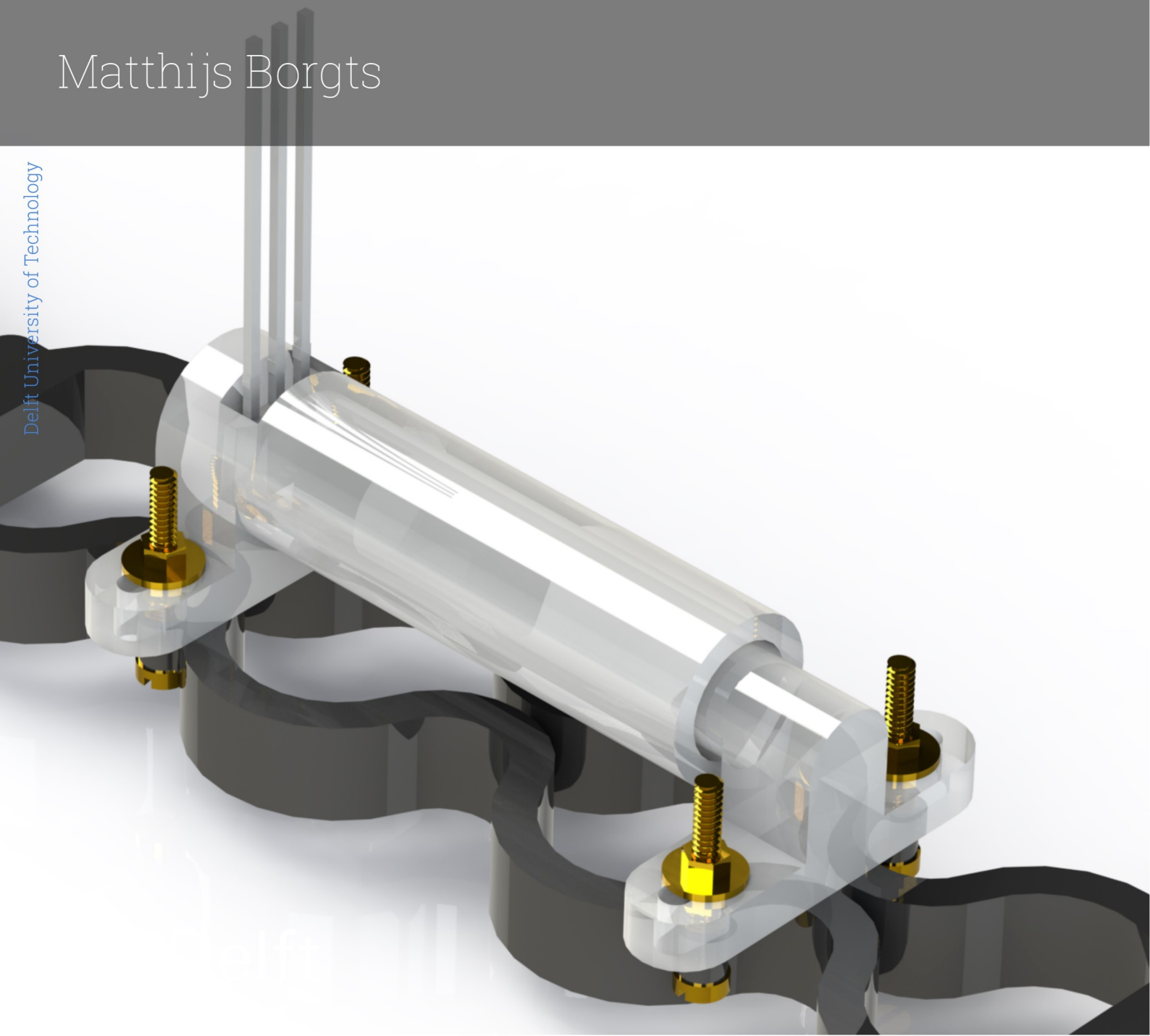
Design and Validation of a Hip Joint Simulator

Additively Manufactured Soft Tissues with Force and Strain Sensors for Orthopaedic Surgeon Training and Implant Analysis

Matthijs Borgts

Delft University of Technology



Delft University of Technology

BioMechanical Engineering
Master's thesis

Design and Validation of a Hip Joint Simulator

Additively Manufactured Soft Tissues with Force and Strain Sensors for Orthopaedic Surgeon Training and Implant Analysis

Author:
Matthijs Borgts
(4610520)

Supervisors:
Jonathan Wei
Tim Horeman

August 15, 2024

Abstract

Most early revisions in total hip arthroplasty (THA) are caused by dislocations. It has been established that surgeon experience is a major component in the need for revision surgeries. Currently, no standardised protocol for intraoperative soft tissue tension assessment exists. Furthermore, orthopaedic surgeons do not have a method of preoperative validation of trial implant configurations. To address this problem, the design of a physical simulator of the hip joint with muscle force and strain sensors is proposed, mostly made of widely available materials, manufactured using fusion deposition modeling (FDM) technology. The goal is to create a training tool for orthopaedic surgeons which will reduce the number of revision operations caused by poor hip balance. In addition, the simulator could be used to review the effects of varying implant geometries, which may reduce operation times. The simulator is hypothesised to have a similar joint balance to an *in vivo* hip, where an increased femoral offset will result in increased tension around the joint. In addition, the increased femoral offset is hypothesised to cause an increased external rotation moment and angle required for subluxation.

A scaling model based on anatomical data is constructed, and an additively manufactured shape is designed which mimics the tensile characteristic of muscle tissue. Sensors are added to almost each phantom muscle, to actively monitor the subjected tension and strain. The simulator is then tested by orthopaedic surgeons, performing two of their preferred movements for hip stability assessment, namely a traction and external rotation test, using three different implant configurations with varying femoral offsets. The muscle strains at the point of subluxation are reported. This experiment is then repeated, but not executed by a specialist, while using an additional implant configuration. During this second traction test, the total traction force at the foot was also measured. During the second external rotation test, the angle and moment required for subluxation were also measured at the foot.

The experiments yielded promising results. Muscle strains were recorded during the traction test for the gluteus maximus, minimus and piriformis which ranged from maxima of 10% to 26% at subluxation. For these muscles, the recorded force and strain were significantly higher ($P \leq 0.05$) for the highest femoral offset compared to the lowest offset. During the external rotation test, the gluteus medius and minimus were maximally strained from 12% to 23%, while showing significantly higher force and strain for the highest offset configuration compared to the lowest offset. An increased femoral offset did cause increased tension in the joint, suggesting that the first hypothesis may be valid. In addition, the surgeons were able to correctly answer if there had been an increase or decrease in femoral offset without knowing the current implant configuration. They were enthusiastic about the potential of the simulator as a training tool, both for surgeons becoming more acquainted with the stability assessment movements and as a general method of understanding the mechanics of the hip joint, specifically how an implant configuration can influence these mechanics. This experiment was followed by an experiment where the same movements were performed, but not by a specialist.

During the second traction test, the gluteus medius, gluteus minimus, pectineus and piriformis showed muscle strains ranging from 7% up to 31%. For these muscles, the force and strain were again significantly higher for the higher femoral offsets compared to the lowest offset. The total traction force ranged from 52N to 74N, where the force was significantly higher for the higher offsets compared to the lowest offset. During the external rotation test, maximum muscle strains ranged from 14% to 18% for the gluteus medius and minimus, where the higher femoral offsets resulted in higher forces and strains compared to the lowest offset, providing additional support for the first hypothesis. The moment required for subluxation ranged from 9N to 12N, where the higher offsets caused a significantly higher moment required for subluxation compared to the lowest offset. Additionally, the angle required for subluxation ranged from 35 to 43 degrees, where the higher offsets again caused significantly higher external rotation angles at subluxation. The results from these tests support the second hypothesis, as the increased femoral offset resulted in an increased external rotation moment and angle required for subluxation.

In conclusion, the hip simulator is promising as a prototype and demonstrates the ability to approximate the stability of an *in vivo* human hip, while being affordable and made of widely available materials. With further improvements, the hip simulator may be implemented in both orthopaedic surgeon and physiotherapist tutoring, potentially decreasing the amount of revision surgeries and time spent in the operation room, to improve patient health and save cost. Furthermore, the simulator may provide additional insight into the functions of specific hip muscles. A method was designed to model the muscles of subjects based on weight, height and limb length, which may be used to model the hip joints of specific patients for a more accurate estimation of their joint mechanics. Lastly, a method was devised to measure the force and strain of flexible materials, which has possible appliances outside the medical field.

Nomenclature

THA Total hip arthroplasty

ROM Range of motion

EVA Ethylene vinyl acetate

TPU Thermoplastic polyurethane

FDM Fusion deposition modelling

$PCSA$ Physiological cross-sectional area

MTU Muscle-tendon unit

TFL Tensor fasciae latae

FEA Finite element analysis

SLA Stereolithography apparatus

PLA Polylactic acid

$PETG$ Polyethylene terephthalate glycol

STD Standard neck

KHO High-offset neck

KLA Lateralised neck

FEM Finite elements method

$PMMA$ Polymethyl methacrylate

ECM Extracellular matrix

Acknowledgements

I would like to thank a few people for their efforts over the course of this project. Firstly, I would like to thank my daily supervisor Jonathan, who continued to give me advice and answer all of my questions during our weekly meetings. I am grateful for the fact that he continued to do so, even after he was no longer working at TU Delft. I would like to thank my other supervisor Tim, whose technical insights and suggestions challenged me to come up with creative solutions to the problems I faced. I would also like to thank the people at the IWM workshop at *TU Delft*, particularly Reinier and Damian, for their willingness to print my many ideas, prototypes and final designs. I would like to thank the doctors Bryan and Hub for their participation and valuable feedback. Furthermore, I would like to thank the people from the *Material Sciences* lab for allowing me to use the tensile testing machine. I would also like to thank the people working in the *MISIT* lab every day for their help and companionship. Lastly, I would like to thank my girlfriend, friends and family for their unconditional support throughout the project.

Contents

1 Introduction

1.1 Background

As the average life expectancy increases, joint implants are becoming increasingly prevalent. One such implant is the hip implant. The procedure of replacing a hip is called total hip arthroplasty (THA). This procedure is executed by replacing both articulating surfaces of the hip joint. The femoral head can be either fully replaced, or trimmed and covered using a cap (this is called hip resurfacing). In both cases, the acetabulum is also replaced. A typical hip implant for THA consists of a stem, femoral head, liner and acetabular cup, shown in Figure 1.

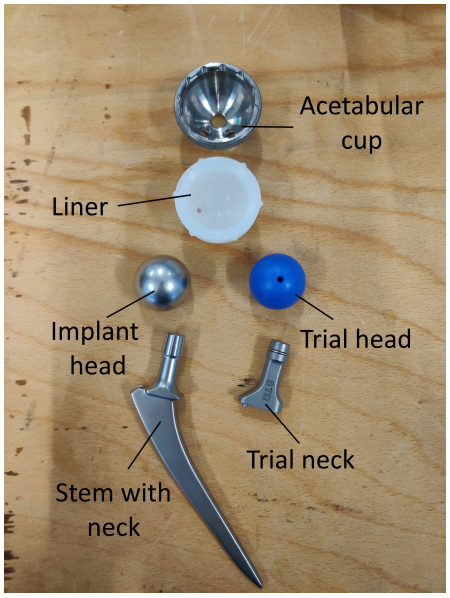


Figure 1: Implant components. Permanent components (stem and femoral head) are shown on the left, trial components are shown on the right. The acetabular cup and liner are shown at the top.

There are multiple ways in which this procedure can be carried out. The surgical procedure is defined by the location of the incision. This can be done using the posterior approach, the lateral approach, the anterolateral approach, and the direct anterior approach [1].

Before a THA surgery procedure, the implant configuration is predetermined using X-ray imaging. During the operation, the surgeon uses modular hip implant parts to assess which geometrical proportions achieve the best fit [2]. These modular implants are also shown in figure 1. The surgeon then performs a series of range-of-motion (ROM) tests to select the permanent configuration. The surgeon can estimate whether the hip is implemented correctly by using this technique.

It has been established that soft tissue balancing is an important component of THA [3]. However, a standardised approach does not exist. Orthopaedic surgeons often have their own preferred methods and protocols. The process of restoring soft tissue tension is not a purely subjective one. Careful preoperative planning precedes the replacement procedure. This procedure always starts with an x-ray of the joint. At this stage, incorrect positioning of the limb or pelvis will cause problems later in the process [4]. Following this, choices are made regarding the material and type of fixation. These choices are based on factors such as the patient's age and activity level [5].

Subsequently, the sizes of the implant configuration need to be estimated. This process is called templating. According to *Shaikh*, the templating phase can be divided into five stages [5]:

1. Anatomical landmarks identification.
2. Analysing quality of the radiograph.
3. Choosing mechanical references.
4. Implant selection and positioning.
5. The templating task.

Determining anatomical landmarks is an important aspect of this process. Using these landmarks, the surgeons can determine the biomechanical properties of the hip, such as the centre of location, the femoral offset, and the leg length [6].

After creating a reference frame of the patient's hip, the hip implant components deemed most suited to the task are selected. Then, a 'template' is created. The proposed implant configuration is laid over the radiograph of the hip, to assess the implant's shape, size and position in all planes. Using this process, surgeons can quantitatively restore the hip's biomechanical function to the best of their abilities.

Based on the aforementioned process, a test implant is assembled to be used during the operation. Then, the operation can commence. The surgeon can also choose to do an intraoperative leg length measurement. However, in the literature, there seems to be a consensus that assessing soft tissue tension is more important [7][8].

During the operation, after the test implant is installed, the surgeon can assess the soft tissue tension using a few different methods. One such method is the 'shuck test'. During this test, the surgeon applies a distal pulling force to the femur in line with the joint, essentially pulling the femoral head out of the acetabular cup [8][9]. This test serves as a method to determine the overall soft-tissue tension around the joint. Another possible test is the 'drop kick test'. The hip is held in extension while the knee is flexed. If the hip implant is too long, the bi-articular hip extensors will become too tight, possibly causing the knee to swing into flexion when the leg is released [8]. This test is a means to prevent this issue.

Other intraoperative methods involve forcing the hip into subluxation, which is defined as a partial dislocation, where the femoral head is on the edge of the

acutabulum. This can be achieved by externally rotating the hip with the leg extended, or by internally rotating the hip with both hip and knee at 90 degrees flexion. By observing the ROM at which the hip dislocates, the surgeon can assess the stability. Especially the internal rotation ROM is a reliable indicator of hip stability [3].

1.2 State of the art

To obtain a better understanding of the forces in the hip joint, implants have been used to measure the contact forces in the joint, namely the *Orthoload* implants [10][11][12]. This research was however confined to measuring the forces during the execution of day-to-day activities, such as walking, cycling, jogging, and standing up from a sitting position. *Wei et al.* constructed a device to measure the soft tissue tension and the range of motion in the hip joint [2]. This research attempted to objectively measure the forces in the hip joint while an orthopaedic surgeon conducted a series of ROM tests, providing a useful insight into the forces at play during these movements. Digital musculoskeletal models exist. One such model is the software package *OpenSim* [13]. This package provides a comprehensible overview of the musculature of the lower extremity. The use of this package in this field is limited, as it is not accurate in replicating passive muscle forces.

1.3 Problem statement

The problem with hip stability assessment is that it relies heavily on the experience of the surgeon. Furthermore, there is no standard testing protocol. ROM testing is a skill accomplished by the surgeon's 'feel'. As a result, inexperienced surgeons often have trouble balancing the hip. In the Netherlands, most early revision operations are caused by dislocation [14]. It was found that surgeon experience is a major factor in the need for revision surgeries [15]. Up to twice as many dislocations are caused by inexperienced surgeons compared to experienced surgeons. This figure decreases by approximately 50% for every ten operations, until a plateau is reached after 30 operations [15][2].

There is currently no training protocol for surgeons to become more proficient at hip balancing movements. Experience is gained by simply operating on patients, which results in many early revision operations. There is no equipment available on the market that would train surgeons for this procedure.

In addition, surgeons do not have a method of predicting how a certain implant configuration would affect hip balance. Surgeons can make estimations based on experience and literature, but tinkering with hip implant configurations and evaluating the effects intraoperatively remains the standardised method for these procedures, which is a costly method as it increases time spent in the operation room.

1.4 Research goal and hypotheses

The design of a physical hip simulator is proposed. The proposed idea consists of a physical model of the hip joint's bone structure and musculature. The phantom (replicated) muscles are equipped with sensors which will estimate the force and strain of each muscle of the hip joint. An artificial lower leg will also be included.

It is hypothesised that a physical simulator of the hip joint can prove to be a valuable training tool for inexperienced orthopaedic surgeons. The proposed idea of a simulator, equipped with force and strain sensors on each muscle, would allow the surgeons to estimate the biomechanics of the hip joint while performing specific motions with the leg. This would not only benefit the surgeon's understanding of the hip joint, but also potentially reduce the amount of revision operations needed due to poor soft tissue tension, as well as saving cost by decreasing operation times. Furthermore, the proposed hip simulator could be used as a physical tool to familiarise surgeons with the movements that an orthopaedic surgeon performs to evaluate hip balance. In all, *the goal of this study is to design, construct and validate a physical simulator of the hip joint, which accurately replicates the passive muscle forces, while actively measuring the phantom muscles' tension and strain.*

The hip simulator is hypothesised to have mechanical properties similar to an *in vivo* hip. This means that specific implant configurations should affect properties such as the force required to achieve subluxation (partial dislocation of the hip joint), and hip range of motion. A larger femoral offset has been shown to increase hip soft tissue tension [16][17][18]. Furthermore, an increased femoral offset implant has been shown to increase external rotation range of motion [19]. Therefore, the following hypotheses are formulated regarding the hip simulator.

An increased femoral offset will result in:

1. Increased muscle tension around the joint.
2. An increased external rotation moment and angle required for subluxation.

1.5 Structure

Firstly, a material is selected and characterised, described in sections 2.1 and 2.2. Then, a model is constructed to scale the dimensions of each muscle (section A.2). A shape is designed to mimic the tensile characteristics of muscle tissue, described in section B.2. In the rest of section 2, the design of a physical hip simulator with force and strain sensors is described, along with its testing methods. The results are shown in section 3 and discussed in section 4. Conclusions are drawn in section 5.

Additional resources can be found in the appendix. These include background theory, the used methods

described in more detail, as well as the results and discussion of tests regarding multiple aspects of the simulator. Furthermore, the appendix contains the code and data used throughout the project.

2 Methods & materials

2.1 Selection

Muscles are known to exhibit hyperelastic properties. To mimic these properties, a shape is designed that exhibits a similar stiffness characteristic to passively lengthened muscle (see appendix A.1). The following requirements are used:

- The material needs to be capable of being made into customised shapes using widely available technology
- The material needs to be capable of high deformation under load
- The material needs to be affordable

It is worth noting that since the material needs to be capable of high deformations, the material is likely to be hyperelastic (see appendix A.3), similar to muscle. The software package *GRANTA Edupack* was used to conduct an initial material search [20]. A graphical representation of this search is shown in supplementary figure 4.

Since human muscle is a hyperelastic material with a non-linear stress-strain characteristic, Young's modulus is not a suitable metric. However, comparison with *EduPack*'s data to other biological materials such as human skin, which is also soft tissue like muscle, and ligament, which is stiffer, yields an indication of the desired stiffness. Furthermore, from *EduPack*'s database it quickly becomes apparent that flexible polymer foams and elastomeric materials are the only material families which approximate the characteristics of soft tissue.

While flexible polymer foams also do exhibit hyperelasticity, these materials are specifically meant for compression, not tension [21]. Thus, this material family is excluded from selection. Consequently, the focus lies on elastomers, namely silicone, polyurethane, rubbers and ethylene vinyl acetate (EVA). Both rubber and EVA have complex and expensive fabrication processes, therefore these materials are not deemed to be feasible. On the other hand, silicone is affordable and can be made into custom shapes by using 3D-printed moulds. Thermoplastic polyurethane (TPU) is also considered a feasible option as this material can be both moulded or additively manufactured by use of fusion deposition modeling (FDM), both with relatively low cost.

For simplicity of fabrication, a 3D-printing technique is preferred over moulding. There is a large variety of TPU-filaments on the market. One of the most flexible available variants is *NinjaFlex* by *NinjaTek* [22]. In addition, it has been characterised using multiple models, notably the Ogden hyperelastic model [23][24][25][26][27]. Due to its reputability and widespread available information, *NinjaFlex* is regarded as a suitable material to use for recreating the mechanical properties of muscle tissue. Just like many other variants though, this type of TPU does have a reputation to be difficult to print. Therefore, a few test prints were carried out at *TU Delft*.

Unfortunately, the printers available for use at *TU Delft* were Ultimaker 3 printers. These printers are fed the material using Bowden tubes. It appeared that flexible filaments such as *NinjaFlex* are troublesome to use with Bowden extruders. There is friction between the tube and the filament, which causes the filament to get stuck in the tube. In addition, the high flexibility of the filament also causes problems in the feeder. For these reasons, *NinjaFlex* does not appear to be a suitable material to use within the project's constraints.

At the available workshop at *TU Delft*, one of the materials in stock was *Ultimaker TPU 95A* [28]. This material is significantly stiffer than NinjaFlex, which makes it less ideal, however, it was already known that this filament was printable using the available machinery at *TU Delft*.

Due to the significantly better printability of this material over *NinjaFlex*, *Ultimaker TPU 95A* is selected as the most suited material [28]. Although silicone is significantly more compliant, it is hypothesised that this difference can be compensated for by changing parameters such as muscle phantom size and shape [29]. As a result, the muscles printed from TPU will be smaller than human muscles, but this is not considered to be a problem.

2.2 Characterisation

Due to the wide array of TPU filaments available on the market, not many studies containing a characterisation of Ultimaker TPU 95A exist. One such study is *Kwon et al.* [29]. The samples in this study were tested at a strain rate of 2 s^{-1}. Unfortunately, this is deemed to be too high for a realistic simulation of the hip muscles. Since elastomers are known to be heavily strain-rate dependent, the choice is made to carry out a new characterisation of the material. In addition, performing a new characterisation of the material will eliminate possible discrepancies in the printing parameters used at the *TU Delft* workshop compared to those used by *Kwon et al.*

Unfortunately, the only grips that are available have a maximum width of 10 mm. This means that the ASTM standard method (D412) could not be used. To accommodate this, a custom dogbone shape is designed with a maximum width of 8 mm. The shape is based on type C of ASTM D412 [30]. The width is reduced, while the length between the shoulders, as

well as the total length, is maintained. This shape is shown in supplementary figure 2.

The samples were tested at 15 mms^{-1}, which corresponds with a strain rate of 0.45 s^{-1}. This strain rate is chosen because it is the fastest that the machine can go without problems. Furthermore, a high speed is preferable to capture the viscoelastic properties of the material, since time-dependent phenomena such as these are also present in soft tissue. The samples are tested until failure. A photograph of a sample being stretched by the Lloyd tensile testing machine is shown in supplementary figure 16.

The data yielded from the tensile test can then be fitted with a hyperelastic model. The fitting is executed by using the curve fitting tool in *ANSYS Workbench* [31]. Although the Ogden model is shown to be more reliable at greater strains (see appendix A.3), the Mooney-Rivlin model is sufficiently accurate at the strain range of interest (0-30%) while also being more computationally simple. Furthermore, the Mooney-Rivlin provides a more accurate fit in this range. Therefore, the Mooney-Rivlin model is used to model the mechanical behaviour of TPU 95A. The fit is performed using a normalised error, and the material is assumed to be incompressible.

2.3 Phantom muscle scaling model

To mimic the muscles' passive mechanical properties, a material has been selected. Now, a method is devised to determine the dimensions of each muscle, as well as the maximum force and strain. In appendix A.1, the literature around passive muscle stiffness is explained in more detail. To summarise, the literature states that there are no specific scaling rules regarding muscle passive stiffness [32]. Despite this, it was found that one of the hip muscles, the gracilis muscle, shows only a modest increase in modulus when going from single fibre to whole muscle [33]. Based on this research, and the fact that active muscle force scales linearly with physiological cross-sectional area (PCSA) [34], the assumption is made that passive muscle stiffness is linearly related to PCSA as well. Since this is relatively uncharted territory in the medical field, the causes of passive stiffness scaling properties are considered to lie outside the scope of this research.

Thus, for each muscle, the passive force response is estimated using the corresponding PCSA. Fortunately, PCSA of all of the hip muscles has been researched in the past, and they are provided by literature [35][36]. *Persad et al.* conducted a unique experiment, where they measured the mechanical properties of the *in vivo* gracilis muscle of 11 individuals [37]. This was possible because the patients were undergoing surgery for free-functioning myocutaneous tissue transfer to restore elbow flexion after a brachial plexus injury. The corresponding graph is shown in figure 2.

Muscle-tendon units (MTUs) are used as a reference for the maximum strain. *Persad et al.* reported an increase in MTU length of 13.4 ± 1.2 cm, which corresponds to 33.6 % MTU strain [33]. Because not every MTU has the same maximum strain, every artificial MTU in the hip simulator is assumed to strain to 30% of its minimal length. Consequently, the whole gracilis passive muscle stress of 28.83 kPa at full stretch, reported by *Persad et al.*, is multiplied with the PCSA of each hip muscle to obtain the maximum force for each corresponding artificial MTU.

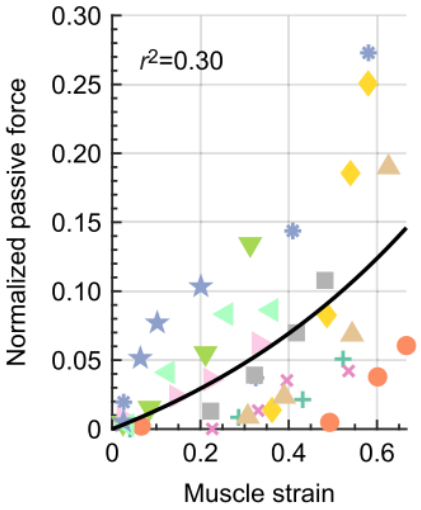


Figure 2: Normalised passive force versus muscle strain of whole gracilis muscle. The force was normalised by the maximum tetanic tension, which is defined as the product of the muscle's PCSA and a specific tension of 22.5 Ncm^{-2}. Taken from *Persad et al.* [37].

Anatomical parameters such as body mass, body height and limb length can be used to calculate muscle architectural properties such as total lower extremity muscle volume, as well as muscle belly length (see appendix A.2) [38]. These properties can then be used to calculate the individual muscle's physiological cross-sectional area, which is assumed to be proportional to passive muscle force. As a consequence, a mathematical model can be constructed which uses body mass, height and limb length as input and yields the corresponding muscle PCSA and length for every lower extremity muscle as output. Furthermore, *Handsfield et al.* provided coefficients for the best linear fit of each muscle's volume relative to the mass-height product. The average height, weight and limb length from *Handsfield et al.* are used as input parameters for this scaling model. Combining equations 6 and 7 and multiplying with the gracilis stress $\sigma_{gracilis}$ at maximum *in vivo* strain, the expression for the individual passive muscle force is shown in equation 1.

$$F_m = \frac{\sigma_{gracilis} f_{v,m}}{l_m f_f}(47mh + 1285) \qquad (1)$$

where F_m is the individual muscle force, $f_{v,m}$ is the individual muscle volume fraction, l_m is the individual muscle length and f_f is the muscle fibre to belly length ratio. Similarly, muscle belly length can be estimated from the subject's limb length l_l, shown in equation 2.

$$l_m = l_l f_{l,m} \tag{2}$$

where $f_{l,m}$ is the individual muscle's length fraction.

For this mathematical model, some assumptions have been made due to some data not being available. The muscle fibre to belly length ratios have been taken from cadavers with a mean age of 83 [35]. Similar to *Handsfield et al.* the assumption is made that these ratios are conserved for each muscle. However, some ratios are not reported by *Ward et al.*. This entails the pectineus, obturator muscles, piriformis, quadratus femoris and the gemelli muscles. Consequently, the missing ratios were taken from *Parvaresh et al.* [36]. In addition, *Parvaresh et al.* lacked the fiber-to-belly length ratio of the tensor fasciae latae (TFL). To solve this issue, the ratio was taken from *Friederich et al.* [39]. This ratio was however not normalised to the optimal sarcomere length.

2.4 Shape design and validation

To design a shape which exhibits a similar tensile characteristic to the one explained in section 2.3, the following design requirements regarding the phantom muscle are formulated:

- The stiffness needs to increase significantly at ∼25% stretch.
- The shape should be a symmetrical pattern across the longitudinal and transversal axis so that it can be repeated to adjust to the muscle belly length.
- The arcs need to be tangent with respect to each other, so that there are no sharp angles which increase initial stiffness.

A shape is designed by changing the dimensions in *SOLIDWORKS* and then simulating a tensile test in *ANSYS*. This trial-and-error approach culminated in an hourglass shape, which can be seen in figure 7a. The design process of the shape is described in more detail in appendix B.2.

After designing a shape with a satisfactory stress-strain characteristic, a tensile test is performed on the hourglass shape. Since there is often a large discrepancy between results simulated using finite elements analysis (FEA) and experimental data, the idea is to validate the hypotheses of the hourglass shape's mechanical behaviour. In turn, the results from this test can be scaled to approximate the force response of each muscle by increasing the thickness.

Hyperelastic material models do not account for strain-rate dependence. Therefore, the same strain rate must be used in this validation experiment as the one from the material characterisation described in section 2.2. Thus, a strain rate of 0.45 s^{-1} is used. The results from this validation test are then used to update the anatomical scaling model of the muscles, to provide a more accurate approximation of each muscle's output force for the phantoms. This is done by calculating the output force of the samples at 30% strain, after which each muscle's thickness is scaled based on the required force. The sample used for this test is shown in supplementary figure 12a.

Figure 3: SOLIDWORKS render of the hourglass shaped design of the phantom muscles.

2.5 Measurement of muscle strain and force

As was mentioned in section 1.4, the hip simulator is deemed to be more effective as a tool for both training orthopaedic surgeons and hip prosthetic evaluation when equipped with sensors to measure the muscle displacement and the muscle force. Hall-effect sensors are widely known to be used as an accurate proximity measurement method [40]. Furthermore, because of their simplicity of use combined with their small size, they would be suited to measure the displacement of each muscle on the simulator. Using tensile testing data of every muscle, the displacement can also be used to calculate the corresponding force of the muscle, so that additional force-sensors are not required.

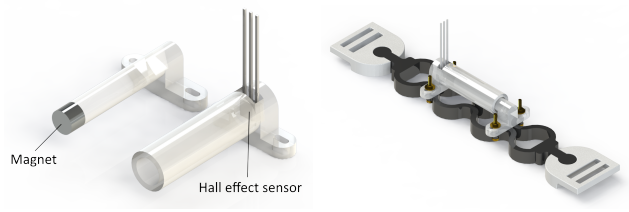
To measure the displacement of each muscle, the A1326 LUA-T model by *Allegro Microsystems* is used. The A1326 is a linear ratiometric Hall effect sensor. This means that the output voltage of the sensor is linearly proportional to the applied magnetic field's flux density. In a quiescent state, the output voltage is 50% of the supply voltage (The supply voltage is 5V). In the presence of a south-polarity magnetic field, the output voltage increases. In the case of a north-polarity field, the output voltage decreases.

This voltage can then be measured using a Redboard by *Sparkfun Electronics*. Since the Redboard only features six analogue I/O pins, CD74HC4051-E analogue multiplexers by *Texas Instruments* are used to connect all of the 25 Hall effect sensors to the Redboard. Analog multiplexers have three digital inputs, which can be used to specify which analogue input is

transmitted to the Redboard. Each of these three inputs represents a bit of the desired analogue input, ranging from 0 to 8. Both the multiplexers and the Hall effect sensors are equipped with decoupling capacitors, to prevent the voltage output from being affected by potential voltage drops from the Redboard's 5V pin.

Each single Hall effect sensor in the circuit shown in supplementary figure 8 is then calibrated. This is achieved by tensile testing every individual phantom muscle, while a sensor is mounted to the muscle. The output voltage V_{EE} is then measured by the *Arduino*. The relation between distance s and output voltage V_{EE} is approximated by using a power law. The same is done for the distance s and the force F_{muscle}, respectively.

To measure the deflection on each muscle, one needs to design a method of attaching the Hall effect sensors to the phantom muscle. Modules are designed to mount the sensors. Attachment holes are added to the muscles along their length. This way, the modules can be attached at any point of choice, since there is a limited amount of space available for this in the simulator. The modules are shown in figure 4. They feature slots, so that the bolts can move laterally and have minimal influence on the stretch of the muscles. The circular profile allows the sensors to still effectively measure the displacement when there is torsion present in the phantom muscle. Each module features an N45 neodymium magnet and a Hall sensor. They are 3D-printed out of *Form Clear* resin, using a *Form 3* stereolithography (SLA) printer from *Formlabs*.



(a) (b)

Figure 4: (a) Sensor mounts with magnet and Hall effect sensor. (b) Sensor mounts attached to TPU phantom muscle with brass M1 bolts.

The *Arduino* continuously reads out all of the 25 Hall effect sensors in ascending order, using a looped script. At the end of each loop, the *Arduino* outputs a line of code, containing the time passed since the script started running, and the reading for each sensor, separated by commas. To get a live reading of the sensors and to store the data, a *MATLAB* script is written. This script asynchronously reads the serial device (*Arduino*) using a callback function. At each call, the comma-separated line is saved to a .txt file and viewed live in a table.

Using the *MATLAB* command 'uitable', the voltage of each sensor is shown in a live table, as well as the force and strain of each muscle, which is calculated in the same callback function as mentioned earlier. The 'uitable' functionality allows the user to also highlight specific muscles, and to read the data more easily. The *MATLAB* and the *Arduino* code can be found in appendices D and E respectively.

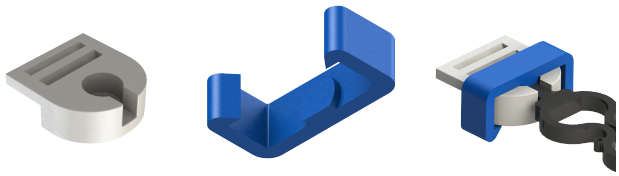
2.6 Design of hip simulator

As previously mentioned, work has been done before at *TU Delft* to find a solution for the issues stated in section 1.3. The second bachelor student group to tackle this issue constructed a physical simulator, which is modified [41]. To approximate an actual hip joint as optimally as possible, all 25 muscles acting on both the hip and knee joint are added. Consequently, some extra holes and threaded inserts are added to the simulator. The function of the knee is simulated by using a steel pipe with a hinge joint. To simulate the patella, a 3D-printed phantom from polylactic acid (PLA) is added. This will provide a more optimal moment arm and slightly increase the strain of the quadriceps muscles, as is the function of the patella in the human body.

