Universidade de Aveiro
2021

**Álvaro Miguel
Cruz Francisco**

Design Algoritmico de um produto baseado em dados do consumidor

Algorithmic design of a product based on customer data

Universidade de Aveiro
2021

Álvaro Miguel Cruz Francisco

Design Algoritmico de um produto baseado em dados do consumidor

Algorithmic design of a product based on customer data

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Mecânica, realizada sob orientação científica de João Alexandre Dias de Oliveira, Professor Auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro, e de Victor Fernando Santos Neto, Professor Auxiliar em Regime Laboral do Departamento de Engenharia Mecânica da Universidade de Aveiro.

Este trabalho foi apoiado pelos projetos UIDB / 00481/2020 e UIDP/00481/2020 - FCT - Fundação para a Ciência e a Tecnologia; e CENTRO-01-0145-FEDER-022083 - Programa Operacional Regional do Centro Portugal (Centro2020), no âmbito do Acordo de Parceria PORTUGAL 2020, através do Fundo Europeu de Desenvolvimento Regional.

o júri / the jury

presidente / president

Prof. Doutor Ricardo José Alves de Sousa
Professor Auxiliar C/ Agregação, Universidade de Aveiro

Doutor João Filipe Moreira Caseiro
Investigador, Centimfe - Centro Tecnológico da Indústria dos Moldes, Ferramentas Especiais e Plásticos

Prof. Doutor Victor Fernando Santos Neto
Professor Auxiliar em Regime Laboral, Universidade de Aveiro (co-orientador)

agradecimentos / acknowledgements

Queria começar os agradecimentos ao Prof. João Oliveira e ao Prof. Victor Neto por me darem a oportunidade para desenvolver este trabalho numa das áreas de Engenharia Mecânica que eu demonstro maior interesse, após ter ficado sem o tema de dissertação atribuído por razões burocráticas. E quero agradecer as horas e a paciência que os professores disponibilizaram para me apoiar nesta etapa final da minha vida académica.

Agradeço aos meus amigos, Simão, Miguel, Tiago, Ricardo, Daniel e Samuel que me aturaram horas a fio em chamadas de voz para explicar as dores de cabeça que esta dissertação causou. Apoiaram todos os meus passos e deram imenso *feedback* dos desenvolvimentos do meu trabalho e sugestões de como o podia melhorar.

Quero também agradecer às pessoas que me acompanharam durante o meu percurso académico, nos bons e nos maus momentos e (quase) nunca dizerem que não a um jantar ou a um copo. Dedico este trabalho ao Afonso, ao Amílcar e ao Martim que estiveram sempre lá para me apoiar nestes maravilhos anos da minha vida. Fizeram-me crescer e levarei para sempre as nossas histórias.

Dedico este trabalho à minha namorada, Joana. Ela esteve sempre pronta para me apoiar em qualquer momento da minha vida, mais ainda durante a elaboração desta dissertação. Fez-me acreditar que tudo é possível e o segredo é nunca desistir. Obrigado por tudo!

Finalmente, mas não menos importante: aos meus pais e família. Obrigado por acreditarem em mim e naquilo que sou realmente capaz de fazer. Obrigado por me darem a força nos momentos que mais precisei. Obrigado pelo apoio a perseguir os meus sonhos. Não há palavras que expressem a gratidão que sinto por serem uma força constante no meu dia-a-dia.

A todos, obrigado!

keywords

Parametric Design, Generative Design, Algorithmic Design, Mass Customization, Hybrid Manufacturing

abstract

There is a growing trend of using computers creatively in order to enrich the design process. There are three Computational Design techniques that stand-out: Parametric Design, Generative Design and Algorithmic Design. This dissertation intends to test the viability of using these techniques in a context of product development. These techniques show tremendous potential for products that can be customizable by consumers, exploring the combination of various manufacturing methods. To achieve these goals a case study with customization potential and the ability to test algorithmic design techniques has been selected. The results originate from 2 approaches: a generative approach and an algorithmic approach, with each having different evaluation methods. The generative approach is able to explore a solution search space and compares the generated curvatures, whilst the algorithmic approach takes advantage of rapid prototyping principles. The performance indicators for the case study's conception stage using CD techniques are very positive, but the production stage needs more research.

palavras-chave

Design Paramétrico, Design Generativo, Design Algorítmico, Customização em Massa, Fabrico Híbrido

resumo

Cada vez mais os computadores são usados de forma criativa para aprofundar o processo de design. Existem três técnicas de design computacional que merecem destaque: design paramétrico, design generativo e design algorítmico. Este trabalho tem como intuito testar a viabilidade do uso destas técnicas num contexto de desenvolvimento de produto. Estas técnicas demonstram um grande potencial para produtos que possam ser customizáveis, explorando a combinação de diferentes métodos de produção. Para isso foi selecionado um caso de estudo com potencial de customização onde seja possível testar a aplicação das técnicas de design algorítmico. Os resultados provêm de 2 abordagens: uma abordagem generativa e uma abordagem algorítmica, com cada abordagem a ter um método de avaliação de resultados. A abordagem generativa varre um espaço de soluções e compara as curvaturas geradas enquanto a abordagem algorítmica aproveita os princípios de prototipagem rápida. Os indicadores obtidos para a fase de conceção do caso de estudo usando as técnicas de CD foram positivos, no entanto a fase da produção necessita mais investigação.

Contents

I Introduction and Background **1**

1 Introduction **3**

1.1 Background . . . 3

1.2 Objectives . . . 4

1.3 Structure . . . 4

2 Concepts and Terminology **7**

2.1 Engineering and Computational Design . . . 7

2.1.1 Parametric Design . . . 8

2.1.2 Generative Design . . . 8

2.1.3 Algorithmic Design . . . 9

2.1.4 Computational Design Tools . . . 10

2.1.5 Concept Overview . . . 11

2.2 Mass Customization and Mass Personalization . . . 12

2.2.1 Industry 4.0 . . . 13

2.2.2 Additive Manufacturing . . . 14

2.2.3 Hybrid Manufacturing . . . 16

II Implementation **17**

3 Case Study **19**

3.1 Framework . . . 19

3.2 Benchmarking . . . 19

4 MUS – Mouse Unravelling Software **25**

4.1 Generative Approach . . . 29

4.1.1 Symmetrical Typology . . . 31

4.1.2 Asymmetrical Typology . . . 38

4.2 Algorithmic Approach . . . 48

4.3 Prototyping . . . 54

4.4 Modularity . . . 60

5 Final remarks **63**
5.1 Main takeaways 63
5.2 Future Works 66

Bibliography **66**

A Code Examples **73**
A.1 Curvature Testing 73
A.2 Modelling Strategy 75

List of Tables

3.1 Comparative measurements between products of the same family. 21

4.1 Default measurements used for the generative process: (a) Symmetrical typology; (b) Asymmetrical typology. 29
4.2 The measurement intervals used for the generative process. 30
4.3 The default measurement used for the generative process of the asymmetrical shape 55

Intentionally blank page.

List of Figures

1.1 Road map to be achieved with this dissertation 4

2.1 Fields of knowledge that make up Engineering Design (adapted from [Montana-Hoyos and Lemaitre 2011]). 7
2.2 CD domain and PD, GD and AD space. 11
2.3 Cost per part using traditional manufacturing methods compared to AM technologies [Tosello *et al.* 2019] 14
2.4 Working principle of a FFF machine [Jerez-Mesa *et al.* 2016] 15

3.1 The rise of gaming during the pandemic [SteamDB 2021]. 20
3.2 Increase of the number of users in a mouse related community in social media [Stats 2021]. 21
3.3 Comparison of the types of grip: (a) Palm Grip; (b) Claw Grip; (c) Fingertip Grip. [GamingGem 2020] 22
3.4 Relevant variables used in mouse design [Razer 2021]. 23
3.5 Market examples of customization 24

4.1 MUS development flowchart 26
4.2 Mice replicas done with with the API 27
4.3 Curvature comparison between a spline constructed in Fusion and other constructed using NURBS-Python. 28
4.4 User interface to generate the multiple mice using a command prompt. . . 29
4.5 Diagram with the spline nomenclature. 30
4.6 Visual comparison of mouse length difference between “Mouse 1” and “Mouse 10”. 31
4.7 Comparative curvature of mice length between “Mouse 1” and “Mouse 10”. 32
4.8 Representation of the rolled edge that caused the anomaly in the plots. . 33
4.9 Comparative curvatures of the generated hump positions for the symmetrical typology along side a benchmark. 33
4.10 Benchmark chosen for comparison with the generated mice [Glorious 2021]. 34
4.11 Visual representation of the variation of hump positions between “Mouse 1” and “Mouse 10”. 34
4.12 Curvatures of the base spline when varying the hump position. 34
4.13 Visual representation of the impact on the base splines by moving the hump position. 35
4.14 Front and top views of the 3 different variations of hump heights. 35
4.15 Different relevant curvatures of the hump heights, alongside with the selected benchmark. 36
4.16 Bottom profiles when generating the front widths. 36

4.17 Curvatures of the base spline when varying the front width. 37
4.18 Curvatures of generated hump heights when varying the front widths. . . 37
4.19 Hump heights affected by the variation of front widths. 38
4.20 Bottom profiles when generating the grip widths widths. 38
4.21 Curvatures of the bottom spline when varying grip widths. 39
4.22 Curvature maps of the hump evolution when varying grip widths. 39
4.23 Hump curvatures when varying grip widths. 40
4.24 Comparison of curvatures on the asymmetrical bottom spline when varying the hump position. 41
4.25 Visual representation of the impact on the base splines by moving the hump position. 41
4.26 Curvature of the mice lengths when varying hump position compared to a benchmark. 42
4.27 Benchmark chosen to compare the curvatures between it and the generated mice [Zowie 2021]. 43
4.28 Comparison between of hump positions between “Mouse 1” and “Mouse 10”. 43
4.29 Curvature comparison of the hump when increasing the hump height. . . 43
4.30 Front and top views to compare some of the effects of increasing the hump height. 44
4.31 Curvatures of the base spline of the asymmetrical mice when varying the front width. 44
4.32 Comparison of curvatures of the base spline when varying the front width of the asymmetrical mice. 45
4.33 Curvature evolution of the humps when varying front widths. 45
4.34 Curvature maps of the humps when varying front widths. 46
4.35 Comparison of curvatures of the bottom spline when varying the grip widths of the asymmetrical mice. 46
4.36 Curvature evolution of the humps when varying grip widths. 47
4.37 Curvature maps of the humps when varying grip widths. 47
4.38 Final Design 49
4.39 Revised hump for the asymmetrical mice 49
4.40 The UI for the current deployment. 50
4.41 The UI for the current deployment. 51
4.42 Top shell sketch. 51
4.43 Sketch to build the second surface. 52
4.44 Representation of the sketch to obtain the asymmetrical side surface. . . . 52
4.45 Creation of the front surface and its stitching to the rest of the body. . . . 53
4.46 Creation of the patch that creates the solid body. 53
4.47 Section view of the hollow body. 53
4.48 Other mouse shapes 54
4.49 Consumer hand size that inspired the prototypes. 55
4.50 Shape 1 56
4.51 Shape 2 56
4.52 Shape 3 57
4.53 Shape 4 57
4.54 Hump Comparison 58

4.55 Model of a mouse for fingertip grip using the third stage of deployment. . 58
4.56 Asymmetrical mouse done with the third stage of deployment. 59
4.57 Mouse geared towards a claw grip user. 59
4.58 Printing errors 60
4.59 Module for mass customization 61

Intentionally blank page.

Part I

Introduction and Background

Chapter 1

Introduction

1.1 Background

Nowadays, Computer Aided Design (CAD) and Engineering (CAE) are common assets in a mechanical engineering environment. These types of tools allow the communication of ideas and the capability of understanding the world virtually, but these ideas are limited to the creativity of the designer/engineer. Currently, there is a gradual shift from modelling parametric solids (implicit modelling) to the freedom provided by modelling surfaces (explicit modelling). This freedom provided by explicit modelling allowed for the designers/engineers to expand their horizons, yet they still remain somewhat limited.

To fight this limitation, designers have been employing the creative use of computers in the design process using different techniques. One technique is the use of integrated environments to model and evaluate the design at the same time and react accordingly, changing parameters to meet requirements. Other is the use of generative models that creates alternate solutions based on hierarchical solutions. Finally, the third alternative is the use of algorithms to enhance the exploration. These three alternatives spawned terms to describe them: Parametric Design (PD), Generative Design (GD) and Algorithmic Design (AD). These terms can overlap themselves in many senses and may cause confusion amongst them.

The democratization of additive manufacturing allowed for these types of design subjects to flourish. The possibility to manufacture complex shapes justified the more prevalent use of explicit modelling. An example of design enhancement is the use of topology optimization. This tool translated itself into the creation of parts that are more interesting, do not compromise on functionality and allow to economize on materials. Not only that, but the proliferation of 3D printers for domestic use meant that consumers are able to design and manufacture objects tailor-made to their specific needs, many times leading common items onto a whole new level of customization, creating a market for mass customized products. All of these factors contribute to the existence of smart production systems revolving around Industry 4.0, allowing to merge traditional manufacturing techniques with contemporary methods in an hybrid manufacturing perspective, offering depth to manufacturing methods.

This works stands in the brink of a 3D modelling revolution in engineering, taking computational resources and expand a designers/engineers creativity beyond what they can envision. To help contextualize and demonstrate these ideas, the title of this dissertation can be deconstructed into a series of questions that serve as road map of this work, as shown in Figure 1.1.

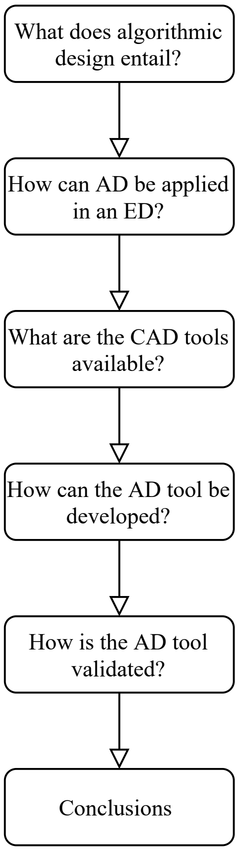


Figure 1.1: Road map to be achieved with this dissertation

1.2 Objectives

The main goal of this work is the exploration of algorithmic design as a tool to develop and customize a specific product to match customer specifications. This tool will focus on a specific market segment with customization potential and explore the manufacturing processes that suit the intended level of customization.

1.3 Structure

This dissertation is divided in three major parts: Part 1 deals with the background and the theoretical concepts that this work is based upon; The implementation presents the case study used, the tool developed and its evaluation.

Part 1 is divided in two chapters: Introduction, comprised of the background where this work stands, the objectives to be achieved and the structure of the thesis; Concepts and Terminology explains the theoretical concepts that serve as basis for this work. The topics of these chapters include an elaboration upon the definition of Design, the various subjects of design and how they can be implemented. The other topic delves into the manufacturing aspect of this work.

The implementation is divided three chapters: Case Study is where the background of the case study is presented alongside a few relevant benchmarks for comparison and contextualization of the potential of customization in this market segment. The second chapter of this part is the presentation of MUS – Mouse Unveiling Software, the tool based in generative and algorithmic design that is used to build the mice.

The final remarks of this work contains the main takeaways of this learning experience. The main takeaways also includes a topic that discusses future works and future research that should be done within this thematic.

Intentionally blank page.

Chapter 2

Concepts and Terminology

2.1 Engineering and Computational Design

The term "design" is coated in ambiguity. When centralizing "design" in the context of product development, there are still remnants of fragmented definitions and subjects. Many of the areas present on this subject are tangent to what Mechanical Engineering encompasses, brought closer together by the process. This underlying interdisciplinarity between scientific and humanistic areas when searching for technical solutions for problems that are inherently human contextualizes Engineering Design. Figure 2.1 clearly shows this dynamic between fields of knowledge and where the scope of this work is located.

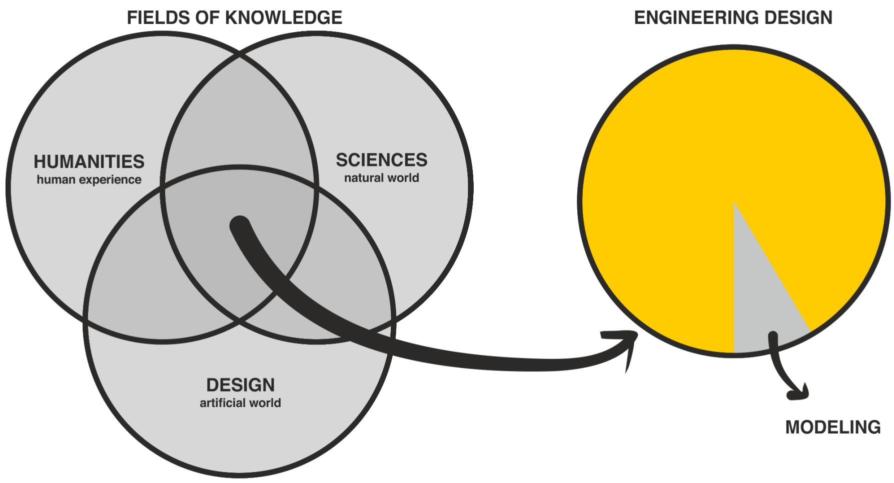


Figure 2.1: Fields of knowledge that make up Engineering Design (adapted from [Montana-Hoyos and Lemaitre 2011]).

Engineering Design can be defined as the process from idea to execution. The formalizing of an idea or a concept into tangible information to be converted in a product intended for production. As the complexity of developed products tend to grow, the need for interdisciplinarity and collaborative workflows also tend to increase, meaning that the information must be transmitted through the use of computers.

Computational Design (CD) uses computers to execute Engineering Design. CD can include everything from low-level processes, like sketches in Microsoft Paint, to high level evaluations using Finite Element Analysis (FEA) or other similar tools. Nowadays there is a push from software developers to evolve Computer Aided Design (CAD) into something more. No longer is CAD used just for modelling, to communicate an object into a three dimensional space, CAD starts to be transverse to Computer Aided Engineering (CAE) and Computer Aided Manufacturing (CAM), making it consistent with the definition presented earlier, that the term "design" encompasses the entire process.

These small revolutions of CD led to the creation of terms that led to the enhancement of Engineering Design. Parametric Design, Generative Design and Algorithmic Design are subjects that facilitate the design process, be it during modelling, material selection [Cavallini *et al.* 2013] or even improve the manufacturing process [Dávila *et al.* 2021]. These subjects are often mislabeled and can come across as ambiguous and it becomes necessary to clearly distinguish them to further contextualize this work.

2.1.1 Parametric Design

In the 1940's a new way of thinking about architectural design was born, when Luigi Moretti coined the term "Parametric Architecture" trying to create a symbiotic relationship between parametric equations and architectural design [Frazer 2016]. Nowadays, PD embraces many fields of application, which can cause general confusion between definitions. When applying PD to modelling, it can be considered as the relationships geometric and dimensional constrains between parameters [Szalapaj 2013]. In a more generalized sense PD can be the algorithmic thinking that expresses the parameters which encode and clarify the relations between design intent and design response [Jabi 2013]. Naturally, with the use of computers in the workflow, this algorithmic thinking evolved into including scripting knowledge into their definition of PD [Oxman 2017, Janssen and Stouffs 2015].

According to what has been said about Parametric Design, it is possible to conclude that something that depends on certain parameters or relationships among these same parameters, can fit in the definition of this term. Yet, since its inception this term has algorithmic thinking intimately connected to its definition. With technology moving forward, new terms start to emerge and start to overlap with what PD originally stood for. It becomes important to contextualize this dissertation at the light of Engineering Design, and although it still subsists some sort of algorithmic strand, the implicit implementation of some of these algorithms in CAD, the algorithmic thinking is not the main focus during most of the design process. Thus, in this context, any modification of a parameter that directly impacts the design process is considered Parametric Design.

2.1.2 Generative Design

One of the first uses of the term Generative Design (GD) in the literature connected to computational design is somewhere in the late 1970's. The author defined it as systems capable of generating solutions to a given problem [Mitchell 1975].

Later, authors started to create their own interpretations of the theme. In the beginning of the new millennia, a designer was able to interface through a generative system capable of solving a problem autonomously, meaning that the designer was not directly involved with the design itself [Caetano *et al.* 2020, Herr and Arch 2002]. GD can have some sort of evolutionary-based system supporting it [Frazer *et al.* 2002] to explore the solution search space, but this can be considered a narrow definition, as there are several other ways to generate solutions using other forms of algorithmic models and descriptors, generally within a set of constrains to achieve an end goal [Krispel *et al.* 2015]. It is also possible for a designer to "brute-force" the same exploration without using any algorithmic approach.

In an engineering context, generative design is starting to rise in relevance, much because of a technique called Topology Optimization (TO). TO is an algorithmic technique that, in its essence, is able to remove material from key areas that are not subject to the loads applied to a body [Sigmund and Maute 2013] using Finite Elements Analysis (FEA) evaluation. GD and TO can overlap themselves in some applications, causing some confusion. Nowadays there is a better understanding of separation of both techniques, yet when both are used together they make up an immensely powerful tool for engineers. The capability of generating multiple outputs for evaluation allow for the engineer to focus in the innovation and just not to get the part done. MSC compiled the "7 Highly Effective Habits for Generative Design" [Deppe *et al.* 2020]:

- Design for exploration;
- Design for usability;
- Design for productivity;
- Design for costing;
- Design for sustainability;
- Design for manufacturability;
- Design for first-time-right.

These 7 habits or dimensions of generative design with topological optimization as a catalyst manifest the capabilities when adopting these techniques in engineering, looking towards the future of product development.

From what has been assumed until this point, a generative system solely focused in solving a problem is to vague and the same system focused in just using algorithms is to narrow, with the added challenge of distinguishing GD from other techniques, defining it can be difficult. It is then assumed that generative design is the act of generating multiple design solutions to a given problem, whether using any sort of algorithmic approach or simply creating variety manually, which can be reduced as populating a feasible region, within hierarchical criteria.

2.1.3 Algorithmic Design

Algorithmic Design (AD) has been developed and matured in the architecture field. AD had its first references in the early 2000's, suggesting the inclusion of the creative use of computers in the design process [Terzidis 2002]. There are also other works that use the term Procedural Design [Krispel *et al.* 2015, Janssen and Stouffs 2015], but they share most definitions with AD.

From here, authors started to connect GD with AD, stating that AD includes GD [Krispel *et al.* 2015], by defining GD as a technique that does not describe the shape but the operations to be performed in order to obtain the shape through an algorithm. This is generally true, but attending to the definition present on the previous section, there is a clear overlap of terminologies. Other author assumed a position that AD is a subset of GD [Caetano *et al.* 2020]. Alternatively AD was defined by designing a part directly thorough code manipulation [Queiroz *et al.* 2015], using explicit instructions written in programs tailor made for specific applications [Oxman 2017, Humppi and Österlund 2016]. The presented definitions are more aligned with what is to be proposed as the definition of AD.

AD can be generally defined as using programming methods to create a unique solution to a problem, while GD outputs multiple solutions. The designer has complete control over the output, being able to manipulate code accurately to refine any design aspect. This allows the creation of complex solutions without overflowing a typical modelling environment.

AD usually implies the use of any algorithmic approach in the design process, for example, an optimization method as briefly mentioned in Section 2.1.2 or a simple algebraic function. AD can also include procedures to automate tedious and repetitive tasks, saving time, reducing complexity and can be adapted to be able to create design variation by editing descriptors [Leitão and Garcia 2021, Krispel *et al.* 2015].

