NTNU – Trondheim
Norwegian University of
Science and Technology

Enhancing Prosthetic Sockets for Alpine Skiers

Josefine Caroline Stokke Haugom

February 2020

MASTER THESIS

TrollLABS
Department of Mechanical and Industrial Engineering
Norwegian University of Science and Technology

Supervisor: Ass. Prof. Knut Einar Aasland, MTP
Co-Supervisor: Prof. Martin Steinert, MTP
Ph.D. Jørgen Falck Erichsen, SIAT, IBM

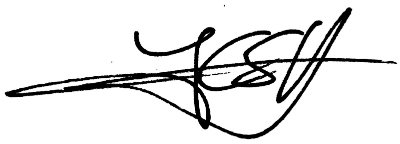
This page is intentionally left blank.

Preface

This is a Master Thesis submitted at the Norwegian University of Science and Technology (NTNU) as part of the Master's Degree Programme Mechanical Engineering (*sivilingeniør i produktutvikling og produksjon*), TMM4960 Engineering Design and Materials. The work has been conducted at the Department of Mechanical and Industrial Engineering (MTP) at NTNU, within the research and prototyping laboratory TrollLABS. The case was presented by Trøndelag Ortopediske Verksted (TOV) with initiator prosthetic engineer Eivind Modalsli, and his team. The cooperation with TOV was mainly presentation of the problem, providing of some prosthetic equipment, user (patient) meeting and a couple of meetings.

This thesis is dealing with an engineering-medical problem and written from a mechanical engineering standpoint. The thesis is written under the assumption that reader has a certain technical knowledge in the field of mechanical engineering.

Trondheim, February 2020



Josefine Caroline Stokke Haugom

Acknowledgment

I would like to thank Trøndelag Ortopediske Verksted (TOV) and Senter for Idrettsanlegg og Teknologi (SIAT) represented by Eivind Modalsli and Jørgen Falck Erichsen respectively for introducing this project. As well, the orthopedic team at TOV for sharing their thoughts, challenges and ideas regarding prosthesis. Gratitude goes to M.Physio. Natalia Bełdowska for support regarding medical terminology and the physiology of amputees. Special thanks to the prosthetic user which did not hesitate to share his first hand knowledge, and showing first-hand how the prosthesis works. Børge Holen and Jan Magnus Granheim Farstad for the training and support using the big 3-axis CNC-mill when producing the stump leg. I would like to thank the students and staff at TrollLABS for sharing their perspectives. And of course my supervisors for all their support and ideas throughout the project, Associate Professor Knut Einar Aasland and Professor Martin Steinert at the Department of Mechanical and Industrial Engineering and Jørgen Falck Erichsen (SIAT) at The Norwegian University of Science and Technology.

- J.C.S.H.

Abstract

Prostheses have evolved greatly over the recent years, improving the life of amputees greatly. Trøndelag Ortopediske Verksted (TOV) are making most of the prosthetic devices used by the amputees in the central part of Norway. With passion, they wish to activate their prosthetic users, showing them that it is possible to do great things even after loosing a limb. Alpine skiing is one of these activities. However, when the above the knee amputees were equipped with alpine skis using their ordinary prostheses they lost all control. Skiing got more terrifying than rewarding. The socket, which is made to secure the prosthesis to the residual limb was slipping, rotating around the limb. As TOV wish to improve the life of their users, they want to find solutions to this problem.

This research has investigated the initial situation, including anatomy of lower limb amputation, prosthetics and socket design as well as development possibilities of enhancing the socket design. In the research, a need for testing sockets without being dependent on the amputee was emphasized and sat as the main objective. The research and development process continued on to explore possibilities of testing rotation in prosthetic sockets. Using various prototypes, mechatronics including force sensors like load cells and flexible force sensitive resistors was explored and implemented in a test rig. The final testing rig, includes a wooden residual limb model, load cells, force sensitive resistors and an user interface intended for usage by TOV or in a following student project.

Keywords: Product Development, Prototyping, Fuzzy Front End, Prosthetic, Amputation, Mechatronics, Sensor System

Sammendrag

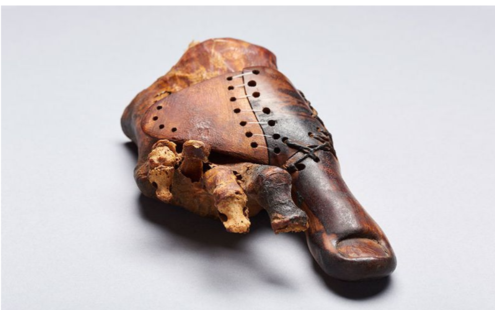
Proteser har utviklet seg kraftig de siste tiårene noe som har økt livskvaliteten til protesebrukere. Trøndelag Ortopediske Verksted (TOV) er hovedaktøren for konstruksjon av proteser for protesebrukere i Midt-Norge. De har en lidenskap til å forbedre livet til brukerne sine og ønsker å vise dem at «alt er mulig» selv etter en amputasjon. I vintersesongen har en gruppe fra TOV tatt med brukerne sine ut i alpinbakken. For enkelte, særlig de med låramputasjoner, var dette mer en skremmende opplevelse enn å være givende. Brukerne ble satt i bakken med vanlige alpinski med deres ordinære proteser. I første leksjon, ploging, mistet brukerne totalt kontroll over skien på grunn av glidning i protesehylsen rundt stumpen, den gjenstående delen av beinet. Med sitt driv om å bedre livskvaliteten til sine brukere, ønsker TOV å finne løsninger på dette rotasjonsproblemet.

I denne masteroppgaven har nå-situasjonen blitt studert, inkludert anatomi og låramputasjoner. Protesen TOV bruker er en av standardproteseløsningene som brukes i verden i dag. I behovsanalysene som ble gjennomført viste det seg at det er en rekke begrensinger, hva gjelder modifikasjon av protesehylsen og sett fra anatomiens side. En del av målene for denne oppgaven var å kartlegge behovene, både for pasienten som skal bruke protesen, men også for videre utvikling. TOV ser på dette som et langsiktig prosjekt, både hos dem og eventuelt nye prosjekt og masteroppgaver. I kartleggingen kom det også frem at TOV ikke har noen muligheter til å teste protesene sine før de prøver dem ut på pasientene. Mulighetene for resultatavhenging forbedring, er avhengig av å ha en måte å måle utfallene på. Hovedmålsettingen med denne masteroppgaven ble derfor å studere og utvikle måter å teste protesehylser på. Utfallet av denne masteroppgaven inkluderer, i tillegg til bruker og behovsanalysene, en testrigg som er ment til bruk av ortopediteknikere og -ingeniører hos TOV og i videre utviklingsprosjekter I testriggen er det implementert sensorer, som kan gi et inntrykk av hvordan protesen føles på stumpen til en amputert når det blir påsatt en kraft som skal simulere skien i alpinbakken.

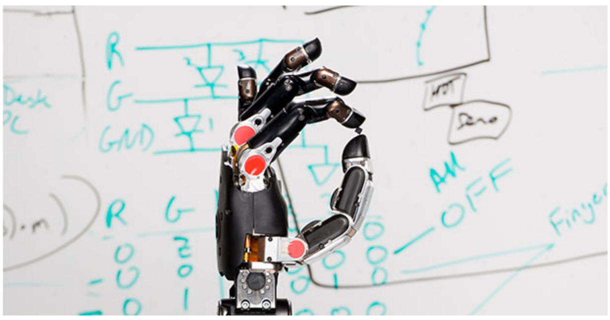
1

Introduction

In medicine, a prosthesis is an artificial device that replaces a missing body part intended to restore all or some of its lost function. The missing body part can as well as a leg or a hand, be a tooth (dental prostheses), jawbone (maxillofacial prosthesis), a hip or other internal or external body parts. The body part might be lost due to a disease, trauma or a congenital disorder (birth defect). Prostheses have presumably been used since the first civilizations, as indicated with the findings of *The Cairo Toe*, Figure 1.1a. Since then, technology have advanced both in design and functionality. In the most recent years, the prostheses capabilities have evolved significantly with implementation of advanced sensors and actuators. Today, you are most likely not realizing how many people that are walking around with a prosthetic leg in developed countries. The Defense Advanced Research Projects Agency (DARPA) is one of the leading groups of researchers which develops advanced, robotic prostheses as shown in Figure 1.1b. Soldiers make up a large group of people that have undergone an amputation, as result of trauma during battle. DARPA is working on getting the soldiers literally back on their feet. For the regular user, the initial prosthetic is usually standardized and comes in a different price range than the above mentioned robotic arm of DARPA. With the standardization the "allround" prostheses does not always hold up to the users needs, which is the reason Trøndelag Ortopediske Verksted have presented this project.



(a) The Greville Chester Great Toe also known as The Cairo Toe was discovered in Sheikh 'Abd el-Qurna and it's claimed to be one of the earliest prosthetic devices known to scientists. It has been dated back to the early first millennium BC. The construction was refitted several times to match the exact shape and fit of the woman who wore it. (Daley, 2017)



(b) The LUKE Arm system has come as far as proclaiming it can make the user feel again. With advanced neurotechnology DARPA, the Defense Advanced Research Projects Agency under the U.S. Department of Defense has developed anthropomorphic modular prototype prosthetic arm systems, which offer users increased dexterity, strength, and range of motion over traditional prosthetic limbs (Emondi, 2017)

Figure 1.1: Ancient and new prosthetic technology

1.1 Lack of Grip In Prostheses When Skiing

Trøndelag Ortopediske Verksted (TOV) is the main orthopedic device workshop in the central region of Norway as a division of St. Olavs Hospital. TOV is responsible for design, manufacturing and adapting orthopedic aids to assist people in need of such device. TOV is passionate about their patients and want them to be active with their new prostheses. Obtaining an active lifestyle is especially important for amputees. With a group of prosthetic users, Orthopedic Engineer Eivind Modalsli and former national team manager in Telemark skiing, went to the slopes and put skis on their artificial and biological leg. When going through the movements of skiing downhill, especially the transfemoral amputees showed difficulties with snowploughing, the action you do to slow down when going downhill on skis. The prosthetic users had control and got the skis into the V-formation, but when going back to parallel posture, the ski on the prosthetic leg did not follow, and was left ploughing. Worth to notice is that these prosthesis users are "normal people", not para-alpine skiers, using their ordinary prosthesis in this situation. Looking into para-alpine skiers, the few that are skiing with two skis, often have special made equipment for them to be able to have a grip. What seems more common among the athletes is to rather go by mono-ski (one ski) or sit-skis. With this issue TOV suggested to explore the possibilities of enhancing the grip in their transfemoral prostheses, in particular the grip of the prosthetic socket.

Investigating the initial situation and development possibilities on this subject, indicated a need for testing socket fit without relying the testing to be done on human beings. As this is a new development project, both for TOV and for the research group at the Department of Mechanical and Industrial Engineering at NTNU, some investigation on prosthetics and anatomy was essential to carry out product development and prototyping to find good solutions for the stakeholder. Through the investigation, including literature studies and interviews with TOV and a patient, important boundary conditions were disclosed. TOV recognized this topic as a larger project from the beginning, implying aspirations to have someone to carry the work further after this master thesis is finished. For that reason, this project have been used to explore different opportunities for development, opening doors, rather than narrowing in to one great solution. But most important, the ground work of this thesis together with the development of a physical testing platform could speed up the process of development of products to improve the socket fit.

> ” *The aim of this master thesis has been to create a test setup for enabling further research and development to enhance the rotational grip of TOVs standard transfemoral prosthetic socket.*
>
> **— JCSH**

1.2 How to Grasp the Problem?

The thesis is built up by introducing basic anatomy and principles of transfemoral amputations, for a reader with a technical/mechanical educational background. Following the medical theory is a description of the standardized prosthetic device which TOV produces for their patients. To keep the product development within the medical and mechanical boundaries, testing mechatronics sensors made a big part of the prototyping in this project. Hence, a theoretical part regarding the use of mechatronics in the test setup is included in the first part of this thesis.

As mentioned, additional important information was obtained through meetings with the prosthetic team of TOV and an interview with a prosthesis user. Following the theoretical chapters comes a thorough case investigation where the problem went was defined and re-defined iteratively. Research and development has been conducted at TrollLABS. It is a research and prototyping laboratory at NTNU, which by rapid prototyping and SCRUM based development teams obtains many radical innovation projects through the extensive. The majority of projects at the lab, including this one goes through a sequence of product development methodologies like design thinking, agile, scrum and lean product development.

"Prosthesis [praːs'θiːsɪs] *- an artificial part of the body. Originates from Latin pros 'in addition' + tithenai 'to place'."*

- Oxford Dictionary

2 Prosthetics and Anatomy of Transfemoral Amputation

This chapter is intended to cover the needed medical theory in which the product development project is based on. Starting out with an introduction to transfemoral amputation, which is the medical term of amputation of a leg where the cut is done through the femur (thigh bone). Then, moving further into the prosthetic solution, NU-FlexSIV prosthetic sockets, that TOV produces for their users, which sets boundaries for the product development work of this project.

2.1 Terminology

To understand the literature regarding amputation and prosthetics, being familiar with some common terms is significant. Figure 2.1 shows the some of the standard anatomical directional terms. In this thesis *distal* and *proximal*, *medial* and *lateral*, along with *posterior* and *anterior* will be used frequently when referring to the placement on the *stump*. A stump is the part of a limb of the body remaining after amputation or a congenital disorder, also called residual limb, as illustrated in figure 2.3 b.

Prosthetics & Anatomy Glossary

Alignment Position of a prosthetic socket in relation to foot and knee

Amputee A person who has had all or part of a limb amputated or is born without a limb

Antagonist muscle The muscles that acts in opposition to another muscle

Bilateral Both sides, bi = both

Brim (socket -) The proximal edge of the socket

Claudication Pain in the limb upon exertion, due to circulatory insufficiency

Disarticulation The limb is amputated through the joint

Dissect To separate into pieces: expose the several parts of (something, such as an animal) for scientific examination

Doffing Taking the prosthesis off (*to doff*)

Donning Putting the prosthesis on (*to don*)

Edema Swelling of the tissues

Endo Inside (as in *endoskeletal prosthesis* - one with internal supporting structures)

Exo Outside (as in *exoskeletal prosthesis* - one with external rigid structure)

Gait training Learning to walk with a prosthesis. Gait - a person's manner of walking.

Ischial Socket Prosthetic socket which depends on using ischium part (distal, characterized by a hole) of the pelvic bone to sit (see figure 2.7)

Interface Inner surface of socket, or portion of prosthesis closest to the skin

Ligate 1) To tie with a ligature, 2) To join together (something) by a chemical process. (Tie up, in contrast to suture)

Liner Soft socket rolled or pulled over the residual limb and used for protection, comfort and in some cases as a suspension device and/or to secure vacuum

Midstance the point in the gait cycle where all weight is put vertically on one leg

Myodesis (in amputation surgery) Is when the muscle is secured to the bone by suturing the distal tendon via pre-drilled holes to the bone.

Myoplasty Attaching sectioned muscles to opposing muscles

Pistoning When a liner stretches so that the stump elongates - or the vertical motion of a residual limb inside a prosthetic socket

(sock) Ply Socket sock thickness, ply thickness can depend on manufacturer [a]

Premorbid Occurring or existing before the occurrence of physical disease or emotional illness

Prosthetists a specialist in prosthetics

Residual limb The remaining part of the limb after amputation (the *stump*)

Subcutaneous Situated under the skin

Socket The part of the prosthesis that fits around the residual limb and fits around the liner or socket insert if one is used

Socket insert A soft form that is contoured to fit around the residual limb and fits inside the socket to provide for some increased padding and comfort for the residual limb

Suction socket A socket that excludes the entry of air and is held to the residual limb by the suction of negative pressure maintained within the socket (*vacuum socket*)

Suture *(Verb:)* to unit, close or secure with sutures. *Noun:* a strand or fiber used to sew parts of the living body; a stitch made with a suture.

Transfemoral (TF) Amputation above the knee, through the femur bone

Transtibial (TT) Amputation below the knee (through the tibia and fibula bones)

[a]Sanders, Cagle, Harrison, Karchin *Amputee socks: how does sock ply relate to sock thickness? (2012)*

[a]This selection of prosthetic/amputation terms was conducted with definitions from the Canadian site Award Prosthetics (Prosthetics, 2012), Merriam-Webster Medical Dictionary (Merriam-Webster, 2019) and Physiopedia.com (Coughlan et al., 2019)

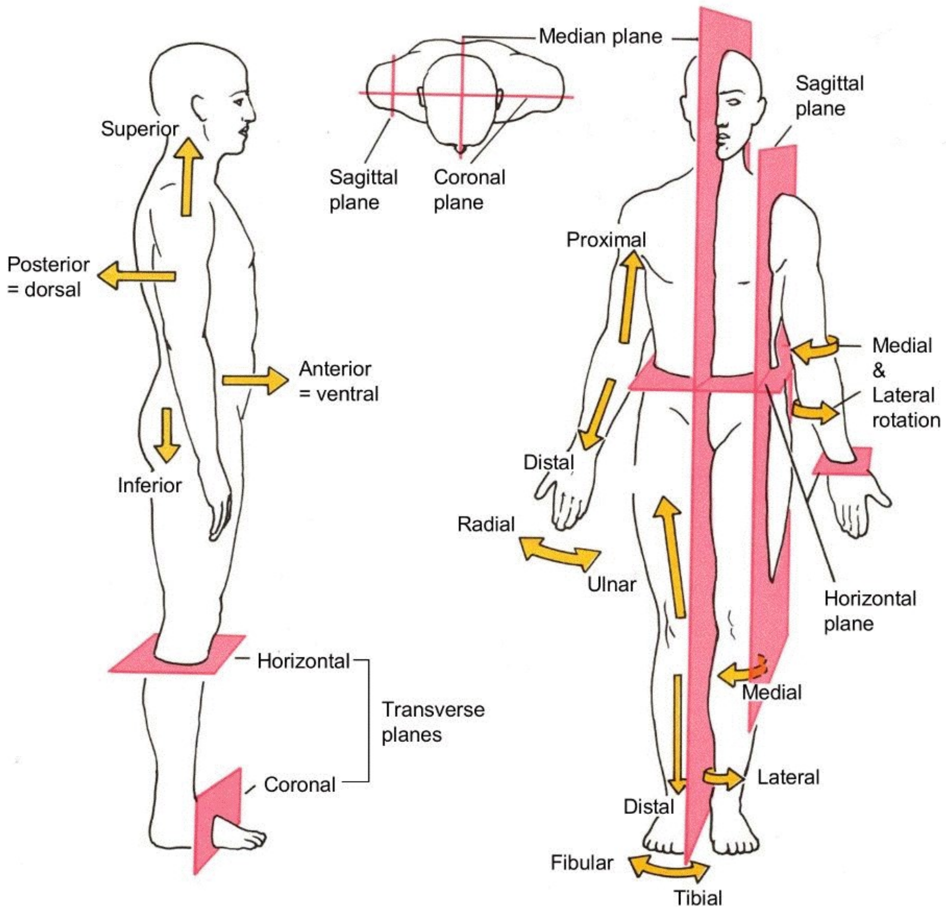


Figure 2.1: Anatomical directions: Chief terms, directions and the main planes of reference in the body (Swenson, 2009)

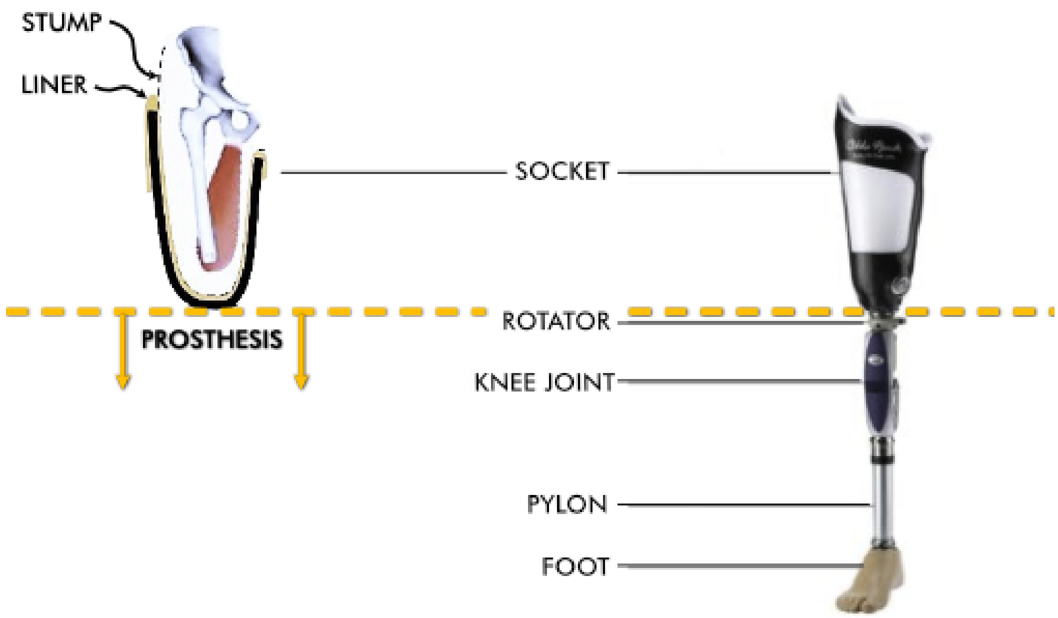


Figure 2.2: Basic orientation of transfemoral prosthesis (combination of illustrations from Caldwell et al. (2016) and Texas (2018)).

2.2 Basic Principles of Transfemoral (TF) Amputations

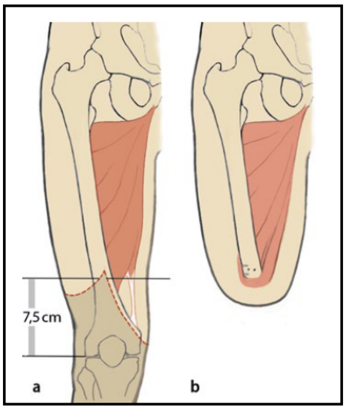


Figure 2.3: The optimum length of residual bone is suggested to be approximately 7.5-10 cm proximal to the superior border of patella (Coughlan et al., 2019)

As the name suggests, transfemoral amputations (TF) are done through the femur. A thumb rule for length of the residual limb is about 2/3 of the untouched femur. Physiopedia suggests that the optimum length would be 7.5-10 cm proximal to the superior border of the patella, as shown in Figure 2.3 a. Shorter stumps tend to end up in abduction because of an imbalance between the adductor and abductor muscles, Figure 2.4. At St. Olavs Hospital they tend to lean towards a 5 cm distance proximal of the patella, making the stump slightly longer (Modalsli et al., 2019). Though these are the guidelines, the condition of the patient is the determinant of where the incision is done.

Nerves The amputation is done by surgically dissecting through the skin to isolate and ligate the nerves and blood vessels. The nerves are dissected on tension to make them retract back into the tissues where they can heal away from the stump end, and in order to minimize the risk of neuropathic pain. The femur is dissected perpendicular to its axis, reducing sharp edges.

Muscle tissue Distal attachments of the thigh muscles are lost through a TF amputation. A myodesis may be performed to anchor the adductor (sometimes also the hamstring) muscles to the bone in order to preserve their function and length. The quadriceps and hamstrings can be sutured together over the distal end of the femur through a myoplasty. To help with padding and covering the distal end of the stump, a myoplasty of the antagonist muscles is performed along with a Gottschalk myodesis (Coughlan et al. (2019), Gottschalk and Stills (1994)).

Stump behavior With new amputees the stump is expected to change its shape and size due to swelling, muscle changes, etc. Eventually the stump will stabilize both in size and shape. When introducing an amputee to a prosthesis, a stabilization period is expected before the stump will adapt to its new environment. For instance, the forces acting on the tissue and lack of airflow, can initially cause the skin to get irritated and having temperature regulation difficulties. Both the internal healing process of the stump and the environment which it is put in (e.g. the socket) will contribute to changes in the stumps appearance and behavior.

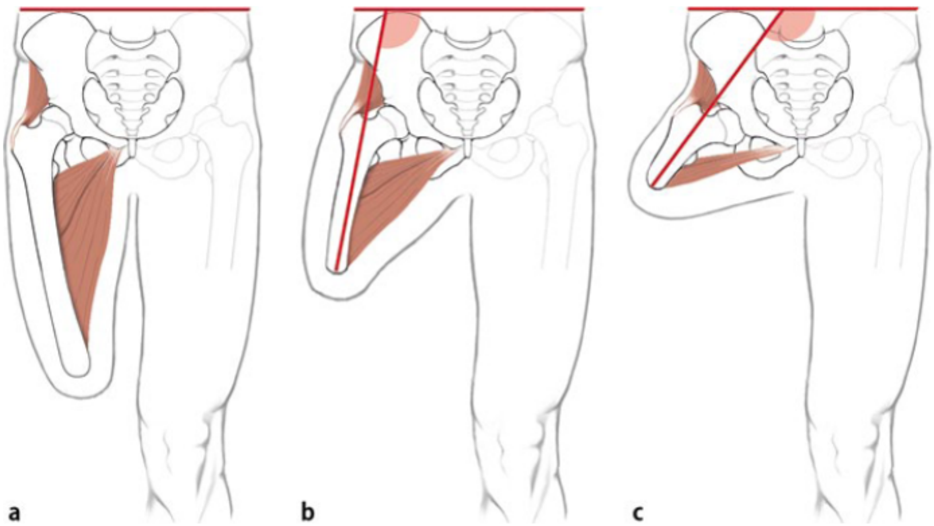


Figure 2.4: The length of the transfemoral stumps affects the muscular stability of the residual muscles. Very short stumps often end up in abduction due to instability. (Coughlan et al., 2019)

2.3 The Prosthesis

Moving over to mechanical constructions, this section intend to cover the basics of TF prostheses with references and links to technical manuals for further reading. As this project is limited to one prosthesis technique, NU-FlexSIV Socket, emphasis is put on this technique with respect to issues that I found useful for this product development project. As Figure 2.2 illustrates, there are more to a prosthesis than just the socket. The rotator, knee joint, pylon and foot are replacing the removed part of the leg. The socket, often along with a liner (see section 2.3.1) and locking mechanisms make up the connection and suspension system between the stump and the prosthetic leg.

2.3.1 Suspension System

The suspension system is basically what keeps the socket on a residual limb. In the NU-FlexSIV-system, this consist of a liner and can be supplemented with locking systems and stump socks.

Figure 2.5: Transfemoral liner by manufacturer Össur *Iceross Dermo Seal-In®*.

Liner

Between the socket and the skin, the NU-FlexSIV sockets require use of a liner. The liner is a protective cover which reduces movement and chafing between the skin and the socket. For the majority of liners, the main inner layer is made of a silicone material providing suctioning covered with a textile to extend durability and ease to don the stump into the socket. There are three main ways of attaching a liner to the socket; locking, cushion and seal-in. The seal-in liner has a Hypobaric Sealing Membrane (HSM), as seen as the white brace near the distal end in Figure 2.5. The Iceross Liner offers a firm hypobaric suspension which protects and stabilize the soft tissue and makes the stump more uniform conical.

The liners are ready-made products ordered directly from a supplier. They come in different sizes, mainly by a measure of the circumference of the stump. For the Iceross liner, the measurement is taken 4cm from the distal end with the tissues hanging down. The right size of the liner is important both to get sufficient tension on the HSM (fit and effect of seal), support of the stump tissue and fit in the socket, see figures on page 4 of the Iceross Technical Manual (Ossur, 2004). The user manual of the Iceross liner, which includes images of sizing, donning and other practical information can be found in Appendix B.2 and B.2.1 (Ossur, 2019).

Figure 2.6: Expulsion Valve by Össur *Icelock® 500 Series, Icelock 552 (Ossur, 2019)*.

Expulsion Valve

To create a stronger attachment, many sockets are made with an expulsion valve like the Össur Icelock in Figure 2.6. A valve moves the air out of the socket, or a pump that sucks it out. As well as creating a stronger attachment, the valve can help the user when donning the prosthesis as the air has an outlet. The valve in Figure 2.6 is an auto-expulsion valve with a push-button for suction release. The core of the valve can be unscrewed to facilitate rapid donning and makes it possible to pull donning socks through the valve (Ossur).

Liner Socks

As the stump may vary its volume, seal-in socks may be added between the socket and the liner to improve the fit in the proximal area of the residual leg. The socks are intended to use above the HSM and should never cross it as that could impede the vacuum suspension. The socks are measured in thickness "ply". The Seal-in Liner is will function with up to 6-ply thickness (2 x 3ply). See the manual for further information (Ossur, 2017).

2.3.2 The NU-FlexSIV Socket

Northwestern University (NU) with Phd, BPO, Stefania Fatone as the principal investigator has developed a sub-ischial prosthesis socket for transfemoral amputation. There are plenty of ways that have been used to fasten a prosthetic leg to an amputee, many which are uncomfortable and reduces mobility of the amputee. To address these issues the NU developed a teachable sub-ischial socket technique, the Northwestern University Flexible Sub-Ischial Vaccum (NU-FlexSIV) Socket that aims to improve comfort without compromising function (Fatone, 2019). Fatone et al. summarize the three important characteristics of the socket:

- Lower trim lines for increased hip range and motion, which not impinge the pelvis
- Vacuum for improved suspension
- Flexibility for improved comfort

  the muscles should be able to move comfortably within the socket as muscles contract during activity and splay during sitting

  -Fatone and Caldwell (2017b)

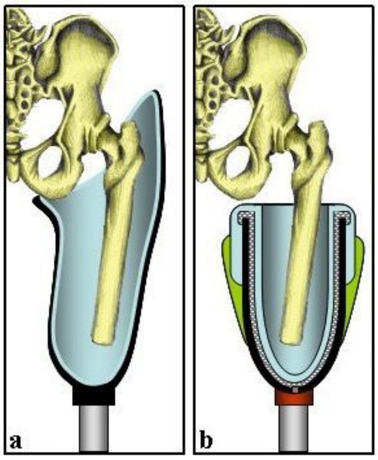


Figure 2.7: Two examples of transfemoral sockets. a) Ischial Containment and b) Sub-Ischial (Fatone, 2019)

This socket is in contrast to the ischial containment socket much shorter (Figure 2.7) and uses vacuum suction as the main suspension mechanism instead of "locking" the socket with the ischial tuberosity as seen on Figure 2.7 a). The socket includes a

- Silicone liner: highly compressive, cylindrical, fabric-covered
- Flexible inner socket
- Rigid outer socket (shorter than the flexible)
- Sealing sleeve (needed to apply vacuum between the liner and inner socket)

Forces and Socket Design

For the prosthetists the aim should be to design the socket in a way that the load is distributed comfortably over the residual limb. Limb volume fluctuation and shape changes calls for more challenges. The limb changes both volume during the day and shape with muscular activity. The NU-FlexSIV socket aims to be compliant so that the muscle can move without pushing against something hard one can accommodate the muscular activity.

The subischial socket has all the weight bearing loads on the thigh as this socket does not provide any pelvic support (unlike the ischial containment does). Figure 2.9 a) shows a model of the forces acting on the socket in midstance. Only the vertical components A1-A6 will be effective in opposing the downward force W exerted by the stump. For the opposing forces to be well distributed the socket needs to have a good tight fit. The NU-FlexSIV is a surface bearing socket systems which globally compresses and preloads the residual soft tissue via an undersized liner (transtibial liner 10-30%

Figure 2.8: TOV uses the NU-FlexSIV method of Northwestern University to make subischial transfemoral prostheses. a) NU-FlexSIV Socket, b) and c) range of motion, d) Single limb stance stability, e) rectification map (from left: medial, posterior, lateral, anterior views) (Fatone, 2019)

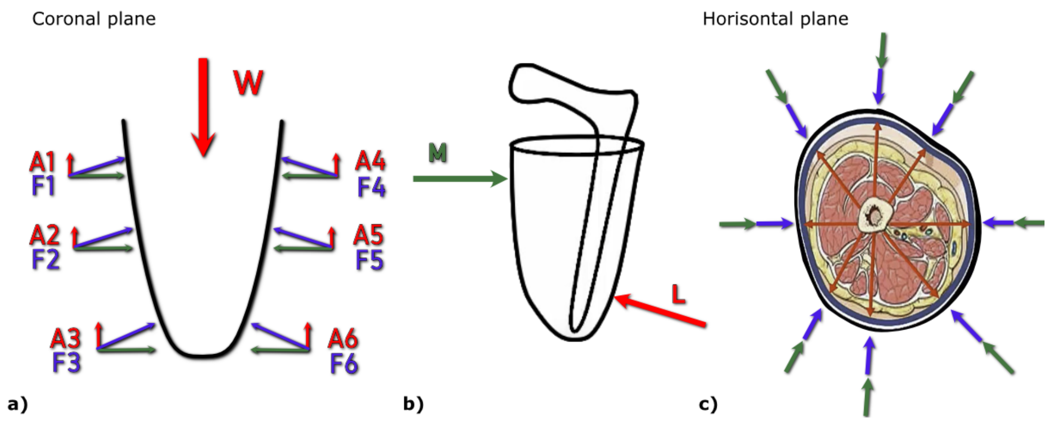


Figure 2.9: An illustration of the force distribution based on the description from the NU-FlexSIV webinar by Fatone and Caldwell (2017b)

undersized straight profile), undersized socket (4-6%) and a compliant socket. Given the cylindrical rather than conical profile provides greater compression of the softer proximal tissue, Fatone prefers to use transtibial liner for transfemoral limbs.

> *Redhead (1979) proposed that if the soft tissues of a transfemoral residual limb were supported in a suitably shaped socket, they would behave under load as an elastic solid in a quasi hydrostatic condition. The patient would be supported upon the relatively incompressible volume of the stump contained within the socket. Downward movement of the skeleton in relation to the socket under an axial compression load would be prevented by the relatively incompressible volume of the stump being contained within the rigid, fixed shape, fixed volume total contact socket.*
>
> **— Redhead (1979) quoted by Fatone and Caldwell (2017b)**

A hydrostatic condition - In their webinar, Fatone and Caldwell (2017b) draw a theory about the hydrostatic condition mentioned by Redhead. They use this theory to move the weight bearing forces at the distal end towards a distribution of normal forces over the residual surface. After preloading of the tissue occurs with the undersized liner and socket, the application of vacuum between the liner and socket is hypothesized to have its own effect on the underlying tissue. If we imagine tissues at the cellular level, we have a resting cell that expands or stretches with the application of vacuum to become a more hydrated cell, as I have illustrated in Figure 2.10 a) based on Caldwells presentation (Fatone and Caldwell, 2017b). It is suggested that this also occurs to the neighboring cells and creates a "locked" structure with minimal internal movement, see Figure 2.10 b) - c). For a transfemoral limb, the ratio between cells and bony structures to lock against is high. Tissue stiffening in this case will require a stiffer liner to compress the tissues prior to hydration than for the transtibial limb.

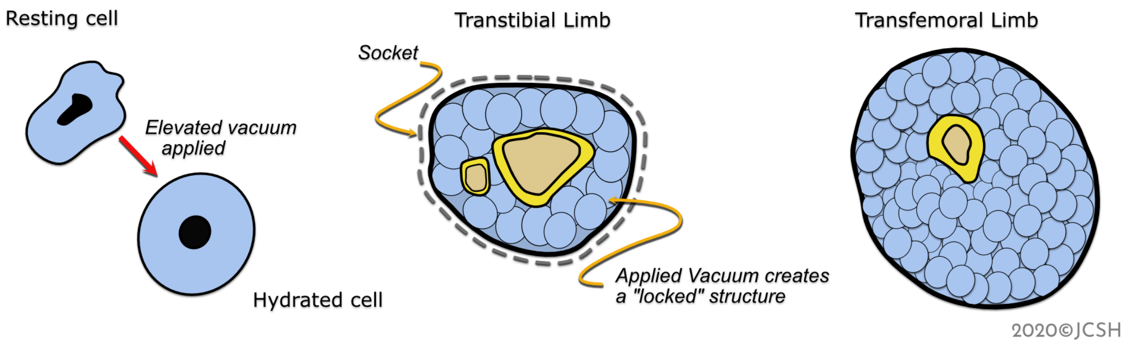


Figure 2.10: Hydrostatic Cell Response. A hypothesis of cell response due to preloading the tissue, illustration are based on the animations of the NU-FlexSIV webinar by Fatone and Caldwell (2017b)

2.3.3 Building a NU-FlexSIV Socket

The main steps of making the prosthetic socket is listed from 1-4 below. Fatone and Caldwell explain the socket technique and steps in their webinar which is to be found by following the QR-code in the margin (Fatone and Caldwell, 2017b).

1. Casting is done in a seated position with a casting liner on the stump. Plaster is used as casting material in most cases.

2. Positive mold is made by filling the negative cast with plaster.

3. Rectification In the NU-FlexSIV technique plaster is removed from specific areas, mark that no plaster is added. Fatone and Caldwell's research group have done studies on the rectification process. The group was mapping areas of the stump where plaster would be expedient to remove, which resulted in a decision making algorithm (see Appendix B.1.2 for the matrix). Figure 2.11 and 2.8 e) show the areas of which plaster is to be removed. In Figure 2.8 e) the amount of removed plaster is illustrated with the color map where the dark red represents the furthest displacement between the non-rectified and the rectified mold. The modifications usually begins between 6-4% proximally gradually reduced to 4-2% distally, mark that this occurs in specific regions rather than globally. Worth noticing is the posterior rectification which acts to flatten the soft tissues, which is beneficial when seating. The lateral reduction eliminate lateral gapping at midstance. Combining the proximal-lateral rectification creates a boomerang pattern. The fully rectified mold can be thought of as a barrel, with a slightly larger midsection than the distal and proximal end.

4. Assembling There are many ways of securing and sealing the flexible socket to the outer rigid socket. One way is to use a silicone sealing ring to the outer surface of the flexible inner socket below the rigid socket trim line. It reduces the relative motion between the two layers and creates a vacuum reservoir to reduce vacuum decay.

Core materials used in the socket

- Rigid outer socket: Laminated carbon fiber
- Flexible inner socket: Össur®FLEX EVA

  Shore hardness D39, recommended heating temperature 148-160°C, lightweight (10mm thick for most TTA limbs, 15mm thick for long limbs), flexible yet firm enough to support axial bodyweight.