Although a method of estimating the length of each muscle has been presented in section 2.3, a more practical approach is chosen for optimal results. The muscle lengths measured by *Handsfield et al.* will be different compared to those in the simulator. This is because every muscle in the body follows a specific path which is dependent on many factors, such as the amount of surrounding tissue, the muscle's thickness and its shape. Since this research only focuses on the muscles themselves and their attachment points to the bone, these conditions are not replicated. As a result, the MTU lengths are measured by attaching ropes to the MTU connection points. This is shown in supplementary figure 17. These ropes are then tightened while the hip joint is maintained in a neutral position. Using this method, the muscles can be pre-tensioned to simulate the hip balance present in real-life conditions. Then, the rope lengths are measured in the simulator.

A form-closed fit is used to attach the muscles to the 3D-printed pelvis and leg. This is preferred over a clamped fit, since a clamped connection would influence the mechanical properties of the phantom muscle. Furthermore, a form-closed fixture can be directly 3D-printed from PLA and integrated into the simulator, whereas designing clamps would be a more complicated solution. Different types of fixtures are designed based on muscle thickness, as bigger muscles require a larger head-to-neck ratio of the form-closed connection, as well as a larger connector to account for the increased force. Such a connector is shown in figure 5a. A phantom muscle with PLA connectors is shown in figure 5. For the smaller muscles (under 10 mm thickness), a 3D-printed polyethylene terephthalate glycol (PETG) clip is designed to hold the muscle in the connector (see

figure 5b). Since this part needs to bend around the connector to 'snap' into the slot which also holds the muscle, the choice is made to use PETG for this part. This is because PLA is too brittle for this use. At both ends of each phantom muscle, the PLA connectors are fixed to the simulator using nylon straps. These straps are secured to the simulator by perforating them with bolts and screwing the bolts into the threaded inserts in the PLA or the threaded holes in the metal leg.



(a) (b) (c)

Figure 5: *SOLIDWORKS* renders of 3D-printed parts used to secure phantom muscles to the simulator. a) PLA fixture to secure the muscle to strap. b) Compliant PETG clip to secure the muscle to the fixture. c) PLA fixture, PETG clip and phantom muscle.

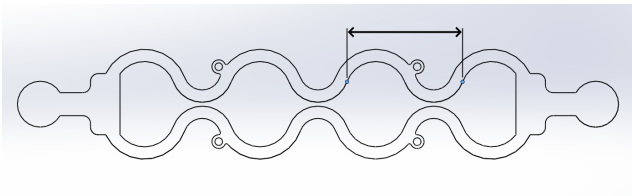


Figure 6: A length increment of a phantom muscle, which is used to extend the muscle belly length.

After the connections have been designed, the muscle belly lengths of each phantom muscle can be calculated. This is done by taking the MTU length l_{MTU} and subtracting the total connector length l, c, tot. Then, the muscle belly length l_{mb} is increased incrementally (shown in figure 6) until a desired strap length l_{strap} remains. The formula is shown in equation 3.

$$l_{strap} = l_{MTU} - l_{c,tot} - l_{mb} \tag{3}$$

Ideally, l_{strap} is close to 10% of l_{MTU} for adjustability. This way, there is some space for extra pretensioning of the phantom muscle if necessary. However, this number may vary to accommodate a low amount of available space in the simulator. Some phantom muscles are too long for the printing bed of the *Ultimaker S5* printer. In those cases, the phantom muscle is split into two equal parts which are connected in the middle of the MTU.

2.7 Material, shape and sensor testing

Additional testing is performed to analyse a few properties of the TPU material, the phantom muscle shape and the measuring method of the shape and size. These include:

- Shelf life of the material
- Material hysteresis
- Strain rate dependence
- Stress relaxation
- Creep
- Overall sensitivity of the measuring method using Hall effect sensors
- Influence of attachment type (T-fit or clamp)
- Influence of sensor mount location
- Influence on sensor output when a muscle is curved around a circular surface
- Agreement between strain and force measurement method and tensile tester

The methods employed to perform these tests are described in detail in appendices B.3 and B.4.

2.8 Sensor calibration

As was mentioned in section 2.5, the Hall effect sensors used are ratiometric. Thus, the sensors need to be calibrated to measure the strain and force of each muscle. This is done by elongating the muscles in a tensile tester while measuring the force. A Hall effect sensor is attached to the muscle in the middle. Then, the muscle is elongated at a strain rate of 0.1 s^{-1}, up to a strain of 50%. Both the Arduino and the tensile tester record the time, so the data can be synchronised by finding the moment at which the voltage starts to increase.

After the tests are performed, the output voltage V_{out} of every Hall sensor is directly fitted, using a power function, to both the displacement and the force, shown in equations 4 and 5, respectively. The muscle end displacement d can be described as:

$$d = aV_{out}^{n} + b \tag{4}$$

where a, b and n are fit parameters specific to the phantom muscle. The muscle phantom output force is given by:

$$F = cV_{out}^{m} + f \tag{5}$$

where c, f and m are fit parameters specific to the phantom muscle.

2.9 Stability testing with orthopaedic surgeons

As was mentioned in section 1.4, it is hypothesised that the hip simulator provides both a training tool to improve the performance of surgeons during hip balancing procedures, as well as a method of preoperatively assessing the influence of specific hip implant configurations. To validate this, a test needs to be performed with experienced medical personnel. The hip implant

trial necks can be seen in supplementary figure 19a, and the trial heads can be seen in supplementary figures 19b and 19c.

For these tests, the following equipment is needed:

- Hip simulator equipped with Hall effect sensors
- Arduino module with USB cable.
- Laptop with MATLAB readout script
- Femoral heads (⌀ 28 mm) with +1 and +9 offset
- STD and KHO femoral necks
- Clamps for fixing simulator to table

First, the resemblance of the simulator to an *in vivo* hip is evaluated. This is a blind test, the surgeon does not know which components are used and the goal is to assess if the subject's experience is comparable to performing ROM tests with an *in vivo* hip, and if the sensors' output data is consistent with the subject's experience. Since seeing the sensor output may cause bias, the subject does not have access to the sensors during this test. The subject is allowed to use the sensors at the end of the test for feedback purposes.

The surgeon's preferred methods of assessing the soft tissue tension are used. These include the traction and the external rotation test (shown in figures 7a and 7b. During the traction test, the surgeon holds the ankle while pulling the leg in the coronal plane, until subluxation occurs. According to the surgeons, subluxation is defined as the moment when the femoral head stands on top of the edge of the acetabular cup. The surgeons can feel when this occurs. Similarly, an external rotation test is performed. During this test, the surgeon externally rotates the leg while the leg is extended at the knee and the hip, until a subluxation occurs. For this test, a standard (STD) and high-offset (KHO) neck is used. The tested configurations are *STD + 9, STD + 1* and *KHO + 9*, The number after the + indicates the depth of the trial head, where a higher number indicates a higher offset, thus a lower depth (see figure 1). The test is conducted as follows:

1. Specific component configuration is installed in the simulator.
2. Surgeon assesses tissue tension by performing a traction and an external rotation test (both tests are repeated three times during this step).
3. Configuration is changed without the surgeon's knowledge.
4. Surgeon reassesses soft tissue tension.
5. Process is repeated two times.
6. Step 2 is repeated while the surgeon observes the sensor data.
7. Surgeon fills out a questionnaire.

After the test, the subject is asked to fill out a questionnaire, in which the surgeon is asked about his suspected changes to the implant configuration, among other things such as the overall likeness of the biomechanics to an *in vivo* hip, the interface of the sensors readings and the potential of the hip simulator as a training tool.

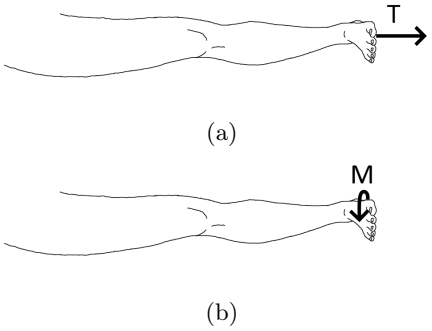


Figure 7: (a) Schematic drawing of the traction test. A traction force T is applied to the leg in the coronal plane, with the knee extended. (b) Schematic drawing of the external rotation test. An external rotation moment M is applied at the foot in the saggital plane, with the knee extended.

2.10 Additional stability testing

The test mentioned in section 2.9 provides sensor data as well as qualitative feedback from the surgeons. To accompany this data, the movements performed by the surgeons are reproduced to provide a more standardised test protocol, as the leg is now kept in the same position for each test and the movements are now produced by the same subject. Furthermore, the traction force and external rotation moment (by exerting a moment perpendicular to the foot) and angle are measured this time, to evaluate the second hypothesis, stated in section 1.4. In addition, the results from this test can be used to see if the sensor data is in accordance with the hypothesised increased range of motion and moment. The heads and necks used for this test can be seen in supplementary figure 19. The following equipment is used for this test:

- Hip simulator equipped with Hall effect sensors
- Arduino module with USB cable.
- Laptop with MATLAB readout script
- Femoral heads (⌀ 28 mm) with +1 and +9 offset
- *STD, KHO* and *KLA* femoral necks
- Newton meter
- Goniometer
- Strap to hold ankle in place
- Clamps for fixing simulator to table

This time, the ankle is held in place for the duration of the test, which can be seen in supplementary figure 13. An additional neck is used, namely a lateralised neck (KLA). Four different implant configurations are used: *STD + 9, STD + 1, KHO + 9* and *KLA + 9*. The sensor output is monitored and saved in a file while testing. The testing protocol is as follows:

1. A specific configuration is installed in the simulator.
2. Ten traction tests are performed while measuring the traction force with a Newton meter.
3. Ten external rotation tests are performed.
4. Steps 1-3 are repeated three times.

After the test, the data is processed and the external rotation moment is calculated based on the dimensions of the lower leg.

3 Results

In this section, the results are presented by the use of figures created using *MATLAB*. The results were tested for statistical significance using the *ANOVA* test, followed by Tukey's multiple comparisons test where possible. The student's t-test was used if there were only two datasets to compare.

3.1 Characterisation

The results from the material characterisation of *Ultimaker TPU 95A* are shown in figure 8. The experimental data along with the curve fit using the Mooney-Rivlin hyperelastic model can be seen.

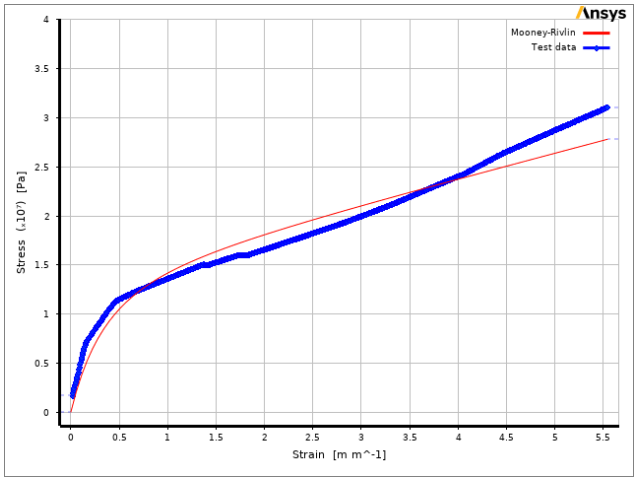


Figure 8: Curve fit of tensile test data using the 2-parameter Mooney-Rivlin hyperelastic model [31].

The material constants $C_{10} = 1.289$ MPa and $C_{01} = 5.519$ MPa are obtained.

3.2 Simulation and experimental testing of hourglass shape

Before every phantom muscle dimension is calculated so that it can be 3D-printed, their tensile properties are analysed through FEA, in the form of a simulated tensile test. Then, the tensile test is conducted using samples where their ends are clamped. The resulting graph is shown in figure 9. As can be seen in the figure, the sample was significantly stiffer than was predicted by FEA ($P \leq 0.05$).

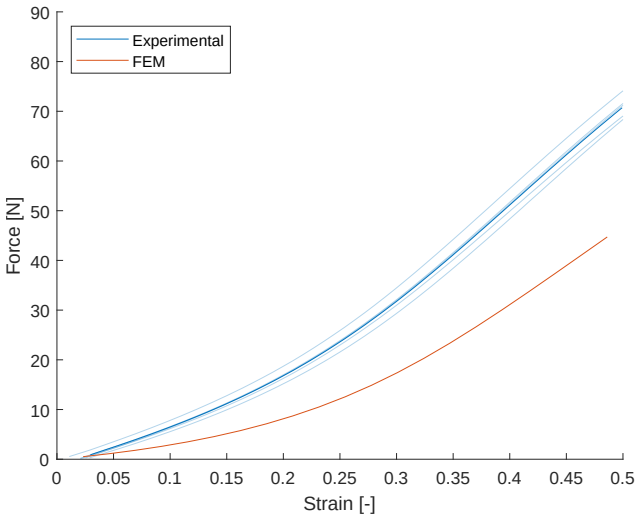


Figure 9: Tensile characteristic of hourglass-shaped sample.

3.3 Phantom muscle dimensions

As was explained in section 2.3, the phantom muscle dimensions are calculated based on the *in vivo* stiffness determined by *Persad et al.* [37]. However, it was suspected that there would be a noticeable difference in stiffness between the experimental results and those resulting from the *FEA*. Therefore, the phantom muscle thickness is scaled based on the experimental results. The result is shown in a spreadsheet containing the length and thickness of each muscle. The output force is calculated using the anatomical scaling model described in section 2.3, while the length was measured using the existing hip simulator, described in section 2.6.

3.4 Material, shape and sensor testing

In section 2.7, multiple tested material and shape properties are described, which could potentially have an impact on the performance of the simulator. The resulting plots are shown in supplementary figure 14. After five months of shelf life, the sample showed a significantly different characteristic, as can be seen in supplementary figure 14a. In supplementary figure 12b, one can observe a significantly stiffer characteristic with a clamped sample than is the case with one attached using a T-fit. The strain-rate dependence test also shows significantly different results for the comparison of each condition, as can be seen in supplementary figure 14d. The stress-relaxation present in supplementary figure 14d can also be observed in supplementary figure 14e. Lastly, material and shape are prone to creep, as can be seen in supplementary figure 14f.

The results of the sensor accuracy test, the sensor location test and the curved surface test, described in

detail in appendices B.3 and B.4, are shown in supplementary figure 15. The range in which the sensor can detect changes can be seen in supplementary figure 15a. As can be seen in the figure, the Hall effect sensors can detect changes up to roughly 11 mm. The influence of the location where the sensors are placed on the muscle can be seen in supplementary figure 15b. All conditions are statistically different from one another. The results of the curved surface test are shown in supplementary figure 15c. All conditions are statistically different, except for the condition where the sensor was mounted on the curved surface versus the condition where the sensor was mounted after the curve.

3.5 Construction of hip simulator

The simulator is built according to the process described in section 2.6. Some minor changes had to be made to the simulator. One of those changes was the removal of the sensor mounts from both the obturator internus and the obturator externus muscles. The reason for this is that these muscles are relatively short and have a line of action that is very close to the joint, specifically the femoral neck. There was simply not enough space for these sensor mounts, as they would have obstructed the movement of the joint. For this reason, the choice was made to keep these muscles in the simulator but remove their sensors. To minimise sensor location influence, most sensors were placed in the middle of their respective muscles. Some were placed in other positions due to a lack of space. A full view photograph of the simulator is shown in figure 18a. More pictures of the simulator can be seen in supplementary figure 18.

3.6 Sensor interface

As was described in section 2.5, an interface is created to visualise the voltage, force and strain values using a *MATLAB* script. The code for this script can be found in appendix D.6. The interface for using the Hall effect sensors is shown in table 1.

3.7 Stability testing by surgeons

The results from the stability tests conducted by the two surgeons can be seen in figure 11. The muscles shown in the graph are the ones that were activated the most during the respective subluxation movement. The results from the traction test can be seen in figures 11a and 11b. The ANOVA test, followed by Tukey's multiple comparisons test, was used to analyse the statistical significance of the differences between measured force and strain across the conditions. Only the muscles that were most active during the movement were analysed. The result is shown in table 2. As can be seen in the table, significant differences where found for the gluteus maximus, gluteus minimus and piriformis, in both force and strain.

The results from the external rotation test, performed by the orthopaedic surgeons, can be seen in figures 11c and 11d. The same statistical analysis method as mentioned above was used. The result is shown in table 3. As is shown in the table, significant differences were observed for the gluteus medius, gluteus minimus and the TFL, in force and strain.

Table 2: Statistical significance of differences between implant configurations for the traction test, conducted by the surgeons.

Muscle	**Comparison**	**p-value (Force)**	**p-value (Strain)**
Gluteus maximus	ANOVA	0.0058	0.0082
	STD + 9 vs. STD + 1	0.645	0.5048
	STD + 9 vs. KHO + 9	0.0348	0.0658
	STD + 1 vs. KHO + 9	0.0058	0.0071
Gluteus minimus	ANOVA	<0.0001	<0.0001
	STD + 9 vs. STD + 1	0.0792	0.0164
	STD + 9 vs. KHO + 9	<0.0001	<0.0001
	STD + 1 vs. KHO + 9	<0.0001	<0.0001
Pectineus	ANOVA	0.2102	0.1877
	STD + 9 vs. STD + 1		
	STD + 9 vs. KHO + 9		
	STD + 1 vs. KHO + 9		
Piriformis	ANOVA	<0.0001	<0.0001
	STD + 9 vs. STD + 1	0.0102	0.0006
	STD + 9 vs. KHO + 9	<0.0001	<0.0001
	STD + 1 vs. KHO + 9	<0.0001	<0.0001

Table 3: Statistical significance of differences between implant configurations for the external rotation test, conducted by the surgeons.

Muscle	**Comparison**	**p-value (Force)**	**p-value (Strain)**
Gluteus medius	ANOVA	<0.0001	<0.0001
	STD + 9 vs. STD + 1	0.049	<0.0001
	STD + 9 vs. KHO + 9	0.0084	0.0175
	STD + 1 vs. KHO + 9	<0.0001	<0.0001
Gluteus minimus	ANOVA	0.0009	0.0006
	STD + 9 vs. STD + 1	0.3839	0.1694
	STD + 9 vs. KHO + 9	0.0128	0.0197
	STD + 1 vs. KHO + 9	0.0008	0.0004
Tensor fasciae latae	ANOVA	<0.0001	<0.0001
	STD + 9 vs. STD + 1	0.0023	<0.0001
	STD + 9 vs. KHO + 9	0.222	0.294
	STD + 1 vs. KHO + 9	<0.0001	<0.0001

3.8 Surgeon's feedback

After the experiment, the surgeons were asked about their experience by the use of a questionnaire. With both configuration changes during the test, both surgeons were able to correctly answer if there had been an offset change. Furthermore, they correctly answered if there had been a positive or negative offset change. They based these answers on the amount of tension that they could feel in the simulator during the test. They felt that this change was noticeable. After viewing the interface, they thought that both the representation of the muscles' strain and force were feasible. They agreed that the simulator could be used as a training tool for surgeons, to improve their capability of assessing hip balance.

Surgeon #1 thought that the information provided by the interface was useful because you can see the difference in force when performing a specific movement. He felt that the pretension of the muscles had greatly improved compared to the previous versions of the hip

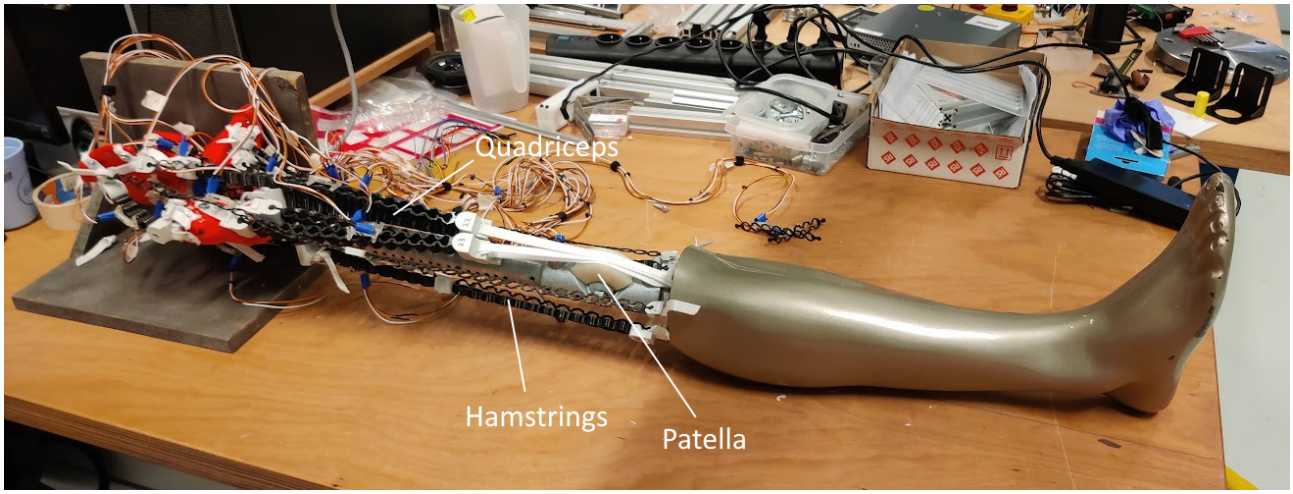


Figure 10: The hip simulator. The quadriceps and hamstrings muscle groups are indicated, along with the patella.

simulator. He found the simulator intuitive in its use but mentioned that it could be a little more robust. He emphasised that the simulator felt close to a real patient.

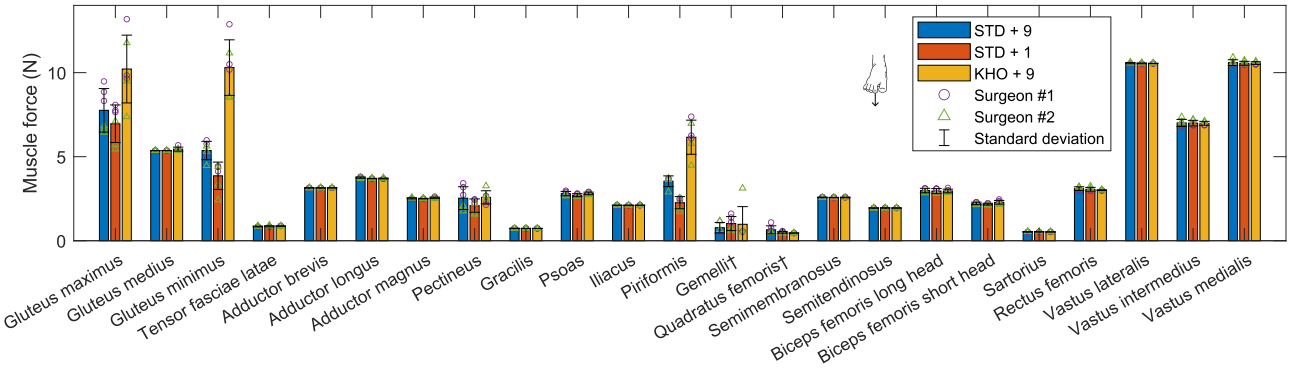
Surgeon #2 strongly agreed that the sensors provided useful information while using the simulator. He found that the consequences of performing a relaxation strategy (such as performing a stretch at the hip using a traction movement) were immediately visible. He also agreed that the simulator felt close to an *in vivo* hip. As a point of critique, he felt that the simulator was not very transparent in use. Because of the high amount of muscles, it was hard to see what exactly was going on in the simulator while performing a certain action with the leg. Furthermore, he mentioned that there were problems due to detachment of two of the muscles during use. Finally, he mentioned that he thought the simulator could not only be used for training orthopaedic surgeons, but also for training physiotherapists. He saw great value in the ability to see what happens to the tension of each muscle while performing a specific motion.

3.9 Additional stability testing

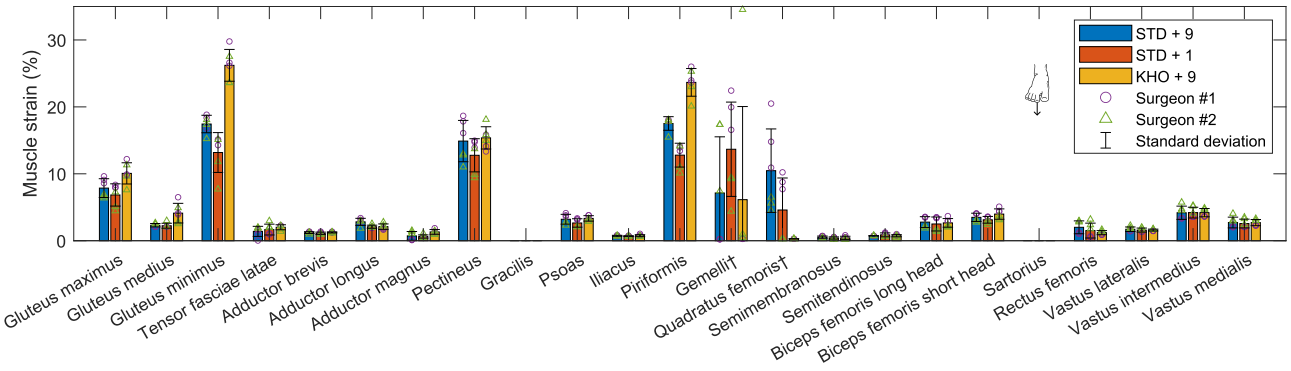
The force measured per muscle during the additional traction test can be seen in figure 12a. The corresponding strain can be seen in figure 12b. To assess the statistical significance for the force and strain across all conditions, the ANOVA test followed by Tukey's multiple comparisons test was, once again, used. Only the muscles that were most activated during the movement were analysed. The result can be seen in table 4. As can be seen in the table, significant differences were observed for the gluteus maximus, minimus, medius, pectineus and piriformis.

The force and strain measured during the additional external rotation test can be seen in figures 13a and 13b respectively. The same statistical analysis as mentioned above was used. The result can be seen in table 5. Significant differences were observed for the gluteus medius, gluteus minimus, TFL and psoas, in both muscle force and strain.

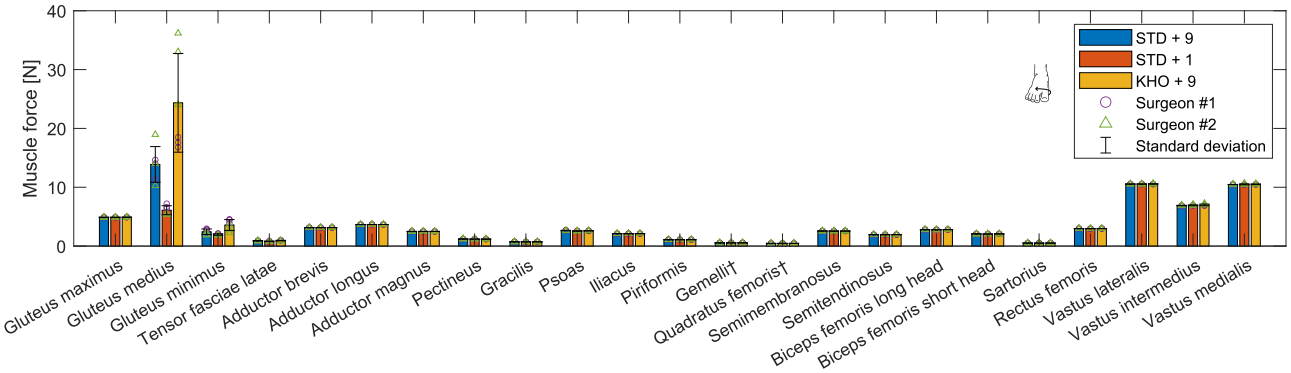
Lastly, the same statistical analysis was performed for the measured properties regarding the whole leg during both tests, namely the total traction force, the external rotation moment and the external rotation angle. The results can be seen in figures 12c, 13c and 13d. The aforementioned statistical analysis methods were also used for these variables, of which the results are shown in 6. As is shown in the table, significant differences were observed for each of these properties.



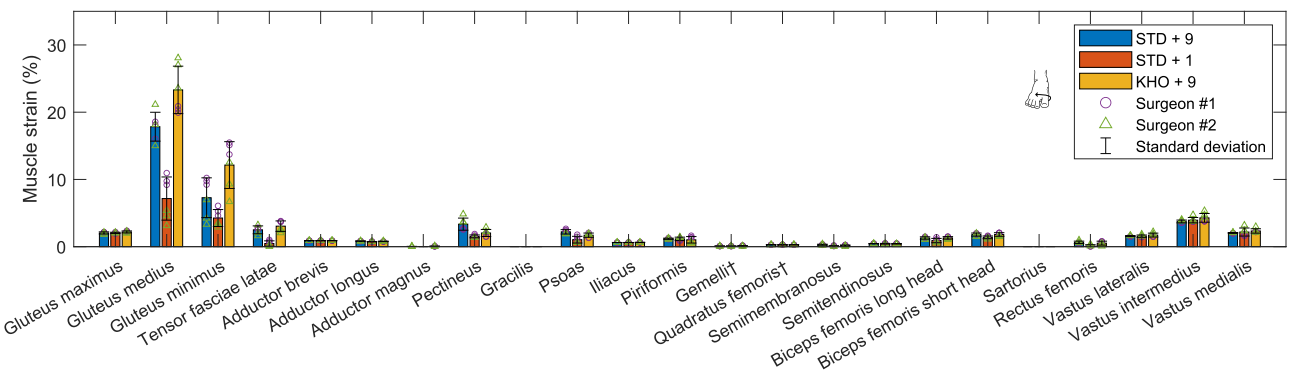
(a)



(b)

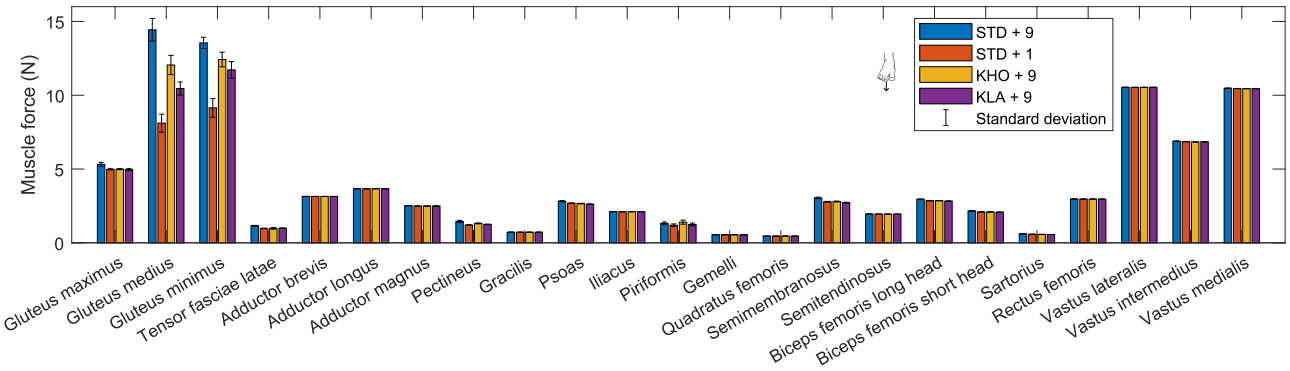


(c)

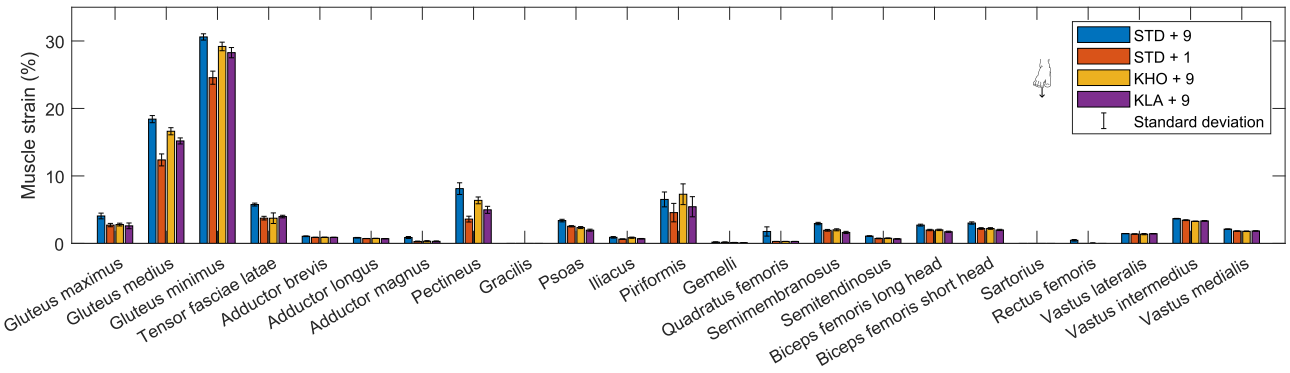


(d)

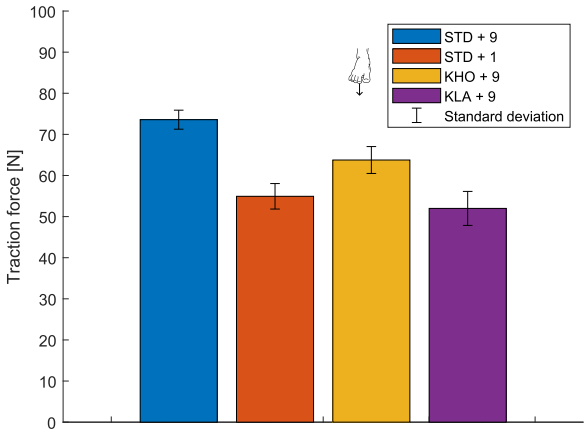
Figure 11: Bar plots showing the data from both the traction and the external rotation test, conducted by two orthopaedic surgeons, for 23 muscles of the lower leg. The movement is depicted to the left of the legend. † Muscle detached during testing. (a) Force measured at subluxation during the traction test. (b) Strain measured at subluxation during the traction test. (c) Force measured at subluxation during the external rotation test. (d) Strain measured at subluxation during the external rotation test.



