

**Entwicklung eines Verfahrens zur Analyse des
Materialreaktionsverhaltens von Mittelsohlen für
verschiedene Laufschuhmodelle über eine simulierte
Marathondistanz.**

Bachelorthesis

für die Prüfung zum

Bachelor of Engineering

im Studiengang angewandte Biomechanik

Fakultät Maschinenbau und Verfahrenstechnik

an der Hochschule für Technik, Wirtschaft und Medien Offenburg

von

Tom-Felix Kraus vom Cleff

23. August 2021

Bearbeitungszeitraum

31.5. – 31.8.2021

1. Gutachter

Prof. Dr. Steffen Willwacher

2. Gutachter

Prof. Dr. Grit Köhler

Eidesstattliche Erklärung

Hiermit versichere ich eidesstattlich, dass ich die vorliegende Arbeit mit dem Thema „Entwicklung eines Verfahrens zur Analyse des Materialreaktionsverhaltens von Mittelsohlen für verschiedene Laufschuhmodelle über eine simulierte Marathondistanz.“

von mir selbstständig und ohne unerlaubte fremde Hilfe angefertigt worden ist, insbesondere, dass ich alle Stellen, die wörtlich oder annähernd wörtlich oder dem Gedanken nach aus Veröffentlichungen, unveröffentlichten Unterlagen und Gesprächen entnommen worden sind, als solche an den entsprechenden Stellen innerhalb der Arbeit durch Zitate kenntlich gemacht habe, wobei in den Zitaten jeweils der Umfang der entnommenen Originalzitate kenntlich gemacht wurde. Die Arbeit lag in gleicher oder ähnlicher Fassung noch keiner Prüfungsbehörde vor und wurde bisher nicht veröffentlicht. Ich bin mir bewusst, dass eine falsche Versicherung rechtliche Folgen haben wird.

Ort, Datum

Vorname Nachname

Abstract

Hersteller von Laufschuhen versprechen ihren Kunden Schuhe, die einen durch gezielte Energierückgabe der Mittelsohlen oder durch eine optimale Dämpfung bei der sportlichen Leistung unterstützen. Als aktueller Teststandard für das Materialreaktionsverhalten von Sportschuhen gilt der ASTM F1976. In diesem Test werden die Schuhe durch einen wiederholten Impulstest untersucht. Allerdings weder Rücksicht auf die zurückgelegte Distanz, noch auf variierende Körpergewichte genommen. In dieser Arbeit wird ein Messverfahren entwickelt, das Sportschuhe über eine simulierte Marathondistanz in Bezug auf die verarbeitete Energiemengen und das Verändern der Festigkeit analysiert. Es wurde ein Schuh der „Ghost“-Reihe der Firma „Brooks“ mit Hilfe einer elektro-dynamischen Prüfmaschine LMT10 der Firma Zwick/Röll zwei Mal im Bereich des Rückfußes mit zwei verschiedenen Lasten analysiert. Die Einstellungsparameter sind recherchierte Durchschnittswerte, die einen erwachsenen Europäer darstellen. Es hat sich herausgestellt, dass ein um 20% geringeres Körpergewicht eines Läufers nicht eine äquivalente geringere umgesetzte Energiemenge bedeutet. Im Vergleich zu der Messung des schwereren Läufers wurden lediglich 65% der Energiemenge umgesetzt. Darüber hinaus nimmt die Festigkeit der Mittelsohle bei der Messung mit 20% weniger Last kontinuierlich ab, wobei sie bei der Messung mit 100% im letzten Drittel der Messung wieder zunimmt. Als Gründe für dieses Verhalten wurden das nichtlineare Reaktionsverhalten von EVA-Schäumen und das anfängliche Verbessern der Gleitfähigkeit der Polymerketten festgestellt.

Inhaltsverzeichnis

1	Einleitung.....	3
2	Literaturübersicht	6
2.1	Faktor Mensch	6
2.1.1	Anatomie des Fußes	6
2.1.2	Gang- & Laufzyklus.....	7
2.1.3	Laufstile	9
2.2	Faktor Laufschuh	10
2.2.1	Aufbau eines Laufschuhs	10
2.2.2	Materialverhalten bei zyklischer Belastung.....	12
2.3	Datenfindung zum Einstellen der Prüfmaschine.....	17
2.3.1	Bodenreaktionskraftkurven	17
2.3.2	Datenanalyse	23
2.3.3	Schrittfrequenzen	26
2.3.4	Körpergewicht	29
2.3.5	Prüfzeit	29
2.3.6	Zusammenfassung der Werte	30
3	Methode	31
3.1	Untersuchungsdesign	31
3.2	Versuchsaufbau	31
3.3	Datenverarbeitung	38
4	Ergebnisse	40
4.1.1	Einführung Messergebnisse	40
4.1.2	Erster Messdurchgang.....	42
4.1.3	Zweiter Messdurchgang	48
4.1.4	Vergleich der Messdurchgänge	54

5	Diskussion	58
5.1	Datenfindung für die Messdurchführung	58
5.2	Messaufbau	59
5.3	Messergebnisse	61
5.3.1	Interpretation der Zeiten und der Kräfte	61
5.3.2	Interpretation der Deformation	62
5.3.3	Interpretation der Energie	63
5.3.4	Interpretation der Steigung	64
5.3.5	Abschluss Messung	65
6	Zusammenfassung	66
	Tabellenverzeichnis	69
	Abbildungsverzeichnis	70
	Literaturverzeichnis	72
	Anhang	76

1 Einleitung

Seitdem die Menschheit seine Kräfte in sportlichen Auseinandersetzungen misst, ist man bestrebt, egal in welcher Form, sich einen Vorteil gegenüber seines Kontrahenten zu verschaffen.

Dabei ist es erst einmal egal in welcher Form man sich einen Vorteil verschaffen kann. Auf der einen Seite versucht man Athleten für bestimmte Sportarten zu finden, die allein schon durch ihre Statur ihrem Gegner überlegen sind. So sind z.B. große athletische Menschen in Sportarten wie Basketball oder Volleyball den Kleineren gegenüber im Vorteil. In anderen Disziplinen wie dem Turnen hingegen ist es besser zu den Kleinen und nicht zu den Großen zu gehören. Auf der anderen Seite ist man bestrebt durch die stetige Verbesserung der Sportutensilien dem Gegner überlegen zu sein. Zum Beispiel hat der Einzug der Leichtmetalle den Radsport revolutioniert (Katzer & Rohdewald, 2017, S.13) und durch das Erfinden der Weichkernmatten wurde es möglich im Hochsprung gänzlich neue Sprungtechniken anzuwenden (Schulke, 2007, S.4).

Der technische Fortschritt hat sich längst nicht mehr nur auf den Profisportbereich beschränkt. Heute ist jeder in der Lage, durch speziell für seine Sportart abgestimmte Ausrüstung, seine körperliche Leistung bestmöglich zu unterstützen, sei es durch das Verwenden von Fitnessarmbändern die einem seine Vitalfunktionen in Echtzeit auf das Smartphone übertragen oder durch die Wahl der neusten Schuhmodelle. „Adidas“ verspricht in seiner Werbung, dass ihre Schuhe, durch das Speichern und Abgeben von auftretenden Energien, beim Laufen einen „Energieschub“ (Adidas, 2021) abgeben und einem so zu neuen persönlichen Bestzeiten verhelfen. Andere Firmen wie „On“ versprechen mit ihren Schuhmodellen durch eine adaptive Mittelsohle ihren Trägern ein Laufen wie auf Wolken zu ermöglichen (On, 2021). Durch das sogenannte „Technik-Doping“ (Schulke, 2007, S.4) werden Diskussionen geführt ob Schuhe wie z.B. der „Vaporfly“ bei offiziellen Marathons überhaupt erlaubt werden dürfen.

Tung und seine Kollegen haben im Jahre 2014 mit ihrer Studie an der University of Colorado den Begriff „cost of cushioning“ geprägt. In der gleichnamigen Studie fanden sie heraus, dass ein gewisses Maß an Dämpfung die Muskelaktivität der

Läufer reduziert. Dadurch resultierend verringert sich der Sauerstoffbedarf und die Probanden waren in der Lage größere Distanzen zurückzulegen (Tung et al., 2014, S.1 ff.).

In den Laufschuhen, die wir tagtäglich, ohne darüber nachzudenken, verwenden steckt offensichtlich deutlich mehr Know-how als man auf den ersten Blick vermuten würde. Doch wie kann man all diese speziellen Eigenschaften der Schuhe quantifizieren?

Für die Industrie ist ein Testverfahren nach den Rahmenbedingungen der „ASTM International“ (ursprünglich American Society for Testing and Materials) standardisiert worden. Dieser Test umfasst eine 8,5 kg schwere Masse, die aus einer Höhe von 50mm auf den zu untersuchenden Schuh fallen gelassen wird. Ein Kraftaufnehmer gibt anschließend eine Aussage über das Reaktionsverhalten dieser verwendeten Sohle. Es soll jeweils im Vor- und Rückfuß eine Analyse über die Energieverarbeitung und das Verformungsverhalten durchgeführt werden. Bei diesem Test ist eine Energierückgabe, für Schuhe mit einer Sohlendicke von über 7 mm, bis zu 5 J zu erwarten (ASTM F1976, 2013). Dieser, zuletzt 2013 aktualisierte Standard, soll eine umfassende Aussage über die Qualität und das Reaktionsverhalten der Sportschuhe geben.

Es wird von einem Impulstest auf ein Materialverhalten rückgeschlossen, bei dem Analyseaspekte wie z.B. die Belastungszeit oder unterschiedliche Körpergewichte nicht beachtet werden.

Diese Arbeit macht es sich zur Aufgabe ein Verfahren zu entwickeln, das es möglich machen soll, verschiedene Laufschuhmodelle über eine simulierte Marathondistanz zu analysieren und die Ergebnisse untereinander vergleichen zu können. Dabei soll im Detail das Werbeversprechen der Energieverarbeitung und die Veränderung der Festigkeit der Mittelsohlen betrachtet werden.

Für die Umsetzung ist es vorgesehen die elektro-dynamische Prüfmaschine LTM10 von Zwick/Roell des Biomechaniklabors der Hochschule Offenburg zu verwenden.

Der erste Teil dieser Arbeit wird sich mit der Recherche der Informationen befassen, die nötig sind um dieses Vorhaben umsetzen zu können und die zur Verfügung

1 Einleitung

stehende Maschine einrichten zu können. Anschließend werden der Messaufbau und die Messergebnisse präsentiert.

Mit der Umsetzung dieser Arbeit wird es hoffentlich möglich sein einen umfassenden Überblick über das Reaktionsverhalten von Mittelsohlen bereitstellen zu können.

2 Literaturübersicht

2.1 Faktor Mensch

2.1.1 Anatomie des Fußes

Der Fuß ist ein aus 26 einzelnen Knochen bestehendes Konstrukt. Aus Abbildung 1 lässt sich entnehmen, dass der Fuß sich in drei unterschiedliche Areale unterteilen lässt. Der proximalste (lat. für dem Körpermittelpunkt am nächsten) Anteil wird Fußwurzel genannt, hier befindet sich die für Kapitel 2.1.2 wichtige Ferse (lat. tuber calcanei). Der distalere (lat. für dem Körpermittelpunkt fernere) Abschnitt wird Mittelfuß genannt, dieser besteht aus den fünf Mittelfußknochen (lat. ossa metatarsalia). Als letztes Areal kategorisiert man die Zehen, bestehend aus der Großzehe und den vier kleinen Zehen (Zimmer & Appell, 2021, S.92ff.)

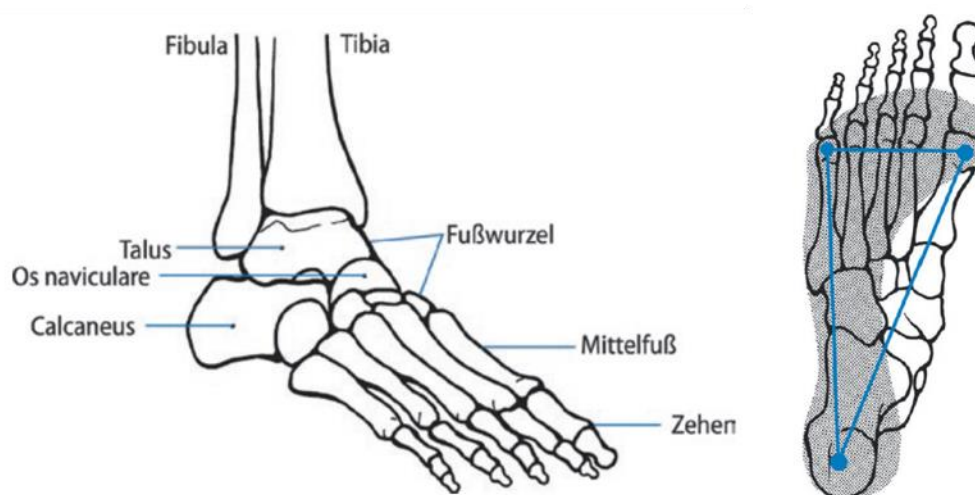


Abbildung 1: Der knöcherne Fuß (Zimmer & Appell, 2021, S.94f.)

Die Hauptaufgabe des Fußes besteht darin Stabilität zu gewähren und entstehende Kräfte in den Boden abzugeben oder aufzunehmen. Für eine optimale Stabilität besitzt der Fuß ein Längs- & Quergewölbe, diese werden durch eine Vielzahl von kurzen Bändern (lat. ligamenta) und kurze Muskeln (lat. muscoli) unterstützt. Wegen dieser Gewölbe entsteht ein über drei Säulen (siehe Abbildung 1) abgestütztes Konstrukt, das entstehende Kräfte und Momente optimal an den Boden abgeben und weitergeben kann (Zimmer & Appell, 2021, S. 93f.).

2.1.2 Gang- & Laufzyklus

Das aufrechte zweibeinige Gehen eines Menschen ist eine sich kontinuierlich wiederholende Abfolge der selben Bewegungsmuster. Es lässt sich präzise beschreiben und an den meisten Menschen, ohne körperliche Einschränkungen, beobachten (Beckers & Deckers, 1997, S.17).

Das Gehen wird durch einen rhythmischen Wechsel zwischen der „Standphase“ und der „Schwungphase“ eines zu betrachtenden Fußes charakterisiert. Ein erster Bodenkontakt der Ferse (engl. heel strike) eines Fußes leitet die „Standphase“ ein, in dieser hat der zu betrachtende Fuß dauerhaft Kontakt zum Boden. Sobald die Großzehe den Kontakt zum Boden verliert (engl. toe off) beginnt die „Schwungphase“, diese endet mit dem erneuten Kontakt der Ferse des selben Fußes am Untergrund. Dieser Vorgang von dem ersten „heel strike“ zum darauffolgenden Fersenkontakt nennt man in der Ganganalyse den „Gangzyklus“ (engl. gait cycle) (Beckers & Deckers, 1997, S.18f.).

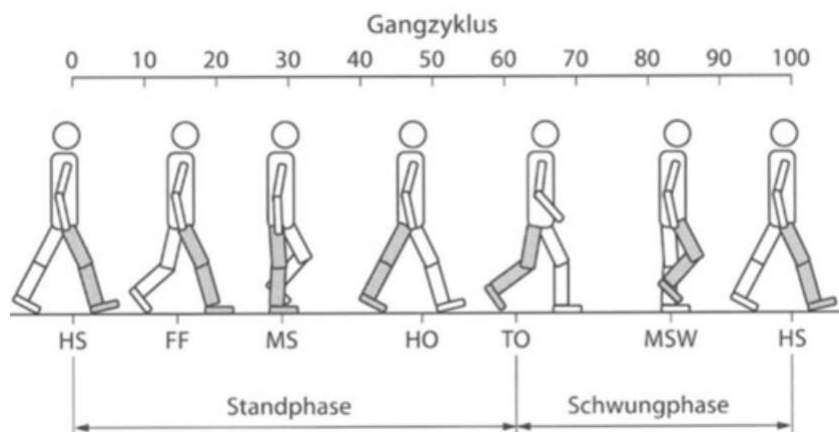


Abbildung 2: Gangzyklus (Beckers & Deckers, 1997, S.19)

Wie man der Abbildung 2 entnehmen kann ist der Gangzyklus genauestens definiert worden, bis zum jeweiligen prozentualen Anteil eines jeden Abschnittes des „gait cycle“. Man erkennt, dass es Zeiträume gibt, in denen beide Füße gleichzeitig Kontakt zum Boden haben. Diesen Zustand nennt man eine „bipedale Phase“ (engl. double limb support), dem gegenüber steht die „monopedale Phase“ (engl. single limb support) in der nur ein Fuß Kontakt zum Boden hat. Dies führt zu einer durchgehenden Verlagerung des Schwerpunktes. In der, für die Durchführung des Laufschuhtests, wichtigen Standphase wandert der Schwerpunkt der sich

fortbewegenden Person von der Ferse beginnend zur Großzehe. Es treten Zustände auf, in denen sich die gesamte Last des Körpers alleine auf der Ferse oder dem Vorfuß befindet. Vom „heel strike“ bis ca. 8% des Gangzyklus, hat lediglich die Ferse Kontakt zum Boden. Danach senkt sich der Vorfuß so weit ab, dass dieser ebenfalls Kontakt zum Untergrund erlangt. Sobald bei 40% des Gangzyklus sich die Ferse beim „heel off“ vom Boden abhebt liegt die gesamte Last auf dem Vorfuß (Beckers & Deckers, 1997, S.19ff.). Dadurch entsteht eine sich kontinuierlich verändernde Beanspruchung des Fußes, darauf wird in Kapitel 2.3.1 umfangreicher eingegangen.

Dem gegenüber steht der Laufzyklus, dieser zeichnet sich ebenfalls durch seine sich zyklisch wiederholenden Bewegungsmuster aus. Allerdings gibt es einige auffällige Unterschiede. Neben der erhöhten Geschwindigkeit ist das Wegfallen der „bipedalen Phase“ die wichtigste Veränderung. Das bedeutet zu keinem Zeitpunkt befinden sich beide Füße gleichzeitig in Kontakt mit dem Boden, stattdessen tritt eine „Flugphase“ (engl. floating phase) auf. In dieser Phase hat der Läufer keinen Kontakt zum Boden. Um eine Flugphase zu ermöglichen muss der Körperschwerpunkt deutlich höher als beim Gehen befördert werden, dies erhöht ebenfalls die Fallhöhe (Beckers & Deckers, 1997, S.17ff.).

Diese Unterschiede ermöglichen es dem Menschen für sich persönlich ein individuelleres „Laufschema“ zu kreieren. Dies macht eine, wie im Gangzyklus, sehr genaue Beschreibung unmöglich.

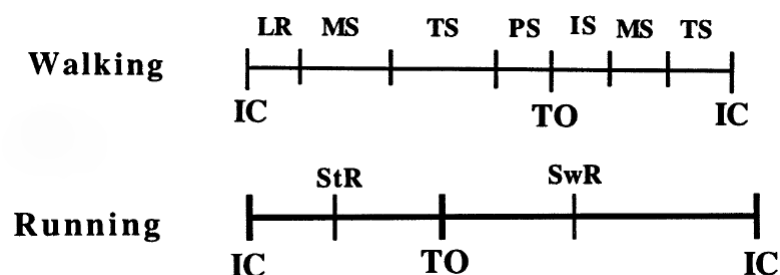


Abbildung 3: Unterschied Gang- & Lauzyklus (Novacheck, 1998, S.79)

Wie man der Abbildung 3 entnehmen kann, reduzieren sich die wichtigen Punkte die klar definierbar sind deutlich. Man spricht jeweils bei der Schwungbeziehungsweise Standphase von dem „Zeitpunkt der Umkehrung“ (engl. stand

phase reversal & swing phase reversal), in Abbildung 3 „StR“ und „SwR,“ an dem der Läufer von der Kraftaufnahme dazu übergeht Kraft für das Abstoßen zu generieren beziehungsweise in der Schwungphase das nach vorne Pendeln der unteren Extremität abbremst (Novacheck, 1998, S.79).

Darüber hinaus wird der individuelle Bewegungsablauf eines Läufers durch sein Trainingsstand und seine körperlichen Voraussetzungen definiert (Novacheck, 1998, S.79).

Die Informationen die in diesem Kapitel erörtert worden sind werden in Kapitel 2.3.1 bei den Bodenreaktionskraftkurven erneut wichtig werden.

2.1.3 Laufstile

Im Gegensatz zum physiologischen Gehen, bei dem fast jeder Mensch sich auf dieselbe Art fortbewegt, kann man bei Läufern prinzipiell drei verschiedene physiologische Stile beobachten.

Der am weitesten verbreitete Laufstil ist der des „Rückfußläufers“ (engl. rear foot runner). Hier setzt der Läufer, wie beim Gehen, zuerst die Ferse auf den Boden ab und rollt dann über den gesamten Fuß ab um sich mit den Zehen abzustößen. Ungefähr 80% der Läufer weltweit verwenden diese Art zu Laufen (Novacheck, 1998, S.78f.)

Wenn der erste Kontakt des Fußes die Mittelfußknochen sind spricht man vom „Mittelfußläufer“ (engl. mid foot runner). Läufer die diese Art des Laufens bevorzugen setzen, wie bereits erwähnt, zuerst den Mittelfuß auf den Boden auf und senken anschließend die Ferse ebenfalls soweit ab bis diese Kontakt zum Boden hat (Novacheck, 1998, S. 78f.).

Den letzten der drei Laufstile nennt man den „Vorfußläufer“. Bei dieser Art des Laufens werden zuerst die Zehen und die Mittelfußknochen auf den Boden abgesetzt. Im Gegensatz zu den beiden bisherigen Varianten wird bei dieser die Ferse nicht auf den Boden abgesetzt. Dadurch kann die Bodenkontaktzeit sehr stark reduziert werden. Die verkürzte Bodenkontaktzeit führt zu einer stärkeren

Muskelkontraktion des Läufers und resultiert so in höheren Geschwindigkeiten. Dieser Laufstil ist sehr oft bei Sprintern zu beobachten (Novacheck, 1998, S.78f.).



Abbildung 4: Laufstile (Ajad Yarra & Pandya, 2020)

In dieser Arbeit wurde der, in Abbildung 4 visualisierte, Stil des „rearfoot runners“ (deu. Rückfußläufer) analysiert, da dieser von den meisten Läufern angewendet wird und somit ein für die Allgemeinheit gültiges Ergebnis liefert.

2.2 Faktor Laufschuh

2.2.1 Aufbau eines Laufschuhs

„Als „Urturnschuh“ gelten bis heute die bekannten „Converse All Star“ der Firma Converse aus dem Jahre 1915 welche sich als erste durch eine **dämpfende Innensohle** und einen **erhöhten bindbaren Schaft** auszeichneten.“ (Kirchhoff et al., 2013)

Seitdem hat sich der Sportschuh deutlich weiterentwickelt. Heute besteht dieser meist aus einer mehrschichtigen Sohle und einem Obermaterial inklusive einer Fersenschale. Das Obermaterial besteht meist aus synthetischen Materialien die mehrere Aufgabe erfüllen. Als wichtigste ist die Stabilität zu nennen. Das Obermaterial hält den Schuh am Fuß und führt den Fuß sicher durch die Abrollbewegung. Darüber hinaus ist das Obermaterial meist ein atmungsaktives netzartiges Kunststoffmaterial, meist bestehend aus Nylon oder Meshmaterialien (zu Maschen gewebter feiner Stoff). Im Bereich der Zehen ist das Obermaterial

verstärkt um ein Durchdrücken der Zehen in der Schwungphase des Laufens zu verhindern (Spanka, 2021).

Neben dem Obermaterial ist zusätzlich die Fersenschale auf den Sohlen befestigt. Diese dient der Fußführung und der Passform des Schuhs. Sie sorgt für ausreichend Stabilität bei unsicherem Untergrund (Spanka, 2021).

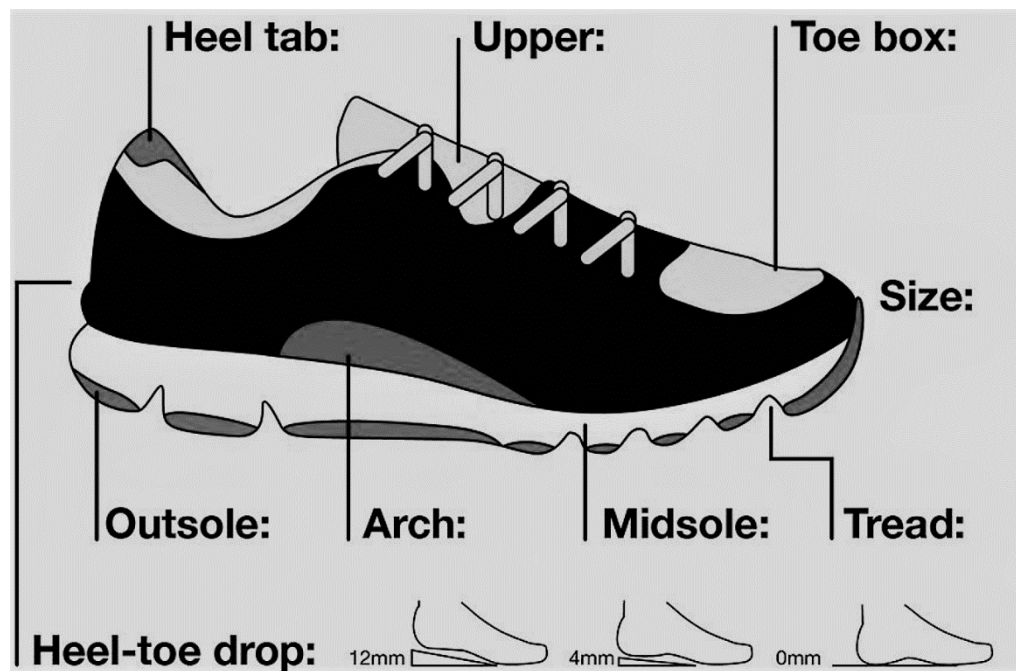


Abbildung 5: Aufbau von Laufschuhen (Beus, 2020)

Die Sohlen lassen sich in Einlege- Zwischen- & Außensohle unterscheiden. Die Einlegesohle dient hauptsächlich der Verbesserung der Passform des Schuhs an den Fuß. Durch ihre nachgiebige Beschaffenheit sorgt sie für eine individuelle Kontaktfläche von der Fußkontur zur Mittelsohle (Spanka, 2021).

Als wichtigstes Element des Schuhs ist die Mittelsohle zu betrachten, die circa zwei Drittel der Dämpfung übernimmt. Darüber hinaus soll die bei der Dämpfung aufgenommene Energie beim Abstoßen wieder abgegeben werden und so den Läufer unterstützen (Kirchhoff et al., 2013). Meist bestehen die Mittelsohlen aus dem Schaummaterial „Ethylen-Vinyl-Acetat“ das sich durch ein geringes Gewicht und ein sehr gutes Dämpfverhalten auszeichnet (Kirchhoff et al., 2013). Die genaue Zusammensetzung variiert von Hersteller zu Hersteller, einige arbeiten mit integrierten Lufteinschlüssen, die ein besseres Dämpfverhalten implementieren sollen, andere arbeiten durchgängig mit dem selben Schaum, verwenden jedoch

innerhalb der Mittelsohle verschiedene Dichten um eine stärkere oder leichtere Dämpfung zu kreieren.

Als letztes Sohlenelement ist die Außensohle zu nennen, die in direktem Kontakt zur Umwelt steht und für den nötigen Halt sorgt. Je nach Sportart variiert die Außensohle, sie ist immer an den bevorzugten Untergrund des Läufers angepasst. Ein Laufschuh für den Wald wird mehr Profiltiefe und eine härtere Beschaffenheit besitzen als der für einen Läufer der hauptsächlich auf Asphalt läuft (Kirchhoff et al., 2013).

Unter dem in Abbildung 5 genannten „heel-toe drop“ (deu. Sprengung) versteht man eine Höhendifferenz der Ferse zu den Zehen. Dies beeinflusst zum Beispiel den Laufstil, mit einem hohen „heel-toe drop“, sprich einer starken Erhöhung der Ferse, ist es für den Läufer sehr schwer zuerst den Mittelfuß abzusetzen. Darüber hinaus beeinflusst die Höhe der Fersenunterlage den im Laufen genutzten Bewegungsradius im Sprunggelenk, dies beeinflusst die Muskelrekrutierung in den unteren Extremitäten (Subic, 2021).

2.2.2 Materialverhalten bei zyklischer Belastung

Um das Materialverhalten der Mittelsohlen verstehen zu können muss man zuerst einen Einblick in den Aufbau des verwendeten Materials erhalten.

Die zu untersuchende Sohle besteht aus einem Kunststoffschäummaterial. Als Kunststoffe bezeichnet man künstlich hergestellte beziehungsweise so nicht natürlich auftretende Materialien. Diese werden auch Polymere genannt, dieser Fachbegriff setzt sich aus den griechischen Silben „poly“ = viel und „meros“ = Teil zusammen und steht für eine Vereinigung vieler Einzelbausteine zu einem Verbundmaterial. Die Eigenschaften des Polymer werden durch die Art der verwendeten Monomere (einmolekulige Ausgangsstoffe) bestimmt. Durch chemische Prozesse werden Monomere zu Makromolekülen und bilden so anschließend als Verbundmaterial das gewünschte Polymer, siehe Abbildung 6 (Weißbach et al., 2015, S.363f.).

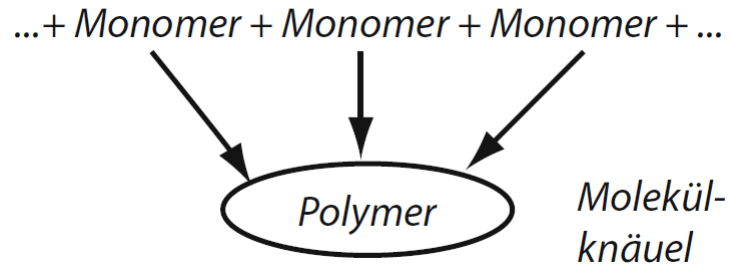


Abbildung 6: Aufbau Polymer (Weißbach et al., 2015, S.364)

Die Art der Verbindung der Makromolekülen bedingt anschließend die Art des Polymers und dessen Eigenschaften.

nicht vernetzte Kunststoffe		vernetzte Kunststoffe
amorph	teilkristallin	
Thermoplaste		Duromere Elastomere

Abbildung 7: Polymerarten (Weißbach et al., 2015, S.365)

Aus Abbildung 7 geht hervor, dass sich Kunststoffe grundsätzlich in vernetzte und unvernetzte Verbindungen einteilen lassen. Für die in dieser Arbeit untersuchten, Mittelsohlen werden Elastomere aus der Klasse der vernetzten Polymere verwendet. Diese zeichnen sich durch ihre meist weitmaschige Vernetzung aus. Auf Grund dieser Vernetzung ist ein Abgleiten der einzelnen Molekülketten erschwert und nach einer Belastung ist es dem Elastomer möglich wieder in seine Ausgangsform zurückzukommen. Es sind Längungen von bis zu 300% der Ausgangsform möglich ohne einen plastische Verformung zu provozieren (Weißbach et al., 2015, S. 390ff.).

Bei Belastung auf ein Polymer verformen sich die gesamten Molekülketten und nicht ein einzelnes Molekül. Darüber hinaus können Molekülketten gegeneinander „verrutschen“ (Weißbach et al., 2015, S.366). Bei der Verformbarkeit spielt die Temperatur, der die Polymere ausgesetzt sind, eine große Rolle, da diese die chemische Vernetzung der Molekülketten beeinflusst (Weißbach et al., 2015, S.366).

In Abbildung 8 wird auf dem linken Graphen beispielhaft das Reaktionsverhalten von Baustahl bei einem Zugversuch abgebildet. Man kann gut den zuerst linearen Anstieg der Kurve erkennen. In diesem Bereich verhält sich die Zugprobe elastisch und wird bei Entfernen der Zugkraft wieder in seine Ursprungsform zurückgehen. Diesen Bereich nennt man den Hook'schen Bereich, sprich den elastischen Bereich (Weißbach et al., 2015, S.505).

Bei Polymeren hingegen bestimmt die Verbindungsart das Spannungs-Dehnungs-Diagramm. In dem rechten Graphen in Abbildung 8 werden verschiedene Arten von Kunststoffen gegenübergestellt, von spröden (Linie 1) bis hin zu gummielastischen (Linie 4) Polymeren. Man erkennt, dass es meist nicht möglich ist den Startpunkt der plastischen Verformung klar zu determinieren, darüber hinaus ist das Verhalten nach Entfernen der Last meist nicht linear. Dieses Verhalten ist mit der Verbindung der Molekülketten, die in Wechselwirkung zu den umliegenden Ketten stehen und so eine Rückstellkraft ausüben können, zu erklären (Weißbach et al., 2015, S.368f.).

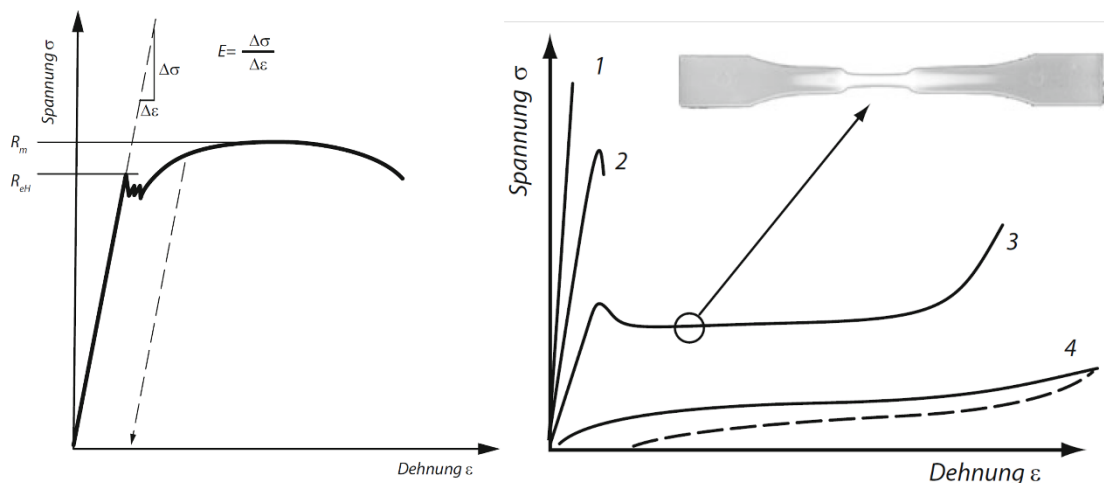


Abbildung 8: Vergleich E-Modul (Weißbach et al., 2015, S.505 & S.369)

Für Schuhmittelsohlen werden gummielastische Polymere verwendet, da sich nach Entfernung der Druck- oder Zuglast die Verformung weitgehend wieder zurückbilden soll.

Durch das Aufschäumen des Elastomers zum Schaummaterial für die Mittelsohle entsteht eine wabenförmige Struktur der Molekülketten die einen Lufteinschluss begrenzen.

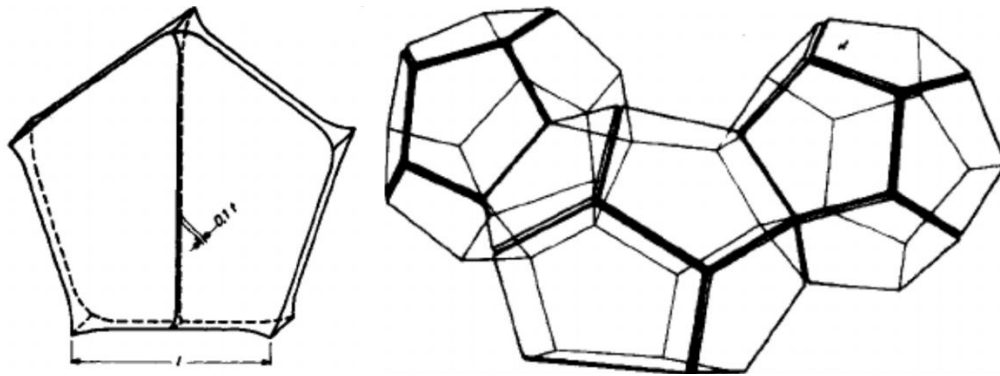


Abbildung 9: Schaumzelle und Schaumzellenverbund (Menges & Knipschild, 1975, S.623)

Diese, in Abbildung 9 zu erkennenden, Wabenstrukturen bilden mit den umliegenden Waben einen Schaumzellenverbund, der das typische Erscheinungsbild eines Schaummaterials induziert (Menges & Knipschild, 1975, S. 623).