Video of NU-FlexSIV Socket Course (Fatone and Caldwell, 2017b)

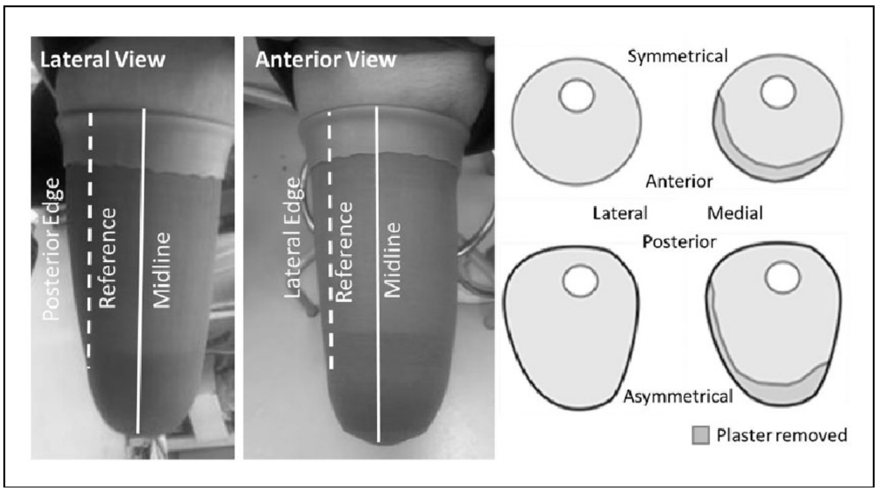


Figure 2.11: An example of a symmetrical residual limb shape. The area of the limb bounded by the reference line and the posterior and lateral edges are approximately the same and represents the area of the material removal during the rectification of the positive model. The cross-section diagrams show where the plaster is removed in the symmetrical and asymmetrical cases (Fatone and Caldwell, 2017a).

3 Mechatronics and Sensor Technology

Mechatronic is the field that combines mechanical engineering with electronics. Typically, mechatronics systems translate physical forces into an electrical signal or vice versa, via sensors and actuators. To control the input and output components, sensors and actuators, a microcontroller board such as an Arduino board is needed (Arduino, 2020)[1]. Modern prosthetics are full of mechatronical components. In leg amputees, this is usually below the socket such as the knee joint or the ankle/foot. In this chapter an introduction of sensors that translate mechanical forces into recordable output will be covered. Focus is put on the sensors which is used in this project, primarily load cells and force sensitive resistors.

3.1 Load Cells

Measuring physical force subjected on a surface, such as the weight of a person can be done by sensors as load cells. Load cells are found in digital kitchen scales, pallet weighing, platform weighing, belt scales, hopper scales and medical equipment, to mention some applications. There are several kinds of load cells, such as *bending beam-, compression-, dual shear-, miniature-, pancake-, shear beam-* and *single point* load cells. All of these have in common that they are transducers that translate mechanical pressure (force) into an electrical signal.

Mainly, there are four ways in which a load cell can translate applied force into a measurable electrical signal; *hydraulic, pneumatic, capacitive* and *strain gauge*.

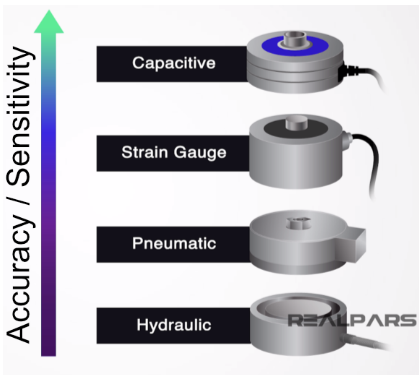


Figure 3.1: Load Cell Accuracy (Prielipp, 2019)

Hydraulic load cells use a piston and cylinder arrangement filled with oil or water, along with an elastic diaphragm and a bourdon tube pressure gauge. Pressure is applied to the load platform, which increase pressure of the liquid. This increase is proportional to the applied force or weight, see Figure 3.2.

Pneumatic load cells are similar to the hydraulic load cells, but uses air instead of fluids, as in Figure 3.2. Air pressure is applied to one end of a diaphragm. The air escapes through a nozzle placed at the bottom of the load cell containing a pressure gauge.

Capacitive load cells are built up by the ability of a system to store an electric charge. They are made up by two flat plates parallel to each other, see Figure 3.2. A current is applied to the plates and once the charge is stable, it gets stored between the plates. The gap between the plates decides the capacitance. When a load is applied, the gap shrinks, making a change in the capacitance (Prielipp, 2019).

[1] The examples mentioned in this thesis will be based on use of an Arduino Uno and Arduino Mega.

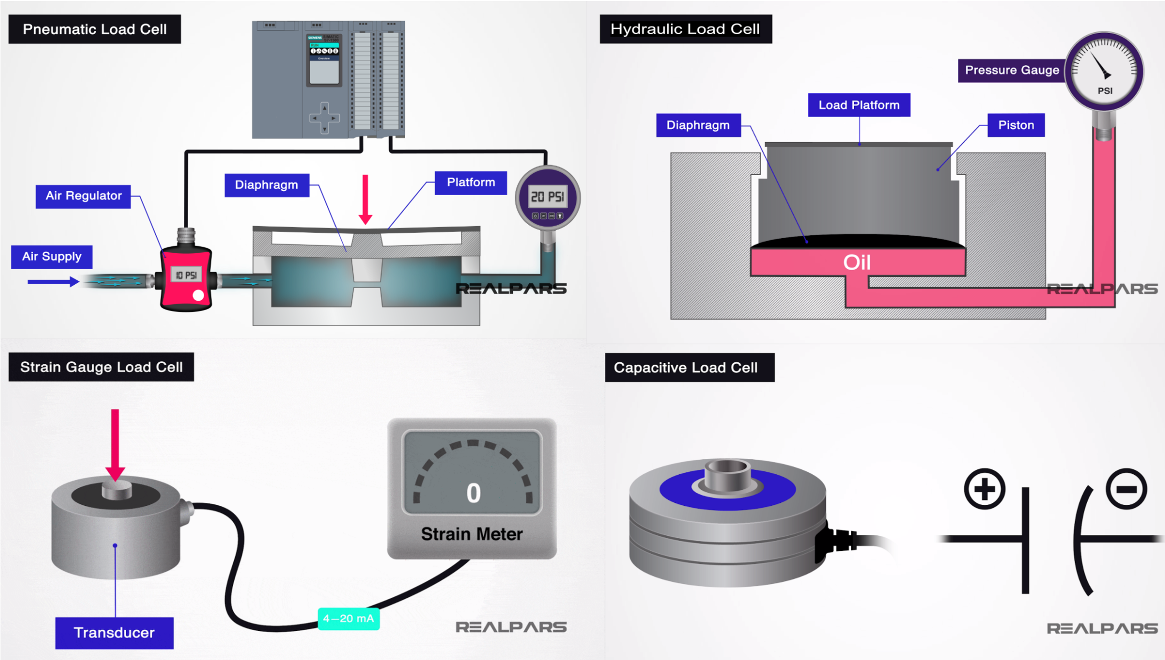


Figure 3.2: Four load cells technologies. Pneumatic, hydraulic, strain gauge and capacitive load cells (Prielipp, 2019).

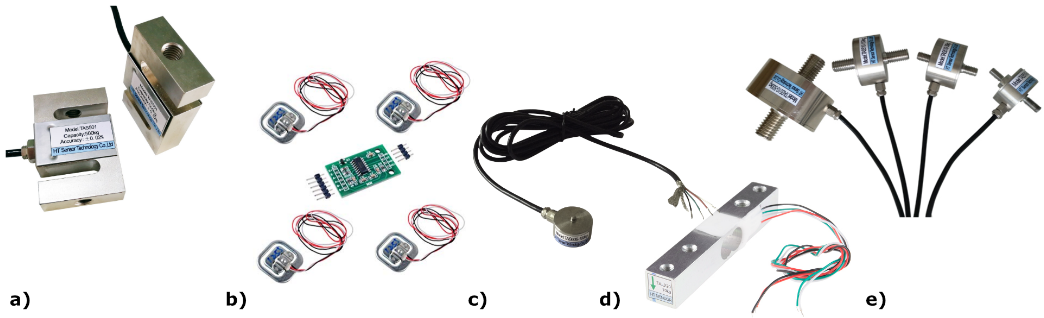


Figure 3.3: Assortment of strain gauge load cells. **a)** S-type load cell can measure tension and compression, **b)** 4 single strain gauge load cells with a combinator, **c)** disc/button-, **d)** straight bar- and **e)** compression/tension disk load cells

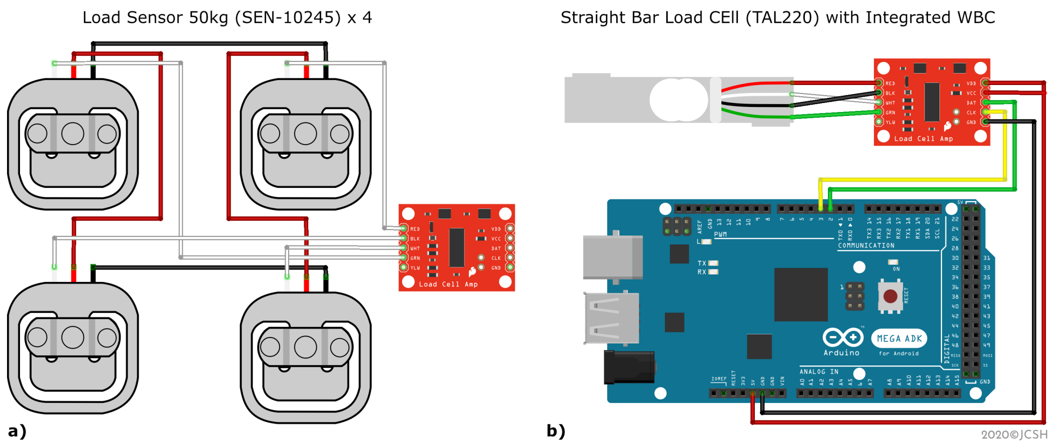


Figure 3.4: Two ways of connecting load cells. **a)** Four single strain gauge load sensors hooked up in a Wheatstone Bridge configuration to a load cell amplifier (which then can be connected to the Arduino the same way as in the second setup). **b)** Straight bar load cell containing four strain gauges within the sensor, giving four wires to be hooked directly up to the amplifier. (Made with Fritzing (fritzing.org))

3.1.1 Strain Gauges - Piezoresistive Sensing

Strain gauge load cells are the most common of the four and among the more accurate (figure 3.1). The force is recorded by the deformation of one or several strain gauges on the element (figure 3.5 c).

$$\text{Strain} = \varepsilon = \frac{\Delta L}{L_0} = \frac{\text{deformation}}{\text{original size}}$$

$$\sigma = E \cdot \varepsilon$$

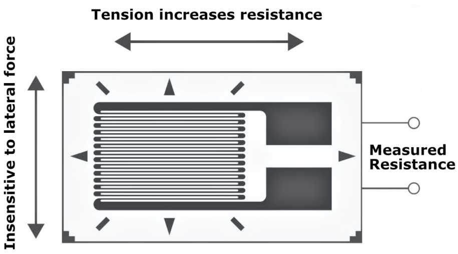


Figure 3.5: Strain Gauge (Naik)

Where σ is the stress (force) and E is the material dependent Young's modulus (PCB Load and Torque Division). Strain gauges work by the means of the piezoelectric effect, meaning that when a mechanical strain is applied to a material, it deforms and its electrical resistance changes (figure 3.5, Naik). The electrical resistance is linear and proportional to the stress or strain applied to the cell. Most strain gauge load cells work by using a *Wheatstone bridge* configuration (WBC). Either the load cell is configured with four strain gauges within the load cell or four single strain gauge load cells need to be hooked up in a WBC through a combinator (e.g. SparkFun Load Sensor Combinator https://www.sparkfun.com/products/13878) or directly to a *load cell amplifier*, like Sparkfuns HX711 (https://www.sparkfun.com/products/13879).

Wheatstone Bridge Configuration

A Wheatstone bridge is an electrical circuit used to measure unknown resistance by balancing two legs of a bridge circuit. As I have illustrated in Figure 3.4, the WBC consists of four resistors (strain gauges in load cells) or loads in a square. A voltmeter is bridging two corners of the square, while power and ground connected to the other corners as seen in Figure 3.6 (Phidgets, 2018). Factors such as gauge placement, temperature and length of the wire that are used to complete the circuit, influence the resistance in the WBC, which can create errors in the measured values. Adding resistance to the bridge can compensate for this. In some load cells, like the coin/button load cell and the straight bar load cell in Figure 3.3, the WBC is implemented in the load cell. Others, as in many bathroom scales, the load cells need to be set up in the configuration by using four load cells and a connection module. Typically load cells containing a WBC have four wires, red (E+), green (O+), black (E-), white (O-) as illustrated in Figure 3.4 b).[2] Single load sensors typically have three wires (power, ground and signal) as seen in Figure 3.4 a).

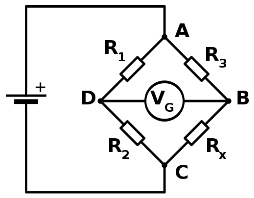


Figure 3.6: Wheatstone bridge configuration of strain gauge load cells. (Phidgets, 2018)

3.1.2 Straight Bar Load Cells

Straight bar load cells (SBLC) are next to the single strain gauge load cells most common among consumer goods (e.g. kitchen and bathroom scales). These sensors are versatile and come in several load capacity ranges (3~2000 kg (HTC, 2019)). It is important to note that the heavy capacity load cells do not always cover the lighter range and a larger domain might affect precision.

In SBLCs the cell is set up in a Z formation, with one fixed end and one free end where the force of which should be measured is applied. Bending distortion will be measured by the four strain gauges on the cell when a torque is applied. Two of the strain gauges measure the tension and two measure compression, as illustrated in Figure 3.7. With a WBC setup the changes in resistance over the gauges can be measured. The resistance is proportional to the force applied in straight bar load cells.

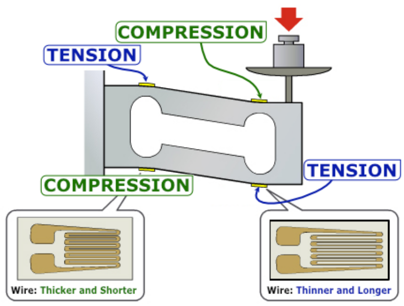


Figure 3.7: Simplified illustration of straight bar load cells mechanics (Al-Mutlaq, 2016)

[2]Load cell nodes either have positive or negative excitation (E+ and E-) or output (O+ and O-).

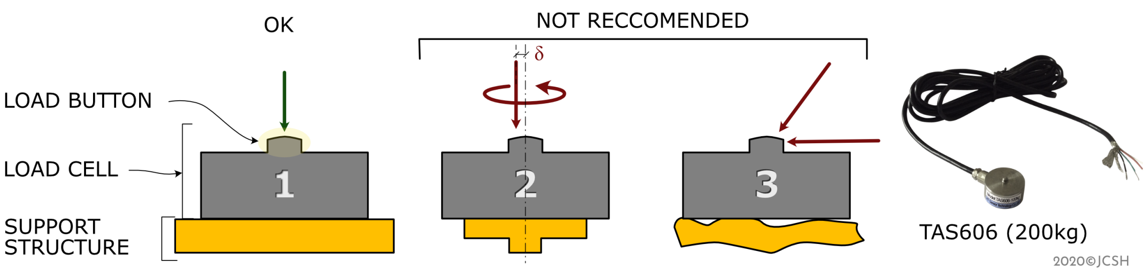


Figure 3.8: Mounting and forces on disk load cells like the TAS606 from HTC (4)(Sparkfun, 2015). **1)** Green arrow shows the proper load vector ⊥ DLC which is placed on a flat surface. **2)** DLC's lower outer ring must be supported, this one is too narrow. Forces displaced from the centerline and torque should be avoided. **3)** Flat support surface is required, side forces should be avoided.

3.1.3 Disk Load Cells

The disk load cells (DLC) are typically smaller in size than the SBLC and can be easier to mount. These load cells are very sensitive to the load orientation. As illustrated in Figure 3.8 the load cell must be placed on a rigid, flat and clean surface which cover all of the lower outer ring (not like example 2 and 3 in the figure). Magnetic and electrical fields (EM) have sometimes shown to create interference voltage within a measuring circuit. The load cell, connection cabling and electronics could be secured in a shielded housing to avoid the EM field. The force must act precise on the center of the load cell button, perpendicular (⊥) to its horizontal bottom plane, as shown on DLC nr 1 (figure 3.8.1). Wrong force orientation would affect the readings, and is said to be one of the difficulties of using this type of load cell (Omega, 2018). Bending, torque and side forces as shown on DLC nr 2 and 3 (figure 3.8.2-3) is not recommended, and can give wrong readings and might affect the lifetime of the load cell. The DLC has as the SBLC an integrated WBC and is connected to the Arduino through a load cell amplifier.

3.1.4 Load Cell Amplifier

Most load cells require an amplifier as the produced signal from the LC may be too weak to be read by the certain components of the measuring system. Load cell amplifiers (LCA) can be standalone components, like HX711 (figure 3.9), or it is present in a signal-conditioning module such as the *OpenScale* and Qwiic Scale NAU7802 from SparkFun (sparkfun.com/categories/143 - Flex/Force). By connecting the LCA to microcontroller it is possible to read the changes in the resistance of the load cell, and with some calibration one can obtain very accurate weight measurements (Omega, 2019).

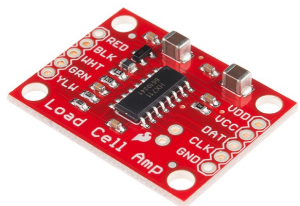


Figure 3.9: SparkFun Load Cell Amplifier HX711 (Sparkfun, 2016)

The LCA uses a two-wire interface, clock (CLK) and data (DAT), meaning that one will need two inputs of an Arduino per LCA. Additionally there are two power inputs (VCC) and one pin to ground (GND). As mentioned in section 3.1.1, the load cells usually have four wires, while you can see an additional fifth input on the HX711. The YLW (yellow, shield) is an optional input pin that is not hooked up to the strain gauge, but is utilized to ground and shield against outside electromagnetic interference (EMI) (Sparkfun, 2016).

3.2 Force Sensitive Resistors

Force sensitive resistors (FSR) are a polymer thick film (PTF) device which exhibits a decrease in resistance with an increase in the force applied to the active surface. There are four basic types of FSRs, *Single-Point, Linear-Potentiometer, 3D Single-Touch* and *3D Multi-Touch*. Only the Single-Point FSR will be discussed further. Its force sensitivity is optimized for use in Human-Machine Interface (HMI) or Machine-Machine Interface (MMI) (Interlink Electronics, Ohmite (2018b)). Generally, FSRs change its resistance inversely proportional to the applied force over the area of the FSR (Figure 3.13a).

Figure 3.10: Force Sensitive Resistors (Ohmite, 2018a)

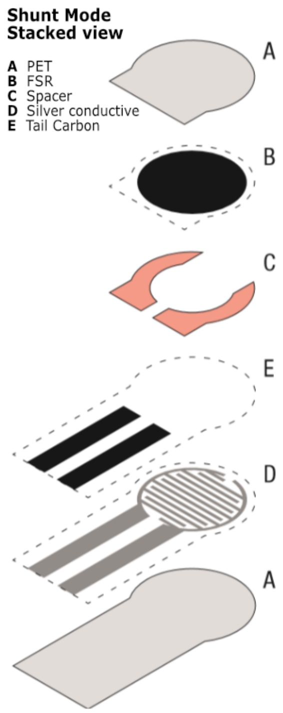


Figure 3.11: Material layers of a Shunt Mode FSR (Ohmite, 2018b)

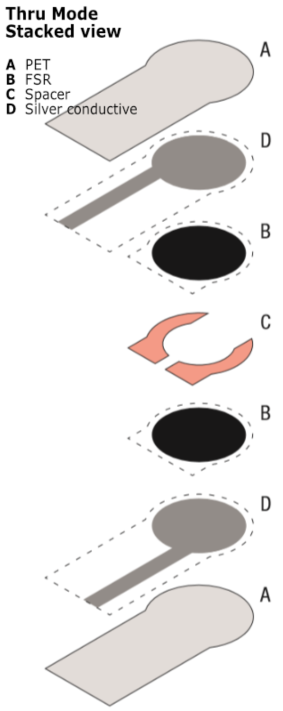


Figure 3.12: Material layers of a Thru Mode FSR (Ohmite, 2018b)

FSRs have a sensing range from about 20g to 5kg. Specific device characteristics will depend on the size, shape and materials used in construction Ohmite (2018b). FSRs are intended to detect changes in applied force and not suited for high accuracy or specific weight measurements.

In Figure 3.13a three zones are marked, as they show where the FSR's three mechanisms work to form the force resistance characteristics. The exact response curve depends on the given FSR, hence no values are given in this graph. *Break force* is the force required for the two membrane layers to contact. In the *Area Effect* zone the size of the contact area between two layers is increasing with force, thereby reducing the resistance. In the *Surface Effect* zone the force is increased as the contact of the two ink surfaces increase at a microscopic level.

Figure 3.11 and 3.12 shows the material layers of two FSR types, respectively in *shunt* and *thru* modes, from Ohmite. They both are an assembly of two separate printed substrate layers which are laminated together with a spacer adhesive around the outside of the sensing area. Figure 3.13b illustrates the working layers of a shunt mode FSR. They exhibit different force-resistance characteristics, as the electrode structure is printed differently. As commented by Ohmite (2018b), the differences includes [3]

Shunt Mode

- All electrical connections on one layer
- Force range up to 5 kg
- Fever print layers and less silver ink required
- Typically less expensive
- Shallower curve giving better control esp. at higher forces i.e. > 100g

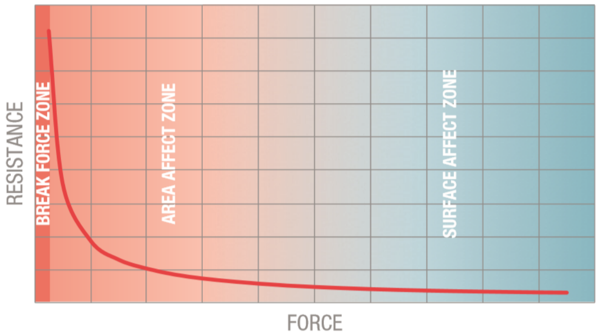
Thru Mode

- Electrical connection required to both top and bottom layers
- Force range typically limited to < 0.5-1 kg
- More print layers and more silver ink cost more than the Shunt Mode sensors
- More expensive

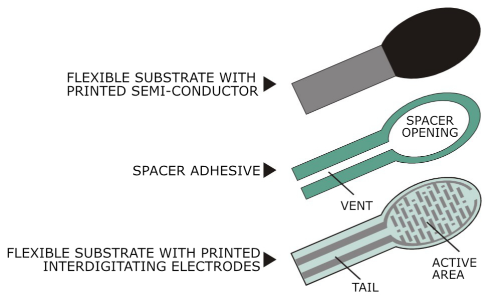
3.2.1 Considerations When Using FSRs

FSRs are typically highly durable, cost effective, low powered and small in size, however, as mentioned not suitable for highly accurate force readings, and not above 5kg. The quality and consistency of the electrical output depends on the quality and consistency of the mechanical input. The manufactures of FSRs (like Ohmite and Interlink) recommend the FSR to be mounted to a flat rigid surface to enable full force transfer and avoid preloading. Ohmite recommends an actuator no larger than 80% of the active area of the sensor (Ohmite, 2018b). Temperature affects the FSR becoming more sensitive when hot and less when cold. Humidity can also impact the FSR stability. The sensors are not fully sealed, so they should not be directly exposed to liquids.

[3] For further reading, Tekscan.com discuss differences between some other FSR technologies here: https://www.tekscan.com/blog/flexiforce/differences-between-force-sensitive-resistor-technologies-assessing-linearity-drift



(a) Typical Force vs Resistance of FSRs (Ohmite, 2018b).



(b) Shunt mode FSR construction showing the functions of the layers (Interlink Electronics).

Figure 3.13

3.3 Setting Up the Sensors with Arduino

Setting up the load cells and FSRs requires some work to get *force* as an output. This section will cover the basics of using the strain gauge load cells together with a load cell amplifier to output the force. Then a walk through of the FSRs voltage to force output.

3.3.1 Strain Gauge Load Cells Setup

To use strain gauge load cells, one will need some reference weights along with the connected load cell, amplifier and Arduino. The HX711 load cell amplifier has some Arduino libraries[4] to choose between. *HX711* by Bogdan Necula, *HX711_ADC* by Olav Knallhovd, *Queuetue HX711* by Scott Russell are only three examples, all with different strengths. The *HX711_ADC* from Olav Kallhovd (2019) is set up for the SI-system and is quite straight forward. In the library he offers both examples of usage and a calibration setup. With a gained calibration value for the load cell, change the value `calValue` in the load cell registration code to get force (or weight) output.

The strengths of the strain gauge load cells are the accuracy and the linearity of the measurements. All though to get a perfect result, the load cell must be well mounted and one must do a thorough calibration before usage.

3.3.2 Voltage Output from FSRs

FSRs can in some ways be quick and easier to set up than load cells, as they can work without doing specific calibrations per test. Alone, they will output the voltage which can be translated into the given pressure through some calculations. To get precise readings, one will need to calibrate the FSR by using some reference weights to get the slope and its changes through the FSRs capacity range. A four-minute walk-through of the steps of calibration can be viewed visiting the link in the QR-code (FlexiForce Tactile Force Sensors (2018) and Tekscan (2019)).

Calibration of FSR (FlexiForce Tactile Force Sensors, 2018)

Simplified Use of FSRs with Predefined Calibration Values

Adafruit introduces a simplified method of using the FSRs without doing the calibration as Tekscan does. Based on approximate values using 5V output and a 10kΩ pull-down resistor they present the values listed in the table of Figure 3.14. The calculations are based on the voltage equation

$$V_0 = V_{cc} \cdot \frac{R}{R + FSR} \tag{3.1}$$

[4] A lot of electronic devices have their respective Arduino libraries to make it easy to implement with pre-defined commands, visit https://www.arduino.cc/en/reference/libraries

as the voltage is proportional to the inverse of the FSR resistance (Nosonowitz and Ada, 2012). Translating the analog voltage output to Newton requires the following steps.

1) Map the analog voltage from $0-1023$ to $0-5000\text{mV}$. In Arduino the code is simply to use the `map`-function `fsrVoltage = map(fsrReading, 0, 1023, 0, 5000);`

2) With $R = 10\text{k}\Omega$ and $V_{CC} = 5000\text{mV}$

$$V = V_{CC} \cdot \frac{R}{R + FSR} \tag{3.2}$$

$$FSR = \frac{R \cdot (V_{CC} - V)}{V} \tag{3.3}$$

$$G = \frac{1}{R} = \frac{1000000}{FSR} \quad \text{due to m}\Omega \tag{3.4}$$

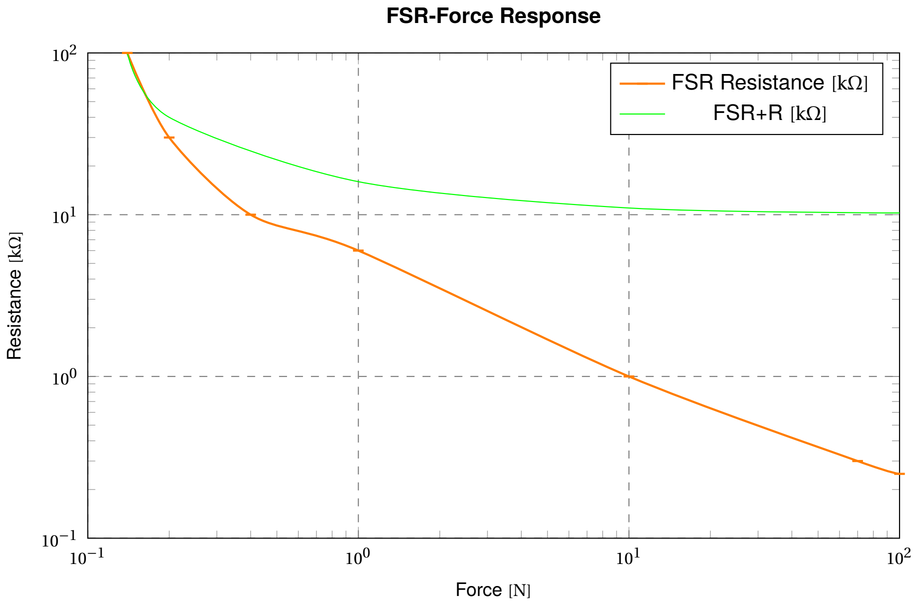
3) By simplifying the FSR-response graph, using two slope values, the force can be calculated by the following formulas.
If $G \leq 1000$

$$F = \frac{G}{80} \quad [\text{mV}] \tag{3.5}$$

If $G > 1000$

$$F = \frac{G - 1000}{30} \quad [\text{mV}] \tag{3.6}$$

If a calibration of the FSR in use is performed, change the slope values in step 3) with obtained values.



Force [N]	FSR Resistance [kΩ]	$FSR+R$ [kΩ]	Current trough $FSR+R$ [mA]	Voltage across R [V]
0	∞	∞	0	0
0.2	30	40	0.13	1.3
1	6	16	0.31	3.1
10	1	11	0.45	4.5
100	0.250	10.25	0.49	4.9

Figure 3.14: Approximate analog voltage based on the sensors force-resistance with a 5V supply and 10kΩ pull-down resistor (Nosonowitz and Ada, 2012)

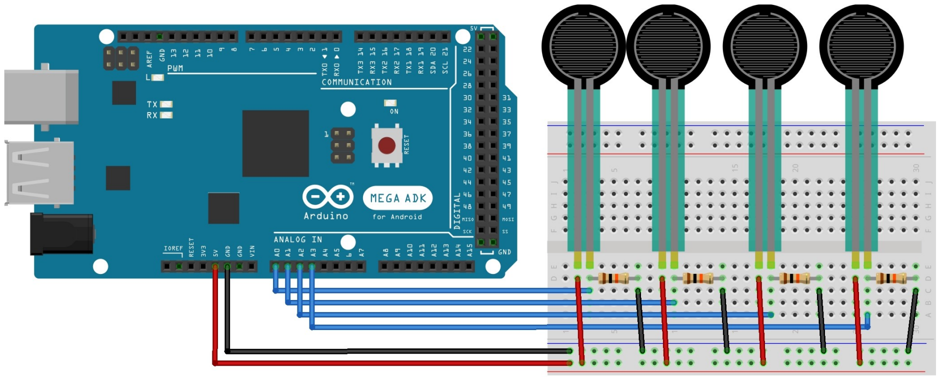


Figure 3.15: Illustration of how to set up four FSRs to an Arduino with 10 kΩ resistors in between the analog inputs and ground.

Connecting Force Sensitive Resistors to an Arduino

Connecting FSRs to an Arduino is fairly easy. As I have illustrated in Figure 3.15, each FSR only requires one analog input on the Arduino, whereas the load cells require two. A $10k\Omega$ resistor should be connected between the FSR, analog input and ground to stabilize the readings.

In this project FSRs are used to get a measurement of how forces are acting on the residual limb when torsion is put on the socket.

Equation variables and units used in chapter 3.3.2

F	Force	**A**	Ampere
FSR	Resistance of FSR	**R**	Resistance
G	Capacitance	**Ω**	Ohm
V	Voltage	**m**	Milli
V_{CC}	Voltage - Power Supply		

4

Case Investigation: Define Re-Define the Problem

Starting out with *eliminating rotation* as the initial problem to solve,the problem space rapidly diverged with the interviews and surrounding research. This chapter will present the results of needfinding in the *fuzzy front end*[1] of this project, defining the direction of developing a prototype. The project has been conducted by the approaches of the research and prototyping laboratory TrollLABS[2]. Based in TrollLABS the project has mainly gone through three stages with three approaches (Steinert, 2017):

1) Rethink the PROBLEM	Design Thinking
2) Find the best SOLUTION	Agile / SCRUM
3) Implement alpha PROTOTYPE	Lean Product Development

4.1 Interviews

A thorough research on the topic *prosthetics for lower limb amputations* was needed to carry out this medical-mechanical product development project. The preliminary research raised important questions of which to ask the stakeholders. Three interviews were conducted to gain better understanding of the problem and allowing the stakeholder (TOV) to share their thoughts. First, a startup meeting with the thigh prosthetics group at TOV was held, allowing them to present the problem of which the master thesis project is based upon. Then, a user interview was conducted with a relatively active amputee which attended the alpine ski day as described in the introduction. Lastly, a shorter meetup at TOV and some email correspondence with Modalsli made a basis for further development.

4.1.1 Meeting with the Prosthetic Team

In the early phase of the project a meeting gathering a group of prosthetists and physiotherapists explaining their experiences with the lower limb amputees and prostheses. With the initial problem *"No ski control due to rotation in the socket"*, the questions developed into an investigation of how the socket works and what the reason for gliding might be. Having the meeting at TOV showed advantages when it came to the explanation of issues that had been shed light upon. As seen in Figure 4.5, a couple of sockets, liners and a locking system were put on the table. This gave us the opportunity to go deeper into the topics of what works and what does not.

In addition to rotation, sweat inside the liner was reported as a problem making the inner silicon wet and slippery, resulting in chafing. Chafing also occurs frequently around the brim area. Volume fluctuation throughout the day, where the volume of the stump reduces in size from the morning to the evening, is a problem for many patients that leads to wishes to have separate a morning- and an evening prosthesis. The following table sums up the challenges of the initial prosthesis solution.

[1] Fuzzy front end (FFE) refers to the early stages of the innovation process in new product development (NPD) (Reinertsen and Smith (1991), Elverum et al. (2014))

[2] TrollLABS, NTNU Trondheim. www.trolllabs.com

Category	Problem	TOV/User Solution
Mounting	Rotational gliding (Figure 4.6b): 1) Socket-Liner and 2) Liner-Stump (skin)	1) KISS-system4.1.1 2) No solution with NU-FlexSIV. Ischial sockets lock in the ischial tuberosity (Figure 2.7), hence lock rotation
	Wear & Tear On clothes	Avoid KISS-system
Skin response	Sweat	Possibility of making small puncture pattern in the silicone, but not in the vacuum part of the socket
	Chafing / blisters	Adjusting the brim to reduce proximal chafing
Dynamic body	Volume fluctuation	In case of too big, a liner socks can be added.
	Muscle contraction	NU-FlexSIV is supposed to comprehend this issue by the way of construction. This is not sufficient for all patients
One socket	... to work in all situations	Made for "everyday life", medium activity. Downside: minimal adjustment

It is worth pointing out that there are large individual varieties in the amount of problems each patient has with the above-mentioned challenges, if experiencing them at all. And this research has most likely not identified all problems each individual transfemoral prosthesis user experience.

"No CASE 1 Slip" with KISS Lanyard System

One apparent solution to fix the gliding of the socket is the lanyard system from KISS Technology Mantelmacher (2011)[3]. It is made to reduce distal pistoning and rotation movements. KISS lanyard system main components are a drop-like reinforcement pad with Velcro on one side to stick to the liner (Figure 4.5 #23), a lanyard (Figure 4.5 #4,6) which goes down to the bottom of the liner and a circular button (Figure 4.5 #24). The button, as seen in Figure 4.5.3), pops out of the hole (hidden behind the liner) in Figure 4.5.4). From the patent document comes Figure 4.1, which illustrates the initial system.

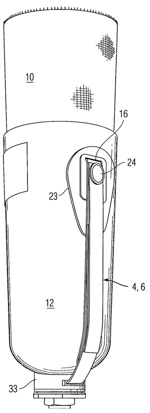


Figure 4.1: KISS-Lanyard system Mantelmacher (2011)

Although this can seem like a prominent solution to the problem, TOV explains downsides including

- Excessive wear and tear of clothes due to the Velcro on the lanyard (big problem)
- Additional wear of the liner due to improper placement and abrasion (liners are expensive)
- Difficulties of attaching the system (in particular the drop) in a proper way on the liner
- The drop has a tendency to curl up (pinching and uncomfortable)
- The lanyard with Velcro makes a lot of excessive material on the outside
- This system only prevents gliding in case 1, socket-liner (Figure 4.6b), as it places a fixed connection in-between these layers. Case 2 sliding, could hypothetically take up more of the rotation, which might lead to discomfort and possibly blisters on the user, although this is not sufficiently tested.

For the mentioned reasons the prosthetic team is not satisfied with the current solution of using the KISS system.

4.1.2 Interview with A Prosthetic User

TOV facilitated a meeting with a TF prosthesis user, for anonymity I call him John. For part of the interview, Modalsli was present and came with some inputs.

John is in his 30s, has a medium physical job and likes to stay active by biking and jogging; which he has managed to continue with also after amputation. John was one of the participants of the ski-day with TOV. He had been skiing a bit pre-morbidity, but not much. John describes the experience in the field as quite terrifying. When talking about how it was to try skiing, he said *"I felt I was drunk or something"* (translated by the author) due to the lack of control. He added *"I had zero control and tried to straighten the leg to gain some control of the ski"* (normally you would have slightly bent legs while skiing). Answering the question if he could see himself ski again, abruptly he

[3] KISS Lanyard System is to be found at kiss-suspension.com and Ottoblock.com

replied *"NO"*. After suggesting use of new equipment which would give him more control, the answer changed to *maybe*. He would rather try snowboarding than skiing if getting back into the slope.

Addressing the challenges enlightened at the TOV startup meeting gave the following answers.

Category/Question	Users Answer	Orthopedic Engineers Answers
Chafing	No big problems with chafing. Sometimes in the brim area.	Some people really have a big problem with this. Some inside the liner, but primarily around the brim.
Socket fit		
Changing throughout the day?	Not much. Use liner sock/ply to fill the socket, worse with donning problem when too tight	Very individual. Some hardly recognize it, while others have big problems with it. Ply is put under the vacuum/silicon-brim (HSE) and up proximally.
Changing from one day to another?	Not really, but with time the circumference of the thigh shrinks. When the socket is too tight and pinching too much, phantom pain appears. Being too tight or not is on a millimeter level.	As above. In addition, women usually have changes corresponding to the menstrual cycle because of the fluid balance in the body
Feels to use a prosthesis	Hard question. The body gets used to it and adapts. When the socket fits good, it feels like the same pressure over the whole socket. To begin with, the skin felt very clammy all the time when using the prosthesis. This improved over time. Now (about 2-3 years later) it is mostly when doing physical activity the stump gets clammy.	
Pressure sensitive areas	Not in particular.	Individual
Appearance (prosthesis)	The appearance does not make a big difference. When they are trying to make the leg look "natural/human", it looks sketchy/unnatural, like the foot on this one - especially on the beach. If I could choose, I would like it to look as much "Ironman" as possible.	
Wear and tear on clothes	No big problems with wear, except from socks on the hard prosthetic foot.	Note that he is not using the KISS-system with his socket

Figure 4.2: Preferred working area based on patient-Johns leg, drawn on a scanned model.

John uses the NU-FlexSIV socket and has a seemingly regular, uncomplicated stump. He started his amputee life with the high, ischial containment socket (2.7), but did not like it because of the design and where the brim hits. It was difficult to sit etc. He quickly went over to the one he uses today and has worked pretty well these ≈ 2 years. On the day of the interview, he came back with a new socket which was too tight - about 1-2 mm too small is enough to make it not function.

Below the socket is the prosthetic knee. His knee is a high-end microprocessor knee, which he is quite happy with and works as it should. He controls the knee via Bluetooth and an app on his smartphone. The knee has settings for most activities, like walking, jogging, biking, swimming etc. The foot blades need to be changed from time to time because of the spring stiffness etc. which affect the gait. He uses the prosthesis regularly throughout the day, but sleeps without it.

