Tom Noë

The frequently replaceable prosthetic hand concept

The design of an appealing solution for growing children with an upper limb deficiency

The frequently replaceable prosthetic hand concept

The design of an appealing solution for growing children with an upper limb deficiency

By

T. Noë

in partial fulfilment of the requirements for the degree of

Master of Science
in BioMedical Engineering

at the Delft University of Technology,
to be defended publicly on Thursday January 31, 2019 at 10:00 AM.

Supervisor:	Dr. ir. Dick H. Plettenburg	TU Delft
Thesis committee:	Prof. dr. Frans C. T. van der Helm,	TU Delft
	Dr. ir. Gerwin Smit,	TU Delft

Contents

1. Introduction

2. Concept Development

3. Final Design

4. Validation

5. Conclusions

Abstract

Children missing an upper limb face a wide range of issues. Besides the difficulties adult prosthetic users face, such as unsatisfactory function, comfort, and appearance, children also have to deal with the challenges associated with growth. The goal of this thesis was to design an appealing solution for these growing children with an upper limb deficiency. With the specific focus on accommodating the physical dimension changes, as well as their ever-changing tastes and preferences.

To find solutions for these issues, a design process was started. Ideas were generated, which were narrowed down into 3 concepts, before the winning concept was further developed into the final design. Crucial aspects of the final design were then validated before finally the thesis work in its entirety was evaluated.

The end result of the design process is the Frequently Replaceable Hand concept (FR Hand). Instead of replacing a prosthetic arm every 18 months, the arm is replaced every few months. This allows the arm to always be the right size and keep up with the changing demands of the growing user. For the concept to work, the prosthetic replacement process needs to become significantly easier, cheaper, and faster. In order to achieve this, two main parts were developed.

First, a natural looking body-powered hand, which is 3D printed using a combination of flexible and rigid parts. The hand can be quickly and easily assembled without using any tools or fasteners. The 3D CAD (Computer Aided Design) model behind the FR Hand is designed from the ground up to be scalable. The hand's dimensions are driven by 7 main independent variables.

Second, a Graphical User Interface (GUI) is developed. This allows the user or their family to generate a perfectly fitting prosthetic hand, without requiring any modeling experience or help from a professional. The resulting 3D model can then be sent to any 3D printing service to be printed and shipped to the user.

Some of the most significant aspects of the FR Hand were validated. The durability of the flexible hinges was tested by cycling the hand 300,000 times, none of the hinges failed. The scalability of the CAD model was tested by generating 1320 hands based on randomized dimensions derived from an anthropometric database, the program was able to generate 96% of the hands without any errors. The hand prototype weighs 99 grams, has an actuation force of 14 Newton and a single finger can handle a tensile load of 157 Newton. The FR hand costs under €6 in material costs and can be delivered to the user for less than €100.

In conclusion, the FR Hand is a functional prosthetic hand system that is aesthetically pleasing, always the right size and keeps up with the user's preferences, which were the goals at the start of this project. Currently, the scalable forearm and socket interface to accompany the FR Hand are missing. Overall, the FR Hand is not ready yet to be deployed in the real world, more development and testing need to be conducted before this is a viable option for growing children missing an upper limb.

1 Introduction

This report documents the thesis work performed by Tom Noë for the Master BioMedical Engineering at the Delft University of Technology. The initial assignment was to work towards a prosthetic solution for children with an upper limb deficiency. To find out more about the problems and solutions for growing children missing an upper limb a literature study was first conducted. This literature study formed the foundation on which the following thesis was built.

1.1. Children with upper limb deficiencies

The literature study estimated that in the Netherlands roughly 10 to a 100 pediatric upper limb deficiency cases occur each year where prosthetic treatment may be an option (Noë, 2016). This includes both congenital and acquired limb loss, as well as varying levels of upper limb deficiency.

The activation of the terminal device typically occurs at an age of around 18 months. The age of fitting these more advanced prostheses is usually based on developmental cues (Shaperman et al. 2003). Muilenburg (2009) states that prostheses are generally replaced every 18 months, though this number seems somewhat anecdotal. These estimates would mean a child uses roughly 11 different prostheses throughout their childhood.

Not every child with an upper limb deficiency decides to wear a prosthesis. Some never wear a prosthesis (16% of the pediatric population), while others wear a prosthesis for some time before rejecting it (45% of all body-powered users, 35% of all myoelectric users) (Biddiss and Chau, 2007). One thing is clear, prosthetic preferences vary per child. Given the opportunity to try body-powered, myoelectric, and passive prostheses results in various patients choosing different types of prostheses. Additionally, some decide to wear no prosthesis at all while some end up using multiple different prostheses for different tasks (Crandall and Tomhave, 2002). This illustrates that there is no “best” prosthesis, it all depends on personal preference and what the user wants from it.

A study by Wagner et al. (2007) found that the main reasons for prosthesis rejection by children with *Unilateral Congenital Below the Elbow Deficiency* (UCBED) were a lack of function and lack of comfort. Another study by Vasluian et al. (2013) found that children with UCBED wore their prosthesis mostly for cosmetic reasons. Weight and limited functionality were other reasons for prosthesis rejection.

When studying the outcome of prosthetic treatment for children with UCBED some interesting results can be found. A study by James et al. (2006) found that the quality of life (QoL) did not vary much between wearers and non-wearers of prostheses. Surprisingly, the differences in quality of life compared to the healthy population are also very small. The only area where children with UCBED scored significantly lower than the general population is in school functioning. In another test within the same study, children with UCBED scored higher in the happiness domain and lower in the physical function domain compared to the general population. This gives some insight into the high rejection rates, if a prosthesis doesn’t significantly improve the quality of life of a user, why would they choose to wear one?

Based on the research performed in the literature study it appears that function, comfort, and cosmetics are the most important factors in successful prosthetic treatment for the end users.

1.2. Problem statement

Based on the conducted literature study it became apparent that there are two areas where prostheses can improve the lives of children with upper limb deficiencies compared to the healthy population, physical function and school functioning. The lack of physical function seems obvious, but children’s issues with school functioning are less prominent. Both of these factors are going to be kept in mind for this thesis. In order to improve upon the currently available prosthetic solutions, it was decided to target the aesthetics of upper limb prostheses for children.

Designing an aesthetically pleasing prosthesis for children is different from designing one for adults. Besides possibly different tastes, one has to keep growth in mind when developing a prosthetic arm for children. In addition to the changing dimensions, the needs and preferences of a child also evolve as they age. There is simply no possibility for a “one size fits all” approach.

“The goal is to develop an aesthetically pleasing prosthesis system that accommodates the growth of children with a below-the-elbow deficiency”

1.3. Currently available products

Besides studying the problems and demographics of children with upper limb deficiencies, the literature study also investigated existing and proposed solutions specifically targeting the growth aspect of pediatric prostheses.

Figure 1 shows a selection of currently available prosthetic arms designed for children. The image credits can be found in Appendix A.1.

Figure 1: Selection of prosthetic arms specifically designed for children. From the top left, clockwise: WILMER appealing prehensor, TRS Child Lite Touch Hand, conventional split hook, 3D printed wrist actuated prosthetic hand, myoelectric arm prosthesis. For image credits see Appendix A.1.

Some of the commercially available terminal devices are available in several different sizes, such as the Lite Touch Hand for children by TRS shown in figure 2 below.

The majority of the currently available solutions fall far short in terms of appearance. Products that do look natural often lack in function or have other downsides such as excessive weight or inadequate sizing.

As stated in the previous section, the goal is to develop an aesthetically pleasing prosthesis that accommodates growth better than current solutions.

Figure 2: TRS prosthetic hands specifically designed for children, available in 4 different sizes

1.4. Requirements

Before starting the design process a list of requirements was made.

Initially, the scope of the requirements was intentionally kept fairly wide. This was done to not limit the generation of possible solutions at the start of the design process. As such the requirements are not specific to an expected solution or technology.

1.4.1. Requirements

The prosthesis should allow the user to grasp both small and large objects. The terminal device should have an adaptive grasp to allow it to hold objects of various shapes. The weight that the user should be able to lift using the prosthesis will vary depending on the user's age, no hard requirements are set. Whether it is required for the prosthesis to be active or passive will depend on the overall concept of the chosen solution. Neither activation of the grasp or the type of actuation is specified by the requirements at the start of the design process.

The prosthesis should be aesthetically pleasing. A natural appearance is preferred. This means an anthropomorphic shape, natural color, and correct size for the user.

The solution should be viable for Dutch children from 4 to 18 years old. The lower end of this age range is chosen because Dutch children go to school at the age of 4. Under the age of four, the options for prostheses are also limited by the mental and physical development of the child. The upper limit is chosen because users over 18 years old can use conventional adult prostheses and the physical growth tapers off.

The design should be reliable for at least 18 months. 18 months is the average lifetime of currently available prostheses (Muilenberg, 2009), which the proposed solution should be able to meet or exceed. Some maintenance is permitted but no critical failures should occur within this timeframe.

The solution should accommodate the growth of a user over an 18-month period. The growth of key dimensions for children between 2 and 12 years old is charted in Appendix B.1. The datapoints in this chart are based on data from the DINED anthropometric database (DINED, n.d.). From the chart, it can be concluded that the growth of arm length and hand length between the ages of 2 and 12 years is roughly linear. In order to accommodate growth over any period of 18 months, the future solution should work for an arm length change of 36 mm and a hand length change of 9 mm.

The solution should cost less than €3500. The solution should be accessible to any user who wishes to use it. The €3500 number is derived from the allowance for an arm prosthesis by a particular Dutch health insurance provider. Arm prostheses under €3500 don't need special authorization from this insurer if prescribed by a contracted health care provider (CZ Groep, 2018).

The design should be as light as possible for comfort. The entire prosthesis should weigh less than roughly 2% of the user's body weight. This number is derived from the combined forearm and hand weight percentages of male (1.62%+0.61%=2.23%) and female (1.38%+0.56%=1.94%) adults (de Leva, 1996). This is a rough estimation of the maximum allowable weight of the prosthesis. For an average 10-year-old male user who weighs 36 kg (DINED, n.d.), this would mean the entire prosthesis (socket, adapter, terminal device) should weigh less than 720g. Ideally, a prosthesis is much lighter than this. Just a passive prosthetic hand for children from Otto Bock weighs roughly 185 grams, the goal for a future solution is to beat this (Otto Bock, 2014).

The design should cater to both male and female users, both from an aesthetic and physiological standpoint.

The design should require less than 1 hour of travel to be fitted or adjusted. In the Netherlands, this shouldn't be a problem. However, this limitation means the solution can't rely on extremely specialized equipment or care.

1.4.2. Wishes

The user wishes the prosthesis…

- Looks natural
- Is Lightweight
- Helps them fit in at school
- Is easy to put on or take off
- Is compatible with multiple types of actuation. E.g. body powered, myoelectric, passive
- Can be fitted or adjusted at home

2 Concept Development

In order to find a better solution for growing children missing an upper limb, a design study was conducted. Based on the literature research, problem definition, and requirements, a wide selection of ideas was generated. These ideas were narrowed down and further developed into three concepts. Based on the previously defined criteria a selection was made for the winning concept. This cycle of diverging (widening scope of thinking), and converging (narrowing down towards solutions) was repeated multiple times over different iterations and different parts of the design before eventually arriving at the final design.

This section of the report describes the concept development part of this process.

2.2. Concepts

The three concepts and the selection process are described in more detail in Appendix C.1.

Concept 1 is a natural looking adjustable arm prosthesis. By using a mechanically adjustable forearm and palm, a wider range of users can be accommodated, as well as allowing for the physical growth of the user. The prosthesis can be used passively or actively, depending on the preference of the user. When the hand is used actively it is actuated through a Bowden cable, either by an electric motor or by the body. The cable doesn't need to be adjusted when the prosthesis is lengthened or shortened. Finally, the prosthesis is covered with a cosmetic glove for a natural appearance. Concept 1 is depicted in figure 3.

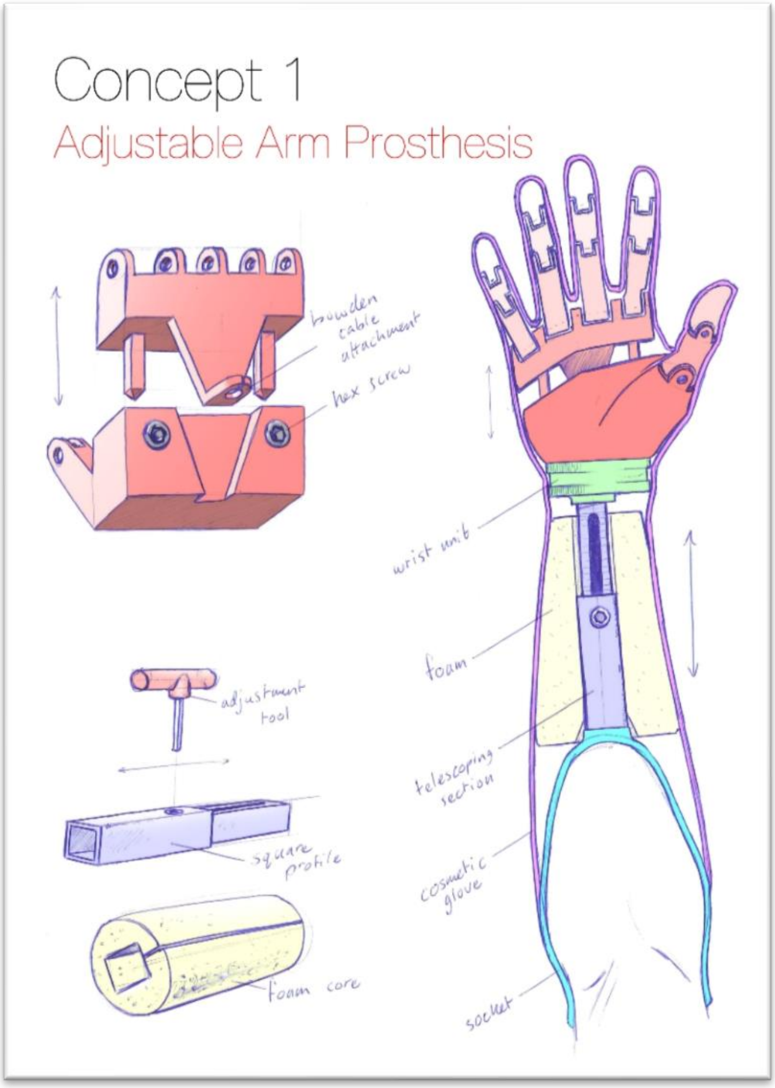


Figure 3: Concept 1, adjustable arm prosthesis

Concept 2 consists of a modular arm system. It utilizes a set of standardized adaptor mounts that allow components to be easily exchanged. For instance, as the arm becomes too short, a longer forearm section can be fitted without the aid of a professional. Different types of terminal devices can be fitted and exchanged. This way the user can try out different technologies or swap out different terminal devices on-the-go based on the situation. The adaptors use a bayonet type fitting that can be unlocked by a button. This concept is shown in figure 4.

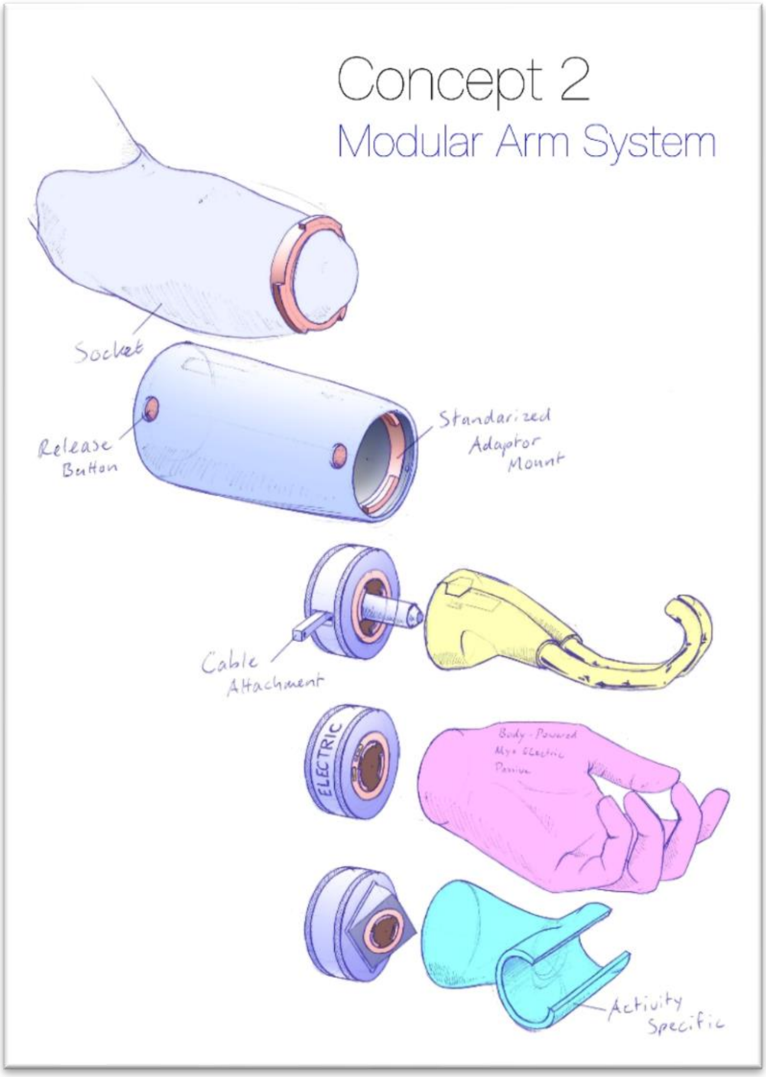


Figure 4: Concept 2, modular arm system

Concept 3 accommodates growth by frequently replacing the user's prosthesis. What prohibits frequent replacements of current prostheses are their costs, as well as the time and effort it takes to get a new prosthesis fitted. This concept utilizes software combined with 3D-scanning and –printing technologies to allow affordable frequent replacements. A 3D scan is made of the residual limb and, optionally, of the other healthy limb. The software is then used to essentially replace the professional labor by creating a 3D CAD (Computer Aided Design) model of the prosthesis that correctly fits and aligns with the residual limb. The result is a ready-to-print file which can be sent off to a 3D printing service. The prosthesis is 3D printed in a single go and doesn't require any complicated assembly. The entire process doesn't require any specialized computer skills from the user or their family. After receiving the prosthesis in the mail, the user wears the prosthesis until it becomes too small or breaks and then repeats the process. This cyclical process can be seen in figure 5.

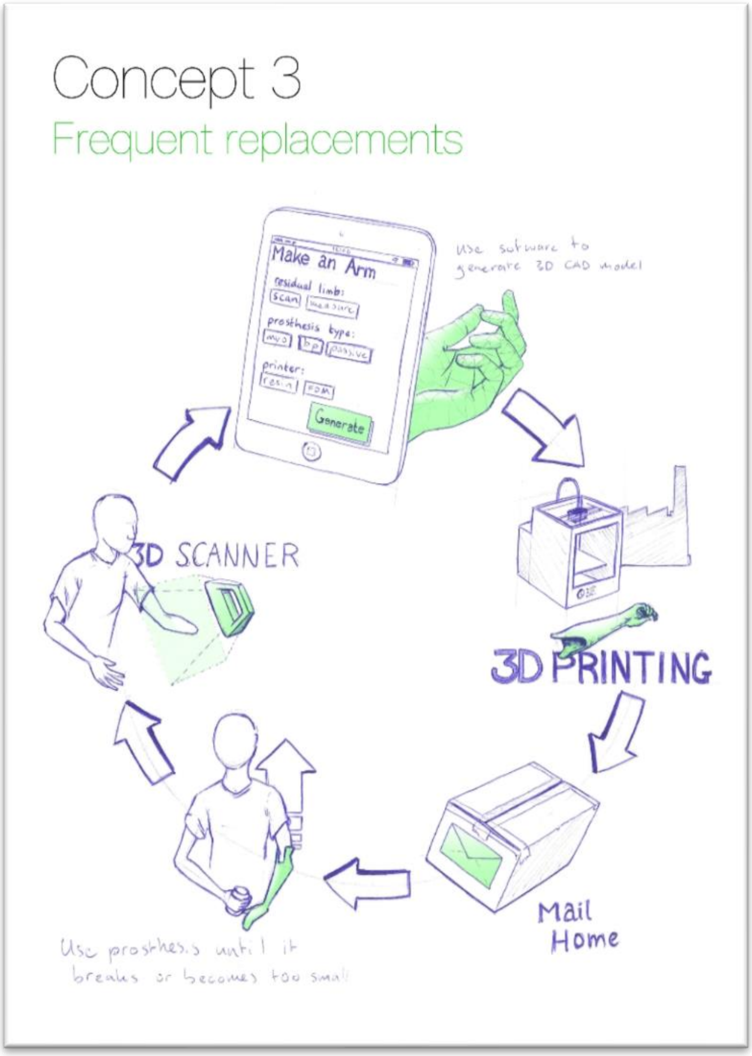


Figure 5: Concept 3, frequent replacements

2.3. Selection

The three concepts are judged based on 5 criteria. The scores are displayed in a Harris profile, with the criteria sorted from the most important at the top to the least important at the bottom. A more detailed description of the criteria can be found in Appendix C.2. Based on this Harris profile a selection of the most promising concept is made.

The Harris profile judging the different concepts is shown below in figure 6.

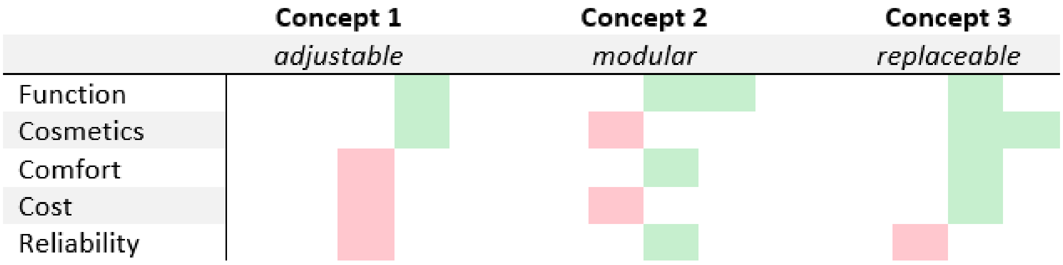


Figure 6: Harris profile comparing the three different concepts

By looking at the Harris profile it becomes clear that concept 1 is the least suitable solution for growing children missing an upper limb. Compared to conventional prostheses the adjustable prosthesis could be slightly more aesthetically pleasing due to its adjustability but this concept adds little in function and is a step backward in comfort, costs, and reliability.

The modular approach of concept 2 could result in a very functional prosthesis due to the easily swappable terminal devices but this approach also has significant drawbacks (see the more detailed concept evaluations in Appendix C.1.). The frequent replacements of Concept 3, on the other hand, don’t add much functionality over conventional prostheses but this concept also has fewer downsides.

In the end, it was decided to move forward with the frequent prosthetic replacements of concept 3 because it aligns most with the interests of growing children. It best accommodates the changes a child goes through as they age, both in physical growth and in terms of other needs. Its customizability can be used to suit the individual aesthetic preferences. The direct input in the prosthetic process also increases the level of involvement for both the child and their family.

While a modular prosthesis (concept 2) could be an improvement over current prostheses, it doesn’t represent enough of a step change to be specifically interesting for growing children.

2.4. Limiting the scope

The previous section selected the winning concept in general but it is not yet fully developed into a final design. For this thesis, some parts of the concept need to be developed further while the scope of other aspects needs to be limited.

An important part of this concept is the 3D-scanning of the residual limb. In addition to the 3D-scan itself, creating a properly fitting socket from the raw data is also critical. Developing these parts of the concept would be a big challenge that falls outside the focus of this thesis. Another member of the Biomedical Engineering department at the Delft University of Technology, J.S. Cuellar, is currently working on this as part of a PhD project. The focus of this thesis will be directed more towards the terminal device. For these reasons, the steps from the residual limb to a 3D CAD model of the socket will be considered a black box.

While the frequent affordable replacements approach might work for different levels of missing upper limbs, this design will focus on one type of deficiency, below-the-elbow deficiencies. The chosen approach could work for both unilateral and bilateral deficiencies, but the focus will be on users with a unilateral deficiency.

Appearance is one of the driving forces behind this project, which is why the main goal is to have a prosthetic arm that is always the right size. Another consequence of this focus is that the terminal device will be developed to look like a natural hand, as opposed to for instance a split hook prosthesis. Other types of terminal devices can be very valuable to children and adults, but since this thesis focusses on appearance, the design of a “hand-like” terminal device is prioritized. A decision was made to focus on a body-powered system for its simplicity, low costs, low weight, and reliability. Myo-electric solutions might be preferred by some users but the properties of a body-powered system suit the strengths of this concept the best.

The final design for the thesis is intended to consist of two main parts:

- A graphical user interface (GUI) that allows the user to generate a custom hand based on measurements, without requiring any 3D modeling experience by the user.
- A working hand-like terminal device that’s scalable by the GUI, that can be easily printed by currently available 3D printers and printing services, which doesn’t require complicated assembly.

2.5. Hinge Development

An essential part of the chosen concept is its ease of assembly. Current 3D printed prostheses often require dozens of screws and need to be assembled by experienced tinkerers. To overcome this challenge, one needs to design functional hinges which don't require assembly. In order to find the best solution, various hinge concepts were generated and tested by 3D printing multiple prototypes.

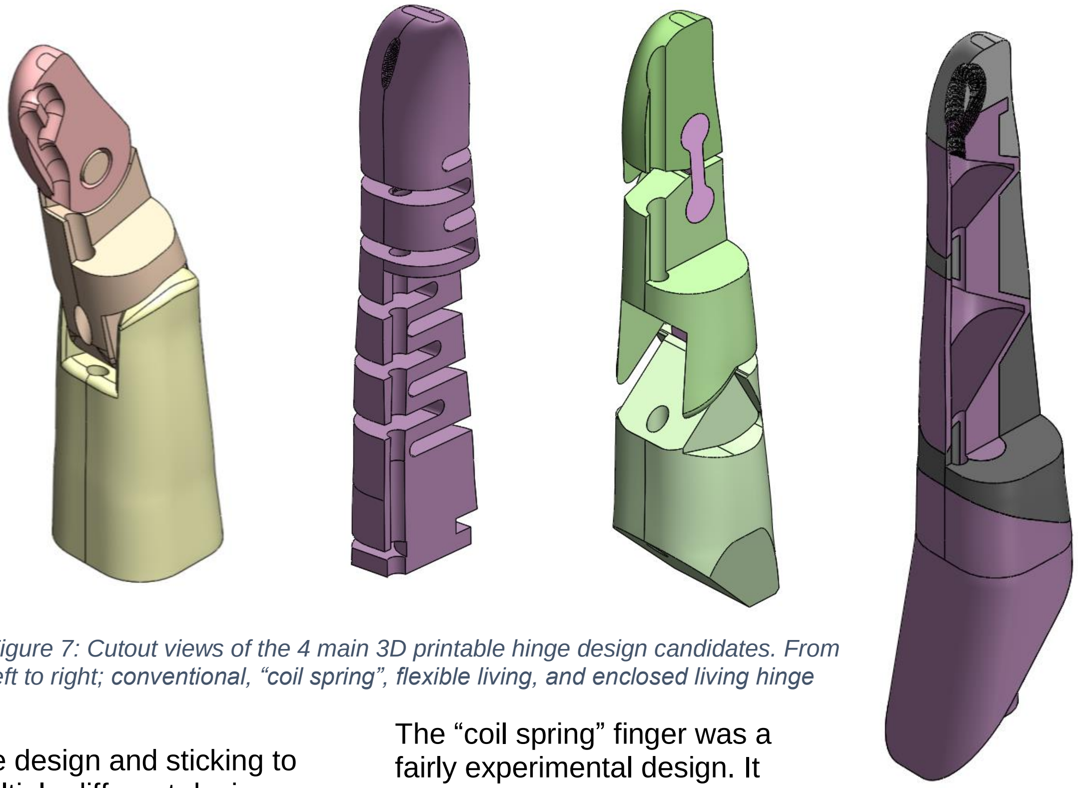


Figure 7: Cutout views of the 4 main 3D printable hinge design candidates. From left to right; conventional, "coil spring", flexible living, and enclosed living hinge

Instead of choosing a hinge design and sticking to it, it was decided to test multiple different designs over several iterations and choosing the winning design after some practical experience. Through the use of 3D printing, this can be done relatively quickly. In total, four types of hinges were tested, a conventional hinge printed in place, a "coil spring" finger, a flexible living hinge, and an enclosed living hinge finger. The different types of hinges are shown in figure 7.

The print-in-place hinge is essentially a conventional pin hinge. The only difference is that the hinge isn't assembled, instead, the different parts are printed in place. This hinge design went through several iterations. Initially, the hinge was printed using PLA (PolyLactic Acid, a rigid plastic) and water-soluble support material on an Ultimaker 3 3D printer. Four test hinges were printed with varying hinge clearances. The use of support material proved impractical. Ideally, a tighter clearance results in a smoother hinge, but the support material could only be removed when using larger clearances. Additionally, the removal of the support material was messy and labor intensive, the opposite of the intended benefits of the print-in-place strategy. Using a single rigid plastic (PLA) with a relatively tight (0.38mm) clearance proved to be easier to print, resulting in a higher quality print, and in the end, a smoother hinge. This type of design still requires some sort of spring to pull the fingers back to their original position after flexion.

