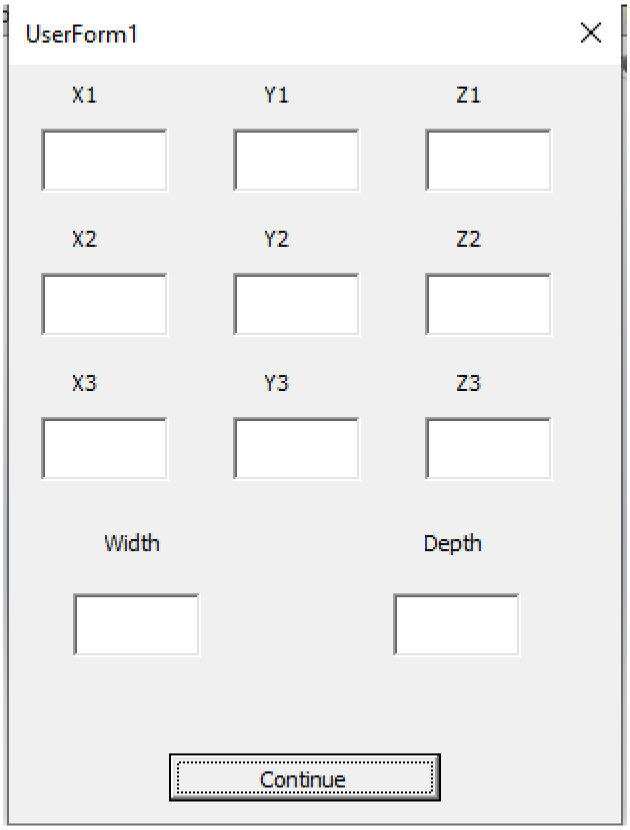

Figure 30 - Slot's user interface

To design and further develop the toolpath of this feature, besides the width and the depth, the program also needs the introduction of 2 X coordinates, 2 Y coordinates and 2 Z coordinates by the user. These coordinates are represented by the two points connected by the center line in the middle of the slot, as shown in Figure 31. Basically, they are the center of each circumference arc in both extremities of the slot.

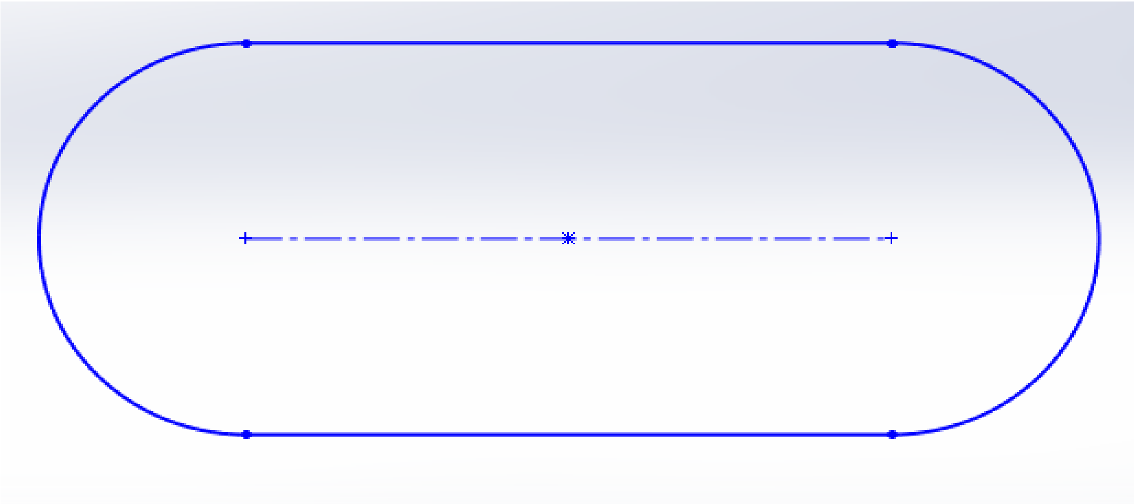

Figure 31 - Slot coordinate points

3.4 Results and critical analysis

In this section, an application of the previous written algorithm is presented. For that, a specific object is designed, following the needed instructions, specified by the user. Furthermore, a critical analysis is made. To do the simulation, the CNCSimulatorPro software was used.

3.4.1 Indicative CNC programming case

Firstly, a workpiece with 200 x 200 x 60 mm^3 was created, as shown in Figure 32. These dimensions were chosen based on the CNC machine utilized.

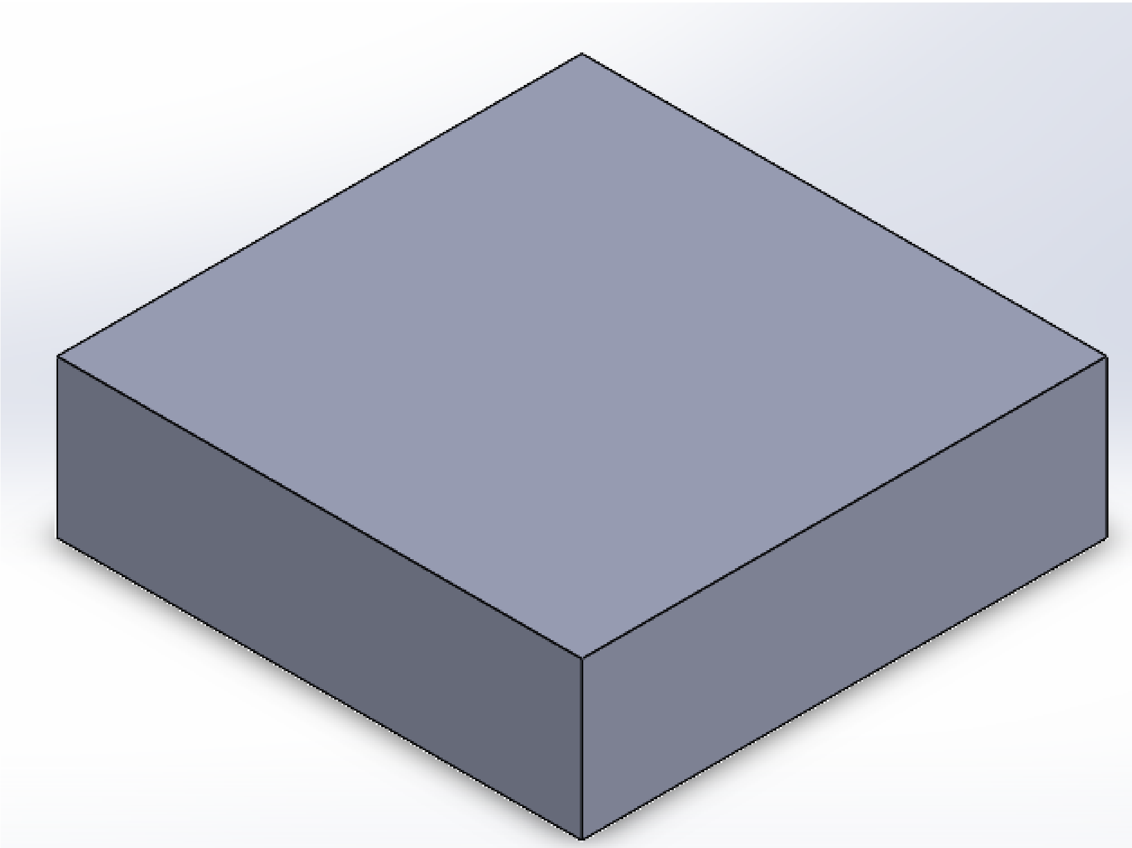

Figure 32 - Workpiece

Figure 33 shows the CNC milling machine used in this simulation. Inside the machine, the workpiece with the same dimensions as the one used in SolidWorks can be seen. Moreover, Figure 34 shows one of the tools needed to design this object. The tool represented on the figure is a drill with a diameter of 10 mm.

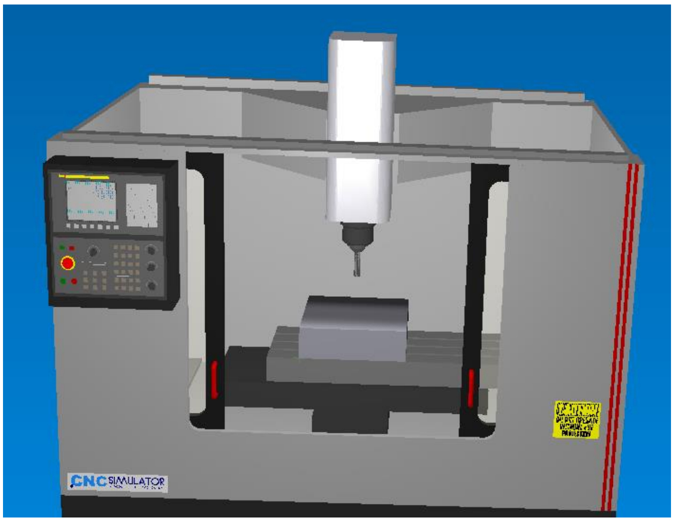

Figure 33 - CNC milling machine

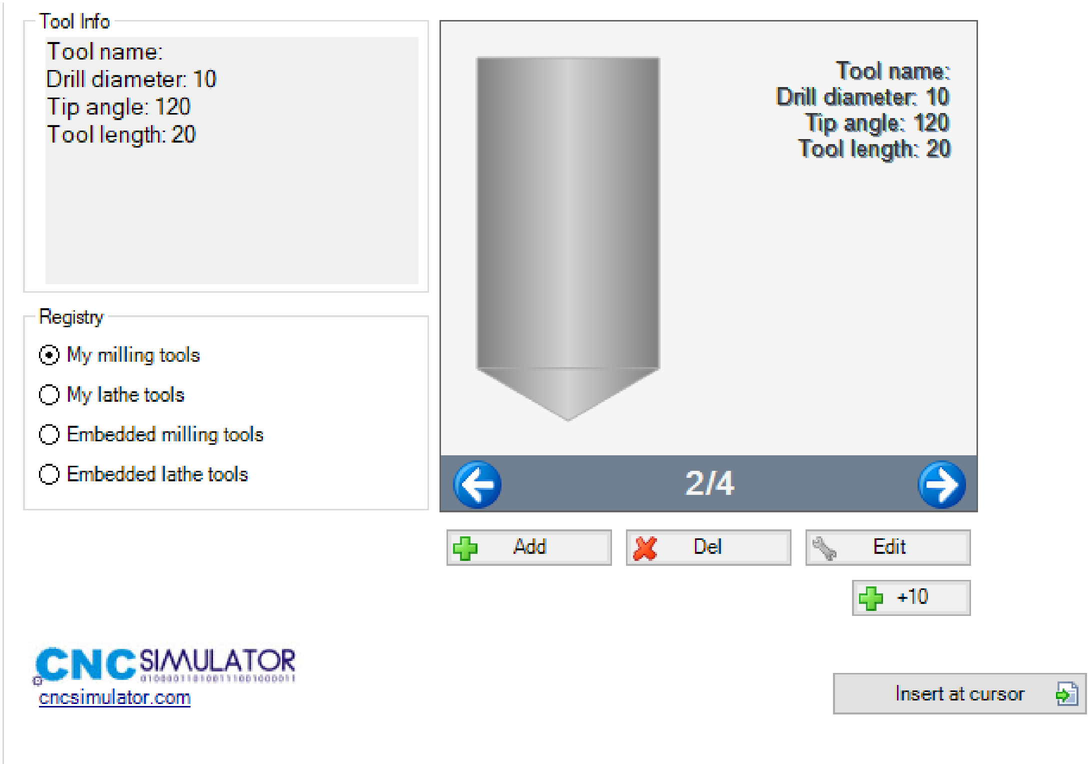

Figure 34 - Tool example

3.4.2 Interactive Process Planning

In order to demonstrate what was explained in the previous chapters, all of the features previously mentioned are used in this object.

Firstly, the macro for the pocket was run. As shown in Figure 35, different pockets were done, so the program ran more than one time. In this figure, there are represented four pockets. All of them done exclusively with the macro created. The two pockets on the top and on the bottom of the workpiece are square pockets with a length and width of 30 mm and a curved corner with a radius of 5 mm. Then, to demonstrate the versatility of this macro, two open pockets on the left and right side of the workpiece were created. These two pockets have a width of 30 mm and a length of 40 mm (in order to go until the edge of the workpiece and, as a consequence, forming an open pocket). As the first two pocket previously mentioned, these also have a curved corner with a radius of 5 mm. Finally, a larger pocket was done in the middle of the workpiece. This pocket has 78 mm length and 50 mm width. Furthermore, it has four curved corners with a radius of 8 mm. Moreover, all the pockets have the same depth, i.e., a 20 mm depth.