(a)

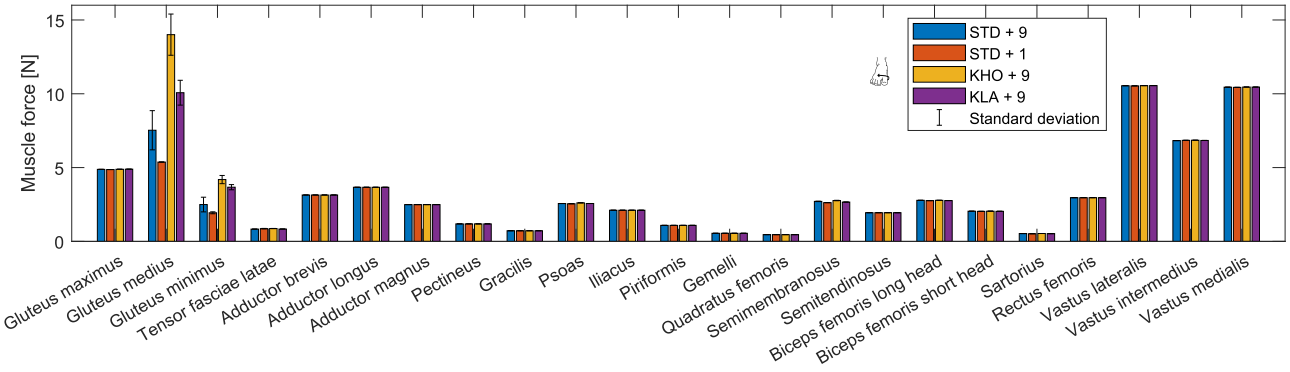


(b)

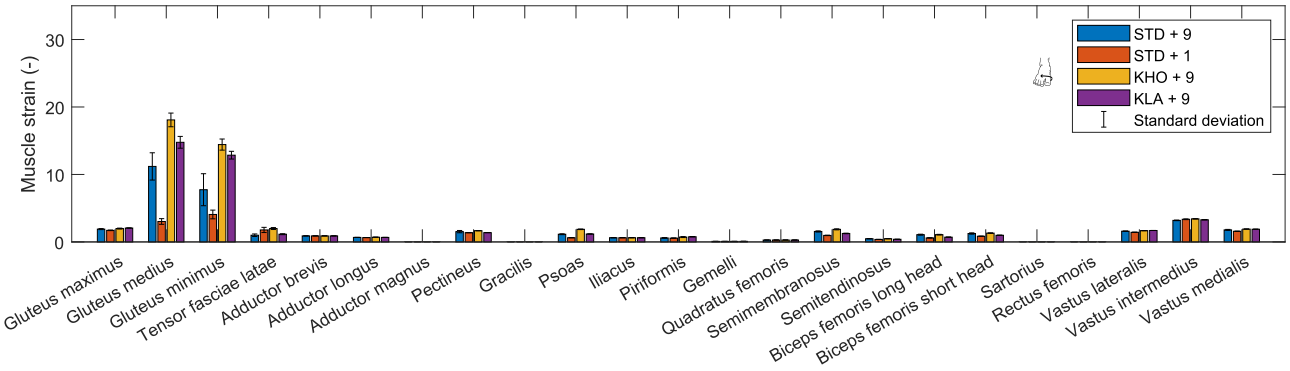


(c)

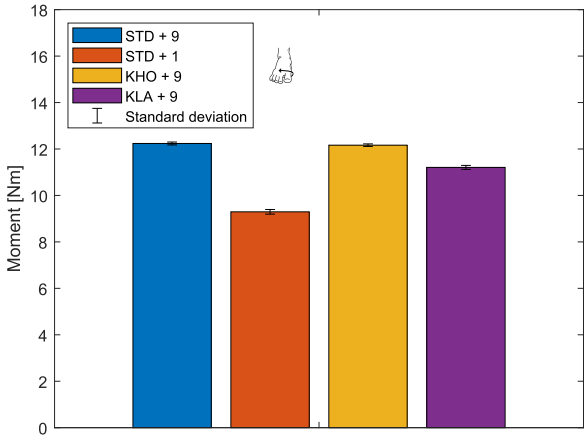
Figure 12: Force and strain plots of additional traction test. The movement is depicted to the left of the legend. (a) Bar plot of additional traction test, showing the measured force in 23 muscles for each configuration. (b) Bar plot of additional traction test, showing the measured strain in five muscles for each configuration. (c) Bar plot of the total traction force measured at the foot, for each configuration.



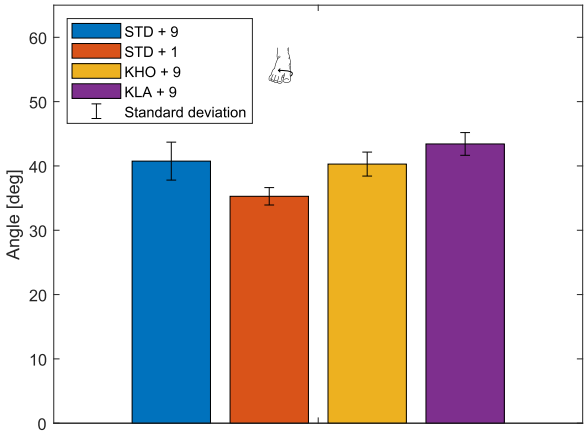
(a)



(b)



(c)



(d)

Figure 13: Plots showing the results of the additional external rotation test. Each implant configuration was tested 10 times. The movement is depicted adjacent to the legend. (a) Bar plot of additional external rotation test, showing the measured force in four muscles for each configuration. (b) Bar plot of additional external rotation test, showing the measured strain in four muscles for each configuration. (c) Bar plot of the subluxation moment at the foot during the additional external rotation test for each configuration. (d) Bar plot of the external rotation angle needed for subluxation, for each configuration.

Table 1: *MATLAB* interface showing the sensor output voltage, muscle force and muscle strain in the simulator. Yellow rows indicate 5-10% muscle strain, orange indicates 10-20% strain, and red indicates $>$ 20% strain.

Muscle	**Bits (0-1023)**	**Voltage [V]**	**Force [N]**	**Strain (%)**
Gluteus maximus	719	3.5142	5.0858	3.2656
Gluteus medius	622	3.0401	13.1962	17.5402
Gluteus minimus	566	2.7664	12.8764	29.7757
Tensor fasciae latae	615	3.0059	1.1146	5.3292
Adductor brevis	961	4.697	3.1435	0.9664
Adductor longus	822	4.0176	3.6668	0.7739
Adductor magnus	763	3.7292	2.5039	0.5968
Pectineus	672	3.2845	1.3233	6.4985
Gracilis	751	3.6706	0.7223	-1.0911
Psoas	671	3.2796	2.7534	3.0005
Iliacus	926	4.5259	2.1111	0.7205
Piriformis	680	3.3236	1.3674	7.1403
Gemelli	833	4.0714	0.549	0.2002
Quadratus femoris	877	4.2864	0.4586	1.3794
Semimembranosus	689	3.3675	2.9363	2.5798
Semitendinosus	750	3.6657	1.9518	0.9777
Biceps femoris long head.	709	3.4653	2.9144	2.4425
Biceps femoris short head	702	3.4311	2.13	2.6788
Sartorius	647	3.1623	0.6036	-0.6972
Rectus femoris	783	3.827	2.9688	0.2591
Vastus lateralis	838	4.0958	10.539	1.4697
Vastus intermedius	738	3.607	6.8953	3.7379
Vastus medialis	740	3.6168	10.4759	2.1062

Table 4: Statistical significance of differences between implant configurations for the additional traction test.

Muscle	**Comparison**	**p-value (Force)**	**p-value (Strain)**
Gluteus maximus	ANOVA	$<$0.0001	$<$0.0001
	STD + 9 vs. STD + 1	$<$0.0001	$<$0.0001
	STD + 9 vs. KHO + 9	$<$0.0001	$<$0.0001
	STD + 9 vs. KLA + 9	$<$0.0001	$<$0.0001
	STD + 1 vs. KHO + 9	0.9871	$<$0.0001
	STD + 1 vs. KLA + 9	0.9862	$<$0.0001
	KHO + 9 vs. KLA + 9	0.9066	0.9808
Gluteus medius	ANOVA	$<$0.0001	$<$0.0001
	STD + 9 vs. STD + 1	$<$0.0001	$<$0.0001
	STD + 9 vs. KHO + 9	$<$0.0001	$<$0.0001
	STD + 9 vs. KLA + 9	$<$0.0001	$<$0.0001
	STD + 1 vs. KHO + 9	$<$0.0001	$<$0.0001
	STD + 1 vs. KLA + 9	$<$0.0001	$<$0.0001
	KHO + 9 vs. KLA + 9	$<$0.0001	$<$0.0001
Gluteus minimus	ANOVA	$<$0.0001	$<$0.0001
	STD + 9 vs. STD + 1	$<$0.0001	$<$0.0001
	STD + 9 vs. KHO + 9	0.0001	0.0007
	STD + 9 vs. KLA + 9	$<$0.0001	$<$0.0001
	STD + 1 vs. KHO + 9	$<$0.0001	$<$0.0001
	STD + 1 vs. KLA + 9	$<$0.0001	$<$0.0001
	KHO + 9 vs. KLA + 9	0.024	0.0374
Pectineus	ANOVA	$<$0.0001	$<$0.0001
	STD + 9 vs. STD + 1	$<$0.0001	$<$0.0001
	STD + 9 vs. KHO + 9	$<$0.0001	$<$0.0001
	STD + 9 vs. KLA + 9	$<$0.0001	$<$0.0001
	STD + 1 vs. KHO + 9	$<$0.0001	$<$0.0001
	STD + 1 vs. KLA + 9	0.1261	$<$0.0001
	KHO + 9 vs. KLA + 9	0.0035	$<$0.0001
Piriformis	ANOVA	0.0006	0.0005
	STD + 9 vs. STD + 1	0.0509	0.0163
	STD + 9 vs. KHO + 9	0.3448	0.6061
	STD + 9 vs. KLA + 9	0.4112	0.3134
	STD + 1 vs. KHO + 9	0.0006	0.0005
	STD + 1 vs. KLA + 9	0.6791	0.5049
	KHO + 9 vs. KLA + 9	0.0127	0.0246

Table 5: Statistical significance of differences between implant configurations for the additional external rotation test.

Muscle	**Comparison**	**p-value (Force)**	**p-value (Strain)**
Gluteus medius	ANOVA	$<$0.0001	$<$0.0001
	STD + 9 vs. STD + 1	0.0003	$<$0.0001
	STD + 9 vs. KHO + 9	$<$0.0001	$<$0.0001
	STD + 9 vs. KLA + 9	$<$0.0001	$<$0.0001
	STD + 1 vs. KHO + 9	$<$0.0001	$<$0.0001
	STD + 1 vs. KLA + 9	$<$0.0001	$<$0.0001
	KHO + 9 vs. KLA + 9	$<$0.0001	$<$0.0001
Gluteus minimus	ANOVA	$<$0.0001	$<$0.0001
	STD + 9 vs. STD + 1	0.001	$<$0.0001
	STD + 9 vs. KHO + 9	$<$0.0001	$<$0.0001
	STD + 9 vs. KLA + 9	$<$0.0001	$<$0.0001
	STD + 1 vs. KHO + 9	$<$0.0001	$<$0.0001
	STD + 1 vs. KLA + 9	$<$0.0001	$<$0.0001
	KHO + 9 vs. KLA + 9	0.0025	0.0556
Tensor fasciae latae	ANOVA	$<$0.0001	$<$0.0001
	STD + 9 vs. STD + 1	$<$0.0001	$<$0.0001
	STD + 9 vs. KHO + 9	$<$0.0001	$<$0.0001
	STD + 9 vs. KLA + 9	0.6797	0.4565
	STD + 1 vs. KHO + 9	0.2154	0.1955
	STD + 1 vs. KLA + 9	$<$0.0001	$<$0.0001
	KHO + 9 vs. KLA + 9	$<$0.0001	$<$0.0001
Psoas	ANOVA	$<$0.0001	$<$0.0001
	STD + 9 vs. STD + 1	$<$0.0001	$<$0.0001
	STD + 9 vs. KHO + 9	$<$0.0001	$<$0.0001
	STD + 9 vs. KLA + 9	0.4897	0.1856
	STD + 1 vs. KHO + 9	$<$0.0001	$<$0.0001
	STD + 1 vs. KLA + 9	$<$0.0001	$<$0.0001
	KHO + 9 vs. KLA + 9	$<$0.0001	$<$0.0001

Table 6: Statistical significance of differences between implant configurations for variables regarding the whole leg during the additional stability tests.

Variable	Comparison	p-value
Total traction force	ANOVA	<0.0001
	STD + 9 vs. STD + 1	<0.0001
	STD + 9 vs. KHO + 9	<0.0001
	STD + 9 vs. KLA + 9	<0.0001
	STD + 1 vs. KHO + 9	<0.0001
	STD + 1 vs. KLA + 9	0.2024
	KHO + 9 vs. KLA + 9	<0.0001
External rotation moment	ANOVA	<0.0001
	STD + 9 vs. STD + 1	<0.0001
	STD + 9 vs. KHO + 9	0.9988
	STD + 9 vs. KLA + 9	0.1792
	STD + 1 vs. KHO + 9	<0.0001
	STD + 1 vs. KLA + 9	0.0024
	KHO + 9 vs. KLA + 9	0.2334
External rotation angle	ANOVA	<0.0001
	STD + 9 vs. STD + 1	<0.0001
	STD + 9 vs. KHO + 9	0.9593
	STD + 9 vs. KLA + 9	0.0319
	STD + 1 vs. KHO + 9	<0.0001
	STD + 1 vs. KLA + 9	<0.0001
	KHO + 9 vs. KLA + 9	0.0091

4 Discussion

4.1 Key findings

The results presented in sections 3.7 to 3.9 suggest that for some muscles, increased femoral offset does result in an increased soft tissue tension around the joint. Furthermore, the experiences of the surgeons during the tests are consistent with these findings. In addition, an increased femoral offset caused an increased external rotation moment and angle required for subluxation. These findings provide support for both hypotheses stated in section 1.4, and are discussed in more detail in section 4.7.

4.2 Characterisation

In section 2.2, the choice was made to select the stiffer *TPU 95A* by *Ultimaker* as the material for printing the phantom muscles of the hip simulator. This choice was made due to the poor printability of *NinjaFlex* filament. Although, at the time, there were doubts that the material would be too stiff, this turned out to be a good decision. The reason for this is that the smaller muscle size meant that there was more room for the sensor mounts. While building the simulator, one could see in person how complex the hip joint is, and even with the small muscle width of 14.36 *mm* (see supplementary figure 10), there were some issues regarding available space.

The curve fitting tool in *ANSYS* using the 2-parameter Mooney-Rivlin hyperelastic model, which can be seen in figure 8, provided a good quality fit. This fit was used later to predict the material's tensile behaviour in specific shapes.

4.3 Shape design using FEA

In figure 9, the difference between the experimental and simulated results using FEA can be seen. The experimental samples were significantly stiffer than was expected from FEA. There are a few possible causes for this. Firstly, the samples were clamped in the tensile tester. This compresses the material, possibly making it less compliant during the test. Secondly, the TPU filament was characterised using only data from a uniaxial test, which may have limited the accuracy of the *Mooney-Rivlin* model. *Kim et al.* have shown that the Mooney-Rivlin model can be inaccurate for this condition [42]. Furthermore, 3D-printed materials are known to have variable material properties, as they are subject to many factors, such as printing quality, printing orientation, layer height and thickness and printing temperature. In the finite elements method (FEM) simulation, the material was assumed to be uniform. However, due to the anisotropic nature of FDM printing, this may have also been a limiting factor. Lastly, FEA can be inaccurate due to limiting factors of the computer. For example, choosing a smaller element size can increase the accuracy at the cost of calculation speed.

Despite the stiffness discrepancy, the hourglass shape did achieve its intended goal. The sample has a clear strain-stiffening effect, which starts at ∼15% strain and achieves its maximum stiffness at ∼30%, which can resemble human muscle tissue [33]. The difference between the experimental and simulated results did not influence the properties of the muscles, as the results from this test were scaled to calculate the phantom muscle dimensions used for the simulator.

The hourglass shape is designed in such a way that it can be scaled to meet the required dimensions of each muscle. Since the shape is a repeated pattern, the length of the muscles should in theory not impact the force profile relative to the strain. The force can be scaled with the thickness, as the cross-section height is proportional to the output force of the phantom muscle.

4.4 Material, shape and sensor testing

As was described in section 2.7, various properties of the hourglass-shaped TPU were analysed. As can be seen in supplementary figure 14a, the material tends to become tougher as time passes (when not loaded). This is something to take into consideration when using the simulator, since the desired strain-stiffening effect tends to become less prevalent, although still visible after a time of five months. Unfortunately, clamping seems to have a significant influence on the characteristics of the shape. This effect can be observed in supplementary figure 12a. As one would expect, clamping compresses the material and therefore increases the stiffness. It is important to consider this effect when calculating the phantom muscle dimensions.

The TPU sample is significantly dependent on strain rate (see supplementary figure 14c). This is likely due to the viscoelastic properties of the polymer, and it is known that TPU is prone to this phenomenon [43]. Strain rate dependence is a viscoelastic property that is also present in muscle, as was shown by *Zhai et al.* [44]. Although there seems to be a more significant stiffness increase reported by *Zhai et al.*, one cannot draw any conclusions as the TPU sample was not tested at a rate as high as $100s^{-1}$, as this was not possible with the available equipment. What does become apparent from these results, is that it is important to note that the elongation speed at which the material is characterised is comparable to that applied to the muscles during balancing procedures.

As can be seen in supplementary figure 14d, the TPU sample is prone to hysteresis. During the test, the machine changed from tension to compression as fast as it could. Still, stress relaxation is visible. This is also the case for the stress relaxation test, seen in supplementary figure 14e. Here, the sample was held in place at a strain of 50% for 30 seconds. From these results, it becomes apparent that the simulator will lose tension over time. Although stress relaxation in *in vivo* muscle is also present, it is not as strong as exhibited by the TPU sample. *Taylor et al.* reported an 11 N decrease in force after 30 seconds while initially stretching the muscle to a force of 78.4 N [45]. *McHugh et al.* reported a decrease of 11.35 ± 1.76 N after 45 seconds, after a peak force of 65.75 ± 5.31 N [46]. The stress relaxation test of the TPU sample resulted in a loss of 10.96 ± 1.09 N from a peak force of 33.4 ± 0.82 N, which was recorded over a time of 30 seconds. This is significantly higher than is the case with muscle. It is however worth noting that the *in vivo* MTUs are not likely to have been stretched to 50 % of their original length. Further research into the mechanical properties of MTUs needs to be conducted to see how close TPU is to soft tissue, although these results seem to indicate that this effect is more pronounced in the polymer.

The results from the creep test can be found in supplementary figure 14f. Even after 10 minutes, creep is still visible in the figure. No studies were found that examined creep in human muscle for this amount of time. However, *Ryan et al.* examined the viscoelastic creep in a human *in vivo* MTU for 30 seconds [47]. This study found that 37% of the total creep could be observed in the muscle during the first 5 seconds. For the TPU muscle, 71% of the creep could be observed during the first 5 seconds. At 10 seconds of constant load, the material crept an additional 11% of the total creep observed after 30 seconds. For human tissue, this number was 24%. This comparison seems to indicate that TPU muscle is more prone to viscoelastic creep, although the results cannot be directly compared, since *Ryan et al.* measured the position in degrees of a joint while applying a constant torque, and not the linear position of a single MTU. From this test and the aforementioned ones, it can be stated that it would be detrimental to the simulator's performance if it is used for extended periods of time, and that the phantom muscles should be given time to return to their original length between sessions.

The Bland-Altman plots shown in supplementary figures 15d and 15e show the sensor method to be accurate, as the mean differences between the sensor method and the gold standard are equal to 0.00 and 0.17 respectively. As can be seen in the figures, there is an increased bias at the lower strains and forces, which was expected, as is explained in section 4.8. For the strain, the bias also increases towards the end of the range, however this is far above the strains measured during testing (which never reached 40%). Overall, the reproducibility coefficients $RPC_\varepsilon = 0.0305$ and $RPC_F = 3.48$ N indicate acceptable precision across the entire range.

4.5 Force and strain measurement

In supplementary figure 15a, the sensor output voltage is measured while increasing the distance between the magnet and the sensor. As can be seen in the figure, the sensors can detect changes up to approximately 15 mm. Since the voltage sensor of the Arduino has a resolution of 1024 bits, the sensors would not be accurate in the 10-15mm range. This is not a problem for the usage of the Hall effect sensors in the simulator. In the figure, one can see that the accuracy significantly starts to decrease after approximately 10 mm. A distance of 10 mm equates to a strain of 46%, which is far above strains that have been reached while using the simulator. In section 2.8, it was explained that each muscle was tested on a tensile testing machine while measuring the sensor output. The goodness-of-fit data can be found in appendix F. For all muscles, $R^2 > 0.99$ for the displacement, and $R^2 > 0.97$. It can therefore be concluded that this part of the used measurement method is accurate, although the viscoelastic properties of the TPU muscles may cause inconsistencies.

In supplementary figure 15b, the influence of sensor location on the voltage is shown. As was shown in section C.1, there is a statistically significant difference between these conditions. Between the means, the largest measured difference amounted to 18 ± 10.6 bits. Depending on the muscle, this may cause significant discrepancies in measured force and strain. Fortunately, this discrepancy mainly occurs in the first 10% of strain, which amounts to a displacement of 2.16 mm in the sensor mount. Due to the voltage following a power function in relation to the displacement (see supplementary figure 15a), a difference of 18 ± 10.6 bits is not as impactful in this range as it is with higher strains. Because of the large standard deviation resulting from this test, which was likely caused by the viscoelastic properties of the samples, further research should be conducted to see how the sensor location ex-

actly impacts the sensor output. For the most accurate measurements, one should attempt to place the sensors as close to the centre of the muscle as possible. If this is not possible due to spatial limitations, one should consider doing the calibration process with the sensors mounted in the same location as where they will be placed in the simulator.

In supplementary figure 15c, it can be seen how wrapping a piece of muscle around a circular surface impacts the sensor output. As was mentioned in appendix B.4, there is no statistically significant difference between mounting the sensor on the curved surface or having it between the curved surface and the load. When the sample is wrapped around the circular profile, there will be friction present between the profile and the sample. Since the force is applied to the bottom part of the sample, the tension in this part will be higher than that in the part between the profile and the attachment (see supplementary figure 11). Therefore, one would expect the output voltage of the sensor to be higher when the sensor is mounted between the attachment and the circular profile (before the curve) since there is less deformation present in this part. The results shown in supplementary figure 15c confirm this. It is important to keep this in mind for future research, as the sensors should output the average force and strain of the phantom muscle, instead of that of a specific part of its length. This could be achieved by minimising the friction between the phantom muscles and the other parts of the simulator.

The interface used for live reading of the phantom muscles' force and strain can be seen in table 1. Colours were used so that the user can quickly see which muscles are under tension. The live interface responded to changes quickly (< 0.5 s), which made it adequate for a live demonstration. Furthermore, the *MATLAB* script always saves the data so that the data can be analysed after using the simulator. As can be seen in the table, the strains of muscles are never exactly zero, and sometimes even negative. This is due to mainly two things. Firstly, it is not possible to achieve a 1:1 fit of the sensor output when using a power function. Secondly, the viscoelastic properties of the muscles make it impossible to always accurately predict the force and deformation of the muscle. This issue presents itself in the simulator because the phantom muscles are under pretension, and simply using the simulator to perform movements will load the phantom muscles and change their length and internal stress.

4.6 Physical hip simulator

The simulator was intuitive in use and an improvement over the design by *van Breukelen et al.* in a few areas [41]. The current version of the hip simulator is the most detailed yet, consisting of all 25 muscles which cross the hip and the knee. This resulted in a more realistic simulation of the hip's balance. Some parts of the previous iteration were used in this version, such as the 3D-printed PLA pelvis and femoral head, the frame to which the pelvis was attached, the steel tubes resembling the femur, the steel joint resembling the knee, and the wooden lower leg. Aside from the amount of phantom muscles, more additions were made. These include the PETG clips, the PLA patella and the sensor system. The design of the muscle connectors was largely based on that of the previous version. It was improved slightly by increasing the head-to-neck ratio of the T-fit, to allow for a more secure connection. The fabrication process of the phantom muscles was also improved. In the previous version, the muscles were created using laser-cut polymethyl methacrylate (PMMA) moulds, in which the TPU was cast. Now, the phantom muscles are directly 3D-printed, which is simpler and a more efficient use of materials.

Besides the enhanced musculature, the most important improvement to the previous version of the hip simulator is the addition of the Hall effect sensors. The sensors greatly improve the potential of the hip simulator as equipment for the education of orthopaedic surgeons and other personnel specialised in physiology. Moreover, when testing different implant configurations, they can be used to validate the differences in hip balance that the surgeon can feel. This provides another improvement over the last version, where the surgeon was only able to feel the difference, which is more prone to confirmation bias.

Aside from the previous version of the hip simulator, perhaps the closest tool to a physical hip simulator is the software package *OpenSim* [13]. This is a convenient package which was also occasionally used for this project to better visualise the hip. Unfortunately, *OpenSim* did not appear to be very reliable for passive muscle stiffness, as both the moment arms and force responses to stretching were not consistent. The hip simulator offers a physical alternative in this regard.

4.7 Stability testing

In figure 11, the results from the stability testing experiment, conducted by surgeons, are shown. As muscle force largely differs per specific muscle, the most information can be obtained by looking at the muscle strain shown in figures 11b and 11d. In figure 11b, one can see that the gluteus maximus, gluteus minimus, pectineus and piriformis muscles were activated the most during the traction movement. For these muscles, most of the differences between implant configurations were statistically significant (see section 3.7). In section 1.4, it was hypothesised that the tension in the muscle would increase with an increased femoral offset. The data for the gluteus maximus, minimus and piriformis muscles suggests that this was the case in the simulator. This was also consistent with the sensation of the corresponding surgeon during testing.

According to one of the surgeons, the strain would

be the most prominent in the rotators of the hip during the traction test. *Gray's Anatomy* states that the lateral rotators of the hip include the piriformis, obturator internus, obturator externus, gemelli, quadratus femoris, gluteus maximus and biceps femoris [48]. The medial rotators of the hip include the gluteus minimus, gluteus medius, adductor longus, adductor brevis, adductor magnus, semitendinosus and semimembranosus. The muscles which were most active all belong to this group. Most muscles in this group are not represented in the results. There were attachment problems with the gemelli and quadratus femoris during testing, which produced inconclusive results. In figure 11b, it does seem likely that these muscles are very active during a traction movement, although precise data is not available for these muscles for this experiment. The obturator internus' and externus' sensors were removed, so these muscles may also have been activated during the experiment. It is not clear why the adductors and hamstrings were not activated during the traction movement. It may be possible that this is due to the leg position, but unfortunately, there is no analysis in the literature on which muscles are most stretched during a traction movement.

The results from the external rotation test can be seen in figures 11c and 11d. The results provide additional support for the first hypothesis stated in section 1.4, as increased tension around the joint would also result in increased external rotation stiffness. As was explained in section 3.7, this effect can be observed the most in the gluteus medius, gluteus minimus and tensor fasciae latae. The same trend is visible as with the traction test; the higher the femoral offset, the higher the tension in the corresponding muscles. External rotation stiffness is caused by the resistance of the medial rotator muscles to stretching. As was also the case with the traction test, the adductors and the hamstrings do not seem to have a significant contribution to external rotation stiffness, possibly due to leg position. In the literature, there appears to be a gap in research on muscle contribution while performing medial rotation. Interestingly, a recent study by *Martins et al.* indicates that the superior portion of the gluteus maximus assists in lateral rotation when the leg is fully extended [49]. The results from this test would also indicate that this is the case. The same study reports that while there is a consensus in anatomy that the TFL is a medial rotator, they observed that the TFL assists in lateral rotation with the hip at 90 degrees of flexion. Unfortunately, the surgeons only performed an external rotation movement with the hip fully extended, as they followed their preferred protocol to assess hip balance. The function of the TFL regarding hip medial and lateral rotation is not precisely known [50]. The results from the external rotation test seem to indicate that the muscle does not have a significant function as a medial rotator, but since the muscle's tension increases with offset, it does appear to have a stabilising role with regard to the hip joint.

After the aforementioned experiments, the traction and external rotation tests were repeated without the surgeons. The results from these tests can be seen in figures 12 and 13. From the traction test shown in figure 12, a notable difference with the test conducted by the surgeons is the tension observed using the *STD + 9* configuration. This configuration does not have the highest femoral offset out of the tested configurations. It is unclear what is the cause of this, though it likely has to do with the fact that this test was not conducted by someone who is specialised in performing these movements (see section 4.8). This may also explain why there is a difference in the muscle contribution compared to the surgeon's experiment. For example, the gluteus maximus, pectineus and piriformis are more stretched during the surgeon experiment, while the gluteus medius and minimus are more stretched during this second experiment. One can however still observe a consistent trend in the fact that *STD + 1* causes far lower tension in the muscles than the higher offset necks do. The KLA neck has a shorter length than *KHO + 9*, but longer than *STD + 1*. A lateralised neck will result in a shorter leg length than a neck with the same length that is not lateralised. Thus, one would expect lower tensions in a traction test than for example the *KHO + 9*. The results from the traction test may confirm this hypothesis. In figure 12c the total force measured at the foot for subluxation is shown. This data shows that the measurements by the sensors are consistent with the total traction force exerted on the leg, since *STD + 9*, *STD + 1* and *KHO + 9* are all statistically different from one another.

The results from the additional external rotation test are depicted in figure 13. These results are more consistent with those obtained from the surgeon experiment. As was the case there, here one can also in figure 13b see that the gluteus medius and gluteus minimus are the muscles most affected by the movement. The configurations are statistically different for these muscles. A noticeable difference with the results from the additional traction test is how the *KLA* neck compares to the other configurations. As the medial rotator muscles have their line of action in the transverse plane, one would expect a more lateralised neck to have a relatively larger effect on joint stiffness than is the case with a traction test. The results support this proposition, as for the gluteus medius, the *KLA* neck, which has approximately the same length as the *STD* (see supplementary figure 19a), falls between the *STD + 9* and *KHO + 9* in terms of tension. For the gluteus minimus, the effect is similar as the *KLA + 9* results in higher tension than the *STD + 9*, although the difference between the *KHO* and *KLA* neck is not statistically significant. Interestingly, the TFL tends to show a higher tension for the *STD + 1* than for the *STD + 9*. Compared to the surgeon experiment, the TFL, pectineus and piriformis do not seem to be as

stretched. This may, once again, be due to improper execution of the subluxation movement. Furthermore, as was stated in section 4.5, the sensors are not accurate at low strains, so no conclusions can be drawn from the data of these phantom muscles.

In figures 13c and 13d, the total moment required for subluxation applied at the foot and the external rotation angle at subluxation are shown. For the subluxation moment, significant differences were found between the *STD + 1* configuration compared to the other three. Additionally, significant differences were found between the subluxation angles as well, as only *STD + 9* vs. *KHO + 9* did not have a significant outcome. The simulator may not be sensitive enough to detect the smaller changes in femoral offset between the longer necks. Nevertheless, as *STD + 1* results in a significantly shorter neck length than the other three, the second hypothesis stated in section 1.4 is supported by the results of this test. Moreover, a more lateralised neck resulting in increased external rotation ROM is consistent with the findings of *Burzynski et al.* as *KLA + 9* resulted in the highest subluxation angle [16]. Further testing is needed to accurately assess if the simulator is sensitive to smaller changes in femoral offset and leg length.

In section 3.8, the feedback from the surgeons is presented. They were able to correctly answer if there had been an offset change, suggesting that the simulator was sensitive to offset changes. They were both enthusiastic about the potential of the simulator, generally in an educational sense. It is perhaps in this aspect that the simulator has the greatest potential, as no such device currently exists. It allows students to obtain a better grasp of the influence of both specific movements and implants on the biomechanics of the hip. In addition, a physical, visual representation of the hip would help in this regard, although there is room for improvement.

4.8 Limitations

This design project was subject to some limitations. Firstly, special grips designed for polymers, such as concentric roller grips, were not available for use at *TU Delft*. For this reason, the TPU sample was clamped during the characterisation tensile tests described in section 2.2. This may have had an impact on the characterisation, modelling the material as stiffer than it is. Furthermore, FDM-printed materials are anisotropic by nature, as they are printed in a layer-by-layer fashion. When modelled in FEA software, however, the material is assumed to be anisotropic, possibly leading to inaccuracies. These choices may have attributed to the discrepancy between the simulated and experimental results of the hourglass shape tensile test shown in figure 9. For future research into FEM of these materials, it is recommended to use adequate grips when characterising the material. In addition, a 2D-tensile test could have been conducted, which could have resulted in a more accurate characterisation using the Mooney-Rivlin hyperelastic model [42]. Additively manufactured materials have varying mechanical properties based on printing settings. This may also have impacted the simulation of the hourglass sample.

About how passive characteristics scale with muscle architecture, much is still unknown [35]. When researching rabbit muscle, it was found that the slope of the stress-strain relationship curve, the modulus, changes non-linearly when increasing the scale from fibre, to bundle, to whole muscle [51]. *Ward et al.* also measured the titin molecular mass, which was previously thought to be the primary determinant of passive stiffness [52]. However, *Ward et al.* suggest that this stiffness can be mainly attributed to the extracellular matrix (ECM). Moreover, *Ward et al.* suggest that passive scaling is muscle-specific. A systematic review into passive muscle stiffness conducted by *Binder-Markey et al.* contradicts this hypothesis, suggesting that species, size scale (fibre/bundle/fascicle/tissue/-muscle) and type of stretch are significant, whereas muscle region is not [32]. Therefore, it can be concluded that further research into passive muscle scaling is required to accurately predict the tension in each muscle. Due to this gap in the literature, the choice was made to scale the passive muscle stiffness based on PCSA. This decision did result in sufficient resemblance to an *in vivo* hip according to the two surgeons. It should also be noted that as is the case with any soft tissue, there is a large natural variability in the tensile characteristic of human muscle tissue [33]. Therefore, one should also question if it is worth the time and resources to attempt to make the simulator more accurate in this regard. The same is true for the assumption of a maximum *in vivo* muscle strain of 30%. As people have varying degrees of joint flexibility, this number is also likely to be highly subjective, and may also vary largely per muscle. Further research into the maximum *in vivo* strain per muscle over a large sample size may increase the accuracy of the simulator. No such study currently exists.

There were also some limitations in the design choices made for the hip simulator. Firstly, to accurately estimate the muscle lengths for proper pretensioning, pieces of rope were attached to the phantom muscles' insertions and origins. This assumes a straight line of action for the muscles, which is an oversimplification. Musculoskeletal models such as *OpenSim* have divided larger muscles with fibres running in multiple directions, such as the gluteus maximus, into different components, which may yield a more accurate simulation of the joint's mechanics. Following the curved trajectory of long muscles such as the sartorius, would require more accurate replication of the muscles' shape and size. The downside of this method is that it would not be possible to use a standardised approach, such as the one used for this project with the scaling of the

hourglass shape. Further research could look into using the hourglass shape as the main tensile element while filling up the rest of the muscle with a soft, compliant material which does not disrupt the tensile characteristic of the TPU. Secondly, the clips caused problems during the experiment with the surgeons. A redesign was however made later, which provided a tighter fit and solved the problem.

TPU 95A proved to be a difficult material to print. For this project, all muscles were printed by the staff of the Employee Workshop at *TU Delft*. They experienced problems in the feeder of the FDM printer, as well as clogging issues at the extruder. Due to this, there was a large variability in the printing quality of the phantom muscles. Many of the TPU muscles were more compliant than intended as a result of this. Whether this higher stiffness would have resulted in a more accurate simulation of the hip joint remains to be seen. The choice was made to use these muscles of lesser quality anyway, due to time constraints. They still caused issues, as the sensors were significantly harder to mount to poorly printed muscles. This was caused by the attachment loops (see supplementary figure 10) often lacking a hole to put the screw into. Fortunately, this problem could often be fixed by using a hand drill with a small diameter. With phantom muscles of good printing quality, mounting the sensors is not an issue. In addition, due to the discrepancy in print quality across all phantom muscles, the influence of phantom muscle length and thickness on the force-strain profile could not be analysed. It is hypothesised that the force scales with thickness and that the length has no influence on the force-strain profile. This was confirmed using FEA, but future testing needs to be conducted to experimentally evaluate these hypotheses.