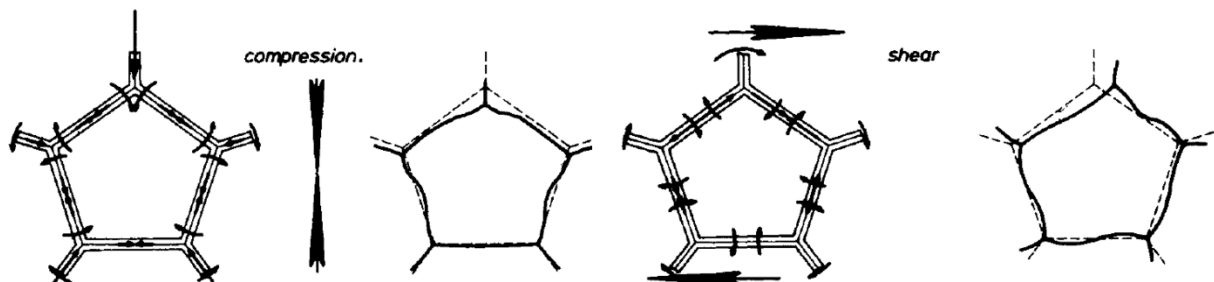


Abbildung 10: Kompression Schaumzelle (Menges & Knipschild, 1975, S.624)

Beim Laufen wird der Schaumzellenverbund einer Schuhmittelsohle Kompressions- und Scherkräften ausgesetzt. In Abbildung 10 ist das Reaktionsverhalten einer einzelnen Schaumwabe schematisch dargestellt.

Unter einer submaximalen zyklischen Last werden nach mehrmaliger Belastung die Verbindungsstege zwischen den Wabenstrukturen, durch Mikrorisse, an Elastizität verlieren. Dieser Vorgang vermindert die Gesamtelastizität des Schaummaterials, bis hin zum irreversiblen Formverlust (Menges & Knipschild, 1975, S. 623ff.).

Durch Variation der Dichte des verwendeten Schaumes kann die Anzahl der Verbindungsstege beeinflusst werden und damit ebenfalls die Dauerfestigkeit des Materials. Allerdings wird bei einer erhöhten Dichte weniger Luft in den Waben eingeschlossen, dies hat Auswirkung auf das Dämpfverhalten des Schuhs und auf sein Gesamtgewicht (Menges & Knipschild, 1975, S. 624).

Wie zuvor erwähnt zeichnet sich das bei Sportschuhen verwendete Polymer durch sein gummielastisches Verhalten aus. Dies bedeutet, dass nach Entfernen einer Last das Polymer sich wieder in seine Ausgangsform zurückbegibt. Dies geschieht, wie ebenfalls bereits erwähnt, nicht linear, wegen der nicht geordneten Orientierung und der durch die ungleichmäßige Anordnung unterschiedlich auf benachbarte Ketten einwirkenden Elektronenpaarbindungen. Wenn man dies graphisch darstellt entsteht eine, für diese Form der Materials typische, Hystereseform. Dies bedeutet, dass das Material, graphisch betrachtet, nicht denselben Verformungsweg unter Belastung wie unter Entlastung zurücklegt. Dadurch entsteht ein Graph der eine Fläche umfasst (siehe Abbildung 11) und keine offene Linie (Ohrndorf, 2011, S.14ff.).

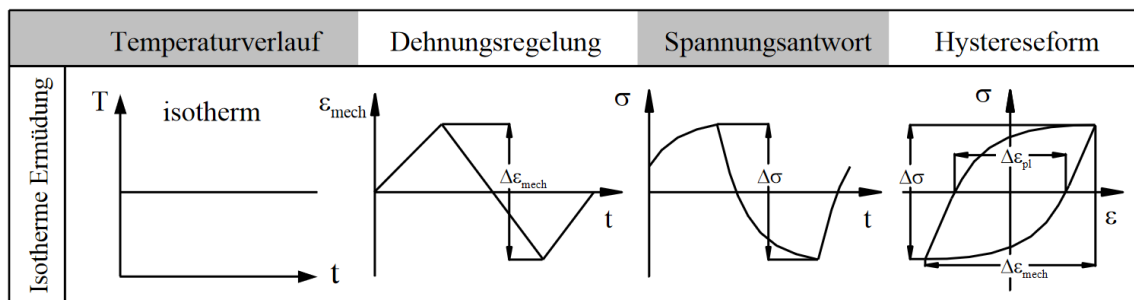


Abbildung 11: Reaktionsverhalten verschiedener Ermüdungsversuche (Ohrndorf, 2011, S.17)

Die typische Hystereseform der Kraftdeformationskurve entsteht indem man den ermittelten Graph, der die Dehnung in Abhängigkeit der Zeit darstellt, und den Graphen, der die Spannung in Abhängigkeit der Zeit darstellt, übereinander projiziert und somit den Faktor der Zeit eliminiert (Ohrndorf, 2011, S.18).

In Abbildung 12 ist zu erkennen, wie dieses gerade beschriebene Verfahren umgesetzt wird, und ein, von der Zeit unabhängiger, Graph erstellt wird, der die aufgebrauchte Spannung und die daraus resultierende Dehnung visualisiert.

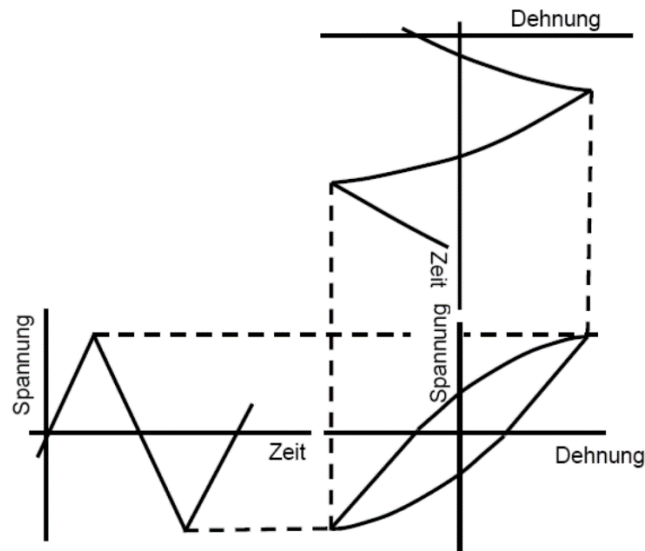


Abbildung 12: Entstehung der hystereseförmigen Kraftdeformationskurve (Ohrndorf, 2011, S.18)

Aus der aufgezeichneten Kraftdeformationskurve kann man einige interessante Merkmale des Materials entnehmen. Neben der offensichtlichen Dehnung und der dazu korrelierenden Spannung kann man an Hand der Steigung der Kraftdeformationskurve die Festigkeit des Materials bestimmen. Je steiler die Kurve ansteigt desto fester verhält sich das Material. Darüber hinaus lässt sich mit Hilfe der Berechnung des Flächeninhaltes unterhalb der Kurve die aufgenommene und die abgegebene Energiemenge bestimmen. Dabei ist besonders interessant die Differenz der Energien zu betrachten, diese stellt einen Energieverlust dar, der in Wärme und Deformation dissipiert.

2.3 Datenfindung zum Einstellen der Prüfmaschine

2.3.1 Bodenreaktionskraftkurven

Nach Newtons dritten Gesetz „actio = reactio“ erzeugt jede Kraft eine vom Betrag ebenso große, entgegengesetzt wirkende, Gegenkraft (Gross et al., 2012, S.38). Durch dieses Naturgesetz ist man in der Lage, wenn man z.B. die im Fuß eines Läufers auftretenden Kräfte quantifizieren möchte, diese über eine indirekte Messung (die Messsensoren werden nicht in den Läufer implantiert) über Kraftmessplattenmessplatten (engl. force plates) am Boden zu ermitteln.

Kraftmessplatten arbeiten meist mit Dehnmessstreifen oder Piezokristallen (Bobbert & Schamhardt, 1990, S.705).

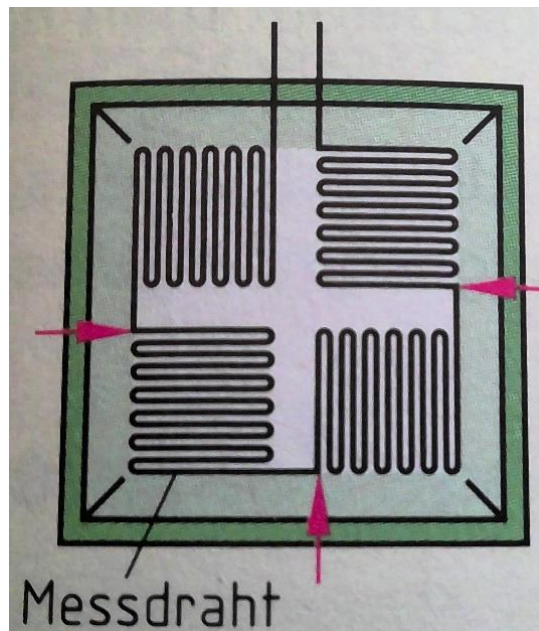


Abbildung 13: Dehnmessstreifen (Tkotz & Bastian, 2008, S.178)

Dehnmessstreifen (folgend DMS), siehe Abbildung 13, sind feine, meist auf Trägerfolien angebrachte Drähte. Wenn sich das Objekt, auf den der DMS aufgeklebt worden ist, durch Einwirken äußerer Kräfte plastisch verformt, wird der DMS ebenfalls verformt. Dadurch verändert sich der Durchmesser und die Länge des Drahtes, was zur Folge hat, dass sich die elektrische Leitfähigkeit verändert. Ein an den DMS angeschlossenes Messgerät nimmt diese Veränderung wahr und kann die spezifisch veränderte Leitfähigkeit in eine Deformation, z.B. in Millimeter, umrechnen (Tkotz & Bastian, 2008, S.178).

Piezokristalle sind Quarze, die bei Krafteinwirkung von außen, eine elektrische Spannung erzeugen. Mit Hilfe einer an die Kristalle angeschlossenen Messeinheit ist es möglich die entstandene Spannung in eine äquivalente Kraft umzurechnen, siehe Abbildung 14.

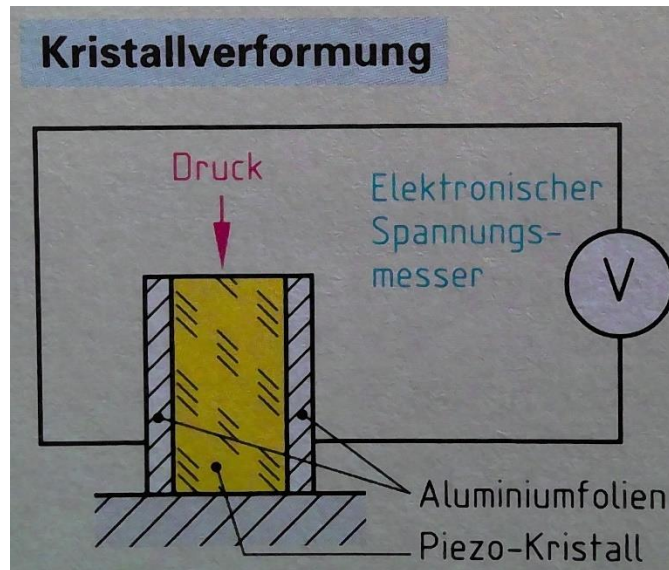


Abbildung 14: Prinzip eines Piezokristalls (Tkotz & Bastian, 2008, S.178)

Diese Kraftmessplatten werden dann so angebracht, dass diese sich auf einer Höhe mit der Laufstrecke befinden auf der der Proband läuft. Wenn der Proband dann beim Laufen auf die Kraftmessplatte tritt wird diese verformt und die Messeinheit kann die auf die Platte einwirkende Kraft quantifizieren. Wenn man sich lediglich auf eine dieser Platten stellen würde, würde diese wie eine Waage funktionieren und das Körpergewicht des Probanden angeben (Novacheck, 1998, S.84f.).

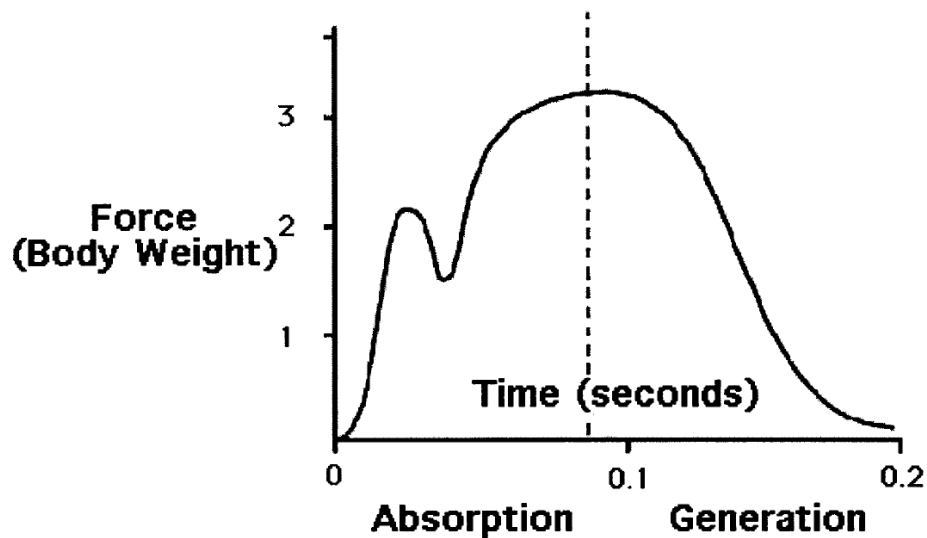


Abbildung 15: Typische Bodenreaktionskraftkurve (Novacheck, 1998, S.85)

In Abbildung 15 ist eine klassische Bodenreaktionskraftkurve (folgend BRK) zu erkennen, die entsteht wenn ein Proband über eine Kraftmessplatte läuft.

Da, wie in Kapitel 2.1.2 beschrieben, der Mensch beim Gehen mit nahezu gestrecktem Knie die Ferse auf den Boden aufsetzt, steigt die Kraftkurve zu Beginn schnell stark an, da keine federnden Elemente den Aufprall schützen. Beim folgenden Absenken der Kraftkurve liegt der Fuß flach auf dem Boden auf und das Körpergewicht des Läufers senkt sich ab. Dadurch fällt die auf die Kraftmessplatte wirkende Kraft ab. Anschließend bremst der Läufer das Absenken des Körperschwerpunktes ab und geht dazu über Kraft für das erneute Abstoßen zu generieren. Deswegen steigt die Kraftkurve erneut (Beckers & Deckers, 1997, S.18ff.).

Wenn ein Proband im Stile eines Vofußläufers sich fortbewegt entsteht eine, wie in Abbildung 16 zu erkennende, veränderte BRK.

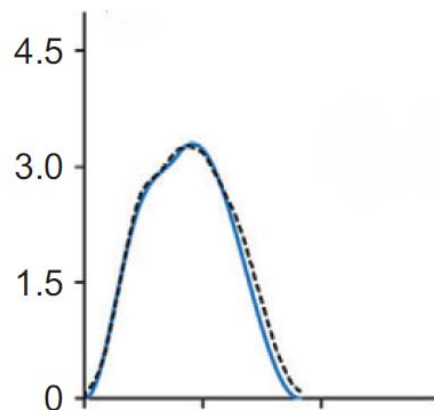


Abbildung 16:BRK Vorfußläufer (Clark et al., 2014, S.2038)

Bei diesem Kraftverlauf fehlt das erste Kraftmaximum, dies ist in dem fehlenden Fersenkontakt mit dem Boden begründet.

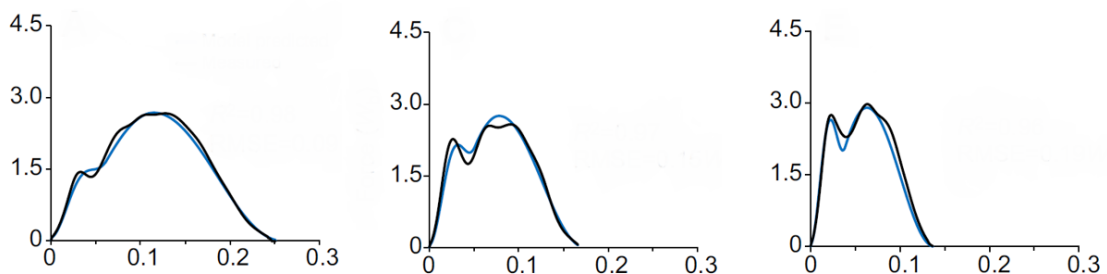


Abbildung 17: BRK in Abhängigkeit von Geschwindigkeiten (Clark et al., 2016, S.252)

Abbildung 17 ist einer Studie von Clark et al. entnommen worden, diese haben es sich zur Aufgabe gemacht ein Modell zu entwickeln, welches die BRK eines Läufers

simulieren kann. Die blaue Linie in den Diagrammen stellt die Simulation und die schwarze stellt die Messung dar (Clark et al., 2016, S.247f.). In Abbildung 17 lässt sich gut erkennen, wie sich die BRK in Abhängigkeit zur Geschwindigkeit des Läufers verändert, diese nimmt von links $3 \frac{m}{s}$ nach rechts auf $8 \frac{m}{s}$ zu. Dabei wird die Bodenkontaktzeit jedes Mal geringer und die Kraftmaxima steigen. Dies ist der, in Kapitel 2.1.2 beschriebenen, gesteigerten Fallhöhe und der verstärkten Muskelrekrutierung geschuldet.

Zu Beginn dieser Arbeit war die Anfangsintention aus den gesammelten BRK einen Durchschnitt für jedes Kraftmaxima der Belastungsphasen abzulesen und diese Werte zum Einstellen der Prüfmaschine zu verwenden. Allerdings sind Im Laufe der Recherche Faktoren aufgetreten die diese Vorgehensweise zunichte gemacht haben. Zum einen war die Grundidee an Hand des genau definierten Gangzyklus, vergleiche Kapitel 2.1.2, und der damit zusammenhängenden Vorhersagbarkeit, zu welchem Zeitpunkt nur die Ferse und wann lediglich der Vorfuß Kontakt zum Boden haben, die unabhängigen Kraftmaxima abzulesen. Da ein Laufschuh über die simulierte Laufstrecke eines Marathon getestet werden soll, ist der Gangzyklus nicht zu verwenden und dem Laufzyklus fehlt die nötige präzise Beschreibung um dieses Vorhaben umsetzen zu können. Darüber hinaus kommt es bei den BRK zu einer Überlagerung der Kräfte im Vor- und Rückfuß, siehe Abbildung 18. Dies ist ein weiterer Punkt der das geplante Vorgehen unmöglich macht.

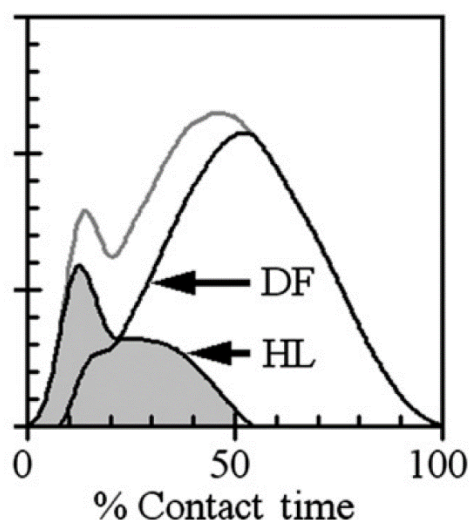


Abbildung 18: Überlagerung der Kräfte (Shorten & Mientjes, 2011, S.46)

In Abbildung 18 stellen Shorten und Mientjes mit der hellgrauen Linie die Gesamtbodenreaktionskraftkurve dar. Mit Hilfe der grau ausgefüllten Kurve wird die BRK für den Rückfuß (HL = heel) und mit der schwarzen Kurve für den Vorfuß (DF = distal foot) dargestellt (Shorten & Mientjes, 2011, S. 46). Man kann klar erkennen, dass sich ab ca. 10% der Gesamtkurve die beiden anderen aufaddieren und somit ein präzises ablesen unmöglich wird.

Um ein weiteres Vorgehen zu ermöglichen wurde nach einer alternativen Datenquelle recherchiert. Diese wird im folgenden Kapitel erläutert.

2.3.2 Datenanalyse

Nachdem in Kapitel 2.3.1 erkannt worden ist, dass die geplante Vorgehensweise nicht zielführend ist, wurden die Rohdaten einer Masterthesis mit dem Namen „Die Interaktion zwischen Mensch und Untergrund für den Vor- und Rückfuß beim Laufen“ verwendet, welche im Jahr 2013 von Alexis Brinkemper an der Sporthochschule Köln verfasst wurde.

In der erstellten Studie mussten 20 Probanden mit einer Geschwindigkeit von $3,5 \frac{m}{s}$ über einen mit Kraftmessplatten präparierten Laufsteg laufen.



Abbildung 19: Aufbau Laufsteg (Alexis Brinkemper, 2013, S.18)

In Abbildung 19 ist die Besonderheit dieser Konstruktion zu erkennen. Es wurden für diese Untersuchung mehrere Kraftmessplatten übereinander positioniert. Als Grundlage dient eine große Kraftmessplatte, diese generiert dieselben Resultate wie es die Studien im vorherigen Kapitel haben. Darüber hinaus wurden zwei kleinere Platten hintereinander auf der größeren angeordnet. Diese kleineren Platten wurden so angebracht, dass die Eine für den Rückfußbereich Messergebnisse liefert und die Andere für den Vorfußbereich (Alexis Brinkemper,

2013, S.12ff.). Die Probanden wurden angewiesen bei jedem Lauf ihre Ferse möglichst nahe der Kante der hinteren oberen Kraftmessplatten zu positionieren.

In Abbildung 20 ist eine Visualisierung der erhobenen Daten zu sehen. Man erkennt von links nach rechts fortlaufend, wie sich das mit visuellen Markern versehene Bein des Probanden der Messapparatur nähert. In den nächsten Bildern wird dargestellt, wie die Ferse zuerst Kontakt auf der hinteren Platte generiert und anschließend über die Messplatten abrollt und den Kontakt wieder beendet.

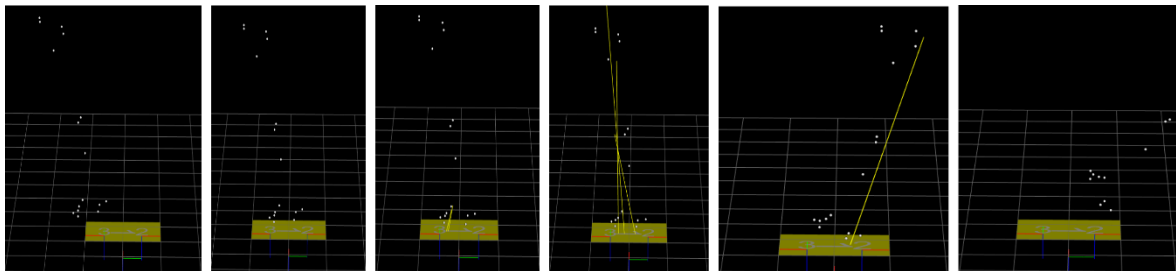


Abbildung 20: Visualisierung der gemessenen Daten

Die gelben Linien stellen den betragsmäßigen Kraftvektor der auf die Messeinheiten einwirkt mit deren Orientierung im Raum dar.

Damit es möglich wurde diese Daten zu verwenden beziehungsweise auszuwerten ist ein Matlabskript verfasst worden. Die verwendeten Rohdaten waren vor Bearbeitungsbeginn bereits bereinigt worden, sprich jeder außerhalb der Belastungsphasen ist als 0 definiert worden. Darüber hinaus waren die Messergebnisse unbearbeitet. Das erstellte Skript ist im Anhang zu finden.

Abbildung 21 zeigt die Ergebnisse, die die Analyse mit Hilfe von Matlab generiert worden sind. Das obere linke Balkendiagramm zeigt die jeweiligen Kraftmaxima für den jeweiligen Vor- bzw. Rückfuß. Im Detail bedeutet dies, die maximal auftretende Kraft im Rückfuß beträgt das 1,49-fache des Körpergewichtes. Der Vorfuß wird maximal mit dem 2,32-fache des Körpergewichtes belastet.

2 Literaturübersicht

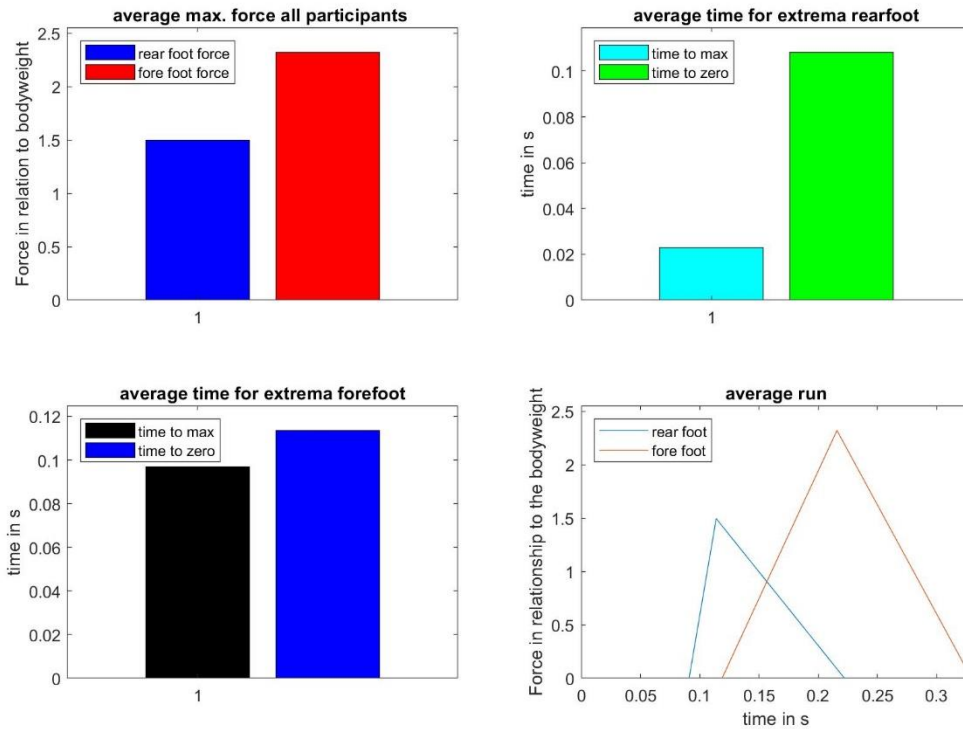


Abbildung 21: Ergebnisse der Matlabauswertung

Die folgenden zwei Diagramme zeigen die Zeiten, die benötigt werden um das Kraftmaximum zu erreichen und um anschließend wieder auf Unbelastet abzufallen. Für den Rückfuß bedeutet dies, dass innerhalb von 0,02 Sekunden die maximale Last erreicht wird und folgend in 0,11 Sekunden wieder auf ein Nullniveau absinkt. Für den Vorfuß lauten die Zeiten 0,09 und 0,11 Sekunden.

In dem letzten Graphen kann man einen durchschnittlichen Kraftverlauf aller Läufer erkennen mit den jeweiligen Zeiten und den auftretenden Kräften.

2.3.3 Schrittfrequenzen

Um die in 2.3.2 ermittelten Kraftmaxima und die Anstiegs- bzw. Abfallzeiten an der Prüfmaschine (diese wird in 3.1 genauer beschrieben) zyklisch einstellen zu können wird zusätzlich noch eine genaue Anzahl an Schritten pro Sekunde benötigt.

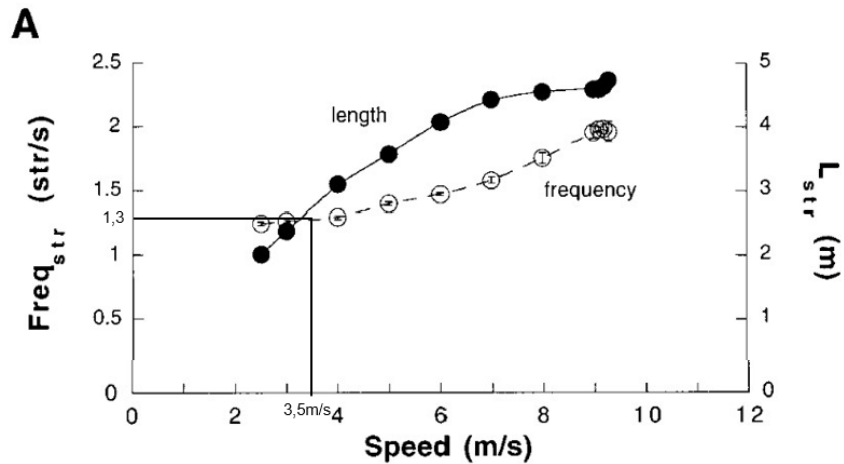


Abbildung 22: Schrittfrequenz I (Weyand et al., 2000, S.1993)

Abbildung 22 stammt aus einer Studie, die mit 33 Studenten an der Harvard Universität durchgeführt worden ist, um einen Zusammenhang zwischen der maximal aufgebrauchten Bodenreaktionskraft und der Höchstgeschwindigkeit zu identifizieren. Für die in dieser Arbeit verwendete Durchschnittsgeschwindigkeit liefert diese Abbildung eine durchschnittliche Frequenz von $1,3 \frac{\text{Schritte}}{\text{Sekunde}}$.

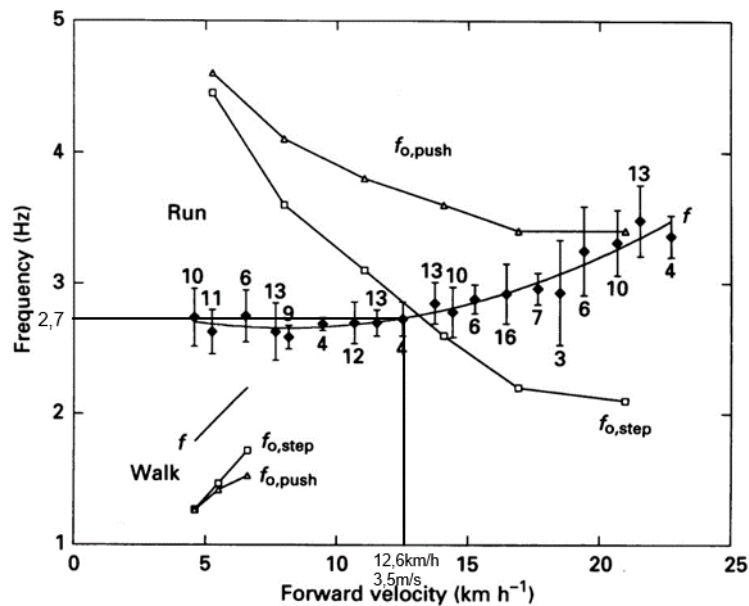


Abbildung 23: Schrittfrequenz II (Cavagna et al., 1991, S.86)

Dem gegenüber hat Cavagna bereits 1991 mit seinen Kollegen die Limitierungen bei der maximal zu erreichenden Laufgeschwindigkeit bei Athleten untersucht. Bei dieser Untersuchung wurde, wie in Abbildung 23 zu erkennen ist, eine durchschnittliche Schrittfrequenz von $2,7 \frac{\text{Schritten}}{\text{Sekunde}}$ bei einer Geschwindigkeit von $12,6 \frac{\text{km}}{\text{h}}$ ermittelt. Dies entspricht einer Geschwindigkeit von $3,5 \frac{\text{m}}{\text{s}}$.

Um die auf den ersten Blick auffällige Diskrepanz der beiden Frequenzen zu relativieren ist nach weiteren Studien gesucht worden.

Die Studie, der Abbildung 24 entnommen worden ist, hat die Variabilität der Schrittfrequenz in Abhängigkeit des Untergrundes, auf dem der Läufer sich befindet, verglichen. Für das Testen der Mittelsole der Laufschuhe wurde sich gegen die ermittelten Werte auf einem Laufband entschieden, da die verarbeiteten Daten der Masterthesis von Alexis Brinkemper auf einer „Laufstegkonstruktion“ (Alexis Brinkemper, 2013, S.18) erhoben worden sind. Auf Grund dessen wurde die hellgraue Kurve aus Abbildung 24 verwendet, diese repräsentiert das Laufen auf dem Boden (OG SF = on ground stepfrequency).

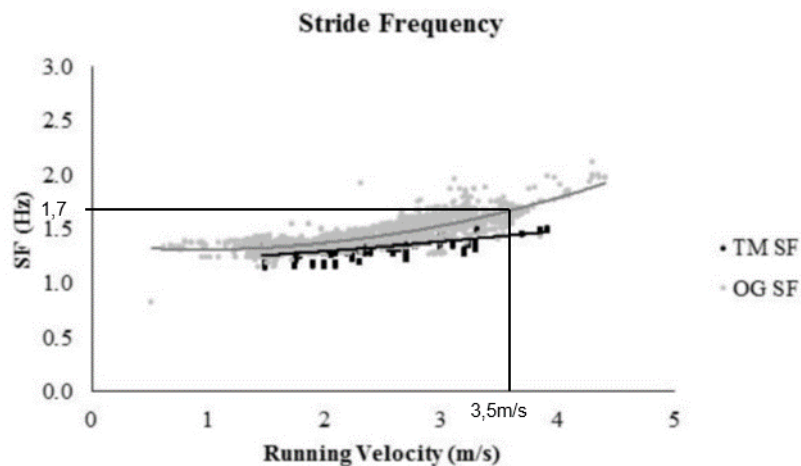


Abbildung 24: Schrittfrequenz III (BAILEY et al., 2017, S.1071)

Um eine allgemein gültige Aussage zu erhalten wurde ein Durchschnitt der drei verwendeten Studien ermittelt.

$$\frac{(1,3 \frac{\text{Schritte}}{\text{Sekunde}} + 2,7 \frac{\text{Schritte}}{\text{Sekunde}} + 1,7 \frac{\text{Schritte}}{\text{Sekunde}})}{3} = 1,9 \frac{\text{Schritte}}{\text{Sekunde}} \quad (1)$$

Dies entspricht einer Zeit pro Schritt von:

$$\frac{1 \text{ Schritt}}{1,9 \frac{\text{Schritte}}{\text{Sekunde}}} = 0,526 \text{ Sekunden} \quad (2)$$

Diese Werte beziehen sich jeweils auf die Zeiten die zwischen dem Bodenkontakt beider Fersen liegen, nicht jedoch auf die Zeit die es benötigt, bis die Ferse des selben Fußes erneut Kontakt mit dem Boden aufnimmt. In der Prüfmaschine soll jeweils nur ein Schuh zur selben Zeit untersucht werden, siehe Kapitel 3.1. Um dies zu ermöglichen müssen die ermittelten Frequenzen auf lediglich einen Fuß umgerechnet werden.

Daraus resultiert, dass sich die Schrittfrequenz von $1,9 \frac{\text{Schritte}}{\text{Sekunde}}$ auf $0,95 \frac{\text{Schritte}}{\text{Sekunde}}$ halbiert und sich dementsprechend die Zeit die pro Schritt benötigt wird von $0,53s$ auf $1,06 s$ verdoppelt.

Mit Hilfe dieser Werte ist es möglich, die zur Durchführung der Messung ebenfalls wichtige Zeitspanne zu berechnen, in denen der Schuh keiner Last ausgesetzt wird. Diese kann ermittelt werden, indem von der Gesamtschrittzeit die Zeitspannen, die zu Be- und Entlastung benötigt werden, subtrahiert werden.

Darauf folgt:

Zeit unbelastet Rückfuß:

$$1,503s - (0,023s + 0,108s) = 1,372s \quad (3)$$

Zeit unbelastet Vorfuß:

$$1,503s - (0,097s + 0,114s) = 1,292s \quad (4)$$

2.3.4 Körpergewicht

Die in Kapitel 2.3.2 ermittelten Kraftmaxima beziehen sich auf das individuelle Körpergewicht der jeweiligen Probanden. Damit ein für die Allgemeinheit gültiges Testergebnis hervorgebracht werden kann musste ein für den durchschnittlichen Europäer geltendes Durchschnittskörpergewicht ermittelt werden.

Eine 2012 von Walpole veröffentlichte Studie beschäftigt sich mit dem ansteigenden Körpergewicht der Weltpopulation und der daraus resultierenden Aufgabe der Versorgung dieser mit genügend Nährstoffen. Sie fanden heraus, dass der durchschnittliche Europäer „70,8 kg“ (Walpole et al., 2012, S.3) wiegt.

Damit dieser Wert in die Prüfmaschine implementiert werden kann wurde dieser in Newton umgerechnet:

$$70,8 \text{ kg} * 9,81 \frac{\text{m}}{\text{s}^2} = 694,584 \text{ N} \quad (5)$$

2.3.5 Prüfzeit

Die in Kapitel 2.3.2 analysierten Rohdaten wurden in der Untersuchung von Herrn Brinkmann bei einer Geschwindigkeit von $3,5 \frac{\text{m}}{\text{s}}$ aufgenommen. Damit ein repräsentatives Ergebnis in dieser Arbeit möglich wird, muss die Prüfmaschine so lange laufen, wie es dauern würde, einen Marathon mit der vorgegebenen Durchschnittsgeschwindigkeit zu absolvieren. Daraus resultiert eine Gesamtprüfzeit für eine von „42,195 km“ (Van den Berghe et al., 2021, S.9) von:

$$\frac{42195 \text{ m}}{3,5 \frac{\text{m}}{\text{s}}} = 12055,71 \text{ s} \quad (6)$$

Dies entspricht einer Zeit von drei Stunden und 21 Minuten.

2.3.6 Zusammenfassung der Werte

Abschließend werden alle für die Messung benötigten Werte zusammengefasst.