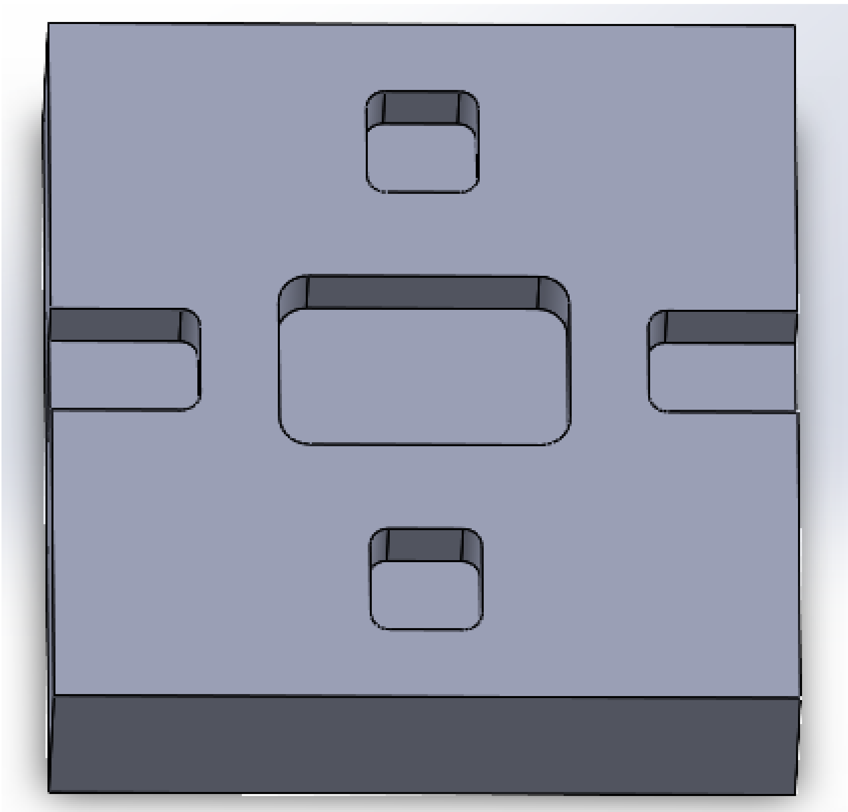

Figure 35 - Pockets in the proposed object

Then, the hole feature macro was run. As a result, four holes were made in each smaller pocket. These holes have all the same dimension. As a matter of fact, each one has 10 mm diameter and 25 mm length. As shown in Figure 36, the holes on the top and bottom pockets are placed in the center of that geometry, while the other two remaining are not. This was made with the intention to show that the user can define the coordinates of the center of the hole, and it does not need to be aligned with the other existing features.

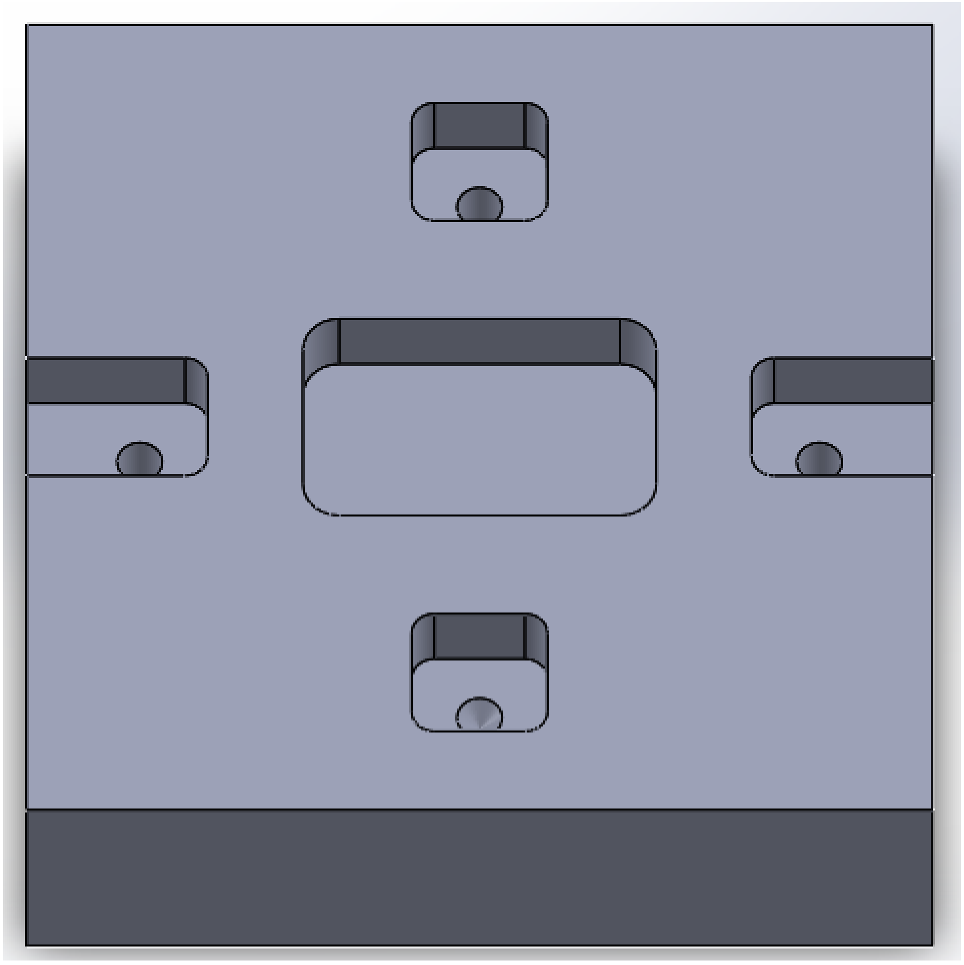

Figure 36 - Holes in the proposed geometry

There is one last macro to be used and that is the slot feature macro. To highlight the different possibilities when using this macro, nine different slots were created, as shown in Figure 37. Note that, in order to enhance this type of feature, the smaller pockets were suppressed in SolidWorks; hence, they were suppressed in this figure. Starting by the center of the geometry, three straight slots with a width of 12 mm were done inside the previous created pocket. In addition, they have a depth of 10 mm. Remembering what was said in previous sub-chapters, the length of this feature is not directly given by the user. In fact, the length is calculated by the software, and, for that, it uses the center point of each arc that belongs to the slot. For instance, the length between both center points in this case is 28 mm. Thus, the CNC machine will have to go until the Z coordinate -30, seeing that the pocket has 20 mm depth, and the slot has 10 mm depth. Actually, this detail shows that the macro can create a feature regardless of the Z coordinate indicated by the user and also that a feature can be created inside another feature. In other words, the program is not restricted to only one Z plan. Afterwards, four slots were created in each corner of the workpiece. Here, the intent is to point out that the feature can have a diagonal orientation. These slots have 18 mm width and 20 mm depth. Lastly, two slots were made; one on the right top corner and one on the left bottom corner of the workpiece. Here, it is possible to see that two features can overlap. Likewise, these features have a width of 18 mm. Nevertheless, based on the points given in the user interface, they have a smaller length. Moreover, as it can be seen in detail in Figure 38, they have a slighter greater depth than the other four slots. In fact, the depth of those pockets is 25 mm, i.e., they are 5 mm deeper than the others. The propose of this detail is to show that the program can design overlapped features with different depths.

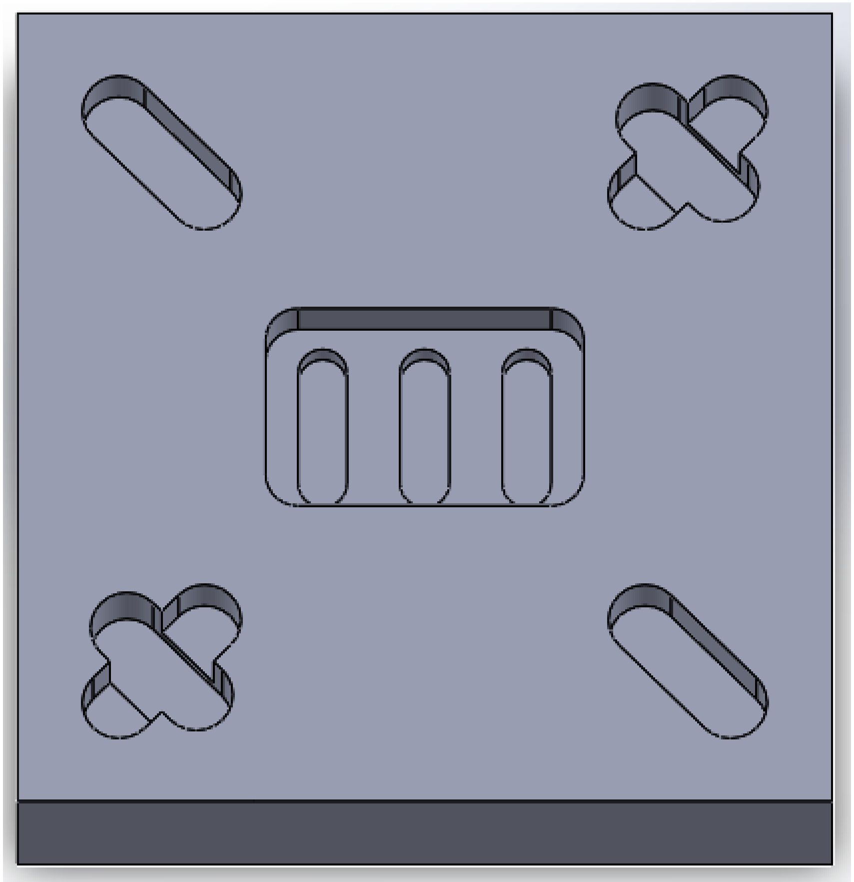

Figure 37 - Slots in the proposed geometry

Figure 39 shows the final state of the workpiece designing using only to the different macros written by the author of this dissertation.

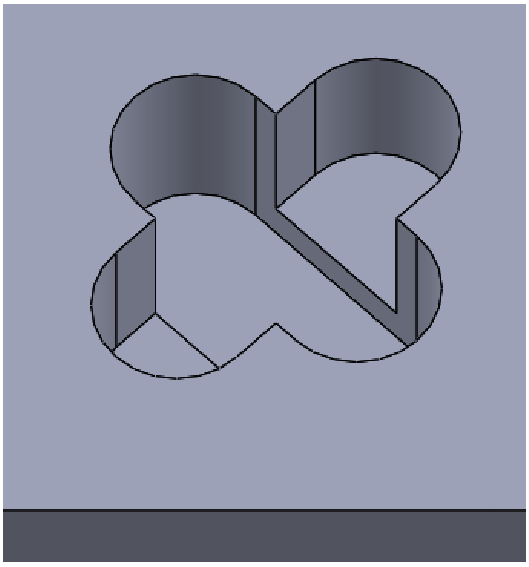

Figure 38 - Detail of the overlapped slots

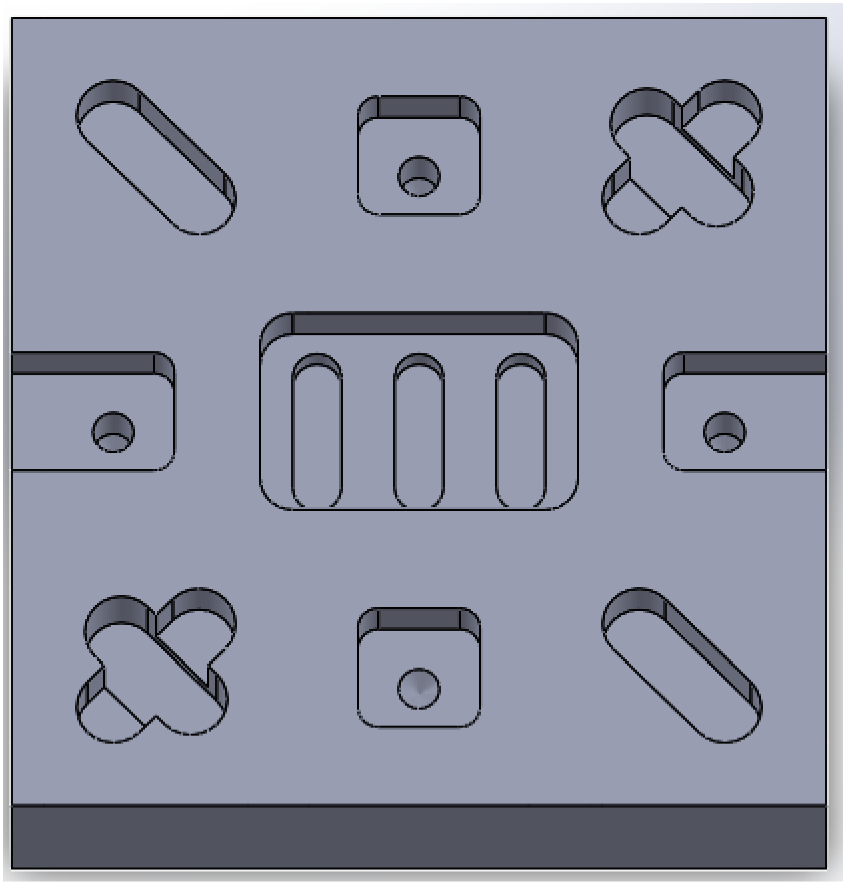

Figure 39 - Designed workpiece

3.4.3 Resulting G-code

As soon as the parameters needed are inserted in the user interface and the continue button is pressed, the algorithm processes the inputs according to the steps approached previously. Besides creating the geometry in SolidWorks, it also created the G-code needed to machine the workpiece. This G-code is automatically saved in the user's desktop as "Gcode". The complete G-code for the designed workpiece is presented on the annex 2. Table 3 shows bits of the code of two different features.

