The design and evaluation of a fully passive variable damping transfemoral prosthesis

ME51035: MSc Thesis

E. S. Vink

Delft University of Technology

TU Delft

The design and evaluation of a fully passive variable damping transfemoral prosthesis

by

E. S. Vink

Main thesis advisor:	G. Smit
Daily thesis advisor:	B. van der Windt
Project Duration:	September, 2024 - April, 2025
Faculty:	Faculty of Mechanical Engineering, Delft

TU Delft

Preface

This thesis marks the end of my master program in Mechanical Engineering for the track BioMechanical Design at Delft University of Technology.

Over the past year I have worked on this research, a year which I very much enjoyed. Every aspect of the project taught me valuable lessons and deepened my passion for academic research and prosthesis design. Throughout this project, everything i wanted to achieve felt challenging in a good way, and I feel like I handled all of them properly. Therefore I am happy to say I am very proud of what I have achieved with this thesis. I truly hope someday this design might somehow help prosthesis users in low-income regions. Additionally, I hope this research serves as a meaningful contribution to the academic community.

This thesis would not be in the current state as it is now without the support of some individuals.

First, I would like to thank my daily supervisor, Bob. The amount of attention you gave me and this project, the insightful meetings together and the collaboration in general were amazing. It was a great pleasure working with you, and I couldn't have hoped for a better daily supervisor.

Secondly, I would like to thank Gerwin, my main supervisor. Your critical questions really pushed me to improve and improve my thesis, and due to that it really feels like I produced something academic. You really taught me how to think critical and academic, and I am very grateful for that.

I could not have done it without my occasional study breaks. Stijn, there will come a day when I fairly beat you in a game of chess. Jenne, I truly hope we will once find the route to the tower of 3mE.

Roommates, thank you for putting up with me throughout my thesis. Your marvelous cooking, our noche de casa and an occasional mariachi song really pulled me through the more intense weeks.

Friends and family, you were amazing as always. Thanks for listening to me talking about prosthetics, even though you probably weren't always as interested as you acted.

Lastly, I would like to thank Reina. I am very happy to have shared the thesis adventure together, and I am proud on how we managed both our thesis and each other. Your support and company really made this year easy and enjoyable for me. Without you, every figure in this thesis would definitely be aesthetically unpleasing (in your opinion).

E. S. Vink
Delft, March 2025

Summary

This research focuses on the design and evaluation of a fully passive variable damping transfemoral prosthesis, aiming to provide a low-cost alternative to existing microprocessor-controlled prosthetic knees (MPKs). Current MPKs, such as the Otto Bock C-Leg 4 and Össur Rheo Knee XC, offer advanced functionality by adjusting knee damping based on sensory input. However, their high cost make them inaccessible to many individuals, particularly in low-income regions. This study addresses this issue by developing a fully passive low-cost transfemoral prothesis with variable knee damping in the swing phase.

The design process involved defining functional requirements, identifying key subfunctions, and generating three concept designs.

The final design provides stability in the stance phase and controlled motion during the swing phase. The main innovation of this design is a centrifugal force-based damping mechanism. The foot's centrifugal acceleration generates a force that is transmitted through a cable system to a friction brake in the knee joint, effectuating velocity-dependent damping. Additionally, knee stability in the stance phase is ensured by a spring latch lock mechanism, which engages at the end of the swing phase and disengages at the end of the stance phase through ankle dorsiflexion. An ankle spring mechanism provides ankle stability in the stance phase, and it provides energy for push-off. The final design is produced using laser-cut metal, 3D-printed components, and off-the-shelf parts to keep production costs low.

The evaluation phase included two key tests:

Swing phase test – Assessed the velocity-dependent damping mechanism by suspending the prosthesis and measuring knee moment at various angular velocities. The results showed some velocity-dependent damping, but inconsistencies due to hysteresis were observed.

Ankle stiffness test – Evaluated the rotational stiffness of the ankle joint using force and displacement measurements. The results showed slightly lower than expected ankle stiffness.

While the prototype shows the feasibility of a fully passive variable damping knee, several limitations were identified. The main limitation was that significant hysteresis was present the damping mechanism, leading to inconsistent results. Furthermore, the swing phase was performed only at walking speeds lower than occur at natural gait, so no conclusion can be drawn on its effectiveness in real world use.

Future research should focus on improving the robustness of the damping mechanism, reducing the hysteresis and conducting full gait cycle tests on human subjects. Additionally, implementing a delay between force sensing and damping activation could enhance the resemblance to natural gait.

In conclusion, this study demonstrates that a low-cost, fully passive variable damping prosthetic knee is possible, demonstrating a promising initial step towards the development of low-cost *K3* prostheses. However, further development and testing is required before the design can be classified as a functional *K3* prosthesis.

Contents

Nomenclature

Abbreviations

Abbreviation	Definition
AI	Artificial Intelligence
CoM	Center of Mass
DoF	Degree of Freedom
FPK	Fluidic Processor Knee
MPK	Microprocessor-controlled prosthetic knee
VGK	Very Good Knee

Symbols

Symbol	Definition	Unit
δ	Displacement	[m]
ϕ_a	Angle between $\vec{g}$ and $\vec{l}_{A-CoM}$	[°]
θ_a	Ankle angle. $0°$ when foot is perpendicular to shank. Positive values indicate dorsiflexion, negative values indicate plantarflexion.	[°]
θ_k	Knee angle. $0°$ when thigh is parallel to shank. Positive values indicate knee flexion.	[°]
θ_s	Angle between suspension points of the ankle spring	[N]
$\vec{g}$	Gravity vector	[m/s^2]
$\vec{l}_{A-CoM}$	Vector from the ankle to the CoM of the foot	[m]
c_d	Damping coefficient	[]
FA	Force amplifier	[]
F_c	Centrifugal force	[N]
F_f	Friction force	[N]
F_l	Leaf spring force	[N]
F_{spring}	Spring force	[N]
k	Stiffness	[N/m] or [N/mm]
K_e	Constant dependent on coil and magnet properties	[$V \cdot s$/rad]
K_t	Torque constant	[Nm/A]
l_s	Spring length	[m] or [mm]
l_{shank}	Distance between knee joint and CoM of the foot	[m] or [mm]
m_b	Body mass	[kg]
m_{foot}	Mass of the foot	[kg]
M_a	Ankle moment	[Nm]
M_d/M_f	Knee moment due to damping	[Nm]
M_{DC}	Torque by DC motor	[Nm]
M_k	Knee moment	[Nm]
r_a	Distance between ankle and ankle spring suspension	[m]
r_k	Distance between knee joint and point on braking surface where friction is applied	[m] or [mm]
R	Resistance	[Ω]
x_m	Magnet displacement	[m]

1

Introduction

1.1. Motivation

Globally, 5-15% of individuals in need of lower-limb prostheses have access to them [1] [2], with an estimated 57.7 million people living with traumatic limb amputations in 2017 [2], and significant access inequalities persisting in low- and middle-income countries due to high costs, limited trained personnel, and infrastructure gaps [3] [4] [5].

The mobility of an individual using a prosthesis is classified using *K*-levels, which define their functional abilities [6]:

K0: The individual does not have the ability to walk safely, with or without assistance.

K1: The individual can walk on level surfaces at a fixed cadence, typically limited to indoor ambulation.

K2: The individual can navigate low-level environmental obstacles such as curbs, uneven surfaces, or stairs.

K3: The individual has the ability to walk with variable cadence, cross most environmental barriers, and move independently.

K4: The individual has the ability for prosthetic ambulation that exceeds ambulation skills, usually applicable to the children, active adults or athletes.

For individuals at the *K3* level, a prosthetic knee must be able to adapt to different walking speeds and environmental conditions. Currently, the most commercially available options for *K3* users are Microprocessor-controlled Prosthetic Knees (MPKs). These semi-active prostheses adjust knee damping based on sensory input to provide smooth and stable gait. While effective, these devices require precision manufacturing and advanced control systems, leading to high production costs [7][8]. As a result, they remain inaccessible to many individuals, particularly those in low-income regions, despite the potential advantages to their mobility, independence, and general health.

1.2. State of the art

The most widely used *K3* knee prostheses are MPKs, which use sensors and microprocessors to dynamically adjust knee damping. Notable examples include:

Otto Bock C-Leg 4[9]: One of the most advanced MPKs, using an advanced microprocessor-controlled hydraulic damping, stumble recovery, and intuitive stance control.

Össur Rheo Knee XC[10]: Knee prosthesis that uses magnetorheological fluid for quick adjustments to walking conditions.

Blatchford Linx[11]: A transfemoral prosthesis with an integrated MPK system with a comparable working principle to the Otto Bock C-Leg 4. Designed for optimal coordination between the knee and foot.

These MPKs offer significant benefits in terms of functionality and adaptability. However, their reliance on electronics, sensors, and precision manufacturing makes them expensive and complex to produce [7][8]. However, there is a cheaper alternative to these MPKs:

Orthomobility Very Good Knee (VGK)[12]: This is a passive device that uses Fluidic Processor Knee (FPK) technology. Instead of using sensors and microprocessors, the VGK uses fluidic sensors and hydraulic damping to automatically adjust to walking conditions. By eliminating electronics, the VGK offers a simpler and more durable alternative to MPKs at a reduced price of at least 3000€. However, despite being more affordable than MPKs, The cost of prosthetic knees is still too high, regardless of the supplier [3].

1.3. Problem statement & Objective

Existing *K3* transfemoral prostheses on the market are expensive and complex, limiting their accessibility to many individuals who would benefit from them. The objective of this research to design a concept for a *K3* transfemoral prosthesis that is fully passive and suitable for variable walking velocities.

2

Gait mechanics

2.1. Natural cadence

Natural cadence refers to the rhythm of movement that moves the body forward while walking on a flat surface [13]. The knee and ankle joints play a crucial role in achieving this cadence. Natural cadence starts with the stance phase, where one leans on the foot and the body pivots over it to move it forward. Then the ankle provides push-off and the swing phase is started. The leg swings forward, and when sufficiently far, the foot is lowered to the ground, and the cycle returns to the stance phase [13]. Figure 2.1 shows a natural human gait cycle.

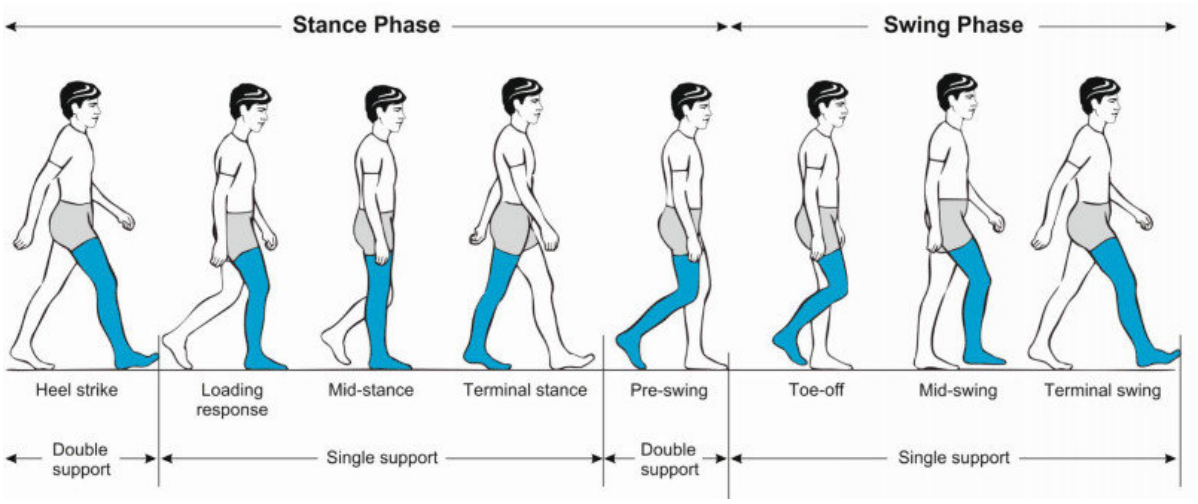

Figure 2.1: Overview of the human gait cycle. Image from Pirker et al. [14].

2.1.1. Function of knee

During the stance phase, the knee provides shock absorption and prevents collapse by increasing its stiffness, therefore providing stance stability [15]. At the end of the swing phase, the knee provides negative work to slow down the shank with respect to the thigh, so that the foot can be properly placed on the ground [16].

2.1.2. Function of ankle

As the stance phase progresses, dorsiflexion increases and the moment in the ankle increases with it, providing stance stability. At the end of the stance phase, the ankle plantarflexes using a burst of positive work to effectuate push-off [16]. The leg is pushed forward into the swing phase. During the swing phase, the ankle ensures foot clearance, preventing the foot from touching the ground and potentially stumbling [13].

2.2. Effect of walking velocity

Walking velocity has an effect on joint mechanics, especially on the moment the knee exerts during the stance phase and the late swing phase. Figure 2.2 demonstrates the effect of walking velocity on joint angles. Similarly, Figure 2.3 shows the moments at different walking velocities.

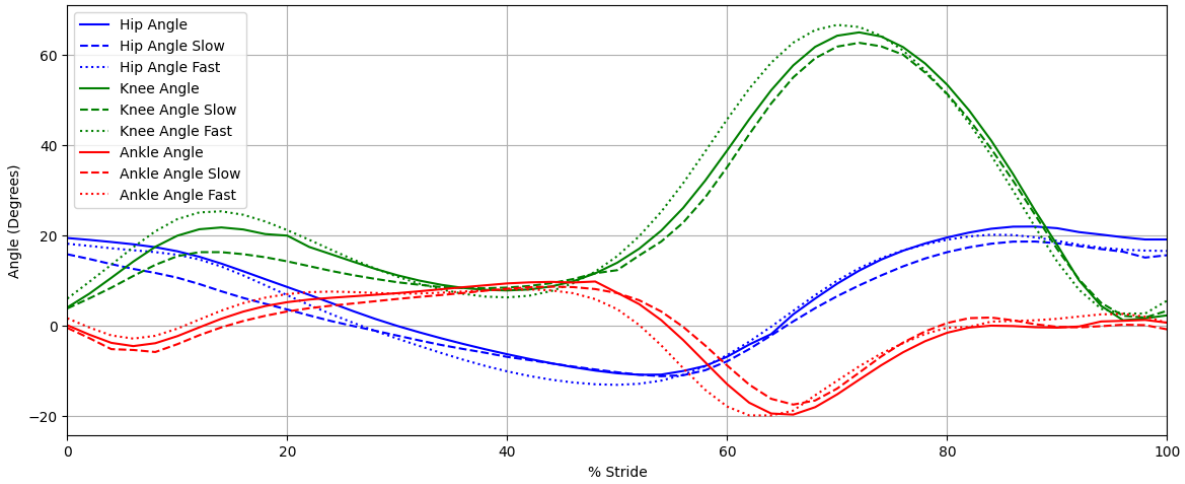

Figure 2.2: Joint angles for slow, normal and fast cadence, which correspond to a cadence of 88, 107 and $125 steps/min$, or 3.7, 4.5, and $5.3 km/h$ respectively. Data by Winter et al. [13]

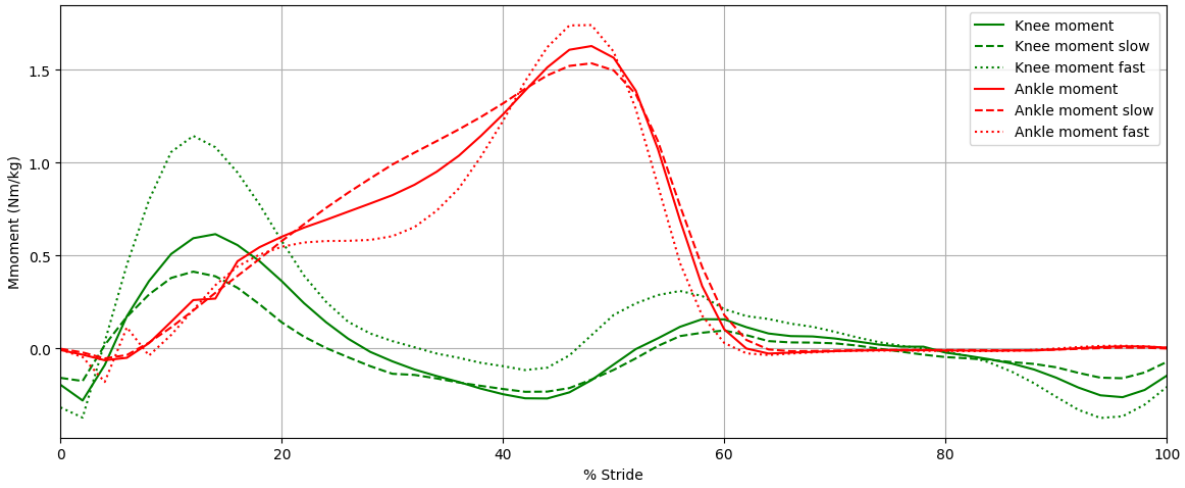

Figure 2.3: Joint moments for slow, normal and fast cadence, which correspond to a cadence of 88, 107 and $125 steps/min$, or 3.7, 4.5, and $5.3 km/h$ respectively. Data by Winter et al. [13]

The unit for moment in Figure 2.3 is expressed as Nm/kg, representing moment normalized by body weight. For the remainder of this thesis, moments will be presented in Nm, assuming a standard body weight of $80 kg$.

2.3. Effect of different terrains

The human gait cycle behaves differently on uneven terrains. During stair ascent, greater peak knee flexion occurs, accompanied by increased knee flexion and ankle plantarflexion moments to support propulsion [17, 18]. In contrast, stair descent requires higher mid-stance knee flexion for controlled movement, along with increased knee flexion and ankle dorsiflexion moments for deceleration [18]. Similarly, walking on an uphill incline results in greater peak knee flexion and increased dorsiflexion [17]. Downhill walking involves higher mid-stance knee flexion for controlled descent [17], as well as increased knee flexion and ankle dorsiflexion moments to control decent [18].