The "coil spring" finger was a fairly experimental design. It was unclear if such a hinge would be useful but with 3D printing such solutions can be easily tested. The spring was initially printed using a flexible material (NinjaFlex) on a FlashForge Creator Pro 3D printer. The thickness and design of the "spring" were varied over several iterations but the finger proved to "floppy". The finger could flex in all directions, as well as shorten and lengthen which is not ideal for a finger. A modified version of this finger was printed in steel but ended up being too rigid.

The living hinge design uses the elastic deformation properties of a material to act as a flexible hinge. This allows the design to be relatively simple. The hinge also returns to its resting position without relying on a separate spring. Living hinges have been used previously in low-cost 3D printed prostheses such as the flexy-hand by Gyrobot (2014). The living hinge finger design was initially printed in two materials in one print using a dual extruder (on a FlashForge Creator Pro). The "phalanges" were printed in rigid plastic (PLA), the flexible hinges in NinjaFlex (see figure 7, 3rd from the left, green = rigid, purple = flexible). Later on, the finger was entirely printed using NinjaFlex, making it easier to print. By varying the infill percentage of the 3D print, the phalanges could still be made fairly rigid while the hinges themselves stayed identical to the previous designs. The resulting hinges turned out to be very flexible in the desired hinge axis and seemed quite robust.

The *enclosed* living hinge functions similarly as the living hinge design, except that the cutouts are enclosed in a thin layer of flexible material. This results in a finger that looks anatomically correct from the outside. This type of hinge design was also part of the J.D. ten Kate's thesis (2016). The enclosed living hinge fingers were initially printed in two materials. Later on, the design was changed to be printed in a single flexible material, a similar design progression as with the non-enclosed living hinge finger described previously. The enclosed living hinge finger has the most aesthetically pleasing exterior out of all prototypes, but the resulting actuation force was also the highest out of all hinge designs. In the end, the enclosed living hinge design was just too rigid to be practical.

Based on the 3D printing experiments and further testing, a hinge design for the final design was selected. The living hinge design was chosen for its (relatively) low actuation force, simplicity, robustness, and ease of printing.

Figure 8 shows some of the 3D printed prototypes that were used to determine the best hinge design for this project.

Finally, an existing 3D hand model was modified to make use of living hinges. This model was then 3D printed to check the feasibility of a prosthetic hand using these living hinges. This prototype is shown in figure 9. The hand is printed in two pieces, the green material is flexible, the red material is rigid.

Figure 8: A selection of 3D printed hinge design prototypes. Every design was printed in place, no assembly was required. The green material is NinjaFlex flexible material, the red material is rigid PLA plastic. The red residues left on some of the green parts are left over support material. One variation of the "coil spring" finger was printed in steel.

Figure 9: A prototype made using an existing adult 3D hand model, modified to use living hinges and to be 3D printed in two parts. The green part is made out of NinjaFlex flexible material, the red part is printed in rigid PLA plastic. Braided Teflon-coated fishing wire is used for the actuation strings.

3 Final Design

This section describes the final design of the prosthetic hand in detail. All ideas, concepts, and selections have culminated in the following design. As described in the previous section, this thesis focusses on two parts of the frequently replaceable prosthesis system. The scalable and 3D printable hand design, and the graphical user interface (GUI) which allows the user to generate the desired hand.

3.1. The Hand

The final design has been named the FR Hand. The name refers to the Frequently Replaceable nature of this concept. The FR Hand system does not just refer to the physical hand itself, it also includes the entire process surrounding it. Figure 10 and 11 show the physical prototype for the FR Hand.

Figure 10: FR Hand, pulling the string actuates the fingers and thumb

Figure 11: The FR Hand. The black parts are 3D printed in a flexible material, the grey parts are printed in rigid PLA plastic.

3.1.1. Design

The prosthetic terminal device is designed to look similar to a natural human hand. Appearance is one of the driving forces behind this project, and while appearance can be subjective, a natural look is typically most appealing to users. Due to the chosen production process the prosthetic hand still allows for differentiation in color should the user so desire. Flexible filaments are available in various skin tones, as well as a wide range of other colors.

The fingers grasp in a “3 jaw chuck” type pinch. All fingers including the thumb are actively actuated, with the load being distributed through a whippletree mechanism. This allows the grip of the prosthetic hand to conform to the shape of an object. Figure 12 shows the FR Hand holding a smartphone.

Figure 12: FR Hand holding a smartphone

Based on testing and experimentation in the previous section of this report, a living hinge design was chosen. The fingers are able to bend by flexing a thin section of material. When the fingers are not actuated, they return back to their natural (extended) position. Based on experiments in the validation section of this report (section 4.1) a hinge thickness of 1mm is used, with a hinge length of 1.25mm. A cross section of a finger including the hinge dimensions is shown in figure 14. These dimensions were chosen for a balance between low actuation force and long-term durability.

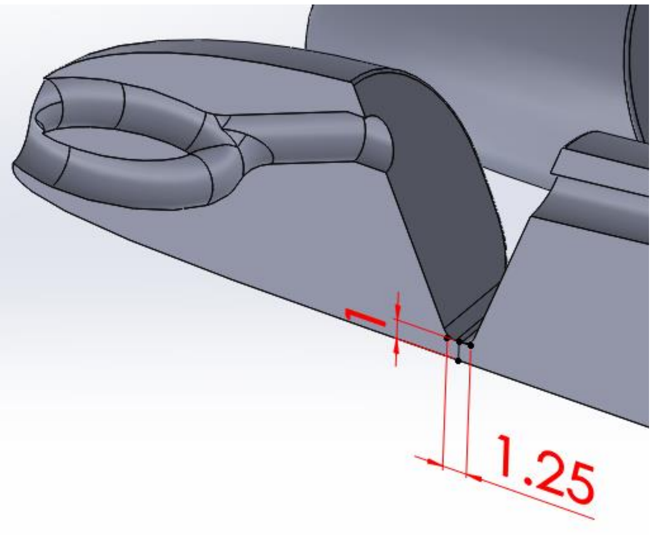


Figure 14: Cross section view of a living hinge, with hinge dimensions

The hand is constructed of mainly three parts, the *finger set*, the *palm*, and the *thumb*. These three parts are assembled without using any screws or tools. The *finger set* is one unit, printed in one go, using a single flexible material. This means the flexible hinges are the same material as the “phalanges” of the fingers. The hinges are printed solidly (100% infill), the “phalanges” are printed with a 40% infill. The phalanges being semi-hollow allows them to be lighter weight and feel soft to touch, while still being strong enough to function as part of the fingers.

Figure 13: FR Hand partially disassembled. Assembly or disassembly doesn’t require fasteners or tools

The hand is made in three main parts to allow all pieces to be 3D printed with minimal support by practically any 3D printer. It also allows the print orientation to be optimized for strength and durability of the hinges. Finally, printing the hand in three pieces allows for easier rigging of the wires and load distribution mechanisms which actuate the fingers. See figure 13 for a partially disassembled prototype.

The finger set connects to the palm using a rail interface. The finger set, completely assembled including wires and load distribution mechanism, is pushed down into the palm and then slid to the ulnar (“pinky finger”) side of the hand. The finger set is held in place by the geometry of the rail, friction prevents the finger set from sliding to the radial side and potentially disconnecting the finger set from the palm. The thumb also uses a rail system but is slid in a single motion into a slot in the palm from above. The thumb is primarily held in place by friction, but a small latch is used on the bottom of the thumb to prevent the thumb from sliding up. The rail interfaces between the different parts are shown in figure 15.

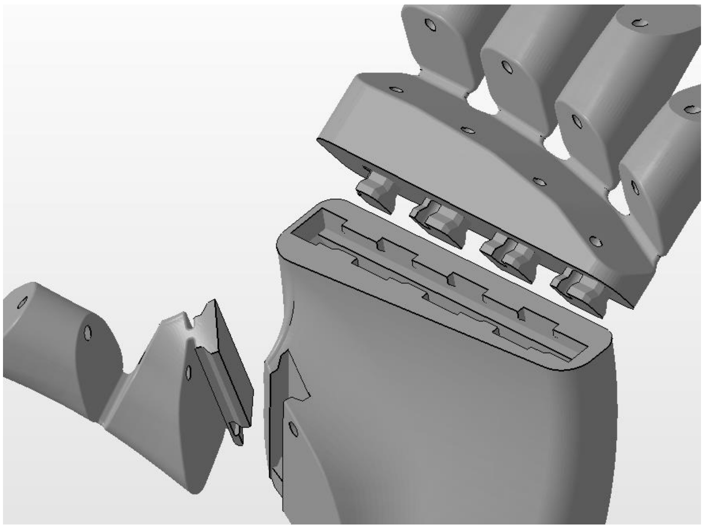


Figure 15: Rail interfaces between the three main parts of the hand

The fingers are actuated using wires running through channels in each finger. For the prototypes, a braided fishing line is used (Caperlan 0.3mm (Decathlon, n.d.)). A 3D printed whippletree mechanism is used to distribute load amongst each individual finger of the *finger set*. When one finger is stopped from bending, for instance by the geometry of an object, the other fingers continue to bend to fully grasp an object. The linkages of the whippletree can be quickly snapped into place without the use of screws. The line coming from the *finger set* whippletree and the *thumb* are tied together in the current prototypes. This means the *thumb* and *finger set* are not able to operate independently, the hand always "grasps", you can't actuate just the thumb.

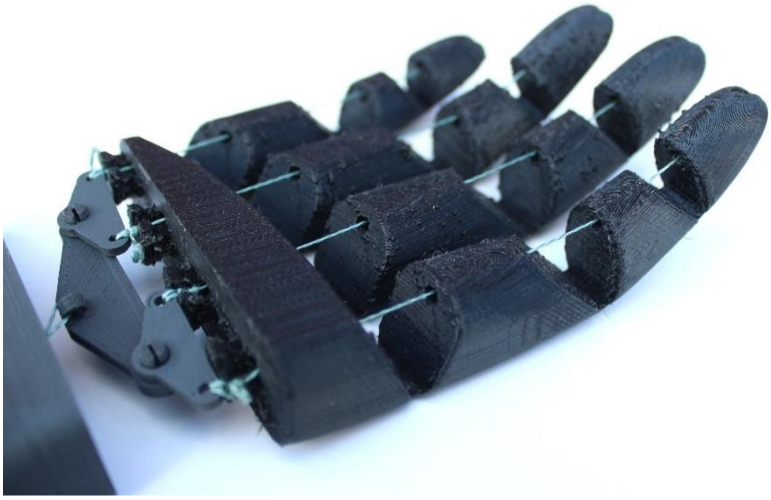


Figure 17: Actuation wires running through the fingers, connecting to the whippletree mechanism, with a single wire running down into the palm

This single line exits the bottom of the hand through the center of an M12 or ½"-20 mounting stud, depending on the wrist unit. This line can be connected to the end of a Bowden cable which in turn can be connected to a shoulder harness. Figures 16 & 17 show the actuation wires running through the hand.

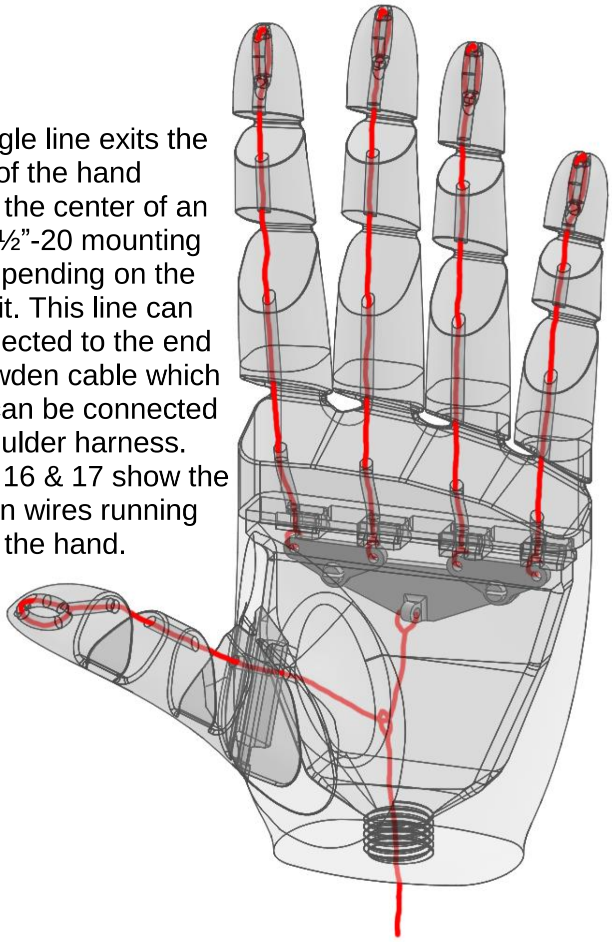


Figure 16: Actuation wire routing through the prosthetic hand

The hand functions as a voluntary closing terminal device. This means the hand is open in its resting state. When desired, the hand can be closed with a specific amount of pinch force controlled by the user.

What differentiates this hand from others is that it's designed from the start to be scalable. The entire 3D CAD hand model is based on 7 main independent variables, almost every other dimension is in some way derived from those independent variables.

The 7 main independent variables are; *hand length, middle finger length, middle finger thickness, hand width, wrist width, thumb thickness, thumb length*. Two other independent variables are the living hinge dimensions (thickness and length) which currently have a fixed value in the model. The rail interfaces of the final design are not included in the scalable proof of concept model. To read more on the independent variables that drive this 3D CAD model see Appendix D.1.

The 7 independent variables can be changed by the user through the Graphical User Interface (GUI) in order to provide a hand that matches the child's other, natural, hand perfectly. The GUI is described in section 3.2 of this report.

3.1.2. Production

The prosthetic FR Hand is designed to be 3D printed on almost any FDM (fused deposition modeling) desktop 3D printer. The design doesn’t require dual extruders or any modifications to the machine. The prototypes for the final design were printed on a Tronxy X8 low-cost 3D printer.

The printing material used was an unbranded spool of flexible thermoplastic elastomer filament (typically called TPE in the 3D printing world). A common name-brand of flexible filament is Ninjaflex which seems more flexible than the filament that was used for the final prototype. Both filaments can be used for this design. Flexible filaments are typically printed more reliably on 3D printers with a direct extruder (as opposed to 3D printers with a Bowden extruder). The palm is printed in PLA rigid plastic, though other rigid plastics (such as ABS (Acrylonitrile butadiene styrene)) should also be sufficient.

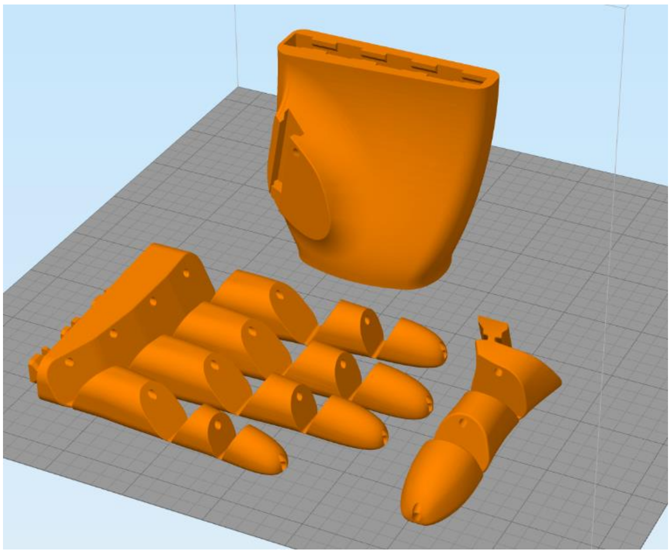


Figure 19: 3D printing orientation of the finger set, thumb, and palm

The orientation of the 3D print is important because it greatly affects the mechanical properties of a print when using an FDM 3D printer. These mechanical properties vary due to the non-homogeneous distribution of the material and varying tensile strength when loaded in certain directions due to the adhesion between layers. For this reason, the parts printed in the flexible material were oriented so the hinge plane is parallel with the print bed. This provides great strength when pulling or bending the finger, while there are almost no scenarios in which the adhesion between layers is pulled apart. The palm is printed in a different, vertical, orientation. This is possible because the palm is not critically loaded and has a different geometry. Figure 19 shows the printing orientations of the different 3D printed parts.

Another benefit of the chosen orientations is that it allows the parts to be printed with minimal support material. The required print orientations were kept in mind from the beginning of the design process and had a big influence on the end result.

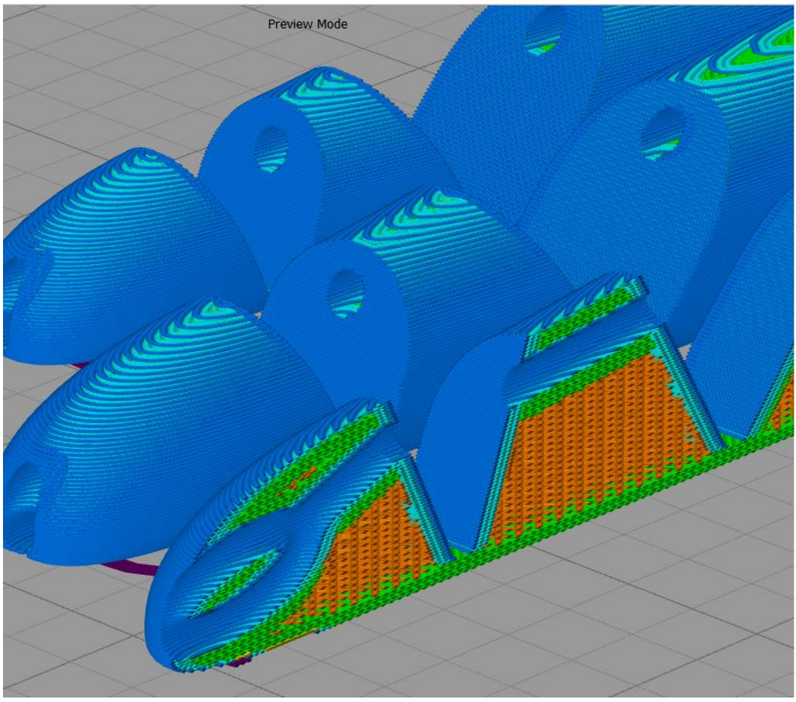


Figure 18: Cross section view of 3D printer toolpath preview. Orange: infill, green: solid layer, blue: outer perimeter, cyan: inner perimeter. Note that the hinges are printed completely solid (green) and that the “phalanges” are printed with a 40% infill (orange).

An infill of 40% was chosen for the flexible fingers and thumb. During earlier prototyping with flexible materials, 3D prints with higher infills of 90% and 100% were tested. These higher infills were more rigid but this is not necessarily desirable for the use in fingers. Higher percentages of infill increase weight, printing time and material costs, so a lower infill percentage was chosen. Because the hinges are only 1mm thick, the hinges themselves are printed completely solid which can be achieved by correctly configuring the printing settings. A layer height of 0.16mm was used, the three bottom and six top layers were printed solidly, which causes any hinge under 1.44mm thick to be printed solid. Figure 18 shows a preview of the toolpath used to 3D print the *finger set*. In total it took roughly 13½ hours to print all the parts that make up the FR hand.

3.2. The Graphical User Interface

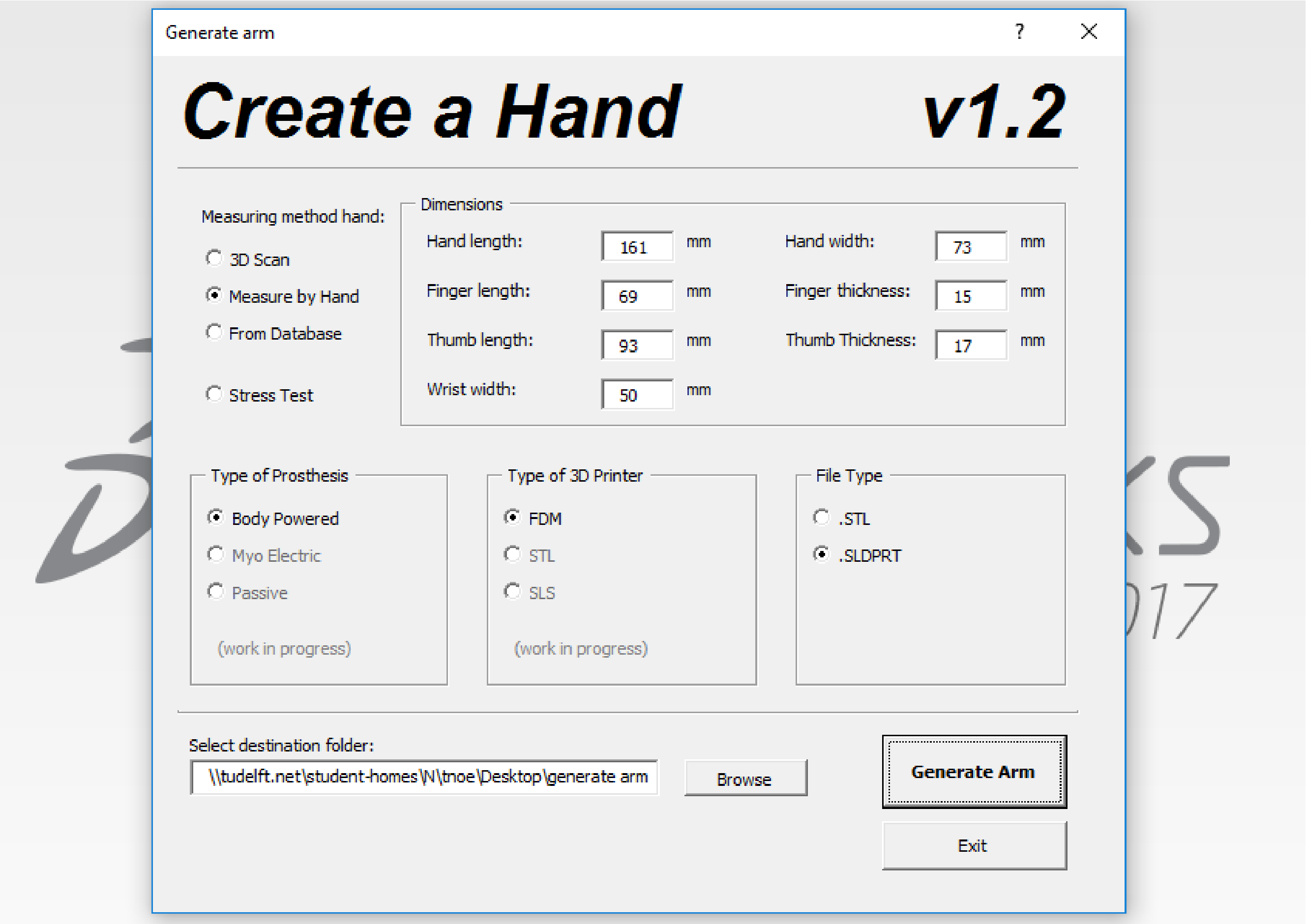
An essential part of the design is the graphical user interface (GUI) which allows the user to generate a perfectly sized hand. This way the user or their family doesn't need any 3D modeling experience or even need to see the 3D CAD model. By filling in a web form with the required measurements, a perfectly sized hand could be generated and sent to a 3D printing service. The hand model is not uniformly scaled, instead 7 independent variable drive the sizing of the 3D hand model (see Appendix D.1. for more details on these independent variables). This non-uniform scaling is important because the hand proportions change as the child ages, and because the goal is to have a hand that closely matches the user's natural hand.

Because the hand generating software is such an essential part of this design, it was decided a proof of concept should be made. This is done to show that a fully scalable 3D hand model can be made and that such a model can be reconfigured by an inexperienced user.

Solidworks was chosen to create the 3D hand model, Visual Basic for Applications (VBA) was used to create the GUI and to control Solidworks, and Excel was used to process data from the DINED database (DINED, n.d.). Other software combinations such as Rhinoceros + Grasshopper were considered but Solidworks was chosen because of the author's familiarity with the software as well as its Application Programming Interface (API) which allows it to be easily programmed using VBA. A real-world product might use a different combination of software and coding languages as the current proof of concept requires a full Solidworks software installation on the user's computer.

For this proof of concept, a simplified 3D hand model was used. Instead of a hand consisting of three pieces (*finger set*, *palm*, *thumb*), a single-part hand was used. This means all the hand geometry and hinges are testable, without having to make the rail interface systems scalable. A 3-part prosthetic hand could definitely be made to be scalable, but this would require significant additional work, which wasn't possible due to time constraints.

Figure 20 shows the GUI. The user can choose different measurement options to generate the required dimensions for the hand.



Measuring method hand:
○ 3D Scan
● Measure by Hand
○ From Database
○ Stress Test

Dimensions
Hand length: 161 mm
Finger length: 69 mm
Thumb length: 93 mm
Wrist width: 50 mm
Hand width: 73 mm
Finger thickness: 15 mm
Thumb Thickness: 17 mm

Type of Prosthesis
● Body Powered
○ Myo Electric
○ Passive
(work in progress)

Type of 3D Printer
● FDM
○ STL
○ SLS
(work in progress)

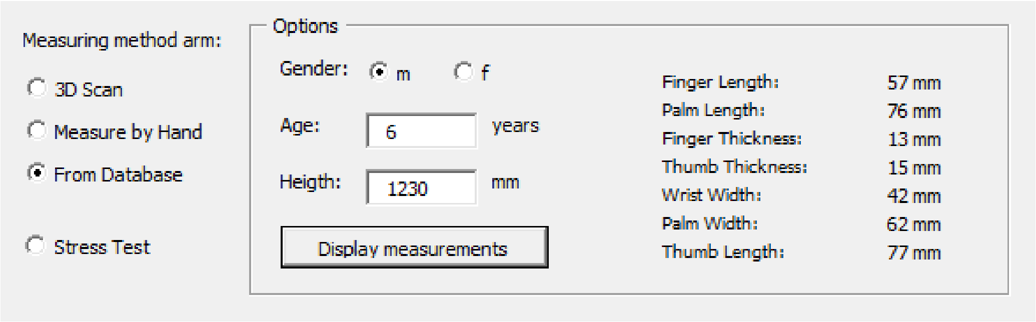
File Type
○ .STL
● .SLDPRT

Select destination folder: \\tudelft.net\student-homes\N\tnoe\Desktop\generate arm

Figure 20: The Graphical User Interface (GUI) that allows a user to generate a custom sized prosthetic hand

The first option ("*3D scan*") is to 3D scan the healthy hand in order to extract the relevant measurements. This option is not functional in this proof of concept as this is not the focus of my thesis (and could be enough work for an entirely separate thesis).

The *"measure by hand"* option allows the user to input 7 measurements. When the user presses the *Generate Hand* button, in the bottom right, these 7 measurements are used to generate a hand with the desired dimensions. This option is selected in figure 20 on the previous page.



Measuring method arm:

○ 3D Scan
○ Measure by Hand
● From Database
○ Stress Test

Options

Gender: ● m ○ f
Age: 6 years
Heigth: 1230 mm

Finger Length: 57 mm
Palm Length: 76 mm
Finger Thickness: 13 mm
Thumb Thickness: 15 mm
Wrist Width: 42 mm
Palm Width: 62 mm
Thumb Length: 77 mm

Figure 21: Available options in the GUI when the "from database" input method is selected

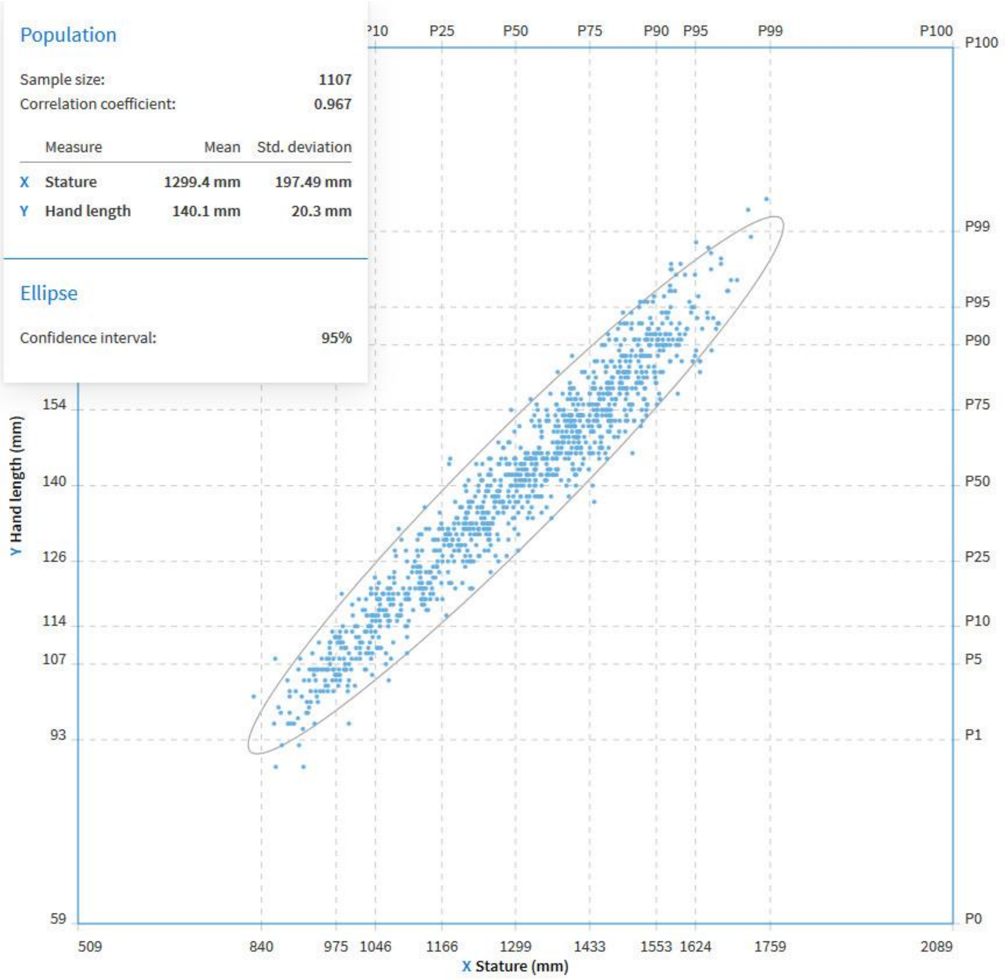
There is also an option to generate the required measurements from an anthropometric database. When the *"from database"* option is selected the user simply fills in the gender, age, and stature. Figure 21 shows the available options in the GUI. These inputs are then used to approximate the 7 required hand measurements using the DINED database. This is possible because most of the hand measurements are strongly correlated to the stature. For example, if you are relatively tall for your age, your hands are likely also longer.

