Friedrich-Alexander-Universität
Faculty of Engineering

Dissertation

November 2024

Eduard S. Scheiterer

Dynamic analysis of a human leg model with a prosthetic foot in the presence of polymorphic uncertainty

Dynamic analysis of a human leg model with a prosthetic foot in the presence of polymorphic uncertainty

Dynamische Analyse eines menschlichen Beinmodells mit Fußprothese in Präsenz polymorpher Unschärfe

Der Technischen Fakultät
der Friedrich-Alexander-Universität
Erlangen-Nürnberg

zur

Erlangung des Doktorgrades Dr.-Ing.

vorgelegt von

Eduard Sebastian Scheiterer

Als Dissertation genehmigt
von der Technischen Fakultät
der Friedrich-Alexander-Universität Erlangen-Nürnberg

Tag der mündlichen Prüfung: 20.11.2024

Gutachter/in: Prof. Dr.-Ing. habil. Sigrid Leyendecker
Prof. Dr.-Ing. habil. Kai Willner

Impressum
Prof. Dr.-Ing. habil. Sigrid Leyendecker
Lehrstuhl für Technische Dynamik
Friedrich-Alexander-Universität Erlangen-Nürnberg
Immerwahrstraße 1
91058 Erlangen

Dynamic analysis of a human leg model with a prosthetic foot in the presence of polymorphic uncertainty

Dynamische Analyse eines menschlichen Beinmodells mit Fußprothese in Präsenz polymorpher Unschärfe

Eduard Sebastian Scheiterer

Schriftenreihe Technische Dynamik
Band 13 2024
Herausgeber: Prof. Dr.-Ing. habil. Sigrid Leyendecker

Preface

Completing this work would have been a lot more difficult without the help from many sources.

I would like to thank Prof. Dr.-Ing. habil. Sigrid Leyendecker for the many years of guidance, advice and fruitful discussions that culminated in this thesis.

My thanks also go to all my former colleagues at the Lehrstuhl für Technische Dynamik for the interesting discussions, many laughs and good times and for creating such an enjoyable atmosphere for working and teaching.

I also appreciate the German Research Foundation (DFG) which funded my work as part of the Priority Programme SPP 1886 'Polymorphic uncertainty modelling for the numerical design of structures' (Grant No. LE 1841/4-2).

Last but certainly not least, I would like to express my gratitude to my family and friends, who have supported me for many more years than just during this thesis. Thank you for being there for me, helping and supporting me in whatever I do.

Erlangen, November 2024 Eduard Sebastian Scheiterer

Abstract

When designing modern structural systems, such as the foot prosthesis considered in this work, analysis of their behaviour via simulations is an essential part of the design process. This process assumes that the model and the procedure used in the simulation both accurately represent the real structure as well as the physical laws governing its behaviour. Ideally, this would mean, that the model is absolutely accurate and the simulation algorithms used are without errors in their calculations. However, this is not possible for multiple reasons. Some stem from simulation limitations, i. e., computational limitations or numerical discretisations. Others are caused by missing information, like for instance measurement inaccuracies or missing data. Thus, to be feasible, a reduction in the model's complexity is necessary although this introduces additional uncertainty into the simulation. This uncertainty has many sources and can either be implicitly considered when interpreting the results by adding safety margins to the results or it can be considered explicitly, by modelling it and including it in the simulation. The goal of this work is to provide an efficient algorithm for considering polymorphic uncertainty in the simulation of human gait with a prosthetic foot and is part of the larger Priority Programme Schwerpunktprogramm (SPP) 1886 'Polymorphic uncertainty modelling for the numerical design of structures'.
To do this, three aspects have to be combined. Firstly, a suitable model needs to be developed. Here, the human leg with a prosthetic foot is modelled as a multibody system with rigid and flexible bodies. Secondly, the uncertainty that will be considered in the simulation has to be formalised, quantified and modelled. Then, the simulation procedure has to be expanded so it can propagate the uncertainty through the model. Finally, the results have to be visualised and interpreted. Two types of uncertainty are considered in this thesis. The Graph Follower algorithm is used to propagate epistemic uncertainty through the developed model, before expanding the uncertainty model to polymorphic uncertainty in the form of fuzzy random numbers which is propagated via the newly developed Fuzzy Random Variable Graph Follower algorithm (FRV-GFA). Thus, this thesis contributes a forward dynamics simulation of the human leg with a flexible prosthesis for two distinct gait phases, namely the swing phase and stance phase, in which parametric uncertainty is explicitly considered. To do this, the existing Graph Follower algorithm's efficiency is greatly improved so it can propagate epistemic uncertainty through the model. The main contributions of this thesis are the development of a new model for marker position errors during optical motion capture and an effi-

cient algorithm, based on the Graph Follower algorithm, which can propagate polymorphic uncertainty modelled with fuzzy random variables through the complex non-linear multibody system of a human leg with a flexible prosthetic foot.

Keywords:
polymorphic uncertainty – Fuzzy Random Variable Graph Follower algorithm – fuzzy random variables – epistemic uncertainty – Graph Follower algorithm – forward dynamics – multi-body dynamics – variational integrators – predeformed geometrically exact beam – foot prosthesis – gait analysis – marker based optical motion capture

Zusammenfassung

Bei der Gestaltung moderner Strukturen, wie beispielsweise der in dieser Arbeit betrachteten Fußprothese, ist die Analyse ihres Verhaltens durch Simulationen ein wesentlicher Bestandteil des Designprozesses. Dabei wird angenommen, dass das Modell und das Verfahren, das in der Simulation verwendet wird, sowohl die tatsächliche Struktur als auch die physikalischen Gesetze, die das resultierende Verhalten steuern, genau repräsentieren. Idealerweise bedeutet dies, dass das Modell absolut genau ist und die in der Simulation verwendeten Algorithmen keine Fehler in ihren Berechnungen aufweisen. Dies ist jedoch aus mehreren Gründen nicht möglich. Einige resultieren aus den Beschränkungen der Simulation, wie z. B. Rechenbeschränkungen oder numerischen Diskretisierungen. Andere werden durch fehlende Informationen verursacht, wie zum Beispiel Messungenauigkeiten oder fehlende Daten. Außerdem muss die Komplexität des Modells oftmals reduziert werden, um die Simulation erst möglich zu machen, obwohl dies zusätzliche Unschärfe in die Simulation einführt. Diese Unschärfe hat viele Quellen und kann entweder implizit berücksichtigt werden, indem Sicherheitsmargen bei den Ergebnissen eingerechnet werden, oder sie kann explizit berücksichtigt werden, indem sie modelliert und in die Simulation einbezogen wird. Das Ziel dieser Arbeit, welche im Rahmen von Teilprojekt 14 des Schwerpunktprogramms SPP 1886 ‘Polymorphe Unschärfe-Modellierung für die numerische Gestaltung von Strukturen’ entstand, ist es, einen effizienten Algorithmus zur Berücksichtigung polymorpher Unschärfe bei der Simulation des menschlichen Gangs mit einer Fußprothese zu liefern.

Um dies zu erreichen, müssen drei Aspekte kombiniert werden. Erstens muss ein geeignetes Modell entwickelt werden. In dieser Arbeit wird das menschliche Bein mit einer Fußprothese als Mehrkörpersystem mit starren und flexiblen Körpern modelliert. Zweitens muss die Unschärfe, die in der Simulation berücksichtigt werden soll, formalisiert, quantifiziert und modelliert werden. Dann muss das Simulationsverfahren erweitert werden, damit es die Unschärfe durch das Modell propagieren kann. Schließlich müssen die Ergebnisse visualisiert und interpretiert werden. In dieser Arbeit werden zwei Arten von Unschärfe berücksichtigt. Der Graph Follower algorithm wird verwendet, um epistemische Unschärfe durch das entwickelte Bein- und Fußprothesenmodell zu propagieren, bevor das Unschärfemodel auf polymorphe Unschärfe in Form von fuzzy random variables erweitert wird, die über den neu entwickelten Fuzzy Random Variable Graph Follower algorithm (FRV-GFA) propagiert wird. Somit ermöglicht diese Arbeit eine Simulation der Vorwärtsdynamik

des menschlichen Beins mit einer flexiblen Prothese für zwei unterschiedliche Gangphasen, nämlich die Schwung- und Standphase, bei der parametrische Unschärfe explizit berücksichtigt wird. Hierfür wird die Effizienz des bestehenden Graph Follower algorithm erheblich gesteigert, damit epistemische Unschärfe durch das Modell propagiert werden kann. Die Hauptbeiträge dieser Arbeit sind die Entwicklung eines neuen Modells für Fehler in der Markerposition während der optischen Bewegungsmessung und ein effizienter Algorithmus, basierend auf dem Graph Follower algorithm, der polymorphe Unschärfe, modelliert mit fuzzy random variables, durch das komplexe nichtlineare Mehrkörpersystem eines menschlichen Beins mit einer flexiblen Fußprothese propagieren kann.

Schlüsselwörter:
polymorphe Unschärfe – Fuzzy Random Variable Graph Follower algorithm – fuzzy random variables – epistemische Unschärfe – Graph Follower algorithm – Vorwärtsdynamik – Mehrkörpersysteme – Variationsintegratoren – vorgekrümmter geometrisch exakter Balken – Fußprothese – Ganganalyse – markerbasierte optische Bewegungsmessung

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Simulations have become integral to modern engineering practices, providing a vital means for engineers to test and optimise designs without the need for costly and time consuming physical prototypes or extensive experimental trials. Furthermore, simulations offer engineers the flexibility to explore diverse scenarios, evaluate trade-offs, and pinpoint potential risks, ultimately contributing to an improved quality and enhancing innovation. Despite their advantages, simulations encounter challenges related to the validity and accuracy of the underlying models and data. Assumptions, simplifications, and approximations that are unfortunately inherent in simulations may not always accurately reflect the complex real-world behaviour or conditions of the examined systems and introduce uncertainty into the simulation and its predictions. This uncertainty is further increased in cases where simulations can not capture all aspects of intricate systems, for instance human and environmental factors or unforeseen events. These factors add another layer of complexity to the modelling and simulation process. While the question, whether we can know everything with absolute precision is an interesting philosophical question, it does not help when classifying or dealing with uncertainty. As Professor John Allen Paulos said in 1945 [Pau03],

> "Uncertainty is the only certainty there is, and knowing how to live with insecurity is the only security."

In a deterministic simulation and subsequent analysis of structures, a model maps input variables to desired output variables. Various parameters may be present in the model and influence the results. Every model is based on assumptions or idealisations of the problem and the environment in which it lives. However, the truth content of a statement based on the simulation results is limited by the model's level of detail and by the accuracy to which the parameters and input variables are known. Both of these are subject to uncertainty, which is in essence a lack of knowledge. According to, e. g., [Möl00] and [Möl04], model uncertainty is an uncertainty in the mapping itself. Thus, an uncertain model leads to uncertain model responses even for crisp input variables. On the other hand, for uncertain input data, even a deterministic model yields uncertain responses. The reasons for uncertainty are generally

very complex and depending on their particular cause, different types of uncertainty and specialised algorithms for their consideration are necessary.

This thesis is a result of subproject 14 of the German Research Foundation's SPP 1886 Priority Program [SPP], which delves into the development of models and structure-preserving methods for simulating fuzzy biomechanical optimal control problems in the presence of uncertainty. The overarching goal, spanning Phases I and II, is to ensure the reliable prediction of human movement with prostheses by incorporating uncertainties that are present in material properties and human gait. The research methodology involves conducting experiments with a specific foot prosthesis, measuring material properties, and modelling them as fuzzy material parameters. Additionally, gait data is recorded and analysed to establish an uncertain model of a human leg. The primary focus lies in advancing simulation methods for forward dynamics and optimal control to accommodate polymorphic uncertainty for analysing prosthetic structures throughout their design and life cycle. Dealing with the computational complexity stemming form the uncertainty is one of the main challenges within this research.
Addressing the core problem, this thesis concentrates on the biomechanical simulation of a human leg with a prosthetic foot while explicitly considering polymorphic uncertainty in the model's parameters. Like many biomechanical systems, simulating the human leg with a prosthesis offers numerous advantages, enabling the calculation of quantities that are challenging or impossible to measure in vivo. For instance, the internally stored energy in a prosthesis, which is linked to the perceived walking comfort of patients, is challenging to measure during use. Simulations offer a valuable option for obtaining insights into such quantities. These can be used to further improve prosthesis design. To enable the simulation of gait with a prosthesis, a mathematical model of the human leg and prosthesis, incorporating multibody dynamics and predeformed geometrically exact beam theory, is developed. The model is then simulated using variational integrators known for their excellent long-term energy behaviour, which makes them especially suitable for energy-related problems. However, being a biomechanical system, there are many sources and causes for uncertainty. Uncertainties in biomechanical systems arise from individual variations in requirements, applications and anthropomorphic measures, as well as from the materials used for prostheses and their associated manufacturing processes.

Finding a balance between considering the most relevant uncertainties and the model's abstraction from reality is crucial. Furthermore, developing suitable algorithms to propagate uncertainty through the model is essential, especially considering the high computational cost typically associated with uncertainty analysis. This becomes even more pronounced when dealing with a non-linear multibody model consisting of rigid (leg) and flexible (prosthesis) bodies, where there are many sources of uncertainty.

This thesis progresses the stated goal with four contributions. In Phase I of the project, the Graph Follower algorithm was developed and applied to different structures, see [Eis19a; Eis19b; Piv19]. The work for this thesis began with the development of a prosthesis model of the Össur Vari-Flex® [Sch19] based on the reverse engineering work done in [Soe19]. Based on this, the first main contribution of this thesis is the development of the leg and prosthesis model, including consideration of the layered structure of the prosthesis via homogenisation. With this model, two distinct gait phases can be simulated with forward dynamics, see [Sch21b; Sch21a]. Then the Graph Follower algorithm was applied to the model [Sch22b] which required the development of substantial optimisations of the algorithm's performance, which are the second major contribution. While measuring human gait with optical marker based motion capture to quantify the underlying uncertainty of the measurement in [Alj20; Hei21] a new error model for marker position errors was developed and proposed in [Sch24b], using the capabilities of the Graph Follower algorithm to efficiently propagate epistemic uncertainty in the form of fuzzy numbers through the joint angle calculations of real gait measurements. This new error model is the third contribution. However, the project's goal is to consider polymorphic uncertainty during gait with a prosthetic foot. The fourth contribution is the development and improvement of the Fuzzy random variable Graph Follower algorithm(FRV-GFA), which is capable of propagating polymorphic uncertainty in the form of fuzzy random variables. A first suggestion for this algorithm, which builds on the already available Graph Follower algorithm, was proposed in [Sch24a] and the algorithm was further improved in this thesis. This work summarises all of the developments and the required theories and introduces an improved version of the FRV-GFA which is capable of propagating polymorphic uncertainty modelled with fuzzy random variables in the model's parameters though the forward dynamics simulation of the human leg with a prosthetic foot for two distinct gait phases.

Outline

This thesis has three different aspects to consider. Firstly, uncertainty has to be quantified and modelled so it can be propagated through a simulation and suitable algorithms for that propagation are required. Secondly, a model of the human leg and of human gait have to be developed. Lastly, the prosthetic foot has to be modelled and a viable forward dynamics simulation of the model has to be implemented. Each of these is their own distinct field. Following this, the first three chapters of this thesis each introduce one field, with its own motivation, literature review, theory and how it pertains to the rest of this work. Then, once the individual parts have been described, their assembly to a whole and the different results are shown. Chapter 2 introduces uncertainty in

general, how it can be classified, how it is quantified for subsequent simulations and the polymorphic uncertainty model for this work is introduced. Chapter 3 describes the lower limb anatomy of humans and the gait cycle before optical marker based motion capture is introduced and a novel error model for marker position uncertainty is proposed in Section 3.5. Foot prostheses are briefly summarised in Chapter 4, before the biomechanical model developed and used in this work is described. This includes the theory required for understanding the forward dynamics simulations. Finally, the application and results of different combinations of the uncertainty theory and models are shown in Chapter 5 before summarising the results produced during this thesis and providing an outlook for the project continuation in Chapter 6.

Scientific contributions

During this thesis, two peer-reviewed journal papers were produced (*Fuzzy forward dynamics of distinct gait phases with a prosthetic foot* [Sch22b; Sch22a], *Marker position uncertainty in joint angle analysis for normal human gait – a new error modelling approach* [Sch24b]) and the results were presented at national and international scientific conferences, (SPP1886 annual meetings, GAMM2020 & 2021, WCCM2022, ICIAM2023) including the published proceedings [Sch21b; Sch21a]. Four student theses [Alj20; Hei21; Hüb22; Blo23] were supervised. Two contributions were made to the book [Kal24]. Chapter [Sch24a] summarises the entire SPP subproject 14 up to 1st of July 2024, which this thesis is part of. A contribution was also made to the Chapter [Sch24c].

Use of previously published works

Since this thesis aims to present the whole picture of my research at the Institute of Applied Dynamics (Lehrstuhl für Technische Dynamik, LTD), some sections quote from my previous works. This is noted at the beginning of every quoted section and summarised here for reference. Section 2.5.2 quotes from [Sch24a]. Section 2.6.1 quotes from [Sch22b]. Section 2.6.2 quotes from [Sch22b]. Section 2.6.3 quotes from [Sch22b]. Section 2.7.1 quotes from [Sch24a]. Section 2.7.2 quotes from [Sch24a]. Section 3.7.1 quotes from [Sch24b]. Section 3.7.2 quotes from [Sch24b]. Section 4.2 quotes from [Sch22b]. Section 4.2.4 quotes from [Sch22b]. Section 5.2 quotes from [Sch22b]. Section 5.3 quotes from [Sch22b]. Section 5.5 quotes from [Sch24b].

Chapter 2

Working with uncertainty

The main focus of this work is the creation of an algorithm capable of propagating input parameters affected with polymorphic uncertainty though a complex multibody forward dynamic simulation with rigid and flexible bodies. To do this, it is necessary to first understand what uncertainty is and what it's sources and causes are. This chapter details the different types of uncertainty, which type is suited to what situation and how they can be combined to form polymorphic uncertainty. Each uncertainty type introduction is followed by methods of propagating that uncertainty, including the algorithms that are improved or developed during this work.

Uncertainty refers to the range of possible values that a parameter, measurement, statement or calculation could have due to errors or unknown factors. For example, when trying to guess how many objects X are in a volume Y, you might have a pretty good idea, but there's always a chance you could be off by a few (or a lot). This unknown possible gap between your guess and the true value is called uncertainty. Of course, in engineering, having a good guess is not really sufficient, especially when safety is a concern. Thus, before methods can be introduced to deal with uncertainty, the different theories are briefly introduced and the models for uncertainty used in this work are described.

Uncertainty is present in manifold ways and is caused by a variety of effects. For instance, [Oha04] detailed some of the uncertainty in computer simulations, such as model uncertainty or sample size in Monte Carlo simulations, and [Kiu09] discusses the question whether the classification of uncertainty matters. In general, uncertainty can be understood as a lack of knowledge.

Previous works Parts of this chapter are based on previously published works. Specifically, the Sections 2.5, 2.6.2 and 2.6.3 are from my previous publication [Sch22b]. Section 2.7 is published in part in [Sch24a].

2.1 Types of uncertainty

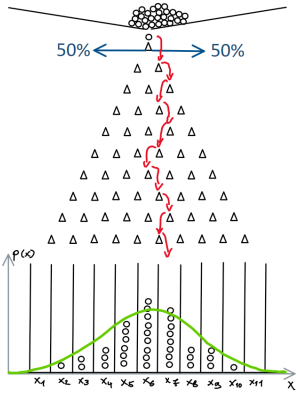
The most common way to classify uncertainty is by the cause for said uncertainty. Generally, three causes are recognised in literature.

Uncertainty caused by variability is usually modelled with stochastic quantities and is called aleatoric uncertainty. It is a type of uncertainty that arises from the inherent randomness or variability of a phenomenon and is also known as statistical or irreducible uncertainty, as it cannot be reduced by collecting more data or improving the model. Classical examples include an ideal dice throw or a Galton board, invented by Francis Galton in 1873 [Gal89] and sketched in Figure 2.1a. It consists of a reservoir of beads on the top from which the beads drop onto a pyramid of pegs. When a bead drops onto a peg, it either continues downward on the left or right side and hits the next peg below, again either going to the left or right. This is repeated for many beads over multiple rows of pegs and the beads are gathered in bins below the final row of pegs. If enough beads are used, the typical bell curve emerges from the stacking height of the beads in the bins. The Galton board is often used to demonstrate the central limit theorem, an important theorem when dealing with aleatoric uncertainty. This type of uncertainty can be represented with probability theory which relies on a probability measure, often a probability distribution, to describe or quantify the uncertainty. This probability measure indicates how likely an event is to occur.

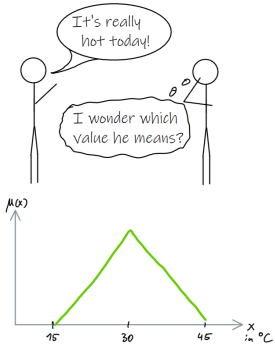
Uncertainty caused by inaccuracy or imprecision is usually modelled with fuzzy quantities, such as fuzzy numbers, and is called epistemic uncertainty. Inaccuracy refers to the deviation of a measurement or value from the true or correct value, while imprecision refers to the lack of consistency or repeatability in the measurements [Tra08]. The underlying idea of this type of uncertainty is, that while a quantity could be known exactly, the required effort to obtain that level of precision or accuracy is not feasible or the quantity is affected by subjective factors. Thus, this type of uncertainty is considered reducible. In contrast to the probability measure, this type of uncertainty uses a possibility measure, which indicates the truth or plausibility of a statement, not its probability. Epistemic uncertainty is caused by factors such as systemic biases or procedural limitations, i. e., accuracy limitations on the measurement equipment. It is also commonly used to model uncertainty associated with language where words allow for interpretation of the specific meaning, see Figure 2.1b.

Uncertainty caused by incompleteness is usually modelled with fuzzy or polymorphic quantities, a combination of stochastic and fuzzy quantities. The underlying idea of this type of uncertainty is, that there is a knowledge gap about the process itself or where exactly the uncertainty may stem from.

The main difference between possibility theory and probability theory is that they use different types of uncertainty measures. Specifically, possibility theory uses a measure, called the membership function, that ranges from 0 to



(a) Sketch of a Galton board used to demonstrate the bell curve of the normal distribution.



(b) Sketch of how language is not always precise. A specific temperature value is certain, but the word "hot" is open to interpretation.

Figure 2.1: Two different causes for uncertainty.

1 and defines the membership of a function or variable in the set. In contrast, probability theory uses one additive measure that sums up to 1, e. g., the cumulative probability function. Possibility theory uses fuzzy sets that allow partial membership, while probability theory uses crisp sets that allow only full membership. Possibility theory relies on expert knowledge or linguistic expressions, while probability theory relies on statistical data or mathematical models. However, possibility theory and probability theory are not incompatible or contradictory. They can be seen as complementary ways of handling uncertainty, depending on the context and the type of information available.

The "correct" uncertainty measure for a given situation is still an open question. In this work, the uncertainty is modelled with fuzzy numbers for epistemic uncertainty and the uncertainty model is later expanded to a polymorphic model with fuzzy random numbers, which combine probability density functions with a method for discretising fuzzy numbers to form the polymorphic uncertainty model.

2.2 A note on the mathematical notation of uncertainty

Parametric uncertainty, as is examined in this work, refers to the lack of knowledge what specific value a parameter p has for a given simulation or

measurement. It is assumed, that the value p is within in a larger space P such that $p \in P \subseteq \mathbb{R}$. As mentioned, to consider the uncertainty in simulations it has to be quantified with an uncertainty measure. For aleatoric (stochastic) uncertainty, the uncertainty is quantified via the probability density function F_{pdf}. Thus, a random variable $\hat{p}$ consists of all pairs $\hat{p} = \{p, F_{\text{pdf}}(p) \mid p \in P\}$. In contrast, epistemic uncertainty which is modelled with fuzzy numbers here uses a membership function $\mu_{\tilde{p}}$ as its uncertainty measure, which asses the "truth" of a value. A fuzzy number $\tilde{p}$ then consists of $\tilde{p} = \{p, \mu_{\tilde{p}}(p) \mid p \in P\}$.

2.3 Quantifying aleatoric uncertainty

Aleatoric uncertainty can be modelled with stochastic quantities, such as random variables or random processes, that are commonly described with probability distributions.

Specifically, a random variable is a mathematical representation of a random event. This means it's value is not known, until the event is observed, termed a realisation of the random variable. For example, a dice roll is a random event with six possible outcomes, namely one, two three, four, five or six. Each of these values is a possible realisation of the random event. For an ideal fair dice, it is assumed that each of these has the equal probability of $\frac{1}{6}$. Of course, this is a very common and simple example of randomness. For more complex probabilities, probability density functions (PDFs), denoted with F_{pdf}, are used to describe and quantify the realisations of the random variable. These distributions mathematically describe the likelihood or probability of different outcomes occurring in a given set of events or experiments and can be continuous or discrete. The probability for the random variable to fall within a particular interval is given by the integral of this variable's PDF over that interval. Sometimes, instead of a PDF a cumulative distribution function (CDF) is provided. While a PDF gives the probability of a specific value or event to occur, a CDF provides the cumulative probability that a specific value or event of a random variable is smaller or equal to a certain value. Thus, it is the integral of the PDF. A random process includes the aspect of time in a random variable, meaning the probability function is both a function of the random variable and time.

Two statistical measures that provide insights into the aleatoric uncertainty's behaviour when represented by a PDF are the mean $\mu_{\hat{p}}$ and standard deviation σ. The mean μ, also called the expected value, is a measure of the central tendency of a probability distribution. The mean of the random variable $\hat{p}$ is given by

$$\mu_{\hat{p}} = \mu[\hat{p}] = \int_{-\infty}^{\infty} p F_{\text{pdf}}(p) \mathrm{d}p, \tag{2.1}$$

where $F_{\text{pdf}}(p)$ is the probability density function at the specific realisation p of the random variable $\hat{p}$. A low standard deviation indicates that the values are likely close to the mean, while a high standard deviation indicates that the values are likely to be spread out over a wider range. Thus, the standard deviation σ is a measure of how far values are dispersed relative to the mean. The standard deviation of $\hat{p}$ is

$$\sigma_{\hat{p}} = \sqrt{\int_{-\infty}^{\infty} (p - \mu_{\hat{p}})^2 F_{\text{pdf}}(p) \mathrm{d}p}. \tag{2.2}$$

The variance of $\hat{p}$ is $\sigma_{\hat{p}}^2$.

The selection of an appropriate PDF for a given scenario greatly affects the outcome and is an important part of the uncertainty modelling and quantification process. Each of these distributions has a specific formula that describes the probability of observing a certain value or range of values of the stochastic quantity.

The normal distribution The normal or Gaussian distribution $N(\mu, \sigma)$ is defined by the PDF on the interval $p \in P = (-\infty, \infty)$

$$F_{\text{pdf}}(p) = \frac{1}{\sigma\sqrt{2\pi}} e^{\xi} \quad \text{with} \quad \xi = \left(-\frac{(p - \mu_{\hat{p}})^2}{2\sigma_{\hat{p}}^2} \right), \tag{2.3}$$

where $\mu_{\hat{p}}$ is the mean and $\sigma_{\hat{p}}$ is the standard deviation of the random variable $\hat{p}$ and results in the typical bell curve, shown in Figure 2.1a and Figure 2.2. The normal distribution is often used to model uncertainty caused by multiple subsequent uncertain processes. Why the normal distribution is so common can be explained by the central limit theorem.

Central limit theorem The central limit theorem (CLT), states that the distribution of the mean of the sum of independent and identically distributed (i.i.d.) random variables with finite variance converges to a normal distribution as the sample size increases [Pis14]. The independent requirement means that drawing a sample does not affect the outcome for future drawings, e. g., tossing a coin and getting heads does not affect the next toss. In contrast, drawing coloured balls from an urn changes the probability of drawing future balls of a specific colour. The 'identically distributed' requirement means that the random variables share the same distribution. The CLT has many applications in statistics and can be stated as follows. Let $\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_n$ be a sequence of i.i.d. random variables with mean $\mu < \infty$ and finite variance $\sigma^2 < \infty$, e. g., the sum of the outcome of the throw of n dice (an outcome

being the number of dots facing upward). The normalised sum is defined as

$$\bar{\hat{p}}_n = \frac{(\hat{p}_1 + \hat{p}_2 + \cdots + \hat{p}_n) - n\mu}{\sigma\sqrt{n}}. \tag{2.4}$$

Note that when adding two i.i.d. random variables $\hat{p}_1$ and $\hat{p}_2$, the mean and variance of the sum are $\mu_{\hat{p}_1+\hat{p}_2} = \mu_{\hat{p}_1} + \mu_{\hat{p}_2}$ and $\sigma^2_{\hat{p}_1+\hat{p}_2} = \sigma^2_{\hat{p}_1} + \sigma^2_{\hat{p}_2}$ [Pis14]. Then, as $n \to \infty$, the CLT states that

$$\lim_{n\to\infty} F_{\text{pdf}}(\bar{p}_n) = N(0,1), \tag{2.5}$$

where $N(0,1)$ denotes the standard normal distribution with $\mu = 0$ and $\sigma = 1$. In other words, if many ($n \to \infty$) independent random variables are added, the probability distribution of the resulting normalised sum resembles the standard normal distribution. The CLT was first formulated by Abraham de Moivre in 1733 for the case of binomial distribution, and later generalized by Pierre-Simon Laplace in 1810. The modern version of the CLT for i.i.d. random variables was proved by Aleksandr Lyapunov in 1901 using his famous Lyapunov's central limit theorem. There are also variants of the CLT that relax the assumptions of identical distribution, such as Lindeberg's central limit theorem. For more detail see [Pis14; Bil95] or, for an excellent summary video, [San23]. In short, the CLT describes that if many independent random variables are combined, a normal distribution emerges and thus explains why the normal distribution is so common.

The beta distribution The beta distribution is a continuous probability distribution that has two parameters, typically denoted by α and β. In contrast to the normal distribution, the Beta distribution is defined for parameter values in the interval $p \in [0,1]$. It can be scaled for other values via multiplication. The probability density function of the beta distribution $B(\alpha,\beta)$ is given by

$$F_{\text{pdf}}(p) = \frac{p^{\alpha-1}(1-p)^{\beta-1}}{G(\alpha,\beta)} \quad \text{with} \quad G(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)} \tag{2.6}$$

where α and β are the shape parameters that determine the shape of the distribution. $G(\alpha,\beta)$ serves as a normalization constant to ensure that the total probability integrates to 1. The beta function is defined in terms of the gamma function $\Gamma(x) = (x-1)!$. The mean of the beta distribution is given by

$$\mu_{\hat{p}} = \frac{\alpha}{\alpha+\beta}, \tag{2.7}$$

and the standard deviation is

$$\sigma_{\hat{p}} = \sqrt{\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}}. \tag{2.8}$$

The Beta distribution is very flexible and can take on many different shapes depending on the values of α and β.

Further examples of common PDFs

There are many other distributions, see for instance [Mun15], like the Uniform distribution or the exponential distribution.
The Uniform distribution assigns equal probability to all values within a designated interval. For example, a random variable $\hat{p}$ uniformly distributed on the interval $[a, b] \in \mathbb{R}$ has the PDF

$$F_{\text{pdf}}(p) = \frac{1}{b-a} \tag{2.9}$$

for $a \leq p \leq b$ and $F_{\text{pdf}}(p) = 0$ elsewhere. Note that for discrete random variables with n possible values, like a dice throw ($n = 6$), the PDF is $F_{\text{pdf}}(p) = \frac{1}{n}$.
The uniform distribution is apt for scenarios devoid of favouritism or bias, such as ideal dice rolls or random number selection.
The exponential distribution exhibits a decreasing curve from a positive value at zero, the exponential distribution's PDF is

$$F_{\text{pdf}}(p) = \lambda e^{-\lambda p} \tag{2.10}$$

for $p \geq 0$ and $F_{\text{pdf}}(p) = 0$ elsewhere, where λ is the rate parameter. This distribution aptly models events with a constant probability per unit time, such as the inter-arrival times of customers, the lifespan of a light bulb, or the decay of radioactive atom. These examples merely scratch the surface, as numerous other PDFs exist, each tailored to model various phenomena with distinct characteristics.

It should be noted, that most probability distributions, like for instance the Normal distribution are infinite. To deal with these numerically, they can be truncated to include a desired percentile, i. e., for a distribution including the 99th percentile, only the interval from 1% likelihood to 99% is included. Some distributions, like the beta distribution, are limited to $[0, 1]$. Depending on the data and problem, a suitable distribution has to be chosen. Fitting the "right" distribution to the available data is a common problem in statistics and is only mentioned here as a task during uncertainty quantification, since it would go beyond the scope of this work.

Sources of aleatoric uncertainty

In this work, engineering and medical aspects need to be considered. In engineering, aleatoric uncertainty can arise from the variability of material properties, loading conditions, environmental factors or manufacturing tolerances, while in medicine, the uncertainty can arise from the variability of patient's anthropomorphic measures, a disease's cause or symptoms and the patient's preferences.

2.4 Propagating aleatoric uncertainty

Once quantified, aleatoric uncertainty can be propagated using various methods. These methods can help estimate the range and likelihood of possible outcomes, as well as the sensitivity and robustness of the model to aleatoric uncertainty. Here, the goal of propagating aleatoric uncertainty is to calculate the uncertain target output or target function f based on the uncertain input parameter $\hat{p}$ with the resulting uncertain target output $\hat{f}(\hat{p}) = \{f(p), F_{\text{pdf}}(f(p)) \; \forall \; p \in P\}$. An example of a target output is how far a cantilever beam deforms in one direction, based on a stochastic distribution of the beam's Young's modulus, a material parameter. Whatever method is used to propagate the uncertainty, one thing all of these methods have in common, is that they require samples to be drawn or generated from the PDF.

Sampling

Sampling is a crucial step in dealing with stochastic uncertainty [Ram98]. Basically, sampling means the process of randomly selecting or drawing a specific sample from a given data space. When simulating with stochastic uncertainty, where the input data is governed by a probability distribution or probability density function, sampling refers to obtaining a specific value of this function, with the aim, that if many samples are drawn, the distribution is well represented. Figure 2.2 shows the main problem with sampling. To accurately represent the underlying distribution, a lot of samples are necessary. The same probability distribution is used across all examples for the sampling in Figure 2.2. Then, a progressively increasing number of samples is drawn from this distribution and used as input data for distribution fitting, i. e., the process of calculating the "best" possible probability distribution for these data. Of course, the definition of "best" varies, depending on the use case and available data. As can be seen in Figure 2.2, if only 10 or 100 samples are used for the fitting, the difference between the generating distribution's mean and standard deviation and the fitted distribution's mean and standard deviation are quite large. However, as more samples are generated and used for the fitting, this difference shrinks. Here, Matlab 2023b's *fitdist* function is

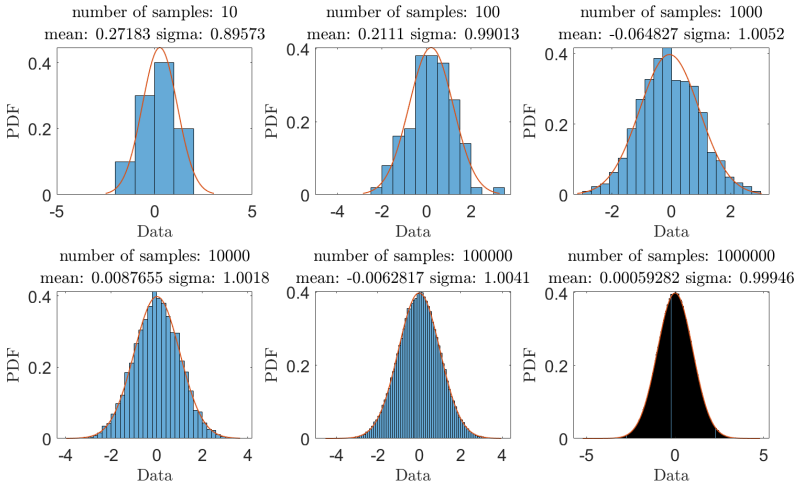


Figure 2.2: Comparing the effect of different sample sizes on the resulting fitting. All samples are generated from the same initial probability distribution, a normal distribution with the mean $\mu = 0$ and standard deviation $\sigma = 1$. The more samples are used for the fitting, the better the match between the initial and fitted distribution. The red line shows the fitted PDF and a histogram of the samples is shown in blue.

used for the fitting of a Normal distribution. *fitdist* uses an automatic binning algorithm to determine the number of bins in the plot. This dependency on sample size is the main reason why stochastic analyses are computationally very expensive. This problem is further compounded, if propagating a sample through a model is also computationally expensive.

However, sampling methods vary in their applicability, accuracy and efficiency depending on the characteristics of the uncertainty distribution and the problem structure. Here, only some examples are shown, to highlight the importance of considering the sampling method when dealing with aleatoric uncertainty. Some possibilities that do not rely on purely random sampling are Markov chain Monte Carlo simulations or Latin Hypercube sampling.

Monte Carlo simulation

A Monte Carlo simulation is a method used to propagate aleatoric uncertainty through a model [Cro20]. In this method, each input variable of the model is randomly sampled from the governing probability distribution and the output of the model is calculated for each sample. If this is repeated many times,

the uncertainty in the output can then be estimated by statistical analysis of the resulting output distribution. It is evident, that the sampling process has a large effect on the results and computational efficiency. Monte Carlo simulations generally require a large number of samples to achieve the desired accuracy, and without advanced sampling methods, problem-specific information is not exploited to reduce the computational cost.

