

Nationaler Wettbewerb SJf 2021

Virtuelles Laufen in anderen Welten

Entwicklung einer neuartigen VR-Treadmill im Kampf gegen Motion-Sickness



Autor

Adresse

Marvin Hotz

Mövenstrasse 9, 8645 Jona

20.03.2021

Inhalt

1	Vorwort	6
2	Einleitung	7
3	Theorie und Recherche.....	8
3.1	Immersion und VR Systeme.....	8
3.1.1	Problematik beim Bewegen in immersiven Räumen.....	9
3.2	Entstehung von Motion Sickness	10
3.3	Methoden um Motion Sickness zu verhindern	10
3.4	Haptisches Feedback	11
3.5	Verschiedene Achsen	12
3.6	Lauf- / Gang-Analyse	12
3.7	ODrive.....	13
3.8	Bestehende Treadmill-Systeme	14
3.8.1	Treadmillsysteme mit Rutschebenen.....	14
3.8.2	Sitzende Treadmillsysteme mit Rollen.....	15
3.8.3	Treadmillsysteme mit Laufbändern.....	15
4	Konzept	16
4.1	Bewegungskonzept.....	16
4.2	Linearachsen (Schwarz)	18
4.3	Riemensystem (Blau)	18
4.4	Erfassungssystem (Blau).....	18
4.5	Haptik System	18
4.6	Gelenkarm (Z-Achse & Pitch) (Rot)	18
4.7	Fusshalterung und Achse (Yaw) (Gelb)	19
4.8	Fussplatten «Virtuelle Bodenplatten» (Grün).....	19
5	Umsetzung.....	20
5.1	3D Teile CAD und 3D Druck.....	20
5.2	Linearschienen (X-Achse).....	22
5.3	Riemensystem (X-Achse).....	24
5.4	Erfassungssystem (X-Achse).....	25
5.5	Force Control (X-Achse).....	26
5.6	Gelenkarm (Z-Achse & Pitch).....	27
5.7	Fusshalterung und Achse (Yaw)	28
5.8	Fussplatten & Erkennung	29
5.9	Elektronik Hardware & Software.....	30
5.9.1	Microcontroller / Arduino / IDE.....	30
5.9.2	Aufbau	31
5.9.3	Kommunikation / Schnittstellen	32

5.9.4	Immersive Umgebung / Unreal Engine	33
5.9.5	Programmablauf	34
6	Resultat	35
6.1	Stand der Entwicklung	35
6.2	Ergebnisse von Tests mit Versuchspersonen.....	35
6.3	Technische Optimierungen für Weiterentwicklung	36
6.3.1	Drehbewegung.....	36
6.3.2	Lauf Unterstützung.....	36
6.3.3	Trägheit der Achsen	36
6.3.4	Referenz System.....	36
6.3.5	Simulierung der Füße in VR	36
6.4	Technische Erweiterungen für Weiterentwicklung	37
6.4.1	Simulieren von virtuellen Objekten.....	37
6.4.2	Simulieren der Physik eines Bodens	37
6.5	Konklusion	38
7	Fazit.....	39
7.1	Zielüberprüfung.....	39
7.2	Verbesserungspotential.....	40
7.3	Organisieren der Teile	40
7.4	Sicherheit.....	40
7.5	Zufriedenheit	40
7.6	Ausblick.....	41
8	Danksagung.....	41
9	Eigenständigkeitserklärung	41
10	Verzeichnisse.....	42
10.1	Abbildungsverzeichnis	42
10.2	Literaturverzeichnis.....	43
11	Anhang	44
11.1	Versuchsergebnisse mit Probanden	45
11.2	Teensy Code als Hauptkontroller	48
11.3	Arduino Nano Code für Fusschlitten	60
11.4	Unreal Engine Pawn Code.....	63
11.5	Teensy 3.2	64
11.6	Arduino Nano	65
11.7	Verbaute Encoder (CUI AMT102-V)	66

Abstract

- Hintergrund** Virtual Reality (VR) Systeme werden immer beliebter. Setzt man eine VR-Brille auf, verlässt man die Realität und kann die endlosen Weiten von VR erkunden. Heutige Systeme für Privatanwender bestehen meist aus einer VR-Brille und VR-Motion-Controllern. Diese werden zum einfachen Navigieren und Interagieren mit der virtuellen Umgebung gebraucht. Diese Arbeit beschäftigt sich mit der Verbesserung des Komforts in VR und dem Ausbau der Interaktionsmöglichkeiten.
- Problem** Obwohl heutige VR-Systeme ein gutes Eintauchen in virtuelle Umgebungen ermöglichen, verursachen realitätsfremde Bewegungen, die vor allem beim Navigieren auftreten, bei vielen Benutzern Motion-Sickness. Dies macht VR für viele Personen ungeeignet, da ihnen nach kurzer Zeit übel oder schwindlig wird.
- Methode** Ziel dieser Arbeit war die Erstellung eines neuen VR-Treadmill- (Deutsch Laufanlage) Konzeptes, welches die Laufbewegung mit der virtuellen Welt synchronisiert und so Motion-Sickness vorbeugt. Basierend auf diesem erstellten Konzept, wurde ein Prototyp entwickelt. Dieser erfasst die Laufbewegungen im Sitzen, mittels Linearschienen. Die daraus resultierenden Positionsdaten werden in die eigens dafür programmierte Demoumgebung übertragen und sorgen so für die Immersion wirklich zu laufen.
- Resultat** Alle Testpersonen konnten problemlos auf dem Prototyp laufen und sich in der virtuellen Umgebung intuitiv bewegen, wobei keine Symptome von Motion-Sickness festgestellt werden konnten. Bei VR-Tests ohne ein VR-Treadmill-Konzept wurde bei den gleichen Testpersonen teils Übelkeit oder ein Unwohlsein festgestellt, was für die Wirksamkeit des Konzepts spricht.
- Fazit** Ein Konzept, welches Probleme von Motion-Sickness löst und eine bessere VR-Benutzererfahrung bietet ist sicher auch für den Massenmarkt interessant. Auf diese Art wird VR einem breiteren Publikum zugänglich gemacht, was unserer Gesellschaft interessante Möglichkeiten bietet. Das erarbeitete Konzept ist neuartig, funktioniert und unterstützt Funktionen, die mit gegenwärtig bekannten Methoden nicht möglich sind.

Demonstrationsvideo:



<https://youtu.be/qZTvMpuDZHA>

Glossar

Begriffe	Erklärung
Haptik	Tastende Wahrnehmung / etwas erfühlen können (aktiv)
Simulation	Ist eine möglichst realitätsnahe Nachbildung von Geschehnissen aus der Wirklichkeit wobei mittels Abstraktion das reale Problem in einem passenden Modell behandelt wird.
Virtuell	Eigenschaft eines Objektes in einer Form zu existieren, in der es nicht zu existieren scheint. / Nicht reale Dinge die trotzdem in einer Form zu existieren scheinen
VR	Virtuelle Realität Interaktive simulierte Umgebung
VR Umgebung	Virtuelle Welt, in der man sich bewegen und agieren kann. / Abgrenzung eines VR-Systems (z.B. ein VR Computer-Spiel)
Taktil	Tastsinn / Etwas spüren (passiv)
Linearmotor	Aktive Achse die das getriebene Objekt linear verschiebt
System	Abgrenzung / Zusammengehörigkeit von Bestandteilen
Aktor	Elektrisches Bauteil, dass elektrisches Signal in Bewegung umsetzt
Microcontroller	Computerprozessor mit Rechen und Peripheriefunktionen
Arduino	Physical-Computing-Plattform die aus Software und Hardware besteht. Entwicklungsboards mit entsprechender Programmierumgebung
Wägezelle	Bauteil zum Messen von Kräften
Abtastrate	Frequenz der Auswertung eines Signals
Encoder	Bauteil zur Messung von Drehwinkel
Serial	Kommunikationsart zwischen Bauteilen
Unreal Engine	Spiele-Engine die bei der Entwicklung von Spielen eingesetzt wird
Open Source	Software, deren Quelltext öffentlich ist und von Dritten eingesehen, geändert und genutzt werden kann.
Visualstudio	Programmierumgebung von Microsoft
Pull-UP-Funktion	Eine Funktion von Arduino-Controllern, die es das einfache Verdrahten und Auslesen von Schaltern ermöglicht
Immersion	Das empfinden einer virtuellen Umgebung als Real.
Treadmill	Eine VR Treadmill ein System, welches wie ein Laufband (Englisch Treadmill) das Laufen stationär am gleichen Ort ermöglicht.
ODrive	Steuerungsplatine die Servomotoren ansteuert

1 Vorwort

Schon als kleiner Junge hatte ich viele kreative Ideen und hinterfragte Systeme, welche uns im Alltag begegnen. In den letzten Jahren hatte ich durch meine technische Ausbildung als Automatiker immer mehr Ideen und auch die Möglichkeiten Projekte umzusetzen. Als ich vom Nationalen Wettbewerb von Schweizer Jugend forscht erfuhr, war mir klar, dass ich mit einem Projekt teilnehmen will. Mich für ein Projekt zu entscheiden, fiel mir persönlich schwer, da ich viele potenzielle Ideen hatte. Mein Ziel war es, eine innovative und anspruchsvolle Idee umzusetzen.

Als FPV-Rendrohnen-Pilot und Gamer liebe ich es in virtuelle Welten einzutauchen. Ich kann mich gut an das erste Mal erinnern, als ich eine VR-Brille aufsetzte. Ich war überwältigt von den Möglichkeiten dieser fernen Realitäten. Leider wurde mir schnell übel, da ich anfällig für Motion-Sickness bin. Bis heute kann ich wegen der Motion-Sickness keine VR-Spiele genießen, in denen man sich Fortbewegen muss. Allerdings hoffe ich, dass ich mit meiner Idee einen Lösungsansatz liefern kann, welcher das Problem beheben oder zumindest lindern kann. Laut Recherchen sollte das Verwenden einer sogenannten «VR Treadmill» Motion Sickness lindern oder gar eliminieren. Die Bewegungen werden in eine VR-Welt übertragen und geben dem Anwender das Gefühl, sich durch eine Welt aktiv zu bewegen. Die Schwierigkeit liegt darin, ein realistisches Laufen in alle Richtungen zu ermöglichen.