2.1.4 Computational Design Tools

One of the tools that architects and designers alike use often is Rhinoceros CAD [Betancourt *et al.* 2014]. Rhinoceros CAD (Rhino for short) uses as the API (Application Programming Interface) traditional textual scripting tools in the form of Python or RhinoScript, which is a modified proprietary language based on VBA (Visual Basic for Applications), a textual scripting language created by Microsoft. Rhino also offers many plugin programs that enhance the functionality of the software, for example a plugin called Grasshopper. Grasshopper is a graphic scripting tool that uses a combination of procedures to create explicit models, so the designer can track and edit every step of the process. This makes Rhino a very powerful and versatile tool for architects, designers and engineers alike.

Another powerful tool is Blender, a free and open-source tool. It is mainly used in animation and video games because of the use of meshes as a modelling method. In engineering its use case is more limited, but it can become specially useful to treat and repair models from more traditional Computer Assisted Design (CAD) programs to 3D print or even to be used in Computer Assisted Engineering (CAE) programs. Blender's open-source nature means that there are many add-ins to enhance its depth as a design tool, from a 3D printing toolbox, to an add-in called *Sverchok* that works in a similar fashion to Grasshopper in Rhino [Nedovizin *et al.* 2021]. Blender also permits textual scripting via Python and this allows access to extra libraries, enhancing the design process further.

One of the major tools used by engineers is SolidWorks. Launched in 1993, this software is currently published by Dassualt Systemes and is widely used by universities and companies all over the world [Warfield 2021]. Over the years SolidWorks have been adding evaluation tools and integrating Computer Assisted Manufacturing (CAM) all within the same environment, making it very straightforward to design, assemble, evaluate and export to be manufactured. And whatever the software lacks, third-party developers can create plugins like ParetoWorks from SciArt. ParetoWorks is a plugin that allows SolidWorks to execute topological optimization that generates multiples solutions thus being a form of Generative Design [SciArt 2021]. In terms of scripting, SolidWorks works with VBA in a similar fashion to Rhino.

The final CAD tool considered is Fusion 360 by Autodesk. It is the most used CAD software nowadays [Warfield 2021], has lots of support from developers and is part of Autodesk's ecosystem, which means that exporting files from CAD to CAM is seamless and easy to do, making it ideal for rapid prototyping and manufacturing. If the designer/engineer decides to execute topological optimization, Autodesk also offers a solution within Fusion 360 in a similar fashion to what ParetoWorks does for SolidWorks, offering extreme versatility inside the same application. Originally used C++ as the API language, it recently added Python as another option for scripting. The community of users can create and share add-ins in the forums to further enhance the capabilities of Fusion 360.

Most of the software presented in this section stand out in particular areas of application, sometimes overlapping one another, meaning that there is not a right or wrong tool to be used, as long as engineers/designers choose a tool that is the most adequate for the goal to be achieved. The CAD tool chosen for the following stages of this dissertation is Fusion 360. The combination of the use of Python as the programming language and the integration of external slicers for prototyping are huge advantages to ensure the success of the study.

2.1.5 Concept Overview

This work presents clear definitions for each subject of Computational Design in order to create some separation, but in practical applications, this separations is not as clear. Figure 2.2 represent a schematic example of the possible overlap between definitions.

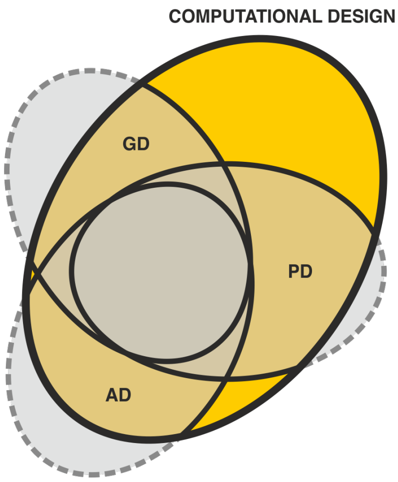


Figure 2.2: CD domain and PD, GD and AD space.

While each type of computational design exists on its own, it is possible to exist areas where these definitions can overlap. In the previous sections was mentioned the strong connections between PD/GD and algorithmic approaches, so, there are instances where both AD and GD can co-exist. If design variation is determined by varying implicit parameters, then GD and PD coexist in the same area. This work stands somewhere in the overlapping region of PD and AD, where programming procedures were used to design a case study and subsequently narrowed down into individual parameters that can be edited. Employing these approaches allows for great flexibility, where it was envisioned another exercise where it used a generative approach to populate a solution space, proving that all three types of design highlighted can definitely co-exist within the same space, thus the ever present difficulty to have clear and distinct definitions for each.

2.2 Mass Customization and Mass Personalization

Intimately connected to the theme of this work, algorithmic design, emerges the discussion about mass customization. In its essence mass customization aims to deliver products and services that best meet individual customers' needs with near mass production efficiency [Tseng *et al.* 1996]. This introduces product differentiation, allowing businesses to be able to add perceived value to the product, while benefiting from competitive advantage.

The underpinnings on how a business could move from mass production to mass customization and its implications on the market was first looked in detail in 1992, being considered "a new frontier" in manufacturing and is still relevant today [Pine 1993]. Mass customization happens when a business has knowledge of the current customer needs through market analysis and categorizes them into different groups. If they are placed inside the same group, they will receive similar products, meaning that products are modular and have certain commonalities between them [Tseng *et al.* 2010].

Some authors started to distinguish mass customization from mass personalization [Kumar 2007, Wang *et al.* 2017] with the latter being an extreme case of mass customization. Mass personalization aims for a market segment of *one* [Kumar 2007] and will present some degree of variation within the base design based on the parameter level to meet unique needs, offering even more value for the customer. One aspect of mass personalization is the requirement for the active involvement of the customer in the design process [Tseng *et al.* 2010].

Nowadays mass customization is implemented in software based applications, where customers can choose options from a list in order to tailor the product for their needs [Heiskala *et al.* 2007, Cannas *et al.* 2020]. A car configurator is a common example of how mass customization can be applied.

An example of application of these principles is what Arburg did in 2015 at Fakuma Trade show where the company tried to customize a pair of scissors, a mass produced item [Gaub 2016]. Arburg had set-up in their booth a computer where customers could choose if the scissor would be right or left handed, have round or pointy tips and they would introduce their name. From there the computer generated a code to be stored digitally on a RFID (Radio Frequency Identifier) chip card. When the time for production the card is read by a controller machine and a human loads the correct blades on to the robotic gripper and the manufacturing process begins. The gripper places the blades on to a plastic injection moulding machine. After moulding the scissor is carried by a conveyor belt to the next station where an autonomous robot loads the product on to a 3D printer where they scan a code embedded in the handle via laser application that contains the relevant information of the customer's name and print a raised logo on to the handle. Another example of mass customization, but related to Algorithmic Design, is what Impact Footwear is proposing with their flip-flops.

Impact's flip-flops are fully customizable and manufactured using 3D printing. The customization is done via a web application and the customer has 9 variables to customize. They are expensive and although they are still a prototype, it shows the potential that this approach has for customers [Footwear 2021].

2.2.1 Industry 4.0

The mass customization and personalization production framework stands heavily on the principles proposed by the implementation of Industry 4.0 technologies [Wang *et al.* 2017]. Listed below are the main pillars of the fourth industrial revolution based on previous literature review [Erboz 2017]:

- Big Data and analytics;
- Autonomous robots;
- Simulation;
- Horizontal and vertical integration;
- Internet of Things (IoT);
- The cloud;
- Additive manufacturing;
- Augmented reality;
- Cyber-Security.

The main pillars that help mass customization and personalization happen can are: Big Data comprises of large databases that collect and manage data that influence organizational decisions regarding market strategies [Erboz 2017]. Autonomous robots are specially useful in flexible production systems, where they are easily reconfigured to support any change in the product design or manufacturing process [Fragapane *et al.* 2020]. IoT integrates communication Machine-to-Machine (M2M) allowing computers to make decisions based on data retrieved from the manufacturing plant, allowing it to be more agile [Erboz 2017]. Cloud computing or "the Cloud" is the on-demand availability of computer system resources, especially data storage and computing power, without direct active management by the user [Montazerolghaem *et al.* 2020]; this allows a shared environment to help manage all aspects of the supply chain [Erboz 2017] and if a business happens to implement mass personalization as defined in the Section 2.2 a customer can also access the cloud to integrate the design process [Wang *et al.* 2017]. Finally, a big component of mass customization and personalization is the use of additive manufacturing, of which will be discussed in more detail in the next section.

2.2.2 Additive Manufacturing

Additive Manufacturing (AM), or 3D printing is the process of building a physical model generated from 3D CAD software by the successive addition of material [Tofail *et al.* 2018]. It can present itself in various forms like sintering of powders, deposition of filaments or material curing with each of these additive processes having their own unique sets of advantages and drawbacks [Shahrubudin *et al.* 2019]. The main focus must be shifted to the exploration of AM as an avenue of application of mass customization and personalization principles.

AM is able to manufacture extremely complex shapes [Chu *et al.* 2008], so manufacturers are no longer limited to traditional methods that are mostly used for mass production and can produce individualized products [Klahn *et al.* 2014]. But, although these products offer more value to the customer, they are still expensive [Attaran 2017].

The big challenge nowadays is to transform additive manufacturing into a valid manufacturing technique for larger production volumes. Figure 2.3 shows a representative graph of the manufacturing cost per part of traditional plastic injection moulding and Stereolithography (SLA), Fused Deposition Modelling (FDM) and Selective Laser Sintering (SLS). The batch sizes have been estimated for economic viability between additive manufacturing (SLA, FDM, SLS) and traditional injection moulding, concluding that for smaller batches additive manufacturing is more viable, but it raises the problem of what's the route that the business can take [Tosello *et al.* 2019].

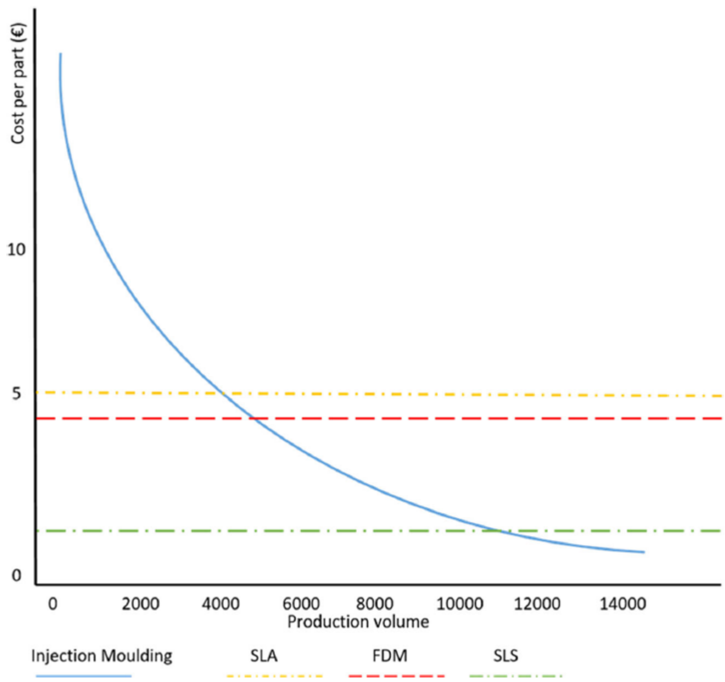


Figure 2.3: Cost per part using traditional manufacturing methods compared to AM technologies [Tosello *et al.* 2019]

Fused Filament Fabrication (FFF) is very popular in the AM world, both in professional or hobbyist settings, is very flexible and most importantly, cheap. FFF is an

AM technology that uses a continuous filament of material, that it is heated through a moving extruding head and deposited on the printed part layer by layer, as seen in Figure 2.4. This technology sees use in several areas like automotive [Arun *et al.* 2018], aerospace [Kumar and Krishnadas Nair 2017], biomedical [Afrose *et al.* 2016] among many others. This inherent flexibility combined with cost effectiveness, not only makes this specific AM technology very attractive, but also allows it to have the largest share in the market since 2011 [Wohlers and Caffrey 2012].

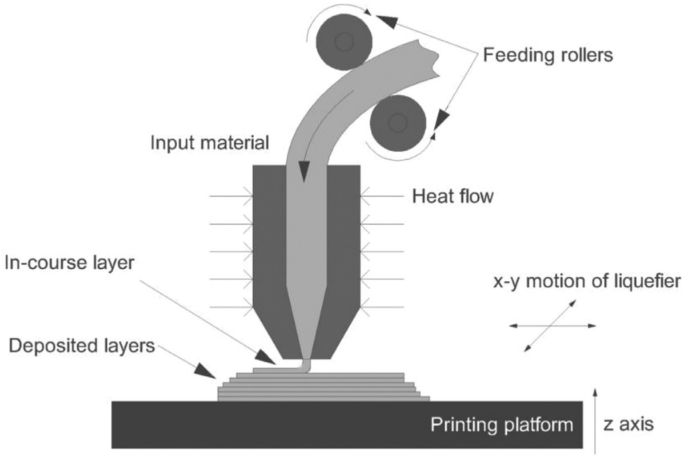


Figure 2.4: Working principle of a FFF machine [Jerez-Mesa *et al.* 2016]

The flexibility of FFF comes from various areas. One of the most preeminent is feed stock material, as it ranges from (mainly) thermoplastics [Singh *et al.* 2020], to metals [Singh and Singh 2015] or ceramics [Singh *et al.* 2017] and the list keeps growing rapidly. The scope of this work focuses more into thermoplastics because of their low-cost, availability and ease of use.

A major strength of FFF is material availability. This technology has flourished within the last decade because the underlying patent expired, which prompted many businesses to manufacture their offers. At the same time, many other companies starting to manufacturing feedstock for these printers with most being comprised of thermoplastics and composites with thermoplastics as a matrix. This feedstock variety leads to the suitability for many applications [Singh *et al.* 2020]. It becomes possible to design and manufacture products per specific application. An example of this concept is mass customization, with customers being able to modify physically and aesthetically a given product to match their preferences.

But FFF comes with some drawbacks, with surface quality being one of them. Its inherent nature of being built on layer upon layer makes it prone to surface roughness. Because AM is mainly used for rapid prototyping a sub-par surface roughness will negatively influence potential assemblies or mechanical properties [Singh *et al.* 2020]. This adds the necessity of post-processing to obtain the surface finish required, which leads to added costs, complexity and turnover time [Singh *et al.* 2020].

Other major drawback is dimensional precision and accuracy. This happens due a multitude of reasons, one is deformation due to internal stress of non-uniform cooling, even leading to failures in the production of parts [Wang *et al.* 2007]. Another issue that persists in dimensional precision is inaccuracies of the printers, be it due to mechanical limitations or through software when creating the paths [Minetola and Galati 2018]. It is possible to minimize the dimensional drawbacks by fine tuning printing parameters and through trial and error [Kaveh *et al.* 2015], but it is not possible to eliminate them altogether.

The final drawback that is very relevant to this work is time. Compared to other traditional manufacturing techniques AM is considerably slower, making its use in an industrial environment very unappealing. Meaning that these technologies are only applicable on manufacturing customized parts and in lower quantities, to keep costs down [Singh *et al.* 2020].

The next section will present an alternative to this manufacturing process that tries to fill in the gaps that exist between large production batches and the cost to execute them.

2.2.3 Hybrid Manufacturing

Hybrid manufacturing can be defined as the combination of two or more manufacturing processes with the goal to overcome either shortcomings. Yet, there is an alternative, narrower way to define hybrid manufacturing and it comprises of simultaneous acting of different processing principles within the same processing zone or machine [Zhu *et al.* 2013]. Based on these definitions it is possible to foresee the amount of combinations possible to execute hybrid manufacturing successfully in a production line.

Although several methods and combination exist, to contextualize this theme to the work being developed there is a need to focus on a single combination. By introducing mass customization principles into an hybrid manufacturing environment, alongside there is a clear connection between Additive Manufacturing and Plastic Injection Moulding. A practical way to implement this idea is using injection moulding for a closed component and then customize upon it with additive manufacturing. This can create a middle ground between mass customization and mass personalization, where it is possible to keep the existence of modules [Pine 1993, Tseng *et al.* 2010, Tseng *et al.* 2017] and have some parts that can be co-designed by the customer [Wang *et al.* 2017], reducing the overall costs and turnover time, while having a highly individualized product.

Part II

Implementation

Chapter 3

Case Study

3.1 Framework

From the beginning of the XXI century, computers have been getting more accessible to consumers. In 2017, over 80% of the European Union population has at least one computer in its household [PORDATA 2021]. These machines are, traditionally, interfaced through two peripherals: the mouse and keyboard. Because of the availability of computers these peripherals tend to be mass made, keeping costs down and not caring for design and ergonomic features.

Nonetheless, this paradigm is gradually changing. With the settling of the COVID-19 pandemic, remote work grew, with video-call platforms seeing unheard usage [Yuan 2020], meaning that people spent more time at the computer. Another curious phenomenon of the pandemic was the growth of the gaming industry as a whole, as seen in Figure 3.1, where the major peak in March corresponds to the beginning of most countries quarantine restrictions. The increase of remote work, with many jobs requiring multitasking, allied to the new need of high performance equipment for e-sports led to the birth of a new type of consumers, eager to explore more about these peripherals participating more and more in public forums about the theme (Figure 3.2).

These new customers are looking for individualized products, while matching their aesthetic requirements. This shift fits perfectly within the framework of this case study, the creation of personalized computer mice using algorithmic design. The human being has different physiognomies, different preferences and different needs, meaning that picking the right mouse for them becomes crucial. A bad choice of this specific peripheral can lead to severe injuries [Chen and Leung 2007]. To mitigate this issue, it is possible to create a tool able to create the perfect mouse for a customers' hand. During the following section the reader will explore the characteristics of the computer mice, what exists in the market in terms of customization and market trends for a benchmark on key points for the design process.

3.2 Benchmarking

This section will present a rundown of certain characteristics of computer mice, highlighting come of the most important. Finally it will be presented a review on other relevant products readily available to create a benchmark for the case study.

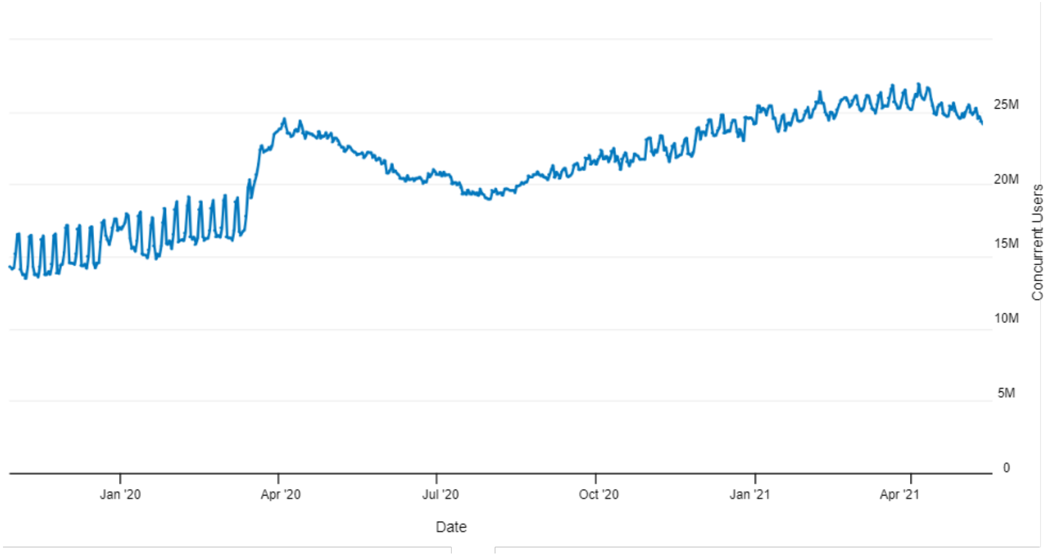


Figure 3.1: The rise of gaming during the pandemic [SteamDB 2021].

Similar to how everyone has a slightly different way of grabbing a pen [Selin 2003], every user has his own way of handling a computer mouse. These different ways of handling are called *Grip Styles* and can be separated in 3 main categories:

- Palm Grip;
- Claw Grip;
- Fingertip Grip.

Figure 3.3 represents examples that show how the hand rests on the mouse. Figure 3.3a is an example of palm grip, Figure 3.3b represents claw and Figure 3.3c is an example of fingertip grip. These different types of grips tend to influence other variables of mouse design, which are not limited by different physiognomies. A person that uses claw grip can have a larger hand than one that uses palm grip, but its preferred mouse can be shorter. Another variable that changes dramatically between grip styles is the *hump*. The hump is the highest point in the mouse and it can vary in height and position. A claw grip user may prefer a lower hump positioned beyond the midway length of the mouse [EpicGear 2021]. The main variables used in mouse design are present in Figure 3.4.

Another important characteristic of the mouse is its symmetry. A mouse can be either symmetric or ambidextrous, where there is less emphasis on individual ergonomics and more on usability by the majority of the population, while fitting most grip styles. An asymmetrical or ergonomic mouse is the opposite, its priority is ergonomics, where it follows stricter design rules, mainly the angle of the sloping. This angle should be around 25 degrees to optimize stress between the forearm and the shoulder muscles [Chen and Leung 2007]. The asymmetrical mouse is more adequate for palm or any other hybrid styles between palm and claw. The case study developed in this dissertation will create both shapes, exploring what can be done in terms of customization.

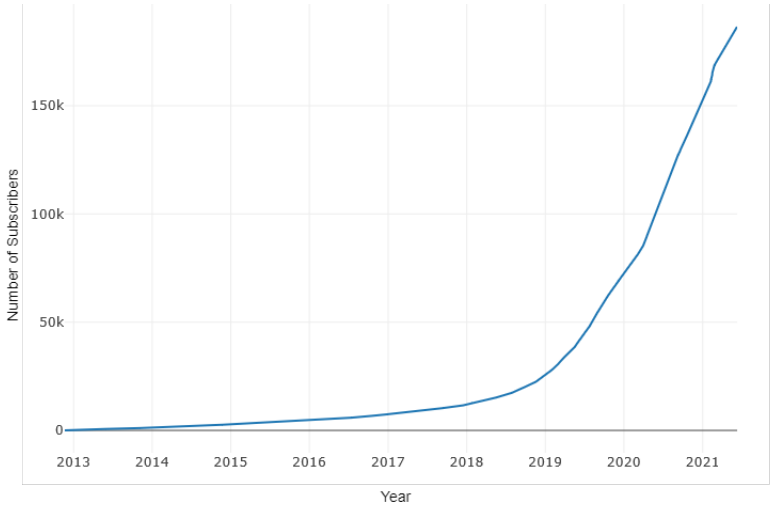


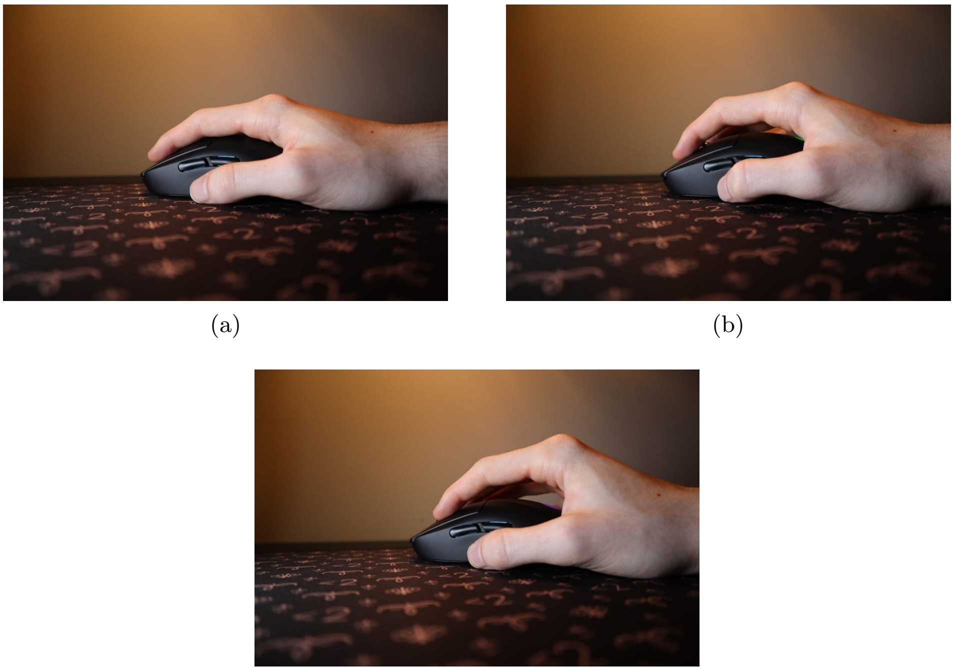
Figure 3.2: Increase of the number of users in a mouse related community in social media [Stats 2021].