Figure 22 shows the correlation analysis of *stature* and *hand length* in children from 2 to 12 years old. As can be seen, these measurements are strongly correlated, as indicated by a correlation coefficient of R=0.967 (Ellipse, n.d.).

One measurement that isn't strongly correlated to stature is the hand thickness (R=0.567). To see the correlation coefficients for each variable compared to stature, see Appendix D.2., Table 8.

One should also keep in mind that not every child follows standard correlations and certainly not for multiple different dimensions at the same time, to quote DINED, "the average person does not exist" (DINED(2), n.d.). For an accurately matching prosthetic hand, manual measurements are preferred. However, database derived measurements can be useful for users who miss both hands, or for users who don't want to manually measure. Additionally, having a way to get approximate measurements also helped immensely with the development and testing of the 3D hand model and GUI.

Finally, the GUI also has an option *("stress test")* to test the scalability and robustness of the 3D hand model. This option is discussed in detail in the *validation* section later in this report.



Population

Sample size: 1107
Correlation coefficient: 0.967

	Measure	Mean	Std. deviation
X	Stature	1299.4 mm	197.49 mm
Y	Hand length	140.1 mm	20.3 mm

Ellipse

Confidence interval: 95%

Figure 22: Correlation analysis of stature and hand length (Ellipse, n.d.)

For this proof of concept, the options for the generated hand itself are limited. The user can select the file type (.STL or .SLDPRT) and save location. If the exported arm already exists in the destination folder the program adds a version number to the name of the file (e.g. arm_export(2).STL).

Figure 23 shows the end result, the automatically generated prosthetic hand 3D CAD model. This particular hand was generated using the “from database” input method, optioned for an 11-year-old male user who is 1500 mm tall. A lineup of hands dimensioned for users between the ages of 2 and 12 years old is shown in figure 24.

To see the GUI in detail, all the code behind it, and the additional modules that are used see Appendix D.2.

To play around with the GUI on a computer and access other digital documents related to this thesis, see the digital resources pack in Appendix F.

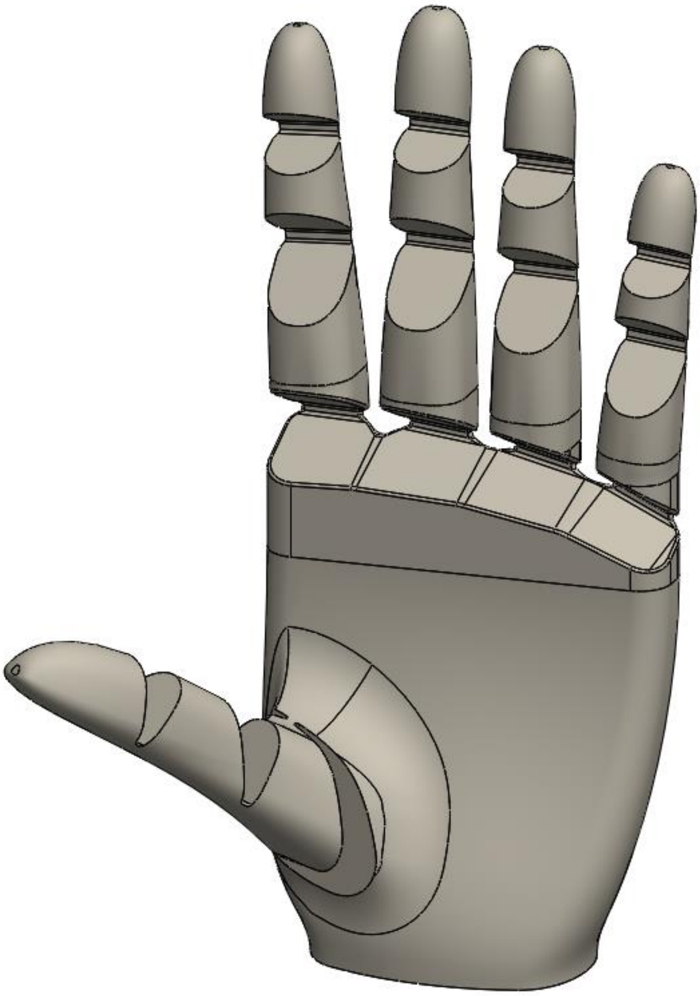


Figure 23: Automatically generated hand. Dimensioned for an 11-year-old, 1500 mm tall, male user. As can be seen the hand is a single part and missing the actuation wire channels.

Figure 24: A series of automatically generated hands. One for each year between the ages of 2 to 12 years old

4 Validation

The final design of the prosthetic hand has to be tested in order to find out if it fulfills the proposed requirements.

4.1. Fatigue

One uncertain part of the design is the reliability of the flexible living hinges that allow the fingers to bend. While such living hinges are not necessarily new in injection molded products, the FR Hand is supposed to be 3D printed. The material in a 3D printed part isn't homogeneously distributed but is instead built up from layers. The adhesion between layers can result in varying mechanical properties. For instance, a part's strength varies based on the printing orientation and the direction in which a part is loaded. This raises questions about the long-term durability of the 3D printed living hinges. Add to this the unknown composition of the unbranded "TPE" (thermoplastic elastomer) printing filament. For these reasons it was decided a durability test of the *finger set* should be conducted.

In order to determine the durability of the hinges a test bench was designed, to put the fingers through a large number of grasp cycles (see figure 25). Besides validating the reliability of the design this test was also used to help determine the final hinge thickness. Ideally, the hinges should be as thin as possible, without being too thin causing the fingers to break prematurely. For this reason, a set of 4 equally long fingers with varying hinge thicknesses and hinge lengths were made. Hinge thicknesses of 0.5mm, 1mm, 1.5mm, and 2mm were chosen. The proximal hinges had a hinge thickness to length ratio of 1:1, meaning the 1mm thick hinge was 1mm long. The distal hinges had a thickness to length ratio of 1:1.5, meaning the 1mm thick hinge had a hinge length of 1.5mm.

The test bench was made using old 3D printing parts that were available. For the control board, a Minitronics v1.0 was used, which is an Arduino

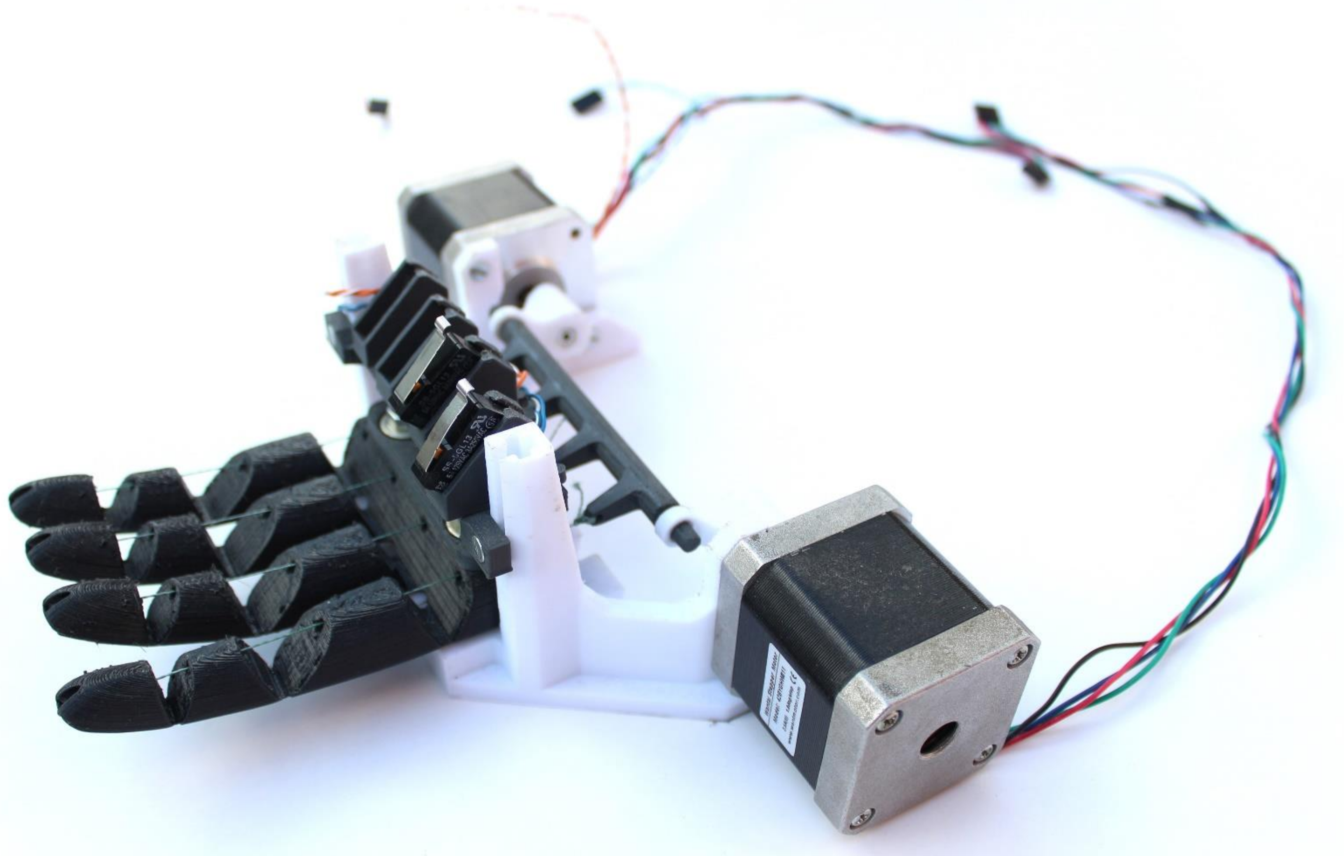


Figure 25: The testbench used to fatigue test the living hinges. Note that only two of the four microswitches are installed in this picture. The stepper motors and microswitches are also not plugged into the Minitronics control board in this photo.

based board (using an ATmega 1281 processor) with integrated stepper drivers designed to control FDM 3D printers (RepRap.org, 2016). Two NEMA 17 stepper motors, connected to a crank, were used to actuate the fingers through braided fishing lines. Four microswitches commonly used as end stops for 3D printers were used to keep track of how many cycles each individual finger completed. The crank and the frame holding the various components together were printed in PLA plastic.

A program was written using the Arduino IDE (Integrated Developer Environment) to control the motors and keep track of the completed number of cycles of each finger. By using four individual microswitches the test bench didn't need to be closely supervised. If a finger fails, the program keeps running, but because the microswitch for the broken finger is no longer triggered, the cycle count for the broken finger no longer increases. Besides the microswitches, the program also uses the number of steps of the stepper motors to keep count of the total numbers of cycles, in case all microswitches fail, without the fingers themselves failing. More detailed information about the test bench setup can be found in Appendix E.1.

The FR Hand is intended to have a shorter life cycle than conventional prostheses. The hand only needs to be used for a maximum of a few months instead of the average 18-month lifespan of an upper limb prosthesis. How many cycles should the final design be able to withstand to prove its reliability? A study by Limehouse et al. (2005) claims a prosthetic hand will undergo 1200 grasping motions per day. Another study (Vinet et al., 1995) claimed a prosthetic should be able to perform 300,000 grasping cycles in its lifetime without failure. To validate the 3D printed living hinges design, a target of 300,000 cycles was chosen. This is likely more than required for this hand but will hopefully demonstrate the reliability with regards to fatigue.

Table 1: Fatigue testing results

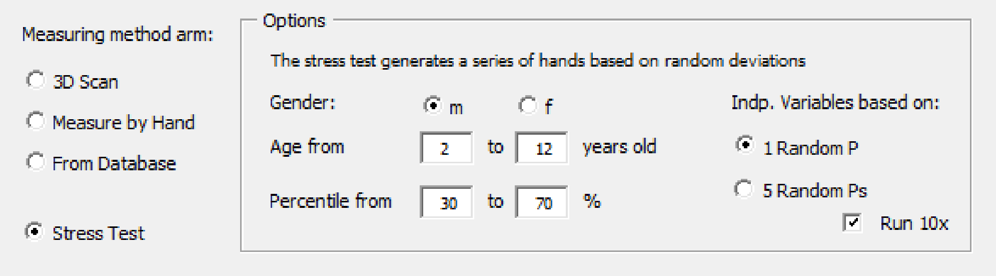
Number of Sessions	*Runtime (hh:mm)*	*Number of Cycles*	*Average Speed (cycles/min)*
19	52:48	298193	94

The four fingers completed a total of 298 thousand cycles on the test bench without any of the hinges or strings breaking, after which the testbench was shut down. The testing occurred over 19 sessions totaling roughly 53 hours of runtime. For more detailed data on the fatigue testing see Appendix E.1. Based on these results fatigue doesn't seem to be an issue for this material (TPE) with these hinge thicknesses (0.5mm – 2mm). The final prosthetic hand design uses 1mm thick and 1.25mm long hinges.

This testing doesn't definitively prove the 3D printed hand's reliability. For that, real-world testing should be performed. The hand may be able to withstand 300 thousand cycles when the hand is loaded in the correct direction, but what happens when a finger is loaded from the side? Or how about the effects of ultraviolet (UV) light and moisture? The durability results are promising so far but more practical testing should be performed to ensure the FR Hand's reliability.

4.2. Scalability 3D model

Another potential issue is the scalability of the 3D CAD/CAM (computer aided design/computer aided manufacturing) model. The Solidworks 3D model was designed from the ground up to be scalable for a wide range of ages and unique proportions. But how robust did the model turn out, and how to test this robustness?



Measuring method arm:
○ 3D Scan
○ Measure by Hand
○ From Database
● Stress Test

Options
The stress test generates a series of hands based on random deviations
Gender: ● m ○ f
Age from 2 to 12 years old
Percentile from 30 to 70 %
Indp. Variables based on:
● 1 Random P
○ 5 Random Ps
☑ Run 10x

Figure 26: Available options when the "stress test" input method is selected

A prosthetic hand model could already be generated using an anthropometric database (DINED, n.d.) through the previously developed Graphical User Interface (GUI). Additional code was written in VBA (Visual Basic for Applications) to perform a stress test of sorts. See figure 26 for the available options in the GUI related to the stress test.

The goal of the stress test is to pick a random deviation (by generating a random percentile value), to then apply a random deviation to each age within a certain age range, and to then generate a series of 3D hand models. For example, if the user specifies an age range of 2-12 years, a percentile range of P49-P51, and a specific gender, the program will generate 11 unique prosthetic hands that are pretty much average for their age. The robustness is tested by slowly widening the percentile range, running 10 of these batches each time (11*10=110 3D hands), and documenting the number of hands that contain errors.

The program can reliably generate hand models between P20 and P80 for the ages 4 to 12 years old, for both males and females. Hands for 2- and 3-year-olds are more sensitive and can't be generated completely reliably. Only a single hand outside this age range failed, a hand for a 6-year-old within the P10-P90 range. Most hands that failed were due to an error of the fillet between the thumb and the palm. The results of the stress test for hands generated using 1 randomized percentile value can be found in Table 2.

The fact that the hands for children between 4 and 12 years old can be reliably generated is a satisfactory result but assumes the hand stays somewhat in proportion. For example, a P85 hand would result in a hand with a palm that is longer than average, but the same percentile value is also used to generate the finger length so the fingers would also be longer, thereby keeping similar proportions. For this reason, an option was added to vary each independent variable individually, which could cause one dimension to be smaller than average while another could be significantly larger than average.

Because the anthropometric database only has 5 of the 7 required independent variables available only 5 randomly generated percentile values are used. The two remaining "independent" variables in the 3D model are derived from the available truly independent variables.

The results of the stress tests when using 5 randomly generated percentile values are shown in Table 3.

When using 5 random percentile values the program seems to be similarly reliable. It's hard to say for certain how reliable the program is due to the possible randomness combined with the small sample size. 110 test hands simply aren't enough to accurately judge the programs ability to reliably generate hands.

Table 2: Results hand generating stress test using various percentile ranges, (1 random P)

1P	*# Faulty hands*	*Out of # hands*	*Faulty hands*	*What's wrong?*
P49-P51	0	110		
P40-P60	0	110		
P30-P70	1	110	3_yo(1)	Thumb-palm fillet
P20-P80	7	110	2_yo(6,8), 3_yo(1,2,5,7,8)	Thumb-palm fillet
P10-P90	9	110	2_yo(12,16,19), 3_yo(10,11,12,13,19), 6_yo(11)	Thumb-palm fillet, *knucklecut* fillet in 6_yo
P01-P99	13	110	2_yo(0,1,3,5,6,7,8), 3_yo(0,1,2,3,4,8)	Thumb-palm fillet

Table 3: Results hand generating stress test using various percentile ranges, (5 random Ps)

5P	*# Faulty hands*	*Out of # hands*	*Faulty hands*	*What's wrong?*
P49-P51	14	110	2_yo(0,1,4), 3_yo(9), 6_yo(7), 11_yo(3,4,6,9), 12_yo(0,1,3,6,8)	palm-thumb fillet in all, also *knucklecut* fillet in 12_yo(3,6,8,0)
P40-P60	2	110	12_yo(3), 10_yo(5)	palm-thumb fillet, palm-thumb fillet and *knucklecut* fillet
P30-P70	1	110	12_yo(3)	palm-thumb fillet
P20-P80	1	110	10_yo(7)	palm-thumb fillet
P10-P90	0	110		
P01-P99	0	110		

More errors started to occur when widening the percentile range when using a single random percentile. This is what you would expect, when the hands are generated using more abnormal values, the number of errors increases. Interestingly, the opposite seemed to be true when using 5 random percentiles. The program became more reliable when choosing variables within a wider percentile range.

Another interesting thing is that most errors occurred in the smaller hands (2-3 years old) when using a single random percentile to generate the required variables. However, when using 5 random percentiles, most errors occurred in the larger hands (10-12 years old).

The causes of these differences are not fully understood.

The reliability of the program doesn't seem to solely depend on the percentile range, some of the errors seem to occur randomly. Overall, the program seems to be reliable for about ~96% of the generated hands (48 total failures out of 1320 generated hands), though again, it's hard to say due to the small sample size.

Of the generated hands with errors, 97.9% of the failures contained an error due to the fillet between the palm and the thumb (see A in figure 27). Roughly 12.5% of the hands with errors contained an error due to the missing *knucklecut* fillet (see B figure 27). 10.4% of the failed hands contained both of these errors. Figure 27 shows two hands that were generated as part of the stress test, one without errors, the other missing both the palm-thumb fillet and the *knucklecut* fillet. This part of the design could be further developed to create a completely reliable hand model. Time constraints prevented a redesign, and re-testing, of the hand model.

It should be noted that the age in the stress test was only varied between 2-12 years because there is no data within the DINED database for children between 12 and 18-years old. The 3D hand model should be scalable for adults as well, though this is not a requirement for this thesis.

Overall, the results of the scalability testing are somewhat inconclusive. The results show that a fully scalable 3D hand model is possible. However, the current CAD model and testing process don't definitively prove that. More work needs to be performed before this could be considered consumer ready.

For a more detailed description of the scalability testing and results see Appendix E.3.

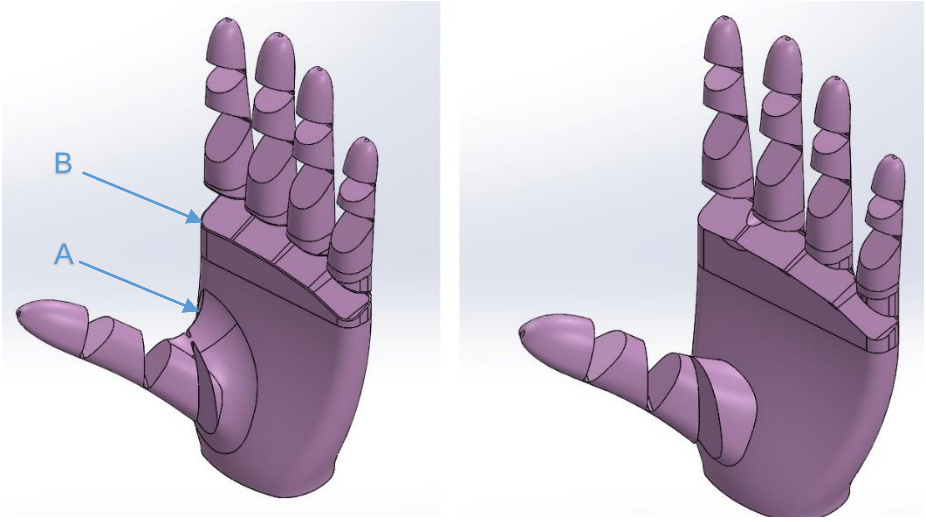


Figure 27: Left, an automatically generated hand without errors. Right, a automatically generated hand missing both palm-thumb(A) and knucklecut fillets(B).

4.3. Function

The scalability and minimal assembly construction are the most distinctive aspects of the FR Hand, compared to more conventional prosthetic hands. But any potential prosthetic hand should also function adequately. For this thesis, the functionality testing is very limited as the focus was directed at other aspects of the design. Some basic measurements of the final design were made. No real-world user testing was conducted. See the discussion section of this report for more discussion on the limited amount of practical testing.

The prototype hand weighs 99 grams including everything except the mounting bolt. The prototype was sized for a 12-year-old male user.

The force required to close the hand was roughly measured using a mass. The prototype was placed vertically, with the actuation wire hanging straight down. At the end of the wire, an empty plastic bottle was placed. The bottle was filled until the hand fully closed (couldn't flex any further). The bottle with the required amount of water was then placed on a scale. A total weight of 1440 grams was required to fully close the prototype hand, this corresponds to roughly 14.1 Newtons of force.

The tensile load at which a 1mm thick hinge breaks was also tested. This load can first be calculated based on the hinge dimensions. The theoretical load at which the hinge will break is 225 Newton (see Table 4). This is based on the cross section of the hinge calculated in Solidworks and the tensile strength of NinjaFlex filament (NinjaTek, 2016). Technical data on the generic TPE filament that was used to print the prototypes is not available, so the tensile strength for NinjaFlex is used as it is a close analogue.

Table 4: Theoretical breaking strength of the fingertip hinge

Cross section fingertip hinge	*Tensile strength NinjaFlex*	***Breaking strength***
8.64 mm^2	26 MPa	**224.64 N**

The reason to test this practically was due to the uncertainty of the 3D printing process and 3D printing material. A spare *finger set* was mounted with the fingers pointing straight down. Initially, a 10-liter plastic container was tied to the fingertip using a string tied to the existing tiedown point in the tip of the finger. This plastic container was then filled with water until the hinge failed. The plastic container was then to be weighed using a scale.

The string attachment point within the fingertip failed at 5.250kg (52N). This tiedown point was only designed to handle the actuation force for each finger. Locking pliers were subsequently used to attach the test mass to the fingertip. Because the 1mm thick hinge did not fail when the 10L container was completely filled, the test mass was switched to a backpack filled with weights. The load was then increased in roughly 500 gram increments. When the hinge failed the total test mass was weighed. Figure 28 illustrates the test setup.

Table 5: Practical measurement of the breaking strength of the fingertip hinge

Ultimate weight of test mass	*Approximate breaking force*
16011 g	157 N

The hinge failed when loaded with a weight of 16kg. The maximum tensile load on a 3D printed 1mm thick flexible hinge is roughly 157 Newton (See Table 5). In practice, this means the user can grab a prosthetic finger, pull the finger straight out as hard as they can with the healthy hand, and the living hinges in the finger will not break.

These measurements are intended as estimates in order to get a better idea of some of the hand's properties. The pinch force was not measured relative to the actuation force. Playing with the prototype makes it clear that the hand is very robust, but exerts very little pinch force.

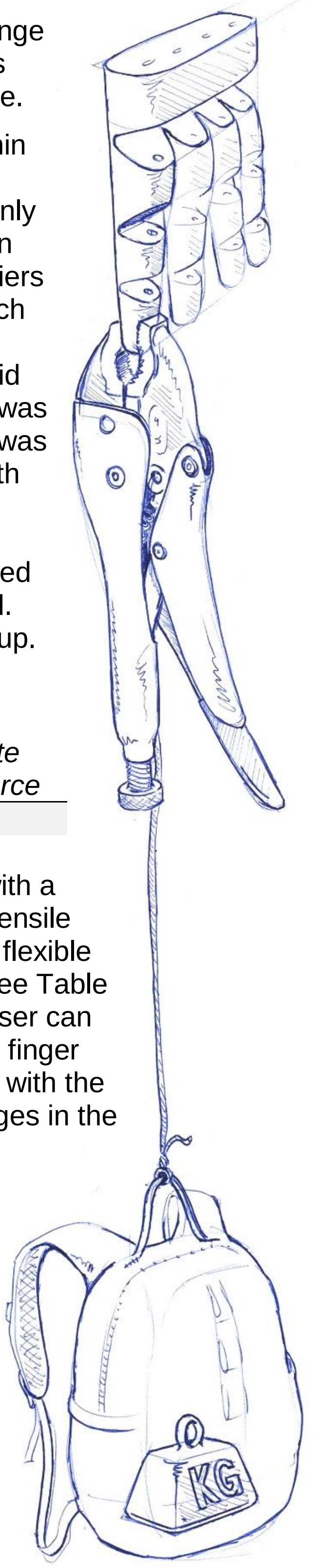


Figure 28: The testing setup to measure approximate breaking strength of the 3D printed living hinges

4.4. Cost

Finally, there is the cost factor. The frequent replacement model hinges on the affordability of the prosthetic hands. The hand can fulfill all technical requirements but still end up not being used if the hands are too expensive. The final design doesn't use any screws or additional fastening hardware. The only non-3D printed parts are the braided lines.

Table 6 shows the cost of the materials for a single prosthetic hand.

These costs assume one already has and can operate a 3D printer which will not be the case for most users. For this reason, a quick inquiry was made with three local 3D printing services through 3dhubs.com in June 2018 (3D Hubs, n.d.). The following 3D printing services were selected; MTB3D located in Delft, G-ICT-3D in Zoetermeer, and Type-R in Rotterdam. Table 7 shows the costs for the 3D printed parts (material costs are included) of this design through a commercial printing service.

Table 6: Material costs and 3D printing time for a single prosthetic hand

Material	***€/m***			***Length used (m)***	***Total cost(€)***
Braided fishing line	0.07			~2	0.14
	€/kg	***Part***	***Printing time (h)***	***Mass used (kg)***	***Total cost (€)***
TPE flexible filament	65	Finger set	05:10	0.052	3.38
		Thumb	02:50	0.014	0.91
PLA rigid filament	20	Palm	05:07	0.059	1.18
		Whippletree	00:15	0.002	0.04
Total material cost			**13:22 (hh:mm)**	**0.127g**	**€5.65**

Additionally, there might also be some labor costs. The 3D printed parts still need to be assembled and rigged with the wires that allow the fingers to be actuated. This can be done by the user's family themselves, in which case the assembly is free, or could be provided through a service.

Table 7: The cost of 3D printed parts for a single prosthetic hand through commercial 3D printing services (June, 2018)

	Material	*Cost (€)*, Type-R (Rotterdam)	*Cost (€)*, G-ICT-3D (zoetermeer)	*Cost (€)*, MTB3D (Delft)
Finger set + Thumb	TPU (thermoplastic polyurethane)	14.53	23.81	30.18
Palm + whippletree	PLA (polylactic acid)	10.55	6.93	17.31
Total printing cost (€)		**25.08**	**30.74**	**47.49**

The rigging of the wires is still a quite fiddly part of the current design, even though it's easier than existing open-source 3D printable prosthetic hands. If we budget half an hour for an experienced person (at €15/h) to assemble and rig the hand we end up with €7.50 in labor costs.