Markov chain Monte Carlo simulations

A Markov chain is a stochastic model describing a sequence of possible events in which the probability of each event depends only on the state attained in the previous event [Chu67]. The defining characteristic of a Markov chain is that no matter how the process arrived at its present state, the possible future states are fixed. Markov chain Monte Carlo simulations (MCMC) are a class of algorithms for sampling from probability distributions based on constructing a Markov chain that has the desired distribution as its equilibrium distribution, see [Gil95b; Gey11; Gil95a]. The state of the chain after a large number of steps is then used as a sample of the desired distribution. MCMC simulations allow for parameter estimation such as means and variances.

Latin Hypercube sampling

Latin Hypercube sampling (LHS) is a statistical method for generating a near-random sample of parameter values from a multidimensional distribution, see for instance [Mck79; Hel03; Men23]. It is often used to construct computer experiments or for Monte Carlo integration, a numerical integration method for definite integrals which is particularly useful in higher dimension and thus benefits from the LHS method. The advantage of LHS is that it ensures a more even distribution of sample points in the parameter space than would be possible with pure random sampling.

In conclusion, sampling methods are essential tools for stochastic uncertainty, but they have different strengths and limitations depending on the nature of the uncertainty and the problem. Therefore, it is important to choose an appropriate sampling method that can capture the relevant features of the uncertainty and provide reliable estimates of the solution.

2.5 Quantifying epistemic uncertainty

Epistemic uncertainty can be modelled with fuzzy quantities, such as fuzzy sets. These were introduced by Zadeh [Zad65] in 1965 and expand classical set theory. In a classical set, membership is binary. If we denote the membership of an element p in the classical set P with a membership function $\mu_{\tilde{P}}(p)$, it

has two possible values. Either an element p is in the set $\mu_{\tilde{p}}(p) = 1 \mid p \in P$ or it is not $\mu_{\tilde{p}}(p) = 0 \mid p \notin P$. For fuzzy sets, the membership of an element p is no longer binary but is governed by a membership function $\mu_{\tilde{p}}(p)$. In [Möl07], "a fuzzy variable $\tilde{p}$ is defined as an uncertain subset of a fundamental set P"[1]

$$\tilde{p} = \{p, \mu_{\tilde{p}}(p) \mid p \in P\}, \tag{2.11}$$

where the uncertainty is defined by the membership function $\mu_{\tilde{p}}(p)$. An example of a fuzzy number is shown in the middle of Figure 2.4. To qualify as a fuzzy number, a fuzzy variable has to fulfil three conditions, namely the membership function has to be normalised

$$0 \leq \mu_{\tilde{p}}(p) \leq 1 \quad \forall \quad p \in P \tag{2.12}$$

and it is required to fulfil convexity

$$\mu_{\tilde{p}}(p_2) \geq \min\left[\mu_{\tilde{p}}(p_1); \mu_{\tilde{p}}(p_3)\right] \quad \forall \quad p_1, p_2, p_3 \in \mathbb{R} \text{ with } p_1 \leq p_2 \leq p_3. \tag{2.13}$$

Finally, the membership function of a fuzzy number can have the value of $\mu_{\tilde{p}}(p_{\text{peak}}) = 1$ only for one value of p_{peak}, also referred to as the peak point.

2.5.1 α-discretisation of fuzzy numbers

A convex fuzzy number can be described with the help of α-level discretisation. The α-level cuts or α-level sets $P_\alpha = [p_{\alpha l}; p_{\alpha r}]$ are defined by

$$p_{\alpha l} = \min[\ p \mid \mu_{\tilde{p}}(p) \geq \alpha] \tag{2.14}$$

$$p_{\alpha r} = \max[\ p \mid \mu_{\tilde{p}}(p) \geq \alpha], \tag{2.15}$$

see for instance [Möl07; Möl04]. The set

$$S_{\tilde{p}} = [p \mid \mu_{\tilde{p}}(p) > 0] \tag{2.16}$$

is referred to as the support of the fuzzy variable and and is referred to as an α-level set with $\alpha = 0$ with the interval boundaries

$$p_{\alpha l} = \lim_{\bar{\alpha} \to +0} [\min[\ p \mid \mu_{\tilde{p}}(p) > \bar{\alpha}]] \quad \text{for} \quad \alpha = 0 \tag{2.17}$$

$$p_{\alpha r} = \lim_{\bar{\alpha} \to +0} [\max[\ p \mid \mu_{\tilde{p}}(p) > \bar{\alpha}]] \quad \text{for} \quad \alpha = 0. \tag{2.18}$$

Due to the convexity of fuzzy numbers, the following relationship between α-level sets holds

$$P_{\alpha_{k+1}} \subseteq P_{\alpha_k} \quad \forall\ \alpha_{k+1}, \alpha_k \in [0; 1] \tag{2.19}$$

[1] The variable names have been changed to fit with the notation in this thesis.

with $\alpha_1 = 0 \leq \alpha_2 \leq \ ... \ \leq \alpha_{k+1} \leq \ ... \ \leq \alpha_k \leq \ ... \ \leq \alpha_{N_\alpha - 1} \leq \alpha_{N_\alpha} = 1$. This implies that a higher α-level set is contained in lower α-level sets.

2.5.2 $l_\alpha r_\alpha$-discretisation of fuzzy numbers

Based on the α-level discretisation introduced in Section 2.5.1, an alternative description of fuzzy numbers from [Möl07] is introduced, which is later used to describe polymorphic uncertainty. The following is quoted from my previous work [Sch24a] Section 2.2. Variable names in the text and figures have been changed to better fit this thesis. Figures and tables have been modified to suit the format of this thesis.
“

Fuzzy numbers can be described with $l_\alpha r_\alpha$-discretisations as shown in [Möl07] and visualised in Figure 2.3[2].

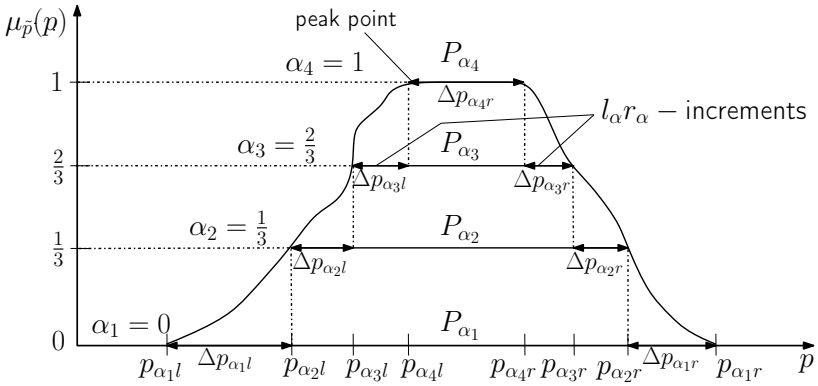


Figure 2.3: The $l_\alpha r_\alpha$-discretisation of a fuzzy number[2], based on [Möl07].

Starting from the peak point $p_{\alpha_{N_\alpha}}$, with $N_\alpha = 4$ in Figure 2.3, each following α-level is given by the $l_\alpha r_\alpha$-increments $\Delta p_{\alpha_i, l}$ and $\Delta p_{\alpha_i, r}$ as

$$\begin{aligned} p_{\alpha_i, l} &= p_{\alpha_{i+1}, l} - \Delta p_{\alpha_i, l} \\ p_{\alpha_i, r} &= p_{\alpha_{i+1}, r} + \Delta p_{\alpha_i, r}. \end{aligned} \tag{2.20}$$

For $\alpha = 1$, only a right increment exists. The increments can either be calculated from a given fuzzy number, based on the differences between the α-levels, or they can be used to define the fuzzy number from a given peak point by calculating subsequent α-levels from predefined increments. With the second

[2]Note that only for $\Delta p_{\alpha_4 r} = 0$, the fuzzy variable in Figure 2.3 is a fuzzy number.

method, which is later used to include randomness in the uncertainty model, the convexity of the fuzzy number has to be ensured via

$$\begin{aligned} \Delta p_{\alpha_i,l} &\geq 0 \quad \text{for} \quad i = 1, 2, ...N_\alpha - 1 \\ \Delta p_{\alpha_i,r} &\geq 0 \quad \text{for} \quad i = 1, 2, ...N_\alpha. \end{aligned} \tag{2.21}$$

”

2.6 Propagating epistemic uncertainty

Similar to the goal of propagating aleatoric uncertainty in Section 2.4, the goal of propagating epistemic uncertainty is to calculate the uncertain target output f based on the uncertain input parameter $\tilde{p}$, such that the resulting uncertain target output reads $\tilde{f}(\tilde{p}) = \{f(p), \mu_{\tilde{f}}(f(p)) \; \forall \; p \in P\}$. This target output can be any quantity of interest of the examined model, for instance a position or angle or something that is calculated based on the model, like internal deformation energy in a beam. With epistemic uncertainty, this means calculating the values for the target output and the membership function $\mu_{\tilde{f}}$. Epistemic uncertainty in the form of fuzzy numbers is propagated in this work with the Graph Follower algorithm. It was developed in [Eis19b; Eis19a] and improved for use with more complex models in [Sch22b]. The following sections summarise its function and the improvements that were produced during this thesis.

2.6.1 Summary of the Graph Follower algorithm

The following is quoted from my previous work [Sch22b] Section 3.1. Variable names in the text and figures have been changed to better fit this thesis. Figures and tables have been modified to suit the format of this thesis.
“

Uncertain parameters $\tilde{p}$ modelled by a triangular fuzzy number, as required for the Graph Follower algorithm, are defined by an interval $\tilde{p} \in [p_{min}, p_{max}]$ and their associated membership function $\mu_{\tilde{p}}(p) \in [0, 1]$. Only one value in the interval has the function value 1, shown in the middle of Figure 2.4, leading to a convex fuzzy number. To get a triangular fuzzy number, the membership function $\mu_{\tilde{p}}(p)$ is assumed to be linear from the extremes of the interval with the value $\mu_{\tilde{p}}(p_{min}) = \mu_{\tilde{p}}(p_{max}) = 0$ to the deterministic parameter $\mu_{\tilde{p}}(p_{det}) = 1$, resulting in a triangular shape for the membership function.

Propagating this uncertain parameter through a forward dynamics simulation has the goal to calculate both the resulting interval of a target output quantity $f(\tilde{p})$ as well as the corresponding membership function μ_f at any time, resulting in the fuzzy target output $\tilde{f}(\tilde{p})$. This results in an optimisation problem to find the extremes of the target output given the uncertain

input interval. The target output f represents a scalar quantity of interest that is calculated from the models trajectory $\mathfrak{q}(\tilde{p}, t)$ and the uncertain parameters $\tilde{p}$. The uncertain parameters $\tilde{p}$ either directly affect the target output or indirectly affect it, by only influencing the forward dynamics $\mathfrak{q}(\tilde{p}, t)$.

$$f = F(\mathfrak{q}(\tilde{p}, t), \tilde{p}, t) \tag{2.22}$$

To compute this optimisation on a computer, a discretisation of the problem is necessary. Using α-level cuts, it is possible to discretise the membership function and intervals, shown in the middle of Figure 2.4 for the the α-level cuts $\alpha = 0.0$ and $\alpha = 0.5$. For $\alpha = 1.0$ the interval of the fuzzy parameters reduce to a single parameter, p_{det}. For more details on this discretisation see for instance [Möl00; Möl04]. The Graph Follower algorithm computes the target output envelopes using α-level optimisation. This means the intervals of the target output function are calculated with an optimisation based on the input's fuzzy parameter interval, while the membership function is inferred from the chosen α-level of the uncertain input parameter, see [Möl00]. This can be done for a singular uncertain parameter or multiple, however, depending on the interaction of these parameters, further unintentional uncertainty may be introduced [Möl00]. The method of propagating fuzzy uncertainty with α-level optimisation is based on the extension principle for fuzzy numbers [Zad65] and Nguyen's note on the extension principle [Ngu78] and is detailed in e.g. [Möl04; Möl00] and only summarised briefly here. The goal of propagating a fuzzy uncertain parameter through a model is to calculate the fuzzy target output $\tilde{f}$, which means both the numerical values f as well as the associated membership function values μ_f. These can be calculated with α-level optimisation which discretises the input fuzzy numbers with α-level cuts and then performs an optimisation to find the extreme values of the target output f_{α_k} for the given α-level cut α_k. When the two extreme values of the target output at a given timestep are known for a given α-level cut, two points of the target output membership function μ_f are known since the values are inherited from the membership function value of the α-level cut due to Nguyen's principle. The target output fuzzy number $\tilde{f}$ can be computed by solving the following optimisation problem for each timestep and multiple α-level cuts as described for instance in [Eis19b; Piv19].

$$\tilde{f}^{(s)}_{\alpha_k}(\tilde{p}^{s,\alpha_k}) = -s \min_{p \in P_{\alpha_k}} (-sf(p)) \quad \text{with} \quad s = \mp 1 \tag{2.23}$$

Solving this optimisation problem includes the evaluation of the deterministic mapping of the target function f_d and thus for this work the forward dynamics simulation. The Graph Follower algorithm combines the α-level optimisation with an approximation of the forward dynamics simulation via Taylor expansion and post-processing steps to efficiently calculate the target

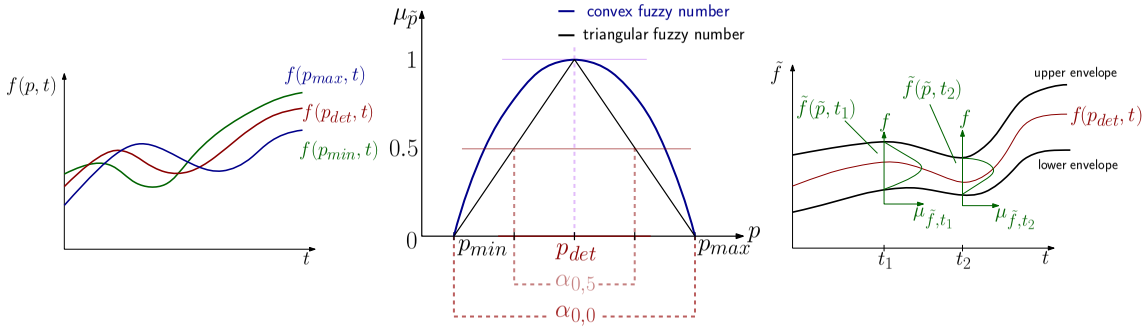


Figure 2.4: "Visualisation of the uncertainty propagation process using the Graph Follower algorithm. The target output quantity $f(\boldsymbol{q}(\tilde{p},t),\tilde{p},t)$ calculated from the discrete parameter interval values p_{min}, p_{det}, p_{max} leads to distinct evolutions of the target output (left). Including epistemic uncertainty in the form of a convex fuzzy number (middle) which is discretised into α-level cuts, the output envelopes of the target output can be calculated with the Graph Follower algorithm (right) resulting in the extreme values for the target output, as well as the corresponding membership function $\mu_{\tilde{p}}(p)$ at every time node." Caption quoted from [Sch22b]. Figure is based on [Sch22b].

output envelope and the target output's membership function over time, resulting in an approximation of the fuzzy target output. The process of forward propagation of epistemic uncertainty to calculate a target output envelope is visualised in Figure 2.4.

The Graph Follower algorithm from [Eis19b] employs two main methods to make α-level optimisation feasible with forward dynamics. The first is the linearisation of the forward dynamics [...] with respect to the last known optimal parameter, allowing for quick computations of the target output at the cost of accuracy. The second method, is the combination of storage and post-processing. When a new parameter is found by the optimisation, the forward dynamics are calculated with it and the target output is computed for the entire simulation time. Both the parameter as well as its associated target output evolution are stored in a library within one α-level. This library of previously computed parameters and target output trajectories is used to chose a parameter for the linearisation of the forward dynamics based on the currently known extreme target output trajectory. The parameter p that is associated with the currently known extreme target output trajectory is used for the linearisation to approximate the forward dynamics and allow an efficient optimisation. Furthermore, at the end of an α-level optimisation for a given α-level, all target output trajectories are compared and only the extreme values are transferred to the target output envelope. This is repeated with several α-levels, to define the membership function value for the envelopes.
"

The process of determining the hulls is visualised in Figure 2.5. The simulation model is an ideal planar pendulum and the scalar target function f is the pendulum's angle at each time node. Figure 2.5(a) shows how the optimisation calculates different evolutions of the target function at different timesteps for one α-level, in this case $\alpha = 0.5$. Once all optimisations are done, the post-processing examines all available trajectories and calculates the upper (Figure 2.5(a) green circles) and lower (Figure 2.5(a) red circles) hull for this α-level. This is repeated for multiple α-levels to obtain the result shown in Figure 2.5(b), which shows the calculated approximation of the fuzzy target output over time.

To summarise, the Graph Follower algorithm propagates an uncertain parameter, that is quantified as a fuzzy number $\tilde{p}$, though a model to calculate an approximation of the fuzzy target output for every time node of the simulation. The target output can be any scalar quantity derived from the system's simulation. The Graph Follower algorithm approximates the fuzzy target output by discretising the input fuzzy number with α-levels and calculating the hulls for each of these levels with α-level optimisation. Once the hulls resulting from the multiple α-levels are known, so is the corresponding membership function of the target output and the fuzzy target output is approximated. The main advantage of using he Graph Follower algorithm is the efficiency in-

crease due to linearising the forward dynamics paired with a post-processing step which improves the resulting hulls based on all previous forward dynamics calculations within one α-level. For more details, see [Eis19a; Eis19b]. The main steps are summarised in the list 'Graph Follower algorithm steps', including the improvements introduced in the following section.

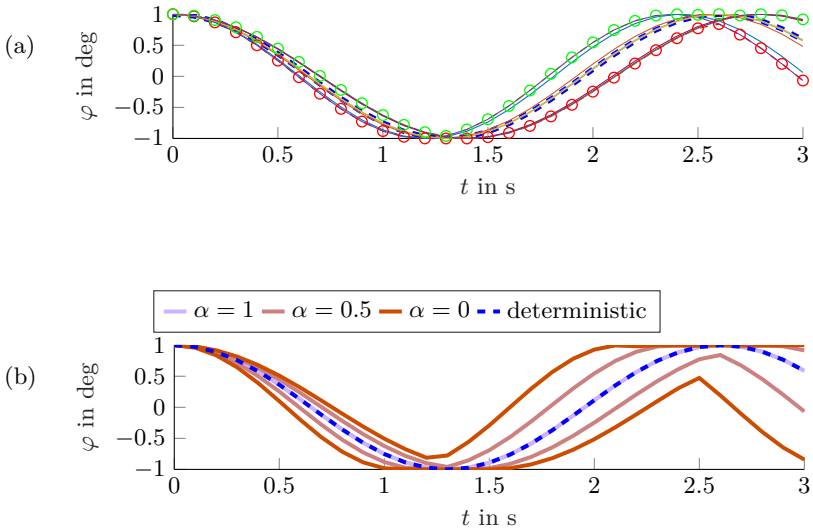


Figure 2.5: Visualisation of how the Graph Follower algorithm calculates the target output hulls. Each optimisation leads to an evolution of the target function, shown as individual lines in (a) for $\alpha = 0.5$. Then, in the post processing, the upper and lower hulls are calculated from all available trajectories, green circles and red circles, respectively, resulting in the $\alpha = 0.5$ hull shown in (b). To approximate the fuzzy target output, this process has to be repeated for multiple α-levels.

2.6.2 Improvements introduced during this thesis

The following is quoted from my previous work [Sch22b] Section 3.3. Variable names in the text and figures have been changed to better fit this thesis. Figures and tables have been modified to suit the format of this thesis.
"

This section details the modifications made to the Graph Follower algorithm

in order to achieve the shown results. These modifications are necessary to keep it computationally feasible for the complexity of the analysed system.

Reduction of evaluation points

To accomplish this [the computational feasibility of the Graph Follower algorithm with the complex model], the timestep of the forward dynamics and the Graph Follower algorithm are separated. To compute stably, the forward dynamics of the leg with a geometrically exact prosthetic foot requires a timestep of 0.001 seconds. Using the same timestep for the Graph Follower algorithm for a simulation time of 2.0 seconds would result in 2000 α-level optimisations for each discrete α-level envelope, meaning once for the upper and once for the lower boundary, for every examined α-level. Additionally, for every new optimal parameter found by the Graph Follower algorithm, a new forward dynamics simulation is necessary to calculate the associated target output. In the worst case, this would lead to $N_\alpha * N_t * 2$ forward dynamics simulations, which is computationally expensive.
To reduce this [the amount of necessary calculations], the α-level optimisation is only performed every 100 steps of the forward dynamics simulation. This leads to greatly reduced computational cost. The resulting envelopes are still calculated with the forward dynamics timestep of $0.001s$, resulting in smooth trajectories. The number of required optimisations is greatly reduced at the cost of possibly not finding all extreme trajectories. Figure 2.6 shows the resulting envelopes for two different optimisation timesteps to compare the effects of this reduced optimisation timestep. Table 2.1 lists the maximum difference between the envelopes. Both simulations were performed using the parameters for the swing configuration, described in detail in the next section[3]. The difference between the two compared optimisation timesteps is negligible for this case.

Table 2.1: The maximum difference between envelopes calculated with reduced optimisation timestep and the same timestep as the forward dynamics simulation, for upper (H_U) and lower (H_L) envelope.

α-level	max ΔH_L	max ΔH_U
1.0	0.0	0.0
0.5	0.0110e-11	0.7254e-11
0.0	0.0916e-8	0.3369e-8

[3]This refers to Section 4.2

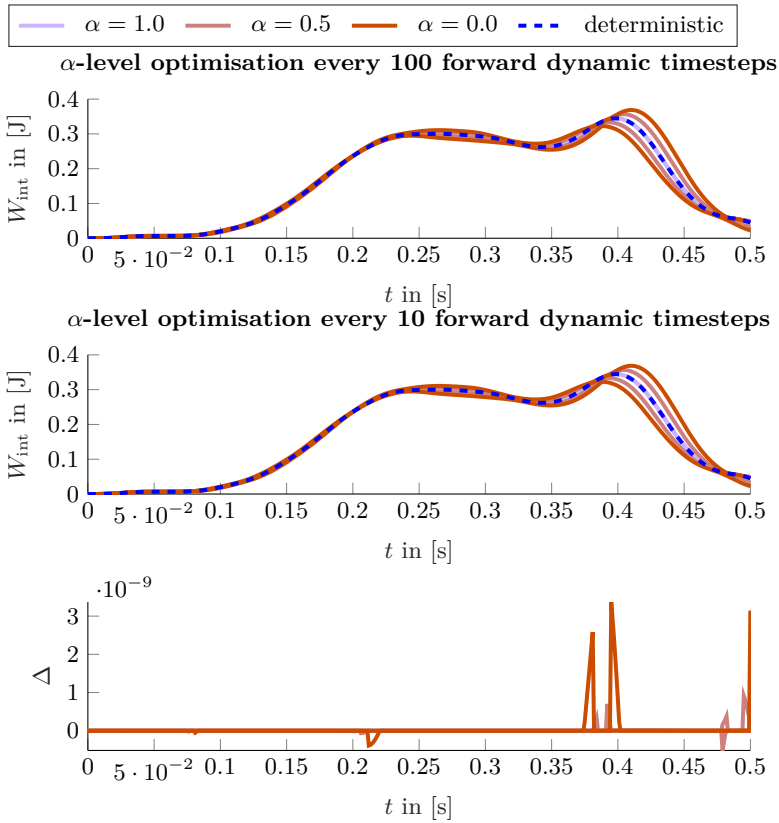


Figure 2.6: "The resulting envelopes for two different α-level optimisation step sizes in the Graph Follower algorithm. The top figure shows the results for an optimisation every 100 timesteps of the forward dynamics simulation, while the middle shows the results for an optimisation every 10 timesteps. The bottom shows the difference between the envelopes for the three examined α-levels." Figure and caption quoted from [Sch22b].

Tolerance distinguishing optimal parameters

The Graph Follower algorithm maximises or minimises the target output function within the parameter interval. If an optimal parameter is found, it is stored in a library and used for the approximation of the forward dynamics

in the optimisation. The last parameter associated with the extreme target output for the current examined [time node] is used to linearise the forward dynamics during the optimisation. This parameter library is also used to check whether the new optimal parameter is indeed new, by comparing it to the parameter used for linearisation. A new forward dynamics simulation is only performed if the difference of the optimised parameter to known parameters exceeds a percentual tolerance. This tolerance can be estimated by a sensitivity analysis of the forward dynamics with respect to the uncertain input parameter. To further reduce the computational cost, the forward dynamics simulation is recalculated and the new target output is calculated from the resulting trajectory only if the found optimal parameter is sufficiently different from all previously calculated parameters. Otherwise, the stored values are used. Ensuring the novelty of a parameter within an α-level greatly reduces the computational cost of the Graph Follower algorithm.

Of course, this tolerance directly affects the accuracy of the target output envelopes. This means it has to be dimensioned to fit to the examined problem. Also, multiple uncertain parameters make it more difficult to apply, since multiple parameters do not affect the target output without interdependencies. In this work, new forward dynamics calculations are performed if the relative difference $p_{\mathrm{diff,rel}}$ between the current fuzzy parameter and the closest stored parameter exceeds p_{tol} of the deterministic value. In the case of a simulation with multiple fuzzy parameters, a new simulation is performed if any individual parameter difference exceeds p_{tol} of the respective deterministic parameter.

$$p_{\mathrm{diff,rel},i} = \frac{|\tilde{p}_i - p_{\mathrm{Bib},i}|}{p_{\mathrm{det},i}} \tag{2.24}$$

The tolerance value $p_{\mathrm{tol}} = 0.01\%$ was determined experimentally. For more precision in the choice of this tolerance, a sensitivity analysis of the target output with respect to the fuzzy parameter can be performed.
”

The main steps of the Graph Follower algorithm with the improvements are summarised in the following list 'Graph Follower algorithm steps'.

Graph Follower algorithm steps

1. Quantify uncertain input parameter $\tilde{p}$ as a fuzzy number
2. Discretise $\tilde{p}$ with N_α α-levels
3. Calculate deterministic forward dynamics and target function to initialise library
4. For every timestep do:
   a) calculate upper hull with
   $f_{\max,t_j,\alpha_k} = \max(f(p, \mathbb{q}(p,t)))$ for $p \in P_{\alpha_k}$
   using linearised forward dynamics and best guess from library
   b) update library with calculated extremal trajectory
   c) calculate lower hull with
   $f_{\min,t_j,\alpha_k} = \min(f(p, \mathbb{q}(p,t)))$ for $p \in P_{\alpha_k}$
   using linearised forward dynamics and best guess from library
   d) update library with calculated extremal trajectory
5. For every timestep do:
   a) improve upper hull by using the maximum of all individual target function calculations from the library at the current time node
   b) improve lower hull by using the minimum of all individual target function calculations from the library at the current time node

2.6.3 Limitations of the Graph Follower algorithm

The following is quoted from my previous work [Sch22b] Section 3.4. Variable names in the text and figures have been changed to better fit this thesis. Figures and tables have been modified to suit the format of this thesis.
"

Before examining the scenarios simulated in this work, the limitations of the Graph Follower algorithm are briefly summarised. The goal of propagating fuzzy input parameters through a model, is to calculate the target output fuzzy number. This includes the interval of the target output f and the associated

membership values $\mu_{\tilde{f}}$. As detailed above, the Graph Follower algorithm uses α-level optimisation and a linearisation of the forward dynamics with respect to the fuzzy input parameter to efficiently calculate the target output envelopes for multiple α-levels, where the membership function is derived from the current α-level, resulting in an approximation of the target output fuzzy number. The α-level optimisation with forward dynamics imposes limitations on the accuracy of the resulting envelopes. To work precisely, the α-level optimisation requires a global optimiser and precise calculation of the target output, see [Möl04]. However, global optimisation is computationally expensive and the accurate calculations of the forward dynamics required for the exact target output calculation are also computationally expensive with such a complex model. Thus, to be feasible, the Graph Follower algorithm only approximates global optimisation, by using Matlab's *MultiStart* function with ten starting points. This significantly increases the result quality with an acceptable increase in computation time, compared to using *fmincon* with just one starting point. The linearisation of the forward dynamics with respect to the fuzzy parameter introduces more inaccuracy. Both of these limitations can lead to an overlap of α-levels. The modifications necessary for the Graph Follower algorithm to work with the complex model of the human leg with a deformable prosthesis, also require a trade-off between accuracy and computational cost. The separation of the timestep of the forward dynamics simulation and the α-level optimisation greatly reduces the amount of necessary optimisations but also reduces the chances to find extremising parameters. Also, the tolerance when comparing new parameters to stored parameters can reduce accuracy if it is chosen to large, however, it greatly reduces the computational cost of the modified Graph Follower algorithm. Nonetheless, the modified Graph Follower algorithm, as is used here, produces an approximation of the target output fuzzy number for the complex multibody system of the human leg with a predeformed geometrically exact beam prosthetic foot. Furthermore, these limitations can be reduced or even become negligible by increasing the available computing power. Also, if the precise membership function value of the target output is not required, the trajectories of the target output of the individual α-levels can be combined to get an accurate global envelope of the target output, providing information on the extreme values of the target output based on the fuzzy input parameter. Thus, even with limitations, the modified Graph Follower algorithm provides valuable information about how epistemic model parameters affect a desired target output for a highly complex multibody system with flexible and rigid bodies, allowing for a quicker examination of the uncertainties effects than simulating many different sets of parameters.
”

2.7 Polymorphic uncertainty

This section is the main contribution of this thesis to the methodology of dealing with uncertainty. To expand fuzzy numbers to polymorphic uncertainty that can also model variability, fuzzy random variables (FRVs) as described in [Möl07] are used here, which in turn are based on [Kwa78; Kwa79; Möl04]. It should be noted that in the literature, the distinction between fuzzy random variables and random fuzzy variables is not always clearly defined.

Fuzzy random variables are introduced in [Kwa78] as random variables whose values are not real numbers but fuzzy numbers. They are often used to handle linguistic label information in statistics or to represent uncertainty about classical random variables. Thus, a fuzzy random variable can be regarded as a fuzzy set of traditional random quantities, each one carrying a certain membership degree [Gud98]. According to [Cou09] there are multiple definitions of fuzzy random variables, see for instance [Kwa78; Pur93; Kru87], and a unified approach is suggested in [Krä01]. In [Möl04], a fuzzy random variable $\tilde{\tilde{P}}$ is defined as convex fuzzy number realisation of a random event. In other words, a realisation or one sample of a fuzzy random variable $\tilde{\tilde{P}}$ is a convex fuzzy number $\tilde{p}$. This means that every random event ω in the space of all random events Ω is assigned a fuzzy realisation $\tilde{p}$ from the set of all convex fuzzy variables $\mathbb{F}$. This is the formulation that is used in this work.

$$\tilde{\tilde{P}}(\omega) : \Omega \to \mathbb{F} \quad | \quad \omega \in \Omega \tag{2.25}$$

Random fuzzy variables (RFV) are type 2 fuzzy variables. They are defined with mathematical possibility theory and are used to represent the entire information associated with a measurement result and an internal possibility distribution and an external possibility distribution called membership functions. The internal distribution is the uncertainty contribution due to the systematic uncertainty while the bounds of the RFV are due to the random contributions.

The difference between FRVs and RFVs lies in their definitions and applications. Fuzzy random variables are defined as the fuzzy perception of an unobservable real-valued random variable [Ber21]. They are actually a description of the fuzziness contained in a traditional random variable often called the original [Ber21]. On the other hand, random fuzzy variables are used in the context of modelling a given random experiment which produces fuzzy sets of a metric space [Ber21]. This random fuzzy set is defined as the "levelwise extension" of the notion of random sets in real-based spaces or as

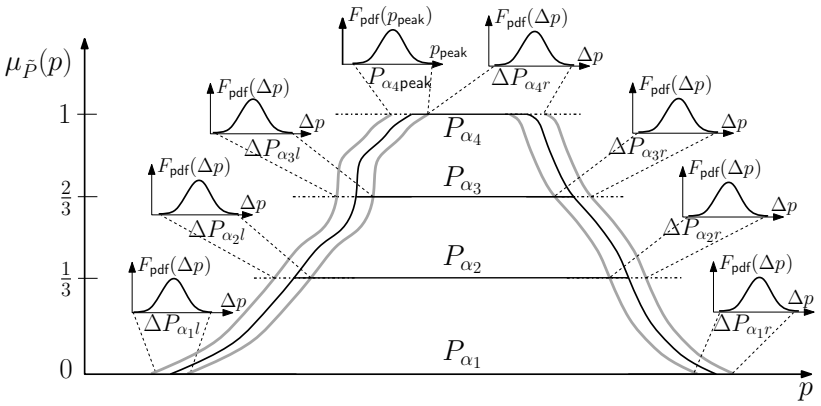


Figure 2.7: "A fuzzy random variable, with randomised $l_\alpha r_\alpha$-increments, based on [Möl07]." Figure and caption quoted from [Sch24a].

random elements defined in the spaces of fuzzy sets [Ber21].

The main reason to use FRVs is the improved Graph Follower algorithm. Due to its capability of efficiently propagating fuzzy numbers through the complex multibody model simulation, it makes sense to expand the uncertainty in a direction that will incorporate this algorithm. In polymorphic uncertainty simulations, it is common to reduce the uncertainty step-wise. FRVs, as described in [Möl04], are a polymorphic uncertainty model, that, when samples are generated and the randomness is thereby collapsed, result in fuzzy numbers which can then be further examined with the Graph Follower algorithm.

2.7.1 Fuzzy random variables (FRV)

In this work, FRVs are used to quantify polymorphic uncertainty. Based on the $l_\alpha r_\alpha$-discretisation introduced in Section 2.5.2, a method to combine the variability of random numbers and the lack of knowledge from fuzzy numbers is described, which can then be propagated with the newly developed FRV-Graph Follower algorithm. The following is quoted from my previous work [Sch24a] Section 2.2. Variable names in the text and figures have been changed to better fit this thesis. Figures and tables have been modified to suit the format of this thesis.
"

A fuzzy random variable is visualised in Figure 2.7. To include randomness in the uncertainty model, the $l_\alpha r_\alpha$-increments from Equation 2.20 in

Section 2.5.2 are randomised, see [Möl07]. This is visualised in Figure 2.7. The $l_\alpha r_\alpha$-increments are governed by a probability density function $F_{\text{pdf}}(\Delta p)$, while the convexity condition from Equation 2.21 holds. To generate a realisation of the fuzzy random variable, a random peak point p_{peak} is drawn, based on the governing probability function $F_{\text{pdf}}(p_{peak})$. Then, the increments for the next lower α-level are drawn from the probability function $F_{\text{pdf}}(\Delta p)$ governing them, while ensuring the convexity condition. This condition introduces a correlation between the increments, but is unavoidable to ensure convexity of the resulting fuzzy number, see [Möl07]. This method results in random intervals $[P_{\alpha_i l}, P_{\alpha_i r}]$ for every α-level and the random membership function $\mu_{\tilde{P}}(p)$. Then, a random fuzzy number $\tilde{\tilde{P}}$ is described by the random sets P_{α_i}

$$\begin{aligned} \tilde{\tilde{P}} = (P_{\alpha_i} &= [P_{\alpha_{i+1} l} - \Delta P_{\alpha_i l}, P_{\alpha_{i+1} r} + \Delta P_{\alpha_i r}] && \mid \alpha_i \in [0,1); \\ P_{\alpha_i} &= [P_{\alpha_{\text{peak}}}, P_{\alpha_{\text{peak}}} + \Delta P_{\alpha_i r}] && \mid \alpha_i = 1). \end{aligned} \tag{2.26}$$

”

2.7.2 Fuzzy random variable Graph Follower Algorithm (FRV-GFA)

The following is quoted from my previous work [Sch24a] Section 4.3 & 4.4. Variable names in the text and figures have been changed to better fit this thesis. Figures and tables have been modified to suit the format of this thesis. “

To enable the propagation of polymorphic uncertainty in the form of fuzzy random variables (FRVs), the realisations of which are fuzzy numbers as described above, a new algorithm is required to propagate this uncertainty through our dynamic model. The steps of our algorithm are shown in Figure 2.8. Similar to most algorithms dealing with polymorphic uncertainty, a nested algorithm is used [Möl04]. This means that the uncertainty is reduced from one step to the next, until a deterministic evaluation of the model with deterministic values is possible. Then the uncertainty is "reassembled". The main steps are the sample generation, where N_s samples are generated from the FRV, examination of the fuzzy number samples with the Graph Follower algorithm, and finally calculating and fitting probability density functions to the target output hull samples. The steps are described in more detail in the following paragraphs.

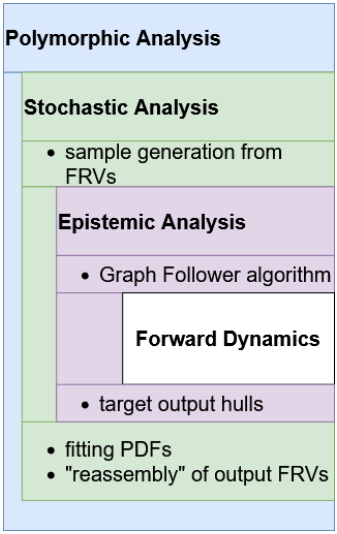


Figure 2.8: “The nested steps of the FRV-GFA developed within the project 14.” Figure and caption quoted from [Sch24a].