3

Design process

3.1. Design requirements

To reach the objective of this research, the design requirements are set in five categories: Functional requirements, optional functional requirements, constraints, performance criteria and specifications. The functional requirements represent the core features that the design must have, while optional functional requirements are preferred but not essential features. The constraints set the limits for the design. The performance criteria are the measurable metrics used to assess the design. The specifications are the predefined standards that the design must meet.

Functional requirements

- No moment is provided by an external power source.
- Velocity-dependent variable damping in the knee in the swing phase.
- Stable stance phase, which means that the knee should not show excessive flexion or rotation, and the ankle should not show excessive dorsiflexion, plantarflexion, eversion, or inversion.

Optional functional requirements

- Energy harvesting mechanism to provide energy for the variable damping mechanism.
- Push-off mechanism in the ankle to initiate the swing phase.
- Walking terrain-dependent damping in the knee.

Constraints

- The knee and ankle joints have 1 Degree of Freedom (DoF).
- Maximum knee weight of $1255g$ excluding prosthetic tube, which is the weight of the Otto Bock C-Leg 4 [9].
- Maximum foot weight of $800g$.
- Maximum circumference of $40cm$, which is the circumference of the opening of a standard set of dress pants [19].
- Maximum knee moment of $30Nm$ in the swing phase, which is the highest moment in natural cadence. This can be seen in Figure 2.3.
- Maximum ankle moment of $144Nm$, which is the highest moment in natural cadence. This can be seen in Figure 2.3.
- Maximum cost of 1000€.

Performance criteria

- *Natural gait resemblance:* This is assessed by comparing the angle and moment data of the design with the data of Winter et al. [13], which can be seen in Figures 2.2 and 2.3.
- *Range of walking velocities:* This is assessed by comparing the damping for various angular velocities of the knee with the velocity dependence of the data of Winter et al. [13], which can be seen in Figure 2.3.
- *Range of walking terrains:* This is assessed by comparing the behavior of the design with the behavior of a natural leg in various terrains, which is described by Gu et al. [17] and Jichuan et al. [18].
- *Stance stability:* This is assessed by the chance of knee or ankle collapse.
- *Cost.*
- *Safety:* This is assessed by predicting the safety hazards.
- *Noise.*
- *Push-off force:* This is assessed with a force measurement.
- *Reproducibility:* This is assessed by the amount and complexity of off-the-shelf and precision manufacturing parts.

Specifications

- Includes knee and ankle joint.
- Length of standard shank, which is $467 \pm 40mm$ for a Dutch person between 20 and 30 years old [20].
- Length of standard foot, which is $253 \pm 20mm$ for a Dutch person between 20 and 30 years old [20].
- Knee flexion angles between $0°$ and $120°$.
- Ankle angles between $-20°$ plantarflexion and $10°$ dorsiflexion.
- Fits onto a standard prosthesis pyramid connector.

3.2. Sub-functions & sub-solutions

The design should be able to perform certain subfunctions. The subfunctions are divided into main subfunctions and subfunctions for the system that controls the variable damping.

The main subfunctions of the design are:

- Knee damping in the swing phase
- Weight acceptance
- Push-off mechanism
- Energy harvesting medium

The subfunctions of the variable damping control system are:

- Type of measurement of gait phase
- Type of measurement of walking velocity
- Sensor type
- Transfer measurement

An overview of solutions for every subfunction can be seen in the morphological chart in Table A.1.

3.3. Concept designs

3.3.1. Concept 1: Centrifugal force brake

The first concept consists of a mass that is suspended by a spring in the lower part of the prosthesis within a control chamber, as seen in Figure 3.1. The mass is connected by wire to a friction brake

that provides torque to the knee joint. Due to its inertia, in the early swing phase the mass m is pulled back into the cavity in the back of the control chamber, as is the state in Figure 3.1. The main working principle of this concept is that when in this cavity, the mass can be pulled down due to the centrifugal force that occurs due to rotation of the hip and knee. As the mass is pulled down, the friction brake on the knee joint is activated.

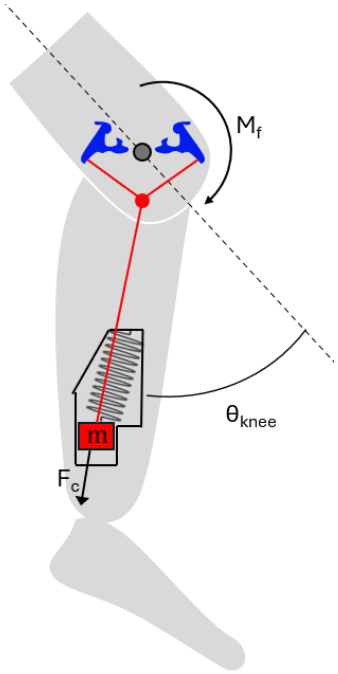

Figure 3.1: Concept 1. The centrifugal force in the shank is coupled to a friction brake in the knee to provide velocity-dependent knee damping.

With an increased walking speed, the centrifugal force F_c in the shank is higher, and therefore more braking torque is applied. This can be seen in the last 20% of the stride in Figure 3.2.

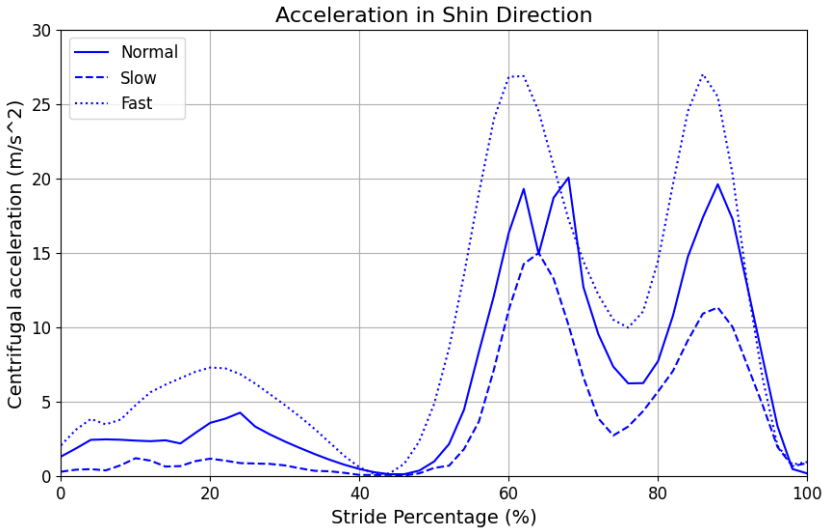

Figure 3.2: Acceleration due to rotational movement for one gait cycle at slow, normal and fast walking velocities, which correspond to a cadence of 88, 107 and $125 steps/min$, or 3.7, 4.5, and $5.3 km/h$ respectively. The acceleration is measured in the lower part of the shank and with respect to the hip. This figure is created by processing data by Winter et al. [13]

When neglecting the gravitational force and the force of the spring by which the mass is hanging,

the moment M_f the friction brake provides through the friction force F_f is calculated by the following formula:

$$M_f = r_k \cdot F_f = r_k \cdot c_d \cdot F_c = r_k \cdot c_d \cdot m \cdot a_c \tag{3.1}$$

where r_k is the distance between the braking point and the axis of rotation, c_d is the damping coefficient, and a_c is the centrifugal acceleration.

The stance stability in the knee is provided by the same friction brake used in the swing phase. As the force F_b due to the body weight m_b is applied to the lower side of the foot, a button is pressed which is coupled to the brake; therefore

$$M_f = r_k \cdot F_f = r_k \cdot c_d \cdot F_c = r_k \cdot c_d \cdot m_b \cdot g. \tag{3.2}$$

Note that when the user is standing on both legs, this formula does not apply.

3.3.2. Concept 2: 1 Degree of freedom

The main working principle of this concept is that the angles of the knee and ankle are kinematically dependent. The relation between the angle of the knee θ_k and ankle θ_a is

$$\theta_k = f(\theta_a) \vee \theta_a = f(\theta_k). \tag{3.3}$$

The angles of both joints are dependent on the position of a pin that slides through a path in a surface. The shape this path should take is illustrated in Figure 3.3. In this path, the x- and y-direction represent θ_k and θ_a and ankle, respectively. The path in the surface is the same as if one were to plot the joint angles during natural cadence against each other. In other words, if the pin is moved along the one-dimensional path, the joint angles move as if in natural cadence. In addition to θ_k and θ_a, Figure 3.3 also shows the moment these joints apply in natural cadence. To ensure that the prosthesis would follow a natural cadence, these moments should also be present in the prosthesis.

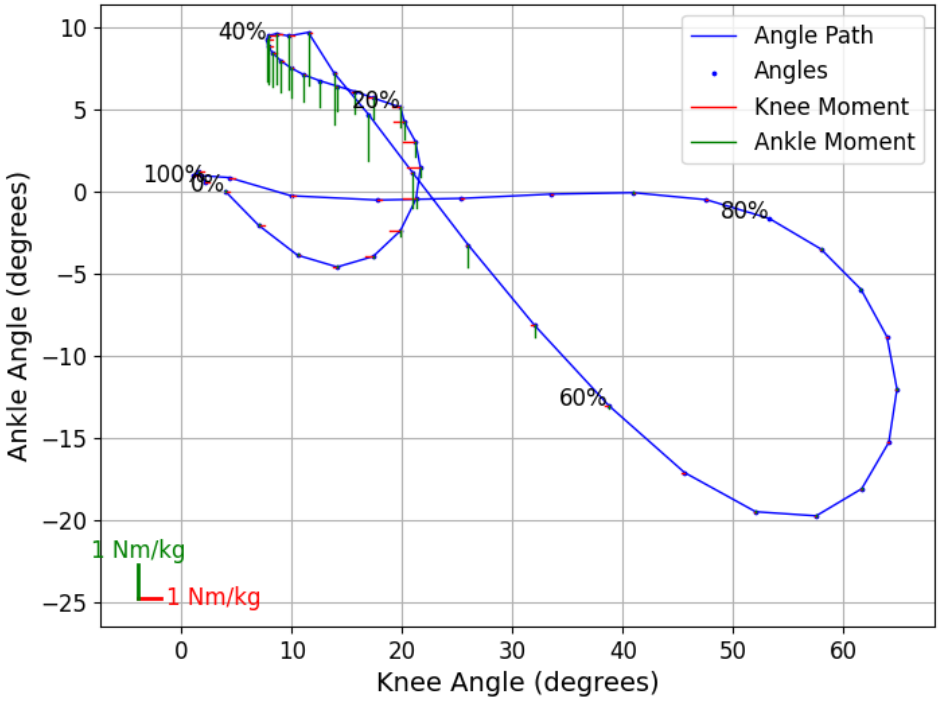

Figure 3.3: Graph showing the knee angle versus the ankle angle and corresponding joint moments for a natural gait cycle with a cadence of $107 steps/min$ and a velocity of $5.3 km/h$ respectively. Contains data which is processed from Winter et al. [13]

Figure 3.4 shows concept 2. As weight acceptance, a spring, marked as **d** in Figure 3.4, is implemented in the x-direction path between 0% and 40% stride in Figure 3.3. This closely approximates the weight acceptance forces in natural cadence [15]. In the swing phase, damping should be applied to the pin.

In the concept, this consists of a frictional damper of which the applied normal force is controlled by an actuator. The value of the normal force applied to the damper is based on the measurement of the sensor of $\dot{\theta}_k$, thus implementing variable damping based on walking velocity.

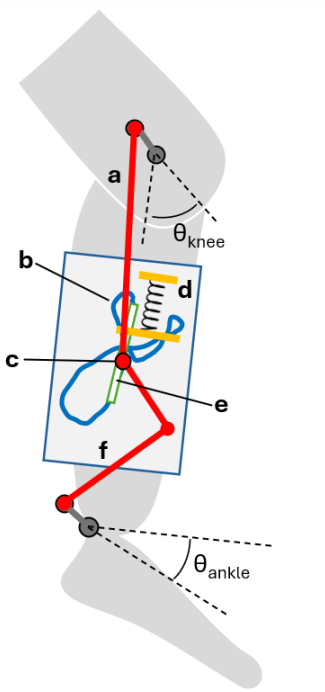

Figure 3.4: Concept 2. **a**: Beam that links the x-position of the pin's path of Figure 3.3 to θ_k. **b**: Path of Figure 3.3 that the pin follows, thereby dictating the movement of θ_k and θ_a. The x-direction is distally and the y-direction is posteriorly. **c**: Pin. **d**: Spring which activates in the stance phase. **e**: Part of path along which damping is applied to the pin by an actuator. The amount of damping is based on information from a sensor in the knee. **f**: Beams that links the y-position of the pin's path of Figure 3.3 to θ_a.

The main design challenge of this concept is that the external forces on the knee and ankle might provide a force in the opposite direction of where the pin should be going, for instance during push-off. This issue might be solved by strategically placing dampers, springs and one-way gates along the path of the pin.

3.3.3. Concept 3: Magnet coil

Figure 3.5 shows Concept 3. This concept consists of a magnet that moves through a coil. The magnet displacement x_m is coupled to θ_k. The coil is put in circuit with a DC motor. When the magnet moves through the coil with a certain velocity $\dot{x}_m$, a voltage V_c is induced, resulting in the torque provided by the DC motor M_{DC}. The following relationship applies:

$$M_{DC} = K_t \cdot \frac{V_c}{R} = \frac{K_t \cdot K_e}{R} \cdot \dot{x}_m \sim \dot{\theta}_k \tag{3.4}$$

where K_t is the torque constant for the motor and K_e is a constant that depends on the coil properties and magnetic field strength, and R is the resistance of the motor windings. As a higher $\dot{\theta}_k$ results in a higher M_{DC}, there is a variable damping system that damps more at higher walking velocities.

In this design, weight acceptance is achieved by activating a leaf spring in the stance phase. As the user stands on their foot, the upper part of the leaf spring is pushed into a locked position with respect to the thigh, activating the spring and providing knee stiffness. A simplified equation for the force of the leaf spring F_l with a spring constant k is given by

$$F_l = k \cdot \delta, \tag{3.5}$$

where δ is the displacement perpendicular to the leaf spring.

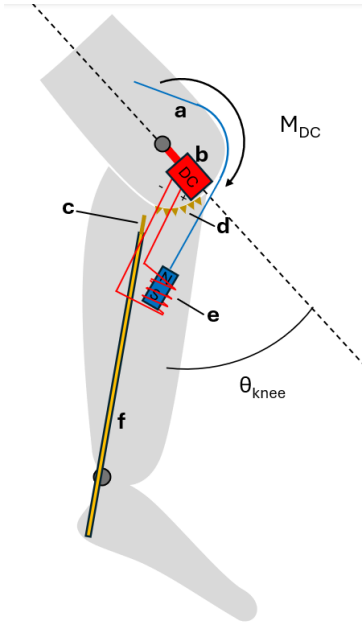

Figure 3.5: Concept 3. **a**: Cable linking θ_k the displacement of the magnet. **b**: DC motor that provides a moment M_{DC} in the knee. **c**: Leaf spring that activates in the stance phase for knee stability. **d**: Teeth in which leaf spring is fixed when activated. **e**: Magnet moving through a coil which connects to the DC motor. **f**: Rod which is pushed upwards in the stance phase, thus activating the leaf spring.

3.4. Concept selection

The concept selection is done by giving each concept scores for weighted performance criteria. The choice of weight according to the criteria is shown in Appendix A in table A.2.

The scores per criterion and the final scores of the concepts are shown in table 3.1. The argumentation for the scores given to the concepts is further discussed in Appendix A. As concept 1: *Centrifugal force brake* has the highest score, this is chosen as the concept to further develop.

Table 3.1: Evaluation of concepts based on performance criteria.

	Weight	**Score (1-10)**	**Total Score**	**Score (1-10)**	**Total Score**	**Score (1-10)**	**Total Score**
Criteria		**Concept 1**		**Concept 2**		**Concept 3**	
How closely does it mimic natural gait?	9	**6**	54	**10**	90	**5**	45
How stable is the weight acceptance?	8	**8**	64	**10**	80	**10**	80
Effective for what range of velocities?	7	**8**	56	**6**	42	**8**	56
Reproducibility	6	**10**	60	**2**	12	**7**	42
Safety	5	**9**	45	**7**	35	**4**	20
Cost	4	**10**	40	**3**	12	**7**	28
Effective for what range of terrains?	3	**6**	18	**2**	6	**7**	21
Push-off force	2	**1**	2	**10**	20	**1**	2
Noise	1	**4**	4	**2**	2	**8**	8
Total Score (Max. = 450)			**343**		**299**		**302**

4

Final design

4.1. Design overview

Figure 4.1 shows the final design in SolidWorks, Figure 4.2 shows a photo and Figure 4.3 shows a lateral cross section, including the cables and springs. The total weight is $3329g$. The mass without thigh is $2581g$, and the distance between the knee and the Center of Mass (CoM) of the design without thigh is $326mm$. The inertia of the shank and foot around the knee joint is $0.375kg \cdot m^2$.

4.1.1. Frame

The thigh, **B** in Figures 4.1, 4.2 and 4.3, is an aluminium X-profile with a brake surface **D** connected to it. The knee joint **C** is a one DoF pivot joint. The shank consists of $1mm$ thick bent stainless steel sheet metal. The ankle joint **J**, in which the shank and foot are connected, is placed posterior and proximal with respect to the foot. The foot consists of two $1mm$ thick stainless steel sheet metal plates which are rigidly connected. Various parts are suspended within the shank and foot.

4.1.2. Variable damping mechanism

Within the variable damping mechanism, the foot functions as a weight which is pulled down during the swing phase due to the rotational acceleration. This pulls the control block, which controls whether or not the prosthesis is in a damping state. When in the damping state, the control block, **I** in Figures 4.1, 4.2 and 4.3, pulls the friction brake arms **F**, which then brake the knee.

As one walks faster, the angular velocity of the knee increases [13], and and thus the centrifugal acceleration in the foot increases. This means the force transferred to the brakes via the control block increases with higher walking velocity. This is the core principle of this design; The centrifugal force is linked to knee damping, thus providing velocity dependent variable damping.

