

Inhaltsverzeichnis

1	Einleitung	1
1.1	Zielsetzung der Arbeit	1
1.2	Gliederung der Arbeit	1
2	Grundlagen	3
2.1	Taktile Geräte	3
2.2	Genetischer Algorithmus	4
2.2.1	Evolution in der Biologie	4
2.2.2	Selektion	4
2.2.3	Variation	4
2.2.4	Gendrift	5
2.2.5	Allgemeiner Vorgang eines Evolutionären Algorithmus	5
2.3	Verwandte Arbeiten	6
2.3.1	Taptic Engine	6
2.3.2	Personalisierte Vibration	7
2.3.3	Personalisierte Smartwatch	7
2.3.4	Fazit	8
2.3.5	Design of a Wearable Tactile Displays	8
3	Analyse	9
3.1	Anforderungen	9
3.2	Existierende Lösungsansätze	11
3.3	Weiterer Abschnitt	12
3.4	Zusammenfassung	12
4	Entwurf	13
4.1	Programm Ablauf	13
4.2	Ausführung des Programms	13
4.2.1	Signal	13
4.3	Evolutionärer Algorithmus	14
4.3.1	Muster	18
4.4	Zusammenfassung	19
5	Implementierung	21
5.1	Signal	21
5.2	Muster	25
5.3	Evolutionärer Algorithmus	25
5.4	DNA	25
5.5	Population	25
5.6	Generationen	25
5.7	Mutation	25

6	Evaluierung	27
6.1	Studiendesign	27
6.1.1	Studiendesign	27
6.2	Analyse	29
6.2.1	Angaben der Probanden	29
6.2.2	Initialisierung der Grenzen	30
6.2.3	Auswertung der Iterationen des Algorithmus	30
6.3	Abschnitt 2	34
6.4	Zusammenfassung	34
7	Zusammenfassung und Ausblick	35
8	Zusammenfassung und Ausblick	37
	Literaturverzeichnis	39

1. Einleitung

Hinweis: In die Einleitung gehört die Motivation und Einleitung in die Problemstellung. Die Problemstellung kann in der Analyse noch detaillierter beschrieben werden.

1.1 Zielsetzung der Arbeit

Was ist die Aufgabe der Arbeit? ...

Mit dieser Bachelorarbeit verfolge ich das Ziel personalisierte Vibrationssignale zu erstellen. Hierbei werden drei verschiedene Vibrationssignale für einen Nutzer so angepasst, dass die Werte speziell für den Benutzer bestimmt werden. Zur Bestimmung der eines Vibrationssignals wird ein Evolutionärer Algorithmus verwendet. Nachdem der Evolutionäre Algorithmus passende Wert gefunden hat, wird überprüft, wie gut die personalisierten Vibrationssignale im Vergleich zu vorgegebenen Werten erkannt werden.

//Die Aufgabe besteht darin, für ein Individuum mit einem Programm drei verschiedene Vibrationssignale für Ihn so anzupassen, dass die Werte von Ihm besser erkannt werden als vorgegebene Werte. Dabei wird zur Bestimmung eines Vibrationssignals ein Evolutionärer Algorithmus verwendet. Nachdem dieser passende Wert gefunden worden ist, wird überprüft, wie gut im Vergleich zu Vorgegebenen Werten die Daten erkannt werden.

Somit wollte ich herausfinden, ob es möglich ist für jedes Individuum eine eigene passende Länge und Stärke von Vibrationssignalen zu erstellen, so dass die Kombination der Signale noch erkannt werden.

Die Hypothese die ich mit dieser Bachelorarbeit beantworten möchte ist, ob man mittels personalisierten Vibrationen eine Folge von Vibrationen besser unterscheiden als vorgegebene Vibrationen. Um dies beantworten zu können verwende ich Wearable zur Darstellung der Vibrationssignale und einen Genetischen Algorithmus um die personalisierten Vibrationen zu ermitteln.

1.2 Gliederung der Arbeit

Was enthalten die weiteren Kapitel? ...

Im Verlauf dieser Bachelorarbeit erläutere ich erst allgemein, was die einzelnen Bestandteile des Evolutionären Algorithmus sind und wie ich den an mein Problem angepasst habe.

Sowie das Problem ansatzweise heutzutage umgesetzt wurde und wie ich an das Problem heranging und wie ich es Implementiert haben.

// auskommentieren In meiner Bachelorarbeit werde ich erst einmal erläutern, was aktuell in den Smartphones und in der Forschung benutzt wird. Im Folgenden werde ich erläutern, was ein Genetischer Algorithmus ist und wie ich diesen nutze genutzt habe.

2. Grundlagen

Die Grundlagen müssen soweit beschrieben werden, dass ein Leser das Problem und die Problemlösung versteht. Um nicht zuviel zu beschreiben, kann man das auch erst gegen Ende der Arbeit schreiben.

Bla fasel...

2.1 Taktile Geräte

Ein Taktiler Gerät ist ein Gerät, dass Informationen an einen Menschen mitteilen möchte, dies geschieht durch die Wahrnehmung der Haut. Ein Taktiler Gerät ist ein Gerät, dass Informationen durch die Wahrnehmung des Menschen mittels Körperkontakt darstellt. [GOS01]

Taktile Geräte werden heutzutage öfter verwendet, als man es eigentlich wahrnimmt. Ein einfaches Beispiel ist das Handy. Eine Person hat sein eigenes Handy in der Hosentasche. Bei einer eingehenden Nachricht, muss der Benutzer mitgeteilt werden, dass eine Nachricht empfangen wurde. Normalerweise geschieht das beim abspielen des Klingeltons. Falls man beschäftigt ist und nicht durch ein lautes klingeln gestört werden will, stellt man den Ton ab. Um dennoch darauf aufmerksam zu machen, dass eine Nachricht eingetroffen ist, nutzt man die Vibration des Handys.

Ein weiteres Beispiel in der Taktile Displays benutzt werden ist für Blinde Menschen um Wissen zu übermitteln [PB03]. Für Blinde gibt es spezielle Karten, die Konturen für die Umrisse der Länder darstellen, solche Karten sind selten vertreten. Um ein solches Ungleichgewicht zwischen den blinden und normalsehenden Schülern zu mindern, hat man sich ein System für entwickelt, das audio und taktile Eindrücke vereint und somit Dabei hat man sich für einen räumlich akustisches Soundsystem entschieden um den Schüler anhand Geräuschen aus verschiedenen Richtungen ein Gefühl zu geben soll wo man sich aktuell befindet und was um einen existiert. Dabei wurden nicht nur Tiergeräusche und Verkehrsgeräusche verwendet, sondern auch den Namen der Region, in der man sich aktuell befindet. Als Taktiles Device hat man sich an schon existierenden Mäusen und Controllern bedient, die Force-Feedback besaßen.

Die Karten die man verwendet hat sind Karten, diese sind nicht so detailliert wie normale Karten sind. Im Vergleich zu der normalen Karten hat man hier weniger Informationen die man übertragen muss und die Details der Ländergrenzen konnte man leichter darstellen.

Damit ein Schüler eine Karte erkunden kann bewegt er mithilfe eines Eingabegeräts über die Karte und drückt eine Taste auf der Tastatur um Informationen über das Gebiet abzurufen, die über Ausdosignale übermittelt werden. Das Taktile Device wurde verwendet um Grenzübergänge zu mittels Vibrationen zu signalisieren.

2.2 Genetischer Algorithmus

Unter dem Begriff der Evaluationären Algorithmen (EA) versteht man eine Ansammlung von Techniken und Methoden die für Optimierungsprobleme eine Lösung findet, die das Problem näherungsweise löst. [Wei15] /////Man hat sich bei dem EA an der Biologischen Evolution von Darwin inspirieren lassen. [Sel03]

2.2.1 Evolution in der Biologie

Im 19. Jahrhundert hat sich Darwin mit den Gedanken über die Evolution von Lebewesen über mehrere Generationen gemacht. Dabei kam hat er auf das Prinzip “Survival of the fittest. Mit dem Prinzip hat Charles Darwin die Entstehung neuer Arten beschrieben. Dabei haben die Arten überlebt, die sich am besten für die Umgebung angepasst haben. Die Arten, die sich an die neue Situation nicht angepasst hatten sind nach ein paar Generationen zur Minderheit geworden. [?]

In der Biologie ist jeder lebende Organismus ein **Individuum**. Jedes Individuum besitzt Erbinformationen in der Form von Chromosomen. Die Erbinformationen werden auch **Gene** oder **DNA** genannt. Eine Gruppe von Individuen wird als **Population** bezeichnet.

Bei dem EA hat man sich das Verhalten des in der Biologie angeschaut, wie dort die Arten über Generationen hinweg überleben und versucht dieses zu adaptieren. [Fli]

2.2.2 Selektion

Eine **Selektion** ist eine Auswahl von Individuen einer Population. Durch Kombination der Gene der ausgewählten Individuen, werden neue Individuen für die nächste Generation erzeugt. Diese Kombination wird **Rekombination** genannt. Dabei gibt es verschiedene Selektionsstrategien. Man versucht die Individuen zu finden, die eine bestmögliche Lösung für ein Problem liefern. [Wei15, Fli]

2.2.3 Variation

Für eine Population sollte man zu Beginn eine große **Variation** von Individuen mit unterschiedlichen Genen besitzen. Nehme man anhand einem Beispiel an, dass man mithilfe einer Evolution eine neue Art von Süßigkeiten entwickeln möchte. Dabei sollten die Form und Farbe neu bestimmt werden. Wenn man jetzt eine Start-Population von Individuen habe, die alle die gleichen Gene besessen würde, so würde man anhand zwei Selektierten Individuen keine Änderung der nächsten Generation erkennen, da die Nachkommen auch alle die gleichen Gene besitzen würden, falls keine Mutation auftreten würde. Damit dieses Verhalten nicht auftritt, versucht man zu Beginn eine große Variation an Individuen zu erzeugen und diese als Anfangs Population für einen Evolutionären Algorithmus zu nutzen. Mittels Rekombination und Mutation wird dabei ein neues Individuum für die nächste Generation erzeugt, dass neue Gene vorweist. Mittels der Rekombination werden verschiedene Gene bei der Fortpflanzung der nächsten Generation vermischt und somit neue Gene für erzeugt. Die Mutation ist eine Veränderung der Gene eines Individuums, die meistens durch Umwelteinflüsse ausgelöst werden. [Fli]

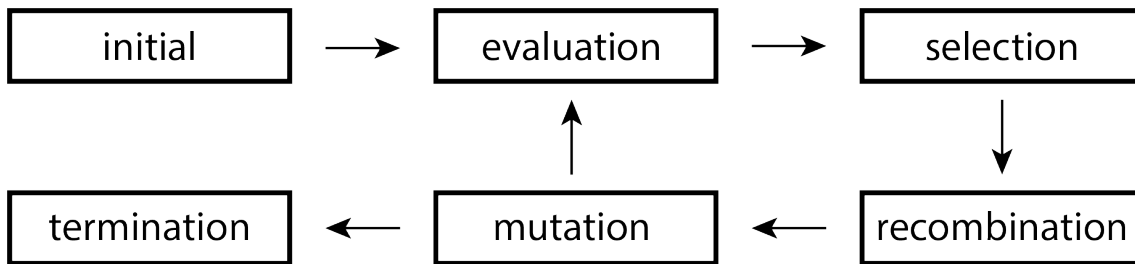


Abbildung 2.1: Ablauf des Algorithmus

2.2.4 Gendrift

Unter **Gendrift** versteht man eine zufällige Veränderung der Genhäufigkeit in einer Population, innerhalb einer Evolution. Ein Gendrift tritt bei kleineren Populationen häufiger auf. [bro5]

2.2.5 Allgemeiner Vorgang eines Evolutionären Algorithmus

Der EA besitzt im allgemeinen immer die gleichen Komponenten. Diese Komponenten werden im folgenden erläutert. Dieses Wissen stammt aus den folgenden Quellen [Shi12, Fli, Wei15]

Initialisierung

In der Initialisierung erzeugt man sich eine Population von Individuen, die eine zahlreiche Variation von Genen besitzen. Für die Anfangs-Population nimmt man in der Praxis zufällige Individuen oder auch die besten bekanntesten Lösungskandidaten.

Bewertung der Individuen

Bevor man eine Evolution für eine Population erzeugen kann, benötigt man Attribute anhand derer man die nächste Generation berechnet. Mit sogenannten **Fitnessfunktion** berechnet anhand der Attribute eines jeden Individuums einen **Fitnesswert**. Der Fitnesswert ist entscheidend für die Selektion.