2 Einleitung

Mit VR-Brillen und Motion-Kontrollern kann heute schon in eine VR-Welt eingetaucht werden. Allerdings gibt es noch keine zufriedenstellenden Bewegungslösungen, welche ein intuitives Laufen in VR-Welten ermöglichen, ohne sich physisch fortzubewegen. Dies bedeutet, dass man sich mit der VR-Brille physisch durch einen Raum oder virtuell mittels eines Joysticks bewegen muss. Da der Raum bei Privatanwendern meist begrenzt ist, wird auf Joystick-Systeme gesetzt, welche bei vielen Menschen Motion Sickness hervorrufen. Motion Sickness wird unter anderem dadurch hervorgerufen, dass sich die visuelle Wahrnehmung und die Bewegungen nicht decken. [1] Wie verbreitet das Problem von Motion Sickness ist, zeigt sich im Internet. Die deutsche Firma Omnia360, welche sich auf VR spezialisiert hat, schreibt auf ihrer Website, dass Motion Sickness derzeit einer der grössten Einsatzbarrieren von VR Systemen sei. [2]

Der vielversprechendste Lösungsansatz ist, die Simulation möglichst real zu gestalten und diese mit dem Körper zu verknüpfen. Die ultimative Umsetzung von diesem Lösungsansatz wäre, den Verstand in eine virtuelle Umgebung hochzuladen und ihm dort einen Körper simulieren, mit dem er sich bewegen, fühlen und alles andere kann, was in der realen Welt möglich ist. Man könnte nicht mehr feststellen, dass man in einer virtuellen Welt ist. Ein solches System ist jedoch weit entfernt von den heutigen technischen Möglichkeiten. Dennoch kann mit heutiger Technik Motion-Sickness gelindert und einen höheren Grad an Immersion erreicht werden, welches das Ziel von jedem VR-System darstellt.

Eine Möglichkeit die virtuelle und die reale Welt näher zusammen zu bringen, ist eine VR-Treadmill. Dies ist eine Laufanlage, mit der man sich auf natürliche Weise in alle Richtungen bewegen und so virtuelle Umgebungen erkunden kann. Eine VR-Treadmill kann man sich wie ein Laufband (engl. Treadmill) vorstellen, welches die Gehbewegungen in die virtuelle Welt überträgt.

Das Ziel dieser Arbeit ist es, nach einem eigenen neuartigen Konzept, einen Prototyp einer VR-Treadmill zu entwickeln. Diese soll ein möglichst intuitives Laufen ermöglichen. Es soll bestmöglich für Heimanwender geeignet sein, ein höheres Immersionslevel ermöglichen und Motion Sickness verhindern. Anhand von diesem Prototyp soll das Konzept getestet werden können. Dese Erkenntnisse sollen als Grundlage für weitere Entwicklungen und Verbesserungen dienen.

3 Theorie und Recherche

3.1 Immersion und VR Systeme

Eine Immersion im VR-Bereich beschreibt, das Eintauchen in eine virtuelle Umgebung. Virtual-Reality-Systeme für Heimanwender mit Brille und Motion Controllern ermöglichen bereits heute eine überzeugende Immersion. Ein solches System ist in Abbildung 1 ersichtlich. Bei einer Immersion gewinnt der Spieler die Wahrnehmung in einer virtuellen Umgebung. Man kommt zur Überzeugung, sich in dieser zu befinden, wobei das Gefühl für die reale Umgebung verloren geht.

Erhältliche VR Systeme haben jedoch zwei Schwachpunkte. Einer davon ist das beschränkte haptische Feedback. Ein weitaus grösseres Problem ist jedoch, dass sich der Anwender nur begrenzt bewegen kann. Begrenzt im Sinne, dass man nicht realistisch durch die unendlich scheinenden VR-Welten bewegen kann, wie es in normalen Spielen üblich ist. Diese Grenze stellen die realen Platzverhältnisse dar. Auf das Problem wird im nächsten Abschnitt 'Problematik beim Bewegen in immersiven Räumen' eingegangen. [2]



Abbildung 1 Oculus Quest VR Headset¹

¹ Bild von digitec.ch \ Oculus Quest 64GB, abgerufen am 28.10.2020

3.1.1 Problematik beim Bewegen in immersiven Räumen

Es gibt verschiedene Konzepte, um sich in VR bewegen zu können. Am besten funktioniert die synchrone Bewegung in VR und der Realität. Möchte man sich bei diesem System im Spiel einen Schritt nach vorne bewegen, muss man in der Realität ebenso einen Schritt nach vorne machen.

Das Problem dabei ist, dass die virtuelle Umgebung meistens grösser ist als die Platzverhältnisse in einem Zimmer. (Visualisiert ist das Problem in der untenliegenden Abbildung)

Der Spieler befindet sich in einem Zimmer an der Wand und kann nicht mehr weiter vorwärts gehen. Befindet er sich in VR bei Position A, sieht er eine Wand, diese muss aber nicht mit der realen Wand übereinstimmen. Es kann gut sein, dass er sich bei Position B befindet oder noch weiter durch eine virtuelle Tür (Pink) laufen will. Dies ist aber nicht möglich, da er sich in der Realität nicht mehr weiter vorwärtsbewegen kann. (Achtung Wand!)

Die Lösung ist, dass sich der Spieler in der virtuellen Umgebung bewegen können muss, ohne sich in der Realität fortzubewegen. Dies wird meistens mit folgenden zwei Methoden umgesetzt.

Der Spieler bewegt sich mit einem **Joystick**, wodurch er sich normal durch den Raum bewegen kann.

Der Spieler **teleportiert** sich durch den Raum, wobei er bei Position A, Position B anvisiert und im nächsten Moment dort ist.

Das Problem ist, dass viele Personen bei solchen Bewegungen, die nicht mit der Realität synchron sind, Motion-Sick werden. Dies tritt weniger auf bei der Teleportation-Methode. Der Nachteil ist, dass bei jeder Teleportation die Orientierung verloren geht². Wird eine Bewegungsanimation beim Teleportieren gemacht, kann der Orientierungsverlust minimiert werden. Dadurch tritt jedoch wieder stärker Motion-Sickness auf, welcher in der Praxis versucht wird, mit einer Verkleinerung des Sichtfeldes entgegenzuwirken. (Getestet mit Oculus Quest 2 und HTC Vive)

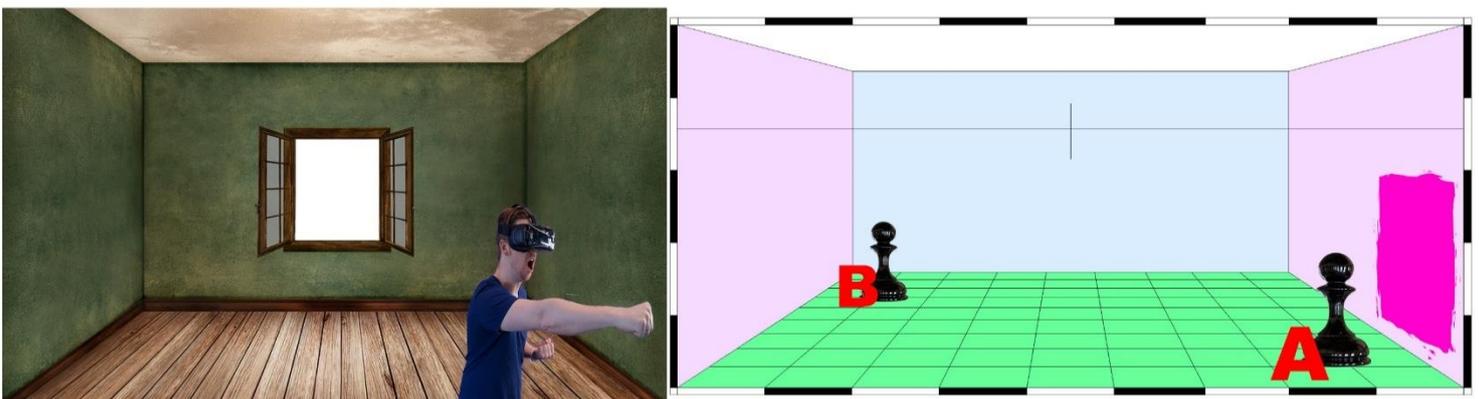


Abbildung 2 Visualisierung Bewegungsproblem

² Teleportation – Instantanen Transport eines Objektes von einem Ort zum anderen ohne den dreidimensionalen Raum dazwischen zu durchqueren. Neuer Ort bedeutet Neuorientierung nötig.

3.2 Entstehung von Motion Sickness

Der Begriff Motion-Sickness ist Englisch und steht für Reisekrankheit oder Bewegungskrankheit. Motion-Sickness wird die körperliche Reaktion auf widersprüchliche Bild- und Bewegungsinformationen im Hirn genannt. Symptome sind unter anderem Übelkeit, Erbrechen, Kopfschmerzen, Blässe und Schwindel.

Die im Kapitel 'Problematik beim Bewegen in immersiven Räumen' erklärte nicht synchrone Bewegung von Bild und Realität führt sehr schnell in VR zu Motion-Sickness. Der gleiche Effekt ergibt sich beim Lesen eines Buches während einer Autofahrt. [1]

Wie viele Personen tatsächlich beim Verwenden von VR Systemen Motion-Sick werden, lässt sich nur schwer ermitteln. Grund dafür ist die Abhängigkeit von zu vielen Faktoren und dass Hersteller von VR-Systemen nicht daran interessiert sind, solche Studien in Auftrag zu geben oder zu publizieren. Bei Recherchen auf Google, zeigten sich Motion-Sickness-Werte von Benutzern, bei etwa 50-90% auf unterschiedlichsten Plattformen. (01.03.2021)

3.3 Methoden um Motion Sickness zu verhindern

Es gibt verschiedene Ansätze, um das Auftreten von Motion Sickness zu verhindern. Momentan sind die zuverlässigsten Lösungen eine statische VR-Umgebung zu verwenden, in der man sich möglichst wenig fortbewegt [2]. Zudem wird oft geraten Pausen einzulegen [2]. Wobei eine Pause einzulegen nicht immer als wirksame Lösung angesehen werden kann, da Betroffene unmittelbar an Motion Sickness Symptomen leiden können [3].

Bei den technischen Ansätzen gegen Motion Sickness wird primär versucht, die virtuelle und reale Welt in Einklang zu bringen. Dies scheint die einzige zuverlässige Methode zu sein, um wirksam gegen Motion Sickness vorzugehen [2].

Eine VR-Treadmill basiert auf dieser Erkenntnis. Die reale Laufbewegung wird in die virtuelle Welt übertragen. So kann Motion-Sickness gelindert oder gar verhindert werden. Auf der Grundlage, dass die vom Hirn wahrgenommenen Informationen in VR mit den ausgeführten Bewegungen übereinstimmen [4]. Zudem wird ein höheres Immersionslevel erreicht was zu einem besseren VR-Erlebnis führt. Ein gutes VR-System mit hohen Bildwiederholfräquenzen, niedriger Latenz und gutem Tracking ist eine Voraussetzung, um die Synchronisation zwischen VR-Welt und VR-System (Realität) möglichst gut zu gewährleisten. Sind diese nicht synchron, besteht ein zusätzliches Risiko für Motion Sickness. [2]

3.4 Haptisches Feedback

Heute haben wahrscheinlich die meisten Menschen bereits Erfahrungen mit haptischem Feedback gemacht. Es ist die Technologie, die uns das Gefühl gibt, beim Tippen auf einer Smartphone Tastatur, mechanisch etwas zu bewirken. In diesem Beispiel sind es Vibrationsmotoren im Smartphone, die diesen Effekt erzeugen [5]. Diese Illusion, etwas Virtuelles fühlen zu können, ist das Ziel des haptischen Feedbacks. Erzeugt wird dieses Gefühl durch Aufbringen von Kräften, Vibrationen oder Kräfte auf den Benutzer [6]. In der Spieleindustrie wird schon seit längerem an solchen Technologien gearbeitet. Game Controller nutzen beispielweise einen Vibrationsmotor, der bei einem Zusammenstoß im Spiel vibriert [7].