The cost calculations don't include a wrist unit to mount the prosthetic hand to the socket. This cost is not included because a conventional prosthetic device would also need a wrist unit and the wrist unit does not have to be replaced each time a hand is swapped.

In conclusion, this design seems very feasible from a cost standpoint. The family of the child could expect a price under €100 per hand. If, for example, the hand is replaced every 3 months this would result in a cost of €400 per year or €600 over a period of 18 months.

5 Conclusions

After the work for this project has been completed it is time to reflect on the results and see if the design goal has been met, and what could be improved upon.

5.1. Discussion

The concept fulfills the majority of set requirements. However, there are still some concerns with the design and thesis work. This section of the report discusses shortcomings of the current project and opportunities for future work.

5.1.1. The overall concept

One of the main elements of the frequent replacement concept is the reduced reliance on the time of a professional. This allows for a cheaper and easier replacement process, as a new hand can be ordered from home. However, this reduced interaction with a professional could also have adverse effects. Some claim professional training and regular monitoring are an essential part of successful prosthetic treatment (Egermann et al., 2008). A professional could also be beneficial for other elements besides the prosthesis itself, such as psychological factors, or education of the family. On the other hand, the process of constantly choosing new prostheses also increases the child and family's involvement with the prosthetic process. Which might lead them to be able to try multiple different options and become more educated overall. When deploying the frequent replacement concept in the wild, attention should be paid to include and emphasize the importance of professional care at specific intervals. This way, one can be assured that the concept has an overall positive effect on the child's wellbeing and prosthetic success.

The work in this thesis only focused on the hand of the prosthesis. But the other elements such as the forearm, socket, and harness are also extremely important. Without proper solutions for some of these elements, this concept is dead in the water. In a way, this thesis only details parts of the frequent replacement concept.

5.1.2. The physical hand

One of the biggest shortcomings of the design is its functionality, or rather the lack thereof. As discussed in the introduction, function is one of the most important things in a prosthesis. If a tool doesn't allow the user to do anything they wouldn't be able to do otherwise, they are not likely to use it.

The main shortcoming in terms of function are the high forces involved with actuating the hand. The hand requires a significant amount of force to simply close, and the hand can barely exert any pinch force on an object. Parts of these problems are due to the actuation method of strings. The strings simply don't have much of a lever in relation to the hinges to act on the fingers. Additionally, there is also friction between the strings and the channels they run through. But besides the actuation method, living hinges also inherently require more force to bend than more conventional hinges.

The actuation mechanism using strings can also be fiddly to rig properly. The strings need to be tied to the whippletree linkages at a precise length of the string, or the fingers won't flex simultaneously. Rigging also requires some knowledge of knots which the average user might not have. For the context of this thesis the rigging was not a problem, it's certainly an easier process than for other 3D printed prosthetic hands found on the internet. But for a final consumer product, the current string actuation is not satisfactory. Improvements could be made in the way the actuation strings attach to different parts of the hand, for instance, by not relying on tied knots. Or the actuation mechanism could be changed entirely, e.g. by using pushrods, linkages, pneumatics or hydraulics instead of strings.

Another problem with the current design is the precision of the grasp. The hand was designed to have a “three jaw chuck” grasp. In practice, the grasp of the prototype isn't very precise and can't really be used to pick up small objects. This could be fixed in a future iteration as there isn't anything that prevents this design from having a precise grasp. It would require finetuning the orientation of the hinges and varying the hinge thicknesses to control the sequence in which the hinges actuate (proximal to distal).

These factors combine in a hand that, currently, can only hold, light, easily graspable objects and these objects will likely be passed from the healthy hand to the prosthetic hand by the user.

The FR Hand excels in terms of weight, robustness, and costs. Even when compared to commercially available prosthetic hands. The ease of assembly, excluding the rigging of the actuation strings, also turned out very well. The hand can be reliably printed on currently available, affordable 3D printers and the hand can be assembled in under a minute from the printer to a complete hand. All without requiring any screws or tools.

Besides the shortcoming of the design itself, there are also shortcomings in the testing of the functionality of the prosthesis. For this thesis, some uncertain elements of the design were tested, but the real-world functionality was barely tested in practice. The actuation force was not accurately tested in relation to the pinch force or displacement. The hand was not tested in practice by a healthy subject using a prosthesis simulator. The hand wasn't tested by the target users. No target users were interviewed during the design process. The lack of practical testing places serious question marks about the real-world usefulness of this project for the target audience. The main reason for the lack of testing were time constraints, this project overall is still in a somewhat early and conceptual stage.

5.1.3. The software

A key part of this project was the design of a scalable hand 3D CAD model, and the user interface to accompany it. The robustness of the scalability was tested in order to validate its usefulness for the target demographic.

One obvious shortcoming is the fact that the fully scalable hand is a simplified compared to the prototype used for physical testing. While it does have the hinges and overall hand geometry, it's made in a single piece. The real-world hand is printed in three pieces which is essential for its production and function. The decision to use a simplified version of the hand was made so the rail interface between the three parts didn't have to be made fully scalable. This would've required significant additional effort. Of course, that means the results of the scalability testing are not directly applicable. With more time a rail system should be able to be made scalable, but as of yet this is unproven.

A smaller concern with the current GUI (graphical user interface) is that it currently runs locally on the user's computer within Solidworks. In the future the program would run on a server, allowing a user to interact with it through a web form. That way the user doesn't need to install any software on their own computer, or even own a computer for that matter.

The testing of the scalability of the 3D hand model was not completely conclusive. While the testing showed that most users (4-12 years old) would likely be able to generate a perfectly sized hand, 100% reliability could not be guaranteed. The reliability seemed to vary pretty unpredictably, sometimes seemingly influenced by unexpected factors such as what computer was used. Overall, the scalability of the hand model could not be proven to be completely reliable due to these uncertain factors and small sample sizes. With more work (redesign of the fillet between the thumb and the palm) the scalability could likely be made to be 100% reliable.

It should be noted that the robustness was only tested for ages between 2-12 years old due to limited data in the DINED database. Though this is unlikely to pose a serious problem in the development of a fully scalable hand.

Like with the physical prototypes, no user testing was conducted for the GUI or the scalability. A future study could significantly add to this project by conducting user tests.

5.1.4. Opportunities

The thesis focused on specific aspects of a prosthetic arm system, what are some areas a future study could look into?

First of all, there is the rest of a prosthetic arm system. This thesis focused exclusively on the hand itself, but the forearm and socket are just as important, especially for children. For the concept in this thesis to work a 3D scanning solution for the socket is necessary. It doesn't matter if the hand is the correct size if the socket doesn't fit well anymore or the forearm has become too short. Traditional elements of a prosthetic arm, such as the harness, also have room for improvements of course.

Besides using a 3D scan to generate a properly fitting socket (which would be a huge challenge), one could also develop a simplified scanning process which allows a user to accurately measure certain dimensions of a hand. The FR

Hand system relies on accurate measurements from the user or their family, which might not be realistic to expect. A simple scan which returns a set of key measurements shouldn’t be too difficult to implement and could significantly increase the practicality of this concept.

Another area for improvement previously touched upon is the actuation mechanism. The string-based actuation mechanism has significant downsides both in terms of function and ease of assembly. A myoelectric version of this concept could also be considered for those who prefer myoelectric actuation. This could result in an aesthetically pleasing prosthetic arm that doesn’t require the user to wear a harness.

Finally, a future study could also look into more advanced 3D printing technologies and materials. While some experimentation was done, the work in this thesis was based on the strengths and particular weaknesses of desktop FDM printers. Using, for instance, a resin-based 3D printer would allow for an even more natural looking and complex prosthesis which wouldn’t require any assembly (printed in one piece). Of course, resin-based 3D printing systems have their own disadvantages, such as high costs and the deterioration of flexible materials over time. It would be interesting to see the results of this concept when using a different 3D printing process as the technologies continue to become cheaper.

5.2. Conclusion

As stated in the introduction the goal was to develop an aesthetically pleasing prosthesis for growing children missing an upper limb. With an underlying focus not just on function, but also on how it allows the user to fit in. The design process that followed resulted in a 3D printed prosthetic hand that is always perfectly scaled to the user as he or she grows. The prosthetic hand can be frequently replaced due to how easily the hand can be generated, manufactured, assembled, and used.

The hand fulfills the majority of the set requirements. The frequent replacement concept results in a correctly sized hand, which is cheap, durable, and lightweight. The user ends up with a hand which is more appealing than most currently available 3D printed prosthetic hands and is always up to date with their ever-changing demands.

Currently, the biggest uncertainty is its real-world performance and usefulness, as more practical testing needs to be conducted. The FR Hand could also still be made to look more natural. As it stands, further development efforts will have to be made before this concept is relevant for real-world children missing an upper limb.

Appendix

Contents

A. Bibliography

1. References

123-3D. (n.d.). *Filament flexibel zwart 1,75 mm TPE 0,5 kg (123-3D huismerk)*. Retrieved on October 30 2018 from https://www.123-3d.nl/123-3D-Filament-flexibel-zwart-1-75-mm-TPE-0-5-kg-123-3D-huismerk-i1541-t224.html

3D Hubs. (n.d.). *Online manufacturing*. Retrieved in June 2018 from https://www.3dhubs.com/3dprint

Biddiss, E. A., & Chau, T. T. (2007). Upper limb prosthesis use and abandonment: A survey of the last 25 years. *Prosthetics and Orthotics International, 31*(3), 236-257. doi:10.1080/03093640600994581

Crandall, R. C., & Tomhave, W. (2002). Pediatric Unilateral Below-Elbow Amputees: Retrospective Analysis of 34 Patients Given Multiple Prosthetic Options. *Journal of Pediatric Orthopaedics, 22*(3), 380-383. doi:10.1097/01241398-200205000-00023

CZ Groep. (2018). *Reglement Hulpmiddelen*, p. 9. Retrieved on October 30 2018 from https://www.cz.nl/-/media/actueel/voorwaarden/reglement-hulpmiddelen.pdf

Decathlon. (n.d.). *Gevlochten vislijn Braid 4 X rookgroen 130m, 0.3mm*. Retrieved on October 30 2018 from https://www.decathlon.nl/p/gevlochten-vislijn-braid-4-x-rookgroen-130-m/_/R-p-149180

DINED. (n.d.). *1D Anthropometric database*. Retrieved on October 30 2018 from http://dined.io.tudelft.nl/en/database/tool

DINED(2). (n.d.). *How it works, Introduction*. Retrieved on October 30 2018 from http://dined.io.tudelft.nl/en/how-it-works

Egermann, M., Kasten, P., & Thomsen, M. (2008). Myoelectric hand prostheses in very young children. *International Orthopaedics, 33*(4), 1101-1105. doi:10.1007/s00264-008-0615-y

Elliot, S. (2014). *modBrowseFolder*. Retrieved on 30 October 2018 from https://www.theswamp.org/index.php?topic=47580.0

Ellipse. (n.d.). *DINED ellipse tool*. Retrieved on October 30 2018 from https://dined.io.tudelft.nl/en/ellipse/tool

Gyrobot. (2014). *Flexy-Hand 2*. Retrieved from https://www.thingiverse.com/thing:380665

James, M. A. (2006). Impact of Prostheses on Function and Quality of Life for Children with Unilateral Congenital Below-the-Elbow Deficiency. *The Journal of Bone and Joint Surgery (American) J Bone Joint Surg Am, 88*(11), 2356. doi:10.2106/jbjs.e.01146

ten Kate, J.D. (2016). *The FA3D Hand, Design and evaluation of a Functional and Anthropomorphic hand prosthesis using the advantages of 3D-printing* (Master’s thesis). Retrieved on October 30 2018 from https://repository.tudelft.nl/islandora/object/uuid%3A7923ea86-8d83-4acc-95e2-530b18920ee2?collection=education

de Leva, P. (1996). Adjustments to Zatsiorsky-Seluyanov's Segment Inertia Parameters . *Journal of Biomechanics 29 (9)*, 1223-1230

Limehouse, J.W., & Farnsworth. T.A. (2005). A preliminary study of 40+ upper extremity patients using the animated control system. *Myoelectric Controls/Powered Prosthetics Symposium*, (pp. 196). Fredricton, NB, Canada.

Muilenburg, T. B. (2009). Prosthetics for Pediatric and Adolescent Amputees. *Cancer Treatment and Research Pediatric and Adolescent Osteosarcoma*, 395-420. doi:10.1007/978-1-4419-0284-9_22

Newman, C. (2014). *VBA Code To Save As A New Version If File Already Exists*. Retrieved on 30 October 2018 from https://www.thespreadsheetguru.com/blog/2014/8/19/save-as-a-new-version-if-file-already-exists

NinjaTek. (2016). *Technical Specifications, NinjaFlex® 3D Printing Filament*. Retrieved on October 30 2018 from https://ninjatek.com/wp-content/uploads/2016/05/NinjaFlex-TDS.pdf

Noë, T. (2016). *The problems and solutions for growing children missing an upper limb, a literature study. (Literature study).* Retrieved from https://www.dropbox.com/sh/b7ubk9vtkt9iy41/AAAaK-1n2u7JHVOztCHC3jBca?dl=0

Otto Bock. (2014). *Prosthetics, Upper Limb*, p. 100. Retrieved on October 30 2018 from https://www.ottobock.co.uk/media/local-media/brochures/prosthetics/catalogues/upper-limb-catalogue-646k6-masteren_2014.pdf

RepRap.org. (2016). *Minitronics 1.0*. Retrieved on October 30 2018 from https://reprap.org/wiki/Minitronics_10

ReprapWorld. (2014). *MINITRONICS v1.1 DATASHEET*. Retrieved on 30 October 2018 from https://reprapworld.com/documentation/datasheet%20minitronics%20v1.1.pdf

Shaperman, J., Landsberger, S. E., & Setoguchi, Y. (2003). Early Upper Limb Prosthesis Fitting: When and What Do We Fit. *JPO Journal of Prosthetics and Orthotics, 15*(1), 11-17. doi:10.1097/00008526-200301000-00004

Vasluian, E., Jong, I. G., Janssen, W. G., Poelma, M. J., Wijk, I. V., Reinders-Messelink, H. A., & Sluis, C. K. (2013). Opinions of Youngsters with Congenital Below-Elbow Deficiency, and Those of Their Parents and Professionals Concerning Prosthetic Use and Rehabilitation Treatment. *PLoS ONE, 8*(6). doi:10.1371/journal.pone.0067101

Vinet, R., Lozac’h, Y., Beaudry, N., & Drouin, G. (1995) Design methodology for a multifunctional hand prosthesis. *Journal of Rehabilitation Research and Development, 32*(4), 316-324

Wagner, L. V., Bagley, A. M., & James, M. A. (2007). Reasons for Prosthetic Rejection by Children With Unilateral Congenital Transverse Forearm Total Deficiency. *JPO Journal of Prosthetics and Orthotics, 19*(2), 51-54. doi:10.1097/jpo.0b013e3180421539

2. Image credits

Figure 1:

From the top left, clockwise:

1. WILMER. (n.d.). [image] Retrieved on 30 October 2018 from https://www.tudelft.nl/en/3me/departments/biomechanical-engineering/research/biomechatronics-human-machine-control/delft-institute-of-prosthetics-and-orthotics/products/prostheses/wilmer-appealing-prehensor/
2. TRS prosthetics. (n.d.). [image]. Retrieved on 30 October 2018 from https://www.trsprosthetics.com/product/child-lite-touch-hand/
3. Reeves, J. L. (2010). [image]. Retrieved on 30 October 2018 from http://www.bornjustright.com/2010/08/helper-arm-buy-in/
4. AOL. (2016). [image]. Retrieved on 30 October 2018 from https://www.aol.com/article/2016/03/30/little-girl-gets-3d-printed-prosthetic-hand-for-free/21335567/
5. Egermann, M., Kasten, P., & Thomsen, M. (2008). Myoelectric hand prostheses in very young children. *International Orthopaedics, 33*(4), 1101-1105. doi:10.1007/s00264-008-0615-y

Figure 2:

TRS prosthetics. (n.d.). [image]. Retrieved on 30 October 2018 from https://www.trsprosthetics.com/product/child-lite-touch-hand/

Figure 30:

DINED. (n.d.). [image]. Modified by author. Retrieved on October 30 2018 from http://dined.io.tudelft.nl/en/database/tool

B. Requirements

1. Growth chart of key measurements

Figure 29 displays the growth over time of the stature, arm length, and hand length. The datapoints were sourced from the DINED database (DINED, n.d.), specifically from the “Dutch children, kima1993” dataset. As can be seen, the datapoints for each dimension follow a roughly linear trajectory. A trendline was added and the equation for the trendline is displayed as well. Based on the chart one can conclude that the arm length of a child grows roughly 24 mm per year and that the hand length grows roughly 6 mm per year.

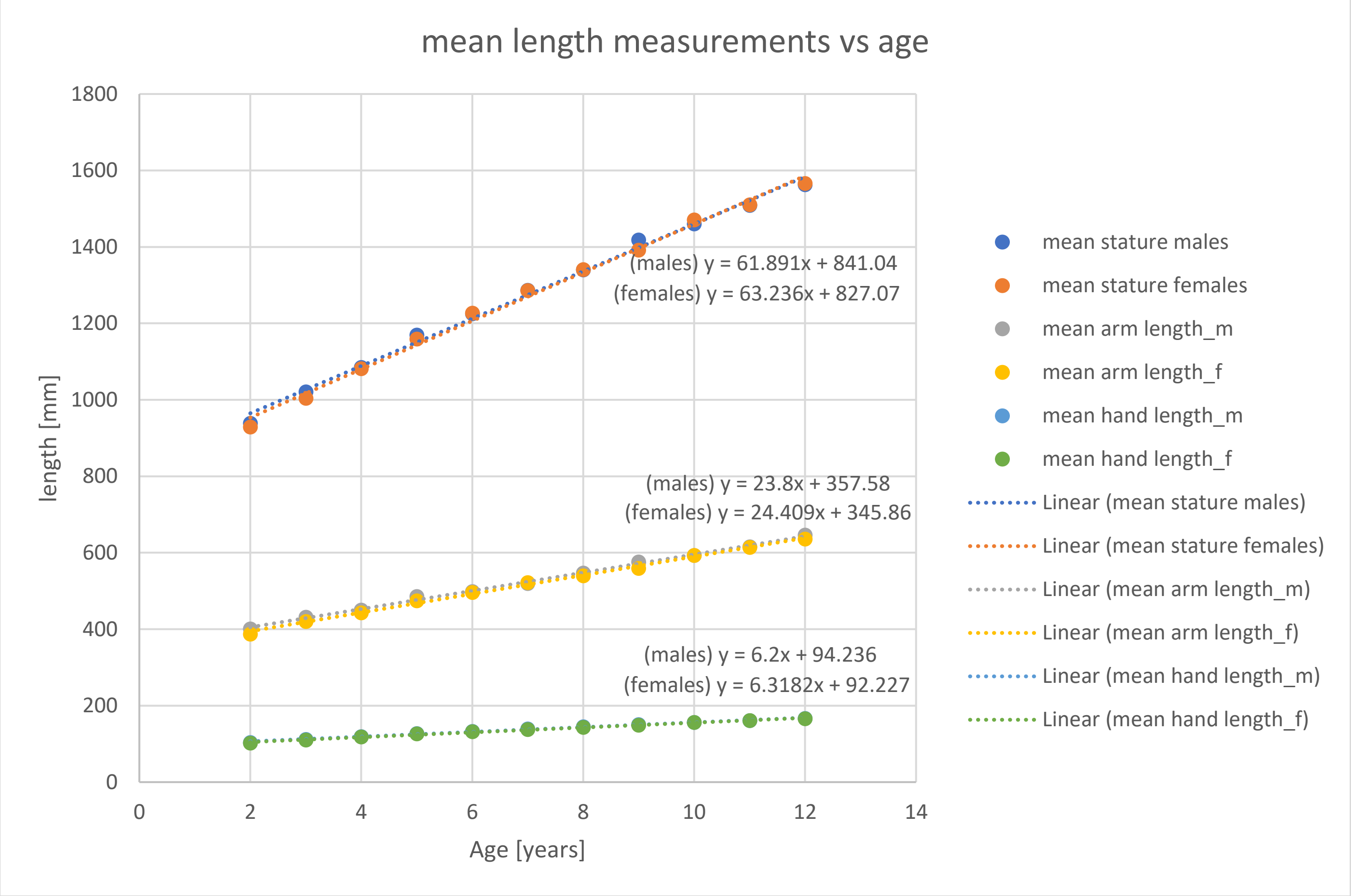


Figure 29: A chart which plots key measurements against the age of a child. Stature, arm length and hand length for both males and females are plotted.

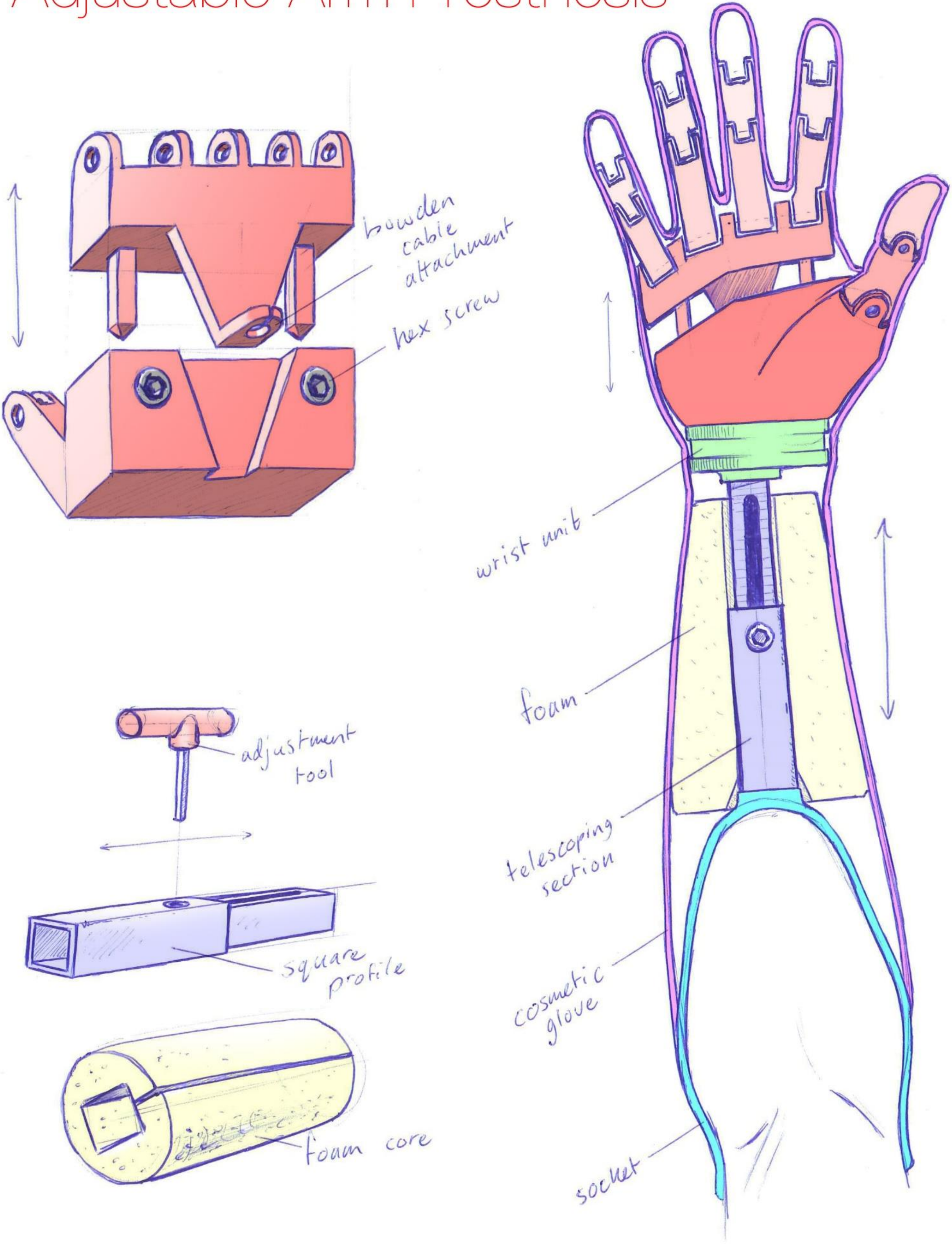
C. Concept development

1. Concepts

This section of the appendix describes the concepts in more detail

Concept 1

Adjustable Arm Prosthesis



Adjustable arm prosthesis

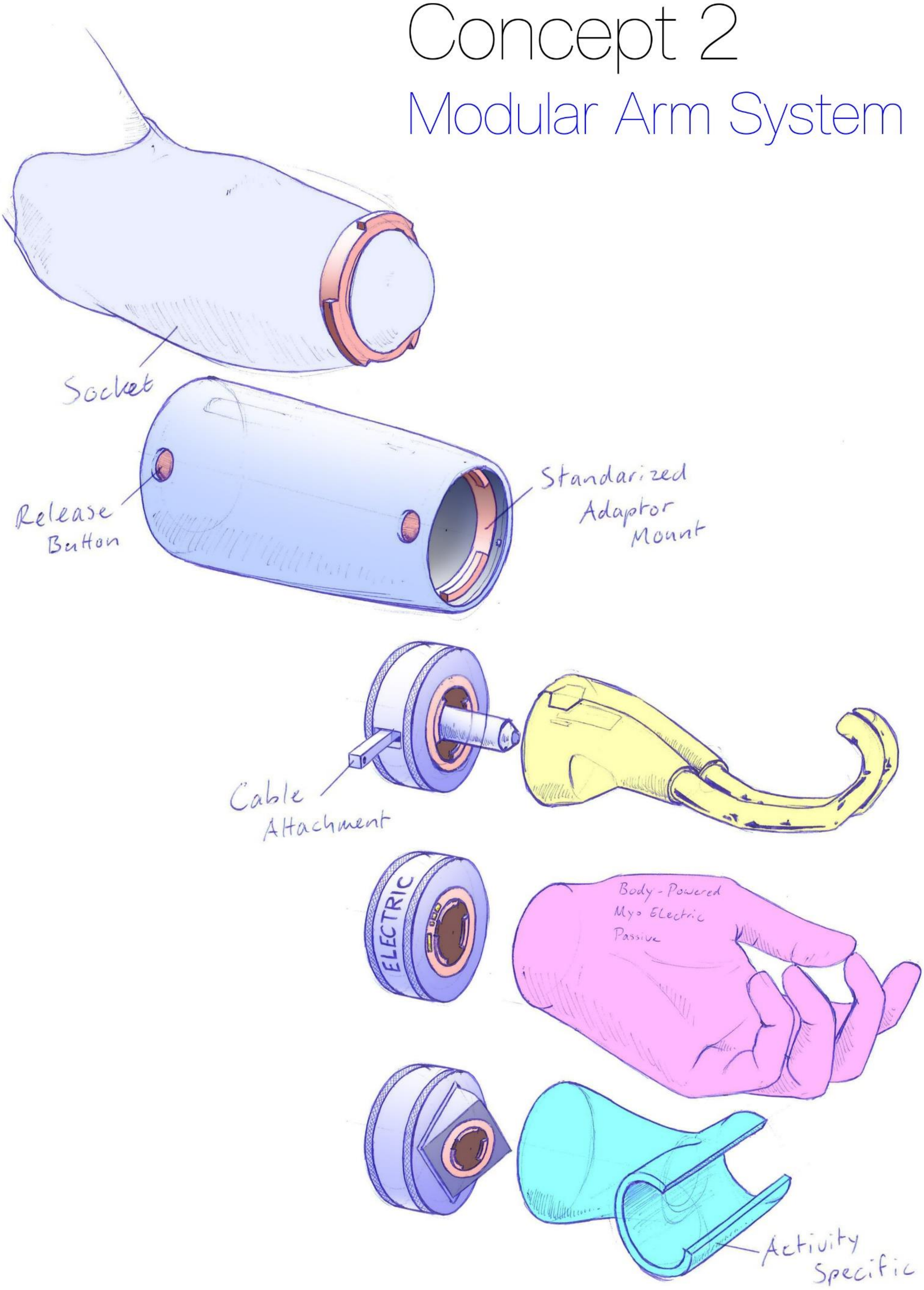
The adjustable arm prosthesis is mechanically adjustable in order to be suitable for a wider range of users and to accommodate growth. It features a natural looking active hand which is actuated through a Bowden cable by either an electric motor or a shoulder harness. The thumb is passively adjustable and the arm can also be configured as a fully passive prosthesis. Length adjustments of the telescoping sections can be made using a single tool. Finally, the prosthesis is covered with a cosmetic glove for a natural appearance.

- 72mm overall length adjustment range (3-year life cycle)
- Adjustable forearm length
- Adjustable palm length
- Fixed finger length
- Square telescoping forearm profile
- Foam forearm core for a natural appearance
- 1 Degree of Freedom actuated hand
- Electrically- or body-powered through a Bowden cable
- No cable adjustment required after length adjustments
- Conventional socket and wrist unit

Adjustability in length allows this prosthesis to be suitable for a wider range of ages and be able to accommodate growth. This can be useful both functionally and cosmetically. The adjustability of this concept also brings with it several drawbacks. This prosthesis will likely be heavier, more expensive, and more complex than comparable conventional prostheses.