Sample generation In general, with aleatoric uncertainty, a large number of samples is generated, then the target quantity is calculated individually for each sample and finally distributions are fitted to the resulting data. Similarly, the proposed algorithm generates N_s samples for the uncertain parameter, using the previously described $l_\alpha r_\alpha$-discretisation from [Möl07]. Currently, all probability distributions in this work are assumed to be normal distributions[4]

$$F_{\mathrm{pdf}}(p) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{1}\left(\frac{p-\mu}{\sigma}\right)^2} \tag{2.27}$$

[4]This refers to the results presented in [Sch24a]. This work expands the FRV-GFA to include other distributions, shown in Section 5.7.1.

defined by the mean μ (not to be confused with the membership function $\mu_{\tilde{p}}(p)$) and the standard deviation σ. Each sample is generated by calculating a random peak point p_{peak}, see Figure 2.3, and adding or subtracting a random Δp to or from the deterministic parameter value p_{det}.

$$p_{\text{peak},i} = p_{\text{det}} + \underbrace{F_{\text{pdf}}(p) \cdot 0.1 \cdot p_{\text{det}}}_{\Delta p_{\alpha=1}} \tag{2.28}$$

The random selection of Δx for the sample i is based on the probability density function $F_{\text{pdf}}(p)$ in Eq. (2.27) with a mean of $\mu = 1$ and a standard deviation of $\sigma = 1$. The PDF's value is drawn and then multiplied by $0.1 \cdot p_{\text{det}}$, relating the scale of Δp to the parameter's magnitude. In the future, the calculation of p_{peak} and the governing probability density function $F_{\text{pdf}}(p)$ can be based on measurements. While lower α-levels use the same distribution as $F_{\text{pdf}}(p)$ (with the same mean and standard deviation), the deviation Δp is slightly different with

$$\Delta p_{\alpha<1} = F_{\text{pdf}}(p) \frac{0.1 p_{\text{det}}}{N_\alpha}. \tag{2.29}$$

The division by the number of α-levels allows for independent control over the expected width of the generated samples of the FRV at α-level zero and the number of α-levels that are examined. Starting from the peak point x_{peak} the steps are

$$p_{\text{l},k-1} = p_{\text{l},k} - \Delta p_{\text{l},k} \qquad \text{and} \qquad p_{\text{r},k-1} = p_{\text{r},k} + \Delta p_{\text{r},k}. \tag{2.30}$$

To ensure convexity, as required for a fuzzy number, the increments have to be greater than zero, $\Delta p_{\text{l},k} > 0$ and $\Delta p_{\text{r},k} > 0$, see Eq. (2.20). In case a randomly drawn increment does not fulfill this requirement, it is regenerated by redrawing the $F_{\text{pdf}}(p)$ value. While this introduces a correlation between the increments, it ensures the convexity of the fuzzy number samples. It should be noted, that this generation truncates the normal distribution currently chosen. Other distributions are of course an option, however, the normal distribution was chosen for this work due to its simplicity and easy validation.

Epistemic analysis Since each individual sample is a fuzzy number, the epistemic analysis can be performed with the previously described Graph Follower algorithm. This returns a fuzzy number for the target output quantity for every examined [time node]. Nested within the Graph Follower algorithm, the forward dynamics of the model are computed, which is the computationally most expensive step.

Reassembling target output FRVs Once the epistemic analysis of every sample is completed, the increments can be calculated for the samples and for

every [time node]. Then a probability density function can be fit to the available data, using Matlab 2023b's *fitdist* function, resulting in a FRV of the target output function for every [time node]. It should be noted, that for consistency, a normal distribution is fit to the data[5] This can be modified in further examinations.

2.7.3 Reduction of computational load

[...]

Since stochastic analyses require a large number of samples, the computational cost [of the improved Graph Follower algorithm] needs to be reduced further. To make the load manageable, a grouping of samples is performed after the sample generation before the epistemic analysis starts. All samples are mapped onto a number of discrete values in the examined parameter interval. This greatly reduces the variability in the parameter and therefore the number of parameters for which calculations are necessary. Currently, the parameter interval (from smallest to largest sample) is mapped onto 120 discrete points, resulting in a difference of approximately $0.1\% \cdot p_{det}$ between these points[6]. Then, the forward dynamics can be calculated for the predefined sample points. This has two advantages. Firstly, it is ensured that all samples lead to a viable forward dynamics simulation. Since a normal probability distribution is in theory infinite, too extreme values for the simulation's parameters might prevent the implicit Newton–Raphson solver in the variational integrator from converging to a solution. In case a sample is so extreme, that the Newton–Raphson solver does not converge, a new sample would have to be generated. So far, this has not occurred. Secondly, no further forward dynamics calculations are necessary after the grouping. During the epistemic analysis, if an optimal parameter is found, it can also be mapped onto the predefined points, similar to the novelty check in the Graph Follower algorithm, eliminating the necessity of recomputing the forward dynamics and reducing the computational cost. Of course this comes at the price of a certain decrease in accuracy. However, we found the reduction in the computational cost greatly outweighs the acceptable reduction in accuracy. Furthermore, by increasing the previously defined discrete values, direct control of the algorithm's accuracy is possible. Due to the precomputation of the forward dynamics and grouping of samples, it would also be possible to eliminate some of the required optimisations, if the sample fuzzy number bounds are already known from previous optimisations. However, this requires

[5] This refers to the results as presented in [Sch24a]. This thesis expands the FRV-GFA to include other distributions, shown in Section 5.7.1.

[6] This refers to the results as presented in [Sch24a]. This work uses different parameters and simulation scenarios, shown in Section 5.7.1

advanced use of data storage and further investigation.
”

It should be noted, that currently the sample generation and fitting of the target output's $l_\alpha r_\alpha$-increments is performed with Matlab 2023b's *fitdist* function while assuming a normal distribution of the increments. This is of course an assumption and not necessarily the case. As mentioned in Section 2.3, there are many other distributions. For instance, the algorithm was also tested using the beta distribution for $\alpha \leq 1$ and the Normal distribution only for the peak points, see Section 5.7.1. While changing the distribution from an infinite one to one that is limited to $[0, 1]$ is not a problem for the sample generation and avoids the truncation necessary with the Normal distribution, when used during fitting in the FRV-reassembly the question arises, how to normalise the results to that interval. For one, this has to be performed for every time node. Then, each side has to be normalised (the left and right increments). And then the question is, whether it should be normalised relative to the maximum (or minimum) of each α-level or to the maximum of that time node. So while other distributions are possible and can be implemented, there are open questions regarding their use. However, this can be changed in the future by incorporating other distributions or more advanced fitting methods, see for instance [Gho19a] and does not impede the algorithm's idea or performance.

2.7.4 Improvements to FRV-GFA

As mentioned, the main issue with uncertainty simulations for aleatoric uncertainty, is the large sample size required to achieve accurate results. Pairing this requirement with the computationally expensive Graph Follower algorithm leads to unfeasibly expensive computations. Thus, more efficiency increases are necessary.

Grouping of samples Similarly to the parameter novelty check introduced to the Graph Follower algorithm in [Sch22b], which limits the recalculations of the forward dynamics during the Graph Follower algorithm by in essence discretising the possible inputs based on past results, the samples required for the aleatoric part of the FRV-GFA can be grouped by mapping them to specific points on a discretised grid of the parameter space, visualised in Figure 2.9. It should be noted, that the convexity of the fuzzy number samples is ensured during the sample generation process, described in Subsection 2.7.2, by calculating all $l_\alpha r_\alpha$-increments and ensuring they fulfil Eq. (2.21). This process can be put within the category of binning. In essence, binning describes the process of grouping data points into discrete bins. In effect, the interval between the smallest and largest value of an uncertain parameter is

discretised into $N_{p\text{-disc}}$ points. Now all samples can be remapped to the middle of the interval defined by two neighbouring points. To determine how many points should be used to discretise parameters, a sensitivity analysis can be used [Sal08], though the available computing power is also a determining factor. The process of grouping samples bears similarities to the Latin Hypercube sampling method, described in a paragraph in Section 2.4. The main difference to Latin Hypercube sampling is, that here the variable is discretised by equidistant points, while Latin Hypercube sampling defines the intervals to have equal probability. Since the goal of this work is to demonstrate the capabilities of the FRV-GFA, and since Latin Hypercube sampling may not be applicable to all uncertainty types, the more general approach of discretising the input domain was chosen. Using this, the forward dynamics

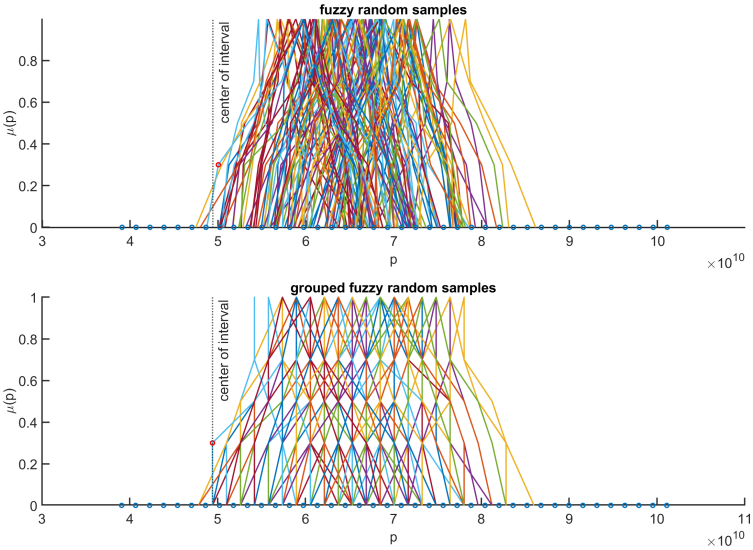


Figure 2.9: The top image shows a selection of the generated samples of the fuzzy random variable p. Each sample is a fuzzy number. The horizontal axis shows the 30 grid points used to discretise the parameter p for the grouping performed in the bottom image. The vertical line highlights the center of an interval and how a sample point not on the grid (top red circle) is remapped to the grid points in the grouping process (bottom red circle).

calculations can be performed on the number of points used for the parameter discretisation $N_{p\text{-disc}}$ in advance and stored in a library $\mathbf{Q}_{\text{lib}}$ and the target function can also be calculated for each parameter value and stored in a li-

brary f_{lib}. Then, the α-level optimisation is performed for each interval from the grouping using the Graph Follower algorithm. However, when an optimal parameter is found, instead of recalculating the forward dynamics and target output, it is mapped to the grid and the results can be looked up in the library or, depending on the tolerance check within the Graph Follower algorithm, a new forward dynamics simulation is calculated and added to the library. This greatly reduces the calculation effort, while still allowing for fine tuning of the result's accuracy by changing the number of points, used to discretise the input parameter. However, this still requires an optimisation for each sample and each α-level of the sample.

Separation of processes and smart storage As can be seen in Figure 2.9, due to the grouping, some of the samples overlap. As a reminder, note that each fuzzy number is described by a set of intervals and the respective α-level value of that interval. To propagate the uncertainty for a single fuzzy number, the α-level optimisation is performed for each α-level on the associated interval. However, the α-level value is only used after the optimisation to approximate the target output fuzzy number. Due to the stochastic uncertainty in the FRV-GFA and the subsequent overlap of fuzzy number samples, multiple optimisations on the same interval are performed, if the samples are treated individually. This is inefficient, for obvious reasons. To avoid this, the different steps in the FRV-GFA can be separated.
The idea is to separate the samples, with their respective parameter intervals, and the optimisations. This avoids unnecessarily repeating optimisations required for the propagation of the epistemic uncertainty of input parameter intervals that show up in multiple samples after the α-level discretisation and grouping. Then, after the optimisation is performed on each interval and the resulting target output envelopes for that interval are known, they can be assigned to each sample and the target output FRV can be reassembled. The number of possible intervals is a combinatorial problem of choosing a distinct objects out of B possible objects and is given by the binomial coefficient

$$\binom{B}{a} = \frac{B!}{a!(B-a)!}. \tag{2.31}$$

In our case, $a = 2$ for the bounds of the interval and $B = N_{p\text{-disc}}$, thus Eq. (2.31) can be calculated with the sum of first integers as well

$$\binom{N_{p\text{-disc}}}{2} = \frac{N_{p\text{-disc}}!}{2!(N_{p\text{-disc}}-2)!} = \sum_{k=1}^{N_{p\text{-disc}}} k = \frac{N_{p\text{-disc}}(N_{p\text{-disc}}+1)}{2}. \tag{2.32}$$

In the example shown in Figure 2.9 with $N_{p\text{-disc}} = 30$ (shown as blue dots in the p-axis) a total of 435 intervals are possible. This means, instead of

performing an α-level optimisation for every α-level of every sample, $N_s * N_\alpha$ α-level optimisations in total, a maximum of $\frac{N_{p\text{-disc}}(N_{p\text{-disc}}+1)}{2}$ optimisations are required. This greatly reduces the total computational cost, while allowing a large number of samples for accuracy in the stochastic uncertainty propagation. As shown in Section 2.4, the number of required samples to accurately represent the underlying aleatoric uncertainty can get quite large. Since for $\alpha = 1$ the interval bounds coincide, no optimisation is necessary and the forward dynamics and target output can be calculated and stored for later use directly. Note also, that not all possible intervals are present in the generated samples. To improve efficiency, an intermediate step ensures that only intervals that are required for at least one sample are considered in the optimisation, further reducing computational cost.
Now that all required intervals are known, the optimisation is performed resulting in the upper and lower envelope for the target output for every time node and for each interval. In a final step, the envelopes are associated with the samples and their α-levels and the resulting FRV of the target output is reassembled. In summary, this approach of grouping the samples greatly reduces the number of required optimisations and unlinks this number from the sample size. This allows for efficient computation of many samples, which is critical for the stochastic evaluation. The worst case for the number of optimisations required Ω for one uncertain parameter depends on number of intervals as calculated in Eq. (2.32) and on the number of time nodes N_t multiplied by two in order to account for the upper and lower hull optimisations

$$\Omega = 2N_t \left(\frac{N_{p\text{-disc}}(N_{p\text{-disc}} + 1)}{2} \right). \tag{2.33}$$

In the case of multiple uncertain parameters, the worst case for optimisations is Eq.(2.33) applied to every parameter, with the respective number of discretisation points, and multiplied with each other. It is obvious, that this can lead to very large numbers very quickly as shown in Figure 2.10, which again highlights the main issue with uncertainty simulations, namely the associated high computational cost. From Eq.(2.33) it is evident, that the main factors governing the computational cost of the proposed algorithm are the number of intervals and the number of points used for the discretisation of the parameter. This is examined in Figure 2.10 by varying the number of generated and grouped samples N_s, which affects the number of intervals that have to be considered, and the number of points used for the discretisation of the uncertain parameter space $N_{p\text{-disc}}$. As expected and shown on the left, the number of possible intervals is independent from the number of samples and drastically increases with the number of points used for the discretisation. Also, as shown in the middle, the number of occurring intervals, that is the

interval is present in at least one sample, is dependant on both the number of samples and the number of points used for the discretisation. If not a lot of samples are generated, not many intervals will occur. On the other hand, the number of relevant intervals rises quickly with increasing sample size but flattens after a sufficiently large sample size is present. This can be used as an indication for the minimum required sample size, given the number of parameters and discretisation points. On the right side of Figure 2.10 it is noteworthy, that the ratio of the occurring intervals to the worst case of all possible intervals drastically decreases with increased number of samples and number of discretisation points. This gets even more drastic with multiple parameters. However, due to the nature of sample creation, not all combina-

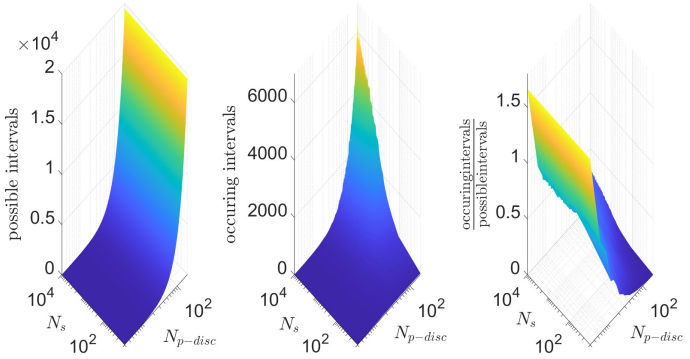


Figure 2.10: Visualisation of how the number of samples and number of points used for the discretisation result in the number of intervals that have to be optimised. Yellow indicates a high number of intervals. From left to right, all possible intervals are shown, then the intervals that are present in the samples and finally the ratio of present intervals to all possible intervals.

tions of intervals will show up in the samples, greatly reducing the number of required optimisations. Furthermore, it should be highlighted that Eq.(2.33), which calculates all possible intervals that may need to be optimised, is independent of the number of generated samples. This means the deciding factor on how many optimisations need to be performed in the worst case no longer depends on the number of samples. In practice, the number of intervals that are present in the samples is much smaller than the number of possible intervals. This is accounted for in the algorithm only generating an entry in the dictionaries, which are used for data storage, if at least one sample has the

respective interval.

The data storage is implemented using Matlab 2023b's dictionary functions. Dictionaries map a keys to any type of data. The main advantage is, that different types of data can be synchronised to the same key across multiple dictionaries. In its current implementation, there are dictionaries for all intervals that are present in at least one sample, the upper and lower hulls, the parameters that lead to extremal trajectories and the extremal trajectories, in short, the same output as is produced by the Graph Follower algorithm. As keys, the left and right interval bounds are used. Since the keys are crucial in keeping the data synchronised, a separate function handles the conversion of interval bounds to keys. The reverse operation can be performed by a dictionary storing the intervals with the respective key. This works for a single uncertain parameter as well as for multiple parameters. Another huge advantage of this method is, that if not enough samples are used in the initial simulation, more can be added with ease. As long as the dictionaries are saved and the grouping of samples does not change, any number of samples can be added to the existing ones. Then the grouping needs to be performed again to map them onto the same intervals from the previous grouping and the reassembly has to be performed. If new intervals are present, these have to be optimised first. These two steps take a fraction of the computational effort of the forward dynamics calculations and optimisation, even for a very large number of samples. (This functionality can be added to the algorithm but is not yet implemented.) It should be noted, that with so many samples, the data storage aspect becomes an issue. Since all values are stored with double precision, the standard in Matlab 2023b, and each double precision data requires 8 bytes, the memory D_{memory} required for such an array with sizes $a_1, a_2, ..., a_I$ in Gigabyte (GB) is

$$D_{\text{memory}} = \frac{\prod_i a_i}{(1024)^3}. \tag{2.34}$$

For example, for 100000 samples ($a_1 = 100000$), 2000 time nodes ($a_2 = 2000$), 5 α-levels ($a_3 = 5$), which is enough to store only the hulls ($a_4 = 2$) of the simulation, a total of 1.86 GB in memory is required. This does not yet account for the storage of the dictionaries for the forward dynamics, the target function, various other simulation parameters or intermediate data during the optimisation. As mentioned in Section 2.4, stochastic accuracy requires a lot of samples, so the required memory can grow very quickly beyond the RAM capacities of normal desktop computers or laptops. The dictionary approach introduced here for the FRV-Graph Follower algorithm can be modified to store data on the hard drive instead of RAM if necessary.

Interaction between GFA novelty check and FRV-GFA sample grouping One question that needs to be answered, is how the novelty check introduced in [Sch22b] interacts with the grouping of samples in the FRV-GFA, developed here. It may seem, that if all the forward dynamics trajectories are precalculated for the existing interval centres, there is no need for the optimisation within the Graph Follower algorithm. A simple comparison of all target output evolutions from the already known forward dynamics trajectories would suffice. However, the goal of the precalculation of the forward dynamics is to skip unnecessary recalculations during the optimisations while at the same time providing better linearisation points for the Taylor approximation used in the optimisation. The novelty check then ensures, that if a new optimal parameter is found and the difference to all currently known parameters, stored with their forward dynamics trajectories and target outputs, is too large, a new forward dynamics trajectory is calculated based on the new optimal parameter. This is then added to the dictionary, improving the dictionary and reducing future forward dynamics calculations. Of course, the grouping discretisation and the tolerance in the novelty check interact. If the grouping is finer than the novelty check tolerance, no new forward dynamics calculations will ever be necessary, while if the grouping is much coarser than the tolerance, most optimisations will incur a forward dynamics calculation. This allows for a fine tuning of computational effort and independent control over the two options, meaning they can be modified to be ideal for a given problem. In the current implementation, the tolerance is chosen slightly smaller than the grouping discretisation in order to demonstrate the interaction, such that sometimes a new calculation of the forward dynamics is triggered and sometimes the dictionary entries are used.

Improving hulls based on previous knowledge One issue pointed out in [Sch22b] for the Graph Follower algorithm, is that due to the simplifications necessary to consider uncertainty for such a complex non-linear multibody model with rigid and flexible bodies and the lack of efficient global optimisation algorithms, some errors are visible in the hulls, see Figure 2.11(a). These show up as overlapping hulls, which obviously is not correct. When an optimisation is performed on an interval, the resulting value cannot be less optimal than the value obtained from performing the optimisation on any of its subintervals. Since the intervals for the α-level optimisation are derived from the α-level cuts and Eq. (2.19) holds ensuring the convexity of the fuzzy number, the results for the target output hulls from lower α-levels have to be at least as large or small as the result from the higher α-level, depending on whether the upper or lower hull is currently being calculated respectively. In addition, the description for fuzzy random variables used here, see Section 2.7.1, requires the increments to be larger than or equal to 0, to fulfil the convexity condi-

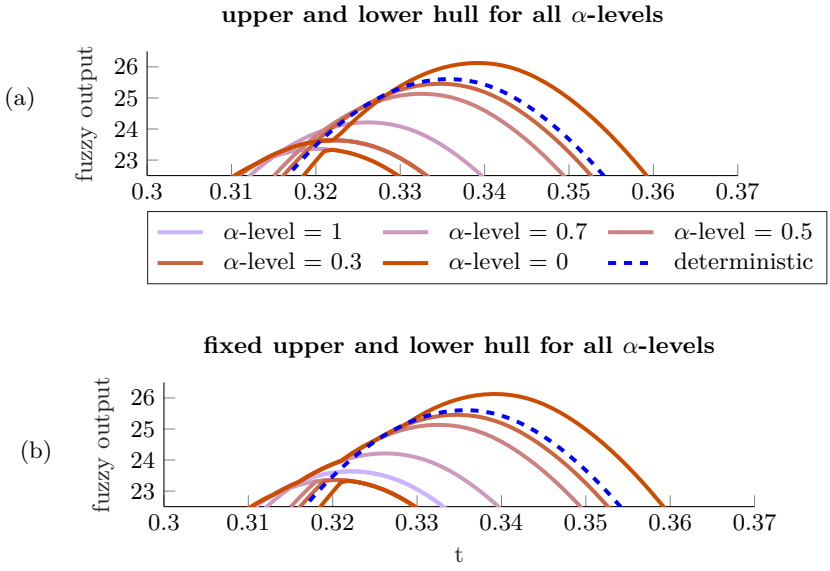


Figure 2.11: Visualisation of how the target output hulls can be improved for one sample. In (a) the hulls overlap in the shown section of the target output evolution at $t = 0.32s$ due to various simplifications required for the Graph Follower algorithm to work with such a complex model. By postprocessing the hulls and checking for intersections, they can be fixed as is shown in (b).

tion of fuzzy numbers in Eq. (2.21). If the hulls of two α-levels intersect, an increment would be negative and the convexity condition would be violated. Thus, to prevent this in the reassembly of the FRV, all of the target output samples are passed through a filter function, which goes from the highest α-level ($\alpha = 1$) to the lowest ($\alpha = 0$) and through all time nodes and checks whether the increments are negative. If a negative increment is detected, its position (time node and sample) is stored, and that specific hull value is set to the same value as the hull of the previous α-level, based on the assumption, that the optimal trajectory of the lower α-level is at least as small or large, depending on whether the lower or upper hull is examined respectively, as that of the previous, higher α-level, resulting in Figure 2.11(b). While this is not a perfect solution to the problem of lacking a computationally efficient global optimisation algorithm, it is a very efficient way to improve the results from the available data and simultaneously ensures that the convexity condition is

satisfied for the target output FRV.

The main steps of the FRV-GFA are summarised in the list 'FRV-GFA steps'.

FRV-GFA steps

1. Generate N_s samples discretised with N_α α-levels for the fuzzy random parameter $\tilde{\hat{p}}$
2. Group the samples into $\binom{N_{p\text{-disc}}}{2}$ intervals
3. Create a unique key for every interval in the samples
4. Create dictionaries based on the keys
5. Use the Graph Follower algorithm to find hulls for the intervals for all keys
6. Improve hulls with postprocessing to ensure convexity of the hulls
7. Assign the hulls to the respective sample using the keys and dictionaries
8. Calculate the deltas of the samples
9. Fit PDF to the deltas or the target function

2.7.5 Further possible improvements

Finally, there are some improvements that are currently not possible due to the limitations of the software used in the current implementation.

Parallelisation Once the samples are generated and grouped, the intervals on which the α-level optimisation has to be performed are known. Since there is no reason to perform the optimisation on the intervals sequentially, an obvious improvement would be parallelising the optimisations. However, part of the model implementation uses CasADi [And19] in the differentiations required for the equations of motion and the linearisation step of the the Graph Follower algorithm. Currently, when paired with Matlab 2023b, this prevents parallelisation, both for the optimisation itself and the process of optimising on all intervals. However, the use of dictionaries would be ideally suited for parallelisation and this should be a focus of future improvements.

Sampling methods As is known from other stochastic applications, the main issue is the number of samples that need to be considered. In the current implementation, the uncertainty is assumed and not based on real world measurements. Depending on the application of the algorithm, the sampling can be improved by focussing it on certain areas in the sample space or using more advanced sampling methods as introduced in Section 2.4. Especially Latin Hypercube sampling may provide a large decrease in computational effort and is already similar to the current implementation.

Inclusion of contained intervals in optimisation Another option to improve results, is to sort the intervals, and start from the smallest with the optimisation. Then any optimisations performed on a larger interval can use the results of the previous one for initial guesses, trajectories etc. This would require some modification of the GFA and some way to sort or cross-reference intervals. The sorting of intervals may be problematic with multiple parameters. While dictionaries should work for this as storage, cross-referencing the intervals likely requires more advanced storage methods.

Summary

In summary, the Graph Follower algorithm was substantially improved by increasing its computational efficiency. This allows its use for propagation of epistemic uncertainty in the form of fuzzy numbers for more complex systems than before. Furthermore, a new algorithm for propagating polymorphic uncertainty in the form of FRVs was proposed and its computational efficiency was increased further.

Chapter 3

Human gait

This thesis examines the propagation of polymorphic uncertainty through a simulation of two distinct gait phases of a human leg with a prosthetic foot. To be able to model the human leg accurately, some knowledge about its anatomy and the process of walking is necessary. Sections 3.1 and 3.2 introduce the human leg anatomy and the human gait cycle. Section 3.4 summarises how this is represented in the model. Section 3.5 outlines the process of measuring human gait as is necessary to understand the consideration of epistemic uncertainty in this process as proposed the journal publication [Sch24b]. The development of methods to include uncertainty and examine its effects is described in Section 3.7. Epistemic uncertainty in the form of triangular fuzzy numbers in the marker positions and its effect on subsequent joint angle calculations has not yet been considered. This chapter is partly based on the student theses [Alj20; Hei21; Sch19] and the journal paper [Sch24b].

3.1 Anatomy of the human leg

This section summarises the anatomy of the human leg. For a more detailed description of human anatomy and movement see [Lip06; Pal11]. This work examines the effects of uncertainty in a foot prosthesis simulation, thus before detailing the models involved, the function of the ankle-foot complex is briefly described. Later on, the effect of marker position errors on subsequent calculations is described, requiring a marker model. This model relates anatomical positions to the markers so, for reference, Figures 3.1, 3.2 and 3.3 from [Big23] are included and visualise the bones with their anatomical names. For the muscles, Figures 3.4, 3.5 and 3.6 from [Big23] are included.

The human leg is subdivided into three segments, the thigh, the shank and the foot, which are connected by joints to allow for movement. The hip joint is a ball-and-socket joint, allowing for the movements flexion/extension, abduction/adduction and internal/external rotation. It is considered the only true ball-and-socket joint in the human body [Whi14]. It is formed by the pelvis (socket) and the head of the femur (ball), see Figure 3.1. The femur is the longest bone in the human body and forms the thigh, with the proximal hip

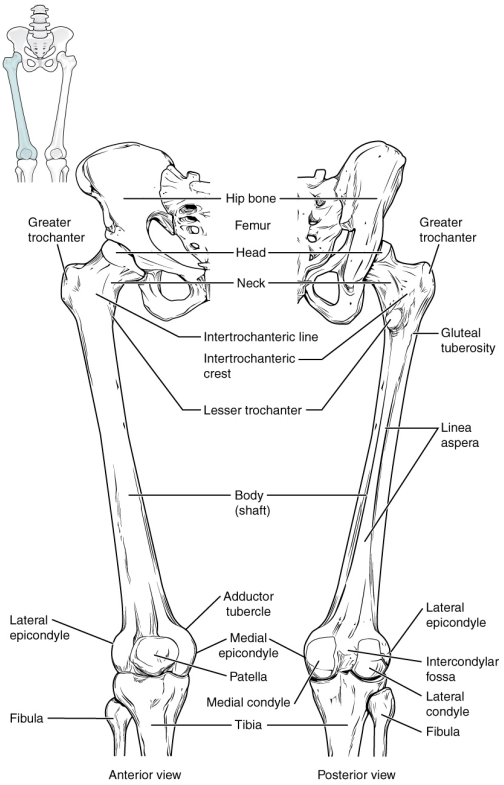


Figure 3.1: The femur and patella bones of the human leg. Copied from [Big23] without adaptation which is published under the CC-BY-SA 4.0 license [CCBYSA]. Therefore this figure is exempt from the copyright covering this thesis.

joint and distal knee joint. The knee joint is formed by the femur and tibia, while the patella is part of the patellofemoral joint [Whi14]. During normal gait, the knee joint's main movements are flexion and extension, though the ligaments involved form a four-bar-linkage and a slight external rotation at full extension can be observed, known as the screw-home mechanism [Whi14]. The shank itself consists of two bones, shown in Figure 3.2, the load bearing tibia and the fibula, which helps stabilise the ankle joint. This joint is formed between the tibia in the shank and the talus in the foot and has only one main motion, namely dorsiflexion/plantarflexion. These correspond to flexion and

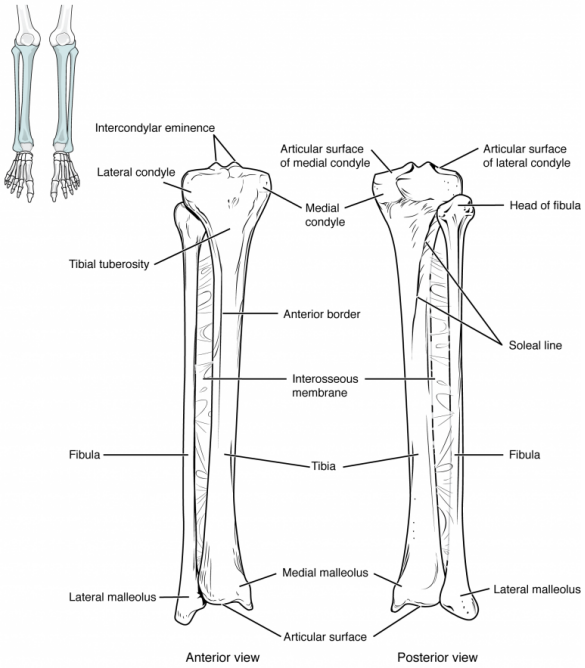


Figure 3.2: The tibia and fibula bones of the human leg. Copied from [Big23] without adaptation which is published under the CC-BY-SA 4.0 license [CCBYSA]. Therefore this figure is exempt from the copyright covering this thesis.

extension in other joints. Of course, the foot has other movement capabilities. However those are in the subtalar joint not in the ankle as [Whi14] points out, and cannot be distinguished during gait analysis and thus the foot's motions are usually referenced with the ankle/subtalar complex. The foot bones are divided into three groups, namely, the tarsals, the metatarsals, and the phalanges, see Figure 3.3. The tarsals are seven bones that form the ankle and the heel. The largest tarsal bone is the calcaneus, which bears most of the body weight when standing. The metatarsals are five long bones that form the arch of the foot and connect to the toes. The phalanges are 14 small bones that make up the digits of the foot where each toe has three phalanges, except

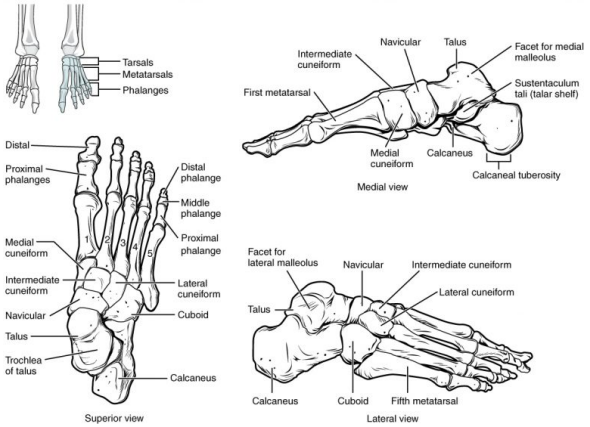


Figure 3.3: The bones of the human foot. Copied from [Big23] without adaptation which is published under the CC-BY-SA 4.0 license [CCBYSA]. Therefore this figure is exempt from the copyright covering this thesis.

for the big toe, which has two. In total, each leg consists of 30 bones (with 26 of them being in the foot) and not counting the pelvis.

Walking is a complex activity that engages about 200 muscles in the body [Cha94]. Often a muscle is attached at its origin point to one bone, which does not move during muscle contraction, over a larger area and then narrows into a tendon at its other end, connecting to the moving bone of the joint at the muscle insertion point. While ligaments and tendons are similar, as a general rule they can be distinguished by what they connect. Ligaments connect two bones while tendons connect a muscle to a bone [Whi14]. While most muscles have multiple actions, the main actions for flexion and extension on each joint are briefly summarised from [Whi14], since those are the largest movements during normal gait. Flexion of the hip joint is driven by the *Psoas major*, the *Iliacus*, the *Sartorius* and *Rectus femoris*, the later of which is part of the quadriceps group, shown in Figure 3.4. Extension of the hip is carried out by the *Gluteus maximus*. Knee extension is driven by the quadriceps, a group of the four muscles, which combine into the quadriceps tendon, see Figure 3.5. The group consists of the *Vastus medialis*, the *Vastus intermedius*, the *Vastus lateralis* and *Rectus femoris*, while the knee flexion is mainly carried out by the hamstrings, the *Semimembranosus*, the *Semitendinosus* and the *Biceps femoris*. Dorsifelxion of the foot is performed mainly by the *Soleus*, while plantarflexion is mainly driven by the *Tibialis anterior*, shown in Figure 3.6.

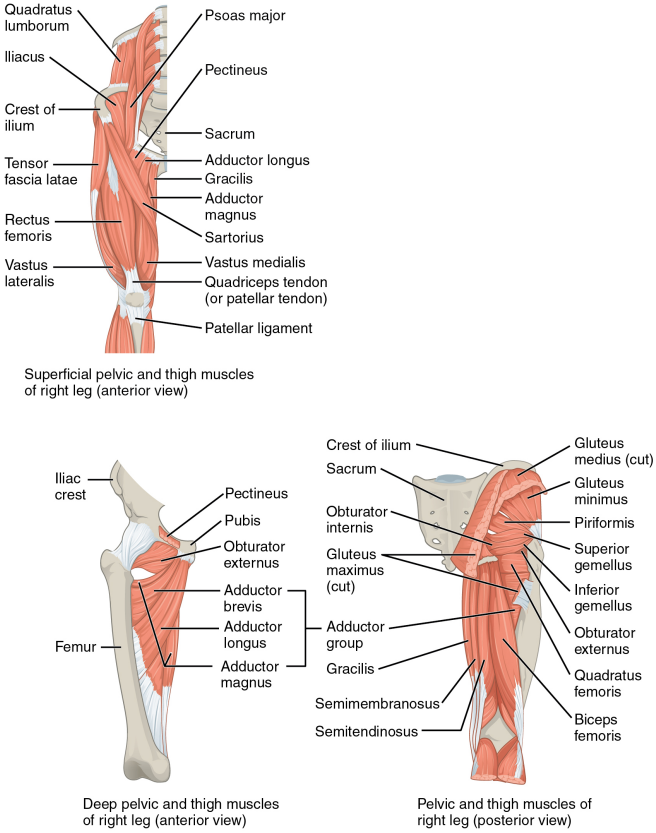


Figure 3.4: The gluteal muscles responsible for moving the femur. Copied from [Big23] without adaptation which is published under the CC-BY-SA 4.0 license [CCBYSA]. Therefore this figure is exempt from the copyright covering this thesis.

The human leg has many more muscles, the scope of which goes beyond this work. However, it should be noted that all of these play an important role in gait, like providing energy and actuation for the walking motion, and their functions need to be adequately replaced by a prosthesis.

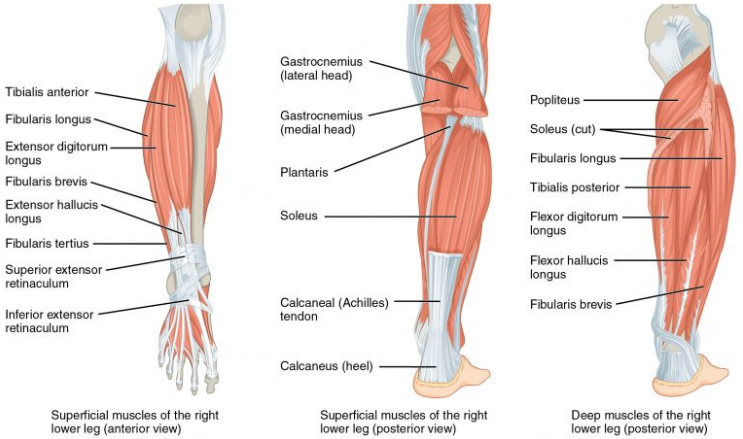


Figure 3.5: The muscles of the human leg that move the foot and toes. Copied from [Big23] without adaptation which is published under the CC-BY-SA 4.0 license [CCBYSA]. Therefore this figure is exempt from the copyright covering this thesis.