Das Unternehmen HaptX entwickelte bereits VR-Handschuhe, welche haptisches Feedback simulieren und dem Nutzer das Gefühl geben, virtuelle Objekte anfassen zu können. Dafür verwenden sie kleine pneumatische Aktoren. [6] In Abbildung 3 ist eine Visualisierung der Physik eines solchen VR-Handschuhs ersichtlich. Wird ein **virtuelles Objekt** berührt, **hält der Handschuh mit Motoren oder Bremsen gegen** die Finger und übt **einen Druck** auf die Haut aus. Obwohl sich nichts zwischen den Fingern befindet, hat man das Gefühl etwas fest zu halten.

Die präzise Bewegungsverfolgung ist ein zentraler Punkt beim haptischen Feedback. Die Anlage muss Kraft, Druck oder Vibration präzise dosieren können, um eine überzeugende Illusion zu erzeugen. Ebenso muss die Anlage oder das Spiel wissen, wann mit einem virtuellen Objekt interagiert wird. Dafür muss das Körperteil, auf welchen man haptisches Feedback übertragen will, virtuell simuliert werden. Es muss eine virtuelle Hand vorhanden sein, um ein virtuelles Objekt fassen zu können und dies haptisch zu simulieren.

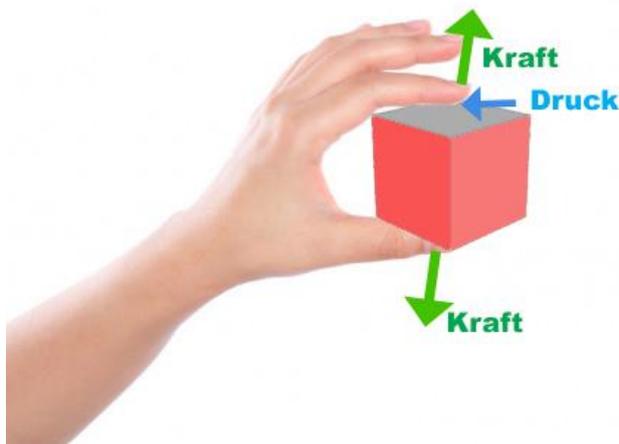


Abbildung 3 Visualisierung Haptisches Feedback

3.5 Verschiedene Achsen

Bei den verschiedenen Achsen und Bezeichnungen bezieht sich diese Arbeit auf das kartesische Koordinatensystem, wie es in Abbildung 4 aufgezeigt wird. Zusätzlich mit der Rotation der verschiedenen Achsen, kann die Lage und Position eines Objektes, sowohl im virtuellen als auch im realen Raum, mit einem Vektor angegeben werden. Die Position des Fusses wird in einem gewissen Zeitintervall von der Laufanlage erfasst. Der Vektor zwischen diesen Positionen stellt dann im Raum pro Zeit den Geschwindigkeitsvektor dar, welcher als Kontrollinput für die VR Welt verwendet wird.

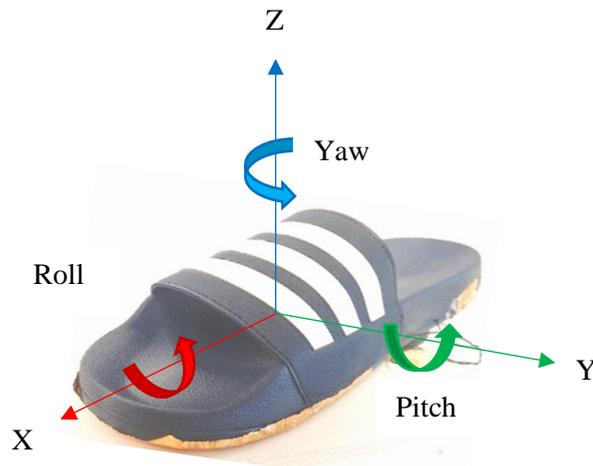


Abbildung 4 Achsen Visualisierung

3.6 Lauf- / Gang-Analyse

Um eine möglichst einfache Treadmill zu entwickeln, mit der man auf natürliche und intuitive Art gehen kann, basiert auf der Analyse des menschlichen Gehens. Die Laufanalyse bestand daraus, zu beobachten, wie der Mensch läuft. Es stellte sich heraus, dass die Laufbewegung auf zwei Achsen in der horizontalen Ebene reduziert werden kann, ohne die Laufbewegung zu fest einzuschränken. Die X-Achse ist in dieser Arbeit die Haupt-Bewegungsrichtung. In diese Richtung kann auch die grösste Bewegungsgeschwindigkeit des Fusses beobachtet werden. Die Yaw-Rotation (in Abbildung 5 mit R dargestellt) dreht um die Z-Achse und entspricht der Rotationsbewegung des Fusses.

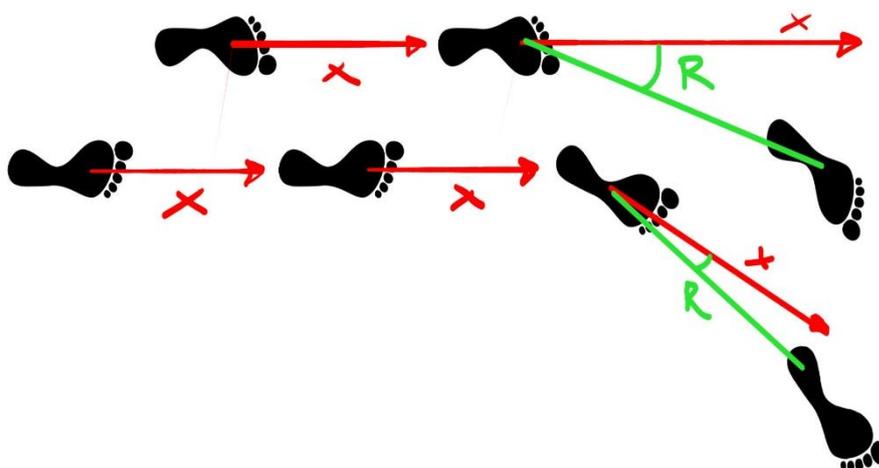


Abbildung 5 Laufanalyse Visualisierung

3.7 ODrive

ODrive ist eine open source Steuer Platine, die aus gewöhnlichen bürstenlosen (BLDC) Motoren starke und präzise Servomotoren macht. Eine solche ODrive Steuerplatine ist in Abbildung 6 ersichtlich. In diesem Projekt werden die Linearachsen mit einer ODrive Platine betrieben, welche ein zentraler Teil der Anlage sind und das haptische Feedback ermöglichen. Mit einer ODrive Steuerplatine können zwei solche Motoren mit über 5 kW Leistung betrieben werden und dies für einen Bruchteil der Kosten von industriellen Servomotor-Systemen. Dies wird erreicht, indem günstige BLDC Motoren aus dem Hobbybereich zusammen mit einem Encoder gekoppelt werden, was in Abbildung 7 zu sehen ist. Diese Motoren werden normalerweise für Drohnen, elektrische Skateboards oder weitere Hobbyapplikationen eingesetzt. Vorteile gegenüber einem Schrittmotor sind die hohe Leistung, Drehmoment, Effizienz, geringe Lautstärke und weitere Funktionen die ODrive bietet. [8]

Der Drehmoment-Modus ist eine entscheidende Funktion für dieses Projekt. Dieser Modus bedeutet, dass die Motordrehzahl irrelevant ist, der Motor jedoch versucht immer ein konstantes Drehmoment zu liefern. Das Drehmoment erzeugt dann übersetzt auf einer Achse ein eine Kräfteinwirkung auf den Fuss. Diese steuerbare Kraft auf den Fuss ermöglicht haptisches Feedback. ODrive unterstützt neben dem Drehmoment-Modus, auch Funktionen wie einen Positions-Modus oder Geschwindigkeits-Modus. [8]

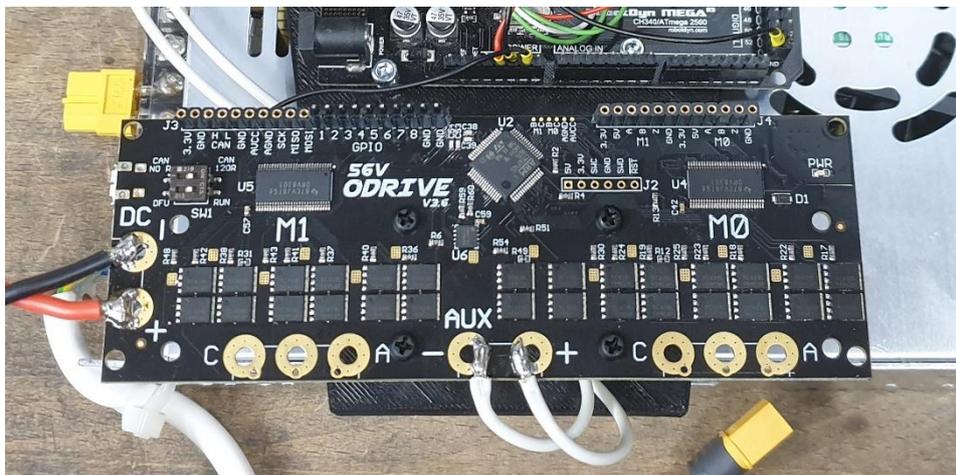


Abbildung 6 ODrive Steuerplatine

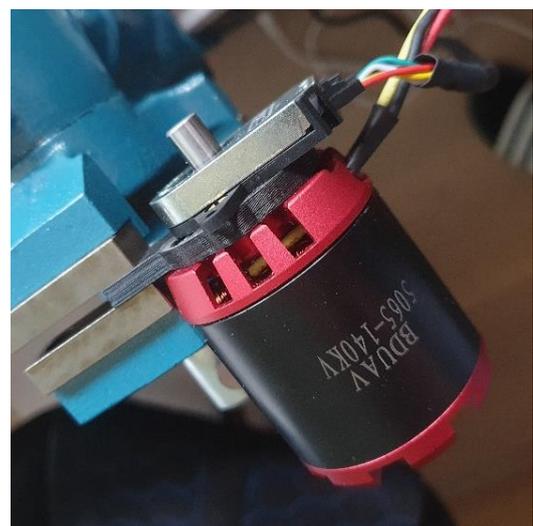


Abbildung 7 BLDC Motor mit Encoder für ODrive

3.8 Bestehende Treadmill-Systeme

Laufbänder, mit denen man sich endlos in VR bewegen kann, sogenannte VR-Treadmills, sind schon seit längerer Zeit in Entwicklung. Man findet beispielsweise auf Plattformen, wie Kickstarter (Plattform zur Umsetzung von neuen Projekten), einige Projektversuche, eine solche VR-Treadmill für Privatanwender zu entwickeln. [9]