With the introduction of the general terminology and basic characteristics of computer mice, the next step is the introduction of the products that occupy the same market space and have some sort of customization. Due to the boom explained in Section 3.1 there are many options available and the next paragraph will cover some options.

Some notorious brands such as *Zowie*, *Glorious* and *Razer* [Zowie 2021, Glorious 2021, Razer 2021] offer the same basic shape, but in different sizes to create variety and match the customers' hand size. Table 3.1 shows the different families from each of these brands and how the dimensions differ from one another. *Xtrfy* went in another route, where its model, the *M42*, comes with a modular part that changes the position and the height of the hump, matching the users' preferred grip style, seen in Figure 3.5a [Xtrfy 2021]. To deepen their customization options *Xtrfy* introduced a configurator where the customer can also choose unique colours combinations for the *M42*, to make it even more personal, seen in Figure 3.5b [Xtrfy 2021]. *Mad C.A.T.Z* created a fully adjustable mouse exemplified in Figure 3.5c [MAD CATZ 2021], but it ended up being too heavy, weighing 120 grams, which goes against a trend of lighter mice (Table 3.1). *Asus* is starting to offer hot-swappable sockets for switches, meaning that the switch can be replaced without any soldering needed, allowing the customer to search for their preferred click, with Figure 3.5d representing the accessibility of the switches in order for them to be replaced [ROG 2021].

Table 3.1: Comparative measurements between products of the same family.

	Zowie		*Glorious*		*Razer*	
	EC1	EC2	O	O$^-$	Viper	Viper Mini
Length (mm)	130	123	128	120	126.8	118.3
Front Width (mm)	64	61	61	58	62.6	59.1
Grip Width (mm)	69	65	66	63	57.6	53.5
Hump Height (mm)	42	42	37.5	36	37.8	38.3
Weight (g)	94	90	67	58	69	61

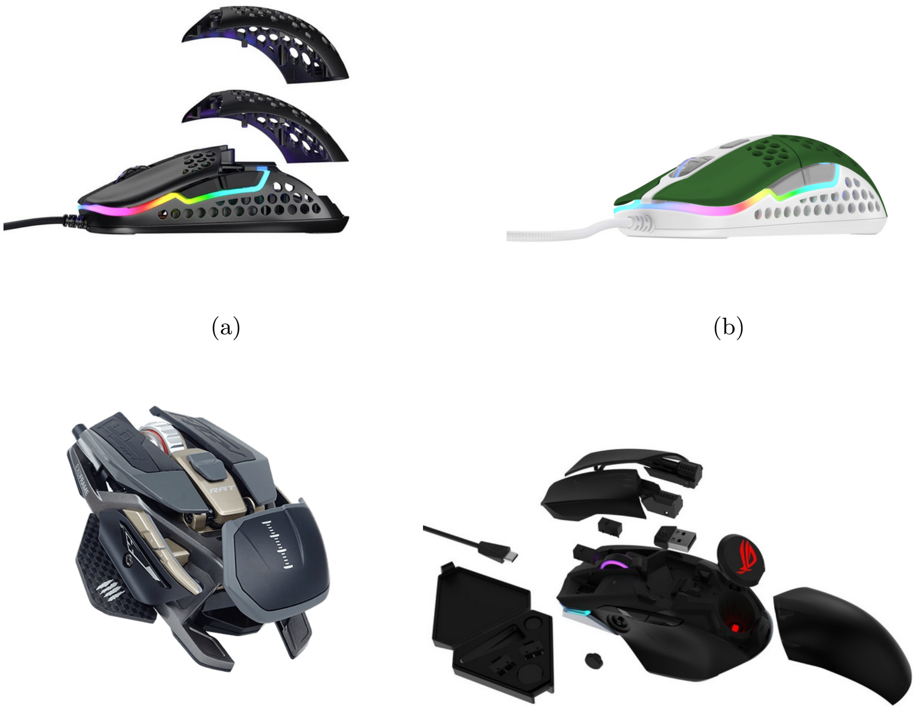


(a) (b) (c)

Figure 3.3: Comparison of the types of grip: (a) Palm Grip; (b) Claw Grip; (c) Fingertip Grip. [GamingGem 2020]

All of the examples given represent the current dynamic in the computer mice market space. Having this much variety is a good thing, but it might cause confusion to the customer as to what they actually want [Chernev *et al.* 2012]. There is the issue of, despite most of the spectrum of hand sizes and grips being covered, there are still compromises that a customer might have to make. With this in mind, the proposition of a fully customized shape can present itself as a good alternative to those compromises.

Figure 3.4: Relevant variables used in mouse design [Razer 2021].



(a) (b)

(c) (d)

Figure 3.5: Market examples of customization: (a) *Xtrfy M42* [Xtrfy 2021]; (b) *Xtrfy M42* with a custom colour scheme [Xtrfy 2021]; (c) *Mad C.A.T.Z R.A.T PRO X3* [MAD CATZ 2021]; (d) *Asus ROG Chakram* [TechPowerUp 2020].

Chapter 4

MUS – Mouse Unravelling Software

After introducing, framing and benchmarking the case study, this chapter serves as a detailed explanation of the stages that comprise the development of the tool MUS (mouse in Latin) and how it is used to explore algorithmic design in product development. A past work by [Zhou *et al.* 2010] explored a similar case study and explored a similar thematic. The goal of that work was to study mass customization using a computer mouse as an applied example, and although the work has some basis in Algorithmic Design (AD), it does not intersect itself with this dissertation. The dissertation "Algorithmic Design of a Product Based on Customer Data" is focuses on the study of AD, while having mass customization as by-product of this exploration.

Figure 4.1 represents a complete flowchart of the development of MUS. It spans from the learning experience to the validation and prototyping and every iterative step along the process.

The chosen software to develop the tool was Fusion 360 because of its API (Application Programming Interface) integration with Python, ease of use, flexibility when exporting the files to a slicer (CAM software for additive manufacturing) and community-driven features to share the source code. Then it was a matter of how to approach the design process.

After the presentation of the framework that supports this work and Computer Assisted Design (CAD) software has been selected, the first stage was learning the API, using and creating small sample programs to familiarize with the environment and language, mostly comprised of modelling a few replicas of preexisting mice to get used to the workflow and test the fidelity of the outcome. From these replicas it was also possible to highlight the relevant variables in mice design. Figure 4.2 showcases both replicas with the real counterparts for comparison.

The goal was to transcript the identified variables onto Fusion 360's work environment so that the designer could alter them without editing the source code, through the use of a command prompt. This step was made easier thanks to the excellent reference manual for Fusion 360's API and available sample programs. The command prompt made the variables available for customization in the modelling environment allowing the designer to experiment and find a good shape interactively, while having direct feedback of the changes that been done. Finally, it was just a matter of testing, validating the versatility

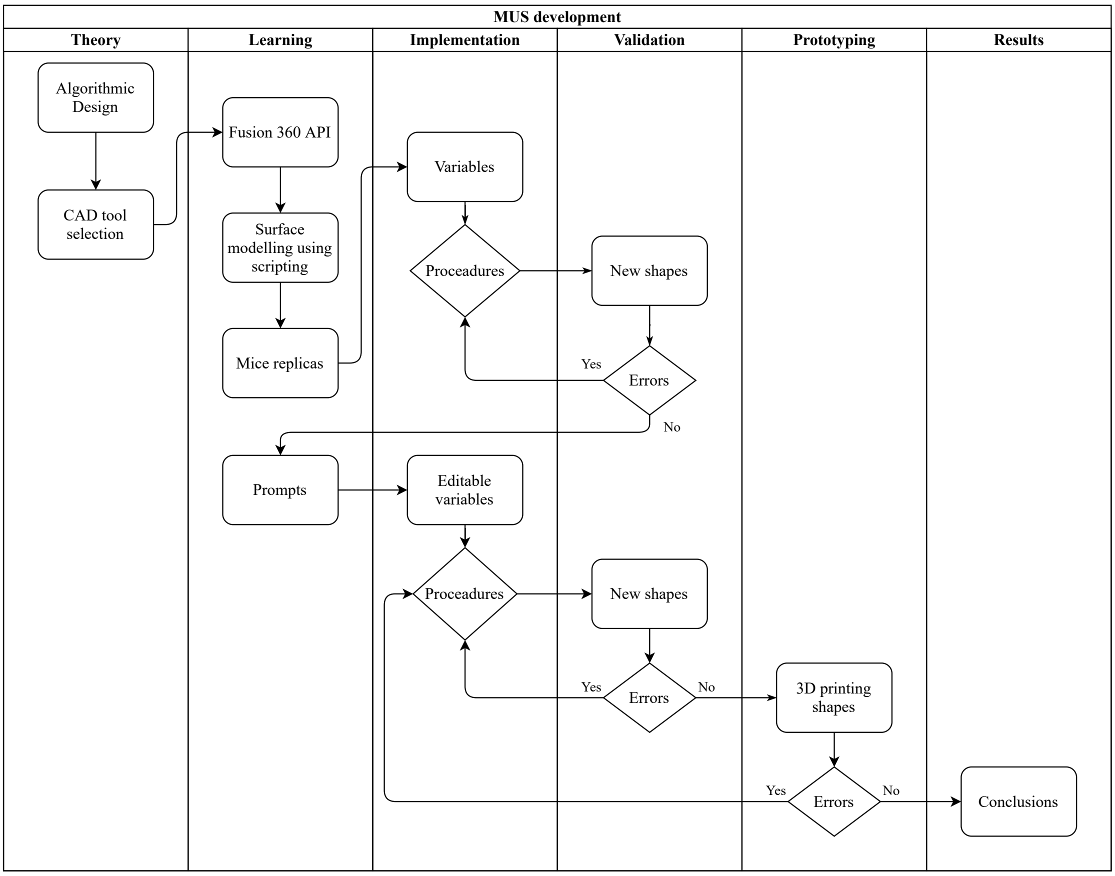


Figure 4.1: MUS development flowchart

and robustness of the tool and advancing to prototyping. Prototyping was done using additive manufacturing, using principles of rapid prototyping to quickly identify any mistakes that were not visible from the modelling environment. All of these steps serve as the foundation for the upcoming chapters where the capabilities of MUS are presented.

This work was based from a direct analysis of the market and had no input from consumers, as its main focus is to study the effectiveness of using algorithmic design in the context of product development. This does not mean that this work is devoid of any consumer interaction, because the prototyping stage will have input from a consumer to build mice according to the data he provides.

The use of explicit surfaces gives a lot more freedom when modeling and this is specially important when the product being modeled has a complex geometry. Most of the geometries of these surfaces are defined by splines, of which most of the results reported are based on their behavior. This means that is important to understand what type of splines are used in CAD software.

There are two spline methodologies that can be used: Bézier Curves or Basis Splines (B-splines). Bézier curves are parametric curves based in the Bernstein polynomials, while B-spline are based in piecewise polynomials [Piegl and Tiller 1997]. Bézier curves rose to popularity in the late 1960's when Pierre Bézier began to use them in the automotive industry [Bézier 1968], but these types of curves/surfaces are a special case of

(a) (b) (c) (d)

Figure 4.2: Creation of mice replicas and comparisons to the real mice: (a) *Logitech G305*; (b) *Logitech G305* replica; (c) *Microsoft WMO 1.1*; (d) *Microsoft WMO 1.1* replica

B-splines. In the work done by [Zhou *et al.* 2010], cited earlier, the mouse surfaces were done using Bézier surfaces where the control points that defined them were editable by the consumer.

In most CAD applications, the most common asset to model splines/surfaces is the use of NURBS (Non-Uniform Rational Basis Splines). This method is as stable as Bézier curves when affine transformation are applied to the curve, is stable numerically and can replicate conical and circular sections. The "rational" portion of the name NURBS comes from the fact that each control point has a certain weight term associated with it, meaning that this term is modifiable and guarantees the interactivity necessary to execute design functions [Piegl and Tiller 1997].

One of the challenges of this work is creating a way to objectively evaluate the changes that each spline endures when a value is edited through the command prompt. Through this evaluation is possible to know where the feasible intervals are located and serves as a way to stress test to the robustness of the program.

A property that is able to track the changes made to a spline is its curvature. Curvature, κ, can be defined as the inverse measure of the radius of a circumference. This means that the lower the radius, the higher the curvature of the circumference, or simply, the tighter it turns [Mary and Brouhard 2019]. The fidelity of the results are dependent on how the CAD software constructs the splines. To verify that Fusion 360 uses NURBS as a method to construct splines it was devised a test: Create an arbitrary fifth degree spline with known control points and through the CAD software export the curvature values directly. Using the same control points, construct the arbitrary spline using the library NURBS-Python [Bingol and Krishnamurthy 2019] and export an array of 200 coordinates placed in the spline to Matlab. Once this data was in Matlab, Equation 4.1 was used to calculate the curvature values, where $\dot{x}$ and $\ddot{x}$ are the first and second derivatives, respectively, of the relative spline length, while $\dot{y}$ and $\ddot{y}$ are the first and second derivatives, respectively, of the height of the points that make up the constructed spline. Figure 4.3 contains both curvature values along the spline plotted for comparison. The code used to export and calculate the curvature values for this test can be found in Appendix A.1.

$$\kappa = \frac{|\dot{x}\ddot{y} - \dot{y}\ddot{x}|}{(\dot{x}^2 + \dot{y}^2)^{\frac{3}{2}}} \tag{4.1}$$

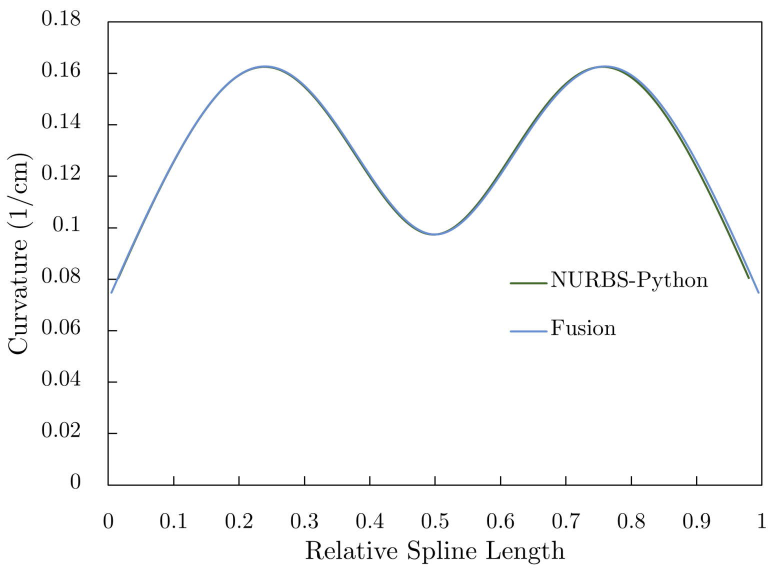


Figure 4.3: Curvature comparison between a spline constructed in Fusion and other constructed using NURBS-Python.

The curvature values and shape of both splines are similar, hinting to the fact that Fusion 360 uses, in fact, fifth degree NURBS when constructing their splines. The only divergence is that Fusion 360 spline is slightly shifted to the right, which might come from discrepancies when translating the spline built by NURBS-Python to the relative values in the x-axis. The NURBS-Python curvature plot is cropped at both ends because of discontinuities caused by the control points located at the start and end positions of the curve.

The development of this work has been divided into two different approaches: Generative Design and Algorithmic Design. Both programs were developed in tandem, where any knowledge from one subject of design can be translated to the other, in a seamless operation. This allowed to fine tune the splines and get data to further improve the design.

4.1 Generative Approach

As defined in Section 2.1.2, generative design can be used to generate design variety and populate a feasible solution domain. To implement this idea a program was developed where some of the variables were put in an interval of ten values equally spaced of each other and then was executed through a command prompt, with the variable selection done via drop down menu Figure 4.4. Figure 4.5 represents a wire-frame of the mouse and has every spline present in the modelling strategy.

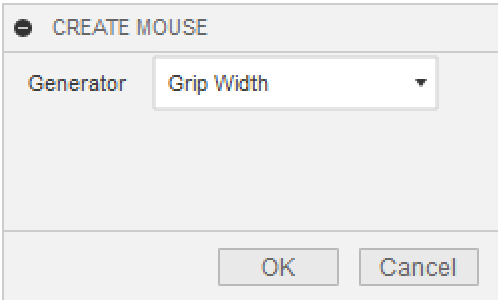


Figure 4.4: User interface to generate the multiple mice using a command prompt.

Besides the exploring nature of the procedure, the results obtained can be used for a sensitivity analysis to determine how influential is each variable in the design process. In general the changes are minute because some of them have intervals that vary within 2 millimeters, but, in practise, these changes can make a difference on how a mouse feels in the hand. Tables 4.1a and 4.1b show the dimensions for the static variables alongside the intervals tested in Table 4.2. These results will be divided in two sections: symmetrical and asymmetrical typologies. This separation is justifiable because of slightly different approaches between typologies thus leading to different results.

Table 4.1: Default measurements used for the generative process: (a) Symmetrical typology; (b) Asymmetrical typology.

(a)

Variables	Default (mm)
Mouse Length	124
Hump Position	80
Hump Height	36
Front Width	54
Grip Width	60

(b)

Variables	Default (mm)
Mouse Length	124
Hump Position	80
Hump Height	36
Front Width	54
Grip Width	62

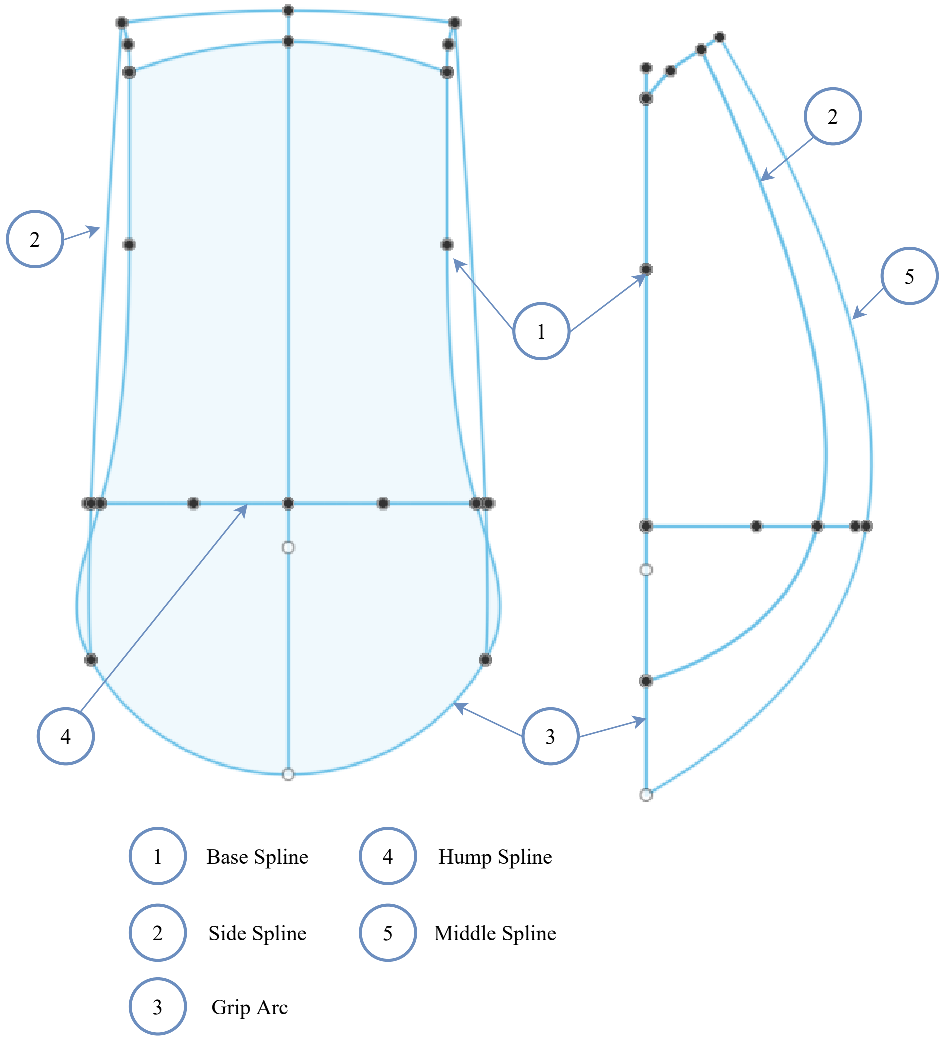


Figure 4.5: Diagram with the spline nomenclature.

Table 4.2: The measurement intervals used for the generative process.

Variables	Tested measurements (mm)									
Length	115	116.7	118.8	120.1	121.8	123.5	125.2	126.9	128.6	130
Hump Pos.	65	67.8	70.6	73.4	76.2	79	81.8	84.6	87.4	90
Hump Hei.	28	29.1	30.2	31.1	32.2	33.3	34.4	35.5	36.6	38
Front Wid.	46	47.56	49.12	50.68	52.24	53.8	55.36	56.92	58.48	60
Grip Wid.	50	52	54	56	58	60	62	64	66	68

Along side the more analytical evaluations based on the curvature analysis presented in Chapter 4, for certain variables there are figures that are visual representations of the behavior of the variables throughout the multiple generated solutions. The following sections will delve into the evaluation of the editable variables present in each mouse typology and how it can influence future deployments of the tool. In the figures that present a curvature plot the x-axis is named "Relative Spline Position" and corresponds to the location of a point relative to the spline length. The y-axis corresponds to the absolute value of curvature in cm^{-1}. When a curvature plot is computed, the values of the "Relative Spline Position" are mirrored in relation to the figures that contain the curvature colour maps. In these curvature colour maps: red shows the maximum values and purples show the minimum values.

4.1.1 Symmetrical Typology

Based upon the definition of Generative Design and the evaluation tools that have been discussed, the analysis starts by looking into the symmetrical typology. The first variable to be studied is mouse length. Usually, the larger the hand, the bigger the mouse length, but this is not always the case as briefly pointed out in Section 3.2, with the incremental length between two specimens for for visual comparison in Figure 4.6.

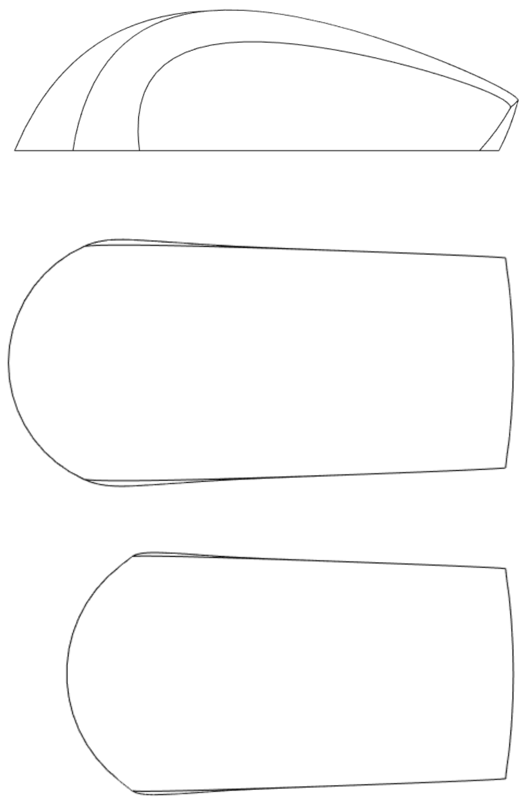


Figure 4.6: Visual comparison of mouse length difference between "Mouse 1" and "Mouse 10".