Table 3 - Bits of G-code from different features

Right side pocket	Bottom left slot
G00 X 60 Y 15 G00 Z 65 M08 G01 Z55 G01 X 100 Y15 G01 Y10 G01 X100 G01 X60 G01 Y5	G00 Z65 G00 X-56 Y-53 G01 Z55 G03 X-53 Y-56 R4 G01 X-74 Y-77 G03 X-77 Y-74 R4 G01 X-56 Y-53 G00 Z65 G00 X0 Y0

After completing all the steps and having the G-code in hand, the next step is to process the workpiece in a machining center. In this case, a simulator is used; thus, the only manual action done was copying the code from the different file text (different features) and paste them in the CNC simulator. After that, the "continue" button was pressed, and the simulator started. Figure 39 and Figure 40 show different phases of the simulation. Lastly, Figure 41 and Figure 42 show different angles thus, all the features of the final machined product can be seen.

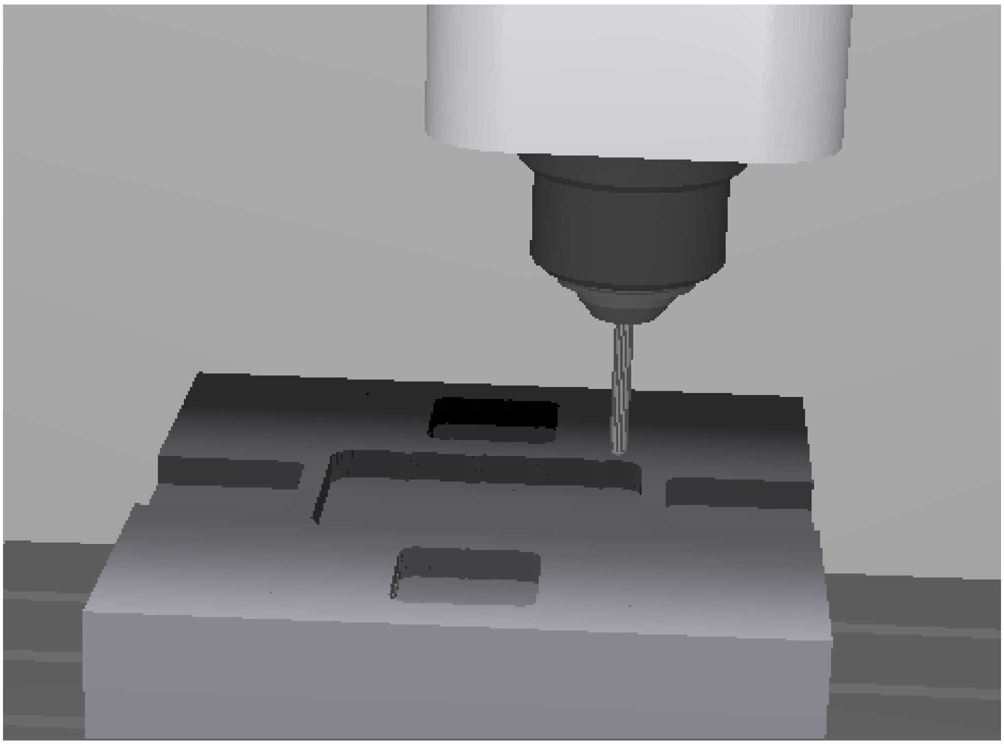

Figure 40 - First simulation frame

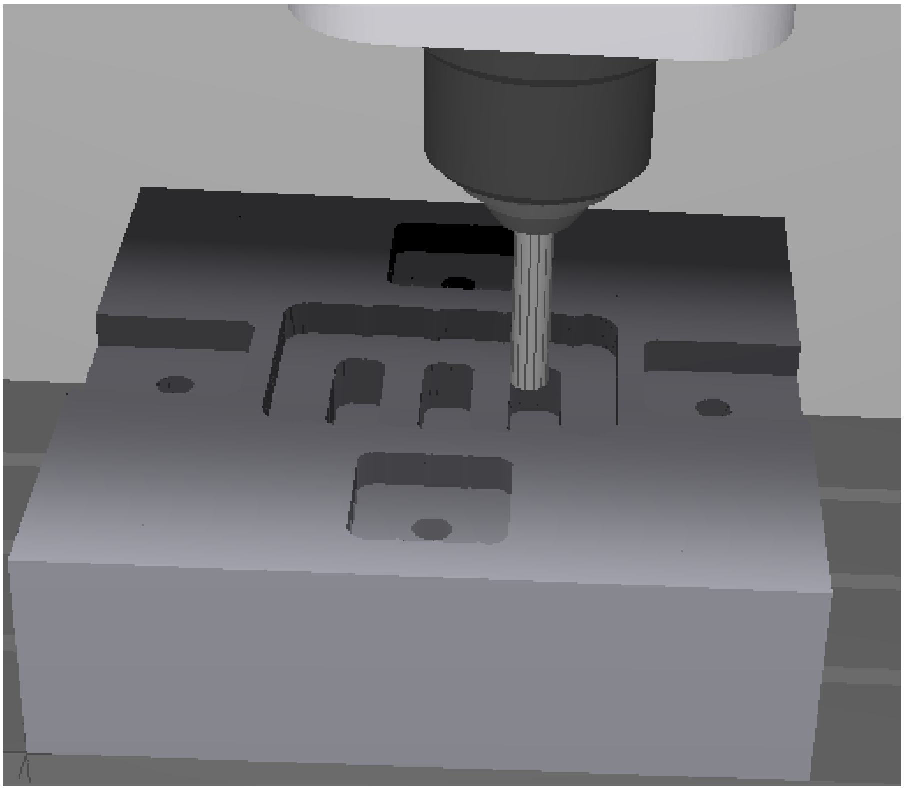

Figure 41 – Second simulation frame

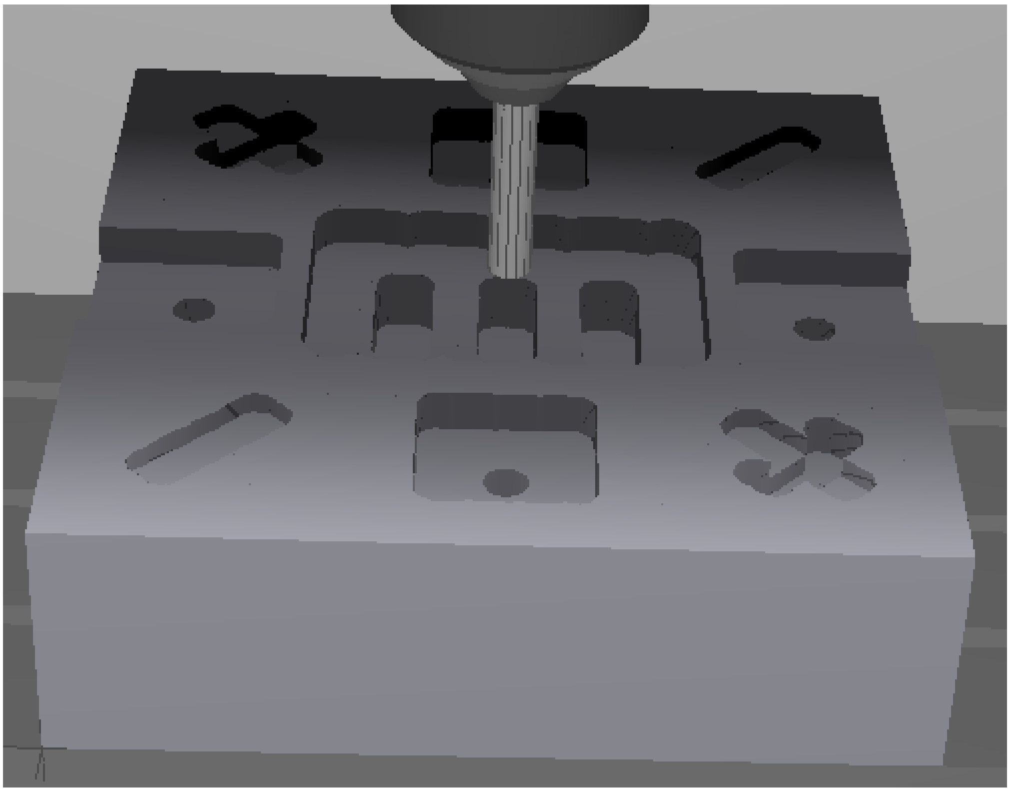

Figure 42 – Final product

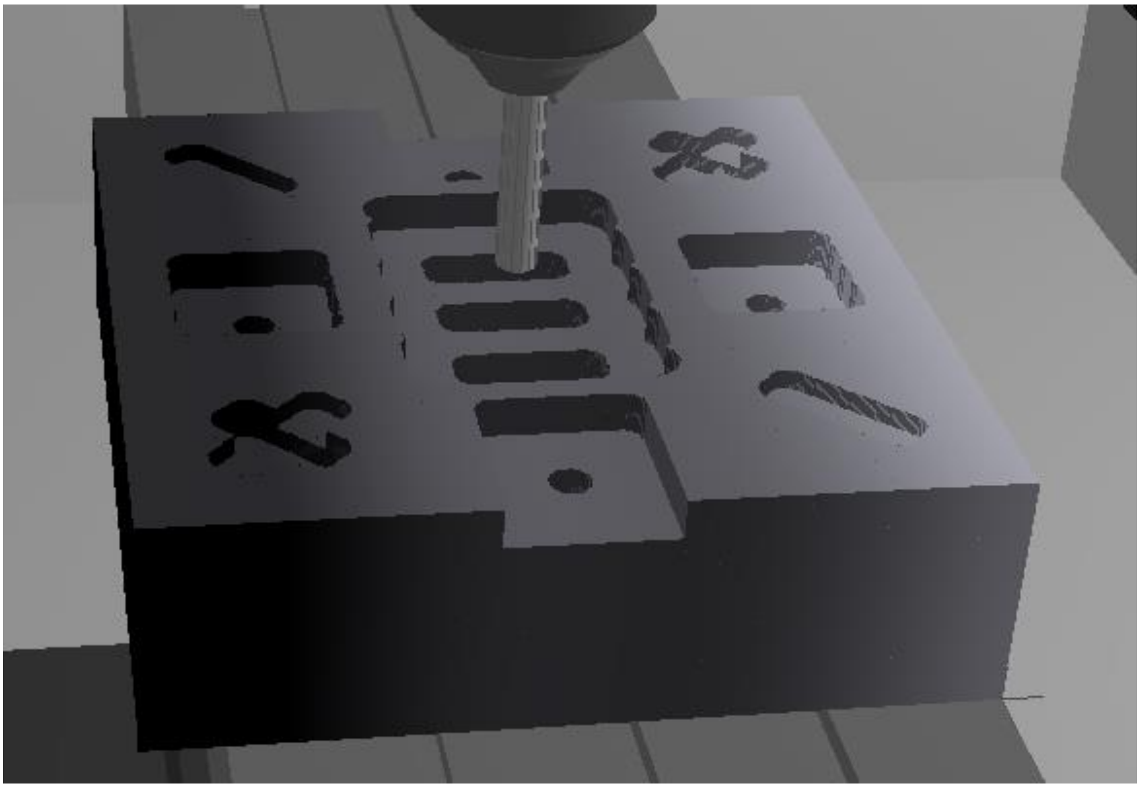

Figure 43 - Another perspective of the final product

3.4.4 Discussion

In this chapter, a global analysis is done in order to highlight the contribution of the proposed program.

As proposed, the developed program addresses a blank space in the literature, due to the fact that similar programs only exist for turning operations. Moreover, as shown in the literature review, and for turning operations, none of the highlighted articles connects the complete process, from the initial iteration to the final product, in a fully automated system. In fact, most of them do not even connect the design step with the manufacturing step. A small number of authors connect CAD and CAM in their algorithm, but even these algorithms require the user to manually design the product in the CAD system. Therefore, in the small amount of studies that enable CAD and CAM integration, there is no algorithm that can automatically design the final product, unlike the one proposed in this dissertation.

Looking at the final product manufactured in the machining center with the help of CNC simulator, it can be seen that a small number of parameters lead to a finished product. Although the simulator is running on the maximum definition possible, it cannot correctly show the walls of the features. Consequently, it appears that a finish cycle was not performed. In fact, if the machining of the proposed product was done in a real environment, this wouldn't happen.

Furthermore, the time required to design and produce a milling product is highly reduced with the proposed algorithm. As a consequence, an experienced user can save time, hence money by using it.

CONCLUSIONS

4.1 CONCLUSIONS

4.2 PROPOSALS OF FUTURE WORKS

4 CONCLUSIONS AND PROPOSALS OF FUTURE WORKS

4.1 Conclusions

In a product whose manufacturing process is machining, the first stage of its life cycle is the design part. Then, with the programming of the G-code, it can be inserted into a machine center. Hence, a workpiece can be transformed into a final product.