3.2 The human gait cycle

While walking is a familiar process, to analyse it, some form of definition is necessary. In [Whi14], walking is defined as "a method of locomotion involving the used of the two legs, alternately, to provide both support and propulsion with at least one foot being in contact with the ground at all times". Often, gait and walking are not differentiated. However, walking is a specific form of gait, which describes a pattern or manner of movement of the limbs while moving. Thus, walking can be described by a gait cycle, introduced in the following. For a more in depth introduction on gait analysis and the gait cycle see [Whi14; Kir06].

The gait cycle describes the cyclic pattern of movement that occurs while walking and is usually counted from one heel strike (when the heel first touches the ground) to the next heel strike of the same leg [Whi14]. The gait cycle can be broken down into two primary phases, the stance phase and the swing phase which each consist of multiple sub-phase, as shown in Figure 3.7.

- Stance Phase – the period of the gait cycle when the foot is on the ground and bearing body weight
  - Heel strike (initial response, contact response, or weight accep-

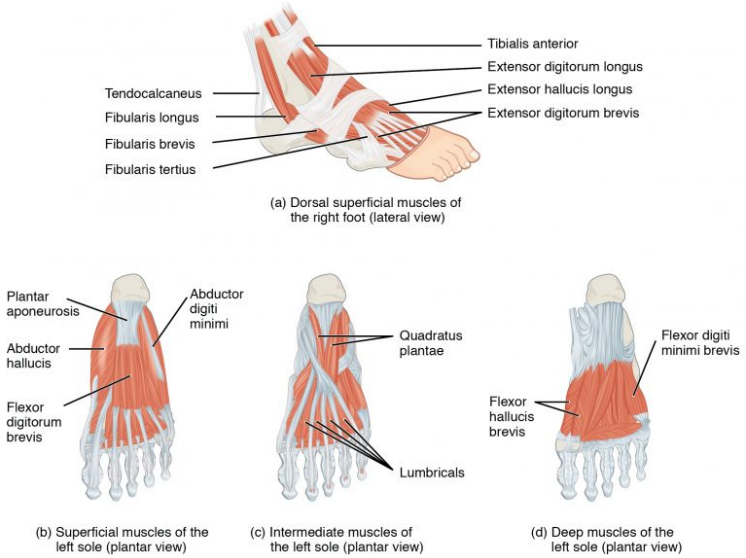


Figure 3.6: The muscles in the human foot. Copied from [Big23] without adaptation which is published under the CC-BY-SA 4.0 license [CCBYSA]. Therefore this figure is exempt from the copyright covering this thesis.

tance) – the heel of the foot first touches the ground
  - Foot flat – the entire foot is on the ground
  - Midstance – the body weight is right above the foot
  - Heel-off – the heel leaves the ground
  - Toe-off – the toes leave the ground
- Swing phase – the period of gait from toe-off to heel strike
  - Initial swing – just after toe-off
  - Mid-swing – the swinging leg is directly underneath the body
  - Terminal swing – just before heel strike

The initial contact of one foot occurs when the other is still on the ground, resulting in a double support period, in which the weight is transferred from one side to the other until the toe-off of the other foot occurs, resulting in a single support, until the process repeats in a mirrored way. Usually, the stance phase lasts 60% of the gait cycle and the swing phase lasts 40%, varying with

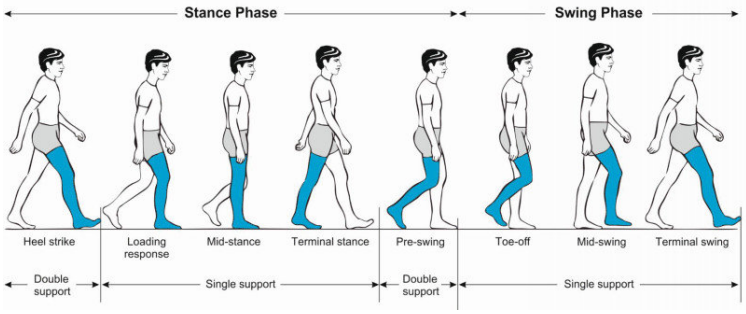


Figure 3.7: The human gait cycle, copied from [Pir17], originally published under CC BY 4.0 DEED [CCBY], thus it is exempt from the copyright of this work.

the speed of movement. The double support period usually takes 10% of the gait cycle. For a detailed analysis of gait cycle timing see [Whi14].

3.3 Replacing the human foot with a prosthesis

In developed countries, the main reason for amputations are circulatory dysfunctions, while in developing countries, most amputations are due to trauma from conflict, or industrial or traffic accidents [Mar01a]. An amputation leads to a permanent disfigurement and loss of function, and therefore drives the need for prosthetic replacements. A foot prosthesis, as is examined in this work, is an artificial device that replaces the function of a missing or amputated foot. The requirements of a foot prosthesis depend on the level and type of amputation, the activity level and lifestyle of the user, and the biomechanical and psychological functions that the prosthesis has to replace [Ver09; Ste18]. There are different types of prosthetic feet, see for instance [Haf05], which provide different levels of functionality and are the result of ongoing technological improvement.

This work examines the Össur Vari-Flex®, a so called energy storage and return (ESAR) prosthesis [VARI]. Such ESAR prostheses are said to improve gait in patients [Haf02] when compared to other passive prostheses. Passive prosthetic feet have no motors or actuators which can provide energy and control to the gait motion, making them cheaper and more independent than active prosthetic feet. While active prosthetic feet can provide energy for the gait cycle of the patient and thereby improve the mobility, they require a power supply, usually a battery, which adds weight and requires control elec-

tronics, increasing the cost of the prosthesis. A foot prosthesis has to replace the functionality of the ankle-foot complex [Ver09]. Some of these are shock absorption, compliance to uneven terrain and push-off [Haf05]. These functions help improve the stability and balance, reduce the energy expenditure and prevent abnormal compensations in the residual limb and other joints. As [Des02] notes, the technical aspect and progress of prosthetics is only one part in fully rehabilitating a patient, since physical rehabilitation may be affected with psychological aspects of amputation. Psychological functions of a foot prosthesis include restoring the appearance and self-image of the user as well as the social perception of an amputation [Des02]. A prosthesis can enhance the confidence of a patient and can for instance facilitate the social participation. These functions help to improve the quality of life, mental health and well-being of the patient [Des02]. Therefore, a foot prosthesis should be designed to meet both the biomechanical and psychological needs of the user, as well as being lightweight, durable, low-maintenance and affordable. Due to the variability of humans and the resulting requirements for a prosthesis, simulations are an important part in the development and uncertainty is present from many sources and should be considered when designing future prosthetics.

3.4 Consequences for the model

The leg in this work is modelled with a thigh, shank and prosthetic foot, described in Section 4.2. The thigh and shank are considered rigid bodies, with their anthropomorphic measures and inertia parameters based on [Cha75]. (These parameters are required for the biomechanical model to be able to perform forward dynamics simulations.) As mentioned, a biomechanical model has to find a balance between accuracy and complexity. The hip joint is modelled as a spherical joint between the reference frame and the thigh and the knee joint is modelled as a revolute joint, a common model simplification [San11]. A foot prosthesis is typically connected to the shank via a socket and pylon. The pylon acts as a support structure [Viv21] and is secured within a socket that cradles the residual limb for a snug fit. The socket is custom-fitted to the individual for optimal comfort and function, while the pylon provides the necessary rigidity and alignment. An osseointegrated socket embeds attachment points in the residual bone through a surgical procedure [Viv21]. Thus, since most foot prostheses are rigidly connected to the shank and the pylon aims to compensate for missing limb length and mass, the prosthesis in the model is rigidly fixed to the end of the shank. Muscles are currently not considered in the model, apart from their contribution to the mass of the leg. This work examines the two distinct phases of the gait cycle separately, see Section 4.2. This has the main advantage of further

reducing the model complexity, by eliminating the need for phase transitions and associated changes in contact and therefore constraints within the model. Without these changes, both phases can be simulated with forward dynamics, reducing the computational effort. The swing phase is examined by modelling the human leg and letting it swing freely under gravity, similarly to a double pendulum. The stance phase is examined by considering the leg in a squatting position and increasing the weight of the shank. This leads to a load on the prosthesis, similarly to the stance phase in gait.

3.5 Human gait analysis with optical motion capture (MoCap)

In order to simulate the human leg with a prosthetic foot, information and data are required about the gait cycle. For a detailed historical development of gait analyses see [Sut01; Sut02; Sut05; Col18]. In general, these consist of athropomorphic measures (leg length or mass) and data regarding the movement, such as limb positions, kinematics and joint angles. In the following sections, motion capture is introduced and some details on optical motion capture, as is used in this work, are described. Motion capture (MoCap) is a technique that records the movement of people, animals or objects and transfers the data to a computer program to enable photorealism in a virtual environment [Kad89] or further examination for research purposes [Whi14; Col18]. There are different types of MoCap, such as optical passive, optical active, markerless and inertial MoCap [Col18]. Optical passive MoCap is a commonly used method, which uses infrared cameras to track retroreflective passive markers attached to the subject [Gho19b]. Optical active MoCap uses LED markers that emit light instead of reflecting it. The main drawback of marker based MoCap is the requirement and dependancy on accurately placed markers. Markerless MoCap relies on software to detect the motion of the subject without markers, avoiding that downside at the cost of accuracy and precision in the measurement and is therefore rarely used in clinical applications or scientific analyses that require highly accurate position data of the motion. Of course, all video based MoCap is subject to common errors afflicting measurements relying on a camera, such as lighting and measurement noise. Inertial MoCap uses inertial measurement units (IMUs) worn by the subject that transmit data wirelessly to a computer or smart device. While requiring less accuracy during the IMU placement (being significantly cheaper and more robust than markers and avoiding camera based issues) IMUs are not as accurate as marker based methods and require a kinematic model and inverse kinematics to reconstruct the motion which introduces its own challenges. While markerless and IMU MoCap require less precision and effort in setup, the lack of accuracy and precision in the data lead to passive marker based optical motion capture being

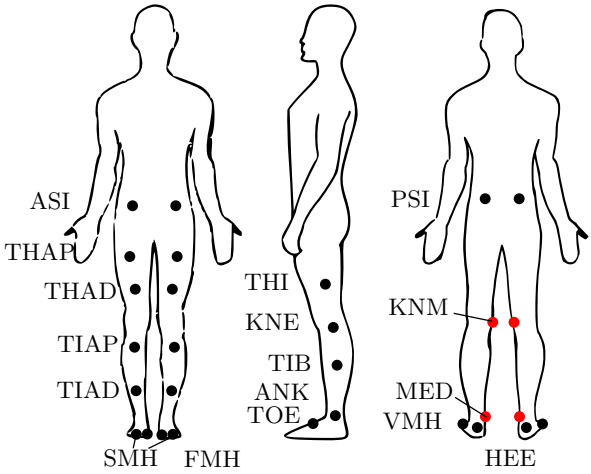


Figure 3.8: "The marker model used for gait analysis in this work, based on the CGM 2.4 marker model [Leb24] and used in [Sch24b]. (ASI – anterior pelvis; PSI – posterior pelvis; THAP – anterior proximal thigh; THAD – anterior distal thigh; THI – lateral thigh; KNE – lateral knee; KNM – medial knee; TIAP – anterior proximal shank; TIAD – anterior distal shank; TIB – lateral shank; ANK – lateral ankle; MED– medial ankle; HEE – heel; TOE – second toe; FMH – first toe; SMH – second toe; VMH – fifth toe). The red markers are only used for the static trial and are removed during gait measurements." Figure and caption quoted from [Sch24b].

the gold standard for MoCap [Cha21]. This technique involves placing reflective markers on specific anatomical landmarks or joints of the subject, such as the head, shoulders, elbows, wrists, hips, knees and ankles. The markers are then illuminated by infrared light sources and tracked by multiple cameras from different angles. The cameras capture the 2D coordinates of the markers in each frame and use triangulation to calculate their 3D positions. The resulting 3D data can then be used to reconstruct the motion of the subject and map it to a 3D model or character.

Marker based optical motion capture has several advantages over other MoCap techniques, such as high accuracy, resolution and reliability. It can also capture complex motions and subtle details that are difficult to achieve with other methods. However, it also has some disadvantages, such as occlusion, which occurs when a marker is hidden from one or more cameras, marker swapping, i. e., when two markers are confused by the software, marker drift, i. e., a marker moves from its original position on the subject, and noise, i. e.,

unwanted signals interfere with the data [Gho19b]. These issues can affect the quality of the motion capture data and require post-processing and editing to reduce them.

In addition to entertainment and gaming applications, marker based optical motion capture is also used for clinical purposes, such as diagnosing injuries, monitoring rehabilitation, assessing movement disorders, validating other measurement methods and evaluating treatment outcomes [Sco22; Gio22; Lam23]. For example, marker based optical motion capture can be used to detect and identify movement limitations or specific movement patterns of patients with certain diseases, such as Parkinson's disease, stroke, cerebral palsy and osteoarthritis. Marker based optical motion capture can also provide accurate and objective measurements of joint angles, velocities, accelerations and forces that can be used to evaluate muscle and joint loading and other biomechanical parameters. Furthermore, marker based optical motion capture can be integrated with other technologies, such as electromyography (EMG), force plates, pressure sensors and virtual reality (VR), to enhance the analysis and feedback of human motion in clinical settings. The main method underlying MoCap is photogrammetry, i. e., the process of obtaining geometric information from images. In marker based optical MoCap, photogrammetry is used to track the motion of reflective or light-emitting markers attached to specific locations on the body of a human or an animal. Multiple cameras capture the images of the markers from different viewpoints, and then triangulation is used to compute the 3D coordinates of the markers. The resulting 3D trajectories are then labelled and mapped to a skeleton model to obtain the 3D pose and motion of the subject [Cha21; Gho19c; OPT; Ces14]. The CGM marker model [Leb24] is a widely used method for estimating lower limb joint kinematics from motion capture data, see [Bak17], and is used for this work's marker model, shown in Figure 3.8. It is validated against a radiographic method and showed good agreement for most joints, except for the ankle, which was affected by soft tissue artefact (STA) in [Dav91]. The clinical utility of the model was demonstrated by applying it to various populations, such as normal adults, children, cerebral palsy patients, and amputees. The model consists of rigid segments connected by functional or anatomical joint axes, and uses a set of markers attached to the segments to define the segment coordinate systems and joint angles. It defines segment axes and joint angles based on [Gro83]. The model has been updated and modified over the years to improve its accuracy and applicability. The CGM marker model was widely adopted by the biomechanics community and became a standard method for reporting lower limb joint kinematics in gait analysis. The International Society of Biomechanics (ISB) recommended the use of the CGM marker model as a common framework for comparing and validating results across different studies and laboratories [Wu02] and the model is also incorporated into various commercial software packages, such as VICON Plug-in Gait. For the cur-

rent marker model and a palpation guide, see [Leb24], since the CGM model is continuously developed, for instance in [PBM19; Leb19]. However, while being a commonly used marker model, CGM is still subject to the same problems as other marker models and marker based measurement procedures, such as marker occlusion or measurement noise [Cha21; Gho19c; Gho19b; Lee19; Yam21]. Several methods have been proposed to address these issues, such as data-driven approaches, using additional markers, functional calibration methods, inverse kinematics or statistical models [Cha21; Gho19c; Gho19b; Yam21]. However, these issues remain a challenge. This highlights the importance of considering uncertainty in the process of subsequent calculations based on marker data.

3.6 Joint angle calculation based on optical marker data

A common application for MoCap is the subsequent calculation and evaluation of joint angles based on the marker position data. The joint angle calculations, as used in [Sch24b] as well as for the results shown later on in this thesis, are briefly summarised here.
The marker position data from the MoCap is preprocessed by checking for large gaps in the data and interpolating them via splines, if necessary. High frequency noise is filtered out via a lowpass Butterworth filter with a cut-off frequency of 6 Hz and order four, as suggested in [Ric08]. From this preprocessed marker position data, the joint angles are calculated directly in Matlab 2023b. In this work, the Grood & Suntay convention [Gro83] is used due to is common use, see [Dre22; Mas22; Wu02]. It assigns a local segment coordinate systems (SCS), consisting of the origin and orthonormal axes, to every body part. The axes are defined from anatomical landmarks and distal and proximal joint centres [Wu02; Bak17]. Based on two SCSs that are adjacent to the joint, the joint angles are calculated for each joint and can be interpreted as flexion, abduction and rotation angles. For the detailed procedure used in this thesis see [Sch24b], which is based on [Gro83]. To summarise, the marker positions as reported form the MoCap are used to calculate antropomorphic measures and the SCSs, from which the joint angles are calculated. It should be noted, that in this work, the anthropomorphic measures are not considered for the uncertainty calculations [Sch24b].

3.7 Epistemic uncertainty in marker positions during optical MoCap

Since optical marker based motion capture is commonly used as the gold standard to evaluate other measurement methods [Cha21], it is important to consider possible errors in the measurement. An idea was contributed to this, by

introducing an error model for marker position errors using epistemic uncertainty, which is then propagated through subsequent joint angle calculations with the Graph Follower algorithm. This resulted in the journal publication [Sch24b]. The proposed error model and method for its consideration allow for an efficient worst-case analysis while simultaneously providing a bidirectional correlation of marker error magnitude and the resulting joint angle calculations. Before detailing the error model, the error sources that are considered and used for the uncertainty quantification are introduced.

3.7.1 Sources of uncertainty in motion capture

The following is quoted from my previous work [Sch24b] Section 1.2. Variable names in the text and figures have been changed to better fit this thesis. Figures and tables have been modified to suit the format of this thesis.
“
Knowledge about the source and magnitude of the errors is required to model the uncertainty appropriately. According to [Lea05; Cap05; Chi05; Del05], the main error sources occuring during marker based optical motion capture are general measurement system uncertainty, placement errors of markers and soft tissue artefacts.

General measurement uncertainty The measurement system introduces a deviation between the physical location of the marker and the reported position by the measurement system. This depends on the system used, the diligence during the calibration, camera's positions, marker diameter and measurement volume and is independent of the marker's purpose.

Errors introduced by marker occlusion and other system-based errors [Top20; Van18; Con21], are not within the scope of this work. As with all modelling approaches, a reduction of complexity is necessary, while accepting the introduction of more uncertainty [Kiu09].

Placement error Incorrect placement of markers by the examiner occurs especially when easy-to-identify anatomical landmarks are not available. In this work, inter-examiner placement errors are used as a reference for the errors introduced by the manual placement of markers. Table 3.1 shows the literature values found in the summary series of [Lea05; Cap05; Chi05; Del05] for inter-examiner accuracy for the various lower body segments along with the assumed errors that are used in the calculations later on.

Soft tissue artefacts Soft tissue artefacts refer to the relative movement between surface markers on the soft tissue (e. g. muscle and skin) and the underlying bone and are the main known source of uncertainty in marker based

Table 3.1: Summary of the error sources affecting the markers, alongside the literature averages (lit. avg., summarised from [Lea05; Cap05; Del05; Chi05]) and assumed errors that are used for the simulation in this work with the maximum modelled deviation $\sum$. The detailed location of the markers can be seen in Figure 3.8. All values are in mm.

markers	segment	measurement system error	placement error		soft-tissue artefact error		$\sum$
			lit. avg.	used	lit. avg.	used	
ASI,PSI	pelvis	2	14.4	15	13.5	15	32
THAP, THAD, THI,KNE	thigh	2	10.3	10	13.5	20	32
TAIP, TIAD, TIB	shank	2	7.2	8	16.3	20	30
ANK	ankle	2	n.a.	8	n.a.	10	20
HEE, TOE, VMH	foot	2	8.5	5	n.a.	0	7

motion capture. These artefacts vary between study participants, the recorded movements, marker location and marker type. Table 3.1 shows values of soft tissue artefacts occurring during normal human gait for different anatomical landmarks based on [Lea05; Cap05; Chi05; Del05].
”

3.7.2 Modelling marker position uncertainty

The following is quoted from my previous work [Sch24b] Chapter 2.5 to 2.6. Variable names in the text and figures have been changed to better fit this thesis. Figures and tables have been modified to suit the format of this thesis.
“

The epistemic uncertainty has to be quantified, based on the data shown in Table 3.1, before it can be propagated from the marker positions to the joint angles. The values in Table 3.1 are based on [Lea05; Cap05; Del05; Chi05]. To account for marker occlusion and other system based errors, a slightly larger measurement system error is assumed. Furthermore, we assume the placement errors to be different for different body segments and markers. We also assume that markers placed on joints have the same error values as the joint's proximal limb. Furthermore, a model needs to be developed that can

apply the now quantified uncertainty to the data used for calculating joint angles.

Specifically, the possible deviation of the actual marker position from the measured position is modelled as a triangular fuzzy number (Figure 3.9(a)).

[...]

Here, the fuzzy number quantifies the possible deviation of the marker's measured position from the it's real position. The size of this deviation is quantified based on literature values [Lea05; Cap05; Del05; Chi05] and summarised in Table 3.1. This assumed deviation limits $p_{\min}, p_{\max}$, the smallest and largest values of the fuzzy number, respectively.

To incorporate uncertainty in the marker's position, we assume that the marker position is located within a sphere around the measured position. The sphere's radius equals the cumulative error $\sum$ from Table 3.1 for the respective marker position on the segment. To model the sphere around the measured position $\boldsymbol{r}_i$ of the marker i, a deviation $\tilde{\boldsymbol{p}}_i$ is added to its measured position (Figure 3.9(c)). To allow the marker to be anywhere inside the sphere, the norm of the total deviation $||\tilde{\boldsymbol{p}}_i||$ is smaller or equal to a maximum allowed deviation such that $||\tilde{\boldsymbol{p}}_i|| \leq \sum_i$ (from Table 3.1). Note that the possible deviation interval is symmetric $p_{i,\max} = \sum_i$ and $p_{i,\min} = -\sum_i$ and

$$\begin{aligned} \tilde{\boldsymbol{r}}_i &= \boldsymbol{r}_i + \tilde{\boldsymbol{p}}_i \\ &\text{with } ||\tilde{\boldsymbol{p}}_i|| \leq p_{max} \end{aligned} \tag{3.1}$$

To model the displacement of a marker i in 3D space, its displacement $\tilde{\boldsymbol{p}}_i = [p_{x,i}, p_{y,i}, p_{z,i}]^T$ consists of three different variables for the optimisation in the Graph Follower algorithm and is subject to the constraint $||\tilde{\boldsymbol{p}}_i|| \leq \sum_i$. However, this constraint induces a dependency between the three spatial directions of $\tilde{\boldsymbol{p}}_i$. In general, the α-level optimisation requires the optimisation variables to be independent of each other [Möl00], since additional uncertainty can be introduced by constraints. To investigate, whether additional uncertainty is introduced by the constraints, the deviation is implemented with Cartesian coordinates and spherical coordinates. Both methods yielded the same results for the joint angle envelopes, suggesting that no additional uncertainty is introduced here by the constraint. Based on this, the Cartesian coordinate formulation is used for all calculations in this work.

3.7.3 Fuzzy joint angle calculation

After modelling the error in the markers' positions with epistemic uncertainty, this uncertainty is propagated through the joint angle calculations with the

Graph Follower algorithm [Eis19b]. The uncertain deviation of all n_M markers $\tilde{\boldsymbol{p}} = [\tilde{\boldsymbol{p}}_1, \tilde{\boldsymbol{p}}_2, \ldots, \tilde{\boldsymbol{p}}_{n_M}]$ is called the input parameter. The goal of propagating input parameters affected with epistemic uncertainty is to calculate the fuzzy target quantity $\tilde{f}(\tilde{\boldsymbol{p}}, t)$, in our case, the joint angle for all timeframes. To do this, the output interval of the target quantity $f(\boldsymbol{p}, t)$, along with its membership function $\mu_{\tilde{f}}(\tilde{\boldsymbol{p}})$ is calculated via α-level optimisation. These two values together approximate the fuzzy target quantity $\tilde{f}(\tilde{\boldsymbol{p}}, t)$.

Figure 3.9 shows two steps in this process. The triangular fuzzy number for the input parameters (Figure 3.9(a)) is discretised with α-level cuts. The k-th α-level $\alpha_k \in [0, 1]$ defines a subinterval $[\boldsymbol{p}_{min,\alpha_k}, \boldsymbol{p}_{max,\alpha_k}]$ of the input parameter uncertainty interval $[\boldsymbol{p}_{\min}, \boldsymbol{p}_{\max}]$ so that

$$\mu_{\tilde{\boldsymbol{p}}}(\tilde{\boldsymbol{p}}_{\alpha_k}) \geq \alpha_k \;\forall\; \tilde{\boldsymbol{p}}_{\alpha_k} \in [\boldsymbol{p}_{min,\alpha_k}, \boldsymbol{p}_{max,\alpha_k}] \tag{3.2}$$

and $\mu_{\tilde{\boldsymbol{p}}}(\boldsymbol{p}_{min,\alpha_k}) = \mu_{\tilde{\boldsymbol{p}}}(\boldsymbol{p}_{max,\alpha_k}) = \alpha_k$. Due to the convexity of the triangular input fuzzy number, the membership function value α_k for the target output $\mu_{\tilde{f}}(\tilde{f}_{\alpha_k})$ is inherited from the input α-level, [Ngu78], in the α-level optimisation. This means the target output membership function value for the current α-level is known

$$\mu_{\tilde{f}}(\tilde{f}_{\alpha_k}) \geq \alpha_k \;\forall\; \tilde{f}_{\alpha_k} \;\in\; [f_{min,\alpha_k}, f_{max,\alpha_k}] \tag{3.3}$$

and $\mu_{\tilde{f}}(f_{min,\alpha_k}) = \mu_{\tilde{f}}(f_{max,\alpha_k}) = \alpha_k$. The time interval $t \in [0, T]$ is discretised with time frames t_j. Then the α-level optimisation calculates the largest (f_{max,α_k}) and smallest (f_{min,α_k}) values of the target output $f(\tilde{\boldsymbol{p}}, t_j)_{\alpha_k}$ (Figure 3.9(b)) at the timeframe t_j, for the α-level α_k. Thus, the fuzzy target output $\tilde{f}(\tilde{\boldsymbol{p}}, t)$ is approximated by calculating the target output interval $[f_{min,\alpha_k}, f_{max,\alpha_k}]$ via optimisation individually for multiple α-levels for all timeframes. Of course, the approximation accuracy increases with the number of examined α-levels. To get the entire joint angle evolution over time, this optimisation is performed at every timeframe of the measurement and for every considered α-level.

In this work, the Graph Follower algorithm [Eis19b] performs the abovementioned steps for α-level optimisation. Specifically, the *OP-II* version of the Graph Follower algorithm is used, which does not linearise the target function for the optimisation. This is possible due to the low computational cost of calculating the joint angles and leads to more accurate envelopes than using a linearised target function for the optimisation (*OP-III* in [Eis19b]). The Graph Follower algorithm uses additional postprocessing steps to further enhance the target output envelopes. The individual trajectories of the target output calculated during one α-level are stored in a library and a postprocessing step ensures that the maximum or minimum of all calculated trajectories is used in the final envelopes for every α-level.

Implementation of the Graph Follower algorithm The Graph Follower algorithm is implemented in Matlab 2021a. The optimisation is performed by the provided *fmincon* function, using MultiStart with 15 points. The algorithm *'interior-point'* is used for the optimisation and the tolerance settings for *fmincon* are set as *'OptimalityTolerance'* $= 1e-12$ and *'StepTolerance'* $= 1e-12$. For more details on the Graph Follower algorithm and its performance see [Eis19a; Eis19b].
"

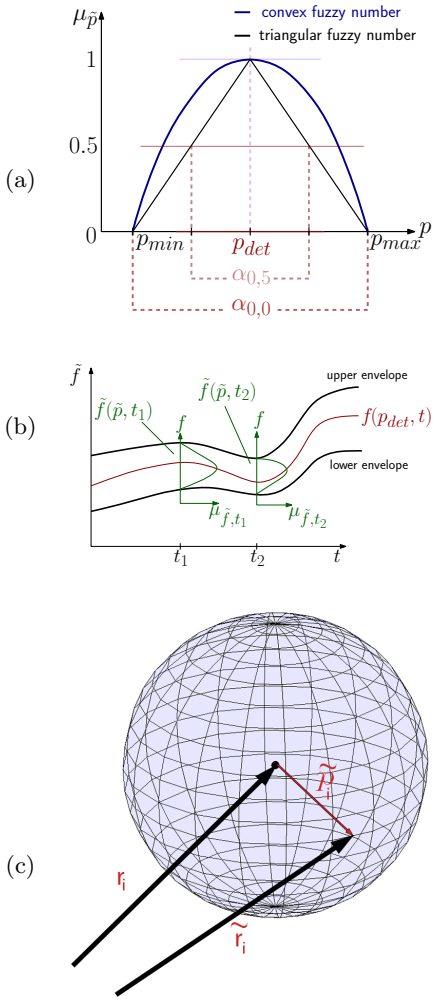


Figure 3.9: "(a) shows a convex fuzzy number and its approximation by a triangular fuzzy number as required for the Graph Follower algorithm, alongside an exemplary α-level discretisation. (b) shows the resulting envelopes when the fuzzy parameter p is propagated to the target output with the Graph Follower algorithm. (c) shows how the measured marker position $\boldsymbol{r}_i$ of marker i is affected with an uncertain deviation $\tilde{\boldsymbol{p}}_i$ allowing the marker position to be anywhere in a sphere around the measured position $\tilde{\boldsymbol{r}}_i$, which has the radius of the maximum allowed deviation." Figure and caption quoted from [Sch24b].

Chapter 4

The biomechanical model

Simulating the human leg with a prosthetic foot in the presence of uncertainty requires a biomechanical model of the human leg and the prosthesis. A biomechanical model is a mathematical representation of the mechanics in a biological system, in this case the human leg with a carbon spring prosthesis. As with other simulations, the main advantage of such a model is, that it can simulate different situations or configurations, for instance prosthesis designs, without requiring cost-intensive real measurements and manufacturing. Especially for biological systems, it has the further advantage that it can enable the observation of quantities that can not be easily measured in vivo. For the prosthesis, one such quantity is the stored energy during the gait cycle. One critical aspect when designing a prosthesis is the prosthesis' energy storage and return behaviour, since it is related to the walking comfort of the patient. However, to be useful, the model has to be as accurate and as detailed as possible, while still being feasible to simulate on current hardware. The consideration of computational effort required to simulate the model is even more important with the many evaluations required for uncertainty simulations. This results in a trade off of model complexity and accuracy with computational cost. In the following sections, the model developed and used in this work is described. The modelling process of the prosthesis with geometrically exact beam theory and the expansion to include a leg model are described, starting with a brief summary of rigid multibody dynamics. Then constrained Lagrangian mechanics is briefly recapped before summarising the derivation of the variational integrator used for the forward dynamics simulation. With the theory recapped, the leg model is described and the prosthesis is modelled with geometrically exact beam theory. Once the model is available, two simulation scenarios are described, based on the human gait cycle followed by including homogenisation of the prosthesis' layers in the model.

Previous works This chapter is based on, and uses parts, of chapters in the following publications. The model is based on the two proceedings [Sch21a; Sch21b], the journal papers [Sch22b; Sch24b] and the book chapter [Sch24a] and on the student thesis [Blo23; Sch19].

4.1 Biomechanical modelling

Creating a biomechanical model involves several steps that, depending on the type of biomechanical model, require expertise in biology, physics, mathematics, and computer science. When creating a model of a given biological system, the goal is to create an as accurate as possible mathematical representation of the system, in this case, the human leg with a prosthetic foot during gait. As with any model, some reduction in complexity is necessary while introducing more uncertainty [Kiu09]. An accurate representation of the leg's dynamics requires parameters such as mass and inertia and anthropomorphic measures such as length and joint positions are necessary. Furthermore, a method of mathematically describing the movement is required. As for the prosthesis, an accurate representation of its elastodynamic behaviour is desired, while at the same time not being computationally expensive. Before detailing the model, the required theories are briefly summarised.

4.1.1 Rigid multibody dynamics

The leg model can be formulated as an open multibody chain with the theory from [Woe11; Lan11; Ley08a; Bet06]. For this, the k-th body is described by its configuration vector $\mathfrak{q}^k$, which consists of the position of its center of mass $\boldsymbol{\varphi}^k$ and an orthonormal director triad $\boldsymbol{d}_i^k$ with $i = 1, 2, 3$ relative to the frame of reference such that

$$\mathfrak{q}^k = \left[\boldsymbol{\varphi}^k,\ \boldsymbol{d}_1^k,\ \boldsymbol{d}_2^k,\ \boldsymbol{d}_3^k\right]^{\mathrm{T}} \quad \in \quad \mathbb{R}^{12}. \tag{4.1}$$

The configuration $\mathfrak{q}^k \in \mathbb{R}^{12}$ has twelve degrees of freedom, so for one body $n_{\mathrm{dof}} = 12$. For a multibody system with n_b individual rigid body $n_{\mathrm{dof}} = 12n_b$. However, a single rigid body only has six degrees of freedom, three rotational ones, three translational ones. To correctly model the rigid body's behaviour, six so called internal constraints $\boldsymbol{g}_{\mathrm{int}}(\mathfrak{q}^k)$ are used, which reduce the degrees of freedom to six,

$$\boldsymbol{g}_{\mathrm{int}}(\mathfrak{q}^k) = \begin{bmatrix} \frac{1}{2}[(\boldsymbol{d}_1^k)^T \cdot \boldsymbol{d}_1^k - 1] \\ \frac{1}{2}[(\boldsymbol{d}_2^k)^T \cdot \boldsymbol{d}_2^k - 1] \\ \frac{1}{2}[(\boldsymbol{d}_3^k)^T \cdot \boldsymbol{d}_3^k - 1] \\ (\boldsymbol{d}_1^k)^T \cdot \boldsymbol{d}_2^k \\ (\boldsymbol{d}_3^k)^T \cdot \boldsymbol{d}_1^k \\ (\boldsymbol{d}_2^k)^T \cdot \boldsymbol{d}_3^k \end{bmatrix} = \mathbf{0} \in \mathbb{R}^6. \tag{4.2}$$

The first three constraints in $\boldsymbol{g}_{\mathrm{int}}(\mathfrak{q}^k)$ ensure that the unit vectors remain at unit length while the last three ensure their orthogonality to each other. For a multibody system, the configuration of the n_b bodies can be grouped into

$\mathfrak{q} \in \mathbb{R}^{12n_b}$ as

$$\mathfrak{q} = \left[\mathfrak{q}^1, \mathfrak{q}^2, \ldots, \mathfrak{q}^k, \ldots, \mathfrak{q}^{n_b}\right]^T \in \mathbb{R}^{12n_b}. \tag{4.3}$$

Here, the model consists of the thigh, shank and prosthesis, thus $n_b = 3$ and $k = 1, 2, 3$. Similarly, the internal constraints can be grouped together, ensuring the rigid body behaviour for all bodies in the model

$$\boldsymbol{g}_{\text{int}}(\mathfrak{q}) = \left[\boldsymbol{g}_{\text{int}}(\mathfrak{q}^1), \boldsymbol{g}_{\text{int}}(\mathfrak{q}^2), \ldots, \boldsymbol{g}_{\text{int}}(\mathfrak{q}^k), \ldots, \boldsymbol{g}_{\text{int}}(\mathfrak{q}^{n_b})\right]^T \in \mathbb{R}^{6n_b}. \tag{4.4}$$

With the velocities $\dot{\mathfrak{q}} = \frac{\mathrm{d}\mathfrak{q}}{\mathrm{d}t}$ and the mass matrix $\boldsymbol{M}$ (which includes the principal values of the Euler tensor with respect to the body's centre of mass), the kinetic energy $\mathbb{T}$ can be defined, as seen for instance in [Woe11], as

$$\mathbb{T}(\dot{\mathfrak{q}}) = \frac{1}{2}\dot{\mathfrak{q}}^T \boldsymbol{M} \dot{\mathfrak{q}}. \tag{4.5}$$

The potential energy $\mathbb{V}$ consists of the external potential energy due to the gravitational field described by the gravitation force vector $\mathfrak{g}$ and the internal potential energy due to the deformation of the prosthesis $\mathbb{V}_{\text{int}}$

$$\mathbb{V} = (\boldsymbol{\varphi} - \boldsymbol{\varphi}_0)^T m\mathfrak{g} + \mathbb{V}_{\text{int}}, \tag{4.6}$$

where $\boldsymbol{\varphi}_0$ is the initial position of the rigid body. The calculation of the internal deformation energy of the prosthesis is described in more detail in Section 4.2.1, see the stored energy function $\mathbb{V}_{\text{int}}$ in Eq. (4.30).
To create the multibody model, it can be built as an open kinematic chain, where the bodies are connected by joints. The human leg in this work is modelled with three types of joints, a spherical joint for the hip, a revolute joint in the knee and the ankle is fixed rigidly to the shank. These joints have the following external constraints $\boldsymbol{g}_{ext}(\mathfrak{q})$, where the index $i = 1, 2$ in $\square^i$ refers to the respective rigid body of the joint, $\boldsymbol{\varrho}$ is the distance to the joint centre and $\boldsymbol{n}$ is the revolute joint's rotational axis, which is fixed to the first body of the joint. The initial configuration of the the joint is defined by η.