The analytical analysis can be done from the plot present in Figure 4.7. The first conclusion drawn form Figure 4.7 is the slope of the middle spline getting flatter past the hump position when the mouse length increases, corroborated by the reduction of the peak value of curvature at around 80% of the spline length. The second is the increase in curvature at the beginning of the spline. This is due to a rolled edge, represented in Figure 4.8, that exists with the phenomenon getting progressively worse as the mouse length increases.

This rolled edge also has influence in the curvature of the middle spline when the hump position are varied. The variation of the hump position influences the curvature value of the middle spline and causes the peak of curvature to move along the x-axis, towards the relative spline position of 1. This is highlighted in Figure 4.9, where there is a comparison between “Mouse 1” and “Mouse 2” with a benchmark mouse for reference (Figure 4.10). This comparison with a benchmark highlights a problem with the curvatures that have been plotted. Apart from the exaggerated peak caused by the rolled edge, mentioned previously, the curvature peak should be closer to the relative position of 0.5, yet this peak happens in the relative position of 0.75. This is caused by that same rolled edge, skewing the results and it should be addressed in a future deployment. The visual impact caused by varying the hump position in the mouse shape can be seen in Figure 4.11. Hump position also influences the curvature of the base splines, seen in Figure 4.12, where moving the hump towards the back of the mouse increases the curvature of the base spline at the front, forming a sharper edge close to the grip, due to tangent relationship between the base spline and the grip arc, resulting in an exponentially higher curvature at the grip. Figure 4.13 helps contextualize the evolution by providing visual reference.

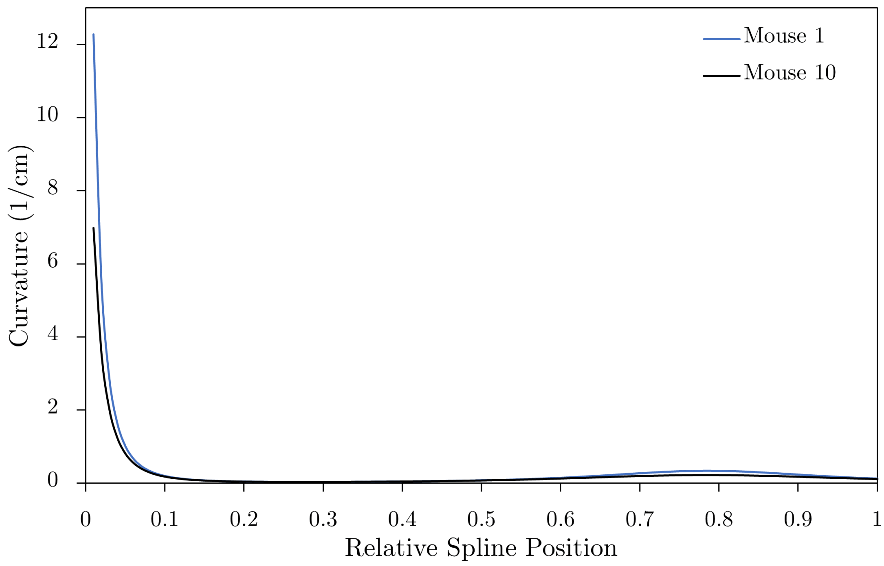


Figure 4.7: Comparative curvature of mice length between “Mouse 1” and “Mouse 10”.

Hump height, in terms of mouse design, is similar to the hump position, where it does influence the type of grip of the user, but remains a subjective variable. Figure 4.14 compares three different generated mice and their curvature maps. The conclusion drawn from this first visual analysis is the taller the hump the rounder the sides. This makes sense due to how that curve was defined (code provided in Appendix A.2), but the lower the hump the more the curve resembles the letter “W”. The curvature graph (Figure 4.15) can corroborate this assessment. From the graph, it is possible to observe

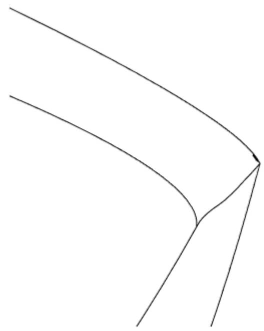


Figure 4.8: Representation of the rolled edge that caused the anomaly in the plots.

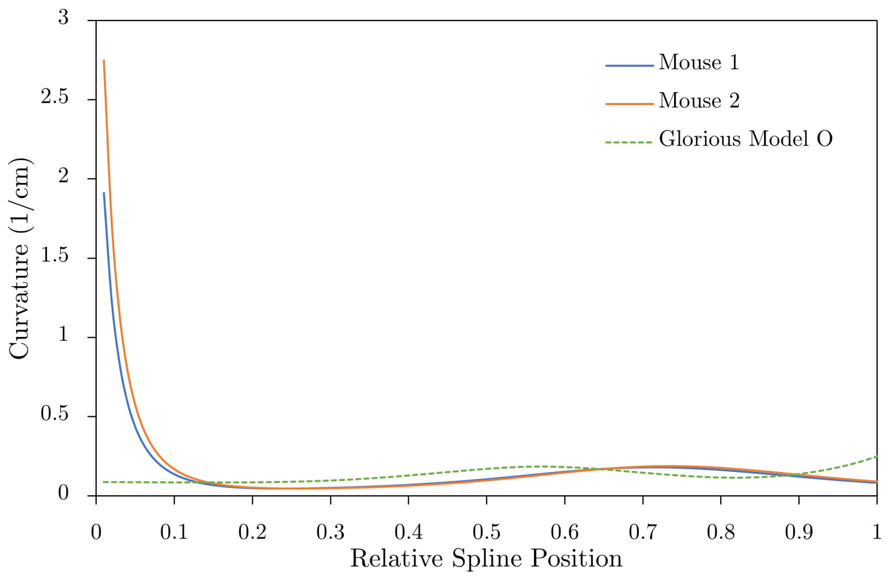


Figure 4.9: Comparative curvatures of the generated hump positions for the symmetrical typology along side a benchmark.

an almost flat spline in both ends, with a single peak in the middle of the hump spline. In comparison the third mouse from the right in Figure 4.14 shows an hump spline that resembles an arc, but seems to have a flat section in the middle of the spline, proved by the green colour provided by curvature map. Figure 4.15 proves this aspect, by showing the curvature peaks at relative positions of 0.25 and 0.75, whilst being flat, with a valley at the halfway point. Figure 4.15 also has the same benchmark plotted as before for comparison with the generated mice.

Next variable to be explored is front width. To compare the entire generation, the bottom profile has been isolated, in order to better visualize the differences, as seen in Figure 4.19. Here, the geometries affected are the base splines and the hump spline. When a front width is generated in its minimum value, this spline is almost straight until 73% of its total length, then, because of the edge caused by the base spline meeting the grip arc, the curvature values increase. But, as the front width tends to higher values, the spline increases its curvature at the front, whilst maintaining a slight indented curve

Figure 4.10: Benchmark chosen for comparison with the generated mice [Glorious 2021].

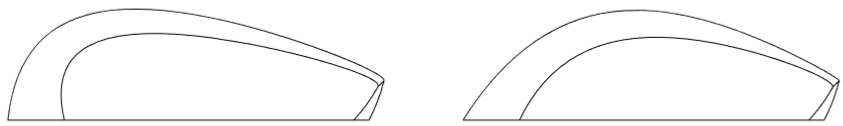


Figure 4.11: Visual representation of the variation of hump positions between “Mouse 1” and “Mouse 10”.

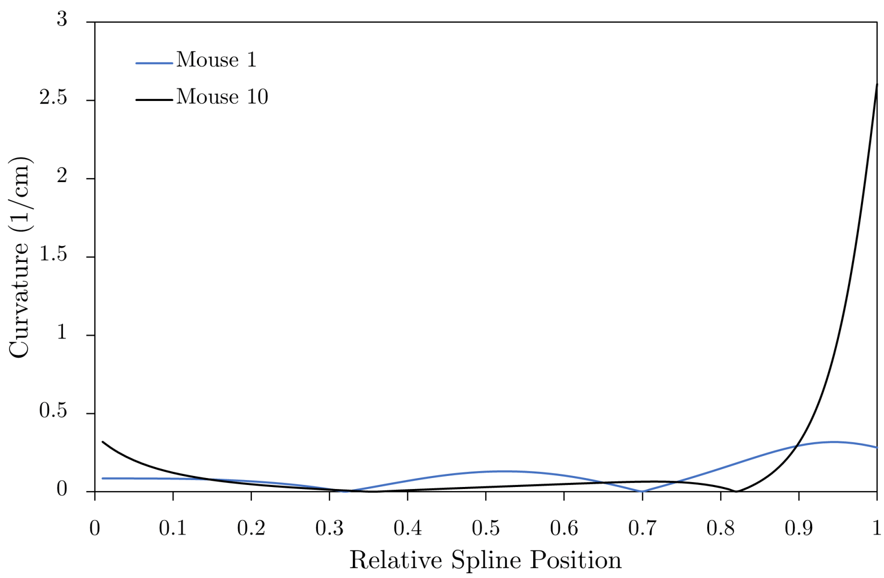


Figure 4.12: Curvatures of the base spline when varying the hump position.

at the middle, causing the slight peak in curvature between 37% and 78% of the total

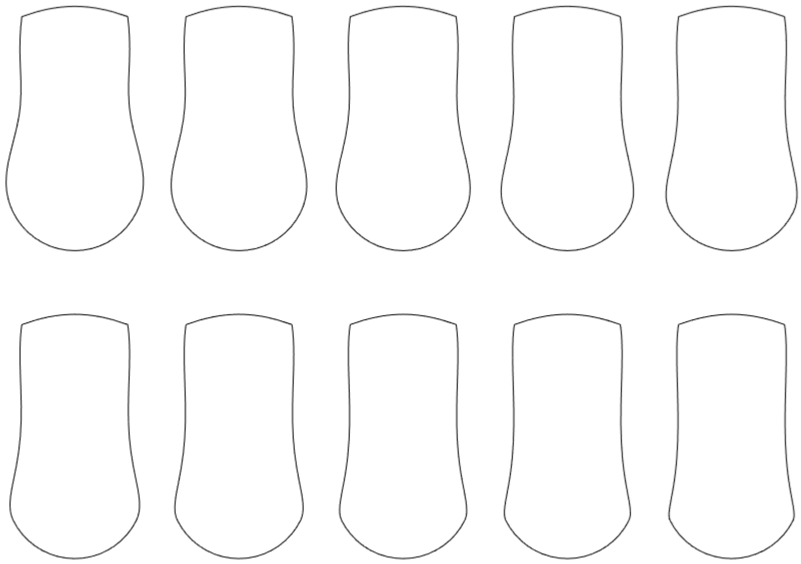


Figure 4.13: Visual representation of the impact on the base splines by moving the hump position.

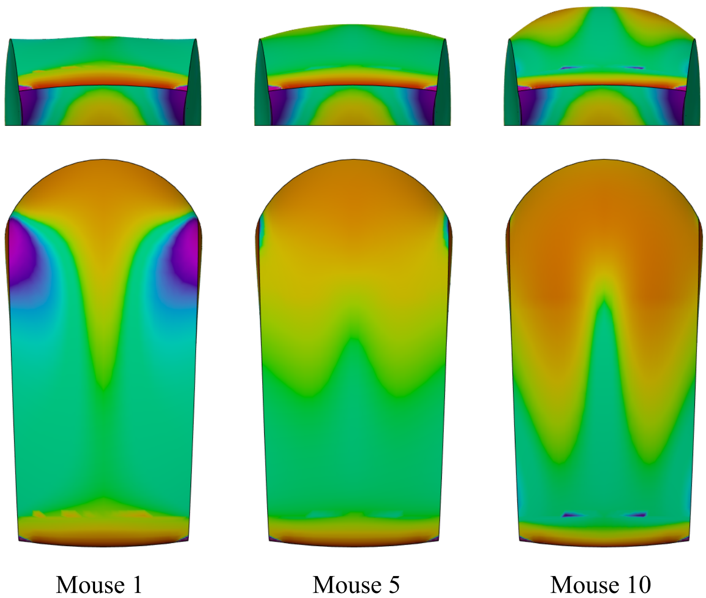


Figure 4.14: Front and top views of the 3 different variations of hump heights.

spline length. The type of curvature of “Mouse 10” seen in Figure 4.17 can be considered undesired because the curvature near the front of the mouse should not be as pronounced it presents itself, and should have a smoother arc throughout the base spline to increase comfort where the fingers rest. The hump curvature is also affected by varying the front

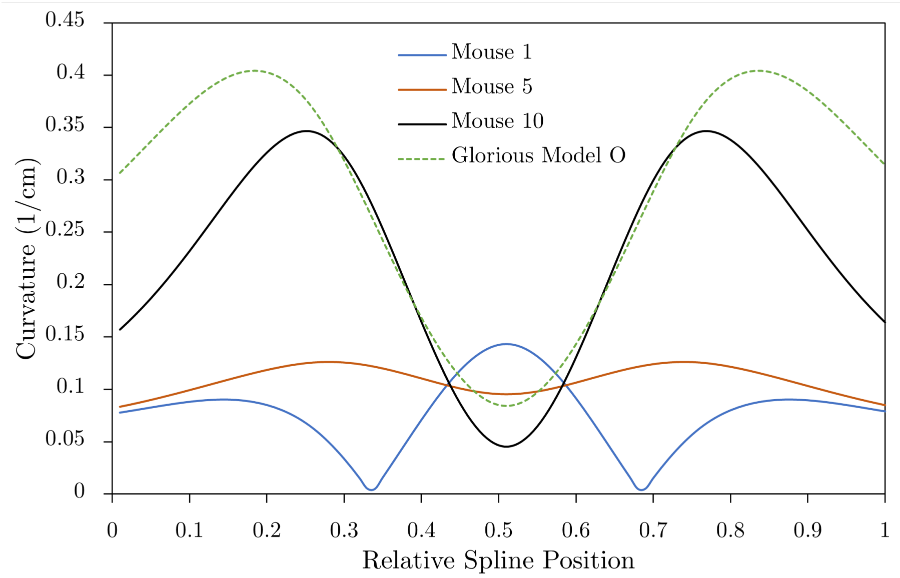


Figure 4.15: Different relevant curvatures of the hump heights, alongside with the selected benchmark.

width of the mouse. Although "Mouse 1" may not be a bad result in terms of curvature, in terms of design, it is wiser to keep both peaks as further apart as possible to make the hand rotate towards a more natural resting position, without compromising the base splines (Figure 4.18).

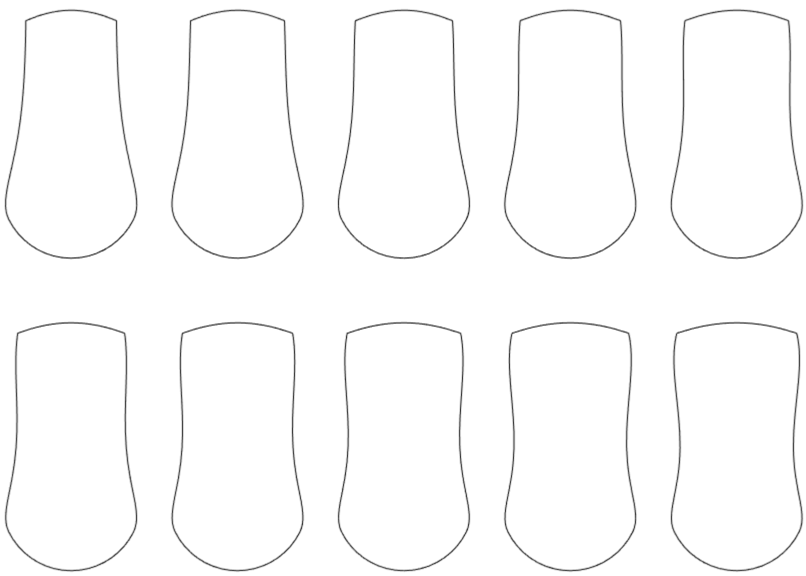


Figure 4.16: Bottom profiles when generating the front widths.

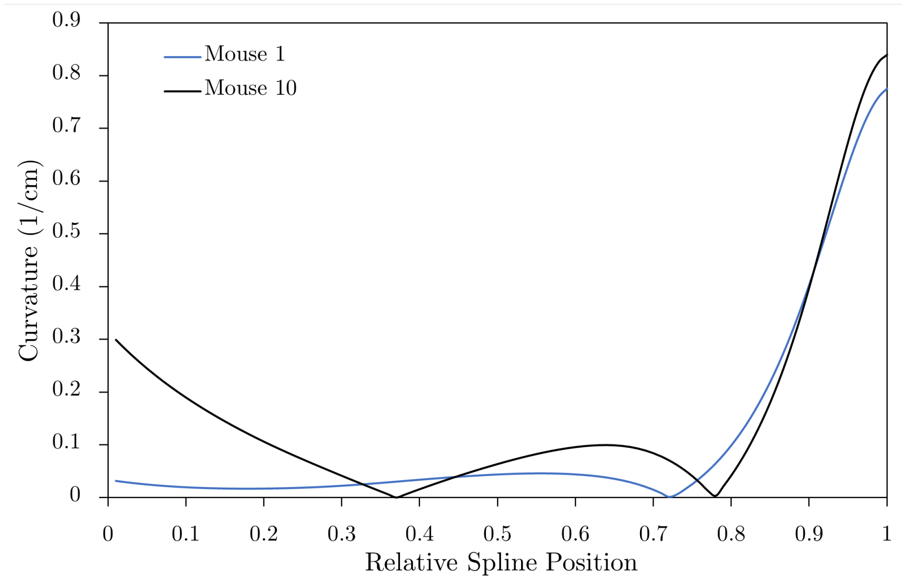


Figure 4.17: Curvatures of the base spline when varying the front width.

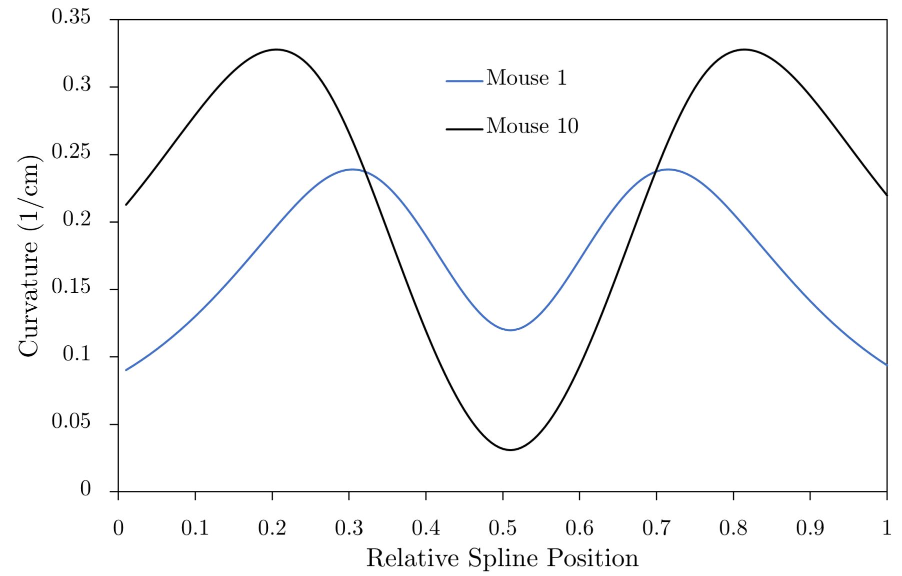


Figure 4.18: Curvatures of generated hump heights when varying the front widths.

Grip widths retain similar conclusions to the ones drawn from the analysis of the front width. The smaller is the back width, the curvature at the beginning of the spline is higher, seen in Figure 4.20, meaning the grip width must be larger than front width, as seen in Figure 4.21. Following Figure 4.22, the hump curvature differs from the one generated by the variation of front widths, where “Mouse 1” has an exacerbated hump, with the two peaks at 20% and 80% and a small valley at the halfway point of the total spline length, while the mouse next to it tends towards a smoother curvature overall, with a single peak in the middle, corroborated by Figure 4.23

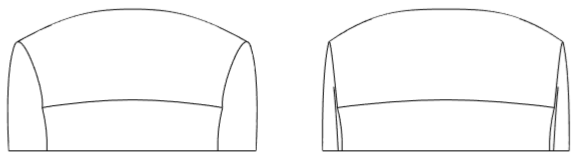


Figure 4.19: Hump heights affected by the variation of front widths.

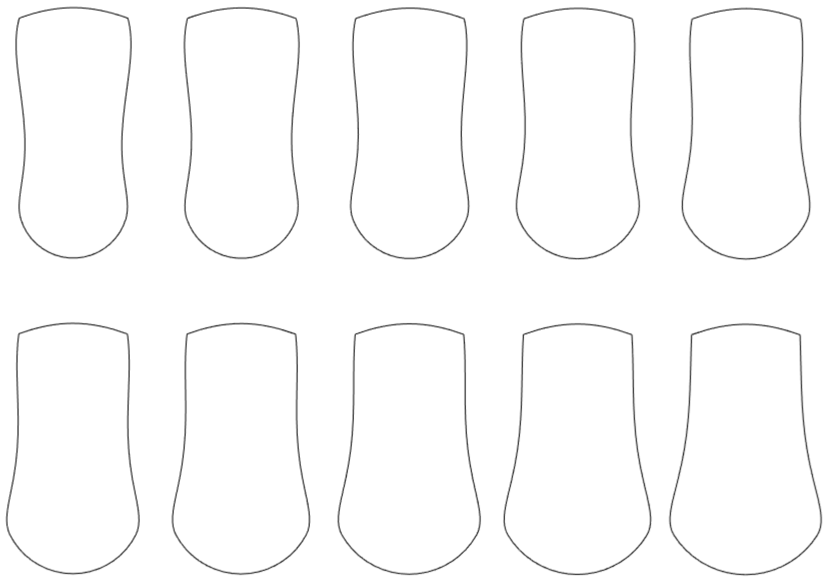


Figure 4.20: Bottom profiles when generating the grip widths widths.

4.1.2 Asymmetrical Typology

Although both typologies are based upon the same modelling strategy, they can yield different results. As such, it becomes interesting to replicate those different solutions keeping the same formula of a sensitivity analysis used in the previous section. There are a few differences in the default measurements as well, as seen in the Table 4.1b, otherwise the tested intervals remain the same.

The first variable to be evaluated is mouse length, but the results and the conclusions remain the same as the ones from Section 4.1.1. Hump position affects the base splines and the middle spline in the same way as reported in the symmetrical typology. The major difference between typologies is the base spline on the left side of the mouse, seen in Figure 4.25. A hump positioned towards the back of the mouse translates to a spline similar to a straight line, with a curvature peak where the base spline meets the grip, due to the sharper edge created by the tangent relationship between curves. The curvature of the middle spline when varying the hump position is shown in Figure 4.26. The used curves correspond to "Mouse 1" and "Mouse 2", with an addition to a reference mouse

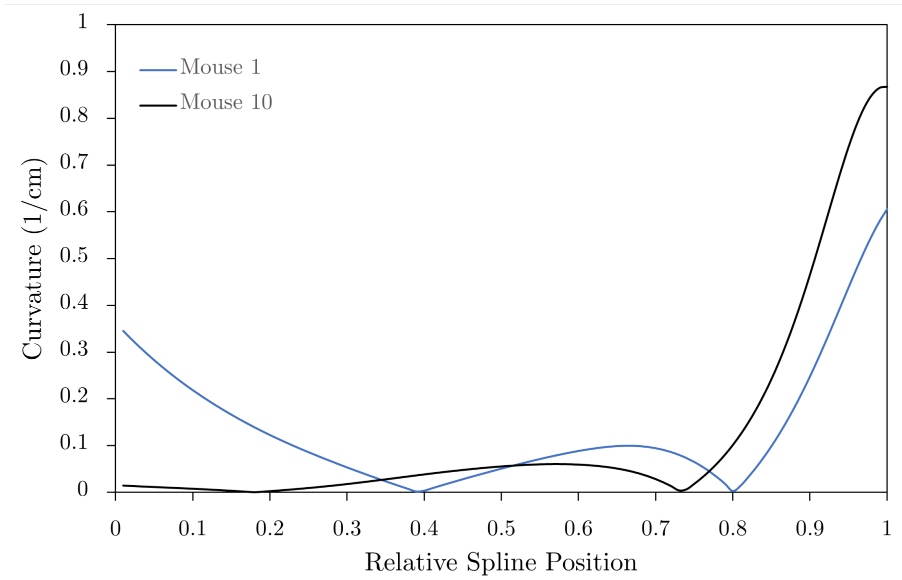


Figure 4.21: Curvatures of the bottom spline when varying grip widths.