Selektion

Die Selektion wählt zwei Individuen als Eltern aus, diese werden anhand der Fitnesswert der Individuen aus einer Population bestimmt. Es werden die Individuen bevorzugt, die einen hohen Fitnesswert aufweisen. Diese ermittelten Eltern werden für die Rekombination und Mutation weiter verarbeitet.

Rekombination

Die Eltern aus der Selektion werden in diesem Schritt einen Nachfahren für die nächste Generation erzeugen. Dabei werden die jeweiligen Gene der Eltern kombiniert und an die nächste Generation vererbt. Die Kombinationsmöglichkeiten hängen davon ab, wie die Gene repräsentiert sind. Die neuen Individuen werden in die Population für die nächste Generation hinzugefügt.

Mutation

Nach der Rekombination besteht eine Chance, dass die Gene der nächste Generation mutieren können.

Terminierung

Man führt für eine Generation die Selektion, Rekombination und Mutation so oft aus, bis man die gleiche Anzahl an Individuen für die nächste Generation erzeugt hat. Der Vorgang der Bewertung, Selektion, Rekombination und Mutation muss für jede neue Generation durchgeführt werden. Die neu erzeugte Population wird als Eingabe für die neue Generation verwendet. Dabei wird der Algorithmus so lange ausgeführt bis eine hinreichende Abbruchbedingung erreicht worden ist. Je nach Definition kann nach einer festen Anzahl an Generationen oder erst nachdem es keine signifikanten Änderungen der Generationen gibt den EA terminieren. Das dabei erzeugte Ergebnis ist eine näherungsweise Lösung für das Problem.

2.3 Verwandte Arbeiten

Hier kommt „Related Work“ rein. Eine Literaturrecherche sollte so vollständig wie möglich sein, relevante Ansätze müssen beschrieben werden und es sollte deutlich gemacht werden, wo diese Ansätze Defizite aufweisen oder nicht anwendbar sind, z. B. weil sie von anderen Umgebungen oder Voraussetzungen ausgehen.

Da heutzutage beinahe jedes Gerät ein Vibrationsmotor verbaut hat, sei es das Handy, die Smartwatches oder Fitnessarmbänder (uvm.), werde ich im folgenden noch auf einige aktuelle Technologien und deren Umsetzung der personalisierten Vibrationsmuster zu sprechen kommen.

2.3.1 Taptic Engine

Die Taptic Engine ist eine von der Firma Apple selbst entwickelte Vibrationsmotor, dass heutzutage in nahezu allen Apple Produkten verbaut ist. Das erste Gerät was die Taptic Engine bekommen hat, war die Apple Watch. Der Name **Taptic** bildet sich aus dem Wörtern "Taktild und "Haptisch". Trotz des neu erfinden einer mechanischen Rückmeldung bietet Apple keine Personalisierung, wie lange eine Rückmeldung, für die Apple Watch, erfolgen soll. Die Einstellungsmöglichkeiten an der Apple Watch ist lediglich die Stärke der Vibration. Diese ist in 3 Stärkestufen unterteilt. Meiner Ansicht nach, kann man daher nicht wirklich von einer personalisierten Vibration sprechen.



Abbildung 2.2: Settings on the apple watch, possible settings on the martian watch

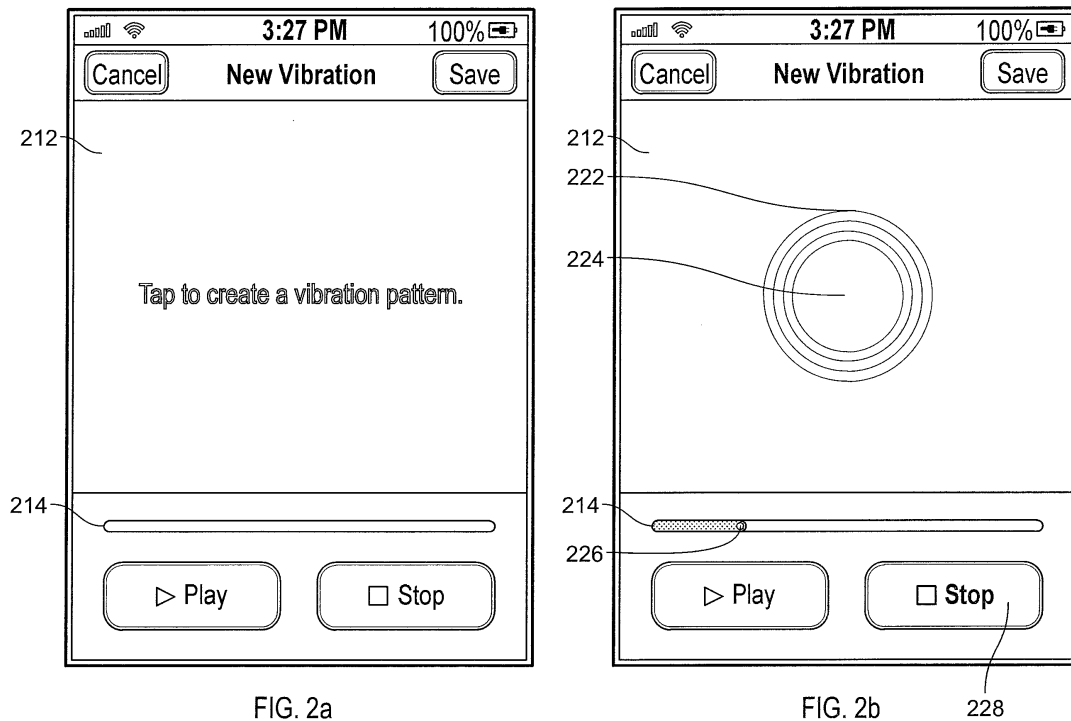


FIG. 2a

FIG. 2b

228

Abbildung 2.3: custom vibrations on the iPhone

2.3.2 Personalisierte Vibration

Der Hersteller Apple hat auch bei dem iPhone eine Möglichkeit geboten eigene Vibrationsmuster zu erstellen, jedoch mit Einschränkungen. Wenn man in die jeweilige Einstellung der iPhones gelangt, erscheint das folgende Bild. Beim drücken auf das Display wird an der Stelle eine Vibration erzeugt. Man hat 10 Sekunden um ein eigenes Muster zu erzeugen, indem man wiederholt auf den Bildschirm drückt. An der Stelle, an der man den Bildschirm berührt hat, erscheint visuell um der Position ein Kreis. Die erzeugten Vibrationen werden in einer Leiste visuell angezeigt. Man kann sich beliebig viele Vibrationsmuster speichern, die bis zu 10 Sekunden lang sind. [FSDS16]

Die Einschränkung die man hier nennen muss ist, dass man die Vibrationsmuster nur für Systeminterne Funktionen benutzen kann. Dies bedeutet, dass man die Funktionen für Klingeltöne, Nachrichtentöne, Erinnerungshinweise, Kalenderhinweise (o.ä.) hinzufügen kann. Für eine andere Anwendung, die nicht im Betriebssystem integriert ist, ist das nicht möglich. Somit können Benachrichtigungen von anderen Entwicklern keine eigene Vibrationsmuster erhalten. Daraus folgt, wenn das iPhone in der Hosentasche ist und ich eine Benachrichtigung von einer Application erhalte, die nicht im System integriert gewesen ist, kann man anhand der Vibrationen des iPhones nicht unterscheiden welche Application dies gewesen ist.

2.3.3 Personalisierte Smartwatch

Das Gerät, dass es nach meiner Recherche am besten gelöst hat, ist eine Smartwatch von einem kleinen StartUp namens Martian. Das Startup hat eine Uhr hergestellt, mit der man mittels einer App auf dem Smartphone die Vibrationsmuster selbst anpassen kann. Die App unterstützt eine große Anzahl an Applications, von anderen Herstellern, die Benachrichtigungen senden. Ein Vibrationsmuster für die Uhr kann man aus mit zu

4 Signalen auf der Uhr darstellen lassen. Die Signale sind als Lang, Kurz und Pause festgelegt. Somit kann man mittels der Vibration der Uhr herausfinden, welche App gerade eine Benachrichtigung auf mein Handy gesendet hat. Die Länge und Stärke eines Signals ist schon im Vorfeld festgelegt.

2.3.4 Fazit

Bei sehr vielen Herstellern ist es aktuell noch gar nicht möglich eigene Vibrationsmuster zu erstellen. Bei Android Geräten ist es aktuell so, dass man nur aus einer Menge von wenigen Vordefinierten Vibrationsmustern sich nur eines auswählen kann. Einige Entwickler haben dieses Problem erkannt und eine eigene Applikationen entwickelt und im Store veröffentlicht..

(BILD)

2.3.5 Design of a Wearable Tactile Displays

In dem Paper von XXX [1] handelt es um Taktile Displays; was man bei der Erschaffung von Taktilen Display beachten soll und was für Verwendungszwecke es noch gibt. Das Paper ist schon mehr als 15 Jahre alt und weist dennoch auf Informationen hin, die heutzutage noch Relevant sind. Die Inhalte des Papers werden im folgenden beschrieben. Taktile Displays sind Geräte, die dazu benutzt werden um Informationen durch Körperkontakt des Menschen über Haptisches Feedback zu übermitteln. Diese bilden keinen Konflikt mit den audio-/visuellen Einflüssen. Die Taktilen Displays sind eine Unterstützung der Darstellung der Informationen. Beispielsweise kann man Blinden oder Tauben Informationen mittels Taktilen Displays vermitteln, die Sie nicht wahrnehmen können. Dabei werden haptische / sensorische assistive Geräte benutzt um die Informationen in der Realen Umgebung in taktile Simulationen umzuwandeln. Ein wichtiger Aspekt im Paper ist es gewesen, wie man ein solches Taktilen Display entwirft und welche Aspekte man beachten soll. Beim Design eines Taktilen Devices sollten leise und leicht, klein sein; wenig Strom verbrauchen; die Taktoren sollten durch Kleidung spürbar sein und am besten so eng wie möglich am Körper liegen. Handys schon damals Vibrationsmotoren gehabt um einem Nutzer darauf Aufmerksam machen wollte, dass man eine Nachricht erhalten hatte. Die Vibration des Handys war eine Metapher dazu, dass eine andere Person einem an die Schulter rütteln würde. [1] Um es nicht nur Theoretisch zu erklären hat man ein Taktilen Display als Weste entworfen, bei dem man die Vibrationsmotoren aus Nokia Handys verwendet hat. Mittels der Weste sollte eine Person vom Punkt A zu Punkt B navigiert werden. Die übermittelten Informationen zur Navigation sind vorwärts, zurück, rechts, links, beschleunigen und verlangsamen gewesen.

3. Analyse

In diesem Kapitel sollten zunächst das zu lösende Problem sowie die Anforderungen und die Randbedingungen einer Lösung beschrieben werden (also nochmal eine präzisierte Aufgabenstellung).

Dann folgt Üblicherweise ein Überblick über bereits existierende Lösungen bzw. Ansätze, die meistens andere Voraussetzungen bzw. Randbedingungen annehmen.

3.1 Anforderungen

Anforderungen und Randbedingungen ...

Die Aufgabenstellung bestand darin, herauszufinden, ob ein personalisiertes Vibrationsmuster besser als ein vorgegebenes Vibrationsmuster erkannt wird. Man sollte in dieser Bachelorarbeit ein System bauen, dass mit dem Wearable kommuniziert und Daten übertragen kann, so dass das Wearable Vibrationen abspielt. Darüber hinaus soll eine Studie erstellt werden, in der man herausfindet, ob es einen signifikanten Unterschied zwischen generischen und genetischen Vibrationen gibt. Dabei sind die generischen Vibrationen fest vorgegeben, wobei die genetischen Vibrationen für den Probanden so angepasst werden sollen, dass diese genau für ihn angepasst werden. Dabei habe man ein Evolutionären Algorithmus verwendet um die genetische Vibration für einen Probanden zu bestimmen.

Um dies zu lösen musste man sich im Vorfeld ein paar Gedanken über die Repräsentation eines Signals, die Dekodierung und über die Übertragung machen. Im Folgenden werden diese Entscheidungsfindungen beschrieben.

Wearable

Damit man Vibrationen überhaupt darstellen kann, hätte man sich ein eigenes Taktils Device entwerfen müssen. In meinem Fall war dies nicht Nötig, da das Wearable vom TECO vorgegeben war. Die Komponenten des Wearables sind zwei Taktoren (TI TLC5971), einen Mikroprozessor (nRF51822 / BLE Nano) und eine Batterie.