3.8.1 Treadmillssysteme mit Rutschebenen

Die meisten Versuche VR-Treadmill auf den Privatanwendermarkt zu bringen, verfolgen das Konzept von Rutschebenen. Diese sind aus vielen technischen Gründen einfacher zu realisieren, wie beispielsweise Laufbänder, da sie nicht motorisiert werden müssen und viel einfacher aufgebaut sind. Dies ermöglicht ein Produkt, das sich preislich in der Nähe des Privatanwender Marktes bewegt (<10'000 CHF). [10]

Benutzer von solchen Rutsch-VR-Treadmills empfinden die Gehbewegungen jedoch als unnatürlich, verglichen mit einem natürlichen Laufprozess. So beschreibt ein Tester, dass die VR-Treadmill Kat Walk C (Abbildung 8) seiner Meinung nach, wie alle VR-Treadmills, nicht unbedingt die Bewegung der realen Welt simuliert, sondern dem Anwender das Gehen in einer komplett neuen Art und Weise beibringt. [11] Bei einer Anfrage bei KAT VR (Produzent von Kat Walk C und anderen ähnlichen Treadmills), ob ihre Treadmills gegen Motion Sickness helfen, erhielt ich von einem Mitarbeiter die Antwort, dass ihre Treadmills gegen Motion Sickness entwickelt wurden und dafür sorgen, dass die Wahrnehmung von Sicht und Körperbewegung übereinstimmen. Er meinte ebenfalls, dass ihr Gerät der beste Weg sei mit diesem Problem umzugehen. [12]



Abbildung 8 Treadmill KAT Walk C

³ Bild von kat-vr.com \ Kat Walk C, abgerufen am 28.10.2020

3.8.2 Sitzende Treadmillsysteme mit Rollen

Ein komplett anderes Konzept verfolgt die österreichische Firma Cybershoes. Sie entwickelten eine Art Schuhaufsatz, der unten an den Schuhen befestigt werden kann. Die Laufbewegung erfolgt im Sitzen, wobei die Bewegung von Rollen erfasst wird (Abbildung 10A). Bei Drehungen wird eine Seitwärtsbewegung mit den Schuhen gemacht, wobei auf einem Stuhl um die eigene Achse gedreht wird. Durch das Sitzen wird das Problem das Gleichgewicht zu halten gelöst, welches viele Treadmills haben. Dies hat den Vorteil, dass der Benutzer sich nicht anbinden oder befestigen muss, wie das bei Steh-Treadmills üblich ist. [10]

Der Verkaufspreis von SFr. 400.-- bis SFr. 500.-- für ein Paket aus Stuhl, Teppich und den Schuhaufsätzen (Abbildung 10B) ist für eine VR Treadmill System eine vergleichsweise günstige Lösung. Sie ist bereits in der Schweiz erhältlich. [13] Die Schuhe ermöglichen weniger ein realistisches Laufen, sondern eine eigene Bewegungsform für VR. Zudem unterstützen die Schuhe kein haptisches Feedback. Das Konzept scheint jedoch gegen Motion Sickness gut zu funktionieren.



Abbildung 10A Cybershoes Paket



Abbildung 10B Cybershoes System mit Rollen

4

3.8.3 Treadmillsysteme mit Laufbändern

Von der Firma Infinadeck gibt es eine VR-Treadmill, die mit Laufbändern arbeitet. Sie nutzen ein System von zwei ineinanderlaufenden Laufbändern, um Bewegungen in alle Richtungen zu ermöglichen. Allerdings brauchen solche Anlagen viel Platz und sind zusätzlich sehr kostspielig, was sie ungeeignet für den Privatanwender machen. Zudem kann man sich nur langsam auf der Anlage bewegen und das haptische Feedback wird nicht unterstützt. [14]

⁴ Bilder aus Cybershoes Business plan [10]

4 Konzept

Das Konzept für die Umsetzung der VR-Treadmill basiert auf den verschiedenen Erkenntnissen, die im Theorie- und Recherchebereich gemacht wurden. In den Teilkonzepten sollen Erkenntnisse aus der Recherche angewendet werden. Auf die einzelnen Anforderungen an die Komponenten und wie diese mit der Konzeptionierung zusammenhängen, wird im entsprechenden Unterkapitel zu den Komponenten eingegangen. Die Konzepte definierten die Anforderungen und sollen eine solide Grundlage für die Ideen und Umsetzung bilden. Die Abbildungen 11 und 15, mit den eingefärbten Elementen dienen zur Orientierung.

4.1 Bewegungskonzept

Um möglichst eine heimanwender-freundliche Treadmill zu bauen, wurde ein System gewählt, bei dem man im Sitzen eine Gehbewegung simuliert, ähnlich wie dies vom Unternehmen Cybershoes umgesetzt wurde. Bei einem solchen System ist das Gewicht des Benutzers hauptsächlich auf dem Stuhl, was zum Vorteil hat, dass die Anlage nur einen Bruchteil des Gewichtes tragen muss und so schlanker ausgelegt werden kann. Durch das Sitzen wird zugleich das Problem mit dem Gleichgewicht gelöst, was sicherheitstechnisch und anwendungstechnisch Vorteile bringt und das bei vergleichsweise geringen Kosten. Bei der Cybershoes-Anlage macht der Benutzer jedoch die Gehbewegung auf Rollen, was nicht als realistisches Laufen überzeugt. Als Alternative setzt dieses Konzept auf die Überlegung von zwei Schlitten, die wie ein beweglicher Boden auf zwei Achsen hin und her fahren und so die Unterlage für das Laufen bilden. Cybershoes kann die Drehbewegung nicht realistisch simulieren. Dies sollte bei der zu konzeptionierenden Anlage, besser gelöst werden. Der gewählte Lösungsansatz ist, die Fussdrehung beim Laufen auf den «Virtuellen Bodenplatten» zu verfolgen und diese als Drehbewegungsinput auf der Yaw Achse zu verwenden. Somit bleibt der Benutzer immer gleich ausgerichtet und kann sich trotzdem in alle Richtungen drehen und sich vor / rückwärts bewegen. Die Füße bleiben immer mit der Linearachse verbunden. Zudem wurde das Ziel gesetzt, die Anlage so hochwertig und robust wie möglich zu konzipieren. (Zur Visualisierung zeigt Abbildung 12 einen Laufzyklus)



Abbildung 11 Bild des Aufbaus (von Vorne)

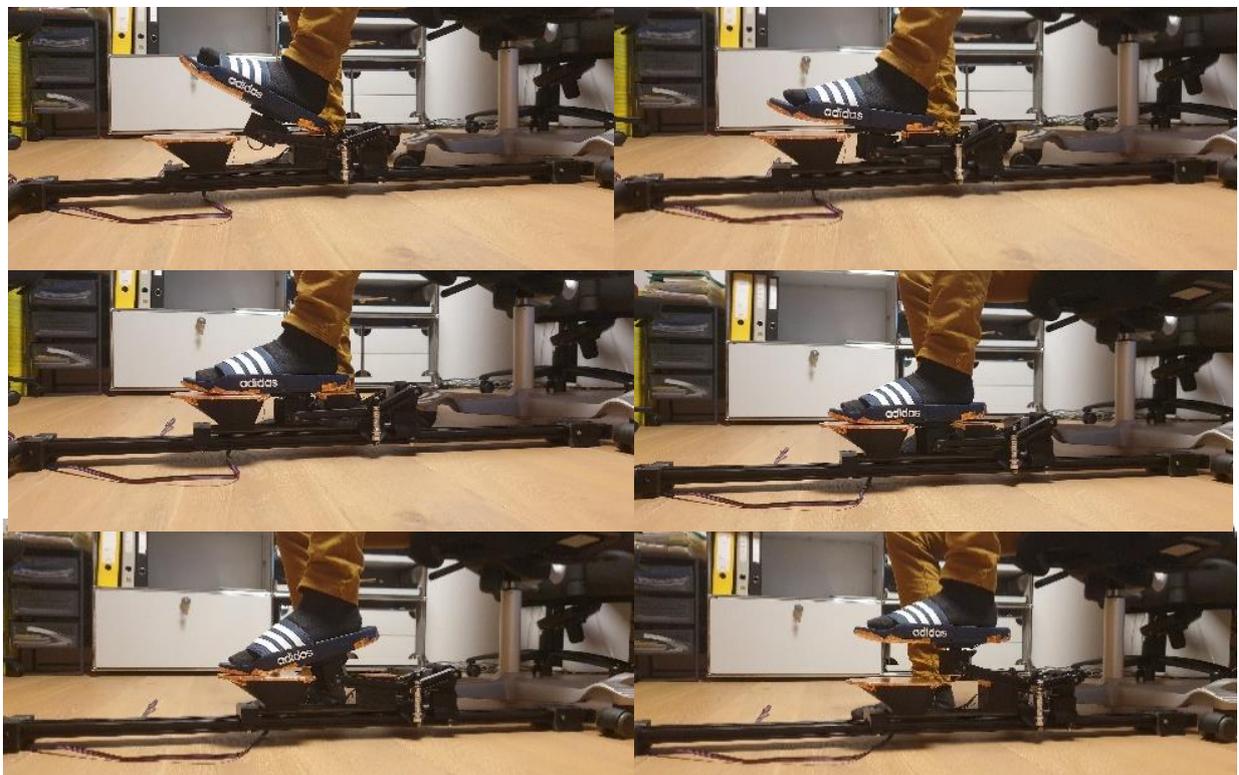


Abbildung 12 Laufbewegung auf Anlage

4.2 Linearachsen (Schwarz)

Die Linearachsen ermöglichen den Hauptteil der Laufbewegung. Sie sind dafür verantwortlich, dass man den Fuss vorwärts und rückwärts bewegen kann und tragen das Gewicht der Beine. Um dem Laufen möglichst nahe zu kommen, sollte man die Anlage möglichst wenig spüren. Dafür müssen die Linearachsen besonders spielfrei und leichtläufig sein.

4.3 Riemensystem (Blau)

Das Riemensystem überträgt die lineare Bewegung des Schlittens in die Rotationsbewegung des Motors mit Encoder. Es muss möglichst spielfrei sein, um präzise Messwerte der Linearachse zu garantieren und um schnell mit den Motoren reagieren zu können. Zudem ist die Robustheit wichtig, dass es genügend Kraft auf die Füße ausüben kann, um diese beispielweise bei einer virtuellen Wand zu stoppen.

4.4 Erfassungssystem (Blau)

Die Geschwindigkeit der Linearachsen entspricht der Geschwindigkeit, mit der sich der Nutzer in der VR-Welt bewegt. Um ein flüssiges Laufen zu ermöglichen, sollte das System möglichst präzise und latenzfrei sein. Zudem sollte die Kraft, die ein Nutzer gegen die Linearachse ausübt, ausgewertet werden. Mit dieser Information kann dann die Reibung reduziert werden und später das haptische Feedback implementiert werden.