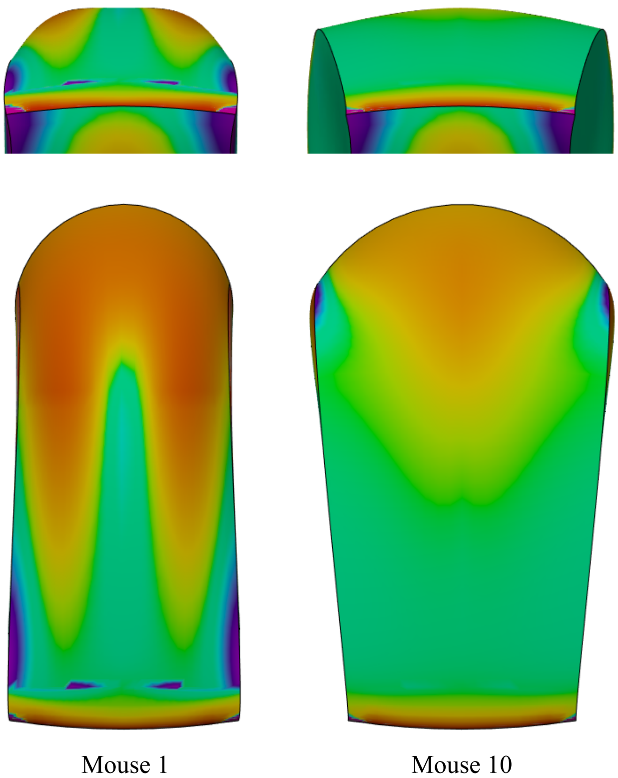


Figure 4.22: Curvature maps of the hump evolution when varying grip widths.

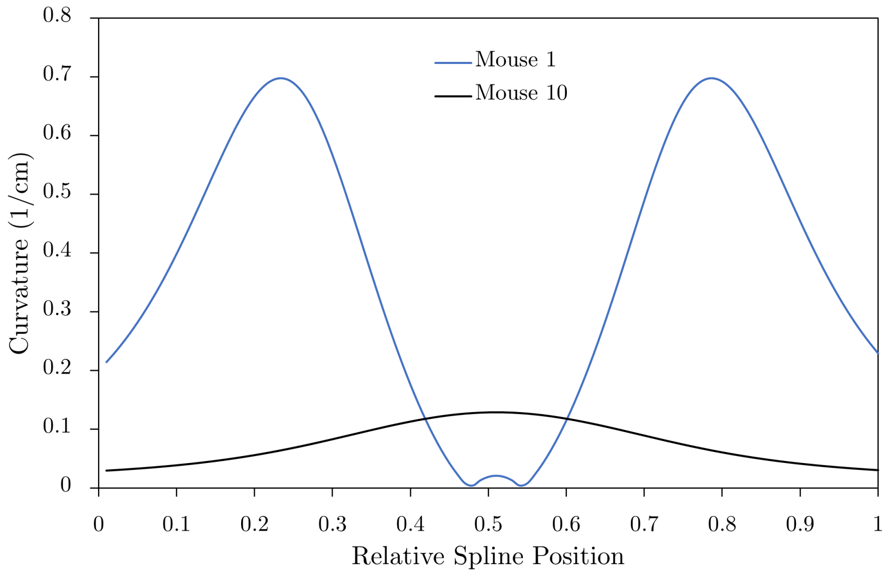


Figure 4.23: Hump curvatures when varying grip widths.

as a benchmark, present in Figure 4.27, to offer a point of comparison. These results are not satisfactory for a couple of factors. According to the intervals of Table 4.2, the hump position of "Mouse 1" is positioned at, approximately, the halfway point of the spline, but, in practise, the curvature peak happens at 70% of the total spline length. When compared to the reference curvature, the hump is placed at the middle of the mouse, based on its peak curvature. The second issue is the excessive curvature at the front of the mouse, caused by the same rolled edge denoted from Section 4.1.1. With the hump moving progressively towards the back of the mouse, the curvature value originated from this rolled edge got worse, with curvature values up to 20 cm^{-1}. Both of these aspects need to be addressed in a future deployment of the tool. Figure 4.28 represent a visual comparison on how the mouse geometry is affected by moving the hump towards the back.

Hump height presents anomalies derived from the nature of how the modeling was implemented (with example code provided in Appendix A.2), in a similar fashion to the ones in the symmetrical typology. Figure 4.30 shows the curvature maps of the same generated mice used in the curvature graph of Figure 4.29, to help contextualize the data.

It is possible to conclude that the curvature of "Mouse 1" falls outside the feasible domain, because the spline is unstable and should form a smooth, outward curve to match the curvature of the hand. By increasing the hump height the curvature close to the starting point of the spline decreases, while increasing the curvature at the end point of the spline. Figure 4.30 helps clarifying this statement. Moreover, the hump spline of the mouse in the middle in Figure 4.30 is also not ideal because it has a small valley near the 0.35 point of the total length of the spline. The result that can be considered ideal is the curvature of the benchmark (Figure 4.27) and the curvature profile of "Mouse 10" tends to approximate the ideal result.

The asymmetrical typology introduces a new base spline, while keeping the other

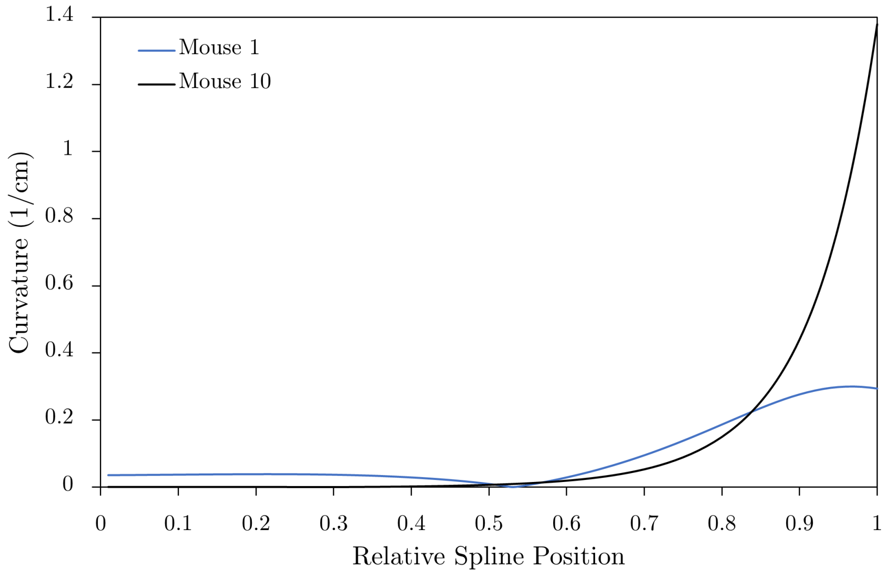


Figure 4.24: Comparison of curvatures on the asymmetrical bottom spline when varying the hump position.

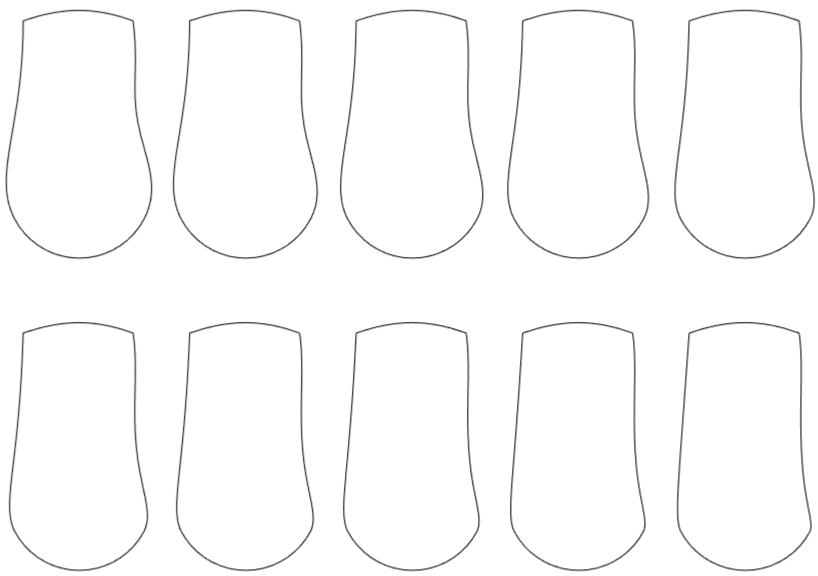


Figure 4.25: Visual representation of the impact on the base splines by moving the hump position.

from the symmetrical typology. This means that the base spline that is kept from the symmetrical typology maintains its behavior as the one studied in Section 4.1.1. Analysing the variation of front widths and the impact on the asymmetrical base spline shows a curvature increase the wider the front gets, while moving the point where the

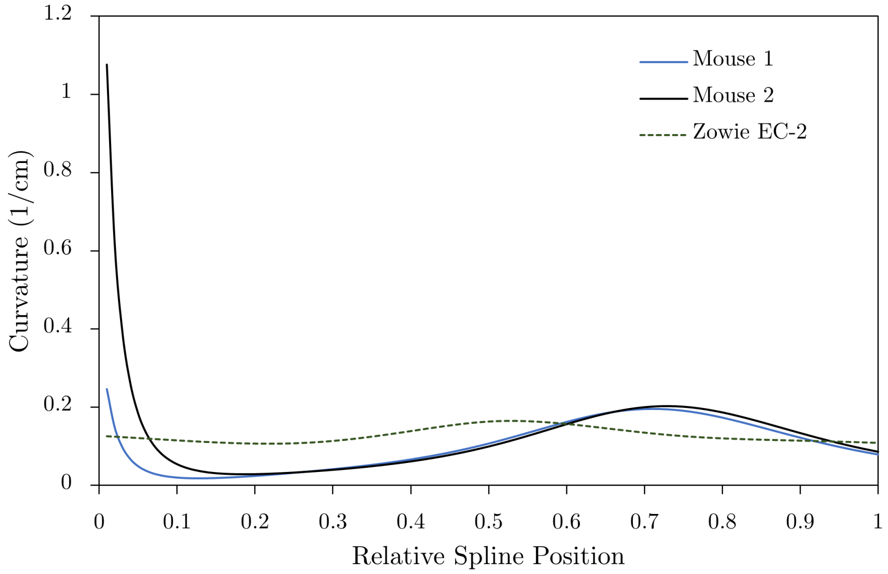


Figure 4.26: Curvature of the mice lengths when varying hump position compared to a benchmark.

base spline meets the grip arc from 50 to 60% of the total spline length. Varying front widths also affects the hump. Figure 4.33 plots the evolution of selected humps throughout the generations. The wider the front of the mouse, the initial curvature of the hump gets higher. Figure 4.34 shows these changes visually. It should be noted that for the widest front value, the curvature of the hump gets close to the curvature deemed "ideal".

Even though the generated values are different due to different interval ratios, grip widths have the reverse effects on the bottom splines, as seen in the previous section and corroborated by Figure 4.35, whereas the effects on the hump do not improve. Figure 4.36 shows the evolution of the hump for selected examples when varying back widths. "Mouse 1" is out of bounds. "Mouse 6", on the other hand exhibits a curvature profile that matches the profile that has been considered "ideal", with a higher curvature at the starting point of the spline and flatter towards the end of the curve. The curvature of the hump of "Mouse 10" starts flatter with the first peak appearing at the halfway point of the hump spline and the spline ends with large values of curvature, roughly presenting the opposite of the hump spline of "Mouse 1". This means that grip width has more influence in the hump profile, unlike what has been verified from varying the front width.

Taking into account some of the observations made across this chapter, it is possible to draw some final conclusions and small corrections to be made in the next deployment. The first glaring issue is trying to regularize the mouse curvature, by removing the rolled edge. This is solved by increasing the height of the beginning of the spline from which that curvature is measured, while decreasing the height of the starting points of the side splines. Another conclusion is that grip width has more weight in the design of the mouse, specially the hump spline, making this variable unstable and difficult to use. When it comes to mouse design, it is needed a careful manipulation of variables.

Figure 4.27: Benchmark chosen to compare the curvatures between it and the generated mice [Zowie 2021].

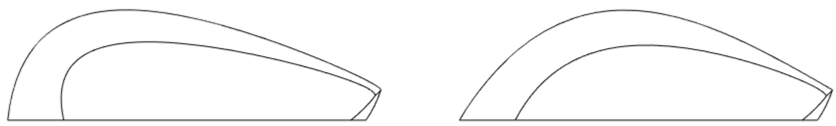


Figure 4.28: Comparison between of hump positions between “Mouse 1” and “Mouse 10”.

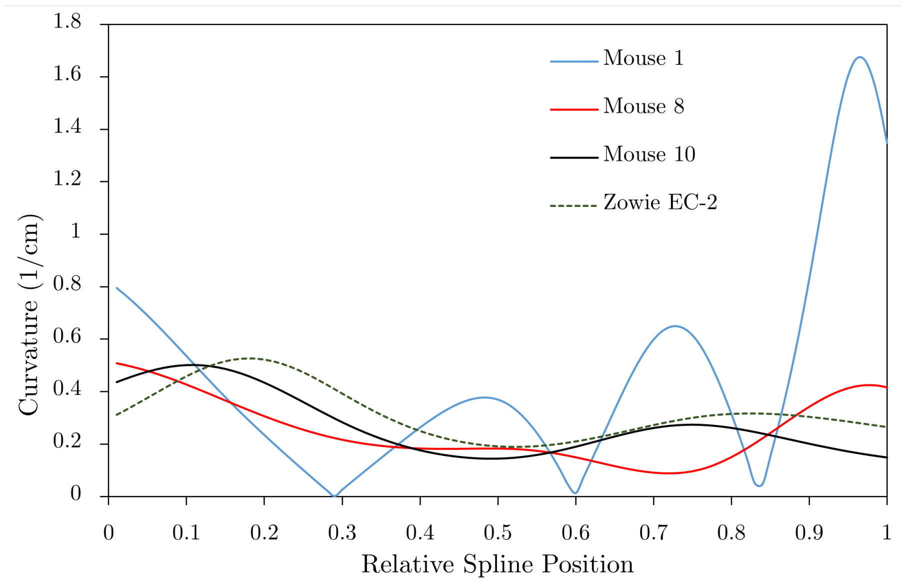


Figure 4.29: Curvature comparison of the hump when increasing the hump height.

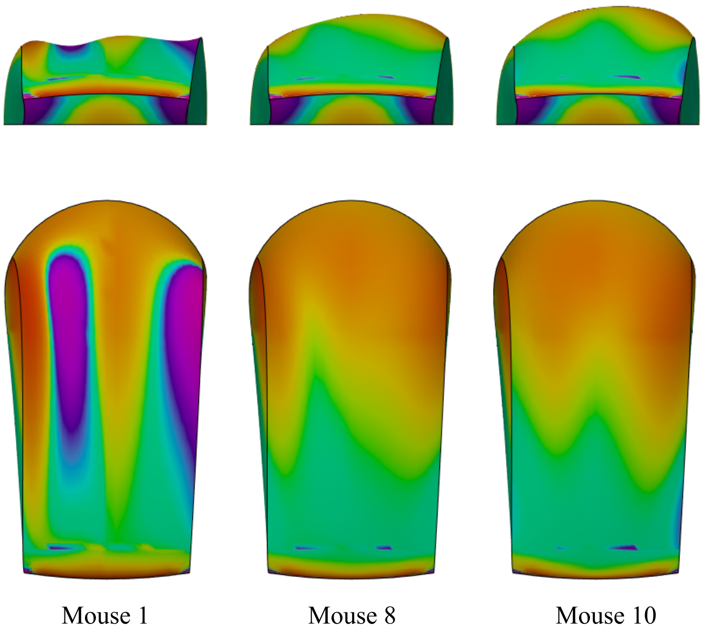


Figure 4.30: Front and top views to compare some of the effects of increasing the hump height.

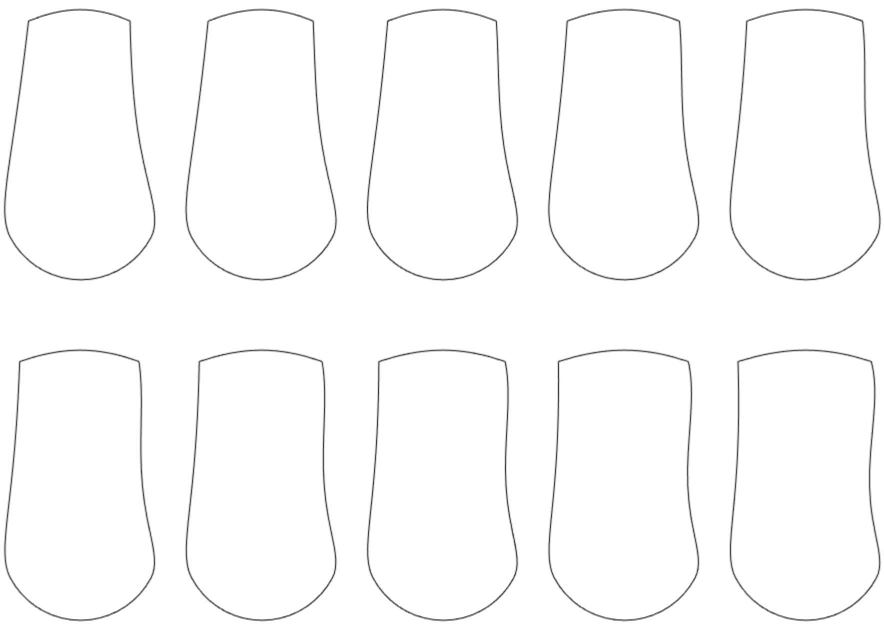


Figure 4.31: Curvatures of the base spline of the asymmetrical mice when varying the front width.

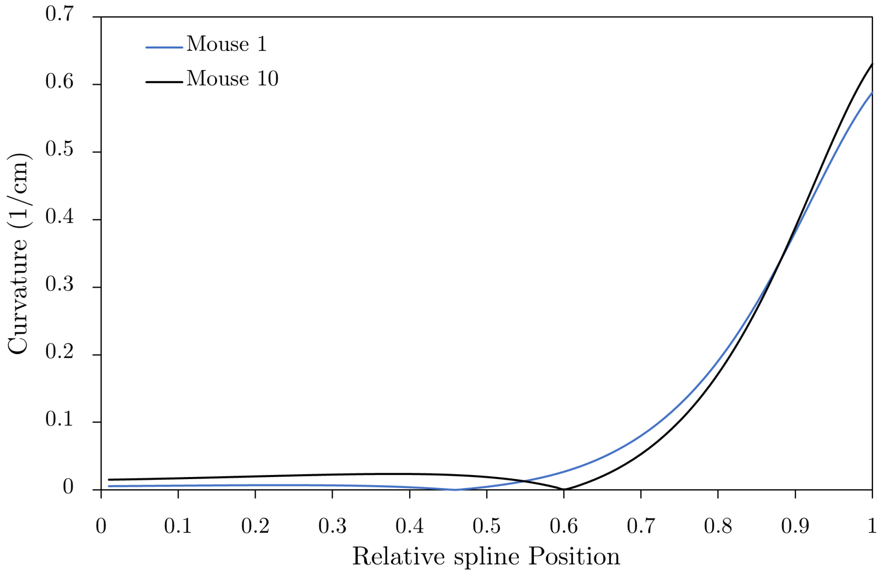


Figure 4.32: Comparison of curvatures of the base spline when varying the front width of the asymmetrical mice.

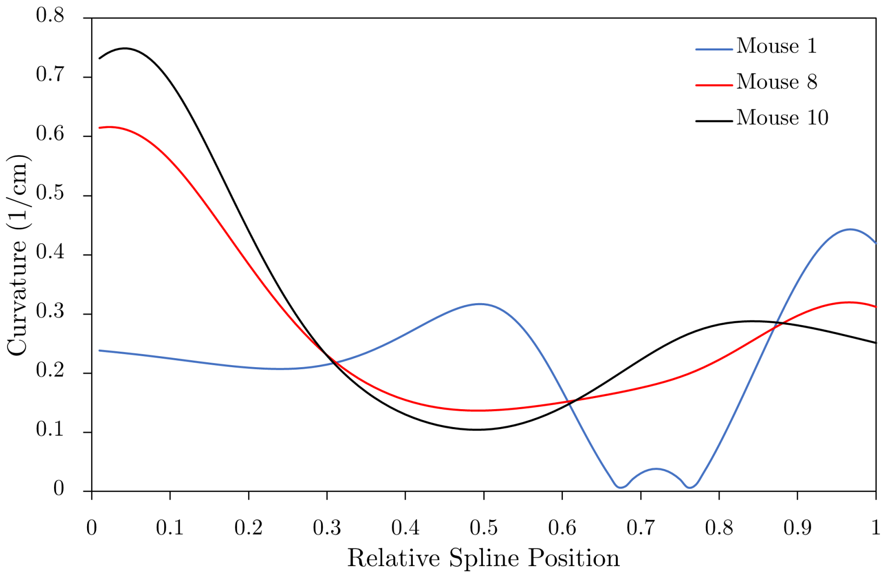


Figure 4.33: Curvature evolution of the humps when varying front widths.

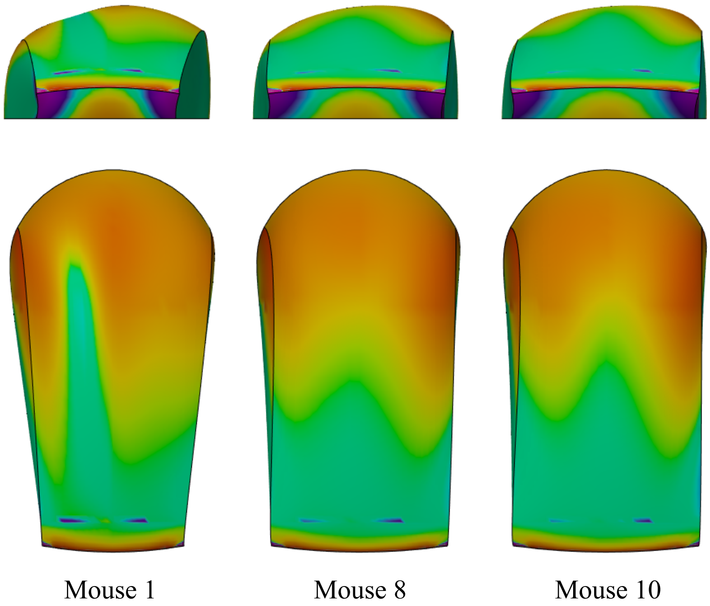


Figure 4.34: Curvature maps of the humps when varying front widths.