Tabelle 1: Zusammenfassung der mit Hilfe von Matlab ausgelesenen Werte

Fußareal:	Kraftmaximum in Newton:	Zeit steigend in Sekunden:	Zeit fallend in Sekunden:
Rückfuß	1 038,40	0,023	0,108
Vorfuß	1 612,82	0,097	0,114

Tabelle 2: Zusammenfassung der recherchierten Prüfungsmerkmale

Prüfungsmerkmal:	Wert:
Gesamtprüfzeit	3:21 Stunden
Frequenz	$0,95 \frac{\text{Schritte}}{\text{Sekunde}}$
Zeit pro Schritt	$1,053 \text{ Sekunden}$
Zeit unbelastet Rückfuß	$1,372s$
Zeit unbelastet Vorfuß	$1,292s$

3 Methode

3.1 Untersuchungsdesign

Bei dieser Messung handelt es sich um eine Materialprüfung, basierend auf Querschnittsdaten. Sowohl bei den aus „Die Interaktion zwischen Mensch und Untergrund für den Vor- und Rückfuß beim Laufen“ entnommenen Rohdaten als auch den verwendeten Schrittfrequenzen und aller daraus resultierenden Prüfungscharakteristika handelt es sich um Durchschnittswerte. Diese wurden verwendet um ein möglichst allgemein gültiges Ergebnis zu generieren.

Mit Hilfe dieser Daten wurde versucht den Laufschuh einer möglichst realitätsnahen Annäherung der Belastungscharakteristika auszusetzen.

Ziel dieser Analyse soll sein das Materialverhalten der Mittelsohle eines Laufschuhs, im Vor- und Rückfuß getrennt voneinander, über eine simulierte Marathondistanz zu dokumentieren.

3.2 Versuchsaufbau

Als Beispielschuh wurde in diesem Versuch ein Modell „Ghost“-Serie von der Sportschuhmarke „Brooks“ verwendet. Es ist die elfte Version des Schuhs gewählt worden. Laut Herstellerangaben hat diese Laufschuhserie eine „DNA-Loft“-Dämpfung, diese ist die „softeste“ Art der Dämpfung über die „Brooks“ verfügt (Brooks, 2021). Die Mittelsohle besteht aus einem „EVA-Schaum, Gummi und Luft“- (Brooks, 2021) Gemisch und ist, laut Hersteller, in der Lage die aufgenommene Energie wieder gut zurückzugeben (Brooks, 2021).

Für die Materialprüfung der Laufschuhmittelsohle wurde die elektro-dynamische Prüfmaschine LTM10 von Zwick/Roell verwendet (Abbildung 25).

Die verwendete Prüfmaschine eignet sich laut Hersteller sehr gut für dynamische „Ermüdungs- und Lebensdauerversuche“ (Zwick/Roell, S.1). Die Maschine ist in der Lage dynamisch eine maximale Kraft von 10 kN , bei einem maximalen Kolbenhub von 60 mm , mit einer Frequenz von maximal 100 Hz auszuführen. Diese

Rahmenbedingungen sind für das Testen der Laufschuhmittelsohle mehr als ausreichend (Zwick/Roell, S. 3).

Das Prüfprotokoll kann abschließend zu jeder Messung als Textdatei ausgegeben werden, mit Hilfe von Matlab soll diese Datei analysiert werden. Das Vorgehen für das erstellte Matlabskript wird in Kapitel 3.3 beschrieben.



Abbildung 25: Zwick/Roell LTM10

Wie in Abbildung 25 zu erkennen, ist auf der roten Messeinheit eine hölzerne Grundplatte montiert. Die zwei Metallplatten die sich zwischen der Grundplatte und der Messeinheit befinden, sind so gelagert, dass eine Verschiebbarkeit in zwei orthogonal zueinander stehenden Achsen, von einigen Millimetern möglich ist. Diese Verschiebung wird in jedem Prüfungszyklus beansprucht, da der

asymmetrische Aufbau der Sohle des Schuhs ein „Setzen“ des Schuhs bei Belastung induziert.

Der soeben beschriebene Aufbau ist in Abbildung 26 erneut zu sehen. Darüber hinaus kann man den Prüfdorn, der die Kraft von der Maschine auf den Schuh überträgt erkennen. Hierbei handelt es sich um einen gedrehten Zapfen mit einem Durchmesser von ca. 20 mm.

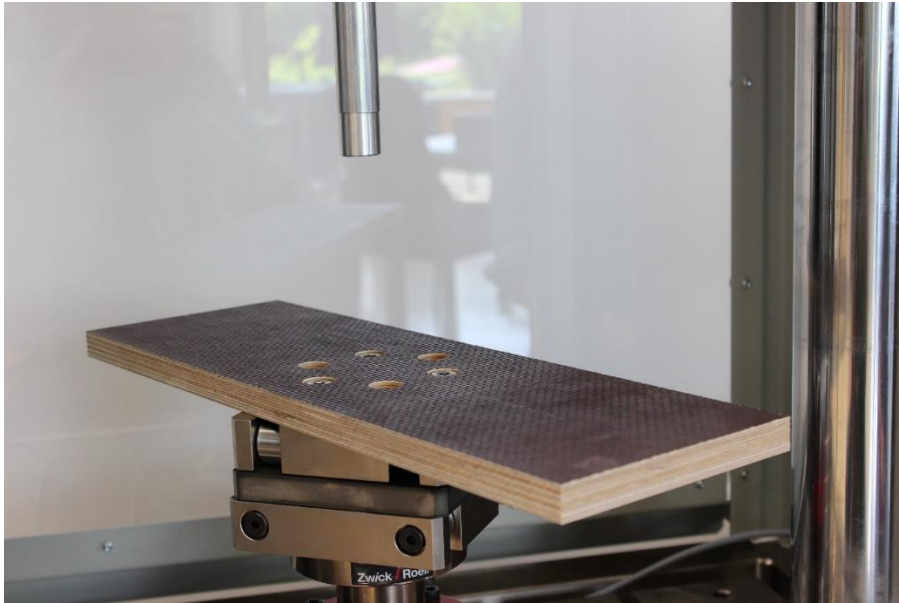


Abbildung 26: Grundplatte und Prüfdorn

Damit die aufgebrachte Kraft nicht in einem zu kleinen Radius auf die Sohle des Schuhs drückt und diesen eventuell zerstören würde, wurde ein, für die Größe des Schuhs passender, Leisten verwendet. Neben der Schonung des Materials entspricht die Kraftverteilung durch den verwendeten Leisten eher einer realen Belastung welcher der Schuh ausgesetzt wird.

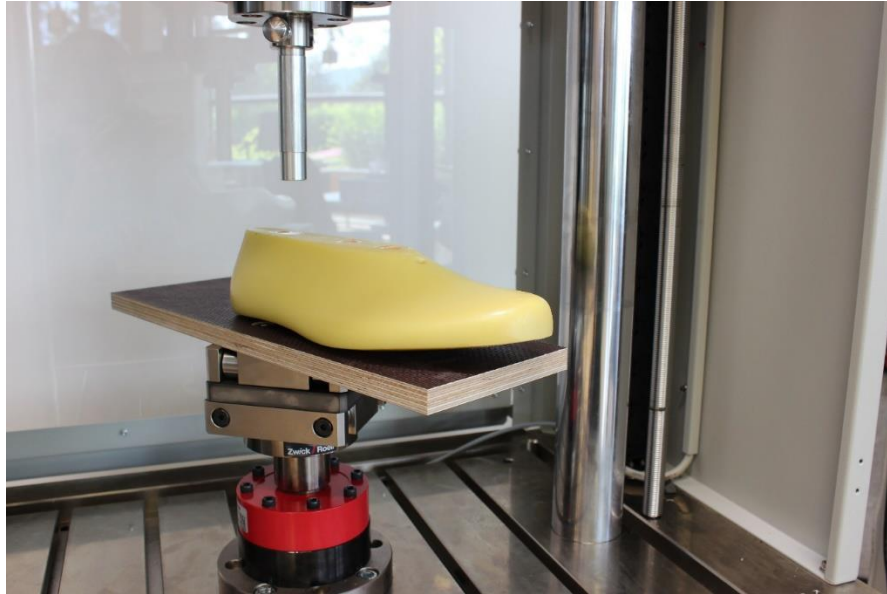


Abbildung 27: Leisten

Zur besseren Reproduzierbarkeit der Messung wurde der Kontaktpunkt des Prüfdornes und des, in Abbildung 27 zu erkennenden, Leisten markiert.



Abbildung 28: Eingespannter Schuh unbelastet

In Abbildung 28 wird der auf der Grundplatte positionierte Schuh abgebildet. Wie bereits die Position des Dornes ist die Position des Schuhs auf der Grundplatte ebenfalls markiert worden. In dem gezeigten Bild hat der Prüfdorn bereits Kontakt zum eingelegten Leisten, allerdings wird noch kein Druck ausgeübt.



Abbildung 29: Eingespannter Schuh im Rückfuß belastet

Abbildung 29 zeigt den Schuh im belasteten Zustand, hier sind einige Auffälligkeiten zu beobachten. Die gelagerten Metallplatten, auf denen die Grundplatte montiert ist, haben sich einige Millimeter gegeneinander verschoben. Darüber hinaus ist die Mittelsohle eingedrückt und die Fußspitze hat sich etwas angehoben. Durch diesen Umstand kam die Befürchtung auf, dass sich die Position, in der sich der Schuh über die gesamte Messzeit über befindet, verändern würde und das Ergebnis nicht repräsentativ sein würde. Diese Befürchtung hat sich als unbegründet herausgestellt, trotz der leichten Positionsänderung während der Belastung hat sich die Gesamtposition während der Messung nicht verändert.

Zur Veranschaulichung des Aufbaus wurden die Schnürsenkel des Schuhs nicht gebunden und die Lasche wurde nach vorne weggeklappt. Bei der Versuchsdurchführung wurde der Schuh mit einer Schleife verschlossen.

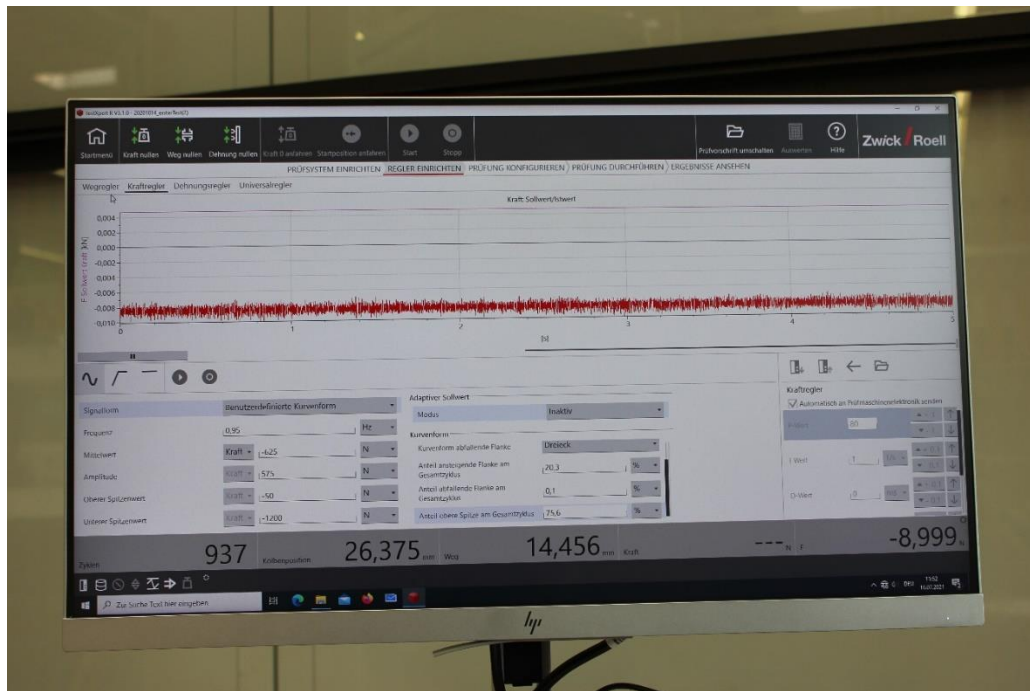


Abbildung 30: Programmoberfläche zum Einstellen der Prüfmaschine

Die in der Prüfmaschine voreingestellten Belastungstests entsprachen nicht der für die Durchführung dieser Messung nötigen asymmetrischen dreieckigen Belastungskurve. Aus diesem Grund wurde eine benutzerdefinierte Kurvenform entworfen, welche möglichst nahe an das gewünschte Belastungsziel heranreicht.

Nachfolgend sind die prozentualen Anteile der Be- und Entlastungskurve berechnet und tabellarisch zusammengefasst worden.

Anteil der ansteigenden Flanke des Rückfußes im Schrittzyklus:

$$\frac{0,023 \text{ s}}{1,053 \text{ s}} * 100\% = 2,18\% \quad (7)$$

Anteil der abfallenden Flanke des Rückfußes im Schrittzyklus:

$$\frac{0,108 \text{ s}}{1,053 \text{ s}} * 100\% = 10,26\% \quad (8)$$

Anteil der unbelasteten Zeit des Rückfußes:

$$\frac{1,053 \text{ s} - (0,023 \text{ s} + 0,108 \text{ s})}{1,053 \text{ s}} * 100\% = 87,56\% \quad (9)$$

Anteil der ansteigenden Flanke des Vorfußes im Schrittzklus:

$$\frac{0,097 \text{ s}}{1,053 \text{ s}} * 100\% = 9,21\% \quad (10)$$

Anteil der abfallenden Flanke des Vorfußes im Schrittzklus:

$$\frac{0,114 \text{ s}}{1,053 \text{ s}} * 100\% = 10,83\% \quad (11)$$

Anteil der unbelasteten Zeit des Rückfußes:

$$\frac{1,053 \text{ s} - (0,097 \text{ s} + 0,114 \text{ s})}{1,053 \text{ s}} * 100\% = 79,96\% \quad (12)$$

Tabelle 3: Zusammenfassung Messeinstellungen

Einstellungsparameter:	Anteil für den Rückfuß:	Anteil für den Vorfuß:
Ansteigende Flanke	2,18%	9,21%
Abfallende Flanke	10,26%	10,83%
Belastungsfrei	87,56%	79,96%

Diese berechneten Werte sind die theoretisch optimalen Einstellungen um das Laufen des Marathons mit einer durchschnittlichen Geschwindigkeit von $3,5 \frac{m}{s}$ zu analysieren. Allerdings war es durch die komplexe Einstellbarkeit des verwendeten PID-Reglers nicht möglich diese Messeinstellungen so zu übernehmen, dass die Prüfmaschine in der Lage gewesen wäre die Messung über den gesamten Zeitraum aufrecht zu erhalten. Aus diesem Grund wurden die errechneten Parameter durch gezielte Abänderungen so angepasst, dass die Kraftkurve der Prüfmaschine möglichst genau den errechneten Belastungswerten angeglichen wurde.

Die endgültig verwendeten Werte sind in Abbildung 30 abgebildet.

3.3 Datenverarbeitung

Zwischen der Fertigstellung der Datenfindung und der ersten Messung lag eine gewissen Zeitspanne. Um diese sinnvoll zu überbrücken wurde bereits vor dem Vorliegen der Messdaten, mit Hilfe einer beispielhaften Ergebnisdatei ein Matlabskript erstellt. Dies geschah in der Annahme, dass die Struktur der jeweiligen Ergebnisdateien identisch ist.

Allerdings hat sich, wie in Abbildung 31 deutlich wird, gezeigt, dass sowohl die Datenstruktur der Ergebnisse als auch der Startpunkt der Messungsaufzeichnung nicht übereinstimmen.

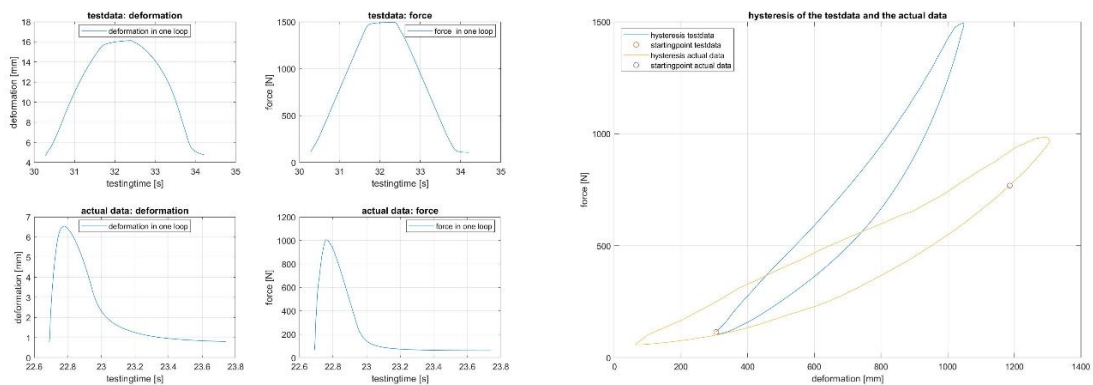


Abbildung 31: Datenstruktur im Vergleich

Deswegen war es erforderlich ein neues Matlabskript zu schreiben, das in der Lage war die Ergebnisdaten so zu ordnen und zu analysieren, dass die gewünschten Merkmale analysiert wurden. Dabei war es möglich Passagen des vorbereiteten Codes zu übernehmen.

Beide geschriebenen Skripte befinden sich im Anhang.

Der Laufschuh wurde zwei Messungen unterzogen. Beide Male im Rückfußbereich, für das erste Mal ist eine durchschnittliche Kraftspitze von $1038,4\text{ N}$ angestrebt worden. Für den zweiten Messdurchgang ist eine 20% geringere Maximalbelastung eingestellt worden.

Für den Vergleich beider Ergebnisse ist das vorhandene Matlabskript erneut angepasst worden. Dieses Skript befindet ebenfalls sich im Anhang.

3 Methode

In Kapitel 4.1.2 wird der erste Lauf vorgestellt, anschließend in Kapitel 4.1.3 der Vergleichsdurchgang und abschließend werden in Kapitel 4.1.4 beide Läufe miteinander verglichen.

4 Ergebnisse

4.1.1 Einführung Messergebnisse

Zum besseren Verständnis der folgenden Kapitel wird in diesem Abschnitt der Arbeit ein kurzer Überblick über die verwendeten Begriffe und das Vorgehen der Messauswertung gegeben.

In dem oberen linken Graphen der Abbildung 32 wird die Bedeutung des „rising-“ und des „falling part“ visualisiert. In Worten stellt der „rising part“ den Anstieg der Kraftverformungskurve von seinem Minimum zum Maximum dar. Der „falling part“ hingegen zeigt das Abfallen vom Maximalwert zum Minimalwert.

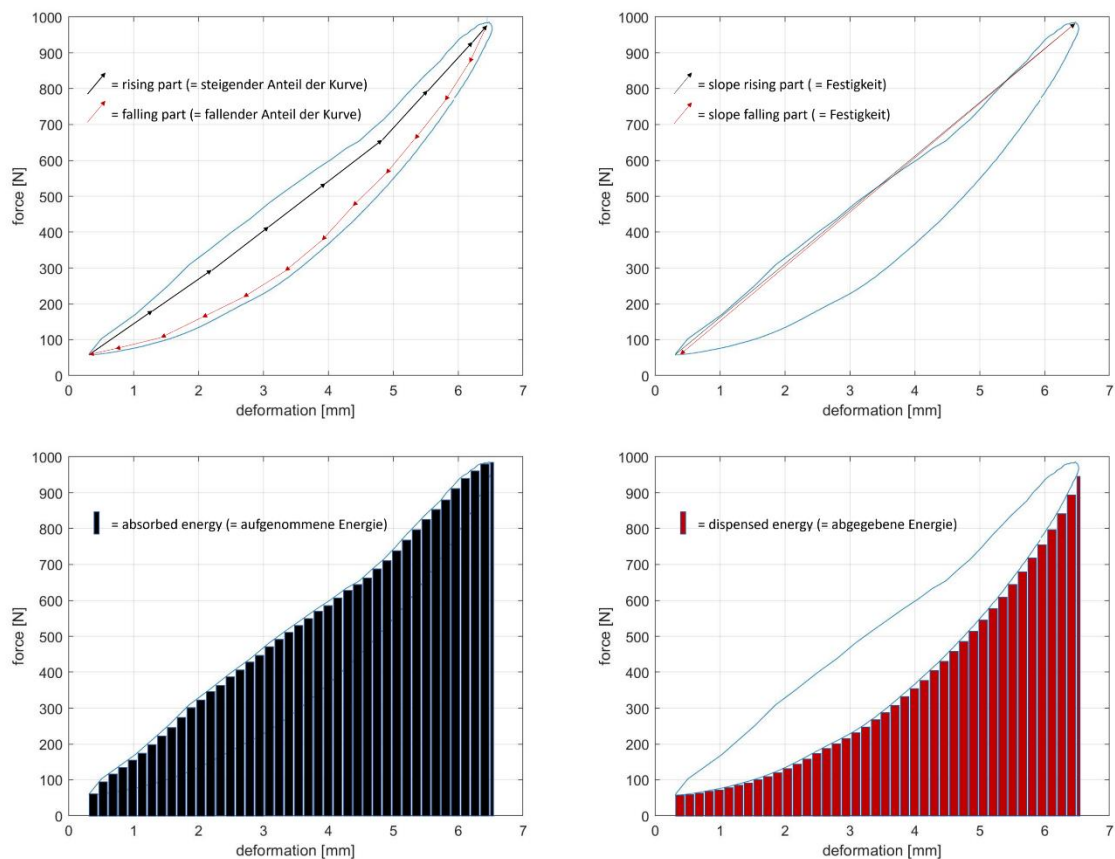


Abbildung 32: Übersicht der Auswertungsaspekte

Als erster wichtiger Analyseaspekt ist die Steigung (engl. slope) in dem oberen rechten Graphen zu erkennen. Diese wird mit Hilfe einer Geraden bestimmt, die durch das Minimum und Maximum eines jeden Messdurchgangs der Kraftdeformationskurve gelegt wird. Dabei wird einmal in die Steigung des „rising-“ und „falling part“ unterschieden, wobei der „falling part“ die Steigung vom Maximum

des Messdurchgangs $n-1$ zum Minimum des Messdurchgangs n beschreibt. Der „rising part“ visualisiert den Anstieg vom Minimum zum Maximum des selben Messdurchganges. Die Steigung repräsentiert die Festigkeit der Mittelsohle.

Darüber hinaus wird der Schuh in Bezug auf seine aufgenommene bzw. abgegebene Energie analysiert. Durch die Bestimmung des Integrals unterhalb der Kraftdeformationskurve wird die Menge der Energien bestimmt. Es wird jeweils in aufgenommene (engl. absorbed) und abgegebene (engl. dispensed) Energie unterschieden. Beides ist in den unteren beiden Graphiken der Abbildung 32 dargestellt worden.

Bei einigen Graphen werden die Ergebnisse sowohl mit numerischen als auch logarithmischen X-Achsen dargestellt. Dies dient zur besseren Visualisierung spezifischer Messabschnitte.

Der Schuh war zu keinem Zeitpunkt unbelastet, er wurde immer mit mindestens 60 N belastet. Bei allen nachfolgenden Graphen sind die Achsen so verschoben worden, dass 60 N einen unbelasteten und undeformierten Zustand darstellt. Dies dient der anschaulicheren Darstellung.

Auf Grund der nicht linearen Abnahme der Last (siehe Abbildung 32 in Kapitel 3.3) musste ein Abschnitt definiert werden, der ein annähernd lineares Verhalten vorweist. Da der Belastungsverlauf beim Laufen ungefähr dem eines Dreiecks entspricht wurde dies nötig. Deshalb wurde das Absinken von der Maximallast bis auf 100 N als absinkende Flanke der Kraftkurve definiert.

Für ein flüssigeres Lesen sind die Messergebnisse im Text auf zwei Stellen nach dem Komma gerundet worden, in den dazugehörigen Formeln sind die exakten Ergebnisse verwendet worden.

4.1.2 Erster Messdurchgang

Als Erstes wird ein Vergleich der Zeitspannen vorgestellt. In der folgenden Abbildung 33 wird jeweils die zuvor ermittelten und angestrebten Zeitwerte mit den bei der Messung umgesetzten Zeitwerten verglichen. Es wird immer zuerst die vorgegebene Zeit dargestellt.

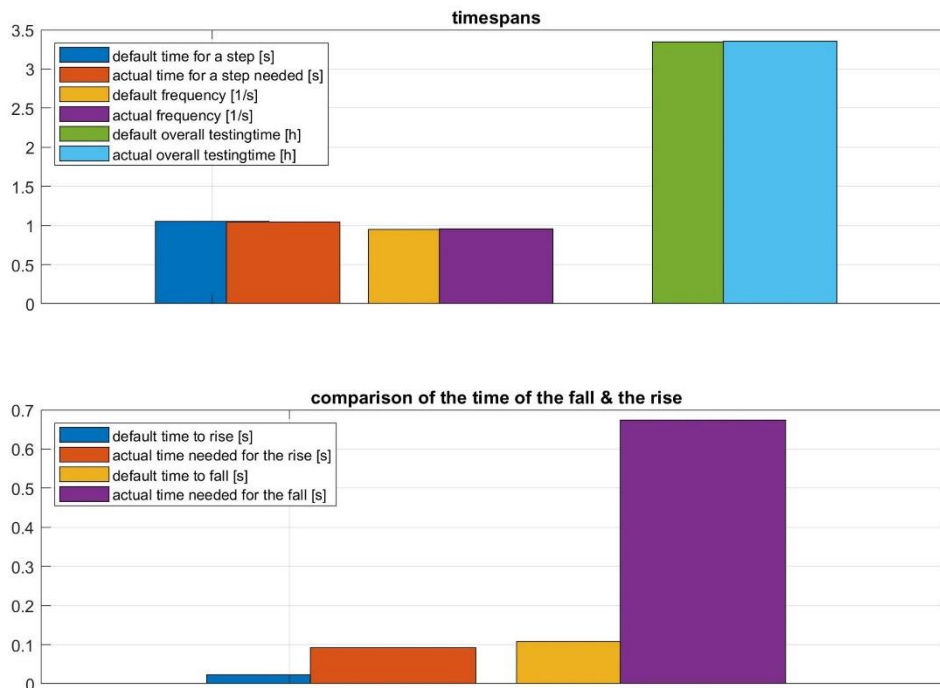


Abbildung 33: Ergebnisse der Zeitanalysen für die Messung mit 1000N

Die angestrebte Zeit die ein einzelner Schritt benötigen sollte betrug 1,05 s. Es war möglich die Maschine eine simulierte Schrittdauer von 1,04 s ausführen zu lassen.

Dies entspricht einer Genauigkeit von:

$$\frac{1,04291 \text{ s}}{1,053 \text{ s}} * 100\% = 99,04\% \quad (13)$$

Der darauf folgende Messwert zeigt die angestrebte Frequenz von $0,95 \frac{1}{s}$ und die umgesetzte Frequenz von $0,96 \frac{1}{s}$.

Dies entspricht einer Genauigkeit von:

$$\frac{0,95 \frac{1}{s}}{0,958859 \frac{1}{s}} * 100 = 99,07\% \quad (14)$$

Als letzter Punkt wird die Gesamtzeit der Messung in Stunden dargestellt. Die vorgesehene Zeitspanne beträgt 3,349 h die Gesamtmessdauer betrug 3,353 h.

Dies entspricht einer Genauigkeit von:

$$\frac{3,34881 h}{3,353361 h} * 100\% = 99,89\% \quad (15)$$

In dem unteren der beiden Graphen, der Abbildung 33, wird zuerst die Zeit dargestellt, die benötigt wird damit die Gesamtlast von dem geringsten Belastungszustand zu der maximalen Krafteinwirkung ansteigt. Diese Zeit wurde, mit Hilfe der Rohdaten der Masterthesis von Herrn Brinkemper, auf 0,023 s bestimmt. Tatsächlich umsetzen ließ sich mit der Prüfmaschine eine Zeitspanne von 0,092 s.

Dies entspricht einer Abweichung von:

$$\frac{0,0926745 s}{0,023 s} * 100\% = 402,93\% \quad (16)$$

Die Prüfmaschine sollte das Absinken der Maximallast auf 100 N in 0,11 s umsetzen. Für den Gesamtzeitraum der Messung betrug sie jedoch durchschnittlich 0,67 s .

Dies entspricht einer Abweichung von:

$$\frac{0,673886 s}{0,108 s} * 100\% = 623,96\%. \quad (17)$$

Im unteren rechten Graph der Abbildung 34 wird die berechnete durchschnittliche Maximalkraft jedes Testzyklus mit der tatsächlich aufgebrauchten Maximalkraft verglichen. Es sollte mit einer Kraft von 1038,4 N geprüft werden. Die Prüfmaschine hat mit einer durchschnittlichen Kraftspitze von 1000,74 N auf die Sohle des Schuhs eingewirkt.

Die entspricht einer Genauigkeit von:

$$\frac{1000,74 \text{ N}}{1038,4 \text{ N}} * 100\% = 96,37\% \quad (18)$$

In dem oberen linken Graphen der Abbildung 34 wird der Verlauf der minimalen Deformation jedes Testzyklus visualisiert. Aus dem Verlauf des Graphen lässt sich ablesen, dass die Mittelsohle über die Gesamtheit der Testzyklen einen Höhenverlust von 1,22 mm erfährt. Die anfänglich abfallende Deformation ist mit dem Annähern der Kraftaufbringung der Prüfmaschine an die vorgegebene Kraftspitze zu begründen, siehe Abbildung 34 im unteren linken Graphen. In den ersten ca. 700 Zyklen ist die Kraft überschießend, danach wird eine relativ konstante Last auf den Schuh aufgebracht.

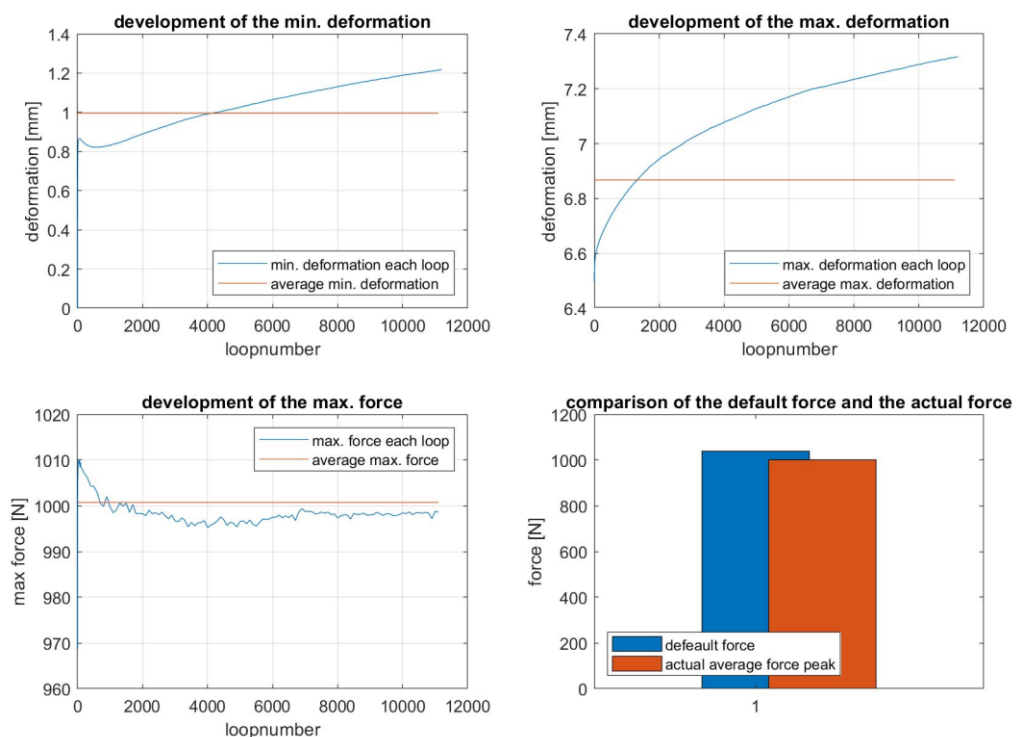


Abbildung 34: Zeitliche Veränderung der Deformation und Kraft für die Messung mit 1000N

Dieses Annähern der durchschnittlichen Last bedingt ebenfalls den Verlauf der maximalen Deformation die der Laufschuh erfährt. In den ersten ca. 700 Zyklen ist die Steigung des Graphen am stärksten, danach flacht diese kontinuierlich ab und

nähert sich der Steigung des Graphen der minimalen Deformation an. Der absolute Höhenverlust unter maximaler Belastung nimmt im Gegensatz zur Deformation bei geringster Belastung nur um $0,82 \text{ mm}$ zu.

$$7,31466 \text{ mm letzter Zyklus} - 6,49 \text{ mm erster Zyklus} = 0,82466 \text{ mm} \quad (19)$$

Als nächster Analyseaspekt wird die aufgenommene bzw. abgegebene Energie der Mittelsohle betrachtet.

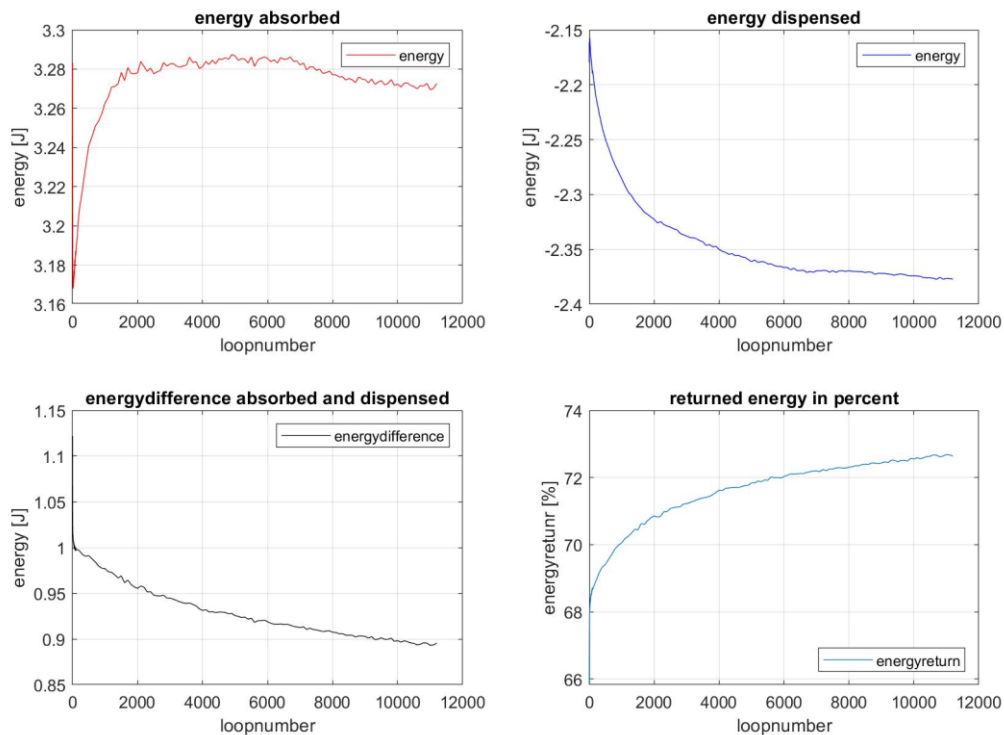


Abbildung 35: Übersicht der auftretenden Energien für die Messung mit 1000N

Im oberen linken Graphen der Abbildung 35 ist der Verlauf der aufgenommenen Energie pro Messzyklus dargestellt. Es ist zu beobachten, dass nach einem anfänglichen Wert von $3,28 \text{ J}$ der absolute Wert der absorbierten Energie in den folgenden 19 Zyklen auf einen minimalen Wert von $3,17 \text{ J}$ abfällt. In den anschließenden 2000 Messzyklen steigt der Graph steil auf $3,28 \text{ J}$ an und steigt folgend relativ konstant auf seinen Maximalwert von $3,29 \text{ J}$ in Zyklus 4901 an. Abschließend nimmt der Graph kontinuierlich über den restlichen Messungszeitraum auf einen Endwert von $3,27 \text{ J}$ im letzten Zyklus langsam ab.

Der Graph neben dem der aufgenommenen Energie stellt die zurückgegebene Energie aus der Mittelsohle dar. Dieser Graph nähert sich, von einem Anfangswert von $-2,16 J$ im ersten Zyklus, asymptotisch einem Endwert von $-2,38 J$ an.

Als weiterer Aspekt wird in Abbildung 35 mit Hilfe der unteren linken Kurve die Differenz der beiden Energiewerte dargestellt. Nach einer anfänglichen hohen Differenz von über $1 J$ in den ersten 50 Zyklen sinkt der Unterschied relativ konstant auf einen Endwert von $0,9 J$ ab.

Die letzte Graphik der Abbildung 35 zeigt das Verhältnis der abgegebenen Energie zur aufgenommenen Energie in Prozent. Dieses steigt von einem Startwert von $65,83\%$ im ersten Messzyklus asymptotisch auf einen Endwert von $72,64\%$ im letzten durchgeführten Zyklus an.