The heavy stress relaxation of the material, observed in 14e, caused difficulties in multiple regards. While calibrating the sensors, the samples became more compliant the more they were stretched, exhibiting hysteresis. This makes it difficult to predict the tensile behaviour of the muscle. For this reason, the same sample was stretched multiple times to obtain an average stiffness profile. The stress-relaxation properties also caused the TPU muscles to lose some of their pretension in the simulator over time. It is difficult to address this problem, as simply increasing the pretension will still result in a loss of tension over time (see section 4.4), and would likely still cause the phantom muscles to become slack. Furthermore, as discussed in section 4.5, the viscoelastic properties of TPU made it difficult to fit the tensile characteristic at lower strains, leading to inaccuracies in this strain range. Another limitation of the material is the shelf life. The strain stiffening characteristic is an important aspect of the replication of the human muscle stretch profile. As this effect becomes less prominent over time, it is important to keep in mind that it may be necessary to often replace the muscles while using the simulator.

Regarding the force and strain measuring method, there are also some limitations to this project. As was explained in section 4.5, sensor placement has a significant effect on the output voltage of the sensors. Interestingly, both the psoas and the iliacus were not activated to a significant extent during the traction test (see figure 11b). As these muscles have their line of action in a direction that is relatively parallel to the direction of the applied force to the leg, one would expect to see a significant strain on these muscles during this type of test. The findings regarding sensor location may explain this, as the sensors were placed between the attachment and the pelvis, and these muscles were one of the few ones which were bent along a surface. This may have caused the sensors to record a smaller strain than was the case along the entire length of the muscles. In the future, this phenomenon could be minimised by reducing the friction between the TPU muscles and the PLA pelvis, for example by placing strips of a smooth material on the surface of the pelvis. This would distribute the tension better across the length of the phantom muscle, resulting in a more accurate displacement measurement. The execution of the sensor method comparison using the Bland-Altman plots was not ideal, as only calibration data was used for this comparison, while a separate testing session after the calibration would have been preferred. A separate session after calibration is recommended for future research, to better evaluate the force and strain measurement method.

Some choices were made regarding the stability testing which may have impeded the results. Due to restricted time with the surgeons, the amount of tests was reduced. It became apparent from the differences between the two testing sessions that adequacy with ROM testing is an important aspect when using the simulator. Ideally, the two tests should have been combined, with all tests conducted by orthopaedic specialists with the four configurations, while keeping the leg in the same position (see supplementary figure 13). This would have likely led to more consistent results, with more clarity about the role of a lateralised implant.

4.9 Advantages and future implications

To imitate the characteristic of muscle tissue, an hourglass shape for additive manufacturing was designed. This concept could be applied in other fields as well. For one, other soft tissues such as skin could be replicated using this shape, in the fields of prosthetics and orthotics. Additionally, these shapes could be made using biocompatible materials for scaffolds in tissue engineering. Another possible application is to make actuators or exoskeletons in the field of soft robotics. Accurate replications of muscle tissue could help prevention or rehabilitation of sports injuries by providing

a better understanding of muscle dynamics. Finally, replications of muscle tissue could be used for crash dummies in impact testing.

A method was devised to measure the force and strain in a flexible material. The method uses a magnet and a Hall effect sensor, glued to an SLA resin sensor mount. These sensor mounts are then screwed on the TPU muscles using small bolts. This method could be interpreted as a universal way of gauging the strain of a flexible material, and its possible applications are not limited to this usage. For one, this technology could be used to measure deformations of materials in soft robotics and prosthetics. Another possible example is to measure the deformation in flexible parts of vehicles or monitor the creeping of flexible materials in buildings.

The hip simulator has promising applications within the medical field. Firstly, it provides a physical model of the hip joint, which can be used to demonstrate the functions of each muscle of the joint and how it contributes to specific motions. The physical representation is supplemented with a numerical one, provided by the Hall effect sensors. As was mentioned in section 3.8, both orthopaedic surgeons and physiotherapists may benefit from using this tool in an educational sense. Another benefit of the simulator is to familiarise orthopaedic surgeons with the motions used to assess hip balance and range of motion. This is another way in which other experts in the field of physiology may also benefit. In addition, the simulator can be used to preoperatively estimate the effects of different implant configurations. The ability of the simulator to serve as a tool for surgeon training and implant influence on soft tissue tension indicates that it has potential to decrease the amount of revisions due to poor hip balance, as well as decrease the time the patient has to spend on the operation table.

While reviewing literature for this research, it became apparent that there is no thorough research on the exact function of some of the hip muscles. For example, the function of the TFL is discussed in section 4.7. Anatomy textbooks such as *Gray's Anatomy* describe the functions for all of the muscles, which is often based on the location of the origin and insertion of the muscles. While the main functions of the muscles can be interpreted in this way, the joints of the body are complex systems where muscles work together for multiple functions. The hip simulator may be used as a basis for this type of research. For example, the simulator may show elongation in a muscle for a motion which is not directly related to that muscle. This function can then be studied with follow-up research.

This report contains methods to calculate the phantom muscle dimensions based on the patient's height, weight and limb length. Furthermore, the shape of the phantom muscles is designed in a way that they are scalable. The principles of the simulator could be used in a patient-specific manner, where the pelvis and femur could be physically modelled using scan data and additive manufacturing. This way, surgeons can approximate the joint geometry and soft tissue tension, predicting the influence of hip implant configurations on the soft tissue tension for specific patients. For patients with difficult joint geometry, this method could also be used to design and test custom implants.

Lastly, the potential of the simulator to reduce revision operations and reduce time spent in the operation room has health and economical benefits. Reducing revisions improves patient health and saves cost. In addition, less time spent in the operation room, as use of the simulator may reduce the need for testing trial implants *in situ*, reduces risk for the patient as well as saving cost.

5 Conclusions

In this research, many promising discoveries were made. Firstly, a model was constructed to predict the PCSA and length of hip musculature and the quadriceps, based on height and weight. Secondly, a method was devised to mimic the passive tensile properties of muscle tissue, which can be scaled to fit any muscle. These methods were then used to create a physical musculoskeletal model of the hip joint. All of these muscles can be directly 3D printed using widely available FDM technology. In addition, a method was designed to accurately measure the displacement of a flexible material, which could also have applications outside this field. This method was used to estimate the force and strain of a phantom muscle.

The end product is a physical hip simulator, equipped with strain and force sensors for the replicated muscles. The prototype was tested by orthopaedic surgeons, who were enthusiastic about its potential as a training tool in their field but also in that of physiotherapy. Furthermore, the simulator can be used to observe the differences caused by varying hip implant configurations, both numerical and by feel. A higher femoral offset causes increased tension in the muscles and increases external rotation ROM, as is the case with an *in vivo* human hip, suggesting that the hypotheses made in this study may be valid. According to specialists, the simulator has a similar joint balance to an *in vivo* hip. In all, the hip simulator is a promising tool to familiarise orthopedic surgeons with the hip joint and specifically implant stability assessment. It can be used preoperatively to assess the influence of implant configurations. Following new design iterations, the simulator could be improved to a more polished end product by implementing the recommendations made in this report. In the future, the hip simulator may decrease revision operations and reduce time spent in the operation room, to improve patient health and reduce cost.

6 Bibliography

[1] G. R. Galakatos, "Direct Anterior Total Hip Arthroplasty," *Missouri Medicine*, vol. 115, p. 537, nov 2018.

[2] J. C. Wei, B. Blaauw, D. G. Van Der Pol, M. C. Saldivar, C. F. Lai, J. Dankelman, and T. Horeman, "Design of an Affordable, Modular Implant Device for Soft Tissue Tension Assessment and Range of Motion Tracking During Total Hip Arthroplasty," *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 10, no. May, 2022.

[3] H. Tanino, T. Sato, Y. Nishida, R. Mitsutake, and H. Ito, "Hip stability after total hip arthroplasty predicted by intraoperative stability test and range of motion: A cross-sectional study," *BMC Musculoskeletal Disorders*, vol. 19, no. 1, pp. 1–7, 2018.

[4] A. Colombi, D. Schena, and C. C. Castelli, "Total hip arthroplasty planning," *EFORT Open Reviews*, vol. 4, no. 11, pp. 626–632, 2019.

[5] A. H. Shaikh, "Preoperative Planning of Total Hip Arthroplasty," in *Total Hip Replacement - An Overview*, vol. 35, pp. 249–257, InTech, nov 2018.

[6] X. Flecher, M. Ollivier, and J. N. Argenson, "Lower limb length and offset in total hip arthroplasty," *Orthopaedics and Traumatology: Surgery and Research*, vol. 102, no. 1, pp. S9–S20, 2016.

[7] M. S. Austin, W. J. Hozack, P. F. Sharkey, and R. H. Rothman, "Stability and leg length equality in total hip arthroplasty," *Journal of Arthroplasty*, vol. 18, no. 3 SUPPL. 1, pp. 88–90, 2003.

[8] M. N. Charles, R. B. Bourne, J. R. Davey, A. S. Greenwald, B. F. Morrey, and C. H. Rorabeck, "Soft-Tissue Balancing of the Hip," *The Journal of Bone Joint Surgery*, vol. 86, no. 5, pp. 1078–1088, 2004.

[9] M. Naito, K. Ogata, and I. Asayama, "Intraoperative limb length measurement in total hip arthroplasty," *International Orthopaedics*, vol. 23, no. 1, pp. 31–33, 1999.

[10] G. Bergmann, F. Graichen, and A. Rohlmann, "Hip joint loading during walking and running, measured in two patients," *Journal of Biomechanics*, vol. 26, no. 8, pp. 969–990, 1993.

[11] M. O. Heller, G. Bergmann, G. Deuretzbacher, L. Claes, N. P. Haas, and G. N. Duda, "Influence of femoral anteversion on proximal femoral loading: Measurement and simulation in four patients," *Clinical Biomechanics*, vol. 16, no. 8, pp. 644–649, 2001.

[12] G. Bergmann, A. Bender, J. Dymke, G. Duda, and P. Damm, "Standardized loads acting in hip implants," *PLoS ONE*, vol. 11, no. 5, pp. 1–23, 2016.

[13] S. L. Delp, F. C. Anderson, A. S. Arnold, P. Loan, A. Habib, C. T. John, E. Guendelman, and D. G. Thelen, "OpenSim: Open-source software to create and analyze dynamic simulations of movement," *IEEE Transactions on Biomedical Engineering*, vol. 54, no. 11, pp. 1940–1950, 2007.

[14] W. P. Zijlstra, B. De Hartog, L. N. Van Steenbergen, B. W. Scheurs, and R. G. Nelissen, "Effect of femoral head size and surgical approach on risk of revision for dislocation after total hip arthroplasty: An analysis of 166,231 procedures in the Dutch Arthroplasty Register (LROI)," *Acta Orthopaedica*, vol. 88, no. 4, pp. 395–401, 2017.

[15] U. Hedlundh, L. Ahnfelt, C. H. Hybbinette, J. Weckström, and H. Fredin, "Surgical experience related to dislocations after total hip arthroplasty," *Journal of Bone and Joint Surgery - Series B*, vol. 78, no. 2, pp. 206–209, 1996.

[16] S. Burzyński, A. Sabik, W. Witkowski, and P. Łuczkiewicz, "Influence of the femoral offset on the muscles passive resistance in total hip arthroplasty," *PLoS ONE*, vol. 16, no. 5 May, pp. 1–12, 2021.

[17] G. Lecerf, M. H. Fessy, R. Philippot, P. Massin, F. Giraud, X. Flecher, J. Girard, P. Mertl, E. Marchetti, and E. Stindel, "Femoral offset: Anatomical concept, definition, assessment, implications for preoperative templating and hip arthroplasty," *Orthopaedics and Traumatology: Surgery and Research*, vol. 95, no. 3, pp. 210–219, 2009.

[18] B. J. McGrory, B. F. Morrey, T. D. Cahalan, K. N. An, and M. E. Cabanela, "Effect of femoral offset on range of motion and abductor muscle strength after total hip arthroplasty," *Journal of Bone and Joint Surgery - Series B*, vol. 77, no. 6, pp. 865–869, 1995.

[19] T. Shoji, T. Inoue, Y. Kato, Y. Fujiwara, J. Sumii, H. Shozen, and N. Adachi, "The impact of increasing femoral offset and stem anteversion on postoperative dislocation in bipolar hemiarthroplasty," *Clinical Biomechanics*, vol. 100, p. 105770, 2022.

[20] Ansys, "Granta EduPack," 2022.

[21] L. M. Yang and V. P. Shim, "A visco-hyperelastic constitutive description of elastomeric foam," *International Journal of Impact Engineering*, vol. 30, no. 8-9, pp. 1099–1110, 2004.

[22] NinjaTek, "NinjaFlex® 3D Printing Filament: Flexible Polyurethane Material for FDM Printers," p. 1, 2016.

[23] T. Reppel and K. Weinberg, "Experimental determination of elastic and rupture properties of printed Ninjaflex," *Technische Mechanik*, vol. 38, no. 1, pp. 104–112, 2018.

[24] M. Robinson, S. Soe, R. Johnston, R. Adams, B. Hanna, R. Burek, G. McShane, R. Celeghini, M. Alves, and P. Theobald, "Mechanical characterisation of additively manufactured elastomeric structures for variable strain rate applications," *Additive Manufacturing*, vol. 27, pp. 398–407, 2019.

[25] A. Zolfagharian, L. Durran, K. Sangtarash, S. E. Ghasemi, A. Kaynak, and M. Bodaghi, *4D printing modeling via machine learning*, vol. 2. Elsevier Inc., 2022.

[26] C. Tawk and G. Alici, "Finite element modeling in the design process of 3D printed pneumatic soft actuators and sensors," *Robotics*, vol. 9, no. 9, 2020.

[27] J. Santoso, E. H. Skorina, D. Guo, Z. Kang, C. Fell, and T. L. Brooks-richards, "Aspects Regarding the Modelling and Design of 3D-printed Bending Soft Pneumatic Actuators Aspects Regarding the Modelling and Design of 3D-printed Bending Soft Pneumatic Actuators," 2022.

[28] Ultimaker, "Ultimaker TPU 95A Technical data sheet," pp. 1–3.

[29] J. Kwon, J. Ock, and N. Kim, "Mimicking the mechanical properties of aortic tissue with pattern-embedded 3d printing for a realistic phantom," *Materials*, vol. 13, no. 21, pp. 1–13, 2020.

[30] "Test Methods for Vulcanized Rubber and Thermoplastic ElastomersTension."

[31] Ansys, "ANSYS Workbench," 2022.

[32] B. I. Binder-Markey, D. Sychowski, and R. L. Lieber, "Systematic review of skeletal muscle passive mechanics experimental methodology," *Journal of Biomechanics*, vol. 129, p. 110839, dec 2021.

[33] L. S. Persad, F. Ates, L. Q. Evertz, W. J. Litchy, R. L. Lieber, K. R. Kaufman, and A. Y. Shin, "Procedures for obtaining muscle physiology parameters during a gracilis free-functioning muscle transfer in adult patients with brachial plexus injury," *Scientific Reports 2022 12:1*, vol. 12, pp. 1–16, apr 2022.

[34] T. Fukunaga, M. Miyatani, M. Tachi, M. Kouzaki, Y. Kawakami, and H. Kanehisa, "Muscle volume is a major determinant of joint torque in humans," *Acta Physiologica Scandinavica*, vol. 172, no. 4, pp. 249–255, 2001.

[35] S. R. Ward, C. M. Eng, L. H. Smallwood, and R. L. Lieber, "Are current measurements of lower extremity muscle architecture accurate?," *Clinical Orthopaedics and Related Research*, vol. 467, pp. 1074–1082, oct 2009.

[36] K. C. Parvaresh, C. Chang, A. Patel, R. L. Lieber, S. T. Ball, and S. R. Ward, "Architecture of the Short External Rotator Muscles of the Hip," *BMC Musculoskeletal Disorders*, vol. 20, pp. 1–6, dec 2019.

[37] L. S. Persad, B. I. Binder-Markey, A. Y. Shin, K. R. Kaufman, and R. L. Lieber, "In vivo human gracilis whole-muscle passive stress–sarcomere strain relationship," *Journal of Experimental Biology*, vol. 224, sep 2021.

[38] G. G. Handsfield, C. H. Meyer, J. M. Hart, M. F. Abel, and S. S. Blemker, "Relationships of 35 lower limb muscles to height and body mass quantified using MRI," *Journal of Biomechanics*, vol. 47, pp. 631–638, feb 2014.

[39] J. A. Friederich and R. A. Brand, "Muscle fiber architecture in the human lower limb," *Journal of Biomechanics*, vol. 23, no. 1, pp. 91–95, 1990.

[40] E. Ramsden, *Hall-Effect Sensors: Theory and Application*. Elsevier Science, 2011.

[41] N. van Breukelen, I. van den Enden, H. Wallenburg, and L. Wielinga, "Design Project of a Hip Simulator with Passive Muscle Behavior for Testing the Stability of a Hip Implant during Total Hip Arthroplasty," Tech. Rep. 5311624, TU Delft, 2023.

[42] B. Kim, S. B. Lee, J. Lee, S. Cho, H. Park, S. Yeom, and S. H. Park, "A comparison among Neo-Hookean model, Mooney-Rivlin model, and Ogden model for Chloroprene rubber," *International Journal of Precision Engineering and Manufacturing*, vol. 13, no. 5, pp. 759–764, 2012.

[43] N. Vidakis, M. Petousis, A. Korlos, E. Velidakis, N. Mountakis, C. Charou, and A. Myftari, "Strain rate sensitivity of polycarbonate and thermoplastic polyurethane for various 3d printing temperatures and layer heights," *Polymers*, vol. 13, no. 16, 2021.

[44] X. Zhai, E. A. Nauman, Y. Nie, H. Liao, R. J. Lycke, and W. W. Chen, "Mechanical response of human muscle at intermediate strain rates," *Journal of Biomechanical Engineering*, vol. 141, apr 2019.

[45] D. C. Taylor, J. D. Dalton, A. V. Seaber, and W. E. Garrett, "Viscoelastic properties of muscle-tendon units," *The American Journal of Sports Medicine*, vol. 18, no. 3, pp. 300–309, 1990.

[46] M. P. McHugh, S. P. Magnusson, G. W. Gleim, and J. A. Nicholas, "Viscoelastic stress relaxation in human skeletal muscle.," *Medicine and science in sports and exercise*, vol. 24, pp. 1375–82, dec 1992.

[47] E. D. Ryan, T. J. Herda, P. B. Costa, A. A. Walter, K. M. Hoge, J. R. Stout, and J. T. Cramer, "Viscoelastic creep in the human skeletal muscle-tendon unit," *European Journal of Applied Physiology*, vol. 108, no. 1, pp. 207–211, 2010.

[48] R. Drake, A. W. Vogl, and A. Mitchell, *Gray's anatomy for students*. Gray's Anatomy, London, England: Churchill Livingstone, 3 ed., 2014.

[49] E. C. Martins, C. Ruschel, E. M. Roesler, G. A. Silvano, M. P. de Castro, W. Herzog, and H. de Brito Fontana, "Tensor fascia latae and gluteal muscles myoelectric responses to increasing levels of hip medial rotation torque," *Journal of Biomechanics*, vol. 132, no. December 2021, p. 110944, 2022.

[50] M. Besomi, L. Maclachlan, R. Mellor, B. Vicenzino, and P. W. Hodges, "Tensor Fascia Latae Muscle Structure and Activation in Individuals With Lower Limb Musculoskeletal Conditions: A Systematic Review and Meta-Analysis," *Sports Medicine*, vol. 50, no. 5, pp. 965–985, 2020.

[51] S. R. Ward, T. M. Winters, S. M. O'Connor, and R. L. Lieber, "Non-linear Scaling of Passive Mechanical Properties in Fibers, Bundles, Fascicles and Whole Rabbit Muscles," *Frontiers in Physiology*, vol. 11, no. March, pp. 1–9, 2020.

[52] L. G. Prado, I. Makarenko, C. Andresen, M. Krüger, C. A. Opitz, and W. A. Linke, "Isoform Diversity of Giant Proteins in Relation to Passive and Active Contractile Properties of Rabbit Skeletal Muscles," *The Journal of General Physiology*, vol. 126, no. 5, p. 461, 2005.

[53] R. Herbert, "The Passive Mechanical Properties of Muscle and Their Adaptations to Altered Patterns of Use," *Australian Journal of Physiotherapy*, vol. 34, no. 3, pp. 141–149, 1988.

[54] C. Y. Tang, G. Zhang, and C. P. Tsui, "A 3D skeletal muscle model coupled with active contraction of muscle fibres and hyperelastic behaviour," *Journal of Biomechanics*, vol. 42, no. 7, pp. 865–872, 2009.

[55] Y. T. Lu, H. X. Zhu, S. Richmond, and J. Middleton, "A visco-hyperelastic model for skeletal muscle tissue under high strain rates," *Journal of Biomechanics*, vol. 43, no. 13, pp. 2629–2632, 2010.

[56] Y. Tang, S. Liu, Y. Deng, Y. Zhang, L. Yin, and W. Zheng, "An improved method for soft tissue modeling," *Biomedical Signal Processing and Control*, vol. 65, no. November 2020, p. 102367, 2021.

[57] Y. C. Fung, *Biomechanics: Mechanical Properties of Living Tissues*. Springer New York, 1981.

[58] R. W. Ogden, *Non-linear Elastic Deformations*. Dover Civil and Mechanical Engineering, Dover Publications, 1997.

[59] B. S. Myers, C. T. Woolley, T. L. Slotter, W. E. Garrett, and T. M. Best, "The influence of strain rate on the passive and stimulated engineering stress–large strain behavior of the rabbit tibialis anterior muscle," *Journal of biomechanical engineering*, vol. 120, no. 1, pp. 126–132, 1998.

[60] K. Wang, C. Wu, Z. Qian, C. Zhang, B. Wang, and M. A. Vannan, "Dual-material 3D printed metamaterials with tunable mechanical properties for patient-specific tissue-mimicking phantoms," *Additive Manufacturing*, vol. 12, pp. 31–37, 2016.

[61] R. Rivlin, "Large elastic deformations of isotropic materials IV. further developments of the general theory," *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, vol. 241, pp. 379–397, oct 1948.

[62] Y. Zhang, K. Yu, K. H. Lee, K. Li, H. Du, and Q. Wang, "Mechanics of stretchy elastomer lattices," *Journal of the Mechanics and Physics of Solids*, vol. 159, no. January, p. 104782, 2022.

[63] V. L. Popov, "Contact mechanics and friction: Physical principles and applications," *Contact Mechanics and Friction: Physical Principles and Applications*, pp. 1–362, 2010.

[64] R. Peck, C. Olsen, and J. Devore, *Introduction to Statistics and Data Analysis*. Cengage Learning, 2011.

A Theory

A.1 Passive muscle stiffness

Passive muscle stiffness has been researched for some time [53] as well as more recently using hyperelastic models and validated using muscle from other species [54][55], and human soft tissue [56]. Furthermore, hyperelastic models have been used to fit experimental stress-strain characteristics from ex vivo muscle [44].

About how passive characteristics scale with muscle architecture, much is still unknown [35]. An extensive systematic review paper by *Binder-Markey et al.* concluded that species, size, and type of stretch have a significant effect on passive muscle stiffness, while muscle region does not seem to be a significant factor [32]. The paper further elucidates that a detailed understanding of passive muscle stiffness is still unclear. Despite this, it was found that one of the hip muscles, the gracilis muscle, shows only a modest increase in modulus when going from single fibre to whole muscle [33]. How this scaling problem is approached in this research, is explained in section 2.3.

A.2 Muscle architecture scaling

Similar to muscle passive stiffness, studies have also researched how muscle architecture scales depending on the subject's anatomical properties such as height and mass [38]. For a pool of 24 healthy, mostly young subjects of both sexes, *Handsfield et al.* discovered correlations for both muscle volume and muscle length, depending on the height-mass product and the subject's limb length respectively.

It was found that the total lower extremity muscle volume can be estimated from the mass-height product with a coefficient of determination (R^2) of 0.92. This relation is shown in equation 6.

$$V_{m,tot} = 47mh + 1285 \tag{6}$$

where $V_{m,tot}$ is the total lower extremity muscle volume, m is the subject's body mass and h is the subject's height. In addition, the study shows that the muscle volume fractions (fraction of a specific muscle relative to the total muscle volume) are conserved for this population. Individual muscle volume thus scales linearly with total muscle volume. All volume correlations are significant ($p < 0.05$), excluding the obturator externus.

The same paper also reports an estimation of individual muscle belly length on the subject's limb length. These length correlations are significant ($p < 0.05$), except for the gemelli, obturator muscles, quadratus femoris, piriformis and pectineus.

When the muscle's architectural properties such as the muscle's volume and belly length are known, the individual muscle's PCSA can also be calculated. *Handsfield et al.* also explain how one can do this. The corresponding formula is shown in equation 7.

$$PCSA = \frac{V_m}{l_m f_f} \tag{7}$$

where V_m is the individual muscle volume, l_m is the individual muscle belly length and f_f is the muscle fibre length divided by the muscle belly length. The optimal fibre/belly length ratio f_f was taken from *Ward et al.*, who obtained this data by analysing cadavers.

A.3 Hyperelastic materials

Muscle, like most soft tissues, is a material which exhibits hyperelastic behaviour [57]. When deformed, hyperelastic materials show a highly non-linear stress-strain relationship [58]. Hyperelastic materials can strain up to as high as 700%. Furthermore, they are almost fully incompressible. A typical stress-strain curve of a hyperelastic material is shown in supplementary figure 1b. When elongating a hyperelastic material, the material's initial response is nonlinear before the material softens. Subsequently, the material reaches a linear state. Then, the material stiffens due to the untwisting of cross-linked polymer chains. After all the polymer chains are fully aligned, another linear region follows before the material softens again due to failure.

The stress-strain response of a muscle specimen is shown in supplementary figure 1a [44]. As can be seen in the figure, the muscle's response is strongly dependent on strain rate. This behaviour has been observed before, and is typical for skeletal muscle, as well as other hyperelastic materials [59]. In skeletal muscle, strain-stiffening can also be observed, which can be attributed to the untwisting and alignment of muscle fibres under tension.

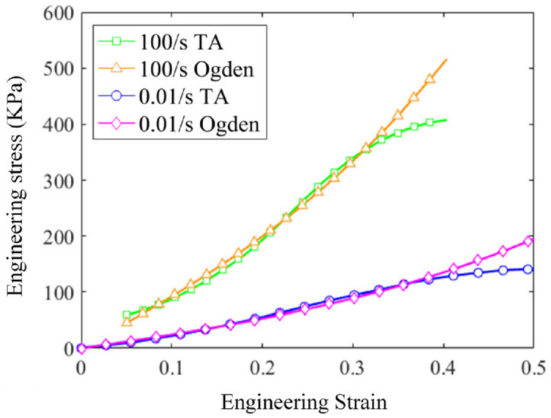
A.4 Hyperelastic models

Since hyperelastic materials have a non-linear stress-strain relation, they cannot be modelled with a constant such as Young's modulus, as is the case with most metals. Instead, the stress-strain response needs to be derived from the strain energy density function. Multiple models have been developed to achieve this. These include the Neo-Hookean model, the Mooney-Rivlin model, and the Ogden model, among others [42]. Each of these has its own strain energy density function, while also being useful for different applications.

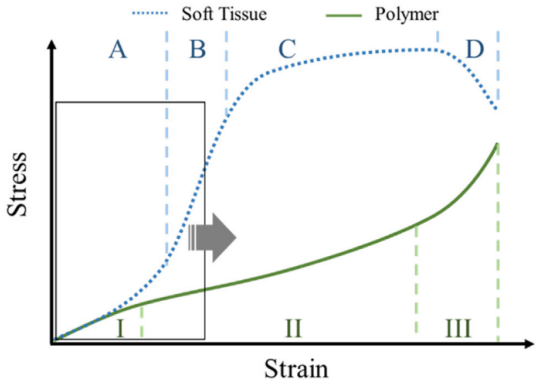
The Mooney-Rivlin model has been widely used to accurately model the mechanical properties of hyperelastic materials. The strain energy density function W of the Mooney-Rivlin model is a linear combination of the two invariants of the left Cauchy-Green deformation tensor [42]. The strain energy density function for an incompressible material is shown in equation 8.

$$W = C_1(\overline{I}_1 - 3) + C_2(\overline{I}_2 - 3) \tag{8}$$

where C_1 and C_2 are empirically obtained material constants and I_1 and I_2 are the first and second invari-



(a)



(b)

Supplementary figure 1: (a) Mechanical response of an *ex vivo* human specimen during elongation, parallel to fibre direction. Fitted with the Ogden hyperelastic model. Taken from *Zhai et al.* [44]. (b) Difference in soft tissue hyperelasticity compared to polymers. Soft tissue: A - toe region, B - elastic region, C - plastic region, D - failure region. Polymer: I - strain-softening region, II - linear elastic region, III - strain-stiffening region. Taken from *Wang et al.* [60].

ants of the deviatoric component of the left Cauchy-Green deformation tensor.

Although the Mooney-Rivlin is not as accurate as the Ogden model for strains closer to material failure, the Mooney-Rivlin model has been proven accurate for strains up to 200% [61]. Furthermore, it is not as computationally demanding as the Ogden model.

A.5 Lattice structures

One of the largest challenges of this design is to mimic the muscles' strain-stiffening behaviour at lower strains. As was elaborated in section A.3, muscles show this behaviour at much lower strains than most elastomeric materials do. This difference is shown in supplementary figure 1b. Specific structures have been designed to tackle this issue. *Wang et al.* designed a soft tissue mimicking phantom, consisting of a wavy stiff elastomer, enclosed by a compliant elastomer [60]. This structure was 3D-printed using SLA technology. The structure allowed them to achieve strain-stiffening behaviour at a strain range of 0%-8%, which is theoretically impossible to achieve when using a single elastomeric material.

The idea mentioned by *Kwon et al.*, however, is complex and requires expensive SLA printing technology. For this research, it is therefore preferred to look at more inexpensive methods. *Zhang et al.* researched the possibilities of stretchy elastomer lattices [62]. It was hypothesised that all lattice structures (bending- or stretching-dominated) are stretching-dominated if subjected to a large stretch.

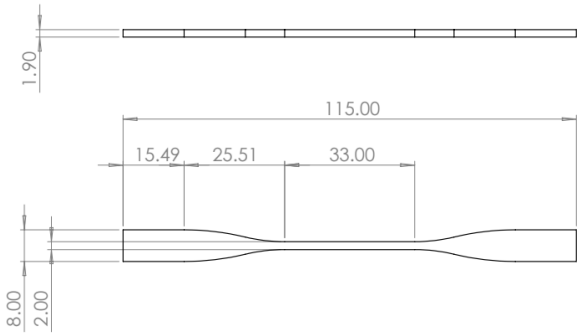
The idea of using lattice structures to tune the strain-stiffening behaviour can also be utilised for the goal of replicating soft tissue's mechanical properties. The idea is that when using a lattice structure, the structure will be mainly loaded in bending. If a material is used which is relatively compliant in bending in comparison to elongation, it is hypothesised that the stiffness will increase at higher stretches. When the material is sufficiently stretched, the lattice's beams will be close to parallel, increasing the stiffness. Furthermore, they investigated the mechanical behaviour of hierarchical lattice structures. As a result, they were able to successfully tweak the strain-stiffening effect, finding that increasing the hierarchical order of the lattice structures makes the steepening of the response curve appear more gradual. Moreover, it was found that strain-stiffening can be achieved at stretch ratios below two using octahedral lattice shapes. At low-volume fraction scenarios, the bending-induced stress was negligible compared to the elongation-induced stress as the stretch increased, resulting in a heavily steepening curve. Despite this, increasing the volume fraction resulted in a more gradual steepening effect.

3D-printed thermoplastic polyurethane (TPU) lattices have been researched in the past to achieve a similar goal [29]. *Kwon et al.* attempted to recreate the compressive properties of the aortic wall. They were able to successfully change the strain at which stiffening occurs. However, they did not succeed in manipulating the material's properties in such a way that strain-stiffening occurs at strains lower twice the original length. It is worth noting that the lattice structures were only two-dimensional, and therefore it is hypothesised that better results can be achieved when a more complex structure is tested.

B Methods

B.1 Lattice structure design

The desired forces are known and the material to be used has been characterised. Before the design of the muscle phantoms can commence, one first needs to determine what the mechanical response of the muscle should look like. As can be seen in figure 2, there is high variability present in the mechanical response of the gracilis muscle. As a result, the data cannot be fitted with a function which results in a high coefficient of determination R^2.



Supplementary figure 2: Test sample design based on ASTM D412-C with dimensions in millimeters, used for the characterisation of the TPU filament [30].