After an exhaustive literature review performed in the themes related to feature-based systems and machining operations, the author realized that there was a gap in the literature. Only a few authors integrated feature-based design with feature-based manufacturing in order to produce machined products. Furthermore, those few articles that have this concept as main theme, they applied it to parts obtained using mainly turning process. Hence, it was proposed to the author of this dissertation to create an algorithm that automatically, can design the features in the CAD software, under given parameters, as well as writing the G-code, that will be used in the CAM software to produce the final product.

Because of that, the main goals of this dissertation laid out in creating different milling features, always considering their versatility in order to achieve different geometries. Moreover, the goals highlighted, included the creation of an algorithm for the design and manufacturing part, as well as to be able to run the code and develop a product automatically since the first stage until the last one.

The algorithm of the present work allows the companies, with experienced employees, to save time and money. In fact, the time from the design stage until the machining stage in a manufacturing center can be drastically reduced. In addition, the algorithm can be adapted to include more features. However, the existing feature can be placed in the geometry in different ways and different dimensions. As a consequence, those features can provide innumerous product geometries.

In this way, the development of this work allowed the author to increase the knowledge about machining. Furthermore, it gave an opportunity to study feature-based systems that otherwise would not be aborded, providing insight on automated machining operations. The range around these topics is constantly increasing, demonstrating that they are becoming more and more important. Thus, the development of this work provided important knowledge for the upcoming challenges in the job market.

4.2 Proposals of future works

As previously mentioned, when developing the algorithm some constraints were considered. For example, those related to the stock depth versus hole depth as well as the compatibility of the different dimensions. However, these constraints are not enough when using this algorithm in a real context.

Thus, a way to improve the algorithm would be by adding more constraints. These constraints could verify for example, if the coordinates given by the user would theoretically place the feature outside the borders of the workpiece. Other possible constraint to be added is the one that can evaluate if a feature will be placed in the same place of another feature already in the workpiece (in the same Z plan). Actually, the implementation of these type of constraints requires the use of a lot of VBA functions as well as a lot of code lines.

Another improvement to this work would be an addition of more features to the algorithm. These new features, could include some made with split lines, allowing even more geometries and different products.

REFERENCES AND OTHER SOURCES OF INFORMATION

REFERENCES AND OTHER SOURCES OF INFORMATION

Adamson, G., Wang, L., & Moore, P. (2017). Feature-based control and information framework for adaptive and distributed manufacturing in cyber physical systems. *Journal of Manufacturing Systems*, *43*, 305–315. https://doi.org/10.1016/j.jmsy.2016.12.003

Adamson, G., Wang, L., & Moore, P. (2019). Feature-based function block control framework for manufacturing equipment in cloud environments. *International Journal of Production Research*, *57*(12), 3954–3974. https://doi.org/10.1080/00207543.2018.1542178

Altintas, Y., Kersting, P., Biermann, D., Budak, E., Denkena, B., & Lazoglu, I. (2014). Virtual process systems for part machining operations. *CIRP Annals - Manufacturing Technology*, *63*(2), 585–605. https://doi.org/10.1016/j.cirp.2014.05.007

Babic, B., Nesic, N., & Miljkovic, Z. (2008). A review of automated feature recognition with rule-based pattern recognition. *Computers in Industry*, *59*(4), 321–337. https://doi.org/10.1016/j.compind.2007.09.001

Battaïa, O., Dolgui, A., & Guschinsky, N. (2017). Integrated process planning and system configuration for mixed-model machining on rotary transfer machine. *International Journal of Computer Integrated Manufacturing*, *30*(9), 910–925. https://doi.org/10.1080/0951192X.2016.1247989

Besharati-Foumani, H., Lohtander, M., & Varis, J. (2020). Fundamentals and new achievements in feature-based modeling, a review. *Procedia Manufacturing*, *51*(2019), 998–1004. https://doi.org/10.1016/j.promfg.2020.10.140

Bissoonauth, K., & Namazi, H. (2019). Complexity-Based decoding of the effect of machining parameters on the machined surface in milling operation. *Fractals*, *27*(5), 1–9. https://doi.org/10.1142/S0218348X19500762

Bodein, Y., Rose, B., & Caillaud, E. (2014). Explicit reference modeling methodology in parametric CAD system. *Computers in Industry*, *65*(1), 136–147. https://doi.org/10.1016/j.compind.2013.08.004

Brinksmeier, E., Meyer, D., Huesmann-Cordes, A. G., & Herrmann, C. (2015). Metalworking fluids - Mechanisms and performance. *CIRP Annals - Manufacturing Technology*, *64*(2), 605–628. https://doi.org/10.1016/j.cirp.2015.05.003

Cardone, A., Gupta, S. K., Deshmukh, A., & Karnik, M. (2006). Machining feature-based similarity assessment algorithms for prismatic machined parts. *CAD Computer Aided Design*, *38*(9), 954–972. https://doi.org/10.1016/j.cad.2006.08.001

Cardoso, C., & Quelhas, V. (2018). Metalurgia e Metalomecânica: número um das exportações. *Portugal Global*, *114*, 1–78.

Cardoso, C., & Quelhas, V. (2020). Portugal Global. *Portugal Global*, *134*, 32–37.

Chang, T., Wysh, R. A., & Wang, H.-P. (1998). *Computer-Aided Manufacturing* (W. J. Fabrycky & J. H. Mize (eds.); 3rd ed.). Prentice HALL.

Chen, N. (2021). An evolutionary neural network approach to machining process planning: A proof of concept. *Procedia Manufacturing*, *53*, 690–696. https://doi.org/10.1016/j.promfg.2021.06.083

Chen, N., Barnawal, P., & Frank, M. C. (2018). Automated post machining process planning for a new hybrid manufacturing method of additive manufacturing and rapid machining. *Rapid Prototyping Journal*, *24*(7), 1077–1090. https://doi.org/10.1108/RPJ-04-2017-0057

Cheng, Y., He, F., Wu, Y., & Zhang, D. (2016). Meta-operation conflict resolution for human–human interaction in collaborative feature-based CAD systems. *Cluster Computing*, *19*(1), 237–253. https://doi.org/10.1007/s10586-016-0538-0

Cheng, Z., & Ma, Y. (2017). A functional feature modeling method. *Advanced Engineering Informatics*, *33*, 1–15. https://doi.org/10.1016/j.aei.2017.04.003

Chu, C. C. P., & Gadh, R. (1996). Feature-based approach for set-up minimization of process design from product design. *CAD Computer Aided Design*, *28*(5), 321–332. https://doi.org/10.1016/0010-4485(95)00052-6

Chu, X., Tang, C., Xue, D., Su, Y., & Sun, X. (2012). Identification of machining features based on available resources of cutting tools. *International Journal of Production Research*, *50*(15), 4141–4157. https://doi.org/10.1080/00207543.2011.596845

Costa, C., Silva, F. J. G., Gouveia, R. M., & Martinho, R. P. (2018). Development of hydraulic clamping tools for the machining of complex shape mechanical components. *Procedia Manufacturing*, *17*, 563–570. https://doi.org/10.1016/j.promfg.2018.10.097

Date, N., Krishnaswami, P., & Motipalli, V. V. S. K. (2009). Automated process planning method to machine A B-Spline free-form feature on a mill-turn center. *Computers and Industrial Engineering*, *56*(1), 198–207. https://doi.org/10.1016/j.cie.2008.05.004

Deja, M., & Siemiatkowski, M. S. (2013). Feature-based generation of machining process plans for optimised parts manufacture. *Journal of Intelligent Manufacturing*, *24*(4), 831–846. https://doi.org/10.1007/s10845-012-0633-x

Denkena, B., Dittrich, M. A., Nguyen, H. N., & Bild, K. (2021). Self-optimizing process planning of multi-step polishing processes. *Production Engineering*, *15*(3–4), 563–571. https://doi.org/10.1007/s11740-021-01042-6

Dipper, T., Xu, X., & Klemm, P. (2011). Defining, recognizing and representing feature interactions in a feature-based data model. *Robotics and Computer-Integrated Manufacturing*, *27*(1), 101–114. https://doi.org/10.1016/j.rcim.2010.06.016

Dittrich, M. A., Uhlich, F., & Denkena, B. (2019). Self-optimizing tool path generation for 5-axis machining processes. *CIRP Journal of Manufacturing Science and Technology*, *24*, 49–54. https://doi.org/10.1016/j.cirpj.2018.11.005

Dodok, T., Čuboňová, N., Císar, M., Kuric, I., & Zajačko, I. (2017). Utilization of Strategies to Generate and Optimize Machining Sequences in CAD/CAM. *Procedia Engineering*, *192*, 113–118. https://doi.org/10.1016/j.proeng.2017.06.020

Duan, W., Zhou, J., & Lai, K. (1993). FSMT: a feature solid-modelling tool for feature-based design and manufacture. *Computer-Aided Design*, *25*(1), 29–38. https://doi.org/10.1016/0010-4485(93)90063-T

Fountas, N. A., Vaxevanidis, N. M., Stergiou, C. I., & Benhadj-Djilali, R. (2019). Globally optimal tool paths for sculptured surfaces with emphasis to machining error and cutting posture smoothness. *International Journal of Production Research*, *57*(17), 5478–5498. https://doi.org/10.1080/00207543.2018.1530468

Frisch, A. M., Jefferson, C., Hernández, B. M., & Miguel, I. (2005). The rules of constraint modelling. *IJCAI International Joint Conference on Artificial Intelligence*, 109–116. https://doi.org/https://doi.org/10.5555/1642293.1642311

Giannuzzi, M., Papadia, G., & Pascarelli, C. (2020). IC.IDO as a tool for displaying machining processes. The logic interface between computer-aided-manufacturing and virtual reality. *Procedia CIRP*, *88*, 145–150. https://doi.org/10.1016/j.procir.2020.05.026

Gouveia, R., Silva, F., Reis, P., & Baptista, A. (2016). Machining Duplex Stainless Steel: Comparative Study Regarding End Mill Coated Tools. *Coatings*, *6*(4), 51. https://doi.org/10.3390/coatings6040051

Han, J. H., Kang, M., & Choi, H. (2001). STEP-based feature recognition for manufacturing cost optimization. *CAD Computer Aided Design*, *33*(9), 671–686. https://doi.org/10.1016/S0010-4485(01)00071-9

He, C., Guan, Z., Yue, L., & Ullah, S. (2018). Set-partitioning-based heuristic for balancing and configuration of automated flexible machining line. *International Journal of Production Research*, *56*(9), 3152–3172. https://doi.org/10.1080/00207543.2018.1436785

He, K., Tang, R., & Jin, M. (2017). Pareto fronts of machining parameters for trade-off among energy consumption, cutting force and processing time. *International Journal of Production Economics*, *185*, 113–127. https://doi.org/10.1016/j.ijpe.2016.12.012

Hoffmann, J. (1989). *Geometric and Solid Modeling*. Morgan Kaufmann Publishers. https://doi.org/10.1007/978-3-540-39607-9_1

Ji, W., Wang, L., Haghighi, A., Givehchi, M., & Liu, X. (2018). An enriched machining feature based approach to cutting tool selection. *International Journal of Computer Integrated Manufacturing*, *31*(1), 1–10. https://doi.org/10.1080/0951192X.2017.1356472

Jing, X., Zhu, Y., Liu, J., Zhou, H., Zhao, P., Liu, X., Tian, G., Ye, H., & Li, Q. (2020). Intelligent generation method of 3D machining process based on process knowledge. *International Journal of Computer Integrated Manufacturing*, *33*(1), 38–61. https://doi.org/10.1080/0951192X.2019.1690687

Jong, W. R., Lai, P. J., & Li, T. C. (2014). Integration and application of feature recognition and mould manufacturing planning navigating process. *International Journal of Production Research*, *52*(20), 5945–5964.

https://doi.org/10.1080/00207543.2014.894259

Jong, W. R., Wu, C. H., & Lee, M. Y. (2011). Feature-based integration of conceptual and detailed mould design. *International Journal of Production Research*, *49*(16), 4833–4855. https://doi.org/10.1080/00207543.2010.504421

Jong, W. R., Wu, C. H., Liu, H. H., & Li, M. Y. (2009). A collaborative navigation system for concurrent mold design. *International Journal of Advanced Manufacturing Technology*, *40*(3–4), 215–225. https://doi.org/10.1007/s00170-007-1328-x

Jurkovic, M., & Jurkovic, Z. (2008). *Machine Tools and Machining Question Bank*. 1–19.

Jurkovic, M., Jurkovic, Z., Mahmic, M., & Karabegovic, E. (2008). Machining and Manufacturing Systems Evolution. *Trends in the Development of Machinery and Associated Technology*, *to 2020*, 85–88.