4.1.3 Additional Information Retrieved From TOV

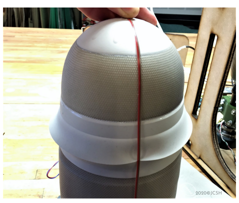


Figure 4.3: Iceross Liner, HSM deflection

TOV is making the prosthetics from outer measurements where fit and slip are the main concerns. Gait analyses and weight of the patients are neglected. The sockets are simply made of the rectified casts, without having a way of testing the strength, fit and slip. Adjustments are done through feedback from the patient.

From what is described in the NU-FlexSIV method (Section 2.3.2), TOV does not use the sealing sleeve as they rather use the liner with HSM to secure

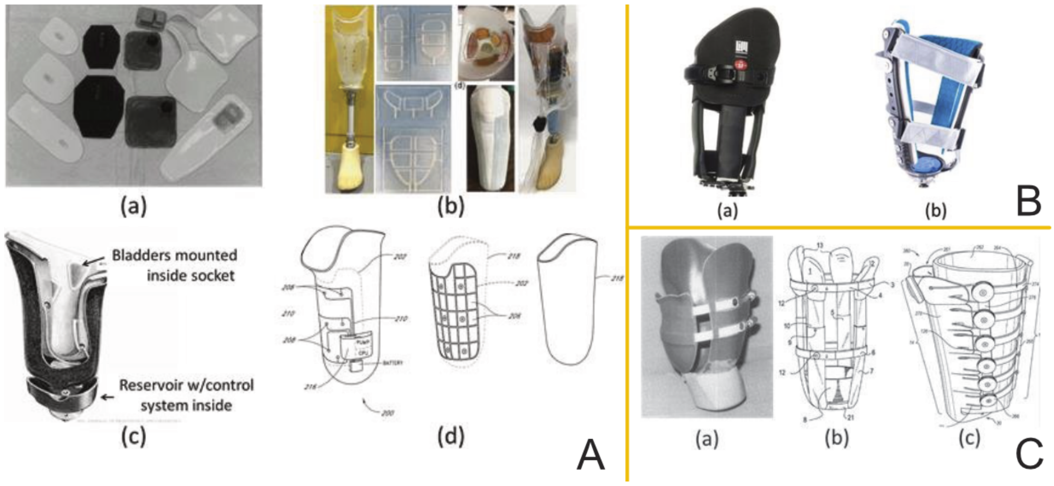


Figure 4.4: Socket technologies investigated by Paterno et al. (2018). A) Inflatable sockets (U.S. Department of Veterans Affair; and American Academy of Orthotists and Prosthetists).
B) The Infinite Socket™ (LIM Innovations) and the Socket-less Socket™ (Martin Bionics).
C) Variable-Volume Sockets (American Academy of Orthotists and Prosthetists)

vacuum in the distal end. On the patient I interviewed, the HSM goes up proximally 12 cm from the distal end (Figure 4.2). One can also mark that the liners used are transfemoral, not transtibial as Fatone and Caldwell (2017b) prefers for transfemoral amputees with their socket. Modalsli informed that the main modifications they do on the socket is changing the flexibility of the white flexible inner layer.

TOV is going through a digitizing process now. They are 3D-scanning the plaster casts to get it in a digital library instead of using hundreds of square meters to store old casts. They are also working towards minimizing the need for plaster casting on the patients, by doing 3D-scanning. As a byproduct, many casts have a designated STL file which opens more product development possibilities for those working with prosthetics and orthotics. For the case of transfemoral amputees, these scans are done with a liner on the stump.

4.1.4 Initial Boundary Conditions

Many engineers have tried to ease the use and comfort of leg prostheses. In Figure 4.4 there is a selection of three concepts, inflatable, open adjustable solution and a variable volume socket. When addressing questions about why it is the NU-FlexSIV socket that is chosen and if any other concepts have been tried, Modalsli responded

> *The prosthetic solution they are using now is widely recognized worldwide, it is one of the "standards". They need a good reason and documentation for changing to another concept. There are many "new and better" solutions which are hyped up by advertisements without being properly tested and documented.*
>
> *Internationally recognized solutions are good for the long term, especially on radical changes. Small tweaks are easier to fix in-house.*
>
> — **Modalsli et al. (2019) (translated by the author)**

For TOV it is further important that they are able to implement solutions in-house. This includes keeping the liner as it is a mass fabricated ordered item and it is rather important for the NU-FlexSIV technology to work properly. Together with the liner and NU-FlexSIV socket, comes the active vacuum and HSE silicon layer which needs to be preserved. This creates some initial boundary conditions. Figure 4.6a illustrates the initial socket construction, with plotted line where TOV suggest not to do modifications. The socket-system should be compliant to

- Body dynamics - rest vs activity
- Blood circulation
- Shrinking, expanding, sweat, cold
- Pressure and trigger points / nerve threshold

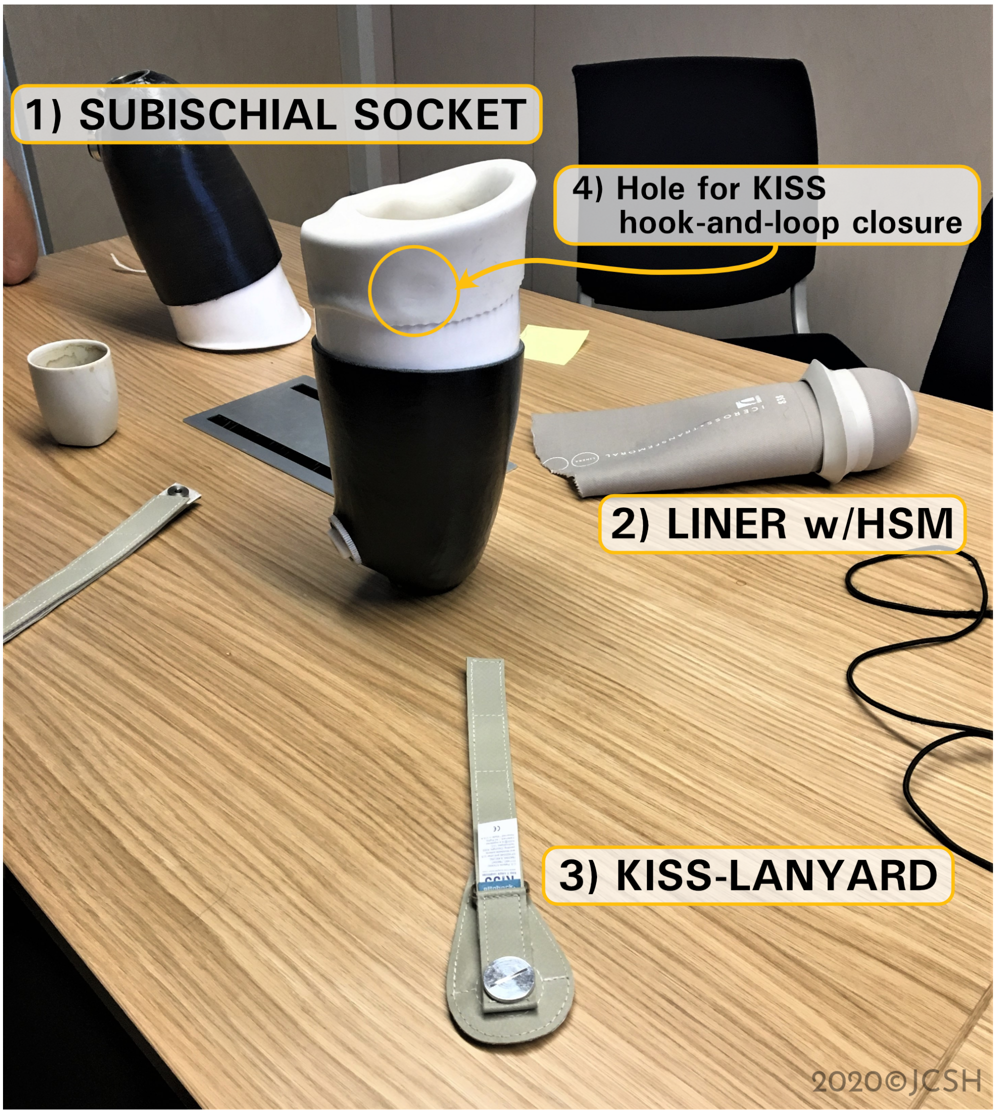
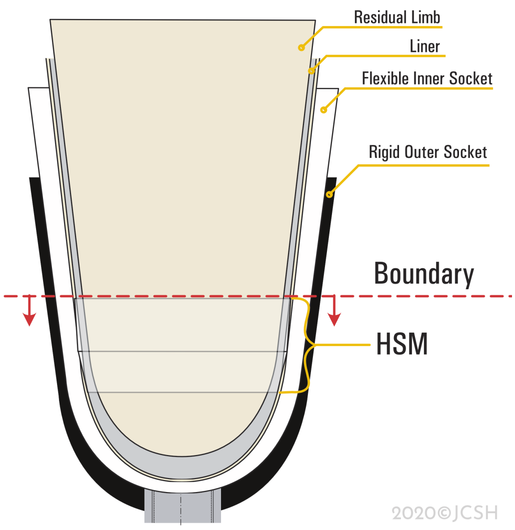
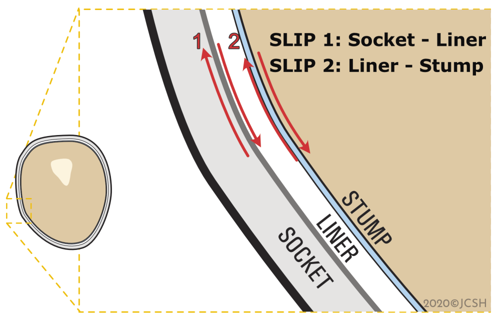


Figure 4.5: Items of interest at the startup meeting at TOV. 1) Two subischial sockets. 2) Standard TF liner with hypobaric sealing membrane (HSM, see 2.3.1). 3) KISS-Lanyard (locking mechanism). 4) Indication of the hole to which the KISS-lock is to be placed.



(a) Development boundaries and rough socket structure including the residual limb, liner, flexible inner socket, rigid outer socket and HSM.



(b) PD: Slip occurrences in the horizontal plane. Case 1: slip between socket and liner. Case 2: slip between liner and stump

Figure 4.6

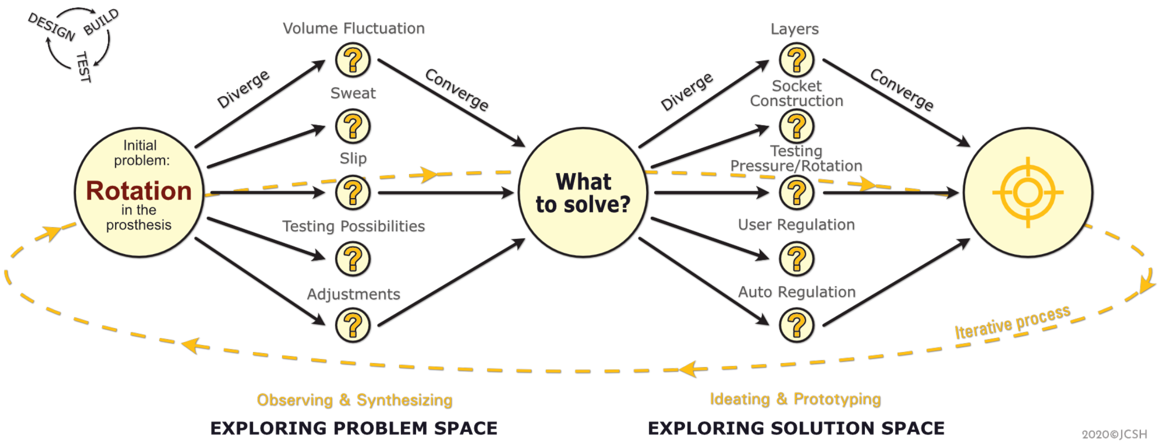


Figure 4.7: A way of illustrating the development process of this project by exploring the problem and solution spaces. For the illustration I have taken inspiration from design thinking models.

4.2 Direction of Development Work

Starting out with "rotation is occurring in the socket when skiing" as the initial issue, the problem space diverged while exploring the case with a design thinking mindset[4]. There might be more to it than just the extra torque brought by the ski in ploughing position that is making the socket-stump-attachment slip. Stump volume fluctuation and sweat might as well be important to address. Adjustment possibilities are missed by the end user, both for different activities and as some experience noticeably size difference in their stump through the day.

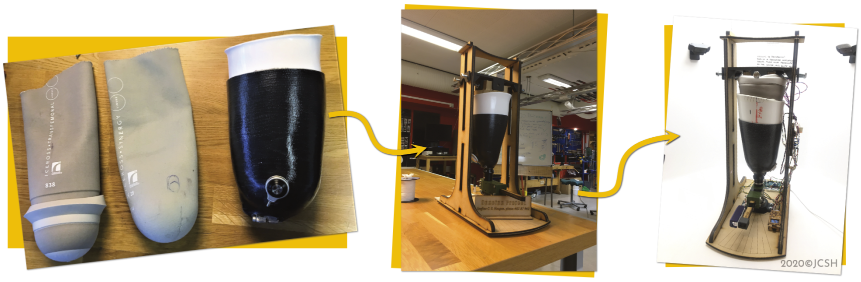
Along with these end-user specific needs, TOV has no user-independent testing possibilities when it comes to fitting and force response on the sockets. Several development possibilities came up while analyzing the situation, the main directions can be categorized into

- Socket Construction
- Layers (add or change initial)
  - Inflatable
  - Soft Robotics
  - Grooves
  - Pattern
  - Material changes, friction
- Adjustable socket or layers (automatic recognition vs. user regulation)
- Test setup for the constructor

As TOV is interested in using this project as part of a larger development project to improve their transfemoral prosthetics solutions, the need for a test rig to use by the constructor and future product developers was favored. When making products to improve the rotation-problem, a way of initially testing the product in a mechanical setup is strongly recommended before presenting it to the end-user. As TOV is reliant of using concepts that are well documented and recognized, getting testable results would make it easier for them to eventually try out new systems.

TOV provided me with two TF prosthetic sockets, two liners (one with and one without HSM) and one socket adapter making it possible to do testing and prototyping on the actual equipment. Along with one of the sockets, they gave me access to a laser scanned model of the patients stump.

[4] With a design thinking mindset, the focus is put on the human values and empathizing with the user to understand the whole picture. Experimentation through prototyping, to visualize, learn, think and find requirements and solutions are some of design thinkings core elements. At innovationtraining.org Eich (2018) compares IDEO's, Standford d.school and PDMAs design thinking mindsets.



5

Exploring Test Rig Possibilities For Transfemoral Sockets by Prototyping

When placing transfemoral amputees on alpine skis in a slope, they experience that the socket is slipping by rotating around the axis of the femur, which gives them no control of the skis. As TOV wish to give their prosthesis users greater control and enjoy being active, they want to improve the socket. When I discovered that TOV have no device for testing the fit of the socket, other than trying it on the amputee, the choice came to explore *how such a testing device can be made* and construct a prototype test rig. This chapter presents the development process of the test rig, in such detail that it can be further developed if desired.

5.1 What Need to be Simulated?

There are a great deal of physics that goes into action in downhill skiing. Simulating a sockets response on the stump to skiing requires extensive simplifications for the time frame available for a master thesis project. What is most important to address and what can be neglected? For this test rig, the main goal is to be able to measure

> *how much the socket is displaced in regards to the stump and liner when a torque is exerted on the socket.*

The stump-liner-socket-adapter is hereby considered to be a closed system, as I have illustrated in Figure 5.1 a. Simplifying the closed socket-liner-stump system even more, neglecting the variable shape and elasticity, the stump can be considered to be a cylindrical cantilever. The socket is press fitted on the stump with a normal force, $\vec{N}$ acting on the system. The rotation from the ski is the torque $\vec{T}$ and the pressure from the press fit is the $\vec{P}$ as illustrated in Figure 5.1 d. The angular displacement which is desired to *reduce or delay*, is illustrated on the figure as $d\theta$ ($d\theta = s/r$ where s is the arch length and r is the radius of the circle).

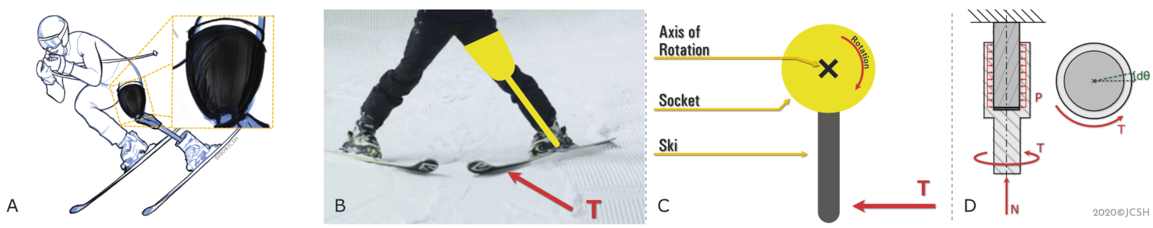


Figure 5.1: Illustrations of the system of forces used as the foundation of the test rig. A) The socket-liner-stump as a closed system. B) Simplification of the ploughing indicating a force acting on the ski which translates to a torque on the socket. C) Simplified illustration of the socket-ski system in the horizontal plane. D) Force diagram of the socket-stump as a simplified cylindrical cantilever system connected through a press fit.

5.2 Test Rig Structure

To achieve the desired simulations and possibility to test out products in the socket, a physical rig is needed. To be able to use the rig on actual sockets, the obtained socket, liner and stump makes up the core of this rig design. Dimensions of the structural design is based on the stump and socket. This section presents the construction of a test stump and the structural design of the test rig.

5.2.1 Creating a Stump Model

The prosthetic socket needs to be filled with some kind of replacement of the stump. All stumps are unique and so are the sockets. The stump needs to be built as a replica of its user. A human thigh is not a solid uniform mass, so how should the stump be made? Considerations regarding stump anatomy in the prototyping of the stump included

- Bone (femur) versus soft tissue
- Muscles and ligaments
- Force distribution within tissues
- Skin layer(s)
- Subcutaneous tissue/fat
- Skin friction (dry vs moist)
- Shape

Two of the stump design ideas are illustrated in Figure 5.2. In design *A* the bone structure is taken into consideration, along with a larger soft tissue layer and two skin layers. In this example the muscle and ligament structure is only taken into some considerations as the bone and the muscle/ligament mass is differentiated. Design *B* shows a greater simplification, where the core of the stump is a solid hard mass with a soft tissue layer and an outer layer to imitate skin friction.

Stump Design A		Stump Design B	
Femur & hard tissue	Wood/MDF milled/laser cut femur and top bracket support	Core	Wood milled according to scanned stump
Soft tissue	Cast foam/silicone (match shape)	Soft tissue	Foam/Silicone/Rubber
Skin layer (hardness)	Silicone	Outer layer	Leather/polyurethane
Skin layer (friction)	Leather/polyurethane		

Realizing that it could be a master project by itself to simulate a stump leg if making it more realistic, with various density and surface friction, the simplified stump leg is milled out from a block of MDF (medium-density fiberboard) . MDF is easy to work with and standard inventory in the MTP workshop, so no delivery time was needed. A wooden stump would be rigid for the job and convenient to make changes to when needed. TOV provided a scan of the patient's leg which was recorded in a standing position with the socket on to be able to make a replica of the leg. This section concentrates on how the wooden stump model was made.

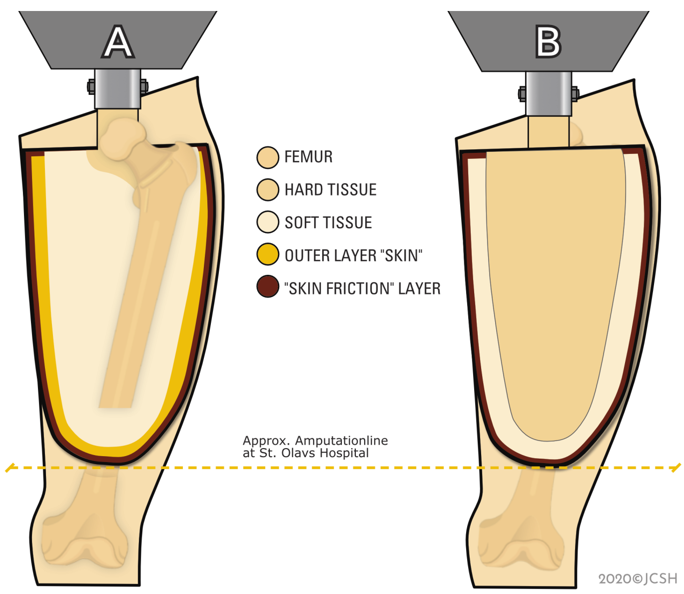


Figure 5.2: Two stump construction directions. Design A consists of four material layers: a hard base representing the femur, soft tissue representing i.e. muscles and ligaments, outer layer representing softer subcutaneous tissue and lastly an outer "skin" layer which could be made of i.e. artificial leather with friction close to the skin. Design B is a simplified version with only three layers: a hard base, a softer tissue layer and lastly an outer "skin" layer.

Figure 5.3: Original scan of the stump corresponding to the prosthesis retrieved from TOV. On the left is the top and bottom view, L: lateral, A: anterior, M: medial and P: posterior side.

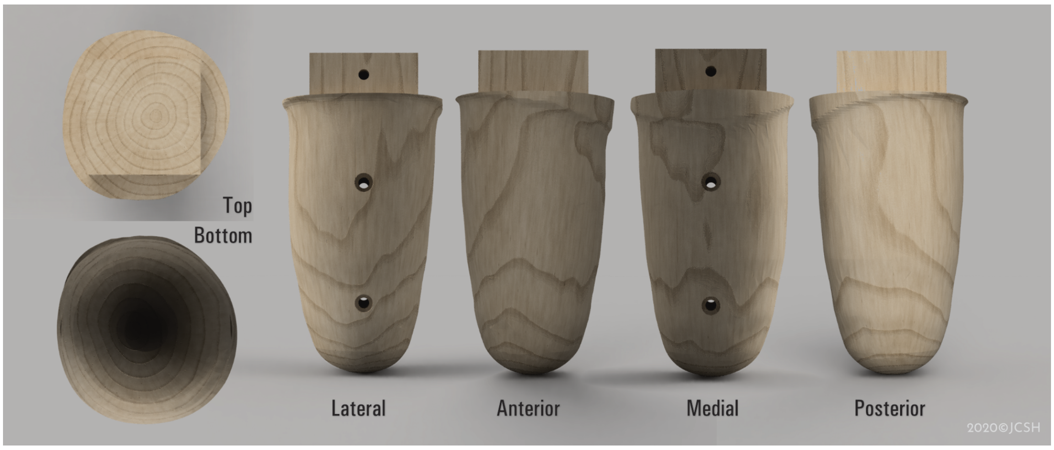


Figure 5.4: CAD Model of the stump after modification to remove the thickness of the liner from the stump. Views corresponding to Figure 5.3

Figure 5.5: Stump model divided for milling. The two first were milled together in the first round and the two last in second round.

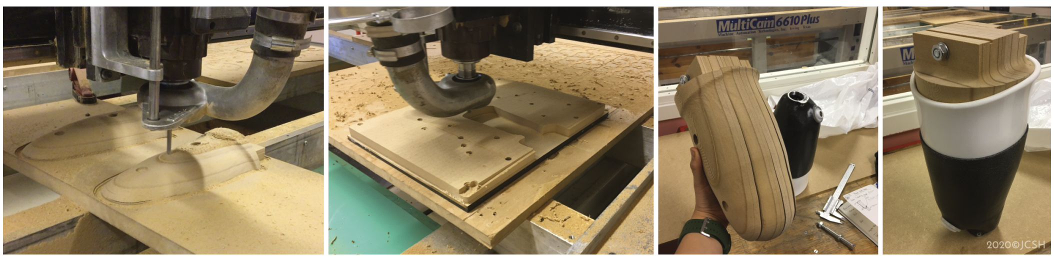


Figure 5.6: Pictures from the milling of the wooden stump model.

Manipulating The Laser Scanned STL-file

TOV provided the original scan of the patient's stump corresponding the socket I had at my disposal. The patient's stump was scanned with a *Creaform GoScan! 50* and saved to a STL-file[1]. Figure reffig:scanstump shows the original scan of the stump retrieved from TOV.

From Chapter 2.3.3 we know that the casting of the stump is done with a casting liner over the stump. To make a model of the stump that would fit in the socket, the liner needs to be subtracted. Modalsli sent the product catalog for the Iceross Seal-In liner along with the STL-file (see Appendix B.2) as the one used for the scan. This liner has a varying thickness, fabricated to be between 2.5 and 14 mm when at rest. Although Figure 5.3 does not indicate any HSM-brim on the model, I went by the assumption it is a socket with these given measurements as I have illustrated on Figure 5.8. The coordinates of the thickness is approximated based on the appearance from the documentation of the liner. The liners change in thickness due to strain when in use on a stump is neglected in both the measurements and when manipulating the STL-model.

For the computer aided design (CAD) a combination of Siemens NX12[2] and Autodesk Fusion 360[3] were used in order to manipulate the STL-model. SOLIDWORKS®2019[4] was also looked into, but not used extensively. Ideally, the CAD software would have some function which could make a variable reduction in size of a meshed non-consistent model like this. Today, this does not appear to exist in NTNUs available CAD software. To make the stump covered with the liner fit inside the socket, the main manipulation steps were as follows:

Modifying the stump model

1. Import STL file
2. Turn off history mode for more mesh options
3. Mesh body from STL *(Modify - Mesh - Mesh to BRep)*
4. Remove unnecessary elements (e.g. proximal of the brim)
5. Smooth surface (selected areas)
6. Re-mesh to downsize the number of mesh points (decrease the accuracy, hence speeding up following processes)
7. Divide the model by horizontal planes into four bodies *(simplification of thickness variation)*
8. Scale each body non-uniform in X and Y direction corresponding to radius reduction (Figure 5.8) which make a staircase look. The distal end body is scaled in X, Y and Z direction due to the 14 mm thick bottom of the liner.
9. Combine bodies by stitching surfaces
10. Fillet and smooth mesh commands were used to reduce the steps between the four previous bodies resulting in the surface in Figure 5.9.
11. Re-mesh to get an acceptable surface

A thick square base was joined onto the proximal end of the stump in the CAD-model to work as a fastening bracket in the test-rig. The shape also ease the post-processing work giving one stable base on a model with otherwise rounded edges. Three through holes were made for mounting.

Figure 5.7: One of the laser scanner TOV uses for development and archiving casts is a Creaform GoScan! 50, an industrial 3D laser scanner.

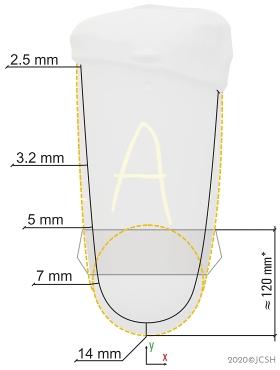


Figure 5.8: Thickness measurements gathered from the liner product documentation and (*) measured from distal end up to above the HSM brim on the patient which stump is used (section 4.1.2).

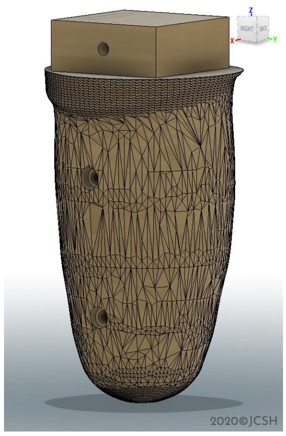


Figure 5.9: Model of the stump after reducing size compliant to the liner.

[1] Stereolithography (STL) is a file format used for 3D-CAD models, https://en.wikipedia.org/wiki/STL_(file_format)
[2] Siemens NX 12: https://www.plm.automation.siemens.com/global/en/products/nx/
[3] Autodesk Fusion 360: https://www.autodesk.com/products/fusion-360
[4] SOLIDWORKS®2019: https://www.solidworks.com/category/3d-cad

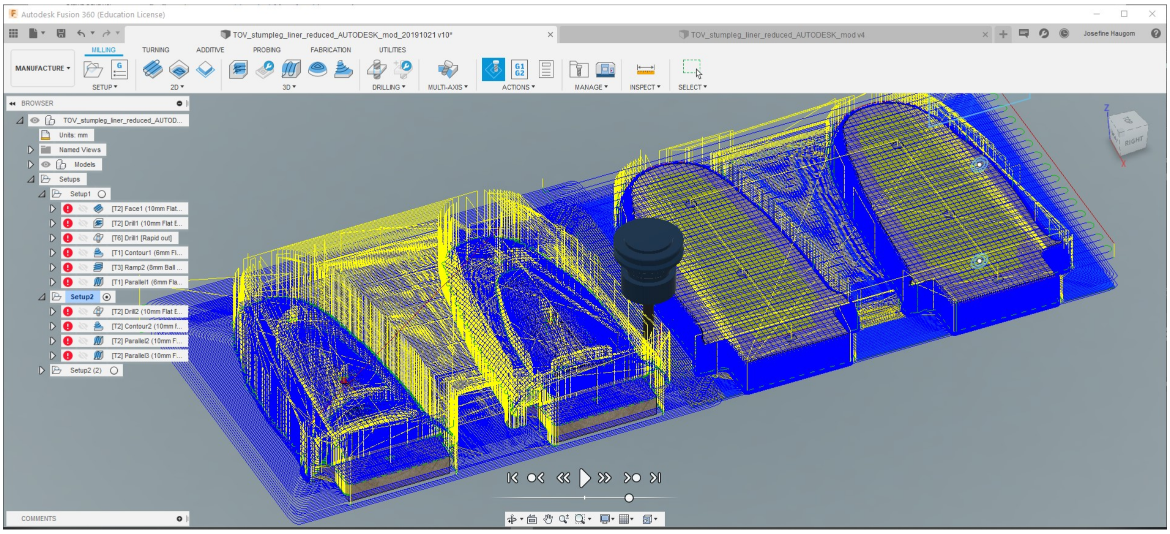


Figure 5.10: Screen shot of the wooden stump milling toolpath simulation in Autodesk Fusion360.

Milling out the stump in MDF

To make the 3D model into a physical workable 1:1 size stump, blocks of MDF were milled to the shape of the model. MDF boards of 3-12 mm thickness were glued together to create blocks measuring 50x330x400 mm and 30x330x400 mm. The machine used for the milling was the *Multicam 6610 Plus 3-axes CNC router* at the workshop at MTP, with table size of 3000x2000 mm and a max working height of 90mm. Max workpiece height depends on the length of the milling tool.

The stump measures approximately 165 mm at the widest point and height of about 305mm. For the workpiece to be compliant to the machine limits, the stump was divided into four parts vertically as shown in Figure 5.5. The two inner pieces have a matching height and were built on the same block and toolpath. The two outer pieces were built on a second block.

Fusion360 has a manufacturing environment which can generate a G-code to run the CNC machine with[5]. With Fusion360s code generator, also comes a simulation option where you can see the route/toolpath of the program, as shown in Figure 5.10. Some edits were made to the G-code output from Fusion360, as the machine needed some adjustments while working.

Toolpaths used for the mid-section block

- 2D Face (Ø10 mm flat endmill)
- 3D Adaptive Clearing (rough milling)
- Drill Rapid Out (6x holes)
- 3D Contour (rough milling, Ø8 mm)
- 3D Ramp (finishing)
- 3D Parallel (finishing)

Toolpaths used for the outer section block

- Drill Rapid Out (Ø10 mm)
- 3D Contour (Ø10 mm flat endmill, rough milling)
- 3D Parallel (along x-axis, finishing)
- 3D Parallel (along y-axis, finishing)

The result of the milling process is shown in Figure 5.6. The wood stump fits well in the socket. However, for the wood-liner-socket system to be assembled, there was a need for further reduction in size of the stump. A few rounds of sandpaper grinding were done in order to fit all together.

We will get back to the stump later, now lets move on - make a section transition here?

[5]G-code is a programming language which is used to control Computer Numerical Control (CNC) Machines. Mach4 is a CNC motion control software which can be used to run e.g. the Multicam 6610 Plus. https://www.machsupport.com/wp-content/uploads/2014/05/Mach4%20Mill%20GCode%20Manual.pdf

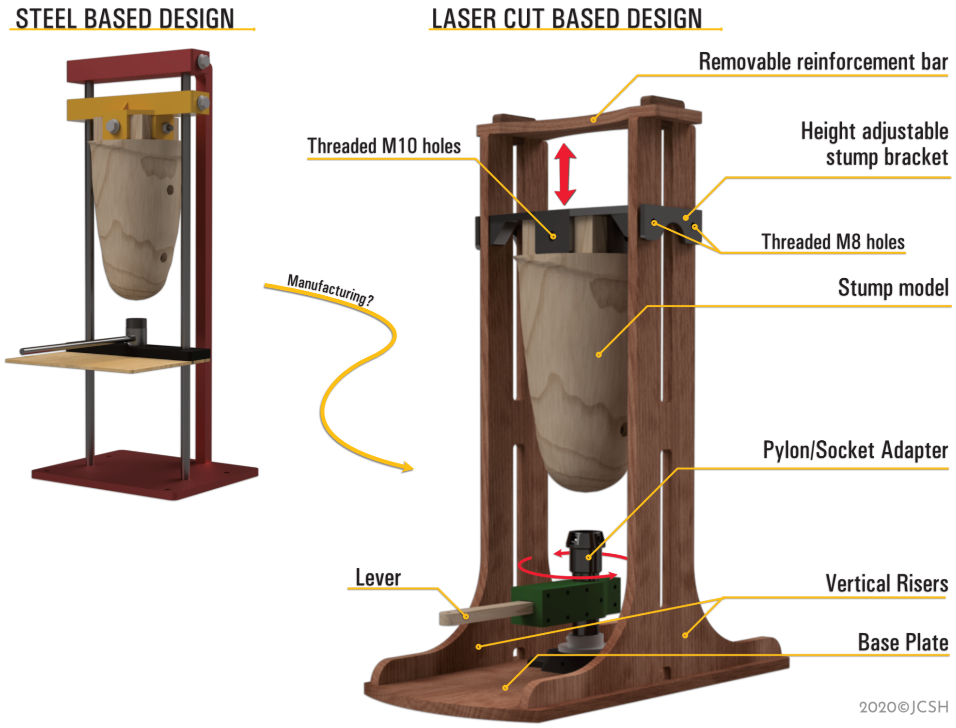


Figure 5.11: Structural design of the test rig. One of the first design ideas is based on a steel structure. The final design is based on manufacturing method for the structure, using TrollLABS laser cutter.

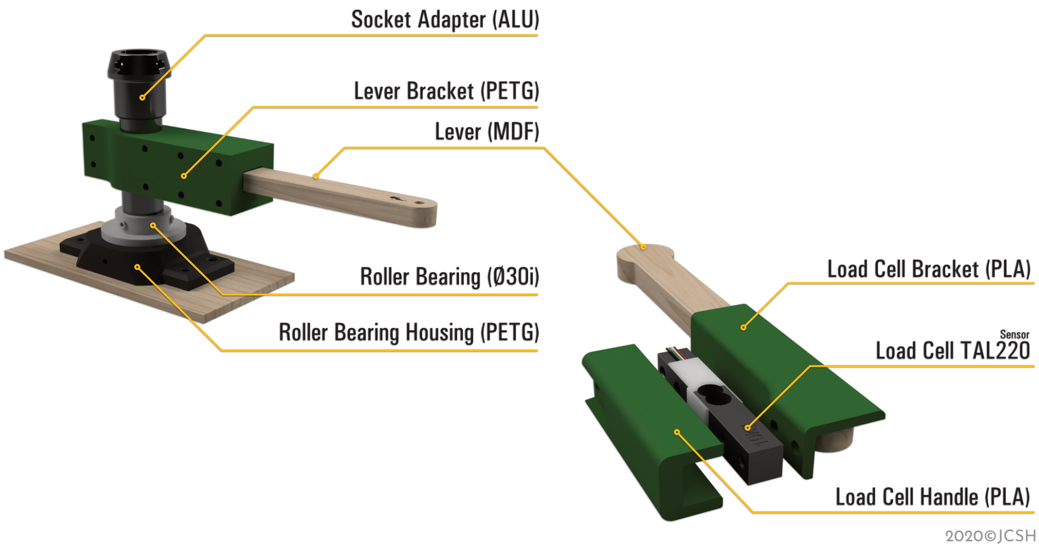


Figure 5.12: CAD model of the test rig lever and bottom bracket including the roller bearing which the pylon and socket adapter is secured in. On the right is an exploded view of the load cell support (excluding screws for mounting).

5.2.2 Designing a Test Rig Structure

Moving from the core of the test rig to the support structure, this section will address the development of the frame of the test rig. In this process, function, strength, materials and manufacturing methods were important considerations. Inspiration was taken from lathes and machines used for torsion testing. The stump need to be secured against following the rotation of the socket when torque is exerted. As well, the stump and socket need to be able to demount relatively easy. Two of the design ideas are illustrated in Figure 5.11. The early design is based on using steel structures. The red parts are welded together to form the support structure. Two steel pipes mounted in the red structure, make rails for the movable parts. One bracket (yellow) secures the wooden stump and the lower structure is where the socket would be mounted on the pylon containing a lever for torque exertion.

As a rapid prototype lab, at TrollLABS students are encouraged to build and test fast, often to verify or discard concepts early in the project. Using rapid prototyping in this test rig, suggest moving away from welded steel structures in the early stages of development due to the processing time and material cost. A laser cut based design was constructed to test if it is sufficient for the job. The structure is made with two dimensional components. The thickness varies on the material and can be stacked together if more strength is needed. It would also be possible to upgrade the strength by cutting out the same structure in for instance metal.

Figure 5.13: PSM Adapter from Fillauer (2020)

The test rig prototype has a structure based on double layer of 6 mm MDF plates. The MDF plates are glued together with regular wooden glue in all parts except the top reinforcement bar. Parts requiring more advanced shape were additive manufactured in Polyethylene Terephthalate Glycol-modified (PETG) or Polylactic Acid (PLA) with Prusa i3 MK3 3D printers. The 3D-printed parts include the stump bracket, which secures the wooden stump to the rig structure; the housing for the roller bearing, which is mounted on the base plate of the rig; and the attachment between the lever and the socket adapter. The 3D-printed parts are fastened in the assembly using machine screws.

Visible at the distal end of the socket is a threaded *locking ring*. To connect the socket to the lever and rig structure, one idea was to make a pylon-lever structure with threads matching the locking ring. Reaching out to TOV, Eivind provided me with an adapter which they use for the sockets. Doing some research on the Fillauers (the manufacturer) website, I discovered what seems to be the whole adapter system (Figure 5.13). The visible parts of the adapter and the socket from TOV resembles *Fillauer's Pyramid Slide Male (PSM) Modular System*[6]. The PSM system makes it possible to adjust the angle between the center axis of the locking ring and the pylon. Using the adapter-system made it easy to get the horizontal plane of the stump bracket perpendicular to the pylon. In the test rig, the adapter is fixed in a roller bearing to only allow rotation of the pylon-socket-system, minimizing reaction forces by friction.