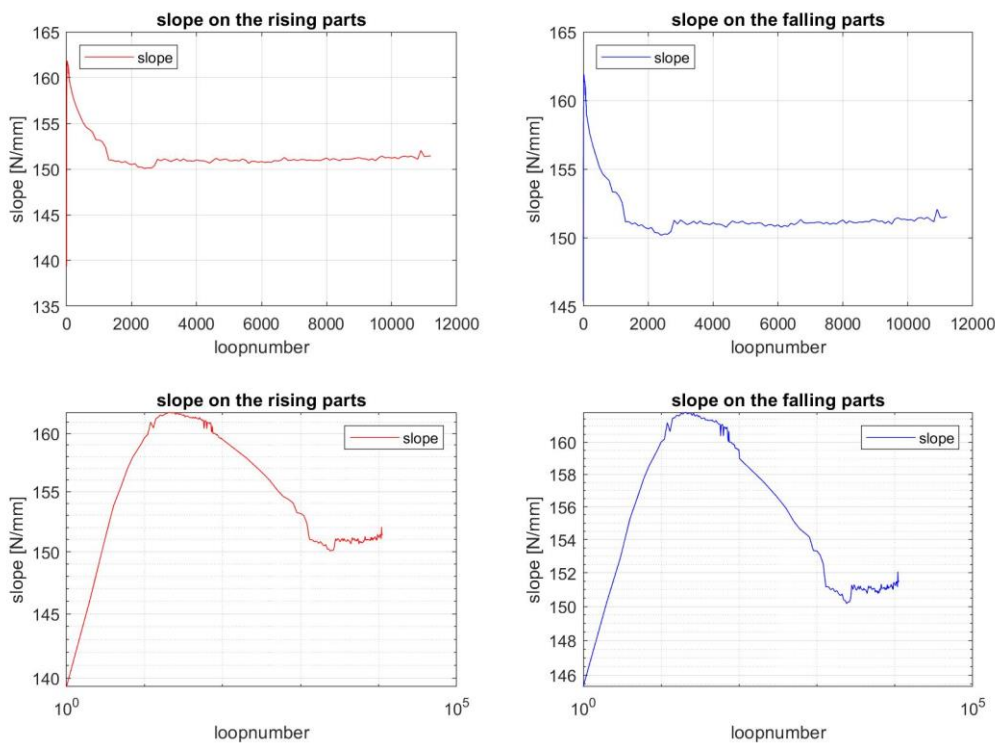


Abbildung 36: Steigung der Kraftdeformationskurve im Verlauf des Messvorganges für die Messung mit 1000N

Der letzte Analyseaspekt stellt die Veränderung der Steigung und somit der Festigkeit dar.

In beiden Darstellungen ist zu erkennen, dass die Steigung des ansteigenden Teils der Kraftverformungskurve in den ersten 24 Zyklen von einem Anfangswert von $139,36 \frac{N}{mm}$ auf einen Wert von $161,85 \frac{N}{mm}$ ansteigt. Anschließend nimmt die Steigung bis Messzyklus 1301 auf den absoluten Wert von $150,99 \frac{N}{mm}$ ab. Bis zum Abschluss der Messung verhält sich die Neigung der Kraftdeformationskurve relativ konstant um einen Wert von $151 \frac{N}{mm}$. Allerdings kann man in der logarithmischen Darstellung ab dem Zyklus 7801 eine steigende Tendenz der Kurve, sprich eine steigende Festigkeit, erkennen.

Die Steigung der Ausgleichsgeraden des abfallenden Teil der Kraftverformungskurve verhält sich identisch zu der des aufsteigenden Teil. Allerdings ist die durchschnittliche Steigung des absinkenden Teil des Graphens, mit $155,65 \frac{N}{mm}$, um $0,14 \frac{N}{mm}$ steiler als die des ansteigenden Teils. Darüber hinaus beginnt die Kurve bereits ab dem 6301. Messzyklus eine ansteigende Tendenz erkennen zu lassen.

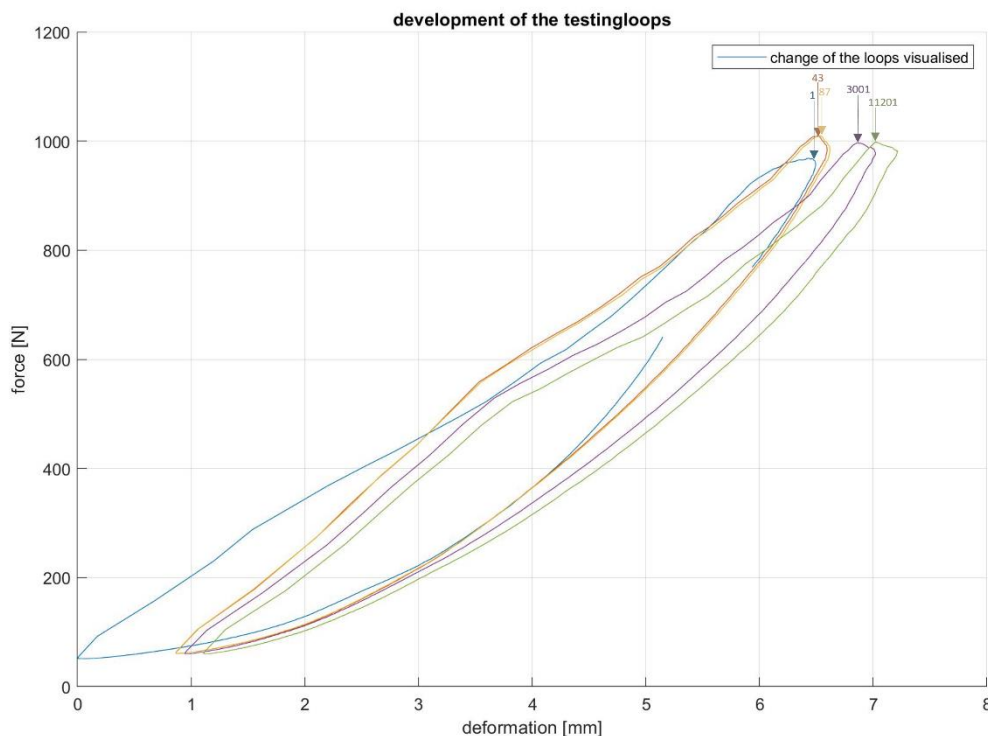


Abbildung 37: Entwicklung der Kraftdeformationskurve von der Messung mit 1000N

Die Summe der gezeigten Analysepunkte spiegelt sich in Abbildung 37 wider. Die genannten Punkte bedingen das Verändern der Kraftverformungskurve im Verlauf der gesamten Messung. Man kann sowohl die Veränderung in der Steigung und die Verschiebung des Startpunktes als auch die Schwankung der Maximallast erkennen.

4.1.3 Zweiter Messdurchgang

Zu Beginn werden wieder die tatsächlich umgesetzten Zeitspannen mit den angestrebten Zeitspannen verglichen, siehe Abbildung 38.

Die ermittelte Schrittdauer lag bei 1,05 s pro Schritt. An der Prüfmaschine war es möglich eine durchschnittliche Dauer von 1,04 s umzusetzen.

Dies entspricht einer Genauigkeit von:

$$\frac{1,04306s}{1,053s} * 100\% = 99,06\% \quad (20)$$

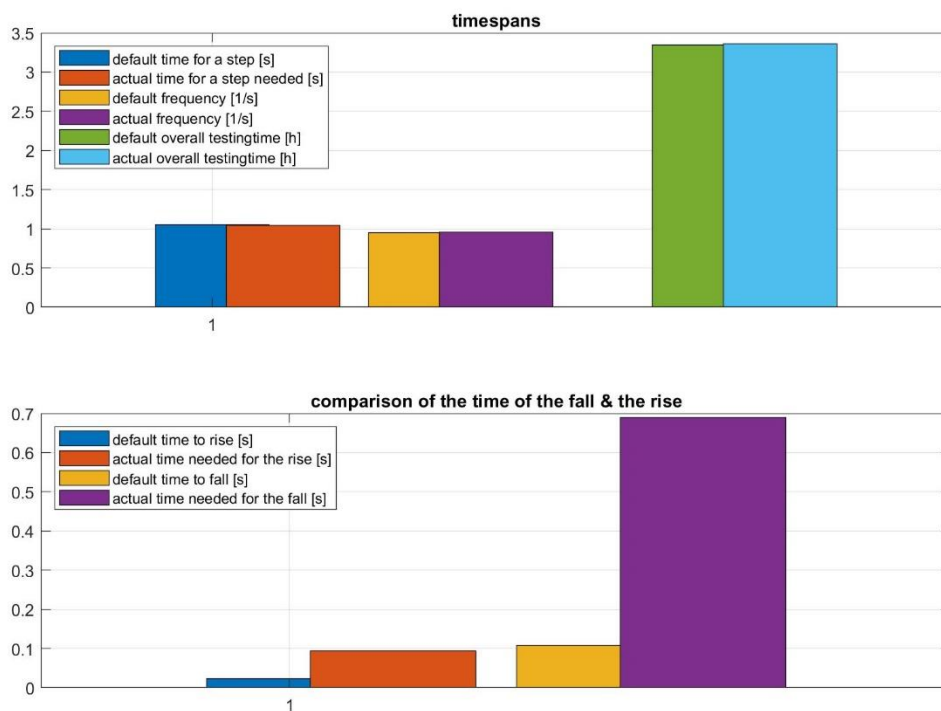


Abbildung 38: Ergebnisse der Zeitanalysen für die Messung mit 800N

Die vorgegebene Frequenz lag bei $0,95 \frac{1}{s}$. Während der Messung ist eine Schrittabfolge mit einer mittleren Frequenz von $0,96 \frac{1}{s}$ simuliert worden.

Dies entspricht einer Genauigkeit von:

$$\frac{1,04306 \text{ s}}{1,053 \text{ s}} * 100\% = 99,06\% \quad (21)$$

Als letzter Wert des oberen Graphen wird die angestrebte Gesamtprüfzeit von $3,35 \text{ h}$ der tatsächlichen Dauer von $3,36 \text{ h}$ gegenübergestellt.

Dies entspricht einer Genauigkeit von:

$$\frac{3,34881 \text{ h}}{3,36113 \text{ h}} * 100\% = 99,63\% \quad (22)$$

Das untere Diagramm der Abbildung 38 stellt, wie bereits für die erste Messung in Kapitel 4.1.1 beschrieben, die Zeiten zum Erreichen der Extrempunkte dar. Für das Erreichen der Maximallast war eine Zeitspanne von $0,02 \text{ s}$ vorgesehen. Umgesetzt worden ist eine Dauer von $0,09 \text{ s}$.

Dies entspricht einer Abweichung von:

$$\frac{0,0932142 \text{ s}}{0,023 \text{ s}} * 100\% = 405,28\% \quad (23)$$

Als letztes Wertepaar für die Abbildung 38 wird die bereits beschriebene Zeit, die zum Absinken der Kraft auf unter 100 N benötigt wird, verglichen. Die Zeit sollte bei $0,11 \text{ s}$ liegen, tatsächlich dauert das Absinken der Messkraft $0,69 \text{ s}$.

Dies entspricht einer Abweichung von:

$$\frac{0,690461 \text{ s}}{0,108 \text{ s}} * 100\% = 639,32\% \quad (24)$$

Abbildung 39 zeigt, in der unteren rechten Darstellung, den Vergleich der angestrebten durchschnittlichen Maximallast von $866,72 \text{ N}$ und der Last, von $803,14 \text{ N}$, welcher der Schuh durchschnittlich ausgesetzt worden ist.

Dies entspricht einer Genauigkeit von:

$$\frac{803,144 \text{ N}}{866,72 \text{ N}} * 100\% = 92,66\% \quad (25)$$

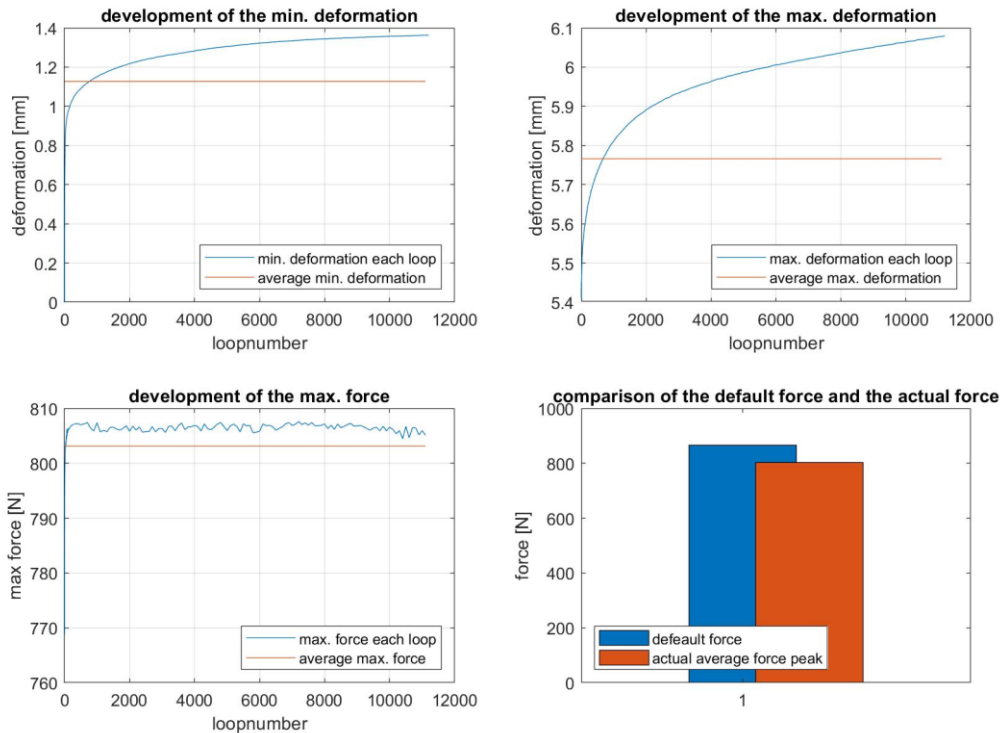


Abbildung 39: Zeitliche Veränderung der Deformation und Kraft für die Messung mit 800N

Die oberen Graphen visualisieren den Verlauf der Deformation der Mittelsohle. Oben links wird die minimale Deformation und oben rechts die maximale Deformation dargestellt. Man kann erkennen, dass die Mittelsohle einen langsam steigenden absoluten Höhenverlust von $1,36 \text{ mm}$ über den Verlauf der Messzyklen erfährt. Die Maximale-Deformationskurve steigt hingegen steiler und erreicht einen Maximalwert von $6,08 \text{ mm}$.

Der untere linke Graph der Abbildung 39 zeigt den Verlauf der Kraftspitze der Messung. Man kann ein starkes Ansteigen der Belastungsspitzen in den ersten 100 Zyklen beobachten, anschließend pendelt der Maximalwert relativ konstant um ca. 806 N .

Die nachfolgende Abbildung 40 zeigt einen Überblick über die Energiemengen die von der Mittelsohle aufgenommen bzw. abgegeben worden sind, deren prozentuales Verhältnis und die Differenz der beiden Energiemengen.

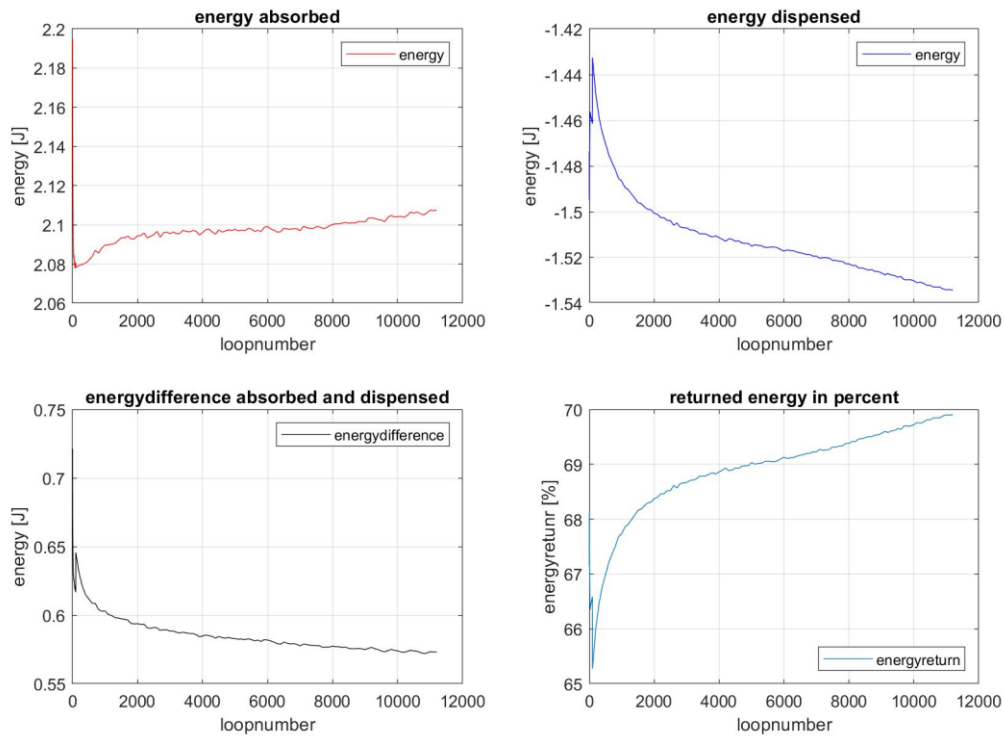


Abbildung 40: Übersicht der auftretenden Energien für die Messung mit 800N

Auffällig beim Kurvenverlauf der aufgenommenen Energie ist, dass in den ersten 100 Zyklen die Energiemenge sehr stark von einem Anfangswert von 2,19 J auf einen Minimalwert von 2,08 J absinkt. Anschließend steigt die Kurve relativ konstant auf einen Endwert von 2,11 J.

Der Wert der abgegebenen Energiemenge beginnt bei einem Startwert von $-1,47 J$ und steigt in den ersten drei Zyklen auf einen Zwischenwert von $-1,49 J$. In den folgenden 30 Messzyklen fällt die absolut abgegebene Energiemenge stark auf $-1,46 J$ ab um dann bis Zyklus 100 wieder anzusteigen. Besonders auffällig ist darauffolgend der große Abfall von $-1,46 J$ auf $-1,43 J$ in Zyklus 101. Im restlichen Verlauf nimmt die resorbierte Energiemenge der Sohle auf einen Endwert von $-1,53 J$ zu.

4 Ergebnisse

Die Energiedifferenz nimmt, nach dem starken Abfallen des Graphen der aufgenommenen Energie in den ersten 100 Messzyklen und dem darauf folgenden starken Sprung der angegebenen Energie, stetig ab. Zum Ende der Messung liegt noch eine Differenz von 0,57 J vor.

In der letzten Graphik der Abbildung 40 wird das prozentuale Verhältnis von abgegebener zu aufgenommener Energiemenge dargestellt. In dieser Graphik ist ebenfalls der starke Sprung von dem 100. auf den 101. Zyklus zu erkennen. Darüber hinaus nähert sich die Kurve in den ersten 6000 Zyklen asymptotisch, von 67,14% aus startend, dem Zwischenwert von 62,28%. Abschließend steigt das Verhältnis relativ konstant auf einen Endwert im letzten Zyklus von 72,8% an.

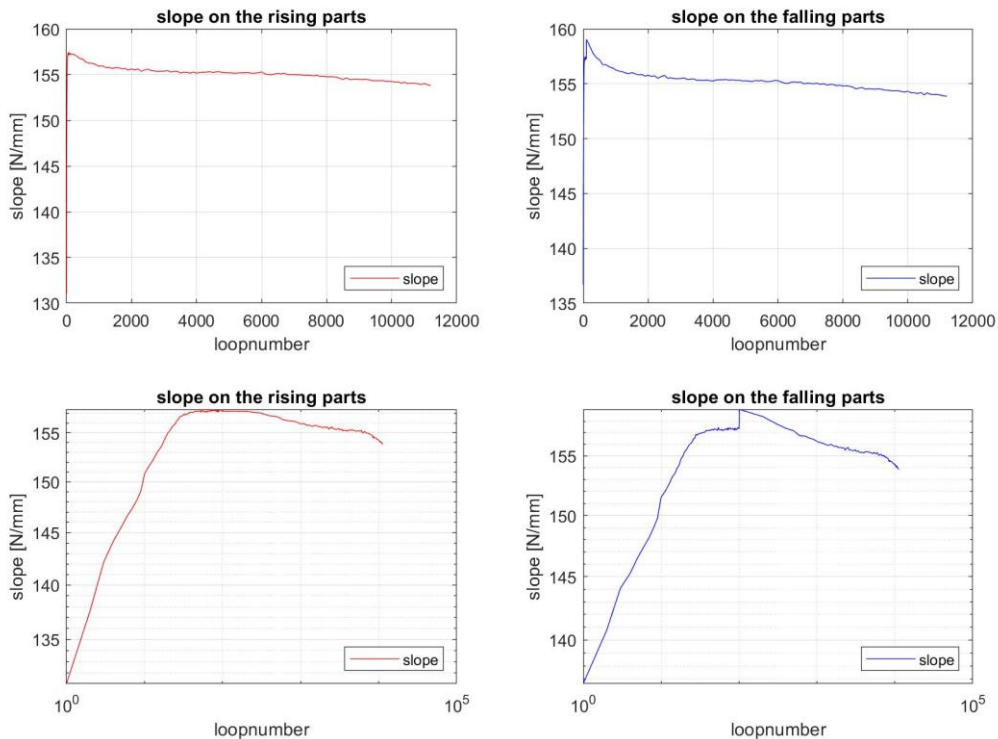


Abbildung 41: Steigung der Kraftdeformationskurve im Verlauf des Messvorganges für die Messung mit 800N

Anschließend werden die Ergebnisse der veränderten Steigung bzw. der Festigkeit der Kraftdeformationskurve besprochen, siehe Abbildung 41.

In den ersten 53 Zyklen nimmt die Steigung des „rising part“ auf einen Spitzenwert von $157,45 \frac{N}{mm}$ stark zu, darauf folgend wird diese relativ gleichmäßig weniger um am Ende der Messung einen Wert von $153,91 \frac{N}{mm}$ zu haben.

Die Kurve, die den „falling part“ der Kraftdeformationskurve widerspiegelt, verläuft ähnlich der des aufsteigenden Teils. Bis Messzyklus 100 ist ein starker Anstieg auf $157,35 \frac{N}{mm}$ zu beobachten, anschließend findet sich der starke Anstieg aus der vorherigen Abbildung 40, der Übersicht der Energien, im Verlauf der Steigung wieder. Innerhalb eines Zyklus steigt dieser auf $159,96 \frac{N}{mm}$ an. Den restlichen Verlauf der Messung über fällt die Steigung auf einen Endwert von $154,02 \frac{N}{mm}$ ab.

Die Steigung des abfallenden Teils der Kraftdeformationskurve ist, wie schon bei der ersten Messung, größer. In diesem Messdurchgang beträgt die Differenz:

$$155,4467 \frac{N}{mm} - 155,298 \frac{N}{mm} = 0,1487 \frac{N}{mm} \quad (26)$$

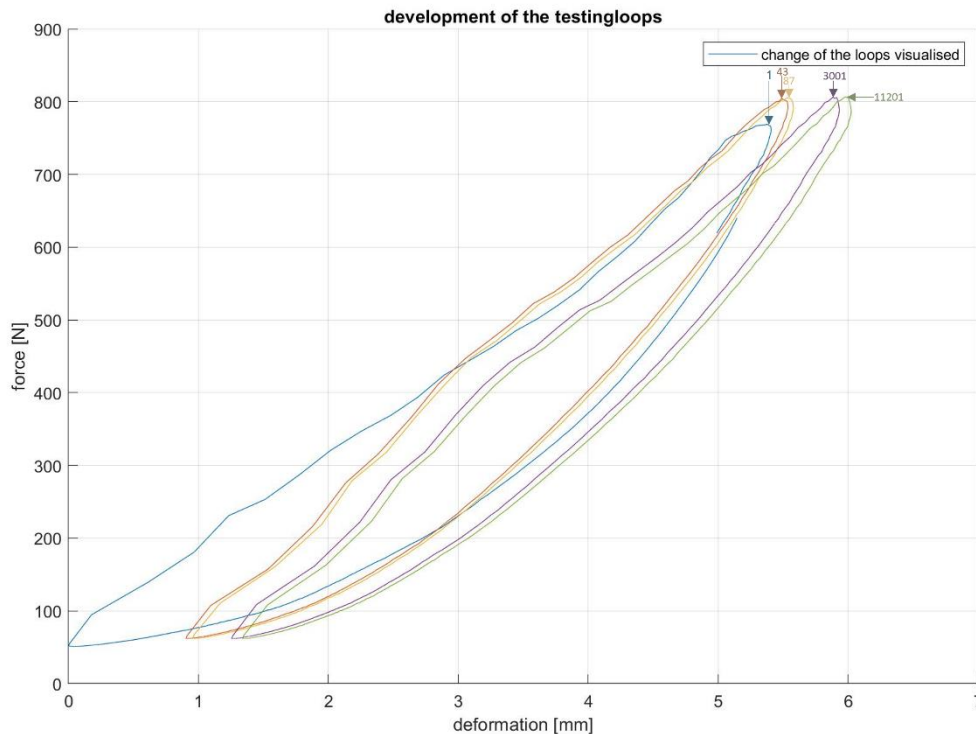


Abbildung 42: Entwicklung der Kraftdeformationskurve von der Messung mit 800N

Die letzte Abbildung für diesen Messdurchgang zeigt die Veränderung der Kraftdeformationskurve über den Verlauf der gesamten Messzeit. Wie bereits in Kapitel 4.1.1 spiegeln sich die Analysepunkt der Messungen in dieser Graphik wider.

4.1.4 Vergleich der Messdurchgänge

Die Verläufe der einzelnen Ergebniskurven wurden bereits in den Kapiteln 4.1.1 und 4.1.2 beschrieben. Folgend werden die Unterschiede beider Kurven analysiert.

Abbildung 43 zeigt die bereits bekannten Verläufe der Deformationen und der Maximallast.

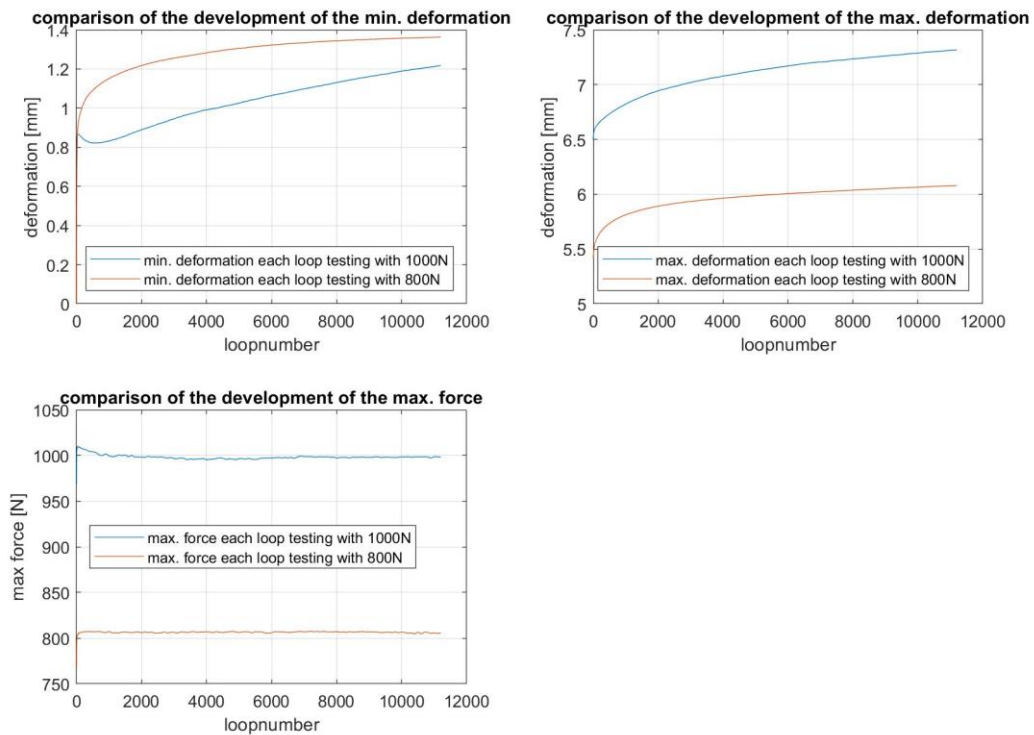


Abbildung 43: Vergleich der zeitlichen Veränderung der Deformation und Kraft beider Messungen

Auffällig ist in dem oberen linken Graphen, dass die mit der geringeren Kraft entstandenen Kurve bereits ab dem 27. Zyklus die Kurve der 1000 N-Messung übersteigt und einen um 1,15 mm höheren Endwert hat.

$$1,36309 \text{ mm} - 1,21667 \text{ mm} = 0,14642 \text{ mm} \quad (27)$$

Dies bedeutet der Maximalwert der zweiten Messung beträgt 109,34% des Maximalwertes des ersten Durchganges.

$$\frac{1,36309 \text{ mm}}{1,24667 \text{ mm}} * 100\% = 109,34\% \quad (28)$$

Bei dem Graphen der maximalen Verschiebung sticht hervor, dass in diesem Analysepunkt die Verformung der höheren Kraftkurve wieder die der geringeren Kraftkurve betragsmäßig übersteigt. In diesem Fall beträgt die Differenz:

$$7,31708 \text{ mm} - 6,07909 \text{ mm} = 1,23799 \text{ mm} \quad (29)$$

Prozentual betrachtet beträgt das Maximum der zweiten Messung 83,1% des Maximums der ersten Messung.

$$\frac{6,07909 \text{ mm}}{7,31708 \text{ mm}} * 100\% = 83,1\% \quad (30)$$

Auch im Verlauf der Lastspitzen gibt es Unterschiede. Auf der einen Seite braucht die Prüfmaschine ca. 700 Messzyklen um die anfänglich überschießende Kraft auf einen relativ konstanten Wert herunterzuregeln. Auf der anderen Seite werden die durchschnittlichen 803 N innerhalb von 20 Zyklen erreicht.

In Abbildung 44 werden beide Messdurchgänge in Bezug auf deren aufgenommenen bzw. abgegebenen Energiemengen miteinander verglichen. Die optische Ähnlichkeit beider Kurven ist besonders auffällig.

Trotz der optisch ähnelnden Verläufe entspricht eine um 20% geringere Maximallast nicht der äquivalenten geringeren umgesetzten Energiemenge.

Bei der aufgenommenen Energiemenge werden lediglich 65,63% der vorherigen Messung aufgenommen.

$$\frac{2,1 \text{ J}}{3,2 \text{ J}} * 100\% = 65,63\% \quad (31)$$

4 Ergebnisse

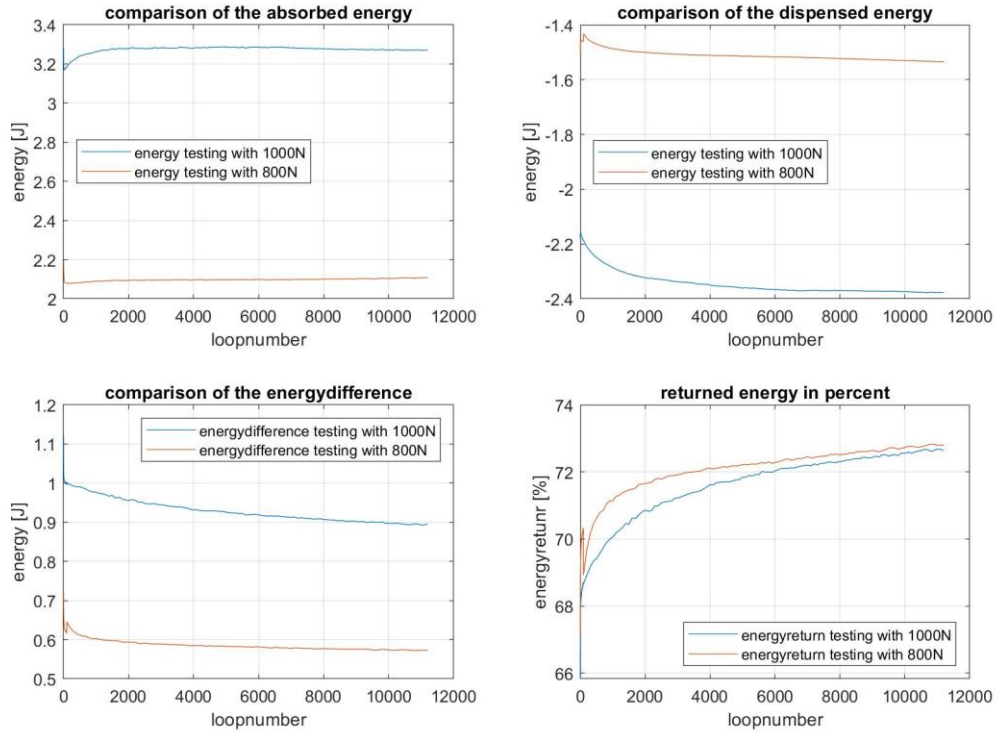


Abbildung 44: Vergleich der auftretenden Energien für beide Messungen

Das Verhältnis der abgegebenen Energiemenge gleicht stark dem der aufgenommenen Energiemenge:

$$\frac{-1,5 J}{-2,4 J} * 100\% = 62,5\% \quad (32)$$

Die Energiedifferenz hat ebenfalls ein ähnliches Verhältnis, wie bereits die zuvor berechneten Werte.

$$\frac{0,6 J}{0,9 J} * 100\% = 66,67\% \quad (33)$$

Darüber hinaus ist in der unteren linken Graphik der Abbildung 44 zu erkennen, dass sich der Verlauf des prozentualen Anteils der abgegebenen Energie in beiden Messdurchgängen sehr ähnelt. Allerdings beträgt der Spitzenwert der Messung mit 800 N 100,23% des Spitzenwertes der Messung mit 1000 N.

$$\frac{72,8022\%}{72,6359\%} * 100\% = 100,23\% \quad (34)$$

Eine Gegenüberstellung der Steigungen, respektive der Festigkeiten, in Abbildung 45, zeigt, dass der Verlauf, der mit 1000 N dokumentiert worden ist, stärkere Veränderungen als die relativ gleichförmig verlaufende Kurve, der geringeren Kraftaufbringung, zeigt. Darüber hinaus fällt die Kurve der Messung mit 800 N ab dem 53. Messzyklus stetig ab. Der Verlauf der anderen Kurve hingegen steigt zum Ende der Messung wieder leicht an, nachdem diese zuvor kontinuierlich abgenommen hatte.

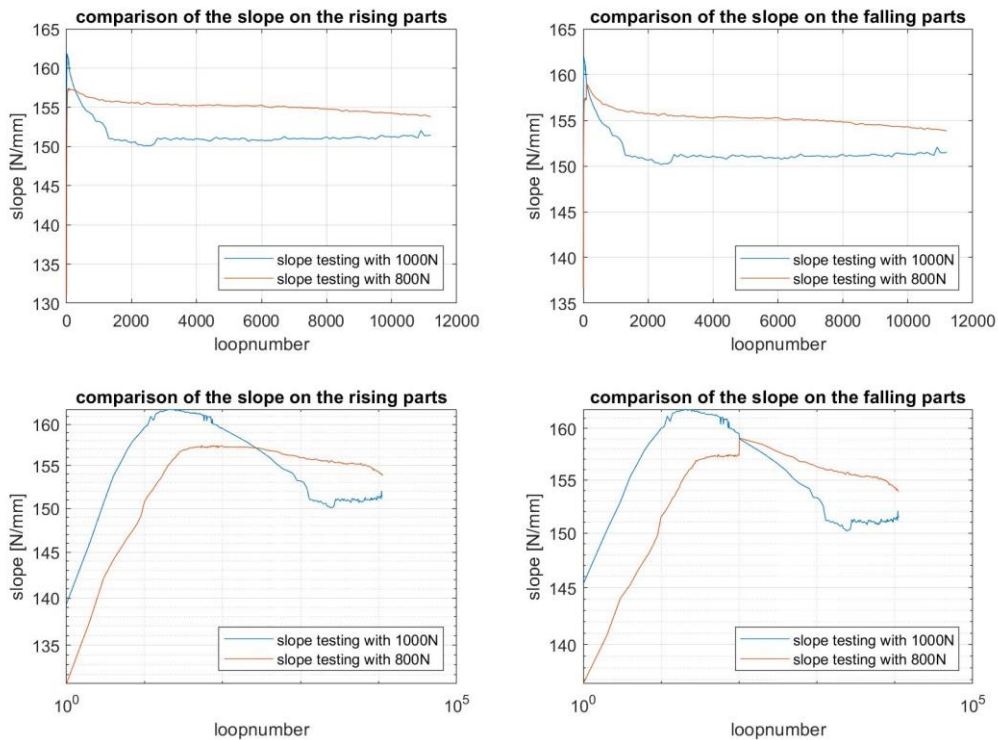


Abbildung 45: Vergleich der Steigungen der Kraftdeformationskurve beider Messungen

5 Diskussion

5.1 Datenfindung für die Messdurchführung

Als einen der größten Kritikpunkte an dem Vorgehen dieser Arbeit ist das anfängliche Fokussieren auf die Recherche der Bodenreaktionskraftkurven zu betrachten. Wäre direkt zu Beginn mehr Zeit investiert worden, um die Bedeutung dieser recherchierten Kurven im Detail zu ergründen, wäre es wahrscheinlich deutlich früher aufgefallen, dass dieser Weg der Datenfindung nicht zielführend war und man hätte früher nach Alternativen recherchieren können.

Nachdem diese Problematik mit Hilfe der zur Verfügung gestellten Rohdaten der Masterthesis von Herrn Brinkemper gelöst worden ist, wurde es möglich, diese spezifisch gesuchten und in der Literatur bisher noch nicht thematisierten Daten zur Einstellung der elektro-dynamischen Prüfmaschine herauszuarbeiten.