Kamrani, A., Abouel Nasr, E., Al-Ahmari, A., Abdulhameed, O., & Mian, S. H. (2015). Feature-based design approach for integrated CAD and computer-aided inspection planning. *International Journal of Advanced Manufacturing Technology*, *76*(9–12), 2159–2183. https://doi.org/10.1007/s00170-014-6396-0

Kardos, C., Kovács, A., & Váncza, J. (2020). A constraint model for assembly planning. *Journal of Manufacturing Systems*, *54*, 196–203. https://doi.org/10.1016/j.jmsy.2019.11.007

Komatsu, W., & Nakamoto, K. (2021). Machining process analysis for machine tool selection based on form-shaping motions. *Precision Engineering*, *67*, 199–211. https://doi.org/10.1016/j.precisioneng.2020.09.023

Kumar, R., Singh, S., Bilga, P. S., Jatin, Singh, J., Singh, S., Scutaru, M. L., & Pruncu, C. I. (2021). Revealing the benefits of entropy weights method for multi-objective optimization in machining operations: A critical review. *Journal of Materials Research and Technology*, *10*(4), 1471–1492. https://doi.org/10.1016/j.jmrt.2020.12.114

Laguionie, R., Rauch, M., Hascoët, J. Y., & Suh, S. H. (2011). An eXtended manufacturing integrated system for feature-based manufacturing with STEP-NC. *International Journal of Computer Integrated Manufacturing*, *24*(9), 785–799. https://doi.org/10.1080/0951192X.2011.592992

Lee, H., Kim, J., & Banerjee, A. (2010). Collaborative intelligent CAD framework incorporating design history tracking algorithm. *CAD Computer Aided Design*, *42*(12), 1125–1142. https://doi.org/10.1016/j.cad.2010.08.001

Lee, S. H., & Lee, K. (2012). Simultaneous and incremental feature-based multiresolution modeling with feature operations in part design. *CAD Computer Aided Design*, *44*(5), 457–483. https://doi.org/10.1016/j.cad.2011.12.005

Lee, Y., Resiga, A., Yi, S., & Wern, C. (2020). The Optimization of Machining Parameters for Milling Operations by Using the Nelder-Mead Simplex Method. *Journal of Manufacturing and Materials Processing*, *4*(3), 1–22. https://doi.org/10.3390/jmmp4030066

Leo Kumar, S. P., Jerald, J., & Kumanan, S. (2015). Feature-based modelling and process parameters selection in a CAPP system for prismatic micro parts. *International Journal of Computer Integrated Manufacturing*, *28*(10), 1046–1062. https://doi.org/10.1080/0951192X.2014.953586

Li, G., Long, X., Zhou, M., Xiong, H., & Wang, W. (2017). A geometric feature-based design system of full parametric association modeling of standard cam for automotive stamping dies. *International Journal of Advanced Manufacturing Technology*, *92*(9–12), 3627–3645. https://doi.org/10.1007/s00170-017-0421-z

Li, Lei, Zheng, Y., Yang, M., Leng, J., Cheng, Z., Xie, Y., Jiang, P., & Ma, Y. (2020). A survey of feature modeling methods: Historical evolution and new development. *Robotics and Computer-Integrated Manufacturing*, *61*, 101851. https://doi.org/10.1016/j.rcim.2019.101851

Li, Lingling, Li, C., Tang, Y., & Yi, Q. (2017). Influence factors and operational strategies for energy efficiency improvement of CNC machining. *Journal of Cleaner Production*, *161*, 220–238. https://doi.org/10.1016/j.jclepro.2017.05.084

Li, R., Hegde, P., & Shih, A. J. (2007). High-throughput drilling of titanium alloys. *International Journal of Machine Tools and Manufacture*, *47*(1), 63–74. https://doi.org/10.1016/j.ijmachtools.2006.02.012

Li, W., Mac, G., Tsoutsos, N. G., Gupta, N., & Karri, R. (2020). Computer aided design (CAD) model search and retrieval using frequency domain file conversion. *Additive Manufacturing*, *36*(July), 101554. https://doi.org/10.1016/j.addma.2020.101554

Li, X., He, F., Cai, X., & Zhang, D. (2012). CAD data exchange based on the recovery of feature modelling procedure. *International Journal of Computer Integrated Manufacturing*, *25*(10), 874–887. https://doi.org/10.1080/0951192X.2011.608721

Li, Z., Wang, J., Anwar, M. S., & Zheng, Z. (2020). An efficient method for generating assembly precedence constraints on 3D models based on a block sequence structure. *CAD Computer Aided Design*, *118*, 102773. https://doi.org/10.1016/j.cad.2019.102773

Liu, H., Liu, Q., Sun, P., Liu, Q., & Yuan, S. (2017). The optimal feedrate planning on five-axis parametric tool path with geometric and kinematic constraints for CNC machine tools. *International Journal of Production Research*, *55*(13), 3715–3731. https://doi.org/10.1080/00207543.2016.1254357

Liu, J., Liu, X., Cheng, Y., & Ni, Z. (2017). An approach to mapping machining feature to manufacturing feature volume based on geometric reasoning for process planning. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, *231*(7), 1204–1216. https://doi.org/10.1177/0954405415585377

López De Lacalle, L. N., Pérez-Bilbatua, J., Sánchez, J. A., Llorente, J. I., Gutiérrez, A., & Albóniga, J. (2000). Using high pressure coolant in the drilling and turning of low machinability alloys. *International Journal of Advanced Manufacturing Technology*, *16*(2), 85–91. https://doi.org/10.1007/s001700050012

Lyu, G., Chu, X., & Xue, D. (2017). Product modeling from knowledge, distributed computing and lifecycle perspectives: A literature review. *Computers in Industry*, *84*, 1–13. https://doi.org/10.1016/j.compind.2016.11.001

Ma, Y. S., Tang, S. H., Au, C. K., & Chen, J. Y. (2009). Collaborative feature-based design via operations with a fine-grain product database. *Computers in Industry*, *60*(6), 381–391. https://doi.org/10.1016/j.compind.2009.02.013

Majerik, J., & Jambor, J. (2015). Computer aided design and manufacturing evaluation of milling cutter when high speed machining of hardened steels. *Procedia Engineering*, *100*(January), 450–459. https://doi.org/10.1016/j.proeng.2015.01.390

Mei, B., Wang, H., & Zhu, W. (2021). Pose and shape error control in automated machining of fastener holes for composite/metal wing-box assembly. *Journal of Manufacturing Processes*, *66*, 101–114. https://doi.org/10.1016/j.jmapro.2021.03.052

Michalik, P., Zajac, J., Duplák, J., & Pivovarnik, A. (2012). CAM software products for creation of programs for CNC machining. *Lecture Notes in Electrical Engineering*, *141 LNEE*(VOL. 1), 421–425. https://doi.org/10.1007/978-3-642-27311-7_56

Möhring, H. C., Wiederkehr, P., Erkorkmaz, K., & Kakinuma, Y. (2020). Self-optimizing machining systems. *CIRP Annals*, *69*(2), 740–763. https://doi.org/10.1016/j.cirp.2020.05.007

Molloy, E., Yang, H., & Browne, J. (1993). Feature-based modelling in design for assembly. *International Journal of Computer Integrated Manufacturing*, *6*(1–2), 119–125. https://doi.org/10.1080/09511929308944562

Nagarajan, S., & Reddy, N. V. (2010). STEP-based automatic system for recognising design and manufacturing features. *International Journal of Production Research*, *48*(1), 117–144. https://doi.org/10.1080/00207540701855419

Núñez, P. J., García-Plaza, E., Martín, A. R., & Egido, A. (2011). An integrated methodology for the teaching of computer aided tools for automated machining. *Materials Science Forum*, *692*, 8–15. https://doi.org/10.4028/www.scientific.net/MSF.692.8

Osman Zahid, M. N., Case, K., & Watts, D. (2014). Optimization of roughing operations in CNC machining for rapid manufacturing processes. *Production and Manufacturing Research*, *2*(1), 519–529. https://doi.org/10.1080/21693277.2014.938277

Otoboni, J. A., Duduch, J. G., Jasinevicius, R., & Gonçalves, A. M. (2013). *Analysis of Zerodur ® Machinability Using Single Point Diamond Turning*. *Cobem*, 8786–8796.

Ovtcharova, J., & Jasnoch, U. (1994). Featured-based design and consistency management in CAD applications: a unified approach. *Advances in Engineering Software*, *20*(2–3), 65–73. https://doi.org/10.1016/0965-9978(94)90050-7

Panda, A., Ranjan Das, S., & Dhupal, D. (2019). Machinability Investigation of HSLA Steel in Hard Turning with Coated Ceramic Tool: Assessment, Modeling, Optimization and Economic Aspects. *Journal of Advanced Manufacturing Systems*, *18*(4), 625–655. https://doi.org/10.1142/S0219686719500331

Patil, L., & Pande, S. S. (2002). An intelligent feature-based process planning system for prismatic parts. *International Journal of Production Research*, *40*(17), 4431–4447. https://doi.org/10.1080/00207540210155855

Pedagopu, Vi. M., & Herano, F. S. (2019). Development of an algorithm for feature based prismatic part. *International Journal of Engineering, Science and Mathematics*, *8*(6), 69–92.

Pelliccia, L., Bojko, M., Prielipp, R., & Riedel, R. (2021). Applicability of 3D-factory simulation software for computer-aided participatory design for industrial workplaces and processes. *Procedia CIRP*, *99*, 122–126. https://doi.org/10.1016/j.procir.2021.03.019

Perez, H., Diez, E., Perez, J., & Vizan, A. (2013). Analysis of machining strategies for peripheral milling. *Procedia Engineering*, *63*, 573–581. https://doi.org/10.1016/j.proeng.2013.08.193

Qian, X., & Dutta, D. (2004). Feature-based design for heterogeneous objects. *CAD Computer Aided Design*, *36*(12), 1263–1278. https://doi.org/10.1016/j.cad.2004.01.012

Ramos, A. M., Relvas, C., & Simões, J. A. (2003). The influence of finishing milling strategies on texture, roughness and dimensional deviations on the machining of complex surfaces. *Journal of Materials Processing Technology*, *136*(1–3), 209–216. https://doi.org/10.1016/S0924-0136(03)00160-2

Roller, D. (1989). Integrating advaned Computer- Aided Design, Manufacturing. *Computers in Industry*, *12*(3), 185–191. https://doi.org/10.1016/0166-3615(89)90065-1

Samanta, K., & Koc, B. (2005). Feature-based design and material blending for free-form heterogeneous object modeling. *CAD Computer Aided Design*, *37*(3), 287–305. https://doi.org/10.1016/j.cad.2004.03.005

Sanfilippo, E. M. (2018). Feature-based product modelling: an ontological approach. *International Journal of Computer Integrated Manufacturing*, *31*(11), 1097–1110. https://doi.org/10.1080/0951192X.2018.1497814

Shahin, T. M. M. (2008). Feature-based design - An overview. *Computer-Aided Design and Applications*, *5*(5), 639–653. https://doi.org/10.3722/cadaps.2008.639-653

Shi, Y., Zhang, Y., Baek, S., De Backer, W., & Harik, R. (2018). Manufacturability analysis for additive manufacturing using a novel feature recognition technique. *Computer-Aided Design and Applications*, *15*(6), 941–952. https://doi.org/10.1080/16864360.2018.1462574

Shipulin, L. V., Dyakonov, A. A., & Al-Aqeeli, M. M. (2017). Development of Module for Automated Abrasive Tool Selection on High-speed Machining Operation. *Procedia Engineering*, *206*, 1211–1214. https://doi.org/10.1016/j.proeng.2017.10.620

Shojaeipour, S. (2010). A novel method for automated tool path optimisation for CNC machining operations. *Solid State Phenomena*, *166–167*, 357–362. https://doi.org/10.4028/www.scientific.net/SSP.166-167.357

Shokrani, A., Dhokia, V., Muñoz-Escalona, P., & Newman, S. T. (2013). State-of-the-art cryogenic machining and processing. *International Journal of Computer Integrated Manufacturing*, *26*(7), 616–648. https://doi.org/10.1080/0951192X.2012.749531

Sousa, V. F. C., Silva, F. J. G., Fecheira, J. S., Lopes, H. M., Martinho, R. P., Casais, R. B., & Ferreira, L. P. (2020). Cutting forces assessment in CNC machining processes: A criticalreview review. *Sensors (Switzerland)*, *20*(16), 1–26. https://doi.org/10.3390/s20164536

Sunil, V. B., Agarwal, R., & Pande, S. S. (2010). An approach to recognize interacting features from B-Rep CAD models of prismatic machined parts using a hybrid (graph and rule based) technique. *Computers in Industry*, *61*(7), 686–701. https://doi.org/10.1016/j.compind.2010.03.011

Suresh Kumar, B., Baskar, N., & Rajaguru, K. (2020). Drilling operation: A review. *Materials Today: Proceedings*, *21*, 926–933. https://doi.org/10.1016/j.matpr.2019.08.160

Tolouei-Rad, M. (2011). An intelligent approach to high quantity automated machining. *Journal of Achievements in Materials and Manufacturing Engineering*, *47*(2), 195–204.