Rigidly connecting two rigid bodies To rigidly connect two rigid bodies to each other, as is used for the connection of the prosthesis to the leg's shank the following constraint can be used

$$\boldsymbol{g}_{\text{ext}}(\mathfrak{q}) = \begin{bmatrix} (\boldsymbol{\varphi}^2 + \boldsymbol{\varrho}^2) - (\boldsymbol{\varphi}^1 + \boldsymbol{\varrho}^1) \\ (\boldsymbol{d}_1^1)^T \cdot \boldsymbol{d}_1^2 - \eta_1 \\ (\boldsymbol{d}_1^2)^T \cdot \boldsymbol{d}_2^2 - \eta_2 \\ (\boldsymbol{d}_1^3)^T \cdot \boldsymbol{d}_3^2 - \eta_3 \end{bmatrix} = \boldsymbol{0} \in \mathbb{R}^6. \tag{4.7}$$

Spherical joint For the spherical joint, used for the hip joint in the model, only the centres of mass of the two bodies are constrained

$$\boldsymbol{g}_{\text{ext}}(\mathbb{q}) = \left[(\boldsymbol{\varphi}^2 + \boldsymbol{\varrho}^2) - (\boldsymbol{\varphi}^1 + \boldsymbol{\varrho}^1)\right] = \mathbf{0} \in \mathbb{R}^3. \tag{4.8}$$

Revolute joint For a revolute joint, rotation is only permitted around one axis, leading to the constraints

$$\boldsymbol{g}_{\text{ext}}(\mathbb{q}) = \begin{bmatrix} (\boldsymbol{\varphi}^2 + \boldsymbol{\varrho}^2) - (\boldsymbol{\varphi}^1 + \boldsymbol{\varrho}^1) \\ (\boldsymbol{n})^T \cdot \boldsymbol{d}_1^2 - \eta_1 \\ (\boldsymbol{n})^T \cdot \boldsymbol{d}_2^2 - \eta_2 \end{bmatrix} = \mathbf{0} \in \mathbb{R}^5. \tag{4.9}$$

This is used to approximate the knee joint in the model.

Anchoring a rigid body To prevent one or more bodies of the multibody system from moving from their initial configuration, constraints can enforce the configuration of the body to stay constant in time,

$$\boldsymbol{g}_{\text{ext}}(\mathbb{q}) = \mathbb{q}^k(t) - \mathbb{q}^k(t_0) = \mathbf{0} \in \mathbb{R}^6. \tag{4.10}$$

This is used for the fixation of two of the prosthesis' nodes in the load scenario, simulating permanent ground contact.

4.1.2 Lagrangian mechanics

This work uses a Variational Integrator (VI), developed in [Hai00; Hai06; Moo03; Ley08b; Mar01b], for the forward dynamics simulation of the model, due to its excellent long term energy behaviour [Moo03; Hai06]. They are derived by applying a discrete version of the variational principle from [Mar01b] to the discrete Lagrangian, resulting in a discrete time-stepping scheme, see [Ley08b; Ley08a]. For constrained systems, the augmented Lagrangian is used. For an excellent summary on mechanics including variational principles see [Cli17].

Continuous constrained Lagrangian mechanics The Lagrange function $L(\mathbb{q}, \dot{\mathbb{q}})$ is defined as the difference between the system's kinetic energy $\mathbb{T}(\dot{\mathbb{q}})$ and its potential energy $\mathbb{V}(\mathbb{q})$

$$L(\mathbb{q}, \dot{\mathbb{q}}) = \mathbb{T}(\dot{\mathbb{q}}) - \mathbb{V}(\mathbb{q}) \quad \in \mathbb{R}. \tag{4.11}$$

In multibody systems and rigid bodies, as described in Section 4.1.1, the number of configuration variables is larger than the degrees of freedom of the system. To correctly represent the desired system, constraint equations must

be satisfied. They are considered via the augmented Lagrangian $\bar{L}(\mathrm{q}, \dot{\mathrm{q}}, \boldsymbol{\lambda})$, which is defined as

$$\bar{L}(\mathrm{q}, \dot{\mathrm{q}}, \boldsymbol{\lambda}) = L(\mathrm{q}, \dot{\mathrm{q}}) - \boldsymbol{g}^T(\mathrm{q}) \cdot \boldsymbol{\lambda}, \tag{4.12}$$

with the system's constraints $\boldsymbol{g}(\mathrm{q})$ and the Lagrange multipliers $\boldsymbol{\lambda}$. Based on this, the system's action is defined as

$$S = \int_{t_0}^{t_{N_t}} \bar{L}(\mathrm{q}, \dot{\mathrm{q}}, \boldsymbol{\lambda}) \mathrm{d}t. \tag{4.13}$$

Here, $t \in \mathbb{R}$ parametrises time in the interval $[t_0, t_{N_t}]$. The principle of stationary action or Hamilton's principle states that a trajectory is a stationary point of the action for variations $\delta\mathrm{q}(t)$ that vanish at the end points $\delta\mathrm{q}(t_0) = \delta\mathrm{q}(t_{N_t}) = \mathbf{0}$, i. e.,

$$\begin{aligned} \delta S &= \delta \int_{t_0}^{t_{N_t}} \bar{L}(\mathrm{q}, \dot{\mathrm{q}}, \boldsymbol{\lambda}) \mathrm{d}t = 0 \quad \forall\, \delta\boldsymbol{\lambda} \;\text{ and }\; \forall\, \delta\mathrm{q} \;:\; \delta\mathrm{q}(t_0) = \delta\mathrm{q}(t_{N_t}) = \mathbf{0} \\ &= \int_{t_0}^{t_{N_t}} \left(\frac{\partial L(\mathrm{q}, \dot{\mathrm{q}})}{\partial \mathrm{q}} \delta\mathrm{q} + \frac{\partial L(\mathrm{q}, \dot{\mathrm{q}})}{\partial \dot{\mathrm{q}}} \delta\dot{\mathrm{q}} - (\boldsymbol{G}^T(\mathrm{q})\boldsymbol{\lambda})^T \delta\mathrm{q} - \boldsymbol{g}^T(\mathrm{q}) \delta\boldsymbol{\lambda} \right) \mathrm{d}t = 0 \end{aligned} \tag{4.14}$$

Applying integration by parts after varying $\bar{L}$ and applying the fundamental lemma of the calculus of variations leads to the well known, constrained second order Euler-Lagrange equations [Mar01b],

$$\begin{aligned} \frac{\partial L}{\partial \mathrm{q}} - \frac{\mathrm{d}}{\mathrm{d}t} \left(\frac{\partial L}{\partial \dot{\mathrm{q}}} \right) - \boldsymbol{G}^T(\mathrm{q}) \cdot \boldsymbol{\lambda} = \mathbf{0} \quad &\in \mathbb{R}^{n_{\mathrm{dof}}} \\ \boldsymbol{g}(\mathrm{q}) = \mathbf{0} \quad &\in \mathbb{R}^{n_c}, \end{aligned} \tag{4.15}$$

where $\boldsymbol{G}(\mathrm{q}) = \frac{\partial}{\partial \mathrm{q}} \boldsymbol{g}(\mathrm{q})$ is the constraint Jacobian and n_c is the number of constrained degrees of freedom. The constraint forces limiting the system's movement are represented by $-\boldsymbol{G}^T(\mathrm{q}) \cdot \boldsymbol{\lambda}$, whose direction and value is given by the Lagrange multipliers $\boldsymbol{\lambda} \in \mathbb{R}^{n_c}$. This allows for the derivation of a system's equations of motion, which can then be solved via numerical integration from the initial conditions $\mathrm{q}(0) = \mathrm{q}_0$ and $\dot{\mathrm{q}}(0) = \dot{\mathrm{q}}_0$. However, not all integrators are suitable to preserve a system's structure. Variational integrators, derived from a discrete version of Lagrangian mechanics, preserve the system's structure and have excellent energy behaviour, leading to their use for all forward dynamics simulations in this work.

Discrete constrained Lagrangian mechanics The discrete augmented Lagrangian is defined similarly to its continuous counterpart, however, the previously continuous in time configuration $\mathfrak{q}(t)$ is discretised on a time grid of equidistant time nodes t_j on the time interval from $[t_0, t_{N_t}]$ with the timestep h, such that $t_j = t_0 + jh$. Denoting the approximation $\mathfrak{q}_j \approx \mathfrak{q}(t_j)$ and approximating the action integral with midpoint quadrature and the velocity $\dot{\mathfrak{q}}(t)$ with finite differences for the small time section $[t_j, t_{j+1}]$, the discrete Lagrangian is

$$L_d(\mathfrak{q}_j, \mathfrak{q}_{j+1}) = hL\left(\frac{\mathfrak{q}_{j+1} + \mathfrak{q}_j}{2}, \frac{\mathfrak{q}_{j+1} - \mathfrak{q}_j}{h}\right) \approx \int_{t_j}^{t_{j+1}} L(\mathfrak{q}, \dot{\mathfrak{q}})\mathrm{d}t. \quad (4.16)$$

To ensure that the constraints $\boldsymbol{g}$ are fulfilled to numerical accuracy at the discrete time nodes and not in between, a generalised trapezoidal rule is used

$$\frac{h}{2}\boldsymbol{g}(\mathfrak{q}_j) \cdot \boldsymbol{\lambda}_j + \frac{h}{2}\boldsymbol{g}(\mathfrak{q}_{j+1}) \cdot \boldsymbol{\lambda}_{j+1} \approx \int_{t_j}^{t_{j+1}} \boldsymbol{g}^T(\mathfrak{q}) \cdot \boldsymbol{\lambda}\mathrm{d}t, \quad (4.17)$$

as seen in [Ley06; Ley10]. This results in the discrete augmented Lagrangian

$$\bar{L}_d(\mathfrak{q}_j, \boldsymbol{\lambda}_j, \mathfrak{q}_{j+1}, \boldsymbol{\lambda}_{j+1}) = L_d(\mathfrak{q}_j, \mathfrak{q}_{j+1}) - \frac{h}{2}\left[\boldsymbol{g}^T(\mathfrak{q}_j) \cdot \boldsymbol{\lambda}_j + \boldsymbol{g}^T(\mathfrak{q}_{j+1}) \cdot \boldsymbol{\lambda}_{j+1}\right]. \quad (4.18)$$

The discrete action S_d approximating the continuous action S in Eq. (4.13) then reads

$$S_d = \sum_{j=0}^{N_t-1} \bar{L}_d(\mathfrak{q}_j, \boldsymbol{\lambda}_j, \mathfrak{q}_{j+1}, \boldsymbol{\lambda}_{j+1}). \quad (4.19)$$

Applying the discrete Hamilton principle leads to

$$\begin{aligned} \delta S_d = \sum_{j=0}^{N_t-1} \Bigg[& \frac{\partial L_d(\mathfrak{q}_j, \mathfrak{q}_{j+1})}{\partial \mathfrak{q}_j} \cdot \delta\mathfrak{q}_j + \frac{\partial L_d(\mathfrak{q}_j, \mathfrak{q}_{j+1})}{\partial \mathfrak{q}_{j+1}} \cdot \delta\mathfrak{q}_{j+1} \\ & \frac{h}{2}\frac{\partial \boldsymbol{g}(\mathfrak{q}_j)^T}{\partial \mathfrak{q}_j} \cdot \boldsymbol{\lambda}_j \cdot \delta\mathfrak{q}_j + \frac{h}{2}\frac{\partial \boldsymbol{g}(\mathfrak{q}_{j+1})^T}{\partial \mathfrak{q}_{j+1}} \cdot \boldsymbol{\lambda}_{j+1} \cdot \delta\mathfrak{q}_{j+1} \\ & \frac{h}{2}\boldsymbol{g}(\mathfrak{q}_j)^T \cdot \delta\boldsymbol{\lambda}_j + \frac{h}{2}\boldsymbol{g}(\mathfrak{q}_{j+1})^T \cdot \delta\boldsymbol{\lambda}_{j+1} \Bigg] = 0 \\ & \forall\ \delta\boldsymbol{\lambda}_j, \delta\mathfrak{q}_j \text{ and } \delta\mathfrak{q}_0 = \delta\mathfrak{q}_{N_t} = \mathbf{0}. \end{aligned} \quad (4.20)$$

Using a discrete version of the fundamental lemma of the calculus of variations and shifting the sum indices to factor out $\delta\mathfrak{q}_j$ and eliminate $\delta\mathfrak{q}_{j+1}$ leads to the discrete constrained Euler-Lagrange equations

$$\begin{aligned} D_1 L_d(\mathfrak{q}_j, \mathfrak{q}_{j+1}) + D_2 L_d(\mathfrak{q}_{j-1}, \mathfrak{q}_j) - \boldsymbol{G}_d^T(\mathfrak{q}_j) \cdot \boldsymbol{\lambda}_j = \mathbf{0} \quad &\in \mathbb{R}^{n_{\text{dof}}} \\ \boldsymbol{g}(\mathfrak{q}_{j+1}) = \mathbf{0} \quad &\in \mathbb{R}^{n_c}, \end{aligned} \tag{4.21}$$

where $D_i F(x_1, \ldots, x_i, \ldots, x_n) = \frac{\partial F(x_1, \ldots, x_i, \ldots, x_n)}{\partial x_i}$ is the differentiation of F with respect to its i^{th} variable and the discrete Jacobian of the constraints $\boldsymbol{G}_d(\mathfrak{q}_j) = h\frac{\partial \boldsymbol{g}(\mathfrak{q}_j)}{\partial \mathfrak{q}_j}$. Note that the constraint forces $-\boldsymbol{G}_d^T(\mathfrak{q}_j) \cdot \boldsymbol{\lambda}_j$ ensure that the constraints are fulfilled to numerical accuracy at the configuration $\mathfrak{q}_{j+1}$.

Including external forces So far the introduced theory is only valid for conservative systems. However, the model in this thesis includes Kelvin-Voigt viscous damping to improve the numerical stability, see Section 4.2.1 Eq. (4.33). This can be considered by including the viscous damping forces $\boldsymbol{f}_{\text{ext}} = \frac{\partial \mathbb{V}_{\text{visc}}}{\partial \dot{\mathfrak{q}}}$ via the Lagrange-d'Alembert principle, see [Wri08; Pen18; Hua22]. The same principle can also be used to consider external forces from, e. g., actuators. For the continuous case, the Lagrange-d'Alembert principle reads

$$\begin{aligned} &\delta \int_{t_0}^{t_{N_t}} \bar{L}(\mathfrak{q}, \dot{\mathfrak{q}}, \boldsymbol{\lambda}) \mathrm{d}t + \int_{t_0}^{t_{N_t}} \boldsymbol{f}_{\text{ext}}(\mathfrak{q}, \dot{\mathfrak{q}}) \cdot \delta\mathfrak{q} \mathrm{d}t = 0 \\ &\qquad \forall\ \delta\boldsymbol{\lambda} \text{ and } \forall\ \delta\mathfrak{q}\ :\ \delta\mathfrak{q}(t_0) = \delta\mathfrak{q}(t_{N_t}) = \mathbf{0}. \end{aligned} \tag{4.22}$$

From the variational principle, the constrained forced Euler-Lagrange equations follow

$$\begin{aligned} \frac{\partial L}{\partial \mathfrak{q}} - \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{\mathfrak{q}}}\right) - \boldsymbol{G}^T(\mathfrak{q}) \cdot \boldsymbol{\lambda} + \boldsymbol{f}_{\text{ext}}(\mathfrak{q}, \dot{\mathfrak{q}}) = \mathbf{0} \quad &\in \mathbb{R}^{n_{\text{dof}}} \\ \boldsymbol{g}(\mathfrak{q}) = \mathbf{0} \quad &\in \mathbb{R}^{n_c}. \end{aligned} \tag{4.23}$$

The inclusion of dissipative forces does not impede the good energy behaviour of variational integrators. The energy change in the system is physically motivated instead of being caused by numerical inaccuracies.
For the discrete setting, the influence of the external forces $\boldsymbol{f}_{\text{ext},j}^+$ and $\boldsymbol{f}_{\text{ext},j}^-$ at the time node t_j, in the time intervals $[t_{j-1}, t_j]$ and $[t_j, t_{j+1}]$, respectively, have to be taken into account. The virtual work is approximated with

$$\boldsymbol{f}_{\text{ext},j}^- \cdot \delta\mathfrak{q}_j + \boldsymbol{f}_{\text{ext},j}^+ \cdot \delta\mathfrak{q}_{j+1} \approx \int_{t_j}^{t_{j+1}} \boldsymbol{f}_{\text{ext}}(\mathfrak{q}, \dot{\mathfrak{q}}) \cdot \delta\mathfrak{q} \mathrm{d}t. \tag{4.24}$$

The position and velocities are approximated via the midpoint rule and finite differences, respectively

$$\begin{aligned} \boldsymbol{f}^{-}_{\text{ext},j} &= \frac{\Delta t}{2} \boldsymbol{f}_{\text{ext}} \left(\frac{\mathbb{q}_j + \mathbb{q}_{j+1}}{2}, \frac{\mathbb{q}_{j+1} - \mathbb{q}_j}{\Delta t} \right) \\ \boldsymbol{f}^{+}_{\text{ext},j} &= \frac{\Delta t}{2} \boldsymbol{f}_{\text{ext}} \left(\frac{\mathbb{q}_{j-1} + \mathbb{q}_j}{2}, \frac{\mathbb{q}_j - \mathbb{q}_{j-1}}{\Delta t} \right). \end{aligned} \tag{4.25}$$

With this, the constrained forced Euler-Lagrange equations read

$$\begin{aligned} D_1 L_d(\mathbb{q}_j, \mathbb{q}_{j+1}) + D_2 L_d(\mathbb{q}_{j-1}, \mathbb{q}_j) - \boldsymbol{G}_d^T(\mathbb{q}_j) \cdot \boldsymbol{\lambda}_j + \boldsymbol{f}^{+}_{\text{ext},j} + \boldsymbol{f}^{-}_{\text{ext},j} &= \mathbf{0} \\ \boldsymbol{g}(\mathbb{q}_{j+1}) &= \mathbf{0}. \end{aligned} \tag{4.26}$$

These non-linear equations can then be solved iteratively with a Newton-Raphson scheme for the unknowns $(\mathbb{q}_{j+1}, \boldsymbol{\lambda}_j)$, if $(\mathbb{q}_{j-1}, \mathbb{q}_j)$ are known.
To initialise the time-stepping scheme, the configurations $(\mathbb{q}_0, \mathbb{q}_1)$ are required. However, $(\mathbb{q}_0, \mathbb{q}_1)$ implicitly define the initial kinetic energy of the system, in dependency of the timestep. To explicitly define the initial velocities and ensure consistent initial conditions, the Legendre transformation can be used to define the conjugate momenta and calculate $\mathbb{q}_1$.

Legendre transformation Defining the conjugate momenta with the discrete Legendre transformation

$$\mathbb{p}_j^{-} = -D_1 L_d(\mathbb{q}_j, \mathbb{q}_{j+1}) + \frac{1}{2} \boldsymbol{G}_d^T(\mathbb{q}_j) \cdot \boldsymbol{\lambda}_j - \boldsymbol{f}^{-}_{\text{ext},j} \tag{4.27a}$$

$$\mathbb{p}_j^{+} = D_2 L_d(\mathbb{q}_{j-1}, \mathbb{q}_j) - \frac{1}{2} \boldsymbol{G}_d^T(\mathbb{q}_j) \cdot \boldsymbol{\lambda}_j + \boldsymbol{f}^{+}_{\text{ext},j}, \tag{4.27b}$$

allows for the constrained forced discrete Euler-Lagrange equations (4.26) to be rewritten as

$$\begin{aligned} -\mathbb{p}_j^{-} + \mathbb{p}_j^{+} &= \mathbf{0} \quad \in \mathbb{R}^{n_{\text{dof}}} \\ \boldsymbol{g}(\mathbb{q}_{j+1}) &= \mathbf{0} \quad \in \mathbb{R}^{n_c}. \end{aligned} \tag{4.28}$$

Then the equations can be reinterpreted as an enforcement of the equality between the left-sided momentum $\mathbb{p}_j^{-}$ and right-sided momentum $\mathbb{p}_j^{+}$ at each time node t_j see [Mar01b].

The value of $\mathbb{p}_0^{-}$ is defined by the system's initial momentum $\mathbb{p}_0$ with

$$\mathbb{p}_0 - \mathbb{p}_0^{-}(\mathbb{q}_0, \mathbb{q}_1) = \mathbf{0} \quad \text{and} \quad \boldsymbol{g}(\mathbb{q}_1) = \mathbf{0}. \tag{4.29}$$

From this $\mathbb{q}_1$ and $\boldsymbol{\lambda}_0$ can be calculated. Then the pair $(\mathbb{q}_0, \mathbb{q}_1)$ is known and $\mathbb{q}_2$ and $\boldsymbol{\lambda}_1$ and subsequent configurations and Lagrange multipliers can be calculated from Eq. (4.15). Using this, consistent initial conditions are

ensured in the simulation.

4.2 Modelling the human leg with a prosthetic foot

The leg is modelled using constrained rigid body dynamics [Bet05; Bet06; Ley08a], while the prosthesis is modelled with predeformed[7] geometrically exact beam theory [Ant95]. The equations of motion are derived via Lagrangian formalism and then solved via a variational integrator [Mar01b]. In general, dynamical systems can not be solved exactly and numerical methods are needed to approximate their solutions. However, not all numerical methods are equally accurate or reliable, and some may introduce errors or artefacts that distort the physical behaviour of the system. The main advantage of VIs is their excellent energy behaviour, due to their structure preserving properties. They approximate the solution of the Euler-Lagrange equations and a discrete version of Noether's theorem can be applied and holds true [Mar01b]. As mentioned in Section 3.3, a prosthesis that stores and releases energy more efficiently during the gait cycle is associated with improved gait in a patient. It is also difficult to measure in vivo, making it a perfect quantity to be examined in simulations. Thus, using an integrator in the simulation that has good energy behaviour is beneficial.

The following is quoted from my previous work [Sch22b] Section 2. Variable names in the text and figures have been changed to better fit this thesis. Figures and tables have been modified to suit the format of this thesis.
“
In this work, parameters based on [Cha75] are used for mass, inertia and size of the thigh and shank and are summarised in Table 4.1. The prosthetic foot is modelled after the Össur Vari-Flex®carbon spring prosthesis [VARI]. The prosthetic foot is rigidly attached to the shank, mimicking a passive prosthetic foot. This means that there is no actuator inputting energy into the prosthetic foot during gait, opposed to for instance the “Empower Prothesenfuss” from [EMPO], which actively controls the prosthesis during gait and can reduce metabolic cost for the wearer [Asi21; Her12]. However, such an active prosthesis requires external energy. Here, a passive prosthetic foot is examined, which only stores and releases energy provided by the patient.

4.2.1 The carbon spring prosthesis model

In this work, the passive prosthetic foot Össur Vari-Flex® from [VARI], shown in the top of Figure 4.1, is modelled in [Sch19; Soe19] with geometrically exact beam theory from [Ant95; Cri99].

[7] Sometimes also termed precurved geometrically exact beam theory in literature.

Table 4.1: Parameters for the thigh and shank used in this works simulations, from [Cha75].

	thigh	shank	unit
length	0.4582	0.3753	[m]
mass	6.5233	2.6857	[kg]
inertia J1	0.1137	0.0391	[kg/m^2]
inertia J2	0.1158	0.0393	[kg/m^2]
inertia J3	0.0225	0.0029	[kg/m^2]
centre of mass distance to proximal centroid	0.1779	0.1943	[m]

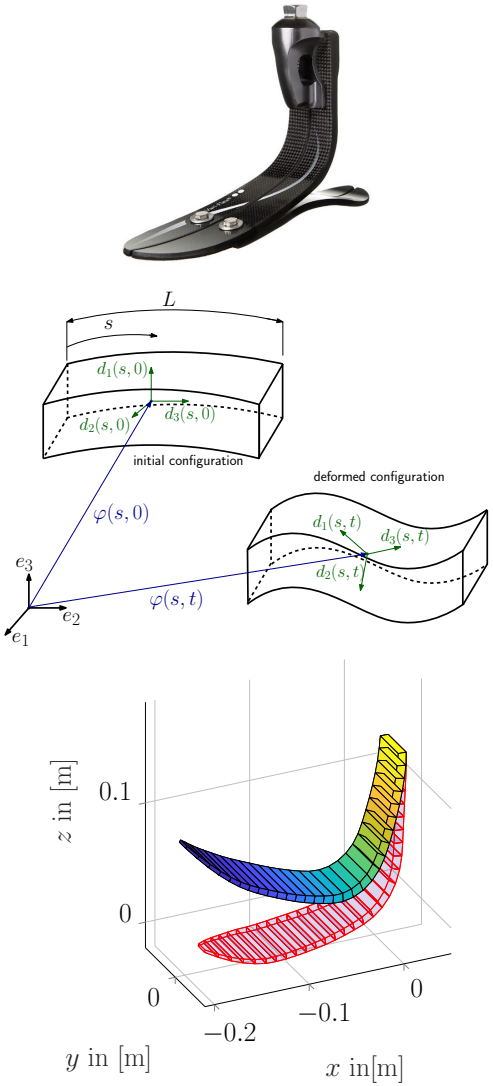


Figure 4.1: "The Össur Vari-Flex® (top) which is modelled with the geometrically exact beam theory (middle, based on [Cri99]) resulting in the high fidelity predeformed prosthesis model in Matlab (bottom) in the reference configuration (transparent) and in a deformed configuration (coloured by stress resultants). The image of the prosthesis is excluded from the copyright covering this work." Figure and caption quoted from [Sch22b].

This formulation allows for accurate representation of complex strain states in the prosthesis while still keeping the computational demand relatively low compared to full 3D finite element models.

[...]

The location of the centerline of the beam is denoted by $\boldsymbol{\varphi}(s,t) \in \mathbb{R}^3$, where s runs along the beam's length and t denotes time. The orientation of a given cross-section is represented by an orthonormal director triad $\boldsymbol{d}_1(s,t), \boldsymbol{d}_2(s,t), \boldsymbol{d}_3(s,t) \in \mathbb{R}^3$. Due to the orthonormality, the cross-sections of the beam are modelled as rigid sections, omitting warping and distortion effects in the model. The prosthesis' deformation energy is modelled with a St. Venant-Kirchhoff-type stored energy function [Lan11; Lan13] which describes an ideally elastic material behaviour.

$$\mathbb{V}_{int}(\boldsymbol{\Gamma}, \boldsymbol{K}) = \frac{1}{2} \int_0^L \left(\boldsymbol{\Gamma}^T \cdot \boldsymbol{D}^\Gamma \cdot \boldsymbol{\Gamma} + \boldsymbol{K}^T \cdot \boldsymbol{D}^K \cdot \boldsymbol{K} \right) \tag{4.30}$$

The material parameters and geometry of the beam are included in the two matrices $\boldsymbol{D}^\Gamma = \mathrm{diag}(GA, GA, EA)$ and $\boldsymbol{D}^K = \mathrm{diag}(EI_1, EI_2, GJ)$ consisting of the Young's modulus E, the shear modulus G and geometry specific parameters e. g. the cross-section area A, the area moments of inertia I_1, I_2 and the polar moment of inertia J. The values derived via reverse engineering in [Soe19] are summarised in Table 4.2.

Table 4.2: Isotropic material parameters for the prosthesis derived in part with reverse engineering in [Soe19] along with the assumed viscous damping parameter.

Young's modulus E	$59958e6$	$[N/m^2]$
Poisson's ration ν	0.265	
shear modulus G	$\frac{E}{2(1+\nu)} = 23.693e6$	$[N/m^2]$
density ρ	$1.71e3$	$[kg/m^3]$
viscous damping η	1.0	

The strain measures $\boldsymbol{\Gamma}(\boldsymbol{q})$ and $\boldsymbol{K}(\boldsymbol{q})$ quantify shear, elongation, flexion and torsion with

$$\Gamma_i = \boldsymbol{d}_i^T \cdot \frac{\mathrm{d}}{\mathrm{d}s}\boldsymbol{\varphi} - [\boldsymbol{d}_i^T \cdot \frac{\mathrm{d}}{\mathrm{d}s}\boldsymbol{\varphi}]\Big|_{t_0} \tag{4.31}$$

$$K_i = \frac{1}{2}\epsilon_{ijk}\left[\boldsymbol{d}_k^T \cdot \frac{\mathrm{d}}{\mathrm{d}s}\boldsymbol{d}_j - [\boldsymbol{d}_k^T \cdot \frac{\mathrm{d}}{\mathrm{d}s}\boldsymbol{d}_j]\Big|_{t_0}\right], \tag{4.32}$$

where $i, j, k = 1, 2, 3$. Figure 4.1 shows the prosthesis in the top, a geometrically exact beam with the centerline and directors in the middle, while the resulting model of the prosthesis from [Sch19; Soe19] in an undeformed[8] and a deformed state, using the predeformed geometrically exact beam formulation for the prosthesis geometry, is shown in the bottom.

To complete the model, viscous damping is included in the beam model. Following [Lan11; Lan13; Sch16], the effective material constitutive equations are extended by a viscous term, proportional to the material strain rates $\dot{\Gamma}_i, \dot{K}_i$, with viscosity parameters for extension η_E and shear η_G. Similar to the energy function above, the Kelvin-Voigt viscous dissipative potential $\mathbb{V}_{\text{visc}}$ is defined as

$$\mathbb{V}_{\text{visc}} = \frac{1}{2} \int_0^L \left(\dot{\boldsymbol{\Gamma}}^T \boldsymbol{D}^{\eta_1} \dot{\boldsymbol{\Gamma}} + \dot{\boldsymbol{K}}^T \boldsymbol{D}^{\eta_2} \dot{\boldsymbol{K}} \right) \tag{4.33}$$

where $\boldsymbol{D}^{\eta_1} = \text{diag}(\eta_G A_1, \eta_G A_2, \eta_E A)$ and $\boldsymbol{D}^{\eta_2} = \text{diag}(\eta_E I_1, \eta_E I_2, \eta_G J)$ combine material and geometric parameters, as for instance in [Lan13]. Since no information on the viscous damping parameters η_E, and η_G for the prosthesis are currently available, this work uses $\eta_E = \eta_G = 1.0$, to counteract high frequency vibrations inside the beam. This continuous formulation of the beam can be discretised using one-dimensional finite elements, see for instance [Zie06; Fis07], and solved numerically. In this work, the prosthesis is discretised with 20 nodes resulting in 19 one dimensional finite elements, striking a balance between accuracy and computational cost. The choice of 20 spatial nodes is based on a convergence study building on the work detailed in [Sch21b] for the conservative case. The viscous damping introduced to the model is included in the convergence study here and the results are shown in Figure 4.2. The prosthesis' highest total deformation energy W_{int} during a 1 second simulation in a gravitational field is used as a comparison value when varying the number of nodes n_{nodes} used to discretise the prosthesis. Figure 4.2 shows the convergence of the internal deformation energy error ϵ relative to a reference solution (computed on a very fine grid of $n_{max} = 250$ nodes), when increasing the number of nodes used to discretise the prosthesis. The error is calculated for each considered number of nodes n_{nodes}.

$$\epsilon = \frac{||\mathbb{V}_{int,n} - \mathbb{V}_{int,n_{max}}||}{||\mathbb{V}_{int,n_{max}}||} \tag{4.34}$$

Increasing the number of nodes leads to an increase in the forward dynamics computation time. Due to the impact the computation time of the forward dynamics has on the fuzzy forward dynamics simulation, 20 nodes are chosen as a balance between accuracy and computational cost in this work.

[8] Undeformed refers to the strain free initial state of the precurved prosthesis.

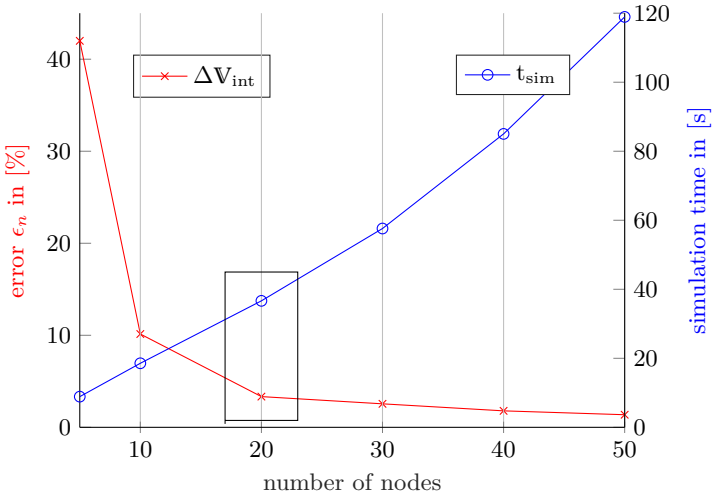


Figure 4.2: "Visualisation of the internal deformation energy error when varying the number of nodes in the model, based on [Sch21b] in order to find a compromise between the computational cost and accuracy of the prosthesis model. The compromise of 20 nodes is highlighted." Figure and caption quoted from [Sch22b].

This examination is based on the forward dynamics simulation, detailed in the next section. Together with the two rigid bodies, the choice of 20 nodes for the prosthesis' discretisation leads to a total of 264 degrees of freedom for the model, constrained by 132 internal constraints and 14 external constraints (three for the hip, five for the knee and six for the ankle joint).

4.2.2 Forward dynamics simulation

To examine the behaviour of the leg with the prosthesis before including parametric uncertainty, the model is placed in a gravitational field orientated in the negative z-direction and held in place at the hip with a spherical joint. The resulting motion of the simulation is a swinging motion of the human leg model with prosthetic foot, similar to the swing phase during the human gait cycle. A variational integrator is used to approximate the dynamics, derived from the discrete action principle for the discretised augmented Lagrangian, see [Ley08b]. All derivatives and gradients in this work are calculated automatically with CasADi [And19]. For the simulation shown in Figure 4.3, a timestep of 0.001s is used. Furthermore, the Young's modulus of the prosthe-

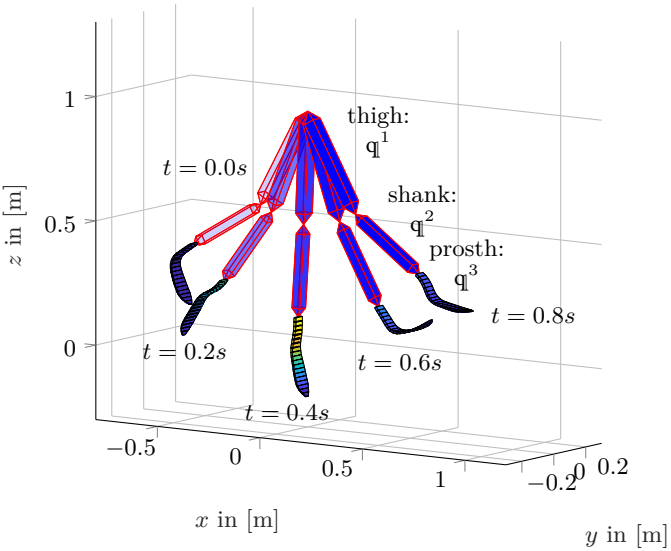


Figure 4.3: "Simulation results of the swing movement for the human leg with a predeformed geometrically exact beam model for the prosthetic foot." Figure and caption quoted from [Sch22b].

sis is reduced and it's density is increased, compared to carbon fibre laminate, in order to emphasise its deformation due to inertia. However, the simulation also runs stably with the original material parameters. As can be seen in Figure 4.4 the energy is exchanged between the potential, kinetic and internal deformation energy. Due to the variational integrator, which is structure preserving, the total energy of the system is not artificially reduced or increased by numerical integration. The viscous damping is present, however, it's influence is so small that it is not visible in the energy evolution plot.
"

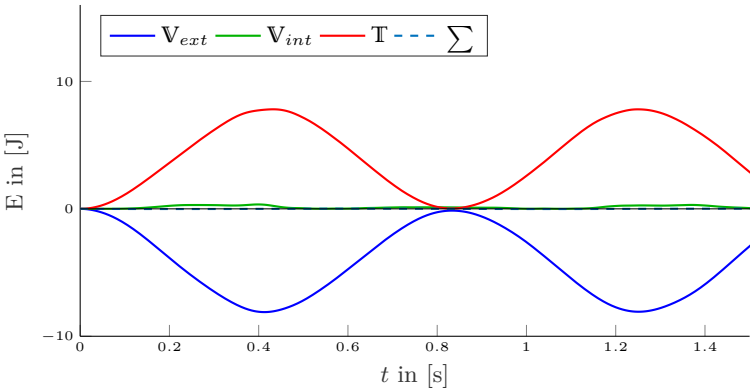


Figure 4.4: "Evolution of the different energy components over time for the swing movement simulation." Figure and caption quoted from [Sch22b].

4.2.3 Note on the finite element discretisation and numerical stability of the simulation

As mentioned in Section 4.2.1, the prosthesis model is discretised with the help of one-dimensional finite elements. The finite element method (FEM) is a numerical technique that is used to obtain approximate solutions of boundary value problems in engineering and mathematical modelling. The basic concept of the FEM is to divide the model into smaller and simpler subdomains called finite elements and then approximate the unknown field variable, such as displacement or pressure. For more details on the FEM method see [Hut04; Zie77; Red93; Log11]. The FEM involves three main steps, namely discretisation, formulation and solution. Discretisation is the process of dividing the domain into elements and nodes. Formulation is the process of deriving a system of algebraic equations that relate the nodal values of the field variable to the given data. Solution is the process of solving the system of equations to obtain the nodal values of the field variable, and then interpolating them within each element to obtain the approximate solution over the entire domain. The formulation step can be done in different ways, depending on the nature of the problem and the desired accuracy. Some common methods are the direct method, weighted residual method and Galerkin method. To achieve accurate results, the element shape and the interpolating shape functions have to be chosen carefully.