Diese generierten Daten verfügten exakt über die für diese Messung benötigten Informationen. Der einzige Kritikpunkt dazu ist, dass lediglich 20 Probanden untersucht worden sind. Für eine allgemeingültige Aussage, die den Querschnitt der Gesellschaft widerspiegelt, wäre eine Untersuchung von deutlich mehr Probanden nötig. Darüber hinaus dürften nicht nur relativ junge und sportliche Personen analysiert werden. Für diese Arbeit waren die verfügbaren Daten dennoch überaus hilfreich.

Für die Berechnung der durchschnittlichen Schrittfrequenz wäre es ebenfalls von Vorteil gewesen, wenn man mehr als die drei verwendeten Studien für das Bilden eines Durchschnittes verwendet hätte. Oder zumindest mehr Studien eines neueren Erstellungsdatums. In dieser Arbeit ist lediglich die Studie von Bailey et al. nach dem Jahre 2015 erstellt worden. Allerdings war es aus Mangel an vorhandenen Studien schlichtweg nicht möglich mehr aussagekräftige Daten zu Rate ziehen zu können.

Dies gilt ebenso für die Studie, der das durchschnittliche Gewicht eines europäischen Erwachsenen entnommen wurde. Diese wurde von im Jahre 2012 von Walpole et al. veröffentlicht. Da dies bereits 9 Jahre zurückliegt kann man argumentieren, dass dieser verwendete Wert, von „70,8 kg“ (Walpole et al., 2012, S.3), zum heutigen Zeitpunkt sicherlich nicht mehr aktuell ist.

5.2 Messaufbau

Wenn man den Aspekt der Reproduzierbarkeit der Messungen betrachtet, wäre es sinnvoll gewesen den verwendeten Schuh mit Hilfe einer anderen Aufnahme unter dem Prüfdorn zu positionieren. Eine einfache, mit einem Stift gezeichnete, Markierung für die Positionierung, wird gerade in Bezug auf die eventuelle Vergleichbarkeit verschiedener Modelle nicht genügen. Allerdings wäre eine umrandete Schuhaufnahme ebenfalls kontraproduktiv, da in diese eventuell breiter ausgelegte Schuhsohlen, bei der selben Schuhgröße, nicht mehr in diese Aufnahme passen könnten. Bei dieser verwendeten Art der Lagerung war es nicht möglich zu beurteilen, ob sich die Position des Schuhs im Laufe der Messung minimal verändert hat.

Das gleiche Problem zeigt sich in Bezug auf die Positionierung des Prüfdorns, dessen Position ebenfalls händisch markiert wurde. Für zukünftige Projekte könnte man eine Sacklochbohrung in den Leisten integrieren um ein eventuelles Verschieben zu verhindern. Darüber hinaus wäre eine Zerteilung des Leistens auf der Höhe des Mittelfußes in Betracht zu ziehen, um gezielt den vorderen bzw. den hinteren Abschnitt der Sohle untersuchen zu können. Wenn der Leisten, wie in dieser Arbeit, im Ganzen in den Schuh eingelegt wird, könnte es sein, dass die Kraftverteilung sich nicht nur auf den Rückfußbereich beschränkt, sondern sich in Richtung Zehen ebenfalls ausbreitet. Dies würde die Aussagekraft der Ergebnisse negativ beeinflussen.

Die Umsetzbarkeit der genannten Verbesserungsideen war im Rahmen dieser Arbeit nicht möglich und könnten für weiterführende Arbeiten zu diesem Thema angewandt werden.

Als weiterer Kritikpunkt ist der Unterschied von der getesteten Belastung zu den realen Bedingungen, denen ein Laufschuh über solche Distanzen ausgesetzt ist, zu betrachten. Auch wenn die zwei gegeneinander verschiebbaren Metallplatten unter der hölzernen Grundplatte bei Belastung eine leichte Verschiebung in der Position des Schuhs zugelassen haben, entfällt doch der Anteil der Scherbelastung auf die Mittelsohle so gut wie vollständig. Dies minimiert die Gesamtbelastung welcher, der Schuh ausgesetzt worden ist und könnte unter Umständen die Ermüdungszeichen des Materials mindern. Dafür könnte der Umstand, dass die Mittelsohle zu keinem

Zeitpunkt der Messung unbelastet war die im 4. Kapitel beschriebenen Ergebnisse ausgeprägter erscheinen lassen, als es bei einem realen Marathon vermutlich der Fall wäre.

Dies gilt auch für den Untergrund, auf dem der Schuh sich während der Messung befunden hat, da ein kontinuierliches Laufen auf einem hölzernen Untergrund als nicht realistisch zu betrachten ist. Allerdings muss dieser Aspekt nicht besonders ins Gewicht fallen, da durch das fehlende Abheben und Aufsetzen der Schuh nicht in eine übermäßige Interaktion mit dem Untergrund getreten ist und das bereits beschriebene Fehlen der Scherkräfte ebenfalls die Belastung auf die Außensohle stark minimiert haben wird.

Zu den bereits erwähnten Unterschieden zum „echten“ Laufen muss noch der Punkt der Gleichförmigkeit der Bewegungsabfolge der Prüfmaschine hinzugezählt werden. Unter realen Bedingungen wird ein Läufer Erschöpfung und dadurch eine eventuelle Änderung des Laufstils erfahren. Dadurch könnte sich eventuell das Areal des ersten Kontaktpunktes mit dem Untergrund verändern oder das Abrollverhalten beim Laufen könnte variiert werden.

Als letzter Aspekt des Messaufbaus lässt sich bemerken, dass für beide Messdurchgänge der selbe Schuh verwendet worden ist. Sprich eventuelle Materialveränderungen die bereits in der er ersten Messung entstanden sind können die Aussagekraft der Ergebnisse der zweiten Messung negativ beeinflussen.

Zusammenfassend lässt sich sagen, dass sich die Bedingungen während der Messung in mehreren Punkten von der Realität unterscheiden, aber diese in den Betrachtungspunkten der Energieverarbeitung und der Festigkeitsänderung des Materials keine so entscheidende Rolle spielen um eine eventuelle Vergleichbarkeit verschiedener Belastungszustände oder unterschiedlichen Sportschuhmodelle zu verhindern.

5.3 Messergebnisse

Bei der Betrachtung der Messergebnisse sind die bereits im vorherigen Kapitel angebrachten Kritikpunkte am generellen Messaufbau zu berücksichtigen.

Folgend werden beiden Messungsergebnisse vergleichend betrachtet und auf Auffälligkeiten beider Messungen eingegangen.

5.3.1 Interpretation der Zeiten und der Kräfte

Als erster wichtiger Punkt ist die Genauigkeit der angestrebten Zeitspannen zu betrachten. Die für das akkurate Simulieren eines Schrittes angestrebte Schrittdauer wurde bei beiden Messdurchgängen mit einer überaus hohen Genauigkeit von jeweils mehr als 99% mehr als zufriedenstellend erfüllt. Dies gilt ebenfalls für die angestrebte Frequenz und die Gesamtdauer der Messdurchgänge, alle Ergebnisse liegen im Bereich oberhalb von 99%.

Kritischer hingegen sind die erreichten Zeiten für den Anstieg, der minimalen Kraft zur Belastungsspitze bzw. dem Abfallen zum Minimum, zu betrachten. Für beide Untersuchungsdurchgänge liegen die von der Prüfmaschine umgesetzten Zeitabschnitte um ein Vielfaches höher als die angestrebten Werte, die sich aus der Rohdatenanalyse ergeben haben. Dies ist im Grunde mit der hohen Komplexität der Einstellbarkeit des PID-Reglers zu erklären. Es war trotz mehrfachen Anpassens der errechneten Einstellungswerte nicht möglich die Prüfmaschine einen gewünschten Kraftverlauf, der das natürliche Laufen widerspiegelt, mit einer höheren Präzision ausführen zu lassen.

Daraus resultiert, dass die Mittelsohle deutlich länger und stärker belastet wurde, als es bei ein realen Lauf der Fall gewesen wäre.

Ein weiterer Punkt der sich auf das suboptimale Einstellen des PID-Reglers zurückführen lässt, ist das Abweichen der erreichten Maximalkräfte im Vergleich zu den angestrebten Kraftwerten. Mit einer Genauigkeit von 96% im ersten Durchgang, und 92% im Folgenden, sind die Abweichungen der Kraftspitzen zwar nicht weit von der Zielvorgabe entfernt, aber es liegt offensichtlich ein Unterschied vor. Dennoch sollte diese Abweichung bei der Bewertung des Erfolgs der Umsetzung nicht

besonders ins Gewicht fallen, da die aufgebrachten Kraftmaxima relativ konstant im Verlauf der Messung geblieben sind und unter realen Bedingungen das Körpergewicht eines Läufers sich auch nicht merklich ändert. Das Ziel, die Visualisierung eines Unterschiedes bei einem variablen Körpergewicht, ist offensichtlich erreicht worden.

5.3.2 Interpretation der Deformation

Bei der Betrachtung der Verformungen ist es interessant zu beobachten, dass der Betrag des minimalen Höhenunterschiedes für die Messung mit geringerer Kraft größer ist als der der Messung unter höherer Last. Die Differenz beträgt zwar lediglich $0,14\text{ mm}$, dennoch ist es ein Ergebnis, das so nicht erwartet worden ist. Es könnte damit zusammenhängen, dass der Schuh bereits das zweite Mal getestet und bereits einer strukturellen Veränderung ausgesetzt worden ist. Wenn dies der Fall wäre, würde man erwarten, dass bei der Betrachtung der maximalen Deformation der Messdurchlauf, unter geringerer Last, ebenfalls eine stärker ausgeprägte Verformung provoziert hätte. Da der Höhenverlust der Messung unter höherer Last in Summe $1,24\text{ mm}$ höher ausgefallen ist, ist dies nicht der Fall. Eine weitere Überlegung worin diese Auffälligkeit begründet sein könnte, wäre eine Betrachtung der Geschwindigkeiten bzw. der Zeitspannen, die die Prüfmaschine benötigt hat um die Extremwerte zu erreichen. Allerdings sind die Zeiten die benötigt werden um auf eine maximale Last zu erreichen, bei beiden Durchgängen, $\sim 0,09\text{ s}$ und für das Mindern der Kraftspitze auf den geringsten Lastzustand wird für den ersten Durchgang $\sim 0,67\text{ s}$ und für den zweiten $\sim 0,69\text{ s}$ benötigt. Eine mögliche Ursache für diese Diskrepanz in diesen nahezu identischen Werten zu begründen wäre als relativ unwahrscheinlich einzustufen. Eventuell kann der unterschiedliche Kurvenverlauf als Anlass für diese Auffälligkeit in Betracht gezogen werden. Die zu Beginn überschießende Maximallast im ersten Messdurchgang könnte die vernetzten Molekülketten des EVA-Schaummaterials bereits in den ersten Messzyklen so weit gegeneinander verschoben haben, dass Ketten, die durch ihre ursprüngliche Vernetzung eine vertikale Verschiebbarkeit der Mittelsohle behindert hätten, bereits gelöst worden sind und von da an ein flüssigeres Reaktionsverhalten der Mittelsohle möglich geworden ist. Die nachfolgende größere Steigung der Kurve

würde dann einen verstärkten Materialverschleiß in den übrigen Messzyklen verdeutlichen. Es könnte auch darin begründet sein, dass der Schuh bis zu dem Zeitpunkt der ersten Messung noch nie belastet worden ist und das Material sich zuerst „einspielen“ muss.

Abschließend lässt sich diese Ursachenfindung, ohne ein bildgebendes Verfahren, das Einblick in die strukturelle Beschaffenheit der Sohle gewährt, nicht aussagekräftig klären.

5.3.3 Interpretation der Energie

Die Ergebnisse der auftretenden Energien befinden sich in Bereichen die erwartet worden sind. Die in Kapitel 1 aufgeführten, erwartbaren maximalen $5 J$ bei einem einmaligen Belastungsimpuls, sind in einem realistischen Verhältnis zu den über die Gesamtzeit zyklisch aufgenommenen Energiemengen von $3,3 J$ und $2,2 J$. Als auffälliges Charakteristikum zeigt sich bei der Messung mit geringerer Last, dass die aufgenommene Energie nach einer zu Beginn starken Abnahme über den Gesamtverlauf der Messung stetig ansteigt. Dies würde erneut für ein „Einlaufen“ bzw. die verbesserte Verschiebbarkeit der Molekülketten des Schuhs in Benutzung sprechen. Ebenfalls interessant ist der bei der abgegebenen Energie zu erkennende Sprung im Verlauf des Graphen, von Zyklus 100 auf Zyklus 101. Dieser Sprung könnte in einer plötzlich auftretenden Materialeigenschaftenveränderung begründet sein, jedoch müsste ein struktureller Schaden auch in der aufgenommenen Energie ablesbar sein, was nicht der Fall ist. Zwar steigt die aufgenommene Energie ab Zyklus 100 stetig, allerdings ist keine plötzliche Veränderung im Verlauf der Kurve zu erkennen. Um einen Messfehler oder ein Verrutschen des Schuhs auf der Grundplatte kann es sich nicht handeln, da alle nachfolgenden Werte an den Wert des im 101. Zyklus entstandenen Sprung anknüpfen und in den Darstellungen der Deformation keine Auffälligkeiten zu erkennen sind. Allerdings vollzieht die Kurve des Graphens, der die Steigung repräsentiert, einen Sprung in demselben Zyklus. Eventuell hat dieser Sprung seine Ursache in der Vernetzung der Polymerketten, dies kann ohne eine Bildgebung jedoch nicht bestätigt werden.

Das Absinken der Kurve der aufgenommenen Energie für den ersten Messdurchgang und bei den Energiedifferenzen beider Messungen, deuten auf eine Materialermüdung im Verlauf der Marathondistanz hin. Dies kann in einer Schädigung der Querbrücken der Schaumstruktur begründet sein und dass Lufteinschlüsse in den Schaumkammern durch diese Schädigungen entweichen konnte und somit die Gesamtelastizität des Systems der Mittelsohle verringert worden ist.

5.3.4 Interpretation der Steigung

Die Steigung zwischen den Extrempunkten der Kraftdeformationskurve repräsentiert die Festigkeit des zu untersuchenden Materials. Der anfänglichen Erwartung, die Mittelsohle wird von Beginn der Belastung an fester bzw. die Festigkeit nimmt durch fortschreitende Materialermüdung kontinuierlich zu, stehen die in den Kapiteln 5.3.2 und 5.3.3 bereits gewonnenen Erkenntnisse gegenüber. In beiden Kapiteln wurde beschrieben, dass ein anfängliches Verbessern des Ineinandergleitens der Polymerketten stattfinden könnte und daraus resultierend die Festigkeit abnimmt. Dieses Phänomen lässt sich ebenfalls in der Auswertung der Steigungen der Kraftdeformationskurve beobachten. Die, bei der Messung mit geringerer Kraft entstandene, Kurve nimmt über ihren gesamten Verlauf ab, dies spricht ebenfalls für eine Minderung der Festigkeit.

Auf der anderen Seite hingegen zeigt die Kurve der höheren Belastung im letzten Drittel der Messung eine steigende Tendenz. Dieser Umstand zeigt, dass die Festigkeit der Mittelsohle zunimmt und man daraus resultierend eine Materialermüdung ablesen kann.

5.3.5 Abschluss Messung

Die gemessenen Ergebnisse sind als realistisch einzuschätzen. In einer Studie haben Hoogkamer und Kollegen drei verschiedene Laufschuhe unter dem Untersuchungsaspekt der prozentualen Energierückgewinnung untersucht. Alle von ihnen angegebenen Messergebnisse sind in einem ähnlichen Rahmen wie die Ergebnisse dieser Arbeit, siehe Abbildung 46.

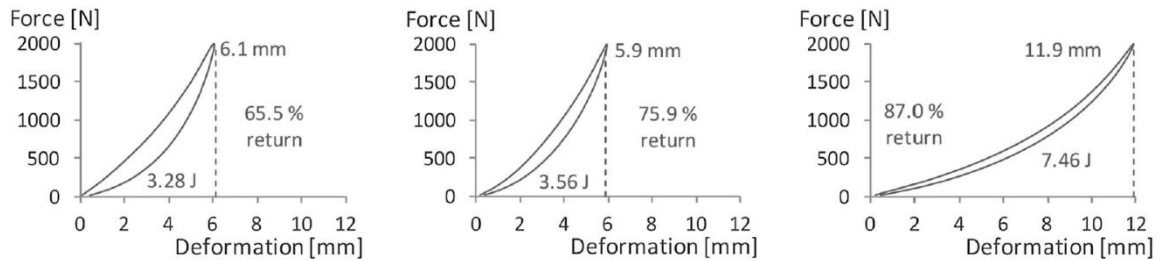


Abbildung 46: Energierückgabe verschiedener Laufschuhmodelle (Hoogkamer et al., 2018, S.1012)

Abschließend lässt sich sagen, dass in allen untersuchten Aspekten das Ausmaß der Veränderungen bei der Messung mit 1000 N ausgeprägter waren als bei der Messung mit geringerer Last. Daraus lässt sich schließen, dass eine Person mit einem höheren Eigengewicht eine stärkere Schwankung in den Materialeigenschaften der verwendeten Laufschuhe zu erwarten hat und dies bei der Wahl des geeigneten Schuhs berücksichtigt werden sollte.

6 Zusammenfassung

Dies Arbeit ist mit dem Ziel begonnen worden, ein Messverfahren zu entwickeln, dass das Materialreaktionsverhalten von Mittelsohlen von Laufschuhen über eine simulierte Distanz eines Marathons analysieren kann.

Der erste große Abschnitt dieser Arbeit hat sich mit der Recherche rund um das Thema Laufen und dessen Charakteristika beschäftigt, die nötig waren um ein möglichst realitätsnahes Simulationsverfahren zu ermöglichen. Nach anfänglichen Schwierigkeiten mit der Fokussierung auf zielführendes Material wurde sehr schnell mit Hilfe der Daten aus der Abschlussarbeit „*Die Interaktion zwischen Mensch und Untergrund für den Vor- und Rückfuß beim Laufen.*“, von Alexis Brinkemper aus dem Jahr 2013, eine Möglichkeit geschaffen die benötigten Daten zu generieren. Das geschriebene Matlabskript konnte gut alle zur Prüfung notwendigen Gesichtspunkte aus den Rohdaten herausfiltern. Zusätzlich zu den Einstellungsbedingungen der elektro-dynamischen Prüfmaschine, wurde in dem Kapitel der Literaturrecherche ein Überblick über die Beschaffenheit und den Aufbau eines Laufschuhs gegeben. Die Bedeutung der zu erwartenden Kraftdeformationskurve in Bezug auf die anschließende Interpretierbarkeit wurde erklärt. Gerade die nachfolgend wichtigen Gesichtspunkte der Bestimmung der umgesetzten Energien durch die Flächenbestimmung unterhalb des Kurvenverlaufes und der Zusammenhang der Festigkeit und der Steigung zwischen den Extrempunkten der Kraftdeformationskurven sind erklärt worden.

Nachdem alle nötigen Einstellungsparameter gefunden worden sind und ein Überblick über das Material und sein zu erwartendes Reaktionsverhalten verschafft worden ist, ist ein Messaufbau entworfen worden, der die Realität nicht zu 100% widerspiegelt, aber gerade auf die Gesichtspunkte der verarbeiteten Energien und der Veränderung der Festigkeit eine mehr als ausreichende Rahmenbedingung für die Messung geschaffen hat.

Der verwendete „Ghost 11“ von dem Hersteller Brooks wurde anschließend zwei Messungen im Rückfußbereich der Sohle unterzogen. In der ersten Messung sind Kraftmaxima, die einem Läufer mit einem Gewicht von $70,8\text{ kg}$ entsprechen, umgesetzt worden. Für die zweite Messung wurde eine 20% leichtere Person

simuliert. Die angestrebten Simulationsparameter sind relativ gut in die Prüfmaschine implementiert worden, für eine bessere Umsetzung wäre ein umfassenderer Kenntnisstand über PID-Reglersysteme von Nöten gewesen.

Interessant war die Erkenntnis, dass sich die Festigkeit in den ersten zwei Dritteln der Messung verringerte und erst im letzten Abschnitt, der Messung mit ca. 1000N sich wieder, wie zu Beginn erwartet, erhöhte. Darüber hinaus konnte man einen absoluten Höhenverlust von 1,22 mm in dieser Messung beobachten. Die Messung unter ca. 800 N hingegen hatte sogar einen Höhenverlust von 1,36 mm. Diese Deformationen werden in der Zerstörung der Querbrücken der Wabenstruktur des Schaummaterials, gekoppelt mit einem Herausdrücken der Luft einschüsse aus den Waben, begründet sein. Man könnte für zukünftige Arbeiten analysieren ob sich diese Kammern wieder mit Luft befüllen und der Schuh so wieder eine vergleichbare Festigkeit einnehmen würde.

Darüber hinaus ist ein immer geringer werdender Unterschied zwischen der aufgenommenen und der abgegebenen Energiemenge beobachtet worden. Bei der Messung unter geringerer Last ist auffällig, dass im Verlaufe der Messung keine Minderung der aufgenommenen Energie beobachtet worden ist. Bei der Messung unter höherer Last ist jedoch ein Rückgang zu beobachten. Dies spricht für eine fortschreitende Materialermüdung.

Auch bei diesem Aspekt wäre es interessant zu sehen, ob der selbe Schuh unter derselben Last, nach einer gewissen Ruhephase, ein ähnliches Reaktionsverhalten gezeigt hätte.

Abschließend lässt sich sagen, dass das Arbeitsziel, ein aussagekräftiges Messverfahren zur Analyse des Reaktionsverhaltens der Mittelsohle von Laufschuhen, zu entwickeln, trotz einiger Abstriche in puncto Reproduzierbarkeit, voll erreicht worden ist. Es ist ein Matlabskript verfasst worden, das in der Lage ist alle zukünftig entstehenden Messungen vergleichend gegenüberzustellen und somit eine detailliertere Ergebnisübersicht zu erstellen, als es mit dem bisherigen Goldstandard der Messung nach dem „ASTM F1976“-Standard möglich war. Auch wenn die Analyse eines einzelnen Schuhs sehr viel mehr Zeit, als der Impulstest nach dem bisherigen Standard in Anspruch nimmt, hat der Endverbraucher mit

diesem neuen Test ein deutlich ausgeprägteres Informationsprofil über die Besonderheiten der aktuell verfügbaren Schuhmodelle als es bisher möglich war. Auch für die Entwicklung neuer Schuhmodelle könnte diese Art der Analyse interessant werden, da man gezielt Schuhe, für Athleten mit einem gewissen Gewicht und einer Sportart die über eine ungefähre Dauer betrieben wird, entwerfen könnte. Die Mittelsohlen der verwendeten Schuhe könnten so angepasst werden, dass der Schuh die erbrachte Leistung optimal unterstützen könnte.

Um dies zu ermöglichen müsste man in Zukunft mehr Schuhmodelle unter verschiedenen Belastungsbedingungen untersuchen. Jedoch ist bereits zu diesem Zeitpunkt ein wichtiger großer Schritt hin zur detaillierten, realitätsnahen Belastungsanalyse für Mittelsohlen verschiedener Laufschuhmodellen gemacht worden.

Tabellenverzeichnis

Tabelle 1: Zusammenfassung der mit Hilfe von Matlab ausgelesenen Werte	30
Tabelle 2: Zusammenfassung der recherchierten Prüfungsmerkmale	30
Tabelle 3: Zusammenfassung Messeinstellungen	37

Abbildungsverzeichnis

Abbildung 1: Der knöchernen Fuß (Zimmer & Appell, 2021, S.94f.)	6
Abbildung 2: Gangzyklus (Beckers & Deckers, 1997, S.19)	7
Abbildung 3: Unterschied Gang- & Lauzyklus (Novacheck, 1998, S.79).....	8
Abbildung 4: Laufstile (Ajad Yarra & Pandya, 2020)	10
Abbildung 5: Aufbau von Laufschuhen (Beus, 2020).....	11
Abbildung 6: Aufbau Polymer (Weißbach et al., 2015, S.364)	13
Abbildung 7: Polymerarten (Weißbach et al., 2015, S.365).....	13
Abbildung 8: Vergleich E-Modul (Weißbach et al., 2015, S.505 & S.369)	14
Abbildung 9: Schaumzelle und Schaumzellenverbund (Menges & Knipschild, 1975, S.623)	15
Abbildung 10: Kompression Schaumzelle (Menges & Knipschild, 1975, S.624) ...	15
Abbildung 11: Reaktionsverhalten verschiedener Ermüdungsversuche (Ohrndorf, 2011, S.17)	16
Abbildung 12: Entstehung der hystereseförmigen Kraftdeformationskurve (Ohrndorf, 2011, S.18).....	17
Abbildung 13: Dehnmessstreifen (Tkotz & Bastian, 2008, S.178)	18
Abbildung 14: Prinzip eines Piezokristalls (Tkotz & Bastian, 2008, S.178)	19
Abbildung 15: Typische Bodenreaktionskraftkurve (Novacheck, 1998, S.85)	19
Abbildung 16: BRK Vorfußläufer (Clark et al., 2014, S.2038).....	20
Abbildung 17: BRK in Abhängigkeit von Geschwindigkeiten (Clark et al., 2016, S.252)	20
Abbildung 18: Überlagerung der Kräfte (Shorten & Mientjes, 2011, S.46)	21
Abbildung 19: Aufbau Laufsteg (Alexis Brinkemper, 2013, S.18).....	23
Abbildung 20: Visualisierung der gemessenen Daten	24
Abbildung 21: Ergebnisse der Matlabauswertung	25
Abbildung 22: Schrittfrequenz I (Weyand et al., 2000, S.1993).....	26
Abbildung 23: Schrittfrequenz II (Cavagna et al., 1991, S.86)	26
Abbildung 24: Schrittfrequenz III (BAILEY et al., 2017, S.1071)	27
Abbildung 25: Zwick/Roell LTM10	32
Abbildung 26: Grundplatte und Prüfdorn	33
Abbildung 27: Leisten.....	34
Abbildung 28: Eingespannter Schuh unbelastet	34

Abbildung 29: Eingespannter Schuh im Rückfuß belastet	35
Abbildung 30: Programmoberfläche zum Einstellen der Prüfmaschine	36
Abbildung 31: Datenstruktur im Vergleich	38
Abbildung 32: Übersicht der Auswertungsaspekte	40
Abbildung 33: Ergebnisse der Zeitanalysen für die Messung mit 1000N	42
Abbildung 34: Zeitliche Veränderung der Deformation und Kraft für die Messung mit 1000N	44
Abbildung 35: Übersicht der auftretenden Energien für die Messung mit 1000N ..	45
Abbildung 36: Steigung der Kraftdeformationskurve im Verlauf des Messvorganges für die Messung mit 1000N	46
Abbildung 37: Entwicklung der Kraftdeformationskurve von der Messung mit 1000N	47
Abbildung 38: Ergebnisse der Zeitanalysen für die Messung mit 800N.....	48
Abbildung 39: Zeitliche Veränderung der Deformation und Kraft für die Messung mit 800N	50
Abbildung 40: Übersicht der auftretenden Energien für die Messung mit 800N	51
Abbildung 41: Steigung der Kraftdeformationskurve im Verlauf des Messvorganges für die Messung mit 800N	52
Abbildung 42: Entwicklung der Kraftdeformationskurve von der Messung mit 800N	53
Abbildung 43: Vergleich der zeitlichen Veränderung der Deformation und Kraft beider Messungen.....	54
Abbildung 44: Vergleich der auftretenden Energien für beide Messungen.....	56
Abbildung 45: Vergleich der Steigungen der Kraftdeformationskurve beider Messungen	57
Abbildung 46: Energierückgabe verschiedener Laufschuhmodelle (Hoogkamer et al., 2018, S.1012)	65

Literaturverzeichnis

- Adidas. (2021). *Adidas Ultraboost 21 Tokyo Laufschuh*. adidas Deutschland.
<https://www.adidas.de/ultraboost-21-tokyo-laufschuh/S23863.html>
- Ajad Yarra, A., & Pandya, A. (2020, November 27). *Foot strike patterns in runners*.
<http://geeksonfeet.com/run/footstrike/>
- Alexis Brinkemper. (2013). *Die Interaktion zwischen Mensch und Untergrund für den Vor- und Rückfuß beim Laufen*. Deutsche Sporthochschule Köln.
- ASTM F1976. (2013, Mai 1). *ASTM F1976—Standard Test Method for Impact Attenuation of Athletic Shoe Cushioning Systems and Materials | Engineering360*.
<https://standards.globalspec.com/std/1627213/ASTM%20F1976>
- BAILEY, J., MATA, T., & MERCER, J. A. (2017). Is the Relationship Between Stride Length, Frequency, and Velocity Influenced by Running on a Treadmill or Overground? *International Journal of Exercise Science*, 10(7), 1067–1075.
- Beckers, D., & Deckers, J. (1997). *Ganganalyse und Gangschulung* (Bd. 38). Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-642-59072-6>
- Beus, M. (2020). Running Shoe Anatomy. *Anatomy Diagram Book*.
<https://grekoulas.blogspot.com/2019/09/running-shoe-anatomy.html>
- Bobbert, M. F., & Schamhardt, H. C. (1990). Accuracy of determining the point of force application with piezoelectric force plates. *Journal of Biomechanics*, 23(7), 705–710. [https://doi.org/10.1016/0021-9290\(90\)90169-4](https://doi.org/10.1016/0021-9290(90)90169-4)
- Brooks. (2021a). *Ghost 14: Straßenlaufschuhe mit Dämpfung für Herren*. Brooks EMEA. https://www.brooksrunning.com/de_de/ghost-14-mens-cushioned-road-running-shoe/110369.html

- Brooks. (2021b). *Was ist die DNA LOFT Dämpfung von Brooks? | Brooks Running*.
Brooks EMEA. https://www.brooksrunning.com/de_de/blog/research-innovation/science-behind-brooks-dna-loft.html
- Cavagna, G. A., Franzetti, P., Heglund, N. C., & Willems, P. (1988). The determinants of the step frequency in running, trotting and hopping in man and other vertebrates. *The Journal of Physiology*, 399(1), 81–92. <https://doi.org/10.1113/jphysiol.1988.sp017069>
- Clark, K. P., Ryan, L. J., & Weyand, P. G. (2016). A general relationship links gait mechanics and running ground reaction forces. *Journal of Experimental Biology*, jeb.138057. <https://doi.org/10.1242/jeb.138057>
- Gross, D., Hauger, W., Schröder, J., & Wall, W. A. (2012). *Technische Mechanik 3*. Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-642-29529-4>
- Hoogkamer, W., Kipp, S., Frank, J. H., Farina, E. M., Luo, G., & Kram, R. (2018). A Comparison of the Energetic Cost of Running in Marathon Racing Shoes. *Sports Medicine*, 48(4), 1009–1019. <https://doi.org/10.1007/s40279-017-0811-2>
- Katzer, N., & Rohdewald, S. (2017). *Sport, Wissenschaft und Technik*. 34.
- Kirchhoff, A., Götz, F., & Biskamp, N. (2013, Juni 24). *WP1304 Laufschuh [Institut für Sportwissenschaft]*. <http://wiki.ifs-tud.de/biomechanik/projekte/ss2013/laufschuh>
- Menges, G., & Knipschild, F. (1975). Estimation of mechanical properties for rigid polyurethane foams. *Polymer Engineering & Science*, 15(8), 623–627. <https://doi.org/10.1002/pen.760150810>
- Novacheck, T. F. (1998). The biomechanics of running. *Gait and Posture*, 19.

- Ohrndorf, A. (2011). *Verformungsverhalten metallischer Schaumwerkstoffe unter isothermen und thermomechanischen Beanspruchungsbedingungen*. 244.
- On. (2021). *Laufen wie auf Wolken: Die Wissenschaft hinter dem ultimativen Laufgefühl*. <https://www.on-running.com/de-de/articles/running-on-clouds-the-science-behind-the-sensation>
- Schulke, H.-J. (2007). *Technik und Sport 2*. AG sportwissenschaftlicher Hochschuleinrichtungen.
- Shorten, M., & Mientjes, M. I. V. (2011). The 'heel impact' force peak during running is neither 'heel' nor 'impact' and does not quantify shoe cushioning effects. *Footwear Science*, 3(1), 41–58. <https://doi.org/10.1080/19424280.2010.542186>
- Spanka, S. (2021). *Laufschuhe Aufbau* | *Joggen.net*. <https://www.joggen.net/laufschuhe/aufbau/>
- Subic, J. (2021, März 18). *Heel to Toe Drop: The Ultimate Guide*. Athletic Shoe Reviews. <https://runrepeat.com/guides/heel-to-toe-drop>
- Tkocz, K., & Bastian, P. (Hrsg.). (2008). *Fachkunde Elektrotechnik: Bilder & weitere Informationen auf DVD* (26., überarb. und erw. Aufl). Verl. Europa-Lehrmittel Nourney, Vollmer.
- Tung, K. D., Franz, J. R., & Kram, R. (2014). A Test of the Metabolic Cost of Cushioning Hypothesis during Unshod and Shod Running. *Medicine & Science in Sports & Exercise*, 46(2), 324–329. <https://doi.org/10.1249/MSS.0b013e3182a63b81>
- Van den Berghe, P., Breine, B., Haeck, E., & De Clercq, D. (2021). One hundred marathons in 100 days: Unique biomechanical signature and the evolution of

- force characteristics and bone density. *Journal of Sport and Health Science*, S2095254621000442. <https://doi.org/10.1016/j.jshs.2021.03.009>
- Walpole, S. C., Prieto-Merino, D., Edwards, P., Cleland, J., Stevens, G., & Roberts, I. (2012). The weight of nations: An estimation of adult human biomass. *BMC Public Health*, 12(1), 439. <https://doi.org/10.1186/1471-2458-12-439>
- Weißbach, W., Dahms, M., & Jaroschek, C. (2015). *Werkstoffkunde*. Springer Fachmedien Wiesbaden. <https://doi.org/10.1007/978-3-658-03919-6>
- Weyand, P. G., Sternlight, D. B., Bellizzi, M. J., & Wright, S. (2000). Faster top running speeds are achieved with greater ground forces not more rapid leg movements. *Journal of Applied Physiology*, 89(5), 1991–1999. <https://doi.org/10.1152/jappl.2000.89.5.1991>
- Zimmer, P., & Appell, H.-J. (2021). *Funktionelle Anatomie: Grundlagen sportlicher Leistung und Bewegung*. Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-662-61482-2>
- Zwick/Roell. (o. J.). *Elektro dynamische Pruefmaschine LTM10*. Abgerufen 19. Juli 2021, von https://www.zwickroell.com/fileadmin/content/Files/SharePoint/user_upload/PI_DE/15_969_Elektro_dynamische_Pruefmaschine_LTM_5_und_LTM_10_DE.pdf

Anhang

Matlabskript zur Analyse der Rohdaten:

```
clc
clear
close all
%% import the resultdata
result = importdata('Test2.tra'); %importing the resultdata
%% starting conditions
k = 1; % a helpvariable
n = 3; %amount of splits in graph
% set the summ of each limit to zero
%rise
summ_x_limit_left_rise(1,n) = zeros;
summ_x_limit_right_rise(1,n) = zeros;
summ_y_limit_left_rise(1,n) = zeros;
summ_y_limit_right_rise(1,n) = zeros;
summ_slope_rise(1,n) = zeros;
%fall
summ_x_limit_left_fall(1,n) = zeros;
summ_x_limit_right_fall(1,n) = zeros;
summ_y_limit_left_fall(1,n) = zeros;
summ_y_limit_right_fall(1,n) = zeros;
summ_slope_fall(1,n) = zeros;
%average energy
summ_energy_rise = 0;
summ_energy_fall = 0;
%% look for the startframes of the different testingloops
for ( i = 1:length(result.data(:,1)) )
    if (result.data(i,2) == k) && (k == 1) %get the startingposition of
the first testingloop
        start_loop(k,1) = i;
        k = k + 1;
    elseif (result.data(i,2) == k) %get the startingposition for each of
the other loops
        start_loop(k,1) = i;
        k = k + 1;
    elseif (result.data(i,2) < k) %if the startingpoint for the last loop
is determined the value "k" will be greater than the amount of
testingloops and every position of the matrix will be marked, but when
the loop ends there will be only the last position saved
        start_loop(k,1) = i;
    end
end
end
%% look for the extrema
for ( j = 1:max(result.data(:,2)))
    %max. deformation each testingloop
    max_deformation(j,1) =
max(result.data(start_loop(j,1):start_loop(j+1,1),4)); %get the max
deformation in each testingloop
    position_max_def(j,1) = find(result.data(:,4) ==
max_deformation(j,1)); %get the position of the maximum deformation in
the overall resultmatrix
    %rise
    start_rise(j,1) = start_loop(j,1); %set the start of each rising part
of the testingloops at the position of each startingposition
```

```

    end_rise(j,1) = position_max_def(j,1); %set the end of each rising
part of the testingloops at the position of each max
    length_rise(j,1) = end_rise(j,1) - start_rise(j,1); %get the amount
of frames of the rise for the overall length
    steps_rise(j,1) = length_rise(j,1) / n; %divide the length by the
amount of analysissteps for the amount of frames each analysisstep will
take
    steps_rise_rounded(j,1) = int32(steps_rise(j,1)); %round the
analysissteplength to make them usable for the frames
    for ( m = 1:n ) %analysis of the rising part
        position_limit_rise_left(j,m) = start_rise(j,1) + ( (m-1) *
steps_rise_rounded(j,1)); %get each left limit with the startingpoints
and the added analysissteplength
        position_limit_rise_right(j,m) = start_rise(j,1) + ( m *
steps_rise_rounded(j,1)); %get each right limit with the startingpoints
and the added analysissteplength
        x_limit_left_rise(j,m) =
result.data(position_limit_rise_left(j,m),4); %set the value of the limit
        x_limit_right_rise(j,m) =
result.data(position_limit_rise_right(j,m),4); %set the value of the
limit
        y_limit_left_rise(j,m) =
result.data(position_limit_rise_left(j,m),3); %set the value of the limit
        y_limit_right_rise(j,m) =
result.data(position_limit_rise_right(j,m),3); %set the value of the
limit
        %
        x_rise = [x_limit_left_rise(j,m), x_limit_right_rise(j,m)];
%create a x-vector for the following polyfit
        y_rise = [y_limit_left_rise(j,m), y_limit_right_rise(j,m)];
%create a y-vector for the following polyfit
        p_r = polyfit(x_rise,y_rise,1); %fit a straight line between the
left and right limit
        p_rise(j,m) = p_r(1,1); %a helpvalue for saving the exact values
for each step of the loops
        p_rise(j,m+n) = p_r(1,2); %a helpvalue for saving the exact
values for each step of the loops
        p_help = [p_rise(j,m), p_rise(j,m+n)]; %create a vector for the
plotting
        y1_rise = polyval(p_help,x_rise); %get the y-values for the
plotting
        %
        slope_rise(j,m) = p_r(1,1); %the first value of the vectors are
the slope
        %
        hold on
        plot(x_rise,y1_rise,'-*');
    end
    %fall
    start_fall(j,1) = position_max_def(j,1); %set the start of each
falling part of the testingloops at the position of each max
    end_fall(j,1) = start_loop(j+1) - 1; %set the end of each falling
part of the testingloops at the position the start of the next
testingloop - 1
    length_fall(j,1) = end_fall(j,1) - start_fall(j,1); %get the amount
of frames of the fall for the overall length
    steps_fall(j,1) = length_fall(j,1) / n; %divide the length by the
amount of analysissteps for the amount of frames each analysisstep will
take
    steps_fall_rounded(j,1) = int32(steps_fall(j,1)); %round the
analysissteplength to make them usable for the frames