Foot mass

The mass in the foot, **R** in Figures 4.1, 4.2 and 4.3, functions as a sensor which measures the walking velocity. The information of the walking velocity, in the shape of force, is transferred through M_a upwards through a cable to the control block **I**. The mass is placed anterior and distal with respect to the ankle joint so that in the late swing phase, the direction of gravity approaches a 90° angle with respect to the axis of rotation of the ankle. This gives an additional boost of M_a and thus force transferred to the control block as the swing phase progresses.

Control block

The control block, **I** in Figures 4.1, 4.2 and 4.3, is suspended by springs in a control chamber. It is also connected by cables to the foot and friction dampers. The spring stiffness is chosen as such that when the leg hangs down perpendicular to the ground, the control block hovers closely over the balcony of the control chamber, which can be seen as **H** in Figures 4.1, 4.2 and 4.3. The chamber is shaped in such a way that when in the early swing phase, the control block is pushed onto the balcony due to its gravity and the centrifugal force from the foot, as seen in Figure 4.10. As θ_k progresses, the direction

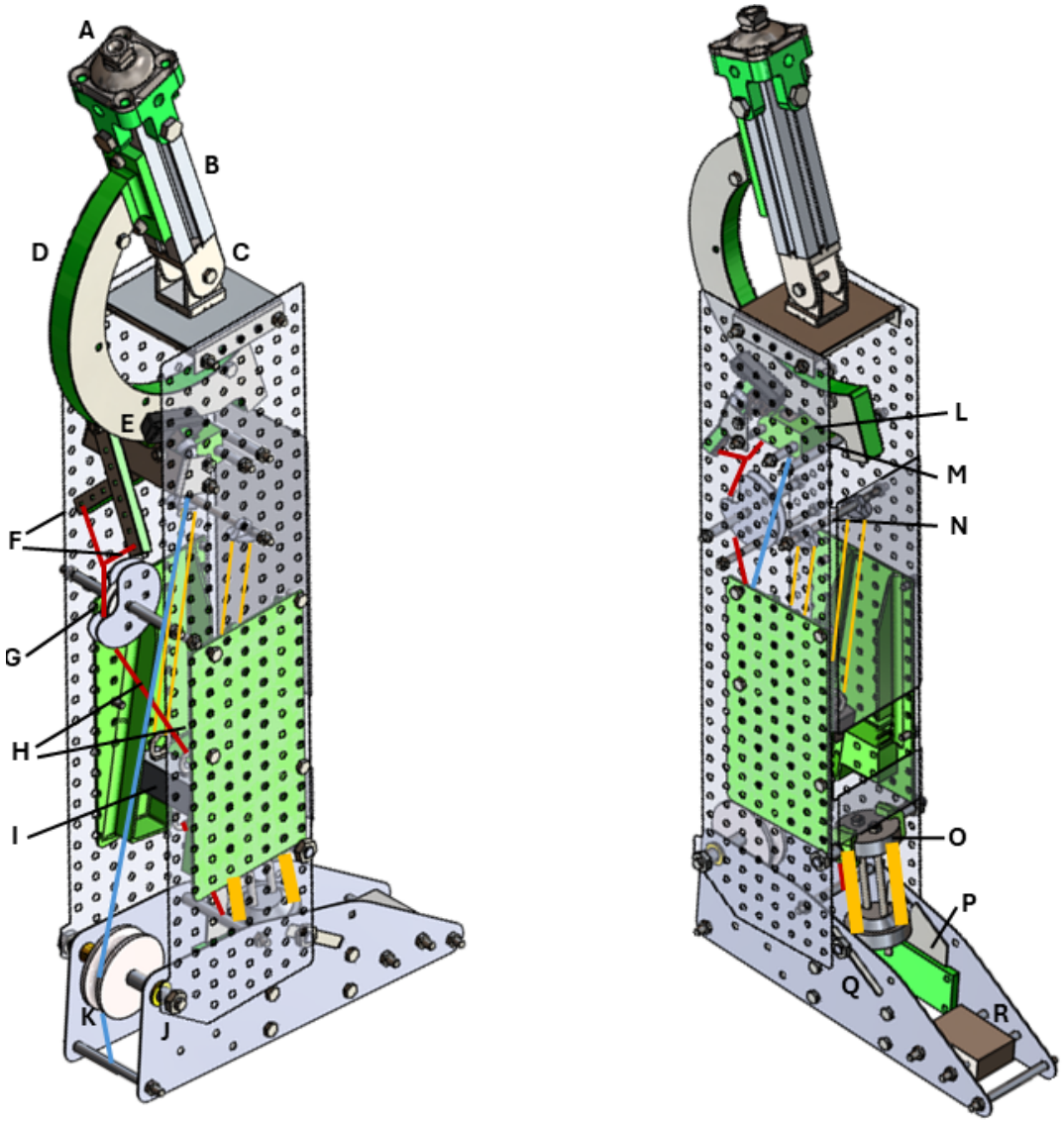

Figure 4.1: An overview of the final design as seen in SolidWorks. The shank is made transparent for a better overview. The red lines are the cables of the variable damping mechanism. The blue line indicates the cable that couples dorsiflexion to the knee lock. The yellow parallel lines show the control block suspension spring and the ankle spring.**A**: Pyramid adapter. **B**: Thigh. **C**: Knee joint. **D**: Brake surface, consisting of a 3D printed center and stainless steel surfaces on the sides. **E**: Friction brake. **F**: Friction damper arms, consisting of 3D printed centers and stainless steel surfaces on the sides. **G**: Pulley guiding the cable from the control block to the friction damper arms. **H**: 3D printed control chamber walls. **I**: Control block, consisting of an aluminium square profile and eye nuts. **J**: Ankle joint. **K**: Pulley guiding cable from foot to knee lock. **L**: Knee lock. **M**: Emergency stop wedge. **N**: Beam at which the control block suspension spring are attached. **O**: Ankle spring suspension. **P**: Ankle spring button. **Q**: Groove through which lower ankle spring suspension can move when the button is not pushed upwards. **R**: Foot mass.

of gravity changes with respect to the control chamber. At a certain point, this gravity, in combination with the centrifugal force of the foot, pushes the control block into the cavity. When in this cavity, the cable towards the friction dampers is tightened, and the centrifugal force that originates from the foot is transferred to the friction dampers, and they activate.

The suspension springs should minimally interfere with the force translation between the centrifugal force and the knee damping, so they should provide minimal additional force as the control mass moves into damping position. This means that the stiffness of the springs should be minimized, and the length maximized. Two Tevema T30927 extension springs with a combined stiffness of $0.1N/mm$ are used.

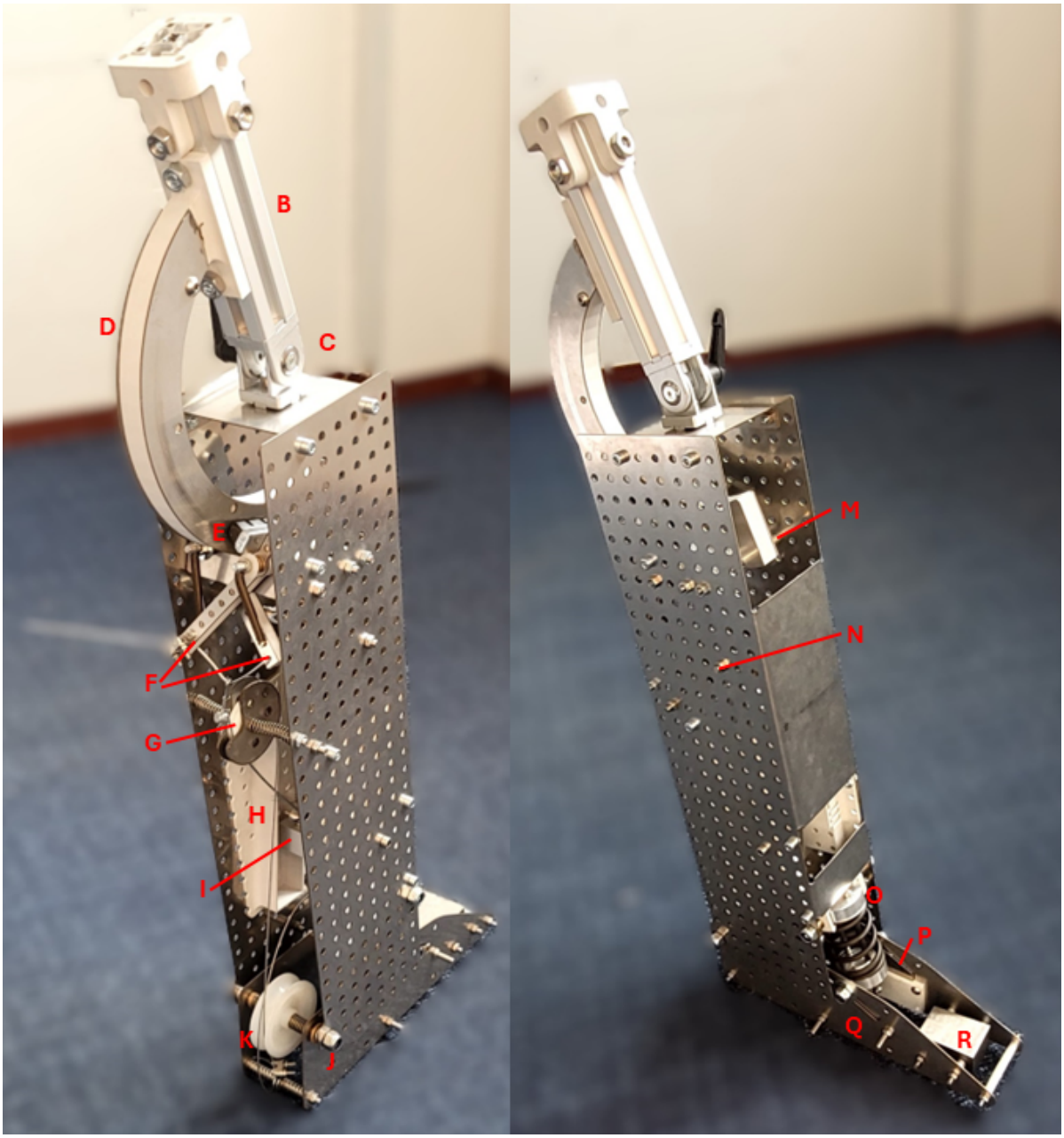

Figure 4.2: A photo of the final design. **A**: Pyramid adapter. Not present in this figure. **B**: Thigh. **C**: Knee joint. **D**: Brake surface, consisting of a 3D printed center and stainless steel surfaces on the sides. **E**: Friction brake. **F**: Friction damper arms, consisting of 3D printed centers and stainless steel surfaces on the sides. **G**: Pulley guiding the cable from the control block to the friction damper arms. **H**: 3D printed control chamber walls. **I**: Control block, consisting of an aluminium square profile and eye nuts. **J**: Ankle joint. **K**: Pulley guiding cable from foot to knee lock. **L**: Knee lock. Not visible in this figure. **M**: Emergency stop wedge. **N**: Beam at which the control block suspension spring are attached in the inside. **O**: Ankle spring suspension. **P**: Ankle spring button. **Q**: Groove through which lower ankle spring suspension can move when the button is not pushed upwards. **R**: Foot mass.

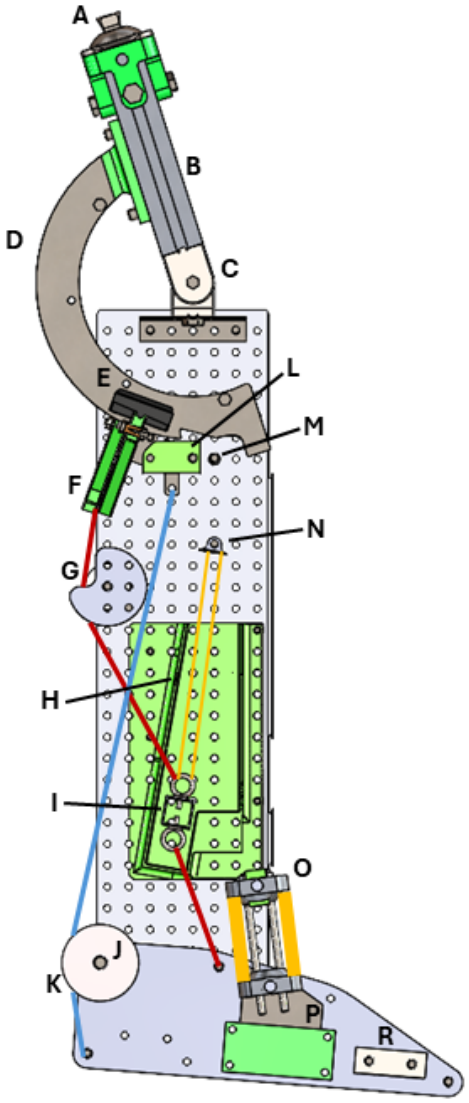

Figure 4.3: A cross section of the final design. The red lines are the cables of the variable damping mechanism. The blue line indicates the cable that couples dorsiflexion to the knee lock. The yellow parallel lines show the control block suspension spring and the ankle spring. **A**: Pyramid adapter. **B**: Thigh. **C**: Knee joint. **D**: Brake surface, consisting of a 3D printed center and stainless steel surfaces on the sides. **E**: Friction brake. **F**: Friction damper arms, consisting of 3D printed centers and stainless steel surfaces on the sides. **G**: Pulley guiding the cable from the control block to the friction damper arms. **H**: 3D printed control chamber walls. **I**: Control block, consisting of an aluminium square profile and eye nuts. **J**: Ankle joint. **K**: Pulley guiding cable from foot to knee lock. **L**: Knee lock. **M**: Emergency stop wedge. **N**: Beam at which the control block suspension spring are attached. **O**: Ankle spring suspension. **P**: Ankle spring button. **Q**: Not visible in this figure. Groove through which lower ankle spring suspension can move when the button is not pushed upwards. **R**: Foot mass.

When the foot gravitational force is exactly compensated, the length of both springs l_s is

$$l_s = \frac{180.0mm}{83.3mm} \cdot 0.6kg \cdot 9.81m/s^2 \cdot \frac{1}{0.1N/mm} = 127.2mm \tag{4.1}$$

assuming the gravitational force on the foot is perpendicular to the point of rotation.

When the control block is in the cavity in a damping state, the ankle angle is at $-15°$, which is a $10.5°$ difference with the neutral state of the ankle when the control block hovers over the balcony. The springs are then lengthened by approximately $83.3mm * sin(10°) = 14mm$. This means in the damping state and when the shank is perpendicular to the level ground, the control block suspension provides a $14mm * 0.1N/mm = 1.4N$ decrease when the force is translated from the foot to the cable

split. This force is then amplified by $\times 13$ due to the force amplifier, so the resulting decrease in M_d is $1.4N \cdot 13 \cdot 0.1m = 1.82N$.

Friction dampers

The damping mechanism consists of two friction damper arms, **F** in Figures 4.1, 4.2 and 4.3, suspended in the shank, and a braking surface **D** that slides between the dampers and connects to the thigh. As the control block is pushed into the cavity, the cable pulls down the secondary cable that connects the two friction damper arms, as can be seen as the red cables near **F** in Figure 4.1. The arms are then pulled together, and the friction dampers clamp on the brake surface. Due to the normal force on the brake surface, which originates from the centrifugal acceleration of the foot, a damping moment is applied to the knee.

Resulting damping moment

To provide a damping moment comparable to that of natural gait, a combination of various parameters could be chosen. The required damping moment can be achieved by applying a friction brake at $r_k = 100mm$ from the knee axis, and applying a force amplifier FA of 26 to the centrifugal force of a mass of the foot $m_{foot} = 0.6kg$. This force amplification is achieved by connecting the cable at $83.3mm$ of the distance from the ankle axis, while the center of mass of the foot is at $180.0mm$. Next, the force is split up to two cables under an angle of $60°$, and lastly the force amplifies through a moment arm, where the cables are connected at $75.3mm$ and the brakes at $12.5mm$. This results in a total force amplification of

$$FA = \frac{180.0mm}{83.3mm} \cdot \frac{1}{cos(60°)} \cdot \frac{75.3mm}{12.5mm} \approx 26 \tag{4.2}$$

This means that, as work $F \cdot s$ should be the same in the whole damping system, the distance the friction brake moves is $\frac{1}{26}$ that of the center of mass of the ankle. The ankle angle is constrained between $-15°$ and $9.5°$, and the center of mass is at $180.0mm$ from the ankle axis, so the displacement range of the friction brakes is $3mm$.

Taking into account all the parts that play a role in the damping system, the formula for knee moment due to damping is

$$M_d = |c_d \cdot r_k \cdot FA \cdot m_{foot} \cdot l_{shank} \cdot \dot{\theta_k}^2 \cdot sin(\phi_a)| - 1.82Nm \tag{4.3}$$

Where c_d is the damping coefficient, r_k is the radius from the knee to the point where the dampers apply friction, FA is the force amplifier, m_{foot} is the mass of the foot, l_{shank} is the distance between the knee and the CoM of the foot, $\dot{\theta_k}$ is the angular velocity of the knee and ϕ_a is the angle between the gravity vector $\vec{g}$ and the direction of $l_{\vec{A-CoM}}$, which is the vector from the ankle point of rotation to the CoM of the foot. The factor $-1.82Nm$ accounts for the losses due to the control block suspension, which is calculated in Chapter 4.1.2. Figure 4.4 shows the feedback control system of the variable damping mechanism.

If the following values are used in Equation 4.3, Figure 4.5 shows the resulting moment in the knee in the late of the swing phase in comparison to the data by Winter et al. [13] for a $80kg$ person. Additionally, Figure 4.6 shows the expected damping moment for certain angular velocities in comparison to the peak moment and peak angular velocity for slow, normal and fast cadence, which is found in the data by Winter et al [13].

- $c_d = 0.7$
- $r_k = 100mm$
- $FA = 26$
- $m_{foot} = 0.6kg$
- $l_{shank} = 495mm$
- $\dot{\theta_k}$ is the angular knee velocities from data by Winter [13] for the late swing phase for slow, normal and fast cadence. This occurs in the last 25% of the gait cycle.
- $\phi_a = 90°$, which is approximately true for the late swing phase.