Figure 5.14: The test rig structure, before sensor implementation

[6]Fillauer Pyramid Adapter System can be found at http://fillauer.eu/prosthetics-lower/alignment/psm. The locking ring can be attached to a *pyramid receiver pylon* by a *threaded pyramid adapter*

5.3 Measuring Forces Applied to the System

Prototypes are used for four purposes: learning, communication, integration and milestones, Ulrich and Eppinger (2007) suggest. Prototypes can serve as learning tools when used to answer questions such as if it will work and how will it meet the customer's needs. Houde and Hill (1997) state that prototypes provide the means for examining design problems and evaluating solution. They suggest a model representing a three dimensional space which corresponds to what prototypes prototype (Figure 5.15). Through the process of applying sensors to the test rig, several prototypes were made for learning purposes. Dividing the rig into sections rather than testing each component directly in the test rig gave great advantages for developing the system and exploring new ideas.

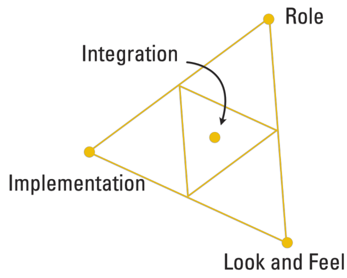


Figure 5.15: Houde and Hill (1997)s four principal categories of prototypes.

> *Given a design problem (of any scope or size), designers can use the model to separate design issues into three classes of questions which frequently demand different approaches to prototyping. Implementation usually requires a working system to be built; look and feel requires the concrete user experience to be simulated or actually created; role requires the context of the artifact's use to be established.*
>
> **— Houde and Hill (1997)**

This section presents prototyping with the goal to make a force measurement system in the rig. Measuring and registering the forces applied to the system along with gathering the systems response are required in a functional test setup. During the project the aim was to explore technology and test out solutions, working a lot in the implementation area of the Houde and Hill model. Each measurement segment of the test rig will be presented consecutively. Mainly, this consists of the force on the lever, an inner axial force and the surface forces on the stump.

5.3.1 Vertical Force on the Lever

A lever is mounted on the pylon, which the socket is fastened on, to be able to initiate rotation on the socket around the stump. The force both need to be applied to the system *and* recorded for the rig to serve as a measuring device. Using a fixed length on the lever (r), the torque ($\vec{T}$) can easily be calculated by knowing the vertical force ($\vec{F}$) applied at a certain point (Figure 5.16).

$$\vec{T} = r \times \vec{F} \tag{5.1}$$

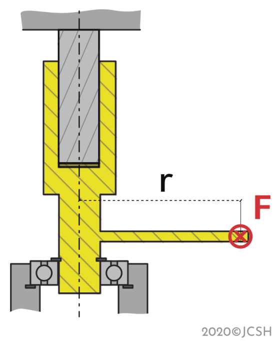
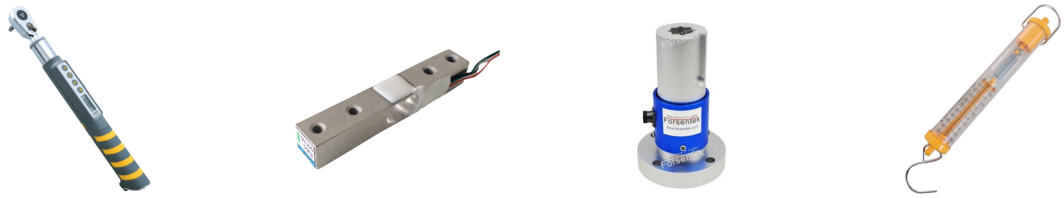


Figure 5.16: Force on the lever creating torque in the socket. The yellow part represents the socket rotating around the stump.

As illustrated in Figure 5.17, four technologies were considered to be used to measure the torque: a) torque wrench, b) load cell, c) torque sensor and d) analogue spring scale. The torque wrench and torque sensor output as the name implies, the torque and would need to be mounted on the rotating axis. Using the load cell or an analogue spring scale, the force $\vec{F}$ from Equation 5.1 would be measured. Considering the cost and availability of the four options along with mounting possibilities, a load cell was chosen. The load cell on the lever is a 10kg TAL220 Straight Bar/Parallel Beam Load Cell (covered in Chapter 3.1.2)[7].

The load cell is mounted on the lever as illustrated in Figure 5.12. Laser cut MDF is for the lever skeleton, which makes it easy to change length and replace it in the lab. The final version of the load cell attachment to the lever is shown in exploded view in Figure 5.12. The attachment contains two 3D-printed parts which is mounted to the load cell with 2xM5 and 2xM4 machine screws. The load cell bracket is formed as a hollow beam which can be slid onto the lever and secured with a M4 screw. The base attachment to the lever is dimensioned to stabilize the load cell when force is

[7] Datasheet of HTC Sensor TAL220 is in Appendix A.3.2



(a) A digital torque wrench from Topeak (canyon.com) **(b)** Strain Gauge Load Cell **(c)** Torque Sensor (forsentek.com) **(d)** Analogue Spring Scale (xump.com)

Figure 5.17: Torque Registration Sensors

applied to the system. The first prototypes had a shorter bracket which was quite unstable. The load cell bracket system is designed to be easy to slide off the lever when calibrating the sensor.

For the purpose of this proof-of-concept prototype, the 10 kg limit is sufficient. The TAL220-series has load cells with capacity up to 200 kg and could be easily changed on the test rig. Scaling up the length (r) of the lever could increase the torque on the socket while keeping the amount of force applied to the load cell unaffected. Going by this last approach could be both cheaper and less time consuming. In the test rig prototype the length (r) between the rotation axis and the mounted load cell handle is 220 mm.

5.3.2 Axial Forces on the Stump

Vacuum suction and friction between the stump and the inner silicon layer of the liner are what keep the stump in the socket. When donning the stump with a HSM liner into the socket, the valve in the bottom lets the air out and secures a vacuum. Trying to doff the stump out of the socket, showed to be a challenge without clamping either the stump or the socket onto a fixed structure and using the button on the valve to release the vacuum. Even when clamping and releasing the vacuum, the socket-liner-stump-system needs some force to take apart.

In addition to the vacuum suction, the weight the patient puts on the socket will in theory contribute to contact forces in the socket. When the socket starts to rotate, contact forces (N) between the stump, liner and socket will tend to build up friction (R) preventing rotation. Frictional force is given by the coefficient of friction μ (depending on the materials which are in contact) and the normal force directed perpendicular to the surface.

$$R = \mu \cdot N \tag{5.2}$$

The force exerted on the socket due to the weight and movement of the patient is a variable which could be interesting to implement to the test rig. Although TOV neglect the weight in their design of sockets (Section 4.1.3), for test rig purposes this measurement could be useful. As with the lever, a load cell would be good for this task, only 10 kg would not be sufficient. Again, using what is available in the lab, a TAS606[8] disk type load cell with a capacity of 200 kg was chosen to measure the preload normal force put on the stump when clamped in the rig.

5.3.3 Implementation Prototype Using Load Cells

In order to explore how the load cells work and how can be utilized in the test rig, the sensors were set up in a separate implementation prototype as shown in Figure 5.18. The setup consist of

- 1 x HTC Sensors Parallel beam load cell TAL220, limit: 10 kg
- 1 x HTC Sensors Button type compression load cell TAS606, limit: 200kg
- 2 x Sparkfun Load Cell Amplifier HX711[9]
- Arduino Mega ADK
- Breadboard
- 2x M5 Machine Screws
- 2x M4 Machine Screws
- 3x M3 Machine Screws
- Laser cut MDF baseplate, load plate and washers on the TAL220
- Connection Wires
- Calibration weights (1-10N)

[8] Datasheet of HTC Sensor TAS606 is in Appendix A.3.3

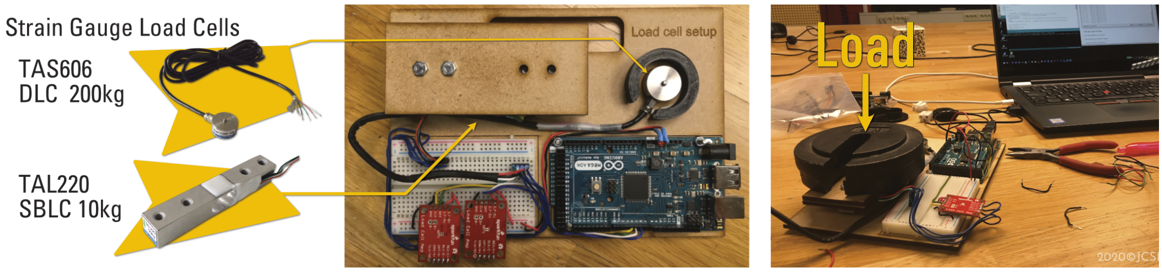


Figure 5.18: Implementation prototype of the torsion and axial force sub-system

The Arduino Mega needs to be programmed using the Arduino programming language for it to execute anything. Arduino uses the name *sketch* for a program in its environment, it is the unit of code that is uploaded to and run on an Arduino board. Working with this project a combination of the open-source Arduino IDE software and Platform IO inside Microsoft's Visual Studio Code was used[10].

As written in Chapter 3.3.1, strain gauge load cells like these two need the HX711 amplifier to work. There are some Arduino libraries to choose from which enables codes to use when writing the sketch to send to the Arduino Mega. While exploring some of the libraries, the HX711_ADC[11] library made most sense to continue with. This is a master library for HX711 24-bit analog to digital converter for weight scales. The library contains five example sketches

HX711_ADC Arduino Library
- Calibration
- Read 1x Load Cell
- Read 1x Load Cell Interrupt Driven
- Read 2x Load Cell
- Testing

These example sketches can be used with the sensor system without change if the sensors are connected to the Arduino in the given inputs, or change the input pin number in the sketch. In the sketch prototyping for this load cell setup inspiration is taken from the examples in HX711_ADC. The sketches used in the prototypes of this project are to be found in Appendix C.

5.4 Force Response on the Stump

Measuring socket angular displacement versus applied torque in the stump-liner-socket system was the initial goal for the test rig. Since the socket is a construction a human being is going to wear comfortably with ability to be active, the designer must balance comfort and forces required to hold the socket in place on the stump. To get a grip on how the human leg is impacted by the socket construction, ways of indicating pinching or pressure force were researched along with how prosthetic engineers have been dealing with this issue up until now.

For many years, prosthetists have used so called *check sockets* when designing and fitting prosthetic sockets. Hamontree and Snelson (1973) define a check socket to be a "socket in a relatively rigid material molded directly over the modified model of the amputation stump and is used to determine the extent of additional modifications that will be required to obtain an optimum fit for the definitive prosthesis." The check sockets are commonly made of transparent thermoplastics, like clear PETG sheets vacuum formed on a stump model as done in the research of Gerschutz et al. (2012). This makes the "fit" of the socket on the stump visible for the engineer which can be used to indicate where the socket needs to be adjusted. Having a transparent socket could to some extent be beneficial for a mechanical test rig, as it would give the analyst a visual image of what is happening inside the socket. However for this test rig and the applicability for the prosthesis developer, making

[10] Arduino IDE: https://www.arduino.cc/en/Main/Software
Platform IO https://platformio.org/
Microsoft Visual Studio Code https://code.visualstudio.com/

[11] Arduino Library HX711_ADC Github link: https://github.com/olkal/HX711_ADC

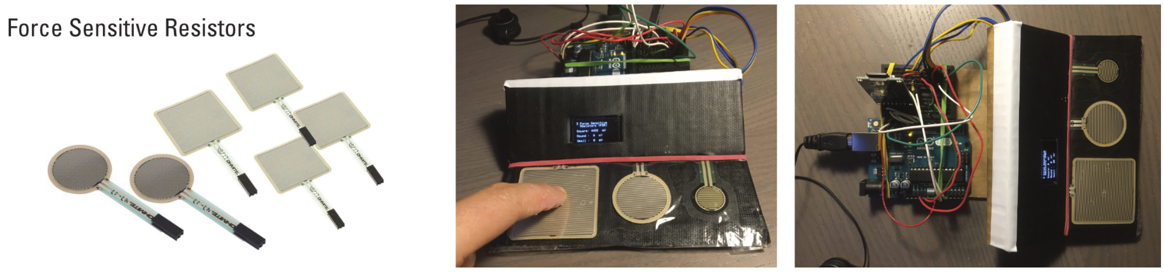


Figure 5.19: Prototype of a FSR array including MicroSD-card module and an OLED display.

a new socket with different material properties (e.g. flexibility and rigidness) than the socket which will be used by the amputee, would not give significant benefits to continue with.

Concluding that the socket will stay non-transparent as it is, another way of measuring contact pressure within the socket could be impression sheets (pressure sensitive paper) as TOV uses to make biomechanical analysis when making foot orthotics. It could hypothetically give a pressure mapping indication. Although, since the rig should simulate what happens in the stump when the socket glides. A need for a dynamic *real-time* analyze method is needed, which brings us to the final technology of this section of the rig - force sensitive resistors.

5.4.1 Implementation Prototyping Using FSRs

Force sensitive/sensing resistors were described in Chapter 3. They are thin, flexible, versatile and often characterized for Human-Machine-Interface. FSRs are used in many studies including medicine and biomechanics, showing various results of the applicability. In a study on using FSRs for pressure monitoring in treatment of chronic venous disorders, Parmar et al. (2017) evaluated the accuracy of five force sensing resistors from different manufacturers. The sensors showed to have different response time, drift and accuracy in static and dynamic testing. Some sensors were better fitted for the dynamic force applications, while others worked better for the static applications.

For proof-of-concept-prototyping, the available FSRs at MTP were used. This includes

- Ohmite FSR01 (square 39.7×39.7mm)
- Ohmite FSR03 (round $\varnothing 25.42$ mm)[12]
- Interlink Electronics FSR 402 ($\varnothing 18.28$mm)[13]

Figure 5.19 shows one of the later prototypes using three FSRs to measure pressure put on their surfaces. The FSRs are connected as described in Section 3.3.2, including a $10\text{k}\Omega$ between the sensor and Arduino. From the initial testing, the response from the FSRs proves to be sufficient for further development implementing them into the test rig.

The FSRs are connected to analogue ports on the Arduino, giving raw output as an integer value between 0 and 1023 representing the voltage passing through the resistor. For the test rig, a choice of which output is desired falls between four options which could be left for the user to decide what is most applicable.

FSR output options

- Voltage (map values 0-1023 to 0-5000 for mV) [V or mV]
- Force in Newton (approximation using the translation described in Section 3.3.2) [N]
- Weight [kg]
- Percentage [%]

For the code to run faster, the best option would clearly be to have as few programming lines as possible as the Arduino. For this implementation prototype, which only consist of FSRs, using the translation to get the voltage output as a force measurement did not cause notably delays. However, putting this in a system with load cells and eventually more components leading to further coding lines could cause unnecessary delays in the system.

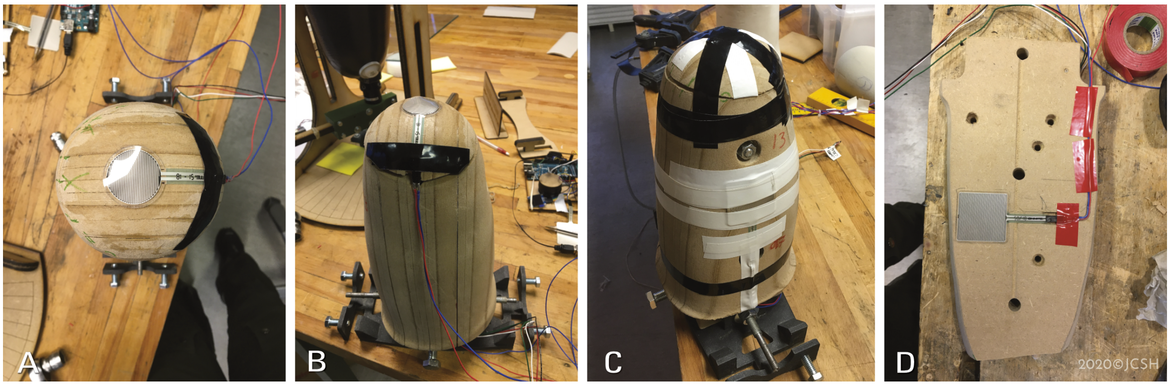


Figure 5.20: Integrating four force sensitive resistors into the stump.

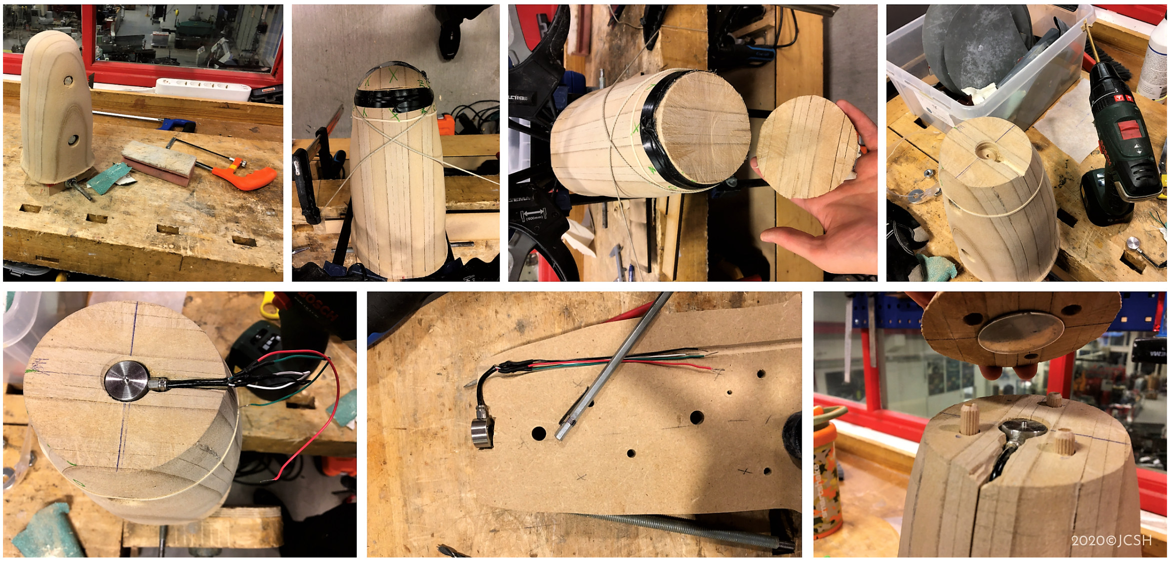


Figure 5.21: The process of integrating a TAS606 disk load cell into the distal end of the stump.

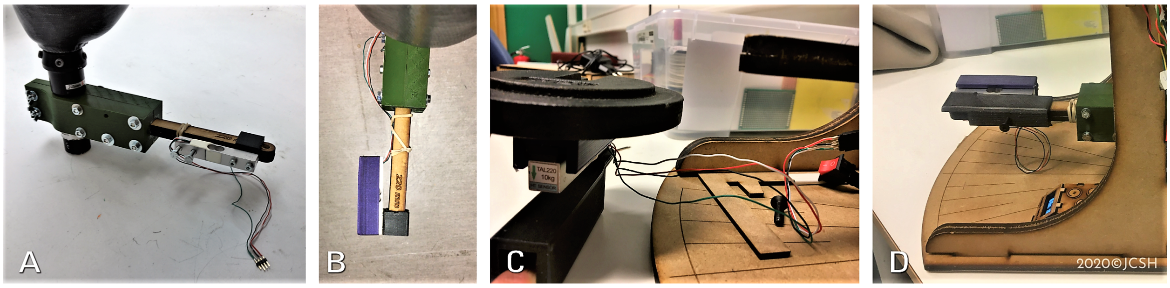


Figure 5.22: Integrating TAL220 load cell onto the test rig lever. a-b) First attachment prototype, c) Calibrating the load cell, d) final load cell attachment onto the lever.

5.5 Integrating Sensors into the Test Rig

Moving from the implementation prototypes and onto the test rig with the two sensor subsystems, new challenges arised. The FSRs are thin and flexible, but also delicate as they can tend to respond different when curved.

> *Interlink Electronics Do be careful if applying FSR devices to curved surfaces. Pre-loading of the device can occur as the two opposed layers are forced into contact by the bending tension. The device will still function, but the dynamic range may be reduced and resistance drift could occur. The degree of curvature over which a FSR can be bent is a function of the size of the active area. The smaller the active area, the less effect a given curvature will have on the FSR's response.*

Using FSR on the surface of the stump, would in most places curve the sensor and could for that reason impact the results. However, when testing the sensors in the rig, even with the curvature, they do output pressure dynamics in a sufficient manner. In this prototyping, the FSRs were placed inside layers of the stump and on the stumps surface as Figure 5.19 shows. Electrical tape, which is slightly ductile, is used to mount FSRs to the surface of the stump. As the adhesiveness of the tape is low for the wood surface of the stump and the FSR, while sufficient to stick to itself, it allows for changing the position of the sensors when needed. Mounting the sensors this way, shows to give some pre-loading on the sensors.

Comprehensive studies can be done regarding mapping stumps for where it would be most convenient to get measurements. Figure 5.23 illustrates general pressure sensitive and pressure tolerant areas of transfemoral stumps. During the interviews with TOV and the amputee, the sensitive areas and pain threshold varies a lot between patients. When using the test rig with a specific patient, sensitive areas could be outlined and used to place the FSRs in the rig.

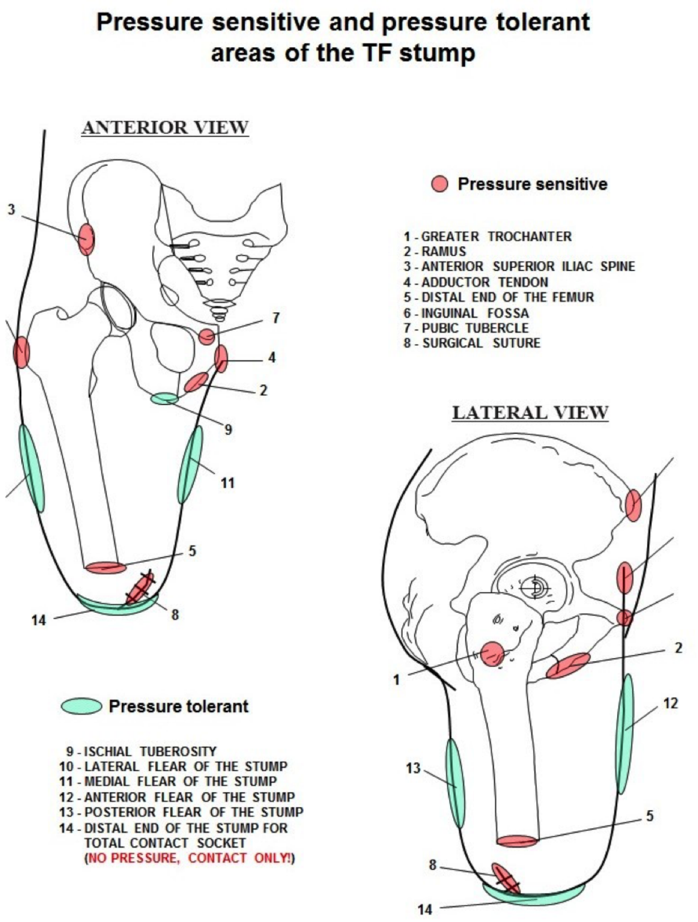


Figure 5.23: General description of pressure sensitive and tolerant areas of transfemoral stumps (van der Stockt and Jackson, 2020)

Using the anatomical findings from the FSR research, the placement and intention of the disk load cell (TAS606 also called button type load cell) changed. It seems that for many transfemoral amputees, the distal end of the stump, where the end of the femur is located, can be quite sensitive. The transfemoral prosthetic sockets are often made to distribute the pressure evenly on the stump and avoid pressure on the distal tip. The DLC got placed inside the distal end of the stump by cutting off the tip and make grooves in the wood as illustrated in Figure 5.21. With the DLC placed inside the socket like this, one could measure preloading at the tip both due to the liner and donning the stump into the socket.

The parallel beam load cell went through some prototyping of the attachment during the project. Figure 5.22 displays the initial attachment which was not sufficiently secured to the lever. Figure 5.22.C shows calibration of the load cell when dismounted from the lever.

5.6 User Experience

The test rig is meant for usage by TOV or future developers of their sockets which might not have a great background with Arduino and programming. The prototyping had a few directions to choose between after sensors were implemented to the system. Either one could go deep into further developing and fine tuning the sensor system in order to increase the accuracy of the readings, new sensors could be tested, the leg could be worked on to become more realistic, or the user experience could be developed. Focusing on user experience, new implementation prototypes were made and implemented to the test rig.

> How would a prosthetic developer like to read the output from the rig? Should it be real-time output or saved to file? Should a computer need to be connected to the rig? These were the main questions when finding a relevant solution for how the measurement output should be.

5.6.1 Measurement Output

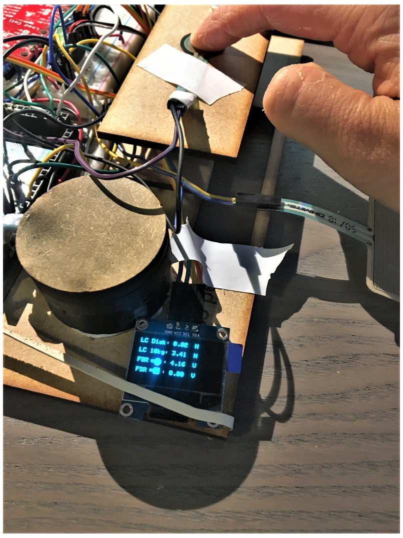
Arduino IDE can display the output in real time either in Serial Monitor or as a plot in Serial Plotter, both cannot be displayed simultaneously. Platforms like Dash[14], Bokeh[15], Realterm[16] and PuTTY[17] were explored to improve the user interface. Both Dash and Bokeh seem prominent, but require an extent of Python programming. Using these platforms would require a computer to be set up with the system. The upside of using these kinds of systems are the amount of information they can give, like displaying plots from the different sensors in each own window wile calculating the slope for instance.

While working with the prototypes an Organic Light-Emitting Diode (OLED) display got connected and added to the sketches, as Figure 5.19 proves. This is a monochrome 0.94" i2c OLED Display with 128x64 pixels, which has a four-wire connection to the Arduino. The OLED became a part of the test rig as it is ready to use straight away if the right sketch is uploaded to the Arduino. The OLED is programmed to guide the user through what the Arduino is doing when running the setup and continue with real-time values from all the connected sensors. The advantage of using this as the output system is its non-reliant on a computer. However, it cannot give too much information at a time and alone it will not save any sensor values to be analyzed later.

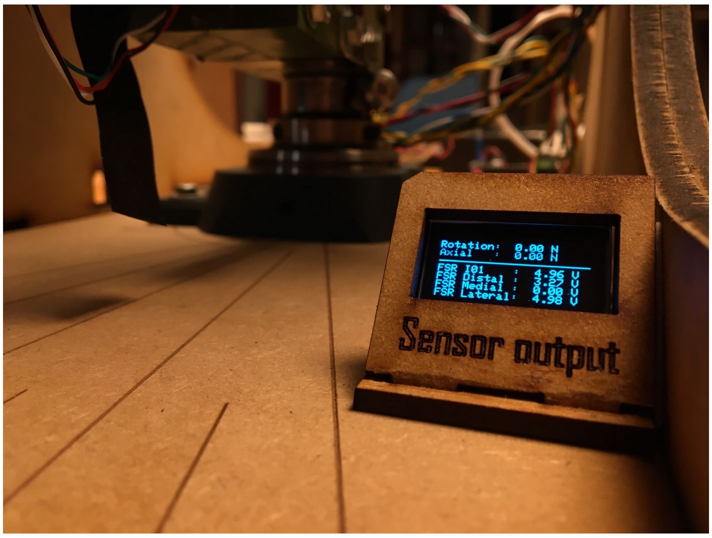
Lastly, an on-board SD-card module was implemented to the system. If the user of the rig lacks experience with Arduino or similar device and cannot connect the rig to a computer, the rig could work by saving the sensor data directly to a SD-card for later to be imported to a computer for analyses. With a datalogger the sensor data can be saved to the SD-card as comma separated values, which can be imported to e.g. Microsoft Excel where the data can be interpreted and plotted if desired.

In the final rig the Arduino IDE (or other terminal reading the serial communication port (COM)), OLED and the SD-card makes up the output channels for the test rig.

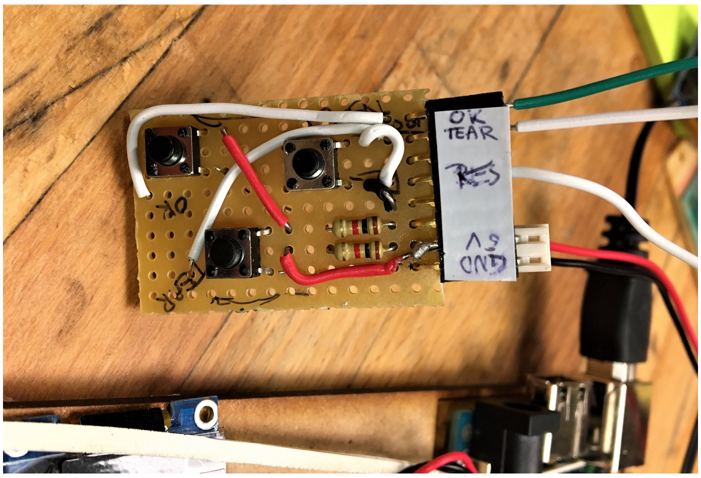
[14] Dash by Plotly: https://plot.ly/dash/
[15] Bokeh: https://bokeh.org/
[16] Realterm: https://realterm.sourceforge.io/
[17] PuTTY: https://www.putty.org/



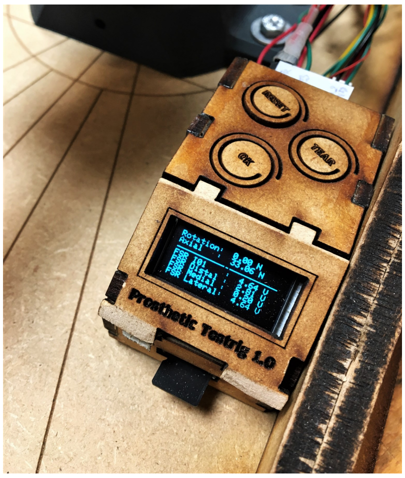
(a) Combining load cells and FSRs in an implementation prototype exploring additions to the mechatronics system



(b) 128x64px 0.94" i2c OLED display showing sensor output



(c) Inside the interface box in (d) pushbuttons are hiding

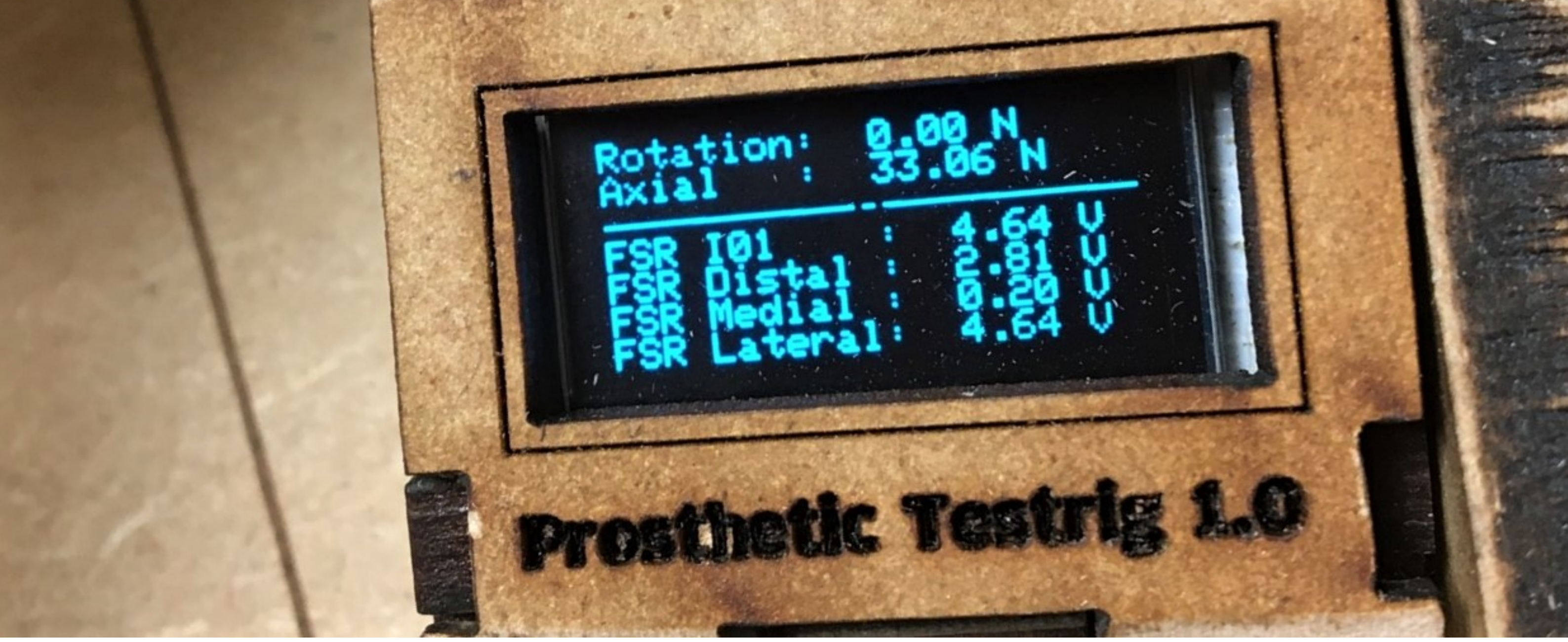


(d) On-rig user interface section including input buttons, 128x64px OLED display showing sensor output and lastly a MicroSD-card slot in the front

5.6.2 User Control

As with kitchen scales, the load cells have a tear function. The function used if you want to measure something inside a bowl without the weight of the bowl included in the number on the screen. In the Arduino HX711 libraries a tear and stabilization sequence is included in the setup for the load cells to work properly. In a larger code, there might be a series of commands which the Arduino must perform before the sensor readings start over again if the program is reset. For that reason, the idea of making interface buttons for the user to among other things, tear the load cells. Using the laser cutter once more, the interface box showed in Figure 5.24d was created. Three button covers are made using a spiral living hinge design. The box also includes the OLED and the microSD-card module.

Figure 5.24: Test Rig Final Version



6

Test Rig Prototype for Transfemoral Prosthetic Sockets

Through research on transfemoral amputation, socket design and exploring ways of how to contribute to enhancing the grip of the NU-FlexSIV socket, a test rig prototype was developed. The rig is built to test how the transfemoral prosthetic socket fits to a model of the stump. With a socket & stump pair, one can test how applying torque is affecting the grip of the socket. In this chapter the final prototype of this master thesis is presented.

6.1 Test Rig Construction

The prosthetic socket test rig is a construction made by rapid prototyping methods. The rig consists of a laser cut frame made in MDF, which enables modifications as thickening the structure using more layers, changing the material or add transversal beams etc. Structure that requires more advanced shape than what would be feasible to do with the laser cutter, was 3D-printed in PLA and PETG[1]. These part is displayed in an exploded view from the CAD software Autodesk Fusion360 in Figure 6.1. Additionally, a depth measurement gauge was laser cut after measurements of the stump. It is built to be attached to the top bracket and reaches down to the distal end of the stump to have an indication of how far into the socket the stump really is.

Through the development process, the rig transformed into a mechatronics system. At the finish line of the project, the test rig contains 6 force sensors, 1 temperature and humidity sensor. Along with these sensors, is an interface system. Two of the sensors are used to measure axial force and torque/force put on the lever when attempting to rotate the socket around the stump. These two sensors have a rigid installment, and would be suggested to keep unchanged in a testing sequence.

Figure 6.1: PD: CAD Test Rig Exploded View

[1] Mechanical drawings of the laser cut and 3D-printed parts can be found in Appendix A.2.

Force sensitive resistors are placed on the stump model. For prototyping purposes, learning how the sensors would react they were placed: 1) between two layers of the MDF, 2) distal end, 3) medial, 4) lateral on the stump as previously shown in Figure 5.20. The four FSRs are connected to the stump using electrical tape to make it easy for the user to change their positions. Adding more FSRs is possible using the current setup, but the Arduino sketch will have to be adjusted.

As both the load cells and the force sensitive resistors can be affected by temperature changes, a *DHT11 Temperature and Humidity Sensor*[2] was added to the system. For this test setup the temperature registration is suggested to be done when calibrating the load cells.

The core electronic setup of the sensor system is illustrated in Figure 6.7. In Appendix C, Arduino Sketches used in the prototypes, including the test rig are attached.
Sketch `20200109_Testrig_All.ino` is the most reliable program which contains all the FSRs, Load Cells and OLED. Datalogging and an Arduino Real Time Clock (RTC) were added to the continuation sketch (Appendix C.1.4). This implementation is inspired by a tutorial at *Random Nerd Tutorials* (https://randomnerdtutorials.com/arduino-temperature-data-logger-with-sd-card-module/). The RTC (DS1307 from Sparkfun) is connected to SCL1 and SDA1 on the Arduino Mega.

As can be seen on the right side of the rig in Figure 6.3, there are a lot of wires connected to a prototyping board. There are also many wires appearing out from the stump, due to all of the sensors. To make disassembly easy, this connection point between the Arduino and the sensors was constructed. The FSRs and Load Cells can be easily detached when needed from the board. The prototype board holds female pin slots, while the wires coming from the sensors are soldered to male pins which are joined, making one connection per sensor.

6.2 Output

The prototype sketches are built to output sensor data as comma separated values to serial monitor and save to a `.txt` file on the SD-card every 2000ms. This is what can be used for analyzing the data in retrospect. To get an idea on what is happening in the rig in real time, a display is placed on the bottom right side of the rig, which should be easy accessible for the tester displaying the following values:

- Rotation: xx [N] *force on lever*
- Axial: xx [N] *axial force in stump*
- FSR I01: xx [V] *the square FSR which is placed in between the stump layers*
- FSR Distal: xx [V] *Distal tip*
- FSR Medial: xx [V] *Medial, approx. mid. axially*
- FSR Lateral: xx [V] *Lateral, approx. mid. axially*

Figure 6.4 shows the Arduino IDE sketching environment and the serial plotter from a test done when integrating FSRs to the dual load cell system (Figure 5.24a). From the example in the HX711_library, the load cell values have a smoothing function:

```
if(millis() > t + 250){
    float a = LoadCell_1.getData();
    float b = LoadCell_2.getData();
    Serial.print(a);
    Serial.pring(",␣");
    Serial.print(b);
    t = millis();       }
```

Adding FSR reading after this if-statement, gave too much delay on the load cells. The values did not correspond to the force applied when the FSR was placed on top of the load cell. Removing the stabilizing if-statement did not do the job, rather, gathering all the sensor inputs into the if-statement gave much better response. As presented in the previous chapter, it is possible to convert the FSR output directly in the Arduino sketch. However, since it uses more power which might be unnecessary from the Arduino, the resulting output is the voltage response.