Die Kommunikation des Armbands konnte man über eine Serielle Schnittstelle, sowie über Bluetooth Low Energie (LE) realisieren. Bei der seriellen Schnittstelle ist ein USB-Kabel nötig gewesen um den PC mit dem Wearable anzusprechen, dabei wäre die Mobilität verloren gegangen. Daher habe man die verfügbare BLE Schnittstelle benutzt um zwischen dem PC und dem Wearable zu kommunizieren.

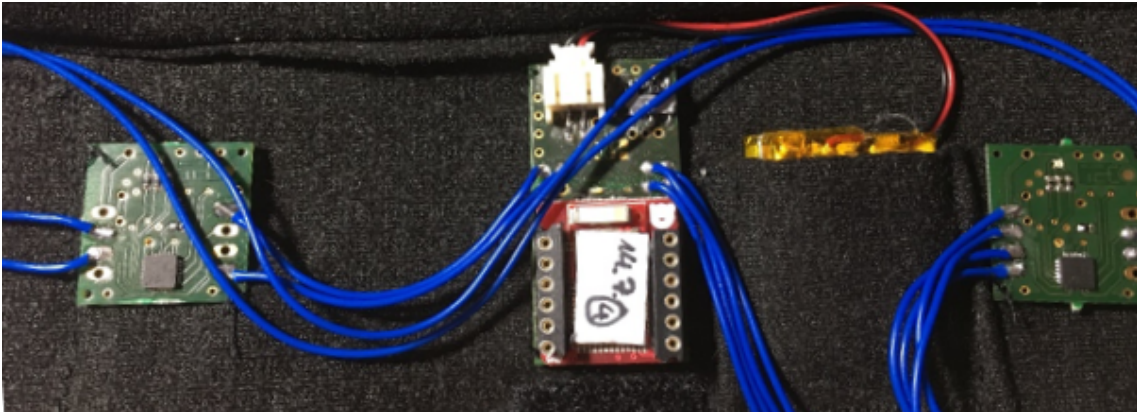


Abbildung 3.1: Armband geöffnet. Komponenten 2 Traktoren, Batterie, und Mikroprozessor



Abbildung 3.2: Armband angezogen

Bluetooth LE überträgt Nachrichten nur in einer maximalen Größe von 20 Bytes. Diese werden jeweils als zwei Bytes Blöcke übertragen.

Anhand der Begrenzung der Datenübertragung hat man sich eine geeignete Dekodierung des Signals überlegen müssen.

Die Programmiersprache um die Kommunikation zwischen dem PC und dem Wearable aufzubauen, konnte frei gewählt werden. Man habe sich hierbei zuvor verschiedenste Programmiersprachen angeguckt. Java hat man nach fehlenden Informationen zum Kommunikationsaufbau ausgeschlossen. Man hat sich letztendlich für C# entschieden, da man eine funktionierende Kommunikation aufgebaut hat und man gleichzeitig die Grafische Oberfläche nutzen kann um Studie darin durchzuführen.

Dabei musste herausfinden was für Einstellungsmöglichkeiten man am Wearable hat. Diese waren die Länge und die Stärke eines Signals.

Signale

Ein Signal repräsentiert eine Vibration, die auf dem Wearable abgespielt wird. Es hat sich die Frage gestellt, wie personalisiert man denn jetzt Signale. Dabei hat man das Signal in drei verschiedene Längen definiert, Kurz, Mittel und Lang.

Dabei hat man als Vorgabe gehabt, dass man einen Evolutionären Algorithmus benutzt. Diesen EA hat man verwendet, um für die Signaltypen ein personalisierten Wert bestimmen.

Dabei hat man sich ein Vorgang überlegen müssen, um für den EA eine Startpopulation zu erzeugen, die Individuen der Population zu bewerten, mittels einer Fitnessfunktion den Fitnesswert für jedes Individuum berechnen, eine geeignete Selektion, Reproduktion und Mutation bestimmen, sowie dass die Terminierungsbedingung.

Eine geeignete Signalrepräsentation musste definiert werden um Signale vom PC zum Wearable über BLE zu senden und abspielen zu lassen. Das bedeutet sowohl am PC als auch am Wearable selbst musste dies definiert werden.

Studie

Es sollte eine Grafische Oberfläche erstellt werden, damit der Benutzer Studie komplett über das Programm durchführen kann indem er die Instruktionen des Programms folgt und er auch selbst entscheiden kann, wann er ein Signal abgespielt haben will.

Es hat sich durch das Definieren der Signale und dem EA ergeben, dass die Studie in drei Schritten unterteilt wird. Der erste Schritt ist es, die Grenzen für eine Startpopulation für einen EA zu bestimmen. Der zweite Schritt ist es, einen genetischen Wert für die eigentliche Studie zu bestimmen, dies passiert durch ausführen mehrerer Generationen des EA. Der letzte Schritt ist es, generische und genetische Signale vom Benutzer bewerten zu lassen.

Alle Daten sollten überprüft werden und es sollte herausgefunden werden ob es am signifikante Unterschiede gibt.

3.2 Existierende Lösungsansätze

Hier kommt eine ausführliche Diskussion von „Related Work“.

Abbildungen sollten möglichst als EPS (Encapsulated Postscript) bzw. PDF eingebunden werden. Zur Erzeugung sauberer EPS-Dateien empfiehlt sich das Tool **ps2eps** zur Nachbearbeitung von Postscript-Dateien. Mit **epstopdf** kann dann eine PDF-Datei zum Einbinden erzeugt werden.

Abbildung 3.3: Testabbildung

3.3 Weiterer Abschnitt

Bla faasel... hat auch schon [?] gesagt und [?, ?, ?] sollte man mal gelesen haben. Abbildung 3.3 auf S. 12 sollte man sich mal anschauen.

Blindtext Blindtext Blindtext Blindtext Blindtext Blindtext Blindtext Blindtext Blindtext Blindtext
Blindtext Blindtext Blindtext Blindtext Blindtext

Blindtext Blindtext Blindtext Blindtext Blindtext Blindtext Blindtext

3.4 Zusammenfassung

Am Ende sollten ggf. die wichtigsten Ergebnisse nochmal in *einem* kurzen Absatz zusammengefasst werden.

4. Entwurf

In diesem Kapitel erfolgt die ausführliche Beschreibung des eigenen Lösungsansatzes. Dabei sollten Lösungsalternativen diskutiert und Entwurfsentscheidungen dargelegt werden.

4.1 Programm Ablauf

4.2 Ausführung des Programms

Zu Beginn hat man sich über das gesamte Problem einen Überblick schaffen müssen. Dabei hat man sich alle einzelnen Bestandteile im Detail angeguckt, um herauszufinden welche Komponenten man wie zusammensetzt. Es haben sich drei / vier große Komponenten ergeben, die man erledigen musste. HMMM

Zuerst habe man sich mit dem Wearable auseinandergesetzt, um zu bestimmen, was man alles mit dem Wearable umsetzen kann. Man fand dabei heraus, dass man zwei Parameter hatte die man einstellen konnte. Diese waren die Zeit, wie lang das Wearable vibriert und die Stärke, wie stark das Armband vibrieret. Um ein Signal zu übertragen hatte man eine Beschränkung von 20 Bytes, die man maximal in einer Nachricht mittels Bluetooth übertragen kann.

Als nächste Komponente des Problems habe man sich eine geeignete Repräsentation eines Signals überlegen müssen. Um nicht außer Bedacht zu lassen ist diese die wichtigste Komponente des Problems, denn es hängen alle anderen Komponenten davon ab, wie diese Daten repräsentiert werden.

Daher stellte sich die Frage, wie sieht ein solches Signal aus?

4.2.1 Signal

Anhand der zwei Parameter, die Länge und die Stärke des Wearables, hat man diese als Attribute eines Signals definiert.

Technisch gesehen hätte man für die Länge eines Signals von 0 ms (0x0000) bis 65535 ms (0xFFFF) nutzen können. Man hat sich jedoch anhand vorheriger Studien [PBB16] orientiert und ein Minimum von 100 ms und ein Maximum von 1000 ms als Grenzen für die Länge übernommen.

Bei der Stärke eines Signals hat man sich auch hier am Wearable orientiert. Die Repräsentation der Stärke musste man aus technischen Gründen ab einem Wert von 32767 (0x7FFF)

Tabelle 4.1: Bewertung eines Probanden die Ideal ist

Signallänge (in ms)	100	200	300	400	500	600	700	800	900	1000
Erkannten Signaltyp	Kurz	Kurz	Kurz	Mittel	Mittel	Mittel	Lang	Lang	Lang	Lang

definieren. Dies war nämlich der Wert, ab dem beide Traktoren angefangen haben spürbar als Vibrationen wahrgenommen zu werden. Vor dem genannten Wert ist zu wenig Strom übertragen worden, sodass kein Traktor funktionierte. Daher hat man sich zwischen den Grenzen von (0x7FFF) und (0xFFFF) für die Stärke festgelegt.

Anhand der Hypothese der Bachelorarbeit wolle man wissen, wie gut sich personalisierte Vibrationen im Vergleich zu generischen Vibrationen verbessern. Dabei musste man sich überlegen, wie man personalisierte Vibrationen mittels dem EA für einen Probanden bestimmen will. Außerdem sollte man herausfinden was für Signale noch gut voneinander unterscheidbar sind, die die Grenzen von 100ms bis 1000ms besitzen. Man habe sich auf drei Typen von Signalen festgelegt: **Kurz**, **Mittel** und **Lang**. Dadurch wolle man wissen, wie ein Signal wahrgenommen wird und welchen Typen man dem Signal zuordnen würde.

Die jeweiligen Typen definieren innerhalb der Grenzen zwischen 100 ms und 1000 ms ein Intervall, die sich nicht überschneiden. Abgesehen davon ist **kurz**, das zahlenwertigste kleinste Intervall, gefolgt von **mittel**, **lang** ist das zahlenwertigste größte Intervall. Allerdings wurde es darauf geachtet, dass die Grenzen nicht aufeinander liegen, sondern einen Abstand zwischen den Grenzen existiert. Als Beispiel habe man für Kurz die Intervallgrenzen 100ms und 300ms, für Mittel habe man dann die Intervallgrenzen von 400ms bis 600ms und für Lang habe man die Intervallgrenzen von 700ms bis 1000ms.

Für die Stärke der Vibration war man an den Darstellbarkeit des Wearable gebunden. Zwischen die Grenzen von (0x7FFF) und (0xFFFF) hat man sich fünf voneinander unterscheidbare Stufen definiert.

4.3 Evolutionärer Algorithmus

Grenzen Initialisieren

Als Eingabe für den Algorithmus benötigt man eine Start Population / Anfangspopulation von Individuen. Die Individuen sind in diesem Fall Signale. Nicht jede Person empfindet ein vorgegebenes Kurzes Signal gleich wie eine andere Person, somit musste man zuvor den Benutzer befragen, was er als Kurz, Mittel und Lang empfindet. Zuerst habe man alle Signale abgespielt, damit der Proband wusste, was Ihn erwartet. Nachdem alle Signale abgespielt wurden, wurde jedes Signal einzeln abgespielt und der Proband hatte die Aufgabe das abgespielte Signal der Kategorie Kurz, Mittel oder Lang zuzuordnen. Jedem Probanden wurde insgesamt 10 Signale mit der gleichen Vibrationsstärke abgespielt, die Signallänge ist dabei gleich verteilt. Dabei wurden alle 10 Signale in einer zufälligen Reihenfolge abgespielt.

Nach der Eingabe des Probanden erhält man beispielsweise folgende Bewertung Tabelle 4.1 erhalten. Diese Eingabe ist ideal, da alle Signaltypen direkt hintereinander vorliegen, so würde man hier einfach die Grenzen aus der Tabelle entnehmen können. Diese belaufen sich für Kurz zwischen 100 und 300 ms, für Mittel zwischen 400ms und 600ms und für Lang zwischen 700 und 1000ms.

Falls die Eingabe jedoch nicht so Ideal sein sollte, wie im Beispiel Tabelle 4.2, mussten ein paar Vorkehrungen getroffen werden. Um einige Sonderfälle auszuschließen, hat man überprüft, ob das Signal mit der Länge von 100ms ein Kurzes Signal ist und das Signal mit der Länge von 1000ms ein Langes Signal ist, sowie man annimmt, dass mindestens

Tabelle 4.2: Bewertung eines Probanden die nicht Ideal ist

Signallänge (in ms)	100	200	300	400	500	600	700	800	900	1000
Erkannten Signaltyp	Kurz	Kurz	Mittel	Mittel	Lang	Mittel	Lang	Lang	Lang	Lang

Tabelle 4.3: Bestimmung der Grenzen

Iterationen	1.	2.	3.	4.	5.	6.	7.	8.	10.
$Kurz_{Min}$	100	100	100	100	100	100	100	100	100
$Kurz_{Max}$	100	200	200	200	200	200	200	200	200
$Mittel_{Min}$	100	200	300	300	300	300	300	300	300
$Mittel_{Max}$	100	200	300	400	400	600	600	600	600
$Lang_{Min}$	100	200	300	400	500	600	700	700	700
$Lang_{Max}$	100	200	300	400	500	600	700	800	1000

jeder Signaltyp mindesten zwei mal ausgewählt wurde. Sollte dies nicht der Fall sein, so würde man den Benutzer dazu bitten, die zehn Signale erneut zu bewerten.