Concept 2

Modular Arm System



Modular prosthetic arm system

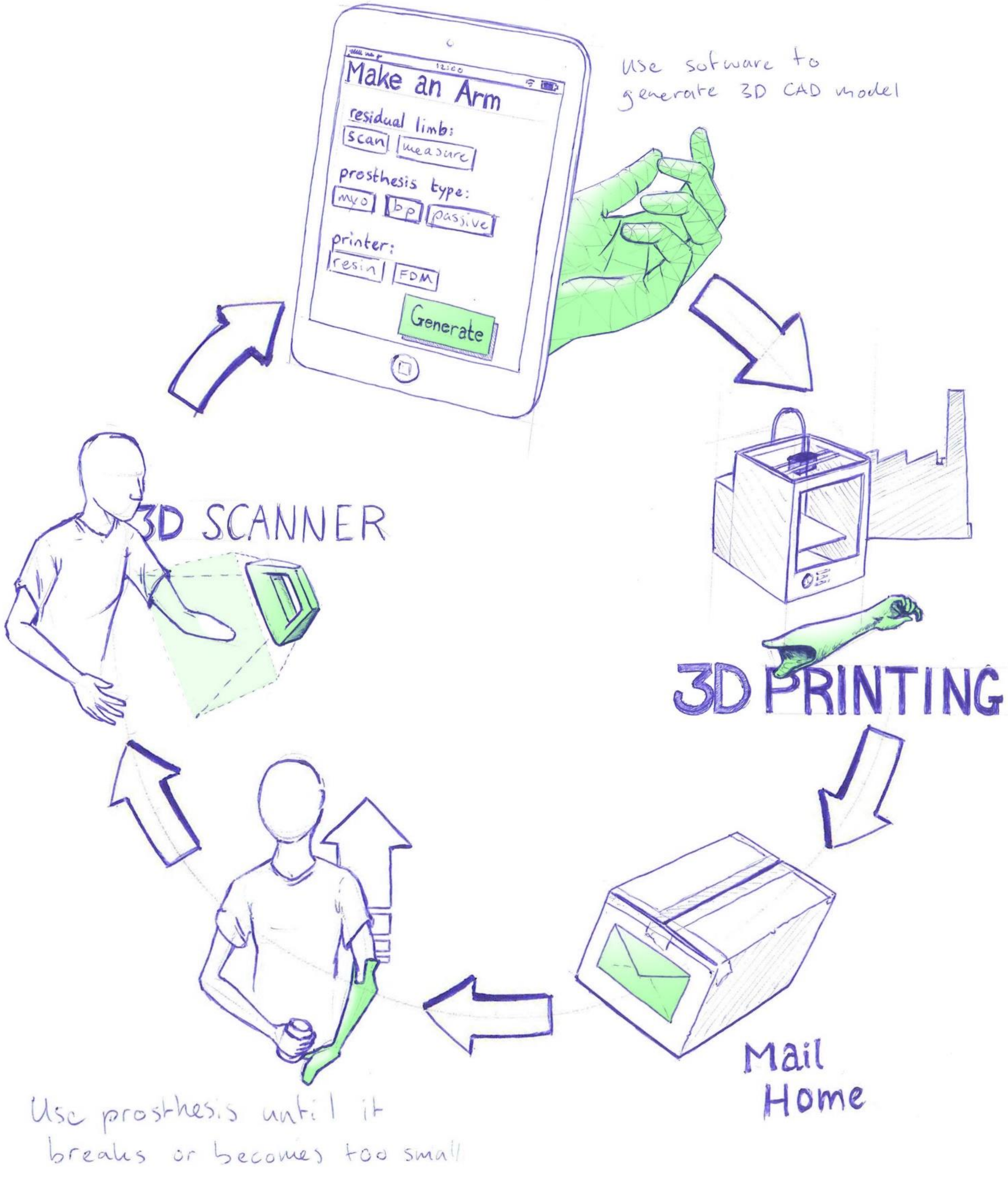
Concept 2 consists of a modular arm system. It utilizes a set of standardized adaptor mounts that allow components to be easily and exchanged. For instance, as the arm becomes too short, a longer forearm section can be fitted without the aid of a professional. Different types of terminal devices can be fitted and exchanged. This way the user can try out different technologies, or swap out different terminal devices on-the-go based on the situation. The adaptors use a bayonet type fitting that can be unlocked by a button.

- Allows quick exchange of different types of terminal devices (myoelectric, body-powered (BP), passive, activity-specific)
- Allows quick exchange of different terminal devices based on the same type (e.g. BP split-hook to BP hand)
- Locking bayonet type fittings
- Components can be exchanged without the aid of a professional
- Allows for more standardized production of prosthetic components
- “monocoque” forearm section (no internal frame)
- 8 forearm sizes for 1-12 years old
- 4 hand sizes for 1-12 years old

The advantage of incorporating modularity is that it allows the user to use and experiment with different types of terminal devices. Additionally, the modular prosthetic arm will always be roughly the right size. The end result is a very functional and robust solution. The main drawback of this concept is the costs. While the standardization of components could drive down costs due to economies of scale, the user is still required to own a large number of different components. Components could possibly be recycled when they become obsolete in order to reduce costs for the user. However, this concept will likely be more expensive than a more conventional approach. It also requires a large amount of components and terminal devices to be developed specifically for this prosthesis system, or a reliance on compatibility between 3rd party manufacturers.

Concept 3

Frequent replacements



Affordable frequent replacements

Another strategy that can be used to accommodate growth is to frequently replace prostheses, which this concept is based around. What prohibits frequent replacements of current prostheses are their costs, as well as the time and effort it takes to get a new prosthesis fitted. This concept utilizes software combined with 3D-scanning and –printing technologies to allow affordable frequent replacements. A 3D scan is made of the residual limb and, optionally, of the other healthy limb. Software is then used to essentially replace the professional labor by creating a 3D CAD model of the prosthesis that correctly fits and aligns with the residual limb. The result is a ready-to-print file which can be sent off to a 3D printing service. The prosthesis is 3D printed in a single go and doesn't require any complicated assembly. The entire process doesn't require any specialized computer skills from the user or their family. Finally, after receiving the prosthesis in the mail, the user wears the prosthesis until it becomes too small or breaks and then repeats the process.

- Doesn't require special software from the user's side (uses web-app)
- Doesn't require CAD/CAM knowledge from the user or their family
- 3D scan results in:
    - 3D model of the residual limb
    - 3D model or measurements from healthy limb (optional)
- Back-end software generates:
    - 3D model of the complete prosthetic arm based on the 3D scan
    - Proper alignment and fitting of prosthetic socket and arm
    - Ready to print 3D CAD file
- 3D printing service provides:
    - "single piece" 3D print of entire prosthetic arm
    - Shipping to users' home
- Allows for myoelectric, body-powered, and passive prostheses

This concept results in a functional, lightweight prosthetic arm that is always the right size. It is able to do so without additional costs or effort compared to conventional prostheses. The main drawback of this concept is its unproven feasibility. The 3D-scanning and -printing technology is adequate and continues to improve year over year. However, there currently doesn't exist any software to automatically generate well-fitting prostheses from a 3D scan. It is important to note that a good socket is not simply the negative of a residual limb. Development of such software would require significant investments which could be a challenge for such a specialized problem.

2. Selection criteria

Function

The most important criteria for a prosthetic arm, as most prostheses worn by children are used as tools to accomplish tasks they wouldn't otherwise be able to do. A prosthesis which lacks function is less likely to add value and be adopted.

Cosmetics

The appearance of a prosthesis is incredibly important for successful adoption and use. Appearance alone can be the sole reason to wear a prosthesis for some users. For children this criterion is even more important than for adults. Cosmetic taste is subjective and will vary between users, but for most, a natural appearance would be preferred.

Comfort

Comfort is important for the continued success of a prosthesis. The main factors influencing comfort are weight (and associated center of gravity) and breathability. Prosthetic designs should aim for minimal weight and bulk, as well as a breathable prosthetic interface. For instance, a concept that is inherently heavier is more likely to be rejected by the user.

Cost

Cost is a factor in prosthesis selection. A lower-cost prosthesis is preferred and would make it accessible to more people.

Reliability

For successful adoption, good reliability is required. A mechanical or electrical failure of a prosthesis could lead to device abandonment.

D. Final Design

This section of the appendix includes additional information relevant to the *final design* part of the main report. This section is divided into two sections, *the hand* (D.1.) and *the GUI* (D.2.).

1. The hand

Independent variables and modeling techniques

The 3D CAD (Computer Aided Design) model of the hand is dimensioned based on 7 main independent variables:

1. middle finger length
2. middle finger thickness
3. wrist width
4. hand width
5. thumb thickness
6. thumb length
7. hand length

These variables are illustrated in figure 30.

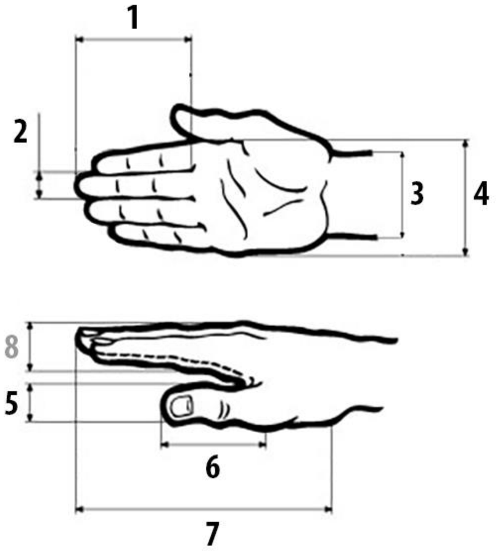
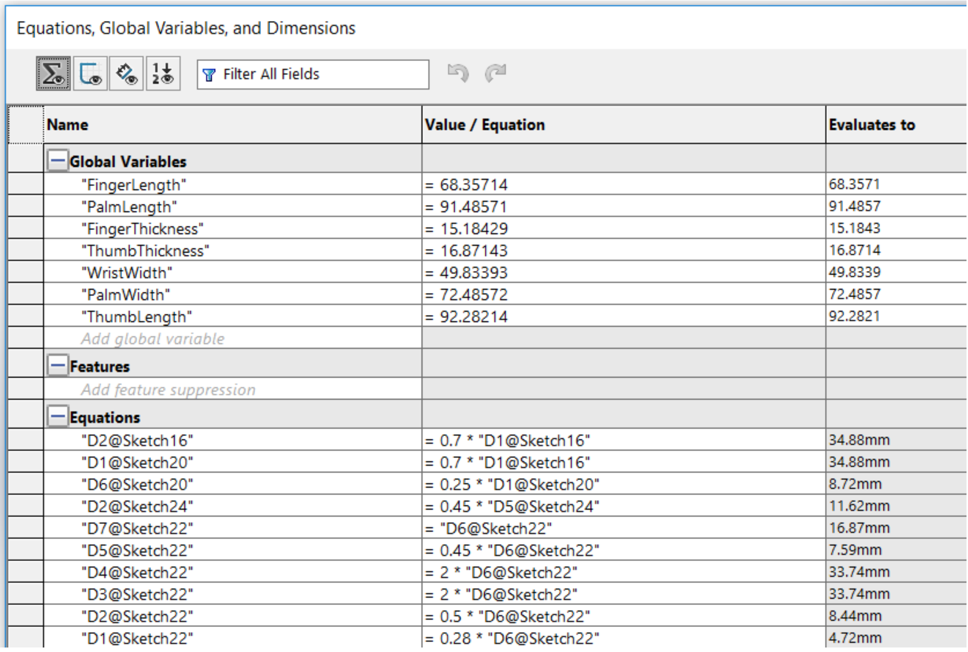


Figure 30: Independent variables that determine the dimensions of the 3D hand model

The initial 7 main variables are defined in the Solidworks model as *global variables*. The global variables are set by the VBA program with the values of the variables depending on the user input. All other dimensions in the Solidworks model are in some way based on these variables. For instance, the length of the index finger is defined as a certain proportion of the middle finger length. This is done for all dimensions mainly by using equations in Solidworks. For example, “D1@Sketch1” = 0.67 * ”FingerLength”. Every sketch is fully defined by proportioning every dimension to existing dimensions. This results in a hand that is fully scalable for a wide range of sizes. Figure 31 shows some of the equations used in the Solidworks hand model.



Name	Value / Equation	Evaluates to
Global Variables		
"FingerLength"	= 68.35714	68.3571
"PalmLength"	= 91.48571	91.4857
"FingerThickness"	= 15.18429	15.1843
"ThumbThickness"	= 16.87143	16.8714
"WristWidth"	= 49.83393	49.8339
"PalmWidth"	= 72.48572	72.4857
"ThumbLength"	= 92.28214	92.2821
Add global variable		
Features		
Add feature suppression		
Equations		
"D2@Sketch16"	= 0.7 * "D1@Sketch16"	34.88mm
"D1@Sketch20"	= 0.7 * "D1@Sketch16"	34.88mm
"D6@Sketch20"	= 0.25 * "D1@Sketch20"	8.72mm
"D2@Sketch24"	= 0.45 * "D5@Sketch24"	11.62mm
"D7@Sketch22"	= "D6@Sketch22"	16.87mm
"D5@Sketch22"	= 0.45 * "D6@Sketch22"	7.59mm
"D4@Sketch22"	= 2 * "D6@Sketch22"	33.74mm
"D3@Sketch22"	= 2 * "D6@Sketch22"	33.74mm
"D2@Sketch22"	= 0.5 * "D6@Sketch22"	8.44mm
"D1@Sketch22"	= 0.28 * "D6@Sketch22"	4.72mm

Figure 31: Global variables and some of the equations used in the Solidworks hand model

There are two variables that are not dependent on the 7 global variables, the hinge thickness and hinge length. The values for these variables are fixed as the function of the living hinges depend on these dimensions, regardless of the scaling of the hand. These variables could be made to be user configurable or scaled relative to one of the 7 global variables if desired. For instance, the hinges could be made thinner for young users, or thicker for older, more active, users.

One critical real-world dimension which is missing from the independent global variables is the hand thickness (number 8 in figure 30). It is important that this variable is independent of the other variables because hand thickness is not strongly correlated to any of the other measurements (a longer hand is not necessarily thicker). However, when initially modeling the hand this measurement was overlooked and made dependent on other dimensions. The way the “knuckles” are currently modeled makes it hard to add hand thickness as an independent global variable afterwards. The global variable *thumb length* (number 6 in figure 30) could likely be removed and made dependent on the middle finger length.

The hand was modeled using simple geometric shapes where possible. This simplified the modeling and makes the model easier to scale and more robust. The “knuckles” are modeled in this way and don’t require any lofts. However, a lot of parts had to be modeled using lofts due to their organic shape. The fingers, thumb and lower palm were all modeled using complicated lofts. The reliability of these lofts when scaled was initially a concern, but when modeled correctly this design method turned out to be very robust. All errors when scaling the final design ended up being caused by failed fillets. This was unforeseen during the start of the design and modeling process as fillets tend to be very reliable in Solidworks for simple parts. When filleting between complex non-geometric parts, Solidworks fillet function proved to be very unreliable and finicky when the hand was scaled. Fillets shouldn’t be used for critical parts of the design such as the transition between the palm and thumb, if the hand were to be modeled again from the ground up.

It should be noted the current proof of concept scalable CAD model is made out of a single piece. The final design consists of three parts that connect using rail interfaces. Before this project can be useful in the real world, the hand model should be further developed to include these rail interfaces and channels through the fingers for the actuation wires. This is a significant amount of work which is why it was not included in the thesis work. A scalable rail interface would be hard to design, especially for the smaller hands, but not impossible.

2. The GUI

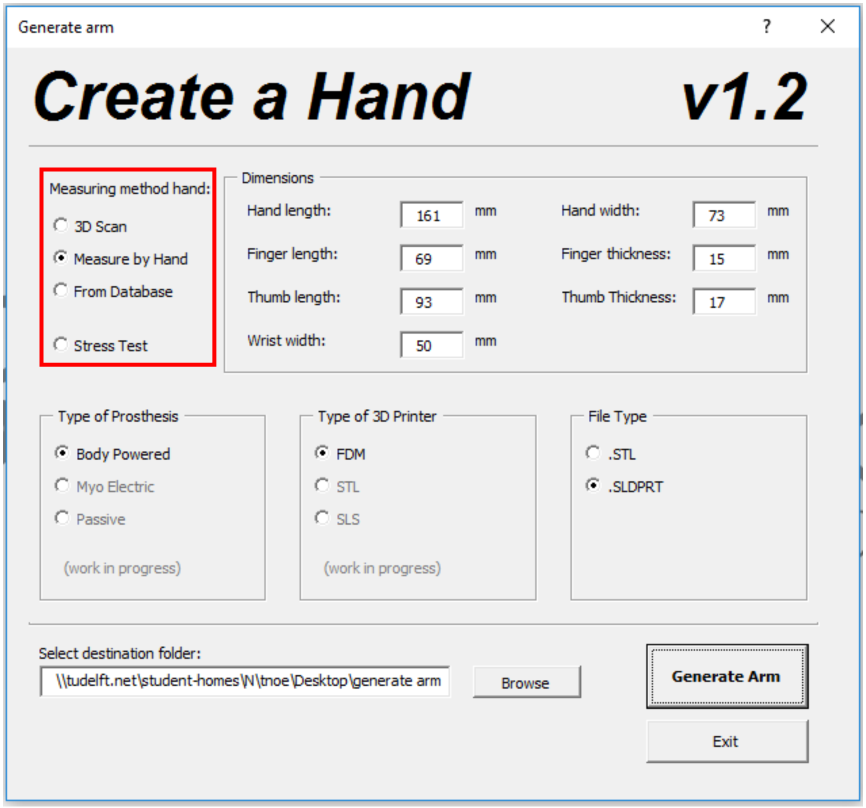
The functionality of the Graphical User Interface (GUI) is described in the main section of the report. This section documents the GUI in more detail.

Measurement input methods

In order to generate 3D models of a hand there are four input methods; 3D scanning (not functional in this thesis), Manual measurements, Measurements from a database, or a Stress test.

The 3D scanning input option is nonfunctioning. It was included because it would be a useful future development. Instead of relying on manual measurements, with their potential user errors, an accurate 3D scan could be made with a smartphone or 3D scanner. This is a subject that could be an entire thesis in and of itself and is currently also being worked on by J.S. Cuellar as part of a PhD project.

The second measurement input method is to use manual measurements. This is the input method most likely used by the end user. It requires the user to input 7 different measurements from the healthy hand which are used to generate a closely matching prosthetic hand. The proof of concept (the hand model and accompanying GUI) was mainly created to test if this process was possible and practical. The GUI with this input method selected is shown in figure 32.



Generate arm

Create a Hand **v1.2**

Measuring method hand:
○ 3D Scan
● Measure by Hand
○ From Database
○ Stress Test

Dimensions
Hand length: 161 mm
Finger length: 69 mm
Thumb length: 93 mm
Wrist width: 50 mm
Hand width: 73 mm
Finger thickness: 15 mm
Thumb Thickness: 17 mm

Type of Prosthesis
● Body Powered
○ Myo Electric
○ Passive
(work in progress)

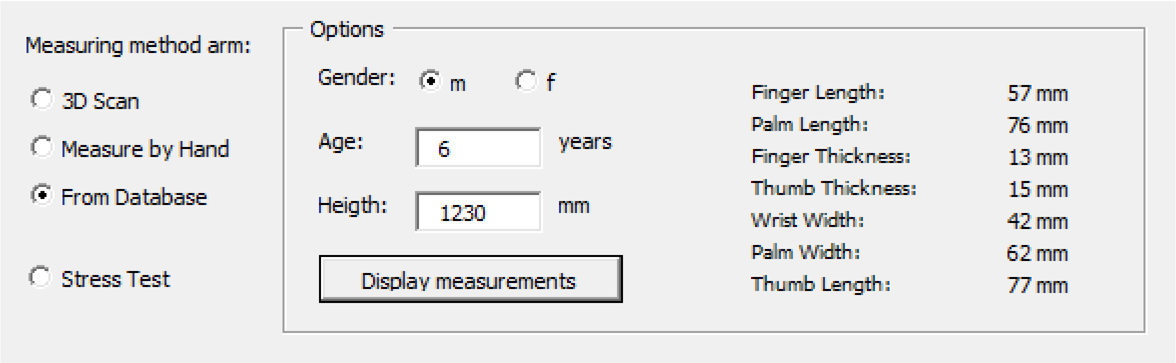
Type of 3D Printer
● FDM
○ STL
○ SLS
(work in progress)

File Type
○ .STL
● .SLDPRT

Select destination folder: \\tudelft.net\student-homes\N\tnoe\Desktop\generate arm

Figure 32: The Graphical User Interface (GUI) with the different input methods highlighted.

The database input method calculates the required measurements based on the gender, age, and stature of a child. This can result in a fairly accurate estimate because most of the desired dimensions are correlated to the stature of a person. The GUI currently uses *gender*, *age*, and *stature* but it was later discovered that using just *gender* and *stature* results in slightly more accurate estimates. The GUI with the database input method selected is shown below in figure 33.



Measuring method arm:
○ 3D Scan
○ Measure by Hand
● From Database
○ Stress Test

Options
Gender: ● m ○ f
Age: 6 years
Heigth: 1230 mm

Finger Length:	57 mm
Palm Length:	76 mm
Finger Thickness:	13 mm
Thumb Thickness:	15 mm
Wrist Width:	42 mm
Palm Width:	62 mm
Thumb Length:	77 mm

Figure 33: The available options when the "From Database" input method is selected

The ellipse tool (Ellipse, n.d.) was used to determine the correlation between the stature and each desired measurement. These correlation coefficients are calculated over an age-range between 2 and 12 years old.

Table 8: Correlation between stature and various other measurements, males and females 2-12 years old

Male	*Stature*
Hand length	0.967
Finger length	0.938
Hand width	0.929
Thumb breadth	0.833
Hand thickness	0.567

Female	*Stature*
Hand length	0.971
Finger length	0.945
Hand width	0.927
Thumb breadth	0.820
Hand thickness	0.572

As can be seen in Table 8 the stature of a child strongly correlates to most hand dimensions. It also becomes clear that there is no correlation between hand thickness and stature. Hand thickness also doesn't correlate to other dimensions available through DINED such as body weight.

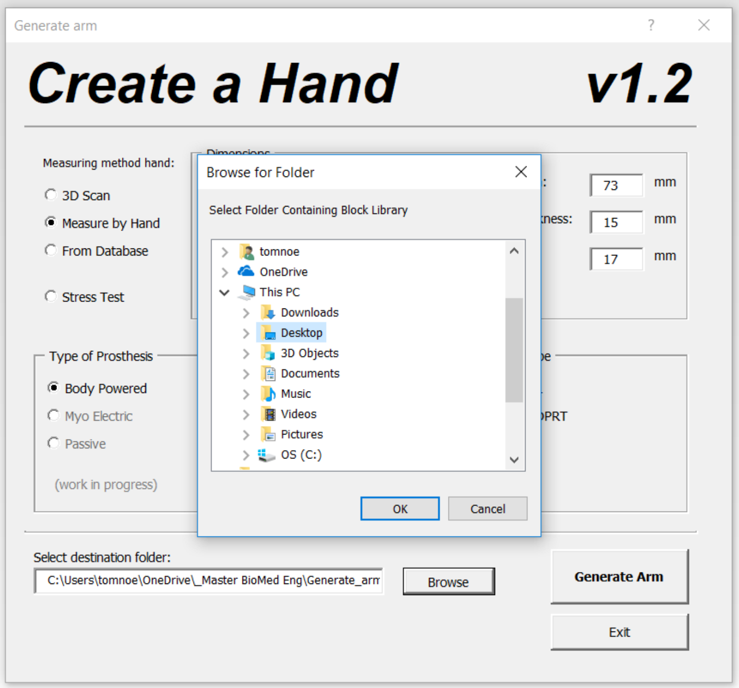
For this reason, manual measurements would be the preferred input method. Besides being able to have an accurate hand thickness it also allows for variations between individual users. Even though there might be a strong correlation between two measurements within a population, these correlations might not apply to every individual.

As mentioned previously in the main report, the *database* input method was also very useful in the development of the hand model and GUI.

Lastly, there is also the *stress test* input method. This option was included to test the robustness of the hand model generation process. The details regarding the testing of this process are described in the validation section of the appendix (Appendix E.2.).

Additional GUI options

Besides the measurement input method options, the GUI offers a few potential options for the CAD file output. In the future, the user would be able to select what type of prosthesis they want and on what type of printer this prosthesis is printed. The current version of this proof of concept is only able to generate body-powered prostheses to be printed on common FDM (fused deposition modeling) 3D printers. There is also the option to select the output file type. The first option is to save as a universal .STL file to be used directly for 3D printing, the other option is to save as a .SLDPRT file which allows the hand model to be viewed and modified in Solidworks. The option to save the output as a Solidworks file was very useful to troubleshoot and develop the hand model and GUI. After a hand was generated it was possible to review the .SLDPRT file and determine what went wrong in the hand generation process. All of the additional options can be seen in figure 32 on the previous page.



Measuring method hand:
○ 3D Scan
● Measure by Hand
○ From Database
○ Stress Test

: 73 mm
ness: 15 mm
17 mm

Type of Prosthesis
● Body Powered
○ Myo Electric
○ Passive
(work in progress)

Select Folder Containing Block Library

Select destination folder: C:\Users\tomnoe\OneDrive_Master BioMed Eng\Generate_ar

Figure 34: The "Browse for Folder" pop up window that allows the user to select the desired destination folder

There are two small options that are currently missing. One is the option to select whether a left or a right hand is generated. The other is to apply an overall scaling factor (e.g. 95%) so the entire hand can be printed slightly smaller than the natural hand if desired. Both these options would be easy to implement in the future.

On the bottom of the GUI it is possible to select the desired destination folder where the output files should be saved to. Figure 34 shows the "Browse for Folder" dialog that pops up. Finally, by clicking the "Generate Arm" button, the CAD hand model is generated based on the selected input method and saved in the desired file location, ready to be reviewed and sent off to be 3D printed.

Excel sheets for “database” input method

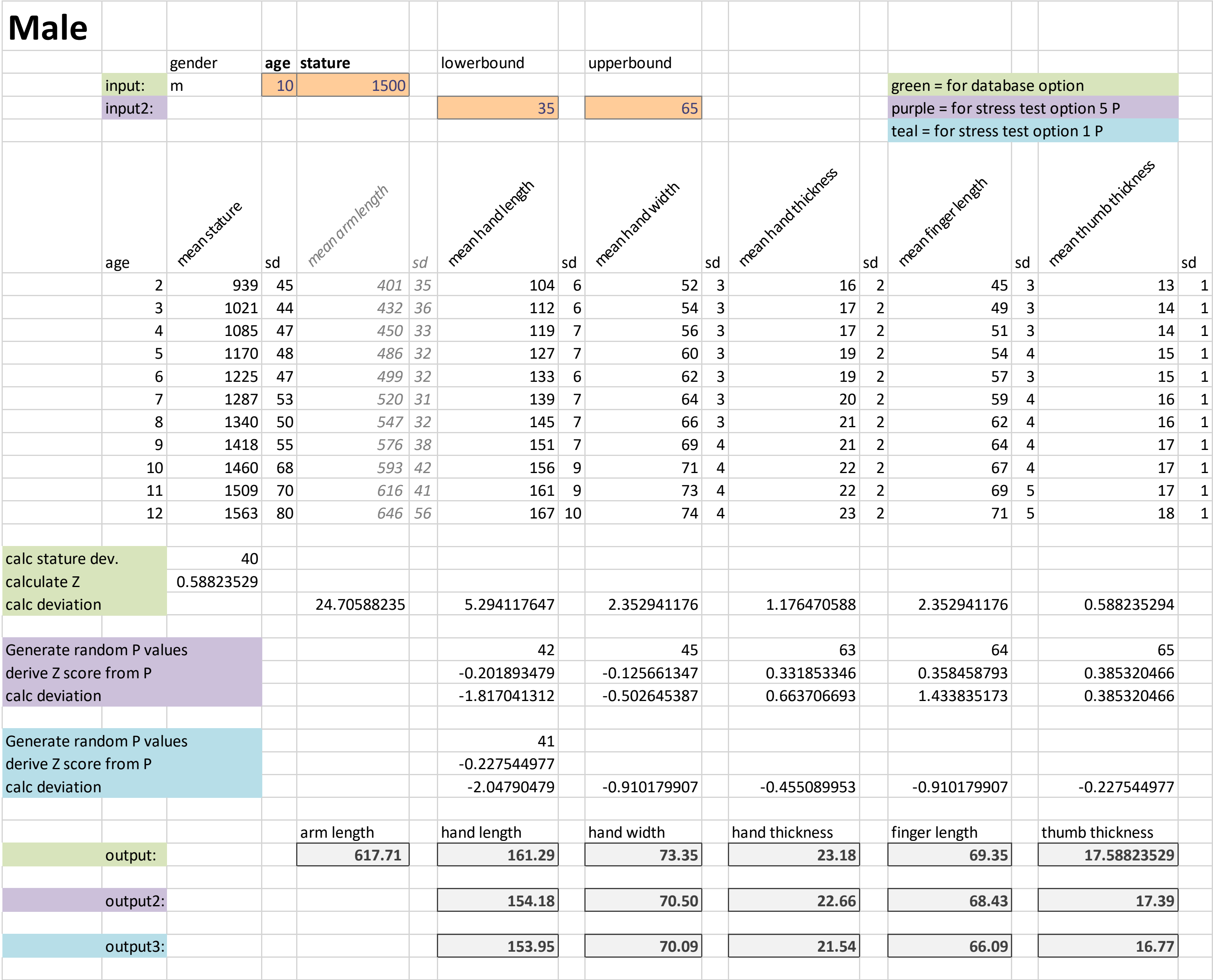
When the *database* input method is selected an excel file (*dined.xlsx*) is used to calculate the required measurements. The user’s age and stature are input into the excel sheet that matches their gender. In these sheets (male and female) the mean values and standard deviations of every required variable at ages from 2 to 12 years old are recorded. These data points were found on DINED, specifically by using the “Dutch children, kima1993” dataset (DINED, n.d.).