4.5 Haptik System

Das Haptik-System ist das grösste Zusatzfeature an der Anlage. Es ermöglicht, virtuelle Objekte fühlen zu können. Ein Feature, das laut Recherchen, in der Art noch nie in einer VR-Treadmill implementiert wurde. Das Spezielle ist, dass die Anlage mit Force Feedback arbeitet, eine Art von haptischem Feedback. Durch die Kräfte, die auf die Füße ausgeübt werden können, kann die Physik eines Bodens simuliert werden. Sind beide Füße abgestellt, werden sie virtuell aneinandergeschlossen und können sich nur noch synchron bewegen. Auch denkbar wäre die Laufbewegung mit abgehobenem Fuss in der Luft leichter zu machen als bei Bodenkontakt. Solche Features können problemlos implementiert und getestet werden.



Abbildung 14 Fuss in der Luft



Abbildung 14 Fuss bei Bodenkontakt

4.6 Gelenkarm (Z-Achse & Pitch) (Rot)

Der Gelenkarm ist die physikalische Verbindung zwischen der Fusshalterung und dem Schlitten. Er überträgt die Kraft des Fusses auf den Schlitten und muss diesem genügend Bewegungsfreiraum lassen. Die Bewegungen des Fusses, wie Abrollen und Heben (Z-Achse), müssen möglichst uneingeschränkt sein. Ebenfalls sollte der Fuss unterstützt werden, wodurch Hebebewegungen einfacher und weniger anstrengend sind. Mechanisch muss der Gelenkarm den Kräften standhalten und sicherheitshalber bei einem Notfall gezielt versagen, falls etwas Ungeplantes eintritt.

4.7 Fusshalterung und Achse (Yaw) (Gelb)

Der Fuss wird physikalisch über die Fusshalterung mit dem Gelenkarm und dem Schlitten verbunden. So können Kräfte vom Fuss auf den Schlitten und das Linearsystem übertragen werden. Dabei muss die Fusshalterung einen guten Tragekomfort bieten und das Drehen des Fusses ermöglichen und messen.

4.8 Fussplatten «Virtuelle Bodenplatten» (Grün)

Die Fussplatten sind die Standflächen, auf denen beim Laufen abgestanden wird. Diese müssen das Gewicht der Beine tragen und das Abstehen erfassen. Der Fuss muss in verschiedenen Positionen abgesetzt und natürlich abgerollt werden können. Zudem müssen sie ein leichtes Drehen um die Yaw Achse des Fusses ermöglichen, welche dann als Drehen in VR umgesetzt wird.

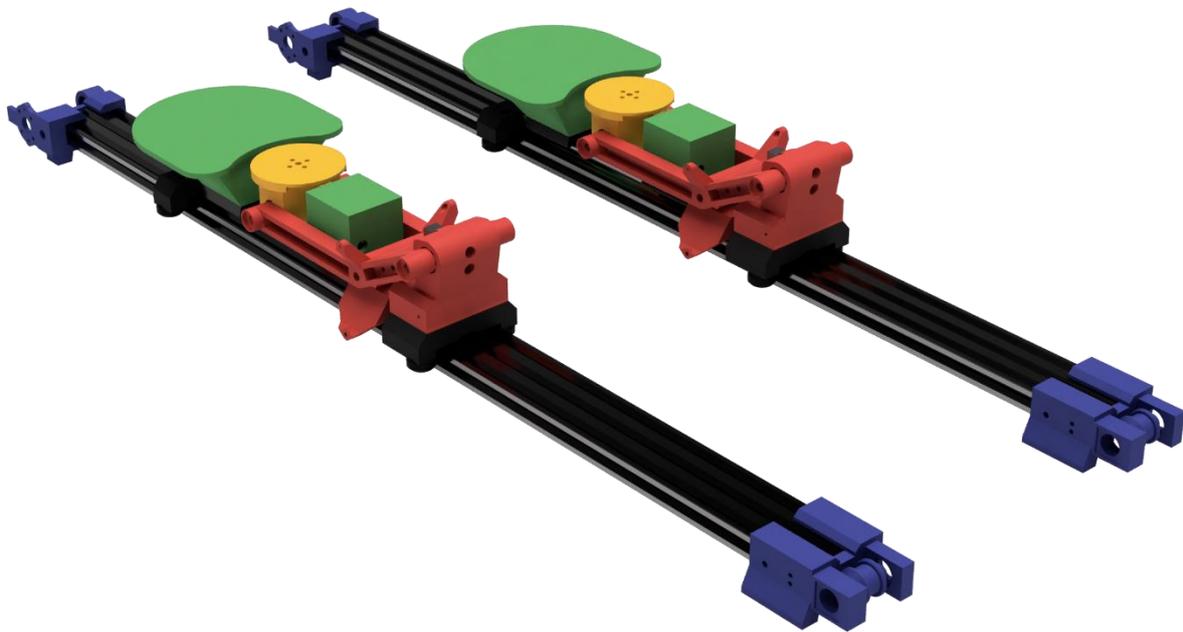


Abbildung 15 Bild des Aufbaus (von Hinten)

5 Umsetzung

5.1 3D Teile CAD und 3D Druck

Als 3D Zeichnungsprogramm wurde Fusion 360 von Autodesk verwendet. Modelliert wird hauptsächlich auf verschiedenen Ebenen mit Skizzen, die dann extrudiert werden. Abbildung 16 zeigt einen Zeichnungsbereich von Fusion 360, in dem die Teile erstellt werden. Die grösste Herausforderung beim Zeichnen war, die Abhängigkeiten richtig zu verknüpfen und die Übersicht zu behalten. Für das gesamte Projekt wurden über 80 solcher Skizzen gezeichnet. Diese bilden die Basis für über 50 3D-gedruckte Einzelteile. Dies bedeutet, dass weit über 50 Teile 3D gedruckt wurden, da noch Änderungen vorgenommen wurden und einige Teile nicht auf Anhieb passten. Insgesamt liefen die 3D Drucker (Abbildung 18A und 18B) etwa 400 Stunden und verbrauchten rund drei Kilogramm Kunststoff-Filament. Als Filament wurde schwarzes "Purefil PLA Filament" von einem Schweizer Hersteller verwendet. Es stellte sich heraus, dass es wichtig ist, immer das gleiche Filament zu verwenden, da sonst die 3D gedruckten Teile in Qualität und Toleranz zu stark abweichen können. Die Toleranzen bei einigen Teilen waren sehr wichtig, da für viele Verbindungen eine Presspassung gewählt wurde. Dies bedeutet, dass zum Beispiel alle Kugellager einfach in ihre Aussparung gepresst sind und auf diese Weise spielfrei sitzen (Abbildung 19).

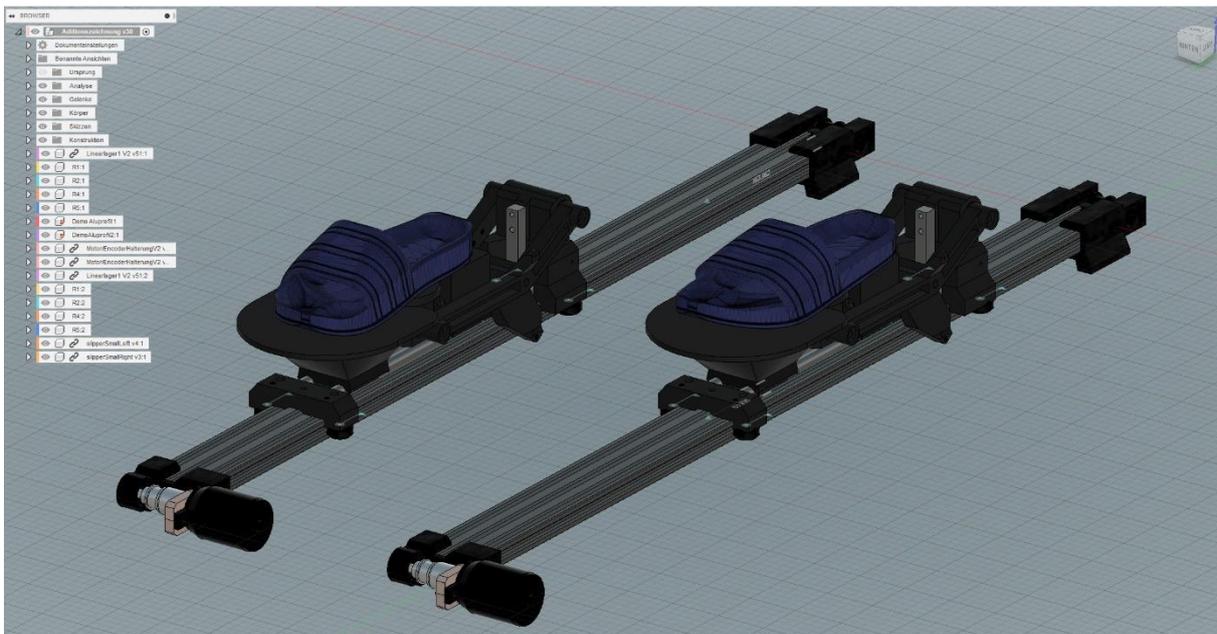


Abbildung 16 Zeichnungsbereich Fusion 360

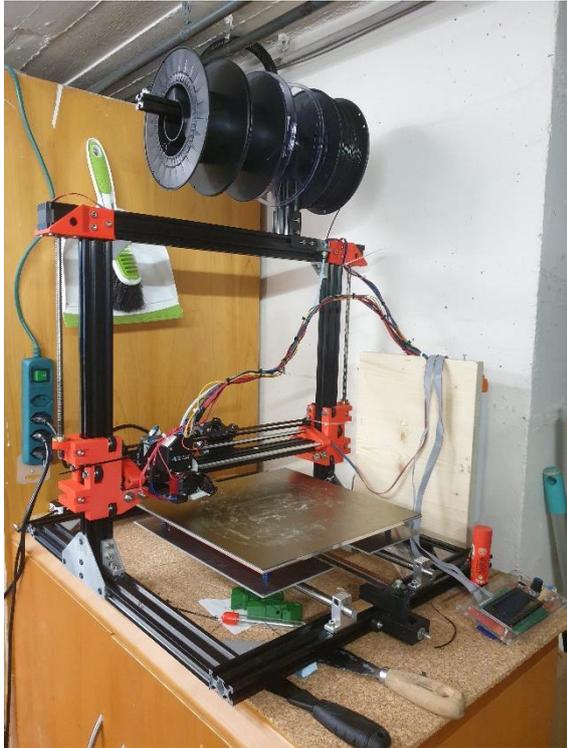


Abbildung 18A Selbstgebauter 3D Drucker



Abbildung 18B 3D Drucker Ender 3 V2

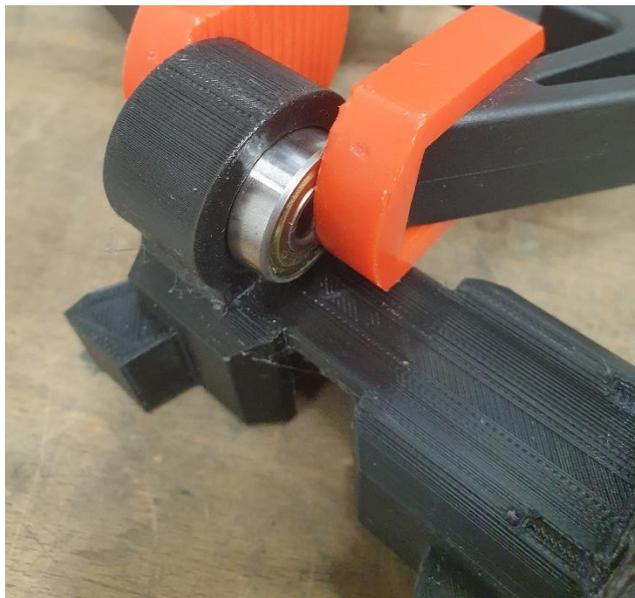


Abbildung 19 Einpressen von Kugellager

5.2 Linearschienen (X-Achse)

Die Linearschienen wurden aus Aluminiumprofilen gefertigt (Abbildung 21A). Diese sind robust und bilden mit V-Slot Rollen (Abbildung 21B) einen sauberen Lauf mit wenig Reibung. Vorteile sind unter anderem, dass sie günstig sind, leise, robust und mit wenig Reibung laufen. Die Nachteile an V-Slot Systemen sind, dass sie nicht sehr präzise und langlebig sind. Da die Präzision bei dieser Applikation etwas vernachlässigt werden kann und die Langlebigkeit der Rollen kein Problem darstellt, begründet die Wahl dieses Systems.