```



```

    for ( m = 1:n ) %analysis of the falling part
        position_limit_left_fall(j,m) = end_fall(j,1) - ( (m-1) *
steps_fall_rounded(j,1)); %get each left limit with the endpoints and
the subtracted analysissteplength
        position_limit_right_fall(j,m) = end_fall(j,1) - ( m *
steps_fall_rounded(j,1)); %get each left limit with the endpoints and
the subtracted analysissteplength
        x_limit_left_fall(j,m) =
result.data(position_limit_left_fall(j,m),4); %set the value of the limit
        x_limit_right_fall(j,m) =
result.data(position_limit_right_fall(j,m),4); %set the value of the
limit
        y_limit_left_fall(j,m) =
result.data(position_limit_left_fall(j,m),3); %set the value of the limit
        y_limit_right_fall(j,m) =
result.data(position_limit_right_fall(j,m),3); %set the value of the
limit
        %
        x_fall = [x_limit_left_fall(j,m), x_limit_right_fall(j,m)];
%create a x-vector for the following polyfit
        y_fall = [y_limit_left_fall(j,m), y_limit_right_fall(j,m)];
%create a x-vector for the following polyfit
        p_f = polyfit(x_fall,y_fall,1); %fit a straight line between the
left and right limit
        p_fall(j,m) = p_f(1,1); %a helpvalue for saving the exact values
for each step of the loops
        p_fall(j,m+n) = p_f(1,2); %a helpvalue for saving the exact
values for each step of the loops
        p_help_fall = [p_fall(j,m), p_fall(j,m+n)]; %create a vector for
the plotting
        y1_fall = polyval(p_help_fall,x_fall); %get the y-values for the
plotting
        % slope of each splitted part of the curve
        slope_fall(j,m) = p_f(1,1); %the first value of the vectors are
the slope
        % plot
        hold on
        plot(x_fall,y1_fall,'-*');
    end
end
%% get the summ of the limits & slope
for ( d = 1:n )
    for ( e = 1:j )
        %rise
        % x_limits_summ
        summ_x_limit_left_rise(1,d) = summ_x_limit_left_rise(1,d) +
x_limit_left_rise(e,d); %add each loop the value of the limit for an
overall summ
        summ_x_limit_right_rise(1,d) = summ_x_limit_right_rise(1,d) +
x_limit_right_rise(e,d); %add each loop the value of the limit for an
overall summ
        % y_limits_summ
        summ_y_limit_left_rise(1,d) = summ_y_limit_left_rise(1,d) +
y_limit_left_rise(e,d); %add each loop the value of the limit for an
overall summ
        summ_y_limit_right_rise(1,d) = summ_y_limit_right_rise(1,d) +
y_limit_right_rise(e,d); %add each loop the value of the limit for an
overall summ
        % slope_rise
        summ_slope_rise(1,d) = summ_slope_rise(1,d) + slope_rise(e,d);
%add each loop the value of the slope for an overall summ

```

```

        %fall
        % x_limits_summ
        summ_x_limit_left_fall(1,d) = summ_x_limit_left_fall(1,d) +
x_limit_left_fall(e,d); %add each loop the value of the limit for an
overall summ
        summ_x_limit_right_fall(1,d) = summ_x_limit_right_fall(1,d) +
x_limit_right_fall(e,d); %add each loop the value of the limit for an
overall summ
        % y_limits_summ
        summ_y_limit_left_fall(1,d) = summ_y_limit_left_fall(1,d) +
y_limit_left_fall(e,d); %add each loop the value of the limit for an
overall summ
        summ_y_limit_right_fall(1,d) = summ_y_limit_right_fall(1,d) +
y_limit_right_fall(e,d); %add each loop the value of the limit for an
overall summ
        % slope_rise
        summ_slope_fall(1,d) = summ_slope_fall(1,d) + slope_fall(e,d);
%add each loop the value of the slope for an overall summ
    end
end
%% create an average curve
for ( o = 1:n)
    %rise
    %limits
    av_x_limit_left_rise(1,o) = summ_x_limit_left_rise(1,o) / j; %divide
the summ of each limits trough the amaount of testingloops to get an
average
    av_x_limit_right_rise(1,o) = summ_x_limit_right_rise(1,o) / j;
%divide the summ of each limits trough the amaount of testingloops to get
an average
    av_y_limit_left_rise(1,o) = summ_y_limit_left_rise(1,o) /j; %divide
the summ of each limits trough the amaount of testingloops to get an
average
    av_y_limit_right_rise(1,o) = summ_y_limit_right_rise(1,o) /j; %divide
the summ of each limits trough the amaount of testingloops to get an
average
    %slope
    av_slope_rise(1,o) = summ_slope_rise(1,o) / j; %divide the summ of
each slope trough the amaount of testingloops to get an average
    %
    %fall
    %limits
    av_x_limit_left_fall(1,o) = summ_x_limit_left_fall(1,o) / j; %divide
the summ of each limits trough the amaount of testingloops to get an
average
    av_x_limit_right_fall(1,o) = summ_x_limit_right_fall(1,o) / j;
%divide the summ of each limits trough the amaount of testingloops to get
an average
    av_y_limit_left_fall(1,o) = summ_y_limit_left_fall(1,o) /j; %divide
the summ of each limits trough the amaount of testingloops to get an
average
    av_y_limit_right_fall(1,o) = summ_y_limit_right_fall(1,o) /j; %divide
the summ of each limits trough the amaount of testingloops to get an
average
    %slope
    av_slope_fall(1,o) = summ_slope_fall(1,o) / j; %divide the summ of
each slope trough the amaount of testingloops to get an average
    end
%% integral of each testingloop
for ( c = 1:j)
    %rise

```

```

        rising_part_each_loop(c,1) = start_loop(c,1); %start rising part
at each start of the testingloops
        rising_part_each_loop(c,2) = position_max_def(c,1); %end rising
part at each individual max of the testingloops
        integral_x_rise =
result.data(rising_part_each_loop(c,1):rising_part_each_loop(c,2),4);
%get a matrix with all x-values of the rise
        integral_y_rise =
result.data(rising_part_each_loop(c,1):rising_part_each_loop(c,2),3);
%get a matrix with all y-values of the rise
        energy_rise(c,1) = trapz(integral_x_rise, integral_y_rise); %get
the integral between the x and y axis of the rise
        %fall
        falling_part_each_loop(c,1) = position_max_def(c,1); %start
falling part at each max of the testingloop
        falling_part_each_loop(c,2) = start_loop(c+1,1) - 1; %end rising
part = startingposition next loop - 1.
        integral_x_fall =
result.data(falling_part_each_loop(c,1):falling_part_each_loop(c,2),4);
%get a matrix with all x-values of the rise
        integral_y_fall =
result.data(falling_part_each_loop(c,1):falling_part_each_loop(c,2),3);
%get a matrix with all y-values of the rise
        energy_fall(c,1) = trapz(integral_x_fall, integral_y_fall); %get
the integral between the x and y axis of the fall
        % difference
        energy_diff(c,1) = energy_rise(c,1) + energy_fall(c,1); %get the
difference between the two integrals, the falling part is negative
because of that there is a "+"
        %create average
        summ_energy_rise = summ_energy_rise + energy_rise(c,1); %add each
value of the energy of the rise for an overall summ
        summ_energy_fall = summ_energy_fall + energy_fall(c,1); %add each
value of the energyof the fall for an overall summ
        if ( c == j)
            av_energy_rise = summ_energy_rise / j; %when all is added
divide by the amaount of testingloops for an average
            av_energy_fall = summ_energy_fall / j; %when all is added
divide by the amaount of testingloops for an average
        end
    end
%% plots
    x0=200;
    y0=100;
    width=950;
    height=650;
    %
    hold on
    plot(result.data(:,4),result.data(:,3))
    ylabel('force [N]');
    xlabel('deformation [mm]');
    title('visualization of the results and the analysissteps');
    legend('resultdata','analysissteps','location','northwest');
    grid on
    set(gcf,'position',[x0,y0,width,height])
    %
    figure %plot of the energy
    subplot(2,2,1)
    bar(1:j,energy_rise,'r')
    xlabel('energy');
    ylabel('testingloop')

```

```

title('energy taken on the rise of each loop')
legend('energy','location','northeast')
grid on
subplot(2,2,2)
bar(1:j,energy_fall,'b')
xlabel('energy');
ylabel('testingloop')
title('energy given on the fall of each loop')
legend('energy','location','northwest')
grid on
subplot(2,2,3)
bar(1:j,energy_diff,'k')
xlabel('energy')
ylabel('testingloop')
title('energydifference between taken and given')
legend('energydiff','location','northeast')
grid on
subplot(2,2,4)
hold on
plot(av_x_limit_left_rise,av_y_limit_left_rise,'-*')
plot(av_x_limit_right_rise,av_y_limit_right_rise,'-*')
plot(av_x_limit_left_fall,av_y_limit_left_fall,'-*')
plot(av_x_limit_right_fall,av_y_limit_right_fall,'-*')
ylabel('force [N]');
xlabel('deformation [mm]');
title('average curve')
legend off
grid on
set(gcf,'position',[x0,y0,width,height])
%
figure %plot of the slope
subplot(2,2,1)
bar(1:n, av_slope_rise,'r')
ylabel('slope');
xlabel('analysissteps');
title('slope on the rising parts');
legend('slope','location','northwest');
grid on
subplot(2,2,2)
bar(1:n,av_slope_fall,'b')
ylabel('slope');
xlabel('analysissteps');
title('slope on the falling parts');
legend('slope','location','northwest');
grid on
subplot(2,2,4)
hold on
plot(av_x_limit_left_rise,av_y_limit_left_rise,'-*')
plot(av_x_limit_right_rise,av_y_limit_right_rise,'-*')
plot(av_x_limit_left_fall,av_y_limit_left_fall,'-*')
plot(av_x_limit_right_fall,av_y_limit_right_fall,'-*')
ylabel('force [N]');
xlabel('deformation [mm]');
title('average curve')
legend off
grid on
set(gcf,'position',[x0,y0,width,height])
%
figure
hold on
plot(av_x_limit_left_rise,av_y_limit_left_rise,'-*')

```

```

plot(av_x_limit_right_rise,av_y_limit_right_rise,'-*')
plot(av_x_limit_left_fall,av_y_limit_left_fall,'-*')
plot(av_x_limit_right_fall,av_y_limit_right_fall,'-*')
ylabel('force [N]');
xlabel('deformation [mm]');
title('average curve')
legend off
grid on
set(gcf,'position',[x0,y0,width,height])
%
figure
plot(result.data(:,4),result.data(:,3))
ylabel('force [N]');
xlabel('deformation [mm]');
title('resultdata')
grid on
set(gcf,'position',[x0,y0,width,height])

```

Matlabskript zur Vorbereitung der Messung:

```

clc
clear
close all
% import the resultdata
result = importdata('Test2.tra'); %importing the resultdata
% starting conditions
k = 1; % a helpvariable
n = 3; %amount of splits in graph
% set the summ of each limit to zero
%rise
summ_x_limit_left_rise(1,n) = zeros;
summ_x_limit_right_rise(1,n) = zeros;
summ_y_limit_left_rise(1,n) = zeros;
summ_y_limit_right_rise(1,n) = zeros;
summ_slope_rise(1,n) = zeros;
%fall
summ_x_limit_left_fall(1,n) = zeros;
summ_x_limit_right_fall(1,n) = zeros;
summ_y_limit_left_fall(1,n) = zeros;
summ_y_limit_right_fall(1,n) = zeros;
summ_slope_fall(1,n) = zeros;
%average energy
summ_energy_rise = 0;
summ_energy_fall = 0;
% look for the startframes of the different testingloops
for ( i = 1:length(result.data(:,1)) )
    if (result.data(i,2) == k) && (k == 1) %get the startingposition of
the first testingloop
        start_loop(k,1) = i;
        k = k + 1;
    elseif (result.data(i,2) == k) %get the startingposition for each of
the other loops
        start_loop(k,1) = i;
        k = k + 1;
    elseif (result.data(i,2) < k) %if the startingpoint for the last loop
is determined the value "k" will be greater than the amount of
testingloops and every position of the matrix will be marked, but when
the loop ends there will be only the last position saved
        start_loop(k,1) = i;
    end
end

```

```

end
%% look for the extrema
for ( j = 1:max(result.data(:,2)))
    %max. deformation each testingloop
    max_deformation(j,1) =
max(result.data(start_loop(j,1):start_loop(j+1,1),4)); %get the max
deformation in each testingloop
    position_max_def(j,1) = find(result.data(:,4) ==
max_deformation(j,1)); %get the position of the maximum deformation in
the overall resultmatrix
    %rise
    start_rise(j,1) = start_loop(j,1); %set the start of each rising part
of the testingloops at the position of each startingposition
    end_rise(j,1) = position_max_def(j,1); %set the end of each rising
part of the testingloops at the position of each max
    length_rise(j,1) = end_rise(j,1) - start_rise(j,1); %get the amount
of frames of the rise for the overall length
    steps_rise(j,1) = length_rise(j,1) / n; %divide the length by the
amount of analysissteps for the amount of frames each analysisstep will
take
    steps_rise_rounded(j,1) = int32(steps_rise(j,1)); %round the
analysissteplength to make them usable for the frames
    for ( m = 1:n ) %analysis of the rising part
        position_limit_rise_left(j,m) = start_rise(j,1) + ( (m-1) *
steps_rise_rounded(j,1)); %get each left limit with the startingpoints
and the added analysissteplength
        position_limit_rise_right(j,m) = start_rise(j,1) + ( m *
steps_rise_rounded(j,1)); %get each right limit with the startingpoints
and the added analysissteplength
        x_limit_left_rise(j,m) =
result.data(position_limit_rise_left(j,m),4); %set the value of the limit
        x_limit_right_rise(j,m) =
result.data(position_limit_rise_right(j,m),4); %set the value of the
limit
        y_limit_left_rise(j,m) =
result.data(position_limit_rise_left(j,m),3); %set the value of the limit
        y_limit_right_rise(j,m) =
result.data(position_limit_rise_right(j,m),3); %set the value of the
limit
        %
        x_rise = [x_limit_left_rise(j,m), x_limit_right_rise(j,m)];
%create a x-vector for the following polyfit
        y_rise = [y_limit_left_rise(j,m), y_limit_right_rise(j,m)];
%create a y-vector for the following polyfit
        p_r = polyfit(x_rise,y_rise,1); %fit a straight line between the
left and right limit
        p_rise(j,m) = p_r(1,1); %a helpvalue for saving the exact values
for each step of the loops
        p_rise(j,m+n) = p_r(1,2); %a helpvalue for saving the exact
values for each step of the loops
        p_help = [p_rise(j,m), p_rise(j,m+n)]; %create a vector for the
plotting
        y1_rise = polyval(p_help,x_rise); %get the y-values for the
plotting
        %
        slope_rise(j,m) = p_r(1,1); %the first value of the vectors are
the slope
        %
        hold on
        plot(x_rise,y1_rise,'-*');
    end
end

```

```

    %fall
    start_fall(j,1) = position_max_def(j,1); %set the start of each
falling part of the testingloops at the position of each max
    end_fall(j,1) = start_loop(j+1) - 1; %set the end of each falling
part of the testingloops at the position the start of the next
testingloop - 1
    length_fall(j,1) = end_fall(j,1) - start_fall(j,1); %get the amount
of frames of the fall for the overall length
    steps_fall(j,1) = length_fall(j,1) / n; %divide the length by the
amount of analysissteps for the amount of frames each analysisstep will
take
    steps_fall_rounded(j,1) = int32(steps_fall(j,1)); %round the
analysissteplength to make them usable for the frames
    for ( m = 1:n ) %analysis of the falling part
        position_limit_left_fall(j,m) = end_fall(j,1) - ( (m-1) *
steps_fall_rounded(j,1)); %get each left limit with the endingpoints and
the subtracted analysissteplength
        position_limit_right_fall(j,m) = end_fall(j,1) - ( m *
steps_fall_rounded(j,1)); %get each left limit with the endingpoints and
the subtracted analysissteplength
        x_limit_left_fall(j,m) =
result.data(position_limit_left_fall(j,m),4); %set the value of the limit
        x_limit_right_fall(j,m) =
result.data(position_limit_right_fall(j,m),4); %set the value of the
limit
        y_limit_left_fall(j,m) =
result.data(position_limit_left_fall(j,m),3); %set the value of the limit
        y_limit_right_fall(j,m) =
result.data(position_limit_right_fall(j,m),3); %set the value of the
limit
        %
        x_fall = [x_limit_left_fall(j,m), x_limit_right_fall(j,m)];
%create a x-vector for the following polyfit
        y_fall = [y_limit_left_fall(j,m), y_limit_right_fall(j,m)];
%create a x-vector for the following polyfit
        p_f = polyfit(x_fall,y_fall,1); %fit a straight line between the
left and right limit
        p_fall(j,m) = p_f(1,1); %a helpvalue for saving the exact values
for each step of the loops
        p_fall(j,m+n) = p_f(1,2); %a helpvalue for saving the exact
values for each step of the loops
        p_help_fall = [p_fall(j,m), p_fall(j,m+n)]; %create a vector for
the plotting
        y1_fall = polyval(p_help_fall,x_fall); %get the y-values for the
plotting
        % slope of each splitted part of the curve
        slope_fall(j,m) = p_f(1,1); %the first value of the vectors are
the slope
        % plot
        hold on
        plot(x_fall,y1_fall,'-*');
    end
end
%% get the summ of the limits & slope
for ( d = 1:n )
    for ( e = 1:j )
        %rise
        % x_limits_summ
        summ_x_limit_left_rise(1,d) = summ_x_limit_left_rise(1,d) +
x_limit_left_rise(e,d); %add each loop the value of the limit for an
overall summ

```

```

        summ_x_limit_right_rise(1,d) = summ_x_limit_right_rise(1,d) +
x_limit_right_rise(e,d); %add each loop the value of the limit for an
overall summ
        % y_limits_summ
        summ_y_limit_left_rise(1,d) = summ_y_limit_left_rise(1,d) +
y_limit_left_rise(e,d); %add each loop the value of the limit for an
overall summ
        summ_y_limit_right_rise(1,d) = summ_y_limit_right_rise(1,d) +
y_limit_right_rise(e,d); %add each loop the value of the limit for an
overall summ
        % slope_rise
        summ_slope_rise(1,d) = summ_slope_rise(1,d) + slope_rise(e,d);
%add each loop the value of the slope for an overall summ
        %fall
        % x_limits_summ
        summ_x_limit_left_fall(1,d) = summ_x_limit_left_fall(1,d) +
x_limit_left_fall(e,d); %add each loop the value of the limit for an
overall summ
        summ_x_limit_right_fall(1,d) = summ_x_limit_right_fall(1,d) +
x_limit_right_fall(e,d); %add each loop the value of the limit for an
overall summ
        % y_limits_summ
        summ_y_limit_left_fall(1,d) = summ_y_limit_left_fall(1,d) +
y_limit_left_fall(e,d); %add each loop the value of the limit for an
overall summ
        summ_y_limit_right_fall(1,d) = summ_y_limit_right_fall(1,d) +
y_limit_right_fall(e,d); %add each loop the value of the limit for an
overall summ
        % slope_rise
        summ_slope_fall(1,d) = summ_slope_fall(1,d) + slope_fall(e,d);
%add each loop the value of the slope for an overall summ
    end
end
%% create an average curve
for ( o = 1:n)
    %rise
    %limits
    av_x_limit_left_rise(1,o) = summ_x_limit_left_rise(1,o) / j; %divide
the summ of each limits trough the amaount of testingloops to get an
average
    av_x_limit_right_rise(1,o) = summ_x_limit_right_rise(1,o) / j;
%divide the summ of each limits trough the amaount of testingloops to get
an average
    av_y_limit_left_rise(1,o) = summ_y_limit_left_rise(1,o) /j; %divide
the summ of each limits trough the amaount of testingloops to get an
average
    av_y_limit_right_rise(1,o) = summ_y_limit_right_rise(1,o) /j; %divide
the summ of each limits trough the amaount of testingloops to get an
average
    %slope
    av_slope_rise(1,o) = summ_slope_rise(1,o) / j; %divide the summ of
each slope trough the amaount of testingloops to get an average
    %
    %fall
    %limits
    av_x_limit_left_fall(1,o) = summ_x_limit_left_fall(1,o) / j; %divide
the summ of each limits trough the amaount of testingloops to get an
average
    av_x_limit_right_fall(1,o) = summ_x_limit_right_fall(1,o) / j;
%divide the summ of each limits trough the amaount of testingloops to get
an average

```



```

    av_y_limit_left_fall(1,o) = summ_y_limit_left_fall(1,o) /j; %divide
the summ of each limits trough the amaount of testingloops to get an
average
    av_y_limit_right_fall(1,o) = summ_y_limit_right_fall(1,o) /j; %divide
the summ of each limits trough the amaount of testingloops to get an
average
    %slope
    av_slope_fall(1,o) = summ_slope_fall(1,o) / j; %divide the summ of
each slope trough the amaount of testingloops to get an average
end
%% integral of each testingloop
for ( c = 1:j)
    %rise
    rising_part_each_loop(c,1) = start_loop(c,1); %start rising part
at each start of the testingloops
    rising_part_each_loop(c,2) = position_max_def(c,1); %end rising
part at each individual max of the testingloops
    integral_x_rise =
result.data(rising_part_each_loop(c,1):rising_part_each_loop(c,2),4);
%get a matrix with all x-values of the rise
    integral_y_rise =
result.data(rising_part_each_loop(c,1):rising_part_each_loop(c,2),3);
%get a matrix with all y-values of the rise
    energy_rise(c,1) = trapz(integral_x_rise, integral_y_rise); %get
the integral between the x and y axis of the rise
    %fall
    falling_part_each_loop(c,1) = position_max_def(c,1); %start
falling part at each max of the testingloop
    falling_part_each_loop(c,2) = start_loop(c+1,1) - 1; %end rising
part = startingposition next loop - 1.
    integral_x_fall =
result.data(falling_part_each_loop(c,1):falling_part_each_loop(c,2),4);
%get a matrix with all x-values of the rise
    integral_y_fall =
result.data(falling_part_each_loop(c,1):falling_part_each_loop(c,2),3);
%get a matrix with all y-values of the rise
    energy_fall(c,1) = trapz(integral_x_fall, integral_y_fall); %get
the integral between the x and y axis of the fall
    % difference
    energy_diff(c,1) = energy_rise(c,1) + energy_fall(c,1); %get the
difference between the two integrals, the falling part is negative
because of that there is a "+"
    %create average
    summ_energy_rise = summ_energy_rise + energy_rise(c,1); %add each
value of the energy of the rise for an overall summ
    summ_energy_fall = summ_energy_fall + energy_fall(c,1); %add each
value of the energyof the fall for an overall summ
    if ( c == j)
        av_energy_rise = summ_energy_rise / j; %when all is added
divide by the amaount of testingloops for an average
        av_energy_fall = summ_energy_fall / j; %when all is added
divide by the amaount of testingloops for an average
    end
end
%% plots
x0=200;
y0=100;
width=950;
height=650;
%
hold on

```

```

plot(result.data(:,4),result.data(:,3))
ylabel('force [N]');
xlabel('deformation [mm]');
title('visualization of the results and the analysissteps');
legend('resultdata','analysissteps','location','northwest');
grid on
set(gcf,'position',[x0,y0,width,height])
%
figure %plot of the energy
subplot(2,2,1)
bar(1:j,energy_rise,'r')
xlabel('energy');
ylabel('testingloop')
title('energy taken on the rise of each loop')
legend('energy','location','northeast')
grid on
subplot(2,2,2)
bar(1:j,energy_fall,'b')
xlabel('energy');
ylabel('testingloop')
title('energy given on the fall of each loop')
legend('energy','location','northwest')
grid on
subplot(2,2,3)
bar(1:j,energy_diff,'k')
xlabel('energy')
ylabel('testingloop')
title('energydifference between taken and given')
legend('energydiff','location','northeast')
grid on
subplot(2,2,4)
hold on
plot(av_x_limit_left_rise,av_y_limit_left_rise,'-*')
plot(av_x_limit_right_rise,av_y_limit_right_rise,'-*')
plot(av_x_limit_left_fall,av_y_limit_left_fall,'-*')
plot(av_x_limit_right_fall,av_y_limit_right_fall,'-*')
ylabel('force [N]');
xlabel('deformation [mm]');
title('average curve')
legend off
grid on
set(gcf,'position',[x0,y0,width,height])
%
figure %plot of the slope
subplot(2,2,1)
bar(1:n, av_slope_rise,'r')
ylabel('slope');
xlabel('analysissteps');
title('slope on the rising parts');
legend('slope','location','northwest');
grid on
subplot(2,2,2)
bar(1:n,av_slope_fall,'b')
ylabel('slope');
xlabel('analysissteps');
title('slope on the falling parts');
legend('slope','location','northwest');
grid on
subplot(2,2,4)
hold on
plot(av_x_limit_left_rise,av_y_limit_left_rise,'-*')

```

```

plot(av_x_limit_right_rise,av_y_limit_right_rise,'-*)
plot(av_x_limit_left_fall,av_y_limit_left_fall,'-*)
plot(av_x_limit_right_fall,av_y_limit_right_fall,'-*)
ylabel('force [N]');
xlabel('deformation [mm]');
title('average curve')
legend off
grid on
set(gcf,'position',[x0,y0,width,height])
%
figure
hold on
plot(av_x_limit_left_rise,av_y_limit_left_rise,'-*)
plot(av_x_limit_right_rise,av_y_limit_right_rise,'-*)
plot(av_x_limit_left_fall,av_y_limit_left_fall,'-*)
plot(av_x_limit_right_fall,av_y_limit_right_fall,'-*)
ylabel('force [N]');
xlabel('deformation [mm]');
title('average curve')
legend off
grid on
set(gcf,'position',[x0,y0,width,height])
%
figure
plot(result.data(:,4),result.data(:,3))
ylabel('force [N]');
xlabel('deformation [mm]');
title('resultdata')
grid on
set(gcf,'position',[x0,y0,width,height])

```

Matlabskript zum Auswerten der Messungen:

```

clc
clear
close all
%% sort the results %%
%
result = importdata('Marathon1.csv'); %importing the resultdata
collums = length(result.data(1,:)); %get amount of collums in results
rows = length(result.data(:,1)); %get amount of rows in results
data = zeros(rows,collums-1); %create a matrix with same size like the
results
%
data(:,1) = result.data(:,3); %put time at first place
data(:,2) = result.data(:,4); %put cyclenumber at second place
data(:,3) = result.data(:,1); %put force at third place
data(:,3) = data(:,3) * (-1); %set the negative values to positive
start_force = min(data(:,3)); %detect the difference between the min
force and zero
%data(:,3) = data(:,3) - start_force; %set the axis to zero
data(:,3) = data(:,3) * 1000; %set force to Newton
data(:,4) = result.data(:,2); %put deformation at fourth place
data(:,4) = data(:,4) * (-1); %set the negative values to positive
start_deformation = min(data(:,4)); %detect the difference between the
min deformation and zero
data(:,4) = data(:,4) - start_deformation; %set the axis to zero
%% starting conditions %%
%
n = 1; %number of analysissteps
start_later = 1; %skip the first x loops

```

```

end_earlier = 5; %skip the last x loops
show_loop = 5; %amount of loops shown at the end
%
%values for creating an average
av_max_f = 0;
av_max_def = 0;
av_min_def = 0;
av_time(1:3,1) = 0;
summ_slope_rise(1,n) = zeros;
summ_slope_fall(1,n) = zeros;
%times it is default to be
%
area = 1; %1=rearfoot ; 2=forefoot
default_time(1:2,1) = 1.053; %overall time for a step
default_time(1:2,2) = 0.95; %supposed frequency
default_time(1:2,3) = 12055.71; %overall testingtime
default_time(1,4) = 0.023; %default time for the rise for the rearfoot
default_time(1,5) = 0.108; %default time for the fall for the rearfoot
default_time(2,4) = 0.097; %default time for the rise for the forefoot
default_time(2,5) = 0.114; %default time for the fall for the forefoot
%
%force it is default to be
default_force(1,1) = 866.72; %rearfoot [N]
default_force(1,2) = 1612.82 ;%forefoot [N]
%definition of the size and the location for the plots
x0=200;
y0=100;
width=950;
height=650;
%% look for the testingloops %%
%
% the following infomatrix of the loop will be positionned like this:
% 1.=Loopnumber      2.=start_position      3.=endposition
4.=max_value
% 5.=max_value_position 6.=min_value      7.=min_value_position
8.=amount of frames on the rise
% 9.=length analysissteps on the rise      10.=amount of frames
on the fall 11.=length analysissteps on the rise
for ( z = 1:1:max(data(:,2))) %finding existing loopnumbers
    for b = 1:1:length(data(:,2)) %looking for the existing loopnumbers
        while finding matching loopnumbers in the results and the counter which
        is counting to the max loopnumber
            if z == data(b,2)
                look_for_loop_number(z,1) = z;
            end
        end
    end
end
d = 1;
for (c = 1:length(look_for_loop_number(:,1))) %eliminating every zeros in
the searching variable
    if look_for_loop_number(c,1) > 0
        loop_number(d,1) = look_for_loop_number(c,1);
        d = d + 1;
    end
end
end
%
for (f = 1:length(loop_number(:,1))) %creating a vector which has the
information of starting each of the loops
    for (g = 1:length(data(:,2)))
        if data(g,2) == loop_number(f,1)
            loop(f,1) = loop_number(f,1); %first collum = loopnumber
        end
    end
end

```

```

        loop(f,2) = g; %second collum = startingframe of each loop
        if (f == length(loop_number(:,1)))
            break
        end
        f = f + 1;
    end
end
end
%add an extra loop
loop(length(loop(:,1))+1,1) = max(loop(:,1)) + 1; %adding an extra
loop for creating the last testingloopending
loop(length(loop(:,1)),2) = length(data(:,1)) + 1; %adding an extra
loop for creating the last testingloopending
%create the distance between the shown loops
shown_loop_steps = length(loop(:,1)) / show_loop; %determinating the
distance between the loops acquired for showing in the end
shown_loop_steps = int32(shown_loop_steps); %get an rounded number to
call the value
%% get the extrema %%
%
for (i = 1:(length(loop(:,1))-1))
    loop(i,3) = loop(i+1,2) - 1; %third collumn = endingframe of each
loop
    %max deformation
    loop(i,4) = max(data(loop(i,2):loop(i,3),4)); %fourth collum = value
of each loop max
    loop(i,5) = find(data(:,4) == loop(i,4)); %get the position of the
maximum deformation in the overall resultmatrix
    av_max_def = av_max_def + loop(i,4); %get the amount of overall max.
deformation throughout all testingloops
    %min deformation
    loop(i,6) = min(data(loop(i,2):loop(i,3),4)); %fourth collum = value
of each loop max
    loop(i,7) = find(data(:,4) == loop(i,6)); %get the position of the
maximum deformation in the overall resultmatrix
    av_min_def = av_min_def + loop(i,6); %get the amount of overall min.
deformation throughout all testingloops
    %max force
    max_force(i,1) = max(data(loop(i,2):loop(i,3),3)); %create a vector
which shows the max. force for each loop
    max_force(i,2) = loop(i,1); %set the loopnumber next to the max.
force so you can see what happened in which loop
    av_max_f = av_max_f + max_force(i,1); %get the amount of overall max.
force throughout all testingloops
    %time loop
    time_loop(i,1) = data(loop(i,3),1) - data(loop(i,2),1);
    time_loop(i,4) = loop(i,1);
    av_time(1,1) = av_time(1,1) + time_loop(i,1);
%
    if ( i == (length(loop(:,1))-1) )
        %create an average max. deformation
        av_max_def = av_max_def / i; %devide the overall amount throug
the number of loops
        av_max_deformation(1:i,1) = av_max_def; %create an matrix with
the average in the first collumn
        av_max_deformation(1:i,2) = loop(1:end-1,1); %get the loopnumbers
to the matrix in the secound collumn
        %create an average min. deformation
        av_min_def = av_min_def / i; %devide the overall amount throug
the number of loops
    end
end
end

```

```

        av_min_deformation(1:i,1) = av_min_def; %create an matrix with
the average in the first collumn
        av_min_deformation(1:i,2) = loop(1:end-1,1); %get the loopnumbers
to the matrix in the secound collumn
        %create an average max. force
        av_max_f = av_max_f / i; %devide the overall amount through the
number of loops
        av_max_force(1:i,1) = av_max_f; %create an matrix with the
average in the first collumn
        av_max_force(1:i,2) = loop(1:end-1,1); %get the loopnumbers to
the matrix in the secound collumn
        %create an average looptime
        av_time = av_time / i;
        frequency = 1 / av_time(1,1);
    end

end

%% analysis of the slope %%
%
for (i = start_later:(length(loop(:,1))-end_earlier))
    %rise
    loop(i,8) = loop(i,5) - loop(i,7); %eight collumn = framenumber of
max. position - framenumber of min. position => amount of frames on the
rising part
    loop(i,9) = loop(i,8) / n; %ninth collumn = amount of frames in each
step to the max. takes
    loop(i,9) = int32(loop(i,9)); %rounded
    %fall
    loop(i,10) = loop(i+1,7) - loop(i,5); %tenth collumn = framenumber of
min. position next loop - max position of this loop => amount of frames
of the fall
    loop(i,11) = loop(i,10) / n; %eleventh collumn = amount of frames in
each step to the min. takes
    loop(i,11) = int32(loop(i,11)); %rounded
    for ( m = 1:n ) %analysis of the rising part
        %get the positions of the limits
        position_limit_left_rise(i,m) = loop(i,7) + ( (m-1) * loop(i,9));
%get each left limit with the endingpoints and the substracted
analysissteplength
        position_limit_right_rise(i,m) = loop(i,7) + ( m * loop(i,9)); %get
each left limit with the endingpoints and the substracted
analysissteplength
        x_limit_left_rise(i,m) = data(position_limit_left_rise(i,m),4); %set
the value of the limit
        x_limit_right_rise(i,m) = data(position_limit_right_rise(i,m),4);
%set the value of the limit
        y_limit_left_rise(i,m) = data(position_limit_left_rise(i,m),3); %set
the value of the limit
        y_limit_right_rise(i,m) = data(position_limit_right_rise(i,m),3);
%set the value of the limit
        %polyfit
        x_rise = [x_limit_left_rise(i,m), x_limit_right_rise(i,m)]; %create
a x-vector for the following polyfit
        y_rise = [y_limit_left_rise(i,m), y_limit_right_rise(i,m)]; %create
a y-vector for the following polyfit
        p_r = polyfit(x_rise,y_rise,1); %fit a straight line between the
left and right limit
        p_rise(i,m) = p_r(1,1); %a helpvalue for saving the exact values for
each step of the loops
        p_rise(i,m+n) = p_r(1,2); %a helpvalue for saving the exact values
for each step of the loops
    end
end