In order to get an estimate of the required variables, the deviation of the stature is first calculated (how much does the user vary from the mean in terms of body height). The deviation of the stature is then used to calculate the z-score. This z-score can be applied to the mean values and standard deviations for the desired unknown variables. This results in the user’s estimated deviation from the mean for every variable. The deviation is then added up to the mean value to end up with the estimated “measurements”.

This method can be used to estimate somewhat accurate measurements because the DINED sample data is normally distributed and the desired variables are correlated to the stature (except for hand thickness).

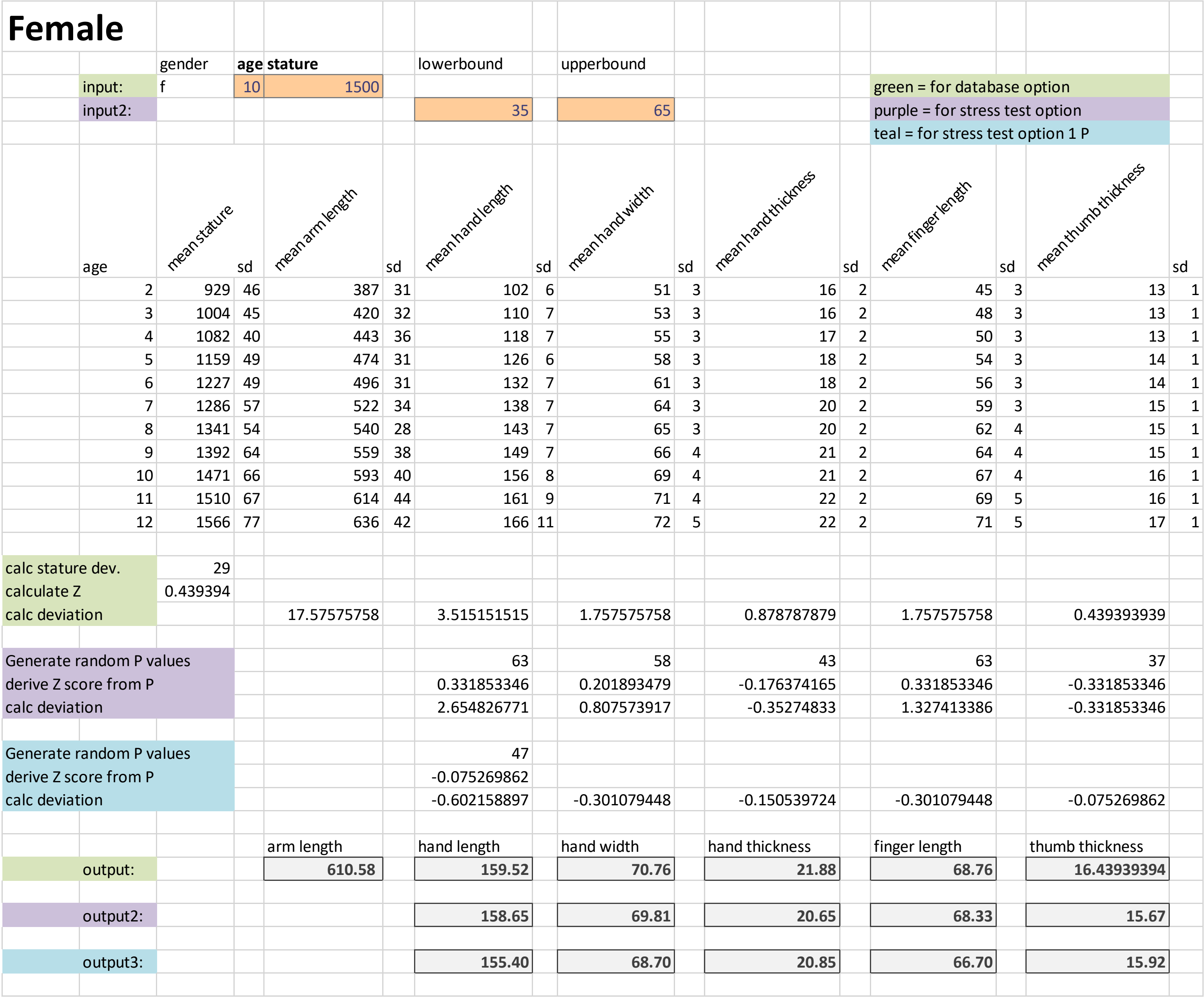
Figures 35 (male) and 36 (female) on the following pages show the excel sheets which are used to generate the required measurements. The inputs and outputs relevant to the “From Database” GUI input method are highlighted with green. The interactive excel file itself can be found in the digital resource pack, see Appendix F, *dined.xlsx*.

The excel sheets also contain the inputs and outputs required for the *stress test* portion of the GUI. The method of generating “measurements” is the same as described previously, except that the deviation is randomly generated within the selected deviation range. The inputs and outputs for the stress test are marked in purple and teal (for 5 random P’s and 1 random P).



Male															
		gender	**age**	**stature**		lowerbound		upperbound							
	input:	m	10	1500								green = for database option			
	input2:					35		65				purple = for stress test option 5 P			
												teal = for stress test option 1 P			
	age	mean stature	sd	*mean arm length*	*sd*	mean hand length	sd	mean hand width	sd	mean hand thickness	sd	mean finger length	sd	mean thumb thickness	sd
	2	939	45	*401*	*35*	104	6	52	3	16	2	45	3	13	1
	3	1021	44	*432*	*36*	112	6	54	3	17	2	49	3	14	1
	4	1085	47	*450*	*33*	119	7	56	3	17	2	51	3	14	1
	5	1170	48	*486*	*32*	127	7	60	3	19	2	54	4	15	1
	6	1225	47	*499*	*32*	133	6	62	3	19	2	57	3	15	1
	7	1287	53	*520*	*31*	139	7	64	3	20	2	59	4	16	1
	8	1340	50	*547*	*32*	145	7	66	3	21	2	62	4	16	1
	9	1418	55	*576*	*38*	151	7	69	4	21	2	64	4	17	1
	10	1460	68	*593*	*42*	156	9	71	4	22	2	67	4	17	1
	11	1509	70	*616*	*41*	161	9	73	4	22	2	69	5	17	1
	12	1563	80	*646*	*56*	167	10	74	4	23	2	71	5	18	1
calc stature dev.		40													
calculate Z		0.58823529													
calc deviation				24.70588235		5.294117647		2.352941176		1.176470588		2.352941176		0.588235294	
Generate random P values						42		45		63		64		65	
derive Z score from P						-0.201893479		-0.125661347		0.331853346		0.358458793		0.385320466	
calc deviation						-1.817041312		-0.502645387		0.663706693		1.433835173		0.385320466	
Generate random P values						41									
derive Z score from P						-0.227544977									
calc deviation						-2.04790479		-0.910179907		-0.455089953		-0.910179907		-0.227544977	
				arm length		hand length		hand width		hand thickness		finger length		thumb thickness	
	output:			**617.71**		**161.29**		**73.35**		**23.18**		**69.35**		**17.58823529**	
	output2:					**154.18**		**70.50**		**22.66**		**68.43**		**17.39**	
	output3:					**153.95**		**70.09**		**21.54**		**66.09**		**16.77**	

Figure 35: The excel sheet that is used to calculate five unknown variables using mean values and standard deviations sourced from DINED. This sheet contains the data for the male population.



Female															
		gender	**age**	**stature**		lowerbound		upperbound							
	input:	f	10	1500								green = for database option			
	input2:					35		65				purple = for stress test option			
												teal = for stress test option 1 P			
	age	mean stature	sd	mean arm length	sd	mean hand length	sd	mean hand width	sd	mean hand thickness	sd	mean finger length	sd	mean thumb thickness	sd
	2	929	46	387	31	102	6	51	3	16	2	45	3	13	1
	3	1004	45	420	32	110	7	53	3	16	2	48	3	13	1
	4	1082	40	443	36	118	7	55	3	17	2	50	3	13	1
	5	1159	49	474	31	126	6	58	3	18	2	54	3	14	1
	6	1227	49	496	31	132	7	61	3	18	2	56	3	14	1
	7	1286	57	522	34	138	7	64	3	20	2	59	3	15	1
	8	1341	54	540	28	143	7	65	3	20	2	62	4	15	1
	9	1392	64	559	38	149	7	66	4	21	2	64	4	15	1
	10	1471	66	593	40	156	8	69	4	21	2	67	4	16	1
	11	1510	67	614	44	161	9	71	4	22	2	69	5	16	1
	12	1566	77	636	42	166	11	72	5	22	2	71	5	17	1
calc stature dev.		29													
calculate Z		0.439394													
calc deviation				17.57575758		3.515151515		1.757575758		0.878787879		1.757575758		0.439393939	
Generate random P values						63		58		43		63		37	
derive Z score from P						0.331853346		0.201893479		-0.176374165		0.331853346		-0.331853346	
calc deviation						2.654826771		0.807573917		-0.35274833		1.327413386		-0.331853346	
Generate random P values						47									
derive Z score from P						-0.075269862									
calc deviation						-0.602158897		-0.301079448		-0.150539724		-0.301079448		-0.075269862	
				arm length		hand length		hand width		hand thickness		finger length		thumb thickness	
	output:			**610.58**		**159.52**		**70.76**		**21.88**		**68.76**		**16.43939394**	
	output2:					**158.65**		**69.81**		**20.65**		**68.33**		**15.67**	
	output3:					**155.40**		**68.70**		**20.85**		**66.70**		**15.92**	

Figure 36: The excel sheet that is used to calculate five unknown variables using mean values and standard deviations sourced from DINED. This sheet contains the data for the female population.

Code for GUI

The code for the GUI is located towards the end of this appendix due to its length, see Appendix G.2.

E. Validation

1. Fatigue

This section of the appendix documents additional details related to the testing of the 3D printed hinges. The goal of these test was to figure out the long-term durability of the 3D printed “living” hinges.

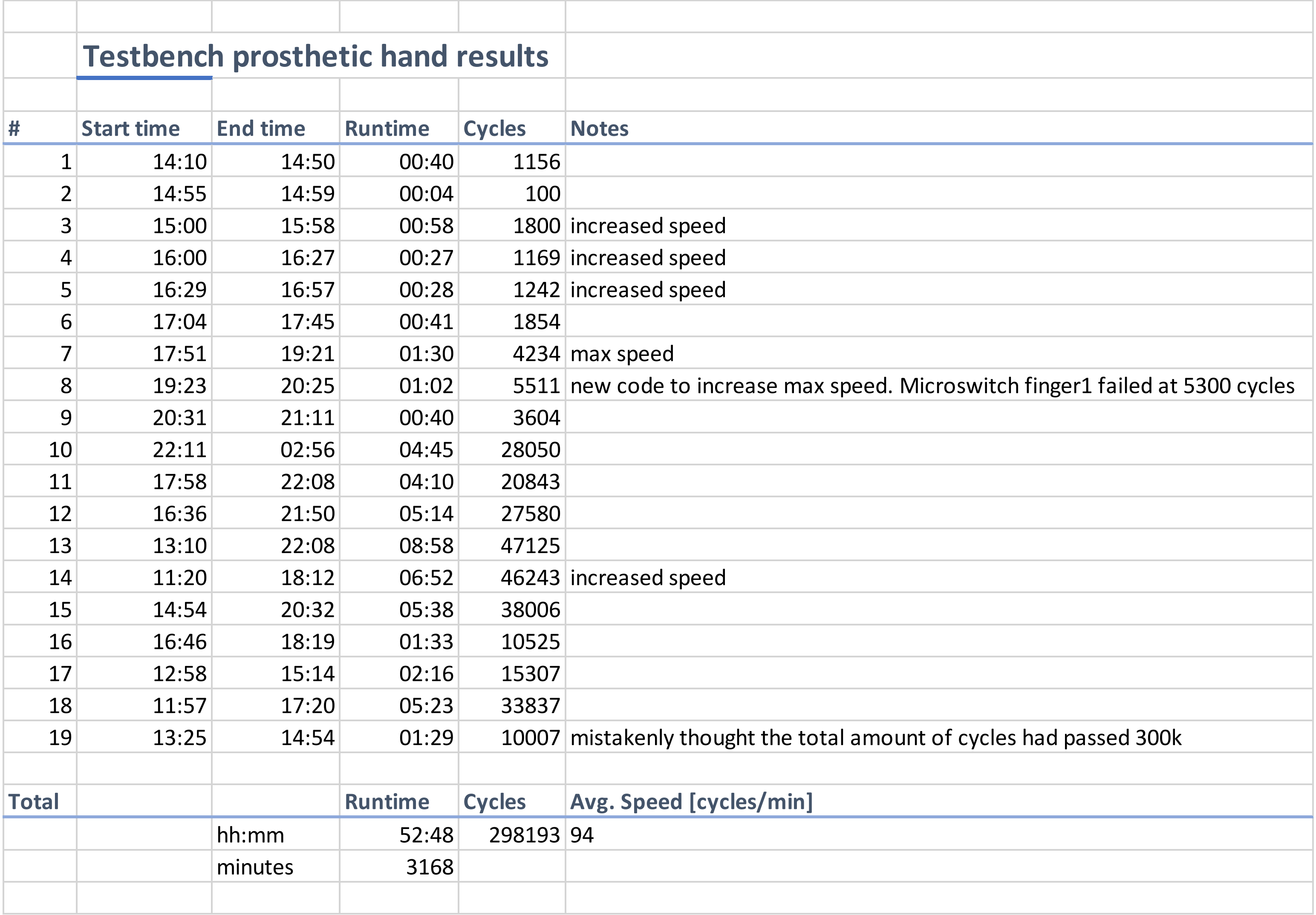
Fatigue test program

Method

In order to test the living hinges’ ability to withstand fatigue, they are put through a set number of grasping cycles. A testbench (Appendix E.1., *Hardware for fatigue testing*) and associated software code (Appendix G.1.) are created to automatically run the prosthetic fingers through the required number of cycles. The target number of cycles for the test fingers was 300,000 cycles, based on research by Limehouse et al. (2005) and Vinet et al., (1995). This number of cycles is far more than likely required but should be the appropriate order of magnitude and should help instill some confidence in the 3D printed living hinges design. The testbench was run in multiple sessions to reach that total number of cycles. Small adjustments to the code and hardware were made throughout the testing.

Results

Figure 37 below shows the results of the fatigue testing. The testing was stopped short of exactly 300,000 because it was mistakenly believed to have passed the number of target cycles.



Testbench prosthetic hand results

#	Start time	End time	Runtime	Cycles	Notes
1	14:10	14:50	00:40	1156	
2	14:55	14:59	00:04	100	
3	15:00	15:58	00:58	1800	increased speed
4	16:00	16:27	00:27	1169	increased speed
5	16:29	16:57	00:28	1242	increased speed
6	17:04	17:45	00:41	1854	
7	17:51	19:21	01:30	4234	max speed
8	19:23	20:25	01:02	5511	new code to increase max speed. Microswitch finger1 failed at 5300 cycles
9	20:31	21:11	00:40	3604	
10	22:11	02:56	04:45	28050	
11	17:58	22:08	04:10	20843	
12	16:36	21:50	05:14	27580	
13	13:10	22:08	08:58	47125	
14	11:20	18:12	06:52	46243	increased speed
15	14:54	20:32	05:38	38006	
16	16:46	18:19	01:33	10525	
17	12:58	15:14	02:16	15307	
18	11:57	17:20	05:23	33837	
19	13:25	14:54	01:29	10007	mistakenly thought the total amount of cycles had passed 300k

Total			Runtime	Cycles	Avg. Speed [cycles/min]
		hh:mm	52:48	298193	94
		minutes	3168		

Figure 37: Results of 19 fatigue test sessions

No failures of the test-hand or string actuation occurred. Additionally, no visible wear of the flexible test hand was observed. The microswitch on finger 1 was the only real failure, this microswitch was a different brand than the other microswitches. The microswitches are only associated with the test setup and not with the actual prosthetic hand design itself. Based on these tests, the long-term durability of the living hinges and string actuation are adequate. At least, when loaded and moved in the intended direction.

These tests were also conducted to determine the optimal living hinge thickness. Since none of the hinges failed, the thinnest (0.5mm) hinge would be sufficient when considering fatigue. A hinge thickness of 1mm was selected for the final design to make the hinges more resilient against peak forces.

Conclusion

The goal of this test was to determine if fatigue is a problem for the chosen “living” hinges. A practical test was chosen due to the uncertainties that are associated with the 3D printing process and the flexible 3D printing material (TPE). Based on the results from the testing, fatigue appears to not be a problem for the flexible 3D printed hinges. The wire actuation method is also durable enough to be used in a practice. The fatigue testing does not prove or disprove real world *overall* reliability, practical user testing should be conducted in order to investigate that further.

Hardware for fatigue testing

The test fingers

The test-hand has 4 fingers of equal length connected to a common set of “knuckles”. The hinge thickness is varied for each finger in 0.5mm increments, from 0.5 to 2mm. the proximal hinges have a thickness-to-length ratio of 1:1 meaning the hinge is as long as it is thick. The most distal hinge in each finger has a thickness-to-length ration of 1:1.5. The testing fingers are saved as *testrig.SLDPRT* and are available through the digital resource pack (see Appendix F.).

The 3D model file *testrig.SLDPRT* is printed in the orientation shown below in figure 38, using a layer thickness of 0.16mm. The flexible material used is sold by 123-3D.nl under the name TPE (ThermoPlastic Elastomer) (123-3D, n.d.).

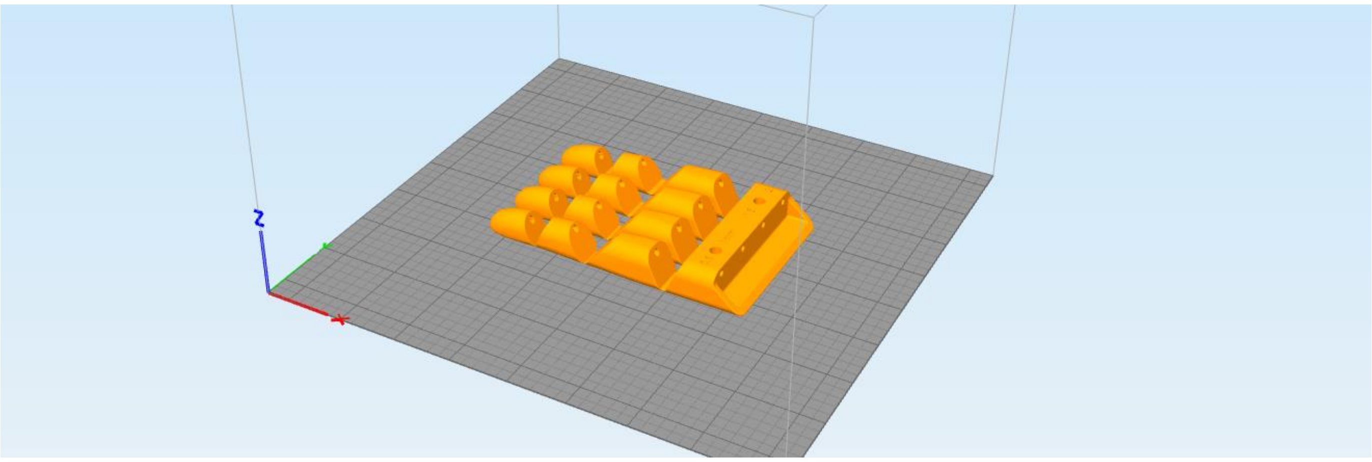


Figure 38: Printing orientation of finger set used for fatigue testing. This is the same orientation the finger set was printed in for the final design

The testing setup

The testbench uses the following electronic components:

- 1x Minitronics v1.0 control board
- 2x NEMA 17 stepper motors
- 4x 3D printer end stop microswitches (wired Normally Open(NO) using internal pull-up resistors)
- Modified 500W computer PSU (power supply unit)

The components are connected to the control board in the following way:

- Left stepper motor on XMOT connector
- Right stepper motor on YMOT connector
- Finger 1 microswitch on AUX1 connector pins 5 & 7
- Fingers 2, 3, 4 microswitches on X, Y, Z connectors

See technical documentation of the minitronics board for more information about the board and the specific connectors and pins that are used (ReprapWorld, 2014).

The testbench uses the following components 3D printed in PLA:

- 1x *testbench.SLDPRT*
- 1x *washer_plate.SLDPRT*
- 1x *endstop_bar2.SLDPRT*
- 4x *endstop_holder*

- 2x *crank.SLDPRT*
- 1x *rod.SLDPRT*

The following additional hardware components are used:

- 2x M3x10mm slotted flat head bolts
- 6x M3x8mm slotted flat head bolts
- 4x M3x10mm slotted pan head bolts
- 12x M3 nuts
- 2x M5x30mm hex bolts
- 2x M5 nuts + washers
- Caperlan 0.30mm braided fishing line

Figure 39 shows the way the described components are put together in an exploded view.

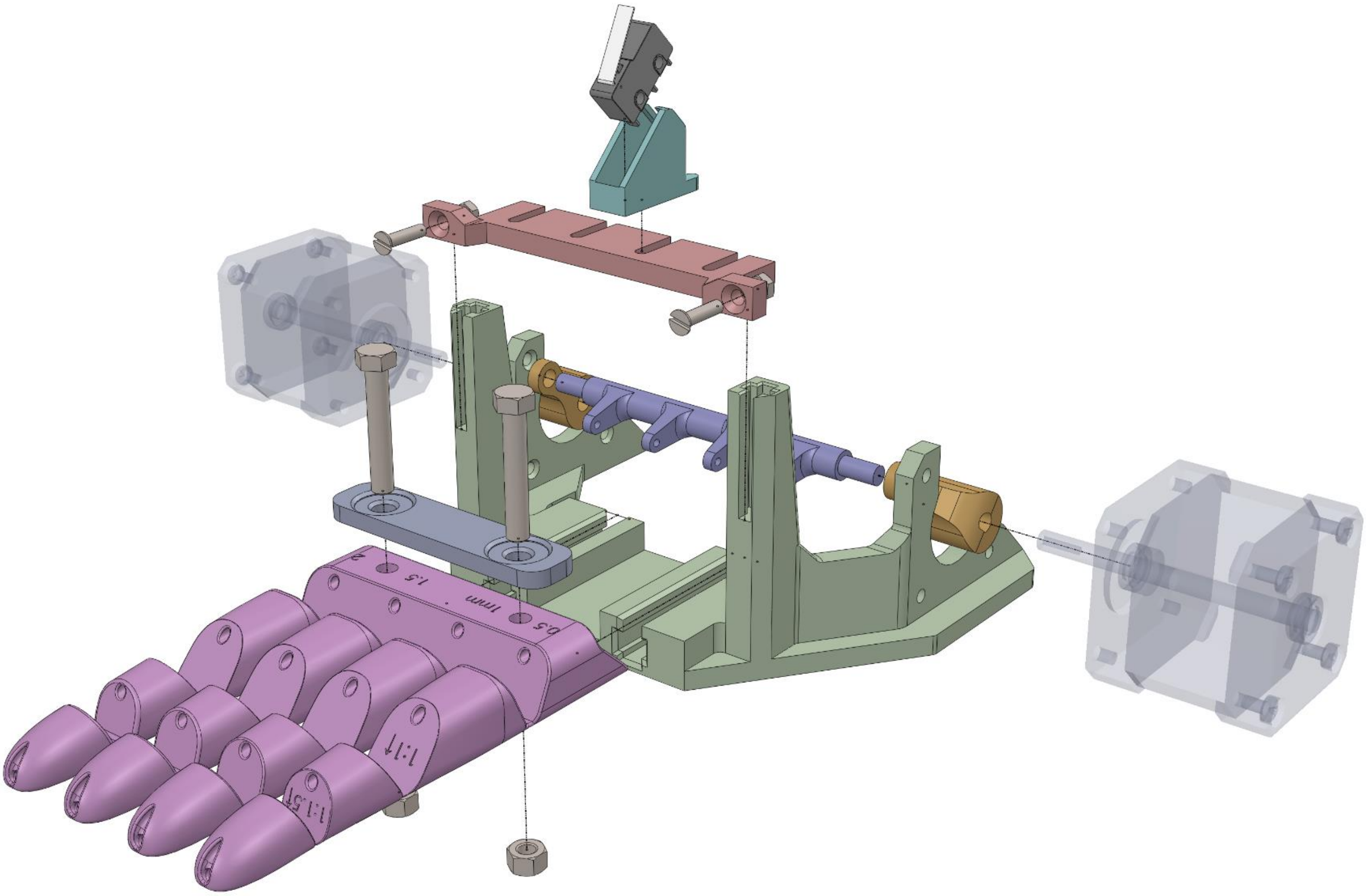


Figure 39: Exploded view of the testbench setup for fatigue testing. Note that only one of the four microswitches is shown

2. Scalability test

Scalability test program

Intro

As described in the main report, a scalability stress-test was performed. The goal of this stress test was to see if the 3D hand model and GUI could reliably generate custom hands for “reasonable” measurements. If it is not possible to reliably generate hands, the frequent replacement concept would not be very functional. DINED data is used to estimate realistic measurements and realistic deviations.

Method

For the stress test, 11 hands were generated for children of every age between 2 and 12 years old. Each hand would have random deviations from the mean. The hand generation process of these 11 hands was then repeated 10 consecutive times (110 hands total). The results (number of errors in which model) are displayed in a message box by the program and manually recorded in an excel file. The generated hands are saved in a Solidworks file format so the failed hands can be reviewed to determine what went wrong.

The range of the deviations is widened in 10 percentile increments at both the lower and upper end of the range. The percentile range includes more and more of the population with each consecutive range increase. The percentile ranges used for scalability testing are shown in Table 9.

Table 9: Widening percentile ranges used for testing the scalability of the hand model

Percentile test ranges
P49-P51
P40-P60
P30-P70
P20-P80
P10-P90
P01-P99

The deviations were applied in two different ways. The first method was to randomly pick a single percentile (within the specified range), and to then apply this same deviation to all 5 independent variables. DINED does not have data for all 7 required variables, so the two missing variables are derived from the other available variables. The generated variables are then used to generate a hand. This is probably fairly realistic but assumes the deviations occur somewhat proportionally. E.g. a randomly chosen percentile value of 67 results in a larger hand size, but since the same percentile is applied to the finger length the fingers scale somewhat proportionally.

The second method generates 5 random percentiles and applies different deviations to each of the 5 independent variables. This is more challenging for the hand model as the hand might have an abnormally short palm combined with exceptionally long fingers.

Results

Table 10 below displays the results of the stress-test for hands where the deviations are calculated by picking a single random percentile value per hand. For every percentile range the program generated a series of hands (2-12 years old) 10 times (110 hands in total).

Table 10: Results stress-test. Single percentile value deviation for every hand, male, 2-12 years old

P range	**Faulty hands**	**Total hands**	**Runtime [mm:ss]**	**comment**
P49-P51	0	110	32:00	
P40-P60	0	110	28:15	
P30-P70	1	110	31:26	1 faulty 3 y.o.
P20-P80	7	110	26:18	2 faulty 2 y.o., 5 faulty 3 y.o.
P10-P90	9	110	26:20	3 faulty 2 y.o., 5 faulty 3 y.o., 1 faulty 6 y.o.
P01-P99	13	110	38:04	7 faulty 2 y.o., 6 faulty 3 y.o.

As can be seen above, the current proof of concept is definitely not perfect. The single percentile value hands were less reliably generated the wider the percentile range was chosen. The failures occurred mostly in the hands for the youngest users (2 and 3 years old), only one faulty hand was generated outside of this group (6-year-old). The part of the hand that failed to be properly generated was the fillet between the palm and the thumb. The failed hand for a 6-year-old was the only failure caused by another error, the fillet surrounding the knuckle-hinge cutout.

Table 11 shows the results for the stress-tests where 5 variables per hand had deviations based on 5 different random percentile values. These tests were run similarly to the single percentile value based tests above, except these tests were run on a different computer.