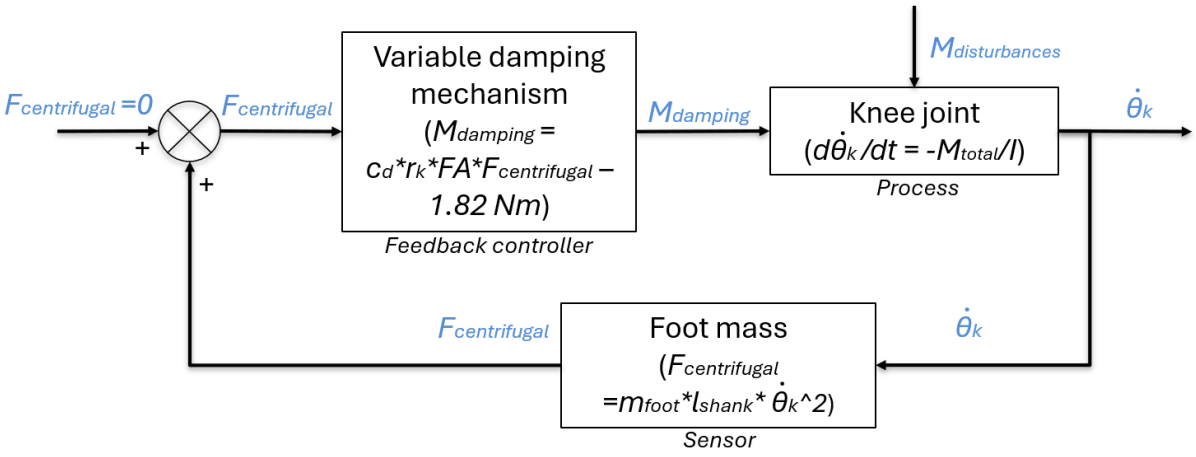

Figure 4.4: The negative feedback control system of the variable damping mechanism. This system applies when the design is in the damping state.

The value for c_d is chosen arbitrarily. The other values are chosen because at these values, the moment of the damping mechanism in the late swing phase is comparable to the data for natural gait, as can be seen in Figure 4.5. Note that this calculation is not the simulation of a dynamic system, but rather an indication of the resulting damping for the angular velocities that occur in the late swing phase for natural gait cycles of slow, normal and fast walking velocities.

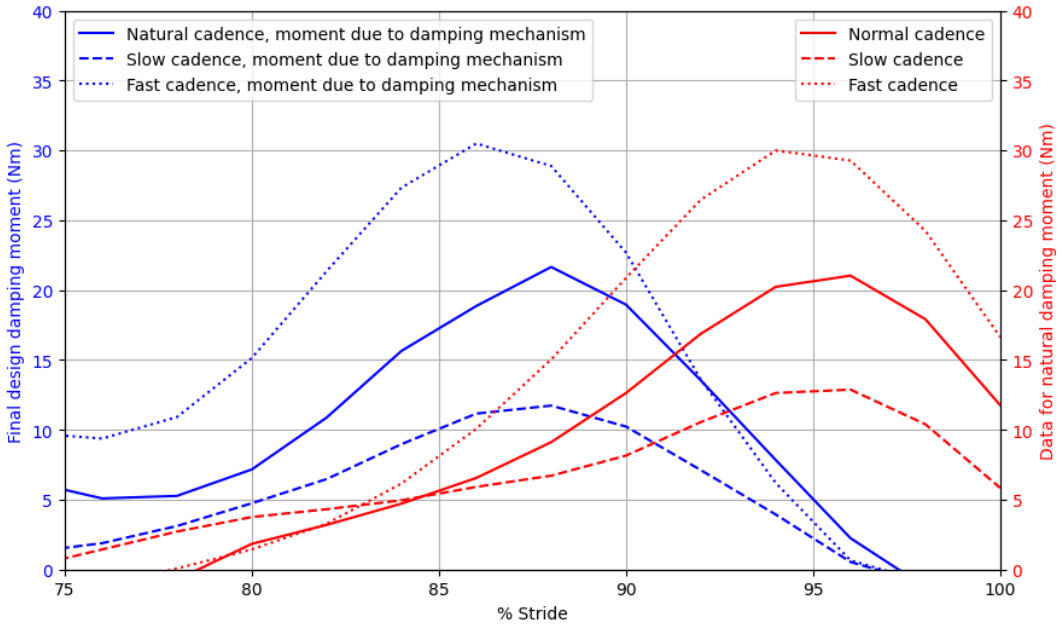

Figure 4.5: Moment due to damping in the late swing phase as found by Winter et al. [13] in red, and an indication for the expected damping in the final design in blue, calculated by Equation 4.3. The slow, normal and fast cadences are 88, 107 and $125 steps/min$, or 3.7, 4.5, and $5.3 km/h$ respectively.

Figure 4.6 shows the expected damping moment for certain angular velocities in comparison to the peak moment and peak angular velocity for slow, normal and fast cadence, which is found in the data by Winter et al. [13].

4.1.3. Ankle spring

Figure 4.7 shows an overview of the ankle spring mechanism. A compression spring, **d** in Figure 4.7, is suspended between the shank and the foot to provide stance stability in the ankle. The lower suspension of the spring can move freely through the groove in the foot plates, **i** in Figure 4.7. When the

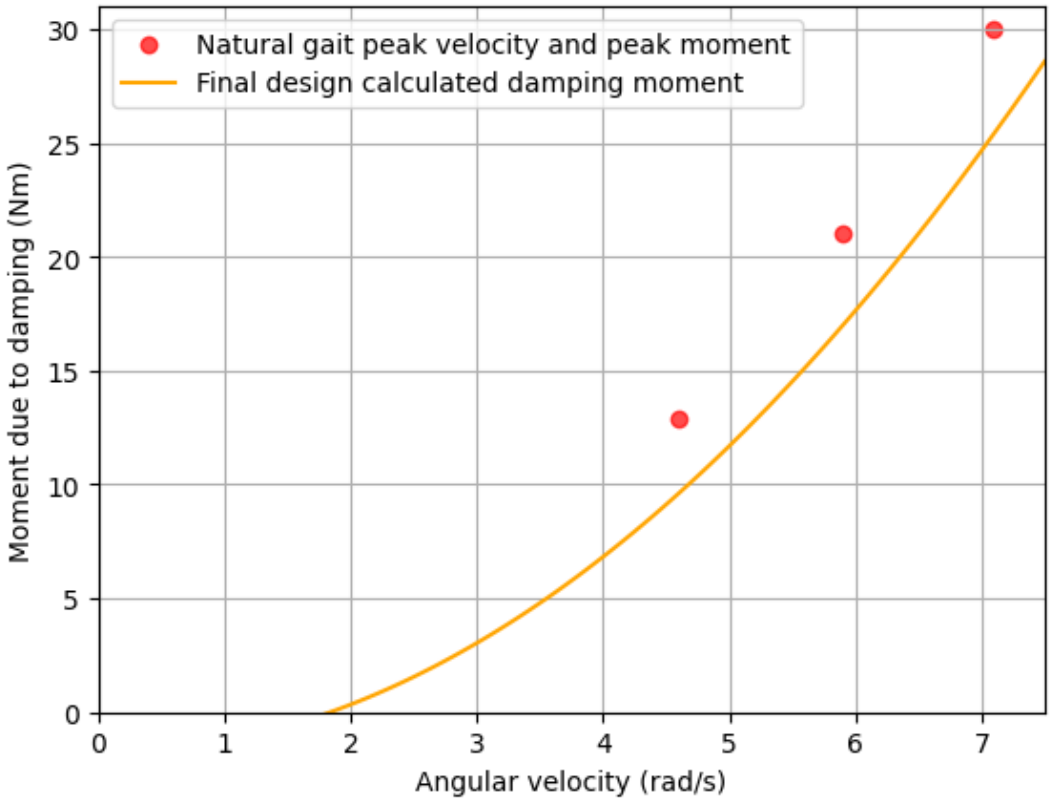

Figure 4.6: Red shows the data by Winter et al. [13] for peak velocity and peak moment in the swing phase for slow, normal and fast cadence, which corresponds to a cadence of 88, 107 and 125 steps$/min$, or 3.7, 4.5, and $5.3km/h$ respectively. Yellow shows the calculated moment of the final design which is calculated with Equation 4.3.

stance phase starts and the foot steps on the ground, two buttons, **P** in Figures 4.1, 4.2 and 4.3, and **f** in Figure 4.7, are pushed upward to limit the movement of the lower suspension within this track. When in this state, the length of the ankle spring is dependent on θ_a. As the stance phase progresses, θ_a and thus M_a increases. Then, as the swing phase is initiated and the foot is lifted, the ankle spring releases its energy and provides push-off. In this situation, the friction force between the lower suspension of the spring and the button ensures that the buttons do not fall down until the spring has released most of its energy.

The stiffness of the ankle is based on the data of normal velocity natural gait. As can be seen in Figure 2.3, the moment varies between 0 and $1.5Nm/kg$ in the stance phase. The point from which the spring starts to provide moment is set at $\theta_a = -4.5°$. At $\theta_a = 9.5°$, M_a should be $\approx 1.5Nm/kg$. The ankle spring provides force at $121mm$ from the point of rotation of the ankle at $\approx 90°$. To get a comparable value for angular stiffness, the spring should therefore provide $\frac{1.5Nm/kg}{14°}$, or for a $80kg$ person $k = 33.6N/mm$.

However, the angle at which the ankle spring provides force on the foot is dependent on the ankle angle itself. Therefore the moment versus ankle angle is not exactly linear, and the expected ankle moment is slightly lower than expected with a spring that provides force at $\approx 90°$. Because of this effect and because lower stressed are preferred, a compression spring with $k = 28.42N/mm$ is chosen. This specific spring, the Tevema D13880, is chosen because of its relatively high inner diameter and wire thickness, which help prevent buckling.

The calculated moment in the ankle due to the ankle spring is

$$M_a = r_a \cdot F_{spring} \cdot cos(0.5 \cdot \theta_s) \tag{4.4}$$

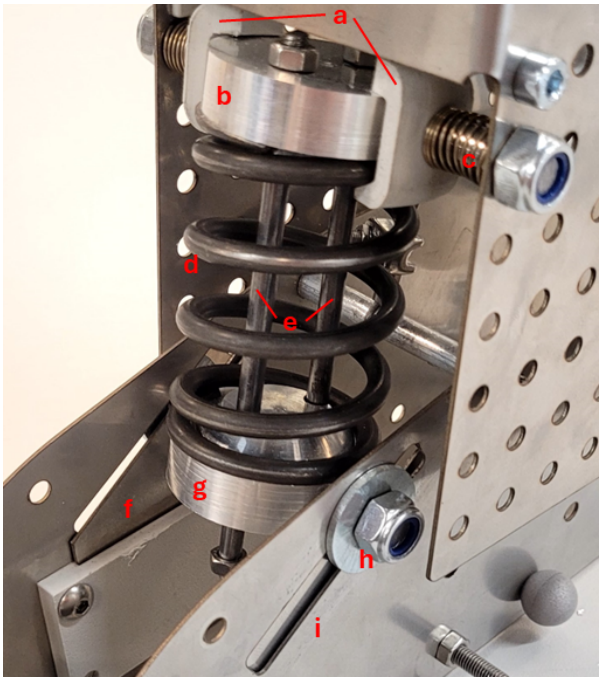

Figure 4.7: Photo of the ankle spring mechanism. **a**: Clamps that fix the spring to the upper suspension. **b**: Upper suspension. **c**: Rotational axis of upper suspension. **d**: Compression spring. **e**: Beams that ensure both suspensions are oriented towards each other. **f**: Spring activation button. **g**: Lower suspension. Rotational axis of lower suspension. This beam can move through the groove when the button is not activated. **h**: Rotational axis of lower suspension. This beam can move through the groove when the button is not activated. **i**: Groove.

, where F_{spring} is dependent on θ_a, so

$$F_{spring} = k \cdot \Delta x = k \cdot r_a \cdot 2 \cdot (sin(0.5 \cdot \theta_s) - sin(0.5 \cdot 29.2°)) \tag{4.5}$$

Where θ_s is the angle between the two suspension points of the spring. $\theta_s = \theta_a + 33.7°$, so $\theta_s = 29.2°$ when $\theta_a = -4.5°$ plantarflexion, the point from which the spring starts to exert force. Figure 4.8 shows the calculated ankle moment, which is calculated using Equation 4.4.

4.1.4. Knee lock

For stance stability in the knee, a lock mechanism is implemented, of which the working principle can be seen in Figure 4.9.

A spring latch, **L** in Figures 4.1 and 4.3, is suspended in the shank in a position under the braking surface which connects to the thigh. A groove is present in the braking surface, such that when the knee is at $11°$, the latch is pushed into the groove and the knee is locked. When in stance phase, as the ankle dorsiflexes a cable that connects the foot to the lock through a pulley in the ankle is pulled. This deactivates the knee lock at sufficient dorsiflexion at the end of the stance phase. The lock remains deactivated until the knee reaches $11°$ degree again at the end of the swing phase, and the lock is pushed into the groove. If the knee overshoots at the end of the swing phase, the brake surface hits an emergency stop wedge, **M** in Figures 4.1, 4.2 and 4.3, at $7°$.

The cable that disengages the lock is coupled to dorsiflexion via a pulley of $r = 22mm$. As θ_a varies between $-4.5°$ and $9.5°$ in the design, this results in a difference in cable length of $5mm$ within a whole gait cycle.

4.1.5. Full gait cycle

Figure 4.10 describes the states of various parts of the design that occur in the design during a full gait cycle.

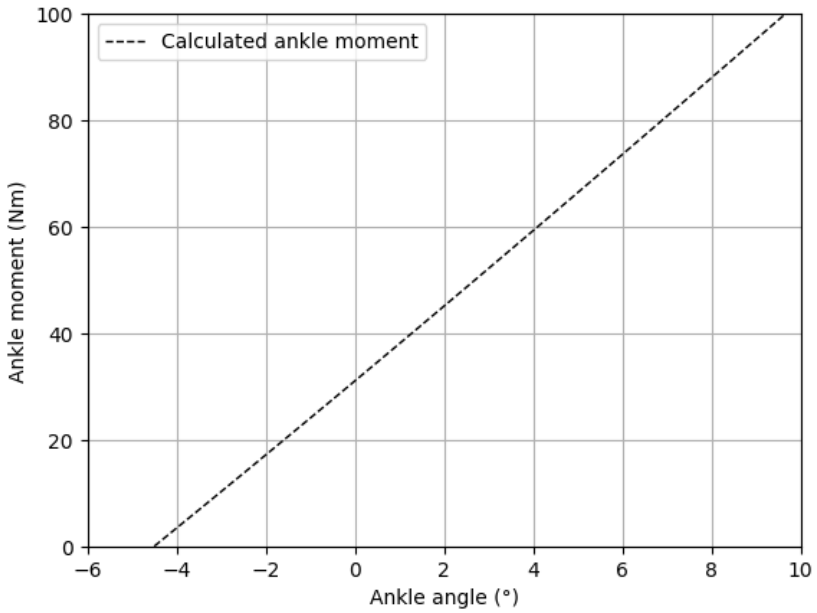

Figure 4.8: Calculated moment in the ankle due to the ankle spring. Positive angles indicate dorsiflexion, negative values indicate plantarflexion. Calculated using Equation 4.4.

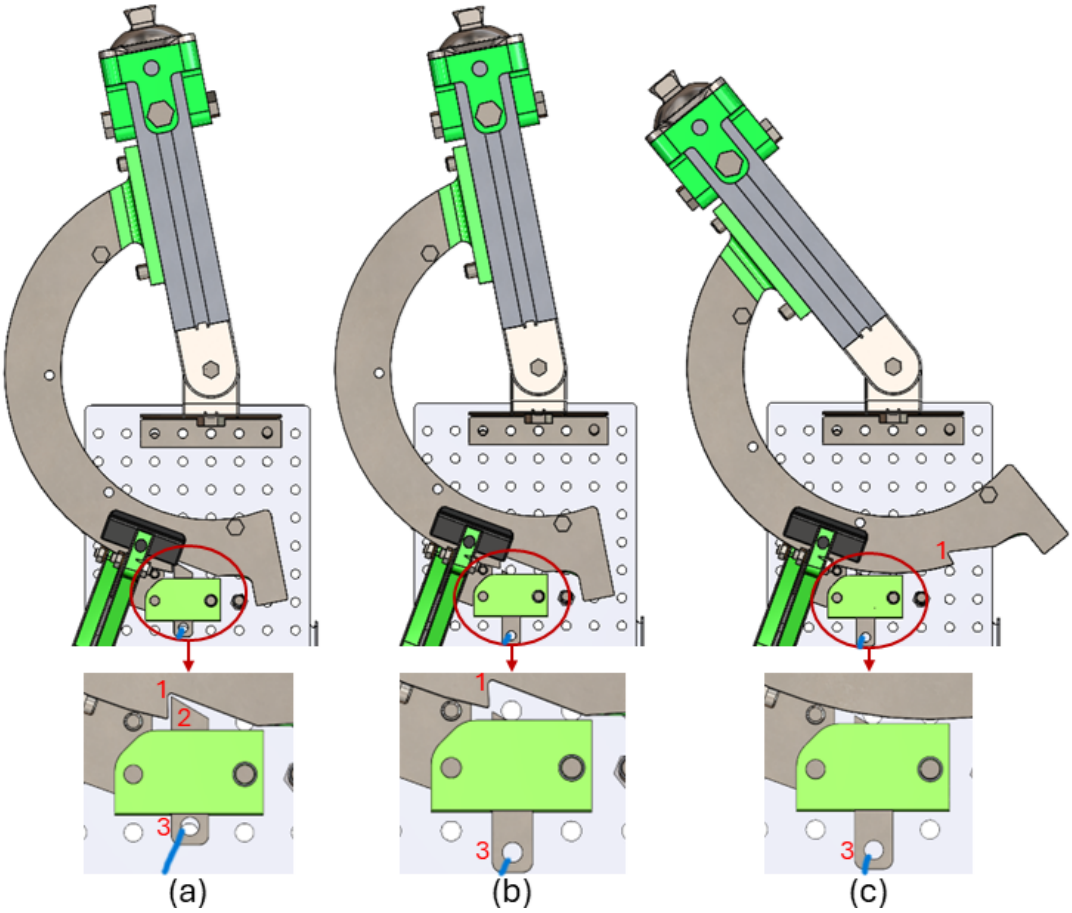

Figure 4.9: The knee lock mechanism in the three possible states. 1: Groove in which spring latch locks during stance phase. 2: Spring latch. 3: Connection point of cable that links dorsiflexion to releasing the spring latch from the groove. (a): The knee is locked during the stance phase. (b): The knee unlocks at the end of the stance phase, as the cable pulls down the spring latch due to dorsiflexion. (c): The knee is unlocked during the swing phase as the spring latch can not enter the groove in the brake surface.