Usubamatov, R., Bin Harun, A., & Bin Sanuddin, A. (2014). Optimisation of machining parameters by criterion of maximum productivity. *International Journal of Production Research*, *52*(10), 2946–2953. https://doi.org/10.1080/00207543.2013.857440

Venu, B., & Komma, V. R. (2017). STEP-based feature recognition from solid models having non-planar surfaces. *International Journal of Computer Integrated Manufacturing*, *30*(10), 1011–1028. https://doi.org/10.1080/0951192X.2016.1268719

Villanueva, E. M., Mamledesai, H., Martinez, P., Poostchi, P., & Ahmad, R. (2021). Design and simulation of an automated robotic machining cell for cross-laminated timber panels. *Procedia CIRP*, *100*, 175–180. https://doi.org/10.1016/j.procir.2021.05.026

Volkmann, M., Legler, T., Wagner, A., & Ruskowski, M. (2020). A CAD feature-based manufacturing approach with OPC UA skills. *Procedia Manufacturing*, *51*, 416–423. https://doi.org/10.1016/j.promfg.2020.10.059

Wang, M. T., Chamberlain, M. A., Joneja, A., & Chang, T. C. (1993). Manufacturing feature extraction and machined volume decomposition in a computer-integrated feature-based design and manufacturing planning environment. *Computers in Industry*, *23*(1–2), 75–86. https://doi.org/10.1016/0166-3615(93)90117-J

Wang, W., Li, Y., Li, H., & Liu, C. (2012). An agent-based collaborative design framework for feature-based design of aircraft structural parts. *International Journal of Computer Integrated Manufacturing*, *25*(10), 888–900.

https://doi.org/10.1080/0951192X.2011.597779

Wang, W., Li, Y., & Tang, L. (2014). Drive geometry construction method of machining features for aircraft structural part numerical control machining. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, *228*(10), 1214–1225. https://doi.org/10.1177/0954405413516953

Xu, X. (2009). *Integrating advanced Computer- Aided Design, Manufacturing, and Numeric Control*.

Yang, W. Z., Xie, S. Q., Ai, Q. S., & Zhou, Z. D. (2008). Recent development on product modelling: A review. *International Journal of Production Research*, *46*(21), 6055–6085. https://doi.org/10.1080/00207540701343895

Yin, C. G., & Ma, Y. S. (2012). Parametric feature constraint modeling and mapping in product development. *Advanced Engineering Informatics*, *26*(3), 539–552. https://doi.org/10.1016/j.aei.2012.02.010

Yingjie, Z., & Liling, G. (2015). Reliability analysis of machining systems by considering system cost. *International Journal of Computer Integrated Manufacturing*, *28*(8), 836–843. https://doi.org/10.1080/0951192X.2014.914630

Youssef, H. A., & El-Hofy, H. (2008). *Machining Technology Machine Tools and Operations* (1st ed.). CRC Press. https://doi.org/https://doi.org/10.1201/9781420043402

Yue, D., Zhang, A., Yue, C., Liu, X., Li, M., & Hu, D. (2021). Optimization Method of Tool Parameters and Cutting Parameters Considering Dynamic Change of Performance Indicators. *Materials*, *14*(20), 6181. https://doi.org/10.3390/ma14206181

Zeid, I. (2004). *Mastering CAD/CAM* (1st ed.). McGraw-Hill Science.

Zhang, W., Zhou, Y., & Zhu, J. (2017). A comprehensive study of feature definitions with solids and voids for topology optimization. *Computer Methods in Applied Mechanics and Engineering*, *325*, 289–313. https://doi.org/10.1016/j.cma.2017.07.004

Zheng, Y., Mohd Taib, J., & Md Tap, M. M. (2012). Decomposition of interacting machining features based on the reasoning on the design features. *International Journal of Advanced Manufacturing Technology*, *58*(1–4), 359–377. https://doi.org/10.1007/s00170-011-3385-4

ANNEXES

ANNEX 1

ANNEX 2

ANNEXES

In this chapter, two annexes are presented. The first one shows the algorithm made during the development of this dissertation. The second presents the G-code obtained by designing the product to test the algorithm.

Annex 1

```
Dim swApp As SldWorks.SldWorks
Dim Part As ModelDoc2
Dim PI As Double

  Sub main()

  Set swApp = Application.SldWorks
  Set Part = swApp.ActiveDoc
  PI = 4 * Math.Atn(1)

  If Part Is Nothing Then
  MsgBox " You should open a document first."
  Exit Sub
  End If

  UserForm1.Show

End Sub

Public Sub CreateHole(InputX As Double, InputY As Double, InputZ As Double, InputDiameter As Double, InputLength As Double, InputStockDepth As Double)

  Dim status As Boolean
  Dim UserCoordX As Double
  Dim UserCoordY As Double
  Dim UserCoordZ As Double
  Dim CoordX As Double
  Dim CoordY As Double
  Dim CoordZ As Double
  Dim Diameter As Double
  Dim Radius As Double
  Dim Lenght As Double
  Dim aux As Double
  Dim stockLenght As Double
```

```
    UserCoordX = InputX
    UserCoordY = InputY
    UserCoordZ = InputZ
    Diameter = InputDiameter
    Radius = Diameter / 2
    Length = InputLength
    StockLength = InputStockLength

    If Length > 8 * Diameter Then
        MsgBox " Deep Hole, You need to use a special tool/machine for that"
        Exit Sub
    End If


    If Length - 2 > StockLength Then
        MsgBox " Hole depth needs to be lower than stock depth"
        Exit Sub
    End If
    'status = Part.Extension.SelectByID2("PLANE2", "PLANE", 0, 0, 0, False, 0, Nothing, 0)
    status = Part.Extension.SelectByRay(UserCoordX, UserCoordY, UserCoordZ, 0, -1, 0,
4.16884171005861E-04, 2, False, 0, 0)
    Part.ClearSelection2 True


    Set sketchLine = Part.SketchManager.CreateCenterLine(UserCoordX, UserCoordY +
0.005, UserCoordZ, UserCoordX, (UserCoordY - Length - Length), UserCoordZ)

    Set sketchLine = Part.SketchManager.CreateLine(UserCoordX, UserCoordY,
UserCoordZ, UserCoordX + Radius, UserCoordY, UserCoordZ)
        CoordX = UserCoordX + Radius

    Set sketchLine = Part.SketchManager.CreateLine(CoordX, UserCoordY, UserCoordZ,
CoordX, (UserCoordY - Length), UserCoordZ)
    CoordY = (UserCoordY - Length)

    aux = Math.Tan(31 * PI / 180) * Radius

    Set sketchLine = Part.SketchManager.CreateLine(CoordX, CoordY, UserCoordZ,
UserCoordX, CoordY - aux, UserCoordZ)

    CoordY = aux + CoordY
```

```
Set sketchLine = Part.SketchManager.CreateLine(UserCoordX, CoordY, UserCoordZ,
UserCoordX, UserCoordY, UserCoordZ)

Part.SetPickMode
Part.ClearSelection2 True

Dim Hole As Object
Set Hole = Part.FeatureManager.FeatureRevolve2(True, True, False, True, False, False,
0, 0, 6.2831853071796, 0, False, False, 0.01, 0.01, 0, 0, 0, True, True, True)

Dim r As Double
Dim DrillId As String
Dim DrillTable(6, 6) As String
Dim i As Integer
Dim j As Integer
Dim Xt As Double
Dim Xb As Double
Dim Yt As Double
Dim Yb As Double
Dim Zt As Double
Dim Zb As Double

Xt = UserCoordX * 1000
Xb = Xt
Zt = UserCoordZ * 1000
Zb = Zt
Yt = UserCoordY * 1000
Yb = Yt + Lenght
i = 0
j = 0
r = Yt + 2

DrillTable(0, 0) = "Drill ID"
DrillTable(0, 1) = "Stock Material"
DrillTable(0, 2) = "Recommended Feed"
DrillTable(0, 3) = "Recommended Speed"
DrillTable(0, 4) = "Drill Diameter"
DrillTable(0, 5) = "Drill Cutting Length"
DrillTable(0, 6) = "Drill Material"


DrillTable(1, 0) = "1"
```

```
    DrillTable(2, 0) = "2"
    DrillTable(3, 0) = "3"
    DrillTable(4, 0) = "4"
    DrillTable(5, 0) = "5"
    DrillTable(6, 0) = "6"

    For i = 1 To 6
        DrillTable(i, 1) = "Steel"
    Next i

    For i = 1 To 6
        If 1 < i < 3 Then
            DrillTable(i, 2) = "100"
        Else
            DrillTable(i, 2) = "50"
        End If
    Next i

    For i = 1 To 6
        If 1 < i < 3 Then
            DrillTable(i, 3) = "1000"
        Else
            DrillTable(i, 3) = "500"
        End If
    Next i

    For i = 1 To 6
        DrillTable(i, 4) = Str(2 * i)
    Next i

    For i = 1 To 6
        DrillTable(i, 5) = Str(5 * i)
    Next i

    For i = 1 To 6
        DrillTable(i, 6) = "Cobalt"
    Next i

    Set fso = CreateObject("Scripting.FileSystemObject")
    Dim ofile As Object
    Set ofile = fso.opentextfile("C:\Users\pmofe\Desktop\GCode.txt", ForAppending,
True)
```

```
    ofile.writeline "o0001"
    ofile.writeline "G21"
    ofile.writeline "G90"

    For i = 1 To 6
        If DrillTable(i, 4) = (Diameter * 1000) Then
            If DrillTable(i, 1) = "Steel" And DrillTable(i, 6) = "Cobalt" Then
                ofile.writeline "T0" & DrillTable(i, 0) & "0" & DrillTable(i, 0)
                DrillId = DrillTable(i, 0)
            End If
        End If
    Next i

    ofile.writeline "M06"
    ofile.writeline "G10 L2 P1 X100 Y100 Z100"
    ofile.writeline "G54 G18"

    For i = 1 To 6
        If DrillTable(i, 0) = DrillId Then
            ofile.writeline "S" & DrillTable(i, 3) & " " & "M03"
        End If
    Next i

    ofile.writeline "G00" & " " & "X" & Str(Xt) & " " & "Z" & Str(Zt)
    ofile.writeline "GOO Y" & Str(r)
    ofile.writeline "M08"

    For i = 1 To 6
        If DrillTable(i, 0) = DrillId Then
            ofile.writeline "G81" & Str(Xt) & "," & Str(Yb) & "," & Str(Zt) & "," & Str(r) & "," &
DrillTable(i, 2)
        End If
    Next i

    ofile.writeline "G00 Z" & Str(r)
    ofile.writeline "M09"
    ofile.writeline "G53 G0 X0 Y0 Z0"
    ofile.writeline "M05"
    ofile.writeline "M30"

End Sub
```

```
Dim swApp As SldWorks.SldWorks
Dim Part As ModelDoc2
Sub main()

   Set swApp = Application.SldWorks
   Set Part = swApp.ActiveDoc

   If Part Is Nothing Then
      MsgBox " You should open a document first."
      Exit Sub
   End If

   UserForm1.Show

End Sub

Public Sub CreatePocket(InputX As Double, InputY As Double, InputZ As Double, InputLength As Double, InputWidth As Double, InputDepth As Double, InputRadius As Double, InputOverlap As Double)

   Dim status As Boolean
   Dim UserCoordX As Double
   Dim UserCoordY As Double
   Dim UserCoordZ As Double
   Dim CoordX As Double
   Dim CoordY As Double
   Dim CoordZ As Double
   Dim Lenght As Double
   Dim Width As Double
```

```
Dim Depth As Double
Dim Radius As Double
Dim Overlap As Double

UserCoordX = InputX
UserCoordY = -InputZ
'UserCoordZ = InputZ
Length = InputLength
Depth = InputDepth
Width = InputWidth
Depth = InputDepth
Radius = InputRadius
Overlap = InputOverlap

If Radius * 2 > Length Or Radius * 2 > Depth Or Radius * 2 > Width Then
    MsgBox " Radius can't be greater then length or depth or width"
    Exit Sub
End If



status = Part.Extension.SelectByID2("PLANE1", "PLANE", 0, 0, 0, False, 0, Nothing, 0)

Part.SketchManager.InsertSketch True
Dim sketch As Object

Set sketch = Part.SketchManager.CreateLine(UserCoordX, UserCoordY, 0#,
UserCoordX, UserCoordY - Width, 0#)

CoordY = UserCoordY - Width
```

Set sketch = Part.SketchManager.CreateLine(UserCoordX, CoordY, 0#, UserCoordX + Length, CoordY, 0#)

CoordX = UserCoordX + Length

Set sketch = Part.SketchManager.CreateLine(CoordX, CoordY, 0#, CoordX, UserCoordY, 0#)

Set sketch = Part.SketchManager.CreateLine(CoordX, UserCoordY, 0#, UserCoordX, UserCoordY, 0#)

status = Part.Extension.SelectByID2("", "SKETCHPOINT", UserCoordX, UserCoordY, UserCoordZ, True, 0, Nothing, 0)

Set sketch = Part.SketchManager.CreateFillet(Radius, 1)

status = Part.Extension.SelectByID2("", "SKETCHPOINT", UserCoordX, CoordY, UserCoordZ, True, o, Nothing, 0)