Man beginnt damit die neuen Intervallgrenzen zu bestimmen. Diese wurden anhand des Beispiels Tabelle 4.3 exemplarisch bestimmt, dabei ist jede Spalte eine Iteration. Dabei bestimmt man zuerst den kleinsten Wert und setzt den für alle Grenzen gleich. Falls ein größerer Wert des für den gleichen Signaltypen existiert, erhalten alle nachfolgenden Grenzen den gleichen Wert zugewiesen. Wenn für den nächsten Signaltyp ein größerer Wert vorliegt werden nur noch die nachfolgenden Signaltypen nach diesem Wert angepasst. Nachdem die 10 Iterationen ausgeführt sind, untersucht man, ob die Grenzen nicht identisch sind und ob die Grenzen jeweils aufsteigend sind.

$Kurz_{Min} < Kurz_{Max} < Mittel_{Min} < Mittel_{Max} < Lang_{Min} < Lang_{Max}$ Fall dies nicht der Fall sein sollte, dann wird der Benutzer erneut aufgefordert die 10 Signale zu bewerten.

Bei einer kleinen Abweichung von zwei nebeneinanderliegen von zwei Werten ist dies noch akzeptabel, bei größeren Abweichungen, hat man Benutzer noch einmal darum gebeten erneut zu bewerten. Im Verlauf der Studie musste man nur 15% aller Probanden ein weiteres mal darum bitten, die Signale neu zu bewerten.

Population

Sind die Grenzen für eine Probanden bestimmt, kann darauffolgend die Startpopulation erzeugt werden. Bei der **Startpopulation** hat man für jeden Signaltypen zehn Signale. Zwei Signale der zehn Signale repräsentieren das Minimum und das Maximum des jeweiligen Signaltypen. Die restlichen acht Signale erhalten eine zufällige Länge innerhalb des Intervalls. Die Stärke eines Signal wird zufällig für jedes Signal zufällig zugewiesen. Man besitzt somit für die Startpopulation dreißig Signale, die jeweils in zehn Kurze, Mittlere und Lange Signale unterteilt sind.

An dieser Stelle muss man besonders betonen, dass man nur beim ersten mal eine **Startpopulation** erzeugt. Für die Erzeugung der Populationen für die nächsten Generationen werden die Grenzen der Signaltypen nicht explizit hinzugefügt.

Bewertung des Algorithmus

Jedes Individuum einer Population muss vor der Ausführung des nächsten Schritts des Algorithmus bewertet werden. Um diese Bewertung zu erhalten hätte man aus einigen Alternativen wählen können wobei ich hier ein weiteres erläutern werde und wieso man dieses nicht gewählt hat.

Tabelle 4.4: Signalstärke

Signalstärke	Namen	Zustandsnamen
0x7FFF	Very Weak	q_0
0x9FFF	Weak	q_1
0xBFFF	OK	q_2
0xDFFF	Strong	q_3
0xFFFF	Very Strong	q_4

Zuerst könnte man hergehen und jedes Signal aus der Population dem Probanden abspielen und fragen was für ein Signaltyp er erkannt hat, wie man es bei den ersten zehn Signalen gemacht hat um die Grenzen der Signaltypen zu bestimmen.

Der erste Ansatz wäre man würde die Anzahl der richtigen Zuweisungen und Abweichungen zählen um mit einer geeigneten Fitnessfunktion den Fitnesswert zu bestimmen. Ferner würde das bedeuten, dass für eine Population ein Individuum mehrmals abgespielt werden müsse, um eine Bewertung des einzelnen Individuums zu erhalten. Bei einer Annahme von fünf Bewertungen pro Individuum, um nur eine Population damit bewerten zu können, müsste der Proband eine Anzahl von 150 Signalen bewerten, damit der Algorithmus eine Generation bestimmen kann. Dies würde bedeuten, dass wenn man vier Generationen bestimmen wollen würde, man auf eine Anzahl von 600 Vibrationen kommt, die der Benutzer bewerten sollte um einen personalisierten Wert zu erhalten.

Aufgrund der Tatsache, dass Probanden nicht so lange an einer freiwilligen Studie teilnehmen wollen, habe man sich für einen anderen Ansatz entschieden. Man muss akzeptieren, dass man über mehrere Generationen hinweg die komplette Population bewerten muss. Das heißt, dass man pro Generation 30 Signale abspielen muss. Um eine Generation zu bestimmen habe man sich Gedanken darüber gemacht, wie man so ein Signal bewerten soll. Dabei ist man zu dem Entschluss gekommen, dass man den Benutzer zuerst fragt, was für ein Signaltyp er erkannt habe. Weiterhin habe man ihn zwei Fragen wie in einem Fragebogen (Likert-Skala) gestellt, wie gut er das Signal erkannt hat und wie man die Stärke des Signals empfand.

Wie schon erwähnt hat jedes Individuum einer Population eine zufällige Stärke zugewiesen. Dabei entspricht die Stärke einem Zustand aus dem Zustandsdiagramm ??, der in der ersten Population zufällig bestimmt worden ist, weshalb auch kein Startpunkt zusehen ist, da jeder Zustand der Startzustand sein kann. Anhand der Bewertung auf die Frage, wie man die Stärke empfunden hat, verändert sich der aktuelle Zustand der Stärke eines aktuellen Individuums um bis zu zwei Zustände. Dabei ist die Wertigkeit der Antwort in der Tabelle Tabelle 4.5 beschrieben. Die Bewertung +2 bedeutet, dass ein Zustandswechsel des aktuellen Zustands um zwei Zustände passiert. Um es an einem Beispiel zu erklären, nehmen wir an, dass der Startzustand q_0 ist. Dem Probanden würde jetzt ein Signal mit der Stärke 0x7FFF abgespielt. Wenn der Benutzer jetzt auf die Frage mit **zu schwach** beantwortet hätte, wechselt der Zustand jetzt zu q_2 . Wenn man immer noch von dem Zufälligen Startzustand von q_0 ausgeht und der Proband nach dem abspielen des Signals mit einer der Antworten **ok**, **stark** oder **zu stark** antworten würde, würde es keine Änderung des Zustandes geben, da es keine schwächere Stärke existiert. Die Ausgabe des Zustandsdiagramms ist die Signalstärke die in der Tabelle Tabelle 4.4 steht. Es wird zu jedem Zustand eine andere Signalstärke abgespielt. Diese Änderung erfolgt direkt nach der Beantwortung des Signals.

Anhand der Frage, was für ein Signal erkannt wurde, ist für das Programm irrelevant gewesen. Diese Information sollte für die Bewertung der Studie dienen.

Mittels der letzten Frage, wie gut ein Proband das Signal erkannt hat, hat man einen

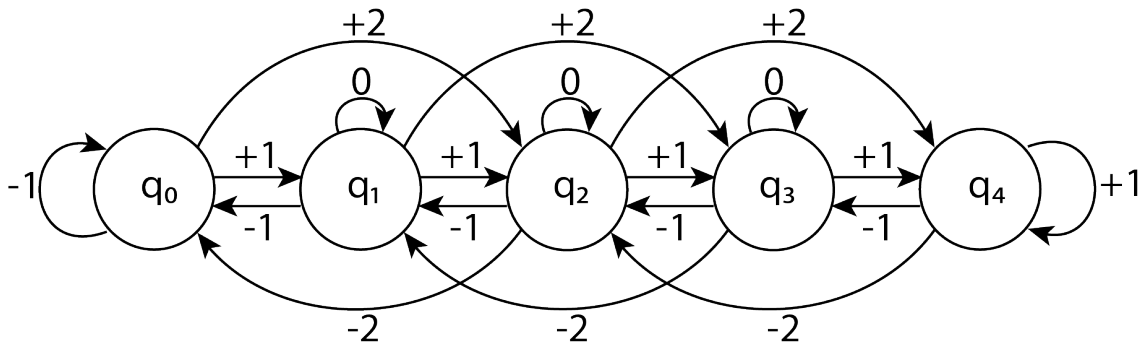


Abbildung 4.1: Zustandsdiagramm für die Stärke

Tabelle 4.5: Bewertung auf die Frage wie man die Stärke empfunden hat

Antwort	zu schwach	schwach	ok	stark	zu stark
Wertigkeit	+2	+1	0	-1	-2

Wert hinter jeder Antwortmöglichkeit gelegt. Die Antwort auf die Frage repräsentiert den Fitnesswert.

Erzeugung der nächsten Generation

Im Anschluss nach der Bewertung erfolgt die Bestimmung der nächsten Generation. Dies erfolgt mit der Selektion, Rekombination und Mutation der Gene.

Selektion

Für die Selektion wird für jeden Signaltyp ein neuer Pool erzeugt. Für jeden Signaltypen bestimmt man das Maximum $Fitness_{Max}$ von allen Fitnesswerten. Anhand der Formel

$$\text{Häufigkeit} = \frac{\text{Fitnesswert eines Individuums}}{Fitness_{Max} \text{ des Signaltyps}} \cdot 100 \quad (4.1)$$

lässt sich die Häufigkeit bestimmen, wie oft ein Individuum in den Pool hinzugefügt wird. Als Beispiel hat man ein Maximum von 5 und einen Fitnesswert von 2, daraus ergibt sich 40; das heißt, es wird 40 mal das jeweilige Individuum in den Pool hinzugefügt. Dabei wird jedes Individuum in das Pool hinzugefügt, welchem Signaltyp es angehört.

Nach der Erzeugung der Pools wird eine Rekombination durch zwei zufällig ausgewählte Individuen ausgeführt.

Rekombination

In der Rekombination wurden die zwei zufällig ausgewählten Individuen als Eltern definiert. Die gefundenen Eltern erzeugen ein neues Individuum, dass in die Population der nächsten Generation hinzugefügt wird.

Dieser Vorgang wurde wie folgt für das Problem angepasst. Man hat die zwei Gene, die Stärke und die Länge eines Signals.

Tabelle 4.6: My caption

Antwort	sehr schlecht	schlecht	ok	gut	sehr gut
Wertigkeit	1	2	3	4	5

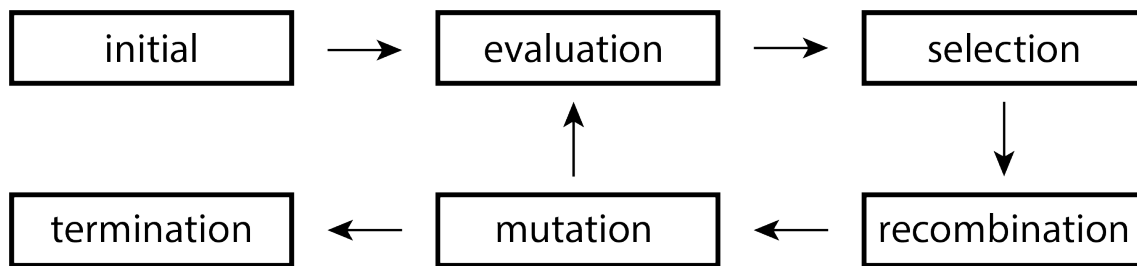


Abbildung 4.2: Ablauf des Algorithmus

Zuerst beginnen wir mit der Länge des Signals. Es wird die Länge des ersten Elternteils und die Länge des zweiten Elternteils genommen und ein Mittelwert von beiden Werten gebildet. Der Mittelwert ist die Länge des neuen Individuums. Bei der Stärke wurde direkt nach der Beantwortung des Signals die Stärke angepasst. Dieser Angepasste Stärkewert wird von beiden Elternteilen genommen und man bildet aus beiden ebenfalls den Mittelwert und rundet diesen ab.

Aus jedem der drei erzeugten Pools werden zehn neue Individuen für die nächste Generation erzeugt, so dass man wieder auf eine Gesamtzahl von 30 Individuen für eine Population kommt.

Mutation

Es besteht eine Chance von 5% dass sich ein Individuum mutieren kann. Das bedeutet in diesem Fall, dass sich lediglich die Länge zufällig verändern kann.