```

```

    p_help = [p_rise(i,m), p_rise(i,m+n)]; %create a vector for the
plotting
    yl_rise = polyval(p_help,x_rise); %get the y-values for the plotting
    %slope of each splitted part of the curve
    slope_rise(i,m) = p_r(1,1); %the first value of the vectors are the
slope set to a matrix
    slope_rise(i,m+1) = loop(i,1); %get the loopnumbers to the matrix in
the second collumn
    %summ slope rise
    summ_slope_rise(1,1) = summ_slope_rise(1,1) + slope_rise(i,m); %add
each loop the value of the slope for an overall summ
    %averagee slope rise
    if ( i == (length(loop(:,1))-end_earlier) )
        av_slope_rise = summ_slope_rise(1,1) / i;
    end
    %plot
    hold on
    plot(x_rise,yl_rise,'-*');
end
for ( m = 1:n ) %analysis of the falling part
    %get the positions of the limits
    position_limit_left_fall(i,m) = loop(i+1,7) - ( (m-1) *
loop(i,11)); %get each left limit with the endingpoints and the
substracted analysissteplength
    position_limit_right_fall(i,m) = loop(i+1,7) - ( m * loop(i,11));
%get each left limit with the endingpoints and the substracted
analysissteplength
    x_limit_left_fall(i,m) = data(position_limit_left_fall(i,m),4);
%set the value of the limit
    x_limit_right_fall(i,m) = data(position_limit_right_fall(i,m),4);
%set the value of the limit
    y_limit_left_fall(i,m) = data(position_limit_left_fall(i,m),3);
%set the value of the limit
    y_limit_right_fall(i,m) = data(position_limit_right_fall(i,m),3);
%set the value of the limit
    %polyfit
    x_fall = [x_limit_left_fall(i,m), x_limit_right_fall(i,m)]; %create
a x-vector for the following polyfit
    y_fall = [y_limit_left_fall(i,m), y_limit_right_fall(i,m)]; %create
a x-vector for the following polyfit
    p_f = polyfit(x_fall,y_fall,1); %fit a straight line between the
left and right limit
    p_fall(i,m) = p_f(1,1); %a helpvalue for saving the exact values
for each step of the loops
    p_fall(i,m+n) = p_f(1,2); %a helpvalue for saving the exact values
for each step of the loops
    p_help_fall = [p_fall(i,m), p_fall(i,m+n)]; %create a vector for
the plotting
    yl_fall = polyval(p_help_fall,x_fall); %get the y-values for the
plotting
    %slope of each splitted part of the curve
    slope_fall(i,m) = p_f(1,1); %the first value of the vectors are the
slope
    slope_fall(i,m+1) = loop(i,1); %get the loopnumbers to the matrix
in the second collumn
    %summ slope fall
    summ_slope_fall(1,1) = summ_slope_fall(1,1) + slope_fall(i,m); %add
each loop the value of the slope for an overall summ
    %averagee slope rise
    if ( i == (length(loop(:,1))-end_earlier) )
        av_slope_fall = summ_slope_fall(1,1) / i;
    end
end

```

```

end

%plot
hold on
plot(x_fall,y1_fall,'-*');
xlabel('deformation [mm]')
ylabel('force [N]')
title('the slope for each hysteresis')
legend('slope','location','northeast')
grid on
set(gcf,'position',[x0,y0,width,height])
end
end
%% analysis of the energy %%
%
for ( c = start_later:i)
    %rise
    rising_part_each_loop(c,1) = loop(c,7); %start rising part at each
start of the testingloops
    rising_part_each_loop(c,2) = loop(c,5); %end rising part at each
individual max of the testingloops
    rising_part_each_loop(c,3) = loop(c,1); %get the loopnumbers to the
matrix in the thrid collumn
    integral_x_rise =
data(rising_part_each_loop(c,1):rising_part_each_loop(c,2),4); %get a
matrix with all x-values of the rise
    integral_y_rise =
data(rising_part_each_loop(c,1):rising_part_each_loop(c,2),3); %get a
matrix with all y-values of the rise
    energy_rise(c,1) = trapz(integral_x_rise, integral_y_rise); %get
the integral between the x and y axis of the rise
    energy_rise(c,2) = loop(c,1); %get the loopnumbers to the matrix in
the second collumn
    %fall
    falling_part_each_loop(c,1) = loop(c,5); %start falling part at
each max of the testingloop
    falling_part_each_loop(c,2) = loop(c+1,7); %end rising part =
startingposition next loop - 1.
    falling_part_each_loop(c,3) = loop(c,1); %get the loopnumbers to
the matrix in the third collumn
    integral_x_fall =
data(falling_part_each_loop(c,1):falling_part_each_loop(c,2),4); %get a
matrix with all x-values of the rise
    integral_y_fall =
data(falling_part_each_loop(c,1):falling_part_each_loop(c,2),3); %get a
matrix with all y-values of the rise
    energy_fall(c,1) = trapz(integral_x_fall, integral_y_fall); %get
the integral between the x and y axis of the fall
    energy_fall(c,2) = loop(c,1); %get the loopnumbers to the matrix in
the second collumn
    %difference
    energy_diff(c,1) = energy_rise(c,1) + energy_fall(c,1); %get the
difference between the two integrals, the falling part is negative
because of that there is a "+"
    energy_diff(c,2) = loop(c,1); %get the loopnumbers to the matrix in
the second collumn
    %energy in percent
    energy_percent(c,1) = (( energy_fall(c,1) * -1 ) / energy_rise(c,1)
) * 100;
    energy_percent(c,2) = loop(c,1);

```


end

%% show the difference between the actual and the other datastructure %%

%

example =

```
[ 30.275800000000000,7,114,837730407700,4,70626351226800,30.2798000000000,7,116,564292907700,4,71234226226800,30.2838000000000,7,118,43586657700,4,72696018219000,30.2878000000000,7,120,43586657700,4,7443550172100,30.2918000000000,7,122,568199157700,4,76443576182700,30.2938000000000,7,123,677574157700,4,7754831340900,30.2958000000000,7,124,826011657700,4,7870798119000,30.2978000000000,7,126,00569157700,4,79928684234600,30.2998000000000,7,127,20864157700,4,8120431900200,30.3018000000000,7,128,443206787100,4,825348940600,30.3038000000000,7,129,709831977100,4,839876422100,30.3058000000000,7,131,005706787100,4,8523203952700,30.3078000000000,7,132,318298787100,4,865800000000,7,133,616956787100,4,88035357100,30.3098000000000,7,135,616956787100,4,8959125200,30.3118000000000,7,135,025914287100,4,8980417251900,30.3138000000000,7,137,42581787100,4,9134225853400,30.3158000000000,7,137,833831787100,4,9282551422100,30.3178000000000,7,139,2171331787100,4,94455051422100,30.3198000000000,7,140,176644287100,4,96017551422100,30.3218000000000,7,142,169769287100,4,9757394906500,30.3238000000000,7,143,638519287100,4,991244975200,30.3258000000000,7,145,122984287100,5,0066232812700,30.3278000000000,7,146,622894287100,5,02182102203400,30.3298000000000,7,148,122984287100,5,0367746353100,30.3318000000000,7,149,630787100,5,05145142700,30.3338000000000,7,151,138519287100,5,0667151031500,30.3358000000000,7,152,65144287100,5,08035687700,30.3378000000000,7,154,169769287100,5,094536956787100,23,237585223400,30.3418000000000,7,172,0528187100,5,2442321340900,30.3438000000000,7,173,49784287100,5,2552463331500,30.3458000000000,7,174,6644287100,5,28592467500,30.3498000000000,7,166,20119287100,5,1974862072100,30.3538000000000,7,167,669769287100,5,2093820571900,30.3578000000000,7,169,138519287100,5,2211614234600,30.3598000000000,7,170,59456787100,23,237585223400,30.3618000000000,7,172,0528187100,5,2442321340900,30.3638000000000,7,173,49784287100,5,2552463331500,30.3658000000000,7,174,6644287100,5,28592467500,30.3698000000000,7,176,365081787100,5,2778650250200,30.3698000000000,7,167,669769287100,5,2093820571900,30.3718000000000,7,179,19287100,5,2211614234600,30.3738000000000,7,180,670269287100,5,31051731109600,30.3758000000000,7,182,0507687100,5,3212053375200,30.3778000000000,7,183,4044287100,5,3321237564900,30.3798000000000,7,184,80281787100,5,342926799600,30.3818000000000,7,186,139287100,5,3537912387700,30.3838000000000,7,187,58381787100,5,3647152894600,30.3858000000000,7,188,96644287100,5,37563812700,30.3878000000000,7,190,357829287100,5,38687229156500,30.3898000000000,7,191,74894287100,5,3980417251900,30.3918000000000,7,193,138519287100,5,4098532906500,30.3938000000000,7,194,536956787100,5,4207468287100,30.3958000000000,7,195,138519287100,5,43269394287100,5,444287100,5,455578750200,30.4018000000000,7,200,177581787100,5,467386987100,5,4038000000000,7,201,607269287100,5,4794628359600,30.4058000000000,7,203,0528187100,5,491608196900,30.4098000000000,7,204,49784287100,5,5039372125200,30.4098000000000,7,205,96644287100,5,5163889312700,30.4118000000000,7,207,443206787100,5,52902317047700,30.4138000000000,7,208,93594287100,5,5418405328400,30.4158000000000,7,210,43586657700,5,55484141656500,4,4781000000000,7,211,950119287100,5,568024635100,30.4198000000000,7,213,49081787100,5,5813923453400,30.4218000000000,7,215,036956787100,5,59494119322100,30.4238000000000,7,216,59456787100,5,6087404937700,30.4258000000000,7,218,177581787100,5,628599284600,30.4278000000000,7,219,71331787100,5,6366281509400,30.4298000000000,7,221,372894287100,5,6508493234600,30.4318000000000,7,222,98269287100,5,6653146743700,30.4338000000000,7,224,615081787100,5,6799020767100,30.4358000000000,7,226,263519287100,5,6946725843400,30.4378000000000,7,227,927581787100,5,7093651265900,30.4398000000000,7,229,607269287100,5,72470181987100,30.4418000000000,7,231,302581787100,30.4438000000000,7,233,005706787100,5,4438000000000,7,235,005706787100,5,4438000000000,7,236,443206787100,5,4438000000000,7,237,443206787100,5,4438000000000,7,238,138519287100,5,4438000000000,7,239,443206787100,5,4438000000000,7,240,443206787100,5,4438000000000,7,241,443206787100,5,4438000000000,7,242,443206787100,5,4438000000000,7,243,443206787100,5,4438000000000,7,244,443206787100,5,4438000000000,7,245,443206787100,5,4438000000000,7,246,443206787100,5,4438000000000,7,247,443206787100,5,4438000000000,7,248,443206787100,5,4438000000000,7,249,443206787100,5,4438000000000,7,250,443206787100,5,4438000000000,7,251,443206787100,5,4438000000000,7,252,443206787100,5,4438000000000,7,253,443206787100,5,4438000000000,7,254,443206787100,5,4438000000000,7,255,443206787100,5,4438000000000,7,256,443206787100,5,4438000000000,7,257,443206787100,5,4438000000000,7,258,443206787100,5,4438000000000,7,259,443206787100,5,4438000000000,7,260,443206787100,5,4438000000000,7,261,443206787100,5,4438000000000,7,262,443206787100,5,4438000000000,7,263,443206787100,5,4438000000000,7,264,443206787100,5,4438000000000,7,265,443206787100,5,4438000000000,7,266,443206787100,5,4438000000000,7,267,443206787100,5,4438000000000,7,268,443206787100,5,4438000000000,7,269,443206787100,5,4438000000000,7,270,443206787100,5,4438000000000,7,271,443206787100,5,4438000000000,7,272,443206787100,5,4438000000000,7,273,443206787100,5,4438000000000,7,274,443206787100,5,4438000000000,7,275,443206787100,5,4438000000000,7,276,443206787100,5,4438000000000,7,277,443206787100,5,4438000000000,7,278,443206787100,5,4438000000000,7,279,443206787100,5,4438000000000,7,280,443206787100,5,4438000000000,7,281,443206787100,5,4438000000000,7,282,443206787100,5,4438000000000,7,283,443206787100,5,4438000000000,7,284,443206787100,5,4438000000000,7,285,443206787100,5,4438000000000,7,286,443206787100,5,4438000000000,7,287,443206787100,5,4438000000000,7,288,443206787100,5,4438000000000,7,289,443206787100,5,4438000000000,7,290,443206787100,5,4438000000000,7,291,443206787100,5,4438000000000,7,292,443206787100,5,4438000000000,7,293,443206787100,5,4438000000000,7,294,443206787100,5,4438000000000,7,295,443206787100,5,4438000000000,7,296,443206787100,5,4438000000000,7,297,443206787100,5,4438000000000,7,298,443206787100,5,4438000000000,7,299,443206787100,5,4438000000000,7,300,443206787100,5,4438000000000,7,301,443206787100,5,4438000000000,7,302,443206787100,5,4438000000000,7,303,443206787100,5,4438000000000,7,304,443206787100,5,4438000000000,7,305,443206787100,5,4438000000000,7,306,443206787100,5,4438000000000,7,307,443206787100,5,4438000000000,7,308,443206787100,5,4438000000000,7,309,443206787100,5,4438000000000,7,310,443206787100,5,4438000000000,7,311,443206787100,5,4438000000000,7,312,443206787100,5,4438000000000,7,313,443206787100,5,4438000000000,7,314,443206787100,5,4438000000000,7,315,443206787100,5,4438000000000,7,316,443206787100,5,4438000000000,7,317,443206787100,5,4438000000000,7,318,443206787100,5,4438000000000,7,319,443206787100,5,4438000000000,7,320,443206787100,5,4438000000000,7,321,443206787100,5,4438000000000,7,322,443206787100,5,4438000000000,7,323,443206787100,5,4438000000000,7,324,443206787100,5,4438000000000,7,325,443206787100,5,4438000000000,7,326,443206787100,5,4438000000000,7,327,443206787100,5,4438000000000,7,328,443206787100,5,4438000000000,7,329,443206787100,5,4438000000000,7,330,443206787100,5,4438000000000,7,331,443206787100,5,4438000000000,7,332,443206787100,5,4438000000000,7,333,443206787100,5,4438000000000,7,334,443206787100,5,4438000000000,7,335,443206787100,5,4438000000000,7,336,443206787100,5,4438000000000,7,337,443206787100,5,4438000000000,7,338,443206787100,5,4438000000000,7,339,443206787100,5,4438000000000,7,340,443206787100,5,4438000000000,7,341,443206787100,5,4438000000000,7,342,443206787100,5,4438000000000,7,343,443206787100,5,4438000000000,7,344,443206787100,5,4438000000000,7,345,443206787100,5,4438000000000,7,346,443206787100,5,4438000000000,7,347,443206787100,5,4438000000000,7,348,443206787100,5,4438000000000,7,349,443206787100,5,4438000000000,7,350,443206787100,5,4438000000000,7,351,443206787100,5,4438000000000,7,352,443206787100,5,4438000000000,7,353,443206787100,5,4438000000000,7,354,443206787100,5,4438000000000,7,355,443206787100,5,4438000000000,7,356,443206787100,5,4438000000000,7,357,443206787100,5,4438000000000,7,358,443206787100,5,4438000000000,7,359,443206787100,5,4438000000000,7,360,443206787100,5,4438000000000,7,361,443206787100,5,4438000000000,7,362,443206787100,5,4438000000000,7,363,443206787100,5,4438000000000,7,364,443206787100,5,4438000000000,7,365,443206787100,5,4438000000000,7,366,443206787100,5,4438000000000,7,367,443206787100,5,4438000000000,7,368,443206787100,5,4438000000000,7,369,443206787100,5,4438000000000,7,370,443206787100,5,4438000000000,7,371,443206787100,5,4438000000000,7,372,443206787100,5,4438000000000,7,373,443206787100,5,4438000000000,7,374,443206787100,5,4438000000000,7,375,443206787100,5,4438000000000,7,376,443206787100,5,4438000000000,7,377,443206787100,5,4438000000000,7,378,443206787100,5,4438000000000,7,379,443206787100,5,4438000000000,7,380,443206787100,5,4438000000000,7,381,443206787100,5,4438000000000,7,382,443206787100,5,4438000000000,7,383,443206787100,5,4438000000000,7,384,443206787100,5,4438000000000,7,385,443206787100,5,4438000000000,7,386,443206787100,5,4438000000000,7,387,443206787100,5,4438000000000,7,388,443206787100,5,4438000000000,7,389,443206787100,5,4438000000000,7,390,443206787100,5,4438000000000,7,391,443206787100,5,4438000000000,7,392,443206787100,5,4438000000000,7,393,443206787100,5,4438000000000,7,394,443206787100,5,4438000000000,7,395,443206787100,5,4438000000000,7,396,443206787100,5,4438000000000,7,397,443206787100,5,4438000000000,7,398,443206787100,5,4438000000000,7,399,443206787100,5,4438000000000,7,400,443206787100,5,4438000000000,7,401,443206787100,5,4438000000000,7,402,443206787100,5,4438000000000,7,403,443206787100,5,4438000000000,7,404,443206787100,5,4438000000000,7,405,443206787100,5,4438000000000,7,406,443206787100,5,4438000000000,7,407,443206787100,5,4438000000000,7,408,443206787100,5,4438000000000,7,409,443206787100,5,4438000000000,7,410,443206787100,5,4438000000000,7,411,443206787100,5,4438000000000,7,412,443206787100,5,4438000000000,7,413,443206787100,5,4438000000000,7,414,443206787100,5,4438000000000,7,415,443206787100,5,4438000000000,7,416,443206787100,5,4438000000000,7,417,443206787100,5,4438000000000,7,418,443206787100,5,4438000000000,7,419,443206787100,5,4438000000000,7,420,443206787100,5,4438000000000,7,421,443206787100,5,4438000000000,7,422,443206787100,5,4438000000000,7,423,443206787100,5,4438000000000,7,424,443206787100,5,4438000000000,7,425,443206787100,5,4438000000000,7,426,443206787100,5,4438000000000,7,427,443206787100,5,4438000000000,7,428,443206787100,5,4438000000000,7,429,443206787100,5,4438000000000,7,430,443206787100,5,4438000000000,7,431,443206787100,5,4438000000000,7,432,443206787100,5,4438000000000,7,433,443206787100,5,4438000000000,7,434,443206787100,5,4438000000000,7,435,443206787100,5,4438000000000,7,436,443206787100,5,4438000000000,7,437,443206787100,5,4438000000000,7,438,443206787100,5,4438000000000,7,439,443206787100,5,4438000000000,7,440,443206787100,5,4438000000000,7,441,443206787100,5,4438000000000,7,442,443206787100,5,4438000000000,7,443,443206787100,5,4438000000000,7,444,443206787100,5,4438000000000,7,445,443206787100,5,4438000000000,7,446,443206787100,5,4438000000000,7,447,443206787100,5,4438000000000,7,448,443206787100,5,4438000000000,7,449,443206787100,5,4438000000000,7,450,443206787100,5,4438000000000,7,451,443206787100,5,4438000000000,7,452,443206787100,5,4438000000000,7,453,443206787100,5,4438000000000,7,454,443206787100,5,4438000000000,7,455,443206787100,5,4438000000000,7,456,443206787100,5,4438000000000,7,457,443206787100,5,4438000000000,7,458,443206787100,5,4438000000000,7,459,443206787100,5,4438000000000,7,460,443206787100,5,4438000000000,7,461,443206787100,5,4438000000000,7,462,443206787100,5,4438000000000,7,463,443206787100,5,4438000000000,7,464,443206787100,5,4438000000000,7,465,443206787100,5,4438000000000,7,466,443206787100,5,4438000000000,7,467,443206787100,5,4438000000000,7,468,443206787100,5,4438000000000,7,469,443206787100,5,4438000000000,7,470,443206787100,5,4438000000000,7,471,443206787100,5,4438000000000,
```



```

    time_loop(j,3) = data(loop(j+1,7),1) - data(time_fall(j,1),1); %get
the time needed for the fall of the max. force to a force lower than 100N
    av_time(3,1) = av_time(3,1) + time_loop(j,3); %get an overall
amount of time needed for all loops
    %
    if ( j == i )
        av_time(3,1) = av_time(3,1) / i; %divide the overall amount of
time through the number of testingloops for an average
        av_time(2,1) = av_time(2,1) / i; %divide the overall amount of
time through the number of testingloops for an average
    %
        if ( area == 1 ) %1 = rearfoot
            actual_time(1:2,1) = av_time(1,1); %first collumn = average
time for one step
            actual_time(1:2,2) = frequency; %second collumn = average
frequency
            actual_time(1:2,3) = max(data(:,1)); %third collumn = overall
testingtime
            actual_time(1,4) = av_time(2,1); %fourth collumn = average
time nedded for the rise of the max. force
            actual_time(1,5) = av_time(3,1); %fifth collumn = average
time nedded for the fall of the force to sub 100N
            elseif ( area == 2 ) %2 = forefoot
                actual_time(1:2,1) = av_time(1,1); %first collumn = average
time for one step
                actual_time(1:2,2) = frequency; %second collumn = average
frequency
                actual_time(1:2,3) = max(data(:,1)); %third collumn = overall
testingtime
                actual_time(2,4) = av_time(2,1); %fourth collumn = average
time nedded for the rise of the max. force
                actual_time(2,5) = av_time(3,1); %fifth collumn = average
time nedded for the fall of the force to sub 100N
            end
        %
        time_difference = default_time - actual_time; %get the
difference between supposed and actual needed time
    end
end
%% plots %%
%% time and frequency %%
%
    figure ('Name','analysis of the important timespans and the
frequency','NumberTitle','off');
    subplot(2,1,1)
    bar(1,default_time(1,1))
    hold on
    bar(1.5,actual_time(1,1))
    bar(2.5,default_time(1,2))
    bar(3,actual_time(1,2))
    bar(4.5,default_time(1,3)/60/60)
    bar(5,actual_time(1,3)/60/60)
    title('timespans')
    legend('default time for a step [s]','actual time for a step needed
[s]','default frequency [1/s]','actual frequency [1/s]','default overall
testingtime [h]','actual overall testingtime [h]','location','northwest')
    grid on
    if area == 1
        subplot(2,1,2)
        bar(1,default_time(1,4))
        hold on

```

```

    bar(1.5,actual_time(1,4))
    bar(2.5,default_time(1,5))
    bar(3,actual_time(1,5))
    title('comparison of the time of the fall & the rise')
    legend('default time to rise [s]','actual time needed for the rise
[s]','default time to fall [s]','actual time needed for the fall
[s]','location','northwest')
    grid on
elseif area == 2
    subplot(2,1,2)
    bar(1,default_time(2,4))
    hold on
    bar(1.5,actual_time(2,4))
    bar(2.5,default_time(2,5))
    bar(3,actual_time(2,5))
    title('comparison of the time of the fall & the rise')
    legend('default time to rise','actual time needed for the
rise','default time to fall','actual time needed for the
fall','location','northwest')
    grid on
end
set(gcf,'position',[x0,y0,width,height])
%% energy normal %%
%
figure('Name','Overview of the energys','NumberTitle','off');
subplot(2,2,1)
plot(energy_rise(:,2),energy_rise(:,1)/1000,'r')
xlabel('loopnumber');
ylabel('energy [J]')
title('energy absorbed')
legend('energy','location','northeast')
grid on
subplot(2,2,2)
plot(energy_fall(:,2),energy_fall(:,1)/1000,'b')
xlabel('loopnumber');
ylabel('energy [J]')
title('energy dispensed')
legend('energy','location','northeast')
grid on
subplot(2,2,3)
plot(energy_diff(:,2),energy_diff(:,1)/1000,'k')
xlabel('loopnumber')
ylabel('energy [J]')
title('energydifference absorbed and dispensed')
legend('energydifference','location','northeast')
grid on
subplot(2,2,4)
plot(energy_percent(:,2),energy_percent(:,1))
xlabel('loopnumber')
ylabel('energyreturnr [%]')
title('returned energy in percent')
legend('energyreturn','location','southeast')
grid on
set(gcf,'position',[x0,y0,width,height])
%% energy logarhitmic %%
%
figure('Name','Overview of the energys
logarhitmic','NumberTitle','off');
subplot(2,2,1)
loglog(energy_rise(:,2),energy_rise(:,1)/1000,'r')
xlabel('loopnumber');

```

```

ylabel('energy [J]')
title('energy energy absorbed')
legend('energy','location','northeast')
grid on
subplot(2,2,2)
loglog(energy_fall(:,2),energy_fall(:,1)/1000,'b')
xlabel('loopnumber');
ylabel('energy [J]')
title('energy dispensed')
legend('energy','location','northeast')
grid on
subplot(2,2,3)
loglog(energy_diff(:,2),energy_diff(:,1)/1000,'k')
xlabel('loopnumber')
ylabel('energy [J]')
title('energydifference absorbed and dispensed')
legend('energydifference','location','northeast')
grid on
set(gcf,'position',[x0,y0,width,height])
%% force and deformation development %%
%
figure('Name','development of the deformation and the
force','NumberTitle','off');
subplot(2,2,1)
plot(loop(0+start_later:end-end_earlier,1),loop(0+start_later:end-
end_earlier,6))
hold on
plot(av_min_deformation(0+start_later:end-
end_earlier,2),av_min_deformation(0+start_later:end-end_earlier,1))
xlabel('loopnumber')
ylabel('deformation [mm]')
title('development of the min. deformation')
legend('min. deformation each loop','average min.
deformation','location','southeast')
grid on
subplot(2,2,2)
plot(loop(0+start_later:end-end_earlier,1),loop(0+start_later:end-
end_earlier,4))
hold on
plot(av_max_deformation(0+start_later:end-
end_earlier,2),av_max_deformation(0+start_later:end-end_earlier,1))
xlabel('loopnumber')
ylabel('deformation [mm]')
title('development of the max. deformation')
legend('max. deformation each loop','average max.
deformation','location','southeast')
grid on
subplot(2,2,3)
plot(max_force(0+start_later:end-
end_earlier,2),max_force(0+start_later:end-end_earlier,1))
hold on
plot(av_max_force(0+start_later:end-
end_earlier,2),av_max_force(0+start_later:end-end_earlier,1))
xlabel('loopnumber')
ylabel('max force [N]')
title('development of the max. force')
legend('max. force each loop','average max.
force','location','southeast')
grid on
subplot(2,2,4)
if area == 1

```

```

    bar(1,default_force(1,1))
    hold on
elseif area == 2
    bar(1,default_force(2,1))
    hold on
end
bar(1.5,av_max_f)
ylabel('force [N]')
title('comparison of the default force and the actual force')
legend('default force','actual average force
peak','location','southwest')
set(gcf,'position',[x0,y0,width,height])
%% overlall slope %%
%
    figure ('Name','analysis of the overall slope of the
hysteresis','NumberTitle','off');
    subplot(2,2,1)
    plot(slope_rise(:,2),slope_rise(:,1),'r')
    ylabel('slope [N/mm]');
    xlabel('loopnumber');
    title('slope on the rising parts');
    legend('slope','location','southeast');
    grid on
    subplot(2,2,2)
    plot(slope_fall(:,2),slope_fall(:,1),'b')
    ylabel('slope [N/mm]');
    xlabel('loopnumber');
    title('slope on the falling parts');
    legend('slope','location','southeast');
    grid on
    subplot(2,2,3)
    loglog(slope_rise(:,2),slope_rise(:,1),'r')
    ylabel('slope [N/mm]');
    xlabel('loopnumber');
    title('slope on the rising parts');
    legend('slope','location','southeast');
    grid on
    subplot(2,2,4)
    loglog(slope_fall(:,2),slope_fall(:,1),'b')
    ylabel('slope [N/mm]');
    xlabel('loopnumber');
    title('slope on the falling parts');
    legend('slope','location','southeast');
    grid on
    set(gcf,'position',[x0,y0,width,height])
%% development of hysteresis %%
%
    figure ('Name','visualisation of the
hysteresischange','NumberTitle','off');
    for ( w = 1:show_loop)
        hold on
        plot(data(loop(1+((w-1) * shown_loop_steps),2):loop(1+((w-1) *
shown_loop_steps),3),4),data(loop(1+((w-1) *
shown_loop_steps),2):loop(1+((w-1) * shown_loop_steps),3),3))
    end
    xlabel('deformation [mm]')
    ylabel('force [N]')
    title('development of the testingloops')
    legend('change of the loops visualised','location','northeast')
    grid on
    set(gcf,'position',[x0,y0,width,height])

```

```

%% visualisation of the datastructure %%
%
    figure ('Name','datastructure','NumberTitle','off');
    subplot(2,2,1)
    plot(example(:,1),example(:,4))
    xlabel('testingtime [s]')
    ylabel('deformation [mm]')
    title('testdata: deformation')
    legend('deformation in one loop','location','best')
    grid on
    subplot(2,2,2)
    plot(example(:,1),example(:,3))
    xlabel('testingtime [s]')
    ylabel('force [N]')
    title('testdata: force')
    legend('force in one loop','location','best')
    grid on
    subplot(2,2,3)
    plot(data(loop(10,7):loop(11,7),1),data(loop(10,7):loop(11,7),4))
    xlabel('testingtime [s]')
    ylabel('deformation [mm]')
    title('actual data: deformation')
    legend('deformation in one loop','location','best')
    grid on
    subplot(2,2,4)
    plot(data(loop(10,7):loop(11,7),1),data(loop(10,7):loop(11,7),3))
    xlabel('testingtime [s]')
    ylabel('force [N]')
    title('actual data: force')
    legend('force in one loop','location','best')
    grid on
    set(gcf,'position',[x0,y0,width,height])
%% difference between testinghysteresis and actual hysteresis %%
%
    figure ('Name','Hysteresis comparison','NumberTitle','off');
    plot(example(:,4)*65,example(:,3))
    hold on
    plot(example(1,4)*65,example(1,3),'o')
    plot(data(loop(2,2):loop(2,3),4)*200,data(loop(2,2):loop(2,3),3))
    plot(data(loop(2,2),4)*200,data(loop(2,2),3),'o')
    xlabel('deformation [mm]')
    ylabel('force [N]')
    title('hysteresis of the testdata and the actual data')
    legend('hysteresis testdata','startingpoint testdata','hysteresis
actual data','startingpoint actual data','location','northwest')
    grid on
    set(gcf,'position',[x0,y0,width,height])

```

Matlabskript zum Vergleichen beider Messdurchgänge:

```

clc
clear
close all
%% sort the results 1 %%
%
result_1 = importdata('Marathon1.csv'); %importing the resultdata
collums_1 = length(result_1.data(1,:)); %get amount of collums in results
rows_1 = length(result_1.data(:,1)); %get amount of rows in results

```



```

data_1 = zeros(rows_1,collums_1-1); %create a matrix with same size like
the results
%
data_1(:,1) = result_1.data(:,3); %put time at first place
data_1(:,2) = result_1.data(:,4); %put cyclenumber at second place
data_1(:,3) = result_1.data(:,1); %put force at third place
data_1(:,3) = data_1(:,3) * (-1); %set the negative values to positive
start_force_1 = min(data_1(:,3)); %detect the difference between the min
force and zero
%data(:,3) = data(:,3) - start_force_1; %set the axis to zero
data_1(:,3) = data_1(:,3) * 1000; %set force to Newton
data_1(:,4) = result_1.data(:,2); %put deformation at fourth place
data_1(:,4) = data_1(:,4) * (-1); %set the negative values to positive
start_deformation_1 = min(data_1(:,4)); %detect the difference between
the min deformation and zero
data_1(:,4) = data_1(:,4) - start_deformation_1; %set the axis to zero
%% starting conditions
%
n_1 = 1; %number of analysissteps
start_later_1 = 1; %skip the first x loops
end_earlier_1 = 5; %skip the last x loops
show_loop_1 = 5; %amount of loops shown at the ende
%
%values for creating an average
av_max_f_1 = 0;
av_max_def_1 = 0;
av_min_def_1 = 0;
av_time_1(1:3,1) = 0;
summ_slope_rise_1(1,n_1) = zeros;
summ_slope_fall_1(1,n_1) = zeros;
%times it is default to be
%
area_1 = 1; %1=rearfoor ; 2=forefoot
default_time_1(1:2,1) = 1.053; %overall time for a step
default_time_1(1:2,2) = 0.95; %supposed frequency
default_time_1(1:2,3) = 12055.71; %overall testingtime
default_time_1(1,4) = 0.023; %default time for the rise for the rearfoot
default_time_1(1,5) = 0.108; %default time for the fall for the rearfoot
default_time_1(2,4) = 0.097; %default time for the rise for the forefoot
default_time_1(2,5) = 0.114; %default time for the fall for the forefoot
%
%force it is default to be
default_force_1(1,1) = 866.72; %rearfoot [N]
default_force_1(1,2) = 1612.82 ;%forefoot [N]
%definition of the size and the location for the plots
x0=200;
y0=100;
width=950;
height=650;
%% look for the testingloops
%
% the following infomatrix of the loop will be positioned like this:
% 1.=Loopnumber 2.=start_position 3.=endposition
4.=max_value
% 5.=max_value_position 6.=min_value 7.=min_value_position
8.=amount of frames on the rise
% 9.=length analysissteps on the rise 10.=amount of frames
on the fall 11.=length analysissteps on the rise
for ( z_1 = 1:1:max(data_1(:,2))) %finding existing loopnumbers

```



```

    for b_1 = 1:length(data_1(:,2)) %looking for the existing
loopnumbers while finding matching loopnumbers in the results and the
counter which is counting to the max loopnumber
        if z_1 == data_1(b_1,2)
            look_for_loop_number_1(z_1,1) = z_1;
        end
    end
end
d_1 = 1;
for (c_1 = 1:length(look_for_loop_number_1(:,1))) %eliminating every
zeros in the searching variable
    if look_for_loop_number_1(c_1,1) > 0
        loop_number_1(d_1,1) = look_for_loop_number_1(c_1,1);
        d_1 = d_1 + 1;
    end
end
end
%
for (f_1 = 1:length(loop_number_1(:,1))) %creating a vector which has the
information of starting each of the loops
    for (g_1 = 1:length(data_1(:,2)))
        if data_1(g_1,2) == loop_number_1(f_1,1)
            loop_1(f_1,1) = loop_number_1(f_1,1); %first collum = loopnumber
            loop_1(f_1,2) = g_1; %second collum = startingframe of each loop
            if (f_1 == length(loop_number_1(:,1)))
                break
            end
        end
        f_1 = f_1 + 1;
    end
end
end
%add an extra loop
loop_1(length(loop_1(:,1))+1,1) = max(loop_1(:,1)) + 1; %adding an
extra loop for creating the last testingloopending
loop_1(length(loop_1(:,1)),2) = length(data_1(:,1)) + 1; %adding an
extra loop for creating the last testingloopending
%create the distance between the shown loops
shown_loop_steps_1 = length(loop_1(:,1)) / show_loop_1;
%determinating the distance between the loops acquired for showing in the
end
shown_loop_steps_1 = int32(shown_loop_steps_1); %get an rounded
number to call the value
%% get the extrema %%
%
for (i_1 = 1:(length(loop_1(:,1))-1))
    loop_1(i_1,3) = loop_1(i_1+1,2) - 1; %third collumn = endingframe of
each loop
    %max deformation
    loop_1(i_1,4) = max(data_1(loop_1(i_1,2):loop_1(i_1,3),4)); %fourth
collum = value of each loop max
    loop_1(i_1,5) = find(data_1(:,4) == loop_1(i_1,4)); %get the position
of the maximum deformation in the overall resultmatrix
    av_max_def_1 = av_max_def_1 + loop_1(i_1,4); %get the amount of
overall max. deformation throughout all testingloops
    %min deformation
    loop_1(i_1,6) = min(data_1(loop_1(i_1,2):loop_1(i_1,3),4)); %fourth
collum = value of each loop max
    loop_1(i_1,7) = find(data_1(:,4) == loop_1(i_1,6)); %get the position
of the maximum deformation in the overall resultmatrix
    av_min_def_1 = av_min_def_1 + loop_1(i_1,6); %get the amount of
overall min. deformation throughout all testingloops
    %max force

```

```

    max_force_1(i_1,1) = max(data_1(loop_1(i_1,2):loop_1(i_1,3),3));
%create a vector which shows the max. force for each loop
    max_force_1(i_1,2) = loop_1(i_1,1); %set the loopnumber next to the
max. force so you can see what happened in which loop
    av_max_f_1 = av_max_f_1 + max_force_1(i_1,1); %get the amount of
overall max. force throughout all testingloops
    %time loop
    time_loop_1(i_1,1) = data_1(loop_1(i_1,3),1) -
data_1(loop_1(i_1,2),1);
    time_loop_1(i_1,4) = loop_1(i_1,1);
    av_time_1(1,1) = av_time_1(1,1) + time_loop_1(i_1,1);
%
    if ( i_1 == (length(loop_1(:,1))-1) )
        %create an average max. deformation
        av_max_def_1 = av_max_def_1 / i_1; %devide the overall amount
through the number of loops
        av_max_deformation_1(1:i_1,1) = av_max_def_1; %create an matrix
with the average in the first collumn
        av_max_deformation_1(1:i_1,2) = loop_1(1:end-1,1); %get the
loopnumbers to the matrix in the secound collumn
        %create an average min. deformation
        av_min_def_1 = av_min_def_1 / i_1; %devide the overall amount
through the number of loops
        av_min_deformation_1(1:i_1,1) = av_min_def_1; %create an matrix
with the average in the first collumn
        av_min_deformation_1(1:i_1,2) = loop_1(1:end-1,1); %get the
loopnumbers to the matrix in the second collumn
        %create an average max. force
        av_max_f_1 = av_max_f_1 / i_1; %devide the overall amount throug
the number of loops
        av_max_force_1(1:i_1,1) = av_max_f_1; %create an matrix with the
average in the first collumn
        av_max_force_1(1:i_1,2) = loop_1(1:end-1,1); %get the loopnumbers
to the matrix in the second collumn
        %create an average looptime
        av_time_1 = av_time_1 / i_1;
        frequency_1 = 1 / av_time_1(1,1);
    end

end

%% analysis of the slope %%
%
for (i_1 = start_later_1:(length(loop_1(:,1))-end_earlier_1))
    %rise
    loop_1(i_1,8) = loop_1(i_1,5) - loop_1(i_1,7); %eight collumn =
framenumber of max. position - framenumber of min. position => amount of
frames on the rising part
    loop_1(i_1,9) = loop_1(i_1,8) / n_1; %ninth collumn = amount of
frames in each step to the max. takes
    loop_1(i_1,9) = int32(loop_1(i_1,9)); %rounded
    %fall
    loop_1(i_1,10) = loop_1(i_1+1,7) - loop_1(i_1,5); %tenth collumn =
framenumber of min. position next loop - max position of this loop =>
amount of frames of the fall
    loop_1(i_1,11) = loop_1(i_1,10) / n_1; %eleventh collumn = amount of
frames in each step to the min. takes
    loop_1(i_1,11) = int32(loop_1(i_1,11)); %rounded
    for ( m_1 = 1:n_1 ) %analysis of the rising part
        %get the positions of the limits