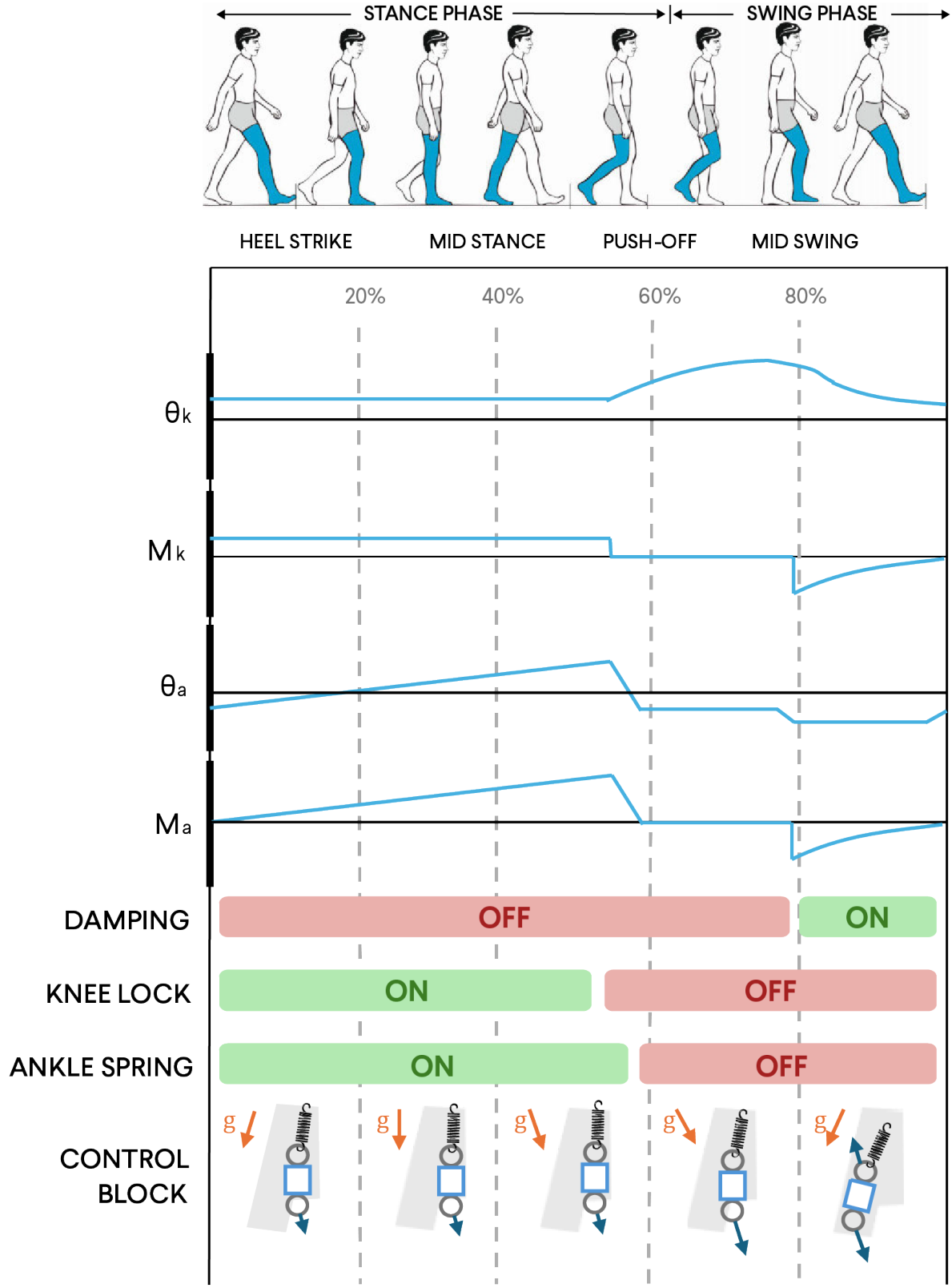

Figure 4.10: An overview of a full gait cycle of the final design. The figure shows an indication for the knee angle θ_k, knee moment M_k , ankle angle θ_a and ankle moment M_a, as well as the state of the damping, knee lock, ankle spring and control block throughout a gait cycle. Image adapted from Pirker et al. [14]

5

Design evaluation

5.1. Swing phase test

To assess the variable damping mechanism, a test is designed that simulates the swing phase of the prosthesis for various walking velocities. for this test, the prosthesis is suspended by a frame in which the thigh is fixed. The test setup is shown in Figure 5.1.

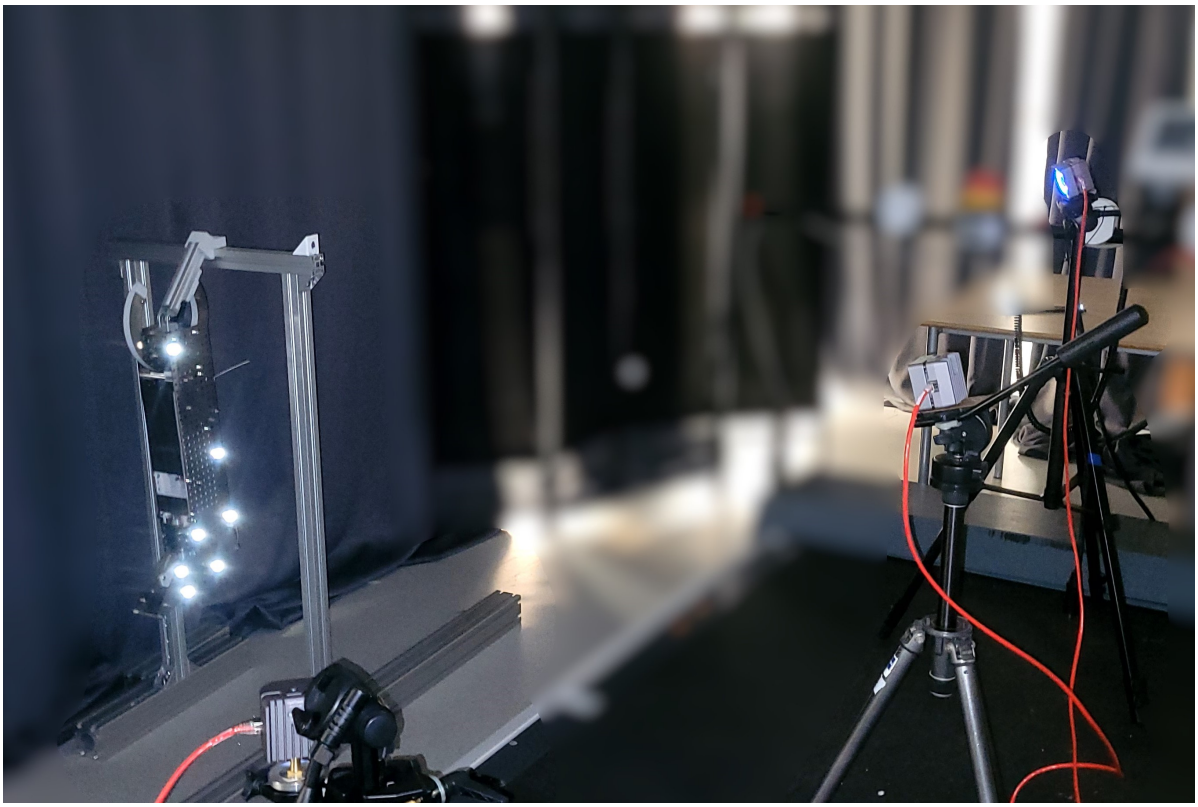

Figure 5.1: The setup for the swing phase test. The shank is given various angular velocities to assess the velocity dependent knee damping of the design. For measuring, markers are placed on the design which can be detected by three cameras. These reflective markers show as bright spots in the image due to the flash of the camera.

In the control chamber, a block is placed on the balcony. This block ensures that the control block can always be pulled into the cavity. This means that when the leg swings, the leg will always enter the damping state when $\dot{\theta}_k$ is sufficient. Additionally, an alternative braking surface is used that does not include an emergency stop wedge, as using this could result in fractures in the 3D printed parts if the

impact is too high.

For the test, the thigh is suspended at a $45°$ angle with respect to the ground. The knee is given various angular velocities from $\theta_k \approx 90°$. The method to accelerate the shank is by giving manual pushes. To assess velocity-dependent damping, the angular velocity of the knee $\dot{\theta}_k$ and the damping moment M_d at a specified θ_k are determined. The specified θ_k, as shown in Figure 5.1, is where the shank hangs still and in equilibrium, so the moment in the knee due to gravity can be considered zero. Data processing is done by first measuring the angles of the prosthesis and then processing the data using Python.

Measurements are made using Motive software in combination with OptiTrack hardware consisting of cameras with a frame rate of $120Hz$ and reflective markers. For the data of the angles, a rolling average filter with a window of 20 is used. Next, for the angular velocity, a rolling average filter with a window of 10 is used. Lastly, a rolling average filter with a window of 2 is used for the angular acceleration, which is then processed further to find the knee moment. The processing is done using a Python code, which can be seen in the Appendix B.1.

5.2. Ankle stiffness test

The rotational stiffness of the ankle joint is evaluated. The knee lock mechanism and control block are disconnected from the foot, so only the moment due to the ankle spring is tested.

This test is done by using a force sensor and displacement sensor on a cable that pulls the foot by the toe, as can be seen in Figure 5.2. By manually controlling a turning wheel, the cable is pulled and the force and displacement in the x-direction are measured using LabView. The data is processed in Python. No filter is used in the processing of the data. The code for the data processing can be seen in Appendix B.2.

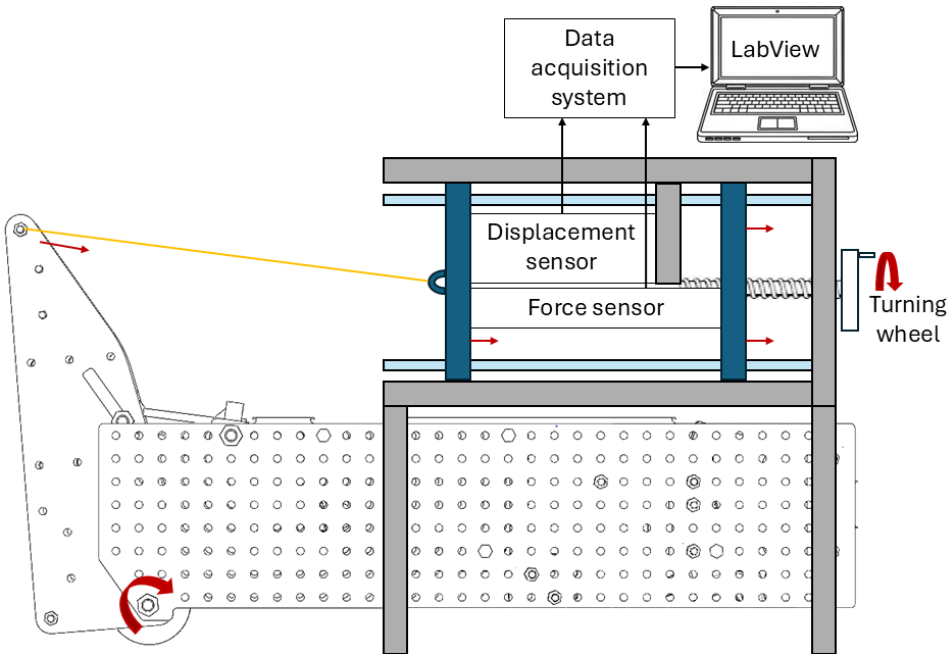

Figure 5.2: A schematic of the setup for the ankle stiffness test. By controlling the turning wheel, the cable connecting to the foot is pulled and the displacement and force are measured.

6

Results

6.1. Swing phase test

Figure 6.1 shows the results of the swing phase test.

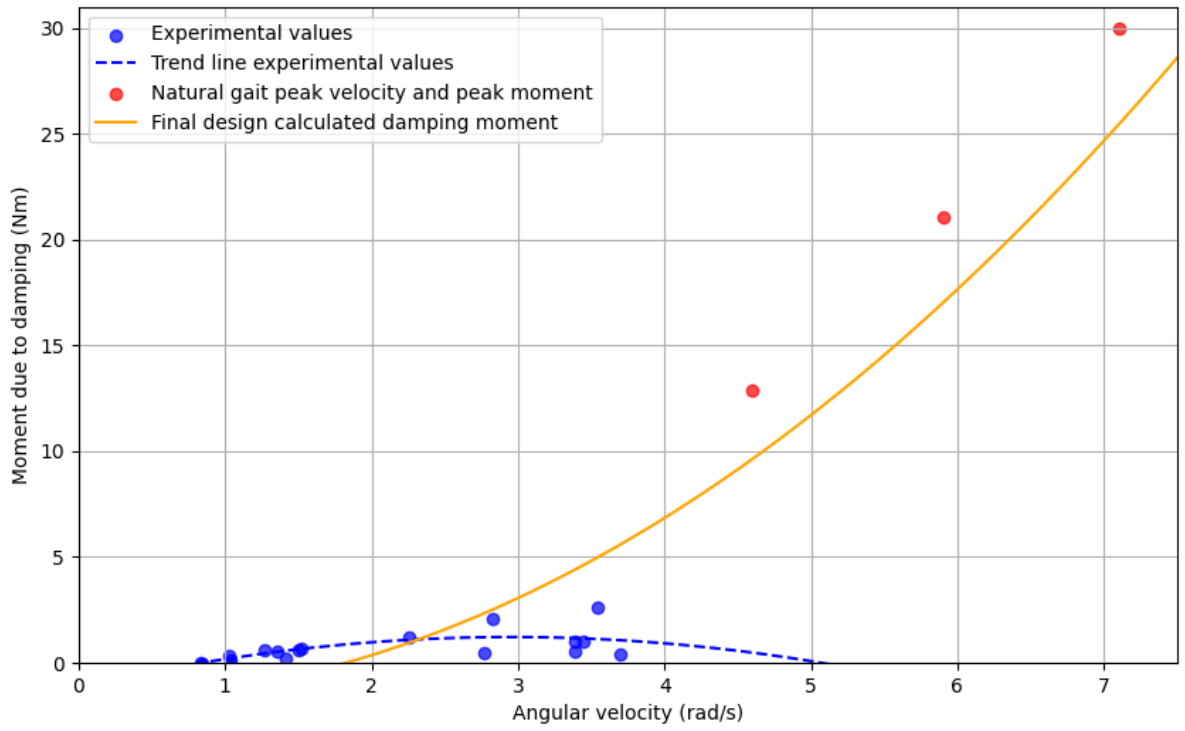

Figure 6.1: Results of the swing phase test. Blue shows the angular velocity and moment of the knee when the shank is perpendicular to the ground. Red shows the data by Winter et al. [13] for peak velocity and peak moment in slow, normal and fast cadence, corresponding to 88, 107 and $125 steps/min$, or 3.7, 4.5, and $5.3 km/h$ respectively. Yellow shows the calculated moment of the final design using Equation 4.3.

The results in Figure 6.1 show the angular velocity and the corresponding moment in the knee, in the instance where it is perpendicular to the ground in the swing phase. Additionally, it shows the calculated moment of the final design which is calculated in Equation 4.3, the data for peak velocity and peak moment in slow, normal and fast cadence. The plot includes a trend line for the experimental data, modelled as a second-order fit.

At angular velocities below $1.8 rad/s$, the measured damping moments remain below $1 Nm$. At higher

angular velocities, some values exhibit similarly low damping moments, while others show an increase. For example, at $2.3rad/s$, the damping moment is $1.2Nm$; at $2.8rad/s$, it is $2.1Nm$; and at $3.5rad/s$, it reaches $2.6Nm$, as shown in Figure 6.1. The fitted trend line forms an inverted parabola, indicating a negative correlation between θ_k^2 and M_d. According to this trend line, the moment would increase with angular velocities up to $3rad/s$ and at higher velocities it decreases.

6.2. Ankle stiffness test

Figure 6.2 shows the result of the ankle stiffness test in comparison to the moment calculated with Equation 4.4.