Abbruchkriterium

Man wiederholt für jede Generation wie den Zyklus ??, bis man eine ausreichende Lösung erhält.

Das Abbruchkriterium ist in diesem Fall erreicht, wenn die vierte Generation erzeugt worden ist. Um hier die Stellung zu der Entscheidung von vier Generationen zu nehmen, ist es so, dass man versucht habe die Anzahl der Vibrationen für einen Probanden niedrig zu belassen. Im Vergleich zum ersten Ansatz, in der man 600 Vibrationen für vier Generationen hat, habe man für die aktuell benutzte Lösung bei 30 Individuen pro Population, für 4 Generationen, 120 Vibrationen.

Nach dem man die letzte Population vom Algorithmus erhält, bestimmt man für jeden Signaltypen das Minimum und Maximum von der Stärke und der Länge von allen Signalen. Anhand derer man den Mittelwert bestimmt und schließlich den personalisierten Wert für den Probanden erhält.

4.3.1 Muster

Damit man schließlich die Hypothese der Bachelorarbeit beantworten kann, muss man zuvor die generische Vibrationen mit genetische Vibrationen vergleichen. Man hat sich im Vorfeld drei Typen von Mustern definiert. Der erste Typ war das dreier Muster. Dabei hat man drei Signale hintereinander erzeugt, die jeweils mit einer Pause getrennt sind. Für das dreier Muster hat man sich auf eine Anzahl von 15 Muster festgelegt. Diese 15 Muster wurden zwei mal erstellt, einmal mit den personalisierten Kurz, Mittel und Lang Werten, sowie die jeweiligen personalisierten Stärke Werten, die nach der Ausführung des Evolutionären Algorithmus bestimmt wurden. Das andere mal hat man sich anhand dem Paper [PBB16] für Kurz, Mittel und Lang jeweils die Werte 200ms, 400ms und 800ms,

sowie die Stärke für *Strong* entschieden. Beim vierer Muster hatte man vier Signale und drei Pausen und beim fünfer Muster hatte man fünf Signale und vier Pausen. Die vierer und fünfer Muster wurden genau wie das dreier Muster erstellt. Man hat also für jeden Mustertypen 30 Muster die auf jeweils 15 generischen und 15 genetischen Mustern bestehen.

Man hat sich als erstes überlegt gehabt für jeden Probanden zufällige Muster zu erstellen. Die Pausen der Muster habe man im Vorfeld fest definiert. Somit haben die dreier Muster entweder 100ms oder 200ms Pausen besessen. Die vierer Muster hatten Pausen, die entweder 100ms, 200ms oder 300ms gewesen sind. Das selbe gilt auch für die fünfer Muster, nur dass hier die Pausen wie beim vierer Muster und zusätzlich noch 400ms sein konnten. Die Pausen wurden für jeden Probanden einmalig festgelegt.

Damals war der Gedankengang so, dass bei den zufällig erzeugten Mustern man vermeiden wollte, dass Muster zwei mal abgespielt werden und aus dem Grund habe man sich die festen Pausen definiert, die für jeden Probanden gleich sind. Als man sich darüber Gedanken gemacht hat, wie man am Ende die Muster / Signale vergleichen wollen würde, hat man festgestellt, dass es nicht Sinnvoll sei zufällige Muster zu erzeugen, da man in der Auswertung keine Vergleiche untereinander machen könnte. Zu dem Zeitpunkt hatte man die festen Pausen schon implementiert und bei der Änderung nicht mehr im Blickwinkel gehabt. So hat man nach der Änderung für alle feste Muster mit festen Pausen, die jedoch zwischen den oben genannten Werten variieren. Dies ist eines der Fehlentscheidungen die ich erst nach der Durchführung der Studie erkannt hatte. Im Nachhinein würde ich für jeden Mustertypen immer gleichlange Pausen zwischen den Signalen definieren.

Während der Studie wurden zuerst alle dreier Muster, gefolgt von allen vierer Muster und schließlich allen fünfer Mustern abgespielt. Es wurde abwechselnd generische und genetische Muster abgespielt. Ein Muster wurde zufällig abgespielt und es wurde kein Muster zwei mal abgespielt. Die Probanden hatten die Aufgabe die Signale eines Muster in erkannter Reihenfolge anzugeben.

Man hat dem Probanden immer die Möglichkeit geboten, ein Signal oder Muster erneut abzuspielen.

4.4 Zusammenfassung

Am Ende sollten ggf. die wichtigsten Ergebnisse nochmal in *einem* kurzen Absatz zusammengefasst werden.

5. Implementierung

5.1 Signal

Das Kapitel **Implementierung** wird anhand dem im Entwurf besprochenen Verlauf der Studie erklärt.

Bei der Entwicklung des Programms hat man sich Gedanken über ein System gemacht, bei dem man alle Daten schon digital abgespeichert hat und nicht noch Informationen vom Probanden von Fragebögen per Hand in den PC eintragen muss. Aus diesem Grund hat man sich für eine Grafische Benutzeroberfläche, auch kurz GUI (graphical user interface) genannt, entschieden. Die Programmiersprache die verwendet wurde ist C# gewesen. Der Entstehungsprozess wurde auf Papier entworfen. Das Design sollte schlicht und einfach sein und den Fokus auf das Programm und nicht auf andere Komponenten des Betriebssystems ablenken. Die Papier Skizzen sind im Anhang vorhanden.

Informationen zur Person

The image shows two side-by-side wireframe sketches of a survey application's GUI. The left sketch, titled "Angaben zur Person", includes a text input field for "Alter", radio buttons for "Geschlecht" (Männlich, Weiblich, N.A.), and radio buttons for "Empfinden Sie sich als musikalisch?" (Ja, Nein). The right sketch, titled "Allgemeine Fragen", includes three questions with radio button options: "Spielen Sie gelegentlich Spiele?" (Ja, Nein), "Haben Sie Erfahrungen mit Taktilem Geräten?" (Ja, Nein), and "Haben Sie schon einmal eine Smart watch verwendet?" (Ja, Nein). Both sketches feature a "Bestätigen" button at the bottom right.

Abbildung 5.1: GUI für die Befragung zu den Angaben zur Person

Bei den Nachforschungen des Studiendesigns [BTT05], habe man sich nach mehreren Entwürfen für folgende Fragen entschieden:

- Wie alt sind Sie?
- Angabe des Geschlechts
- Empfinden Sie sich als musikalisch?

Bitte bewerten Sie das Signal, wie Sie es empfunden haben.

Kurz Mittel Lang

↶

Abbildung 5.2: GUI zur Bestimmung der Grenzen

- Spielen Sie gelegentlich Computer Spiele?
- Haben Sie Erfahrungen mit Taktilen Geräten?
- Haben Sie schon einmal eine Smartwatch verwendet?

Da man für den Probanden nicht mit allen Fragen auf einmal überhäufen wollte, hat man diese Fragen auf zwei Seiten dargestellt Abbildung 6.2. Bei der Frage um das Geschlecht habe man ebenfalls drei Antwortmöglichkeiten geboten, ob man *männlich*, *weiblich* oder *keine Angabe*. Im Hintergrund hatte man eine einfache Klasse **Person**, die die Informationen der Antworten dieser Befragung gespeichert hat. Um zu vermeiden, dass Daten verloren gehen, habe man die Daten zur Person schon nach diesem Schritt in einer externen Datei gespeichert. Für jeden Probanden hat man im Vorfeld eine anonymisierte Datei angelegt, die im Nachhinein nicht mehr auf den Probanden zurückschließen konnte.

Signal

Wie schon im Entwurf beschrieben, ist das Signal eines der wichtigsten Bestandteile des Programms.

Ein **Signal** besitzt die Attribute Länge in ms, Stärke, die den jeweiligen Zustand der Stärke speichert **??**, den Signaltypen, sowie die Grenzen des Signaltyps und die Anzahl wie oft ein Signal erneut abgespielt wurde. Da man in den nächsten beiden Phasen dem Signal einen Signaltypen zuordnen soll, wurde dafür auch ein weiteres Attribut angelegt, sowie die Zeit, die benötigt wurde um die Frage zu beantworten.

In der Phase indem der Algorithmus durchgeführt wird, werden noch zwei zusätzliche Fragen gestellt für die man in der Klasse **Signal** auch noch diese beiden Werte gespeichert hat.

Bestimmung der Grenzen durch Bewertung durch den Nutzer

Nachdem der Proband die Angaben zur Person beantwortet hat, erscheint ein Benachrichtigungsfenster, indem Informationen über den die aktuelle Phase erklärt wird. Falls

der Proband zu einem Zeitpunkt Fragen haben sollte, werden diese durch den Aufseher beantwortet. Für die Bestimmung der Grenzen wurden 10 Signale erzeugt, die gleichverteilt sind. Mithilfe einer Zufallsfunktion werden diese Signale zufällig ausgewählt und in dieser Reihenfolge abgespielt. Der Proband soll für jeden der 10 Signalen einen der Signaltypen *Kurz*, *Mittel* oder *Lang* zuordnen. Dabei habe man die im Entwurf besprochenen Sonderfälle überprüft und bei erfolgreicher Bildung der Grenzen ist man in die nächste Phase übergegangen.

Wenn die Bestimmung der Grenzen nicht erfolgreich gewesen ist, wird der Benutzer gebeten die gleiche zufällige Reihenfolge erneut zu bewerten.

Sollte ein Proband den Replay-Button drücken, wird das letzte Signal erneut abgespielt.

Für die GUI ?? hat man sich auf das wesentliche beschränkt. Die Anweisung an den Benutzer war immer die selbe, daher hat man nur den gleichen Text dargestellt. Als Antwortmöglichkeiten des Signals wurde für jeden Signaltyp je ein Radio-Button erstellt. In den Phasen, in denen ein Signal abgespielt wurde, wurde der Replay-Button immer an der gleichen feste Position plziert.

Der Ablauf einer Bewertung wird im folgenden beschrieben. Nach der Bestätigung des Benachrichtigungsfenster wurden im Hintergrund die 10 Signale in zufälliger Reihenfolge ausgewählt. Darauf wurde das erste Signal über BLE an das Wearable gesendet. Das Wearable hat dieses Signal empfangen, ausgewertet und hat die Informationen als Vibration abgespielt. Das was der Benutzer dadurch empfunden hat wurde über den PC mittels der Maus bewertet. Nach einer Bewertung gab es keine Möglichkeit die Bewertung eines vorherigen Signals erneut anzupassen. Nachdem das Signal vom Probanden bewertet wurde, wurde der Maus-Cursor durch dem Programm an eine Anfangsposition gesetzt, damit der Benutzer nicht unabsichtlich doppelt auf ein Radio-Button drücken konnte. Um dem Probanden zu signalisieren, dass er die Checkbox gedrückt hat, hat man die Check-Box einen Augenblick visuell markiert gelassen, nachdem man den Maus-Cursor neu positioniert hat. Mit der Neupositionierung der Maus hat man erreicht, dass man immer den gleichen Weg hatte um die Checkboxes zu erreichen, aber man hatte auch nach mehreren Fragen die Maus anheben weiter nach oben legen müssen, denn man hat schon das Ende der Tischkannte erreicht. Nach einer Bewertung wurde die vorherige Auswahl entfernt und man hat das nächste Signal übertragen.

3. Ausführen des Algorithmus

Die Abbildung zeigt zwei GUI-Screenshots. Der linke Screenshot ist eine Bewertungsförme mit dem Titel 'Bitte bewerten Sie das Signal.' und drei Fragen: 'Was für einen Signaltypen haben Sie erkannt?' (mit Optionen kurz, mittel, lang), 'Wie gut haben Sie das Signal erkannt?' (mit Optionen sehr schlecht, , sehr gut) und 'Wie empfanden Sie die Stärke des Signals?' (mit Optionen zu schwach, , zu stark). Ein Replay-Button (Pfeil) befindet sich unten rechts. Der rechte Screenshot zeigt die Stimmungsbewertung mit dem Titel 'Stimmung Bewerten' und der Frage 'Bitte bewerten Sie Ihre Stimmung'. Darunter befinden sich fünf Emojis, die von traurig bis glücklich reichen, jeweils mit einem Radio-Button.