```

```

        position_limit_left_rise_1(i_1,m_1) = loop_1(i_1,7) + ( (m_1-1) *
loop_1(i_1,9)); %get each left limit with the endpoints and the
substracted analysissteplength
        position_limit_right_rise_1(i_1,m_1) = loop_1(i_1,7) + ( m_1 *
loop_1(i_1,9)); %get each left limit with the endpoints and the
substracted analysissteplength
        x_limit_left_rise_1(i_1,m_1) =
data_1(position_limit_left_rise_1(i_1,m_1),4); %set the value of the
limit
        x_limit_right_rise_1(i_1,m_1) =
data_1(position_limit_right_rise_1(i_1,m_1),4); %set the value of the
limit
        y_limit_left_rise_1(i_1,m_1) =
data_1(position_limit_left_rise_1(i_1,m_1),3); %set the value of the
limit
        y_limit_right_rise_1(i_1,m_1) =
data_1(position_limit_right_rise_1(i_1,m_1),3); %set the value of the
limit
        %polyfit
        x_rise_1 = [x_limit_left_rise_1(i_1,m_1),
x_limit_right_rise_1(i_1,m_1)]; %create a x-vector for the following
polyfit
        y_rise_1 = [y_limit_left_rise_1(i_1,m_1),
y_limit_right_rise_1(i_1,m_1)]; %create a y-vector for the following
polyfit
        p_r_1 = polyfit(x_rise_1,y_rise_1,1); %fit a straight line between
the left and right limit
        p_rise_1(i_1,m_1) = p_r_1(1,1); %a helpvalue for saving the exact
values for each step of the loops
        p_rise_1(i_1,m_1+n_1) = p_r_1(1,2); %a helpvalue for saving the
exact values for each step of the loops
        p_help_1 = [p_rise_1(i_1,m_1), p_rise_1(i_1,m_1+n_1)]; %create a
vector for the plotting
        y1_rise_1 = polyval(p_help_1,x_rise_1); %get the y-values for the
plotting
        %slope of each splitted part of the curve
        slope_rise_1(i_1,m_1) = p_r_1(1,1); %the first value of the vectors
are the slope set to a matrix
        slope_rise_1(i_1,m_1+1) = loop_1(i_1,1); %get the loopnumbers to the
matrix in the second column
        %summ slope rise
        summ_slope_rise_1(1,1) = summ_slope_rise_1(1,1) +
slope_rise_1(i_1,m_1); %add each loop the value of the slope for an
overall summ
        %averagee slope rise
        if ( i_1 == (length(loop_1(:,1))-end_earlier_1) )
            av_slope_rise_1 = summ_slope_rise_1(1,1) / i_1;
        end
        %plot
        hold on
        plot(x_rise_1,y1_rise_1,'-*');
    end
    for ( m_1 = 1:n_1 ) %analysis of the falling part
        %get the positions of the limits
        position_limit_left_fall_1(i_1,m_1) = loop_1(i_1+1,7) - ( (m_1-1) *
loop_1(i_1,11)); %get each left limit with the endpoints and the
substracted analysissteplength
        position_limit_right_fall_1(i_1,m_1) = loop_1(i_1+1,7) - ( m_1 *
loop_1(i_1,11)); %get each left limit with the endpoints and the
substracted analysissteplength
    end
end

```

```

        x_limit_left_fall_1(i_1,m_1) =
data_1(position_limit_left_fall_1(i_1,m_1),4); %set the value of the
limit
        x_limit_right_fall_1(i_1,m_1) =
data_1(position_limit_right_fall_1(i_1,m_1),4); %set the value of the
limit
        y_limit_left_fall_1(i_1,m_1) =
data_1(position_limit_left_fall_1(i_1,m_1),3); %set the value of the
limit
        y_limit_right_fall_1(i_1,m_1) =
data_1(position_limit_right_fall_1(i_1,m_1),3); %set the value of the
limit
        %polyfit
        x_fall_1 = [x_limit_left_fall_1(i_1,m_1),
x_limit_right_fall_1(i_1,m_1)]; %create a x-vector for the following
polyfit
        y_fall_1 = [y_limit_left_fall_1(i_1,m_1),
y_limit_right_fall_1(i_1,m_1)]; %create a x-vector for the following
polyfit
        p_f_1 = polyfit(x_fall_1,y_fall_1,1); %fit a straight line between
the left and right limit
        p_fall_1(i_1,m_1) = p_f_1(1,1); %a helpvalue for saving the exact
values for each step of the loops
        p_fall_1(i_1,m_1+n_1) = p_f_1(1,2); %a helpvalue for saving the
exact values for each step of the loops
        p_help_fall_1 = [p_fall_1(i_1,m_1), p_fall_1(i_1,m_1+n_1)]; %create
a vector for the plotting
        y1_fall_1 = polyval(p_help_fall_1,x_fall_1); %get the y-values for
the plotting
        %slope of each splitted part of the curve
        slope_fall_1(i_1,m_1) = p_f_1(1,1); %the first value of the vectors
are the slope
        slope_fall_1(i_1,m_1+1) = loop_1(i_1,1); %get the loopnumbers to
the matrix in the secound collumn
        %summ slope fall
        summ_slope_fall_1(1,1) = summ_slope_fall_1(1,1) +
slope_fall_1(i_1,m_1); %add each loop the value of the slope for an
overall summ
        %averagee slope rise
        if ( i_1 == (length(loop_1(:,1))-end_earlier_1) )
            av_slope_fall_1 = summ_slope_fall_1(1,1) / i_1;
        end

        %plot
        hold on
        plot(x_fall_1,y1_fall_1,'-*');
        xlabel('deformation [mm]')
        ylabel('force [N]')
        title('the slope for each hysteresis')
        legend('slope','location','northeast')
        grid on
        set(gcf,'position',[x0,y0,width,height])
    end
end
%% analysis of the energy %%
%
for ( c_1 = start_later_1:i_1)
    %rise
        rising_part_each_loop_1(c_1,1) = loop_1(c_1,7); %start rising part
at each start of the testingloops

```

```

        rising_part_each_loop_1(c_1,2) = loop_1(c_1,5); %end rising part at
each individual max of the testingloops
        rising_part_each_loop_1(c_1,3) = loop_1(c_1,1); %get the
loopnumbers to the matrix in the thrid collumn
        integral_x_rise_1 =
data_1(rising_part_each_loop_1(c_1,1):rising_part_each_loop_1(c_1,2),4);
%get a matrix with all x-values of the rise
        integral_y_rise_1 =
data_1(rising_part_each_loop_1(c_1,1):rising_part_each_loop_1(c_1,2),3);
%get a matrix with all y-values of the rise
        energy_rise_1(c_1,1) = trapz(integral_x_rise_1, integral_y_rise_1);
%get the integral between the x and y axis of the rise
        energy_rise_1(c_1,2) = loop_1(c_1,1); %get the loopnumbers to the
matrix in the secound collumn
        %fall
        falling_part_each_loop_1(c_1,1) = loop_1(c_1,5); %start falling
part at each max of the testingloop
        falling_part_each_loop_1(c_1,2) = loop_1(c_1+1,7); %end rising part
= startingposition next loop - 1.
        falling_part_each_loop_1(c_1,3) = loop_1(c_1,1); %get the
loopnumbers to the matrix in the third collumn
        integral_x_fall_1 =
data_1(falling_part_each_loop_1(c_1,1):falling_part_each_loop_1(c_1,2),4)
; %get a matrix with all x-values of the rise
        integral_y_fall_1 =
data_1(falling_part_each_loop_1(c_1,1):falling_part_each_loop_1(c_1,2),3)
; %get a matrix with all y-values of the rise
        energy_fall_1(c_1,1) = trapz(integral_x_fall_1, integral_y_fall_1);
%get the integral between the x and y axis of the fall
        energy_fall_1(c_1,2) = loop_1(c_1,1); %get the loopnumbers to the
matrix in the secound collumn
        %difference
        energy_diff_1(c_1,1) = energy_rise_1(c_1,1) + energy_fall_1(c_1,1);
%get the difference between the two integrals, the falling part is
negative because of that there is a "+"
        energy_diff_1(c_1,2) = loop_1(c_1,1); %get the loopnumbers to the
matrix in the secound collumn
        %energy in percent
        energy_percent_1(c_1,1) = (( energy_fall_1(c_1,1) * -1 ) /
energy_rise_1(c_1,1) ) * 100;
        energy_percent_1(c_1,2) = loop_1(c_1,1);
end
%% compare to the original data %%
%
for ( j_1 = 1:i_1)
    time_loop_1(j_1,2) = data_1(loop_1(j_1,5),1) -
data_1(loop_1(j_1,7),1); %get the time needed for the rise of the min.
force to the max. force
    av_time_1(2,1) = av_time_1(2,1) + time_loop_1(j_1,2); %get an
overall amount of time needed for all loops
    %
    for ( k_1 = loop_1(j_1,5):loop_1(j_1+1,7) ) %get the frames needed
for the fall of the max. force to a force lower than 100N
        if ( (data_1(k_1,3) < 100) && (data_1(k_1-1,3) > 100) )
%looking for the first frame where the force is the first time lower than
100N
            time_fall_1(j_1,1) = k_1; %get the position where the
force is the first time lower than 100N
        end
    end
end
%
```

```

        time_loop_1(j_1,3) = data_1(loop_1(j_1+1,7),1) -
data_1(time_fall_1(j_1,1),1); %get the time needed for the fall of the
max. force to a force lower than 100N
        av_time_1(3,1) = av_time_1(3,1) + time_loop_1(j_1,3); %get an
overall amount of time needed for all loops
    %
    if ( j_1 == i_1 )
        av_time_1(3,1) = av_time_1(3,1) / i_1; %divide the overall
amount of time through the number of testingloops for an average
        av_time_1(2,1) = av_time_1(2,1) / i_1; %divide the overall
amount of time through the number of testingloops for an average
    %
    if ( area_1 == 1 ) %1 = rearfoot
        actual_time_1(1:2,1) = av_time_1(1,1); %first collumn =
average time for one step
        actual_time_1(1:2,2) = frequency_1; %second collumn = average
frequency
        actual_time_1(1:2,3) = max(data_1(:,1)); %third collumn =
overall testingtime
        actual_time_1(1,4) = av_time_1(2,1); %fourth collumn =
average time nedded for the rise of the max. force
        actual_time_1(1,5) = av_time_1(3,1); %fifth collumn = average
time nedded for the fall of the force to sub 100N
    elseif ( area_1 == 2 ) %2 = forefoot
        actual_time_1(1:2,1) = av_time_1(1,1); %first collumn =
average time for one step
        actual_time_1(1:2,2) = frequency_1; %second collumn = average
frequency
        actual_time_1(1:2,3) = max(data_1(:,1)); %third collumn =
overall testingtime
        actual_time_1(2,4) = av_time_1(2,1); %fourth collumn =
average time nedded for the rise of the max. force
        actual_time_1(2,5) = av_time_1(3,1); %fifth collumn = average
time nedded for the fall of the force to sub 100N
    end
    %
    time_difference_1 = default_time_1 - actual_time_1; %get the
difference between supposed and actual needed time
    end
end
%% end 1 %%
%% start 2 %%
%% sort the results_2 %%
%
result_2 = importdata('Marathon2.csv'); %importing the resultdata
collums_2 = length(result_2.data(1,:)); %get amount of collums in results
rows_2 = length(result_2.data(:,1)); %get amount of rows in results
data_2 = zeros(rows_2,collums_2-1); %create a matrix with same size like
the results
%
data_2(:,1) = result_2.data(:,3); %put time at first place
data_2(:,2) = result_2.data(:,4); %put cyclenumber at second place
data_2(:,3) = result_2.data(:,1); %put force at third place
data_2(:,3) = data_2(:,3) * (-1); %set the negative values to positive
start_force_2 = min(data_2(:,3)); %detect the difference between the min
force and zero
%data(:,3) = data(:,3) - start_force_2; %set the axis to zero
data_2(:,3) = data_2(:,3) * 1000; %set force to Newton
data_2(:,4) = result_2.data(:,2); %put deformation at fourth place
data_2(:,4) = data_2(:,4) * (-1); %set the negative values to positive

```

```

start_deformation_2 = min(data_2(:,4)); %detect the difference between
the min deformation and zero
data_2(:,4) = data_2(:,4) - start_deformation_2; %set the axis to zero
%% starting conditions
%
n_2 = 1; %number of analysissteps
start_later_2 = 1; %skip the first x loops
end_earlier_2 = 5; %skip the last x loops
show_loop_2 = 5; %amount of loops shown at the end
%
%values for creating an average
av_max_f_2 = 0;
av_max_def_2 = 0;
av_min_def_2 = 0;
av_time_2(1:3,1) = 0;
summ_slope_rise_2(1,n_2) = zeros;
summ_slope_fall_2(1,n_2) = zeros;
%times it is default to be
%
area_2 = 1; %1=rearfoor ; 2=forefoot
default_time_2(1:2,1) = 1.053; %overall time for a step
default_time_2(1:2,2) = 0.95; %supposed frequency
default_time_2(1:2,3) = 12055.71; %overall testingtime
default_time_2(1,4) = 0.023; %default time for the rise for the rearfoot
default_time_2(1,5) = 0.108; %default time for the fall for the rearfoot
default_time_2(2,4) = 0.097; %default time for the rise for the forefoot
default_time_2(2,5) = 0.114; %default time for the fall for the forefoot
%
%force it is default to be
default_force_2(1,1) = 866.72; %rearfoot [N]
default_force_2(1,2) = 1612.82 ;%forefoot [N]
%% look for the testingloops
%
% the following infomatrix of the loop will be positionned like this:
% 1.=Loopnumber      2.=start_position      3.=endposition
4.=max_value
% 5.=max_value_position 6.=min_value      7.=min_value_position
8.=amount of frames on the rise
% 9.=length analysissteps on the rise      10.=amount of frames
on the fall 11.=length analysissteps on the rise
for ( z_2 = 1:1:max(data_2(:,2))) %finding existing loopnumbers
    for b_2 = 1:1:length(data_2(:,2)) %looking for the existing
loopnumbers while finding matching loopnumbers in the results and the
counter which is counting to the max loopnumber
        if z_2 == data_2(b_2,2)
            look_for_loop_number_2(z_2,1) = z_2;
        end
    end
end
end
d_2 = 1;
for (c_2 = 1:length(look_for_loop_number_2(:,1))) %eliminating every
zeros in the searching variable
    if look_for_loop_number_2(c_2,1) > 0
        loop_number_2(d_2,1) = look_for_loop_number_2(c_2,1);
        d_2 = d_2 + 1;
    end
end
end
%
for (f_2 = 1:length(loop_number_2(:,1))) %creating a vector which has the
information of starting each of the loops
    for (g_2 = 1:length(data_2(:,2)))

```

```

        if data_2(g_2,2) == loop_number_2(f_2,1)
            loop_2(f_2,1) = loop_number_2(f_2,1); %first collum = loopnumber
            loop_2(f_2,2) = g_2; %second collum = startingframe of each loop
                if (f_2 == length(loop_number_2(:,1)))
                    break
                end
            f_2 = f_2 + 1;
        end
    end
end
end

%add an extra loop
loop_2(length(loop_2(:,1))+1,1) = max(loop_2(:,1)) + 1; %adding an
extra loop for creating the last testingloopending
loop_2(length(loop_2(:,1)),2) = length(data_2(:,1)) + 1; %adding an
extra loop for creating the last testingloopending
%create the distance between the shown loops
shown_loop_steps = length(loop_2(:,1)) / show_loop_2; %determinating
the distance between the loops acquired for showing in the end
shown_loop_steps = int32(shown_loop_steps); %get an rounded number to
call the value
%% get the extrema %%
%
for (i_2 = 1:(length(loop_2(:,1))-1))
    loop_2(i_2,3) = loop_2(i_2+1,2) - 1; %third collumn = endingframe of
each loop
    %max deformation
    loop_2(i_2,4) = max(data_2(loop_2(i_2,2):loop_2(i_2,3),4)); %fourth
collum = value of each loop max
    loop_2(i_2,5) = find(data_2(:,4) == loop_2(i_2,4)); %get the position
of the maximum deformation in the overall resultmatrix
    av_max_def_2 = av_max_def_2 + loop_2(i_2,4); %get the amount of
overall max. deformation throughout all testingloops
    %min deformation
    loop_2(i_2,6) = min(data_2(loop_2(i_2,2):loop_2(i_2,3),4)); %fourth
collum = value of each loop max
    loop_2(i_2,7) = find(data_2(:,4) == loop_2(i_2,6)); %get the position
of the maximum deformation in the overall resultmatrix
    av_min_def_2 = av_min_def_2 + loop_2(i_2,6); %get the amount of
overall min. deformation throughout all testingloops
    %max force
    max_force_2(i_2,1) = max(data_2(loop_2(i_2,2):loop_2(i_2,3),3));
%create a vector which shows the max. force for each loop
    max_force_2(i_2,2) = loop_2(i_2,1); %set the loopnumber next to the
max. force so you can see what happened in which loop
    av_max_f_2 = av_max_f_2 + max_force_2(i_2,1); %get the amount of
overall max. force throughout all testingloops
    %time loop
    time_loop_2(i_2,1) = data_2(loop_2(i_2,3),1) -
data_2(loop_2(i_2,2),1);
    time_loop_2(i_2,4) = loop_2(i_2,1);
    av_time_2(1,1) = av_time_2(1,1) + time_loop_2(i_2,1);
    %
    if ( i_2 == (length(loop_2(:,1))-1) )
        %create an average max. deformation
        av_max_def_2 = av_max_def_2 / i_2; %divide the overall amount
through the number of loops
        av_max_deformation_2(1:i_2,1) = av_max_def_2; %create an matrix
with the average in the first collumn
        av_max_deformation_2(1:i_2,2) = loop_2(1:end-1,1); %get the
loopnumbers to the matrix in the second collumn
        %create an average min. deformation

```



```

        av_min_def_2 = av_min_def_2 / i_2; %devide the overall amount
through the number of loops
        av_min_deformation_2(1:i_2,1) = av_min_def_2; %create an matrix
with the average in the first collumn
        av_min_deformation_2(1:i_2,2) = loop_2(1:end-1,1); %get the
loopnumbers to the matrix in the secound collumn
        %create an average max. force
        av_max_f_2 = av_max_f_2 / i_2; %devide the overall amount throug
the number of loops
        av_max_force_2(1:i_2,1) = av_max_f_2; %create an matrix with the
average in the first collumn
        av_max_force_2(1:i_2,2) = loop_2(1:end-1,1); %get the loopnumbers
to the matrix in the secound collumn
        %create an average looptime
        av_time_2 = av_time_2 / i_2;
        frequency_2 = 1 / av_time_2(1,1);
    end

end

%% analysis of the slope %%
%
for (i_2 = start_later_2:(length(loop_2(:,1))-end_earlier_2))
    %rise
    loop_2(i_2,8) = loop_2(i_2,5) - loop_2(i_2,7); %eight collumn =
framenumber of max. position - framenumber of min. position => amount of
frames on the rising part
    loop_2(i_2,9) = loop_2(i_2,8) / n_2; %ninth collumn = amount of
frames in each step to the max. takes
    loop_2(i_2,9) = int32(loop_2(i_2,9)); %rounded
    %fall
    loop_2(i_2,10) = loop_2(i_2+1,7) - loop_2(i_2,5); %tenth collumn =
framenumber of min. position next loop - max position of this loop =>
amount of frames of the fall
    loop_2(i_2,11) = loop_2(i_2,10) / n_2; %eleventh collumn = amount of
frames in each step to the min. takes
    loop_2(i_2,11) = int32(loop_2(i_2,11)); %rounded
    for (m_2 = 1:n_2) %analysis of the rising part
        %get the positions of the limits
        position_limit_left_rise_2(i_2,m_2) = loop_2(i_2,7) + (m_2-1) *
loop_2(i_2,9)); %get each left limit with the endingpoints and the
substracted analysissteplength
        position_limit_right_rise_2(i_2,m_2) = loop_2(i_2,7) + (m_2 *
loop_2(i_2,9)); %get each left limit with the endingpoints and the
substracted analysissteplength
        x_limit_left_rise_2(i_2,m_2) =
data_2(position_limit_left_rise_2(i_2,m_2),4); %set the value of the
limit
        x_limit_right_rise_2(i_2,m_2) =
data_2(position_limit_right_rise_2(i_2,m_2),4); %set the value of the
limit
        y_limit_left_rise_2(i_2,m_2) =
data_2(position_limit_left_rise_2(i_2,m_2),3); %set the value of the
limit
        y_limit_right_rise_2(i_2,m_2) =
data_2(position_limit_right_rise_2(i_2,m_2),3); %set the value of the
limit
        %polyfit
        x_rise_2 = [x_limit_left_rise_2(i_2,m_2),
x_limit_right_rise_2(i_2,m_2)]; %create a x-vector for the following
polyfit

```

```

        y_rise_2 = [y_limit_left_rise_2(i_2,m_2),
y_limit_right_rise_2(i_2,m_2)]; %create a y-vector for the following
polyfit
        p_r_2 = polyfit(x_rise_2,y_rise_2,1); %fit a straight line between
the left and right limit
        p_rise_2(i_2,m_2) = p_r_2(1,1); %a helpvalue for saving the exact
values for each step of the loops
        p_rise_2(i_2,m_2+n_2) = p_r_2(1,2); %a helpvalue for saving the
exact values for each step of the loops
        p_help_2 = [p_rise_2(i_2,m_2), p_rise_2(i_2,m_2+n_2)]; %create a
vector for the plotting
        y1_rise_2 = polyval(p_help_2,x_rise_2); %get the y-values for the
plotting
        %slope of each splitted part of the curve
        slope_rise_2(i_2,m_2) = p_r_2(1,1); %the first value of the vectors
are the slope set to a matrix
        slope_rise_2(i_2,m_2+1) = loop_2(i_2,1); %get the loopnumbers to the
matrix in the secound collumn
        %summ slope rise
        summ_slope_rise_2(1,1) = summ_slope_rise_2(1,1) +
slope_rise_2(i_2,m_2); %add each loop the value of the slope for an
overall summ
        %averagee slope rise
        if ( i_2 == (length(loop_2(:,1))-end_earlier_2) )
            av_slope_rise_2 = summ_slope_rise_2(1,1) / i_2;
        end
        %plot
        hold on
        plot(x_rise_2,y1_rise_2,'-*');
    end
    for ( m_2 = 1:n_2 ) %analysis of the falling part
        %get the positions of the limits
        position_limit_left_fall_2(i_2,m_2) = loop_2(i_2+1,7) - ( (m_2-1) *
loop_2(i_2,11)); %get each left limit with the endingpoints and the
substracted analysissteplength
        position_limit_right_fall_2(i_2,m_2) = loop_2(i_2+1,7) - ( m_2 *
loop_2(i_2,11)); %get each left limit with the endingpoints and the
substracted analysissteplength
        x_limit_left_fall_2(i_2,m_2) =
data_2(position_limit_left_fall_2(i_2,m_2),4); %set the value of the
limit
        x_limit_right_fall_2(i_2,m_2) =
data_2(position_limit_right_fall_2(i_2,m_2),4); %set the value of the
limit
        y_limit_left_fall_2(i_2,m_2) =
data_2(position_limit_left_fall_2(i_2,m_2),3); %set the value of the
limit
        y_limit_right_fall_2(i_2,m_2) =
data_2(position_limit_right_fall_2(i_2,m_2),3); %set the value of the
limit
        %polyfit
        x_fall_2 = [x_limit_left_fall_2(i_2,m_2),
x_limit_right_fall_2(i_2,m_2)]; %create a x-vector for the following
polyfit
        y_fall_2 = [y_limit_left_fall_2(i_2,m_2),
y_limit_right_fall_2(i_2,m_2)]; %create a x-vector for the following
polyfit
        p_f_2 = polyfit(x_fall_2,y_fall_2,1); %fit a straight line between
the left and right limit
        p_fall_2(i_2,m_2) = p_f_2(1,1); %a helpvalue for saving the exact
values for each step of the loops

```

```

        p_fall_2(i_2,m_2+n_2) = p_f_2(1,2); %a helpvalue for saving the
exact values for each step of the loops
        p_help_fall_2 = [p_fall_2(i_2,m_2), p_fall_2(i_2,m_2+n_2)]; %create
a vector for the plotting
        y1_fall_2 = polyval(p_help_fall_2,x_fall_2); %get the y-values for
the plotting
        %slope of each splitted part of the curve
        slope_fall_2(i_2,m_2) = p_f_2(1,1); %the first value of the vectors
are the slope
        slope_fall_2(i_2,m_2+1) = loop_2(i_2,1); %get the loopnumbers to
the matrix in the second collumn
        %summ slope fall
        summ_slope_fall_2(1,1) = summ_slope_fall_2(1,1) +
slope_fall_2(i_2,m_2); %add each loop the value of the slope for an
overall summ
        %averagee slope rise
        if ( i_2 == (length(loop_2(:,1))-end_earlier_2) )
            av_slope_fall_2 = summ_slope_fall_2(1,1) / i_2;
        end

        %plot
        hold on
        plot(x_fall_2,y1_fall_2,'-*');
        xlabel('deformation [mm]')
        ylabel('force [N]')
        title('the slope for each hysteresis')
        legend('slope','location','northeast')
        grid on
        set(gcf,'position',[x0,y0,width,height])
    end
end
%% analysis of the energy %%
%
for ( c_2 = start_later_2:i_2)
    %rise
        rising_part_each_loop_2(c_2,1) = loop_2(c_2,7); %start rising part
at each start of the testingloops
        rising_part_each_loop_2(c_2,2) = loop_2(c_2,5); %end rising part at
each individual max of the testingloops
        rising_part_each_loop_2(c_2,3) = loop_2(c_2,1); %get the
loopnumbers to the matrix in the thrid collumn
        integral_x_rise_2 =
data_2(rising_part_each_loop_2(c_2,1):rising_part_each_loop_2(c_2,2),4);
%get a matrix with all x-values of the rise
        integral_y_rise_2 =
data_2(rising_part_each_loop_2(c_2,1):rising_part_each_loop_2(c_2,2),3);
%get a matrix with all y-values of the rise
        energy_rise_2(c_2,1) = trapz(integral_x_rise_2, integral_y_rise_2);
%get the integral between the x and y axis of the rise
        energy_rise_2(c_2,2) = loop_2(c_2,1); %get the loopnumbers to the
matrix in the second collumn
    %fall
        falling_part_each_loop_2(c_2,1) = loop_2(c_2,5); %start falling
part at each max of the testingloop
        falling_part_each_loop_2(c_2,2) = loop_2(c_2+1,7); %end rising part
= startingposition next loop - 1.
        falling_part_each_loop_2(c_2,3) = loop_2(c_2,1); %get the
loopnumbers to the matrix in the third collumn

```

```

        integral_x_fall_2 =
data_2(falling_part_each_loop_2(c_2,1):falling_part_each_loop_2(c_2,2),4)
; %get a matrix with all x-values of the rise
        integral_y_fall_2 =
data_2(falling_part_each_loop_2(c_2,1):falling_part_each_loop_2(c_2,2),3)
; %get a matrix with all y-values of the rise
        energy_fall_2(c_2,1) = trapz(integral_x_fall_2, integral_y_fall_2);
%get the integral between the x and y axis of the fall
        energy_fall_2(c_2,2) = loop_2(c_2,1); %get the loopnumbers to the
matrix in the second collumn
        %difference
        energy_diff_2(c_2,1) = energy_rise_2(c_2,1) + energy_fall_2(c_2,1);
%get the difference between the two integrals, the falling part is
negative because of that there is a "+"
        energy_diff_2(c_2,2) = loop_2(c_2,1); %get the loopnumbers to the
matrix in the second collumn
        %energy in percent
        energy_percent_2(c_2,1) = (( energy_fall_2(c_2,1) * -1 ) /
energy_rise_2(c_2,1) ) * 100;
        energy_percent_2(c_2,2) = loop_2(c_2,1);
end
%% compare to the original data %%
%
for ( j_2 = 1:i_2)
    time_loop_2(j_2,2) = data_2(loop_2(j_2,5),1) -
data_2(loop_2(j_2,7),1); %get the time needed for the rise of the min.
force to the max. force
    av_time_2(2,1) = av_time_2(2,1) + time_loop_2(j_2,2); %get an
overall amount of time needed for all loops
    %
    for ( k_2 = loop_2(j_2,5):loop_2(j_2+1,7) ) %get the frames needed
for the fall of the max. force to a force lower than 100N
        if ( (data_2(k_2,3) < 100) && (data_2(k_2-1,3) > 100) )
%looking for the first frame where the force is the first time lower than
100N
            time_fall_2(j_2,1) = k_2; %get the position where the
force is the first time lower than 100N
        end
    end
    %
    time_loop_2(j_2,3) = data_2(loop_2(j_2+1,7),1) -
data_2(time_fall_2(j_2,1),1); %get the time needed for the fall of the
max. force to a force lower than 100N
    av_time_2(3,1) = av_time_2(3,1) + time_loop_2(j_2,3); %get an
overall amount of time needed for all loops
    %
    if ( j_2 == i_2 )
        av_time_2(3,1) = av_time_2(3,1) / i_2; %divide the overall
amount of time through the number of testingloops for an average
        av_time_2(2,1) = av_time_2(2,1) / i_2; %divide the overall
amount of time through the number of testingloops for an average
        %
        if ( area_2 == 1 ) %1 = rearfoot
            actual_time_2(1:2,1) = av_time_2(1,1); %first collumn =
average time for one step
            actual_time_2(1:2,2) = frequency_2; %second collumn = average
frequency
            actual_time_2(1:2,3) = max(data_2(:,1)); %third collumn =
overall testingtime
            actual_time_2(1,4) = av_time_2(2,1); %fourth collumn =
average time nedded for the rise of the max. force

```

```

        actual_time_2(1,5) = av_time_2(3,1); %fifth column = average
time nedded for the fall of the force to sub 100N
        elseif ( area_2 == 2 ) %2 = forefoot
        actual_time_2(1:2,1) = av_time_2(1,1); %first column =
average time for one step
        actual_time_2(1:2,2) = frequency_2; %second column = average
frequency
        actual_time_2(1:2,3) = max(data_2(:,1)); %third column =
overall testingtime
        actual_time_2(2,4) = av_time_2(2,1); %fourth column =
average time nedded for the rise of the max. force
        actual_time_2(2,5) = av_time_2(3,1); %fifth column = average
time nedded for the fall of the force to sub 100N
        end
    %
        time_difference_2 = default_time_2 - actual_time_2; %get the
difference between supposed and actual needed time
        end
end
%% plots %%
%
%% compare deformation & force
%
    figure ('Name','development of the deformation and the
force','NumberTitle','off');
    subplot(2,2,1)
        plot(loop_1(0+start_later_1:length(loop_1(:,1))-
end_earlier_1,1),loop_1(0+start_later_1:length(loop_1(:,1))-
end_earlier_1,6))
        hold on
        plot(loop_2(0+start_later_2:length(loop_2(:,1))-
end_earlier_2,1),loop_2(0+start_later_2:length(loop_2(:,1))-
end_earlier_2,6))
        xlabel('loopnumber')
        ylabel('deformation [mm]')
        title('comparison of the development of the min. deformation')
        legend('min. deformation each loop testing with 1000N','min.
deformation each loop testing with 800N','location','southeast')
        grid on
    subplot(2,2,2)
        plot(loop_1(0+start_later_1:length(loop_1(:,1))-
end_earlier_1,1),loop_1(0+start_later_1:length(loop_1(:,1))-
end_earlier_1,4))
        hold on
        plot(loop_2(0+start_later_2:length(loop_2(:,1))-
end_earlier_2,1),loop_2(0+start_later_2:length(loop_2(:,1))-
end_earlier_2,4))
        xlabel('loopnumber')
        ylabel('deformation [mm]')
        title('comparison of the development of the max. deformation')
        legend('max. deformation each loop testing with 1000N','max.
deformation each loop testing with 800N','location','southeast')
        grid on
    subplot(2,2,3)
        plot(max_force_1((0+start_later_1:length(loop_1(:,1))-
end_earlier_1),2),max_force_1((0+start_later_1:length(loop_1(:,1))-
end_earlier_1),1))
        hold on
        plot(max_force_2((0+start_later_2:length(loop_2(:,1))-
end_earlier_2),2),max_force_2((0+start_later_2:length(loop_2(:,1))-
end_earlier_2),1))

```

```

xlabel('loopnumber')
ylabel('max force [N]')
title('comparison of the development of the max. force')
legend('max. force each loop testing with 1000N','max. force each
loop testing with 800N','location','west')
grid on
set(gcf,'position',[x0,y0,width,height])
%% compare energy
figure('Name','Overview of the energys','NumberTitle','off');
subplot(2,2,1)
plot(energy_rise_1(:,2),energy_rise_1(:,1)/1000)
hold on
plot(energy_rise_2(:,2),energy_rise_2(:,1)/1000)
xlabel('loopnumber');
ylabel('energy [J]')
title('comparison of the absorbed energy')
legend('energy testing with 1000N','energy testing with
800N','location','west')
grid on
subplot(2,2,2)
plot(energy_fall_1(:,2),energy_fall_1(:,1)/1000)
hold on
plot(energy_fall_2(:,2),energy_fall_2(:,1)/1000)
xlabel('loopnumber');
ylabel('energy [J]')
title('comparison of the dispensed energy')
legend('energy testing with 1000N','energy testing with
800N','location','west')
grid on
subplot(2,2,3)
plot(energy_diff_1(:,2),energy_diff_1(:,1)/1000)
hold on
plot(energy_diff_2(:,2),energy_diff_2(:,1)/1000)
xlabel('loopnumber')
ylabel('energy [J]')
title('comparison of the energydifference')
legend('energydifference testing with 1000N','energydifference
testing with 800N','location','northeast')
grid on
set(gcf,'position',[x0,y0,width,height])
subplot(2,2,4)
plot(energy_percent_1(:,2),energy_percent_1(:,1))
hold on
plot(energy_percent_2(:,2),energy_percent_2(:,1))
xlabel('loopnumber')
ylabel('energyreturn [%]')
title('returned energy in percent')
legend('energyreturn','location','southeast')
grid on
set(gcf,'position',[x0,y0,width,height])
%% compare slope %%
%
figure('Name','analysis of the overall slope of the
hysteresis','NumberTitle','off');
subplot(2,2,1)
plot(slope_rise_1(:,2),slope_rise_1(:,1))
hold on
plot(slope_rise_2(:,2),slope_rise_2(:,1))
ylabel('slope [N/mm]');
xlabel('loopnumber');
title('comparison of the slope on the rising parts');

```

```

    legend('slope testing with 1000N', 'slope testing with
800N', 'location', 'southeast');
    grid on
    subplot(2,2,2)
    plot(slope_fall_1(:,2),slope_fall_1(:,1))
    hold on
    plot(slope_fall_2(:,2),slope_fall_2(:,1))
    ylabel('slope [N/mm]');
    xlabel('loopnumber');
    title('comparison of the slope on the falling parts');
    legend('slope testing with 1000N', 'slope testing with
800N', 'location', 'southeast');
    grid on
    subplot(2,2,3)
    loglog(slope_rise_1(:,2),slope_rise_1(:,1))
    hold on
    loglog(slope_rise_2(:,2),slope_rise_2(:,1))
    ylabel('slope [N/mm]');
    xlabel('loopnumber');
    title('comparison of the slope on the rising parts');
    legend('slope testing with 1000N', 'slope testing with
800N', 'location', 'southeast');
    grid on
    subplot(2,2,4)
    loglog(slope_fall_1(:,2),slope_fall_1(:,1))
    hold on
    loglog(slope_fall_2(:,2),slope_fall_2(:,1))
    ylabel('slope [N/mm]');
    xlabel('loopnumber');
    title('comparison of the slope on the falling parts');
    legend('slope testing with 1000N', 'slope testing with
800N', 'location', 'southeast');
    grid on
    set(gcf, 'position', [x0,y0,width,height])

```