[2]Adafruit DHT11 Humidity Sensor https://www.adafruit.com/product/386
Arduino library for DHT11 https://github.com/adafruit/DHT-sensor-library

A sensor response based on saved *Comma Separated Values* (CSV) from the SD-card gave the results which are displayed in Figure 6.5. The program had been running from before the stump was donned into the socket to see the affect on the disk load cell and the FSRs. Worth noticing are the preload values which can be seen at the FSRs. In the given graph the values of the FSRs are multiplied by 100, meaning 4.42 V is displayed as 442 for readability in the plot together with the load cells. Two of the load cells were close to reach their maximum value even before doing the *real testing* - rotation testing in the socket-stump system. One can also mark how fluctuating the internal FSR is in this test (grey, FSR1).

Summarized, the sensor output form which the test rig is set up to do are chosen to be possible via

- On rig OLED display
- Serial Plotter and Serial Monitor via a connected computer
- Saved as CSV to microSD-card

The angular displacement of the socket has not been investigated much when it comes to using electronic sensors to read the displacement during this project. Strain gauges were considered, but at this stage analogue measurements have been acceptable. Rather than using electronics, the lines which are engraved like a protractor on the base plate of the rig increment with 5°(10°per full line) give an indication of the displacement. Additionally, markings on the white, flexible part of the socket can give a closer indication of how far the socket slid from its initial position.

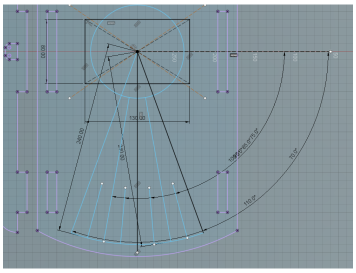


Figure 6.2: Lines are engraved with 5°increment on the base plate of the test rig.

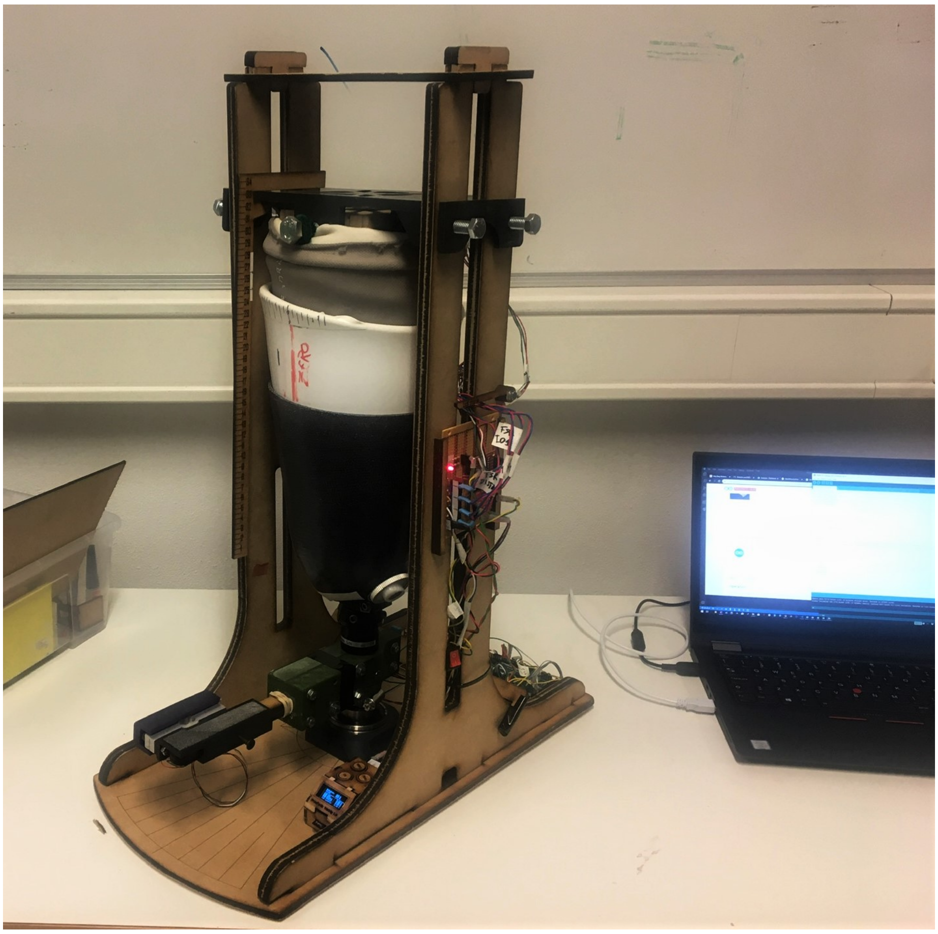


Figure 6.3: The working test rig prototype displays the sensor readings from the four FSRs along with the two load cell outputs. The red light on the right side of the rig indicates that a test is running. Connected to a computer it is possible to follow the plot or the serial monitor showing the values in numbers. Hanging on the bracket left of the socket is a depth measurement gauge which can be used as an indication on how far down into the socket the stump is.

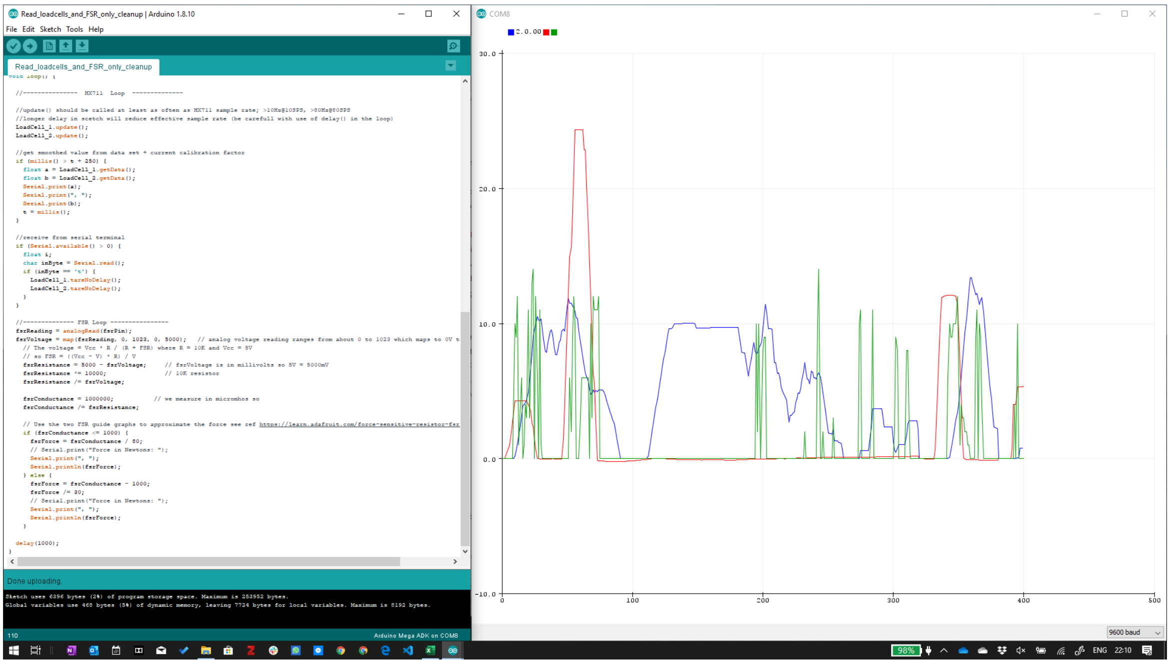
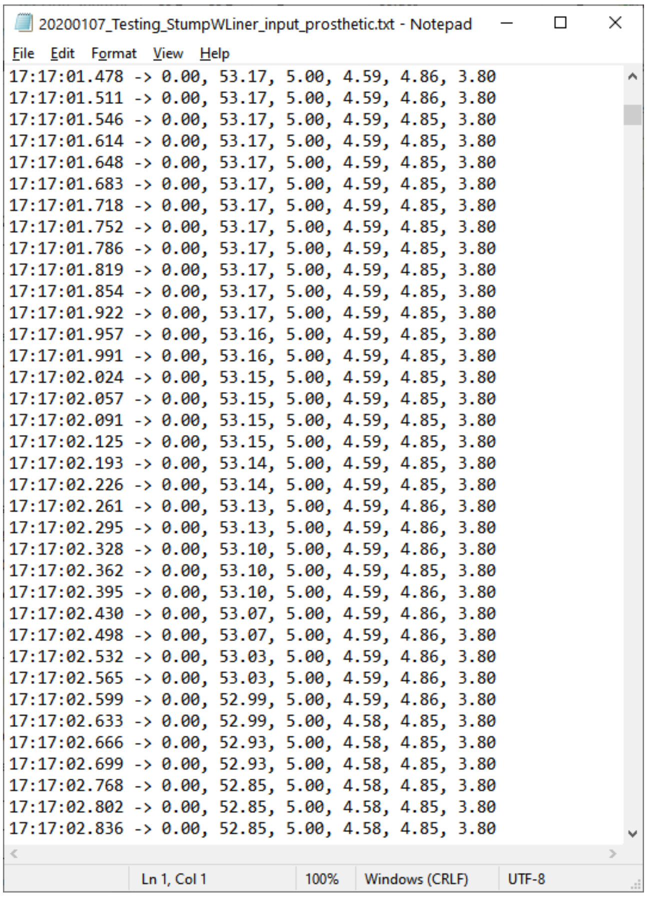


Figure 6.4: Arduino sketch and the response from the sensors while monitoring the *Serial Plotter*. The blue graph shows the response from the TAL220 load cell. Red graph is the response from the TAS606 disk load cell. The green graph is the response form a FSR, calculated to Newton force.



20200107_Testing_StumpWLiner_input_prosthetic.txt - Notepad

File Edit Format View Help

```
17:17:01.478 -> 0.00, 53.17, 5.00, 4.59, 4.86, 3.80
17:17:01.511 -> 0.00, 53.17, 5.00, 4.59, 4.86, 3.80
17:17:01.546 -> 0.00, 53.17, 5.00, 4.59, 4.85, 3.80
17:17:01.614 -> 0.00, 53.17, 5.00, 4.59, 4.85, 3.80
17:17:01.648 -> 0.00, 53.17, 5.00, 4.59, 4.85, 3.80
17:17:01.683 -> 0.00, 53.17, 5.00, 4.59, 4.85, 3.80
17:17:01.718 -> 0.00, 53.17, 5.00, 4.59, 4.85, 3.80
17:17:01.752 -> 0.00, 53.17, 5.00, 4.59, 4.85, 3.80
17:17:01.786 -> 0.00, 53.17, 5.00, 4.59, 4.85, 3.80
17:17:01.819 -> 0.00, 53.17, 5.00, 4.59, 4.85, 3.80
17:17:01.854 -> 0.00, 53.17, 5.00, 4.59, 4.85, 3.80
17:17:01.922 -> 0.00, 53.17, 5.00, 4.59, 4.85, 3.80
17:17:01.957 -> 0.00, 53.16, 5.00, 4.59, 4.85, 3.80
17:17:01.991 -> 0.00, 53.16, 5.00, 4.59, 4.85, 3.80
17:17:02.024 -> 0.00, 53.15, 5.00, 4.59, 4.85, 3.80
17:17:02.057 -> 0.00, 53.15, 5.00, 4.59, 4.85, 3.80
17:17:02.091 -> 0.00, 53.15, 5.00, 4.59, 4.85, 3.80
17:17:02.125 -> 0.00, 53.15, 5.00, 4.59, 4.85, 3.80
17:17:02.193 -> 0.00, 53.14, 5.00, 4.59, 4.85, 3.80
17:17:02.226 -> 0.00, 53.14, 5.00, 4.59, 4.85, 3.80
17:17:02.261 -> 0.00, 53.13, 5.00, 4.59, 4.86, 3.80
17:17:02.295 -> 0.00, 53.13, 5.00, 4.59, 4.86, 3.80
17:17:02.328 -> 0.00, 53.10, 5.00, 4.59, 4.86, 3.80
17:17:02.362 -> 0.00, 53.10, 5.00, 4.59, 4.85, 3.80
17:17:02.395 -> 0.00, 53.10, 5.00, 4.59, 4.86, 3.80
17:17:02.430 -> 0.00, 53.07, 5.00, 4.59, 4.86, 3.80
17:17:02.498 -> 0.00, 53.07, 5.00, 4.59, 4.86, 3.80
17:17:02.532 -> 0.00, 53.03, 5.00, 4.59, 4.86, 3.80
17:17:02.565 -> 0.00, 53.03, 5.00, 4.59, 4.86, 3.80
17:17:02.599 -> 0.00, 52.99, 5.00, 4.59, 4.86, 3.80
17:17:02.633 -> 0.00, 52.99, 5.00, 4.58, 4.85, 3.80
17:17:02.666 -> 0.00, 52.93, 5.00, 4.58, 4.85, 3.80
17:17:02.699 -> 0.00, 52.93, 5.00, 4.58, 4.85, 3.80
17:17:02.768 -> 0.00, 52.85, 5.00, 4.58, 4.85, 3.80
17:17:02.802 -> 0.00, 52.85, 5.00, 4.58, 4.85, 3.80
17:17:02.836 -> 0.00, 52.85, 5.00, 4.58, 4.85, 3.80
```

Ln 1, Col 1 100% Windows (CRLF) UTF-8

Figure 6.5: Screenshot of the output in *Serial Monitor* in Arduino IDE.

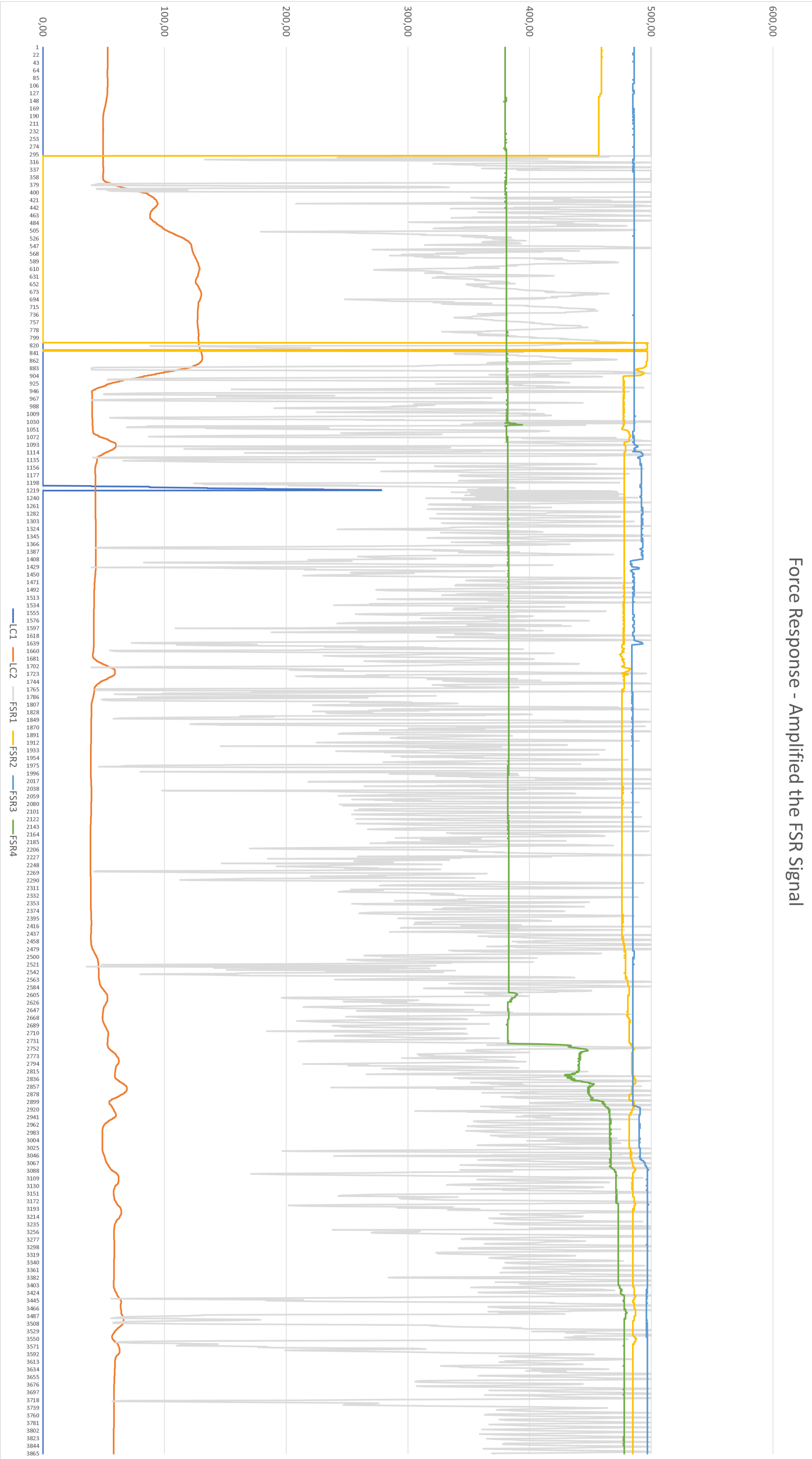


Figure 6.6: Result of testing force output when adding the liner to the stump.

6.3 User Control of the System

With a wish to give the user a way of controlling the test rig without the need for an extensive programming background resulted in the interface box. Three buttons were placed together with the display and the microSD-slot. Two of them are in (proper) use in the final sketches.

- Tear
- Reset
- OK

To make it intuitive for the user, the buttons are engraved with these names. The reset button is just an extension of Arduino's built in reset button, as the Arduino is placed away from the user. The reset button simply resets the program so it starts from line one (that would include the setup which most times only run once). The tear button is programmed to start the HX711 *tear function*, zeroing the values of both of the load cells. The code below shows the fragment of the code which is responsible for the tear function from Appendix C.1.3

```
if (buttonStateTear == HIGH){
digitalWrite(ledPinRed, LOW);
delay(100);
digitalWrite(ledPinRed, HIGH);
Serial.println("Tear Button pushed");
LoadCell_1.begin();
LoadCell_2.begin();
long stabilisingtime = 2000; // tare preciscion can be improved by adding a few seconds of stabilising time
byte loadcell_1_rdy = 0;
byte loadcell_2_rdy = 0;
while ((loadcell_1_rdy + loadcell_2_rdy) < 2) { //run startup, stabilization and tare, both modules simultaniously
if (!loadcell_1_rdy) loadcell_1_rdy = LoadCell_1.startMultiple(stabilisingtime);
if (!loadcell_2_rdy) loadcell_2_rdy = LoadCell_2.startMultiple(stabilisingtime);
}
LoadCell_1.setCalFactor(calValue_1); // user set calibration value (float)
LoadCell_2.setCalFactor(calValue_2); // user set calibration value (float)

digitalWrite(ledPinRed, LOW);
delay(100);
digitalWrite(ledPinRed, HIGH);
delay(300);
digitalWrite(ledPinRed, LOW);
}
```

Additionally as indicated in this sketch, two LEDs are connected to the test rig sensor connection point (right side of the rig structure). These are used to give a notice to the user that something is happening - people are often attracted to a visual response, especially when it seems that "nothing is happening". Using the LEDs are also a good tool for the prototyper, as it can be used for troubleshooting if the code does not seem to do what it is suppose to. For instance, the red LED lights up when the Arduino runs through the setup, and the green LED lights up when the two load cells are finished with their calibration. Also, when doing the tear using the pushbutton, the red LED blinks.

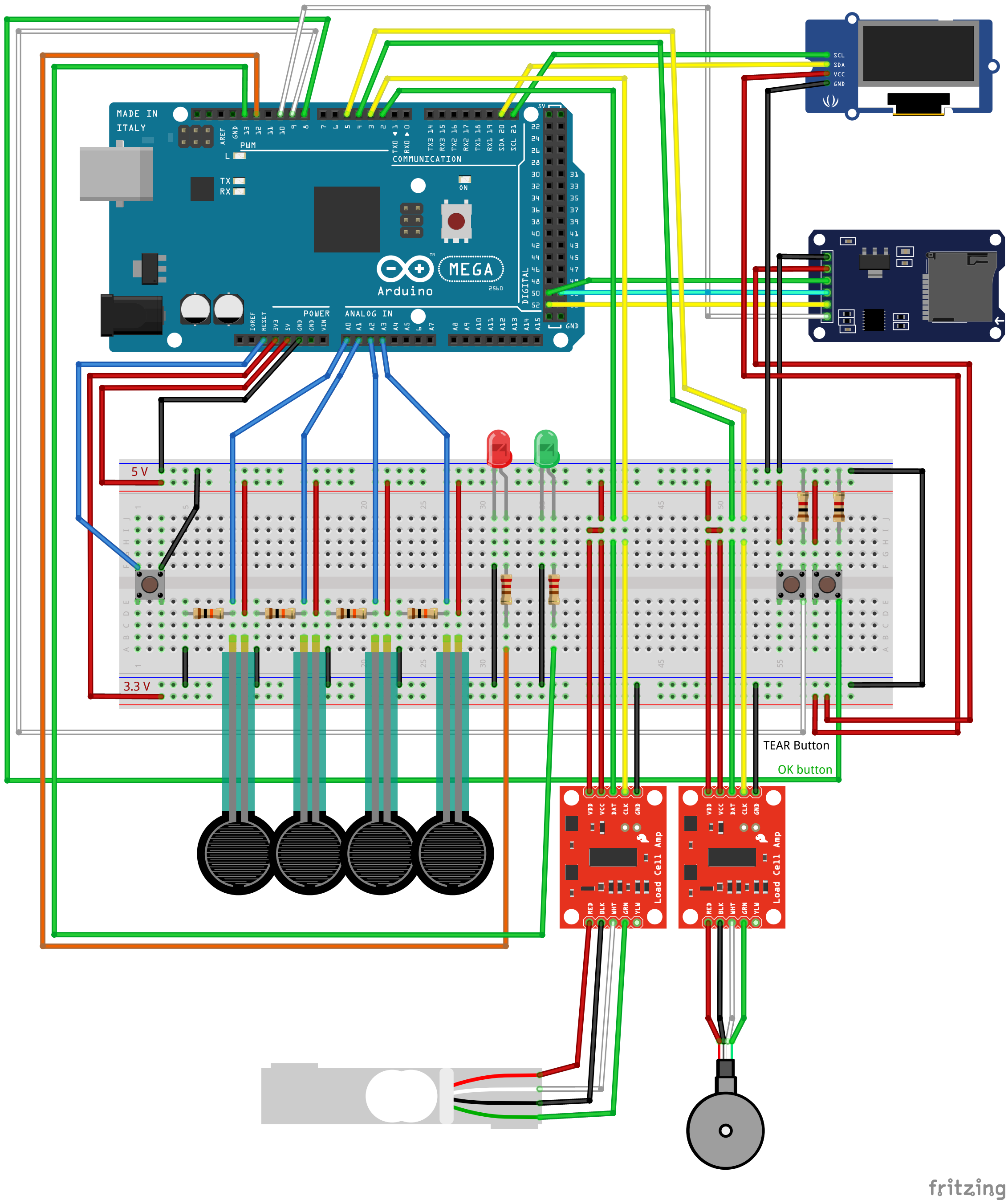


Figure 6.7: Diagram of the core setup of the sensor system including 4x FSR, 2 x Load Cells incl. HX711 amplifiers, microSD-module, OLED Display, 2x LEDs, 3x Pushbuttons and one Arduino Mega

7 Conclusion and Future Work

The main outcome of this project is the mapping and understanding of what is important to consider for mechanical engineers when developing solutions to improve prosthetic sockets. Through the interviews information which would not be found by literature studies alone was gathered and sat in system. Discovering and testing out various sensors which can be used to mimic human sensation in the socket resulted in a 1:1 size test rig. With the the rig as it is now can only be used for the socket and stump which it is made for, but with a production of a new stump, the setup would be the same for other prosthetic sockets. Hypothetically, it could also be used for other limb prosthetics than just the transfemoral.

7.1 Future Work

The prototype is not intended to be "a final testing solution", rather it can serve as a base for further research and development. Main direction which I was considering when working on this project includes

- Improving the interface: make a standalone rig which does not need a computer to calibrate the sensors. Optional "save to SD-card"
- The stump design has a lot of potential for further work, both including sensors and to make it "feel like a human leg"
- Structural design. The doffing (detach) of the stump is not convenient as it is now as much forces are required to release the seal.
- Lastly, improving the accuracy of the sensors

Bibliography

Sarah Al-Mutlaq. Getting Started with Load Cells - learn.sparkfun.com, 2016. URL *https://learn.sparkfun.com/tutorials/getting-started-with-load-cells*.

Arduino. Arduino - Introduction, 2020. URL *https://www.arduino.cc/en/Guide/Introduction*.

Ryan Caldwell, L Faaop, and Stefania Fatone. Development of the Northwestern University Flexible Sub-Ischial Vacuum (NU-FlexSIV) Socket for Persons with Transfemoral Amputation. *12th Annual Lewis Landsberg Research Day, Northwestern University, Chicago, IL*, page 1, April 2016.

Lucy Coughlan, Aicha Benyaich, Tarina van der Stockt, Rachael Lowe, Tony Lowe, and Sheik Abdul Khadir. Principles of Amputation, August 2019. URL *https://www.physio-pedia.com/Principles_of_Amputation*. Page Version ID: 218947.

Jason Daley. This 3,000-Year-Old Wooden Toe Shows Early Artistry of Prosthetics, June 2017. URL *https://www.smithsonianmag.com/smart-news/study-reveals-secrets-ancient-cairo-toe-180963783/*.

Darin J. Eich. Design Thinking Mindsets for Human-Centered Design, November 2018. URL *https://www.innovationtraining.org/design-thinking-mindsets/*.

Christer Elverum, Torgeir Welo, and Martin Steinert. The Fuzzy Front End: Concept Development in the Automotive Industry. August 2014. doi: 10.1115/DETC2014-35138.

Al Emondi. Revolutionizing Prosthetics, 2017. URL *https://www.darpa.mil/program/revolutionizing-prosthetics*.

Stefania Fatone. Development of Subischial Prosthetic Sockets with Vacuum-Assisted Suspension for Highly Active Persons with Transfemoral Amputations: Northwestern University Prosthetics-Orthotics Center: Feinberg School of Medicine Northwestern University, 2019. URL *https://www.nupoc.northwestern.edu/research/projects/lowerlimb/dev_subischial.html*.

Stefania Fatone and Ryan Caldwell. Northwestern University Flexible Subischial Vacuum Socket for persons with transfemoral amputation-Part 1: Description of technique. *Prosthetics and Orthotics International*, 41(3):237–245, June 2017a. ISSN 1746-1553. doi: 10.1177/0309364616685229.

Stefania Fatone and Ryan Caldwell. NU FlexSIV Socket Course, May 2017b. URL *https://www.youtube.com/watch?v=1y8gfOg1WUU*.

Stefania Fatone and Ryan Caldwell. Sub-Ischial Socket Work Form 2019, 2019.

Fillauer. Pyramid Slide Male (PSM) Modular System, 2020. URL *http://fillauer.eu/prosthetics-lower/alignment/psm*.

FlexiForce Tactile Force Sensors. How to Calibrate FlexiForce Sensors for an Embedded Application, April 2018. URL *https://www.youtube.com/watch?time_continue=241&v=xMykO8LRTzA&feature=emb_title*.

Maria J. Gerschutz, Michael L. Haynes, Derek Nixon, and James M. Colvin. Strength evaluation of prosthetic check sockets, copolymer sockets, and definitive laminated sockets. *The Journal of Rehabilitation Research and Development*, 49(3):405, 2012. ISSN 0748-7711. doi: 10.1682/JRRD.2011.05.0091. URL *http://www.rehab.research.va.gov/jour/2012/493/pdf/gershutz493.pdf*.

F A Gottschalk and M Stills. The biomechanics of trans-femoral amputation. *Prosthetics and Orthotics International*, 18:12–17, January 1994. URL *http://www.oandplibrary.org/poi/pdf/1994_01_012.pdf*.

Samuel Hamontree and Roy Snelson. The Use of Check Sockets in Lower-Limb Prosthetics. *Orth. and Pros.*, 27(4):30–33, December 1973.

Stephanie Houde and Charles Hill. What do Prototypes Prototype? 2, December 1997. doi: 10.1016/B978-044481862-1/50082-0.

HTC. Load Cell Series-HT SENSOR TECHNOLOGY CO.,LIMITED, 2019. URL *http://www.htc-sensor.com/products/?id=8*.

Interlink Electronics. FSR Integration Guide. URL *https://www.generationrobots.com/media/FSR400-Series-Integration-Guide.pdf*.

Olav Kallhovd. HX711_adc Arduino Library, November 2019. URL *https://github.com/olkal/HX711_ADC*. original-date: 2017-09-29T22:42:50Z.

H Lee Mantelmacher. Anti-slip prosthetic attachment system (Patent US7883547b2), February 2011. URL *https://patentswarm.com/patents/US7883547B2*.

Merriam-Webster. Merriam-Webster Medical Dictionary, 2019. URL *https://www.merriam-webster.com/browse/medical/a*.

Eivind Heilmann Modalsli, Tobias Goihl, Linus Appelkvist, Øyvind Bernhardsen, and et. al. Interview 1 at TOV: Prosthetic Team, September 2019.

Suketu Naik. Lecture 5 Force Sensing. URL *https://faculty.weber.edu/snaik/ECE5900_ECE6900/05Lec05_Force.pdf*.

Danny Nosonowitz and Lady Ada. Force Sensitive Resistor (FSR), July 2012. URL *https://learn.adafruit.com/force-sensitive-resistor-fsr/using-an-fsr*.

Ohmite. FSR Series Datasheet, 2018a. URL *http://www.farnell.com/datasheets/2737025.pdf*.

Ohmite. Integration Guide - Force Sensing Resistor, January 2018b. URL *https://www.mouser.com/pdfdocs/Ohmite-FSR-Integration-Guide-V1-0_11-01-18.pdf*.

Omega. Installing a Load Cell: Best Practices, September 2018. URL *https://www.omega.com/en-us/resources/load-cell-installation*.

Omega. What is a Load Cell Amplifier?, November 2019. URL *https://www.omega.com/en-us/resources/load-cell-amplifier*.

Ossur. Icelock 500 Series. URL *https://assets.ossur.com/library/19261/ICELOCK*.

Ossur. Iceross Transfemoral Seal In Technical manual, December 2004. URL *https://assets.ossur.com/library/7922/Iceross%20Transfemoral%20Seal-In%c2%a0%20Technical%20manual.pdf*.

Ossur. Iceross Seal-In Liner Instructions for Use, 2017. URL *https://assets.ossur.com/library/7732/Iceross%20Transfemoral%20Seal-In%c2%a0%20Instructions%20for%20use.pdf*.

Ossur. Leg amputation Information for New Amputees, 2019. URL *https://www.ossur.com/en-us/prosthetics/prosthetic-users-info/information-for-lower-limb-amputees/leg-amputation*.

Suresh Parmar, Iryna Khodasevych, and Olga Troynikov. Evaluation of Flexible Force Sensors for Pressure Monitoring in Treatment of Chronic Venous Disorders. *Sensors (Basel, Switzerland)*, 17(8), August 2017. ISSN 1424-8220. doi: 10.3390/s17081923. URL *https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5580323/*.

Linda Paterno, Michele Ibrahimi, Emanuele Gruppioni, and Leonardo Ricotti. Sockets for Limb Prostheses: A Review of Existing Technologies and Open Challenges. *IEEE Transactions on Biomedical Engineering*, PP(99)(1-1), January 2018. doi: 10.1109/TBME.2017.2775100. URL *https://www.researchgate.net/publication/322667772_Sockets_for_Limb_Prostheses_A_Review_of_Existing_Technologies_and_Open_Challenges*.

PCB Load and Torque Division. Load Cell Handbook: A Technical Overview and Selection Guide. URL *https://www.pcb.com/contentstore/MktgContent/LinkedDocuments/Load_Torque/LT-LoadCellHandbook_LowRes.pdf*.

Phidgets. Load Cell Primer - Phidgets Support, November 2018. URL *https://www.phidgets.com/docs/Load_Cell_Primer*.

Luke Prielipp. What is a Load Cell? | Types of Load Cells | RealPars, August 2019. URL *https://realpars.com/load-cell/*.

Award Prosthetics. Amputee and Prosthetic Terminology at Award Prosthetics, 2012. URL *http://www.awardprosthetics.com/terminology.html*.

Donald G. Reinertsen and Paul G. Smith. The strategist's role in shortening product development. *The Journal of business strategy*, 1991. doi: 10.1108/eb039425.

Sparkfun. TAS606 - Load Cell - 200kg, Disc - SEN-13332 - SparkFun Electronics, June 2015. URL *https://www.sparkfun.com/products/13332*.

Sparkfun. HX711 - SparkFun Load Cell Amplifier - SEN-13879 - SparkFun Electronics, July 2016. URL *https://www.sparkfun.com/products/13879*.

Ralf Martin Steinert. TMM4245 Fuzzy Front End - Introduction Session, January 2017. URL *https://www.ntnu.no/studier/emner/TMM4245#tab=omEmnet*.

Rand Swenson. Basic Human Anatomy Directional Terms, 2009. URL *https://www.dartmouth.edu/~humananatomy/figures/chapter_1/1_1.HTM*.

Tekscan. How Does a Force Sensing Resistor (FSR) Work?, July 2019. URL *https://www.tekscan.com/blog/flexiforce/how-does-force-sensing-resistor-fsr-work*.

LeTourneau Prosthetics and Orthotics Southeast Texas. Prosthetics | Artificial Limbs | Limb Replacement For Amputees, December 2018. URL *https://www.llop.com/prosthetics/*.

Karl T. Ulrich and Steven D. Eppinger. *Product Design and Development*. McGraw-Hill Higher Education, New York, NY, USA, 4th edition, 2007. ISBN 978-0-07-310142-2.

Tarina van der Stockt and Kim Jackson. Lower Limb Prosthetic Sockets and Suspension Systems, January 2020. URL *https://www.physio-pedia.com/Lower_Limb_Prosthetic_Sockets_and_Suspension_Systems*.

List of Figures

Appendix contents

A

Documentation and Resources

A.1 Supplementary Images of the Test Rig

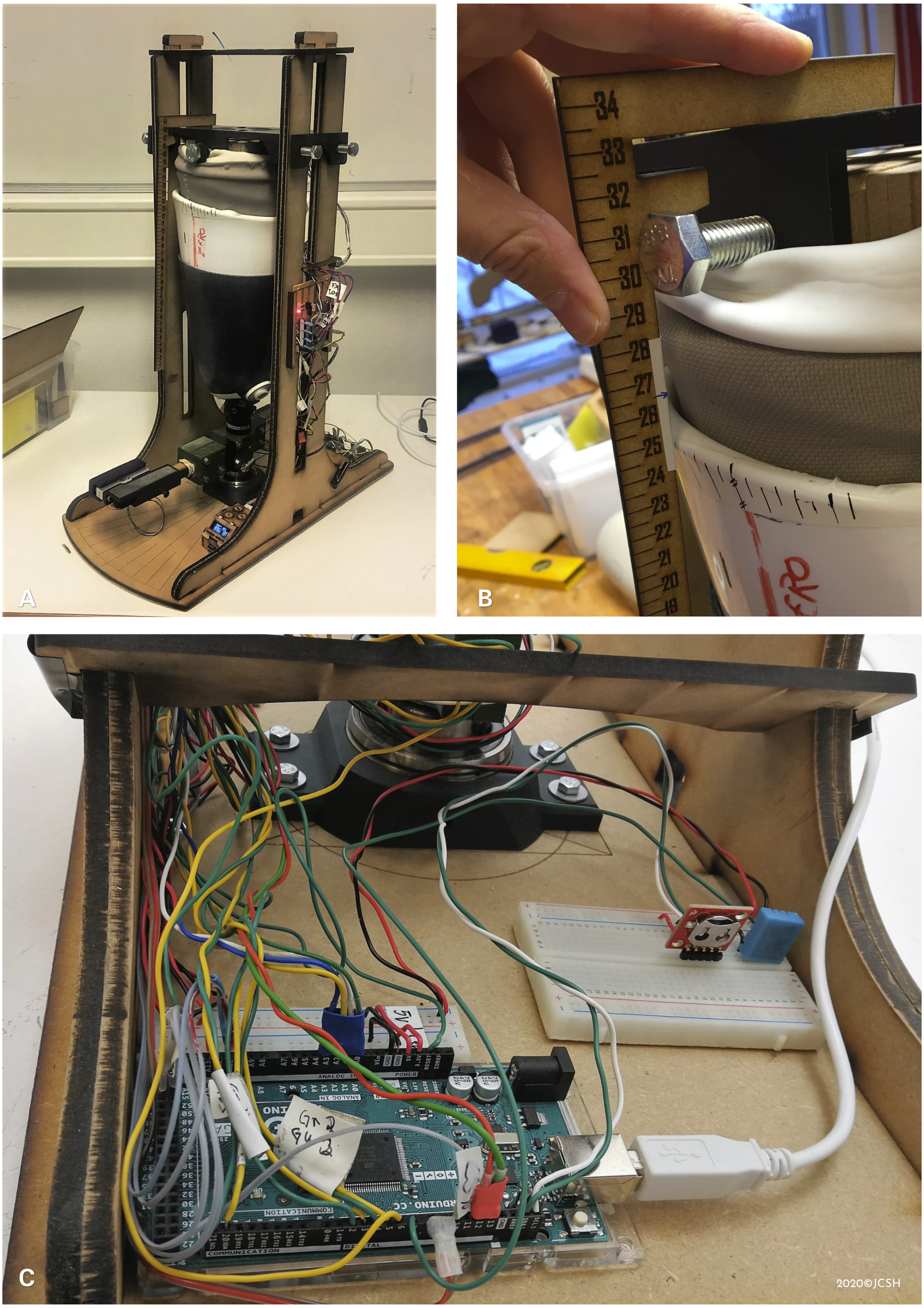


Figure A.1: The final test rig design. b) Indication of about 15 mm space between the socket and stump at the most distal point of the stump.