Set sketch = Part.SketchManager.CreateFillet(Radius, 1)

status = Part.Extension.SelectByID2("", "SKETCHPOINT", CoordX, CoordY, UserCoordZ, True, 0, Nothing, 0)

Set sketch = Part.SketchManager.CreateFillet(Radius, 1)

status = Part.Extension.SelectByID2("", "SKETCHPOINT", CoordX, UserCoordY, UserCoordZ, True, 0, Nothing, 0)

Set sketch = Part.SketchManager.CreateFillet(Radius, 1)

Dim pocket As Object

Set pocket = Part.FeatureManager.FeatureCut4(True, False, False, 0, 0, Depth, 0.01, False, False, False, False, 1.74532925199433E-02, 1.74532925199433E-02, False, False, False, False, False, True, True, True, True, Flase, 0, 0, False, False)

```
Dim ToolTable(6, 6) As String
Dim ToolID As String
Dim r1 As Double
Dim Cx As Double
Dim Cy As Double
Dim Cz As Double
Dim Xb As Double
Dim Zb As Double
Dim dpc As Double
Dim Cty As Double
Dim Xt As Double
Dim Zt As Double
Dim zba As Double
Dim Depth1 As Double
Dim i As Integer
Dim j As Integer
Dim a As Double
Dim b As Double
Dim overlapf As Double
Overlap = 1 - Overlap
b = 0
a = 0
i = 0
j = 0
r1 = Radius * 1000
Cx = UserCoordX * 1000
Cy = UserCoordY * 1000
Cz = UserCoordZ * 1000
Xb = Cx + r1
```

```
Zb = Cz - r1

Cty = Cy - dpc

Xt = (Length - Radius) * 1000

Zt = (Length - Radius) * 1000

Depth1 = Depth * 1000


If Depth Mod 2 = 0 And Depth < 12 Then
    dpc = 2
Else
    If Depth Mod 2 = 0 And Depth > 12 Then
        dpc = 4
    Else
        If Depth Mod 2 <> 0 And Depth > 1 Then
            dpc = 3
        Else
           dpc = 1
        End If
    End If
End If


ToolTable(0, 0) = "Tool ID"
ToolTable(0, 1) = "Stock Material"
ToolTable(0, 2) = "Recommended Feed"
ToolTable(0, 3) = "Recommended Speed"
ToolTable(0, 4) = "Tool Diameter"
ToolTable(0, 5) = "Tool Cutting Length"
ToolTable(0, 6) = "Tool Material"
```

```
ToolTable(1, 0) = "1"

ToolTable(2, 0) = "2"

ToolTable(3, 0) = "3"

ToolTable(4, 0) = "4"

ToolTable(5, 0) = "5"

ToolTable(6, 0) = "6"

For i = 1 To 6
   ToolTable(i, 1) = "Steel"
Next i

For i = 1 To 6
   If 1 < i < 3 Then
      ToolTable(i, 2) = "100"
   Else
      ToolTable(i, 2) = "50"
   End If
Next i

For i = 1 To 6
   If 1 < i < 3 Then
      ToolTable(i, 3) = "1000"
   Else
      ToolTable(i, 3) = "500"
```

```
    End If
Next i

    ToolTable(1, 4) = "2"
    ToolTable(2, 4) = "4"
    ToolTable(3, 4) = "6"
    ToolTable(4, 4) = "10"
    ToolTable(5, 4) = "12"
    ToolTable(6, 4) = "26"

For i = 1 To 3
    ToolTable(i, 5) = Str(2)
Next i

For i = 3 To 6
    ToolTable(i, 5) = Str(3)
Next i

For i = 1 To 6
    ToolTable(i, 6) = "Carbride"
Next i

Set fso = CreateObject("Scripting.FileSystemObject")
Dim ofile As Object
Set ofile = fso.opentextfile("C:\Users\pmofe\Desktop\GCodePocket.txt", 8, True)

ofile.WriteLine "o0002"
ofile.WriteLine "G21"
ofile.WriteLine "G90"
```

```
For i = 1 To 6
   If ToolTable(i, 4) = (Radius * 1000) Then
      If ToolTable(i, 1) = "Steel" And ToolTable(i, 6) = "Carbride" Then
         ofile.WriteLine "T0" & ToolTable(i, 0) & "0" & ToolTable(i, 0)
         ToolID = ToolTable(i, 0)
      End If
   End If
Next i

ofile.WriteLine "M06"
ofile.WriteLine "G10 L2 P1 X10 Y10 Z10"
ofile.WriteLine "G54 G18"

For i = 1 To 6
   If ToolTable(i, 0) = ToolID Then
      ofile.WriteLine "S" & ToolTable(i, 3) & " " & "M03"
   End If
Next i

ofile.WriteLine "G00" & " " & "X" & Str(Xb) & " " & "Z" & Str(Zb)
ofile.WriteLine "GOO Y" & Str(Cy)

ofile.WriteLine "M08"

Zb = Cz - r1
zba = Abs(Zb)
For a = 0 To Depth1
```

```
        Zb = Cz - r1
        ofile.WriteLine "G01" & " " & "Y" & Str(Cty)
        ofile.WriteLine "G01" & " " & "X" & Str(Xt) & " " & "Z" & Str(Zb)
        Zb = Zb - dpc * Overlap
        'ofile.WriteLine "G03" & " " & "X" & Str(Xt) & " " & "Z" & Str(Zb) & " " & "R" &
Str(r1)
        Zb = Zb - dpc * Overlap
        ofile.WriteLine "G01" & " " & "X" & Str(Xb) & " " & "Z" & Str(Zb)
        Zb = Zb - dpc * Overlap
        ofile.WriteLine "G01" & " " & "X" & Str(Xb) & " " & "Z" & Str(Zb)
        b = 0
        For b = 0 To 50
          If Abs(Zb) + (Radius * 1000) >= Width * 1000 Then
            'ofile.WriteLine "G03" & " " & "X" & Str(Xt) & " " & "Z" & Str(Zb) & " " & "R"
& Str(r1)
            ofile.WriteLine "G01" & " " & "X" & Str(Xb) & " " & "Z" & Str(Zb)
            b = 50
         Else

            ofile.WriteLine "G01" & " " & "X" & Str(Xt) & " " & "Z" & Str(Zb)
            Zb = Zb - dpc * Overlap
            ofile.WriteLine "G01" & " " & "X" & Str(Xt) & " " & "Z" & Str(Zb)

            ofile.WriteLine "G01" & " " & "X" & Str(Xb) & " " & "Z" & Str(Zb)
            Zb = Zb - dpc * Overlap
            ofile.WriteLine "G01" & " " & "X" & Str(Xb) & " " & "Z" & Str(Zb)
            Zb = Zb - dpc * Overlap
            ofile.WriteLine "G01" & " " & "X" & Str(Xt) & " " & "Z" & Str(Zb)
            b = b + 1
          End If
```

```
            Next b

        Cty = Cty - dpc
        a = a + dpc
        Zb = Cz - r1

        ofile.WriteLine "G00" & " " & "X" & Str(Xb) & " " & "Z" & Str(Zb)
    Next a
    Cty = Depth - dpc
    ofile.WriteLine "G00" & " " & "X" & Str(Cx) & " " & "Y" & " " & Str(Cty) & " " & " " & "Z"
& " " & Str(Cz)
    ofile.WriteLine "G01" & " " & "X" & Str(Xt) & " " & "Z" & " " & Str(Zb)
    ofile.WriteLine "G03" & " " & "X" & Str(Xt) & " " & "Z" & Str(Zb) & " " & "R" & Str(r1)
    ofile.WriteLine "G01" & " " & "X" & Str(Xt) & " " & "Z" & Str(Zt)
    Xt = Xt - 2
    ofile.WriteLine "G03" & " " & "X" & Str(Xt) & " " & "Z" & Str(Zb) & " " & "R" & Str(r1)
    ofile.WriteLine "G01" & " " & "X" & Str(Xb) & " " & "Z" & Str(Zb)
    ofile.WriteLine "G03" & " " & "X" & Str(Xb) & " " & "Z" & Str(Zb) & " " & "R" & Str(r1)
    ofile.WriteLine "G01" & " " & "X" & Str(Cx) & " " & "Z" & Str(Cz)
    ofile.WriteLine "G00 Y" & Str(Cy)
    ofile.WriteLine "M09"
    ofile.WriteLine "G53 G0 X0 Y0 Z0"
    ofile.WriteLine "M05"
    ofile.WriteLine "M30"
End Sub
Private Sub CommandButton1_Click()
    Dim X As Double
    Dim Y As Double
    Dim Z As Double
```

```
    Dim Length As Double
    Dim Depth As Double
    Dim Width As Double
    Dim Radius As Double
    Dim Overlap As Double

    X = Val(TextBox1.Text) / 1000
    Y = Val(TextBox2.Text) / 1000
    Z = Val(TextBox3.Text) / 1000
    Length = Val(TextBox4.Text) / 1000
    Depth = Val(TextBox5.Text) / 1000
    Width = Val(TextBox6.Text) / 1000
    Radius = Val(TextBox7.Text) / 1000
    Overlap = Val(TextBox8.Text) / 100

    Call CreatePocket(X, Y, Z, Length, Depth, Width, Radius, Overlap)

End Sub

Dim swApp As SldWorks.SldWorks
Dim Part As ModelDoc2
Dim boolstatus As Boolean

Sub main()

    Set swApp = Application.SldWorks
    Set Part = swApp.ActiveDoc

    If Part Is Nothing Then
```

```
  MsgBox " You should open a document first."
  Exit Sub
  End If
  UserForm1.Show

End Sub
 Public Sub CreateSlot(InputUserCoordX1 As Double, InputUserCoordY1 As Double, InputUserCoordZ1 As Double, InputUserCoordX2 As Double, InputUserCoordY2 As Double, InputUserCoordZ2 As Double, InputUserCoordX3 As Double, InputUserCoordY3 As Double, InputUserCoordZ3 As Double, InputWidth As Double, InputDepth As Double)

  Dim status As Boolean
  Dim CoordX1 As Double
  Dim CoordY1 As Double
  Dim CoordZ1 As Double
  Dim CoordX2 As Double
  Dim CoordY2 As Double
  Dim CoordZ2 As Double
  Dim CoordX3 As Double
  Dim CoordY3 As Double
  Dim CoordZ3 As Double

  Dim Width As Double
  Dim Depth As Double

  CoordX1 = InputUserCoordX1
  CoordX2 = InputUserCoordX2
  CoordY1 = InputUserCoordY1
  CoordY2 = InputUserCoordY2
  CoordZ1 = InputUserCoordZ1
```

```
CoordZ2 = InputUserCoordZ2

CoordX3 = InputUserCoordX3

CoordY3 = InputUserCoordY3

CoordZ3 = InputUserCoordZ3


Depth = InputDepth

Width = InputWidth


boolstatus = Part.Extension.SelectByID2("PLANE3", "PLANE", 0, 0, 0, False, 0, Nothing, 0)


Part.SketchManager.InsertSketch True


Dim SketchSlot As Object


Set SketchSlot = Part.SketchManager.CreateSketchSlot(swSketchSlotCreationType_e.swSketchSlotCreationType_line, swSketchSlotLengthType_e.swSketchSlotLengthType_CenterCenter, Width, CoordX1, -CoordZ1, 0, CoordX2, -CoordZ2, 0, 0, 0, 0, 1, False)


'Set mySketchSlot = Part.SketchManager.CreateSketchSlot(swSketchSlotCreationType_e.swSketchSlotCreationType_line, swSketchSlotLengthType_e.swSketchSlotLengthType_CenterCenter, 0.298608817942409, 0.100032498575075, -0.198926375567055, 0, 0.100032498575075, 4.38482370753661E-02, 0, 0, 0, 0, 1, False)

Dim Slot As Object

Set Slot = Part.FeatureManager.FeatureCut4(True, False, False, swEndCondBlind, 0, Depth, 0.01, False, False, False, False, 1.74532925199433E-02, 1.74532925199433E-02, False, False, False, False, False, True, True, True, True, False, 0, 0, False, False)


Part.SelectionManager.EnableContourSelection = False
```

```
Dim ToolTable(6, 6) As String
Dim ToolID As String
Dim Depth1 As Double
Dim a As Integer
Dim dpc As Double
Dim GCoordX1 As Double
Dim GCoordX2 As Double
Dim GCoordX3 As Double
Dim GCoordY1 As Double
Dim GCoordY2 As Double
Dim GCoordY3 As Double
Dim GCoordZ1 As Double
Dim GCoordZ2 As Double
Dim GCoordZ3 As Double
Dim GcoordK As Double
Dim CoordF1 As Double
Dim CoordF2 As Double

GCoordX1 = CoordX1 * 1000
GCoordX2 = CoordX2 * 1000
GCoordX3 = CoordX3 * 1000
GCoordY1 = CoordY1 * 1000
GCoordY2 = CoordY2 * 1000
GCoordY3 = CoordY3 * 1000
GCoordZ1 = CoordZ1 * 1000
GCoordZ2 = CoordZ2 * 1000
GCoordZ3 = CoordZ3 * 1000
CoordF1 = Abs(CoordZ3 * 1000) - Width * 1000
CoordF2 = Abs(CoordZ3 * 1000) + Width * 1000
```

```
Depth1 = Depth * 1000
a = 0

If Depth Mod 2 = 0 And Depth < 12 Then
   dpc = 2
Else
   If Depth Mod 2 = 0 And Depth > 12 Then
      dpc = 4
   Else
      If Depth Mod 2 <> 0 And Depth > 1 Then
         dpc = 3
      Else
        dpc = 1
      End If
   End If
End If

ToolTable(0, 0) = "Tool ID"
ToolTable(0, 1) = "Stock Material"
ToolTable(0, 2) = "Recommended Feed"
ToolTable(0, 3) = "Recommended Speed"
ToolTable(0, 4) = "Tool Diameter"
ToolTable(0, 5) = "Tool Cutting Length"
ToolTable(0, 6) = "Tool Material"



ToolTable(1, 0) = "1"


ToolTable(2, 0) = "2"
```

```
ToolTable(3, 0) = "3"

ToolTable(4, 0) = "4"

ToolTable(5, 0) = "5"

ToolTable(6, 0) = "6"

For i = 1 To 6
   ToolTable(i, 1) = "Steel"
Next i

For i = 1 To 6
   If 1 < i < 3 Then
      ToolTable(i, 2) = "100"
   Else
      ToolTable(i, 2) = "50"
   End If
Next i

For i = 1 To 6
   If 1 < i < 3 Then
      ToolTable(i, 3) = "1000"
   Else
      ToolTable(i, 3) = "500"
   End If
Next i
```

```
    ToolTable(1, 4) = "2"
    ToolTable(2, 4) = "4"
    ToolTable(3, 4) = "6"
    ToolTable(4, 4) = "10"
    ToolTable(5, 4) = "12"
    ToolTable(6, 4) = "26"



For i = 1 To 3
    ToolTable(i, 5) = Str(2)
Next i

For i = 3 To 6
    ToolTable(i, 5) = Str(3)
Next i

For i = 1 To 6
    ToolTable(i, 6) = "Carbride"
Next i



Set fso = CreateObject("Scripting.FileSystemObject")
Dim ofile As Object
Set ofile = fso.opentextfile("C:\Users\pmofe\Desktop\GCodeSlot.txt", 8, True)

ofile.writeline "o0003"
ofile.writeline "G21"
```

```
ofile.writeline "G90"


For i = 1 To 6
   If ToolTable(i, 4) = (Radius * 1000) Then
      If ToolTable(i, 1) = "Steel" And ToolTable(i, 6) = "Carbride" Then
         ofile.writeline "T0" & ToolTable(i, 0) & "0" & ToolTable(i, 0)
         ToolID = ToolTable(i, 0)
      End If
   End If
Next i


ofile.writeline "M06"
ofile.writeline "G10 L2 P1 X10 Y10 Z10"
ofile.writeline "G54 G18"
For i = 1 To 6
   If ToolTable(i, 0) = ToolID Then
      ofile.writeline "S" & ToolTable(i, 3) & " " & "M03"
   End If
Next i
ofile.writeline "M08"


ofile.writeline "G00 X0 Z0"
ofile.writeline "G00 Y0"


For a = 0 To Depth1
   GCoordZ3 = CoordZ3
   GCoordZ3 = GCoordZ3 - 1
   ofile.writeline "G01" & " " & "X" & Str(GCoordX3) & " " & "Z" & Str(GCoordZ3)
   GcoordK = GCoordZ3 / 2
```

```
    ofile.writeline "G03" & " " & "X" & Str(GCoordX1) & " " & "Z" & Str(GCoordZ1) & " " & "I" & Str(GCoordX1) & " " & "K" & Str(GcoordK)

    CoordZ3 = GCoordZ3 - 1

    ofile.writeline "G01" & " " & "X" & Str(GCoordX2) & " " & "Z" & Str(GCoordZ2)

    GcoordK = GCoordZ3 / 2

    ofile.writeline "G02" & " " & "X" & Str(GCoordX2) & " " & "Z" & Str(GCoordZ3) & " " & "I" & Str(GCoordX2) & " " & "K" & Str(GcoordK)

    GCoordZ3 = GCoordZ3 - 1

    ofile.writeline "G01" & " " & "X" & Str(GCoordX1) & " " & "Z" & Str(GCoordZ3)


    If GCoordZ3 < CoordF1 Or GCoordZ3 < CoordF2 Then


      GcoordK = GCoordZ3 / 2

      ofile.writeline "G03" & " " & "X" & Str(GCoordX2) & " " & "Z" & Str(GCoordZ3) & " " & "I" & Str(GCoordX2) & " " & "K" & Str(GcoordK)

      CoordZ3 = GCoordZ3 - 1

      ofile.writeline "G01" & " " & "X" & Str(GCoordX2) & " " & "Z" & Str(GCoordZ2)

      GcoordK = GCoordZ3 / 2

      ofile.writeline "G02" & " " & "X" & Str(GCoordX2) & " " & "Z" & Str(GCoordZ3) & " " & "I" & Str(GCoordX2) & " " & "K" & Str(GcoordK)

      GCoordZ3 = GCoordZ3 - 1

      ofile.writeline "G01" & " " & "X" & Str(GCoordX1) & " " & "Z" & Str(GCoordZ3)


    End If


    a = a + dpc


  Next a


  ofile.writeline "G00 Y" & Str(GCoordY2)
```

```
    ofile.writeline "M09"
    ofile.writeline "G53 G0 X0 Y0 Z0"
    ofile.writeline "M05"
    ofile.writeline "M30"
End Sub

Private Sub CommandButton1_Click()
    Dim X1 As Double
    Dim Y1 As Double
    Dim Z1 As Double
    Dim X2 As Double
    Dim Y2 As Double
    Dim Z2 As Double
    Dim X3 As Double
    Dim Y3 As Double
    Dim Z3 As Double
    Dim Depth As Double
    Dim Width As Double


    X1 = Val(TextBox1.Text) / 1000
    Y1 = Val(TextBox2.Text) / 1000
    Z1 = Val(TextBox3.Text) / 1000
    X2 = Val(TextBox4.Text) / 1000
    Y2 = Val(TextBox5.Text) / 1000
    Z2 = Val(TextBox6.Text) / 1000
    X3 = Val(TextBox9.Text) / 1000
    Y3 = Val(TextBox10.Text) / 1000
    Z3 = Val(TextBox11.Text) / 1000
```

```
    Width = Val(TextBox7.Text) / 1000

    Depth = Val(TextBox8.Text) / 1000



    Call CreateSlot(X1, Y1, Z1, X2, Y2, Z2, X3, Y3, Z3, Width, Depth)


End Sub
```