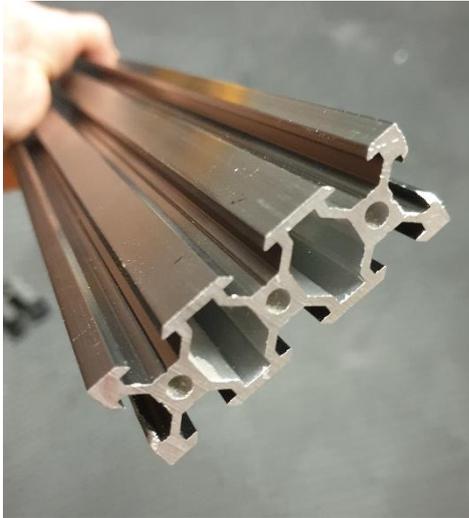


Abbildung 21A Aluminium Profil



Abbildung 21B V-Slot Rollen

Kritisch sind die Hebelkräfte, welche die Linearachsen aushalten müssen. Diese entstehen beim Versuch, die Füße seitlich zu bewegen. Je höher die Position der Füße, desto grösser wird die Hebelkraft. Um diese Kräfte gut aufzunehmen und die Stabilität zu gewährleisten, wurden breite Profile (60 x 20 mm) verwendet. Abbildung 22 zeigt einen entsprechenden Schnitt eines Schlittens mit den genannten Hebelkräften, die auftreten können.

Wie man im Schnitt auch erkennen kann, sind beide Rollen am gleichen Schlittenteil befestigt. Dies bedeutet, dass kein Spannsystem entworfen wurde, was für diese Rollen üblich wäre. Dafür wurde die Flexibilität von 3D-gedruckten Teilen ausgenutzt.

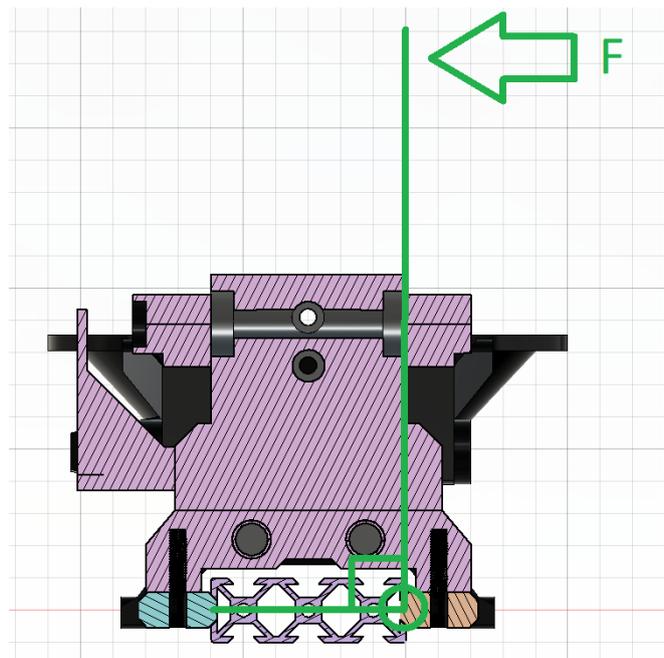


Abbildung 22 Schnitt eines Schlittens

Der Linearschlitten (Abbildung 23), auf dem die restliche Konstruktion befestigt wird, besteht aus zwei Schlittenteilen, an denen die Rollen, sowie zwei 12 mm Stahlrohre, befestigt sind. Diese Rohre sind günstig und sorgen für eine steife, stabile Konstruktion.

Der Vorteil des Schlittensystems ist, dass es modular ist und Komponenten, wie die Fussplatte, einfach verschoben oder ersetzt werden können.



Abbildung 23 Linearschlitten

5.3 Riemensystem (X-Achse)

Das Riemensystem wurde mit Standard GT2 Riemen umgesetzt, welche bei 3D-Druckern üblich sind. Dieser läuft einmal komplett um das Aluminium Profil herum und ist an der Wägezelle befestigt. Der Riemen überträgt die Kraft des Motors auf den Linearschlitten. Auf die Wägezelle wird im Kapitel Erfassungssystem eingegangen. Bei der Konstruktion wurde versucht, möglichst viele Normteile zu verwenden. Beispielsweise besteht die Welle bei der Umlenkungsrolle (Abbildung 25B) aus einem günstigen Aluminiumstab mit 8 mm Durchmesser. Die dazu passenden Motor-Pullys und Kupplungen konnten direkt in China bestellt werden. Die verwendeten Kugellager für 8 Millimeter Wellen, werden ebenfalls in Rollbrettern verwendet und sind somit gut erhältlich und günstig. Damit das System möglichst leise und reibungsarm läuft, sind alle Pullys zweifach kugellagert.

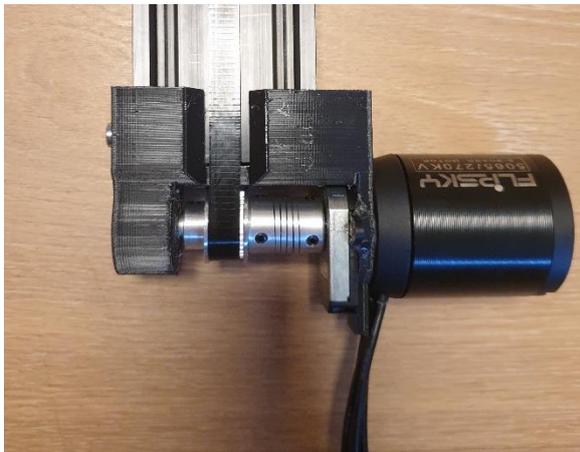


Abbildung 25A Riemensystem Antriebsseite



Abbildung 25B Riemensystem mit 3D-gedrucktem Pully

5.4 Erfassungssystem (X-Achse)

Die Bewegung der Linearschlitten wird über das Riemensystem auf die Motoren und Encoder übertragen (Abbildung 26). Die Encoder werden vom ODrive ausgewertet und die Geschwindigkeit sowie die Position werden anschliessend in jeweils einen Parameter umgerechnet. Die Encoder haben eine Auflösung von über 8'000 Schritten pro Umdrehung. Diese können mit dem dazugehörigen Kabel zur ODrive Steuerplatine, im ODrive Shop gekauft werden. Die theoretische Auflösung im System beruht sich mit diesen Encodern auf über einen 1/10 Millimeter. Realistisch betrachtet liegt die Auflösung mit dieser Umsetzung im Millimeterbereich, was ausreichend ist.

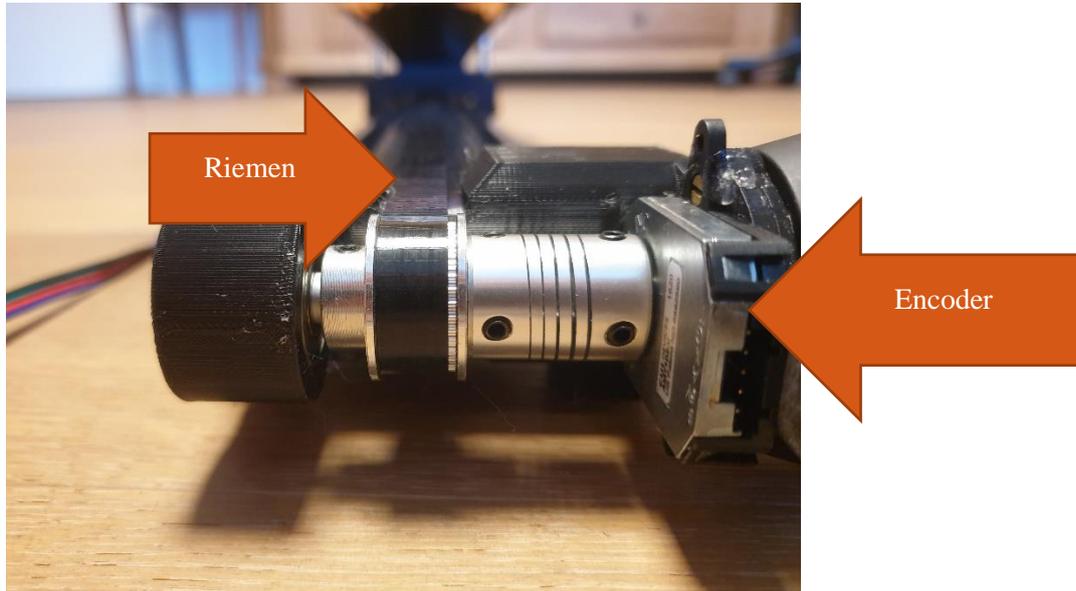


Abbildung 26 Übertragung mit Riemensystem

Für Funktionen, wie das haptische Feedback, muss die Kraftübertragung gemessen werden. Dies wird von einer Wägezelle ausgeführt. Eine Wägezelle misst die Kraft, die vertikal auf die Wägezelle drückt und gibt diese als analoges Signal heraus. Bei der ersten Version (Abbildung 28A) wurde versucht, die Kraft auf dem Gelenkarm, den Fuss und Schlitten zu messen. Dies führte zu vielen Problemen, weshalb in der zweiten Version (Abbildung 28B) die Kraft auf dem Riemen gemessen wird, der nun direkt an der Wägezelle befestigt ist.