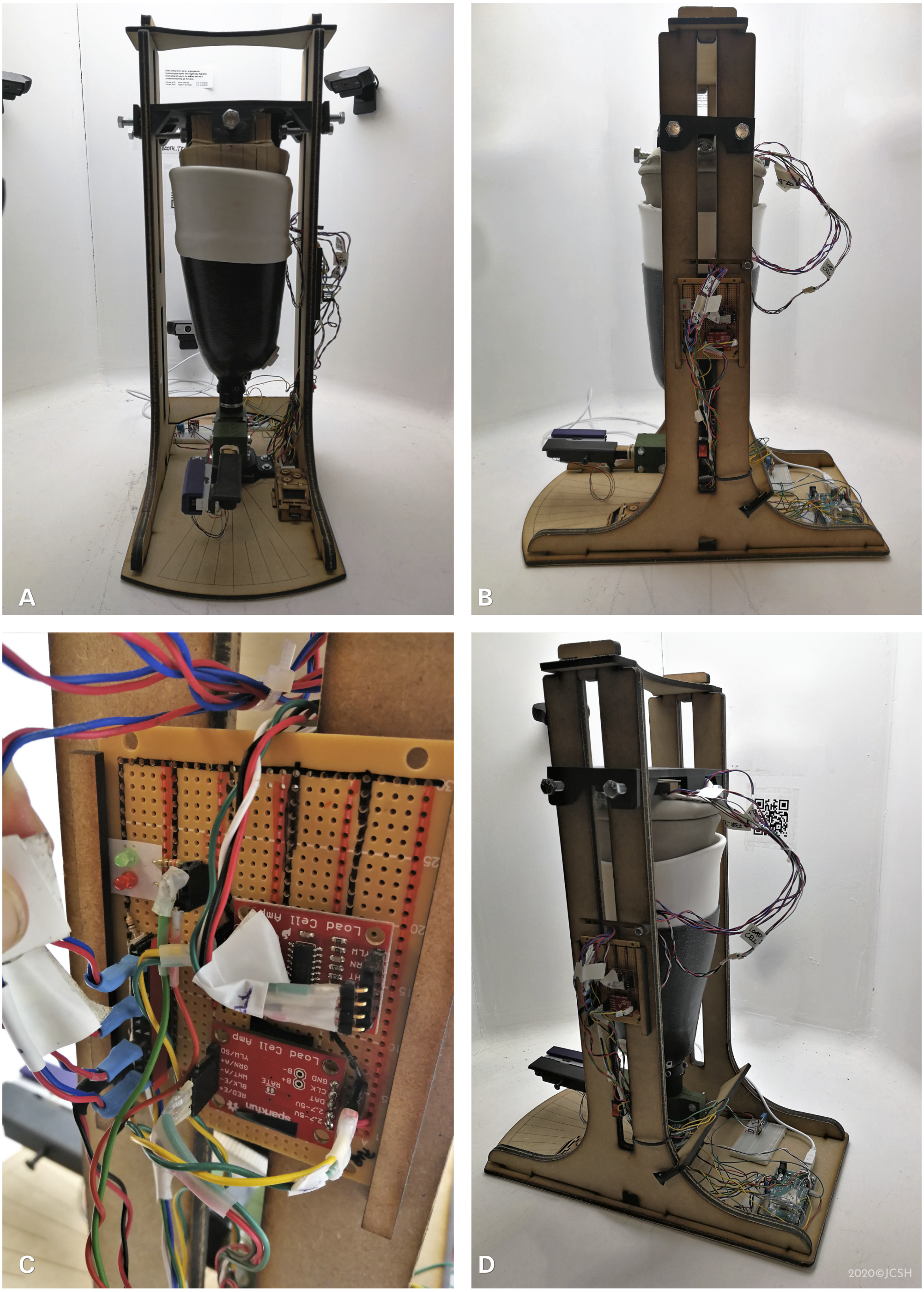


Figure A.2: Closeup image of the sensor connection prototype board (c) along with different views of the rig.

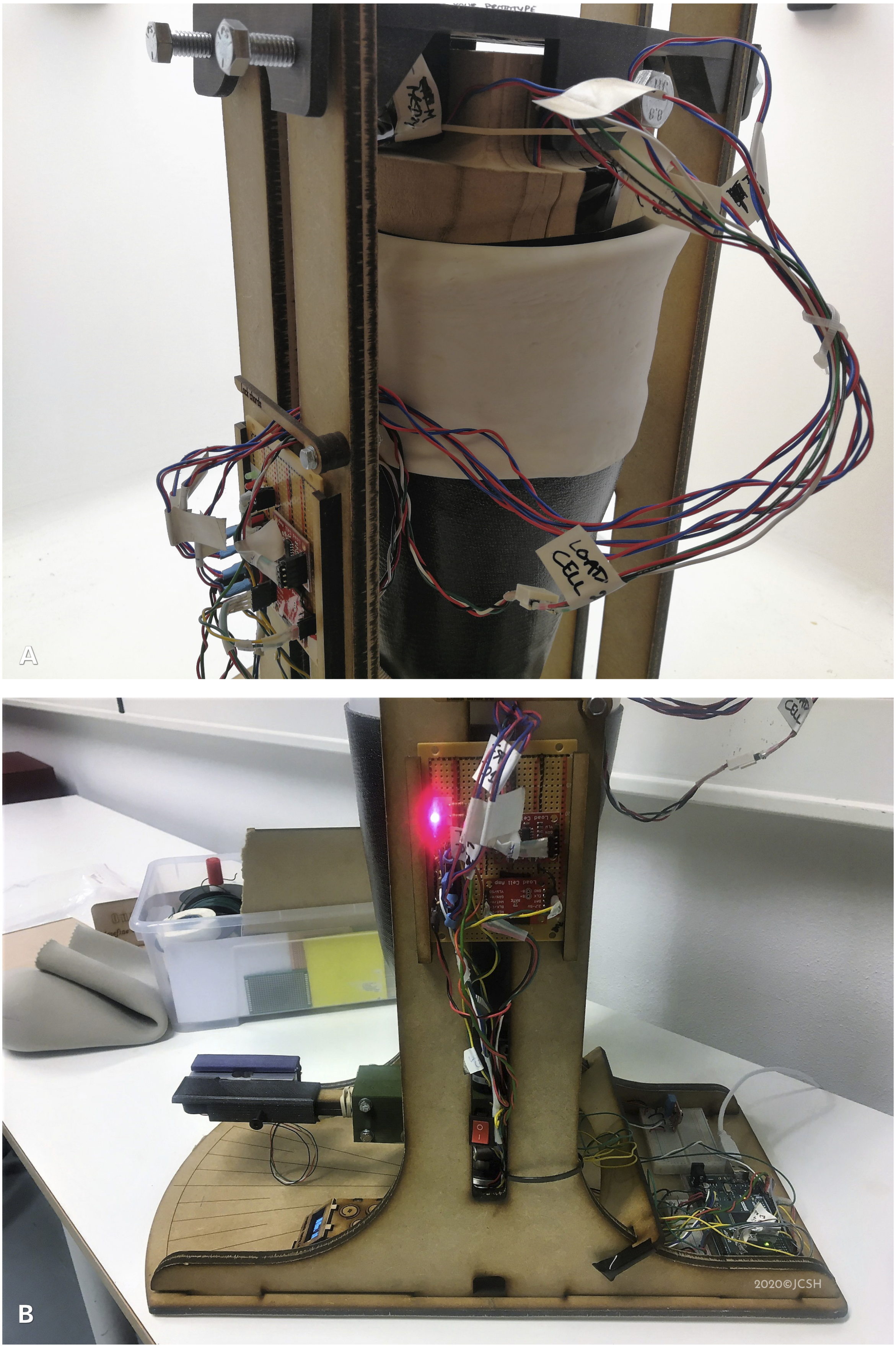
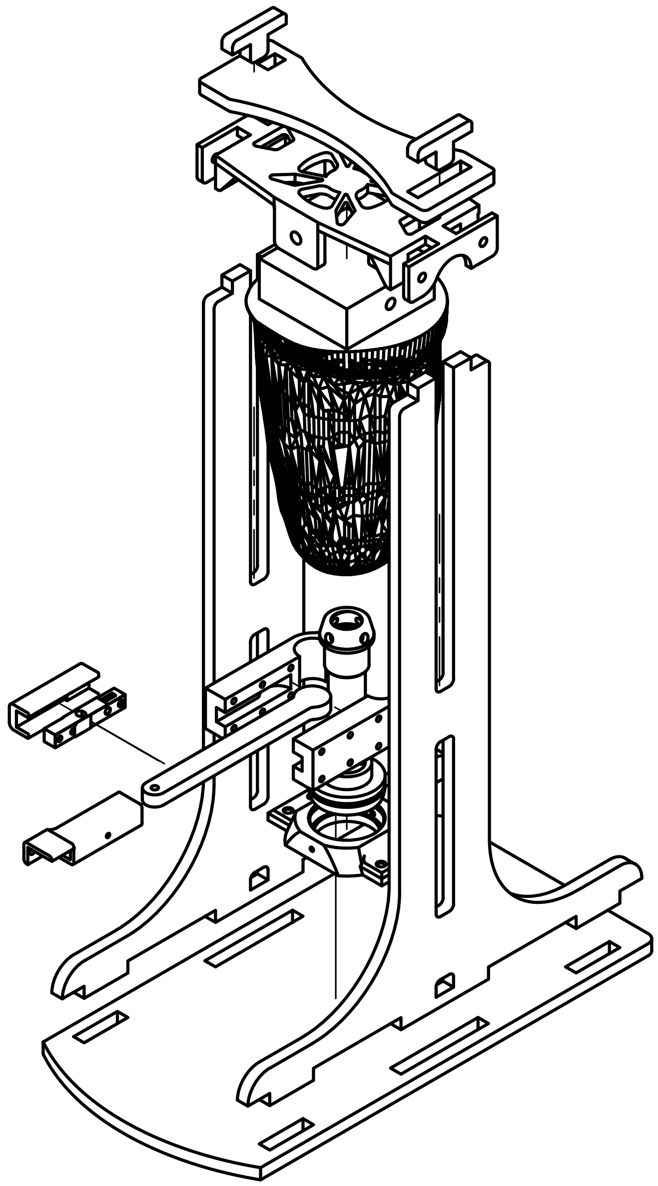


Figure A.3: Closeup images of the sensor system

A.2 Technical Drawings

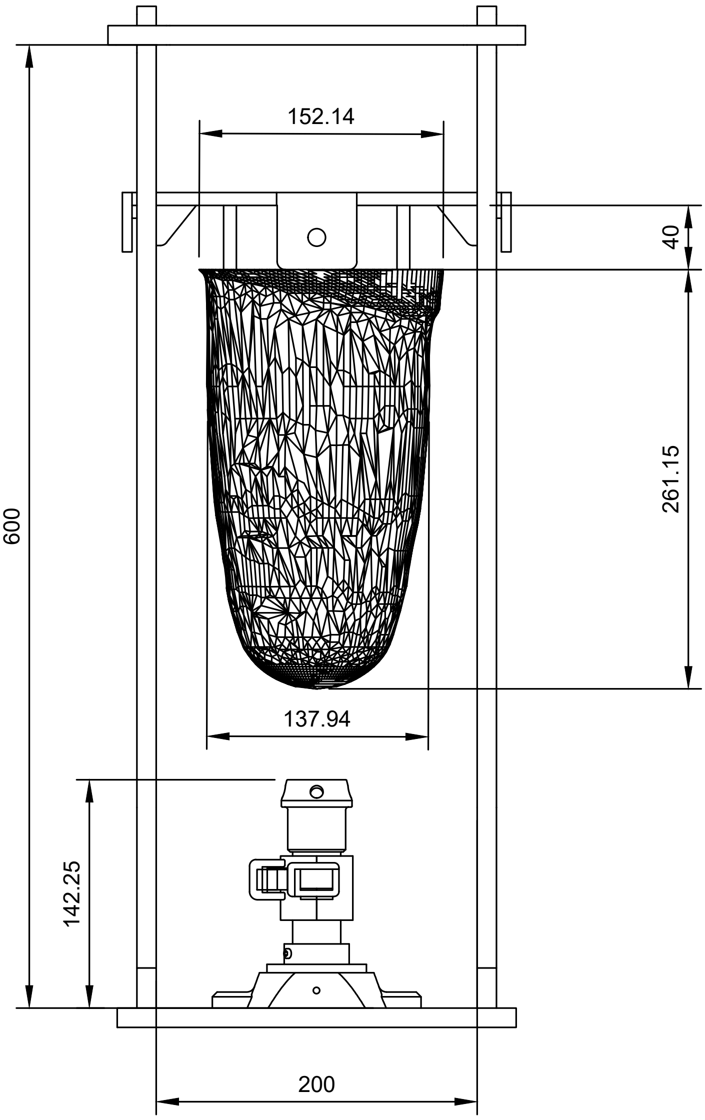


Dept.	Technical reference	Created by	Approved by
MTP	RIG_0002	Josefine Haugom 27.02.2020	JCSH 2020.02.27

NTNU
Faculty of Engineering
Department of Mechanical and Industrial Engineering
TROLLLABS

Document type	Document status
	Final

Title	DWG No.		
Socket Test Rig Assembly Exploded View	2020.0001		
	Rev.	Date of issue	Sheet
		2020.02.27	1/6



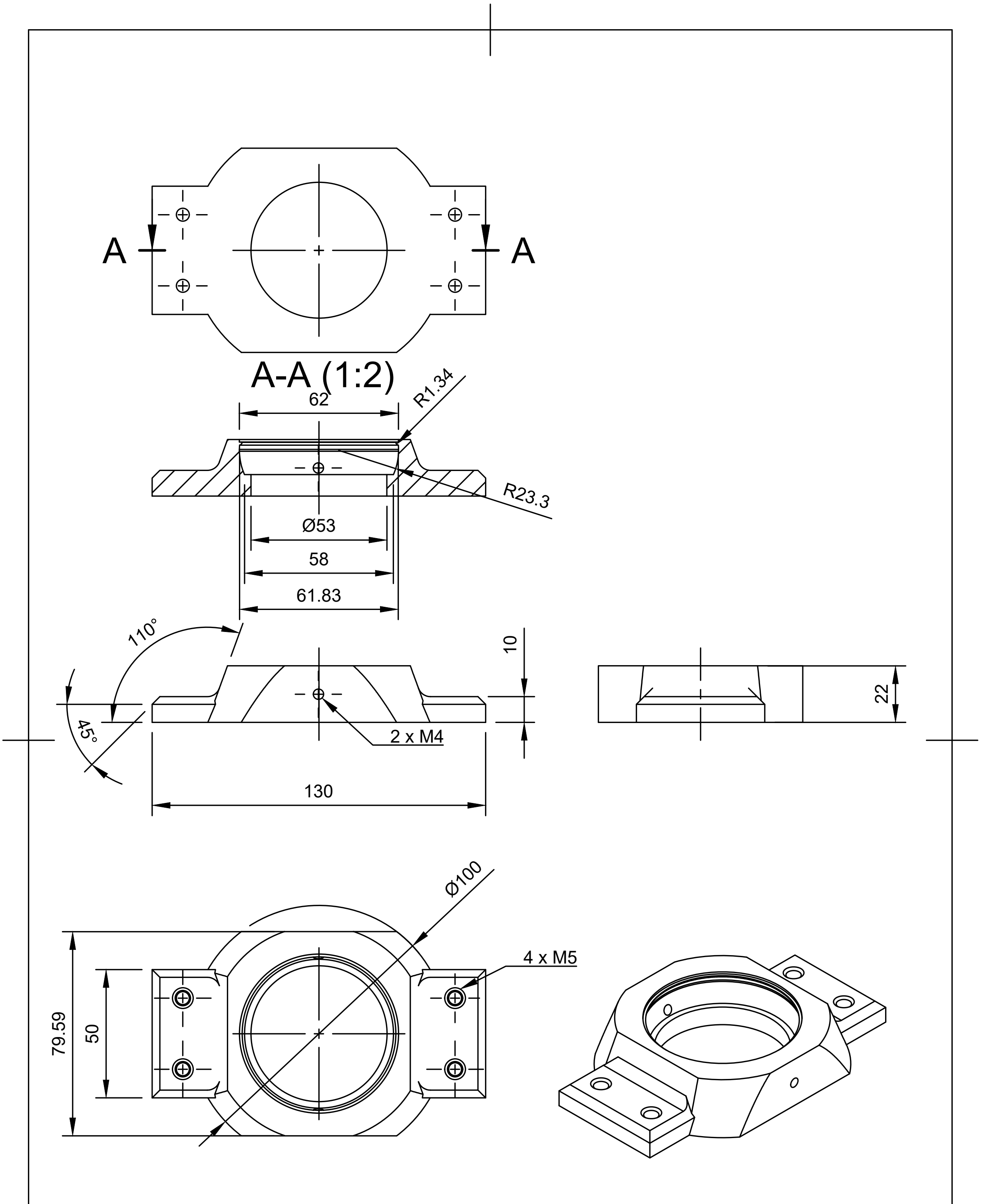
Dept.	Technical reference	Created by	Approved by
MTP	RIG_0002	Josefine Haugom 27.02.2020	JCSH 2020.02.27

NTNU
Faculty of Engineering
Department of Mechanical and Industrial Engineering
TROLLLABS

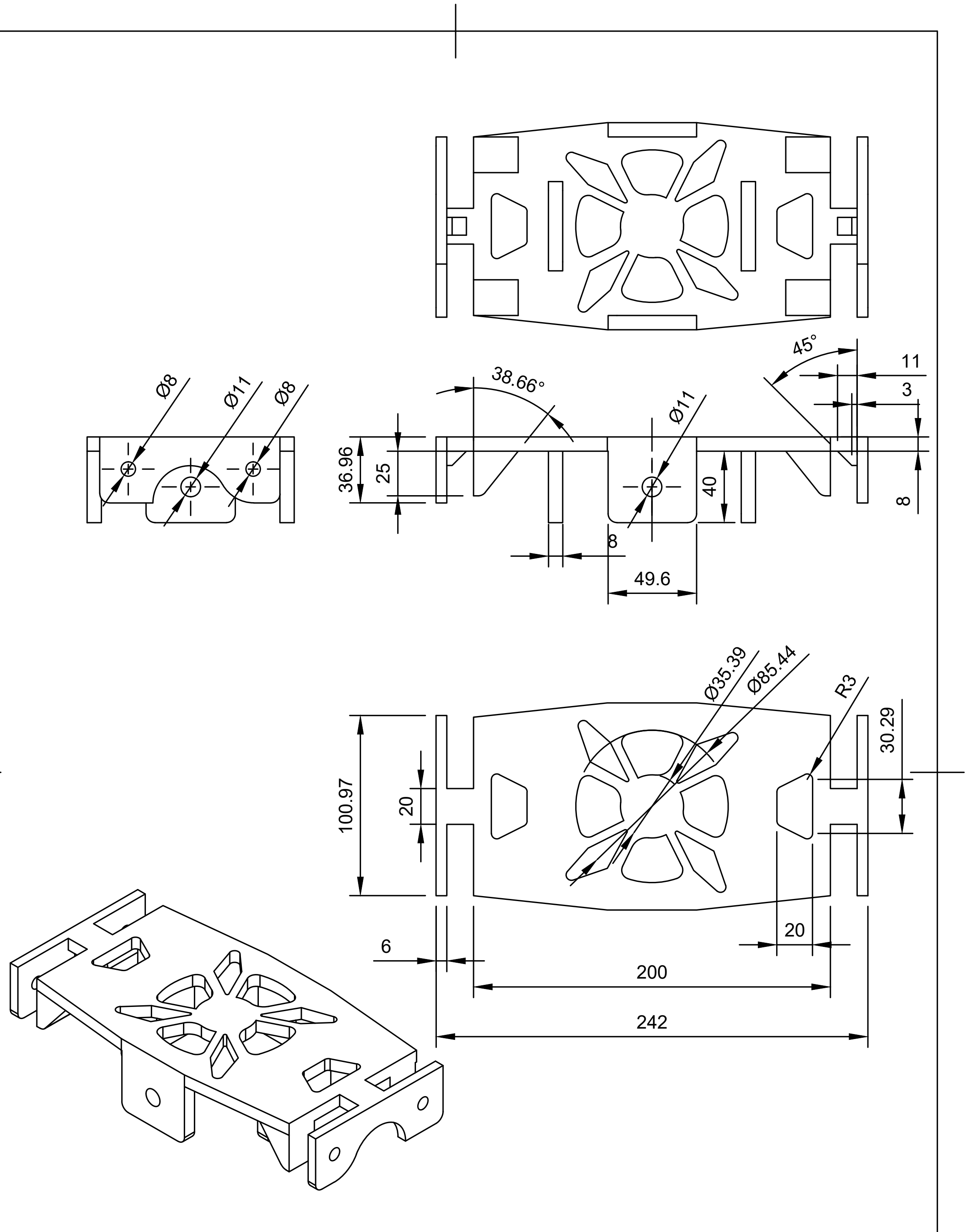
Document type	Document status
	Final

Title	DWG No.
Socket Test Rig Assembly	2020.0001

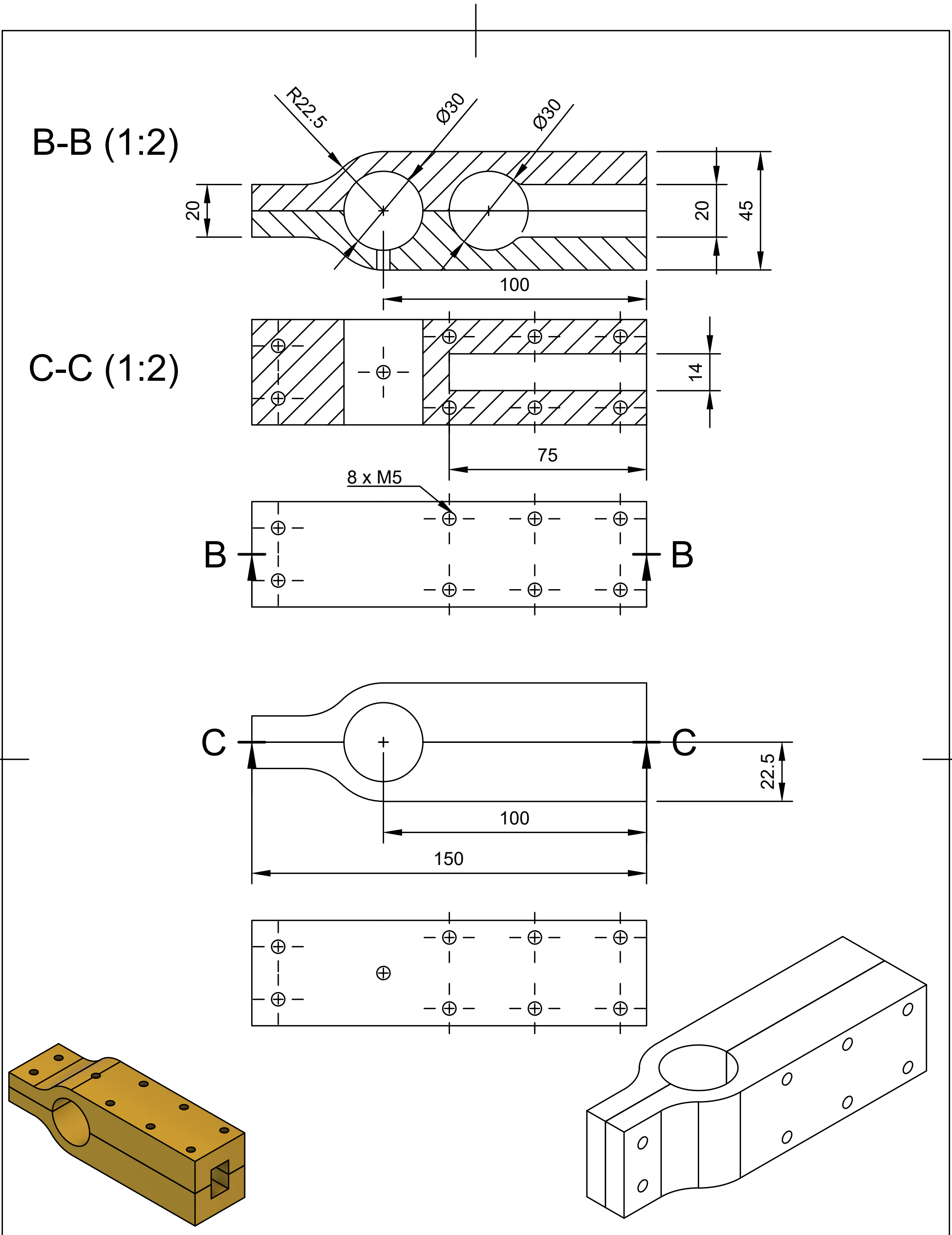
Rev.	Date of issue	Sheet
	2020.02.27	2/6



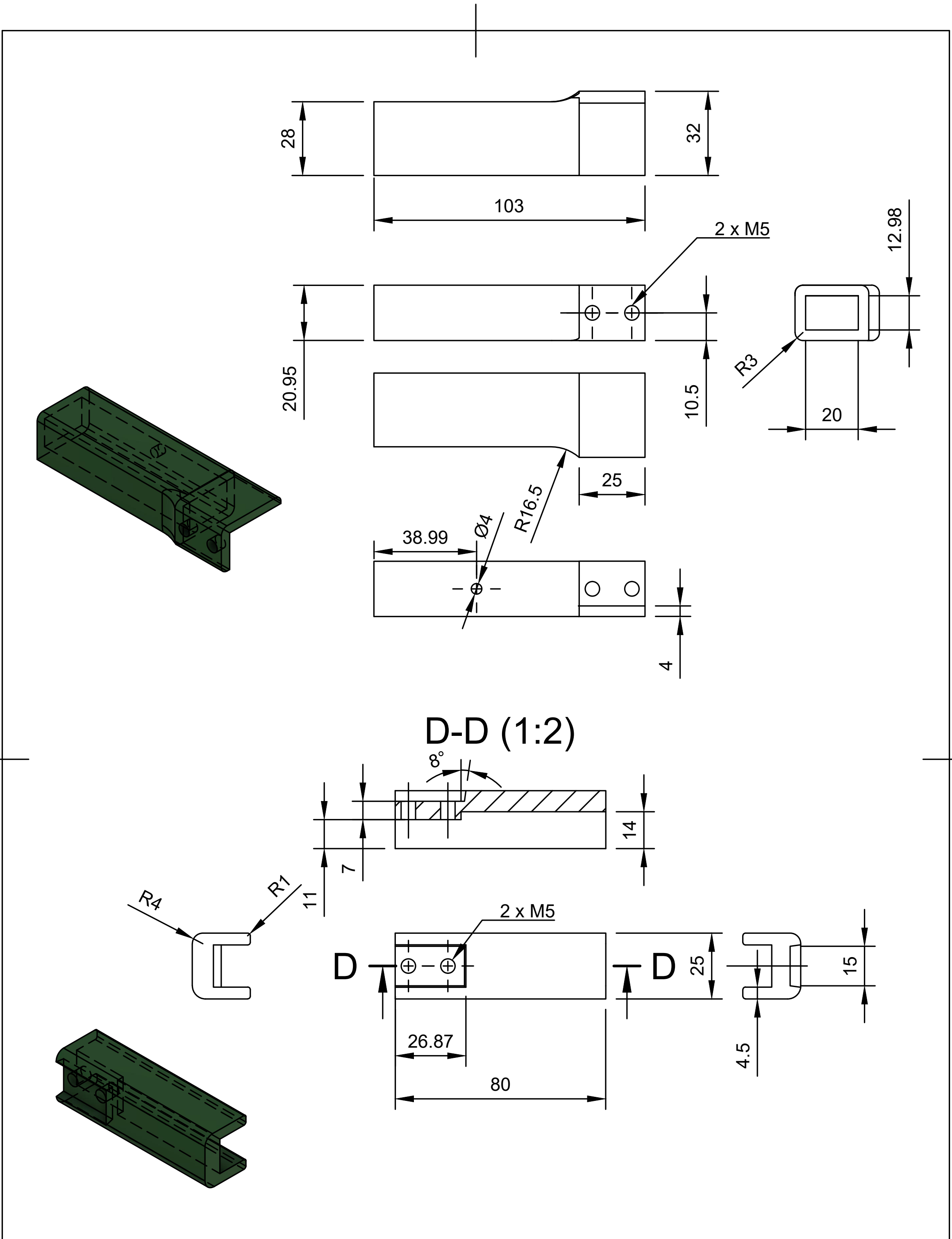
Dept.	Technical reference	Created by	Approved by
MTP	RIG_0002	Josefine Haugom 27.02.2020	JCSH 2020.02.27
NTNU Faculty of Engineering Department of Mechanical and Industrial Engineering TrollLABS		Document type	Document status: Final
		Title: Socket Test Rig Roller Bearing Adapter	DWG No.: 2020.0001
			Rev. / Date of issue: 2020.02.27 / Sheet: 3/6



Dept.	Technical reference	Created by	Approved by
MTP	RIG_0002	Josefine Haugom 27.02.2020	JCSH 2020.02.27
NTNU Faculty of Engineering Department of Mechanical and Industrial Engineering TROLLLABS		Document type	Document status: Final
		Title: Socket Test Rig Roller Bearing Adapter	DWG No.: 2020.0001
			Rev.: — Date of issue: 2020.02.27 Sheet: 4/6



Dept. MTP	Technical reference RIG_0002	Created by Josefine Haugom 27.02.2020		Approved by JCSH 2020.02.27	
NTNU Faculty of Engineering Department of Mechanical and Industrial Engineering TROLLLABS		Document type		Document status Final	
		Title Socket Test Rig Bracket Lever-Pylon		DWG No. 2020.0001	
			Rev.	Date of issue 2020.02.27	Sheet 5/6



Dept.	Technical reference	Created by	Approved by
MTP	RIG_0002	Josefine Haugom 27.02.2020	JCSH 2020.02.27

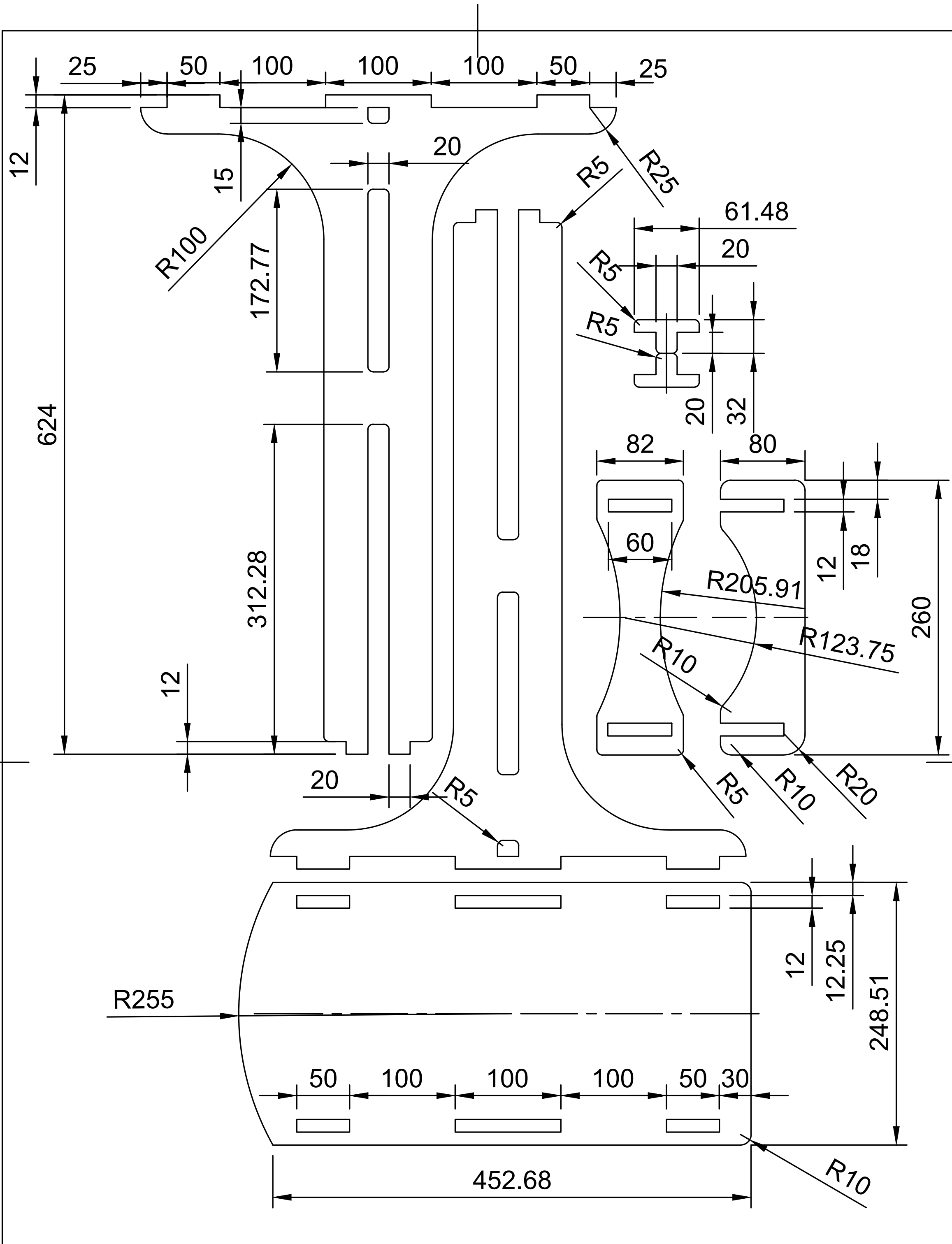
NTNU
Faculty of Engineering
Department of Mechanical and Industrial Engineering

TROLLLABS

Document type	Document status
	Final

Title	DWG No.
Socket Test Rig Load Cell Attachment	2020.0001

Rev.	Date of issue	Sheet
	2020.02.27	6/6



Dept.	Technical reference	Created by	Approved by	
MTP	RIG_0001	Josefine Haugom 04.11.2019	JCSH	05.11.2019

NTNU
Faculty of Engineering
Department of Mechanical and Industrial Engineering
TROLLLABS

Document type	Document status		
	Final		
Title	**DWG No.**		
Socket Test Rig Laser Cutting 6mm MDF	2019.0001		
	Rev.	Date of issue	Sheet
	2	05.11.2019	1/1

A.3 Documentation: Mechatronics

A.3.1 Bill of Materials

Amount	Part Type	Properties	Manufacturer	M.Part No	Part Price	Supplier
4	10kΩ Resistor	1/6th Watt, +/- 5% tolerance PTH resistor	Sparkfun	COM-11508	0.03 USD	Sparkfun.com
2	1kΩ Resistor	1/4 Watt, +/- 5% tolerance PTH resistor	Sparkfun	PRT-14492	0.05 USD	Sparkfun.com
2	220Ω Resistor	250mW, +/- 5% tolerance	MCM	66-220	9.66 NOK	Farnell.com
1	Arduino Mega 2560	Development Board, ATmega2560 MCU	Arduino	A000067	422.66 NOK	Farnell.com
1	Force Sensing Resistor	Round 5 kg, sensing area Ø25.42 mm	OHMITE	FSR03CE	120.00 NOK	Farnell.com
3	Force Sensing Resistor	Square, 5 kg, sensing area 39.7x39.7mm	OHMITE	FSR01CE	95.40 NOK	Farnell.com
1	Green LED	LED, Low Power, Green, Through Hole, T-1 3/4 (5mm)	Kingbright	L-1503GC	1.49 NOK	Farnell.com
1	Load Cell	Parallel Beam Load Cell, TAL220, 10 kg	HTC Sensor	TAL220	8.50 USD	Sparkfun.com
1	Load Cell	Button Type Compression Load Cell, TAS606, 200 kg	HTC Sensor	TAS606	59.95 USD	Sparkfun.com
2	Load Cell Amplifier	SparkFun Load Cell Amplifier HX711	Sparkfun	SEN-13879	9.95 USD	Sparkfun.com
1	MicroSD Card	TS16GUSDC10 - MicroSD Memory Card 16 GB, 30 MB/s, 30 MB/s, Transcend	Transcend	TS16GUSDC10	76.60 NOK	Elfadistrelec.no
1	MicroSD Card Module	254 - MicroSD card breakout board, Adafruit	Adafruit	254	73.20 NOK	Elfadistrelec.no
1	OLED Display	0.96", 4pin, monochrome, I2C 128x64 OLED	Geekcredit	958196	29.34 NOK	Banggood
3	Pushbutton	Momentary pushbutton, 12mm	Sparkfun	COM-09190	0.50 USD	Sparkfun.com
1	Red LED	LED, Low Power, Red, Through Hole, T-1 3/4 (5mm)	Kingbright	L-1503SRD	1.75 NOK	Farnell.com
1	Temperature and Humidity Sensor	386 - DHT11 Temperature Humidity Sensor, 5V, Adafruit incl. resistor	Adafruit	386	45.10 NOK	Elfadistrelec.no
1	Real Time Clock	Adafruit PCF8523 Real Time Clock Assembled Breakout Board	Adafruit	3295	44.10 NOK	Elfadistrelec.no

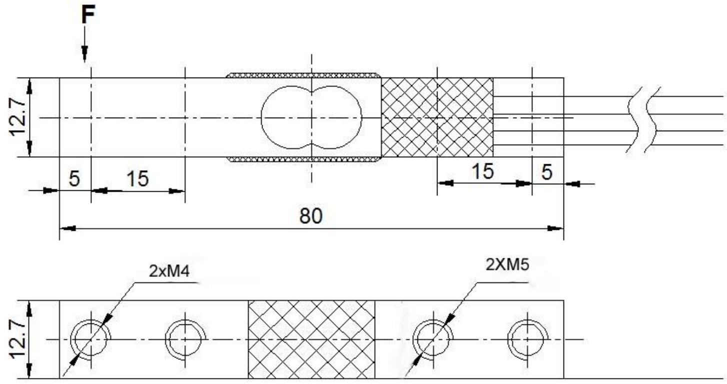
A.3.2 Datasheet: TAL220 Parallel Beam Load Cell

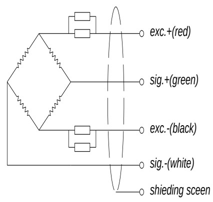
TAL220 PARALLEL BEAM LOAD CELL

WWW.HTC-SENSOR.COM

Features:
◆Capacity： 3~200kg
◆Material: aluminum-alloy or alloy steel
◆Type: Parallel beam type
◆Defend grade: IP65
◆Application : Palm scale, kitchen scale, electronic balance, fishing scale, electronic platform scale and other electronic weighing devices.

Electrical connection and Dimensions:(dimension unit: mm)





Specifications:		
capacity	kg	3,5,10,20,25,30,50(aluminum); 80,100,120,200(alloy steel)
safe overload	%FS	120
ultimate overload	%FS	150
rated output	mV/V	1.0 ± 0.15
excitation voltage	Vdc	5 ~ 10
combined error	%FS	± 0.05
zero unbalance	%FS	± 0.1
non-linearity	%FS	± 0.05
hysteresis	%FS	± 0.05
repeatability	%FS	± 0.03
creep	%FS/3min	± 0.05
input resistance	Ω	1000 ± 15
output resistance	Ω	1000 ± 10
insulation resistance	M Ω	≥ 2000
operating temperature range	℃	-10 ~ +55
compensated temperature range	℃	-10 ~ +40
temperature coefficient of SPAN	%FS/10℃	± 0.05
temperature coefficient of ZERO	%FS/10℃	± 0.05
Electrical connection	cable	4 color wire (standard)or 4 shielded PVC cable, Ø0.8 × 220 mm

※**Ordering code:** model-capacity- rated output-accuracy-defend grade- the length of cable

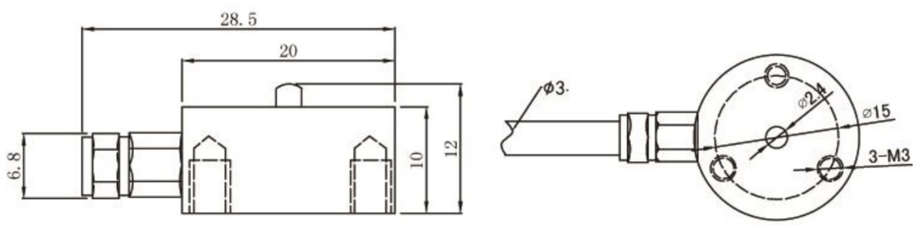
A.3.3 Datasheet: TAS606 Disk Type Load Cell

http://www.htc-sensor.com/products/151.html
Product Numbers: TAS606

Product Description:

- Capacity: 5 500kg
- Material Stainless Steel
- Defend grade: IP66
- Application: Suitable for force control in production control, vibratory feeding equipment and other electronic weighing or force measuring fields.