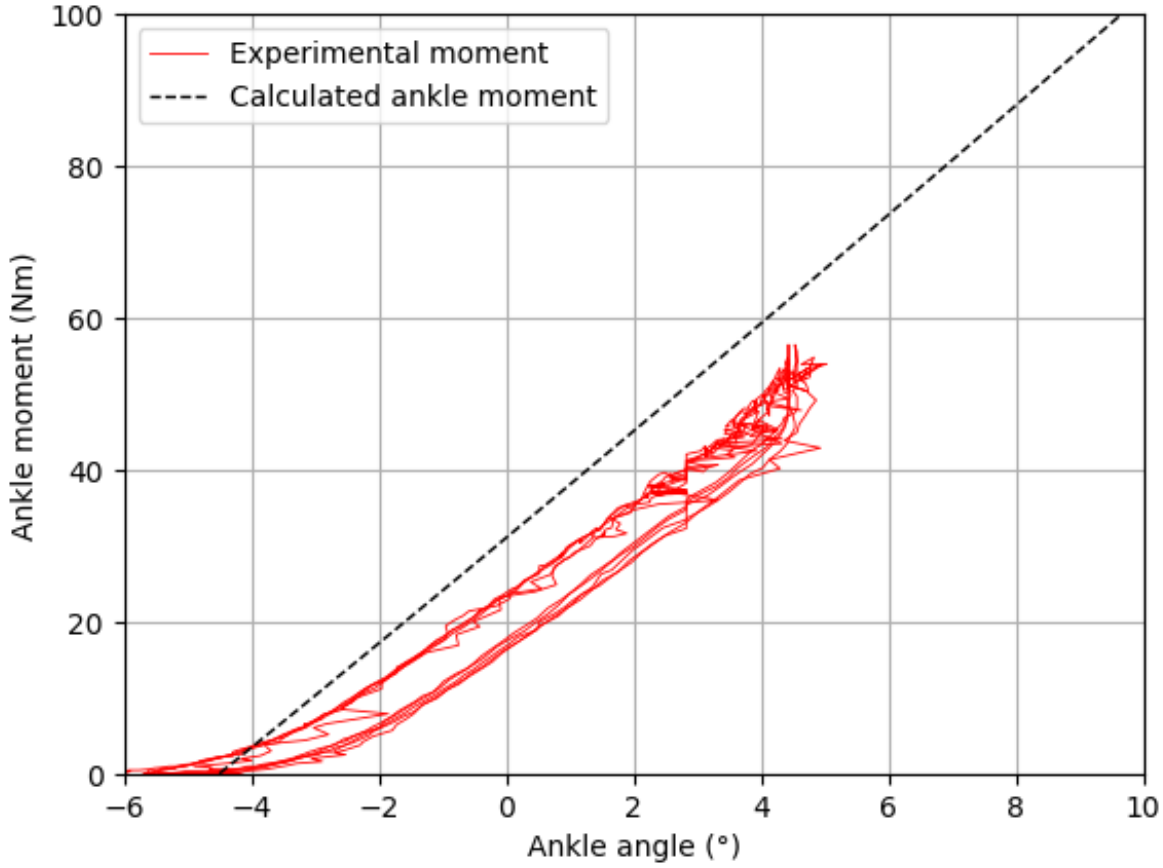

Figure 6.2: Results of the ankle stiffness test in comparison to the calculated moment of Equation 4.4, showing M_a based on θ_a. Plantarflexion gives a negative angle and dorsiflexion gives a positive angle.

The offset of the angle at which the experimental ankle moment is $0Nm$, is arbitrary and chosen to be such that M_a is $3Nm$ at $\theta_a = -4.5°$ plantarflexion. When the measured force exceeds $250N$, the sheet metal plates which make up the foot start to deflect laterally. Therefore the test is performed up to measurements of $200N$.

7

Discussion

7.1. Interpretation of results

7.1.1. Swing phase test

The swing phase test demonstrated that the mechanism provides knee moment up to $0.8Nm$ at angular velocities below $1.8rad/s$, which is the point where the variable damping mechanism starts providing moment according to Equation 4.3. The damping at low angular velocities can be attributed to friction within the design other than the variable damping mechanism. Especially the spring latch mechanism that locks the knee exerts significant friction on the braking surface.

The values for angular velocities higher than $1.8rad/s$ show a different behavior. Some data points show values higher than $0.8Nm$ for the knee moment, indicating that the variable damping mechanism provides friction. These data points are for instance at $2.3rad/s$ & $1.2Nm$, $2.8rad/s$ & $2.1Nm$ and $3.5rad/s$ & $2.6Nm$ in Figure 6.1. However, other data points above $1.8rad/s$ show a knee moment of approximately $0.8Nm$, indicating that the variable damping mechanism does not provide any damping in these cases. These data points are for instance at $2.8rad/s$ & $0.4Nm$, $3.4rad/s$ & $0.5Nm$ and $3.7rad/s$ & $0.4Nm$ in Figure 6.1. The inconsistency between the knee moments at angular velocities higher than $1.8rad/s$ could be attributed to hysteresis within the damping mechanism. Observations suggest that this effect could arise from the pulley that guides the cable from the control block to the damper arms. This problem can be solved by using a pulley with less friction.

7.1.2. Ankle stiffness test

The measured stiffness appears to be slightly lower than anticipated. Additionally, due to deflection in the lateral direction at forces higher than $250N$, tests are not conducted for the full range as specified in Chapter 3.1. For dorsiflexion angles lower than $4°$ the results remain clear and relatively conclusive. However, at angles higher than $4°$ the hypothesis is that the measured stiffness of the ankle spring would be lower due to the lateral deflection of the foot plates. Furthermore, a hysteresis is clearly visible in the data, which can be attributed to a combination of the ankle spring and the hinge of the ankle.

7.2. Limitations and strengths

7.2.1. Design limitations

The following design requirements were not fully met:

- Walking velocity dependent knee damping is not consistently present in the design. As discussed in Chapter 7.1.1, some values indicate its presence and some values do not. This variability can be attributed to hysteresis in the variable damping mechanism.
- Terrain-dependent knee damping is not present, as it is an optional functional requirement. However, since no tests have been conducted on different terrains, no conclusions can be drawn about terrain-dependent behavior.

- The weight of the shank and foot together is $2581g$, which exceeds the maximum value of $2055g$. For the goal of this research this is acceptable, but when the design is further developed, this problem should be solved.
- The circumference of the design is $430mm$, which slightly exceeds the maximum value of $400mm$. This means that the prosthesis does not fit in a regular sized pants. For the goal of this research this is acceptable, but when the design is further developed, this problem should be solved.
- The knee flexion angles vary between $7°$ and $90°$ instead of the specified $0°$ and $120°$. This means that in the late swing phase, the knee has less space to fully decrease its angular velocity, and the moment in the knee will be higher at heel strike as the knee does not extend fully. Additionally, the user can not comfortably sit with the prosthesis, as a comfortable knee flexion angle for sitting is $120°$.
- The ankle plantarflexion reaches $-15°$ instead of the specified $-20°$. However, this is not a significant issue for the design as the full angle is only necessary for a push-off comparable to natural gait, which the design does not provide. In natural gait, the push-off ankle moment is present up to $144Nm$, whereas the design is theoretically capable of $100Nm$ and in practice $56Nm$.
- In practice the ankle dorsiflexion in the stance phase can only occur up to $4°$ instead of the specified $10°$, due to lateral deflection of the foot plates at higher ankle moments.

Various features of the design result in a different gait phase for the design compared to natural gait. One of the main differences is that θ_k does not change during the stance phase, whereas a natural knee can be approximated by a spring which would result in a more natural gait [15]. Additionally, the placement of the ankle joint differs from its natural position, being located more posteriorly. This misalignment could result in unnatural movement patterns or even physiological complications. To counteract this, a higher moment than in a natural ankle may be required to compensate for that. In addition, the design lowers the foot during the swing phase. This deliberate foot drop can cause stumbling, which is undesired. A possible solution for this could be to either limit the allowed plantarflexion in the swing phase while still linking the centrifugal force of the foot to knee damping, or one could link knee damping to a different mass than the foot, so no plantarflexion is required for damping. Another contradictory design feature is the use of the foot's mass. Increasing it benefits the knee damping, however, it conflicts with user comfort, as a lighter foot requires less energy to be moved. Lastly, the design has a different expected knee moment in the late swing phase, which can bee seen when comparing the shapes of M_k in Figure 4.10 and Figure 4.5. The difference in damping pattern results in a higher velocity reduction in the mid-swing phase for the design with respect to natural gait. In the late swing phase, the knee of the design would then show a lower angular velocity compared to natural gait. This might result in stumbling, as the knee takes longer to fully stretch, so the foot hovers closer to the ground. Additionally, the knee might still not being fully stretched at heel strike. In the worst case scenario, the knee might not yet be locked at heel strike and knee collapse occurs.

the shape of M_k in the swing phase in Figure 4.10 shows an instantaneous increase in knee moment as the damping is activated. This could potentially harm the user, as the abrupt force could place excessive strain on the user's residual limb and hip, leading to discomfort, fatigue, or injury. This problem could be solved by implementing a delay between the centrifugal force and the knee damping.

The coupling in the final design between angular velocity and knee damping might result in an interesting effect: As the knee goes faster, the mechanism slows it down, and it decreases the braking itself. This means the mechanism reduces its own effect. This might especially be disadvantageous in lower walking velocities, where the centrifugal force can become too low to provide sufficient knee damping. When this happens, the knee overshoots due to a lack of damping, and the emergency stop is reached at a certain angular velocity. This might also pose problems at higher walking velocities, because if damping occurs without any delay due to hysteresis, it slows down the knee until a velocity is reached where damping does not occur. This means that if gravity and friction are neglected, the variable damping mechanism can only reduce the velocity to a certain point.

7.2.2. Design strengths

The final design has several advantages. It is theoretically capable of completing a full gait cycle at various walking velocities and includes a robust knee lock and ankle stiffness to ensure stability during the stance phase. The variable damping mechanism does not yet perform as desired, but this might be improved by reducing hysteresis and fragility. The knee lock and unlock mechanisms operate automatically in response to the gait phase, so theoretically no manual operations are required. Additionally, a push-off mechanism is present which enhances propulsion. However, these advantages are only theoretical, as they have not been tested. A practical advantage is that the design is fully passive, requiring no battery or external power source. Lastly, it is cost-effective and easy to manufacture, consisting only of laser-cut metal, 3D-printed components, and low cost off-the-shelf parts.

7.2.3. Research limitations

Several limitations were present in this research. The experiments were conducted only at lower velocities than in natural gait, limiting the applicability of the results to natural gait. The highest angular velocity used in the test is $3.7 rad/s$, which corresponds to a walking velocity of $2.4 km/h$. This is significantly lower than the walking velocities of 3.7, 4.5, and $5.3 km/h$ for slow, normal and fast natural gait respectively. The system may behave differently at higher angular velocities, especially because $M_k \sim \dot{\theta}_k^2$ according to Equation 4.3. This implies that the knee moment increases quadratically with a linear increase in angular velocity. Therefore the difference moment can be measured more clearly, as the noise due to friction is relatively lower. Additionally, a higher angular velocity results in higher forces in the cable, which may be sufficient to overcome the hysteresis. However, testing at higher angular velocities is challenging due to the need for a high force manual push to accelerate the shank to give sufficient angular velocity. Due to the required high push forces, it is difficult to provide a clean push without giving unwanted lateral vibrations. Additionally, at these higher forces, the risk of breaking increased, as the 3D-printed parts of the design could fail under stress. Furthermore, no testing is performed on a human subject, and a full gait cycle test is not conducted. Another limitation is that no tests are conducted to assess the terrain-dependent behavior of the design. Due to these limitations in testing, no conclusion can be drawn on whether the final design can be classified as a $K3$-prosthesis.

7.3. Future Research Perspectives

Further research should focus on experimental validation and optimization of the design. For experimental validation, an alternative should be found to the manual pushes as an acceleration method in the swing phase test. This could for instance be done by using springs. The behavior of the design could then be assessed at angular velocities comparable to natural gait velocities to make the design more applicable to practice. Furthermore, performing tests with participants using a transfemoral prosthesis simulator would be valuable. This could be used to evaluate the ability of performing a full gait cycle, as well as various other relevant data, such as measuring the adaptability to different terrains.

In addition to experimental validation, design optimization is necessary, especially in improving the robustness of critical parts. Additional optional enhancements include decreasing the overall mass and reducing the impact of hysteresis. Another promising research direction is to eliminate the reliance on the foot as a mass linked to damping, which could resolve the issue of foot drop in the design. Lastly, a possible research direction would be implementing a delay between the centrifugal force as input and the resulting knee damping as an output. This would make the damping pattern in the late swing phase comparable to that of natural gait if the method in Chapter 4.1.2 is used. This would therefore be a promising research direction. Implementing a mechanical delay would be particularly interesting, as it would allow the design to remain entirely passive. A potential solution for achieving this mechanical delay could involve a spring-damper mechanism. The effect of implementing a delay is visualized in Figure 7.1.

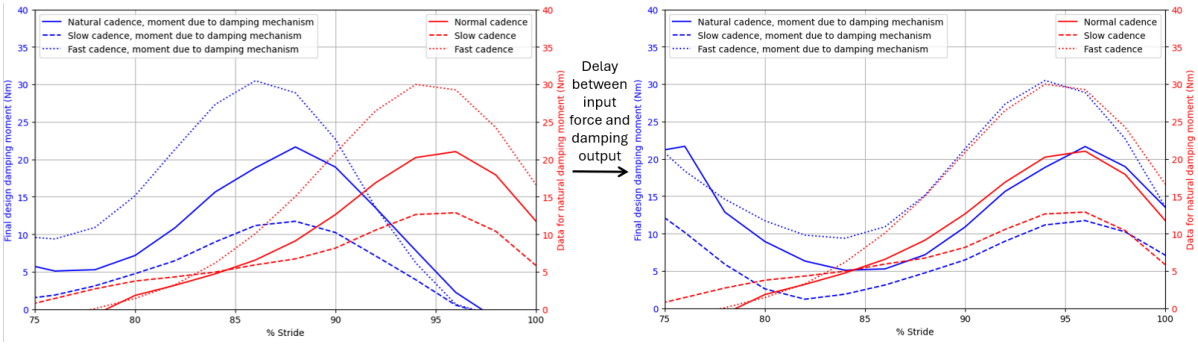

Figure 7.1: On the left, knee moment due to damping in the late swing phase as found by Winter et al. [13] in red, and an indication for the expected damping in the final design in blue, calculated by Equation 4.3. On the right, the same graph where there is a delay present between measurement and resulting damping in the design. The slow, normal and fast cadences are 88, 107 and $125 steps/min$, or 3.7, 4.5, and $5.3 km/h$ respectively.

8

Conclusion

The objective of this research is to propose a concept for a transfemoral *K3* prosthesis that is fully passive and suitable for variable walking velocities by implementing velocity-dependent knee damping in the swing phase.

The main working principle of the final design is that in the swing phase, the design links the centrifugal acceleration of the foot to friction damping in the knee using cables, thereby implementing a velocity dependent damping moment in the knee. In the stance phase, the design locks the knee using a spring latch. An ankle spring is activated, increasing the moment of the ankle as dorsiflexion increases. At sufficient dorsiflexion, a cable is pulled that unlocks the knee and the ankle spring releases its energy, and it initiates the swing phase.

The design is theoretically capable of a complete gait cycle, fully passive and suitable for variable walking velocities. However, due to the acceleration method of the shank and fragility of 3D printed parts, testing the variable knee damping mechanism is performed at angular velocities up to $3.6 rad/s$. This corresponds to a walking velocity of up to $2.4 km/h$, whereas natural cadence occurs approximately between $3.7 - 5.3 km/h$. Therefore, no clear conclusion can be made on the performance of the device in natural walking velocities. Some velocity-dependent behavior can be observed at lower velocities, but inconsistencies in knee damping are present, which can be attributed to hysteresis in the damping mechanism. Therefore, the results remain inconclusive. Lastly, the design is only assessed on walking on a flat surface.

Due to the limitations in the design and the experimental validation the concept can not yet be classified as a *K3* prosthesis. To reach that goal, the design should be improved to contain less hysteresis and be more robust. Additionally, testing needs to be performed at higher angular velocities with an alternative acceleration method. However, this research demonstrates the potential for a low-cost, fully passive variable damping knee prosthesis, which a promising initial step toward the development of low cost *K3* prostheses.

References

[1] Jefferson K. Rajah and B. Kopainsky. “Enabling Mobility: A Simulation Model of the Health Care System for Major Lower-Limb Amputees to Assess the Impact of Digital Prosthetics Services”. In: *Syst* (2023).

[2] Cody L. McDonald and D. Kartin. “Global prevalence of traumatic non-fatal limb amputation.” In: *Prosthetics and Orthotics International* 44 (2020), pp. 57–77.

[3] Dominik Wyss and J. Andrysek. “Priorities in lower limb prosthetic service delivery based on an international survey of prosthetists in low- and high-income countries”. In: *Prosthetics and Orthotics International* 39.5 (2015), pp. 361–370.

[4] Robert Ngarambe and D. Tumusiime. “The status and use of prosthetic devices by persons with lower limb amputation in Rwanda”. In: *African Journal of Disability* 11 (2022), pp. 645–654.

[5] Benedict Mulindwa and R. Ssekitoleko. “Evaluation of the current status of prosthetic rehabilitation services for major limb loss: a descriptive study in Ugandan Referral hospitals”. In: *Disability and Rehabilitation* 45.19 (2023), pp. 3494–3502.

[6] Health Care Finance Administration. *Common Procedure Coding System (HCPCS) 2001*. Washington, DC: U.S. Government Printing Office, 2001. URL: https://ntrl.ntis.gov/NTRL/dashboard/searchResults/titleDetail/PB2001100221.xhtml.

[7] M. U. Qadir and Mohamed A. Ismail. “Design, Analysis, and Development of Low-Cost State-of-the-Art Magnetorheological-Based Microprocessor Prosthetic Knee”. In: *Sensors (Basel, Switzerland)* (2024).

[8] Lucas Galey and Roger V. Gonzalez. “Optimization of a Cost-Constrained, Hydraulic Knee Prosthesis Using a Kinematic Analysis Model”. In: *Biomechanics* (2023).

[9] Ottobock. *C-Leg 4 Microprocessor Knee*. Accessed: 2025-02-09. 2025. URL: https://www.ottobock.com/nl-nl/product/3C88-3~23C98-3.

[10] Össur. *Rheo Knee XC*. Accessed: 2025-02-09. 2025. URL: https://www.ossur.com/nl-nl/prosthetics/knieen/rheo-knee-xc.

[11] Blatchford Mobility. *Linx Microprocessor-Controlled Leg*. Accessed: 2025-02-09. 2025. URL: https://www.blatchfordmobility.com/en-us/products/limb-systems/linx-microprocessor-controlled-leg/.

[12] Very Good Knee. *Fluidic Processor Knees*. Accessed: 2025-02-09. 2025. URL: https://www.verygoodknee.com/fluidic-processor-knees/.

[13] David A. Winter. “Energy Generation and Absorption at the Ankle and Knee during Fast, Natural, and Slow Cadences”. In: *Clinical Orthopaedics and Related Research®* 175 (May 1983), p. 147. ISSN: 0009-921X. URL: https://journals.lww.com/clinorthop/abstract/1983/05000/energy_generation_and_absorption_at_the_ankle_and.21.aspx (visited on 06/21/2024).