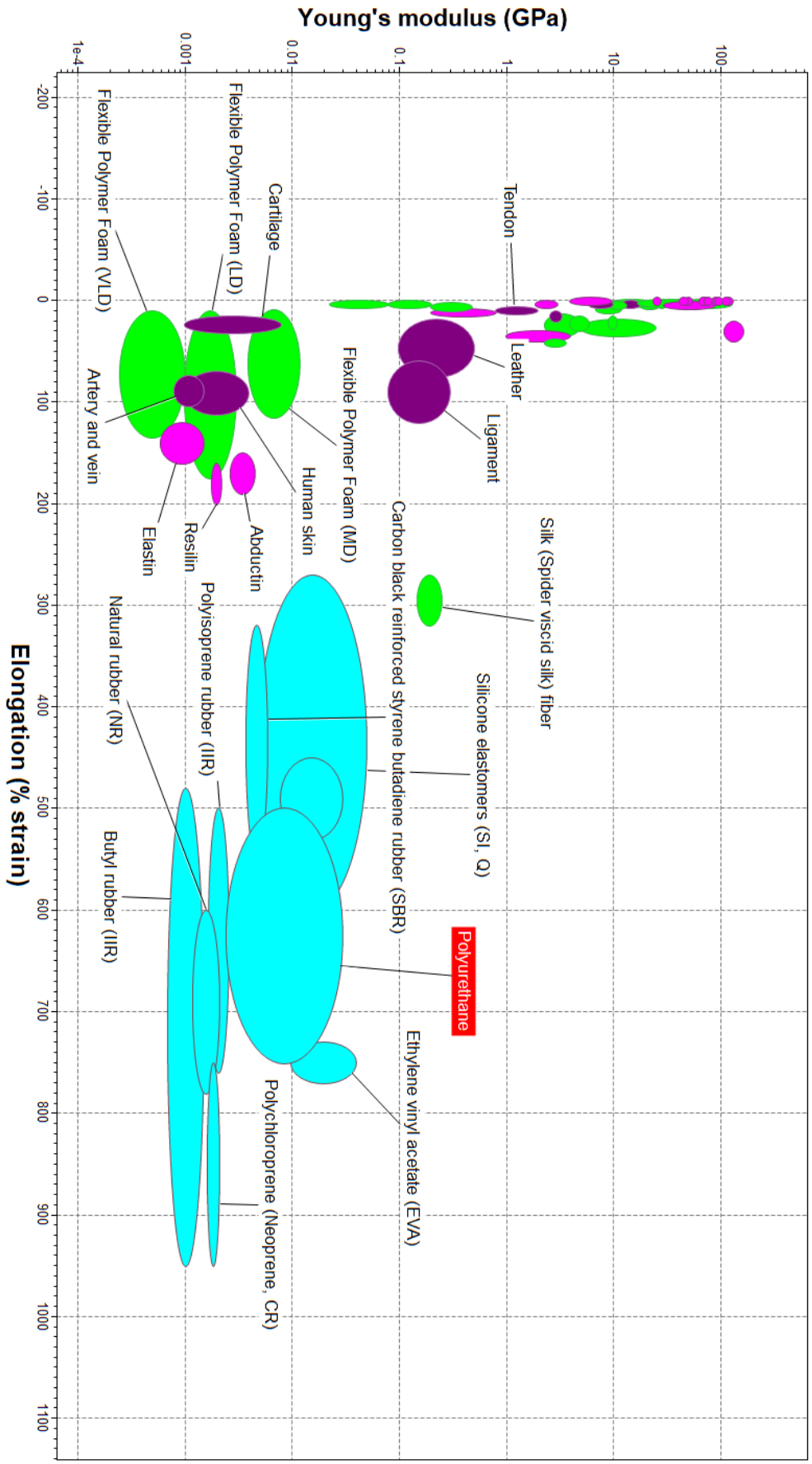
This report aims to achieve a mechanical response which mimics the response experienced by a surgeon when performing a series of ROM tests. This means that when the muscle is approaching maximum lengthening, the stress response needs to increase significantly. As is explained in section 2.3, the muscles are assumed to stretch up to 30%. Therefore, stiffening needs to occur shortly before that, at $\tilde{2}5\%$. Inspired by the literature mentioned in appendix B.1, the possibility of recreating the passive mechanical properties of muscle using 3D-printed lattice structures is researched.

Supplementary figure 3: 3D-printed TPU 95A sample with octahedronal lattice structure.

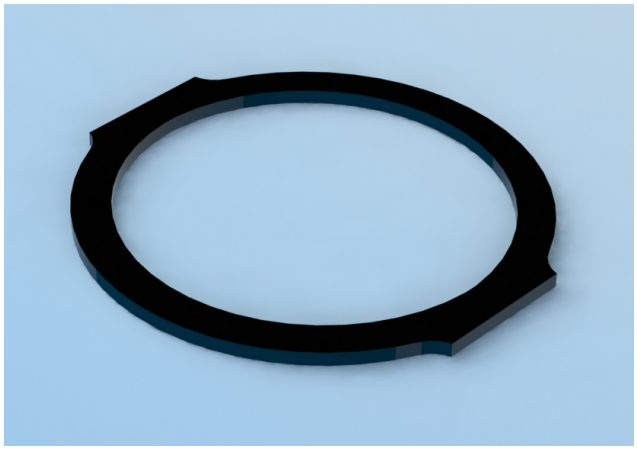
To test the feasibility of lattice structures using Ultimaker TPU 95A, a simple lattice structure was printed. This pattern is shown in supplementary figure 3. When using FDM printers, it is generally not recommended to print circular, unsupported profiles at diameters less than 3 mm. Consequently, the minimum thickness of one beam in the lattice is 3 mm. However, when extrapolating the data reported by tensile testing by *Kwon et al.*, one finds that elongating a beam with a diameter of 3 mm will already result in a force of roughly 42N. Based on an extrapolation using the *in vivo* passive muscle stiffness data reported by *Persad et al.* (see section 2.3 for more detail), it is concluded that creating FDM-printed lattice structures is not feasible using this material. This is a result of the material not being compliant enough. Correspondingly, a different approach was chosen, which is elaborated further in section B.2.

B.2 Hourglass shape design

It is hypothesised that strain-stiffening can also be achieved by utilising a flat, circular profile, as shown in supplementary figure 5.

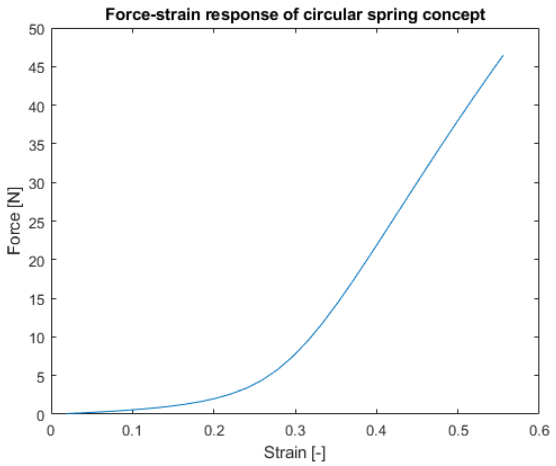


Supplementary figure 4: Material selection using GRANTA EduPack [20]. Four material families are shown: Molecular building blocks (lilac), soft tissue (dark purple), polymer foams (green), and elastomers (blue). The selected material, polyurethane, is highlighted.



Supplementary figure 5: 3D-render of circular spring concept. A force is applied perpendicular to both square surfaces to create a force with increasing stiffness modulus.

As the ring is pulled apart at the thickened edges, the bending-dominated stress transforms to tensile stress, increasing the total stress significantly. A SOLIDWORKS render of this concept is shown in supplementary figure 5. This concept is tested in ANSYS using the previously obtained Mooney-Rivlin material constants (see section 2.2). Using finite element analysis (FEA), a simulated tensile test is carried out using an input displacement. The result is shown in supplementary figure 6.



Supplementary figure 6: Force-strain response curve of circular spring concept. Obtained using FEA in ANSYS.

As can be seen in supplementary figure 6, this shape can theoretically be used to achieve the goal of stiffening the material at a specific strain. However, this is not a feasible shape to use in a hip simulator. There are more than twenty lower extremity muscles that need to be modelled, and a circular shape takes up as much width as it does length. Consequently, a more spatially efficient shape needs to be designed.

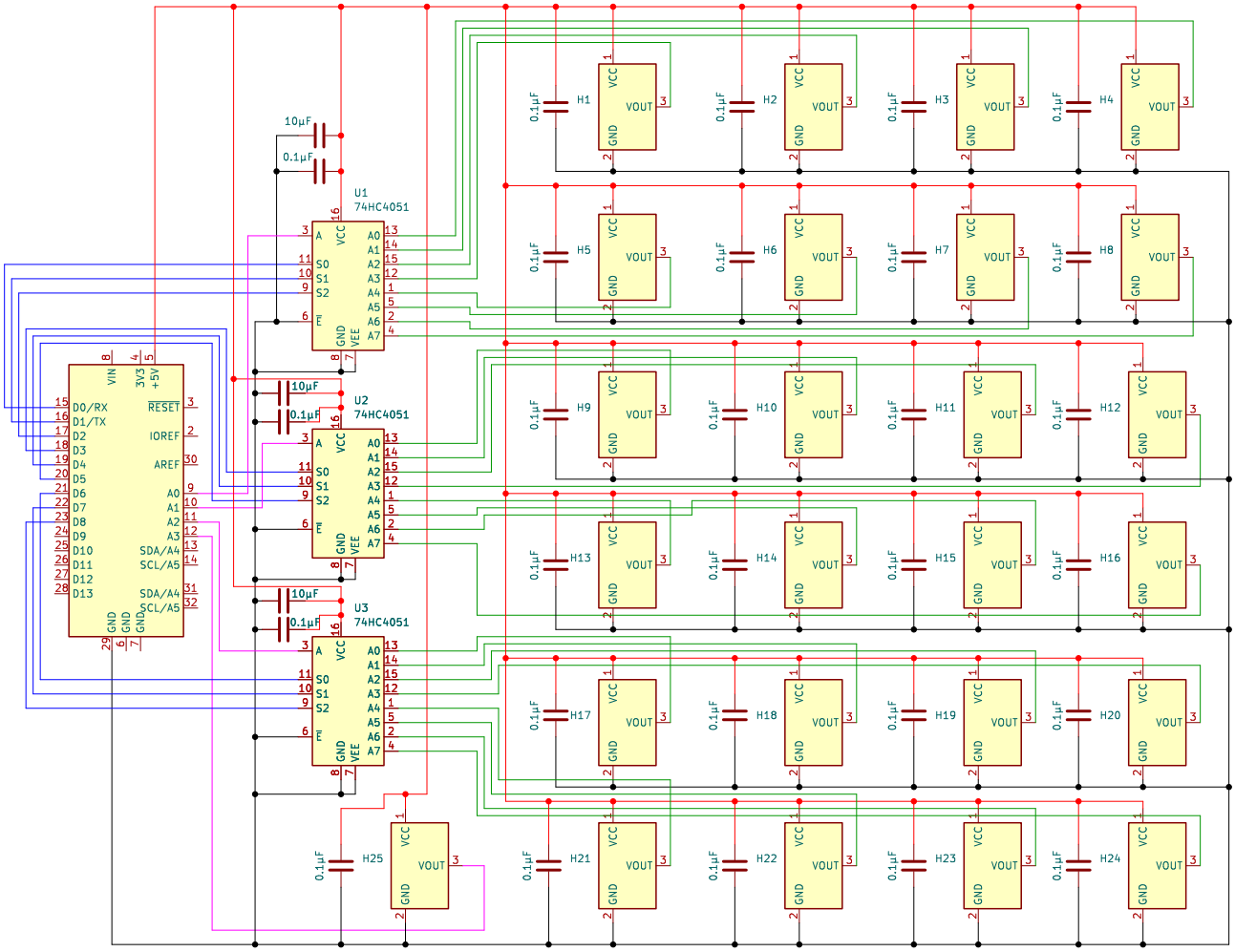
A circular shape is not the only way of transforming bending-dominated stress into tensile-dominated stress. It is hypothesised that a similar result can be achieved while using an S-shape. An optimisation process is carried out to create an appropriate S-shaped spring. Since the mechanical response of both the material and the shapes are highly non-linear, this optimisation process follows a trial-and-error approach.

The shape was found by applying a few constraints:

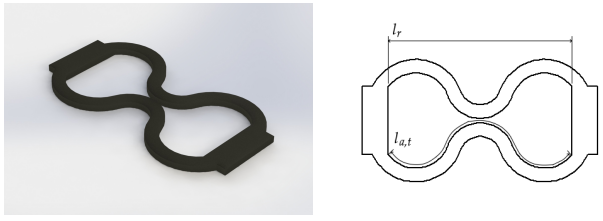
- The stiffness needs to increase significantly at $\sim$25% stretch
- The shape should be a symmetrical pattern across the longitudinal and transversal axis so that it can be repeated to adjust to the muscle belly length
- The arcs need to be tangent with respect to each other, so that there are no sharp angles which increase initial stiffness

To ensure a significant stiffness increase, the initial stiffness of the muscle needs to be kept to a minimum. Therefore, the amount of arcs needs to be as low as possible. For a symmetrical shape, however, an odd amount of arcs is required. The choice is made to implement three arcs on each side. This is the optimal choice since a higher amount of arcs would result in a higher initial stiffness. The reference length l_r, which is defined as the length between the two opposing rectangular surfaces inside the spring shape, is kept constant. The dimensions of the clamped ends are also kept constant.

It is hypothesised that the total arc length $l_{a,t}$ will determine which stretch the stiffness will increase. While retaining some space between the spring leaves, the total arc length $l_{a,t}$ is changed until a satisfactory force-strain response is obtained from simulating tensile tests using FEA in ANSYS. These tests were conducted using a strain rate of 0.45 s^{-1}, elongating up to 50% strain. Changing the total arc length while the central arcs' midpoints are kept in line with the transversal centerline of the muscle, means that the arcs' radius, angle and centre position are the affected properties. As was done in section 2.2, an input displacement is used to simulate a tensile testing machine. The obtained hourglass-shaped spring design is shown in supplementary figure 7.



Supplementary figure 8: Schematic drawing of the circuit used to measure the muscles' displacement. Red wires represent wires connected to the Arduino's supply voltage pin, while black wires are connected to the ground. Blue represents the wires used to select the pin on the multiplexers, green wires are connected to the sensors' output voltage pins and purple wires are connected to the Arduino's analogue pins.



(a) (b)

Supplementary figure 7: a) SOLIDWORKS render of the hourglass-shaped spring. b) Drawing of the hourglass-shaped spring. The reference length l_r and the total arc length $l_{a,t}$ are indicated in the figure.

B.3 Material property testing

Some further tests need to be performed to fully characterise the TPU's mechanical behaviour in the relevant conditions. Hyperelastic materials are known to exhibit visco-elastic properties, such as stress-relaxation and creep [63]. To completely analyse the performance of the hip-simulator, all of these potentially impactful properties are investigated. These properties are as follows:

- Shelf life
- Material hysteresis
- Strain rate dependence
- Stress relaxation
- Creep

Firstly, a series of tensile tests is performed using a Zwick tensile tester. The testing settings are shown in supplementary table 1. Schematic drawings of each sample can be found in supplementary figure 12. The first test is meant to assess the shelf life of the TPU material, as well as establish a possible difference in print quality. The 'old' samples mentioned in supplementary table 1 are the same ones as those used in the testing session described in section 2.4. The time passed between these two sessions is approximately five months. After that, samples fastened using the T-fit

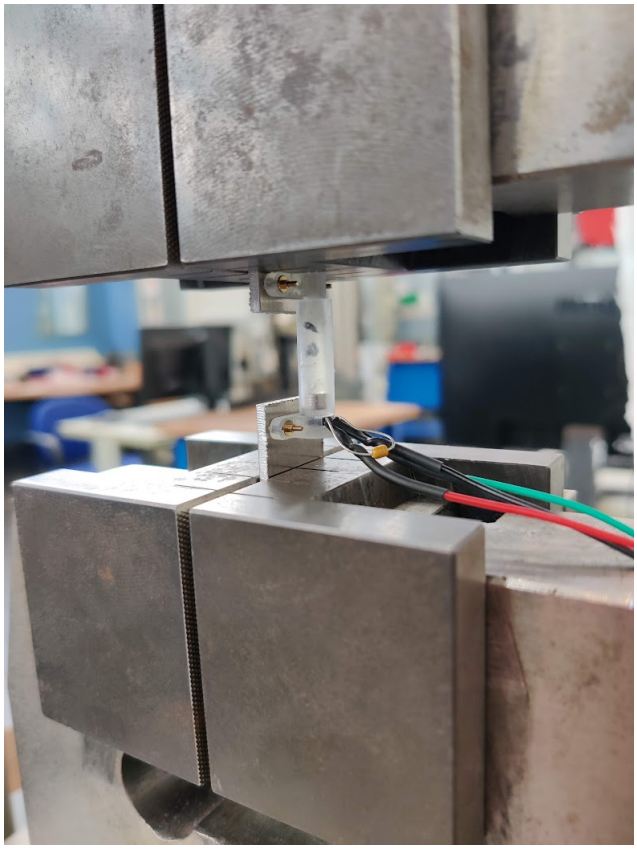
method are tested in loading and unloading to analyse material hysteresis. The samples are also extended at slower speeds to determine the influence of strain-rate dependence. In the next test, the samples are extended, held in place for 30 seconds and unloaded, to analyse the influence of stress relaxation. Creep is also researched by loading the sample with a force of 10 N and measuring the length for 10 minutes.

B.4 Sensor testing

To obtain a better understanding of what factors could influence the performance of the simulator, some additional testing is conducted. These include:

- Precision of Hall effect sensors
- Influence of attachment type (T-fit or clamp)
- Influence of sensor mount location
- Influence on sensor output when a muscle is curved around a circular surface
- Agreement between strain and force measurement method and tensile tester.

The sensor sensitivity is assessed by mounting the two sensor mount parts shown in figure 4a to two aluminium plates. Each plate is clamped by using the grips of the Zwick tensile tester. The magnet was inserted into the tube with the Hall effect sensor, after which the magnet and sensor were separated (see supplementary figure 9). The settings can be found in supplementary table 1. During this test, the output voltage of the Hall effect sensor is measured by the *Arduino*.



Supplementary figure 9: Sensor accuracy test in the Zwick tensile tester.

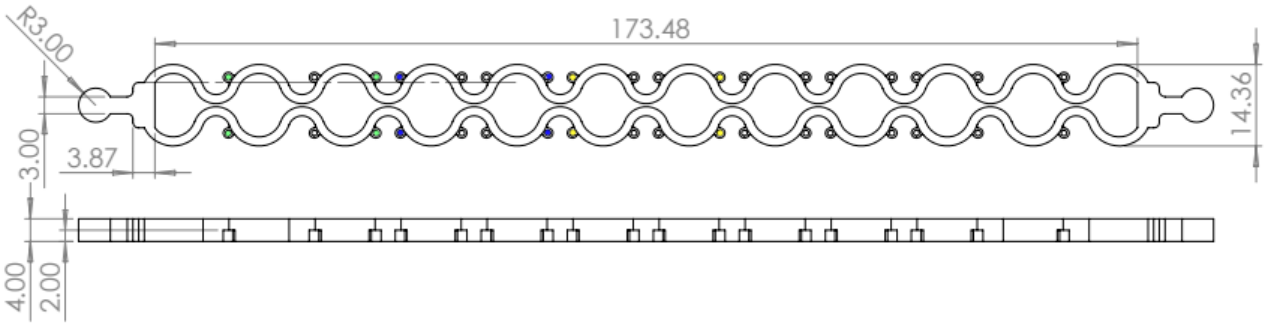
The influence of the attachment type is also tested on the Zwick tensile tester. The data of the clamped sample, described in section 2.4, is compared to that of the loading phase of the T-fit (see supplementary table 1 at a crosshead speed of 9.72 mm/s.

The design of the 3D-printed muscles allows for the placement of the sensor mounts along its entire length. This decision was made for practical purposes. However, there is a possibility that placing the sensor closer to the edge of the muscle instead of in the middle could influence the output voltage of every sensor. Therefore, the sensor is placed in three positions, specifically at the edge of a sample muscle, at roughly a third of the length of the muscle, and in the middle (see figure 10). This test is performed 10 times for each of the three samples. The testing settings can be found in supplementary table 1.

The data from the middle position was also used to assess the agreement between the measurement method using Hall effect sensors and the gold standard, namely the tensile tester. The data is fitted using the same method as described in section 2.8. Then, the force and strain are calculated using the obtained curve fits. The measured force and strain using the Hall effect sensors are compared to the force and strain measured using the tensile tester, using a Bland-Altman plot. A non-parametric statistical analysis was performed. The reproducibility coefficient (RPC) is calculated as

Supplementary table 1: Testing settings for the third tensile test.

Sample type	*Crosshead speed [mm/s]*	*Elongation [mm]*	*Repetitions*	*Notes*
Clamped (old)	9.72	10.8	5	
Clamped (fresh)	9.72	10.8	5	
T-fit	9.72	10.8	5	Loading and unloading
T-fit	4.32	10.8	5	
T-fit	0.972	10.8	5	
T-fit	9.72	10.8	5	Loading, hold for 30 seconds, unloading
T-fit	9.72	10.8	5	Hold for 10 minutes
Plates with sensor mounts	1.00	40.0	10	
T-fit	17.348	86.74	10	Three specimens with sensor mounted in three positions



Supplementary figure 10: Top and side view of sample muscle used for testing the sensor location influence and the curved surface influence. The different colours indicate the positions in which the sensor was mounted to test the sensor location influence. Only the yellow position was used for the curved surface test.

follows:

$$RPC = 1.45IQR \tag{9}$$

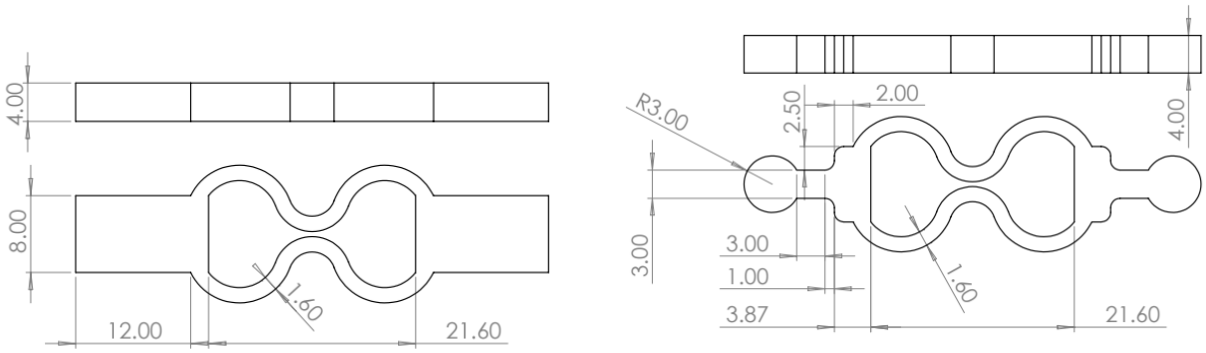
where IQR is the interquartile range [64].

It is suspected that the output voltage of each Hall effect sensor can be affected when the muscle is curved around a surface (such as the pelvis or parts of the femur). To simulate this condition, a sample muscle is curved around a circular surface. The muscle is then loaded with weights using increments of 500 g, up to a total load of 6 kg. The sensor is mounted in the middle of the muscle, but the muscle is moved for each condition. These conditions are the following:

1. The sensor is mounted between the fixture and the circular profile
2. The sensor is mounted on top of the circular profile
3. The sensor is mounted between the circular profile and the applied load

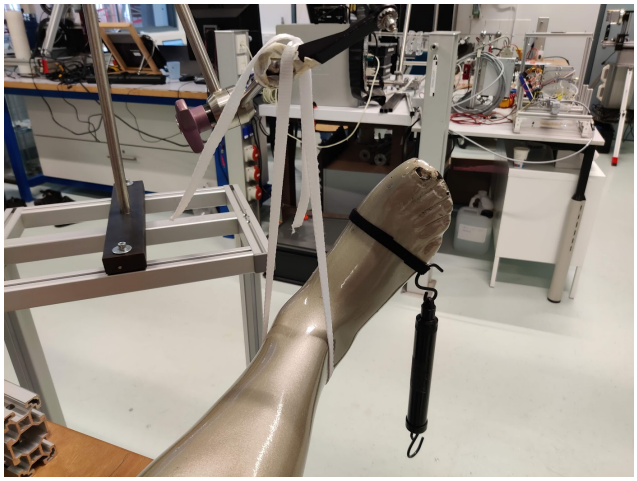
Supplementary figure 11: Setup of the curved surface test.

To reduce viscoelastic effects, the muscle is stretched using the maximum load for a minute before initiating the test. This test is repeated using three different samples, with the same dimensions as those used to test the sensor location influence (see supplementary figure 10). The test setup can be seen in supplementary supplementary figure 11.



(a) (b)

Supplementary figure 12: Schematic drawings of TPU samples used for tensile testing. Top and side views are shown. (a) Clamped TPU sample (b) Sample with T-fit.



Supplementary figure 13: Ankle is held in place during the additional stability testing.

C Results

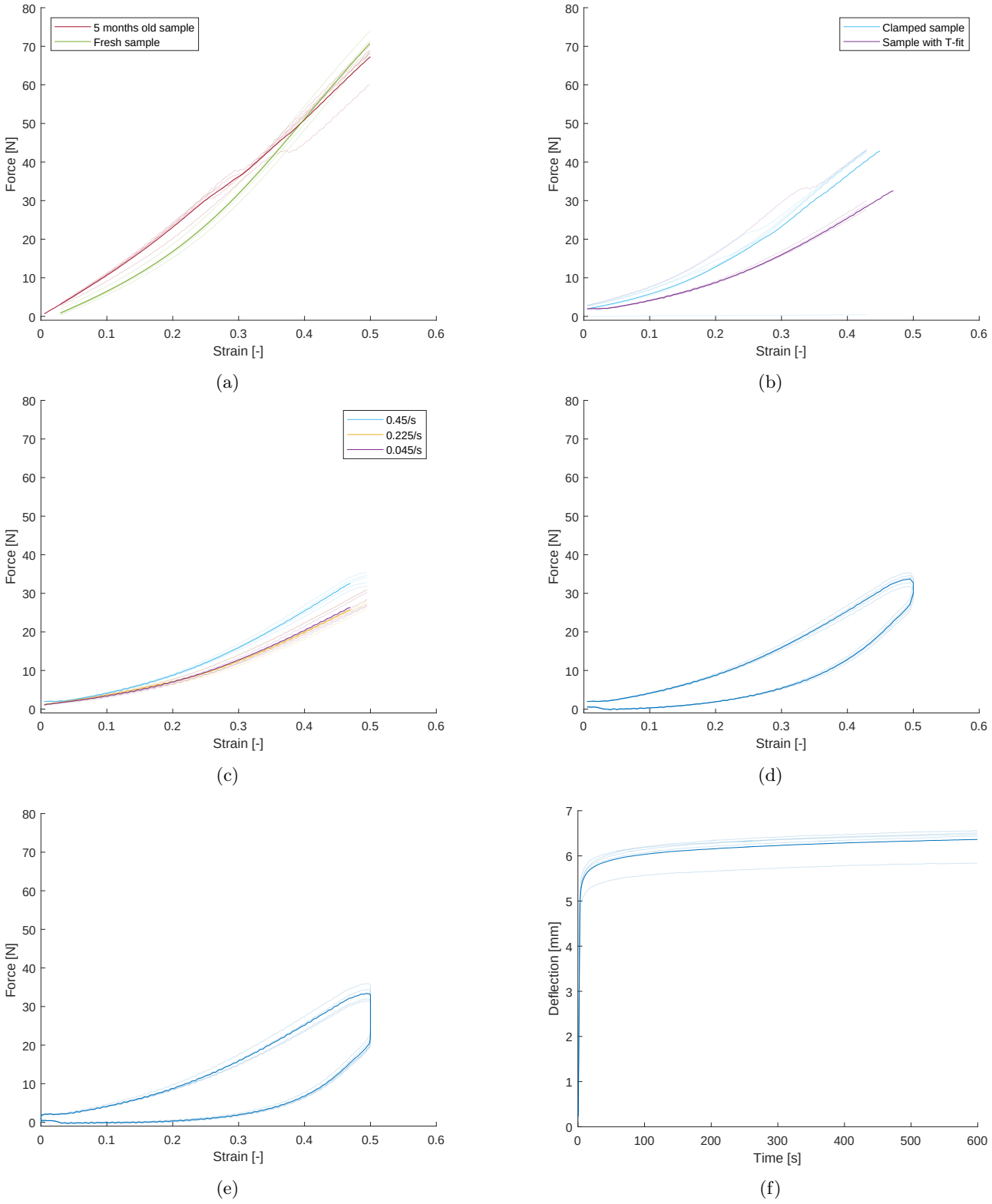
C.1 Material and shape property testing

In section B.3, a testing method was described to assess multiple material and shape properties, which could potentially have an impact on the performance of the simulator. The resulting plots are shown in supplementary figure 14. After five months of shelf life, the sample showed a significantly different characteristic ($P < 0.0001$), as can be seen in supplementary figure 14a. In supplementary figure 12b, one can observe a stiffer ($P < 0.0001$) characteristic with a clamped sample than is the case with one attached using a T-fit. The strain-rate dependence test also shows statistically different results ($P < 0.0001$) for the comparison of each condition, as can be seen in supplementary figure 14d. The stress-relaxation present in supplementary figure 14d can also be observed in supplementary figure 14e. Lastly, material and shape are prone to creep, as can be seen in supplementary figure 14f.

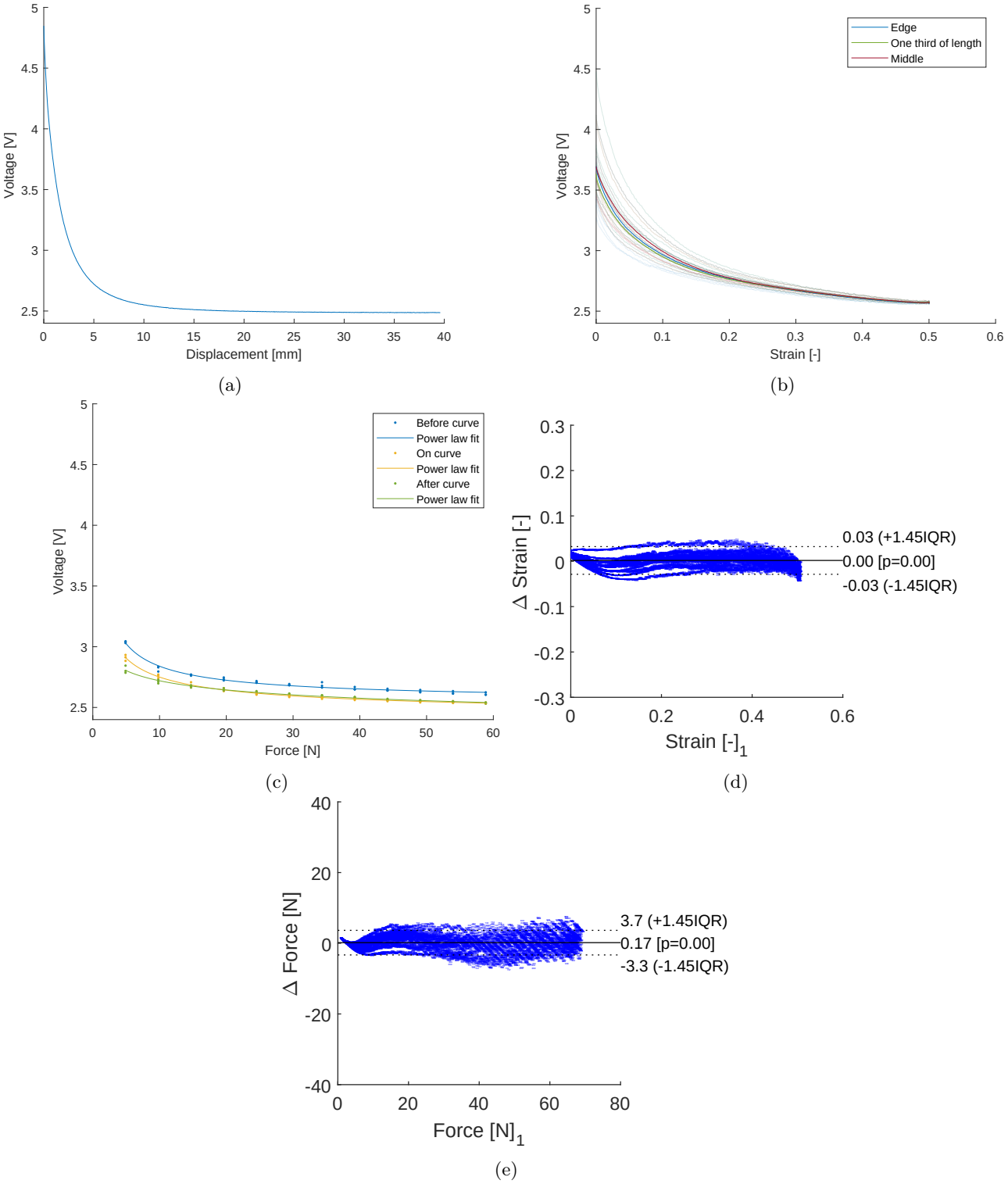
C.2 Sensor testing

The results of the sensor accuracy test, the sensor location test the curved surface test, and the measurent method comparison are shown in supplementary figure 15. The range in which the sensor can detect changes can be seen in supplementary figure 15a. As can be seen in the figure, the Hall effect sensors can detect changes up to roughly 11 mm. The influence of the location where the sensors are placed on the muscle can be seen in supplementary figure 15b. All conditions are statistically different from one another ($P < 0.0001$). The results of the curved surface test are shown in supplementary figure 15c. All conditions are statistically different ($P < 0.0001$), except for the condition where the sensor was mounted on the curved surface versus the condition where the sensor was mounted after the curve ($P = 0.4978$).

The Bland-Altman plots for the strain and force measurement methods are shown in supplementary figures 15d and 15e respectively. The non-parametric analysis resulted in $RPC_\varepsilon = 0.0305$ and $RPC_F = 3.48$ N.



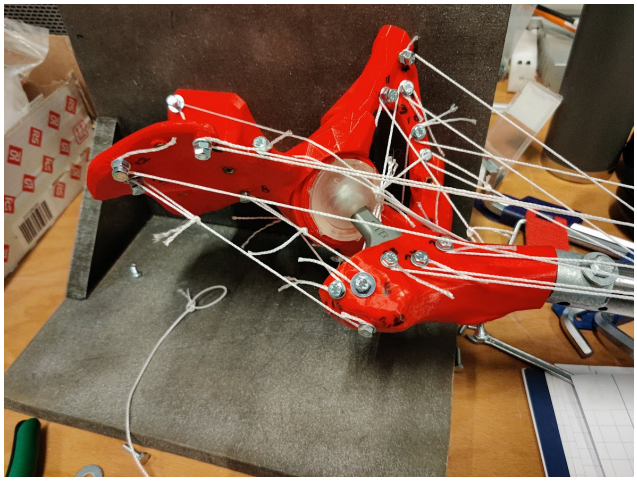
Supplementary figure 14: Plots showing the results of different tests on the influence of shape and material properties on the muscles' force-strain characteristic curves. The more transparent lines indicate the individual datasets, while the less transparent lines indicate the mean. (a) Force-strain characteristic of old samples versus freshly printed samples. (b) Influence of attachment method on force-strain characteristic curve of TPU muscle sample. (c) TPU muscle samples, connected with T-fit and tested at different strain rates. (d) Hysteresis loop of TPU muscle sample. (e) Stress relaxation of TPU muscle sample, secured with T-fit. (f) Creep of TPU muscle sample, secured with T-fit.



Supplementary figure 15: The more transparent lines indicate the individual datasets, while the less transparent lines indicate the mean. (a) Results of sensor accuracy test. (b) Results of sensor location influence test. (c) Results of curved surface test. Each set of data is fitted using a power law. (d) Bland-Altman plot of strain measurement method using Hall effect sensors in reference to tensile tester data (e) Bland-Altman plot of force measurement method using Hall effect sensors in reference to tensile tester data.

C.3 Photos

Supplementary figure 16: Tensile test of 3D-printed samples using a Lloyd tensile testing machine.