Abbildung 28A Kraftmessung V1

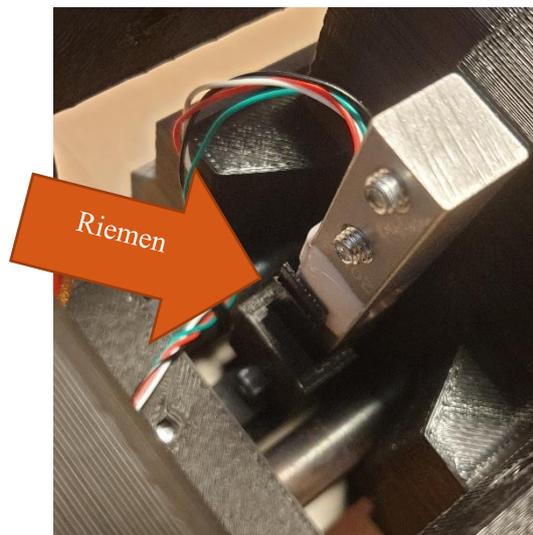


Abbildung 28B Kraftmessung V2

5.5 Force Control (X-Achse)

Bei der Anlage wurde auf ein Force-Haptik-System gesetzt. Dieses setzt tatsächlich Kraft auf den Fuss an und gibt das Gefühl, dass man wirklich gegen ein virtuelles Objekt drückt. Dieses ist anders im Vergleich zu den meisten haptischen Systemen, da diese nur auf die Haut wirken, in dem sie Druck oder Vibrationen erzeugen. Erzeugt wird diese Kraft mit den BLDC Motoren an den Linearachsen, welche von der ODrive Platine gesteuert werden. Drückt nun der Fuss in der virtuellen Welt gegen ein Objekt, erhält die ODrive Platine den Befehl, dieses Drücken in die Realität umzusetzen. In Abbildung 29 wird dies visualisiert.



Abbildung 29 Haptisches Force Feedback

Die Motoren ermöglichen nicht nur haptisches Feedback, sie tragen auch zu einem natürlicheren Laufen bei, indem sie den Widerstand einer Bewegung erhöhen oder verkleinern. So reduziert das System bei gehobenem Fuss (Fuss in der Luft) den Widerstand auf ein Minimum. Am Boden hingegen, erzeugt es beim Laufen ein natürliches Beschleunigungsgefühl. Auch die Füße aneinander zu koppeln, falls beide abgestellt sind und nur noch synchrone Bewegungen zu erlauben, würde der normalen Laufphysik entsprechen und könnte Sinn machen. Diese Funktionen können in der Software einfach implementiert und dann evaluiert werden.

Die Anlage kann schnell grosse Kraft auf die Fusschlitzen und somit auf die Füße bringen. Diese ist momentan aus Sicherheitsgründen auf 75W pro Motor in der Software limitiert. Möglich wäre das über 10-Fache, was für genügend Spielraum sorgen sollte, um realistische Kräfte auf den Fuss bringen zu können. Die Geschwindigkeit wurde ebenfalls auf 20 Umdrehungen in der Sekunde limitiert.

5.6 Gelenkarm (Z-Achse & Pitch)

Der Gelenkarm (Abbildung 31A& 31B) ermöglicht das Heben (Z-Achse) und Abrollen (Pitch) des Fusses. Dies wurde mit zwei kugelgelagerten Achsen und einem 3D gedruckten Arm realisiert. Um den Fuss beim Laufen zu unterstützen und das Gewicht der Anlage zu kompensieren, wurden 140 Millimeter Federn verbaut. Der Fuss kann von den Bodenplatten bis zu 11 Zentimetern angehoben werden, was für eine normale Laufbewegung laut Tests ausreichend ist. Durch die Biegung im Arm wird für genügend Bewegungsfreiheit beim Abrollen gesorgt. Um Querkräfte besser aufnehmen zu können, wurde ein Verbindungsteil 3D-gedruckt, welches die Arme mit massiven Schrauben verbindet (Abbildung 32).

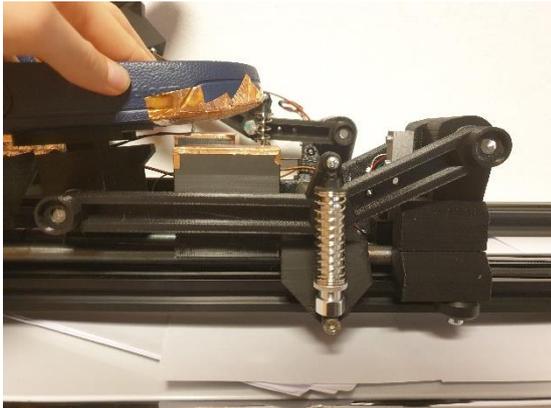


Abbildung 31A Untere Position Gelenkarm

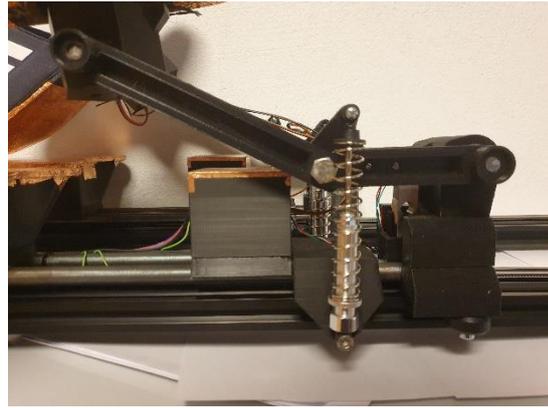


Abbildung 31B Obere Position Gelenkarm

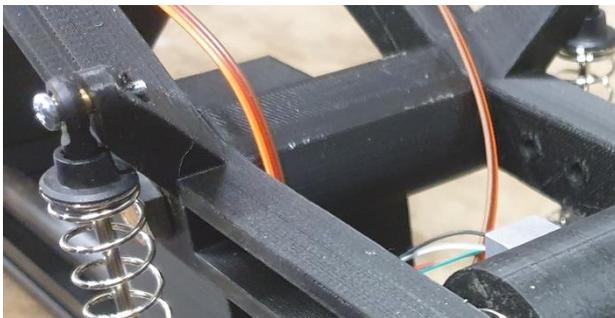


Abbildung 32 Verbinder zwischen Armen

Die Arme wurden so dimensioniert, dass diese die Kräfte bei der Laufbewegung gut aufnehmen können, jedoch bei einem Notfall (beispielsweise umstürzen in der Anlage) brechen würden. Dies ist sehr unwahrscheinlich, beugt jedoch Knöchelverletzungen vor.

5.7 Fusshalterung und Achse (Yaw)

Um die Füße mit der Anlage komfortabel zu verbinden und ein angenehmes Laufen ermöglichen, wurde als Fusshalterung Sandalen von Adidas (Adiletten) (Abbildung 35) genommen. Diese haben eine glatte Oberfläche, welche gutes gleiten auf den Fussplatten ermöglicht. Zudem lassen sie sich schnell an und ab ziehen und haben einen guten Tragekomfort. Der Halt in den Sandalen zeigte sich als genügend hoch. Um die Adiletten mit der Anlage zu verbinden, wurde ein 3D-gedruckter Adapter auf die Unterseite geklebt (Abbildung 35B und Abbildung 35C). Um die Füße auf der Yaw Achse drehen und tracken zu können, wurde ein Verbindungsteil 3D-gedruckt, in welchem ein Potentiometer die Drehbewegung auswertet. Geführt wird die Bewegung durch ein grosses Kugellager.



Abbildung 35A Sandalen von Adidas



Abbildung 35B Fusshalterung / Fussverbinder



Abbildung 35C Aufrauen der Oberfläche für Klebevorgang

5.8 Fussplatten & Erkennung

Um ein realistisches Absteigen und Abrollen (Abbildung 37) auf der Anlage zu ermöglichen, wurden Fussplatten 3D-gedruckt, die mit einem Spannmechanismus an den Rohren befestigt werden. Der Spannmechanismus ist auf der Abbildung 36 im Schnitt ersichtlich. Er ermöglicht ein einfaches Justieren der Platten und macht so die Anlage flexibel. Zudem muss erfasst werden, ob die Füße abgestellt sind, also Kontakt mit der Fussplatte haben oder in der Luft sind. Dies wurde mit einer Kupferschicht erreicht, die aufgeklebt wurde. Die Schicht an der Fusssohle ist mit Masse (0 V Potential) verbunden und die Schicht an den Fussplatten mit einem Pin des Microcontrollers, welcher mit der eingebauten Pull-Up-Funktion den Kontakt zwischen diesen Flächen feststellen kann. Ein weiterer Vorteil der Kupferschicht ist, dass ein gutes Gleiten zwischen Schuh und Fussplatte ermöglicht wird, was bei Drehbewegungen wichtig ist. In der hinteren Fussplatte befindet sich zudem die ganze Elektronik (Abbildung 38) eines Fusschlitzens. Diese wertet die Messzelle, den Potentiometer in der Fusshalterung (Yaw) und ob der Fuss abgesetzt ist aus.

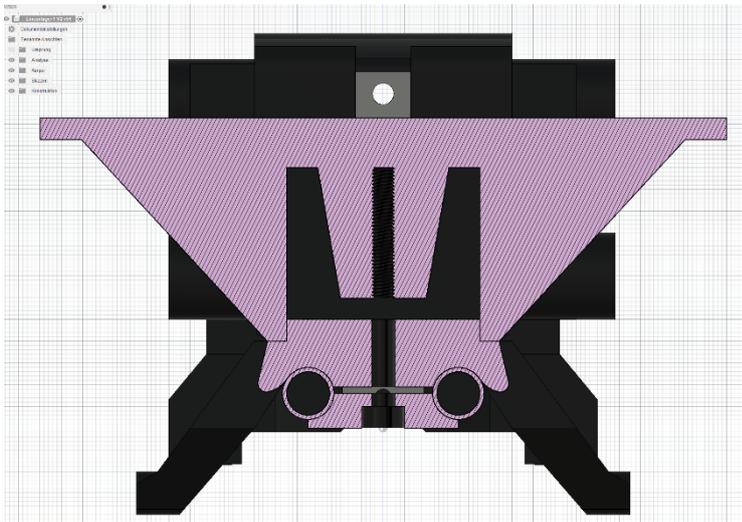


Abbildung 36 Schnitt vordere Fussplatte mit Spannmechanismus

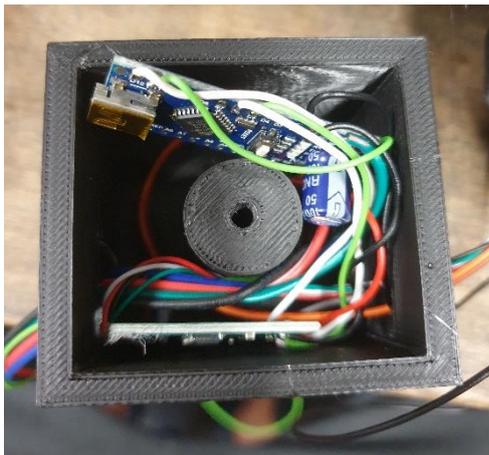


Abbildung 38 Elektronik in der hinteren Fussplatte



Abbildung 37 Abrollen des Fusses

5.9 Elektronik Hardware & Software

5.9.1 Microcontroller / Arduino / IDE

Die Aufgabe der Microcontroller ist es, alle Sensoren zu erfassen und daraus die Bewegungen zu errechnen, welche in die VR-Umgebung übertragen wird. Da die Entwicklungszeit für das Projekt sehr begrenzt war, wurde eine flexible Arduino-Plattform (Abbildung 40) gewählt. Die Arduino-Umgebung ist simpel aufgebaut und ermöglicht es Codes aus vorherigen Projekten wiederzuverwenden. Ausserdem war ein grosser Teil der Komponenten vorhanden und musste nicht extra bestellt werden. Arduino ist eine Open Source Entwicklungsplattform, die aus Software und Hardware besteht. Genutzt wird sie grösstenteils für semiprofessionelle Anwendungen und in der Bastler-Szene. Viele 3D Drucker laufen beispielweise auf Arduino Plattformen. Programmiert werden diese Microcontroller in der eigens dafür entwickelten Arduino IDE. Diese ist simpel aufgebaut und ist einfach zu bedienen, bietet aber auch nicht alle Funktionen. [15]

Da dieses Projekt etwas komplexer ist, wurde in die Entwicklungsumgebung Visual Studio von Microsoft gewechselt. Da diese nicht direkt das Programmieren von Arduino Microcontrollern unterstützt, wurde das Plugin Visual Mirco verwendet. Dieses Plugin ermöglicht es in einer Anwendung mehrere Microcontroller gleichzeitig zu programmieren und alle im seriellen Monitor zu debuggen. Zudem können serielle Befehle an die Microcontroller gesendet werden. Eigenschaften, die beispielweise PlatformIO (andere bekannte Arduino IDE) nicht unterstützt, aber für den Aufbau von dem seriellen Bussystem wichtig waren.