[14] Walter Pirker. “(PDF) Gait disorders in adults and the elderly: A clinical guide”. In: *ResearchGate* (Dec. 9, 2024). DOI: 10.1007/s00508-016-1096-4. URL: https://www.researchgate.net/publication/309362425_Gait_disorders_in_adults_and_the_elderly_A_clinical_guide (visited on 02/08/2025).

[15] Matthys Arnout et al. “Concept and design of the HEKTA (Harvest Energy from the Knee and Transfer it to the Ankle) transfemoral prosthesis”. In: *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. 2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob). ISSN: 2155-1782. June 2012, pp. 550–555. DOI: 10.1109/BioRob.2012.6290833. URL: https://ieeexplore.ieee.org/document/6290833 (visited on 03/27/2024).

[16] R. Unal et al. “Towards a fully passive transfemoral prosthesis for normal walking”. In: *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. 2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob). ISSN: 2155-1782. June 2012, pp. 1949–1954. DOI: 10.1109/BioRob.2012.6290837. URL: https://ieeexplore.ieee.org/document/6290837 (visited on 03/27/2024).

[17] Hong Gu and Juan Li. “Knee and ankle kinematics in different walking conditions”. In: *2016 International Conference on Advanced Robotics and Mechatronics (ICARM)*. 2016.

[18] Z. Jichuan. “Normal gait patterns on different terrain”. In: *Journal of Tsinghua University* (2000).

[19] Silver Bobbin. *How to Measure Leg Opening*. Accessed: 2025-03-12. n.d. URL: https://silverbobbin.com/how-to-measure-leg-opening/.

[20] Dined Database. *DINED Anthropometric Database - Tool*. Accessed: 2025-02-24. 2025. URL: https://dined.nl/en/database/tool.

A

Design process

A.1. Sub-functions & sub-solutions

An overview of all sub-functions the design should contain is seen in the first column of the morphological chart, as seen in table A.1. For every sub-function, all relevant solutions are presented.

A.2. Concept selection

The concept selection is done by giving each concept scores for weighted performance criteria. The choice for weight per criteria is shown in table A.2.

The scores per criteria and final scores of the concepts are shown in table A.3. As concept 1 has the highest score, this is chosen as the concept to expand on and turn into a more detailed design. For gait mimicry, concept 2 scores the highest as it follows the exact path of natural gait. Concept 3 scores slightly less than concept 1 as it is expected to have troubles in reaching sufficient damping torque. Natural stance stability behaves comparably to a spring, therefore concepts 2 and 3 score higher than concept 1. The velocity ranges of concept 1 and 3 are expected to be comparable as both have a variable damping system which is based on $\dot{\theta}_k$. Concept 2 scores less as it allows for only one gait pattern. Concept 1 & 3 score high on reproducibility because they are producible with mostly off-the-shelf products and 3D-printing. Concept 3 scores lower than 1 as the gait pattern mechanism of concept 2 requires precision tools to manufacture, so it scores lower. Regarding safety, the only hazard for concept 1 could be that something could get stuck between the moving parts, but that is expected to be solvable. Concept 2 might induce unexpected dangerous movements when unexpected forces are applied to the joints, and therefore scores the lowest. The electronics in concept 3 pose slight safety problems, as well as having a magnet in a prosthesis; therefore, it scores lower than concept 1. As the costs are closely linked to ease of manufacturing, the scores are comparable to reproducibility. For different terrain ranges, a concept should be adaptable to different situations. Concept 2 is not, due to its 1 degree-of-freedom constraint. Concept 3 scores slightly higher than concept 1 due to the stance stability spring, which is more versatile than a friction system because it is able to provide positive torque, which might be useful in terrains other than flat ground. Concept 2 is the only concept with a push-off mechanism, which is expected to be very effective due to gait pattern. Concept 1 is expected to be quite noisy due to the control mass moving and hitting the walls of the control chamber. Concept 2 also might produce a lot of noise due to friction as well as the amount of moving parts. Concept 3 is expected to be less noisy as the concept requires less physical contact between parts.

Table A.1: Morphological chart, showing all subfunctions and their according solutions.

Main subfunctions							
Knee damping	Friction	Electro-mechanical	Pneumatic	Hydraulic			
Weight acceptance	Friction	Spring	Electro-mechanical	Static joint lock	Dynamic joint lock	Polycentric knee	
Push-off mechanism	Electro-mechanical	Spring	Displacement coupling to other joint	Hydraulic pressure	Pneumatic pressure		
Energy harvesting medium	Electro-mechanical	Spring	Height	Kinetic energy	Rotational kinetic energy	Fluid pressure	
Variable damping control system subfunctions							
Gait phase measurement	Centrifugal force of shank or thigh	Ankle, knee or hip angle	Stress in thigh, shank or foot	Moment in hip, knee or ankle	Distance between foot and ground		
Walking velocity measurement	Angular velocity of hip, knee or ankle	Centrifugal force in thigh, shank oor foot	Stress in hip, shank or foot				
Sensor types	Mechanical force sensor	Mechanical displacement sensor	Electronic force sensor	Electronic displacement sensor	Muscle activity sensor (EMG)	Magnet moving through coil	
Transfer measurement	Displacement	Force	Velocity	Electrical	Fluid pressure	Chemical	Wireless signals

Table A.2: Performance Criteria Evaluation

Performance criteria	Mimics natural gait	Range of velocities	Range of terrains	Stance stability	Cost	Safety	Noise	Push-off force	Reproducibility	Total	Weight
How closely does it mimic natural gait?		+	+	+	+	+	+	+	+	8	**9**
Effective for what range of velocities?	-		+	-	+	+	+	+	+	6	**7**
Effective for what range of terrains?	-	-		-	-	-	+	+	-	2	**3**
How stable is the stance?	-	+	+		+	+	+	+	+	7	**8**
Cost	-	-	+	-		-	+	+	-	3	**4**
Safety	-	-	+	-	+		+	+	-	4	**5**
Noise	-	-	-	-	-	-		-	-	0	**1**
Push-off force	-	-	-	-	-	-	+		-	1	**2**
Reproducibility	-	-	+	-	+	+	+	+		5	**6**

Table A.3: Evaluation of concepts based on performance criteria.

	Weight	**Score (1-10)**	**Total Score**	**Score (1-10)**	**Total Score**	**Score (1-10)**	**Total Score**
Criteria		**Concept 1**		**Concept 2**		**Concept 3**	
How closely does it mimic natural gait?	9	**6**	54	**10**	90	**5**	45
How stable is the weight acceptance?	8	**8**	64	**10**	80	**10**	80
Effective for what range of velocities?	7	**8**	56	**6**	42	**8**	56
Reproducibility	6	**10**	60	**2**	12	**7**	42
Safety	5	**9**	45	**7**	35	**4**	20
Cost	4	**10**	40	**3**	12	**7**	28
Effective for what range of terrains?	3	**6**	18	**2**	6	**7**	21
Push-off force	2	**1**	2	**10**	20	**1**	2
Noise	1	**4**	4	**2**	2	**8**	8
Total Score (Max. = 450)			**343**		**299**		**302**

B

Data processing

B.1. Swing phase test

```
# README: Every comments applies to all lines below

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt


# Define file paths for calibration and test datasets
file_calibration = "new␣callibration.xlsx"
file_main_test = "superfast␣1.xlsx"
file_sneller = "extreem␣1.xlsx"
file_snelst = "extreem␣2.xlsx"
file_main_test2 = "superfast␣3.xlsx"
file_sneller2 = "extreem␣4.xlsx"
file_snelst2 = "new␣8.xlsx"
file_main_test3 = "superfast␣4.xlsx"
file_sneller3 = "extreem␣3.xlsx"
file_snelst3 = "new␣9.xlsx"

# Load datasets from Excel files
# The header is set to None as the column titles will be extracted later
df_calibration = pd.read_excel(file_calibration, header=None)
df_main_test = pd.read_excel(file_main_test, header=None)
df_sneller = pd.read_excel(file_sneller, header=None)
df_snelst = pd.read_excel(file_snelst, header=None)
df_main_test2 = pd.read_excel(file_main_test2, header=None)
df_sneller2 = pd.read_excel(file_sneller2, header=None)
df_snelst2 = pd.read_excel(file_snelst2, header=None)
df_main_test3 = pd.read_excel(file_main_test3, header=None)
df_sneller3 = pd.read_excel(file_sneller3, header=None)
df_snelst3 = pd.read_excel(file_snelst3, header=None)

# Extract column titles from the 7th row (index 6)
column_titles = df_main_test.iloc[6, 2:5].values  # Assume these are columns R, S, T

# Process the calibration data
data_calibration = df_calibration.iloc[1:50, 2:5]  # Extract relevant rows and columns
data_calibration.columns = column_titles  # Assign column names
data_calibration = data_calibration.apply(pd.to_numeric, errors='coerce')  # Convert to
    numeric
data_calibration = data_calibration * 1e-6  # Convert from microdegrees to degrees
data_calibration_rad = np.deg2rad(data_calibration)  # Convert degrees to radians

# Define a function to process test datasets
def process_test_data(df, row_start, row_end):
    data = df.iloc[row_start:row_end, 2:5]  # Extract relevant rows and columns
    data.columns = column_titles  # Assign column names
```

```
    data = data.apply(pd.to_numeric, errors='coerce')  # Convert to numeric
    data = data.fillna(method='ffill')  # Forward-fill missing values
    data = data * 1e-6  # Convert from microdegrees to degrees
    return np.deg2rad(data)  # Convert degrees to radians

# Process all test datasets
data_main_test_shifted = process_test_data(df_main_test, 150, 410)
data_main_test_shifted2 = process_test_data(df_main_test2, 100, 330)
data_main_test_shifted3 = process_test_data(df_main_test3, 80, 340)
data_sneller_shifted = process_test_data(df_sneller, 140, 360)
data_sneller_shifted2 = process_test_data(df_sneller2, 90, 360)
data_sneller_shifted3 = process_test_data(df_sneller3, 90, 320)
data_snelst_shifted = process_test_data(df_snelst, 90, 360)
data_snelst_shifted2 = process_test_data(df_snelst2, 50, 350)
data_snelst_shifted3 = process_test_data(df_snelst3, 50, 350)

# Define compensation and calibration adjustment
compensation_value = np.sqrt(3 * (20 ** 2))  # Calculate compensation factor
total_rotation_calibration = np.sqrt(((data_calibration_rad)**2).sum(axis=1))  # Compute
    total rotation for calibration
calibrate_final = total_rotation_calibration[10]  # Take a reference value from calibration

# Compute total rotation for each dataset, applying compensation and calibration

def compute_total_rotation(data):
    return np.sqrt(((data + 20) ** 2).sum(axis=1)) - compensation_value - calibrate_final

# Calculate total rotation for each test dataset
total_rotation_main_test = compute_total_rotation(data_main_test_shifted)
total_rotation_main_test2 = compute_total_rotation(data_main_test_shifted2)
total_rotation_main_test3 = compute_total_rotation(data_main_test_shifted3)
total_rotation_sneller = compute_total_rotation(data_sneller_shifted)
total_rotation_sneller2 = compute_total_rotation(data_sneller_shifted2)
total_rotation_sneller3 = compute_total_rotation(data_sneller_shifted3)
total_rotation_snelst = compute_total_rotation(data_snelst_shifted)
total_rotation_snelst2 = compute_total_rotation(data_snelst_shifted2)
total_rotation_snelst3 = compute_total_rotation(data_snelst_shifted3)

# Generate time steps for plotting
time_main_test = np.arange(len(total_rotation_main_test))
time_main_test2 = np.arange(len(total_rotation_main_test2))
time_main_test3 = np.arange(len(total_rotation_main_test3))
time_sneller = np.arange(len(total_rotation_sneller))
time_sneller2 = np.arange(len(total_rotation_sneller2))
time_sneller3 = np.arange(len(total_rotation_sneller3))
time_snelst = np.arange(len(total_rotation_snelst))
time_snelst2 = np.arange(len(total_rotation_snelst2))
time_snelst3 = np.arange(len(total_rotation_snelst3))

# Plot the total angle of rotation over time
plt.figure(figsize=(10, 5))
plt.plot(time_main_test, total_rotation_main_test, label="Main␣Test␣1", color="black",
    linestyle='-', linewidth=1)
plt.plot(time_sneller, total_rotation_sneller, label="Sneller␣Test␣1", color="blue",
    linestyle='-', linewidth=1)
plt.plot(time_snelst, total_rotation_snelst, label="Snelst␣Test␣1", color="red", linestyle='-
    ', linewidth=1)
plt.plot(time_main_test2, total_rotation_main_test2, label="Main␣Test␣2", color="black",
    linestyle='--', linewidth=1)
plt.plot(time_sneller2, total_rotation_sneller2, label="Sneller␣Test␣2", color="blue",
    linestyle='--', linewidth=1)
plt.plot(time_snelst2, total_rotation_snelst2, label="Snelst␣Test␣2", color="red", linestyle=
    '--', linewidth=1)
plt.plot(time_main_test3, total_rotation_main_test3, label="Main␣Test␣3", color="black",
    linestyle=':', linewidth=1)
plt.plot(time_sneller3, total_rotation_sneller3, label="Sneller␣Test␣3", color="blue",
    linestyle=':', linewidth=1)
plt.plot(time_snelst3, total_rotation_snelst3, label="Snelst␣Test␣3", color="red", linestyle=
    ':', linewidth=1)

# Configure plot labels and appearance
```

```
plt.xlabel("Time Step")
plt.ylabel("Total Rotation Angle (rad)")
plt.title("Total Rotation Angle Over Time")
plt.legend()
plt.grid()
plt.ylim(-1, 0.6)
plt.show()

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Convert pandas Series to numpy arrays for easier manipulation
angles_main_test = total_rotation_main_test.to_numpy()
angles_snelst = total_rotation_snelst.to_numpy()
angles_sneller = total_rotation_sneller.to_numpy()

angles_main_test2 = total_rotation_main_test2.to_numpy()
angles_snelst2 = total_rotation_snelst2.to_numpy()
angles_sneller2 = total_rotation_sneller2.to_numpy()

angles_main_test3 = total_rotation_main_test3.to_numpy()
angles_snelst3 = total_rotation_snelst3.to_numpy()
angles_sneller3 = total_rotation_sneller3.to_numpy()

# Define smoothing window sizes for different calculations
window_size = 20  # Window size for smoothing angles
window_size2 = 10  # Window size for smoothing angular velocity
window_size3 = 2   # Window size for smoothing angular acceleration

# Apply moving average smoothing to the angular data
angles_main_test_smooth = pd.Series(angles_main_test).rolling(window=window_size, center=True
    ).mean().to_numpy()
angles_snelst_smooth = pd.Series(angles_snelst).rolling(window=window_size, center=True).mean
    ().to_numpy()
angles_sneller_smooth = pd.Series(angles_sneller).rolling(window=window_size, center=True).
    mean().to_numpy()

angles_main_test_smooth2 = pd.Series(angles_main_test2).rolling(window=window_size, center=
    True).mean().to_numpy()
angles_snelst_smooth2 = pd.Series(angles_snelst2).rolling(window=window_size, center=True).
    mean().to_numpy()
angles_sneller_smooth2 = pd.Series(angles_sneller2).rolling(window=window_size, center=True).
    mean().to_numpy()

angles_main_test_smooth3 = pd.Series(angles_main_test3).rolling(window=window_size, center=
    True).mean().to_numpy()
angles_snelst_smooth3 = pd.Series(angles_snelst3).rolling(window=window_size, center=True).
    mean().to_numpy()
angles_sneller_smooth3 = pd.Series(angles_sneller3).rolling(window=window_size, center=True).
    mean().to_numpy()

# Define time step between frames (assuming 120 frames per second)
dt = 1 / 120

# Compute angular velocity by taking the derivative of angle with respect to time
angular_velocity_main_test = np.diff(angles_main_test_smooth) / dt
angular_velocity_snelst = np.diff(angles_snelst_smooth) / dt
angular_velocity_sneller = np.diff(angles_sneller_smooth) / dt

angular_velocity_main_test2 = np.diff(angles_main_test_smooth2) / dt
angular_velocity_snelst2 = np.diff(angles_snelst_smooth2) / dt
angular_velocity_sneller2 = np.diff(angles_sneller_smooth2) / dt

angular_velocity_main_test3 = np.diff(angles_main_test_smooth3) / dt
angular_velocity_snelst3 = np.diff(angles_snelst_smooth3) / dt
angular_velocity_sneller3 = np.diff(angles_sneller_smooth3) / dt

# Smooth the angular velocity data
angular_velocity_main_test_smooth = pd.Series(angular_velocity_main_test).rolling(window=
    window_size2, center=True).mean().to_numpy()
```

```
angular_velocity_snelst_smooth = pd.Series(angular_velocity_snelst).rolling(window=window_size2, center=True).mean().to_numpy()
angular_velocity_sneller_smooth = pd.Series(angular_velocity_sneller).rolling(window=window_size2, center=True).mean().to_numpy()

angular_velocity_main_test_smooth2 = pd.Series(angular_velocity_main_test2).rolling(window=window_size2, center=True).mean().to_numpy()
angular_velocity_snelst_smooth2 = pd.Series(angular_velocity_snelst2).rolling(window=window_size2, center=True).mean().to_numpy()
angular_velocity_sneller_smooth2 = pd.Series(angular_velocity_sneller2).rolling(window=window_size2, center=True).mean().to_numpy()

angular_velocity_main_test_smooth3 = pd.Series(angular_velocity_main_test3).rolling(window=window_size2, center=True).mean().to_numpy()
angular_velocity_snelst_smooth3 = pd.Series(angular_velocity_snelst3).rolling(window=window_size2, center=True).mean().to_numpy()
angular_velocity_sneller_smooth3 = pd.Series(angular_velocity_sneller3).rolling(window=window_size2, center=True).mean().to_numpy()

# Compute angular acceleration by taking the derivative of angular velocity
angular_acceleration_main_test = np.diff(angular_velocity_main_test_smooth) / dt
angular_acceleration_snelst = np.diff(angular_velocity_snelst_smooth) / dt
angular_acceleration_sneller = np.diff(angular_velocity_sneller_smooth) / dt

angular_acceleration_main_test2 = np.diff(angular_velocity_main_test_smooth2) / dt
angular_acceleration_snelst2 = np.diff(angular_velocity_snelst_smooth2) / dt
angular_acceleration_sneller2 = np.diff(angular_velocity_sneller_smooth2) / dt

angular_acceleration_main_test3 = np.diff(angular_velocity_main_test_smooth3) / dt
angular_acceleration_snelst3 = np.diff(angular_velocity_snelst_smooth3) / dt
angular_acceleration_sneller3 = np.diff(angular_velocity_sneller_smooth3) / dt

# Smooth the angular acceleration data
angular_acceleration_main_test_smooth = pd.Series(angular_acceleration_main_test).rolling(window=window_size3, center=True).mean().to_numpy()
angular_acceleration_snelst_smooth = pd.Series(angular_acceleration_snelst).rolling(window=window_size3, center=True).mean().to_numpy()
angular_acceleration_sneller_smooth = pd.Series(angular_acceleration_sneller).rolling(window=window_size3, center=True).mean().to_numpy()

angular_acceleration_main_test_smooth2 = pd.Series(angular_acceleration_main_test2).rolling(window=window_size3, center=True).mean().to_numpy()
angular_acceleration_snelst_smooth2 = pd.Series(angular_acceleration_snelst2).rolling(window=window_size3, center=True).mean().to_numpy()
angular_acceleration_sneller_smooth2 = pd.Series(angular_acceleration_sneller2).rolling(window=window_size3, center=True).mean().to_numpy()

angular_acceleration_main_test_smooth3 = pd.Series(angular_acceleration_main_test3).rolling(window=window_size3, center=True).mean().to_numpy()
angular_acceleration_snelst_smooth3 = pd.Series(angular_acceleration_snelst3).rolling(window=window_size3, center=True).mean().to_numpy()
angular_acceleration_sneller_smooth3 = pd.Series(angular_acceleration_sneller3).rolling(window=window_size3, center=True).mean().to_numpy()

# Plot angular velocity comparison
plt.figure(figsize=(12, 8))
plt.subplot(2, 1, 1)

plt.plot(angular_velocity_main_test_smooth, label='Main Test Angular Velocity (rad/s)', color='blue')
plt.plot(angular_velocity_snelst_smooth, label='Snelst Angular Velocity (rad/s)', color='orange')
plt.plot(angular_velocity_sneller_smooth, label='Sneller Angular Velocity (rad/s)', color='green')

plt.xlabel('Time (frames)')
plt.ylabel('Angular Velocity (rad/s)')
plt.title('Angular Velocity Comparison')
plt.legend()
plt.grid(True)