Electrical connection and Dimensions:(dimension unit: mm)

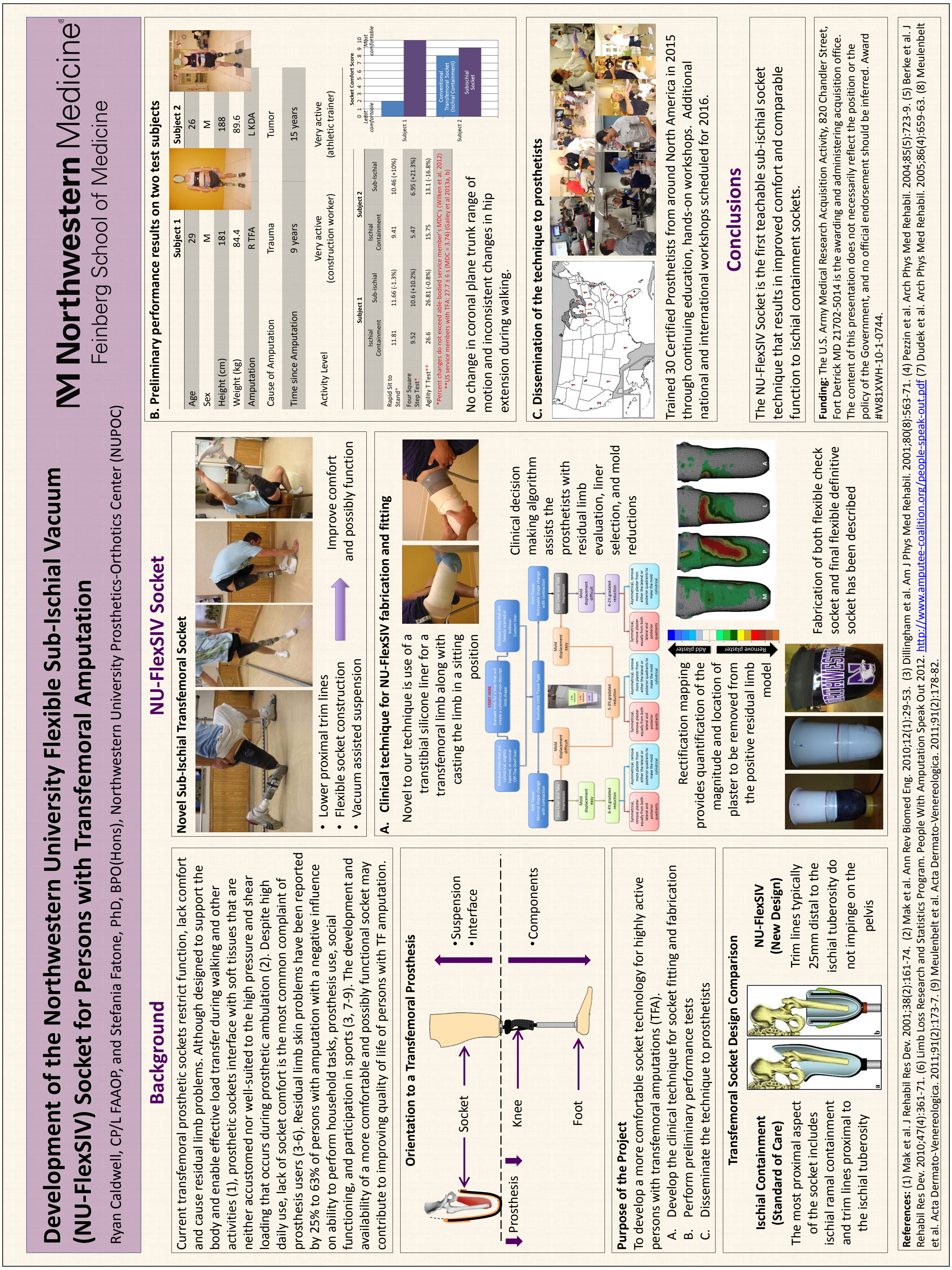


B
Background Material: Prosthetics

Included in this appendix are some background material for understanding how the socket system works. It is also some supplementary information about the liner and valve.

B.1 NU-FlexSIV Sockets from Northwestern University

B.1.1 NU-FlexSIV Socket - Technique Poster (Caldwell et al., 2016)



B.1.2 NU FlexSIV & NU FlexSIS Sockets - *Work Form* (Fatone and Caldwell, 2019)

NU FlexSIV & NU FlexSIS Sockets Work Form

PATIENT INFORMATION:

Name: ____

Date: __________________ Amputation Side: ☐Left ☐Right

Age (years): ___________ Weight (kg): _______________

Sex: ☐ M ☐ F Activity K-Level: _____________

PATIENT MEASUREMENTS:

For liner selection:

Circumference 4cm above distal end when seated (cm):

____

Liner Type/Size: ________________________________

For prosthesis/socket set up:

(Numbers correspond to image at top right)

2 Ischial Tuberosity to Distal End (cm): ____________

3 Ischial Tuberosity to Distal Femur (cm): __________

Socket Flexion (degrees): ____________________

Socket Adduction (degrees): _________________

For socket rectification:

(Numbers correspond to image at top right)

1 1 U proximal M-L (cm):______________

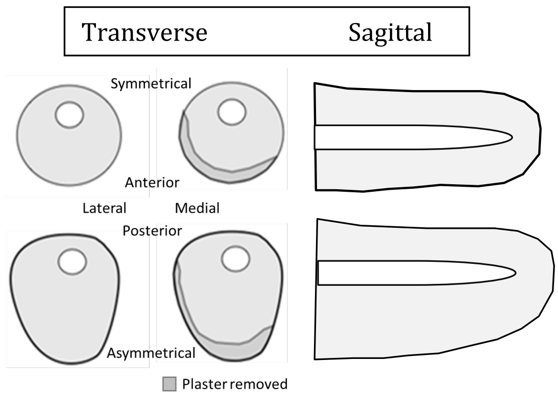
1 1 Compressed proximal M-L (cm):____________

(Refer to Mold Reduction Algorithm)

Limb Tissue Type: ☐Soft Tissue ☐Firm Tissue

Mold Displacement: ☐Easy ☐Difficult

Limb Profile: ☐Symmetrical ☐Asymmetrical



Coronal View

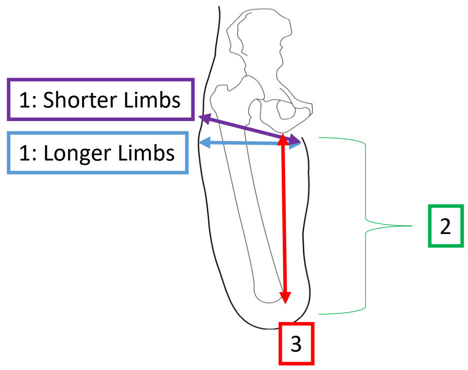
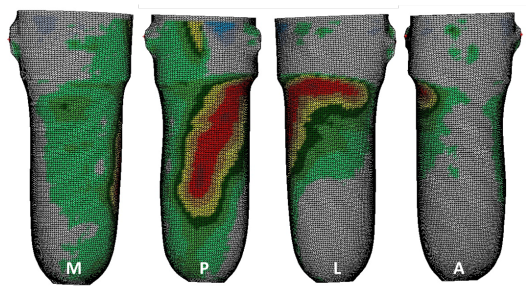


Figure adapted from CAD Measurement Chart, Advanced O&P Solutions, Hickory Hills, IL 60457. Used with permission.

Rectification Reference:



Pictured (left to right): medial, posterior, lateral, anterior

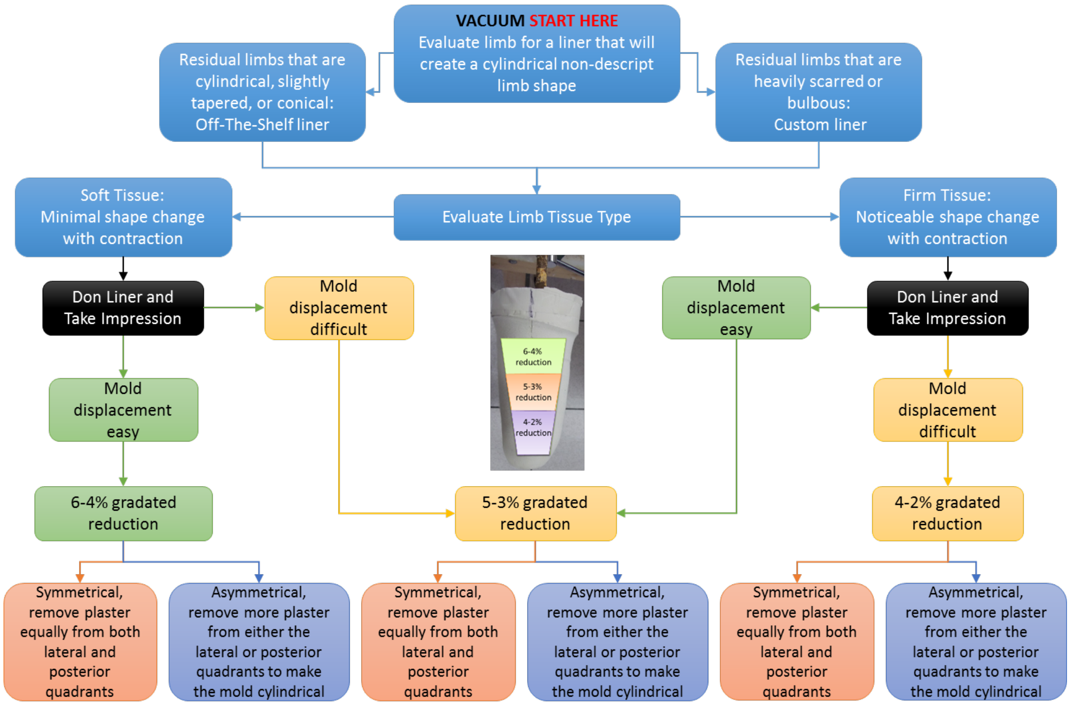
Mold Measurements:

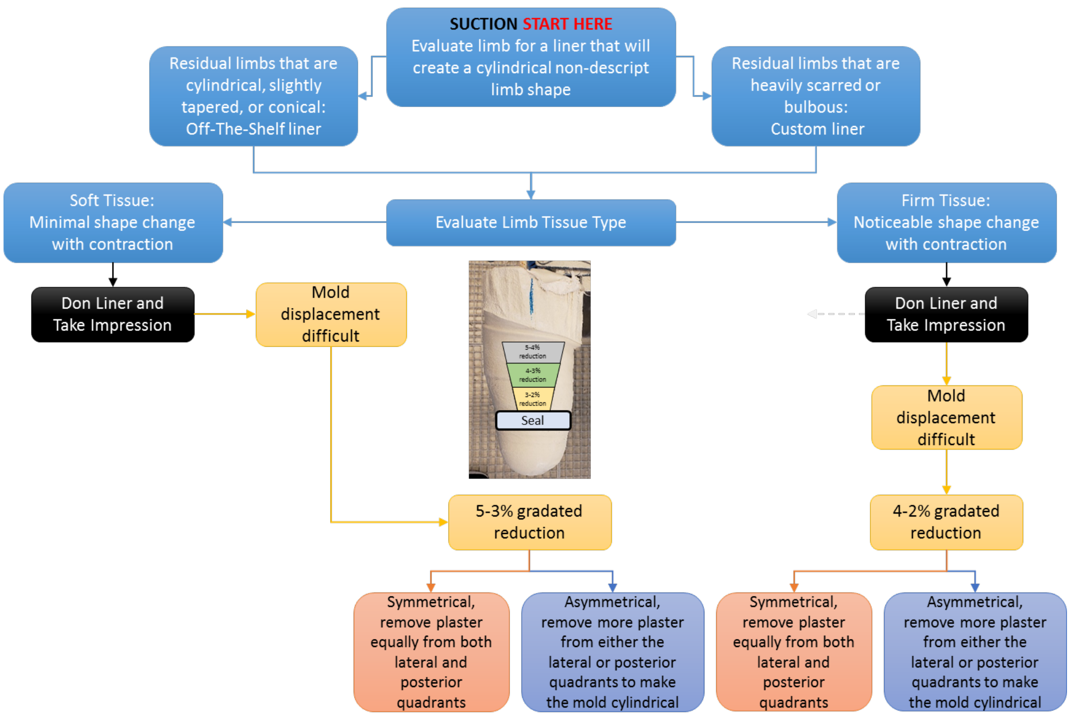
Starting 1" (2.5cm) below proximal medial trim line

Location	% Reduction	Actual Circumference	Goal Circumference
1" (2.5cm)			
2" (5 cm)			
3" (7.5 cm)			
4" (10 cm)			
5" (12.5 cm)			
6" (15 cm)			
7" (17.5 cm)			
8" (20 cm)			
9" (22.5cm)			
10" (25 cm)			
11" (27.5 cm)			
12" (30 cm)			

Mold Reduction Algorithm for NU-FlexSIV Socket

Page 2 of 2: NU FlexSIV & NU FlexSIS Sockets - *Work Form* (Fatone and Caldwell, 2019)



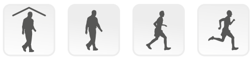


B.2 Product catalog: Össur Liner Iceross Seal-In

Product website: https://www.ossur.com/en-us/prosthetics/liners/iceross-seal-in-transfemoral

ÖSSUR

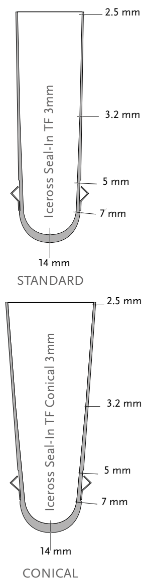
ICEROSS SEAL-IN® TRANSFEMORAL



The Iceross Seal-In Transfemoral liner offers uncompromised security and freedom of movement for low to extremely active transfemoral amputees. With textile that has improved the adhesion by 25%, the fabric cover is extremely durable and allows radial stretch and comfortable elasticity.

Össur recommends that Iceross Seal-In Transfemoral Liner is used in conjunction with the Icelock® 500 Series Expulsion Valve.

User Information	
Amputation Level:	Transfemoral
Impact Level:	All
Liner Information	
Size Standard:	25,26.5,28,30,32,34,36,38,40,45,50,55.
Size Conical:	25,26.5,28,30,32,34,36,38.
Profile:	3mm
Matrix:	Full Length



SUSPENSION METHOD

SEAL-IN

FEATURES

STABILIZING MATRIX | FABRIC COVER | ACTIVE SKIN CARE | SILKEN INNER SURFACE | CONICAL

MINIMUM RECOMMENDED LENGTH OF RESIDUAL LIMB

Liner Sizes	Mininum Length
25-30	125 mm
32-40	135 mm
45-50	155 mm
55	165 mm

PART NUMBER INFORMATION - ICEROSS TRANSFEMORAL SEAL-IN

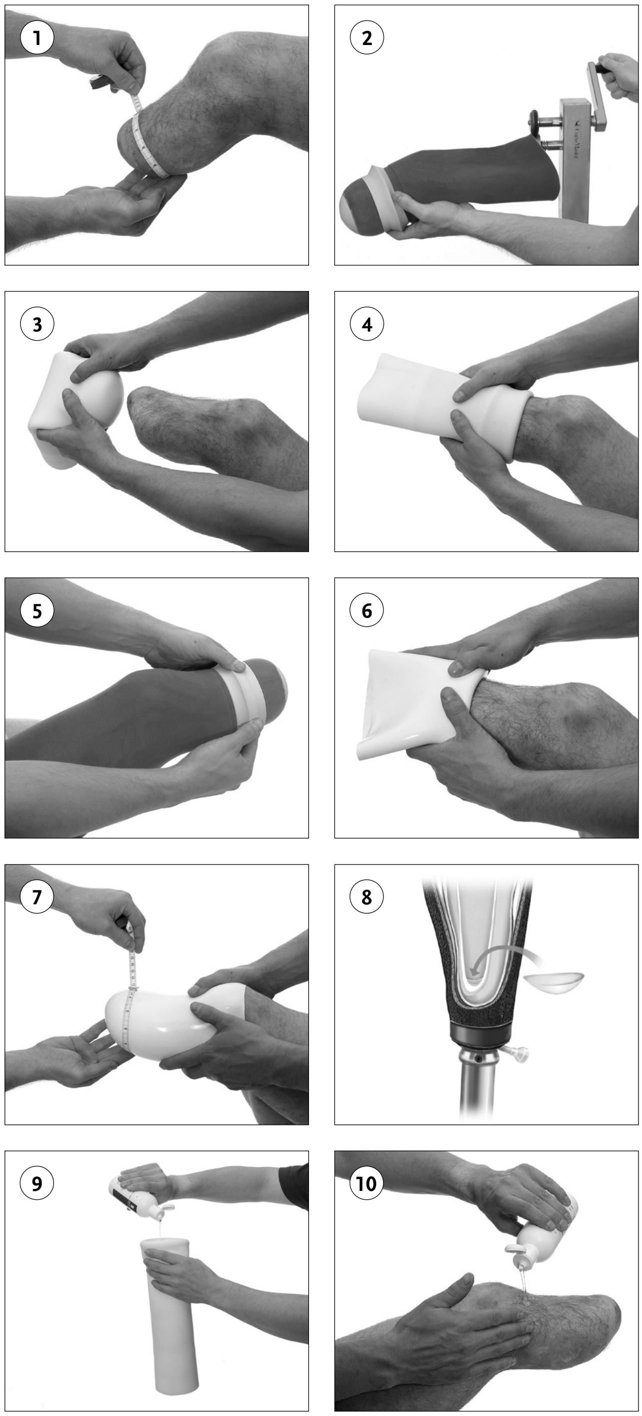
Part#	Profile	Suspension Method	Matrix
I-7532XX	Standard 3mm	Seal-In	Full Length
I-7632XX	Conical 3mm	Seal-In	Full Length

B.2.1 Instruction for use: Össur Liner Iceross Seal-In

ÖSSUR®
LIFE WITHOUT LIMITATIONS

Instructions for Use

ICEROSS SEAL-IN® LINER



3

Instruction for use: Össur Liner Iceross Seal-In

ENGLISH

INTRODUCTION
Thank you for choosing a high quality silicone liner from Össur. We can assure you that you have made the right choice. Please carefully review and follow these instructions before fitting a patient with the liner or accessory products. Remember that informing the patient concerning proper handling and care of the liner, as set forth in these instructions, is important to achieve patient comfort and maximum product life.

PRODUCT SELECTION AND FITTING (Figure 1)
Liner Sizing
To determine the correct Iceross size:
1. Measure the circumference of the residual limb at 4cm from the distal end.
2. Choose the Iceross size, which corresponds to the measurement or which, is immediately below the indicated measurement.
3. Never choose a liner size, which exceeds the measurement, or round up.

Example: If the residual limb measures 24.5cm at the indicated location, choose Iceross size 23.5 NOT size 25.

It is essential that the Hypobaric Sealing Membrane (HSM™) sits correctly against the liner around its full circumference. To ensure the HSM is correctly positioned, run your fingers around the seal and assess the tension.

To ensure optimal suspension with the Iceross Seal-In liner there must be a slight tension around the proximal edge of the HSM seal. If the seal is at all loose than a crease may form when donning the socket, this may be prevented by selecting the next liner size below.If the liner is too tight, this may cause pistoning and/or numbness and blistering. A Seal-In Liner that is too tight will also produce excessive ring tension to the residual limb from the HSM. If the liner is too loose, increased perspiration and/or movement of the limb inside the liner may occur. Perspiration and movement can lead to blisters and rashes. If any of the above symptoms or any other indication of improper fit are noted, the user should be instructed to contact his/her physician or prosthetist immediately.

Note: The matrix present within the liner extends 10cm from the distal end of the liner. Ensure that this matrix does not extend over the fibular head.

Trimming an Iceross Liner (Figure 2)
Iceross liners may be trimmed for user preference and to increase range of motion. The Iceross CurveMaster is specifically designed for this function and to produce a smooth, rounded edge, which will minimize the risk of skin irritation and/or tears of the liner. Do not cut the liner below the level of the matrix or below the proximal trim-lines of the socket. Excessive trimming may reduce the natural suction between the liner and limb and/or compromise suspension.

Donning an Iceross Liner
- When donning the liner, turn the liner inside out and grip it as shown in the illustration (Figure 3).

Note: Be very careful that the inside of the liner is clean, dry and free from any foreign objects that can cause skin irritation.

- After exposing as much of the distal end of the liner as possible, position it against the residual limb and with light compression roll upward onto the limb. Check that no air pockets are present (Figure 4).
- Roll the liner all the way up the limb, taking care not to damage it with fingernails. Do not tug or pull (Figure 5).
- Ensure the Silicone Membrane is not inverted and sits correctly around its full circumference (Figure 6).

Note: Inverting the Iceross Seal-In Liner and returning it to its original position will soften the silicone and assist with donning of the liner when new.

Iceross Distal Cup
Iceross Distal Cups provide enhanced comfort for conical limbs with poor soft tissue coverage. They can also be used as a fitting tool in conjunction with the Iceross Original liners to ensure total contact between the silicone and residual limb. The DERMOGEL silicone in the Iceross Distal Cup conforms to the limb, thus minimizing pressure over bony prominence during gait.

To correctly fit a Distal Cup:
1. Measure the circumference of the residual limb at 4cm from the distal end (Figur 1).
2. Choose the Distal Cup two or three sizes smaller than the measurement.
3. Roll the Distal Cup directly over the limb (Figure 6).
4. When using an Iceross liner in combination with a Distal Cup, the liner measurement (4cm from the distal end) should be made with the Distal Cup in place (i.e. OVER the Distal Cup) (Figure 7).
5. Choose the correct Iceross liner size based on the measurement obtained with the Distal Cup in place

Iceross Pads (Figure 8)
Iceross Pads can be used to fill in any problem areas (i.e. voids or scars) to ensure better fit and to minimize trapped air. Iceross Pads may also be used to distribute body pressure and to protect the skin. In some cases, custom-formed pads may be required for irregular shapes or problem areas.

Liner Care (Figure 9)
Proper product care is important. The liner needs to be washed inside and out every day, following use. Your patient should be instructed to remove the liner, turn the liner inside out and wash it with pH balanced, 100% fragrance- and dye-free soap. The liner can also be machine washed (40°C/Hot) with a mild detergent. Fabric softeners or bleaches or other products or cleaning solutions may damage the liner and should NOT be used. After cleaning, the patient should rinse the liner thoroughly with water and wipe both sides dry with a lint-free cloth.

Instruction for use: Össur Liner Iceross Seal-In

An Iceross liner can be used directly after washing. It does not need to dry over night. Always return the liner to its normal position with the distal attachment facing out as soon as possible after cleaning. Ensure the Silicone Membrane is positioned correctly on the liner following washing. The Silicone Membrane must be clean and free from dirt prior to use.

Note: Misuse may result in potential loss of suspension. Iceross liners should always be checked for damage or wear. Any damage may weaken the effectiveness of the liner and should be reported to the user's prosthetist immediately. Care must be taken by both the prosthetist and the patient not to expose a liner to glass or carbon fibers or other foreign particles. Such substances can become embedded in the silicone causing aggravation of the skin. Washing alone may not be sufficient to eliminate the problem and the patient should be instructed to return the liner to the prosthetist if the liner is inadvertently exposed to any foreign substances or chemicals.

DermoSil and DermoGel silicone contain Aloe Vera extract. This acts as a natural moisturizer and is known for its active healing and soothing effects on skin. Initially, the inner surface of the liner should be moist to the touch. For these reasons, the liner should not be left inside out for extended period or exposed to excessive heat or sunlight.

***Skin Care* (Figure 10)**
Daily cleaning of the residual limb is also essential. We recommend use of a mild liquid, pH balanced, 100% fragranceand dye-free soap.
If dry skin is noted, apply a pH balanced, 100% fragrance- and dye-free lotion to nourish and soften the skin.
Do **NOT** apply lotion to the residual limb immediately prior to donning the liner.

Note: Many common household or bath products, including soaps, deodorants, perfumes, aerosol or alcohol sprays or abrasive cleaners may cause or contribute to skin irritation and the patient should be so advised.

Product Warranty
Össur offers a written warranty on the Iceross Seal-In Liner for the indicated period from the invoice date. Said warranty shall be in lieu of all others, whether express or implied by law.

Iceross Seal-In Liner	6 months

ICEROSS SEAL-IN SOCK
If volume reduction is experienced in the residual limb, Seal-in socks may be added to restore the socket fit.

Seal-in sock application:
- Invert the HSM seal completely in the distal direction.
- Pull the Seal-in sock over the inverted HSM seal until the distal end of the sock is level with the proximal border of the seal.
- Reflect the HSM seal over the end of the sock.
- Don the prosthesis as normal.

Note: Loss of suspension may be encountered if a sock is placed over the

HSM seal. The Seal-in Liner will function with up to 6-ply thickness (2 x 3ply).

To be washed at 50°C with light coloured fabric.
Do not tumble dry.

MADE OF:

95%	Polyester
5%	Lycra

For further information, please contact Össur or your local Össur distributor. You can also visit the Össur web page at www.ossur.com.

B.3 Product catalog: Össur Icelock 500 Series

ICELOCK® 500 SERIES

Both Icelock Expulsion Valve 551 and 552 are designed to be used with all suction suspension systems. The valves are positioned in a housing that seals securely against the inner surface of the hard socket near the distal end. The valves expell air under positive pressure when entering a socket. During removal of the prosthesis the push button needs to be compressed to allow air back into the hard socket. The Icelock 500 Series can be used for all impact levels.

PART#	DESCRIPTION
L-551002	Icelock Expulsion Valve 551

The Icelock 551 Valve is an auto-expulsion and push button suction release in the same compact, easy to install package. (M 10). The valve is recommended for both Transtibial and Transfemoral users. The kit includes a flexible socket housing.

PART#	DESCRIPTION
L-552000	Icelock Expulsion Valve 552

Icelock 552 Valve is an auto-expulsion valve with push button for suction release. The core of the valve can be unscrewed to facilitate rapid donning and it makes it possible to pull donning socks through the valve. The valve can be used with a flexible interface and is recommended for Transfemoral users.

ICELOCK®

C
Arduino Sketches

Contents of Appendix C

C.1 Test Rig Sketches

C.1.1 20200109 Testrig All.ino

This sketch has been most stable during my testing in this project. It includes realtime output through serial monitor and OLED, but does not save to SD-card.

```
//------------- Reading Loadcells and Force Sensitive Resistors with Arduino Mega ADK  ------
//    Written by Josefine C. S. Haugom, NTNU
//    Trondheim, February 2020
//    Based on HX711_ADC.h library and FRS testing sketch from
//    https://learn.adafruit.com/force-sensitive-resistor-fsr
//    Load Cell 1 - Load cell 10 kg - via HX711 Loadcell amplifier - input to pin 2, 3
//    Load Cell 2 - Load cell disk 200 kg - via HX711 Loadcell amplifier - input to pin 4, 5
//    FSRs connected to 5V, input to A0-A3, connected with 10kOhm resistor
//    I2C OLED Display 128x64px is connected to SCL (pin 21) and SDA (pin 20)
//
//    Both Load Cells (LC) needs calibration. Calibration values achieved
//    by HX711_ADC-librarys example code
//-------------------------------------------------------------------------------------------

#include <Arduino.h>   // Required if code is used in PlatformIO
#include <HX711_ADC.h> // Load Cell amplifier HX711 library

//------------  Screen: SSD1306 OLED 0.96" 128 x 64px ---------------------------------------
// OLED with 4 output pins: GND, Power (NB: 3.3V), SDA and SLC (NB: different pinnr between Uno and Mega )
//
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

// 'ntnu-logo', 128x64px
const unsigned char ntnu[] PROGMEM = {
\\BITMAP REMOVED DUE TO PDF COMPILING CAPACITY};

//-------------------------------------------------------------------------------------------

/******      BUTTONS     **********/
const int buttonOK = 8;    // OK-button input D8
const int buttonTEAR = 9;  // TEAR-button input D9

int buttonStateOK = 0;   // variable for reading the pushbutton status
int buttonStateTear = 0; // variable for reading the pushbutton status

/****** CONSTANTS - INPUTS *******/

//------- HX711 constructor (dout pin, sck pin)
HX711_ADC LoadCell_1(2, 3); //HX711 1 - 10 kg Load Cell
HX711_ADC LoadCell_2(4, 5); //HX711 2 - 200 kg Disk Load Cell
long t;

//-------------------- FSR constants
// Each FSR has two sensor outlets, one direct to ground,
// second connected to ANALOG PIN with 10kOhm resistor between (Sensor - 10kOhm - A0)
// Four FSRs placed on the test rig stump leg:
//    -   I01     - A0 -    Square FSR - Between MDF pos: medial, mid axial (          |           )
//    -   Medial  - A1 -    Square FSR - Medial - mid axial                   (    A2   |   A0    ) A1
//    -   Lateral - A2 -    Square FSR - Lateral - mid axial                  (         |         )
//    -   Distal  - A3 -    Round FSR -  Distal tip                            \_____ D4/D5______/
//    -------------------------------------------------------                    \______________/
//                                                                                    A3
int fsr_I01 = A0;
int fsr_Read_I01;
float fsr_V_I01;

int fsr_MedM = A1;
int fsr_Read_MedM;
float fsr_V_MedM;

int fsr_LatM = A2;
int fsr_Read_LatM;
```

```
float fsr_V_LatM;

int fsr_DISTAL = A3;
int fsr_Read_Distal;
float fsr_V_Distal;

const int ledPinRed = 12;   //Status LEDs - program start - led on
const int ledPinGreen = 13; //Status LEDs - loop start    - led on

float calValue_1; // calibration value load cell 1
float calValue_2; // calibration value load cell 2

void setup()
{
    Serial.begin(9600);
    Serial.println("Starting␣up");
    calValue_1 = 6.80;    //Use HX711_ADC Calibration library example to find calibration factor of the day
    calValue_2 = 1543.60; //Use HX711_ADC Calibration library example to find calibration factor of the day - for each load cell

    pinMode(ledPinRed, OUTPUT);    //  Defining led pin as output
    pinMode(ledPinGreen, OUTPUT);  //  --
    digitalWrite(ledPinRed, HIGH); //  Arduino setup going - Red Led on!

    if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C))
    { // OLED  - Address 0x3D for 128x64 Check right adress if its not working
        Serial.println(F("SSD1306␣allocation␣failed"));
        for (;;)
        ;
    }
    delay(2000);

    display.clearDisplay();   // Starting up OLED - Welcome with NTNU logo
    display.invertDisplay(1); // Inverted logo = white on black background
    display.drawBitmap(0, 0, ntnu, 128, 64, WHITE);
    display.display(); // Make abow display commands visible on OLED
    delay(2000);       // Let stand for 2000 ms

    display.clearDisplay();   // Initiallizing - could go simultaniously with tear ?
    display.invertDisplay(0); // If power consuming, drop this
    display.setTextSize(1);
    display.setTextColor(WHITE);
    display.setCursor(10, 10);
    display.println("Initializing␣\n␣sensors.␣.␣.␣");
    display.setTextSize(2);
    display.setCursor(40, 30);
    display.write(2);
    display.display();
    display.startscrolldiagright(0x00, 0x07);
    delay(4000);
    display.stopscroll(); //--------- Until this step -------------//

    LoadCell_1.begin();
    LoadCell_2.begin();
    long stabilisingtime = 2000; // tare preciscion can be improved by adding a few seconds of stabilising time
    byte loadcell_1_rdy = 0;
    byte loadcell_2_rdy = 0;
    while ((loadcell_1_rdy + loadcell_2_rdy) < 2)
    { //run startup, stabilization and tare, both modules simultaniously
        if (!loadcell_1_rdy)
        loadcell_1_rdy = LoadCell_1.startMultiple(stabilisingtime);
        if (!loadcell_2_rdy)
        loadcell_2_rdy = LoadCell_2.startMultiple(stabilisingtime);
    }
    LoadCell_1.setCalFactor(calValue_1); // user set calibration value (float)
    LoadCell_2.setCalFactor(calValue_2); // user set calibration value (float)

    digitalWrite(ledPinRed, LOW);
    delay(100);
    digitalWrite(ledPinRed, HIGH);
    delay(100);
    digitalWrite(ledPinRed, LOW);
    delay(100);
    digitalWrite(ledPinRed, HIGH);
}
/*   UNCOMMENT if the calibration vaules are desired to be showed on the OLED display
void cal_values(){

    display.clearDisplay();
```

```
    display.setTextSize(1);
    display.setCursor(5, 20);
    display.println("Calibration values: \n");
    display.write(26);
    display.print(F("  LC 1 = "));
    display.println(calValue_1);
    display.write(25);
    display.print(F("  LC 2 = "));
    display.println(calValue_2);
    display.display();
    delay(8000);
}

*/

void loop()
{

    /* TESTING BUTTONS */
    buttonStateOK = digitalRead(buttonOK);     // variable for reading pushbutton status (OK)
    buttonStateTear = digitalRead(buttonTEAR); // variable for reading pushbutton status (TEAR)

    if (buttonStateOK == HIGH)
    {
        digitalWrite(ledPinRed, LOW);
        Serial.println("OK␣Button␣pushed");
    }
    if (buttonStateTear == HIGH)
    {
        digitalWrite(ledPinRed, HIGH);
        Serial.println("Tear␣Button␣pushed");
    }

    /* TESTING BUTTONS FINISH */
    digitalWrite(ledPinGreen, HIGH);

    //---------------  HX711  Loop  --------------
    //update() should be called at least as often as HX711 sample rate; >10Hz@10SPS, >80Hz@80SPS
    //longer delay in scetch will reduce effective sample rate (be carefull with use of delay() in the
        loop)
    LoadCell_1.update();
    LoadCell_2.update();

    //get smoothed value from data set + current calibration factor
    /**/
    if (millis() > t + 250)
    {
        float a = LoadCell_1.getData();
        float b = LoadCell_2.getData();
        /* ***Uncomment the ifs to remove the extreme values when the
        *****Load Cells are calibrating (espceially when using serial plot.)
        if (a < 300) Serial.print(a);
        if (a > 300) Serial.print("0");
        Serial.print(", ");
        if (b < 300) Serial.print(b);
        if (b > 300) Serial.print("0");
        Serial.print(", ");
        */
        Serial.print(a);    // COMMENT THIS AWAY IF USING THE if (a < 300 ) ... above
        Serial.print(",␣"); //
        Serial.print(b);
        Serial.print(",␣");

        t = millis();

        //-------------- FSR Loop ----------------
        fsr_Read_I01 = analogRead(fsr_I01);
        fsr_V_I01 = map(fsr_Read_I01, 0, 1023, 0, 5000);

        fsr_Read_Distal = analogRead(fsr_DISTAL);
        fsr_V_Distal = map(fsr_Read_Distal, 0, 1023, 0, 5000);

        fsr_Read_MedM = analogRead(fsr_MedM);
        fsr_V_MedM = map(fsr_Read_MedM, 0, 1023, 0, 5000);

        fsr_Read_LatM = analogRead(fsr_LatM);
        fsr_V_LatM = map(fsr_Read_LatM, 0, 1023, 0, 5000);

        Serial.print(fsr_V_I01 / 1000);
        Serial.print(",␣");
        Serial.print(fsr_V_Distal / 1000);
```

```
        Serial.print(",␣");
        Serial.print(fsr_V_MedM / 1000);
        Serial.print(",␣");
        Serial.println(fsr_V_LatM / 1000);

        //OLED Loop
        display.clearDisplay();
        display.invertDisplay(0);
        display.setTextSize(1);
        display.setTextColor(WHITE);
        display.setCursor(0, 8);

        display.print("Rotation:␣␣");
        if (a < 2000)
        display.print(a);
        display.println("␣N");

        display.print("Axial␣␣␣:␣␣");
        if (b < 2000)
        display.print(b);
        display.println("␣N␣\n");

        display.drawLine(0, 28, 60, 28, WHITE);
        display.drawLine(63, 28, 65, 28, WHITE);
        display.drawLine(68, 28, 128, 28, WHITE);

        display.print("FSR␣I01␣␣␣␣:␣␣");
        display.print(fsr_V_I01 / 1000);
        display.println("␣V");

        display.print("FSR␣Distal␣:␣␣");
        display.print(fsr_V_Distal / 1000);
        display.println("␣V");

        display.print("FSR␣Medial␣:␣␣");
        display.print(fsr_V_MedM / 1000);
        display.println("␣V");

        display.print("FSR␣Lateral:␣␣");
        display.print(fsr_V_LatM / 1000);
        display.println("␣V");

        display.display();
    }
    /**/

    //receive from serial terminal
    if (Serial.available() > 0)
    {
        float i;
        char inByte = Serial.read();
        if (inByte == 't')
        {
            LoadCell_1.tareNoDelay();
            LoadCell_2.tareNoDelay();
        }
    }

    //delay(2000);
}
```

C.1.2 Calibrating the Load Cells

```
//----------------------- HX711 LOAD CELL CALIBRATION -------------------------------------
// Modified by Josefine C. S. Haugom, January 2020
//Based on Arduino Master library HX711_ADC.h for HX711
//               24-Bit Analog-to-Digital Converter for Weigh Scales by Olav Kallhovd sept2017
// LoadCell 1: TAL220 Parallel Beam Load Cell 10kg
// LoadCell 2: TAS606 Button Type Load Cell 200kg
//-------------------------------------------------------------------------------------
// This is an example sketch on how to use this library
// Settling time (number of samples) and data filtering can be adjusted in the config.h file

// This example shows how to calibrate the load cell and optionally save the calibration
// value to EEPROM, and also how to change the value.
// The value can later be fetched from EEPROM in your project sketch.