Table 11: Results stress-test. 5 percentile value deviations for every hand, male, 2-12 years old

P range	**Faulty hands**	**Total hands**	**Runtime [mm:ss]**	**comment**
P49-P51	13	110	13:52	3 faulty 2 y.o., 1 faulty 3 y.o., 1 faulty 6 y.o., 4 faulty 11 y.o., 4 faulty 12 y.o.
P40-P60	2	110	08:15	1 faulty 10 y.o., 1 faulty 12 y.o.
P30-P70	1	110	07:28	1 faulty 12 y.o.
P20-P80	1	110	07:00	1 faulty 10 y.o.
P10-P90	0	110	07:14	
P01-P99	0	110	07:13	

A significant number of failures occurred in the hands generated using five different random percentile values. The hand generation process seems to become more reliable when wider percentile ranges for the deviation are chosen. This is opposite of what seems to occur with the single random percentile generated hands. The failures seemed to occur more in the hands for older users, 12 out of 17 failures occurred in hands for users older than 10 years old. Most failures were caused by the fillet between the palm and thumb, sometimes in combination with a failure to generate the fillet surrounding the knuckle-hinge cutout.

Discussion

The proof of concept definitely has room for improvement. Every failed hand was caused by Solidworks not being able to generate one of two fillets, or both. Specifically, the fillet between the palm and thumb and the fillet surrounding the knuckle-hinge cutout. The Solidworks model could be tweaked to be more reliable or a different modeling method could be chosen to generate the problematic fillets.

The main shortcoming of this small stress test experiment is the inconsistency in results. The stress test only generates a small sample size, 110 hands per percentile range. When tweaking the 3D hand model and working on the GUI, the same test runs would sometimes give significantly different results, because of the randomly selected deviations. A large sample size could counter these inconsistencies and give more confidence in the results, but would require more time and effort. Ultimately, this was a quick and dirty test to see what the current state of the hand model was.

The results also seemed to vary based on what computer and what version of Solidworks was used. Because of the small sample sizes and inconsistent results, it’s hard to judge whether this is really occurring or is just a coincidence.

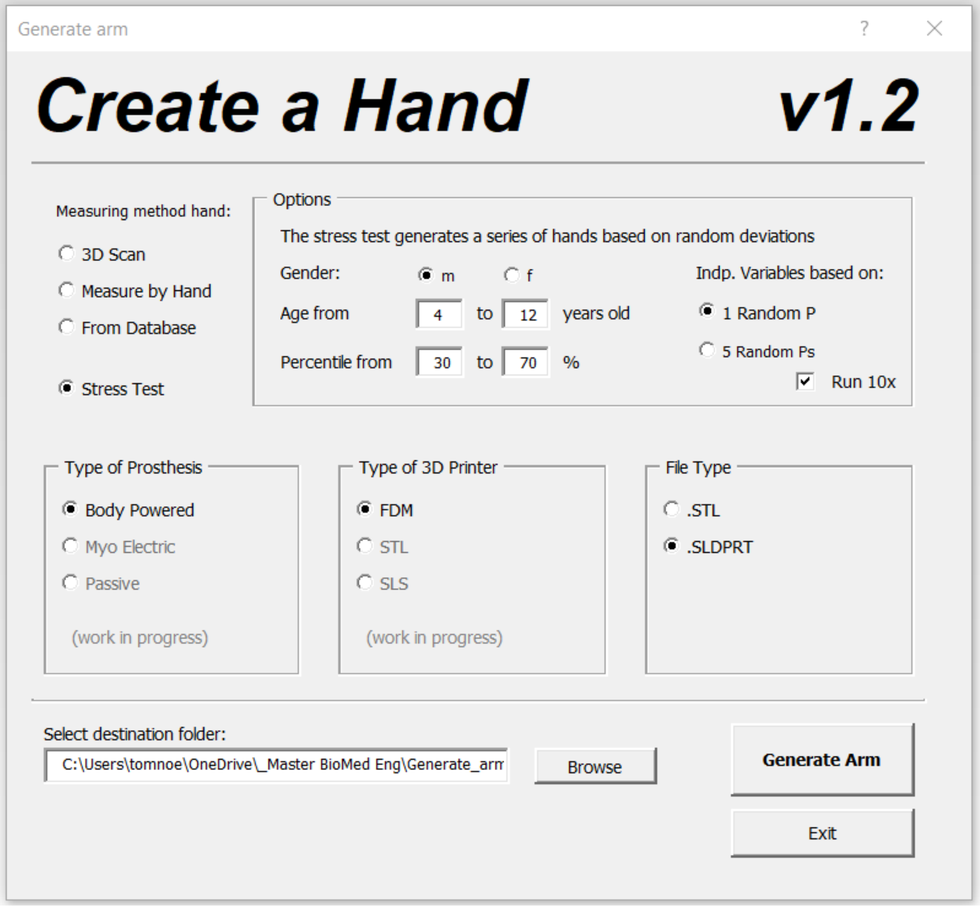
Another interesting trend is that the 5 random percentile based hand generation process seems to become more reliable the wider deviation range is chosen. The 1 random percentile based hand generation is the opposite, the narrower the range of deviations, the fewer errors occur.

Because of the inconsistencies, it is hard to draw any strong conclusions from this test. Though it can be said that the hand model and GUI are able to generate hands for the majority of the population. Even when accommodating for P01-P99 of the population most hands were generated without errors.

Conclusion

Overall, custom generating 3D CAD (Computer Aided Design) hand models for individual users seems doable. The proof of concept has room for improvement, but was able to generate hands for the majority of users. This was the goal behind the making of a proof of concept. A more robust hand model, subjected to more rigorous testing, should be made for a future consumer-ready product.

The stress test GUI/Program



Measuring method hand:
○ 3D Scan
○ Measure by Hand
○ From Database
● Stress Test

Options
The stress test generates a series of hands based on random deviations
Gender: ● m ○ f
Age from 4 to 12 years old
Percentile from 30 to 70 %
Indp. Variables based on:
● 1 Random P
○ 5 Random Ps
☑ Run 10x

Type of Prosthesis
● Body Powered
○ Myo Electric
○ Passive
(work in progress)

Type of 3D Printer
● FDM
○ STL
○ SLS
(work in progress)

File Type
○ .STL
● .SLDPRT

Select destination folder:
C:\Users\tomnoe\OneDrive_Master BioMed Eng\Generate_arm

Figure 40: the available options when the “stress test” input method is selected

Interface

Figure 40 above shows the GUI (Graphical User Interface) when the “Stress Test” input method is selected. In this mode, a set of hands is generated using random hand measurements within a specified population percentile range. The variables that are used to generate the hand can be randomized individually (“5 random Ps”), or randomized as a somewhat proportional group (“1 random P”). Finally, there is an option present to run this hand generation process 10 times in order to increase the sample size.

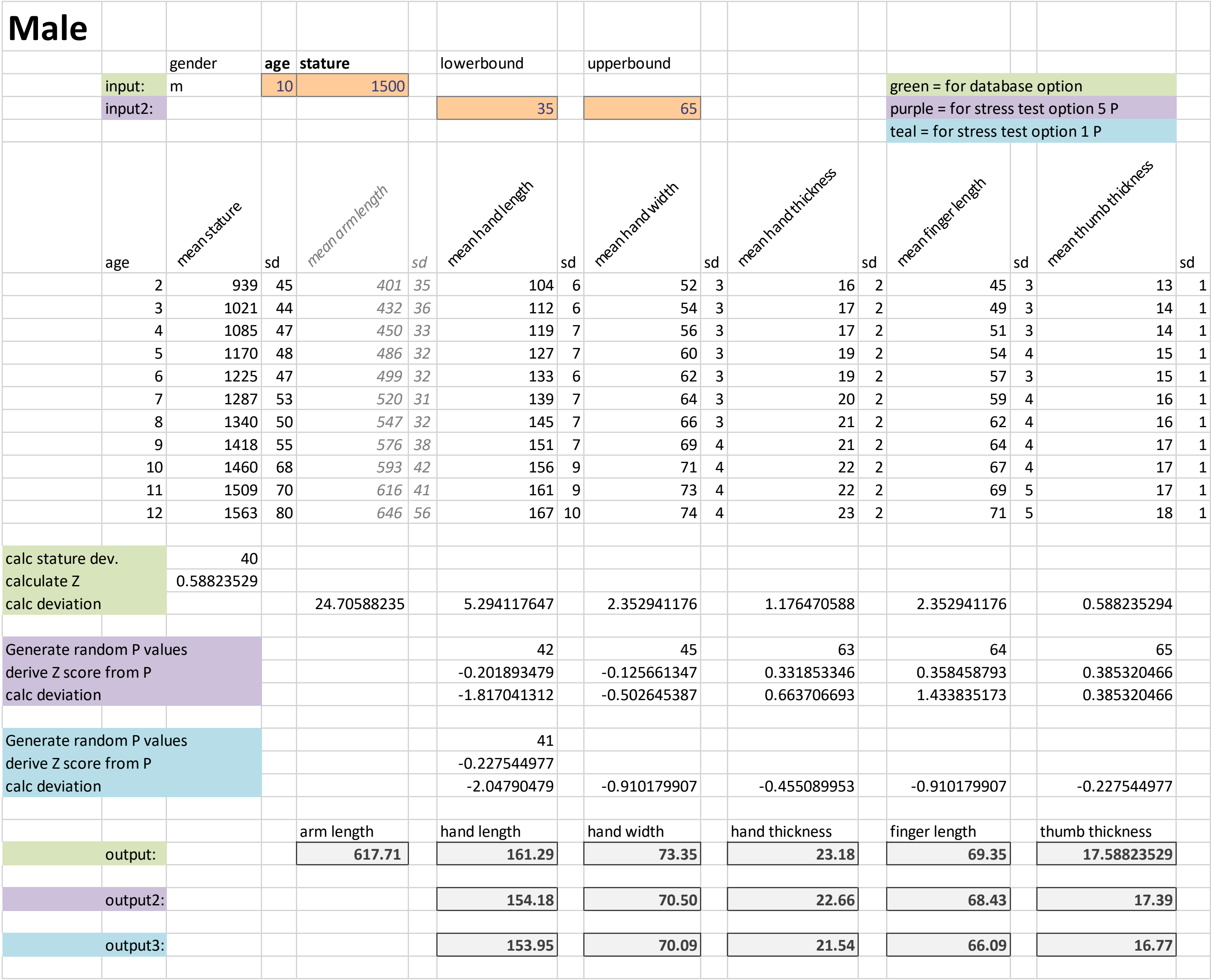
The parts of the GUI that are relevant to the stress test are included in the normal use GUI. The excel sheets and code needed to run the stress test are documented on the following pages. The code relevant to the “stress test” is also included in the complete code for the GUI in general which can be found in Appendix G.2. The Solidworks macro can be accessed through the digital resource pack (see Appendix F.). Opening the macro in Solidworks makes it easier to play around in the GUI and read the code behind it.

Excel sheets

The excel sheets are exactly the same as the excel sheets used for the “from database” input method (dined.xlsx). These excel sheets can be found earlier in Appendix D.2. on pages 49 & 50. The lines relevant to the stress test are highlighted in purple and teal.

When the “1 random P” option is selected in the GUI the teal lines in the excel sheet are used. Only one random P is generated within the specified percentile range. The resulting z-score is then applied to all 5 variables.

When the “5 random Ps” option is selected in the GUI the purple lines in the excel sheets are used. 5 random P’s are generated, a different one for each variable. From this, the 5 output variables are calculated which are then used to generate a 3D CAD hand. Figure 41 shows the excel sheet for males.



Male															
		gender	**age**	**stature**		lowerbound		upperbound							
	input:	m	10	1500								green = for database option			
	input2:					35		65				purple = for stress test option 5 P			
												teal = for stress test option 1 P			
	age	mean stature	sd	*mean arm length*	*sd*	mean hand length	sd	mean hand width	sd	mean hand thickness	sd	mean finger length	sd	mean thumb thickness	sd
	2	939	45	*401*	*35*	104	6	52	3	16	2	45	3	13	1
	3	1021	44	*432*	*36*	112	6	54	3	17	2	49	3	14	1
	4	1085	47	*450*	*33*	119	7	56	3	17	2	51	3	14	1
	5	1170	48	*486*	*32*	127	7	60	3	19	2	54	4	15	1
	6	1225	47	*499*	*32*	133	6	62	3	19	2	57	3	15	1
	7	1287	53	*520*	*31*	139	7	64	3	20	2	59	4	16	1
	8	1340	50	*547*	*32*	145	7	66	3	21	2	62	4	16	1
	9	1418	55	*576*	*38*	151	7	69	4	21	2	64	4	17	1
	10	1460	68	*593*	*42*	156	9	71	4	22	2	67	4	17	1
	11	1509	70	*616*	*41*	161	9	73	4	22	2	69	5	17	1
	12	1563	80	*646*	*56*	167	10	74	4	23	2	71	5	18	1
calc stature dev.		40													
calculate Z		0.58823529													
calc deviation				24.70588235		5.294117647		2.352941176		1.176470588		2.352941176		0.588235294	
Generate random P values						42		45		63		64		65	
derive Z score from P						-0.201893479		-0.125661347		0.331853346		0.358458793		0.385320466	
calc deviation						-1.817041312		-0.502645387		0.663706693		1.433835173		0.385320466	
Generate random P values						41									
derive Z score from P						-0.227544977									
calc deviation						-2.04790479		-0.910179907		-0.455089953		-0.910179907		-0.227544977	
				arm length		hand length		hand width		hand thickness		finger length		thumb thickness	
	output:			**617.71**		**161.29**		**73.35**		**23.18**		**69.35**		**17.58823529**	
	output2:					**154.18**		**70.50**		**22.66**		**68.43**		**17.39**	
	output3:					**153.95**		**70.09**		**21.54**		**66.09**		**16.77**	

Figure 41: dined.xlsx sheet for males, shows how a specified percentile range is transformed in a set of random hand “measurements”. These “measurements” are used to stress test the 3D CAD hand model.

Code

The code for the stress test is part of the overall GUI program. The complete code, including the parts relevant to the scalability stress test, can be found towards the end of this Appendix in section G.2. Two sections of code particularly relevant to the stress test are highlighted here.

When the “generate arm” button is pressed the following section of code is used if the *stresstest* radio button is selected.

```
' Stress test, use random values to generate arm
    Dim startTime As Double
    Dim MinutesElapsed As String
    startTime = Timer

    If optTest.Value = True Then
        If chk10x = True Then     'run stresstest 10 times if 10x checkbox is clicked
in GUI
            Dim k
            For k = 0 To 9
                Call GenStressTest
            Next k
        Else                      'run stress test once
            Call GenStressTest
        End If

        MinutesElapsed = Format((Timer - startTime) / 86400, "hh:mm:ss")
        MsgBox "stresstest completed! " & vbNewLine & totErrorCounter & " errors in " _
        & faultyHandsCounter & " hands" & vbNewLine & "Runtime: " & MinutesElapsed _
        & " hh:mm:ss" & vbNewLine & "faulty hands" & faultyHands

    End If
```

The code calls *GenStressTest* which generates a randomized hand for every age within the specified age-range. It does so using existing parts of the code found in section G.2. of the Appendix (*GetDimensionsFromExcel*, *ConvertDimensions*, *OpenSetExportClose*).

```
Sub GenStressTest()

    Dim AgeRange As Integer
    Dim dimensions() As Double
    AgeRange = txtAgeMax.Value - txtAgeMin.Value
    ReDim dimensions(AgeRange, 7)
    Dim i As Integer

    For i = 0 To AgeRange
        GenAge = txtAgeMin.Value + i
        Call GetDimensionsFromExcel
        Call ConvertDimensions

        dimensions(i, 0) = FingerLength
```

```
        dimensions(i, 1) = PalmLength
        dimensions(i, 2) = FingerThickness
        dimensions(i, 3) = ThumbThickness
        dimensions(i, 4) = WristWidth
        dimensions(i, 5) = PalmWidth
        dimensions(i, 6) = ThumbLength

        Call OpenSetExportClose
    Next i

End Sub
```

3. Function

The functionality of the prosthesis was only tested very superficially. Attention was focused on other aspects of the prosthesis (scalability and minimal assembly construction) as the design is still somewhat in its early stages. The testing in this section is intended to roughly measure two of the performance properties of the hand, the required actuation force and the tensile strength of the hinges.

Weight

The prototype weighs 99 grams, measured using a kitchen scale. This weight includes the entire hand except the mounting bolt. The printed prototype hand was sized for a 12-year-old male user. The weight of the hand could be farther reduced by choosing different 3D printing settings, such as a lower infill (prototype used 90% infill).

Actuation force

The amount of force required to completely close the hand was measured using a plastic bottle filled with water. The hand was positioned vertically with the actuation wire exiting from the bottom. An empty bottle was tied to this wire and filled with water until the hand was fully closed (hand can't flex any further). The bottle with water was then measured using a kitchen scale.

Table 12: Measurement of actuation force

Bottle weight (grams)	*Actuation force (Newton)*
1440 g	14.1 N

The pinch force in relation to the actuation force was not measured. The prototype prosthetic hand exerted very little pinch force, this is an element of the design on which future developments could definitely improve on.

Tensile strength hinges

An extra *finger set* was used to test the tensile strength of a 1mm thick 3D printed living hinge. The *finger set* was positioned vertically with the fingers pointing down, with a weight connected to the tip of the finger. Initially, the testing weight was suspended using the existing *connection loop* used for the actuation of the finger. At higher forces this connection loop tore as it was only designed to handle actuation forces. From then on, locking pliers were used to attach the test weight to the tip of the finger. The test weight initially consisted of a 10-liter container which was incrementally filled with water to increase its mass. Later on, the test mass was switched to a backpack incrementally filled with more weights. Roughly 0.5 kg was added each time until the hinge failed. The locking pliers, connection strings, backpack and its contents were weighed afterwards.

Table 13: Measurement of breaking strength

Ultimate weight of test mass (grams)	*Approx. breaking force (Newton)*
16011 g	157 N

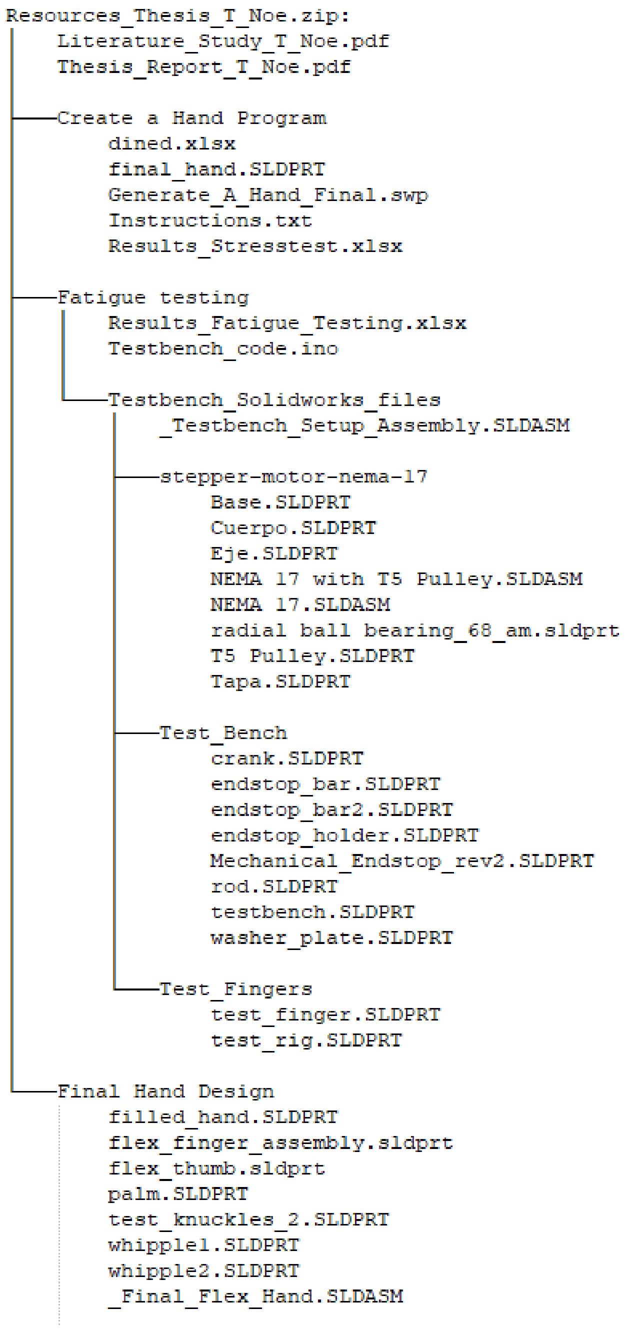
F. Digital Resources Thesis

Due to the nature of a lot of the work performed for this thesis, several important digital files were created. Referring to these digital files in the appendix or the main report isn’t always useful or practical. For instance, interacting with the functional Graphical User Interface is not possible on paper. For this reason, a digital pack of resources in the form of a .zip file was created to accompany this thesis.

The *Resources_Thesis_T_Noe.zip* archive can be downloaded from:

https://www.dropbox.com/sh/b7ubk9vtkt9iy41/AAAaK-1n2u7JHVOztCHC3jBca?dl=0

Figure 42 shows the files and file structure within *Resources_Thesis_T_Noe.zip*.



```
Resources_Thesis_T_Noe.zip:
│   Literature_Study_T_Noe.pdf
│   Thesis_Report_T_Noe.pdf
│
├───Create a Hand Program
│       dined.xlsx
│       final_hand.SLDPRT
│       Generate_A_Hand_Final.swp
│       Instructions.txt
│       Results_Stresstest.xlsx
│
├───Fatigue testing
│   │   Results_Fatigue_Testing.xlsx
│   │   Testbench_code.ino
│   │
│   └───Testbench_Solidworks_files
│       │   _Testbench_Setup_Assembly.SLDASM
│       │
│       ├───stepper-motor-nema-17
│       │       Base.SLDPRT
│       │       Cuerpo.SLDPRT
│       │       Eje.SLDPRT
│       │       NEMA 17 with T5 Pulley.SLDASM
│       │       NEMA 17.SLDASM
│       │       radial ball bearing_68_am.sldprt
│       │       T5 Pulley.SLDPRT
│       │       Tapa.SLDPRT
│       │
│       ├───Test_Bench
│       │       crank.SLDPRT
│       │       endstop_bar.SLDPRT
│       │       endstop_bar2.SLDPRT
│       │       endstop_holder.SLDPRT
│       │       Mechanical_Endstop_rev2.SLDPRT
│       │       rod.SLDPRT
│       │       testbench.SLDPRT
│       │       washer_plate.SLDPRT
│       │
│       └───Test_Fingers
│               test_finger.SLDPRT
│               test_rig.SLDPRT
│
└───Final Hand Design
        filled_hand.SLDPRT
        flex_finger_assembly.sldprt
        flex_thumb.sldprt
        palm.SLDPRT
        test_knuckles_2.SLDPRT
        whipple1.SLDPRT
        whipple2.SLDPRT
        _Final_Flex_Hand.SLDASM
```

Figure 42: File structure of Resources_Thesis_T_Noe.zip

G. Code

1. Software code used for fatigue testing

This section documents the Arduino code used for the fatigue testing. The code itself can be accessed through the digital resource pack, as described in Appendix F. The code itself could be cleaned up but was sufficient for the fatigue testing.

```
// Tom Noe    4091744
// TU Delft   2018

//******* Hardware ********//
 // minitronics v1.0
 // NEMA 17 stepper motors
 // left stepper motor on XMOT
 // right stepper motor on YMOT
 // finger 1 microswitch on AUX1 pins 5 & 7
 // fingers 2,3,4 microswitches on X,Y,Z
//*************************//

 #define leftStepPin 48              // Left stepper motor pins
 #define leftDirPin 47
 #define leftEnablePin 49

 #define rightStepPin 39 // A6       // Right stepper motor pins
 #define rightDirPin 40 // A0
 #define rightEnablePin 38

 #define f1MinPin 19                 // finger 1 microswitch pin
 #define f2MinPin 5                  // finger 2 microswitch pin
 #define f3MinPin 2                  // finger 3 microswitch pin
 #define f4MinPin 6                  // finger 4 microswitch pin


 #define LEDPin 46                   // LED pin



void setup() {

  pinMode(f1MinPin  , INPUT_PULLUP);
  pinMode(f2MinPin  , INPUT_PULLUP);
  pinMode(f3MinPin  , INPUT_PULLUP);
  pinMode(f4MinPin  , INPUT_PULLUP);

  pinMode(LEDPin  , OUTPUT);

  pinMode(leftStepPin  , OUTPUT);
  pinMode(leftDirPin    , OUTPUT);
  pinMode(leftEnablePin    , OUTPUT);

  pinMode(rightStepPin  , OUTPUT);
  pinMode(rightDirPin    , OUTPUT);
  pinMode(rightEnablePin    , OUTPUT);


  digitalWrite(leftEnablePin    , LOW);
  digitalWrite(rightEnablePin    , LOW);


  Serial.begin(115200);

  Serial.println("TESTING...");

;
```

```
}

//**************** stepper motor settings ********************//
  int steps = 200;
  int microstepping = 32;
  int stepsPerRev = steps * microstepping;

//*********** intialize variables ************//
long stepCounter = 1;

int prevTriggerTime = 0;
int prevTSLT = 0;
int triggerTime = 0;
int timeSinceLastTrigger = 0;

long cycles[] = {0, 0, 0, 0};
int lastSwitchState[] = {0, 0, 0, 0};

int pause = 0;



void loop () {

//********* blink LED while running ***********//
  if (millis() %1000 <500)
   digitalWrite(LEDPin, HIGH);
  else
   digitalWrite(LEDPin, LOW);


//*************** run testbench ******************//

  digitalWrite(leftDirPin    , HIGH); // direction left stepper motor
  digitalWrite(rightDirPin    , LOW); // direction right stepper motor (opposite)

  int triggerTime = 0;
  int timeSinceLastTrigger = 0;

  triggerTime = prevTriggerTime;
  timeSinceLastTrigger = prevTSLT;

   //**** one step ****//
   // single step is repeated multiple times per loop to increase testbench speed
   // could be cleaned up...
   digitalWrite(leftStepPin,HIGH);
   digitalWrite(rightStepPin,HIGH);

   digitalWrite(leftStepPin,LOW);
   digitalWrite(rightStepPin,LOW);
   stepCounter++;

      //**** one step ****//
   digitalWrite(leftStepPin,HIGH);
   digitalWrite(rightStepPin,HIGH);

   digitalWrite(leftStepPin,LOW);
   digitalWrite(rightStepPin,LOW);
   stepCounter++;

      //**** one step ****//
   digitalWrite(leftStepPin,HIGH);
   digitalWrite(rightStepPin,HIGH);

   digitalWrite(leftStepPin,LOW);
   digitalWrite(rightStepPin,LOW);
   stepCounter++;

      //**** one step ****//
```