In the current case, the beam is discretised into n_{nodes} along the beam's

centreline and two nodes form one element. Then the internal deformation energy of the beam can be calculated based on the director triad and position of the nodes, see Section 4.2.1. Each node is associated with a position and orientation, in essence turning each node into a rigid cross-section and the deformation energy is quantified by two neighbouring nodes via their orientation and distance to each other.

While working with the model, it's numerical stability is very sensitive to parameter changes. Depending on the combination of timestep, material parameters and number of nodes used in the discretisation, the model would compute stably and fast, or require more time and in the worst-case, not compute at all. One possible reason for this sensitivity is the following condition.

The Courant–Friedrichs–Lewy (CFL) condition The Courant-Friedrichs-Lewy condition, first introduced by Courant, Friedrichs and Lewy in 1928 see [Cou28], is a necessary condition for the convergence and stability of numerical methods that solve certain partial differential equations (PDEs). It arises in the numerical analysis of explicit time integration schemes, when these are used for the numerical solution of PDEs that involve convection or wave phenomena. The CFL condition states that the numerical domain of dependence must contain the true domain of dependence, that is, the information required to determine the solution at a given point must be accessible by the numerical scheme. Mathematically, the CFL condition can be expressed as $C = \frac{ah}{\Delta x} \leq C_{max}$ where C is the CFL number, a is the wave speed in the medium, h is the timestep, Δx is the spatial step, and C_{max} is a constant that depends on the numerical scheme and the order of accuracy. The CFL condition ensures that the numerical solution does not introduce unrealistic oscillations or instabilities that are not present in the true solution. It also implies a trade-off between accuracy and efficiency. An example for the application of the CFL condition in computational fluid dynamics is shown in [Bla15]. For more general references on numerical methods and stability for PDEs see [Lev07].

4.2.4 Simulation scenarios

The following is quoted from my previous work [Sch22b] Section 3.5. Variable names in the text and figures have been changed to better fit this thesis. Figures and tables have been modified to suit the format of this thesis.
“

In the two exemplary scenarios analysed in this work, the leg moves under its weight due to being placed in a gravitational field oriented in the negative z-direction.

The first examined scenario is a pendulum swing of the multibody system, where the prosthesis deforms due to inertia, as was already used to check the models implementation with forward dynamics, shown as a timelapse in Figure 4.3. The pendulum movement is similar to the swing phase occurring during human gait (for more information on the biomechanics of the human leg and the human gait cycle see for instance [Whi14]). The parameters used for the simulation are summarised in Table 4.3.

Table 4.3: Prosthesis parameters and initial joint angles used for the fuzzy forward dynamics simulation of the swinging motion.

Young's modulus E	$59.95e4$	$[N/m^2]$
Poisson's ration ν	0.265	
shear modulus G	$\frac{E}{2(1+\nu)} = 23.69e4$	$[N/m^2]$
density ρ	$8.55e3$	$[kg/m^3]$
viscous damping η	1.0	
Δt	0.001	s
T	2	s
hip extension	$\frac{\pi}{6}$	rad
knee flexion	$\frac{\pi}{3}$	rad

The second scenario simulates the leg in a squatting position, that is the thigh is horizontal with the shank vertical. The prosthetic foot is held in place at two nodes. This is done by constraining two nodes to remain in their initial position. However, rotation of the cross-sections at these nodes is permitted. The prosthesis deforms under the leg's weight periodically. Figure 4.5 shows the initial configuration of the leg alongside a deformed configuration of the resulting movement. The deformation is due to the mass of the leg in a gravitational field. To emulate the process of realistic load on the prosthesis, the mass of the shank is increased to represent the influence of body weight, while the prosthesis uses the parameters derived with reverse engineering. The parameters used for the simulation are summarised in Table 4.4.
"

4.3 Considering the layered structure of the prosthesis

As mentioned in Section 4.2, the parameters used there are based on the results of the reverse engineering in [Soe19]. Due to the layered structure of the prosthesis, the values can not be used exactly. As can be seen in Figure 4.6, the prosthesis consists of three layers. The two faces consist of carbon fibre laminate, which are woven carbon fibre cloth in a resin matrix, visible on the CT-scan in Figure 4.6 from [Soe19; Blo23] as intersecting dark lines (car-

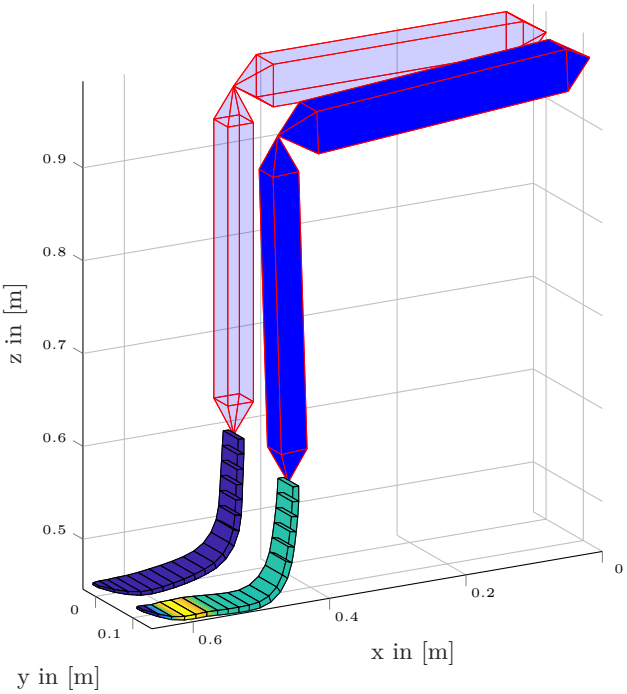


Figure 4.5: "The initial configuration of the leg with prosthetic foot for the second scenario is shown in the back, with a deformed configuration shown in the foreground." Figure and caption quoted from [Sch22b].

bon fibres) separated by lighter regions (resin). The faces are separated by a core layer, which has parallel lines running throughout in the CT-Scan. This points towards a uniform foam or resin with the lines resulting from the manufacturing process or another fibre resin layer, with parallel fibres. The core layer's material could not be determined exactly, since destructive methods would have been required. Due to the similar contrast of the faces and core in the CT-scan, the density of the layers is likely similar as well. To include the layered structure in the model, two approaches are possible. Firstly, each layer can be modelled as an individual geometrically exact beam with the re-

Table 4.4: Prosthesis parameters and initial joint angles used for the fuzzy forward dynamics simulation of the prosthesis under load.

Young's modulus E	$59.958e7$	$[N/m^2]$
Poisson's ration ν	0.265	
shear modulus G	$\frac{E}{2(1+\nu)} = 23.69e7$	$[N/m^2]$
density ρ	$8.55e3$	$[kg/m^3]$
viscous damping η	1.0	
Δt	0.001	s
T	2	s
hip flexion	$\frac{\pi}{2}$	rad
knee flexion	$\frac{\pi}{2}$	rad
shank weight	107	kg

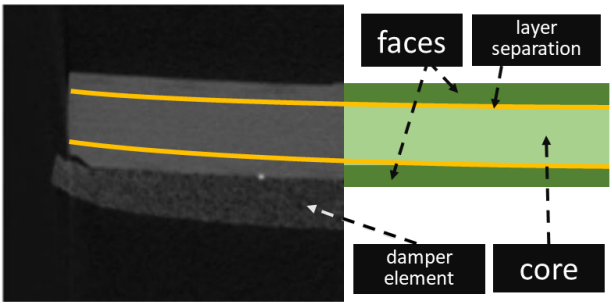


Figure 4.6: A close up of the layers of the prosthesis from the CT-scan, based on [Soe19; Blo23].

spective layer's geometric and material parameters. Secondly, the layers can be homogenised mathematically into a single layer, that represents the effect of the layered structure. While the first approach allows for more modelling accuracy, the gain in accuracy is likely not large enough when compared to the huge disadvantage that it incurs. Due to the fact that each layer is modelled as its own beam, the degrees of freedom of the model are also tripled in this case. Furthermore, the constraints modelling the interaction between layers further complicates the model adding to the computational cost of the simulation. Thus, to avoid this increase in computational cost, the homogenisation approach is chosen for this work. The following is based on the student thesis [Blo23]. In mathematical terms, the separable stiffness matrices in Eq. (4.35)

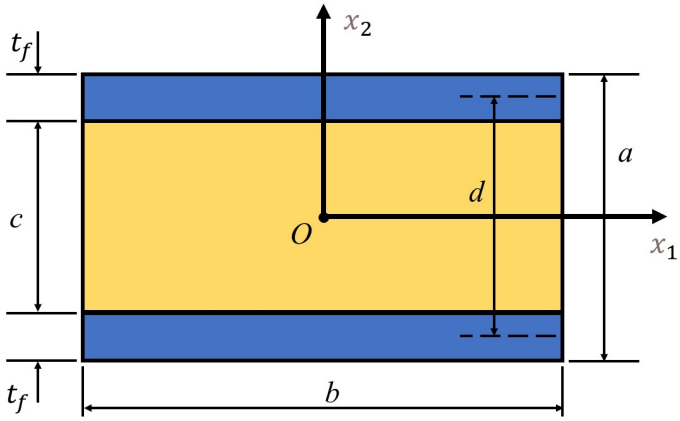


Figure 4.7: The dimensions required for the homogenisation approach, based on [Bîr13; Blo23].

$$\boldsymbol{D}^{\Gamma} = \begin{bmatrix} G\kappa_1 A & 0 & 0 \\ 0 & G\kappa_2 A & 0 \\ 0 & 0 & GA \end{bmatrix} \quad \text{and} \quad \boldsymbol{D}^{K} = \begin{bmatrix} EI_1 & 0 & 0 \\ 0 & EI_2 & 0 \\ 0 & 0 & GJ \end{bmatrix} \tag{4.35}$$

are replaced with the homogenised and no longer separable stiffness matrices in Eq. (4.36)

$$D_h^{\Gamma} = \begin{bmatrix} A_1 & 0 & 0 \\ 0 & A_2 & 0 \\ 0 & 0 & A_3 \end{bmatrix} \quad \text{and} \quad D_h^{K} = \begin{bmatrix} C_1 & 0 & 0 \\ 0 & C_2 & 0 \\ 0 & 0 & C_3 \end{bmatrix}. \tag{4.36}$$

Here, A_1, A_2 denote the effective shear stiffnesses, A_3 denotes the effective axial stiffness, C_1, C_2 denote the effective bending stiffnesses and C_3 denotes the effective torsional stiffness. The term separable refers to how in Eq. (4.35) can be separated into geometry parameters and material parameters, while in Eq. (4.36) this distinction is no longer visible. In [Blo23], two different homogenisation approaches are examined. It should be noted, that the homogenisation described here is only valid for a rectangular cross-section with two face layers and one core layer, which fits the prosthesis modelled in this work. The homogenisation approach assumes, that the face layers have the same thickness and consist of the same material. Additionally, each layer's material is assumed to be homogeneous and isotropic and the Poisson ratios

of all materials is assumed to be equal [Bîr13; Sta22]. The standard approach, denoted with "std", amounts to a summation over the individual layers, proportional to their respective cross sections. The faces, which in this case have the same dimensions and material parameters are denoted with f while the core is denoted with c. The stiffness parameters for the standard approach read

$$\begin{aligned} C_1^{\text{std}} &= 2E_f I_{f1} + E_c I_{c1} \\ C_2^{\text{std}} &= 2E_f I_{f2} + E_c I_{c2}, \\ C_3^{\text{std}} &= 2G_f J_f + G_c J_c, \\ A_1^{\text{std}} &= 2G_f \kappa_1 A_f + G_c \kappa_1 A_c, \\ A_2^{\text{std}} &= 2G_f \kappa_2 A_f + G_c \kappa_2 A_c, \\ A_3^{\text{std}} &= 2G_f A_f + G_c A_c \end{aligned} \tag{4.37}$$

While simple, this approach neglects the interaction between the layers, such as layer adhesion. It should be noted, that I_{fi} has to consider the distance from the centreline of the beam to the faces, due to the parallel axis theorem, see [Cli17; Kuy16].

In order to account for the interlayer effects, a coupled approach developed in [Mus53] and simplified for rectangular cross-sections with three layers in [Bîr13] is used here. According to [Bîr13], the relevant differences in stiffness occur in C_1, C_3 and A_2, which are the effective bending stiffness, the torsional rigidity and effective shear stiffness respectively. The dimensions of the layers are visualised in Figure 4.7. All values are based on the parameters determined with reverse engineering in [Soe19]. The layer thickness $t_f = 0.5\text{mm}$ is constant throughout the prosthesis, while the core layer thickness c varies resulting in the total prosthesis cross-section height $a = 2t_f + c$. The width of the prosthesis cross-section is denoted with b and also varies along the prosthesis' centreline. The stiffness parameters for the coupled approach, denoted with "coup", then read

$$C_1^{\text{coup}} = E_f \frac{b t_f d^2}{2} + E_f \frac{b t_f^3}{6} + E_c \frac{b c^3}{12} \tag{4.38}$$

$$C_3^{\text{coup}} = \frac{b^3}{3} \left[(cG_c + 2t_f G_f) - \frac{192b}{\pi^5} \sum_{n=0}^{\inf} \frac{C_{(n)}}{(2n+1)^5} \right] \tag{4.39}$$

$$A_2^{\text{coup}} = \kappa \frac{12b}{a^2} \frac{cG_c + 2t_f G_f}{c\rho_c + 2t_f \rho_f + c(\rho_f - \rho_c) F(\frac{\pi c}{2a})} \left(\rho_f \frac{t_f d^2}{2} + \rho_f \frac{t_f^3}{6} + p_c \frac{c^3}{12} \right). \tag{4.40}$$

The term $C_{(n)}$ and function $F(x)$ are defined in [Bîr13] as

$$C_{(n)} = \frac{\left[G_c(G_c\cosh(mt_f)\tanh\left(\frac{mc}{2}\right) + G_f\sinh(mt_f)) \right.}{G_c\cosh(m_t f) + G_f\sinh(mt_f)\tanh\left(\frac{mc}{2}\right)} \\ \left. + 2G_f(G_c - G_f)(1-\cosh(mt_f))\tanh\left(\frac{mc}{2}\right)\right], \quad (4.41)$$

and

$$F(x) = \frac{\cos(x)}{x}\ln\left(\frac{1+\sin(x)}{\cos(x)}\right) \quad \text{for} \quad x \neq 0,\ F(0) = 1. \quad (4.42)$$

It is noted in [Bîr13], that the series in Eq. (4.39) converges very rapidly and that $F(x)$ is continuous.

4.4 Possible model improvements

The model introduced here can be improved in a few ways. So far, only forward dynamics are considered. To completely simulate the human gait cycle, the model would have to be transferred to an optimal control problem, which considered things like ground contact opening and closing and control torques and activation, see for instance [Koc17]. Furthermore, the shank and thigh are currently modelled as complete and healthy bodies, as they are in a healthy human being. While this is a valid assumption for this work, in general, a prosthesis is not directly attached to the shank, but connects to a shaft, that then connects to what remains of the patient's shank. This is currently not considered in this work, but could be modelled in future works by changing the shank's parameters. Similarly, the data for the thigh and shank model's mass and moments of inertia are based on a singular source. More expansive modelling in this regard would further improve the model.

Chapter 5

Applications

This chapter demonstrates the use of the previously introduced algorithms and methods to propagate the uncertainty through the developed model of the human leg with a prosthetic foot. The application of the Graph Follower algorithm to examine the effect of marker position errors with the proposed error model is also shown.

5.1 Validation of the code

Before looking at the different applications of the improved Graph Follower algorithm and newly developed FRV-GFA, it should briefly be mentioned how the code was verified. In this case, verification means to ensure that the code implementation of the algorithm does what is intended. The Graph Follower algorithm has been verified in [Eis19a; Eis19b]. The improvements made in this work, are in turn examined in [Sch22b], including the difference between the modifications introduced during this thesis and the original Graph Follower algorithm. The behaviour of the model can be validated in different ways. The two cases examined here have similarities to classical mechanical models. The swing case is similar to a double pendulum and the stance case has similarities with a cantilever beam. To check the implementation of the prosthesis and the underlying geometrically exact beam model, different simulations were performed and checked for plausibility and compared to analytical solutions where possible. With the individual parts of the algorithm validated, different applications are possible and described in the following sections.

5.2 Fuzzy forward dynamics of the prosthesis

The first uncertainty propagations were performed with the improved Graph Follower algorithm and the leg and prosthesis model without homogenisation.

5.2.1 Emulation of the swing phase

The following is quoted from my previous work [Sch22b] Section 3.6 & 3.7. Variable names in the text and figures have been changed to better fit this thesis. Figures and tables have been modified to suit the format of this thesis. “

In the simulation of the leg's swing movement, the chosen target output f is the total deformation energy stored in the prosthesis at a certain time t_j.

$$f = \sum_{el=1}^{n_{el}} W_{int,el}\left(\boldsymbol{\Gamma}(\mathfrak{q}(\tilde{p}, t_n), \tilde{p}, t_j), \boldsymbol{K}(\mathfrak{q}(\tilde{p}, t_j), \tilde{p}, t_j)\right) \tag{5.1}$$

The total deformation energy stored and released during gait in the prosthetic foot is related to the burden of gait for a patient, see for instance [Wat99]. This means it is directly related to the walking comfort of the patient with the prosthetic foot and is therefore an important measure for the prosthesis' design and it's acceptance by the patient. Predicting the prosthesis' behaviour with respect to the target output is important during the design process. However, predicting the movement of a double pendulum is difficult, given its chaotic nature. Adding a flexible body with uncertain stiffness to the end of the double pendulum further complicates the prediction. Including fuzzy uncertainty and a target output that depends on both the behaviour of the multibody model and the fuzzy uncertainty increases the need for algorithmic computation of the desired target output. Figure 5.1 shows the calculated trajectories for the minimal (0.9 E_{det}), deterministic (E_{det}) and maximum (1.1 E_{det}) values of the prosthesis' Young's modulus. The Graph Follower algorithm is used to calculate the target output envelopes. Analysing the uncertainty propagation allows for an efficient calculation of the correlation between the parametric input uncertainty, in this case the Young's modulus, and the target output, the stored internal deformation energy of the prosthesis. The results are shown in Figure 5.2. The target output envelopes for the different alpha levels are shown alongside the deterministic simulation's result. The $\alpha = 1$-level envelope matches the deterministic solution, as expected. The total deformation energy of the prosthesis oscillates which is represented well by the variational integrator used in the forward dynamics which avoids artificial energy loss due to numerical integration. As shown in the examination of the forward dynamics of this model, the energy drain of the viscous damping is negligible for the chosen parameters. While the input uncertainty is triangular and symmetric, the output envelopes shown in Figure 5.2 do not inherit these properties.

The limitations of the Graph Follower algorithm can be seen at $t = 1.4s$ when the upper envelope from $\alpha = 0.0$-level intersects the envelope from $\alpha = 0.5$-level.

However, when considering the entire evolution of the envelopes, the inac-

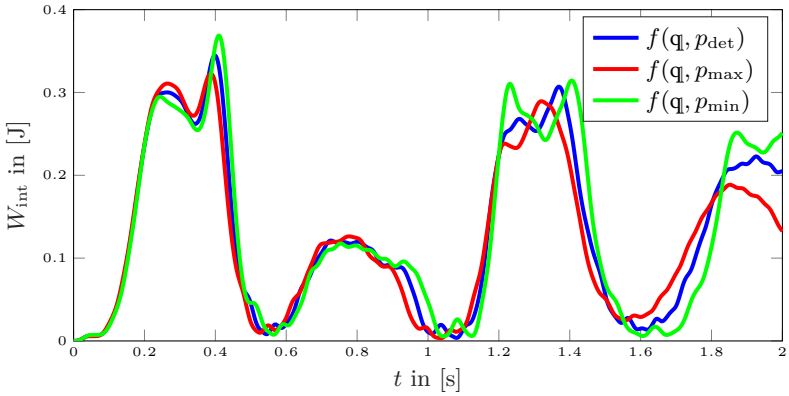


Figure 5.1: "Evolution of the target output calculated for the minimal, deterministic and maximum values for the Young's modulus." Figure and caption quoted from [Sch22b].

curacy is very localised to parts of the target output where it varies strongly in a short amount of time. If the main interest is in the envelopes magnitude and not the membership function of the target output, the results can be further improved, by combining all calculated α-levels envelopes and calculating the resulting combined envelope, shown in Figure 5.3. In this figure, only the highest and lowest values for the target output are considered alongside the deterministic solution. This omits any knowledge about the membership function and is thus no longer a fuzzy number[9]. However, this can be used to check conditions set for the design of the prosthesis.

5.2.2 Emulation of the prosthesis during a squatting exercise

The second target output analyses the maximal local deformation energy over all finite elements during the simulation at a certain time t_j. This is related to the structural integrity of the prosthesis during a load scenario.

$$f = \max_{n_{el}}(W_{int,el}(\mathfrak{q}, t_j)) \tag{5.2}$$

The calculations use the squatting pose configuration of the model and more realistic parameters for the material of the prosthesis. Simulations were

[9] While details for shape of the membership function is discarded, the result can be seen as a triangular fuzzy number, since the extremal target output trajectories correspond to the alpha-level 0 cut of the input parameter.

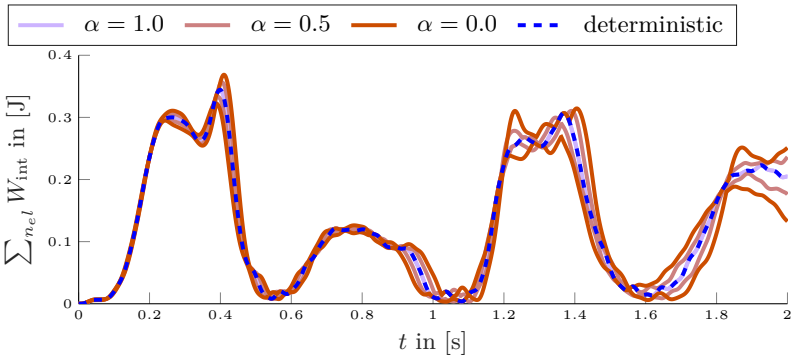


Figure 5.2: "Visualisation of the upper and lower envelope during the swinging motion of the human leg with the prosthesis for the target output function being the total stored energy within the prosthesis. The deterministic solution is shown with a dashed line, while the envelopes for the different α-levels are shown as full lines." Figure and caption quoted from [Sch22b].

performed with realistic parameters. However, due to the prosthesis being lightweight and stiff the Courant-Friedrichs-Lewy-condition [Cou28] requires the simulation to run with a very small timestep which leads to very computationally intensive forward dynamics simulations. The load applied to the prosthesis emulates the load of the body during a squat exercise. The envelopes of the target function are shown in Figure 5.4. As expected, the prosthesis acts as a spring under the load of the shank. The leg moves up and down periodically, while the uncertain Young's modulus affects both the amplitude and the period of the oscillation. Since the amplitude and frequency of the oscillating movement of the leg is directly related to the prosthesis' deformation, the oscillating movement can also be seen in the resulting envelopes for the chosen target function. Artefacts from the simplifications necessary for the Graph Follower algorithm to run with the high number of degrees of freedom of the model are present, but can be reduced by increasing computational power.
"

5.3 Examining multiple fuzzy parameters

The following is quoted from my previous work [Sch22b] Section 4. Variable names in the text and figures have been changed to better fit this thesis. Figures and tables have been modified to suit the format of this thesis.

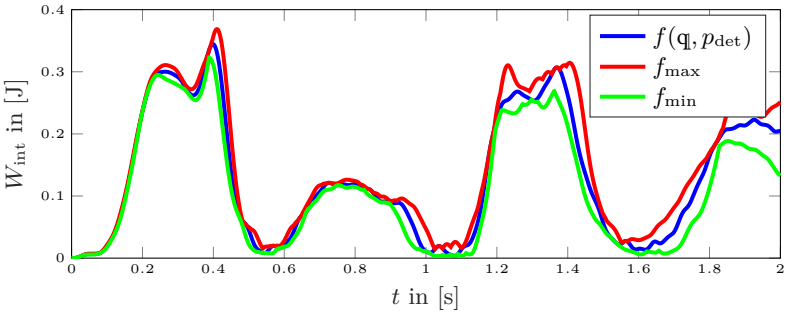


Figure 5.3: “The minimal and maximum envelopes for total internal deformation energy in the prosthesis.” Figure and caption quoted from [Sch22b].

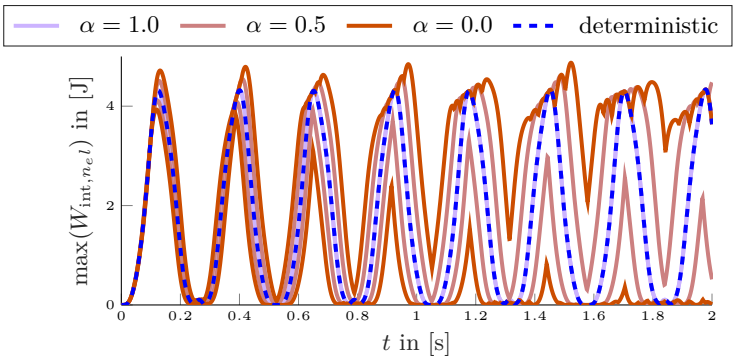


Figure 5.4: “Upper and lower hulls for the α-level optimisation approximation with the Graph Follower algorithm for the α-levels 1 (blue dashed) 0.5 (light brown) and 0 (dark brown) with the maximum local deformation energy as targeted output.” Figure and caption quoted from [Sch22b].

“

After demonstrating the application of the Graph Follower algorithm on the complex multibody system of the human leg with a geometrically exact beam prosthesis with a single fuzzy uncertain parameter, the next step is to examine the structure with multiple uncertain parameters. For this, two aspects are of interest. How does the method perform with regards to the increased number of optimisation parameters and which parameters can be affected with uncertainty.

5.3.1 Numerical efficiency of Graph Follower algorithm with multiple uncertain parameters

Here a brief evaluation of the Graph Follower algorithm with respect to its numerical efficiency is discussed. A comprehensive evaluation of the Graph Follower algorithm and comparison to other methods can be found in [Eis19b]. There, it is shown that the Graph Follower algorithm is suitable for propagating multiple fuzzy parameters since its computing time scales much less with the amount of uncertain parameters than the benchmark algorithm *Qua.Si.III*. To compare the modified Graph Follower algorithm used in this work, the runtime for four simulations are compared, one with a single fuzzy parameter and one with 11 fuzzy parameters, each for one and two seconds with 1000 and 2000 timsteps respectively. The resulting runtimes are shown in Table 5.1.

Table 5.1: Runtime of the simulations. The relative change is displayed on the outside, with the increased complexity runtime divided by the simpler simulation runtime.

Runtime	$N_t = 1000$	$N_t = 2000$	
$N_p = 1$	$3.8551e5$	$6.1432e5$	1.59
$N_p = 11$	$4.0819e5$	$5.4863e5$	1.34
	1.05	0.89	

This shows, that an increase in the amount of uncertain parameters does not affect the computation time of the simulation strongly. This also means, that focus should be put on reducing the computational time of the forward dynamics simulation, to reduce the computational cost when using the Graph Follower algorithm to propagate fuzzy uncertainty.

5.3.2 Choice of fuzzy parameters

The main limiting factor when choosing which parameters to model as uncertain, is the requirement, that the parameters are independent of each other and without interaction [Möl04]. Interdependence or interaction between fuzzy numbers can negatively affect the α-level optimisation by introducing unwanted uncertainty. Possible uncertain parameters are the leg's segment length, the segment's mass and moments of inertia, the segment's initial orientation and the prosthesis material parameters, namely Young's modulus, density and Poisson's ratio. Here, affecting the segment length or initial position with uncertainty would lead to fuzzy configurations of the leg. However, the configuration is not independent, since it has to satisfy the internal constraints of the rigid bodies of the segments and the joint constraints and can therefore not be affected by uncertainty without further investigation. The

prosthesis' Young's modulus, Poisson's ratio and density parameters can safely be assumed to be independent and are affected with uncertainty. For the leg, its mass is affected with uncertainty, including the moments of inertia. These values are not independent of each other, however the exact distribution of the mass along the leg is unknown for an individual and difficult to measure in vivo, which allows for an independent treatment in this work. This leads to a simulation with a total of 11 uncertain parameters. All fuzzy parameters are discretised with the same α-level cuts and for comparability also have the same parameters and interval of $p_{\text{det}} \pm 0.1 p_{\text{det}}$ as the squatting simulation.

5.3.3 Results

As in the previous simulation the movement is an oscillation. With the increased number of fuzzy parameters, the period and magnitude of the oscillations differ more than in the simulation with only the fuzzy Young's modulus leading to larger envelopes, shown in Figure 5.5. Due to the multiple uncertain

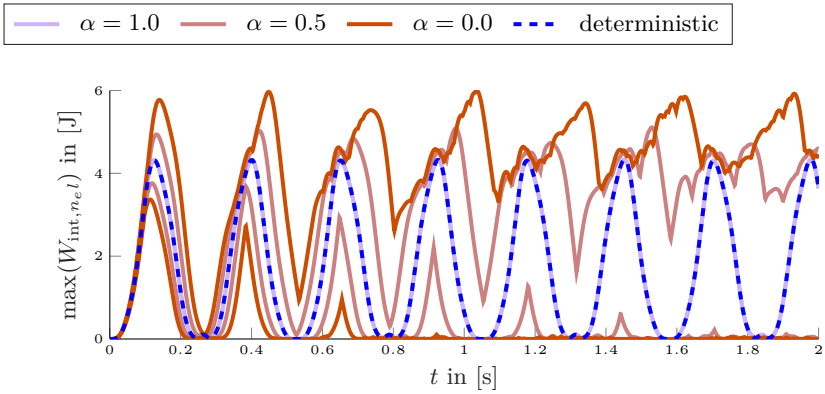


Figure 5.5: "Upper and lower hulls for the α-level optimisation approximation with the Graph Follower algorithm for the α-levels 1 (blue dashed) 0.5 (light brown) and 0 (dark brown) with the maximum local deformation energy as targeted output and 11 uncertain parameters." Figure and caption quoted from [Sch22b].

parameters, visualisation of the parameter distribution and their correlation to the envelopes is no longer trivial. Figure 5.6 visualises the parameter distribution for the $\alpha = 0$ level with the individual trajectories calculated from the corresponding parameter set and the resulting upper and lower envelope. The left side shows the relative difference with respect to the upper (green)

and lower (red) bounds for the parameters with the deterministic parameters in the middle (blue). As can be seen, all of the parameters vary and lead to individually extreme target outputs for a given [time node], shown on the right of Figure 5.6. It can also be seen, that for instance the Poisson's ratio (ν) leading to the envelope varies a lot less than for instance the mass of the thigh segment ($L_{1,M}$), giving an indication of which parameters have a strong effect on the target output. Also of note, the moments of inertia for the z-direction ($L_{i,T3}$) do not have an effect on the forward dynamics, since the movement has no rotation around the z-axis. This means, that these parameters can

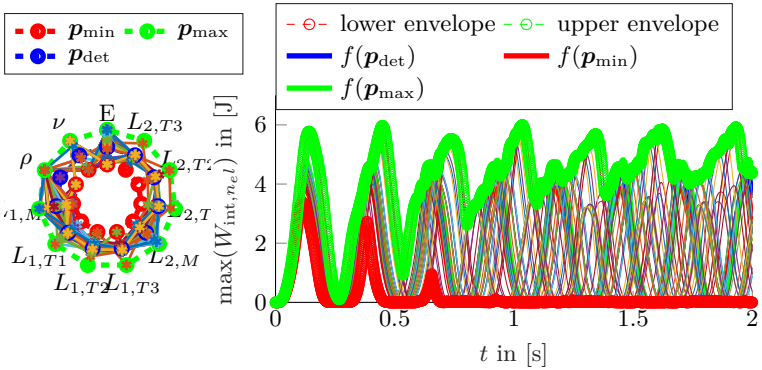


Figure 5.6: "Visualisation of individual parameter sets $\tilde{p}_{\alpha_k,i}$ (left) from which the individual target outputs $f_{d,\alpha k,i}$ (right) are calculated via forward dynamics. The upper and lower hulls for the α-level k are then calculated based on all of these individual trajectories. The 11 fuzzy parameters are Young's modulus (E), Poisson's ration (ν), density (ρ), thigh mass ($L_{1,M}$), thigh moments of inertia ($L_{1,T1-3}$), shank mass ($L_{2,M}$), shank moments of inertia ($L_{2,T1-3}$)" Figure and caption quoted from [Sch22b].

vary without affecting the target output. This results in most trajectories of the simulation shown in Figure 5.6 being based on the deterministic value.
"

5.4 Fuzzy simulation of the homogenisation procedure

As mentioned in Section 4.3, the prosthesis has three layers that are considered in the more advanced model examined here. Before including this model in the leg model, the effects of uncertain parameters on the homogenisation is examined separately. The Young's moduli of the face and core are modelled with fuzzy numbers and additionally the thickness of the face is assumed to

be uncertain in height but constant along the prosthesis, following the reverse engineering results from [Soe19]. It should be noted, that the effect of the uncertainty on the stiffness matrices (Eq. (4.36)) cannot be examined directly with the proposed methods, since the methods require a scalar target function. However, it is possible to examine the uncertainty's effects on the prosthesis model and use the previously introduced target functions. As be-

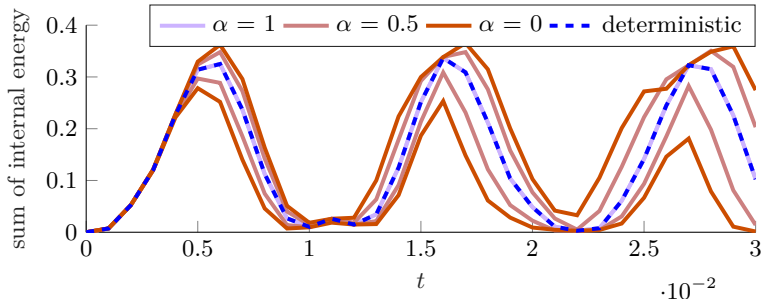


Figure 5.7: Hulls for the total internal deformation energy of the prosthesis when considering homogenisation with uncertain parameters.

fore, two target outputs are considered, namely the position of the last node, see Figure 5.7 and the total deformation energy, see Figure 5.8. The prosthe-

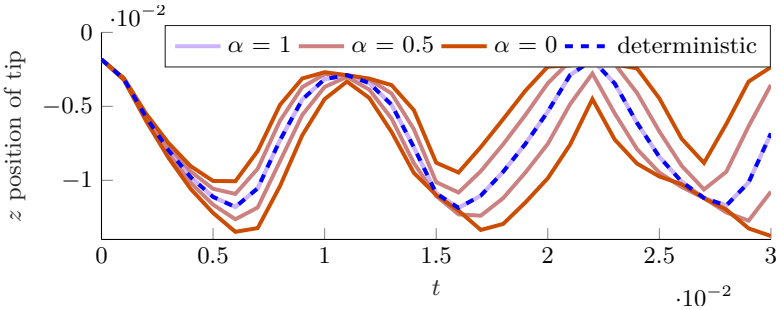


Figure 5.8: Hulls for the position of the last node of the prosthesis when considering homogenisation with uncertain parameters.

sis is fixed at the ankle and deforms due to a load applied to the tip of the prosthesis. The uncertain parameters affect the stiffness of the prosthesis, as described in Section 4.3 by modifying the stiffness matrix. However, since the Graph Follower algorithm requires a scalar quantity for the optimisation, so

the matrices cannot be the target output. Either a single entry in the stiffness matrix can be observed or the target output functions previously defined can be used. Since from an engineering and design perspective, the resulting behaviour of the prosthesis is of interest, the target output for movement and energy was used.

5.5 Fuzzy joint angle calculations

The following is quoted from my previous work [Sch24b] Chapter 3 & 4. Variable names in the text and figures have been changed to better fit this thesis. Figures and tables have been modified to suit the format of this thesis.
“

The epistemic marker position uncertainty is propagated through the joint angel calculations for the flexion angle of the three joints (hip, knee, ankle) of the left human leg during normal gait. All results are based on the same measurement and on the values shown in Table 3.1 of the largest possible marker deviations considered here. The results in Figures 5.10, 5.11 and 5.9(c) show the envelope of possible joint angle evolutions.

Effect of the individual uncertainty sources on joint angles

Measurement system errors, marker (mis)placement and soft tissue artefacts from Table 3.1, are examined for their effect on the flexion angles individually. The results are shown in Figure 5.9. As expected, the joint angle envelopes get larger if the considered maximum marker deviation increases from (a) to (c).