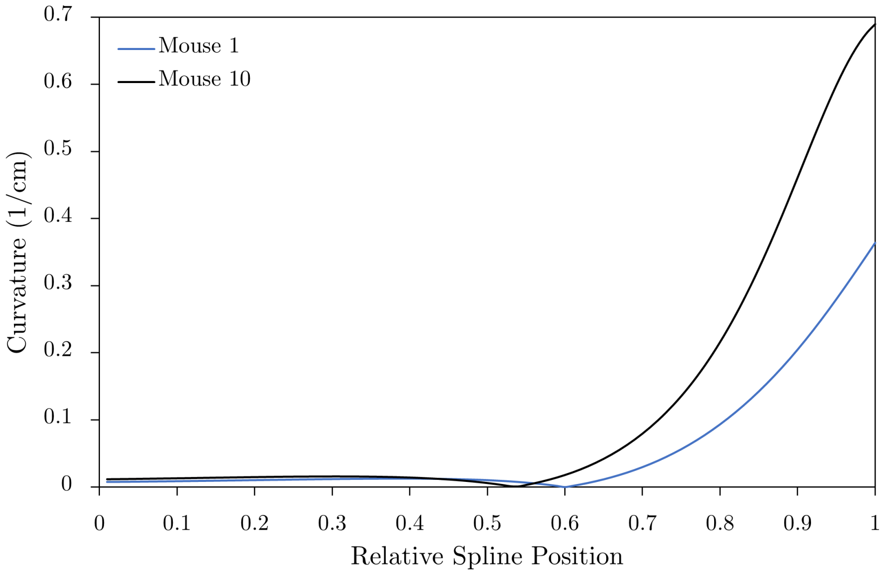


Figure 4.35: Comparison of curvatures of the bottom spline when varying the grip widths of the asymmetrical mice.

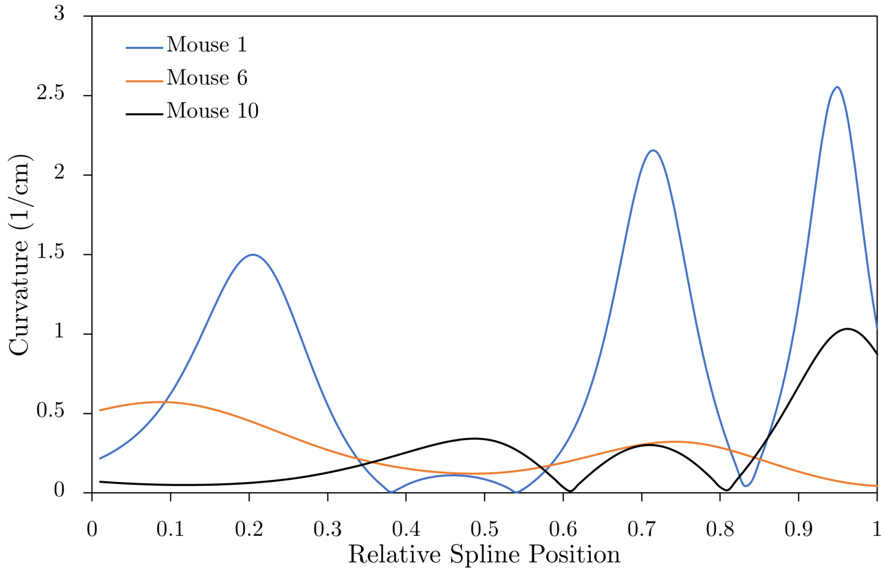


Figure 4.36: Curvature evolution of the humps when varying grip widths.

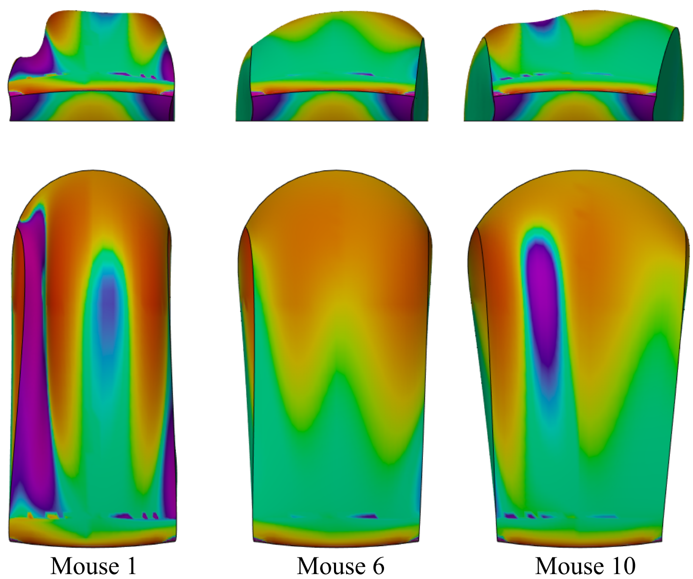


Figure 4.37: Curvature maps of the humps when varying grip widths.

Another issue is the hump curvature. This analysis was made with an hump angled at about 12.5 degrees which is not the best solution. To address that issue, the hump was redesigned to conform to the curvatures of the selected benchmark. Although some of the issues presented here cannot be completely eradicated, the implementation of a verification procedure with certain combinations of measurements can be done in a future work.

This is the absolute main strength of generative design, it allows the designer to acknowledge the possible solutions and decide upon them. This approach led to conclusions that otherwise would not have been possible.

4.2 Algorithmic Approach

From the data that has been gathered and compiled during the generative design process, this section will present the program in its final form, how it is intended to be used and some other aspects that can be improved upon. The generative process implemented was built to explore intervals of values for each variable used in mouse design, following a modelling strategy. This exploration is useful to locate the underlying limitations of the modelling strategy implemented.

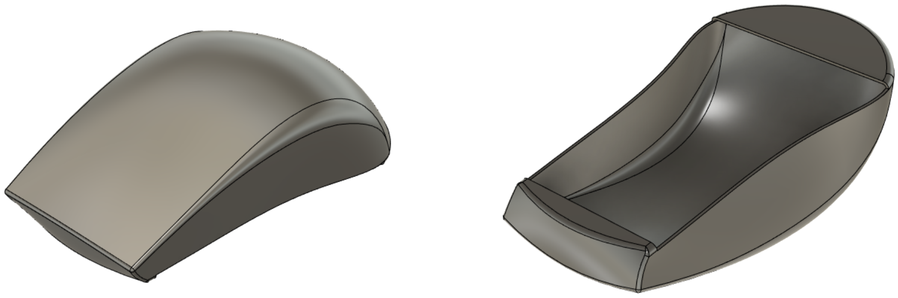
The algorithmic approach uses the same modelling strategy as the generative process, but adds some corrections and other layers to it, focusing the approach towards the end-user. The goal of implementing an algorithmic approach is to have the possibility of introducing a product that has the capability of being customized by multiple users in order to match their personal preferences. The program must be as versatile and robust as possible because it has to support many unique hand-sizes and unique grip styles. The end-user must also be capable of tracking the changes he is performing in real time and adjust any measurement accordingly.

The biggest issue found in this approach was how difficult it was to create a manufacturable shell reliably. Some methods were tried, but none yielded consistent results. In a first deployment every surface was computed independently with thickness added and then combined all the solid bodies together. This solution generated a rough mouse with many clipping edges, that posed a challenge when trying to build fillet features autonomously.

The second deployment implemented a different modelling strategy, of which the generative approach was based upon. Each surface that made up the mouse was computed independently, in a similar fashion to the first deployment. These surfaces were then stitched together and, finally, a thickness would be added to create a solid shell. This strategy ended up limiting the amount of feasible solutions because some shapes were not possible to rebuild. If the mouse was able to be built, then the software would store the edges of the mouse in an array, which is useful when creating a fillet command on specific edges, but due to the modelling strategy implemented, every time the mouse was rebuilt with new dimensions this edge array would be rebuilt, meaning that the edge of one mouse might not be the same as another. This meant that the fillet feature would not work autonomously, defeating part of the purpose of this approach.

Finally, the third deployment of MUS adds upon the second deployment a patch at the base of the mouse. This way, the mouse turns to a solid body, meaning the the edges stored in the array would occupy the same positions, disregarding the dimensions used.

This enabled the capability of automating the fillet features. Although the versatility of shapes possible increased, there are still many instances where the rebuild might not be possible. With the solid body created it is possible to use a shell command to make it hollow. The final procedure is a simple cut extrude to accommodate part modularity (Figure 4.38).



(a) (b)

Figure 4.38: Final design of the symmetric mouse: (a) Top Shell; (b) Bottom side.

With what has been analysed thus far, there was a need to tackle the deformed hump in the asymmetrical typology. The hump was tuned by changing the dependency of the control points of the hump spline (code provided in Appendix A.2). Other thing that needed revision was the sloping of the hump. The hump had a slope of, approximately, 12.5 degrees, which was not ideal in terms of ergonomics and has been revised, currently standing at an angle of 22.5 degrees (Figure 4.39) which is more in line with the conclusions of [Chen and Leung 2007].

Figure 4.39: Revised hump for the asymmetrical mice

Another issue with the modelling procedures that the generative process made clear was the base splines getting deformed when certain variables were modified, seen throughout Section 4.1. To tackle this issue, instead of using a single spline that connects the front of the mouse to the grip arc, this spline has been divided into two different sections.

The first section is a small straight line and the second section is a spline that keeps the same properties as before, but because it is shorter, it will not deform as severely as it did in the generative process. Figure 4.40 shows the new sketches in the context of the base profile.

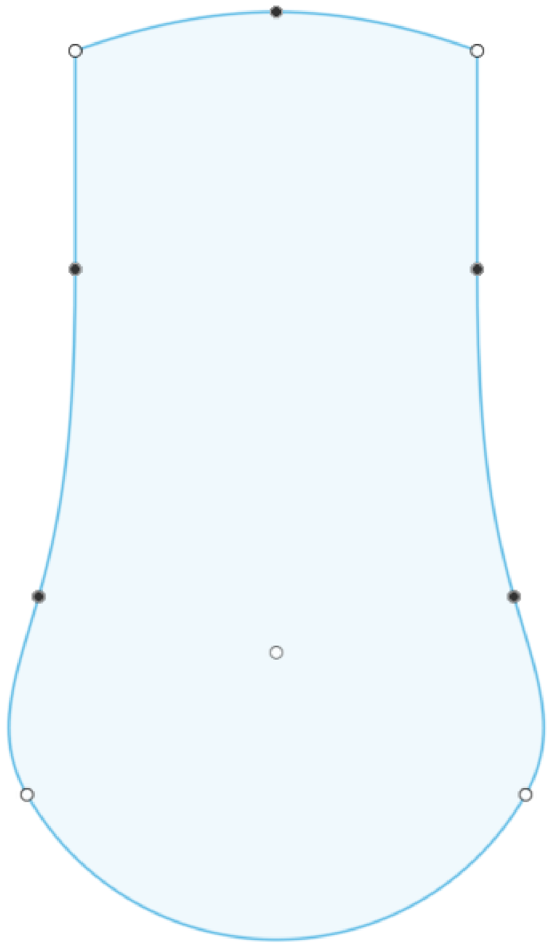


Figure 4.40: The UI for the current deployment.

From this review of previous versions of MUS, a more in depth step-by-step should done to help the reader better understand the entire process in a pedagogic perspective. Figure 4.41 shows the User Interface (UI) with the finished shapes.

Both shape typologies share most of the code, so when explaining the modelling strategy employed it is transverse between both typologies. First thing that is generated by the program is the top surface. This surface has three splines that must be lofted through and then controlled by three rails as seen in Figure 4.42. Once every face is generated independently, the next face to be generated is one of the side surfaces. This

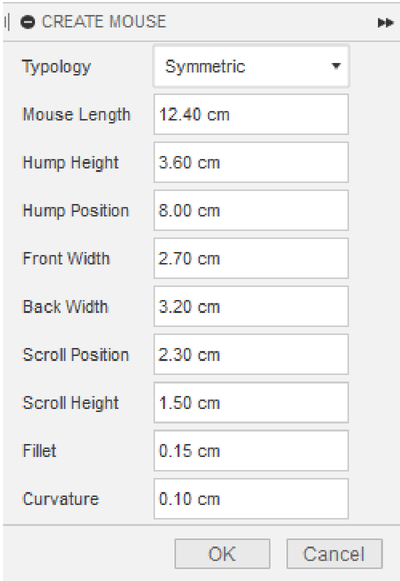


Figure 4.41: The UI for the current deployment.

surface has two splines that intersect themselves at the back and this poses a problem because if both splines are present in the same sketch the software recognises it as a single curve, meaning that it cannot generate the loft. To combat this, a new sketch was initiated with only the spline in the bottom and since that the top spline is shared with the top surface, the edge of that surface was extracted and used to compute the loft so the software no longer recognizes it as a single sketch. To guide this surface two rails were used, one at the front and one at the middle. To build these rails it is still needed the information of the top rail, that is why it is shown in Figure 4.43

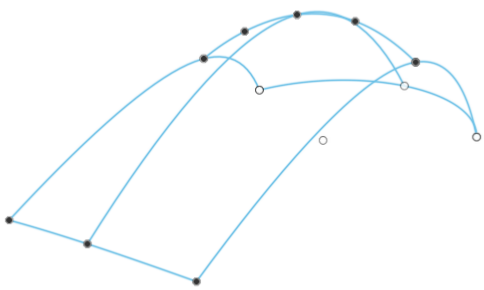


Figure 4.42: Top shell sketch.

In the case of the asymmetrical typology the different side surface had an extra step, where the guide rail at the middle of the mouse is tangent to the rail that controls the hump, creating a bulge where the two surfaces met. To fix this bulge the loft should have tangent chaining to the top surface, meaning that the front rail can no longer be used to guide the surface. In this case, that rail has been abolished and the edge created by the surface will be the new rail (Figure 4.44).

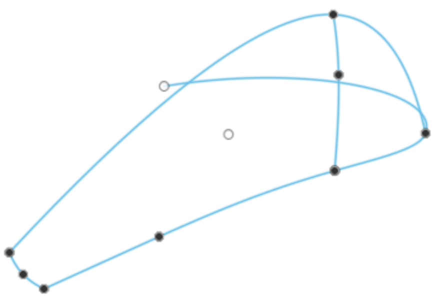


Figure 4.43: Sketch to build the second surface.

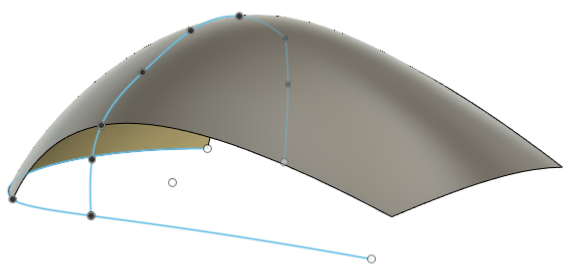


Figure 4.44: Representation of the sketch to obtain the asymmetrical side surface.

When two surfaces are created they must be stitched together to form a new body. This procedure is important because the software can store the information of the edges of the created body in an array. This edge can be then selected and extracted to form a new surface. When building the side surface of the opposite side, the procedure is repeated.

One of the rails used for the top surface becomes a spline that is lofted through to form the front face. Completing this loft results in the final shape of the mouse, but without thickness associated (Figure 4.45). This third deployment looked for another method of adding thickness to the model, because the software could not compute efficiently and reliably a thick shell if the body is comprised of self-intersecting surfaces. To address this, using the edges of the body it was possible to use a patch command to close the surface (Figure 4.46). Upon closure of the surface, the set of surfaces turn into a solid body. From this solid body it is possible to create an hollow body reliably (Figure 4.47).

The hollow body created is not a definitive solution. Unfortunately, to be able to rebuild the body, it must be created with 1 mm thick walls, which is not enough to guarantee structural rigidity, specially when the taking the mouse buttons into account. Although, with some testing it is possible to obtain 2 mm thick walls, it is not reliable and only possible manually because the shell command is dependant on the starting point.

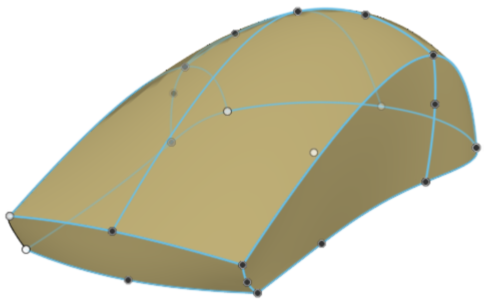


Figure 4.45: Creation of the front surface and its stitching to the rest of the body.

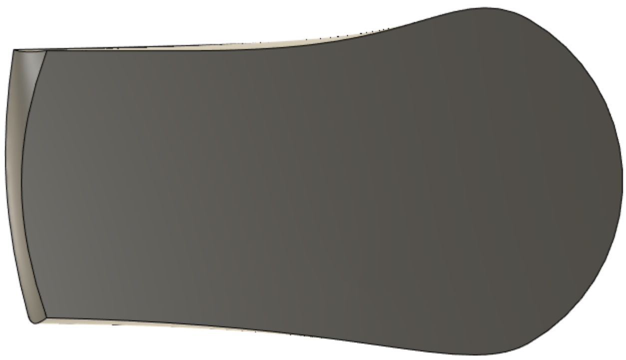


Figure 4.46: Creation of the patch that creates the solid body.

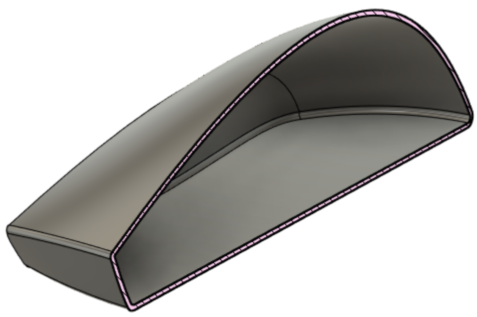


Figure 4.47: Section view of the hollow body.

Moving past this issue, the final procedure at this stage of implementation is the simple cut extrude to accommodate a closed component that holds the electronics of the mouse. Within the framing of this work the most important part is to understand how good is algorithmic design for customization and not evaluate the features or final fit.

The results reported are satisfactory, but can definitely be improved upon. In the UI there are some variables that do not work or their influence is minimal. The variable “Fillet” is not active in this deployment, but it can only control the edges that joins the top surface to the side surfaces. This happens because the other rounded edges are limited to just one value, otherwise the model cannot be rebuilt. This variable controls the “squareness” of the top shell of the mouse, meaning that the smaller the radius of the fillet, the more rectangular is the mouse.

Other variable present in the UI is “Curvature”. This variable controls the bulge on the rail of the side surface and although it is active and working, its influence in the design is negligible, thus the omission from any sort of evaluation. Both variables “Scroll Position” and “Front Height” have a much larger influence on the possible outcomes of the design, they had to be fixed values because of the implementation of the closed component. This module is to be manufactured using traditional methods, so its dimensions are fixed.

Although the implementation has been relatively successful, there are a few things that should be mentioned besides the variables. The basic shapes are supported, but the tool is limited for specialized mice. Vertical mice are not possible to replicate at all with the current state of deployment (e.g. *Logitech MX Master Vertical*, like in Figures 4.48a). The same goes for “horizontal” shapes, specifically the ones that have an extended base for the thumb, similar to the *Logitech MX Master 3* (Figure 4.48b)

(a) (b)

Figure 4.48: Specialist shapes: (a) *Logitech MX Master 3*; (b) *Logitech MX Vertical* [Logitech 2021b, Logitech 2021a]

To validate these results throughout the different phases of design there was a need to physically try and assess the different shapes to possibly improve something beyond that might have been undetected from the evaluations employed. The next section will delve into the physical evaluation side of the work.

4.3 Prototyping

Prototyping takes the concepts from Section 2.2.2 and applies them to the work developed in Sections 4.1 and 4.2. This section uses the Computer Assisted Design (CAD) files created from Section 4.2 and translates them to physical products to validate the created shapes. Table 4.3 documents the dimensions of the specimens used for prototyping in the various stages of development. The prototyping has been executed for the top shell only, as its the most influential part of the process of design and the one that is directly impacted by the customers preference.

Table 4.3: The default measurement used for the generative process of the asymmetrical shape

	Shape 1	Shape 2	Shape 3	Shape 4	MUS 1	MUS 2	MUS 3
Length	124	124	124	124	118	128	122
Hump Pos.	80	65	80	70	80	64	70
Hump Hei.	36	36	36	35	33	38	34
Front Wid.	54	54	54	54	58	54	52
Grip Wid.	64	58	64	60	60	64	60
Front Hei.	10	15	15	15	15	15	15
Curvature			1	1	0	1	1

The first four specimens, named "Shape" represent the various test prototypes that were done to gather more tangible data on the performance of each state of deployment and correct any design oversights in order to improve the results. The last three specimens on Table 4.3 with the name "MUS", were printed based on a single consumer's preference and focuses on exploring various options for different grip styles. Figure 4.49 represents the hand used as a basis to build the mice in this chapter.

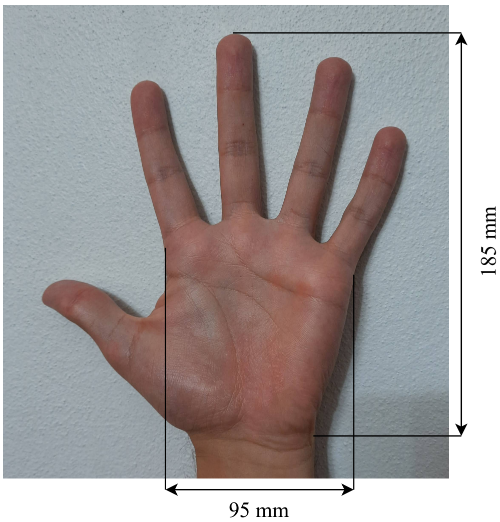


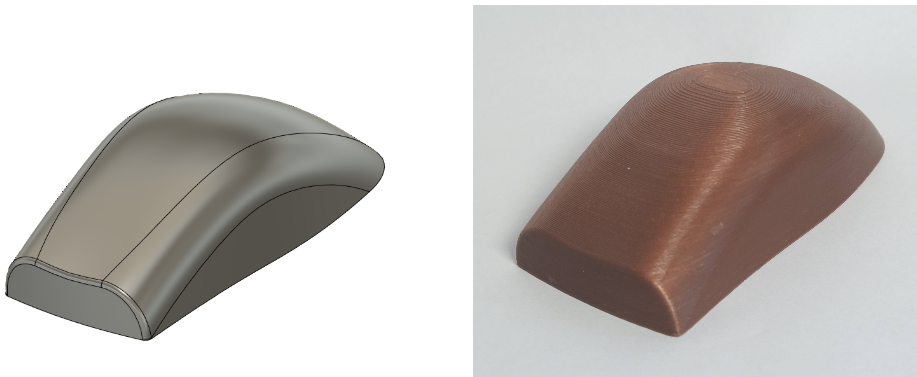
Figure 4.49: Consumer hand size that inspired the prototypes.

The prototyping was executed on a FFF printer, a Creality Ender 3V2, a budget offering in this segment. The material used was PLA at 100% infill and a layer height of 0.2 mm. "Shape 1" and "Shape 2" were printed in a copper colour and represent the first stage of deployment. "Shape 3" and "Shape 4" represent the second deployment and is by far the most developed, but when correcting many of the aspects to reach the third deployment, it was decided to roll back some extra features and focus solely on the final shape, were printed in black. The custom "MUS" mice, which are based in the third deployment, were printed in an aluminium colour. Each colour of prototype was chosen to facilitate the differentiation between each deployment.

Figures 4.50 and 4.51 were the first specimens printed, with intention to understand how the variables interacted physically. “Shape 1” forces the user to employ claw grip on the mouse in order to be comfortable. “Shape 2” manages to correct this with a much thinner grip width and advancing the hump towards the middle the mouse meaning that the hand can rest on top of it. This means that the user can use palm grip and not just claw. A higher front also helps relaxing both the middle and pointer fingers. The biggest issue with this deployment was the modelling, where there were many instances of manual editing in order to print the prototype.