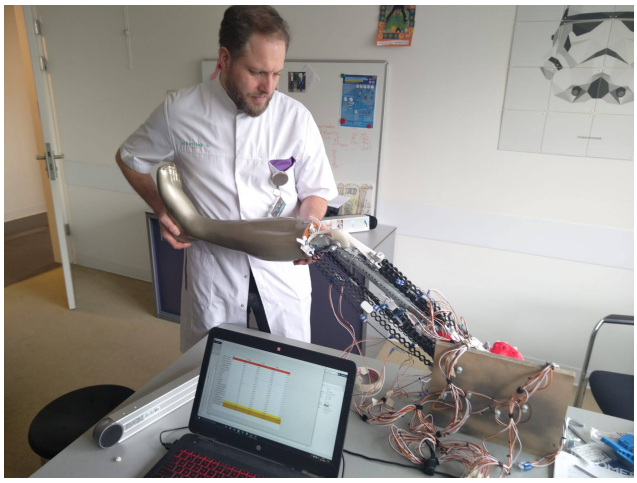
Supplementary figure 17: Side view of the hip joint part of the simulator, with ropes attached to measure the required MTU lengths.

(a)

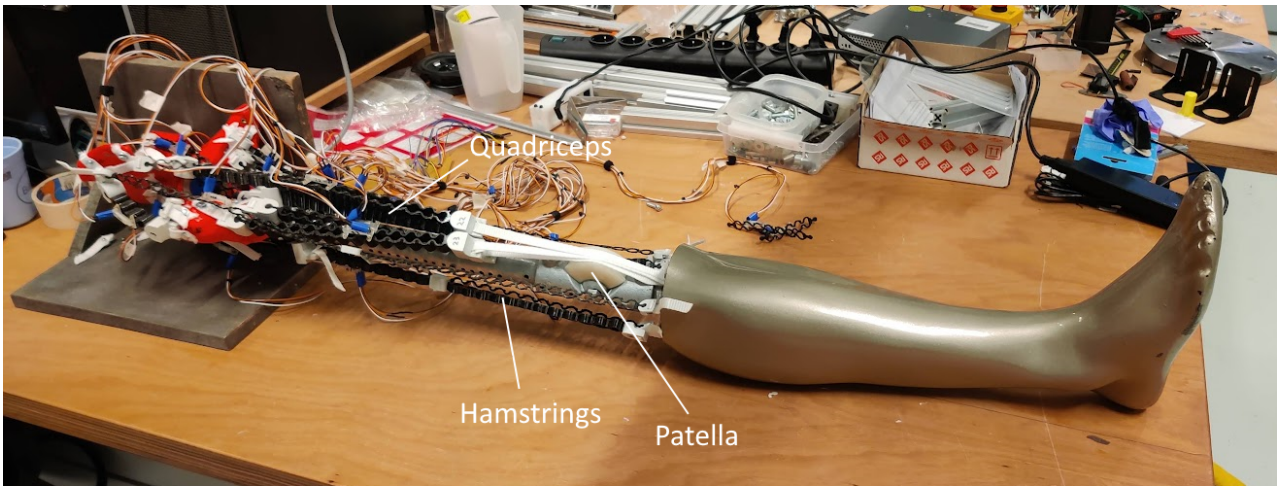
(b)

(c)

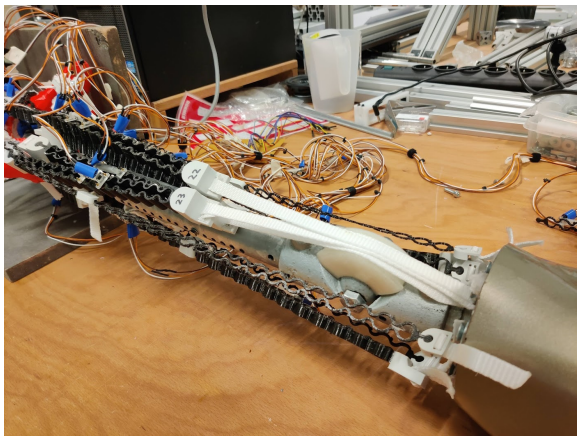
Supplementary figure 19: (a) Trial necks used for the implant configurations. 1. STD neck 2. KHO neck 3. KLA-125 neck. Only necks 1 and 2 were used for the experiment with the surgeons (b) *STD* neck with a +9 offset trial head. (c) *STD* neck with a +1 offset trial head.



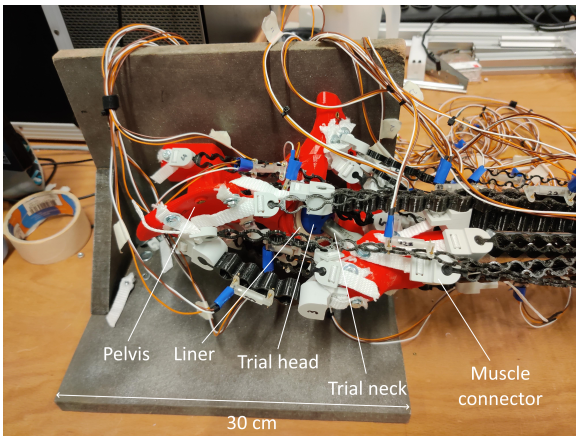
Supplementary figure 20: One of the surgeons testing the simulator.



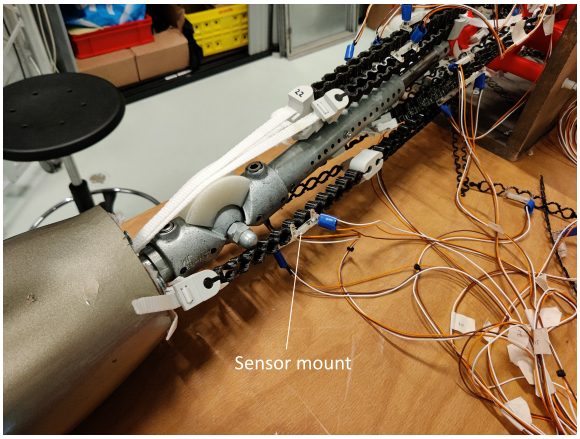
(a)



(b)



(c)



(d)

Supplementary figure 18: (a) Full view of the hip simulator. (b) Patella and quadriceps. (c) Side view of the pelvis. The joint is also visible. (d) View of the hamstring muscle group on the simulator.

D *MATLAB* code

D.1 Characterisation data processing

```
%% Stress strain curves of TPU 95A

clc
clear all
close all

%% Reading data

X51 = readmatrix('20230307_15mms_01.txt'); %strain rate of 0.45/s
X52 = readmatrix('20230307_15mms_02.txt');
X53 = readmatrix('20230307_15mms_03.txt');

X051 = readmatrix('20230307_1.5mms_01.txt'); %strain rate of 0.045/s
X052 = readmatrix('20230307_1.5mms_02.txt');
X053 = readmatrix('20230307_1.5mms_03.txt');

X0051 = readmatrix('20230307_0.15mms_01.txt'); %strain rate of 0.0045/s
X0052 = readmatrix('20230307_0.15mms_02.txt');
X0053 = readmatrix('20230307_0.15mms_03.txt');

%% Creating force and displacement matrices

F5(:,1) = X51(5:1000,2);
F5(:,2) = X52(5:1000,2);
F5(:,3) = X53(5:1000,2);

F05(:,1) = X051(2:1000,2);
F05(:,2) = X052(2:1000,2);
F05(:,3) = X053(2:1000,2);

F005(:,1) = X0051(3:1000,2);
F005(:,2) = X0052(3:1000,2);
F005(:,3) = X0053(3:1000,2);

d5(:,1) = X51(5:1000,3);
d5(:,2) = X52(5:1000,3);
d5(:,3) = X53(5:1000,3);

d05(:,1) = X051(2:1000,3);
d05(:,2) = X052(2:1000,3);
d05(:,3) = X053(2:1000,3);

d005(:,1) = X0051(3:1000,3);
d005(:,2) = X0052(3:1000,3);
d005(:,3) = X0053(3:1000,3);

% Creating alternate arrays using deflection from preload

FF51 = X51(66:1000,2);
FF52 = X52(53:1000,2);
FF53 = X53(194:1000,2);

dd51 = X51(66:1000,4);
dd52 = X52(53:1000,4);
dd53 = X53(194:1000,4);

%% Constants

w = 2; %width in mm
t = 1.9; %thickness in mm
l0 = 33; %original length between shoulders in mm (reference length)

A = w*t; %cross-sectional area

%% Stress and strain matrices

sigma5 = F5/A;
sigma05 = F05/A;
sigma005 = F005/A;
```

```
eps5 = d5/l0;
eps05 = d05/l0;
eps005 = d005/l0;

% Creating alternate arrays using deflection from preload

sigma51 = FF51/A;
sigma52 = FF52/A;
sigma53 = FF53/A;

eps51 = dd51/l0;
eps52 = dd52/l0;
eps53 = dd53/l0;

% Making them the same length and combining them into one matrix

sigma5_true = [sigma51(1:length(sigma53)) sigma52(1:length(sigma53)) sigma53];
eps5_true = [eps51(1:length(eps53)) eps52(1:length(eps53)) eps53];

% Finding the mean curve by using linear interpolation

numPoints = 1000; % Whatever resolution you want.
xCommon = linspace(max(eps5_true(1,:)), 6, numPoints);
ySum = zeros(1, numPoints);
for k = 1 : 3
    % Somehow get this particular set of x and y
    thisx = eps5_true(:,k);
    thisy = sigma5_true(:,k);
    % Interpolate y so that it's using a common x axis.
    yCommon = interp1(thisx, thisy, xCommon);
    % Add it in to the other curves.
    ySum = ySum + yCommon;
end
% Divide the sum by the number of curves to get the average curve.
yAverage = ySum / 3;

eps5_mean = xCommon';
sigma5_mean = yAverage';

data = [eps5_mean sigma5_mean];
data = data(1:924,:)

%% Plotting

figure
title('Tensile properties of TPU 95A samples')
plot(eps5,sigma5,'k',eps05,sigma05,'b',eps005,sigma005,'r')
grid minor
legend('Strain rate 0.45s^-1','','','Strain rate 0.045s^-1','','', ...
    'Strain rate 0.0045s^-1','','')
xlabel('Engineering strain [-]')
ylabel('Engineering stress [MPa]')

figure
plot(eps5,sigma5)
title('Stress-strain of 0.45/s using extension values')
xlabel('Engineering strain [-]')
ylabel('Engineering stress [MPa]')

figure
plot(eps51,sigma51,eps52,sigma52,eps53,sigma53,eps5_mean,sigma5_mean)
title('Stress-strain of 0.45/s using deflection from preload')
xlabel('Engineering strain [-]')
ylabel('Engineering stress [MPa]')
legend('Sample 1','Sample 2','Sample 3','mean')
grid minor

figure
plot(eps5,F5)
xlabel('Engineering strain [-]')
ylabel('Force [N]')
```

D.2 Shape optimisation

```
%% Optimisation

clc
close all
clear

%% Reading data

Xnomiddle = readmatrix('No middle beam R=32.10 t=2.5.txt');
Xnomiddlet5 = readmatrix('No middle beam R=32.10 t=5.txt');
Xnomiddlescaled = readmatrix('No middle beam d0=80 t=5.txt');
XS7arc = readmatrix('S-shape 7 arcs d0=54.txt');
XS9arc = readmatrix('S-shape 9 arcs d0=54.txt');
XS3arc = readmatrix('S-shape 3 arcs d0=54 t=1.4.txt');
XSt5di9 = readmatrix('S-shape 3 arcs d0=54 t=5 di=9.txt');
XSt3la71 = readmatrix('S-shape 3 arcs d0=54 t=3 la=71.txt');
XSt4la715 = readmatrix('S-shape 3 arcs d0=54 t=4 la=71.5.txt');
XSt4la645 = readmatrix('S-shape 3 arcs d0=54 t=4 la=64.5.txt');
XSt4la679 = readmatrix('S-shape 3 arcs d0=54 t=4 la=67.9.txt');
XSt41a679long = readmatrix('S-shape 3 arcs d0=92 t=4 la=67.9.txt');


d = 54; %Length of middle leaf in mm

%% Creating matrices

Fnomiddle = Xnomiddle(:,4);
Fnomiddlet5 = Xnomiddlet5(:,4);
Fnomiddlescaled = Xnomiddlescaled(:,4);
FSt5di9 = XSt5di9(:,4);
FSt3la71 = XSt3la71(:,4);
FSt4la715 = XSt4la715(:,4);
FSt4la645 = XSt4la645(:,4);
FSt4la679 = XSt4la679(:,4);
FSt4la679long = XSt41a679long(:,4);

dnomiddle = Xnomiddle(:,5);
dnomiddlet5 = Xnomiddlet5(:,5);
dnomiddlescaled = Xnomiddlescaled(:,5);
dSt5di9 = XSt5di9(:,5);
dSt3la71 = XSt3la71(:,5);
dSt4la715 = XSt4la715(:,5);
dSt4la645 = XSt4la645(:,5);
dSt4la679 = XSt4la679(:,5);
dSt4la679long = XSt41a679long(:,5);

enomiddle = dnomiddle/d;
enomiddlet5 = dnomiddlet5/d;
enomiddlescaled = dnomiddlescaled/80;
eSt5di9 = dSt5di9/d;
eSt3la71 = dSt3la71/d;
eSt4la715 = dSt4la715/d;
eSt4la645 = dSt4la645/d;
eSt4la679 = dSt4la679/d;
eSt4la679long = dSt4la679long/92;


%% Plotting

figure
plot(enomiddle,Fnomiddle,enomiddlet5,Fnomiddlet5,eSt3la71,FSt3la71, ...
    eSt4la715,FSt4la715,eSt4la645,FSt4la645,eSt4la679,FSt4la679, ...
    eSt4la679long,FSt4la679long)
xlabel('Strain [-]')
ylabel('Force [N]')
legend('No middle beam R=32.10, t=2.5','No middle beam R=32.10, t=5', ...
    'S-shape 3 arcs t=3 la=71','S-shape 3 arcs t=4 la=71.5', ...
    'S-shape 3 arcs t=4 la=64.5','S-shape 3 arcs t=4 la=67.9', ...
    'S-shape 3 arcs t=4 la=67.9 lengthened')

% figure
```

```
% plot(enomiddlet5,Fnomiddlet5,enomiddlescaled,Fnomiddlescaled)
% xlabel('Strain [-]')
% ylabel('Force [N]')
% legend('d0 = 54','d0 = 80')

figure
plot(enomiddle,Fnomiddle)
xlabel('Strain [-]')
ylabel('Force [N]')
title('Force-strain response of circular spring concept')
```

D.3 Hourglass shape tensile test

```
%% S-shape tensile test simulation
clc
clear all
close all

%% Data
X08 = readmatrix('Tensile test t=0.8.txt'); % simulated tensile test 0.8mm thickness
X4 = readmatrix('Tensile test t=4.txt'); % simulated tensile test 4mm thickness

% Experimental data
Xv1 = readmatrix('Trek Flex Matthijs1.csv');
Xv2 = readmatrix('Trek Flex Matthijs2.csv');
Xv3 = readmatrix('Trek Flex Matthijs3.csv');
Xv4 = readmatrix('Trek Flex Matthijs4.csv');
Xv5 = readmatrix('Trek Flex Matthijs5.csv');

d0 = 21.6;

%% Cleaning data


%% Force, deformation and strain
% Simulated tests
F4 = X4(:,4);
F08 = X08(:,4);

d4 = X4(:,5);
d08 = X08(:,5);

e4 = d4/d0;
e08 = d08/d0;

% Experimental data
Fv1 = Xv1(:,4);
Fv2 = Xv2(:,4);
Fv3 = Xv3(:,4);
Fv4 = Xv4(:,4);
Fv5 = Xv5(:,4);

dv1 = Xv1(:,1)-Xv1(1,1);
dv2 = Xv2(:,1)-Xv2(1,1);
dv3 = Xv3(:,1)-Xv3(1,1);
dv4 = Xv4(:,1)-Xv4(1,1);
dv5 = Xv5(:,1)-Xv5(1,1);

ev1 = dv1/d0;
ev2 = dv2/d0;
ev3 = dv3/d0;
ev4 = dv4/d0;
ev5 = dv5/d0;

%% Plotting
figure
plot(e4,F4,ev1,Fv1,ev2,Fv2,ev3,Fv3,ev4,Fv4,ev5,Fv5);
xlabel('Strain [-]')
ylabel('Force [N]')
legend('Simulated results','Experimental results 1', ...
    'Experimental results 2','Experimental results 3', ...
```

```
    'Experimental results 4','Experimental results 5')

figure
plot(d4,F4)
xlabel('Deflection [mm]')
ylabel('Force [N]')

F_max = interp1(e08,F08,0.5)
```

D.4 Live readout script for characterisation

```
    %% Sensor readout tensile tester
% Run this section to stop reading and writing

try
    fclose(s);
    delete(s);
catch
end

clc;
clear all;
close all;

%% Initialisation

% Prompt
prompt = "Specify file name >> ";
filename = input(prompt,"s");

% Ensure the file name has a .txt extension
if ~endsWith(filename, '.txt')
    filename = [filename, '.txt'];
end

% Check if the file already exists
if exist(filename,'file') == 2
    error('File name already in use. Please enter a different file name.')
end

% Open file
fid = fopen(filename,'a');

% Check if the file was opened successfully
if fid == -1
    error('Unable to open the file for writing.');
end

% Connect to Arduino
s = serialport("COM5", 9600); % Connect to serial device (Arduino)


%% Callback function
% Set up callback function
callbackFunction = @(src,~) readAndUpdate(src,fid);

% Set up asynchronous reading
configureCallback(s, "terminator", callbackFunction);

function readAndUpdate(src,fid)
    persistent r myfile saveline;
    if isempty(r)
        r = zeros(26, 1);
        myfile = fid;
        saveline = "0";
    end
    sensordata = readline(src); % Read Arduino output
    try
        sensordata_s = strrep(sensordata, ',', ' ') % separate data
        r = str2num(sensordata_s)'; % Convert data to array
        saveline = sensordata; % Save line for printing to file
```

```
    catch
        % Handle data conversion errors
    end
    fprintf(myfile,saveline);

    end
```

D.5 Sensor calibration

```
    % Script for calibrating hall sensors using tensile test data of every
% muscle
clear all
close all
clc

%% Initialisation
% Enter muscle number
musclenum = 25;

%% Tensile test data
% Create cell
Xt = cell(1,5);
Yt = cell(1,5);

% Read data
for i = 1:5
    specimen = ['Specimen ', num2str(i)];
    Xt{i} = readmatrix([num2str(musclenum), '.xlsx'], 'Sheet', specimen);
end

% Get muscle extension values
d_max = readmatrix('MuscleExtensions.xlsx');

% Clean data
for i = 1:5
    % Order columns
    Yt{i} = [Xt{i}(:,4) Xt{i}(:,1:2)];
    % Remove zero displacement values and highest strain values
    Yt{i} = Yt{i}(Yt{i}(:,2) > 0.001 & Yt{i}(:,2) < 0.98*d_max(musclenum),:);
    % Set time to 0
    Yt{i}(:,1) = Yt{i}(:,1) - Yt{i}(1,1);
end

%% Calibration data
% Create cell
Xs = cell(1,5);
Ys = cell(1,5);

% Read data
for i = 1:5
    Xs{i} = readmatrix([num2str(musclenum), '_', num2str(i)]);
end

% Select time and relevant sensor number
for i = 1:5
    Ys{i} = [Xs{i}(:,1)/1000 Xs{i}(:,musclenum+1)]; % Also convert from ms to s
end

% i_exclude = 5; % Exclude one set if necessary

% Find the start of the test
i_start = zeros(1,5); % Create empty array
threshold = 20;
for i = 1:5
        drop = ischange(Ys{i}(:,2),'mean','Threshold',threshold);
        indices = find(drop == 1);
        i_start(i) = indices(1);
end

% Brute force deviant starting point
% i_start(1) = 47;
```

```
64  % Plot to check starting points
65  for i = 1:5
66      figure
67      plot(Ys{i}(:,1),Ys{i}(:,2),'.',Ys{i}(i_start(i),1),Ys{i}(i_start(i),2),'ro','MarkerSize',5);
68  end

70  % Clean data
71  for i =1:5
72      % Set start of test as first point
73      Ys{i} = Ys{i}(i_start(i):end,:);
74      % Set time at first point to 0
75      Ys{i}(:,1) = Ys{i}(:,1) - Ys{i}(1,1);
76      % Remove data after end of test
77      Ys{i} = Ys{i}(Ys{i}(:,1) <= max(Yt{i}(:,1)),:);
78  end

80  % Plot to check
81  figure
82  for i = 1:5
83      plot(Ys{i}(:,1),Ys{i}(:,2));
84      hold on
85  end
86  legend

88  %% Interpolate data at sensor timestep
89  % Find the lowest final timestamp across all tests
90  t_last = zeros(1,5); % Create empty array for final timesteps
91  numRows = zeros(1,5); % Create empty array for vector lengths

93  for i = 1:5
94      t_last(i) = Ys{i}(end,1); % Find last stamps
95      numRows(i) = length(Ys{i}); % Find number of rows
96  end

98  t_lastmin = min(t_last); % Find lowest t value
99  numRowsmin = min(numRows); % Find lowest timestep number

101 % Create time, displacement, force and voltage arrays
102 % t = linspace(0,t_lastmin,numRowsmin); % Create common time array
103 t = linspace(0,t_lastmin,1000);
104 d = zeros(length(t),5);
105 F = zeros(length(t),5);
106 V = zeros(length(t),5);

108 for i = 1:5
109 %     if i ~= i_exclude % Uncomment this and 'end' to exclude a bad dataset
110     d(:,i) = interp1(Yt{i}(:,1),Yt{i}(:,2),t);
111     F(:,i) = interp1(Yt{i}(:,1),Yt{i}(:,3),t);
112     V(:,i) = interp1(Ys{i}(:,1),Ys{i}(:,2),t);
113 %     end
114 end

116 % Convert bits to volts
117 V = V/1023 * 5;

119 %% Create mean arrays
120 d_avg = mean(d,2);
121 F_avg = mean(F,2);
122 V_avg = mean(V,2);

124 % % Use this method instead when excluding a dataset
125 % % Create empty array for sum of columns
126 % d_sum = zeros(length(t),1);
127 % F_sum = zeros(length(t),1);
128 % V_sum = zeros(length(t),1);
129 %
130 % for i = 1:5
131 %     if i ~= i_exclude
132 %         d_sum = d_sum + d(:,i);
133 %         F_sum = F_sum + F(:,i);
134 %         V_sum = V_sum + V(:,i);
135 %     end
```

```
136 % end
137 %
138 % % Calculate average
139 % d_avg = d_sum/4;
140 % F_avg = F_sum/4;
141 % V_avg = V_sum/4;

143 % Plot to check average curves
144 % figure
145 % for i = 1:5
146 %     plot(t,d(:,i))
147 %     hold on
148 % end
149 % hold on
150 % plot(t,d_avg)
151 % legend
152 %
153 % figure
154 % for i = 1:5
155 %     plot(t,F(:,i))
156 %     hold on
157 % end
158 % hold on
159 % plot(t,F_avg)
160 % legend

162 %% Fit data

164 % Create fit objects
165 [fitfn_d, gof_d] = fit(V_avg,d_avg,'power2');
166 [fitfn_F, gof_F] = fit(V_avg,F_avg,'power2');

168 % Save gof data into matrix
169 gofdata = [gof_d.sse gof_d.rsquare gof_d.dfe gof_d.adjrsquare gof_d.rmse;
170     gof_F.sse gof_F.rsquare gof_F.dfe gof_F.adjrsquare gof_F.rmse];

172 %% Check fits
173 % Create voltage array from 2.5V - 5V
174 Vfit = linspace(2.5,5,100);

176 % Calculate fitted data
177 dfit = fitfn_d(Vfit);
178 Ffit = fitfn_F(Vfit);

180 % Plot to check fits
181 figure
182 plot(V_avg,d_avg,Vfit,dfit)
183 title('Displacement fit')
184 xlabel('Voltage [V]')
185 ylabel('Displacement [mm]')
186 legend('Experimental data','Fitted data')

188 figure
189 plot(V_avg,F_avg,Vfit,Ffit)
190 title('Force fit')
191 xlabel('Voltage [V]')
192 ylabel('Force [N]')
193 legend('Experimental data','Fitted data')

195 %% Save coefficients prompt
196 response = input('Save coefficients? (yes/no): ','s');

198 if strcmpi(response, 'yes')
199     % Save coefficients
200     filename = ['Fit coefficients/coeff_',num2str(musclenum),'.txt'];
201     coeff = [coeffvalues(fitfn_d); coeffvalues(fitfn_F)]; % Put coefficients into matrix
202     fid = fopen(filename,'w'); % Open file
203     % Check if file was opened successfully
204     if fid == -1
205         error('Cannot open file for writing');
206     end
207     fprintf(fid,'%d\t%d\t%d\n',coeff');
208     fclose(fid);
```

```
    % Save GOF data
    filename_gof = ['GOF data/GOF_',num2str(musclenum),'.txt'];
    fid_gof = fopen(filename_gof,'w');
    % Check if file was opened successfully
    if fid_gof == -1
        error('Cannot open file for writing');
    end
    fprintf(fid,'%d\t%d\t%d\t%d\t%d\n',gofdata');
    fclose(fid);

    disp('File saved');
elseif strcmpi(response, 'no')
    disp('File not saved');
else
    disp('Invalid response. Please enter "yes" or "no".');
end
```

D.6 Live table

```
% Run this section to close the table and to stop reading data
try
    all_fig = findall(0, 'type', 'figure'); % Close ui figure
    close(all_fig)
catch
end

try
    fclose(s);
    delete(s);
catch
end

clc;
clear all;
close all;

%% Initialisation

% Prompt
prompt = "Specify file name >> ";
filename = input(prompt,"s");

% Ensure the file name has a .txt extension
if ~endsWith(filename, '.txt')
    filename = [filename, '.txt'];
end

% Check if the file already exists
if ~strcmp(filename, 'test.txt') % Test file can be overwritten
    if exist(filename,'file') == 2
        error('File name already in use. Please enter a different file name.')
    end
end

% Open file
fid = fopen(filename,'a');

% Check if the file was opened successfully
if fid == -1
    error('Unable to open the file for writing.');
end

% Connect to Arduino
s = serialport("COM5", 9600); % Connect to serial device (Arduino)

%% Initialise variables
b = zeros(23, 4); % Create empty array for storing sensor data

% Read coefficients
coeff_d = load('coeff_d.txt');
```

```
coeff_F = load('coeff_F.txt');

% Read extensions
d0_read = readmatrix('d0.xlsx');

%% Create UI table
fig = uifigure('position',[100, 100, 1200, 740]);
uit = uitable(fig, "Data", b,'ColumnName',["bits (0 - 1023)","Voltage [V]", ...
    "Force [N]","Strain (%)"],'position',[50 50 800 600]);
musclenames = {'Gluteus maximus';'Adductor magnus';'Gluteus medius';'Psoas';
    'Iliacus';'Sartorius';'Adductor longus';'Gluteus minimus';
    'Adductor brevis';'Gracilis';'Pectineus';'Tensor fasciae latae';
    'Obturator externus';'Piriformis';'Quadratus femoris';
    'Obturator internus';'Gemelli';'Semimembranosus';'Biceps femoris long head';
    'Semitendinosus';'Biceps femoris short head';
    'Rectus femoris';'Vastus lateralis';'Vastus intermedius';'Vastus medialis'};

order = [1 3 8 12 9 7 2 11 10 4 5 13 16 14 17 15 18 20 19 21 6 22 23 24 25]; % Reorder
musclenames = musclenames(order,:);
musclenames([12, 13],:) = [];
uit.RowName = musclenames;

%% Update table

% Set up callback function
callbackFunction = @(src,~) readAndUpdate(src,uit,fid,coeff_d,coeff_F,d0_read);

% Set up asynchronous reading
configureCallback(s, "terminator", callbackFunction);

function readAndUpdate(src,uit,fid,coeff_d,coeff_F,d0_read)
    persistent uitHandle r b V F d eps myfile saveline a_d b_d c_d a_F b_F c_F;
    persistent d0 colorData order data;
    if isempty(uitHandle)
        uitHandle = uit;
    end
    if isempty(r) % Run once
        r = zeros(26, 1);
       % d0 = 53.7;
        myfile = fid;
        saveline = "0";
        % Create arrays for displacement and force coefficients
        data = zeros(25,4); % Empty array for storing all data

        a_d = coeff_d(:,1);
        b_d = coeff_d(:,2);
        c_d = coeff_d(:,3);

        a_F = coeff_F(:,1);
        b_F = coeff_F(:,2);
        c_F = coeff_F(:,3);
        % Unloaded lengths
        d0 = d0_read;
        % Initialise colors
        colorData = repmat([1 1 1],25,1);
        % Change order based on muscle group
        order = [1 3 8 12 9 7 2 11 10 4 5 13 16 14 17 15 18 20 19 21 6 22 23 24 25];
    end
    sensordata = readline(src); % Read Arduino output
    try
        sensordata_s = strrep(sensordata, ',', ' '); % separate data
        r = str2num(sensordata_s)'; % Convert data to array
        b = r(2:26); % Remove timestamps
        V = (b/1023)*5; % Calculate voltage
        F = a_F.*V.^b_F + c_F; % Calculate force
        d = a_d.*V.^b_d + c_d; % Displacement
        eps = (d./d0)*100; % Strain
        saveline = sensordata; % Save line for printing to file
        data = [b V F eps]; % Store data in matrix
        data = data(order,:); % Order data

        % Change row background color depending on strain value
        white_indices = eps < 5;
```

```
        yellow_indices = eps >= 5 & eps < 10;
        orange_indices = eps >= 10 & eps < 20;
        red_indices = eps >= 20;

        % White color for strain less than 5
        colorData(white_indices,:) = repmat([1,1,1],sum(white_indices),1);
        % Yellow color for strain between 5 and 10
        colorData(yellow_indices,:) = repmat([1,1,0],sum(yellow_indices),1);
        % Orange for strains between 10 and 20
        colorData(orange_indices,:) = repmat([0.9290 0.6940 0.1250],sum(orange_indices),1);
        % Red color for strain greater than or equal to 20
        colorData(red_indices,:) = repmat([1,0.2,0.2],sum(red_indices),1);


    catch
        % Handle data conversion errors
    end
    data([12, 13], :) = []; % Remove rows 13 and 16
    colorData = colorData(order,:);
    colorData([12, 13],:) = [];
    set(uitHandle,'Data',data,'BackgroundColor',colorData); % Update table
    fprintf(myfile,saveline);
end
```