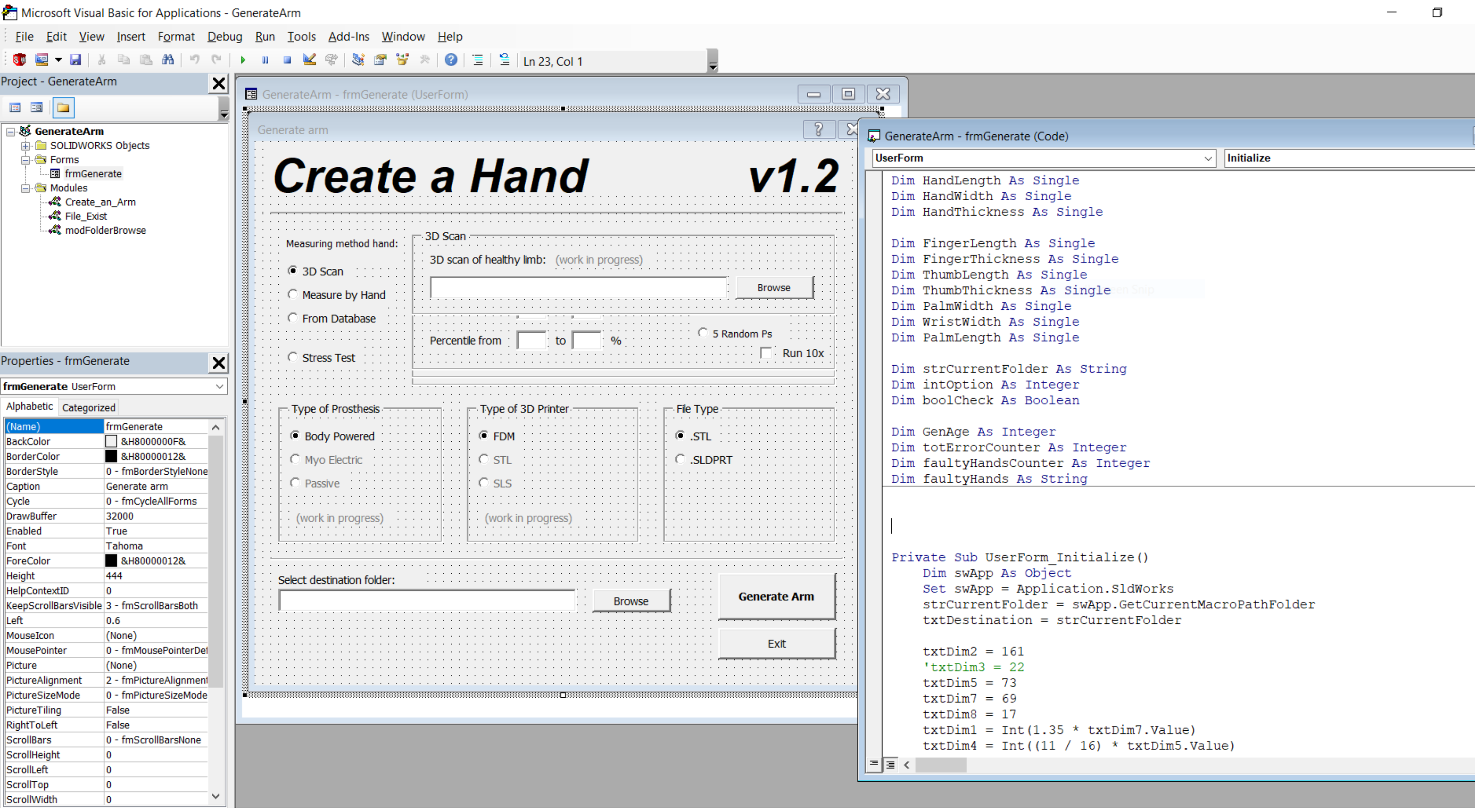
```
   digitalWrite(leftStepPin,HIGH);
   digitalWrite(rightStepPin,HIGH);

   digitalWrite(leftStepPin,LOW);
   digitalWrite(rightStepPin,LOW);
   stepCounter++;

      //**** one step ****//
   digitalWrite(leftStepPin,HIGH);
   digitalWrite(rightStepPin,HIGH);

   digitalWrite(leftStepPin,LOW);
   digitalWrite(rightStepPin,LOW);
   stepCounter++;

   long totalRev = stepCounter/stepsPerRev;



   //********* state change detection ***********//
   int switchState[]={digitalRead(f1MinPin), digitalRead(f2MinPin), digitalRead(f3MinPin),
digitalRead(f4MinPin)};

   for (int i=0; i<4; i++) {
     if (switchState[i] != lastSwitchState[i]) {

      if (switchState[i] == LOW){
        cycles[i]++;                                    // count number of cycles for each finger

        Serial.print(" finger1: ");                     // display number of cycles in serial
monitor
        Serial.print(cycles[0]);
        Serial.print("    finger2: ");
        Serial.print(cycles[1]);
        Serial.print("    finger3: ");
        Serial.print(cycles[2]);
        Serial.print("    finger4: ");
        Serial.print(cycles[3]);
        Serial.print("    ||      total revs: ");
        Serial.println(totalRev);

        triggerTime = millis();                         // record time of last trigger
        prevTriggerTime = triggerTime;
      }

      delay(35);                                        // delay to prevent bouncing of
microswitch
     }

    lastSwitchState[i] = switchState[i];

    timeSinceLastTrigger = millis() - triggerTime;      // calculate time since last trigger
    prevTSLT = timeSinceLastTrigger;
   }

//******** pause if all fingers are broken ********//
  while(timeSinceLastTrigger > 10000){      // pause if none of the switches have been triggered
in 10 seconds

    //do nothing


    if (pause == 0){                       // display pause message only once
      Serial.println();
      Serial.println(" ********** TESTING STOPPED **********");
      Serial.print(" total stepper revolutions: ");
      Serial.println(totalRev);
      Serial.println();
      Serial.println(" total revolutions measured by the microswitches: ");
```

```
      Serial.print(" finger1: ");
      Serial.print(cycles[0]);
      Serial.print("    finger2: ");
      Serial.print(cycles[1]);
      Serial.print("    finger3: ");
      Serial.print(cycles[2]);
      Serial.print("    finger4: ");
      Serial.println(cycles[3]);
      Serial.println(" **********                **********");
    }
    pause = 1;

    if (digitalRead(f1MinPin) == LOW ||
       digitalRead(f2MinPin) == LOW ||
       digitalRead(f3MinPin) == LOW ||
       digitalRead(f4MinPin) == LOW){     // resume cycles if any of the switches is re-triggered
       triggerTime = millis();
       timeSinceLastTrigger = millis() - triggerTime;
       prevTriggerTime = triggerTime;
       prevTSLT = timeSinceLastTrigger;

       pause = 0;
    }

  }

}
```

2. Code GUI

This section documents the code behind the GUI. The GUI was made in VBA (Visual Basic for Applications) and uses the standard *Visual Basic* coding language. Besides the code written by the author, two modules created by outside sources were used. The first module, *ModBrowseFile*, creates the normal Windows popup window that allows a user to select the destination folder (Elliott, 2014). The second module, *FileExists*, checks whether a file already exists in the desired destination folder (Newman, 2014).

The image below (Figure 43) shows the development environment in VBA. The *project* window in the top left shows the from and modules that make up the hand generation GUI. The window in the middle shows the “UserForm”, the part of the interface the user will see and use. The window on the right shows the code behind the form.



Properties - frmGenerate

frmGenerate UserForm

Alphabetic | Categorized

(Name)	frmGenerate
BackColor	☐ &H8000000F&
BorderColor	■ &H80000012&
BorderStyle	0 - fmBorderStyleNone
Caption	Generate arm
Cycle	0 - fmCycleAllForms
DrawBuffer	32000
Enabled	True
Font	Tahoma
ForeColor	■ &H80000012&
Height	444
HelpContextID	0
KeepScrollBarsVisible	3 - fmScrollBarsBoth
Left	0.6
MouseIcon	(None)
MousePointer	0 - fmMousePointerDel
Picture	(None)
PictureAlignment	2 - fmPictureAlignment
PictureSizeMode	0 - fmPictureSizeMode
PictureTiling	False
RightToLeft	False
ScrollBars	0 - fmScrollBarsNone
ScrollHeight	0
ScrollLeft	0
ScrollTop	0
ScrollWidth	0

```
Dim HandLength As Single
Dim HandWidth As Single
Dim HandThickness As Single

Dim FingerLength As Single
Dim FingerThickness As Single
Dim ThumbLength As Single
Dim ThumbThickness As Single
Dim PalmWidth As Single
Dim WristWidth As Single
Dim PalmLength As Single

Dim strCurrentFolder As String
Dim intOption As Integer
Dim boolCheck As Boolean

Dim GenAge As Integer
Dim totErrorCounter As Integer
Dim faultyHandsCounter As Integer
Dim faultyHands As String



Private Sub UserForm_Initialize()
    Dim swApp As Object
    Set swApp = Application.SldWorks
    strCurrentFolder = swApp.GetCurrentMacroPathFolder
    txtDestination = strCurrentFolder

    txtDim2 = 161
    'txtDim3 = 22
    txtDim5 = 73
    txtDim7 = 69
    txtDim8 = 17
    txtDim1 = Int(1.35 * txtDim7.Value)
    txtDim4 = Int((11 / 16) * txtDim5.Value)
```

Figure 43: Visual Basic for Applications development environment. Gererate_A_Hand_Final.swp

The code behind the form and the three modules are documented below. The code could be condensed considerably and could be made to be easier to review by splitting up the program into multiple modules/functions. This program was written as part of a proof of concept and as such is a bit messy and long.

Forms

frmGenerate

```
Dim HandLength As Single
Dim HandWidth As Single
Dim HandThickness As Single

Dim FingerLength As Single
Dim FingerThickness As Single
Dim ThumbLength As Single
Dim ThumbThickness As Single
Dim PalmWidth As Single
Dim WristWidth As Single
Dim PalmLength As Single

Dim strCurrentFolder As String
Dim intOption As Integer
Dim boolCheck As Boolean

Dim GenAge As Integer
Dim totErrorCounter As Integer
Dim faultyHandsCounter As Integer
Dim faultyHands As String




Private Sub UserForm_Initialize()
    Dim swApp As Object
    Set swApp = Application.SldWorks
    strCurrentFolder = swApp.GetCurrentMacroPathFolder
    txtDestination = strCurrentFolder

    txtDim2 = 161
    'txtDim3 = 22
    txtDim5 = 73
    txtDim7 = 69
    txtDim8 = 17
    txtDim1 = Int(1.35 * txtDim7.Value)
    txtDim4 = Int((11 / 16) * txtDim5.Value)
    txtDim6 = Int(0.9 * txtDim8)

    txtAgeMin = 4
    txtAgeMax = 12
    txtPMin = 30
    txtPMax = 70

    txtAge = 6
    txtHeight = 1230


    frmScan.Visible = False
    frmMeasure.Visible = True
    frmData.Visible = False
```

```
    frmTest.Visible = False
    optMeasure.Value = True

    optSLDPRT.Value = True

    boolCheck = True

    totErrorCounter = 0
    faultyHandsCounter = 0



End Sub


Private Sub cmdBrowseResidual_Click()
    MsgBox "work in progress"
End Sub

Private Sub cmdBrowseHealthy_Click()
    MsgBox "work in progress"
End Sub

Private Sub cmdBrowseDestination_Click()
   Dim sPath As String
'                          FolderBrowse(DialogBoxTitle, StartingPath)
   sPath = modFolderBrowse.FolderBrowse("Select Folder Containing Block Library", "")
   If sPath <> "" Then
      If modFolderBrowse.FolderExists(sPath) Then
         txtDestination.Text = sPath
      End If
   End If
End Sub



Private Sub opt3Dscan_Click()

    frmScan.Visible = True
    frmMeasure.Visible = False
    frmData.Visible = False
    frmTest.Visible = False

End Sub


Private Sub optMeasure_Click()
    frmScan.Visible = False
    frmMeasure.Visible = True
    frmData.Visible = False
    frmTest.Visible = False

End Sub

Private Sub optDatabase_Click()
```

```
    frmScan.Visible = False
    frmMeasure.Visible = False
    frmData.Visible = True
    frmTest.Visible = False

End Sub

Private Sub optTest_Click()

    frmScan.Visible = False
    frmMeasure.Visible = False
    frmData.Visible = False
    frmTest.Visible = True

End Sub



Sub CheckForErrors()
Dim age As String
Dim height As String



    If optDatabase.Value = True Then
        age = txtAge.Text
        height = txtHeight.Text
        If age = Empty Then
            MsgBox "Age is empty"
            boolCheck = False
        ElseIf Not IsNumeric(age) Then
            MsgBox "Age is not numeric"
            boolCheck = False
        ElseIf age < 2 Or age > 12 Then
            MsgBox "Fill in the age of the user. (2-12 years old)"
            boolCheck = False

        ElseIf height = Empty Then
            MsgBox "Height is empty"
            boolCheck = False
        ElseIf Not IsNumeric(height) Then
            MsgBox "Height is not numeric"
            boolCheck = False
        ElseIf height < 750 Or height > 1750 Then
            MsgBox "Fill in the height of the user in millimeters. (750-1750 mm)"
            boolCheck = False
        Else
            boolCheck = True

        End If

    End If
```

```
    If optMeasure.Value = True Then
        age = txtAge.Text
        If txtDim1 = Empty Or txtDim1 = Empty Or txtDim2 = Empty _
        Or txtDim4 = Empty Or txtDim5 = Empty Or txtDim6 = Empty Or txtDim7 = Empty _
        Then
            MsgBox "Fill in all measurements"
            boolCheck = False
        ElseIf Not IsNumeric(age) Then
            MsgBox "One or more measurements are not numeric"
            boolCheck = False
        Else
            boolCheck = True
        End If
    End If


Dim age1 As Integer
Dim age2 As Integer

    If optTest.Value = True Then
        age1 = txtAgeMin.Text
        age2 = txtAgeMax.Text
        PMin = txtPMin.Text
        PMax = txtPMax.Text

        If age1 = Empty Then
            MsgBox "Age is empty"
            boolCheck = False
        ElseIf Not IsNumeric(age1) Then
            MsgBox "Age is not numeric"
            boolCheck = False
        ElseIf age1 < 2 Or age1 > 12 Then
            MsgBox "Fill in the age of the user. (2-12 years old)"
            boolCheck = False


        ElseIf age1 = Empty Then
            MsgBox "Age is empty"
            boolCheck = False
        ElseIf Not IsNumeric(age2) Then
            MsgBox "Age is not numeric"
            boolCheck = False
        ElseIf age2 < 2 Or age2 > 12 Then
            MsgBox "Fill in the age of the user. (2-12 years old)"
            boolCheck = False
        ElseIf age1 > age2 Then
            MsgBox "Fill in the age from young to old"
            boolCheck = False


        ElseIf PMin = Empty Then
            MsgBox "Percentile is empty"
            boolCheck = False
        ElseIf Not IsNumeric(PMin) Then
```

```
            MsgBox "Percentile is not numeric"
            boolCheck = False
        ElseIf PMin < 0 Or PMin > 100 Then
            MsgBox "Fill in the desired percentiles. (0-100 %)"
            boolCheck = False


        ElseIf PMax = Empty Then
            MsgBox "Percentile is empty"
            boolCheck = False
        ElseIf Not IsNumeric(PMax) Then
            MsgBox "Percentile is not numeric"
            boolCheck = False
        ElseIf PMax < 0 Or PMax > 100 Then
            MsgBox "Fill in the desired percentiles. (0-100 %)"
            boolCheck = False
        ElseIf PMin > PMax Then
            MsgBox "Fill in the percentiles from small to large"
            boolCheck = False

        Else
            boolCheck = True
        End If

    End If



End Sub

Sub GetDimensionsFromExcel()
    Dim xlApp As Excel.Application
    Dim xlWB As Excel.Workbook
    Set xlApp = CreateObject("Excel.Application")
    Set xlWB = xlApp.Workbooks.Open(strCurrentFolder & "\dined.xlsx")


    If optGenderM.Value Or optGenderM2.Value = True Then intOption = 1
    If optGenderF.Value Or optGenderF2.Value = True Then intOption = 2
    Dim Row As Integer

    If optDatabase.Value = True Then
        Row = 32
        With xlWB.Worksheets(intOption)
            .range("D3").Value = txtAge.Value
            .range("E3").Value = txtHeight.Value

            HandLength = .range("G" & Row).Value
            HandWidth = .range("I" & Row).Value
            HandThickness = .range("K" & Row).Value
            FingerLength = .range("M" & Row).Value
            ThumbThickness = .range("O" & Row).Value
        End With
    End If
```

```
    If optTest.Value = True Then
        If opt1P = True Then Row = 36
        If opt5P = True Then Row = 34

        With xlWB.Worksheets(intOption)
            .range("D3").Value = GenAge
            .range("G4").Value = txtPMin.Value
            .range("I4").Value = txtPMax.Value

            HandLength = .range("G" & Row).Value
            HandWidth = .range("I" & Row).Value
            HandThickness = .range("K" & Row).Value
            FingerLength = .range("M" & Row).Value
            ThumbThickness = .range("O" & Row).Value
        End With
    End If

    xlWB.Close False
    xlApp.Quit
    Set xlApp = Nothing
    Set xlWB = Nothing

End Sub



Sub ConvertDimensions()

    If optDatabase.Value Or optTest.Value = True Then
        ThumbLength = 1.35 * FingerLength
    Else
        ThumbLength = 0.85 * FingerLength
    End If


    WristWidth = (11 / 16) * HandWidth
    PalmLength = HandLength - FingerLength
    FingerThickness = ThumbThickness * 0.9
    'FingerThickness = HandThickness * 0.73

End Sub



Private Sub cmdDinedData_Click()
    'check inputs
    Call CheckForErrors
    If boolCheck = False Then Exit Sub

    Call GetDimensionsFromExcel
    Call ConvertDimensions

        lbl1.Caption = Int(ThumbLength) & " mm"
```

```
        lbl2.Caption = Int(PalmLength) & " mm"
        lbl4.Caption = Int(WristWidth) & " mm"
        lbl5.Caption = Int(HandWidth) & " mm"
        lbl6.Caption = Int(FingerThickness) & " mm"
        lbl7.Caption = Int(FingerLength) & " mm"
        lbl8.Caption = Int(ThumbThickness) & " mm"


End Sub

Sub GenStressTest()

    Dim AgeRange As Integer
    Dim dimensions() As Double
    AgeRange = txtAgeMax.Value - txtAgeMin.Value
    ReDim dimensions(AgeRange, 7)
    Dim i As Integer

    For i = 0 To AgeRange
        GenAge = txtAgeMin.Value + i
        Call GetDimensionsFromExcel
        Call ConvertDimensions

        dimensions(i, 0) = FingerLength
        dimensions(i, 1) = PalmLength
        dimensions(i, 2) = FingerThickness
        dimensions(i, 3) = ThumbThickness
        dimensions(i, 4) = WristWidth
        dimensions(i, 5) = PalmWidth
        dimensions(i, 6) = ThumbLength

        Call OpenSetExportClose
    Next i

End Sub



Sub OpenSetExportClose()

    ' %%%%%%%% open doc %%%%%%%%%%%%%%
    Dim swApp As Object
    Set swApp = Application.SldWorks
    Dim Part As SldWorks.ModelDoc2
    Dim swModelDocExt As SldWorks.ModelDocExtension
    Dim swEquationMgr   As SldWorks.EquationMgr
    Dim boolstatus As Boolean
    Dim longstatus As Long
    Dim longwarnings As Long
    Dim numWrong As Integer

    Set Part = swApp.OpenDoc6(strCurrentFolder & "\final_hand.SLDPRT", 1, 1, "", longstatus,
longwarnings)
    swApp.ActivateDoc2 "final_hand.SLDPRT", False, longstatus
    Set Part = swApp.ActiveDoc
```

```
    Set swModelDocExt = Part.Extension
    Set swEquationMgr = Part.GetEquationMgr


    ' %%%%%%%% set dimensions %%%%%%%%
    Dim j As Integer
    Dim vSplit As Variant
    For j = 0 To swEquationMgr.GetCount - 1
        vSplit = Split(swEquationMgr.Equation(j), "=")
        vSplit(0) = Replace(vSplit(0), Chr(34), Empty)

        If vSplit(0) = "FingerLength" Then _
            swEquationMgr.Equation(j) = Replace(swEquationMgr.Equation(j), vSplit(1),
FingerLength)
        If vSplit(0) = "PalmLength" Then _
            swEquationMgr.Equation(j) = Replace(swEquationMgr.Equation(j), vSplit(1), PalmLength)
        If vSplit(0) = "FingerThickness" Then _
            swEquationMgr.Equation(j) = Replace(swEquationMgr.Equation(j), vSplit(1),
FingerThickness)
        If vSplit(0) = "ThumbThickness" Then _
            swEquationMgr.Equation(j) = Replace(swEquationMgr.Equation(j), vSplit(1),
ThumbThickness)
        If vSplit(0) = "WristWidth" Then _
            swEquationMgr.Equation(j) = Replace(swEquationMgr.Equation(j), vSplit(1), WristWidth)
        If vSplit(0) = "PalmWidth" Then _
            swEquationMgr.Equation(j) = Replace(swEquationMgr.Equation(j), vSplit(1), HandWidth)
        If vSplit(0) = "ThumbLength" Then _
            swEquationMgr.Equation(j) = Replace(swEquationMgr.Equation(j), vSplit(1),
ThumbLength)


    Next j


    ' %%%%%%%% export part %%%%%%%%%%%
    Dim FileExt As String
    Dim FileName As String
    Dim FilePath As String
    Dim strType
    Dim X As Integer
    Dim saved As Boolean


    If optSTL.Value = True Then FileExt = ".STL"
    If optSLDPRT.Value = True Then FileExt = ".SLDPRT"

    If optTest.Value = True Then
        FileName = "\stress_test_" & GenAge & "_yo"
    Else
        FileName = "\arm_export"
    End If


    saved = False
    X = 1
```

```
    FilePath = txtDestination.Text & FileName
    If FileExist(FilePath & FileExt) = False Then
        longstatus = Part.SaveAs3(FilePath & FileExt, 0, 0)
        saved = True
    End If

    Do While saved = False
        If FileExist(FilePath & "(" & X & ")" & FileExt) = False Then
            longstatus = Part.SaveAs3(FilePath & "(" & X & ")" & FileExt, 0, 0)
            saved = True
        Else
            X = X + 1
        End If
    Loop


    ' %%%%%% count errors and faulty hands %%%%%%
    numWrong = swModelDocExt.GetWhatsWrongCount

    totErrorCounter = totErrorCounter + numWrong
    If numWrong > 0 Then
        faultyHandsCounter = faultyHandsCounter + 1
        faultyHands = faultyHands & ", " & FileName & "(" & X & ")"
        'MsgBox "something wrong"
    End If


    ' %%%%%%%% close doc %%%%%%%%%%%%%
    'swApp.CloseDoc "arm_generate"
    boolstatus = swApp.CloseAllDocuments(True)

End Sub


Private Sub cmdGenerateArm_Click()
    ' %%% zero values previous run %%%
    totErrorCounter = 0
    faultyHandsCounter = 0
    faultyHands = ""

    ' %%% check inputs %%%
    Call CheckForErrors
    If boolCheck = False Then Exit Sub


    ' %%% Dimensioning methods %%%

    ' 3D Scanning (not functional)
    If opt3Dscan.Value = True Then
        MsgBox "3D scanning is still a work in progress"
        Exit Sub
    End If

    ' Measure manually
```

```
    If optMeasure.Value = True Then
        'retrieve inputs
        HandLength = txtDim2.Value
        ThumbLength = txtDim1.Value
        WristWidth = txtDim4.Value
        HandWidth = txtDim5.Value
        'HandThickness = txtDim3.Value
        FingerLength = txtDim7.Value
        FingerThickness = txtDim6.Value
        ThumbThickness = txtDim8.Value

        PalmLength = HandLength - FingerLength
        MsgBox "HL:" & HandLength & " FL:" & FingerLength & " PL:" & PalmLength
        Call OpenSetExportClose

        MsgBox "Hand generated succesfully!" & vbNewLine _
        & "With " & totErrorCounter & " errors" _
        '& vbNewLine & "in " & faultyHands
    End If


    ' Use Dined database
    If optDatabase.Value = True Then
        Call GetDimensionsFromExcel
        Call ConvertDimensions
        Call OpenSetExportClose

        MsgBox "Hand generated succesfully!" & vbNewLine & "With " & totErrorCounter & " errors"
    End If


    ' Stress test, use random values to generate arm
    Dim startTime As Double
    Dim MinutesElapsed As String
    startTime = Timer

    If optTest.Value = True Then
        If chk10x = True Then
            Dim k
            For k = 0 To 9
                Call GenStressTest
            Next k
        Else
            Call GenStressTest
        End If

        MinutesElapsed = Format((Timer - startTime) / 86400, "hh:mm:ss")
        MsgBox "stresstest completed! " & vbNewLine & totErrorCounter & " errors in " _
        & faultyHandsCounter & " hands" & vbNewLine & "Runtime: " & MinutesElapsed _
        & " hh:mm:ss" & vbNewLine & "faulty hands" & faultyHands

    End If


End Sub
```

```
Private Sub cmdExitButton_Click()

    ' %%% Close interface %%%
    End

End Sub
```

Modules

Create_an_Arm

```
Sub main()
frmGenerate.Show

End Sub
```

File_Exists

Chris Newman, https://www.thespreadsheetguru.com/blog/2014/8/19/save-as-a-new-version-if-file-already-exists

```
Function FileExist(FilePath As String) As Boolean
'PURPOSE: Test to see if a file exists or not
'SOURCE: www.TheSpreadsheetGuru.com/The-Code-Vault
'RESOURCE: http://www.rondebruin.nl/win/s9/win003.htm

Dim TestStr As String

'Test File Path (ie "C:\Users\Chris\Desktop\Test\book1.xlsm")
  On Error Resume Next
    TestStr = Dir(FilePath)
  On Error GoTo 0

'Determine if File exists
  If TestStr = "" Then
    FileExist = False
  Else
    FileExist = True
  End If

End Function
```

modFolderBrowse

Steven Elliott

https://www.theswamp.org/index.php?topic=47580.0

```
'#####################################################################################
'#                                                                                   #
'#                               modFolderBrowse                                     #
'#                              code compiled by                                     #
'#                               Steven Elliott                                      #
'#                         from many sources on the net                              #
'#                          Released to Public Domain                                #
'#                                                                                   #
'#####################################################################################

Option Explicit

Private Declare PtrSafe Function SHBrowseForFolder Lib "shell32.dll" _
  Alias "SHBrowseForFolderA" (lpBrowseInfo As BrowseInfo) As Long

Private Declare PtrSafe Function SHGetPathFromIDList Lib "shell32.dll" _
        (ByVal pidList As Long, ByVal lpBuffer As String) As Long

Public Declare PtrSafe Function SendMessageA Lib "user32" _
      (ByVal hWnd As LongPtr, ByVal wMsg As Long, _
       ByVal wParam As LongPtr, lParam As Any) As LongPtr

Private Declare PtrSafe Sub CoTaskMemFree Lib "ole32.dll" (ByVal hMem As Long)

Private Const BIF_RETURNONLYFSDIRS  As Long = 1
Private Const CSIDL_DRIVES          As Long = &H11
Private Const WM_USER               As Long = &H400
Private Const MAX_PATH              As Long = 260 ' Is it a bad thing that I memorized this
value?

'// message from browser
Private Const BFFM_INITIALIZED     As Long = 1
Private Const BFFM_SELCHANGED      As Long = 2
Private Const BFFM_VALIDATEFAILEDA As Long = 3 '// lParam:szPath ret:1(cont),0(EndDialog)
Private Const BFFM_VALIDATEFAILEDW As Long = 4 '// lParam:wzPath ret:1(cont),0(EndDialog)
Private Const BFFM_IUNKNOWN        As Long = 5 '// provides IUnknown to client. lParam: IUnknown*

'// messages to browser
Private Const BFFM_SETSTATUSTEXTA   As Long = WM_USER + 100
Private Const BFFM_ENABLEOK         As Long = WM_USER + 101
Private Const BFFM_SETSELECTIONA    As Long = WM_USER + 102
Private Const BFFM_SETSELECTIONW    As Long = WM_USER + 103
Private Const BFFM_SETSTATUSTEXTW   As Long = WM_USER + 104
Private Const BFFM_SETOKTEXT        As Long = WM_USER + 105 '// Unicode only
Private Const BFFM_SETEXPANDED      As Long = WM_USER + 106 '// Unicode only


Public Type BrowseInfo
```

```
  hWndOwner As LongPtr
  pIDLRoot As Long
  pszDisplayName As String
  lpszTitle As String
  ulFlags As Long
  lpfnCallback As LongPtr
  lParam As LongPtr
  iImage As Long
End Type

Private Function PtrToFunction(ByVal lFcnPtr As LongPtr) As LongPtr
  PtrToFunction = lFcnPtr
End Function

Private Function CorrectPath(ByVal sPath As String) As String
  If Right$(sPath, 1) = "\" Then
    If Len(sPath) > 3 Then sPath = Left$(sPath, Len(sPath) - 1) ' Strip backslash from non-root
  Else
    If Len(sPath) = 2 Then sPath = sPath & "\"                  ' Append backslash to root
  End If
  CorrectPath = sPath
End Function

Public Function FolderBrowse(ByVal sDialogTitle As String, ByVal sPath As String) As String
  Dim ReturnPath As String

  Dim b(MAX_PATH) As Byte
  Dim pItem       As Long
  Dim sFullPath   As String
  Dim bi          As BrowseInfo
  Dim ppidl       As Long

  sPath = CorrectPath(sPath)

  bi.hWndOwner = 0 'Screen.ActiveForm.hwnd

  'SHGetSpecialFolderLocation bi.hWndOwner, CSIDL_DRIVES, ppidl

  bi.pIDLRoot = 0 'ppidl

  bi.pszDisplayName = VarPtr(b(0))
  bi.lpszTitle = sDialogTitle
  bi.ulFlags = BIF_RETURNONLYFSDIRS
  If FolderExists(sPath) Then bi.lpfnCallback = PtrToFunction(AddressOf BFFCallback)
  bi.lParam = StrPtr(sPath)

  pItem = SHBrowseForFolder(bi)

  If pItem Then ' Succeeded
    sFullPath = Space$(MAX_PATH)
    If SHGetPathFromIDList(pItem, sFullPath) Then
      ReturnPath = Left$(sFullPath, InStr(sFullPath, vbNullChar) - 1) ' Strip nulls
      CoTaskMemFree pItem
    End If
  End If
```

```
  If Right$(ReturnPath, 1) <> "\" And ReturnPath <> "" Then  'Could be "C:"
    FolderBrowse = ReturnPath & "\"
  End If

End Function


Public Function BFFCallback(ByVal hWnd As Long, ByVal uMsg As Long, ByVal lParam As Long, ByVal
sData As String) As Long
  If uMsg = BFFM_INITIALIZED Then
    SendMessageA hWnd, BFFM_SETSELECTIONA, True, ByVal sData
  End If
End Function

Public Function FolderExists(ByVal sFolderName As String) As Boolean
   Dim att As Long
   On Error Resume Next
   att = GetAttr(sFolderName)
   If Err.Number = 0 Then
      FolderExists = True
   Else
      Err.Clear
      FolderExists = False
   End If
   On Error GoTo 0
End Function
```