```

```
#include <HX711_ADC.h>
#include <EEPROM.h>

//HX711 constructor (dout pin, sck pin):
HX711_ADC LoadCell(2, 3); //loadcell 1
HX711_ADC LoadCell(4, 5); //loadcell 2 - comment the one which should not be calibrated this round

int eepromAdress = 0;

long t;

void calibrate() {
Serial.println("***");
Serial.println("Start␣calibration:");
Serial.println("It␣is␣assumed␣that␣the␣mcu␣was␣started␣with␣no␣load␣applied␣to␣the␣load␣cell.");
Serial.println("Now,␣place␣your␣known␣mass␣on␣the␣loadcell,");
Serial.println("then␣send␣the␣weight␣of␣this␣mass␣(i.e.␣100.0)␣from␣serial␣monitor.");
float m = 0;
boolean f = 0;
while (f == 0) {
LoadCell.update();
if (Serial.available() > 0) {
m = Serial.parseFloat();
if (m != 0) {
Serial.print("Known␣mass␣is:␣");
Serial.println(m);
f = 1;
}
else {
Serial.println("Invalid␣value");
}
}
}
float c = LoadCell.getData() / m;
LoadCell.setCalFactor(c);
Serial.print("Calculated␣calibration␣value␣is:␣");
Serial.print(c);
Serial.println(",␣use␣this␣in␣your␣project␣sketch");
f = 0;
Serial.print("Save␣this␣value␣to␣EEPROM␣adress␣");
Serial.print(eepromAdress);
Serial.println("?␣y/n");
while (f == 0) {
if (Serial.available() > 0) {
char inByte = Serial.read();
if (inByte == 'y') {
#if defined(ESP8266)
EEPROM.begin(512);
#endif
EEPROM.put(eepromAdress, c);
#if defined(ESP8266)
EEPROM.commit();
#endif
EEPROM.get(eepromAdress, c);
Serial.print("Value␣");
Serial.print(c);
Serial.print("␣saved␣to␣EEPROM␣address:␣");
Serial.println(eepromAdress);
f = 1;

}
else if (inByte == 'n') {
Serial.println("Value␣not␣saved␣to␣EEPROM");
f = 1;
}
}
}
Serial.println("End␣calibration");
Serial.println("For␣manual␣edit,␣send␣'c'␣from␣serial␣monitor");
Serial.println("***");
}

void changeSavedCalFactor() {
float c = LoadCell.getCalFactor();
boolean f = 0;
Serial.println("***");
Serial.print("Current␣value␣is:␣");
Serial.println(c);
Serial.println("Now,␣send␣the␣new␣value␣from␣serial␣monitor,␣i.e.␣696.0");
while (f == 0) {
if (Serial.available() > 0) {
```

```
c = Serial.parseFloat();
if (c != 0) {
Serial.print("New␣calibration␣value␣is:␣");
Serial.println(c);
LoadCell.setCalFactor(c);
f = 1;
}
else {
Serial.println("Invalid␣value,␣exit");
return;
}
}
}
f = 0;
Serial.print("Save␣this␣value␣to␣EEPROM␣adress␣");
Serial.print(eepromAdress);
Serial.println("?␣y/n");
while (f == 0) {
if (Serial.available() > 0) {
char inByte = Serial.read();
if (inByte == 'y') {
#if defined(ESP8266)
EEPROM.begin(512);
#endif
EEPROM.put(eepromAdress, c);
#if defined(ESP8266)
EEPROM.commit();
#endif
EEPROM.get(eepromAdress, c);
Serial.print("Value␣");
Serial.print(c);
Serial.print("␣saved␣to␣EEPROM␣address:␣");
Serial.println(eepromAdress);
f = 1;
}
else if (inByte == 'n') {
Serial.println("Value␣not␣saved␣to␣EEPROM");
f = 1;
}
}
}
Serial.println("End␣change␣calibration␣value");
Serial.println("***");
}

void setup() {
Serial.begin(9600); delay(10);
Serial.println();
Serial.println("Starting...");
LoadCell.begin();
long stabilisingtime = 2000; // tare preciscion can be improved by adding a few seconds of stabilising
      time
LoadCell.start(stabilisingtime);
if (LoadCell.getTareTimeoutFlag()) {
Serial.println("Tare␣timeout,␣check␣MCU>HX711␣wiring␣and␣pin␣designations");
}
else {
LoadCell.setCalFactor(1.0); // user set calibration value (float)
Serial.println("Startup␣+␣tare␣is␣complete");
}
while (!LoadCell.update());
calibrate();
}
void loop() {
//update() should be called at least as often as HX711 sample rate; >10Hz@10SPS, >80Hz@80SPS
//longer delay in sketch will reduce effective sample rate (be carefull with delay() in the loop)
LoadCell.update();

//get smoothed value from the data set
if (millis() > t + 250) {
float i = LoadCell.getData();
Serial.print("Load␣cell␣output␣val:␣");
Serial.println(i);
t = millis();
}

//receive from serial terminal
if (Serial.available() > 0) {
float i;
char inByte = Serial.read();
if (inByte == 't') LoadCell.tareNoDelay();
```

```
else if (inByte == 'c') changeSavedCalFactor();
}

//check if last tare operation is complete
if (LoadCell.getTareStatus() == true) {
Serial.println("Tare complete");
}

}
```

C.1.3 Test Rig Sketch Including Datalogger and Tear Button

```
//---------------------------- TESTRIG SENSOR SKETCH 20200119              -----------------
//------------- Reading Loadcells and Force Sensitive Resistors With Button Tear -------------
//-
//-   Written by Josefine C. S. Haugom ----------------------
//-   January 2020, NTNU Trondheim
//-
//-----------------------------------------------------------
//    Based on HX711_ADC.h library, Arduino Datalogger and FRS testing sketch from
//    https://learn.adafruit.com/force-sensitive-resistor-fsr
//    Load Cell 1 - Load cell 10 kg - via HX711 Loadcell amplifier - input to pin 4, 5
//    Load Cell 2 - Load cell disk 200 kg - via HX711 Loadcell amplifier - input to pin 2, 3
//    FSRs connected to 5V, input to A0-A3, connected with 10kOhm resistor
//
//    Both Load Cells (LC) needs calibration. Calibration values achieved
//    by HX711_ADC-librarys example code
//--------------------------------------------------------------------------------------------

#include <Arduino.h>        // Required if code is used in PlatformIO
#include "inputpins.h"

void setup() {
Serial.begin(9600);
Serial.println("Starting up");
calValue_1 = 22008.00;
calValue_2 = 1141.20; //Use HX711_ADC Calibration library example to find calibration factor of the day - for each load cell


pinMode(ledPinRed, OUTPUT);                        //  Defining led pin as output
pinMode(ledPinGreen, OUTPUT);                      //  --
digitalWrite(ledPinRed, HIGH);                     //  Arduino setup going - Red Led on!

if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // OLED  - Address 0x3D for 128x64 Check right adress if its not working
Serial.println(F("SSD1306 allocation failed"));
for(;;);
}
delay(2000);

display.clearDisplay();                            // Starting up OLED - Welcome with NTNU logo
display.invertDisplay(1);                          // Inverted logo = white on black background
display.drawBitmap(0, 0,  ntnu, 128, 64, WHITE);
display.display();                                 // Make abow display commands visible on OLED
delay(2000);                                       // Let stand for 2000 ms

LoadCell_1.begin();
LoadCell_2.begin();
long stabilisingtime = 2000; // tare preciscion can be improved by adding a few seconds of stabilising time
byte loadcell_1_rdy = 0;
byte loadcell_2_rdy = 0;
while ((loadcell_1_rdy + loadcell_2_rdy) < 2) { //run startup, stabilization and tare, both modules simultaniously
if (!loadcell_1_rdy) loadcell_1_rdy = LoadCell_1.startMultiple(stabilisingtime);
if (!loadcell_2_rdy) loadcell_2_rdy = LoadCell_2.startMultiple(stabilisingtime);
}
LoadCell_1.setCalFactor(calValue_1); // user set calibration value (float)
LoadCell_2.setCalFactor(calValue_2); // user set calibration value (float)

digitalWrite(ledPinRed, LOW);
delay(100);
digitalWrite(ledPinRed, HIGH);
delay(200);

```

```
/*------------------- SD CARD SETUP -------------*/
// Open serial communications and wait for port to open:
Serial.begin(9600);
while (!Serial) {
; // wait for serial port to connect. Needed for native USB port only
}

Serial.print("Initializing␣SD␣card...");
display.clearDisplay();
display.invertDisplay(0);
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(0, 8);
display.println("Initializing␣SD␣card...␣\n␣\n");
display.display();

// see if the card is present and can be initialized:
if (!SD.begin(chipSelect)) {
Serial.println("Card␣failed,␣or␣not␣present");
display.println("Card␣failed,␣or␣not␣present");
display.display();
// don't do anything more:
//while (1);
}
if(SD.begin(chipSelect)){
Serial.println("card␣initialized.");
display.println("card␣initialized.");
display.display();
}
delay(2000);

digitalWrite(ledPinRed, LOW);
delay(100);
digitalWrite(ledPinRed, HIGH);
}

void loop() {

/* TESTING BUTTONS */
buttonStateOK   = digitalRead(buttonOK);    // variable for reading pushbutton status (OK)
buttonStateTear = digitalRead(buttonTEAR);  // variable for reading pushbutton status (TEAR)

if (buttonStateOK == HIGH){
digitalWrite(ledPinRed, LOW);
Serial.println("OK␣Button␣pushed");
}
if (buttonStateTear == HIGH){
digitalWrite(ledPinRed, LOW);
delay(100);
digitalWrite(ledPinRed, HIGH);
Serial.println("Tear␣Button␣pushed");
LoadCell_1.begin();
LoadCell_2.begin();
long stabilisingtime = 2000; // tare preciscion can be improved by adding a few seconds of stabilising time
byte loadcell_1_rdy = 0;
byte loadcell_2_rdy = 0;
while ((loadcell_1_rdy + loadcell_2_rdy) < 2) { //run startup, stabilization and tare, both modules simultaniously
if (!loadcell_1_rdy) loadcell_1_rdy = LoadCell_1.startMultiple(stabilisingtime);
if (!loadcell_2_rdy) loadcell_2_rdy = LoadCell_2.startMultiple(stabilisingtime);
}
LoadCell_1.setCalFactor(calValue_1); // user set calibration value (float)
LoadCell_2.setCalFactor(calValue_2); // user set calibration value (float)

digitalWrite(ledPinRed, LOW);
delay(100);
digitalWrite(ledPinRed, HIGH);
delay(300);
digitalWrite(ledPinRed, LOW);

}

/* TESTING BUTTONS FINISH */



digitalWrite(ledPinGreen, HIGH);

//--------------- HX711 Loop --------------
//update() should be called at least as often as HX711 sample rate; >10Hz@10SPS, >80Hz@80SPS
```

```
//longer delay in scetch will reduce effective sample rate (be carefull with use of delay() in the loop)
LoadCell_1.update();
LoadCell_2.update();

//get smoothed value from data set + current calibration factor
/**/
if (millis() > t + 250) {
float a = LoadCell_1.getData();
float b = LoadCell_2.getData();


if (a < 300) Serial.print(a);
if (a > 300) Serial.print("0.00");
Serial.print(",␣");
if (b < 300) Serial.print(b);
if (b > 300) Serial.print("0.00");
Serial.print(",␣");


t = millis();



//-------------- FSR Loop ----------------
fsr_Read_I01 = analogRead(fsr_I01);
fsr_V_I01 = map(fsr_Read_I01, 0, 1023, 0, 5000);

fsr_Read_Distal = analogRead(fsr_DISTAL);
fsr_V_Distal = map(fsr_Read_Distal, 0, 1023, 0, 5000);

fsr_Read_MedM = analogRead(fsr_MedM);
fsr_V_MedM = map(fsr_Read_MedM, 0, 1023, 0, 5000);

fsr_Read_LatM = analogRead(fsr_LatM);
fsr_V_LatM = map(fsr_Read_LatM, 0, 1023, 0, 5000);

Serial.print(fsr_V_I01/1000);
Serial.print(",␣");
Serial.print(fsr_V_Distal/1000);
Serial.print(",␣");
Serial.print(fsr_V_MedM/1000);
Serial.print(",␣");
Serial.println(fsr_V_LatM/1000);


//OLED Loop
display.clearDisplay();
display.invertDisplay(0);
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(0, 8);

display.print("Rotation:␣␣");
if (a < 2000) display.print(a);
display.println("␣N");

display.print("Axial␣␣␣:␣␣");
if (b < 2000) display.print(b);
display.println("␣N␣\n");

display.drawLine(0, 28, 60, 28, WHITE);
display.drawLine(63, 28, 65, 28, WHITE);
display.drawLine(68, 28, 128, 28, WHITE);


display.print("FSR␣I01␣␣␣␣:␣␣");
display.print(fsr_V_I01/1000);
display.println("␣V");

display.print("FSR␣Distal␣:␣␣");
display.print(fsr_V_Distal/1000);
display.println("␣V");

display.print("FSR␣Medial␣:␣␣");
display.print(fsr_V_MedM/1000);
display.println("␣V");

display.print("FSR␣Lateral:␣␣");
display.print(fsr_V_LatM/1000);
display.println("␣V");

```

```
display.display();


/*****  DATALOGGER ******************************/
String dataString = "";    //Make a string for assembling the data to log.

//OUTPUT
//a , b , fsr_V_I01/1000, fsr_V_Distal/1000, fsr_V_MedM/1000, fsr_V_LatM/1000;

dataString += String( a );
dataString += ", ";
dataString += String( b );
dataString += ", ";
dataString += String(fsr_V_I01 /1000);
dataString += ", ";
dataString += String(fsr_V_Distal/1000);
dataString += ", ";
dataString += String(fsr_V_MedM/1000);
dataString += ", ";
dataString += String(fsr_V_LatM/1000);

File dataFile = SD.open("sensorlog.txt", FILE_WRITE);

if (dataFile) {
dataFile.println(dataString);
dataFile.close();
}
else {
Serial.println("error opening sensorlog.txt");
}
}
/**/






/****** END DATALOGGER *************************/


//receive from serial terminal
if (Serial.available() > 0) {
float i;
char inByte = Serial.read();
if (inByte == 't') {
LoadCell_1.tareNoDelay();
LoadCell_2.tareNoDelay();
}
}



//delay(2000);
}
```

C.1.4 Continuation of Test Rig Sketch Using Custom Libraries

The following sketch is a program where setup for different subsystems in the test rig are divided into library files (.h). Note: There has been some instability with the microSD-module during the testing.

C.1.5 20200119 Testrigg All.ino

```
//---------- TESTRIG SENSOR SKETCH 20200119 -----------------
//-   Written by Josefine C. S. Haugom ----------------------
//-   January 2020, NTNU Trondheim
//-   Prototype program
//-   Some problems of finding SD card with Arduino ADK
//-----------------------------------------------------------

#include <Arduino.h>
#include <Wire.h>
#include "realtime.h"
#include "user_control.h"
#include "fsr_setup.h"
```

```
#include "hx711_setup.h"
#include "oled_setup.h"
#include "sd_setup.h"

void setup()
{
Serial.begin(9600);
//  timeSetup();

Serial.print("Starting␣up␣at:␣");
Serial.println("␣");

controlSetup();
OLEDsetup();
calValue_1 = 12.44;    //Use HX711_ADC Calibration library example to find calibration factor of the day
calValue_2 = 1543.60; //Use HX711_ADC Calibration library example to find calibration factor of the day
      - for each load cell
hx711Setup();
}

void loop()
{

//  timeLoop();
loopTearing();
sensorLoop();
dispSensors();
datalogging();

delay(250);
}
```

fsr_setup.h

```
//---------- TESTRIG SENSOR SKETCH 20200119 -----------------
//----------------    fsr_setup.h    -------------------------
//-   Written by Josefine C. S. Haugom ----------------------
//-   January 2020, NTNU Trondheim
//----------------------------------------------------------

int fsr_I01 = A0;
int fsr_Read_I01;
float fsr_V_I01;


int fsr_MedM = A1;
int fsr_Read_MedM;
float fsr_V_MedM;

int fsr_LatM = A2;
int fsr_Read_LatM;
float fsr_V_LatM;

int fsr_DISTAL    = A3;
int fsr_Read_Distal;
float fsr_V_Distal;


// void fsr_loop(void){
//      String fsrString ="";

//      fsr_Read_I01 = analogRead(fsr_I01);
//      fsr_V_I01 = map(fsr_Read_I01, 0, 1023, 0, 5000);
//      fsr_Read_Distal = analogRead(fsr_DISTAL);
//      fsr_V_Distal = map(fsr_Read_Distal, 0, 1023, 0, 5000);

//      fsr_Read_MedM = analogRead(fsr_MedM);
//      fsr_V_MedM = map(fsr_Read_MedM, 0, 1023, 0, 5000);

//      fsr_Read_LatM = analogRead(fsr_LatM);
//      fsr_V_LatM = map(fsr_Read_LatM, 0, 1023, 0, 5000);

//      fsrString += String(fsr_V_I01);
//      fsrString += ", ";
//      fsrString += String(fsr_V_Distal);
//      fsrString += ", ";
//      fsrString += String(fsr_V_MedM);
//      fsrString += ", ";
```

```
//     fsrString += String(fsr_V_LatM);

//     /*
//     Serial.print(fsr_V_I01/1000);
//     Serial.print(", ");
//     Serial.print(fsr_V_Distal/1000);
//     Serial.print(", ");
//     Serial.print(fsr_V_MedM/1000);
//     Serial.print(", ");
//     Serial.println(fsr_V_LatM/1000);
//     */
//    Serial.print("FSR values: ");
//    Serial.println(fsrString);
// }
```

hx711_setup.h

```
//---------- TESTRIG SENSOR SKETCH 20200119 -----------------
//----------------   hx711_setup.h    -------------------------
//-   Written by Josefine C. S. Haugom ----------------------
//-   January 2020, NTNU Trondheim
//----------------------------------------------------------


#include <HX711_ADC.h> // Load Cell amplifier HX711 library

//------- HX711 constructor (dout pin, sck pin)
HX711_ADC LoadCell_1(2, 3); //HX711 1 - 10 kg Load Cell
HX711_ADC LoadCell_2(4, 5); //HX711 2 - 200 kg Disk Load Cell
long t;
int sensorString = "";

float calValue_1; // calibration value load cell 1
float calValue_2; // calibration value load cell 2
int a = "";
int b = "";

void hx711Setup(void)
{
LoadCell_1.begin();
LoadCell_2.begin();
long stabilisingtime = 2000; // tare preciscion can be improved by adding a few seconds of stabilising time
byte loadcell_1_rdy = 0;
byte loadcell_2_rdy = 0;
while ((loadcell_1_rdy + loadcell_2_rdy) < 2)
{ //run startup, stabilization and tare, both modules simultaniously
if (!loadcell_1_rdy)
loadcell_1_rdy = LoadCell_1.startMultiple(stabilisingtime);
if (!loadcell_2_rdy)
loadcell_2_rdy = LoadCell_2.startMultiple(stabilisingtime);
}
LoadCell_1.setCalFactor(calValue_1); // user set calibration value (float)
LoadCell_2.setCalFactor(calValue_2); // user set calibration value (float)
}

//--------------  HX711  Loop  -------------
//update() should be called at least as often as HX711 sample rate; >10Hz@10SPS, >80Hz@80SPS
//longer delay in scetch will reduce effective sample rate (be carefull with use of delay() in the loop)

void sensorLoop(void)
{
LoadCell_1.update();
LoadCell_2.update();

String sensorString = "";

//get smoothed value from data set + current calibration factor
if (millis() > t + 250)
{
float a = LoadCell_1.getData();
float b = LoadCell_2.getData();

if (a < 400)
{
sensorString += String(a);
sensorString += ",␣";
}
```

```
if (a > 400)
{
sensorString += String("0, ");
}

Serial.print(", ");
if (b < 400)
{
sensorString += String(b);
sensorString += ", ";
}
if (b > 400)
{
sensorString += "0, ";
}

fsr_Read_I01 = analogRead(fsr_I01);
fsr_V_I01 = map(fsr_Read_I01, 0, 1023, 0, 5000);

fsr_Read_Distal = analogRead(fsr_DISTAL);
fsr_V_Distal = map(fsr_Read_Distal, 0, 1023, 0, 5000);

fsr_Read_MedM = analogRead(fsr_MedM);
fsr_V_MedM = map(fsr_Read_MedM, 0, 1023, 0, 5000);

fsr_Read_LatM = analogRead(fsr_LatM);
fsr_V_LatM = map(fsr_Read_LatM, 0, 1023, 0, 5000);

sensorString += String(fsr_V_I01);
sensorString += ", ";
sensorString += String(fsr_V_Distal);
sensorString += ", ";
sensorString += String(fsr_V_MedM);
sensorString += ", ";
sensorString += String(fsr_V_LatM);

Serial.print("Sensor values: ");
Serial.print(sensorString);

t = millis();
}
//receive from serial terminal
if (Serial.available() > 0)
{
float i;
char inByte = Serial.read();
if (inByte == 't')
{
LoadCell_1.tareNoDelay();
LoadCell_2.tareNoDelay();
}
}
}

void loopTearing()
{
if (buttonStateTear == HIGH) //Tearing HX711 when button is pressed
{
//receive from serial terminal
if (Serial.available() > 0)
{
float i;
char inByte = Serial.read();
if (inByte == 't')
{
LoadCell_1.tareNoDelay();
LoadCell_2.tareNoDelay();
}
}
}
}
```

oled_setup.h

```
//---------- TESTRIG SENSOR SKETCH 20200119 -----------------
//-----------------    oled_setup.h    -------------------------
//-   Written by Josefine C. S. Haugom ----------------------
//-   January 2020, NTNU Trondheim
```

```
//-------------------------------------------------------------

//------------  Screen: SSD1306 OLED 0.96" 128 x 64px ----------------------------------------
// OLED with 4 output pins: GND, Power (NB: 3.3V), SDA and SLC (NB: different pinnr between Uno and Mega )
//
//#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

// 'ntnu-logo', 128x64px
const unsigned char ntnu[] PROGMEM = {
//BITMAP REMOVED DUE TO PDF COMPILING ERROR
};

void OLEDsetup(void)
{
if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C))
{ // OLED  - Address 0x3D for 128x64 Check right adress if its not working
Serial.println(F("SSD1306␣allocation␣failed"));
for (;;)
;
}
delay(100);

display.clearDisplay();   // Starting up OLED - Welcome with NTNU logo
display.invertDisplay(1); // Inverted logo = white on black background
display.drawBitmap(0, 0, ntnu, 128, 64, WHITE);
display.display(); // Make abow display commands visible on OLED
delay(2000);

display.setCursor(0, 0);
display.setTextSize(1);
display.setTextColor(WHITE);
}

void dispSensors(void)
{
//OLED Loop
display.clearDisplay();
display.invertDisplay(0);
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(0, 8);

display.print("Rotation:␣␣");
if (a < 2000)
display.print(a);
display.println("␣N");

display.print("Axial␣␣␣:␣␣");
if (b < 2000)
display.print(b);
display.println("␣N␣\n");

display.drawLine(0, 28, 60, 28, WHITE);
display.drawLine(63, 28, 65, 28, WHITE);
display.drawLine(68, 28, 128, 28, WHITE);

display.print("FSR␣I01␣␣␣␣:␣␣");
display.print(fsr_V_I01 / 1000);
display.println("␣V");

display.print("FSR␣Distal␣:␣␣");
display.print(fsr_V_Distal / 1000);
display.println("␣V");

display.print("FSR␣Medial␣:␣␣");
display.print(fsr_V_MedM / 1000);
display.println("␣V");

display.print("FSR␣Lateral:␣␣");
display.print(fsr_V_LatM / 1000);
display.println("␣V");

```

```
display.display();
}
```

realtime.h

```
/******************************************************************************
DS1307_RTC_Demo.ino
Jim Lindblom @ SparkFun Electronics
original creation date: October 2, 2016
https://github.com/sparkfun/SparkFun_DS1307_RTC_Arduino_Library

Configures, sets, and reads from the DS1307 real-time clock (RTC).

Resources:
Wire.h - Arduino I2C Library

Development environment specifics:
Arduino 1.6.8
SparkFun RedBoard
SparkFun Real Time Clock Module (v14)
******************************************************************************/
#include <SparkFunDS1307RTC.h>
//#include <Wire.h>

// Comment out the line below if you want month printed before date.
// E.g. October 31, 2016: 10/31/16 vs. 31/10/16
#define PRINT_USA_DATE

#define SQW_INPUT_PIN 2   // Input pin to read SQW
#define SQW_OUTPUT_PIN 13 // LED to indicate SQW's state

void timeSetup()
{
// Use the serial monitor to view time/date output
pinMode(SQW_INPUT_PIN, INPUT_PULLUP);
pinMode(SQW_OUTPUT_PIN, OUTPUT);
digitalWrite(SQW_OUTPUT_PIN, digitalRead(SQW_INPUT_PIN));

rtc.begin(); // Call rtc.begin() to initialize the library
// (Optional) Sets the SQW output to a 1Hz square wave.
// (Pull-up resistor is required to use the SQW pin.)
rtc.writeSQW(SQW_SQUARE_1);

// Now set the time...
// You can use the autoTime() function to set the RTC's clock and
// date to the compiliers predefined time. (It'll be a few seconds
// behind, but close!)
rtc.autoTime();
// Or you can use the rtc.setTime(s, m, h, day, date, month, year)
// function to explicitly set the time:
// e.g. 7:32:16 | Monday October 31, 2016:
// rtc.setTime(16, 32, 7, 2, 31, 10, 16);  // Uncomment to manually set time
// rtc.set12Hour(); // Use rtc.set12Hour to set to 12-hour mode
}



void printTime()
{
Serial.print(String(rtc.hour()) + ":"); // Print hour
if (rtc.minute() < 10)
Serial.print('0'); // Print leading '0' for minute
Serial.print(String(rtc.minute()) + ":"); // Print minute
if (rtc.second() < 10)
Serial.print('0'); // Print leading '0' for second
Serial.print(String(rtc.second())); // Print second

if (rtc.is12Hour()) // If we're in 12-hour mode
{
// Use rtc.pm() to read the AM/PM state of the hour
if (rtc.pm()) Serial.print("␣PM"); // Returns true if PM
else Serial.print("␣AM");
}

Serial.print("␣|␣");

// Few options for printing the day, pick one:
Serial.print(rtc.dayStr()); // Print day string
//Serial.print(rtc.dayC()); // Print day character
//Serial.print(rtc.day()); // Print day integer (1-7, Sun-Sat)
Serial.print("␣-␣");
#ifdef PRINT_USA_DATE
Serial.print(String(rtc.month()) + "/" +   // Print month
```

```
String(rtc.date()) + "/");  // Print date
#else
Serial.print(String(rtc.date()) + "/" +    // (or) print date
String(rtc.month()) + "/"); // Print month
#endif
Serial.println(String(rtc.year()));        // Print year
}

void timeLoop(void)
{
static int8_t lastSecond = -1;

// Call rtc.update() to update all rtc.seconds(), rtc.minutes(),
// etc. return functions.
rtc.update();

if (rtc.second() != lastSecond) // If the second has changed
{
printTime(); // Print the new time

lastSecond = rtc.second(); // Update lastSecond value
}

// Read the state of the SQW pin and show it on the
// pin 13 LED. (It should blink at 1Hz.)
digitalWrite(SQW_OUTPUT_PIN, digitalRead(SQW_INPUT_PIN));
}
```

sd_setup.h

```
#include <SD.h>
const int chipSelect = 10;


void sdSetup(){
while (!Serial) {
; // wait for serial port to connect. Needed for native USB port only
}


Serial.print("Initializing SD card...");

// see if the card is present and can be initialized:
if (!SD.begin(chipSelect)) {
Serial.println("Card failed, or not present");
// don't do anything more:
while (1);
}
Serial.println("card initialized.");
}


void datalogging() {
// make a string for assembling the data to log:


// open the file. note that only one file can be open at a time,
// so you have to close this one before opening another.
File dataFile = SD.open("datalog.txt", FILE_WRITE);

// if the file is available, write to it:
if (dataFile) {
dataFile.println(sensorString);
dataFile.close();
// print to the serial port too:
Serial.println(sensorString);
}
// if the file isn't open, pop up an error:
else {
Serial.println("error opening datalog.txt");
}
}
```

user_control.h

```
//---------- TESTRIG SENSOR SKETCH 20200119 -----------------
//---------------   user_control.h    ------------------------
//-   Written by Josefine C. S. Haugom ---------------------
//-   January 2020, NTNU Trondheim
//-------------------------------------------------------------
//---- Library for buttons and leds in the test-rig

/******      BUTTONS     **********/
const int buttonOK = 8;   // OK-button input D8
const int buttonTEAR = 9; // TEAR-button input D9

int buttonStateOK = 0;   // variable for reading the pushbutton status
int buttonStateTear = 0; // variable for reading the pushbutton status

/*****      LEDS         **********/
const int ledPinRed = 12;   //Status LEDs - program start - led on
const int ledPinGreen = 13; //Status LEDs - loop start    - led on

void controlSetup(void)
{
pinMode(ledPinRed, OUTPUT);   //  Defining led pin as output
pinMode(ledPinGreen, OUTPUT); //  --
pinMode(buttonOK, INPUT);
pinMode(buttonTEAR, INPUT);
digitalWrite(ledPinRed, HIGH);
}

void ledSeq4(void)
{
digitalWrite(ledPinRed, LOW);
delay(100);
digitalWrite(ledPinRed, HIGH);
delay(100);
digitalWrite(ledPinRed, LOW);
delay(100);
digitalWrite(ledPinRed, HIGH);
}

void loopButtonread(){
buttonStateOK   = digitalRead(buttonOK);     // variable for reading pushbutton status (OK)
buttonStateTear = digitalRead(buttonTEAR);   // variable for reading pushbutton status (TEAR)
}
```

D
Additional Documents

D.1 Risk Assessment MTP

NTNU	Kartlegging av risikofylt aktivitet	Utarbeidet av	Nummer	Dato
HMS		HMS-avd.	HMSRV2601	09.09.2019
		Godkjent av		Erstatter
		Rektor		

Enhet: Department of Mechanical and Industrial Engineering **Dato: 2019.09.09**

Linjeleder: Øyvind Haave

Deltakere ved kartleggingen (m/ funksjon)**:** Knut Einar Aasland, veileder/Martin Steinert, co-veileder/ Josefine C. Stokke Haugom, student.
(Ansv. veileder, student, evt. medveiledere, evt. andre m. kompetanse)

Kort beskrivelse av hovedaktivitet/hovedprosess: Masteroppgave student Josefine C. S. Haugom. Forbedret innfesting av protese for alpinutøvere.

Er oppgaven rent teoretisk? (JA/NEI)**:** *«JA» betyr at veileder innestår for at oppgaven ikke inneholder noen aktiviteter som krever risikovurdering. Dersom «JA»: Beskriv kort aktiviteteten i kartleggingskjemaet under. Risikovurdering trenger ikke å fylles ut*

.

Signaturer: *Ansvarlig veileder: Knut Einar Aasland, Martin Steinert* *Student: Josefine Caroline Stokke Haugom*

ID nr.	Aktivitet/prosess	Ansvarlig	Eksisterende dokumentasjon	Eksisterende sikringstiltak	Lov, forskrift o.l.	Kommentar
1	Bruk av Trolllabs workshop.	JCSH	Romkort	Romkort		
1a	Bruk av roterende maskineri	JCSH	Maskinens brukermanual	Ukjent	Ukjent	
1b	Bruk av laserkutter	JCSH	Maskinens brukermanual	Ukjent	Ukjent	
1c	Bruk av 3D printer	JCSH	Maskinens brukermanual	Ukjent	Ukjent	
1d	Bruk av skjæreverktøy	JCSH	Ukjent			
1e	Bruk av samenføynigsmidler (lim og lignende.)	JCSH	Produktets brukermanual og datablad	Datablad	Ukjent	

D.2 Risk Assessment MTP

NTNU / HMS	Kartlegging av risikofylt aktivitet	Utarbeidet av	Nummer	Dato
		HMS-avd.	HMSRV2601	09.09.2019
		Godkjent av		Erstatter
		Rektor		

2	Tilstedeværelse ved arbeid utført av andre.	Andre	Andres HMSRV2601	Andres HMSRV2601	Prosessavhengig	
3	Eksperimentelt arbeid	JCSH	Egen risikovurdering- må gjøres for hvert enkelt eksperiment		Prosessavhengig	

D.3 Risk Assessment MTP

NTNU / HMS	Risikovurdering	Utarbeidet av	Nummer	Dato
		HMS-avd.	HMSRV2601	09.09.2019
		Godkjent av		Erstatter
		Rektor		

ID nr	Aktivitet fra kartleggings-skjemaet	Mulig uønsket hendelse/ belastning	Vurdering av sannsyn-lighet	Vurdering av konsekvens:				Risiko-Verdi (menn-eske)	Kommentarer/status Forslag til tiltak
			(1-5)	Menneske (A-E)	Ytre miljø (A-E)	Øk/ materiell (A-E)	Om-dømme (A-E)		
1	Bruk av Trolllabs workshop.								
1a-i	Bruk av roterende maskineri	Stor kuttskade	2	D	A	A	D	2D	Sørg for at roterende deler tilstrekkelig sikret/dekket. Vær nøye med opplæring i bruk av maskineri.
1a-ii		Liten kuttskade	3	B	A	A	A	3B	Vær nøye med opplæring i bruk av maskineri. Ikke ha løse klær/tilbehør på kroppen.
1a-iii		Klemskade	2	D	A	A	C	2D	Vær nøye med opplæring i bruk av maskineri. Ikke ha løse klær/tilbehør på kroppen.
1a-iv		Flygende spon/gjenstander	3	C	A	A	B	3C	Bruk øyevern og tildekk hurtig roterende deler (Fres og lignende.)
1a-v		Feil bruk-> ødelagt utstyr	3	A	A	C	A	3C	Vær nøye med opplæring i bruk av maskineri
1b-i	Bruk av laserkutter	Klemskade	2	D	A	A	C	2D	Vær nøye med opplæring i bruk av maskineri. Ikke ha løse klær/tilbehør på kroppen.
1b-ii		Brannskade	3	B	A	A	A	3B	Vær nøye med opplæring i bruk av maskineri. Bruk hansker ved håndtering av varme materialer.

D.4 Risk Assessment MTP

NTNU HMS	Risikovurdering	Utarbeidet av	Nummer	Dato
		HMS-avd.	HMSRV2601	09.09.2019
		Godkjent av		Erstatter
		Rektor		

1b-iii		Øyeskade-laser	2	D	A	A	C	2D	Bruk øyevern! Skru av laser når maskinen ved oppsett.
1b-iv		Brann	2	B	A	D	C	2B	Vær nøye med opplæring i bruk av maskin. Ha slukkeutstur tilgjengelig
1c-i	Bruk av 3D-printer	Brannskade	3	B	A	A	A	3B	Vær nøye med opplæring i bruk av maskin.
1c-ii		Innhalering av plast/ printemateriale	5	A	A	A	A	5A	Bruk åndedretsvern/ vernebriller
1c-iii		Feil bruk-> ødelagt maskineri	3	A	A	C	A	3A	Vær nøye med opplæring i bruk av maskin.
1d-i	Bruk av skjæreverktøy	Stor kuttskade	2	D	A	A	D	2D	Bruk skapre verktøy og riktig skjæreunderlag
1d-ii		Liten kuttskade	3	B	A	A	A	3B	Bruk skapre verktøy og riktig skjæreunderlag
1e-i	Bruk av samenføynigsmidler (lim og lignende.)	Eksponering på øyet	2	D	A	A	B	2D	Bruk øyevern, ha datablad tilgjengelig
1e-ii		Eksponering hud	4	A	A	A	A	4A	Bruk hansker, ha datablad tilgjengelig
1e-iii		Eksponering åndedrett	4	A	A	A	A	4A	Bruk åndedretsvært/ god ventilasjon. Ha datablad tilgjengelig.
1e-iv		Søl	4	A	B	A	A	4A	Ha papir/ rengjøringsmateriell tilgjengelig. Ha datablad

D.5 Risk Assessment MTP

NTNU	Risikovurdering	Utarbeidet av	Nummer	Dato
HMS		HMS-avd.	HMSRV2601	09.09.2019
		Godkjent av		Erstatter
		Rektor		

									tilgjengelig.
2	Tilstedeværelse ved arbeid utført av andre.	Se andres risikovurdering om sikkerhet betviles.	3	C	C	C	C	3C	Hold et øye med hva som foregår rundt deg.
3-i	Eksperimentelt arbeid	Test av rotasjonsglidning – klem i apparat	3	B	A	A	B	3B	Vær nøye med oppsetting av rigg. Ikke ha løse klær og gjenstander på kroppen.
3-ii									

Sannsynlighet vurderes etter følgende kriterier:

D.6 Risk Assessment MTP

NTNU / HMS	Risikovurdering	Utarbeidet av	Nummer	Dato
		HMS-avd.	HMSRV2601	09.09.2019
		Godkjent av		Erstatter
		Rektor		

Svært liten 1	Liten 2	Middels 3	Stor 4	Svært stor 5
1 gang pr 50 år eller sjeldnere	1 gang pr 10 år eller sjeldnere	1 gang pr år eller sjeldnere	1 gang pr måned eller sjeldnere	Skjer ukentlig

Konsekvens vurderes etter følgende kriterier:

Gradering	Menneske	Ytre miljø Vann, jord og luft	Øk/materiell	Omdømme
E **Svært Alvorlig**	Død	Svært langvarig og ikke reversibel skade	Drifts- eller aktivitetsstans >1 år.	Troverdighet og respekt betydelig og varig svekket
D **Alvorlig**	Alvorlig personskade. Mulig uførhet.	Langvarig skade. Lang restitusjonstid	Driftsstans > ½ år Aktivitetsstans i opp til 1 år	Troverdighet og respekt betydelig svekket
C **Moderat**	Alvorlig personskade.	Mindre skade og lang restitusjonstid	Drifts- eller aktivitetsstans < 1 mnd	Troverdighet og respekt svekket
B **Liten**	Skade som krever medisinsk behandling	Mindre skade og kort restitusjonstid	Drifts- eller aktivitetsstans < 1uke	Negativ påvirkning på troverdighet og respekt
A **Svært liten**	Skade som krever førstehjelp	Ubetydelig skade og kort restitusjonstid	Drifts- eller aktivitetsstans < 1dag	Liten påvirkning på troverdighet og respekt

Risikoverdi = Sannsynlighet x Konsekvens
Beregn risikoverdi for Menneske. Enheten vurderer selv om de i tillegg vil beregne risikoverdi for Ytre miljø, Økonomi/materiell og Omdømme. I så fall beregnes disse hver for seg.

Til kolonnen "Kommentarer/status, forslag til forebyggende og korrigerende tiltak":
Tiltak kan påvirke både sannsynlighet og konsekvens. Prioriter tiltak som kan forhindre at hendelsen inntreffer, dvs. sannsynlighetsreduserende tiltak foran skjerpet beredskap, dvs. konsekvensreduserende tiltak.

D.7 Risk Assessment MTP

NTNU	Risikomatrise	utarbeidet av	Nummer	Dato
		HMS-avd.	HMSRV2604	09.09.2019
HMS/KS		godkjent av		Erstatter
		Rektor		

MATRISE FOR RISIKOVURDERINGER ved NTNU

KONSEKVENS	Svært alvorlig	E1	E2	E3	E4	E5
	Alvorlig	D1	D2	D3	D4	D5
	Moderat	C1	C2	C3	C4	C5
	Liten	B1	B2	B3	B4	B5
	Svært liten	A1	A2	A3	A4	A5
		Svært liten	Liten	Middels	Stor	Svært stor
		SANNSYNLIGHET				

Prinsipp over akseptkriterium. Forklaring av fargene som er brukt i risikomatrisen.

Farge		Beskrivelse
Rød		Uakseptabel risiko. Tiltak skal gjennomføres for å redusere risikoen.
Gul		Vurderingsområde. Tiltak skal vurderes.
Grønn		Akseptabel risiko. Tiltak kan vurderes ut fra andre hensyn.