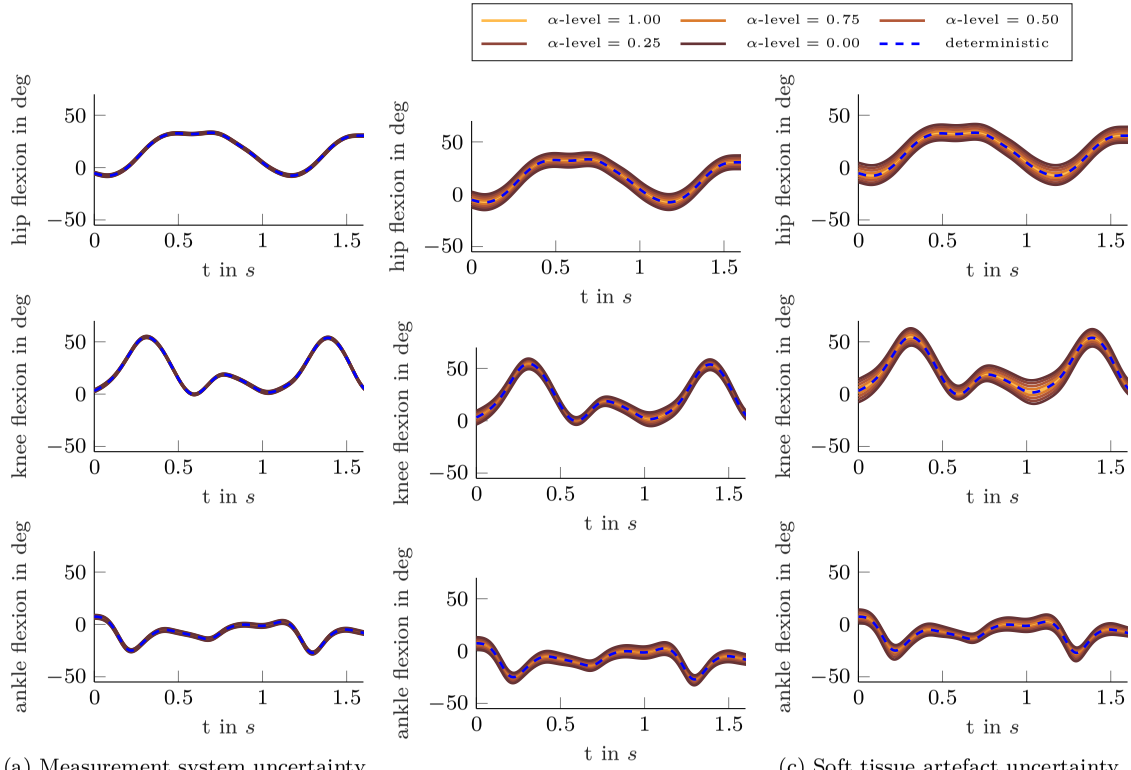
Worst case scenario – cumulative uncertainty

In the worst case, all three considered uncertainty sources can be present at the same time and add up to an even larger displacement in the marker positions. The resulting envelopes are shown in Figure 5.10. The worst possible joint angle error can be quantified by calculating a relative error for all α-levels.

This relative error $\Delta\Phi$ relates the maximum difference between the upper joint angle envelope, denoted with $\Phi^u_{\alpha_k}(t)$ for an α-level α_k, and the lower joint angle envelope, denoted with $\Phi^l_{\alpha_k}(t)$, to the maximum measured joint angle without uncertainty $\Phi(t)$.

$$\Delta\Phi_{\alpha_k} = \frac{\max\limits_{t\in[0,T]} \left(\Phi^u_{\alpha_k}(t) - \Phi^l_{\alpha_k}(t)\right)}{\max\limits_{t\in[0,T]} \left(\Phi(t)\right)} \tag{5.3}$$

The resulting relative errors of the flexion angle for all joints are summarised



(a) Measurement system uncertainty.

(b) Marker placement uncertainty.

(c) Soft tissue artefact uncertainty.

Figure 5.9: "Fuzzy envelopes for flexion angles of all three joints of the human leg, namely hip (top), knee (middle) and ankle (bottom) for the three considered uncertainty sources, namely measurement system (a), marker placement (b) and soft tissue artefacts (c)." Figure and caption quoted from [Sch24b].

in Table 5.2. The uncertainty of the joint angle, quantified by the differences in the envelopes, can be very large throughout the entire gait cycle, especially for joint angles with a small maximum value.

Table 5.2: The maximum joint angle envelope difference (Δ) relative to the largest measured joint angle over time for the various joints for every considered α-level. All values are in %. The relative error can be large for small angles.

	α_k	0.75	0.50	0.25	0.00
Δhip flexion		27	54	81	107
Δknee flexion		17	35	51	68
Δankle flexion		96	199	303	425

Comparing different methods of modelling the markers' position error

For comparison and plausibility check of results from the new method, two other methods of propagating the uncertainty in the marker positions, namely specific calculations of different marker deviations and a calculation with random marker deviations, are briefly examined. Figure 5.11(a) shows three trajectories of the knee flexion angle, calculated with no deviation (blue) from the measured positions and the largest positive (red) and largest negative (green) deviations. The effect of random noise on all markers is shown in Figure 5.11(b) and the result of the new method with epistemic uncertainty and the Graph Follower algorithm is presented in Figure 5.11(c). All three methods use the same spherical error model introduced in this work and the same largest allowed deviation in the marker positions from Table 3.1 and are applied to the same measurement data. When comparing Figure 5.11(a) to Figure 5.11(b), it is visible that assuming a fixed error does not accurately depict possible errors in the joint angle calculation. While Figure 5.11(b) already provides a more accurate representation of the effect of marker position errors on the joint angle, it is difficult to draw conclusions about the level of accuracy due to the randomness involved. In contrast, the proposed method (Figure 5.11(c)) clearly shows the effect of errors on the joint angle calculations while at the same time providing a correlation between the marker position error's magnitude and the resulting error in the joint angle.

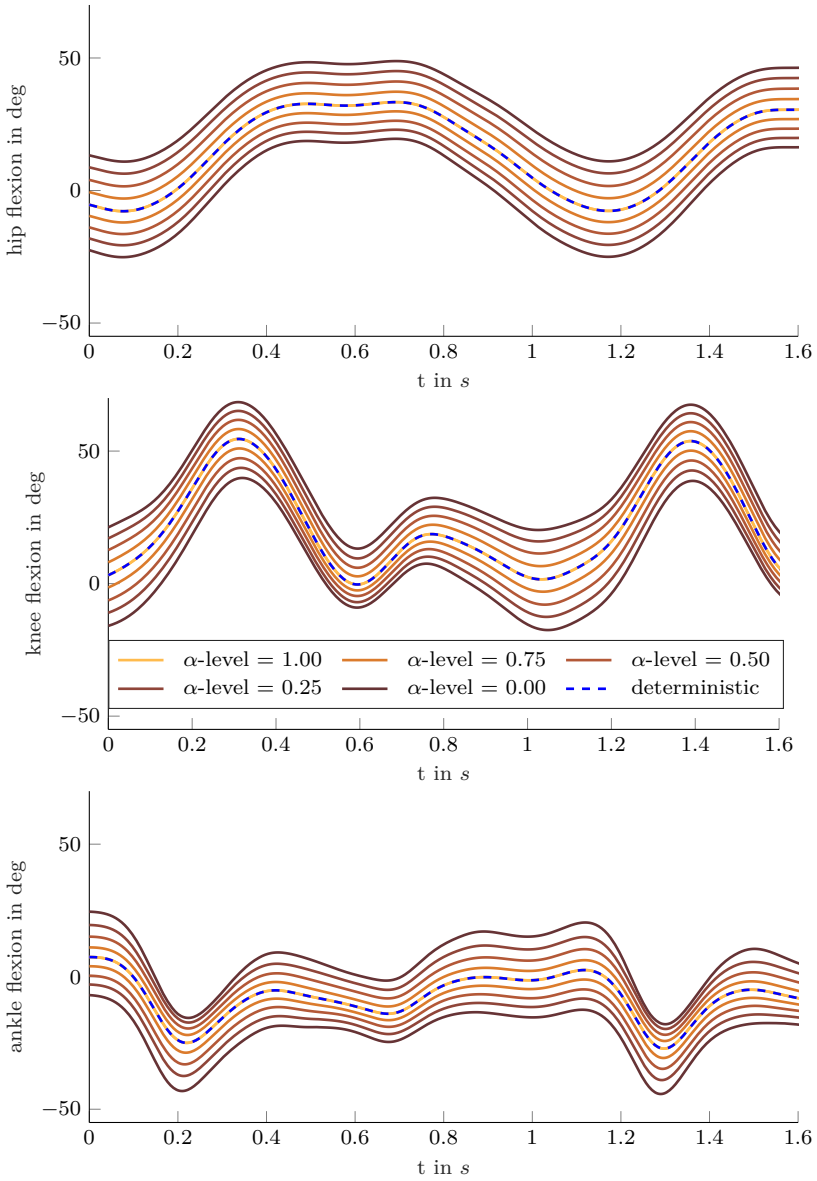


Figure 5.10: "The resulting envelopes when considering all error sources cumulatively." Figure and caption quoted from [Sch24b].

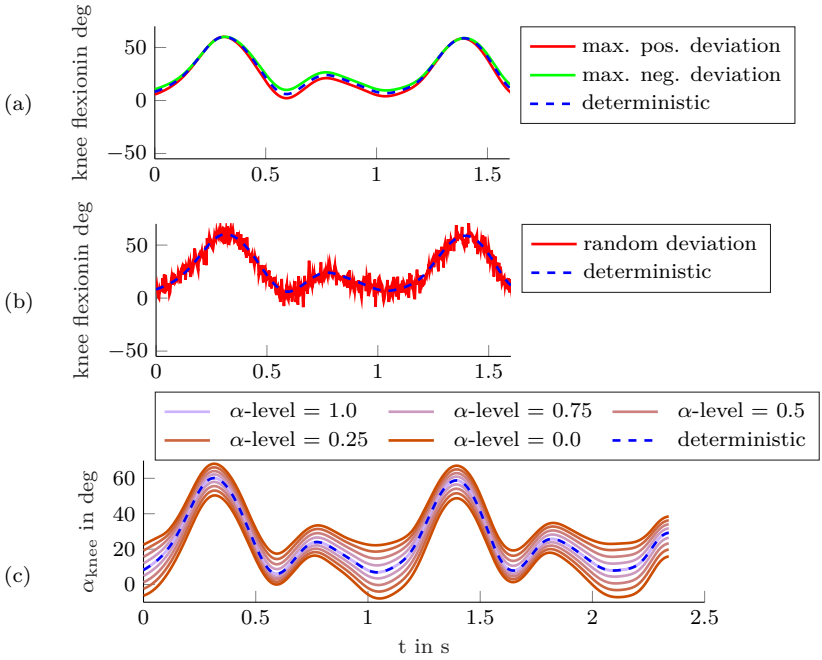


Figure 5.11: "Comparison of three methods to consider uncertainty in the marker positions for knee flexion, namely calculating different specific cases, using random noise and the calculation of the target output envelopes as introduced in this work. (a) knee flexion calculated for no deviation and the maximum positive and negative deviation. (b) knee flexion calculated with the markers having random deviations. (c) the resulting envelopes from α-level optimisation. All three examples use the same maximum allowed marker deviations, while each marker's deviation is modelled individually." Figure and caption quoted from [Sch24b].

5.5.1 Discussion

This work[10] examines the effect of epistemic uncertainty in the marker position data from optical motion capture for gait analysis. We introduce a new modelling method for marker position errors and a new method to analyse the effects of these errors on subsequent joint angle calculations. As mentioned in the introduction, 3D marker based optical motion capture is often used to benchmark other sensors without considering that the data gathered by the optical system is affected by errors in the reported position of the markers. Therefore, an estimation of how these errors affect subsequent calculations and the possible validations of other sensors is crucial and methods to model these are required. Previous research has examined the effects of errors on subsequent motion capture data [Uch22; Mye15; Cam17], but epistemic uncertainty and the propagation of an error model as introduced in this work has not been previously performed even though it allows for a more detailed examination of the effects of marker position errors. Propagating the uncertainty in the markers' positions with the Graph Follower algorithm (Figure 5.11(c)), has the advantage of not only showing the worst possible cases over the whole measurement period, but simultaneously provides more granular information on the possible joint angle errors with respect to the marker deviations. Furthermore, this process also provides the reverse information by showing the acceptable marker deviation, to still achieve a desired level of accuracy for the joint angle. In comparison, just calculating three different trajectories (Figure 5.11(a)) does not necessarily represent the possible worst cases. While using random noise to model the uncertainty gives a better impression of the possible joint angle error (Figure 5.11(b)), many calculations would be necessary to allow any kind of reliable prediction of what the worst joint angle at a given timeframe could be, which is a common attribute for stochastic approaches. In contrast, fuzzy uncertainty is suitable for the type of data uncertainty present in optical motion capture and efficiently calculates the possible joint angle errors for multiple marker position deviations up to the worst case. This bidirectional functionality of the suggested model and method allows for better examination of marker position errors and how they affect subsequent calculations. This contributes to the development of more sophisticated methods of reducing errors during the measurements which will benefit any subsequent calculations and diagnoses.

Furthermore, the results show and confirm, that even with small errors in the marker positions, large deviations of the joint angle results may occur. Due to the low computational effort required for this method and modelling approach, it is especially well suited for clinical applications, allowing for an evaluation of the joint angle during gait, alongside confidence intervals based on possible errors during the measurement. This allows for better evalua-

[10] This refers to the publication [Sch24b].

tions, without increasing the measurement or processing effort. Of course, this method can be applied to other joint angles or body segments and movements. Since the confidence in any subsequent diagnoses or treatment suggestions for a patient, that are based on such a gait analysis, is crucial, the provided details on errors and their effect will improve the information available for making these decisions and suggestions.

An additional benefit of the method used here to propagate the uncertainty through the joint angle calculations is, that it enables extracting the deviations of individual markers that lead to the respective maximum joint angle for any given time point of the measurement. This means that for any given timeframe, the marker deviations that led to the extreme joint angle for a given α-level can be obtained from the algorithm's results. It should be noted, that the deviations that lead to the largest deviation of the joint angle are not constant and are not necessarily on the boundary of the error intervals. This means the largest possible error in the joint angle, can already occur at much smaller deviations in the markers' positions, than are assumed from literature values. For instance the KNE_l marker only needs to deviate by 10 mm of the possible 32 mm for the largest joint angle error to occur for the given timeframe. This is also visible from the envelopes. The error interval for a given timeframe of the envelopes does not stay constant throughout the movement and is not symmetric. Including this knowledge in measurements could improve confidence in the measurement and diagnoses, since errors are explicitly considered without noticeably increasing the postprocessing effort.
”

Figure 5.12 shows the Euclidean norm of the marker's displacement that leads to the maximum joint angle error. It can be seen, that in some cases, as for instance with $||p_{\alpha,KNE_l}||$ for the upper envelope, the displacement does not need to equal the maximum possible deviation based on the known error sources from Table 3.1 to lead to a maximum joint angle error.

5.5.2 Conclusion and outlook

The following is quoted from my previous work [Sch24b] Section 5. Variable names in the text and figures have been changed to better fit this thesis. Figures and tables have been modified to suit the format of this thesis.
“

[...] The new approach is used to examine the effect of deviations in the marker positions based on known error sources and are quantified from literature values. The proposed method allows for an easy worst-case examination of calculated joint angles based on assumed errors in the marker positions. Furthermore, the opposite can also be examined with this method. If the worst possible error in the joint angles has to be limited, the largest allowable error in marker positions can be derived. This provides additional information

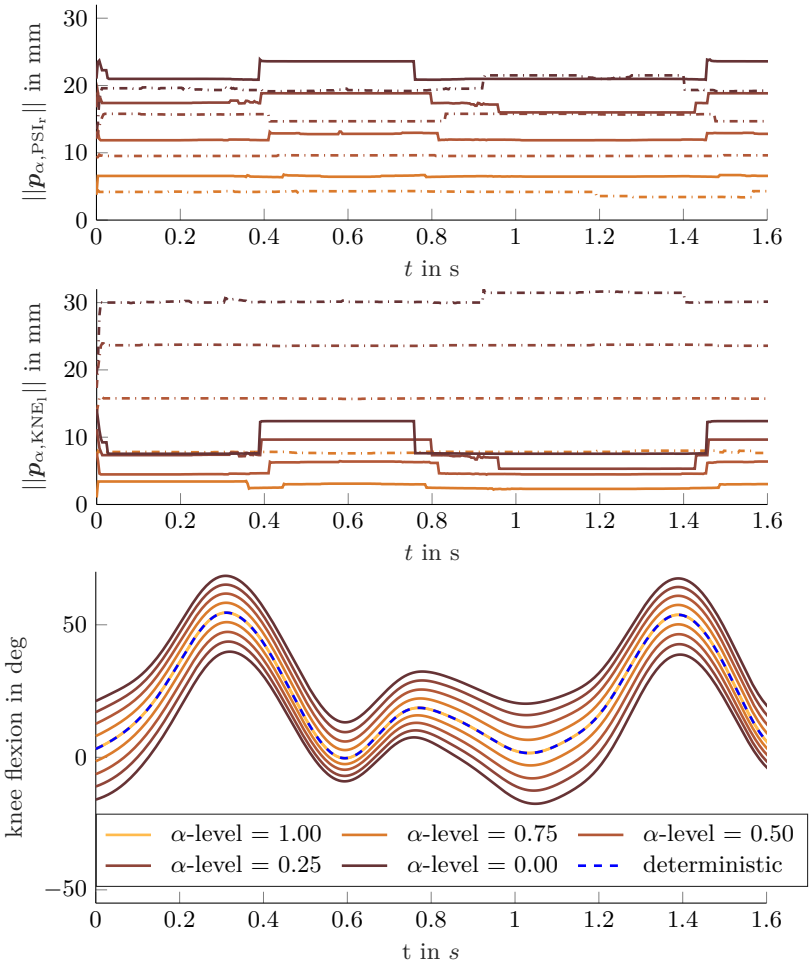


Figure 5.12: Example of the Euclidean norm of individual marker deviations for two markers (PSI (top) and KNE (middle), see Figure 3.8), during left knee flexion (bottom). The full line shows the deviations resulting in the upper envelopes and the dash-dotted line shows the deviations for the lower envelopes. The colours correspond to the respective α-levels.

for decisions that are based on motion capture data and subsequent calculation.

The method introduced here could also be used to investigate the sensitivity of subsequent calculations to marker models or to evaluate the validity of postprocessing optimisation procedures for motion capture. These optimisation methods aim to compensate for uncertainty introduced by the motion capture. However, the objective comparison of different compensation methods remains difficult. As shown in [Eis19a] the Graph Follower algorithm can be applied within optimisation problems which could allow for better evaluation of error reducing methods during gait analysis in the future. The new model for marker position errors during motion capture introduced in this work can also be expanded by basing the deviation of the markers on different, unsymmetrical shapes and the limb's motion and acceleration.
"

5.6 Failure probabilities from epistemic uncertainty analysis

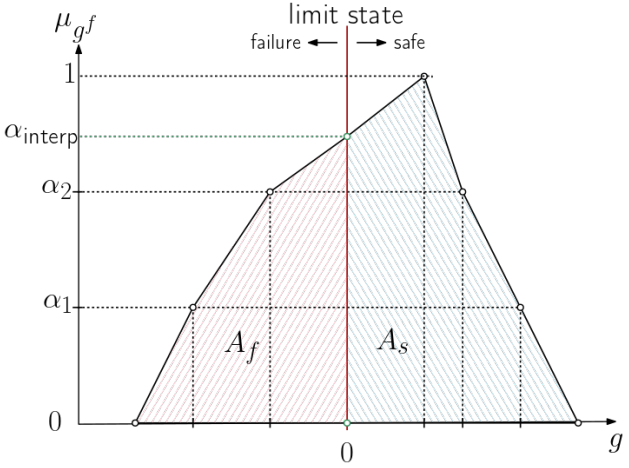


Figure 5.13: Visualisation of the calculation of a probability from a fuzzy number, based on [Sch24c].

To facilitate decisions, it may be helpful to reduce the uncertainty to a single value, for instance a failure probability v. However, as mentioned in Section 2.1 epistemic uncertainty is not a probability but a possibility. Thus, a method to transform the possibility into probability is required. One method, used for comparisons between methods in the joint work [Sch24c] based on [Rez20] is to relate the areas (or for higher dimensional cases the volume) underneath certain parts of the fuzzy number, as shown in Figure 5.13. To do

this, a limit state function $g(\tilde{f}(p))$ has to be defined for which, as an example, a value larger than zero signifies failure while a value smaller or equal to zero signifies the safe operation of the system under examination. This limit state, based on the target output function of the uncertainty analysis with fuzzy numbers, then divides the area under the target output into a failure area A_f and a safe area A_s, as shown in Figure 5.13. It should be noted, that it may be necessary to interpolate the membership function value of the limit state. The failure probability can then be calculated with

$$v = \frac{A_f}{A_f + A_s}. \tag{5.4}$$

As mentioned in [Sch24c], the modelling of the input uncertainty with fuzzy numbers greatly affects the resulting failure probability. Using the same bounds as for a probabilistic approach results in greatly overestimated failure probabilities, due to the discretisation and optimisation involved in the epistemic analysis. This highlights the importance of considering the combined aspects of uncertainty modelling, uncertainty propagation and the subsequent conclusions as a whole, and not as independent parts.

5.7 Polymorphic simulation of the human leg with a prosthetic foot

To demonstrate the function of the FRV-GFA, see Section 2.7.2, it is applied to the sitting configuration of the model of the human leg with a prosthesis, detailed in Section 4.2.

5.7.1 Simulation of the sitting configuration with polymorphic uncertainty

As before, this case reflects the prosthesis' behaviour under load. The Young's modulus of the prosthesis is affected with polymorphic uncertainty. Table 5.3 summarises the parameters for the sample generation. The fuzzy number samples of the fuzzy random variable are generated based on Eqs. (2.20) with the values from Table 5.3. The prosthesis' Young's modulus is affected with uncertainty. The $l_\alpha r_\alpha$-increments for the FRV are generated according to Eq. (2.28) and Eq. (2.29) with a normal distribution $N(1, 1)$, see Eq. (2.3), for $\Delta p_{\alpha=1}$ and $\Delta p_{l/r,\alpha<1}$. Convexity is ensured by redrawing deltas that are negative, resulting in the truncated normal distribution visible in Figure 5.14. The mean for the generation of the deltas are the same for all α-levels with $\Delta p_1 = \Delta p_{l,\alpha<1} = \Delta p_{r,\alpha<1} = 11.922e9$ [N/m^2]. A total of $N_s = 10000$ samples are generated and $N_{p\text{-disc}} = 100$ points are used for the grouping. The resulting minimum Young's modulus in the samples is $p_{\min} = 29.422e9$ [N/m^2]

and the maximum is $p_{\max} = 97.288e9$ [N/m^2]. The model's parameters are summarised in Table 5.3.

Table 5.3: Prosthesis parameters and initial joint angles used for the fuzzy forward dynamics simulation of the prosthesis under load with polymorphic uncertainty.

Young's modulus E	$59.958e9$	[N/m^2]	
Poisson's ration ν	0.265		
shear modulus G	$\frac{E}{2(1+\nu)} = 23.69e9$	[N/m^2]	
density ρ	$8.55e3$	[kg/m^3]	
viscous damping η	1.0		
Δt	0.001	s	
T	2	s	
hip flexion	$\frac{\pi}{2}$	rad	
knee flexion	$\frac{\pi}{2}$	rad	
shank weight	107	kg	

As mentioned in Section 2.7, the first step is to generate samples of the FRV, shown in Figure 5.14. The top plot shows 100 of the input parameter samples, while the plots below show the distributions generating the deltas for the different α-levels. The next step is grouping the samples. Figure 5.15 shows 100 of the total of 10000 generated samples, before grouping and after grouping, if the interval between largest and smallest drawn values for the Young's modulus is discretised with 100 points. The advantage of this grouping is that the number of necessary optimisations (twice per interval between two discretisation points) is no longer tied to the number of samples, see Section 2.7. From the figures it is evident, that while greatly reducing the computational cost required, the variability in the samples introduced from the randomness is still well preserved. After the grouping, all possible combinations for the intervals are created and checked if at least one sample requires the interval. Any intervals that do not show up in at least one sample are discarded. Then, the optimisation is performed for all intervals and the resulting hulls are stored in a dictionary. Finally, all samples are assigned their hulls from the dictionary based on the sample's interval values, which is the same result as if the Graph Follower algorithm had been applied to that sample individually. To reassemble the FRV for the target output, a PDF is fitted to the results for every α-level, either to the target output samples or the respective deltas. Figure 5.16 shows the resulting FRV for a single time node, when only examining the target output sample distribution. As can be seen, the normal distribution fits the data well and the uncertainty of the input parameter is clearly propagated through the model. However, since the

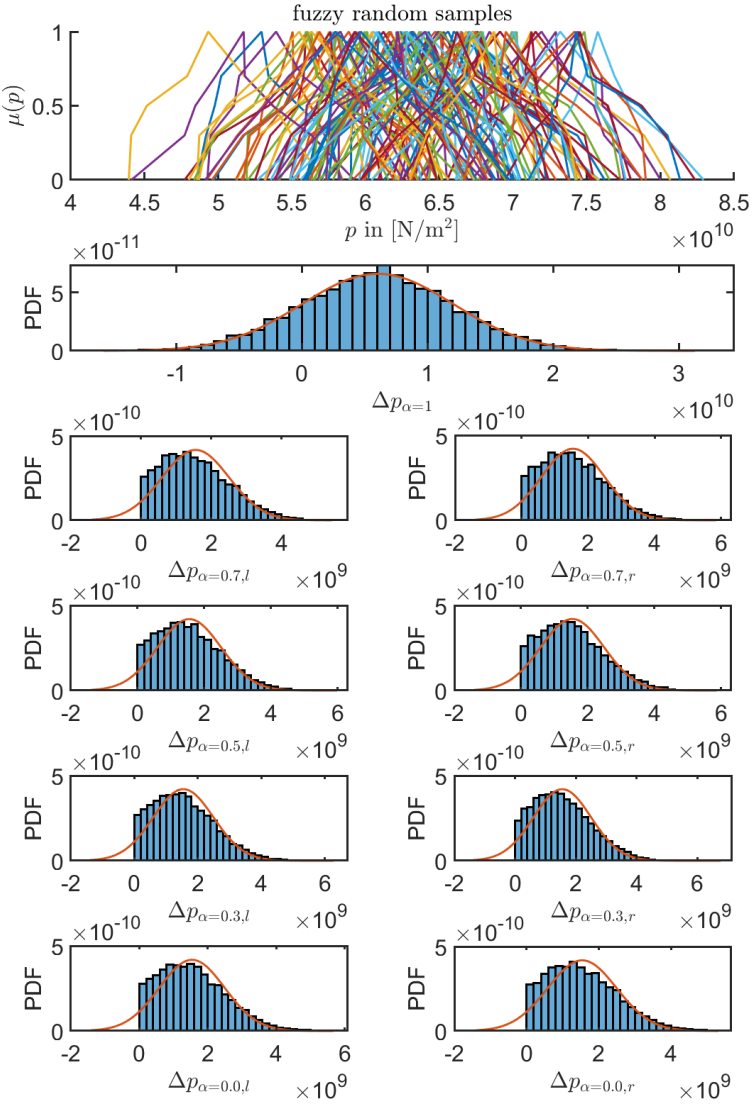


Figure 5.14: 100 of the input samples for the uncertain Young's modulus alongside the PDFs for the $l_\alpha r_\alpha$-increments.

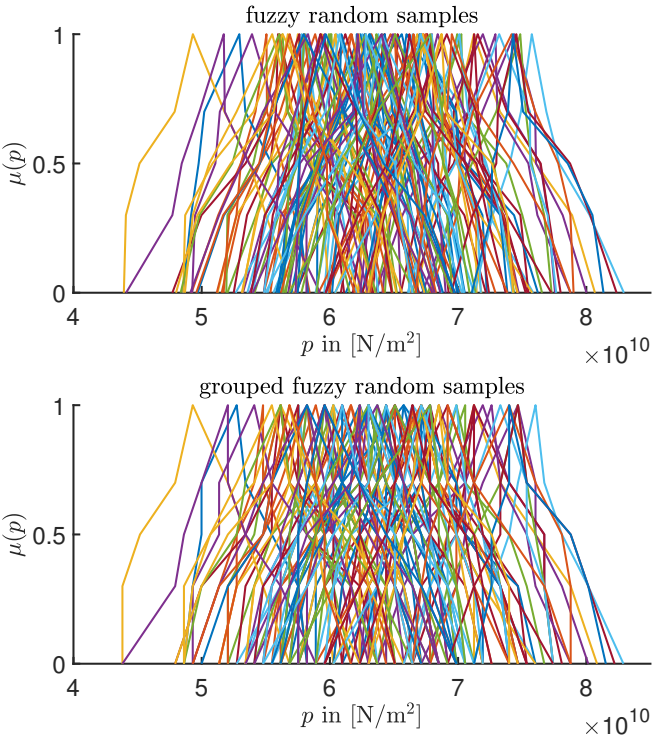


Figure 5.15: Showing 100 of the generated samples for the uncertain Young's modulus before and after grouping.

FRV is defined by the deltas, see Eq. (2.26), Figure 5.17 shows the same time node when fitting the PDF to the deltas. Due to the grouping preformed in sample generation, the deltas do not vary strongly. This distorts the fitting process and the normal distribution is no longer suitable as a distribution. This should be considered for further uncertainty analysis based on the results of this uncertainty propagation.

The goal of the uncertainty propagation is to calculate the target output FRV for every time node. Due to the multidimensional nature of the resulting time series of FRVs, visualisation is not trivial. A common way to visualise stochastic data is to show the mean value μ along side one or multiple steps

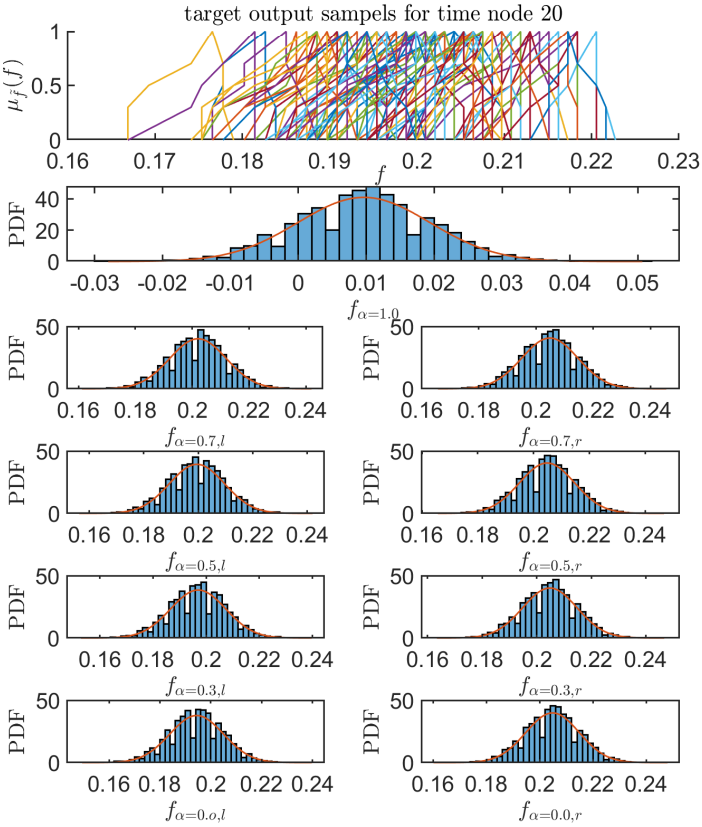


Figure 5.16: Resulting PDF for the target output samples for every α-level for time node 20 when fitting a normal distribution to the target output values different α-levels.

based on the standard deviation σ. Figure 5.18 shows the time series of the target output every 25 time nodes. All time nodes were calculated, but more time nodes would drastically reduce readability. Similarly, only 10 samples are shown per time node, since more would again reduce readability. This highlights one problem when dealing with uncertainty. The multidimensional nature of the resulting data and, especially with aleatoric uncertainty, the high number of samples make it difficult to visualise the results. To highlight the expected oscillating motion, the mean of the target output is shown in

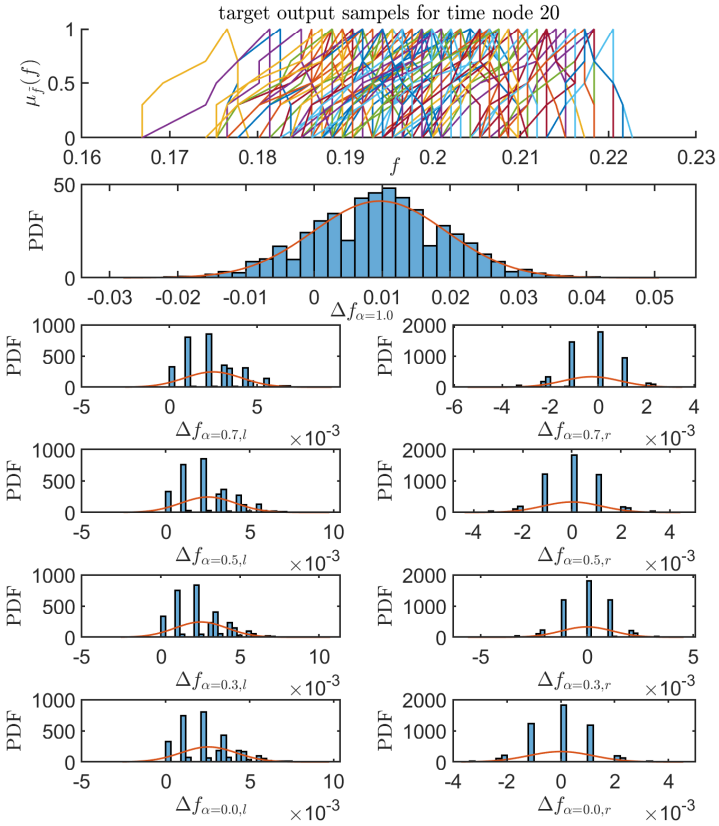


Figure 5.17: Resulting PDF for the target output samples for every α-level for time node 20 when fitting a normal distribution to the deltas of the target output for each α-levels.

blue for $\alpha = 1$. Figure 5.19 shows another alternative to visualise the target output FRV. Here, two images are shown of the mean and standard deviation for the FRV's deltas. The red line indicates the mean for each considered α-level. The left image shows the standard deviation of the distribution for the left deltas, while the right image shows the standard deviation of the distribution for the right deltas, each shaded from the mean to a maximum of one standard deviation.

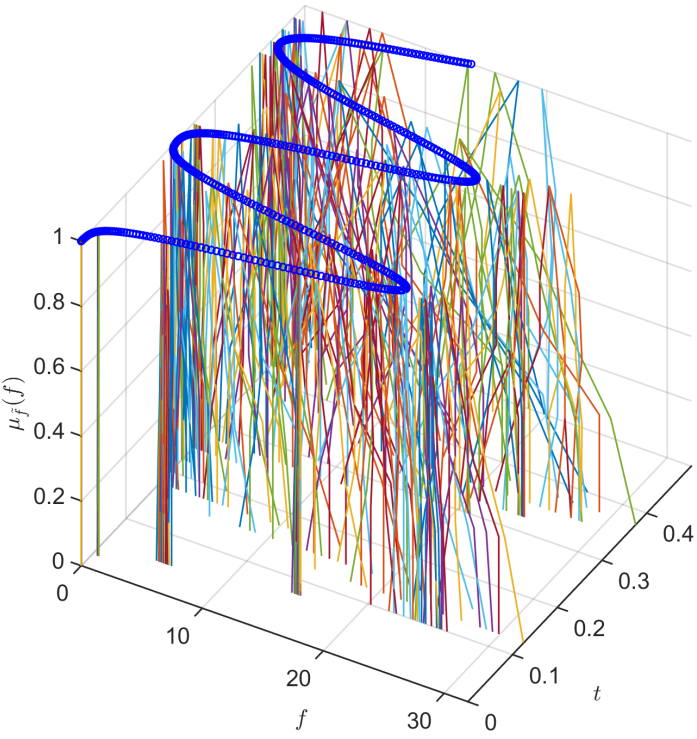


Figure 5.18: Visualisation of the temporal evolution for some samples of the target output. The mean of the target output for $\alpha = 1$ is shown in blue.

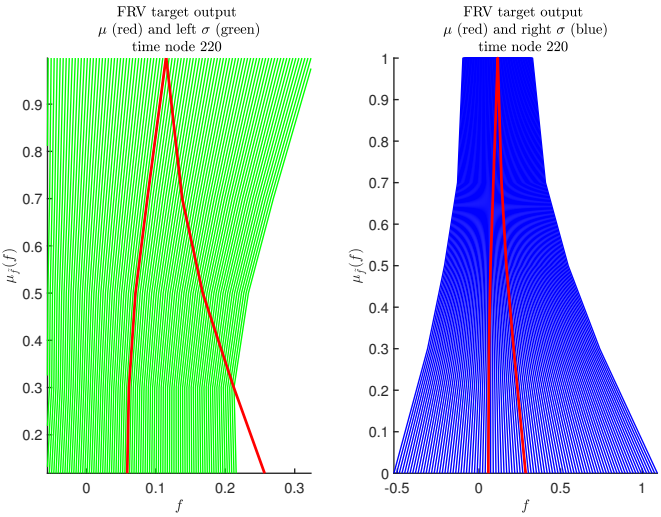


Figure 5.19: Visualisation of the fuzzy random target output for time node 220. The red line shows the mean μ of the FRV for each α-level. The left image shows the standard deviation σ (green) for the left delta distribution and the right shows the standard deviation σ (blue) for the right delta distribution. The colour's opacity transitions from full colour at a standard deviation of 0 to almost complete transparency at a standard deviation of 1.

5.7.2 Interpretation

The results show the expected motion. For the squatting scenario, the model behaves similarly to a cantilever beam, deflecting and returning to its origin position due to the conservation of energy. The propagation of the parameter uncertainty in the form of FRVs is successful, however the visualisation of the result is not trivial. Some examples have been proposed, but depending on the future use of the results, some visualisations are more helpful than others. For example, if the result of these simulations is to be used for further uncertainty simulations, visualising the FRV of the target output for a certain time node as in Figure 5.16 or 5.17 provides the most information. On the other hand, if the temporal evolution is of interest, Figure 5.18 may be more useful. As can be seen in the results shown in Section 5.7.1, due to the convexity condition and the required redraw (see Subsection 2.7.2), the normal distribution is not ideal for generating the $l_\alpha r_\alpha$-increments $\Delta p_{\alpha_i,l}$ and $\Delta p_{\alpha_i,r}$, visible in Figure 5.14. Alternatively, the beta distribution can be used to avoid the redraw process ensuring the convexity of the input parameter samples. However, this raises the question of which data and what distribution should be used for the fitting of the distributions for the target output FRV. When fitting the target output FRVs using a normal distribution fitted to the α-level cut values $f_{\alpha_i,l/r}$ of the target output parameter, as in Figure 5.16, the uncertainty is clearly propagated through the system and well preserved and can thus directly be used as input for further uncertainty analysis. However, when fitting the target output FRVs to the deltas of the target output $\Delta f_{\alpha_i,l/r}$, see Figure 5.17, the resulting distribution is no longer suitable for the further use. Due to the grouping of the input samples, the deltas for α-levels smaller than $\alpha = 1$ have the similar size, distorting the fitting process. This highlights, that when generating samples and fitting distributions to data, careful consideration of the uncertainty model, propagation method and goal of the simulation is required.