```

```
220 # Plot angular acceleration comparison
221 plt.subplot(2, 1, 2)
222 plt.plot(angular_acceleration_main_test_smooth, label='Main␣Test␣Angular␣Acceleration␣(rad/s²
    )', color='blue')
223 plt.plot(angular_acceleration_snelst_smooth, label='Snelst␣Angular␣Acceleration␣(rad/s²)',
    color='orange')
224 plt.plot(angular_acceleration_sneller_smooth, label='Sneller␣Angular␣Acceleration␣(rad/s²)',
    color='green')
225
226 plt.xlabel('Time␣(frames)')
227 plt.ylabel('Angular␣Acceleration␣(rad/s²)')
228 plt.title('Angular␣Acceleration␣Comparison')
229 plt.legend()
230 plt.grid(True)
231
232 plt.tight_layout()
233 plt.show()
234
235 # Plot angular acceleration against angular velocity
236 plt.figure(figsize=(12, 6))
237 plt.plot(angular_velocity_main_test_smooth[1:], angular_acceleration_main_test_smooth, label=
    'Main␣Test', color='blue')
238 plt.plot(angular_velocity_snelst_smooth[1:], angular_acceleration_snelst_smooth, label='
    Snelst', color='orange')
239 plt.plot(angular_velocity_sneller_smooth[1:], angular_acceleration_sneller_smooth, label='
    Sneller', color='green')
240
241 plt.xlabel('Angular␣Velocity␣(rad/s)')
242 plt.ylabel('Angular␣Acceleration␣(rad/s²)')
243 plt.title('Angular␣Acceleration␣vs␣Angular␣Velocity')
244 plt.legend()
245 plt.grid(True)
246 plt.tight_layout()
247 plt.show()
248
249 import numpy as np
250 import matplotlib.pyplot as plt
251
252 # Define constants
253 # Moment of inertia and related parameters
254 m = 2.581  # Mass of the leg (kg)
255 r_leg = 0.327  # Distance from the pivot to the center of mass (m)
256 I_com = 0.0985262392  # Moment of inertia about the center of mass (kgm²)
257 I = m * (r_leg ** 2) + I_com  # Total moment of inertia (kgm²)
258
259 tolerance = 0.01  # Tolerance for filtering data based on angle
260 measurement_angle = 0.0  # Target angle for filtering
261
262 # Gravitational moment decrease at the measurement angle
263 moment_decrease_g = np.sin(measurement_angle) * m * 9.81 * r_leg
264 standard_damping = 0  # Base damping value (excluding variable damping system)
265
266 # Function to filter data where angle  measurement_angle
267 def filter_by_angle(angle_data, velocity_data, acceleration_data, target_angle=-
    measurement_angle, tol=tolerance):
268     """
269     Filters data based on a specified target angle within a given tolerance.
270     Returns velocity and acceleration values corresponding to the filtered angles.
271     """
272     indices = np.where(np.abs(angle_data - target_angle) < tol)[0]  # Find indices where
        angle  target_angle
273     return velocity_data[indices], acceleration_data[indices]
274
275 # Apply filtering for multiple test datasets
276 vel_main_test, acc_main_test = filter_by_angle(angles_main_test_smooth[:-2],
    angular_velocity_main_test_smooth[:-1], angular_acceleration_main_test_smooth)
277 vel_snelst, acc_snelst = filter_by_angle(angles_snelst_smooth[:-2],
    angular_velocity_snelst_smooth[:-1], angular_acceleration_snelst_smooth)
278 vel_sneller, acc_sneller = filter_by_angle(angles_sneller_smooth[:-2],
    angular_velocity_sneller_smooth[:-1], angular_acceleration_sneller_smooth)
279
```

```
vel_main_test2, acc_main_test2 = filter_by_angle(angles_main_test_smooth2[:-2], angular_velocity_main_test_smooth2[:-1], angular_acceleration_main_test_smooth2)
vel_snelst2, acc_snelst2 = filter_by_angle(angles_snelst_smooth2[:-2], angular_velocity_snelst_smooth2[:-1], angular_acceleration_snelst_smooth2)
vel_sneller2, acc_sneller2 = filter_by_angle(angles_sneller_smooth2[:-2], angular_velocity_sneller_smooth2[:-1], angular_acceleration_sneller_smooth2)

vel_main_test3, acc_main_test3 = filter_by_angle(angles_main_test_smooth3[:-2], angular_velocity_main_test_smooth3[:-1], angular_acceleration_main_test_smooth3)
vel_snelst3, acc_snelst3 = filter_by_angle(angles_snelst_smooth3[:-2], angular_velocity_snelst_smooth3[:-1], angular_acceleration_snelst_smooth3)
vel_sneller3, acc_sneller3 = filter_by_angle(angles_sneller_smooth3[:-2], angular_velocity_sneller_smooth3[:-1], angular_acceleration_sneller_smooth3)

# Scatter plot for Angular Velocity vs. Moment due to damping
plt.figure(figsize=(10, 6))

def plot_scatter(velocity, acceleration):
    """Helper function to plot experimental values."""
    plt.scatter(-velocity, acceleration * I - moment_decrease_g - standard_damping, color='blue', alpha=0.7)

# Plot experimental values
plot_scatter(vel_main_test, acc_main_test)
plot_scatter(vel_snelst, acc_snelst)
plot_scatter(vel_sneller, acc_sneller)
plot_scatter(vel_main_test2, acc_main_test2)
plot_scatter(vel_snelst2, acc_snelst2)
plot_scatter(vel_sneller2, acc_sneller2)
plot_scatter(vel_main_test3, acc_main_test3)
plot_scatter(vel_snelst3, acc_snelst3)
plot_scatter(vel_sneller3, acc_sneller3)

# Combine all x (velocity) and y (moment) values for fitting a trend line
x_values_unf = np.concatenate([
    -vel_main_test, -vel_snelst, -vel_sneller,
    -vel_main_test2, -vel_snelst2, -vel_sneller2,
    -vel_main_test3, -vel_snelst3, -vel_sneller3
])

y_values_unf = np.concatenate([
    acc_main_test * I - moment_decrease_g - standard_damping,
    acc_snelst * I - moment_decrease_g - standard_damping,
    acc_sneller * I - moment_decrease_g - standard_damping,
    acc_main_test2 * I - moment_decrease_g - standard_damping,
    acc_snelst2 * I - moment_decrease_g - standard_damping,
    acc_sneller2 * I - moment_decrease_g - standard_damping,
    acc_main_test3 * I - moment_decrease_g - standard_damping,
    acc_snelst3 * I - moment_decrease_g - standard_damping,
    acc_sneller3 * I - moment_decrease_g - standard_damping
])

# Remove invalid values (negative and NaN values)
valid_mask = (x_values_unf >= 0) & (y_values_unf >= 0) & ~np.isnan(x_values_unf) & ~np.isnan(y_values_unf)
x_values = x_values_unf[valid_mask]
y_values = y_values_unf[valid_mask]

# Fit a quadratic model to the data
coefficients = np.polyfit(x_values, y_values, 2)
poly = np.poly1d(coefficients)

# Generate points for the trend line
x_trend = np.linspace(min(x_values) - 0.5, max(x_values) + 5, 100)
y_trend = poly(x_trend)
plt.plot(x_trend, y_trend, label="Trend line experimental values", color='blue', linestyle='--')

# Define natural gait peak velocity and peak moment
piek_vels = np.array([4.6, 5.9, 7.1])
peak_moment = np.array([0.161, 0.263, 0.375]) * 80  # Scaled by bodyweight
```

```

plt.scatter(piek_vels, peak_moment, color='red', label='Natural gait peak velocity and peak moment', alpha=0.7)

# Define expected quadratic moment function
def quadratic_function(x):
    return 0.4859 * x**2 - 1.82

# Plot expected moment function
x_cal = np.linspace(0, 7.5, 100)
y_cal = quadratic_function(x_cal)
plt.plot(x_cal, y_cal, label="Expected moment", color="orange")

# Plot formatting
plt.xlabel("Angular velocity (rad/s)")
plt.ylabel("Moment due to damping (Nm)")
plt.legend()
plt.grid(True)
plt.xlim(0, 7.5)
plt.ylim(0, 31)
plt.show()
```

B.2. Ankle stiffness test

```
# README: Every comments applies to all lines below

import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# List of Excel files
excel_files = [f"Reductie {i}.xlsx" for i in range(1, 6)]

# Column names for data extraction
columns = ['time', 'pull force v', 'displacement v', 'pull force N', 'displacement mm']

# Define stiffness and ratio for calculations
stiffness = 28.42
afstandsverhouding = 258/121 # mm ratio

displacement_array = np.arange(0,30,1)
force_array = stiffness * displacement_array / (afstandsverhouding * afstandsverhouding)

# Dictionary to store data from each file
data = {column: [] for column in columns}

try:
    # Load and process each Excel file
    for file in excel_files:
        df = pd.read_excel(file)
        df.columns = columns  # Rename columns to match expected format
        for column in columns:
            data[column].append(df[column])  # Append column data

    # Plot displacement vs. pull force for all Excel files
    plt.figure(figsize=(10, 6))
    for i, file in enumerate(excel_files):
        plt.plot((data['displacement v'][i] / 1000 - 18), -data['pull force N'][i] / 10000,
            label=f"{file}", linewidth=0.5)

    # Plot expected force curve
    plt.plot(
        displacement_array,
        force_array,
        color="black",            # Black color
        linewidth=2,              # Thicker line
        linestyle="--",           # Dashed line
        label="Expected force"    # Line label
    )

```

```
    # Label axes and add title
    plt.xlabel("Displacement V")
    plt.ylabel("Pull Force N")
    plt.title("Ankle spring stiffness test: Displacement vs Pull force")
    plt.legend()
    plt.grid(True)
    plt.show()

except FileNotFoundError as e:
    print(f"Error: {e}. Please check the file names and paths.")
except Exception as e:
    print(f"An error occurred: {e}")

import math
from scipy.optimize import minimize  # Import minimize function for optimization

def calculate_angles(displacements, pivot, toe_start):
    px, py = pivot
    t0x, t0y = toe_start

    # Radius of the arc (distance from pivot to toe start)
    R = math.sqrt((t0x - px)**2 + (t0y - py)**2)

    def angle_equation(theta, d):
        """
        Computes the difference between cable length and expected displacement.
        theta: Toe angle around the pivot.
        d: Displacement (cable length from (0,0) to the toe).
        """
        tx = px + R * np.cos(theta)
        ty = py + R * np.sin(theta)
        cable_length = np.sqrt(tx**2 + ty**2)
        return cable_length - d

    angles = []
    for d in displacements:
        if d > R + np.sqrt(px**2 + py**2):
            raise ValueError(f"Displacement {d} exceeds maximum possible cable length.")

        # Solve for theta using minimize with bounds (0 < theta < pi)
        initial_guess = 1  # Initial guess for angle
        result = minimize(lambda theta: abs(angle_equation(theta, d)), initial_guess, bounds
            =[(0, np.pi)])

        if result.success:
            theta = result.x[0]  # Extract solution
            angles.append(np.degrees(theta))  # Convert to degrees
        else:
            raise ValueError(f"Optimization failed for displacement {d}.")

    return angles

# Define pivot and toe start positions
pivot = (440, -185)
toe_start = (530, 60)

# Compute angles for displacement data
angles = calculate_angles(((data['displacement mm'][0]/10000) + 43.8), pivot, toe_start)

# Compute theoretical angles
angles_simple = lambda d: d * 360 / (2 * 258 * np.pi)
add = -6.9
addii = 43.8

angles1 = [angle + add for angle in angles_simple((data['displacement mm'][0]/10000 + addii))
    ]
angles2 = [angle + add for angle in angles_simple((data['displacement mm'][1]/10000 + addii))
    ]
angles3 = [angle + add for angle in angles_simple((data['displacement mm'][2]/10000 + addii))
    ]
angles4 = [angle + add for angle in angles_simple((data['displacement mm'][3]/10000 + addii))
```

```
    ]
angles5 = [angle + add for angle in angles_simple((data['displacement␣mm'][4]/10000 + addii))
    ]

# Compute forces
force1 = -data['pull␣force␣N'][0] / 10000
force2 = -data['pull␣force␣N'][1] / 10000
force3 = -data['pull␣force␣N'][2] / 10000
force4 = -data['pull␣force␣N'][3] / 10000
force5 = -data['pull␣force␣N'][4] / 10000

# Compute force components adjusted for ankle angle
force1x = force1 * np.cos((np.deg2rad(angles1) - np.deg2rad(15))) / np.cos(np.deg2rad(15))
force2x = force2 * np.cos((np.deg2rad(angles2) - np.deg2rad(15))) / np.cos(np.deg2rad(15))
force3x = force3 * np.cos((np.deg2rad(angles3) - np.deg2rad(15))) / np.cos(np.deg2rad(15))
force4x = force4 * np.cos((np.deg2rad(angles4) - np.deg2rad(15))) / np.cos(np.deg2rad(15))
force5x = force5 * np.cos((np.deg2rad(angles5) - np.deg2rad(15))) / np.cos(np.deg2rad(15))

# Compute moments (force * lever arm distance)
moment1x = force1x * 0.258
moment2x = force2x * 0.258
moment3x = force3x * 0.258
moment4x = force4x * 0.258
moment5x = force5x * 0.258

# Compute expected moment curve
angle_array = np.arange(-4.5, 30.5, 1)
angle_array_2 = np.deg2rad(abs(angle_array + 4.5 - 29.2))
force_array_spring = -(stiffness * 121 * 2 * (np.sin(0.5 * angle_array_2) - np.sin(0.5 * np.
    deg2rad(abs(-29.2)))))
force_array = force_array_spring * 121 / 258
moment_array = force_array_spring * 0.121 * np.cos(0.5 * angle_array_2)

# Plot results
plt.plot(angles1, moment1x, color='red', linewidth=0.6, label="Experimental␣moment")
plt.plot(angles2, moment2x, color='red', linewidth=0.6)
plt.plot(angles3, moment3x, color='red', linewidth=0.6)
plt.plot(angles4, moment4x, color='red', linewidth=0.6)
plt.plot(angles5, moment5x, color='red', linewidth=0.6)
plt.plot(angle_array, moment_array, color='black', linestyle='--', linewidth=1, label="
    Expected␣moment")
plt.xlabel("Ankle␣angle␣(°)")
plt.ylabel("Ankle␣moment␣(Nm)")
plt.xlim(-6, 10)
plt.ylim(0, 100)
plt.legend()
plt.grid(True)
plt.show()
```

C

Use of artificial intelligence

In the writing of this thesis, the *Artificial Intelligence (AI)* by Writefull which is implemented in Overleaf is used. This AI gives suggestions when a written sentence is not grammatically correct.

In the data processing, ChatGPT is used to assist with coding in Python. An example of a used prompt is:
Input: "*[my code], this is my code. Implement a moving average filter with a window of 20 over the data of the angles.*"
Output: "*[my code with an implemented moving average filter]*"
The code provided is always checked by the writer of this thesis, and the writer guarantees that only code which is fully understood is used.

Undermind AI is used occasionally to find sources. An example of a prompt is:
Input: *"Give me papers describing the effect of stair walking and incline walking on knee and ankle angles and moments."*
Output: *[A list of papers describing the effect of stair walking and incline walking on knee and ankle angles and moments, as well as a small summary]*