Abbildung 5.3: GUI des Algorithmus (links) und der Bewertung der Stimmung (rechts)

Für den Algorithmus habe man sich mehrere Klassen erzeugt. Die Klasse **DNA** beinhaltet ein Signal und Hilfsfunktionen, die eine Rekombination und eine Mutation von zwei Signalen erzeugen. In der Klasse **Population** hat man eine Liste aus 30 Elementen, von einer Klasse **DNA** erstellt, dabei sind beinhalten die ersten 10 DNA-Objekte Signale, die

den Signaltypen *Kurz* haben. Für die nachfolgenden DNA-Objekte gilt das selbe nur mit *Mittel* gefolgt von *Lang*. Bei der Erzeugung der Signale habe man mittels einer Zufallsfunktion sich Signale mit einer zufälligen Stärke und einer zufälligen Länge zwischen den Grenzen des jeweiligen Signaltyps erstellt. Beim ersten Aufruf der Klasse *Population* habe man die Grenzen der Signaltypen ebenfalls als ein Signal erzeugt und in der Liste hinzugefügt.

Für die Grafische Benutzeroberfläche hat man sich für die drei folgenden Fragen entschieden:

- Was für einen Signaltypen haben Sie erkannt?
- Wie gut haben Sie das Signal erkannt?
- Wie empfanden Sie die Stärke des Signals?

Es ist bis auf die Anzahl der Fragen genau das gleiche Prozedere gewesen. Es wurde zuerst ein Erklärungsfenster dargestellt, so dass der Benutzer weiß, was in dieser Phase von Ihm erwartet wird. Dabei wird im Hintergrund wie gerade erwähnt eine *Population* erzeugt. Diese *Population* wird DNA für DNA abgearbeitet. Dabei wird für jede DNA enthaltene Signal über BLE an das Wearable gesendet und die Information in Vibration umgewandelt. Dabei hat der Benutzer die drei Fragen zu dem Signal beantworten sollen. Es musste die komplette *Population* bewertet werden, es wurde keine DNA erneut bewertet. Man habe dabei zufällig eine DNA aus der *Population* gewählt, die bewertet werden sollte.

Die erste Frage diene nur dazu um in der späteren Auswertung herauszufinden, was der Benutzer im Vergleich zum Programm erkannt hat. Die zweite Frage wurde als eigentlicher Fitnesswert gespeichert. Die Repräsentation der Frage ist in der Tabelle ?? zu sehen. Die Stärke wurde direkt nach der Beantwortung aller drei Fragen nach dem Zustandsdiagramm verändert.

Die Informationen der Stärke und des Fitnesswerts habe man in der Klasse *Signal* noch mitgespiechert.

Nachdem eine *Population* komplett berechnet wurde, habe man den Generations-zähler um eins erhöht, die bisherigen Informationen in der externen Datei gespeichert und eine neue *Population* aus den Werten erzeugt.

Selektion

Für jeden Signaltypen hat man einen Pool aus DNA-Objekten erstellt, dessen Signale den jeweiligen Signaltypen besitzen. Anhand der Formel X (im Entwurf) habe man eine Anzahl berechnet, wie oft ein DNA-Objekt in das jeweilige Pool hinzugefügt wird. Im Schnitt hatte man nach dem Hinzugügen aller DNA-Objekte knapp 1000 Elemente in einem der Pools.

Rekombination

Bei der Rekombination verwendet man

4. Muster Erkennung

Ein Signal beinhaltet die Signallänge, die in Millisekunden gespeichert wird, und eine Signalstärke, die in 5 Stärkestufen eingeteilt ist.

Ich habe meinen Evolutionären Algorithmus so angepasst, dass bei mir ein Individuum ein Signal ist. Ich habe dem Benutzer das Signal mit dem Wearable abspielen lassen und im Anschluss Fragen beantworten lassen. Er sollte bewerten wie gut er das Signal erkannt hat. Die Bewertung vom Benutzer war entscheidend um nach der kompletten Bewertung der *Population*

5.2 Muster

5.3 Evolutionärer Algorithmus

5.4 DNA

5.5 Population

5.6 Generationen

5.7 Mutation

6. Evaluierung

Hier kommt der Nachweis, dass das in Kapitel 4 entworfene Konzept auch funktioniert. Leistungsmessungen einer Implementierung werden auch immer gerne gesehen.

6.1 Studiendesign

6.1.1 Studiendesign

Im folgenden Programmablauf Abbildung 6.1 hat man sich orientiert um die Studie zu entwerfen. Über Mailinglisten und Bekanntenkreis haben sich 32 Probanden bereiterklärt an der Studie teilzunehmen. Dabei waren 72 Prozent Männer und 28 Prozent Frauen. Das Alter der Probanden war zwischen 12 bis 54 Jahren vertreten und das Durchschnittsalter war 22 Jahre Abbildung 6.2. Die Studie hat zwischen 30 Minuten und einer Stunde gedauert. Die Studien wurden an drei verschiedenen Orten durchgeführt, im TECO in Karlsruhe, in einem Seminarraum an der Hochschule Darmstadt und in einem Arbeitszimmer in Meschede.

Für jeden teilgenommenen Probanden ist der gleiche Ablauf durchgeführt. Vor der Studie wurde ein Termin mit dem Probanden vereinbart. Nachdem der Proband zur abgemachten Zeit am vorgegebenen Ort angekommen ist, wurde Ihm erklärt wofür die Studie ist, was man mit der Studie herausfinden will und welche Erwartungen man an den Probanden hat. Nachdem der Proband alles verstanden hat und die Einverständniserklärung verstanden und unterschrieben hat, wurde Ihm das Armband angezogen. Dabei haben die Rechtshänder das Armband an dem linken Arm und die Linkshänder an dem rechten Arm angelegt.

Da man für die Studie ein Programm mit einer Grafischen Oberfläche (GUI) entworfen hatte, wurde der Proband mithilfe des Programms durch die Studie geführt. Dabei wurden dem Probanden ein paar Personalien abgefragt, wie das Alter, das Geschlecht, ob sich die Person als Musikalisch empfindet, ob man Computerspiele spielen würde, ob man schon einmal eine Smartwatch benutzt habe und ob die Person schon mal ein Tactiles Gerät benutzt habe. Falls vom Probanden Fragen während der Studie Fragen aufgekommen sind wurden diese sofort beantwortet.

Nach der Aufnahme der Personalien, wurde dem Benutzer erklärt, was Ihn als nächstes erwartet und von Ihm verlangt wird. Man hat dem Nutzer im ersten Schritt 10 Signale abgespielt, um Ihn ein Gefühl für Signale zu geben. Im Anschluss wurde dem Probanden jedes Signal erneut einmalig abgespielt. Dabei sollte er das Signal zu drei jeweiligen Kategorien zuordnen. Diese Kategorien waren ob es ein Kurzes, Mittleres oder Langes Signal

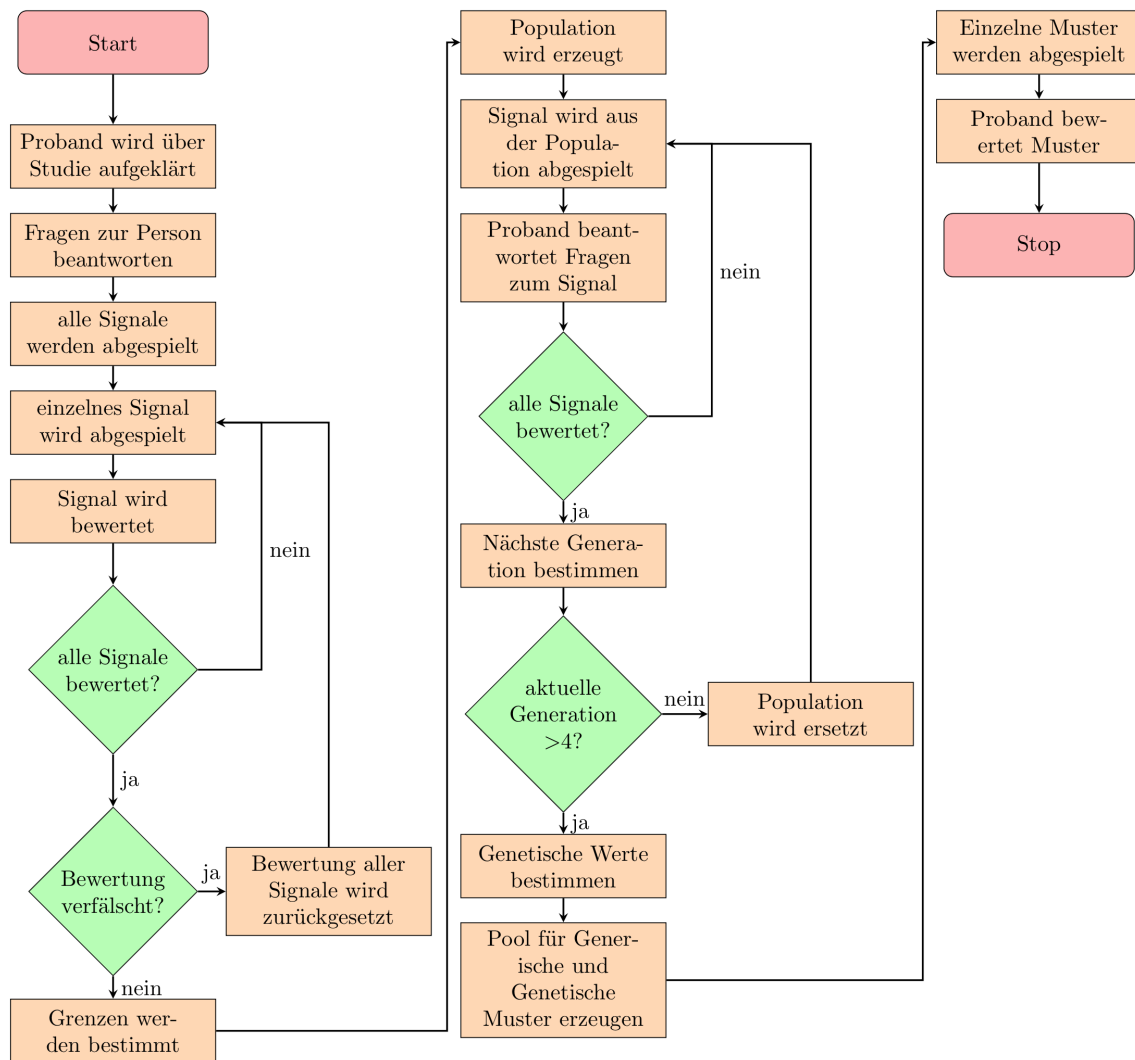


Abbildung 6.1: Studienablauf

für ihn gewesen ist. Dieser Schritt war dafür notwendig um für den Benutzer die Grenzen für die jeweilige Kategorien Kurz, Mittel und Lang zu bestimmen. Diese Grenzen sind für die Initialisierung des Algorithmus notwendig gewesen.

Als die 10 Signale bewertet wurden, wurde der Benutzer darüber aufgeklärt, was ihm als nächsten Schritt erwartet. Es wurde ihm anhand seiner Eingaben ein zufälliges Signal abgespielt, das er bewerten sollte (BILD). Anhand der drei Fragen wurde das Signal bewertet. Um eine Iteration komplett zu bewerten wurde dieser Vorgang 30 mal wiederholt. Im Anschluss wurde gefragt wie der Benutzer sich derzeit fühlt (BILD). Anhand einer komplett bewerteten Iteration wurde dem Benutzer neue Werte berechnet. Es wurden insgesamt vier Iterationen durchgeführt um einen möglichst genauen Wert für den Benutzer zu bestimmen.

Im letzten Schritt wurde dem Benutzer aufgeklärt, dass ab dem Zeitpunkt nur noch Folgen von Signalen, die man ab jetzt Muster nennt, abgespielt werden. Die Probanden sollten angeben in welcher Reihenfolge was für Signaltypen abgespielt wurden. Es wurde für alle Probanden im Vorfeld alle Muster definiert, damit jeder die gleichen Muster abgespielt bekommt. Es gab zwei Arten von Muster, die generischen Muster und die genetischen Muster. Der einzige Unterschied zwischen den beiden Arten waren die Werte, die die Signale in einem Muster zugewiesen wurden. Das bedeutet es wurden zwei mal das selbe Muster abgespielt mit lediglich anderen Werten. Die Genetischen Muster hatten die Werte, die nach dem Algorithmus erzeugt wurden übernommen, wobei die generischen Muster einen vordefinierten Wert zugewiesen bekommen hat. Der generische Wert ist für jeden Probanden gleich gewesen und hat man sich wie im Entwurf schon beschrieben am Paper [PBB16] orientiert. Dabei gab es Muster mit drei, vier und fünf Signalen. Nacheinander wurde dem Nutzer zuerst alle Muster mit drei Signalen. Dabei wurde das genetische Muster abwechselnd zum generischen Muster abgespielt.