(a) (b)

Figure 4.50: “Shape 1”: (a) CAD model; (b) Physical prototype



(a) (b)

Figure 4.51: “Shape 2”: (a) CAD model; (b) Physical prototype

In Section 4.2 was discussed the changes made to the code in order to improve the results and reach the second stage of deployment. The second deployment allowed for more freedom when creating shapes and the asymmetrical typology has been implemented. “Shape 3” (Figure 4.52b) represents the symmetrical mouse and has the same dimensions as “Shape 1” (apart from the front height), but because the modelling strategy changed the end result is different. Because the second deployment sketches a semi-circumference at the grip, it allows the user to hug the mouse more comfortably. The presence of a taller hump towards the back causes the use of claw grip for users with larger hands, otherwise palm grip is now possible.

(a) (b)

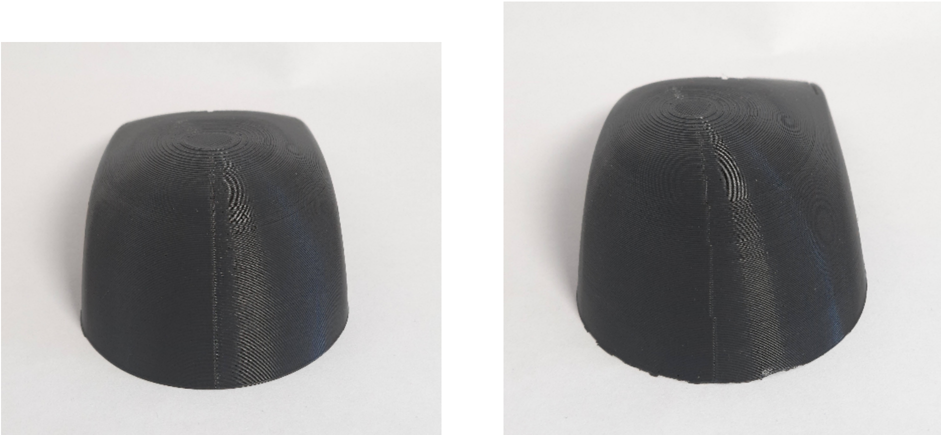
Figure 4.52: “Shape 3”: (a) CAD model; (b) Physical prototype

“Shape 4” (Figure 4.53b) is the first asymmetrical mouse created and because it is designed focusing on the usage of palm grip, the hump was moved slightly forward and the grip width was narrowed. The lower hump height is negligible. The mouse is not asymmetrical enough, with the hump slope not being noticeable (Figure 4.54), apart from a slight shift on the highest point, identifiable by the location of the top layer on the prototype. This also had a negative impact in the side surface on the outside of the mouse, where it provided minimal support for the pinkie and the ring fingers and barely promoted the rotation of the wrist towards a more natural position [Chen and Leung 2007].

(a) (b)

Figure 4.53: “Shape 4”: (a) CAD model; (b) Physical prototype

When confronted with the results from the curvature analysis in Section 4.1, the modelling had some portions that could be corrected. The third deployment allowed for increased flexibility when creating shapes and led to better solutions for the asymmetrical mice. This deployment is able to create custom mice focused in each grip style with the inclusion of a modular system. “MUS 1” is designed for fingertip grip, “MUS 2” is oriented towards palm grip and “MUS 3” tries to go for a claw grip style mouse. As



(a) (b)

Figure 4.54: Comparison of the hump profiles: (a) Symmetrical; (b) Asymmetrical

briefly mentioned before, all of these shapes have been based on the data provided by one consumer, whose had size is present in Figure 4.49.

The first custom mouse is designed towards fingertip use, where the hump is shallower and placed on the back of the mouse. This mouse is also wider at the front giving it a very rectangular look as seen in Figure 4.55. Although it fulfills its goal, the mouse should be thinner in general.

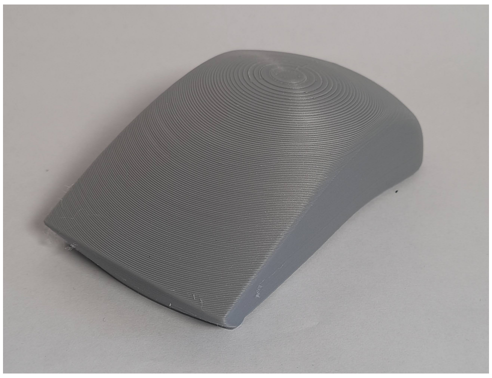


Figure 4.55: Model of a mouse for fingertip grip using the third stage of deployment.

The second custom mouse, represented in Figure 4.56, is built using the third stage of deployment and it is geared towards palm grip users. This mouse is longer, taller and wider than “MUS 1”. To improve the general comfort of the mouse, it is also asymmetrical. This mouse fixes most of the issues noted in the prototype “Shape 4”, with the side surface on the outside providing much better support for the fingers (pinkie and ring finger) and the hump profile allows the wrist to take a much more natural resting position. The hump position should be placed at around 55-60% of the mouse length, instead of the 50% selected, otherwise this mouse provides an excellent result.

Finally, “MUS 3”, seen in Figure 4.57, represents a mouse designed for claw grip, with a shallower hump located near the middle of the mouse and is longer than the

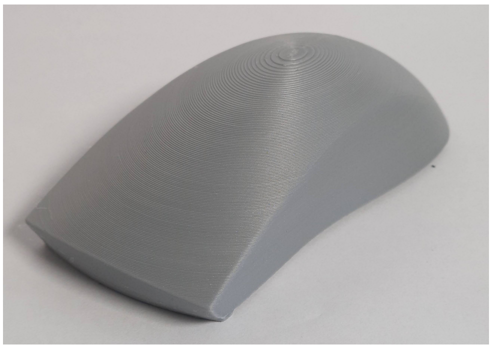


Figure 4.56: Asymmetrical mouse done with the third stage of deployment.

fingertip mouse. Curiously, this mouse turned out better for fingertip that the mouse designed specifically for it, much because it is thinner. For the consumer's preferences, a longer mouse is better suited for claw grip (the default length of 124 mm is perfect).

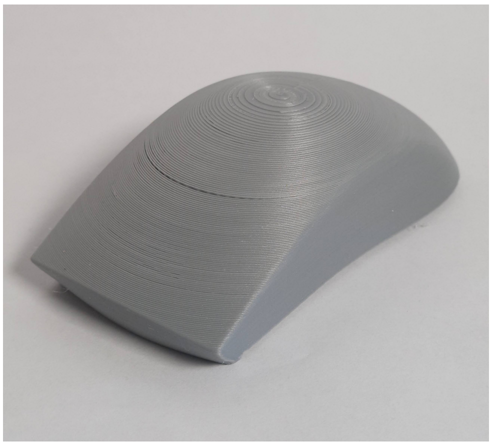
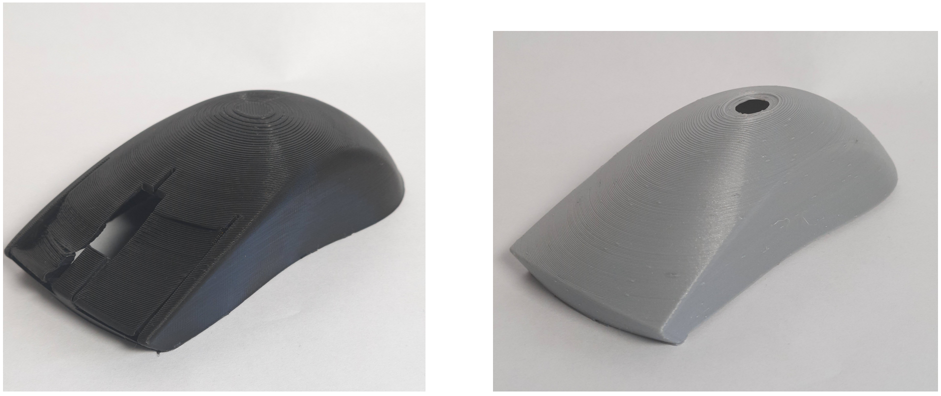


Figure 4.57: Mouse geared towards a claw grip user.

These experiments with physical models represent another way to test and validate ideas besides the use of analytical tools, like the curvature analysis. With a tangible product it is possible to identify further issues that are not possible to discern from the virtual world. For the results to be successful, prototyping needs to becomes an essential tool for this work. There are still a few issues that need to be solved, like the thickness of the shell. By printing the models, it becomes more apparent that the structural fragility of the models is not good.

The second issue is print quality: the nature of manufacturing an object by deposition of material layer by layer leads to a very poor surface finish. To improve the surface finish the layers must be smaller and the mouse must be printed in another position than the one shown in the previous figures, increasing material usage, costs and print time. Adding a post processing stage to sand down the object to have a smooth finish worsens the turnover time and costs even further. There is also a chance for prints to fail, due to a multiple factors. Figures 4.58 represent examples of failed prints, where Figure 4.58a is an example of lack of proper supports and Figure 4.58b is caused by under-extrusion of material. "MUS 3" also suffered from under-extrusion, but had no severe consequences.



(a) (b)

Figure 4.58: Printing errors: (a) bad supports while printing; (b) under-extrusion on the second to last layer

4.4 Modularity

Throughout the development of this dissertation, most of the work was focused in the generated component and the discussion towards the potential for mass customization. Section 4.3 studies the personal preference of a user through the creation of physical mice that have different dimensions. Creating a base for each unique mouse is possible, but requires more time and extra costs. To combat this issue and reach true mass customization potential the existence of modules is an integral part of this concept and must be discussed

A module can be considered a closed component of the design process. It remains immutable, despite the generated component being dynamic. The philosophy of mass customization is to increase value for the customer through a product that has been tailor made for his specifications, without compromising turnover times or costs. This thematic has been explored in more detail in Section 2.2.

The potential for this philosophy applied to this case study means that the generated component can have any shape that the customer desires, whilst the closed component that holds the electronic hardware of the mouse remains the same. Concepts of hybrid manufacturing are possible, having the closed component manufactured using plastic injection moulding and the generated component manufactured with additive manufacturing. This symbiotic relationship between manufacturing methods materializes the possibility to achieve mass customization potential of this product.

The design process of this closed component deviates itself from the Algorithmic Design principles that have been discussed before and has a much more direct approach to its modelling because it remains in a "steady-state". This part has been modeled taking into account the Printed Circuit Board (PCB) mounting points of a *Logitech G305*, as seen in Figure 4.59. To be truly modular it must be easily assembled into the customized shell and the selected fastening method is four bolted connections at the front and back, two in each end. The bolted connection are very easy to design and implement. There are also toolless approaches that can be implemented (e.g snap-fits).

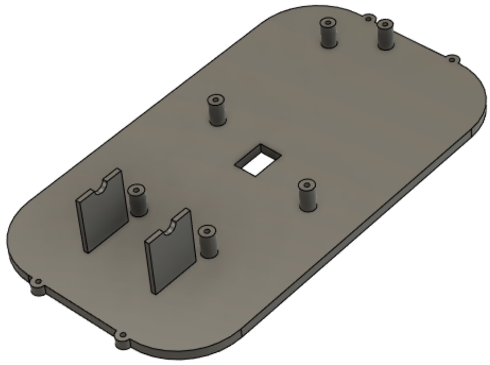


Figure 4.59: Module for mass customization

The major aspect that should be discussed is the fact that part is not yet modelled accounting manufacturing. Because the manufacturing is to be done via plastic injection moulding there are many features that need to be redesigned in a perspective of Design for Manufacturing (DFM). The second issue that might arise is the fact that the PCB used might not the have the perfect fit and end up with a misplaced sensor, that affects the performance of the mouse, but that can be revised and fixed by creating a custom PCB for the closed component.

This case study represents an exercise in mass customization and it is possible to understand the true potential of this approach. If produced in big enough quantities it can help reduce the cost and turnover time of an otherwise expensive operation if it would have been done using just additive manufacturing. And while it might not yield a deeper work related to this module, this part must be revisited to get it ready for eventual production.

Intentionally blank page.

Chapter 5

Final remarks

5.1 Main takeaways

This work is based on the application of Computational Design (CD) techniques in order to enrich the design process applied to product development. There are three major subjects of design that take advantage of these novel techniques that have been identified. Parametric Design (PD) is the subject that defines most of the traditional procedures used in Computer Assisted Design (CAD), where any parameter change that directly intervenes in the design process can be considered part of PD.

Generative Design (GD) is the design subject that is used to explore multiple solutions and generate variety within the same product family. This particular subject is gaining traction within the engineering field because it offers a way to expand the possible solutions based on strict design requirements without limiting designers creativity. This subject's strength relies on presenting multiple solutions to the designer and they are expected to have critical thinking in order to make informed decisions. There are instances of CAD tools including built-in solutions to implement GD, helping the overall popularity of this subject, making it more accessible for multiple applications.

Algorithmic Design (AD) uses programming procedures in order to automate models or create parametric relationships to help and deepen the modelling process. The presented definitions can overlap themselves in many senses, often being mixed up and creating confusion. Although this work assumes an underlined position of where each term stands, it cannot be denied the many applications where these subjects can converge. Even when analysing the work that has been developed, there is an overlap of the different subjects, where the editing of parameters impacts product design, besides the underlying generative and algorithmic design principles.

The case study developed throughout this dissertation was selected because of its potential as a customizable product in order to study and application of AD. The computer mouse is something that is used daily and there are many people that struggle with discomfort and can even lead to injury because it might not correspond to the ideal shape for their physiognomy.

With the challenge of covering multiple instances of human physiognomy, it has been identified a market segment with customization potential which is geared towards ergonomics and user comfort, solving a problem that is inherently human. Once this potential has been identified, the principles of mass customization must be discussed.

Mass customization happens when the user has the capability of taking part in the product conception either directly or indirectly, in order to get a product that corresponds to their own expectations and specifications. Mass customization is a strategy to increase consumer's perception of value of the product, waging in product individualization without sacrificing production costs. For this strategy to be viable in terms of time and costs, products are designed to be modular where the consumer matches them with the goal of creating something that better fits their use cases. This work differs slightly from this approach because consumers do not combine various closed components to create the final product. This case study lays itself in a philosophy of having both a generated component and a closed component.

This way, the consumer is able to generate, without limitations (inside the admissible intervals), the body of the computer mouse that better suits their needs and then complete the final product by adding the closed component, that the user cannot modify staying the same for the entire production line. This closed component contains the electronic components of the mice, no matter their shape.

This philosophy is possible by applying principles of hybrid manufacturing. Hybrid manufacturing is a manufacturing technique that exploits the advantages of multiple manufacturing methods and combines them in order to mitigate their drawbacks. This symbiotic relationship between manufacturing methods is advantageous because it allows to combine the flexibility of additive manufacturing to manufacture the generated component and the cost effectiveness of plastic injection moulding of the closed component, in a way that the final product can be customized minutely, without penalizing too much turnover times and production costs.

The closed component was obtained through traditional CAD methods, because it remains immutable throughout its life cycle, unlike the generated component. The generated component was created through the introduction of the CD techniques that serve as the basis for this work. To achieve the state of development where it currently sits, the generated component was subject the various techniques to evaluate and execute the results reported. In a first stage, AD was useful to settle on a modelling strategy and then the tool was mutated to include generative procedures so it can cover the solution search space and gather more knowledge of the behavior of each variable when altered. This learning experience allowed drawing conclusions about the modelling strategy implemented, with the goal of altering and adding layers to improve results.

The evaluation of the GD was done through comparing the curvature values of key splines that make up the modelling strategy. These curvatures were extracted directly from the CAD software (Fusion 360) and treated in Excel so they could plotted and compared. This evaluation was important to understand any shortcoming of the implemented strategy, get to understand the problem better and identifying the limits of this approach.

With the knowledge of the results from the generative evaluation it is possible to fine tune the tool that implements the AD principles to improve results. This is an iterative process of continuous improvement, where the strategy is developed, the solution search space is evaluated and then the least positive results are corrected and the process is repeated. This continuous improvement process allows the final tool to be robust and versatile, to ensure that the solutions searched by the customers match their expectations. The fact that this work has a single complete iteration can be considered insufficient, but given the multiple dimension involved in this work, this can serve as a demonstrator of potential of including these approaches in product development.

The developed tool stands in the principles of AD by establishing relationships between parameters to reduce the number of variables that can be edited by the consumer, reducing complexity and transforming the tool to be more intuitive for a customer that has little to no experience with CAD software. Another goal of AD is the automation of modelling procedures so it can save time and effort to the designer. The designer does not need to repeat the same base model, being able to program common procedures or procedures that repeat themselves many times. Programming the procedures can also be time consuming in the beginning, but makes up for this drawback, by saving time in boring and repetitive tasks. Despite not being clearly highlighted in this work, because it is mostly oriented to be used by the end-consumer, it is possible to denote these principles by the fact that there is no need to model the default mouse every time a new customization is started.

The algorithmic tool is in a state that can be considered finished, then it is possible to advance to the physical prototyping to validate the results. Prototyping is an advanced evaluation method that can be used when a product is close to be ready for production and allows to report errors or detect other issues that otherwise could not have been possible to detect digitally. All prototypes done in this work follow the input of a single consumer, his physiognomy and his preference, so these results cannot be generalized. These prototypes were manufactured using a domestic 3D printer, neglecting printing quality and general finishing of the product as they were modelled to be printed as quickly as possible. In general, the results obtained were satisfactory, corresponding to the expectations set by the consumer apart from one solution, but it was dictated by human error. According to the methodology set in Chapter 4, the development of the tool MUS did not preview any customer interaction apart from prototyping, but in retrospect, the results obtained are not representative of the versatility of MUS, because it is focused around a single customer. To extract better and more significant results about the state of development of the tool, this dimension must have more consumers giving back feedback about the physical aspect to combat any oversight that might persist.

What has been done in this work serves to start a discussion about the implementation of these CD techniques in the workflow of product development. As the work went on, the knowledge of the CAD software used to develop the AD tool grew stronger. This led to the work not following a clear, continuous flow like described. The first change in modelling strategy was fed from information extracted from an early prototype. Only after that change it was implemented the generative evaluation and followed the natural execution of a work of this nature, using the generative evaluation to improve results cyclically. Then, prototyping should be used to validate results and change any detail that has been overlooked before production.

The tool presented in this work is not perfect as there are still some solutions that are not possible. This means that might exist some shapes that consumers prefer, but are not possible to obtain. The CAD software struggles in some situations where it is not capable to compute fillets or a shell command because the resulting geometry from the changes applied by the customer are not favourable and this should be subject to corrections.

Overall, the results related to the demonstrations of the applications of the CD techniques that have been discussed thus far are very promising. Their implementation in the workflow can lead to an evolution to the way that conception and evaluation of the design process can be done, allowing the creation of better and more creative products. It can also increase product variety by implementing mass customization and hybrid manufacturing principles.

5.2 Future Works

Taking into account what been developed during the course of this work in terms of studying of the application of the different CD techniques in the workflow of product development and the shortcomings that have been noted, there is still further research to be made. Although there were multiple small references to mass customization and hybrid manufacturing, there are not any performance indexes that evaluates their feasibility in any significant way. To ensure the general success of these techniques, a study that encompasses every stage of the design process, from conception (this work) to execution should be held. This continuation work must focus in the manufacturing aspect. An in-depth study of various manufacturing methods should be held and then explore the ones that represent the best balance of customization potential, costs and turnover time, considering the possibility of applying hybrid manufacturing that allows to "marry" various manufacturing methods. Understanding how to interact with the customer helps applying Industry 4.0 principles, by developing a bridge between the consumer and the manufacturing systems.

Customer interaction is another sub-theme connected to this thematic. This represents a social driven component, where it studies how the customization reach the consumer, how he/she should be addressed and how does the consumer reaches the final product. This dissertation disregards the care to explore the interaction between designer and consumer and only focuses in obtaining the shapes through direct manipulation of parameters, not worrying about if the program is user friendly. This future work must create a fully autonomous tool capable of generating a product through the information given by the customer. Then a survey should be held to determine which method of interaction the consumer prefers: active (the consumer is the designer) or passive (the autonomous tool creates the product) and report the results. A final stage should be the development of an User Interface (UI) that is intuitive for all consumer knowledge levels.

A secondary work that can be done based upon this product with the current state of development of the tool is based upon the structural study of the generated computer mice. One of the defects detected is the side walls of the mice not being thick enough to be viable for production. The results from the structural study can introduce new modelling strategies and even the possibility of structural calculation within the developed tool.

Bibliography

[Afrose *et al.* 2016] M. F. Afrose, S. H. Masood, P. Iovenitti, M. Nikzad and I. Sbarski. Effects of part build orientations on fatigue behaviour of FDM-processed PLA material. *Progress in Additive Manufacturing*, 1(1-2):21–28, 2016.

[Arun *et al.* 2018] K. Arun, K. Aravindh, K. Raja, P. A. Jeeva and S. Karthikeyan. Metallization of PLA plastics prepared by FDM-RP process and evaluation of corrosion and hardness characteristics. *Materials Today: Proceedings*, 5(5):13107–13110, 2018.

[Attaran 2017] M. Attaran. The rise of 3-D printing: The advantages of additive manufacturing over traditional manufacturing. *Business Horizons*, 60(5):677–688, 2017.

[Betancourt *et al.* 2014] M. C. Betancourt, L. M. Quintero and G. Cereceda. A discussion on algorithmic thinking in product design process. *Proceedings of International Design Conference, DESIGN*, 2014-Janua:1035–1042, 2014.

[Bézier 1968] P. E. Bézier. How renault uses numerical control for car body design and tooling. *SAE Technical Papers*, 1968.

[Bingol and Krishnamurthy 2019] O. R. Bingol and A. Krishnamurthy. NURBS-Python: An open-source object-oriented NURBS modeling framework in Python. *SoftwareX*, 9:85–94, 2019.

[Caetano *et al.* 2020] I. Caetano, L. Santos and A. Leitão. Computational design in architecture: Defining parametric, generative, and algorithmic design. *Frontiers of Architectural Research*, 9(2):287–300, 2020.

[Cannas *et al.* 2020] V. G. Cannas, A. Masi, M. Pero and T. D. Brunø. Implementing configurators to enable mass customization in the Engineer-to-Order industry: a multiple case study research. *Production Planning & Control*, 0(0):1–21, 2020.

[Cavallini *et al.* 2013] C. Cavallini, A. Giorgetti, P. Citti and F. Nicolaie. Integral aided method for material selection based on quality function deployment and comprehensive VIKOR algorithm. *Materials and Design*, 47:27–34, 2013.

[Chen and Leung 2007] H. M. Chen and C. T. Leung. The effect on forearm and shoulder muscle activity in using different slanted computer mice. *Clinical Biomechanics*, 22(5):518–523, 2007.

[Chernev *et al.* 2012] A. Chernev, U. Böckenholt and J. Goodman. Choice overload: A conceptual review and meta-analysis. *Journal of Consumer Psychology*, 25(2):333–358, 2012.

[Chu *et al.* 2008] C. Chu, G. Graf and D. W. Rosen. Design for additive manufacturing of cellular structures. *Computer-Aided Design and Applications*, 5(5):686–696, 2008.

[Dávila *et al.* 2021] J. Luis Dávila, P. Inforçatti Neto, P. Y. Noritomi, R. T. Coelho and J. V. Lopes da Silva. Algorithms-aided design applied to the tool-paths generation for hybrid manufacturing. *Manufacturing Letters*, 27:53–57, 2021.

[Deppe *et al.* 2020] G. Deppe, T. Reiher, R. Dua, K. Hanna and V. Mensing. *The 7 habits of highly effective generative design*. MSC Software, 2020.

[EpicGear 2021] EpicGear. *Types of Mouse Grip*. `http://www.epicgear.com/en/technologies/types-of-mouse-grip`, 2021. Accessed: 2021-06-11.

[Erboz 2017] G. Erboz. How to Define Industry 40: The Main Pillars of Industry 4.0. In *Managerial Trends in the Development of Enterprises in Globalization Era*, pp. 761–767, 2017.