D.7 Material and shape property testing

```
    %% Data processing of final tensile tests
clc;
clear all;
close all;

%% Reading matrices
% New sample test
Xns1 = readmatrix("New sample test.xlsx", 'Sheet', 'Specimen 1');
Xns2 = readmatrix("New sample test.xlsx", 'Sheet', 'Specimen 2');
Xns3 = readmatrix("New sample test.xlsx", 'Sheet', 'Specimen 3');
Xns4 = readmatrix("New sample test.xlsx", 'Sheet', 'Specimen 4');
Xns5 = readmatrix("New sample test.xlsx", 'Sheet', 'Specimen 5');

% Old sample test
Xos1 = readmatrix("Old sample test.xlsx", 'Sheet', 'Specimen 7');
Xos2 = readmatrix("Old sample test.xlsx", 'Sheet', 'Specimen 2');
Xos3 = readmatrix("Old sample test.xlsx", 'Sheet', 'Specimen 3');
Xos4 = readmatrix("Old sample test.xlsx", 'Sheet', 'Specimen 4');
Xos5 = readmatrix("Old sample test.xlsx", 'Sheet', 'Specimen 5');

% T-fit loading and unloading at 9.72mm/s crosshead speed
Xtffast1 = readmatrix("T-fit 972 loading unloading.xlsx", 'Sheet', 'Specimen 1');
Xtffast2 = readmatrix("T-fit 972 loading unloading.xlsx", 'Sheet', 'Specimen 2');
Xtffast3 = readmatrix("T-fit 972 loading unloading.xlsx", 'Sheet', 'Specimen 3');
Xtffast4 = readmatrix("T-fit 972 loading unloading.xlsx", 'Sheet', 'Specimen 4');
Xtffast5 = readmatrix("T-fit 972 loading unloading.xlsx", 'Sheet', 'Specimen 5');

% T-fit loading 4.32mm/s
Xtfmed1 = readmatrix("T-fit 432.xlsx", 'Sheet', 'Specimen 1');
Xtfmed2 = readmatrix("T-fit 432.xlsx", 'Sheet', 'Specimen 2');
Xtfmed3 = readmatrix("T-fit 432.xlsx", 'Sheet', 'Specimen 3');
Xtfmed4 = readmatrix("T-fit 432.xlsx", 'Sheet', 'Specimen 4');
Xtfmed5 = readmatrix("T-fit 432.xlsx", 'Sheet', 'Specimen 6');

% T-fit loading 0.972mm/s
Xtfslow1 = readmatrix("T-fit 0972.xlsx", 'Sheet', 'Specimen 1');
Xtfslow2 = readmatrix("T-fit 0972.xlsx", 'Sheet', 'Specimen 2');
Xtfslow3 = readmatrix("T-fit 0972.xlsx", 'Sheet', 'Specimen 3');
Xtfslow4 = readmatrix("T-fit 0972.xlsx", 'Sheet', 'Specimen 4');
Xtfslow5 = readmatrix("T-fit 0972.xlsx", 'Sheet', 'Specimen 5');

% T-fit loading at 9.72mm/s, hold 30 seconds, unload 9.72mm/s
Xtfhold1 = readmatrix("Load hold unload.xlsx", 'Sheet', 'Specimen 1');
Xtfhold2 = readmatrix("Load hold unload.xlsx", 'Sheet', 'Specimen 2');
Xtfhold3 = readmatrix("Load hold unload.xlsx", 'Sheet', 'Specimen 3');
```

```
Xtfhold4 = readmatrix("Load hold unload.xlsx", 'Sheet', 'Specimen 4');
Xtfhold5 = readmatrix("Load hold unload.xlsx", 'Sheet', 'Specimen 5');

% Old sample old test
Xosot = readmatrix("Old sample old test.xlsx");
Xosot_all = readmatrix("Old sample old test (all datasets)");

% FEM simulation
Xsim = readmatrix('Simulated S-shape tensile test.txt');

%% Cleaning data

d0 = 21.6; % Unloaded vertical length of sample
n = 100; % number of points
xCommon = linspace(0, 0.5*d0, n)';

% Separate Xtffast in loading and unloading part
% Loading
Xtffast1l = Xtffast1(diff(Xtffast1(:, 1), 1, 1) > 0, :);
Xtffast2l = Xtffast2(diff(Xtffast2(:, 1), 1, 1) > 0, :);
Xtffast3l = Xtffast3(diff(Xtffast3(:, 1), 1, 1) > 0, :);
Xtffast4l = Xtffast4(diff(Xtffast4(:, 1), 1, 1) > 0, :);
Xtffast5l = Xtffast5(diff(Xtffast5(:, 1), 1, 1) > 0, :);

% Unloading
Xtffast1u = Xtffast1(diff(Xtffast1(:, 1), 1, 1) < 0, :);
Xtffast2u = Xtffast2(diff(Xtffast2(:, 1), 1, 1) < 0, :);
Xtffast3u = Xtffast3(diff(Xtffast3(:, 1), 1, 1) < 0, :);
Xtffast4u = Xtffast4(diff(Xtffast4(:, 1), 1, 1) < 0, :);
Xtffast5u = Xtffast5(diff(Xtffast5(:, 1), 1, 1) < 0, :);

% Remove extension values > 10.8mm
Xtfmed1 = Xtfmed1(Xtfmed1(:, 1) < 0.5*d0, :);
Xtfmed2 = Xtfmed2(Xtfmed2(:, 1) < 0.5*d0, :);
Xtfmed3 = Xtfmed3(Xtfmed3(:, 1) < 0.5*d0, :);
Xtfmed4 = Xtfmed4(Xtfmed4(:, 1) < 0.5*d0, :);
Xtfmed5 = Xtfmed5(Xtfmed5(:, 1) < 0.5*d0, :);

% Split Xtfhold in load, hold and unload arrays
% Remove vales > 10.8mm extension
% Xtfhold1lu = Xtfhold1(Xtfhold1(:, 1) < 10.79, :);
% Xtfhold2lu = Xtfhold2(Xtfhold2(:, 1) < 10.79, :);
% Xtfhold3lu = Xtfhold3(Xtfhold3(:, 1) < 10.79, :);
% Xtfhold4lu = Xtfhold4(Xtfhold4(:, 1) < 10.79, :);
% Xtfhold5lu = Xtfhold5(Xtfhold5(:, 1) < 10.79, :);
%
% Xtfhold1l = Xtfhold1lu(diff(Xtfhold1lu(:, 1), 1, 1) > 0, :);
% Xtfhold2l = Xtfhold2lu(diff(Xtfhold2lu(:, 1), 1, 1) > 0, :);
% Xtfhold3l = Xtfhold3lu(diff(Xtfhold3lu(:, 1), 1, 1) > 0, :);
% Xtfhold4l = Xtfhold4lu(diff(Xtfhold4lu(:, 1), 1, 1) > 0, :);
% Xtfhold5l = Xtfhold5lu(diff(Xtfhold5lu(:, 1), 1, 1) > 0, :);
%
% Xtfhold1h = Xtfhold1(Xtfhold1(:, 1) > 10.79, :);
% Xtfhold2h = Xtfhold2(Xtfhold2(:, 1) > 10.79, :);
% Xtfhold3h = Xtfhold3(Xtfhold3(:, 1) > 10.79, :);
% Xtfhold4h = Xtfhold4(Xtfhold4(:, 1) > 10.79, :);
% Xtfhold5h = Xtfhold5(Xtfhold5(:, 1) > 10.79, :);
%
% Xtfhold1u = Xtfhold1lu(diff(Xtfhold1lu(:, 1), 1, 1) < 0, :);
% Xtfhold2u = Xtfhold2lu(diff(Xtfhold2lu(:, 1), 1, 1) < 0, :);
% Xtfhold3u = Xtfhold3lu(diff(Xtfhold3lu(:, 1), 1, 1) < 0, :);
% Xtfhold4u = Xtfhold4lu(diff(Xtfhold4lu(:, 1), 1, 1) < 0, :);
% Xtfhold5u = Xtfhold5lu(diff(Xtfhold5lu(:, 1), 1, 1) < 0, :);

%% Get the averaged curves

Fns = getAverageCurve(Xns1,Xns2,Xns3,Xns4,Xns5,xCommon);
Fos = getAverageCurve(Xos1,Xos2,Xos3,Xos4,Xos5,xCommon);
Ftffastl = getAverageCurve(Xtffast1l,Xtffast2l,Xtffast3l,Xtffast4l,Xtffast5l,xCommon);
Ftffastu = flip(getAverageCurve(Xtffast1u,Xtffast2u,Xtffast3u,Xtffast4u,Xtffast5u,xCommon));
Ftffast = [Ftffastl; Ftffastu];
Ftfmed = getAverageCurve(Xtfmed1,Xtfmed2,Xtfmed3,Xtfmed4,Xtfmed5,xCommon);
Ftfslow = getAverageCurve(Xtfslow1,Xtfslow2,Xtfslow3,Xtfslow4,Xtfslow5,xCommon);
```

```
% Ftfhold = [getAverageCurve(Xtfhold1l, Xtfhold2l, Xtfhold3l, Xtfhold4l, Xtfhold5l,xCommon);
%       getAverageCurve(Xtfhold1h, Xtfhold2h, Xtfhold3h, Xtfhold4h, Xtfhold5h,xCommon);
%       getAverageCurve(Xtfhold1u, Xtfhold2u, Xtfhold3u, Xtfhold4u, Xtfhold5u,xCommon)];
Ftfhold = mean([Xtfhold1(:,2), Xtfhold2(:,2), Xtfhold3(:,2), Xtfhold4(:,2), Xtfhold5(:,2)],2);

% Old sample old test (already averaged in separate file)
xosot = Xosot(:,1);
Fosot = Xosot(:,2);

% x point sets to match data
xtffast = [xCommon; flip(xCommon)];
xtfhold = mean([Xtfhold1(:,1), Xtfhold2(:,1), Xtfhold3(:,1), Xtfhold4(:,1), Xtfhold5(:,1)],2);

% Compensate for higher preload of T-fit sample
preload = min(Ftffastl);
cor = Fns > preload; % Logical array to remove forces below 1N
Fns_c = Fns(cor);
xns_c = xCommon(cor) - xCommon(length(Fns(Fns < preload))); % Set elongation at 1N to 0

%% Calculate strain curves
epsCommon = xCommon/d0;
epsns_c = xns_c/d0;
epsosot = xosot/d0;
epstffast = xtffast/d0;
epstfhold = xtfhold/d0;

%% Extra cleaning before plotting
% Make one matrix for x and y
Ytffastl = [epsCommon Ftffastl];
Ytfmed = [epsCommon Ftfmed];
Ytfslow = [epsCommon Ftfslow];

% Remove rows under 0.45 strain
keeprows = epsCommon < 0.47;
Ytffastl = Ytffastl(keeprows,:);
Ytfmed = Ytfmed(keeprows,:);
Ytfslow = Ytfslow(keeprows,:);

%% Plotting

ax = [0 0.6 -1 80];

% % Comparison of fresh and old clamped sample
% figure;
% plot(epsCommon, Fos, 'r', epsosot, Fosot, 'g');
% xlabel('Strain [-]');
% ylabel('Force [N]');
% legend('Old sample', 'Fresh sample')
% axis(ax)
%
% % Comparison of clamp and T-fit
% figure;
% plot(epsns_c, Fns_c, 'c', Ytffastl(:, 1), Ytffastl(:, 2), 'm');
% xlabel('Strain [-]');
% ylabel('Force [N]');
% legend('Clamped sample', 'Sample with T-fit')
% axis(ax)
%
% % Strain-rate dependence
% figure;
% plot(Ytffastl(:,1), Ytffastl(:,2), 'r', Ytfmed(:,1), Ytfmed(:,2), 'g', ...
%       Ytfslow(:,1), Ytfslow(:,2), 'b');
% xlabel('Strain [-]');
% ylabel('Force [N]');
% legend('0.45/s', '0.225/s', '0.045/s')
% axis(ax)
%
% % Hysteresis
% figure;
% plot(epstffast, Ftffast);
% xlabel('Strain [-]');
% ylabel('Force [N]');
% axis(ax)
```

```
%
% % Stress relaxation
% figure;
% plot(epstfhold, Ftfhold);
% xlabel('Strain [-]');
% ylabel('Force [N]');
% axis(ax)

%% Statistics processing

% Old vs new sample test
% Fos_int = interp1(epsCommon,Fos,epsosot);
x_osot_all = Xosot_all(:,1);
Fos_int = getInterp(Xos1,Xos2,Xos3,Xos4,Xos5,x_osot_all);
eps_osot_all = x_osot_all/d0;
Y_oldvsnew = [eps_osot_all Xosot_all(:,2:end) Fos_int];


% Clamp vs T-fit
% Ytffast_int = interp1(Ytffastl(:,1),Ytffastl(:,2),epsns_c);
% Y_clampvstfit = [epsns_c, Fns_c, Ytffast_int];
Ytffastl_int = getInterp(Xtffast1l,Xtffast2l,Xtffast3l,Xtffast4l, ...
    Xtffast5l,xCommon); % Interpolate T-fit sample at same points
Fns_all = getInterp(Xns1,Xns2,Xns3,Xns4,Xns5,xCommon);
% Compensate for higher preload of T-fit sample
preload = min(min(Ytffastl_int));
cor = all(Fns_all > preload,2); % Logical array to remove forces below 1N
Fns_all_c = Fns_all(cor,:);
% Set elongation at 1N to 0
xns_all_c = xCommon(cor) - xCommon(length(Fns_all) - length(Fns_all_c));
Ytffastl_int_c = getInterp(Xtffast1l,Xtffast2l,Xtffast3l,Xtffast4l, ...
    Xtffast5l,xns_all_c); % T-fit set corrected
% Everything in one matrix for graphpad
Y_clampvsfit = [xns_all_c/d0 Fns_all_c Ytffastl_int_c];
Y_clampvsfit = Y_clampvsfit(all(~isnan(Y_clampvsfit),2),:); % Remove NaN values


% Strain rate dependence
% Y_strratedep = [Ytffastl(:,1) Ytffastl(:,2) Ytfmed(:,2) Ytfslow(:,2)];
% Y_strratedep = Y_strratedep(3:end,:); % remove nan rows
Ytfmed_all = getInterp(Xtfmed1,Xtfmed2,Xtfmed3,Xtfmed4,Xtfmed5,xCommon);
Ytfslow_all = getInterp(Xtfslow1,Xtfslow2,Xtfslow3,Xtfslow4,Xtfslow5,xCommon);
Y_strratedep = [epsCommon Ytffastl_int Ytfmed_all Ytfslow_all];
Y_strratedep = Y_strratedep(all(~isnan(Y_strratedep),2),:);

% FEM vs new T-fit test
% Ysim = [Xsim(:,5) Xsim(:,4)];
% Ysim(any(isnan(Ysim),2),:) = []; % Remove NaN values
% Ytffastl_intshort = getInterp(Xtffast1l,Xtffast2l,Xtffast3l,Xtffast4l, ...
%     Xtffast5l,Ysim(:,1));
% Y_FEMvsexp = [Ysim Ytffastl_intshort];
% Y_FEMvsexp(any(isnan(Y_FEMvsexp),2),:) = []; % Remove NaN rows

%

%% Plots with all datasets

% Comparison of old and fresh sample
figure
hold on
for i = 1:5
    plot(Y_oldvsnew(:,1),Y_oldvsnew(:,i+1),'color', ...
        [0.4660 0.6740 0.1880 0.18],'Handlevisibility','off')
    plot(Y_oldvsnew(:,1),Y_oldvsnew(:,i+6),'color', ...
        [0.6350 0.0780 0.1840 0.18],'Handlevisibility','off')
end
plot(epsCommon, Fos,'color',[0.6350 0.0780 0.1840]);
plot(epsosot, Fosot, 'color',[0.4660 0.6740 0.1880])
xlabel('Strain [-]');
ylabel('Force [N]');
legend('5 months old sample', 'Fresh sample','Location','northwest')
axis(ax)
```

```
% Clamp vs T-fit
figure
hold on
for i = 1:5
    plot(Y_clampvsfit(:,1),Y_clampvsfit(:,i),'color', ...
        [0.3010 0.7450 0.9330 0.18],'Handlevisibility','off');
    plot(Y_clampvsfit(:,1),Y_clampvsfit(:,i+5),'color', ...
        [0.4940 0.1840 0.5560 0.18],'Handlevisibility','off');
end
plot(epsns_c, Fns_c,'color',[0.3010 0.7450 0.9330])
    plot(Ytffastl(:, 1), Ytffastl(:, 2),'color',[0.4940 0.1840 0.5560]);
xlabel('Strain [-]');
ylabel('Force [N]');
legend('Clamped sample', 'Sample with T-fit')
axis(ax)

% Strain-rate dependence
figure
hold on
for i = 1:5
    plot(Y_strratedep(:,1),Y_strratedep(:,i+1),'Color', ...
        [0.3010 0.7450 0.9330 0.18],'HandleVisibility','off')
    plot(Y_strratedep(:,1),Y_strratedep(:,i+6),'Color', ...
        [0.9290 0.6940 0.1250 0.18],'HandleVisibility','off')
    plot(Y_strratedep(:,1),Y_strratedep(:,i+11),'Color', ...
        [0.4940 0.1840 0.5560 0.18],'HandleVisibility','off')
end
plot(Ytffastl(:,1),Ytffastl(:,2),'color',[0.3010 0.7450 0.9330]);
plot(Ytfmed(:,1),Ytfmed(:,2), 'Color',[0.9290 0.6940 0.1250])
plot(Ytfslow(:,1),Ytfslow(:,2),'Color',[0.4940 0.1840 0.5560])
xlabel('Strain [-]');
ylabel('Force [N]');
legend('0.45/s', '0.225/s', '0.045/s')
axis(ax)

% Hysteresis
Ftffastl_all = getInterp(Xtffast1l,Xtffast2l,Xtffast3l,Xtffast4l, ...
    Xtffast5l,xCommon); % Get all datasets
Ftffastu_all = flip(getInterp(Xtffast1u,Xtffast2u,Xtffast3u,Xtffast4u, ...
    Xtffast5u,xCommon));
Ytffast_all = [epsCommon Ftffastl_all;
    flip(epsCommon) Ftffastu_all]; % Put everything in one matrix
% Remove rows with NaN entries
Ytffast_all = Ytffast_all(all(~isnan(Ytffast_all),2),:);
figure
hold on
for i = 1:5
    plot(Ytffast_all(:,1),Ytffast_all(:,i+1),'color', ...
        [0 0.4470 0.7410 0.18], 'HandleVisibility','off')
end
plot(epstffast, Ftffast,'color',[0 0.4470 0.7410]);
xlabel('Strain [-]');
ylabel('Force [N]');
axis(ax)

% Stress relaxation
Ytfhold{1} = Xtfhold1;
Ytfhold{2} = Xtfhold2;
Ytfhold{3} = Xtfhold3;
Ytfhold{4} = Xtfhold4;
Ytfhold{5} = Xtfhold5;
figure
hold on
for i = 1:5
    plot(Ytfhold{i}(:,1)/d0,Ytfhold{i}(:,2),'color', ...
        [0 0.4470 0.7410 0.18], 'HandleVisibility','off')
end
plot(epstfhold, Ftfhold,'color',[0 0.4470 0.7410]);
xlabel('Strain [-]');
ylabel('Force [N]');
axis(ax)

%% Standard error of the mean
```

```
% Put all data in one array
Ftfhold_all = zeros(length(Xtfhold1),5);
for i = 1:5
    Ftfhold_all(:,i) = Ytfhold{i}(:,2);
end

% Calculate std and SEM
Ftfhold_std = std(Ftfhold_all,0,2);
n = 5;
Ftfhold_sem = Ftfhold_std / sqrt(n);

% Confidence Interval
alpha = 0.05;
t_val = tinv(1 - alpha/2, n - 1);
CI_lower = Ftfhold - t_val * Ftfhold_sem;
CI_upper = Ftfhold + t_val * Ftfhold_sem;

% Calculate difference and error margin for loss of force
i = 125;  % First point index
j = 431;  % Second point index
difference_Fdrop = Ftfhold(i) - Ftfhold(j);
error_margin_Fdrop = sqrt(Ftfhold_sem(i)^2 + Ftfhold_sem(j)^2);

% Calculate maximum force
[Ftfhold_max, iFtfhold_max] = max(Ftfhold)
Ftfhold_max_std = Ftfhold_sem(iFtfhold_max)
```

D.8 Sensor location test and Bland-Altman plot

```
   %% Sensor location test
clc
clear all
close all
%% Initialisation
% Create cells
Xt_e = cell(5,3);
Xt_t = cell(5,3);
Xt_m = cell(5,3);

Xs_e = cell(5,3);
Xs_t = cell(5,3);
Xs_m = cell(5,3);

% Read tensile tester data
for i = 1:15
    Xt_e{i} = readmatrix('sensorlocation_edge.xlsx','Sheet',['Specimen ',num2str(i)]);
    Xt_t{i} = readmatrix('sensorlocation_third.xlsx','Sheet',['Specimen ',num2str(i)]);
    Xt_m{i} = readmatrix('sensorlocation_mid.xlsx','Sheet',['Specimen ',num2str(i)]);
end

% Read sensor data
for i = 1:3
    for j = 1:5
        Xs_e{j,i} = readmatrix(['spec',num2str(i),'_e',num2str(j),'.txt']);
        Xs_t{j,i} = readmatrix(['spec',num2str(i),'_t',num2str(j),'.txt']);
        Xs_m{j,i} = readmatrix(['spec',num2str(i),'_m',num2str(j),'.txt']);
    end
end

% Unloaded length
d0 = 173.48;

%% Clean data
for i = 1:15
    % Remove third row
    Xt_e{i}(:,3) = [];
    Xt_t{i}(:,3) = [];
    Xt_m{i}(:,3) = [];

    % Remove steps with 0 strain
    Xt_e{i} = Xt_e{i}(Xt_e{i}(:,1) > 0.001,:);
```

```
43      Xt_t{i} = Xt_t{i}(Xt_t{i}(:,1) > 0.001,:);
44      Xt_m{i} = Xt_m{i}(Xt_m{i}(:,1) > 0.001,:);
45
46      % Convert time to s
47      Xs_e{i}(:,1) = Xs_e{i}(:,1)/1000;
48      Xs_t{i}(:,1) = Xs_t{i}(:,1)/1000;
49      Xs_m{i}(:,1) = Xs_m{i}(:,1)/1000;
50  end
51
52  %% Find sensor starting points
53  close all
54  % Find the start of the test
55  i_start = zeros(3,5); % Create empty array
56  threshold = 50;
57  % In the loop below the points are checked for each dataset manually, then
58  % entered into a separate matrix below.
59  % for i = 1:15
60  %         rows = 1:length(Xs_m{i});
61  %         drop = ischange(Xs_m{i}(:,2),'mean','Threshold',threshold);
62  %         indices = find(drop == 1);
63  %         i_start(i) = indices(1);
64  %         figure
65  %         plot(1:length(Xs_m{i}(:,1)),Xs_m{i}(:,2),'-o',rows(i_start(i)), ...
66  %             Xs_m{i}(i_start(i),2),'ro','MarkerSize',5)
67  % end
68
69  i_start_e = [774 258 286 294 209
70      366 257 191 242 192;
71      220 206 223 180 268];
72
73  i_start_t = [323  197 224 147 201;
74      184 200 200 184 219;
75      205 205 177 163 177];
76
77  i_start_m = [194  178 187 157 160
78      166 194 256 169 160
79      166 182 161 169 156];
80
81  %% Synchronise data
82  % Remove t0 from sensor dataset
83  for i = 1:15
84      % Remove t0 from sensor dataset
85      Xs_e{i}(:,1) = Xs_e{i}(:,1) - Xs_e{i}(i_start_e(i),1);
86      Xs_t{i}(:,1) = Xs_t{i}(:,1) - Xs_t{i}(i_start_t(i),1);
87      Xs_m{i}(:,1) = Xs_m{i}(:,1) - Xs_m{i}(i_start_m(i),1);
88
89      % Remove rows before starting point
90      Xs_e{i} = Xs_e{i}(i_start_e(i):end,:);
91      Xs_t{i} = Xs_t{i}(i_start_t(i):end,:);
92      Xs_m{i} = Xs_m{i}(i_start_m(i):end,:);
93
94      % Subtract t0 from tensile tester data
95      Xt_e{i}(:,3) = Xt_e{i}(:,3) - Xt_e{i}(1,3);
96      Xt_t{i}(:,3) = Xt_t{i}(:,3) - Xt_t{i}(1,3);
97      Xt_m{i}(:,3) = Xt_m{i}(:,3) - Xt_m{i}(1,3);
98  end
99
100 % Create averaged dataset
101 xCommon = linspace(0,5.6,1000)';
102
103 Xt_e_int = cell(5,3);
104 Xt_t_int = cell(5,3);
105 Xt_m_int = cell(5,3);
106
107 Xs_e_int = cell(5,3);
108 Xs_t_int = cell(5,3);
109 Xs_m_int = cell(5,3);
110
111 Xt_BA_int = cell(5,3); % Cell for bland-altman plot
112
113 for i = 1:15
114     Xt_e_int{i} = interp1(Xt_e{i}(:,3),Xt_e{i}(:,1),xCommon);
115     Xt_t_int{i} = interp1(Xt_t{i}(:,3),Xt_t{i}(:,1),xCommon);
```

```
116     Xt_m_int{i} = interp1(Xt_m{i}(:,3),Xt_m{i}(:,1),xCommon);

118     Xs_e_int{i} = interp1(Xs_e{i}(:,1),Xs_e{i}(:,2),xCommon);
119     Xs_t_int{i} = interp1(Xs_t{i}(:,1),Xs_t{i}(:,2),xCommon);
120     Xs_m_int{i} = interp1(Xs_m{i}(:,1),Xs_m{i}(:,2),xCommon);

122     Xt_BA_int{i} = interp1(Xt_m{i}(:,3),Xt_m{i}(:,2),xCommon);
123 end

125 % Create mean sets
126 Yt_e_sum = zeros(size(xCommon));
127 Yt_t_sum = zeros(size(xCommon));
128 Yt_m_sum = zeros(size(xCommon));

130 Ys_e_sum = zeros(size(xCommon));
131 Ys_t_sum = zeros(size(xCommon));
132 Ys_m_sum = zeros(size(xCommon));

134 Yt_BA_sum = zeros(size(xCommon));

136 for i = 1:15
137     Yt_e_sum = Yt_e_sum + Xt_e_int{i};
138     Yt_t_sum = Yt_t_sum + Xt_t_int{i};
139     Yt_m_sum = Yt_m_sum + Xt_m_int{i};

141     Ys_e_sum = Ys_e_sum + Xs_e_int{i};
142     Ys_t_sum = Ys_t_sum + Xs_t_int{i};
143     Ys_m_sum = Ys_m_sum + Xs_m_int{i};

145     Yt_BA_sum = Yt_BA_sum + Xt_BA_int{i};
146 end

148 % Create mean datasets and convert bits to voltage
149 f_conv = 5/1023; % Conversion factor

151 Y_e = [Yt_e_sum/15 (Ys_e_sum/15)*f_conv];
152 Y_t = [Yt_t_sum/15 (Ys_t_sum/15)*f_conv];
153 Y_m = [Yt_m_sum/15 (Ys_m_sum/15)*f_conv];
154 Y_BA = [Yt_m_sum/15 Yt_BA_sum/15]; % Left column displacement, right column force

156 %% Plot data
157 % Mean plot
158 figure
159 plot(Y_e(:,1),Y_e(:,2),Y_t(:,1),Y_t(:,2),Y_m(:,1),Y_m(:,2));
160 xlabel('Extension [mm]');
161 ylabel('Voltage [V]')
162 legend('Edge','One third of length','Middle')

164 %% Statistical analysis
165 Y_s = [Y_e(:,2) Y_t(:,2) Y_m(:,2)];

167 Ys_e_all = []; % Create empty matrix
168 Ys_t_all = [];
169 Ys_m_all = [];

171 Yt_e_all = [];
172 Yt_t_all = [];
173 Yt_m_all = [];

175 Yt_BA_all = [];

177 % Put everything into columns
178 for j = 1:size(Xs_e_int,2)
179     for i  = 1:size(Xs_e_int,1)
180         Ys_e_all = [Ys_e_all Xs_e_int{i,j}];
181         Ys_t_all = [Ys_t_all Xs_t_int{i,j}];
182         Ys_m_all = [Ys_m_all Xs_m_int{i,j}];

184         Yt_e_all = [Yt_e_all Xt_e_int{i,j}];
185         Yt_t_all = [Yt_t_all Xt_t_int{i,j}];
186         Yt_m_all = [Yt_m_all Xt_m_int{i,j}];

188         Yt_BA_all = [Yt_BA_all Xt_BA_int{i,j}];
```

```
    end
end

Ys_all = [Ys_e_all Ys_t_all Ys_m_all];
Yt_all = [Yt_e_all Yt_t_all Yt_m_all];
V_all = Ys_all/1023 * 5;

% Calculate largest difference
Y_diff = Y_m(:,2) - Y_t(:,2);
Y_diff_max = max(Y_diff)
Y_diff_max_V = Y_diff_max/f_conv;

% Standard deviation and SEM
Ys_m_std = std(Ys_m_all,0,2);
Ys_t_std = std(Ys_t_all,0,2);
Ys_m_sem = Ys_m_std/sqrt(15);
Ys_t_sem = Ys_t_std/sqrt(15);

% SEM of largest difference
Y_diff_max_sem = sqrt(Ys_m_sem(Y_diff == Y_diff_max)^2 + Ys_t_sem(Y_diff == Y_diff_max)^2)


%% Plot all data
epsCommon = xCommon/d0;
colors = [0 0.4470 0.7410;
    0.4660 0.6740 0.1880;
    0.6350 0.0780 0.1840];
transp = 0.1;
figure
hold on
for i = 1:15
    plot(Yt_all(:,i)/d0,V_all(:,i),'color',[colors(1,:) transp],'HandleVisibility','off');
    plot(Yt_all(:,i+5)/d0,V_all(:,i+5),'color',[colors(2,:) transp],'HandleVisibility','off');
    plot(Yt_all(:,i+10)/d0,V_all(:,i+10),'color',[colors(3,:) transp],'HandleVisibility','off');
end
plot(Y_e(:,1)/d0,Y_e(:,2),'color',colors(1,:))
plot(Y_t(:,1)/d0,Y_t(:,2),'color',colors(2,:))
plot(Y_m(:,1)/d0,Y_m(:,2),'color',colors(3,:));
xlabel('Strain [-]');
ylabel('Voltage [V]')
legend('Edge','One third of length','Middle')
ax = gca;
ax.YLim = [2.4 5];

%% Prepare Bland-Altman data
% Create separate datasets for plot
d_BA_all = Yt_m_all;
F_BA_all = Yt_BA_all;
V_BA_all = V_all(:,31:end);

% Curve fit using the mean
[dfit,dgof] = fit(Y_m(:,2),Y_BA(:,1),'power2'); % Displacement fit
[Ffit,Fgof] = fit(Y_m(:,2),Y_BA(:,2),'power2'); % Force fit

% Create set of fitted data
for i = 1:size(V_BA_all,2)
    dfit_data(:,i) = dfit(V_BA_all(:,i));
    Ffit_data(:,i) = Ffit(V_BA_all(:,i));
end

%% Create Bland-Altman plot
close all;
% Create Bland-Altman plot using 'Bland-Altman and Correlation Plot'
% Copyright (c) 2017, Ran Klein
% All rights reserved.
% d_BA = BlandAltman(dfit_data/d0,d_BA_all/d0,'Strain [-]',[],[],'baYLimMode',[-0.02 0.04]);
% F_BA = BlandAltman(Ffit_data,F_BA_all,'Force [N]',[],[],'baYLimMode',[-3 3]);

[rpc_d,figBA_d,stats_d] = BlandAltman(d_BA_all/d0,dfit_data/d0,'Strain [-]', ...
    [],[],'symbols','num','markerSize',1.1,'data1Mode','truth','baStatsMode', ...
    'Non-parametric','baInfo',[]);
[rpc_F,figBA_F,stats_F] = BlandAltman(F_BA_all,Ffit_data,'Force [N]',[],[], ...
    'symbols','num','markerSize',1.1,'data1Mode','truth','baStatsMode', ...
```

```
    'Non-parametric','baInfo',[]);
```

D.9 Sensor accuracy test

```
    %% Sensor accuracy test
clear all
clc
close all
%%
% Create cells for storing data
Xt = cell(1,10);
Xs = cell(1,10);

% Read data
for i = 1:10
    Xt{i} = readmatrix('sensoraccuracy_data.xlsx','sheet',['Specimen ',num2str(i)]);
    Xs{i} = readmatrix(['acc_',num2str(i),'.txt']);
end

%% Calculation
% Tensile data
Yt = cell(1,10);

for i = 1:10
    Yt{i} = [Xt{i}(:,4) Xt{i}(:,1)]; % Time in left common, displacement in right
    Yt{i} = Yt{i}(Yt{i}(:,2) > 0.001,:); % Remove all columns except displacement,
    % remove zero displacement values
    Yt{i}(:,1) = Yt{i}(:,1) - Yt{i}(1,1); % Set time to 0
end

% Sensor data

% Find the start of the test
i_start = zeros(1,10); % Create empty array
threshold = 500;
for i = 1:10
        drop = ischange(Xs{i}(:,2),'linear','Threshold',threshold);
        indices = find(drop == 1);
        i_start(i) = indices(1);
        % figure
        % plot(Xs{i}(:,1),Xs{i}(:,2),'.',Xs{i}(i_start(i),1), ...
        %     Xs{i}(i_start(i),2),'ro','MarkerSize',5)
end

% Clean sensor data
Ys = cell(1,10);

for i = 1:10
    Ys{i} = Xs{i}(i_start(i):end,:); % Remove values before start of test
    % Set t to 0 at start of test and convert to second
    Ys{i}(:,1) = (Ys{i}(:,1) - Ys{i}(1,1))/1000; s
    Ys{i}(:,2) = Ys{i}(:,2)/1023 * 5; % Convert to volts
end

% Interpolate data at common timesteps
n = 1000; % Amount of steps
xCommon = linspace(0,40,n);

Y = cell(1,10);
Y_mean = zeros(n,2);
Y_gr = []; % Graphpad

for i = 1:10
    Y{i}(:,1) = interp1(Yt{i}(:,1),Yt{i}(:,2),xCommon); % Interpolate displacement
    Y{i}(:,2) = interp1(Ys{i}(:,1),Ys{i}(:,2),xCommon); % Interpolate voltage
    Y_mean = Y_mean + Y{i};

    Y_gr = [Y_gr Y{i}(:,2)];% Data for graphpad
end

Y_mean = Y_mean/10;
```

```
%% Plotting
figure
% axes('XScale', 'log', 'YScale', 'log')
hold on
% for i = 1:10
%     plot(Y{i}(:,1),Y{i}(:,2),'color',[0 0.4470 0.7410 0.18]);
% end
plot(Y_mean(:,1),Y_mean(:,2),'.','color',[0 0.4470 0.7410]);
xlabel('Displacement [mm]')
ylabel('Voltage [V]')
ax = gca;
ax.YLim = [2.4 5];
```

D.10 Curved surface test

```
    %% Curved surface plots
clc
clear all
close all
%% Initialisation
% Read data
X = cell(3,3);

for i = 1:3
    % Top row
    X{1,i} = readmatrix('Before_curve.xlsx','Sheet',['Sheet',num2str(i)]);
    % Middle row
    X{2,i} = readmatrix('On_curve.xlsx','Sheet',['Sheet',num2str(i)]);
    % Bottom row
    X{3,i} = readmatrix('After_curve.xlsx','Sheet',['Sheet',num2str(i)]);
end

%% Calculation
V = cell(size(X)); % Voltage cell

% Remove first row
for i = 1:numel(X)
    X{i}(1,:) = [];
    V{i} = X{i}(:,2)/1023 *5;
end

% Calculate average
b_sum = cell(length(X),1);
b_mean = cell(length(X),1);

for i = 1:size(X,2)
    b_sum{i} = zeros(length(X{i}),1);
    for j = 1:length(b_mean)
        b_sum{i} = b_sum{i} + X{i,j}(:,2);
    end
    b_mean{i} = b_sum{i}/length(X);
end

% Calculate voltage
V_mean = cell(size(b_mean));
V_std = V_mean; % Standard deviation
for i = 1:length(b_mean)
    V_mean{i} = b_mean{i}/1023 * 5;
    V_std{i} = std([V{i,1} V{i,2} V{i,3}],0,2);
end

% Create one cell for all data
Y = cell(length(V_mean),1);

for i = 1:length(Y)
    Y{i}(:,1) = X{i}(:,1)/1000*9.81; % Convert to force in N
    Y{i}(:,2) = V_mean{i};
end

% Make one array with all data for graphpad
```

```
Y_gr = zeros(length(Y{1}),1+numel(X));
Y_gr(:,1) = Y{1}(:,1);

count = 1;
for i = 1:3
    for j = 1:3
        count = count+1;
        Y_gr(:,count) = V{i,j};
    end
end

%% Curve fitting
fits = cell(size(Y));
fitteddata = fits;
gof = fits;
x_fit = linspace(Y{1}(1,1),Y{1}(end,1),1000);

for i = 1:length(fits)
    [fits{i}, gof{i}] = fit(Y{i}(:,1),Y{i}(:,2),'power2');
    fitteddata{i} = fits{i}(x_fit);
end

%% Plot
figure
hold on
% colors = zeros(3:3); % Array for saving color data
colors = [0 0.447000000000000 0.741000000000000;
    0.929000000000000 0.694000000000000 0.125000000000000;
    0.466000000000000 0.674000000000000 0.188000000000000];
transp = 1; % Transparency

% Plot all data
for i = 1:3
    if i == 1
        plot(Y_gr(:,1),Y_gr(:,1+i),'.','color',[colors(1,:) transp], ...
            'displayname','Before curve')
        plot(x_fit,fitteddata{i},'Color',colors(1,:),'DisplayName', ...
            'Power law fit')
        plot(Y_gr(:,1),Y_gr(:,4+i),'.','color',[colors(2,:) transp], ...
            'displayname','On curve')
        plot(x_fit,fitteddata{i+1},'Color',colors(2,:),'DisplayName', ...
            'Power law fit')
        plot(Y_gr(:,1),Y_gr(:,7+i),'.','color',[colors(3,:) transp], ...
            'displayname','After curve')
        plot(x_fit,fitteddata{i+2},'Color',colors(3,:),'DisplayName', ...
            'Power law fit')
    else
        plot(Y_gr(:,1),Y_gr(:,1+i),'.','color',[colors(1,:) transp], ...
            'HandleVisibility','off')
        plot(Y_gr(:,1),Y_gr(:,4+i),'.','color',[colors(2,:) transp], ...
            'HandleVisibility','off')
        plot(Y_gr(:,1),Y_gr(:,7+i),'.','color',[colors(3,:) transp], ...
            'HandleVisibility','off')
    end
end
%
% for i = 1:length(V_mean)
%     % p = plot(Y{i}(:,1),Y{i}(:,2),'*');
%
%     % colors(i,:) = p.Color;
% end

% Plot errorbars
% for i = 1:length(V_mean)
    % errorbar(Y{i}(:,1),Y{i}(:,2),V_std{i},'LineStyle','none','Marker', ...
    %     'none','Color',colors(i,:))
    % errorbar(Y{i}(:,1),Y{i}(:,2),V_std{i},'LineStyle','none','Marker', ...
    %     'none','color','k')
% end



% title('Influence of curved surface on sensor output voltage')
```

```
xlabel('Force [N]')
ylabel('Voltage [V]')
% legend('Before curve (measured)','Before curve (power law fit)', ...
    % 'On curve (measured)','On curve (power law fit)', ...
    % 'After curve (measured)','After curve (power law fit)')
legend show
ax = gca;
ax.YLim = [2.4 5];
```