Nachdem alle Muster von dem Probanden bewertet wurden, haben Sie ihre e-Mail noch angegeben um an einer automatischen Verlosung von zwei Gutscheinen teilzunehmen. Bei Interesse wurde ihnen ihre Werte gezeigt und erklärt, was genau im Hintergrund passiert worden ist. Während der ganzen Studie standen dem Probanden ausreichend Süßigkeiten zur Verfügung, bei denen Sie sich frei bedienen konnten.

6.2 Analyse

6.2.1 Angaben der Probanden

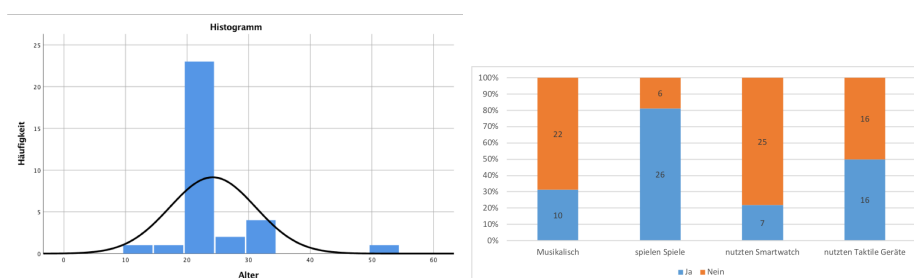


Abbildung 6.2: Alter aller Probanden (links) und Auswertung der Fragen (rechts)

Nach der Befragung der Probanden hat sich ergeben, dass 31% musikalisch sind, 81% gelegentlich Computerspiele spielen. 21% haben schon mal eine Smartwatch benutzt und genau die Hälfte haben schon mal taktile Geräte benutzt.

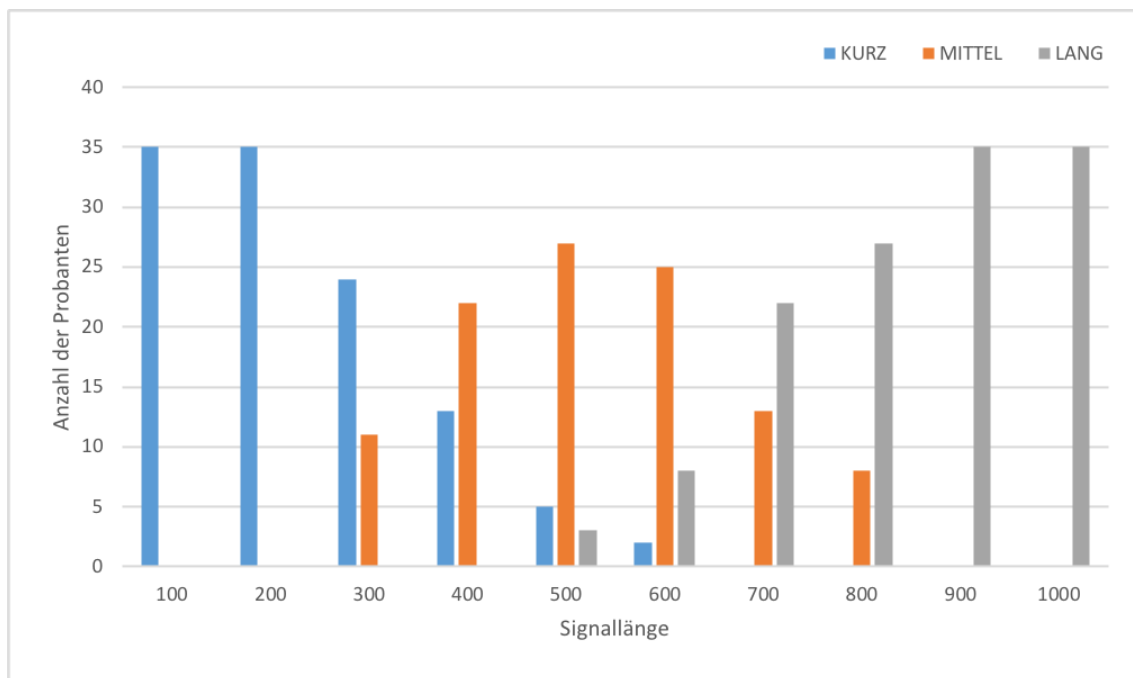


Abbildung 6.3:

6.2.2 Initialisierung der Grenzen

Für die Bestimmung der Grenzen von Kurz, Mittel und Lang hat der Benutzer 10 Signale zu der jeweiligen Kategorie einordnen sollen. Da man bei 16% der Probanden bei diesem Schritt keine eindeutigen Grenzen bestimmen konnte, musste der Schritt ein weiteres mal wiederholt werden.

Obwohl die 10 normalverteilten Signale in zufälliger Reihenfolge abgespielt wurden, wurde bei 84% aller Probanden die zehn Signale so bewertet, dass zuerst alle Kurz, gefolgt von nur Mittel und anschließend nur Lang bewertet wurden. Die restlichen Probanden haben an den Grenzen nur ein Signal von Kurz und Mittel oder Mittel und Lang vertauscht gehabt.

In der Auswertung?? haben sich drei Normalverteilungen gebildet. Daraus kann man entnehmen, dass Werte 100ms und 200 ms eindeutig als Kurz, sowie auch Werte 900ms und 1000ms als Lang erkannt worden sind. Einen eindeutigen Hochpunkt für Mittel hat man bei dem Wert von 500ms, dennoch gab es einige Probanden für die dieser Wert noch als Kurz oder Lang empfunden wurde.

Aus dem Diagramm kann man sehr schön erkennen, dass man eine Personalisierung von Vibrationen benötigen könnte, da außer den Rändern keine Eindeutige Zuweisung von Kurz, Mittel oder Lang passieren konnte. Man kann zwar im Vorfeld Werte für Signale definieren, wie es derzeit einige Firmen machen, diese Werte könnten jedoch von Personen unterschiedlich wahrgenommen werden.

6.2.3 Auswertung der Iterationen des Algorithmus

Verlauf der Grenzen

Der Algorithmus wurde insgesamt vier mal ausgeführt, anschließend wurde aus der letzten Population der personalisierte Wert gemittelt.

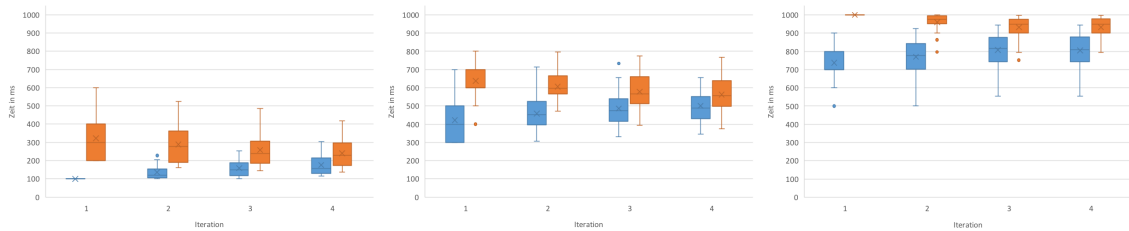


Abbildung 6.4: Darstellung des Minimum (in blau) und Maximum (in rot) im Verlauf der 4 Iterationen des Algorithmus. Das erste Bild zeigt Kurz, gefolgt von Mittel und Lang.

Kurz

Eine der größten Veränderungen im Diagramm ?? gibt es bei **Kurz**; die oberste Grenze vom Maximum ist nach vier Iterationen von 600ms auf 415ms gesunken. Die obere Grenze des Minimum ist von 100ms auf 300ms gestiegen. Der Bereich verkleinert sich insgesamt vom Intervall von 100 bis 600ms auf 140 bis 420 ms. Der Median von den Maximum und Minimum konvergieren in die gleiche Richtung. Nach vier Iterationen ist der Median vom Minimum um 80ms nach oben und der Median von dem Maximum auch um 80ms nach unten gewandert. Das Minimum und Maximum überschneidet sich in der 2. Iteration nur minimal. In der 4. Iteration sich Minimum und Maximum um 150ms überschneiden. Das 50% Quartil vergrößert sich auf ein Intervall bis zu 100ms, wobei das 50% Quartil von einem 20ms Intervall auf ein 120ms Intervall verdichtet.

Mittel

In der ersten Iteration besitzt man schon Überschneidungen des Minimum und Maximum. Von der ersten zur zweiten Iteration verschieben sich die Grenzen des Minimum um ca. 10 ms nach oben, diese werden im Verlauf zur dritten und vierten Iteration kleiner. Die Grenzen des Maximums hingegen werden größer und verschieben sich anschließend minimal nach unten. Das 50% Quartil verändert sich beim Minimum von einem 200ms Intervall auf ein 120ms Intervall und verschiebt sich anschließend im Verlauf der folgenden Iterationen nur nach oben.

Lang

Die meisten Ausreißer sind in bei Lang vertreten, hier hat man in der zweiten Iteration schon ein Maximum von 1000ms auf unter 800ms und 880ms erhalten, das ist eine Änderung eines Maximums um 200ms nach nur einer Iteration bei einem Probanden. Die 50% Quartile des Maximums werden Minimal größer und wachsen nach unten. Bei den 50% Quartile des Minimums werden die kurzzeitig größer ändern sich aber wieder auf mit dem Ausreißer der ersten Iteration auf das gleiche Intervall und verschieben sich pro Iteration leicht nach oben.

Unter den Signaltypen selbst überschneiden sich die Iterationen die Grenzen selbst zwischen den Iterationen selbst. Diese Überschneidungen verringern sich nach den 4 Iterationen.

Das Minimum steigt stetig und das Maximum verringert sich. Denn genau diesen Effekt wollte man auch erzielen, dass der Benutzer zu seinen personalisierten Wert konvergiert. Bei Kurz und Mittel überschneiden sich die 50% Quartile. Bei Lang überschneiden diese in keiner Iteration.

Schlussfolgernd kann man sagen, dass die man anhand der 50% Quartile vom Maximum und Minimum aller Probanden ein Kurzes Signal in dem Bereich von 130ms bis 300ms,

ein Mittleres Signal zwischen 430ms und 640ms und ein Langes Signal zwischen 740ms und 980ms die gebildet haben.

Grenzen nach dem Algorithmus

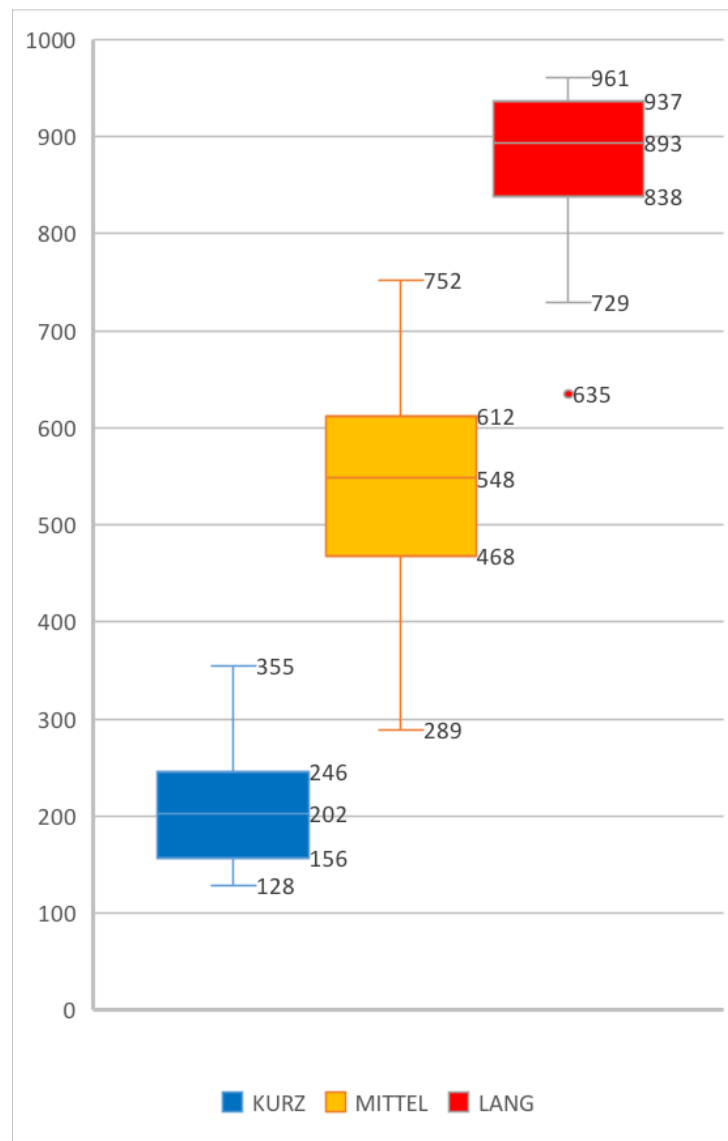


Abbildung 6.5: Nach der Vierten Iteration gemittelte personalisierte Werte in ms

In dem Diagramm ?? sind alle gemittelten personalisierten Werte nach der vierten Iteration abgebildet. Daraus lässt sich ableiten, dass zwischen den Kategorien von Kurz, Mittel und Lang sich die Grenzen überschneiden. Für die Einzelnen Kategorien hat sich der Median für Kurz bei 202ms, für Mittel bei 548ms und Lang bei 893ms festgelegt. Die oberen 25% bei Lang und die unteren 25% von Kurz weisen nur eine minimale Abweichung auf. Im Gegensatz dazu gibt es eine Abweichung von genau 109ms bei den oberen 25% von Kurz und den unteren 25% von Lang. Die Grenzen überschneiden sich von Kurz und Mittel zwischen 289ms bis 355ms und von Mittel und Lang zwischen 729ms und 752ms. Es gibt einen Ausreißer bei Lang mit einem Wert von 635ms, dabei handelt es sich nicht um keinen Fehler, sondern einen Probanden der genau dies für sich als Personalisierten Wert bestimmt hat. Im Vergleich der inneren 50% von den jeweiligen Kategorien weißt Kurz einen Intervall von 90 ms und Lang einen Intervall von 99ms auf, wobei Mittel diesen Wert von 144 ms am größten Bereich ist und um nahezu 50ms größer als Kurz und Lang ist.