[Footwear 2021] Impact Footwear. *Breaking The Mold*. `https://impactfootwear.com/pages/breaking-the-mold`, 2021. Accessed: 2021-06-11.

[Fragapane *et al.* 2020] G. Fragapane, D. Ivanov, M. Peron, F. Sgarbossa and J. O. Strandhagen. Increasing flexibility and productivity in Industry 4.0 production networks with autonomous mobile robots and smart intralogistics. *Annals of Operations Research*, 2020.

[Frazer 2016] J. Frazer. Parametric computation: History and future. *Architectural Design*, 86(2):18–23, 2016.

[Frazer *et al.* 2002] J. Frazer, J. Frazer, L. Xiyu, T. Mingxi and P. Janssen. Generative and Evolutionary Techniques for Building Envelope Design. *5th International Generative Art Conference GA2002*, 2002(December):1–16, 2002.

[GamingGem 2020] GamingGem. *The Best Mouse Grip For Gaming - Palm vs Claw vs Fingertip*. `https://gaminggem.com/best-mouse-grip-for-gaming/`, 2020. Accessed: 2021-06-11.

[Gaub 2016] H. Gaub. Customization of mass-produced parts by combining injection molding and additive manufacturing with Industry 4.0 technologies. *Reinforced Plastics*, 60(6):401–404, 2016.

[Glorious 2021] Glorious. *Glorious Model O*. `https://www.pcgamingrace.com/products/glorious-model-o-black`, 2021. Accessed: 2021-06-11.

[Heiskala *et al.* 2007] M. Heiskala, J. Tiihonen, Kaija-Stiina Paloheimo and T. Soinienen. Mass Customization with Configurable Products and Configurators: A Review of Benefits and Challenges. in T. Blecker and G. Friedrich, editores, *Mass Customization Information Systems in Business*, pp. 75–106. Springer, Hersey, 2007.

[Herr and Arch 2002] C. M. Herr and D. Arch. Generative Architectural Design and Complexity Theory. *Proceedings of the Generative Art Conference*, pp. 1–13, 2002.

[Humppi and Österlund 2016] H. Humppi and T. Österlund. Algorithm-Aided BIM. In *Complexity & Simplicity – Proceedings of the 34th eCAADe Conference*, Vol. 2, pp. 601–609, 2016.

[Jabi 2013] W. Jabi. *Parametric Design for Architecture*. Laurence King Publishing, 2013.

[Janssen and Stouffs 2015] P. Janssen and R. Stouffs. Types of parametric modelling. In *International Conference of the Association for Computer-Aided Architectural Design Research*, Hong Kong, 2015.

[Jerez-Mesa *et al.* 2016] R. Jerez-Mesa, J. A. Travieso-Rodriguez, X. Corbella, R. Busqué and G. Gomez-Gras. Finite element analysis of the thermal behavior of a RepRap 3D printer liquefier. *Mechatronics*, 36(April):119–126, 2016.

[Kaveh *et al.* 2015] M. Kaveh, M. Badrossamay, E. Foroozmehr and A. Hemasian Etefagh. Optimization of the printing parameters affecting dimensional accuracy and internal cavity for HIPS material used in fused deposition modeling processes. *Journal of Materials Processing Technology*, 226:280–286, 2015.

[Klahn *et al.* 2014] C. Klahn, B. Leutenecker and M. Meboldt. Design for additive manufacturing - Supporting the substitution of components in series products. *Procedia CIRP*, 21:138–143, 2014.

[Krispel *et al.* 2015] U. Krispel, C. Schinko and T. Ullrich. A survey of algorithmic shapes. *Remote Sensing*, 7(10):12763–12792, 2015.

[Kumar 2007] A. Kumar. From mass customization to mass personalization: A strategic transformation. *International Journal of Flexible Manufacturing Systems*, 19(4):533–547, 2007.

[Kumar and Krishnadas Nair 2017] L. J. Kumar and C. G. Krishnadas Nair. Current Trends of Additive Manufacturing in the Aerospace Industry. in D. Wimpenny, P. Pandey and L. J. Kumar, editores, *Advances in 3D Printing and Additive Manufacturing Technologies*, pp. 39–54. Springer, Singapore, 2017.

[Leitão and Garcia 2021] . Leitão and S. Garcia. Reverse Algorithmic Design. *Design Computing and Cognition DCC*, (4):323–334, 2021.

[Logitech 2021a] Logitech. *Logitech MX Master 3*. `hhttps://www.logitech.com/en-us/products/mice/mx-master-3.910-005692.html`, 2021. Accessed: 2021-09-13.

[Logitech 2021b] Logitech. *Logitech MX Master Vertical*. `https://www.logitech.com/en-us/products/mice/mx-vertical-ergonomic-mouse.910-005447.html`, 2021. Accessed: 2021-09-13.

[MAD CATZ 2021] MAD CATZ. *R.A.T. PRO X3 Supreme*. `https://www.madcatz.com/En/Product/Detail/rat-pro-x3-supreme`, 2021. Accessed: 2021-06-11.

[Mary and Brouhard 2019] H. Mary and G. J. Brouhard. Kappa (κ): Analysis of Curvature in Biological Image Data using B-splines. *bioRxiv*, 2019.

[Minetola and Galati 2018] P. Minetola and M. Galati. A challenge for enhancing the dimensional accuracy of a low-cost 3D printer by means of self-replicated parts. *Additive Manufacturing*, 22(May):256–264, 2018.

[Mitchell 1975] W. J. Mitchell. The theoretical foundation of computer-aided architectural design. *Environment and Planning B: Planning and Design*, 2(2):127–150, 1975.

[Montana-Hoyos and Lemaitre 2011] C. Montana-Hoyos and F. Lemaitre. Systems thinking, disciplinarity and critical thinking in relation to creativity within contemporary arts and design education. *Studies in Learning, Evaluation Innovation and Development*, 8(December):12–25, 2011.

[Montazerolghaem *et al.* 2020] A. Montazerolghaem, M. H. Yaghmaee and A. Leon-Garcia. Green Cloud Multimedia Networking: NFV/SDN Based Energy-Efficient Resource Allocation. *IEEE Transactions on Green Communications and Networking*, 4(3):873–889, 2020.

[Nedovizin *et al.* 2021] A. Nedovizin, N. Gorodetskiy, L. Yng, A Gimenez, D. McArdle, K. Vorobiew, I. Portnov, E. Howick, W. Perdan, M. Giurgi, S. Soluyanov, I. Prytov, V. Doval, D. Moult and A. Zomparelli. *Sverchok*. `https://github.com/nortikin/sverchok`, 2021. Accessed: 2021-06-12.

[Oxman 2017] R. Oxman. Thinking difference: Theories and models of parametric design thinking. *Design Studies*, 52:4–39, 2017.

[Piegl and Tiller 1997] L. Piegl and W. Tiller. *The NURBS Book*. Springer, Berlin, Heidelberg, 2nd edition, 1997.

[Pine 1993] B. J. Pine. *Mass Customization: The New Frontier in Business Competition*. Harvard Business School Press, 1993.

[PORDATA 2021] PORDATA. Agregados domésticos com computador e com ligação à Internet (%). `https://www.pordata.pt/Europa/Agregados+dom%c3%a9sticos+privados+com+computador++com+liga%c3%a7%c3%a3o+%c3%a0+Internet+em+casa+e+com+liga%c3%a7%c3%a3o+%c3%a0+Internet+atrav%c3%a9s+de+banda+larga+(percentagem)-1494`, 2021. Accessed: 2021-06-11.

[Queiroz *et al.* 2015] N. Queiroz, N. Dantas, C. Nome and C. Vaz. Designing a Building envelope using parametric and algorithmic processes. In *Project Information for Interaction – Proceedings of the 19th SIGraDi conference*, pp. 797–801, 2015.

[Razer 2021] Razer. *Razer Viper*. `https://www.razer.com/pc/gaming-mice/viper-family`, 2021. Accessed: 2021-06-11.

[ROG 2021] ROG. ROG Keris Wireless. `https://rog.asus.com/pt/mice-mouse-pads/mice/wireless/rog-keris-wireless-model/spec`, 2021. Accessed: 2021-06-11.

[SciArt 2021] SciArt. *Pareto: An Overview*. `https://www.sciartsoft.com/pareto`, 2021. Accessed: 2021-06-15.

[Selin 2003] A. S. Selin. *Pencil grip: A descriptive model and four empirical studies.* Åbo Akademi University Press, Åbo, 2003.

[Shahrubudin *et al.* 2019] N. Shahrubudin, T. C. Lee and R. Ramlan. An overview on 3D printing technology: Technological, materials, and applications. *Procedia Manufacturing*, 35:1286–1296, 2019.

[Sigmund and Maute 2013] O. Sigmund and K. Maute. Topology optimization approaches: A comparative review. *Structural and Multidisciplinary Optimization*, 48(6):1031–1055, 2013.

[Singh and Singh 2015] S. Singh and R. Singh. Wear modelling of Al-Al2O3 functionally graded material prepared by FDM assisted investment castings using dimensionless analysis. *Journal of Manufacturing Processes*, 20:507–514, 2015.

[Singh *et al.* 2017] S. Singh, S. Ramakrishna and R. Singh. Material issues in additive manufacturing: A review. *Journal of Manufacturing Processes*, 25:185–200, 2017.

[Singh *et al.* 2020] S. Singh, G. Singh, C. Prakash and S. Ramakrishna. Current status and future directions of fused filament fabrication. *Journal of Manufacturing Processes*, 55(January):288–306, 2020.

[Stats 2021] Subreddit Stats. *r/MouseReview subreddit stats (Mice, Mousepads, and More!).* `https://subredditstats.com/r/MouseReview`, 2021. Accessed: 2021-06-11.

[SteamDB 2021] SteamDB. *Concurrent Steam Users.* `https://steamdb.info/app/753/graphs/`, 2021. Accessed: 2021-06-11.

[Szalapaj 2013] P. Szalapaj. *CAD Principles for Architectural Design.* Taylor & Francis, 2013.

[TechPowerUp 2020] TechPowerUp. *Asus ROG Chakram.* `https://tpucdn.com/review/asus-rog-chakram-mouse/images/title.jpg`, 2020. Accessed: 2021-09-13.

[Terzidis 2002] K. Terzidis. The nature of computation Algorithmic Design: A Paradigm Shift in Architecture? pp. 201–207, 2002.

[Tofail *et al.* 2018] S. A. M. Tofail, E. P. Koumoulos, A. Bandyopadhyay, S. Bose, L. O'Donoghue and C. Charitidis. Additive manufacturing: scientific and technological challenges, market uptake and opportunities. *Materials Today*, 21(1):22–37, 2018.

[Tosello *et al.* 2019] G. Tosello, A. Charalambis, L. Kerbache, M. Mischkot, D. B. Pedersen, M. Calaon and H. N. Hansen. Value chain and production cost optimization by integrating additive manufacturing in injection molding process chain. *International Journal of Advanced Manufacturing Technology*, 100(1-4):783–795, 2019.

[Tseng *et al.* 1996] M. M. Tseng, J. Jiao and M. E. Merchant. Design for Mass Customization. *CIRP Annals - Manufacturing Technology*, 45(1):153–156, 1996.

[Tseng *et al.* 2010] M. M. Tseng, R. J. Jiao and C. Wang. Design for mass personalization. *CIRP Annals - Manufacturing Technology*, 59(1):175–178, 2010.

[Tseng *et al.* 2017] M. M. Tseng, Y. Wang and R. J. Jiao. Mass Customization. in The International Academy for Production Engineering, L. Laperrière and G. Reinhart, editores, *CIRP Encyclopedia of Production Engineering*, pp. 1–8. Springer, Berlin, 2017.

[Wang *et al.* 2007] T. M. Wang, J. T. Xi and Y. Jin. A model research for prototype warp deformation in the FDM process. *International Journal of Advanced Manufacturing Technology*, 33(11-12):1087–1096, 2007.

[Wang *et al.* 2017] Y. Wang, H. S. Ma, J. H. Yang and K. S. Wang. Industry 4.0: a way from mass customization to mass personalization production. *Advances in Manufacturing*, 5(4):311–320, 2017.

[Warfield 2021] B. Warfield. *CNCCookbook 2021 CAD Survey*. `https://tinyurl.com/249mcubx`, 2021. Accessed: 2021-09-03.

[Wohlers and Caffrey 2012] T. Wohlers and T. Caffrey. *Additive manufacturing and 3D printing state of the industry: annual worldwide progress report*. Wohlers Associates, 2012.

[Xtrfy 2021] Xtrfy. *M42 Black*. `https://xtrfy.com/mice/m42/`, 2021. Accessed: 2021-06-11.

[Yuan 2020] E. S. Yuan. *A Message to Our Users*. `https://blog.zoom.us/a-message-to-our-users/`, 2020. Accessed: 2021-08-12.

[Zhou *et al.* 2010] W. Zhou, D. Wu, X. Ding and D. W. Rosen. Customer co-design of computer mouse for mass customization without causing mass confusion. In *Proceedings - 2010 International Conference on Manufacturing Automation, ICMA 2010*, pp. 8–15. IEEE, 2010.

[Zhu *et al.* 2013] Z. Zhu, V. G. Dhokia, A. Nassehi and S. T. Newman. A review of hybrid manufacturing processes - State of the art and future perspectives. *International Journal of Computer Integrated Manufacturing*, 26(7):596–615, 2013.

[Zowie 2021] Zowie. *ZOWIE EC2 Mouse for e-Sports*. `https://zowie.benq.com/content/dam/game/en/product/mouse/ec2/gallery/ec2-05-left.png`, 2021. Accessed: 2021-09-13.

Appendix A

Code Examples

Occasionally, conveying the work that has been done to achieve certain results can be puzzling. This appendix has the goal to share with the reader additional information about the practical aspects, attempting to contextualize the decision making process across this dissertation development. The appendix is divided in two sections: the first section shares the procedures done to calculate the curvature using the library NURBS-Python and Matlab. The second section shares some portions of code of MUS.

A.1 Curvature Testing

This section is dedicated to share with the reader the procedures that were used in order to understand what type of algebraic method Fusion 360 used to compute its splines. An arbitrary spline was created in both Fusion 360 and in Python using the library NURBS-Python. Listing A.1 shows the information given to NURBS-Python in order to compute the same spline.

```
1
2 from geomdl import NURBS
3 from geomdl import exchange
4
5 ctrlpts = [[-32.0, 28.8483], [-24.28, 32.36], [-10.454, 37.231], [0,
    35.465], [10.454, 37.231], [24.28, 32.36], [32.0, 28.8483]] #control
    points
6 wei = [1,1,1,1,1,1,1] #control points weights
7 crv = NURBS.Curve() #type of curve
8 crv.degree = 5 #curve degree
9 crv.ctrlpts = ctrlpts #control points of the NURBS
10 crv.knotvector = [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.5, 1.0, 1.0, 1.0, 1.0,
    1.0, 1.0] #knot vector
11 crv.weights = wei
12 crv.delta = 0.005 #interval between points that define the curve
13 curve_points = crv.evalpts
14
15 exchange.export_csv(crv, file_name="control_points.csv", point_type='
    evalpts') #exports the points set by delta
```

Listing A.1: Python code used to create a NURBS curve

This is a simple code to compute a fifth degree spline using NURBS-Python and then using the code in Line 14 it is possible to export a CSV (Comma Separated Values) file with the coordinates of the points that lie on the computed curve. With the curve delta set to 0.005, there are 200 coordinates that are exported to have its curvature calculated in Matlab. This curvature calculation uses as its basis the formulation presented in Equation 4.1, with the code used listed in Listing A.2

```
% extract the coordinates into seperate arrays
xpts = controlpoints{:,{'dim1'}}';
ypts = controlpoints{:,{'dim2'}}';

dh = gradient(xpts); % x first derivative
dht = gradient(dh); % x second derivative

dy = gradient(ypts); % y first derivative
ddy = gradient(dy); % y second derivative

curv = abs((dh.*ddy)-(dy.*dht))./sqrt((dy.^2 + dh.^2).^3) * 10; % curvature calculation
```

Listing A.2: Matlab code used to calculate the curvature from the NURBS curve created with NURBS-Python

The curvature values are multiplied by a factor of 10. The default units used in the Application Programming Interface (API) of Fusion 360 are in centimeters, so the values of curvature that are computed directly from Fusion 360 are given in centimeters. When calculating the curvature values using this Matlab code, the units used are in millimeters. To normalise both curvature values it must be considered the factor of 10.

As far as the Fusion 360 curve used for comparison, it was built manually using the Control Point curve option, but it is interesting to look at the procedure that is able to export the curvature information directly from Fusion 360. This procedure remains the same for every evaluation done throughout the work (Listing A.3).

```
import adsk.core, adsk.fusion, adsk.cam, traceback

def run(context):
    ui = None
    try:
        app = adsk.core.Application.get() # call application
        activeSelection = adsk.fusion.Design.cast(app.activeProduct) # target the active component and sketch
        MySketch = activeSelection.rootComponent.sketches.itemByName('Sketch1') # select the sketch from the environment

        Line = MySketch.sketchCurves.item(6) # select the line from the array of lines stored

        allParms = activeSelection.allParameters # activate parameters

        evaluator = Line.geometry.evaluator # activate evaluator
        (ret, stParam, endParam) = evaluator.getParameterExtents() # get the parameters of the selected item
        step = (endParam - stParam) / 200 # number of parameters to extract

        filename = 'C:/temp/Curv.txt' # append information into a .txt file
        f = open(filename, 'a')

```

```
        param = stParam
        while param < endParam:
            (ret, dire, curv) = evaluator.getCurvature(param) # get curvature at the parameters
            param += step

            f.write(str(param) + ',' + str(curv) + '\n') # write in the file the parameter and the curvature

        f.close()

    except:
        if ui:
            ui.messageBox('Failed:\n{}'.format(traceback.format_exc()))
```

Listing A.3: API code used to export the curvature values from any spline in Fusion 360

The function created is specific of Fusion 360's API with its own library. It is useful to export the information of the curve into a file to be analysed, making it possible to compare two different methods of computing curves.

A.2 Modelling Strategy

As previously alluded, this section sheds more light onto the programming procedures used in MUS, mainly on the parameter relationships. Part of the definitions presented for Parametric Design and Algorithmic Design focus on the scripting logic and on the parametric relationships to transform the designing process, making it more intuitive. This also brings an advantage for the consumer because it reduces the amount of variables available to customise, allowing the design process to be quicker and easier. Listing A.4 lists the Python code developed that showcase some of the parametric relationships between variables.

```
        # Create sketch instance
        sketches = newComp.sketches
        sketch = sketches.add(newComp.xYConstructionPlane)

        sketchSplines = sketch.sketchCurves.sketchFittedSplines

        # Sketch main profile
        pointsProfile = adsk.core.ObjectCollection.create()
        pointsProfile.add(adsk.core.Point3D.create(0 , 0, self.scrollHeight - 0.3))
        pointsProfile.add(adsk.core.Point3D.create(0, self.humpPosition, 0.97 * self.humpHeight))
        pointsProfile.add(adsk.core.Point3D.create(0, self.mouseLength, 0))

        splineProfile = sketchSplines.add(pointsProfile)

        # Sketch side profile right (top)
        pointsSideProfile = adsk.core.ObjectCollection.create()
        pointsSideProfile.add(adsk.core.Point3D.create(self.frontWidth, 0.2, self.scrollHeight - 0.6))
        pointsSideProfile.add(
            adsk.core.Point3D.create(self.backWidth, self.humpPosition,
```

```
                                        self.backWidth - (math.tan(math.pi / 32) * self.backWidth)))
    pointsSideProfile.add(adsk.core.Point3D.create(self.backWidth, 0.85 * self.mouseLength, 0))

    splineSideProfile = sketchSplines.add(pointsSideProfile)

    # Sketch side profile left (top)
    pointsSideProfileL = adsk.core.ObjectCollection.create()
    pointsSideProfileL.add(adsk.core.Point3D.create(-self.frontWidth, 0.5, self.scrollHeight - 0.6))
    pointsSideProfileL.add(
        adsk.core.Point3D.create(-self.backWidth * 0.9, self.humpPosition,
                                        self.backWidth - (math.tan(math.pi / 8) * self.backWidth)))
    pointsSideProfileL.add(adsk.core.Point3D.create(-self.backWidth, 0.85 * self.mouseLength, 0))

    splineSideProfileL = sketchSplines.add(pointsSideProfileL)

    # Sketch the top rail
    pointsTopRail = adsk.core.ObjectCollection.create()
    pointsTopRail.add(
        adsk.core.Point3D.create(self.backWidth, self.humpPosition,
                                        self.backWidth - (math.tan(math.pi / 32) * self.backWidth)))
    pointsTopRail.add(
        adsk.core.Point3D.create(self.frontWidth * 0.6, self.humpPosition, self.humpHeight))
    pointsTopRail.add(adsk.core.Point3D.create(0, self.humpPosition, 0.97 * self.humpHeight))
    pointsTopRail.add(
        adsk.core.Point3D.create(-self.frontWidth * 0.58, self.humpPosition, 0.78 * self.humpHeight))
    pointsTopRail.add(
        adsk.core.Point3D.create(-self.backWidth * 0.9, self.humpPosition,
                                        self.backWidth - (math.tan(math.pi / 8) * self.backWidth)))

    splineTopRail = sketchSplines.add(pointsTopRail)

    # Sketch the spline that connect the three points at the front (top)
    pointsFrontProfileTop = adsk.core.ObjectCollection.create()
    pointsFrontProfileTop.add(adsk.core.Point3D.create(self.frontWidth, 0.2, self.scrollHeight - 0.6))
    pointsFrontProfileTop.add(adsk.core.Point3D.create(0, 0, self.scrollHeight - 0.3))
    pointsFrontProfileTop.add(adsk.core.Point3D.create(-self.frontWidth, 0.5, self.scrollHeight - 0.6))

    splineFrontProfileTop = sketchSplines.add(pointsFrontProfileTop)

    # bottom arc that serves as rail
    arcBackRail = sketch.sketchCurves.sketchArcs
```

```
        arc1 = arcBackRail.addByThreePoints(adsk.core.Point3D.create(self.backWidth, 0.85 * self.mouseLength, 0),
                                            adsk.core.Point3D.create(0, self.mouseLength, 0),
                                            adsk.core.Point3D.create(-self.backWidth, 0.85 * self.mouseLength, 0))

        # Create loft of top surface
        loftInput = newComp.features.loftFeatures.createInput(adsk.fusion.FeatureOperations.NewBodyFeatureOperation)
        section1 = loftInput.loftSections.add(newComp.features.createPath(splineSideProfile, False))
        section2 = loftInput.loftSections.add(newComp.features.createPath(splineProfile, False))
        section3 = loftInput.loftSections.add(newComp.features.createPath(splineSideProfileL, False))
        rail0 = loftInput.centerLineOrRails.addRail(newComp.features.createPath(splineFrontProfileTop, False))
        rail1 = loftInput.centerLineOrRails.addRail(newComp.features.createPath(splineTopRail, False))
        rail2 = loftInput.centerLineOrRails.addRail(newComp.features.createPath(arc1, False))
        loftInput.isSolid = False
        loft1 = newComp.features.loftFeatures.add(loftInput)

        # Get edges from the lofted surface. Useful for the next loft process to create tangent relationships between geometries
        body1 = loft1.bodies.item(0)
        face = loft1.faces.item(0)
        edge = face.edges.item(2)
        edge1 = face.edges.item(0)
```

Listing A.4: Python code used to model the top surface of the asymmetrical mice

There is plenty to dissect from the portion of code presented in Listing A.4. This portion of code builds the top surface of the asymmetrical mouse, with the main profile being the middle spline, side profiles are the side splines, top rail is the hump spline and arc rail is the grip arc. There are splines that take the information of multiple variables and applying simple arithmetic operations they are able to be built. Most of the modelling strategy presented in this portion of code is transverse to the general strategy, where the splines and the lofts are created in similar fashion.