Chapter 6

Summary & Outlook

Summary This thesis was part of the SPP 1886 Prioprity programme subproject 14 titled 'Dynamische Analyse prosthetischer Strukturen mit polymorpher Unschärfe', which, when translated, reads 'Dynamic analysis of prosthetic structures in the presence of polymorphic uncertainty'. As evident from the title, the goal is to examine a prosthesis in the presence of polymorphic uncertainty. This can be subdivided into multiple sub-tasks. First of all, a model, in this case the human leg with a foot prosthesis, has to be created. Once the model is available, it has to be simulated dynamically, requiring an efficient forward dynamics algorithm. Lastly, and the main focus of this work, is the formulation of an algorithm that can deal with polymorphic uncertainty, that is a combination of aleatoric and epistemic uncertainty.

Four major advancements were contributed to this goal in this thesis. The first contribution was the creation of a full leg model with a flexible prosthesis. A biomechanical model of the human leg was developed, based on anatomical data. The prosthesis was modelled with the help of predeforemd geometrically exact beam theory. A more complex variant of the model was also developed by including the layered structure of the prosthesis in the model via homogenisation. With this model, two distinct phases of the gait cycle, namely the stance and swing phase, can be simulated, providing data on the movement of the leg and the prosthesis under load. Following the successful development of the model, epistemic uncertainty in multiple parameters was propagated through this model, after substantially increasing the computational efficiency of the Graph Follower algorithm. The second contribution consists of this computational efficiency increase of the Graph Follower algorithm, by separating the timestep for the forward dynamics and the α-level optimisation and adding a novelty check after the optimisation step of the algorithm. The third major contribution, was applying the knowledge about epistemic uncertainty propagation to develop a new error model and analysis possibility for optical marker based motion capture. Epistemic uncertainty in the marker positions during optical marker based motion capture has not yet been considered elsewhere. With the new model, the effect of various error sources during the measurements on subsequent joint angle calculations was examined. The main advantages of the introduced method are that the er-

ror model is well suited for the uncertainty present in marker position errors and its bi-directionality. The method provides an efficient worst-case analysis of the effects errors in the markers' positions have on the joint angles, while simultaneously providing information on acceptable errors in the position to prevent unacceptably large errors in the joint angles. The fourth and final contribution is the expansion of the uncertainty model from purely epistemic uncertainty to polymorphic uncertainty in the form of fuzzy random variables. This type of uncertainty can model both variability and inaccuracy or imprecision, allowing for broader representation of occurring uncertainty. Various efficiency increases in the algorithm were implemented to enable it to propagate fuzzy random variables through the model of the human leg with a flexible prosthesis. Overall, this thesis adds new methods to the big field of uncertainty analysis.

Findings The main conclusion from working with uncertainty during the creation of this thesis, is that while uncertainty is omnipresent and therefore critical to explicitly consider, the required computational cost is still a problem. Thus, when working with uncertainty or trying to include it in simulations, a large part of the focus should be put on making the deterministic simulation at the core of the uncertainty analysis as efficient as possible. It is also beneficial for the uncertainty consideration, if the implementation of the core deterministic simulation allows for parallelisation of calculations. Another conclusion is, that while the different aspects of the developed algorithms stand alone, it is difficult to modularise them since the uncertainty model has to be adapted and specialised for the underlying deterministic simulation. Furthermore, the proposed model and method for joint angle calculations in the presence of marker position uncertainty demonstrate that the effects of marker position errors on subsequent calculations can be quite large. The development of accurate models for these errors and their propagation through the calculations helps in the development and verification of methods for compensating and reducing the errors and their effects. Lastly, while working with uncertainty in various forms, it became clear that the uncertainty model and propagation method should not be considered without also considering its interactions with the simulation and the desired goal of the calculations, as was evident when calculating a failure probability based on a fuzzy target output.

Outlook There are many ways with which the progress of this work can be continued. The most obvious expansion is the model itself. At the moment, a forward dynamics simulation of the multibody model with rigid and flexible bodies is at the core of the uncertainty analysis. Ultimately however, human gait uses two legs and mathematically leads to a optimal control problem with opening and closing ground contact. Of course, this type of model is much

more computationally expensive, than the model currently used to evaluate the algorithms introduced here. While in theory the application of the uncertainty propagation algorithms is not limited by model size, the computational time and/or power required becomes problematic. Two approaches can be suggested to counter this problem. For one, the current or future model could be optimised to greatly reduce the calculation time required. One possibility would be to include the null-space matrix method into the model, see [Bet05], however, this proved problematic when paired with CasADi [And19]. Alternatively, the implementation of the code can be further improved. Lastly, the algorithms can be investigated for further improvements. Section 2.7.5 details some of the possible options, the biggest one being parallelisation.

References

[Alj20] Y. Aljammal. “Gelenkwinkelbestimmung beim menschlichen Gang aus optischen Tracking – Ein Vergleich”. Bachelor’s Thesis. Institute of Applied Dynamics – Friedrich-Alexander-Universität, 2020.

[And19] J. A. E. Andersson et al. “CasADi – A software framework for nonlinear optimization and optimal control”. In: *Mathematical Programming Computation* 11 (2019), pp. 1–36.

[Ant95] S. S. Antman. *Nonlinear problems of elasticity*. Springer, 1995.

[Asi21] M. Asif et al. “Advancements, trends and future prospects of lower limb prosthesis”. In: *IEEE Access* 9 (2021), pp. 85956–85977.

[Bak17] R. Baker et al. “The Conventional Gait Model - Success and Limitations”. In: *Handbook of Human Motion*. Ed. by B. Müller et al. Cham: Springer International Publishing, 2017, pp. 1–19. ISBN: 978-3-319-30808-1. DOI: 10.1007/978-3-319-30808-1_25-2.

[Ber21] R. Berkachy. “Fuzzy Random Variables and Fuzzy Distributions”. In: *The Signed Distance Measure in Fuzzy Statistical Analysis: Theoretical, Empirical and Programming Advances* (2021), pp. 87–113.

[Bet05] P. Betsch. “The discrete null space method for the energy consistent integration of constrained mechanical systems: Part I: Holonomic constraints”. In: *Computer Methods in Applied Mechanics and Engineering* 194.50-52 (2005), pp. 5159–5190.

[Bet06] P. Betsch and S. Leyendecker. “The discrete null space method for the energy consistent integration of constrained mechanical systems. Part II: Multibody dynamics”. In: *International journal for numerical methods in engineering* 67.4 (2006), pp. 499–552.

[Big23] L. M. Biga et al. *Anatomy & Physiology – Open Textbook*. https://open.oregonstate.education/aandp/. Licensed under CC BY-SA 4.0. Accessed: 2023-12-11. 2023.

[Bil95] P. Billingsley. “Measure and probability”. In: *John Wiley & Sons* (1995).

[Bîr13] M. Bîrsan et al. “Mechanical behavior of sandwich composite beams made of foams and functionally graded materials”. In: *International Journal of Solids and Structures* 50.3-4 (2013), pp. 519–530.

[Bla15] J. Blazek. *Computational Fluid Dynamics: Principles and Applications*. 3rd. Elsevier, 2015.

[Blo23] J. Bloehs. “Homogenisation of the material parameters of a multilayered carbon fiber spring prosthetic foot”. Bachelor’s Thesis. Institute of Applied Dynamics – Friedrich-Alexander-Universität, 2023.

[Cam17] V. Camomilla et al. “Methodological factors affecting joint moments estimation in clinical gait analysis: A systematic review”. In: *Biomedical engineering online* 16.1 (2017), pp. 1–27.

[Cap05] A. Cappozzo et al. “Human movement analysis using stereophotogrammetry. Part 1: Theoretical Background”. In: *Gait & Posture* 21.2 (Feb. 2005), pp. 186–196. ISSN: 09666362. DOI: 10.1016/S0966-6362(04)00025-6. (Visited on 04/28/2020).

[Ces14] E. Ceseracciu, Z. Sawacha, and C. Cobelli. “Comparison of markerless and marker-based motion capture technologies through simultaneous data collection during gait: Proof of concept”. In: *PloS one* 9.3 (2014), e87640.

[Cha94] C. W. Chan and A. Rudins. “Foot biomechanics during walking and running”. In: *Mayo Clinic Proceedings*. Vol. 69. Elsevier. 1994, pp. 448–461.

[Cha75] R. F. Chandler et al. *Investigation of inertial properties of the human body*. Vol. 53. Aerospace Medical Research Laboratory Wright-Patterson Air Force Base, OH, USA, 1975.

[Cha21] A. Chatzitofis et al. “DeMoCap: Low-Cost Marker-Based Motion Capture”. In: *International Journal of Computer Vision* 129.11 (2021), pp. 3338–3366. URL: https://link.springer.com/article/10.1007/s11263-021-01526-z.

[Chi05] L. Chiari et al. “Human Movement Analysis Using Stereophotogrammetry. Part 2: Instrumental Errors”. In: *Gait & Posture* 21.2 (Feb. 1, 2005), pp. 197–211. ISSN: 0966-6362. DOI: 10.1016/j.gaitpost.2004.04.004. URL: https://www.sciencedirect.com/science/article/pii/S0966636204000682 (visited on 02/21/2022).

[Chu67] K. L. Chung. “Markov chains”. In: *Springer-Verlag, New York* (1967).

[Cli17] D. Cline. *Variational principles in classical mechanics*. University of Rochester River Campus Librarie, 2017. URL: https://digitalshowcase.oru.edu/cgi/viewcontent.cgi?article=1001&context=library_books.

[Col18] S. L. Colyer et al. "A review of the evolution of vision-based motion analysis and the integration of advanced computer vision methods towards developing a markerless system". In: *Sports medicine-open* 4.1 (2018), pp. 1–15.

[Con21] M. Conconi et al. "Quantification of the Errors Associated with Marker Occlusion in Stereophotogrammetric Systems and Implications on Gait Analysis". In: *Journal of Biomechanics* 114 (Jan. 2021), p. 110162. ISSN: 00219290. DOI: 10.1016/j.jbiomech.2020.110162. URL: https://linkinghub.elsevier.com/retrieve/pii/S0021929020305868 (visited on 06/29/2022).

[Cou28] R. Courant, K. Friedrichs, and H. Lewy. "Über die partiellen Differenzengleichungen der mathematischen Physik". In: *Mathematische Annalen* 100.1 (1928), pp. 32–74.

[Cou09] I. Couso and D. Dubois. "On the variability of the concept of variance for fuzzy random variables". In: *IEEE Transactions on Fuzzy Systems* 17.5 (2009), pp. 1070–1080.

[CCBYSA] *Creative Commons Attribution-ShareAlike 4.0 International*. https://creativecommons.org/licenses/by-sa/4.0/. Accessed: 2023-12-11.

[CCBY] *Creative Commons Attribution 4.0 International*. https://creativecommons.org/licenses/by/4.0/. Accessed: 2023-12-11.

[Cri99] M. A. Crisfield and G. Jelenic. *Objectivity of strain measures in the geometrically exact three-dimensional beam theory and its finite-element implementation*. Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences, 1999.

[Cro20] S. Crowder et al. "Monte Carlo methods for the propagation of uncertainties". In: *Introduction to statistics in metrology* (2020), pp. 153–180.

[Dav91] R.B. Davis III et al. "A gait analysis data collection and reduction technique". In: *Human movement science* 10.5 (1991), pp. 575–587.

[Del05] U. Della Croce et al. “Human movement analysis using stereophotogrammetry. Part 4: Assessment of anatomical landmark misplacement and its effects on joint kinematics”. In: *Gait & Posture* 21.2 (Feb. 2005), pp. 226–237. ISSN: 09666362. DOI: 10.1016/j.gaitpost.2004.05.003. URL: https://linkinghub.elsevier.com/retrieve/pii/S0966636204000785 (visited on 04/28/2020).

[Des02] D. Desmond and M. MacLachlan. “Psychological issues in prosthetic and orthotic practice: A 25 year review of psychology in Prosthetics and Orthotics International”. In: *Prosthetics and orthotics international* 26.3 (2002), pp. 182–188.

[Dre22] M. J. Dreyer et al. “European Society of Biomechanics S.M. Perren Award 2022: Standardized Tibio-Femoral Implant Loads and Kinematics”. In: *Journal of Biomechanics* 141 (Aug. 2022), p. 111171. ISSN: 00219290. DOI: 10.1016/j.jbiomech.2022.111171.

[Eis19a] M. Eisentraudt and S. Leyendecker. “Epistemic Uncertainty in Optimal Control Simulation”. In: *Mechanical Systems and Signal Processing* 121 (Apr. 2019), pp. 876–889. ISSN: 08883270. DOI: 10.1016/j.ymssp.2018.12.001. URL: https://linkinghub.elsevier.com/retrieve/pii/S0888327018307787 (visited on 09/10/2019).

[Eis19b] M. Eisentraudt and S. Leyendecker. “Fuzzy Uncertainty in Forward Dynamics Simulation”. In: *Mechanical Systems and Signal Processing* 126 (July 2019), pp. 590–608. ISSN: 08883270. DOI: 10.1016/j.ymssp.2019.02.036. URL: https://linkinghub.elsevier.com/retrieve/pii/S0888327019301207 (visited on 09/10/2019).

[EMPO] *The Empower prosthesis*. https://www.ottobock.de/prothesen/beinprothesen/prothesenfuesse/empower/. Accessed: 2021-07-07.

[Fis07] J. Fish and T. Belytschko. *A first course in finite elements*. Wiley, 2007.

[Gal89] F. Galton. *The Galton Board*. https://en.wikipedia.org/wiki/Galton_board. Accessed: 2024-01-04. 1889.

[Gey11] C. J. Geyer. “Introduction to Markov chain Monte Carlo”. In: *Handbook of Markov chain Monte Carlo* 20116022 (2011), p. 45.

[Gho19a] B. Ghojogh et al. “Fitting a mixture distribution to data: Tutorial”. In: *arXiv preprint arXiv:1901.06708* (2019).

[Gho19b] N. Ghorbani and M. J. Black. “SOMA: Solving optical marker-based MoCap automatically”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 11117–11126.

[Gho19c] S. Ghorbani, A. Etemad, and N. F. Troje. “Auto-labelling of Markers in Optical Motion Capture by Permutation Learning”. In: *Computer Graphics International Conference*. Springer. 2019, pp. 167–178.

[Gil95a] W. R. Gilks, S. Richardson, and D. Spiegelhalter. *Markov chain Monte Carlo in practice*. CRC press, 1995.

[Gil95b] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. “Introducing Markov chain Monte Carlo”. In: *Markov chain Monte Carlo in practice* 1 (1995).

[Gio22] L. Gionfrida et al. “Validation of two-dimensional video-based inference of finger kinematics with pose estimation”. In: *PloS one* 17.11 (2022), e0276799.

[Gro83] E. S. Grood and W. J. Suntay. “A Joint Coordinate System for the Clinical Description of Three-Dimensional Motions: Application to the Knee”. In: *Journal of Biomechanical Engineering* 105.2 (May 1983), pp. 136–144. ISSN: 0148-0731, 1528-8951. DOI: 10.1115/1.3138397.

[Gud98] S. Gudder. “Fuzzy probability theory”. In: *Demonstratio Mathematica* 31.1 (1998), pp. 235–254.

[Haf05] B. J. Hafner. “Clinical prescription and use of prosthetic foot and ankle mechanisms: A review of the literature”. In: *JPO: Journal of Prosthetics and Orthotics* 17.4 (2005), S5–S11.

[Haf02] B. J. Hafner et al. “Energy storage and return prostheses: Does patient perception correlate with biomechanical analysis?” In: *Clinical Biomechanics* 17.5 (2002), pp. 325–344.

[Hai00] E. Hairer and C. Lubich. “Long-time energy conservation of numerical methods for oscillatory differential equations”. In: *SIAM journal on numerical analysis* 38.2 (2000), pp. 414–441.

[Hai06] E. Hairer et al. “Geometric numerical integration”. In: *Oberwolfach Reports* 3.1 (2006), pp. 805–882.

[Hei21] S. Heinrich. “Lower body joint angle determination during human gait based on optical segment positioning using CGM2.4 and marker clusters”. Project Thesis. Institute of Applied Dynamics – Friedrich-Alexander-Universität, 2021.

[Hel03] J. C. Helton and F. J. Davis. “Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems”. In: *Reliability Engineering & System Safety* 81.1 (2003), pp. 23–69.

[Her12] H. M. Herr and A. M. Grabowski. “Bionic ankle-foot prosthesis normalizes walking gait for persons with leg amputation”. In: *Proc. Roy* 279.1728 (2012), pp. 457–464.

[Hua22] D. Huang and S. Leyendecker. “An electromechanically coupled beam model for dielectric elastomer actuators”. In: *Computational Mechanics* 69.3 (2022), pp. 805–824.

[Hüb22] J. Hübner. “Estimating spinal curvature and spinal mobility based on intertial measurement unit data”. Project Thesis. Institute of Applied Dynamics – Friedrich-Alexander-Universität, 2022.

[Hut04] D. V. Hutton. *Fundamentals of finite element analysis*. McGraw-Hill, 2004.

[Kad89] M. P. Kadaba, H. K. Ramakrishnan, and M. E. Wootten. “Measurement of lower extremity kinematics during level walking”. In: *Journal of orthopaedic research* 7.3 (1989), pp. 383–392.

[Kir06] C. Kirtley. *Clinical gait analysis: Theory and practice*. Elsevier Health Sciences, 2006.

[Kiu09] A. D. Kiureghian and O. Ditlevsen. “Aleatory or Epistemic? Does It Matter?” In: *Structural Safety* 31.2 (Mar. 2009), pp. 105–112. ISSN: 01674730. DOI: 10.1016/j.strusafe.2008.06.020. URL: https://linkinghub.elsevier.com/retrieve/pii/S0167473008000556 (visited on 09/23/2019).

[Koc17] M. W. Koch, M. Ringkamp, and S. Leyendecker. “Discrete mechanics and optimal control of walking gaits”. In: *Journal of Computational and Nonlinear Dynamics* 12.2 (2017), p. 021006.

[Krä01] V. Krätschmer. “A unified approach to fuzzy random variables”. In: *Fuzzy sets and systems* 123.1 (2001), pp. 1–9.

[Kru87] R. Kruse and K. D. Meyer. *Statistics with vague data*. Vol. 6. Springer Science & Business Media, 1987.

[Kuy16] F. Kuypers. *Klassische Mechanik*. John Wiley & Sons, 2016.

[Kwa78] H. Kwakernaak. “Fuzzy random variables – I. Definitions and theorems”. In: *Information sciences* 15.1 (1978), pp. 1–29.

[Kwa79] H. Kwakernaak. “Fuzzy random variables – II. Algorithms and examples for the discrete case”. In: *Information sciences* 17.3 (1979), pp. 253–278.

[Lam23] W. W .T. Lam, Y. M. Tang, and K. N. K. Fong. "A systematic review of the applications of markerless motion capture (MMC) technology for clinical measurement in rehabilitation". In: *Journal of NeuroEngineering and Rehabilitation* 20.1 (2023), p. 57.

[Lan13] H. Lang, S. Leyendecker, and J. Linn. "Numerical experiments for viscoelastic Cosserat rods with Kelvin-Voigt damping". In: *Proceedings of the ECCOMAS Thematic Conference on Multibody Dynamics*. 2013.

[Lan11] H. Lang, J. Linn, and M. Arnold. "Multi-body dynamics simulation of geometrically exact Cosserat rods". In: *Multibody System Dynamics* 25.285-312 (2011), pp. 11044–010.

[Lea05] A. Leardini et al. "Human movement analysis using stereophotogrammetry. Part 3. Soft tissue artifact assessment and compensation". In: *Gait & Posture* 21.2 (Feb. 1, 2005), pp. 212–225. ISSN: 0966-6362. DOI: 10.1016/j.gaitpost.2004.05.002. URL: https://www.sciencedirect.com/science/article/pii/S0966636204000773 (visited on 02/21/2022).

[Leb24] F. Leboeuf, R. Baker, and M. Sangeux. *PyCGM2 — Documentation of the CGM2 Project*. https://pycgm2.netlify.app. Accessed: 2024-04-15.

[Leb19] F. Leboeuf et al. "The conventional gait model, an open-source implementation that reproduces the past but prepares for the future". In: *Gait & posture* 69 (2019), pp. 235–241.

[Lee19] M. Lee and A. I. T. Salo. "A review of the evolution of vision-based motion analysis and the integration of advanced computer vision methods towards developing a markerless system". In: *Sports Medicine-Open* 5.1 (2019), pp. 1–15.

[Lev07] R. J. LeVeque. *Finite difference methods for ordinary and partial differential equations: Steady-state and time-dependent problems*. SIAM, 2007.

[Ley06] S. Leyendecker. "Mechanical integrators for constrained dynamical systems in flexible multibody dynamics". PhD thesis. Technische Universität Kaiserslautern, 2006.

[Ley08a] S. Leyendecker, P. Betsch, and P. Steinmann. "The discrete null space method for the energy-consistent integration of constrained mechanical systems. Part III: Flexible multibody dynamics". In: *Multibody System Dynamics* 19 (2008), pp. 45–72.

[Ley08b] S. Leyendecker, J. E. Marsden, and M. Ortiz. "Variational integrators for constrained dynamical systems". In: *ZAMM-Journal of Applied Mathematics and Mechanics* 88 (2008), pp. 677–708.

[Ley10] S. Leyendecker et al. “Discrete mechanics and optimal control for constrained systems”. In: *Optimal Control Applications and Methods* 31.6 (2010), pp. 505–528.

[Lip06] L. Lippert et al. *Clinical kinesiology and anatomy*. FA Davis Philadelphia, PA, 2006.

[Log11] D. L. Logan Jr. *A first course in the finite element method*. Cengage Learning, 2011.

[Kal24] M. Kaliske, W. Graf (Editors). *Polymorphic Modelling for the Numerical Design of Structures*. in preparation. Lecture Notes in Applied and Computational Mechanics, Springer, 2024.

[Mar01a] L. J. Marks and J. W. Michael. “Artificial limbs”. In: *British Medical Journal* 323.7315 (2001), pp. 732–735.

[Mar01b] J. E. Marsden and M. West. “Discrete mechanics and variational integrators”. In: *Acta numerica* 10 (2001), pp. 357–514.

[Mas22] G. Mascia et al. “A Functional Calibration Protocol for Ankle Plantar-Dorsiflexion Estimate Using Magnetic and Inertial Measurement Units: Repeatability and Reliability Assessment”. In: *Journal of Biomechanics* 141 (Aug. 2022), p. 111202. ISSN: 00219290. DOI: 10.1016/j.jbiomech.2022.111202.

[Mck79] M. D. McKay, R. J. Beckman, and W. J. Conover. “A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code”. In: *Technometrics* 21.2 (1979), pp. 239–245.

[Men23] D. van der Mensbrugghe. *A Summary Guide to the Latin Hypercube Sampling (LHS) Utility*. 2023.

[Möl04] B. Möller and M. Beer. *Fuzzy randomness: Uncertainty in civil engineering and computational mechanics*. Springer, 2004.

[Möl00] B. Möller, W. Graf, and M. Beer. “Fuzzy structural analysis using α-level optimization”. In: *Computational Mechanics* 26.6 (2000), pp. 547–565.

[Möl07] B. Möller and U. Reuter. *Uncertainty forecasting in engineering*. Springer, 2007.

[Moo03] B. Moore and S. Reich. “Backward error analysis for multisymplectic integration methods”. In: *Numerische Mathematik* 95 (2003), pp. 625–652.

[Mun15] J. Mun. "Understanding and choosing the right probability distributions". In: *Advanced Analytical Models* (2015), pp. 899–917. URL: https://www.researchgate.net/profile/Elvis-Ganyaupfu-2/post/Problem_in_fitting_data/attachment/60d891736b95310001571aa8/AS%3A1039324787261441%401624805747404/download/Understanding+and+Choosing+the+Right+Probability+Distributions.pdf.

[Mus53] N. I. Muskhelishvili et al. *Some basic problems of the mathematical theory of elasticity*. Vol. 15. Noordhoff Groningen, 1953.

[Mye15] C. A. Myers et al. "A probabilistic approach to quantify the impact of uncertainty propagation in musculoskeletal simulations". In: *Annals of biomedical engineering* 43.5 (2015), pp. 1098–1111.

[Ngu78] H. T. Nguyen. "A Note on the Extension Principle for Fuzzy Sets". In: *Journal of Mathematical Analysis and Applications* 64.2 (June 1978), pp. 369–380. ISSN: 0022247X. DOI: 10.1016/0022-247X(78)90045-8. URL: https://linkinghub.elsevier.com/retrieve/pii/0022247X78900458 (visited on 12/02/2019).

[Oha04] A. O'Hagan and J. E. Oakley. "Probability is perfect, but we can't elicit it perfectly". In: *Reliability Engineering & System Safety* 85.1-3 (2004), pp. 239–248.

[OPT] *Optical Motion Capture Systems*. https://metamotion.com/motion-capture/optical-motion-capture-1.htm Accessed 30 November 2023.

[Pal11] N. Palastanga and R. Soames. *Anatomy and human movement, structure and function with PAGEBURST access, 6: Anatomy and human movement*. Elsevier Health Sciences, 2011.

[Pau03] J. A. Paulos. *A mathematician plays the stock market*. Elsevier Brasil, 2003.

[Pen18] J. Penner and S. Leyendecker. "Optimization based muscle wrapping in biomechanical multibody simulations". In: *PAMM* 18.1 (2018), e201800311.

[PBM19] A. Peters, R. Baker, and M. Morris. "A proposal for an update to the conventional gait model: Removing wand markers from thighs and shanks". In: *Gait & posture* 68 (2019), pp. 507–510.

[Pir17] W. Pirker and R. Katzenschlager. "Gait disorders in adults and the elderly: A clinical guide". In: *Wiener Klinische Wochenschrift* 129.3-4 (2017). https://pubmed.ncbi.nlm.nih.gov/27770207/ Accessed: 2023-12-11, pp. 81–95.

[Pis14] H. Pishro-Nik. *Introduction to probability, statistics, and random processes*. Kappa Research, LLC Blue Bell, PA, USA, 2014.

[Piv19] D. Pivovarov et al. "Fuzzy dynamics of multibody systems with polymorphic uncertainty in the material microstructure". In: *Computational Mechanics* 64 (2019), pp. 1601–1619.

[Pur93] M. L. Puri, D. A. Ralescu, and L. Zadeh. "Fuzzy random variables". In: *Readings in fuzzy sets for intelligent systems*. Elsevier, 1993, pp. 265–271.

[Ram98] M. H. Ramsey. "Sampling as a source of measurement uncertainty: techniques for quantification and comparison with analytical sources". In: *Journal of Analytical Atomic Spectrometry* 13.2 (1998), pp. 97–104.

[Red93] J. N. Reddy. *An introduction to the finite element method*. McGraw-Hill, 1993.

[Rez20] M. Rezaei et al. "Fuzzy uncertainty analysis and reliability assessment of aeroelastic aircraft wings". In: *The Aeronautical Journal* 124.1275 (2020), pp. 786–811.

[Ric08] J. Richards. *Biomechanics in clinic and research*. Churchill Livingstone, 2008.

[Sal08] A. Saltelli et al. *Global sensitivity analysis: The primer*. John Wiley & Sons, 2008.

[San23] G. Sanderson. *But what is the Central Limit Theorem?* https://www.youtube.com/watch?v=zeJD6dqJ5lo, 3Blue1Brown, YouTube. Accessed: 31.07.2023.

[San11] A. Sandholm et al. "Evaluation of a geometry-based knee joint compared to a planar knee joint". In: *The Visual Computer* 27 (2011), pp. 161–171.

[Sch19] E. S. Scheiterer. "Simulation of a prosthetic foot modelled by a predeformed geometrically exact beam". Master's Thesis. Institute of Applied Dynamics – Friedrich-Alexander-Universität, 2019.

[Sch24a] E. S. Scheiterer, M. Eisentraudt, and S. Leyendecker. "Dynamic analysis of prosthetic structures with polymorphic uncertainty". In: *Polymorphic Modelling for the Numerical Design of Structures*. This work quotes from the version submitted to the editors on 7th July 2023. Lecture Notes in Applied and Computational Mechanics, Springer, 2024.

[Sch21a] E. S. Scheiterer and S. Leyendecker. “Forward dynamics simulation of a human leg model with a geometrically exact beam model as prosthetic foot.” In: *PAMM* 21.1 (2021), e202100096.

[Sch21b] E. S. Scheiterer and S. Leyendecker. “Predeformed geometrically exact beam model for a dynamic-response prosthesis”. In: *PAMM* 20.1 (2021), e202000152.

[Sch22a] E. S. Scheiterer and S. Leyendecker. “Correction to: Fuzzy forward dynamics of distinct gait phases with a prosthetic foot”. In: *Computational Mechanics* 70.3 (2022), pp. 515–515.

[Sch22b] E. S. Scheiterer and S. Leyendecker. “Fuzzy forward dynamics of distinct gait phases with a prosthetic foot”. In: *Computational Mechanics* 70.3 (2022), pp. 501–513.

[Sch24b] Scheiterer, E. S. and Heinrich, S. and Liphardt, A.-M. and Leyendecker, S. “Marker position uncertainty in joint angle analysis for normal human gait — Demonstration of a new error-modelling approach”. In: *Biomedical Signal Processing and Control* (2024). This work quotes from the version submitted for the reply to the first peer review process by the journal, on 26th January 2024.

[Sch16] T. Schlögl and S. Leyendecker. “Electrostatic–viscoelastic finite element model of dielectric actuators”. In: *Computer Methods in Applied Mechanics and Engineering* 299 (2016), pp. 421–439.

[Sch24c] A. Schmidt et al. “Strategies for reliability analysis in the presence of polymorphic uncertainty applied to a membrane structure”. In: *Polymorphic Modelling for the Numerical Design of Structures*. Submitted to the editors on 7th July 2023. Lecture Notes in Applied and Computational Mechanics, Springer, 2024.

[Sco22] B. Scott et al. “Healthcare applications of single camera markerless motion capture: A scoping review”. In: *PeerJ* 10 (2022), e13517.

[Soe19] M. Söhnlein. “Qualifzierung von Simulationsparametern einer Fussprothese durch numerische und experimentelle Modalanalyse”. MA thesis. Institute of Applied Dynamics – Friedrich-Alexander-Universität, 2019.

[SPP] *Dynamische Analyse prosthetischer Strukturen mit polymorpher Unschärfe*. https://tu-dresden.de/bu/bauingenieurwesen/sdt/forschung/spp1886/teilprojekte/zweite-foerderphase/tp14-2. Accessed: 2023-12-29.

[Sta22] M. Stavole et al. “Homogenization of the constitutive properties of composite beam cross-sections”. In: *ECCOMAS Congress 2022-8th European Congress on Computational Methods in Applied Sciences and Engineering*. 2022.

[Ste18] P. M. Stevens, J. Rheinstein, and S. R. Wurdeman. “Prosthetic foot selection for individuals with lower-limb amputation: A clinical practice guideline”. In: *Journal of Prosthetics and Orthotics* 30.4 (2018), p. 175.

[Sut01] D. H. Sutherland. “The evolution of clinical gait analysis: Part I – Kinesiological EMG”. In: *Gait & posture* 14.1 (2001), pp. 61–70.

[Sut02] D. H. Sutherland. “The evolution of clinical gait analysis: Part II – Kinematics”. In: *Gait & posture* 16.2 (2002), pp. 159–179.

[Sut05] D. H. Sutherland. “The evolution of clinical gait analysis: Part III – Kinetics and energy assessment”. In: *Gait & posture* 21.4 (2005), pp. 447–461.

[Top20] M. Topley and J. G. Richards. “A comparison of currently available optoelectronic motion capture systems”. In: *Journal of biomechanics* 106 (2020). DOI: 10.1016/j.jbiomech.2020.109820.

[Tra08] G. Trajković. “Measurement: Accuracy and precision, reliability and validity”. In: *Encyclopedia of Public Health. Springer Holanda* (2008).

[Uch22] T. K. Uchida and A. Seth. “Conclusion or Illusion: Quantifying Uncertainty in Inverse Analyses From Marker-Based Motion Capture due to Errors in Marker Registration and Model Scaling”. In: *Frontiers in Bioengineering and Biotechnology* 10 (2022), p. 874725.

[Van18] E. van der Kruk and M. M. Reijne. “Accuracy of Human Motion Capture Systems for Sport Applications; State-of-the-Art Review”. In: *European Journal of Sport Science* 18.6 (July 2018), pp. 806–819. ISSN: 1746-1391, 1536-7290. DOI: 10.1080/17461391.2018.1463397.

[VARI] *The Össur Vari-Flex® prosthesis*. https://www.ossur.com/de-de/prothetik/fusse/vari-flex. Accessed: 2021-07-07.

[Ver09] R. Versluys et al. “Prosthetic feet: State-of-the-art review and the importance of mimicking human ankle–foot biomechanics”. In: *Disability and Rehabilitation: Assistive Technology* 4.2 (2009), pp. 65–75.

[Viv21] L. G. de Vivo Nicoloso et al. “Towards 3D printing of a monocoque transtibial prosthesis using a bio-inspired design workflow”. In: *Rapid Prototyping Journal* 27.11 (2021), pp. 67–80.

[Wat99] R. L. Waters and S. Mulroy. “The energy expenditure of normal and pathologic gait”. In: *Gait & posture* 9.3 (1999), pp. 207–231.

[Whi14] M. W. Whittle. *Gait analysis: An introduction*. Butterworth-Heinemann, 2014.

[Woe11] C. Woernle. *Mehrkörpersysteme*. Springer, 2011.

[Wri08] Peter Wriggers. *Nonlinear finite element methods*. Springer Science & Business Media, 2008.

[Wu02] G. Wu et al. “ISB Recommendation on Definitions of Joint Coordinate System of Various Joints for the Reporting of Human Joint Motion – Part I: Ankle, Hip, and Spine”. In: *Journal of Biomechanics* 35.4 (Apr. 2002), pp. 543–548. ISSN: 00219290. DOI: 10.1016/S0021-9290(01)00222-6.

[Yam21] Y. Yamazaki et al. “Real-Time Motion Analysis System Using Low-Cost Web Cameras and Photogrammetry”. In: *Frontiers in Bioengineering and Biotechnology* 9 (2021), p. 790764.

[Zad65] L. A. Zadeh. “Fuzzy Sets”. In: *Information and Control* 8.3 (June 1, 1965), pp. 338–353. ISSN: 0019-9958. DOI: 10.1016/S0019-9958(65)90241-X. URL: https://www.sciencedirect.com/science/article/pii/S001999586590241X (visited on 01/24/2022).

[Zie77] O. C. Zienkiewicz. *The finite element method*. McGraw-Hill, 1977.

[Zie06] O. C. Zienkiewicz and K. Morgan. *Finite elements and approximation*. Courier Corporation, 2006.

Schriftenreihe Technische Dynamik

bereits veröffentlicht wurden:

- Maas, R. "Biomechanics and optimal control simulations of the human upper extremity". *Dissertation*, LTD, FAU.
  Band 1, 2014
- Koch, M. "Structure preserving simulation of non-smooth dynamics and optimal control". *Dissertation*, LTD, FAU.
  Band 2, 2016
- Schlögl, T. "Modelling, simulation and optimal control of dielectric elastomer actuated systems". *Dissertation*, LTD, FAU.
  Band 3, 2018
- Budday, D. "High-Dimensional Robotics at the Nanoscale — Kino-Geometric Modeling of Proteins and Molecular Mechanisms". *Dissertation*, LTD, FAU.
  Band 4, 2019
- Gail, T. "Numerical studies on variational multirate integrators and their use in discrete mechanics and optimal control". *Dissertation*, LTD, FAU.
  Band 5, 2021
- Penner, J. "A discrete variational approach to muscle wrapping in musculoskeletal optimal control simulations". *Dissertation*, LTD, FAU.
  Band 6, 2023
- Leitz, T. "Galerkin Lie group variational integrators". *Dissertation*, LTD, FAU.
  Band 7, 2023
- Phutane, U. "Optimal control simulations of human hand grasping". *Dissertation*, LTD, FAU.
  Band 8, 2024
- Martonová, D. "Computational modelling and simulation of rat heart electromechanics – from (smoothed) finite element methods towards a ligand-receptor model". *Dissertation*, LTD, FAU.
  Band 9, 2024
- Holz, D. "On Aspects of Cardiac and Artificial Muscle Modelling– Insights into Orthotropic Tissue Structure and Dielectric Elastomer Actuators". *Dissertation*, LTD, FAU.
  Band 10, 2024

- Chen, X. “Kinematic assessment to characterize protein and macromolecular conformations”. *Dissertation*, LTD, FAU.
  Band 11, 2024

- Phansalkar, D. “Phase-field modeling of fracture with a spatially varying length variable and adaptive mesh refinement”. *Dissertation*, LTD, FAU.
  Band 12, 2024