D.11 Surgeon experiment

```
    %% Surgeon experiment plots
clc
clear all
close all
%% Initialisation

% Read data
for i = 1:3
    % Traction tests
    STD9_T(i,:) = dlmread(['Bryan_STD9_T',num2str(i),'.txt']); %#ok<*SAGROW>
    STD9_T(i+3,:) = dlmread(['Huub_STD9_T',num2str(i),'.txt']);

    STD1_T(i,:) = dlmread(['Bryan_STD1_T',num2str(i),'.txt']);
    STD1_T(i+3,:) = dlmread(['Huub_STD1_T',num2str(i),'.txt']);

    KHO9_T(i,:) = dlmread(['Bryan_KHO9_T',num2str(i),'.txt']);
    KHO9_T(i+3,:) = dlmread(['Huub_KHO9_T',num2str(i),'.txt']);

    % External rotation tests
    STD9_E(i,:) = dlmread(['Bryan_STD9_E',num2str(i),'.txt']);
    STD9_E(i+3,:) = dlmread(['Huub_STD9_E',num2str(i),'.txt']);

    STD1_E(i,:) = dlmread(['Bryan_STD1_E',num2str(i),'.txt']);
    STD1_E(i+3,:) = dlmread(['Huub_STD1_E',num2str(i),'.txt']);

    KHO9_E(i,:) = dlmread(['Bryan_KHO9_E',num2str(i),'.txt']);
    KHO9_E(i+3,:) = dlmread(['Huub_KHO9_E',num2str(i),'.txt']);
end

% Fit coefficients
coeff_d = readmatrix('coeff_d.txt');
coeff_F = readmatrix('coeff_F.txt');

% Unloaded muscle lengts
d0 = readmatrix('d0.xlsx')';

a_d = coeff_d(:,1)';
b_d = coeff_d(:,2)';
c_d = coeff_d(:,3)';

a_F = coeff_F(:,1)';
b_F = coeff_F(:,2)';
c_F = coeff_F(:,3)';

% Putting everything into a cell

% Top row = traction, bottom row = external rotation, columns = configuration
bits = cell(2,3);

bits{1,1} = STD9_T;
bits{1,2} = STD1_T;
bits{1,3} = KHO9_T;

bits{2,1} = STD9_E;
bits{2,2} = STD1_E;
bits{2,3} = KHO9_E;

% Remove time
for i = 1:6
    bits{i} = bits{i}(:,2:end);
```

```
61 end
63 %% Calculation
64 % Create cell for voltage, forces and strains
65 V = cell(2,3);
66 F = cell(2,3);
67 d = cell(2,3);
68 eps = cell(2,3);
70 for i = 1:6
71     for k = 1:6
72         V{i} = bits{i}/1023 * 5; % Calculate voltage
73         F{i}(k,:) = a_F.*V{i}(k,:).^b_F + c_F; % Calculate force
74         d{i}(k,:) = a_d.*V{i}(k,:).^b_d + c_d; % Displacement
75         eps{i}(k,:) = (d{i}(k,:)./d0).*100;
76     end
77 end
79 % Calculate average forces and strains
80 F_mean = cell(2,3);
81 eps_mean = cell(2,3);
83 for i = 1:6
84     F_mean{i} = mean(F{i},1);
85     eps_mean{i} = mean(eps{i},1);
86 end
88 % Calculate standard deviation across each column
89 F_std = cell(2,3);
90 eps_std = cell(2,3);
92 for i = 1:6
93     F_std{i} = std(F{i},0,1);
94     eps_std{i} = std(eps{i},0,1);
95 end
98 %% Plotting
99 % Select muscles for plotting
100 % tractionmuscles = [1 8 11 14];
101 tractionmuscles = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 ...
102     21 22 23 24 25];
103  % Same muscle order as live table
104 order = [1 3 8 12 9 7 2 11 10 4 5 14 17 15 18 20 19 21 6 22 23 24 25];
105 tractionmuscles = tractionmuscles(order);
107 % Create matrices for traction test plotting
108 for i = 1:length(tractionmuscles)
109     for k = 1:3
110         yF_T(i,k) = [F_mean{1,k}(tractionmuscles(i))];
111         yF_T_std(i,k) = [F_std{1,k}(tractionmuscles(i))];
113         yeps_T(i,k) = [eps_mean{1,k}(tractionmuscles(i))];
114         yeps_T_std(i,k) = [eps_std{1,k}(tractionmuscles(i))];
115     end
116 end
118 % Plot force figure traction
119 f = gcf;
120 fdim = [100 100 1200 300]; % Figure dimensions
121 set(f, 'Position', fdim); % Set dimension
122 musclenames = ["Gluteus maximus","Adductor magnus","Gluteus medius",...
123     "Psoas","Iliacus","Sartorius","Adductor longus","Gluteus minimus",...
124     "Adductor brevis","Gracilis","Pectineus","Tensor fasciae latae",...
125     "Obturator externus","Piriformis","Quadratus femoris",...
126     "Obturator internus","Gemelli","Semimembranosus",...
127     "Biceps femoris long head","Semitendinosus",...
128     "Biceps femoris short head","Rectus femoris","Vastus lateralis",...
129     "Vastus intermedius","Vastus medialis"];
130 tractionmusclenames = musclenames(order);
131 b_T = bar(tractionmusclenames,yF_T);
132 hold on
```

```
% Calculate the number of groups and number of bars in each group
[ngroups,nbars] = size(yF_T);

% Get the x coordinate of the bars
x_err_T = nan(nbars, ngroups);
for i = 1:nbars
    x_err_T(i,:) = b_T(i).XEndPoints;
end
x_err_T = x_err_T';

% Plot the errorbars
% errorbar(x_err_T(:),yF_T(:),yF_T_std(:),'k','linestyle','none');

ylabel('Muscle force (N)');
% title('Mean muscle force at subluxation during traction test by surgeons');
legend('STD + 9', 'STD + 1', 'KHO + 9', 'Standard deviation', 'Location',...
    'northeast');
ax = gca;
ax.YLim = [0 14]; % Y limit
hold on

% Make arrays to plot Bryan's and Hub's data
xAllDataPoints = zeros(length(tractionmuscles),9);
yBryan_F_T = xAllDataPoints;
yBryan_eps_T = xAllDataPoints;
yHub_F_T = xAllDataPoints;
yHub_eps_T = xAllDataPoints;

for i = 1:length(tractionmuscles)
    for j = 1:3
        for k = 1:3
            xAllDataPoints(i,(j-1)*3+k) = x_err_T(i,j);
            yBryan_F_T(i,(j-1)*3+k) = F{1,j}(k,tractionmuscles(i));
            yBryan_eps_T(i,(j-1)*3+k) = eps{1,j}(k,tractionmuscles(i));
            yHub_F_T(i,(j-1)*3+k) = F{1,j}(k+3,tractionmuscles(i));
            yHub_eps_T(i,(j-1)*3+k) = eps{1,j}(k+3,tractionmuscles(i));
        end
    end
end

% Plot Bryan's and Hub's data into figure
scatter(xAllDataPoints(:),yBryan_F_T(:),10,'o','MarkerFaceAlpha',0.01)
scatter(xAllDataPoints(:),yHub_F_T(:),10,'^','MarkerFaceAlpha',0.01)
errorbar(x_err_T(:),yF_T(:),yF_T_std(:),'k','linestyle','none');
legend('STD + 9', 'STD + 1', 'KHO + 9', 'Surgeon #1','Surgeon #2',...
    'Standard deviation','Location', 'northeast');

%%

% strain figure traction
figure
bar(tractionmusclenames,yeps_T);
hold on
% Plot Bryan's and Hub's data into figure
scatter(xAllDataPoints(:),yBryan_eps_T(:),10,'o','MarkerFaceAlpha',0.01)
scatter(xAllDataPoints(:),yHub_eps_T(:),10,'^','MarkerFaceAlpha',0.01)
errorbar(x_err_T,yeps_T,yeps_T_std,'k','linestyle','none');
ylabel('Muscle strain (%)');
% title('Mean muscle strain at subluxation during traction test by surgeons');
legend('STD + 9', 'STD + 1', 'KHO + 9','Surgeon #1','Surgeon #2',...
    'Standard deviation', 'Location', 'northeast');
ax = gca;
ax.YLim = [0 35];
f = gcf;
set(f, 'Position', fdim); % Set dimension

% Create matrices for external rotation test plot
extrotationmuscles = tractionmuscles;

for i = 1:length(extrotationmuscles)
    for k = 1:3
        yF_E(i,k) = [F_mean{2,k}(extrotationmuscles(i))];
        yF_E_std(i,k) = [F_std{2,k}(extrotationmuscles(i))];
```

```
208             yeps_E(i,k) = [eps_mean{2,k}(extrotationmuscles(i))];
209             yeps_E_std(i,k) = [eps_std{2,k}(extrotationmuscles(i))];
210         end
211     end
212
213     figure
214     extrotationmusclenames = tractionmusclenames;
215     b_E = bar(extrotationmusclenames,yF_E);
216
217     % Errorbars
218     [ngroups,nbars] = size(yF_E);
219
220     x_err_E = nan(nbars, ngroups);
221     for i = 1:nbars
222         x_err_E(i,:) = b_E(i).XEndPoints;
223     end
224
225     x_err_E = x_err_E';
226
227     % Make arrays to plot Bryan's and Hub's data
228     xAllDataPoints = zeros(length(extrotationmuscles),9);
229     yBryan_F_E = xAllDataPoints;
230     yBryan_eps_E = xAllDataPoints;
231     yHub_F_E = xAllDataPoints;
232     yHub_eps_E = xAllDataPoints;
233
234     for i = 1:length(extrotationmuscles)
235         for j = 1:3
236             for k = 1:3
237                 xAllDataPoints(i,(j-1)*3+k) = x_err_E(i,j);
238                 yBryan_F_E(i,(j-1)*3+k) = F{2,j}(k,extrotationmuscles(i));
239                 yBryan_eps_E(i,(j-1)*3+k) = eps{2,j}(k,extrotationmuscles(i));
240                 yHub_F_E(i,(j-1)*3+k) = F{2,j}(k+3,extrotationmuscles(i));
241                 yHub_eps_E(i,(j-1)*3+k) = eps{2,j}(k+3,extrotationmuscles(i));
242             end
243         end
244     end
245
246     % Force
247     hold on
248     scatter(xAllDataPoints(:),yBryan_F_E(:),10,'o','MarkerFaceAlpha',0.01)
249     scatter(xAllDataPoints(:),yHub_F_E(:),10,'^','MarkerFaceAlpha',0.01)
250     errorbar(x_err_E,yF_E,yF_E_std,'k','linestyle','none');
251     ylabel('Muscle force [N]');
252     % title('Mean muscle force at subluxation during external rotation test by surgeons');
253     legend('STD + 9', 'STD + 1', 'KHO + 9','Surgeon #1','Surgeon #2',...
254         'Standard deviation', 'Location', 'northeast');
255     f = gcf;
256     set(f, 'Position', fdim); % Set dimension
257
258     % Strain
259     figure
260     bar(extrotationmusclenames,yeps_E);
261     hold on
262     scatter(xAllDataPoints(:),yBryan_eps_E(:),10,'o','MarkerFaceAlpha',0.01)
263     scatter(xAllDataPoints(:),yHub_eps_E(:),10,'^','MarkerFaceAlpha',0.01)
264     errorbar(x_err_E,yeps_E,yeps_E_std,'k','linestyle','none');
265     ylabel('Muscle strain (%)');
266     % title('Mean muscle strain at subluxation during external rotation test by surgeons');
267     legend('STD + 9', 'STD + 1', 'KHO + 9','Surgeon #1','Surgeon #2',...
268         'Standard deviation', 'Location', 'northeast');
269     ax = gca;
270     ax.YLim = [0 35];
271     f = gcf;
272     set(f, 'Position', fdim); % Set dimension
273
274     %% Create data table for import into graphpad
275     % Create matrix where every 3 columns represent one muscle in 4
276     % configurations.
277     % Every row is one dataset
278     % F_T_gp = zeros(6,75);
```

```
% for i = 1:25
%     for j = 1:3
%         F_T_gp(:,(i-1)*3+j) = F{1,j}(:,i);
%     end
% end

% Create matrix where every 6 columns represent 6 different points for the
% same muscle. Every row is one configuration.
% F_T_gp = zeros(3,6*25);
% for i = 1:25
%     for j = 1:3
%         F_T_gp() =
%     end
% end

F_gluteusminimus = zeros(6,3);

for j = 1:3
    F_gluteusminimus(:,j) = F{1,j}(:,8);
end

% Traction test
F_T_gp = cell(1,length(tractionmuscles));
eps_T_gp = cell(1,length(tractionmuscles));

for i = 1:length(tractionmuscles)
    for k = 1:3
        F_T_gp{i}(:,k) = F{1,k}(:,tractionmuscles(i));
        eps_T_gp{i}(:,k) = eps{1,k}(:,tractionmuscles(i));
    end
end

% External rotation test
F_E_gp = cell(1,length(extrotationmuscles));
eps_E_gp = cell(1,length(extrotationmuscles));

for i = 1:length(extrotationmuscles)
    for k = 1:3
        F_E_gp{i}(:,k) = F{2,k}(:,extrotationmuscles(i));
        eps_E_gp{i}(:,k) = eps{2,k}(:,extrotationmuscles(i));
    end
end
```

D.12 Additional stability test

```
    %% Stability testing of hip simulator
clc
clear all
close all
%% Initialisation
% Read data
F_T = readmatrix('balancing_data.xlsx','Sheet','F_trac')*9.81; % Calculate force in N
F_ext = readmatrix('balancing_data.xlsx','Sheet','F_ext')*9.81; % Calculate force in N
alfa_ext = readmatrix('balancing_data.xlsx','Sheet','alfa_sub');

% Read coefficients and muscle lengths
coeff_d = readmatrix('coeff_d.txt');
coeff_F = readmatrix('coeff_F.txt');
d0 = readmatrix('d0.xlsx')';

a_d = coeff_d(:,1)';
b_d = coeff_d(:,2)';
c_d = coeff_d(:,3)';

a_F = coeff_F(:,1)';
b_F = coeff_F(:,2)';
c_F = coeff_F(:,3)';

% Read sensor data
sensordata = cell(2,4); % Top row is traction test, bottom row external rotation
```

```
% Put data into cell
sensordata{1,1} = readmatrix("STD9_T.txt");
sensordata{1,2} = readmatrix("STD1_T.txt");
sensordata{1,3} = readmatrix("KHO9_T.txt");
sensordata{1,4} = readmatrix("KLA9_T.txt");
sensordata{2,1} = readmatrix("STD9_E.txt");
sensordata{2,2} = readmatrix("STD1_E.txt");
sensordata{2,3} = readmatrix("KHO9_E.txt");
sensordata{2,4} = readmatrix("KLA9_E.txt");

%% Find peaks
close all
% Find measurement points for traction test by hand

musclenumber = 7; % Muscle number used for finding peaks

% for i = 1:4
%     figure
%     numrows = 1:size(sensordata{1,i},1);
%     % threshold = 100000;
%     % tf = ischange(sensordata{i}(:,9),'Threshold',threshold);
%     % indices = find(tf==1);
%     % mpoints = sensordata{i}(indices,:);
    % % plot(sensordata{i}(:,1),sensordata{i}(:,9),mpoints(:,1),mpoints(:,9), ...
        % % 'ro','MarkerSize',5)
%     plot(numrows,sensordata{1,i}(:,musclenumber+1));
% end

%% Calculation
% Extract points of subluxation from separate file
peaks_T = readmatrix('peaks_sensordata.xlsx','Sheet','trac');
peaks_E = readmatrix('peaks_sensordata.xlsx','Sheet','ext');

% Get relevant rows from sensordata
sensordata_peaks = cell(2,4);

for i = 1:4
    sensordata_peaks{1,i} = sensordata{1,i}(peaks_T(:,i),:);
    sensordata_peaks{2,i} = sensordata{2,i}(peaks_E(:,i),:);
end

for j = 1:8
    sensordata_peaks{j} = sensordata_peaks{j}(:,2:26); % Remove time
end


% Calculate displacement, strain and force
V = cell(2,4);
d = cell(2,4);
F = cell(2,4);
eps = cell(2,4);

for i = 1:8
    for k = 1:10
        V{i} = sensordata_peaks{i}/1023 * 5; % Calculate voltage
        F{i}(k,:) = a_F.*V{i}(k,:).^b_F + c_F; % Calculate force
        d{i}(k,:) = a_d.*V{i}(k,:).^b_d + c_d; % Displacement
        eps{i}(k,:) = (d{i}(k,:)./d0).*100;
    end
end

% Create mean and standard deviation cells for sensor data
F_mean = cell(2,4);
eps_mean = cell(2,4);

F_std = cell(2,4);
eps_std = cell(2,4);

for i = 1:8
    F_mean{i} = mean(F{i},1);
    eps_mean{i} = mean(eps{i},1);
    F_std{i} = std(F{i},0,1);
    eps_std{i} = std(eps{i},0,1);
```

```
100 end

102 % Subtract neutral angle from measured angles
103 alfa0 = 100.7; % Neutral angle measured during experiment
104 alfa_ext = alfa_ext - alfa0;
105 % alfa_ext = deg2rad(alfa_ext); % convert to radians

107 % Calculate external rotation moment
108 r = 0.07; % moment arm in m
109 M = F_ext*r;

111 % Mean and std for measured data
112 F_T_tot_mean = mean(F_T,1);
113 M_mean = mean(F_ext,1);
114 alfa_mean = mean(alfa_ext,1);

116 F_T_tot_std = std(F_T,0,1);
117 M_std = std(M,0,1);
118 alfa_std = std(alfa_ext,0,1);

120 %% Plot
121 % Select muscles for plotting
122 tractionmuscles = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 ...
123     21 22 23 24 25];
124 % Same muscle order as live table
125 order = [1 3 8 12 9 7 2 11 10 4 5 14 17 15 18 20 19 21 6 22 23 24 25];
126 tractionmuscles = tractionmuscles(order);

128 % Create matrices for traction test plotting
129 for i = 1:length(tractionmuscles)
130     for k = 1:4
131         yF_T(i,k) = [F_mean{1,k}(tractionmuscles(i))]; %#ok<*SAGROW>
132         yF_T_std(i,k) = [F_std{1,k}(tractionmuscles(i))];

134         yeps_T(i,k) = [eps_mean{1,k}(tractionmuscles(i))];
135         yeps_T_std(i,k) = [eps_std{1,k}(tractionmuscles(i))];
136     end
137 end

139 % Plot force figure traction
140 figure
141 musclenames = ["Gluteus maximus","Adductor magnus","Gluteus medius",...
142     "Psoas","Iliacus","Sartorius","Adductor longus","Gluteus minimus",...
143     "Adductor brevis","Gracilis","Pectineus","Tensor fasciae latae",...
144     "Obturator externus","Piriformis","Quadratus femoris",...
145     "Obturator internus","Gemelli","Semimembranosus",...
146     "Biceps femoris long head","Semitendinosus",...
147     "Biceps femoris short head","Rectus femoris","Vastus lateralis",...
148     "Vastus intermedius","Vastus medialis"];
149 tractionmusclenames = musclenames(order);
150 b_T = bar(tractionmusclenames,yF_T);
151 f = gcf;
152 fdim = [100 100 1200 300]; % Figure dimensions
153 % fdim = [100 100 1200 350]; % Figure dimensions
154 set(f, 'Position', fdim); % Set dimension
155 ax = gca;
156 Flim = [0 16];
157 axdim = [0.130000000000000 0.323519209935670 0.775000000000000 ...
158     0.601480790064330]; % Plot dimensions
159 % ax.Position = axdim;
160 ax.YLim = Flim;
161 hold on

163 % Calculate the number of groups and number of bars in each group
164 [ngroups,nbars] = size(yF_T);

166 % Get the x coordinate of the bars
167 x_err_T = nan(nbars, ngroups);
168 for i = 1:nbars
169     x_err_T(i,:) = b_T(i).XEndPoints;
170 end

172 % Plot the errorbars
```

```
173 errorbar(x_err_T',yF_T,yF_T_std,'k','linestyle','none','marker', ...
174     'none','CapSize',2.5);
175 ylabel('Muscle force (N)');
176 % ax.XTickLabel = {}; % Remove x axis label
177 % title('Mean muscle force at subluxation during traction test');
179 % Plot total traction force
180 % colors = {"#0072BD","#D95319","#EDB120","#7E2F8E"};
181 % b_T_F_tot = bar("Whole leg",F_T_tot_mean);
182 % for k = 1:4 % Specify color
183 %     b_T_F_tot(k).FaceColor = colors{k};
184 % end
185 % ylabel("Traction force [N]")
186 % set(gca,'YScale','log'); % Logarithmic y axis
188 % Total force error bar
189 % x_err_T_F_tot = nan(size(F_T_tot_mean'));
190 % for i = 1:length(x_err_T_F_tot)
191 %     x_err_T_F_tot(i,:) = b_T_F_tot(i).XEndPoints;
192 % end
194 % errorbar(x_err_T_F_tot',F_T_tot_mean,F_T_tot_std,'k','linestyle','none', ...
195 %     'marker','none')
196 legend('STD + 9', 'STD + 1', 'KHO + 9','KLA + 9','Standard deviation', ...
197     'Location','northeast')
199 % strain figure traction
200 figure
201 b_T = bar(tractionmusclenames,yeps_T);
202 hold on
203 errorbar(x_err_T',yeps_T,yeps_T_std,'k','linestyle','none','marker', ...
204     'none','CapSize',2.5);
205 ylabel('Muscle strain (%)');
206 % title('Mean muscle strain at subluxation during traction test');
207 legend('STD + 9', 'STD + 1', 'KHO + 9','KLA + 9','Standard deviation', ...
208     'Location', 'northeast');
209 f = gcf;
210 set(f, 'Position', fdim); % Set dimension
211 ax = gca;
212 epslim = [0 35];
213 ax.YLim = epslim;
214 % ax.Position = axdim;
215 % ax.XTickLabel = {}; % Remove x axis label
218 % Create matrices for external rotation test plot
219 extrotationmuscles = tractionmuscles;
221 for i = 1:length(extrotationmuscles)
222     for k = 1:4
223         yF_E(i,k) = [F_mean{2,k}(extrotationmuscles(i))];
224         yF_E_std(i,k) = [F_std{2,k}(extrotationmuscles(i))];
226         yeps_E(i,k) = [eps_mean{2,k}(extrotationmuscles(i))];
227         yeps_E_std(i,k) = [eps_std{2,k}(extrotationmuscles(i))];
228     end
229 end
232 figure
233 extrotationmusclenames = musclenames(order);
234 b_E = bar(extrotationmusclenames,yF_E);
236 % Errorbars
237 [ngroups,nbars] = size(yF_E);
239 x_err_E = nan(nbars, ngroups);
240 for i = 1:nbars
241     x_err_E(i,:) = b_E(i).XEndPoints;
242 end
244 % Force
```

```
hold on
errorbar(x_err_E',yF_E,yF_E_std,'k','linestyle','none','CapSize',2.5);
ylabel('Muscle force [N]');
% title('Mean muscle force at subluxation during external rotation test');
legend('STD + 9', 'STD + 1', 'KHO + 9','KLA + 9', 'Standard deviation', ...
    'Location', 'northeast');
% set(gca,'YScale','log'); % Logarithmic y axis
f = gcf;
set(f, 'Position', fdim); % Set dimension
ax = gca;
ax.YLim = Flim;
% ax.XTickLabel = {}; % Remove x axis label
% ax.Position = axdim;

% Strain
figure
bar(extrotationmusclenames,yeps_E);
hold on
errorbar(x_err_E',yeps_E,yeps_E_std,'k','linestyle','none','CapSize',2.5);
ylabel('Muscle strain (-)');
% title('Mean muscle strain at subluxation during external rotation test');
legend('STD + 9', 'STD + 1', 'KHO + 9','KLA + 9', 'Standard deviation', ...
    'Location', 'northeast');
f = gcf;
set(f, 'Position', fdim); % Set dimension
ax = gca;
ax.YLim = epslim;
% ax.Position = axdim;

%%
% Plot total traction force
figure
hold on
colors = {"#0072BD","#D95319","#EDB120","#7E2F8E"};
b_T_F_tot = bar(1,F_T_tot_mean);
ax = gca;
ax.XTickLabel = {}; % Remove x axis label
ax.YLim = [0 100];
for k = 1:4 % Specify color
    b_T_F_tot(k).FaceColor = colors{k};
end
ylabel("Traction force [N]")

% Total force error bar
x_err_T_F_tot = nan(size(F_T_tot_mean'));
for i = 1:length(x_err_T_F_tot)
    x_err_T_F_tot(i,:) = b_T_F_tot(i).XEndPoints;
end

errorbar(x_err_T_F_tot',F_T_tot_mean,F_T_tot_std,'k','linestyle','none', ...
    'marker','none')

% title('Total traction force during subluxation')
legend('STD + 9', 'STD + 1', 'KHO + 9','KLA + 9', 'Standard deviation', ...
    'Location', 'northeast');

%%
%%% Subluxation force and angle during external rotation
% figure
% plot(alfa_ext,M,'*')
% title('Moment and angle required for subluxation')
% xlabel('External rotation angle [deg]')
% ylabel('External rotation moment [Nm]')
% legend('STD + 9','STD + 1','KHO + 9','KLA + 9', 'location','northwest')
% ax = gca;
% axis([0 50 0 1])

% Moment plot
figure
b_M = bar(1,M_mean);
ax = gca;
ax.XTickLabel = {}; % Remove x axis label
ax.YLim = [0 18];
```

```
% title('Subluxation moment during external rotation test')
ylabel('Moment [Nm]')
hold on
% Errorbars
x_err_M = nan(size(M_mean'));

for i = 1:length(x_err_M)
    x_err_M(i,:) = b_M(i).XEndPoints;
end
errorbar(x_err_M,M_mean,M_std,'k','linestyle','none','marker','none')
legend('STD + 9', 'STD + 1', 'KHO + 9','KLA + 9', 'Standard deviation', ...
    'Location', 'northwest');

% Angle plot
figure
b_alfa = bar(1,alfa_mean);
ax = gca;
ax.XTickLabel = {}; % Remove x axis label
ax.YLim = [0 65];
% title('External rotation angle at subluxation')
ylabel('Angle [deg]')
hold on
% Errorbars
x_err_alfa = nan(size(alfa_mean'));

for i = 1:length(x_err_M)
    x_err_alfa(i,:) = b_alfa(i).XEndPoints;
end
errorbar(x_err_alfa,alfa_mean,alfa_std,'k','linestyle','none','marker','none')
legend('STD + 9', 'STD + 1', 'KHO + 9','KLA + 9', 'Standard deviation', ...
    'Location', 'northwest');

%% Statistics
% Make empty arrays with all data points for selected muscles
F_s = cell(2,4);
eps_s = cell(2,4);

% Put selected muscle data points into cells
for i = 1:length(F_s)
    F_s{1,i} = F{1,i}(:,tractionmuscles);
    eps_s{1,i} = eps{1,i}(:,tractionmuscles);
end

for i = 1:length(F_s)
    F_s{2,i} = F{2,i}(:,extrotationmuscles);
    eps_s{2,i} = eps{2,i}(:,extrotationmuscles);
end

% Make empty arrays
F_gr_T = cell(1,length(tractionmuscles));
eps_gr_T = F_gr_T;

F_gr_E = cell(1,length(extrotationmuscles));
eps_gr_E = F_gr_E;

for j = 1:length(tractionmuscles)
    for k = 1:length(F)
        F_gr_T{j}(:,k) = F_s{1,k}(:,j);
        eps_gr_T{j}(:,k) = eps_s{1,k}(:,j);
    end
end

for j = 1:length(extrotationmuscles)
    for k = 1:length(F)
        F_gr_E{j}(:,k) = F_s{2,k}(:,j);
        eps_gr_E{j}(:,k) = eps_s{2,k}(:,j);
    end
end
```

E Arduino code

E.1 Sensor live readout script

```
int r0 = 0;
int r1 = 0;
int r2 = 0;


int s2 = 2;
int s3 = 3;
int s4 = 4;
int s5 = 5;
int s6 = 6;
int s7 = 7;
int s8 = 8;
int s9 = 9;
int s10 = 10;

int count = 0;
int V[25] = {};
int i = 0;

void setup() {
  // put your setup code here, to run once:
pinMode(A0,INPUT);
pinMode(A1,INPUT);
pinMode(A2,INPUT);
pinMode(A3,INPUT);
pinMode(s2,OUTPUT);
pinMode(s3,OUTPUT);
pinMode(s4,OUTPUT);
pinMode(s5,OUTPUT);
pinMode(s6,OUTPUT);
pinMode(s7,OUTPUT);
pinMode(s8,OUTPUT);
pinMode(s9,OUTPUT);
pinMode(s10,OUTPUT);

  Serial.begin(9600);
}

void loop()


{
  float voltage;
   for (count=0; count<=7; count++) {



    // select the bit

    r0 = bitRead(count,0);

    r1 = bitRead(count,1);

    r2 = bitRead(count,2);
```

```
58      digitalWrite(s2, r0);

60      digitalWrite(s3, r1);

62      digitalWrite(s4, r2);



66      //Either read or write the multiplexed pin here
67      voltage = analogRead(A0);

69      //save voltage to array
70      V[count] = voltage;

72      /*voltage = (voltage/1023.0)*5;*/

74    }

76    for (count=0; count<=7; count++) {



80      // select the bit

82      r0 = bitRead(count,0);

84      r1 = bitRead(count,1);

86      r2 = bitRead(count,2);



90      digitalWrite(s5, r0);

92      digitalWrite(s6, r1);

94      digitalWrite(s7, r2);



98      //Either read or write the multiplexed pin here
99      voltage = analogRead(A1);

101     //save voltage to array
102     V[count+8] = voltage;

104     /*voltage = (voltage/1023.0)*5;*/


107   }


110 for (count=0; count<=7; count++) {
```

```
    // select the bit

    r0 = bitRead(count,0);

    r1 = bitRead(count,1);

    r2 = bitRead(count,2);



    digitalWrite(s8, r0);

    digitalWrite(s9, r1);

    digitalWrite(s10, r2);



    //Either read or write the multiplexed pin here
    voltage = analogRead(A2);

    //save voltage to array
    V[count+16] = ;

  }


// Sensor 25
voltage = analogRead(A3);
V[24] = voltage;


Serial.print(millis());
Serial.print(',');
for(i = 0; i <= 24; i++)
{
  Serial.print(V[i]);
  Serial.print(',');
}

Serial.println();

}
```

F Goodness of fit data

Supplementary table 2: Goodness of fit data for the displacement fit.

Muscle	SSE	R^2	DFE	adj. R^2	RMSE
Gluteus maximus	90.67959	0.997261	44	0.997136	1.435583
Adductor magnus	31.81046	0.998385	39	0.998303	0.903135
Gluteus medius	22.44579	0.991391	39	0.99095	0.758639
Psoas	33.69417	0.998679	41	0.998614	0.906537
Iliacus	36.31342	0.997357	44	0.997237	0.908463
Sartorius	126.34	0.997774	41	0.997666	1.75541
Adductor longus	106.3562	0.996952	40	0.9968	1.630615
Gluteus minimus	10.42766	0.993275	164	0.993193	0.252157
Adductor brevis	81.9458	0.996857	43	0.996711	1.380477
Gracilis	8.65361	0.999773	44	0.999763	0.443478
Pectineus	20.86894	0.995219	40	0.99498	0.722304
Tensor fasciae latae	374.8066	0.993959	40	0.993657	3.061073
Obturator externus	4.741267	0.998983	43	0.998936	0.332057
Piriformis	10.81379	0.998799	39	0.998738	0.526571
Quadratus femoris	1.89114	0.99582	39	0.995605	0.220206
Obturator internus	16.77132	0.998341	44	0.998266	0.617387
Gemelli	95.63165	0.990439	997	0.99042	0.309709
Semimembranosus	1464.111	0.997745	997	0.99774	1.211824
Biceps femoris long head	760.9505	0.998867	997	0.998865	0.873636
Semitendinosus	1115.943	0.998549	997	0.998546	1.05797
Biceps femoris short head	1025.027	0.998688	997	0.998685	1.013958
Rectus femoris	90.90709	0.999758	717	0.999757	0.356073
Vastus lateralis	2066.421	0.997774	997	0.997769	1.439666
Vastus intermedius	2323.341	0.997087	997	0.997081	1.526542
Vastus medialis	1935.773	0.997116	997	0.99711	1.393412

Supplementary table 3: Goodness of fit data for the force fit.

Muscle	**SSE**	**R^2**	**DFE**	**adj. R^2**	**RMSE**
Gluteus maximus	585.4393	0.997198	44	0.997071	3.647662
Adductor magnus	386.9374	0.997429	39	0.997298	3.149837
Gluteus medius	285.272	0.995732	39	0.995513	2.704564
Psoas	39.34658	0.998444	41	0.998368	0.979629
Iliacus	22.42361	0.998418	44	0.998346	0.713882
Sartorius	4.918004	0.997045	41	0.996901	0.34634
Adductor longus	103.1794	0.996587	40	0.996417	1.606077
Gluteus minimus	34.29266	0.98439	164	0.984199	0.457276
Adductor brevis	42.58734	0.997868	43	0.997769	0.99519
Gracilis	6.759152	0.997086	44	0.996954	0.39194
Pectineus	9.224861	0.99482	40	0.994561	0.480231
Tensor fasciae latae	12.38301	0.995233	40	0.994995	0.556395
Obturator externus	5.577913	0.99775	43	0.997645	0.360165
Piriformis	5.753341	0.997884	39	0.997775	0.384085
Quadratus femoris	0.72389	0.992331	39	0.991938	0.13624
Obturator internus	2.577841	0.998945	44	0.998897	0.242048
Gemelli	48.18877	0.977587	997	0.977542	0.219849
Semimembranosus	1664.185	0.999218	997	0.999216	1.291972
Biceps femoris long head	1661.42	0.99863	997	0.998627	1.290899
Semitendinosus	300.1456	0.998609	997	0.998607	0.548679
Biceps femoris short head	295.3155	0.998635	997	0.998632	0.544246
Rectus femoris	1114.207	0.993821	717	0.993804	1.246589
Vastus lateralis	22883.04	0.997129	997	0.997124	4.790814
Vastus intermedius	9026.87	0.996447	997	0.99644	3.008992
Vastus medialis	43469.65	0.993446	997	0.993433	6.603064