Stimmung im Verlauf des Algorithmus

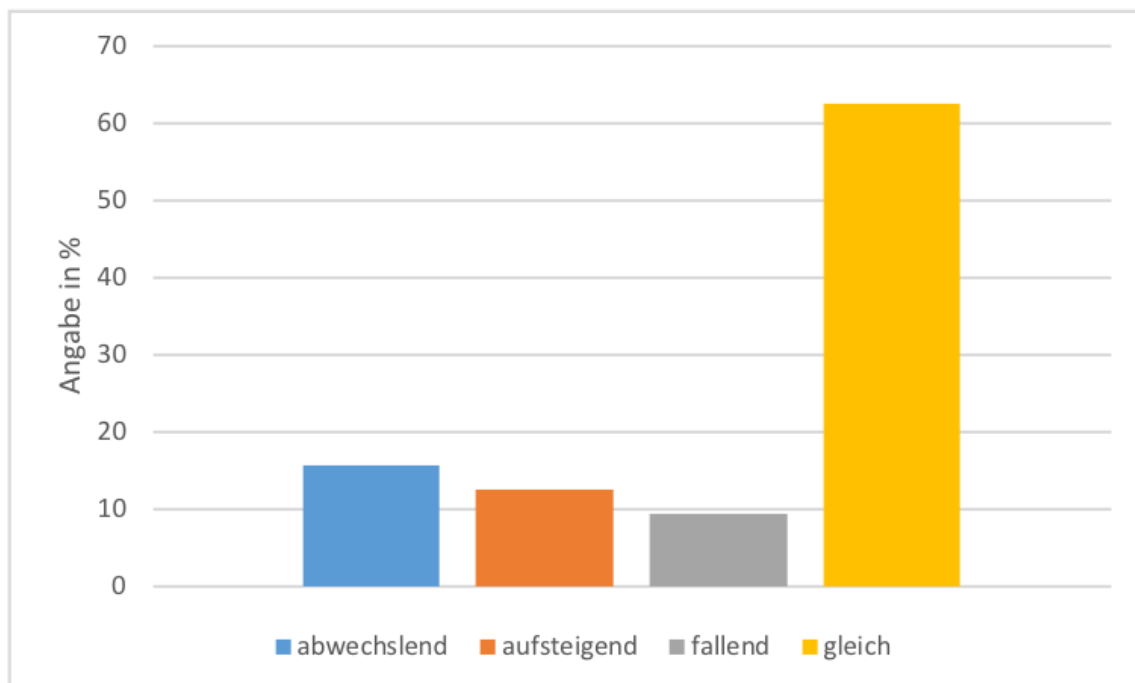


Abbildung 6.6: Bewertung der Stimmung

Nach jeder Iteration wurden die Benutzer gefragt wie Ihre Stimmung ?? aktuell sei. Dabei hat es bei 62% aller Probanden keine Änderung der Stimmung gegeben. Diese Probanden hatten Ihre Stimmung mit **Gut** oder **Sehr gut** bewertet gehabt. Im Gegensatz dazu hat sich die Stimmung bei 16% der Probanden entweder zwischen **Gut** und **Sehr gut** oder **Gut** und **OK** abgewechselt. Des Weiteren gab es bei 13% der Probanden eine aufsteigende Stimmung und bei den restlichen 9% ist die Stimmung pro Iteration gefallen.

Somit kann man feststellen, dass die Anzahl von vier Iterationen keinen wirklichen Einfluss auf die Stimmung des Benutzers ausgewirkt hat.

Replays eines Signals

Man hat dem Probanden die Möglichkeit geboten, ein Signal wiederholen zu dürfen. Im Verlauf des Algorithmus wollte man herausfinden wie oft unter allen Probanden auf Replay für ein bestimmten Intervall Replay gedrückt wurde ?. Die Anzahl der wie oft auf Replay gedrückt wurde

In der ersten Iteration haben 44% der Probanden in dem Intervall von 700-799ms die größten Schwierigkeiten gehabt ein Signal zuzuordnen und sich das Signal erneut abgespielt. Der Bereich, der am wenigsten wiederholt abgespielt werden musste war 200-299ms. Die Restlichen Intervalle der ersten Iteration wurden mit einem Mittelwert von 32% erneut abgespielt.

In der zweiten Iteration hat sich die Anzahl der Replays um einiges geringer hier war der Mittelwert insgesamt bei 22% mit einem Minimum bei 200-299ms von 15% und einem Maximum bei 500-599ms mit 27%. Die dritte Iteration hat eine Verbesserung bei allen nahezu allen Werten gehabt, bis auf das Intervall von 700-799ms was auf 35% angestiegen ist; insgesamt ist der Mittelwert in dieser Iteration bei 19%.

Selbst in der letzten Iteration hat sich ein globales Minimum bei 800-899ms von 9,5% gebildet. Der globale Maximum von 700-799 in der ersten Iteration mit 43% ist auch in der vierten Iteration, trotz Verbesserungen zwischen durch, auch das Maximum mit 42,5%.

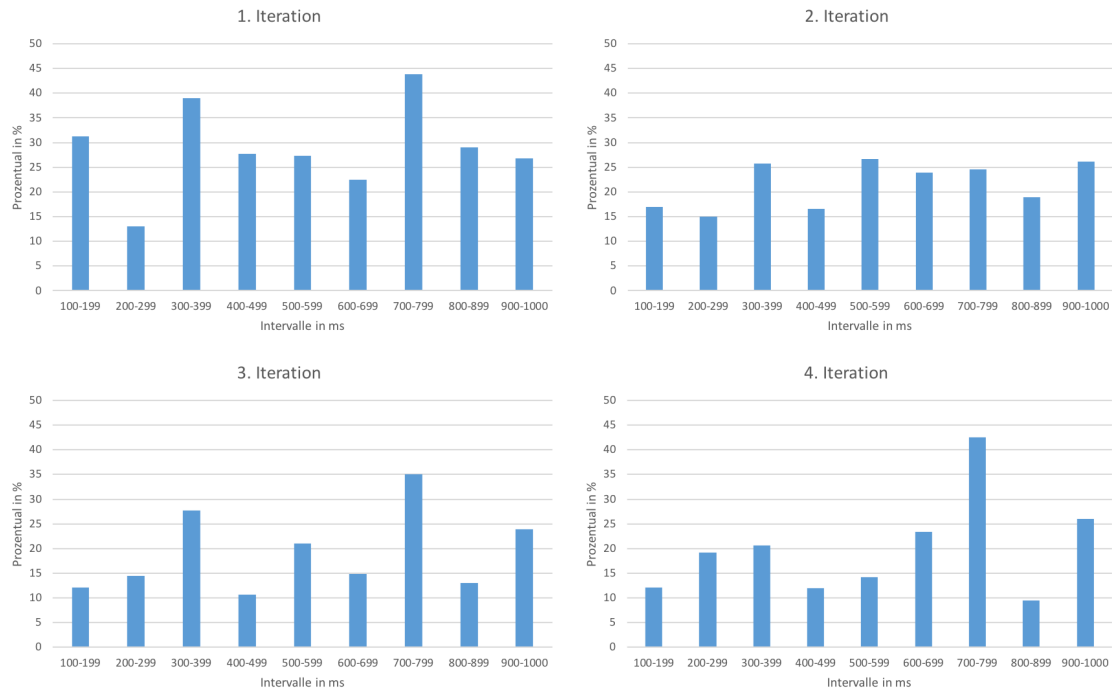


Abbildung 6.7: Anzahl der Replays innerhalb von Zeitintervallen über dem Verlauf des Algorithmus

Man kann daraus erkennen, dass die Probanden genau an dem Bereich von Kurz zu Mittel, was sich in der 4. Iteration bei 200-399ms bewegt, und Mittel zu Lang, dass zwischen 600-799ms liegt, am meisten den Replay Knopf gedrückt haben. Ein viertel aller Probanden haben bei 900-1000ms in jeder Iteration das Signal erneut abgespielt. Jedoch kann man in der vierten Iteration sehr gut erkennen, dass die Signale mit den Werten 100-199ms, 400-599ms und 800-899ms am geringsten erneut abgespielt werden mussten. Im Schnitt wurde, falls man ein Signal erneut abgespielt haben wollte, 1,7mal der Replay Button gedrückt, bei einem Minimum von einmal und einem Maximum von 9 mal, im Verlauf des Algorithmus.

6.3 Abschnitt 2

6.4 Zusammenfassung

Am Ende sollten ggf. die wichtigsten Ergebnisse nochmal in *einem* kurzen Absatz zusammengefasst werden.

7. Zusammenfassung und Ausblick

Bla fasel...

(Keine Untergliederung mehr!)

8. Zusammenfassung und Ausblick

Bla fasel...

(Keine Untergliederung mehr!)

GLOSSAR

Force Feedback:

Literaturverzeichnis

- [app18] *Apple Watch Settings*, 2018 (accessed May 1, 2018). <https://support.apple.com/de-de/HT204793>.
- [bro5] *Brockhaus-Enzyklopädie online*, 2005-. <http://gso.gbv.de/DB=2.2/PPNSET?PPN=517415410>, Druckausgabe u.d.T.: Brockhaus. Enzyklopädie in 30 Bänden (2006).
- [BTT05] David Benyon, Phil Turner und Susan Turner: *Designing interactive systems: People, activities, contexts, technologies*. Pearson Education, 2005.
- [Fli] Patrick Flick: *Evolutionäre Algorithmen*.
- [FSDS16] Christopher B Fleizach, Eric Taylor Seymour und Joel M Lopes Da Silva: *Custom vibration patterns*, jul # -5" 2016. US Patent 9,383,820.
- [GOS01] Francine Gempeler, Nathan Ota und Dan Siewiorek: *Design of a wearable tactile display*. In: *Wearable Computers, 2001. Proceedings. Fifth International Symposium on*, Seiten 5–12. IEEE, 2001.
- [PB03] Peter Parente und Gary Bishop: *BATS: the blind audio tactile mapping system*. In: *Proceedings of the ACM Southeast regional conference*, Seiten 132–137. Citeseer, 2003.
- [PBB16] Erik Pescara, Michael Beigl und Matthias Budde: *RüttelFlug: a wrist-worn sensing device for tactile vertical velocity perception in 3d-space*. In: *Proceedings of the 2016 ACM International Symposium on Wearable Computers*, Seiten 172–175. ACM, 2016.
- [Sel03] Bianca Selzam: *Genetische Algorithmen*. Dortmund: TU Dortmund, 2003.
- [Shi12] Daniel Shiffman: *The Nature of Code: Simulating Natural Systems with Processing*. Daniel Shiffman, 2012.
- [Wei15] Karsten Weicker: *Evolutionäre Algorithmen*. Springer-Verlag, 2015.

Erklärung

Ich versichere, dass ich die Arbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, die wörtlich oder inhaltlich übernommenen Stellen als solche kenntlich gemacht und die Satzung des KIT zur Sicherung guter wissenschaftlicher Praxis in der jeweils gültigen Fassung beachtet habe. Die Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt und von dieser als Teil einer Prüfungsleistung angenommen.

Karlsruhe, den 22. Mai 2018

(thomas)