Annex 2

```
G21
G90
T1 M06

G54
G17
G00 X 60 Y 15
G00 Z 65
M08
G01 Z55
G01 X 100 Y15
G01 Y10
G01 X100
G01 X60
G01 Y5
G01 X100
G01 X60
G01 Y0
G01 X100
G01 X60
G01 Y-5
G01 X100
G01 X60
G01 Y-10
G01 X100
G01 X60
G01 Y-15
G01 X100
G01 X60
G00 Z50
G01 X60 Y15
G01 X 100 Y15
G01 Y10
G01 X100
G01 X60
G01 Y5
G01 X100
G01 X60
G01 Y0
G01 X100
G01 X60
```

```
G01 Y-5
G01 X100
G01 X60
G01 Y-10
G01 X100
G01 X60
G01 Y-15
G01 X100
G01 X60

T4 M6
G00 X100 Y15
G00 Z50
G01 X65 Y15
G03 X60 Y10 R5
G01 X60 Y-10
G03 X65 Y-15 R5
G01 X100 Y-15

T1 M6
G00 Z65
G00 X-60 Y15
G01 Z55
G01 X-100 Y15
G01 Y10
G01 X-100
G01 X-60
G01 Y5
G01 X-100
G01 X-60
G01 Y0
G01 X-100
G01 X-60
G01 Y-5
G01 X-100
G01 X-60
G01 Y-10
G01 X-100
G01 X-60
G01 Y-15
G01 X-100
G01 X-60

G00 Z50
G01 X-60 Y15
G01 X -100 Y15
G01 Y10
G01 X-100
G01 X-60
G01 Y5
G01 X-100
G01 X-60
G01 Y0
G01 X-100
G01 X-60
G01 Y-5
G01 X-100
G01 X-60
G01 Y-10
G01 X-100
```

```
G01 X-60
G01 Y-15
G01 X-100
G01 X-60

G00 Z65
G00 X15 Y-80
G01 Z55
G01 Y-50
G01 X10
G01 Y-80
G01 X5
G01 Y-50
G01 X0
G01 Y-80
G01 X-5
G01 Y-50
G01 X -10
G01 Y-80
G01 X-15
G01 Y-50

G00 Z65
G00 X15 Y-80
G01 Z50
G01 Y-50
G01 X10
G01 Y-80
G01 X5
G01 Y-50
G01 X0
G01 Y-80
G01 X-5
G01 Y-50
G01 X -10
G01 Y-80
G01 X-15
G01 Y-50


G00 Z65
G00 X15 Y80
G01 Z55
G01 Y50
G01 X10
G01 Y80
G01 X5
G01 Y50
G01 X0
G01 Y80
G01 X-5
G01 Y50
G01 X -10
G01 Y80
G01 X-15
G01 Y50

G00 Z65
G00 X15 Y80
G01 Z50
```

```
G01 Y50
G01 X10
G01 Y80
G01 X5
G01 Y50
G01 X0
G01 Y80
G01 X-5
G01 Y50
G01 X -10
G01 Y80
G01 X-15
G01 Y50
T3 M6
G00 Z65
G00 X40 Y25
G01 Z55
G01 Y-25
G01 X30
G01 Y25
G01 X20
G01 Y-25
G01 X10
G01 Y25
G01 X0
G01 Y-25
G01 X-10
G01 Y25
G01 X-20
G01 Y-25
G01 X-30
G01 Y25
G01 X-40
G01 Y-25
G00 Z65
G00 X40 Y25
G01 Z50
G01 Y-25
G01 X30
G01 Y25
G01 X20
G01 Y-25
G01 X10
G01 Y25
G01 X0
G01 Y-25
G01 X-10
G01 Y25
G01 X-20
G01 Y-25
G01 X-30
G01 Y25
G01 X-40
G01 Y-25
G00 Z65

T4M6

G00 X40 Y25
G00 Z50
```

```
G01 Y-25
G01 X-40
G01 Y25
G01 X40
G00 Z65

T2 M6
G00 X-75 Y0
G00 Z65
G81 Z30 R1 M03 M8
G00 Z65
G00 X75 Y0
G81 Z30 R1 M03 M8
G00 Z65
G00 X0 Y-65
G81 Z30 R1 M03 M8
G00 Z65
G00 X0 Y65
G81 Z30 R1 M03 M8
G00 Z65
G80
T1 M6
G00 Z65
G00 X-2 Y-14
G01 Z35
G03 X2 Y-14 R4
G01 X2 Y14
G03 X-2 Y14 R4
G01 X-2 Y-14
G00 Z65
G00 X0 Y0

G00 Z65 F70
G00 X-27 Y-14
G01 Z35
G03 X-23 Y-14 R4
G01 X-23 Y14
G03 X-27 Y14 R4
G01 X-27 Y-14
G00 Z65
G00 X0 Y0

G00 Z65 F70
G00 X23 Y-14
G01 Z35
G03 X27 Y-14 R4
G01 X27 Y14
G03 X23 Y14 R4
G01 X23 Y-14
G00 Z65
G00 X0 Y0

G00 Z65
G00 X53 Y-55
G01 Z55
G03 X55 Y-53 R4
G01 X76 Y-74
G03 X74 Y-76 R4
G01 X53 Y-55
G00 Z65
```

```
G00 X0 Y0

G00 Z65
G00 X72 Y-55
G01 Z50
G03 X75 Y-58 R4
G01 X58 Y-75
G03 X55 Y-72 R4
G01 X55 Y-72
G00 Z65
G00 X0 Y0

G00 Z65
G00 X-56 Y-53
G01 Z55
G03 X-53 Y-56 R4
G01 X-74 Y-77
G03 X-77 Y-74 R4
G01 X-56 Y-53
G00 Z65
G00 X0 Y0

G00 Z65
G00 X-77 Y74
G01 Z55
G03 X-74 Y77 R4
G01 X-53 Y56
G03 X-56 Y53 R4
G01 X-77 Y74
G00 Z65
G00 X0 Y0

G00 Z65
G00 X-58 Y75
G01 Z50
G03 X-55 Y72 R4
G01 X-72 Y55
G03 X-75 Y58 R4
G01 X-58 Y75
G00 Z65
G00 X0 Y0

G00 Z65
G00 X75 Y77
G01 Z55
G03 X77 Y74 R4
G01 X56 Y53
G03 X53 Y56 R4
G01 X75 Y77
G00 Z65
G00 X0 Y0

M09
M05
M30
```