Der in Abbildung 40 markierte Arduino Mega wurde zuerst verbaut. Da er jedoch nicht über genügend Leistung und Funktionen verfügte wurde auf ein Teensy 3.2 (Abbildung 39) gewechselt. Dies ist ein Arduino kompatibler Microcontroller, der denselben Code unterstützt und ein Vielfaches der Rechenleistung hat.

5

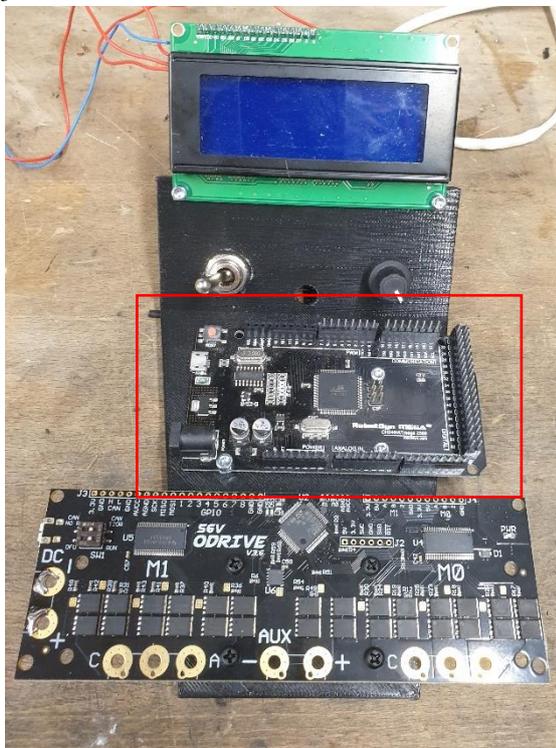


Abbildung 40 ArduinoMega oberhalb von ODrive



Abbildung 39 Teensy 3.2

⁵ Bild Teensy 3.2 von <https://www.pjrc.com/teensy/> abgerufen am 10.03.2021

5.9.2 Aufbau

Augenfällig beim Betrachten des Schemas (Abbildung 41, Anhang) ist, dass drei Microcontroller verwendet werden. Dieses wurde so umgesetzt, damit die Verkabelung vereinfacht wird und bessere Messwerte garantiert werden können. Zu den Fusschlitzen führen jeweils nur vier Adern in einem Flachbandkabel. Darin geführt ist die Stromversorgung und die serielle Kommunikation. Darüber werden die Daten von den Sensoren in den Fusschlitzen übertragen, die von einem Arduino Nano ausgelesen und so digitalisiert werden. Dies hat zum Vorteil, dass die analogen Sensoren mit möglichst wenig Rauschen ausgelesen werden können und viel Kabel gespart wird. Ansonsten müssten zu jedem Fusschlitzen etwa zehn Adern geführt werden. Zudem würde dies die Anlage unflexibel für weitere Erweiterungen machen.

Die «TuchPlate» Sensoren sind die Kupferflächen an den Schuhsohlen und auf den Fussblöcken, die feststellen, ob der Fuss aufgesetzt ist. Der ODrive ist ebenfalls mit dem Teensy verbunden und kommuniziert mit diesem über TX1 und RX1. Gespiesen wird der ODrive direkt mit 24V DC vom Netzteil (PSU).

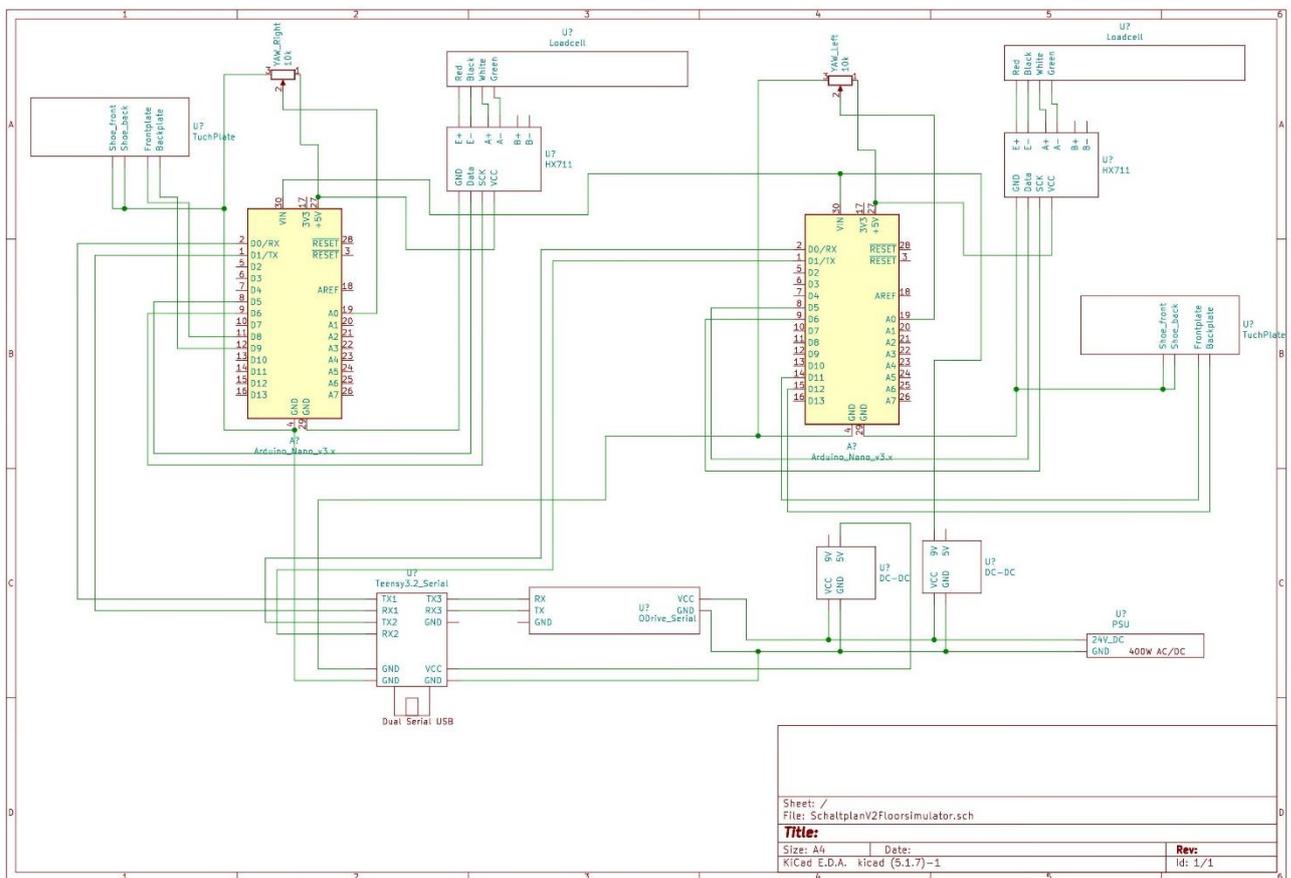


Abbildung 41 Schema

5.9.3 Kommunikation / Schnittstellen

Wie man im Schaltplan erkennen kann, sind die Microcontroller jeweils über RX und TX miteinander verbunden. Diese RX & TX - Pins sind die seriellen Schnittstellen an den Microcontrollern und ermöglichen das Austauschen von Daten. Im System dieses Projektes werden Strings hin und her gesendet, welche einfache Zeichenketten sind (z.B. 'X5Y9' oder 'Hello World'). Sendet nun ein Microcontroller einen String an einen anderen Microcontroller, wird dieser von dem Microcontroller empfangen, ohne diesen bei seiner eigentlichen Aufgabe zu stören. Dadurch können in einem solchen seriellen Netzwerk effizient und zuverlässig Daten ausgetauscht werden, mit möglichen Datenraten von über einer Million Zeichen in der Sekunde. [16]

Der Teensy ist in dem Sinne speziell, dass er drei Serielle-Schnittstellen hat. Dadurch kann er problemlos mit mehreren Geräten gleichzeitig kommunizieren. Im Aufbau kommuniziert er über Serial mit den beiden Arduino Nanos, dem ODrive und dem PC. Die Kommunikation zwischen Anlage und PC wurde ebenfalls via Serieller-Schnittstelle implementiert, damit Daten für das haptische Feedback ausgetauscht werden können. Normalerweise werden Bewegungsinputs von Gamecontrollern mit einem anderen, standardisierten USB-Protokoll übertragen. Dies macht verschiedene Spiele und Inputgerätehersteller miteinander kompatibel, bietet jedoch für diese Anwendung nicht die gewünschten Funktionen.

5.9.4 Immersive Umgebung / Unreal Engine

Durch die Wahl einer seriellen Schnittstelle, müssen die Bewegungsinputs und Ausgabe für das haptische Feedback selbst in eine Umgebung/ Spiel implementiert werden, da dies standardmässig nicht unterstützt wird. Als Game Engine, die Entwicklungsumgebung, in welcher Spiele werden, wurde die Unreal Engine gewählt. Diese hat gute Beispielspiele (Abbildung 42), in welchen ein solcher serieller Input und Output relativ einfach implementiert werden konnte. Für das Lesen und Schreiben der seriellen Schnittstelle wurde ein Plugin namens UE4Duino verwendet, welches, wie der ganze Rest, graphisch programmiert werden kann.

Um wissen zu können, wann Kraft an den Füßen angesetzt werden muss, wird der Fuss in der VR-Welt auf Interaktionen mit virtuellen Objekten untersucht. Dafür müssen in der VR-Welt die Füße theoretisch simuliert werden. Da dies bei Spielen normalerweise nicht nötig ist und somit nicht impliziert ist, muss dies speziell programmiert werden.

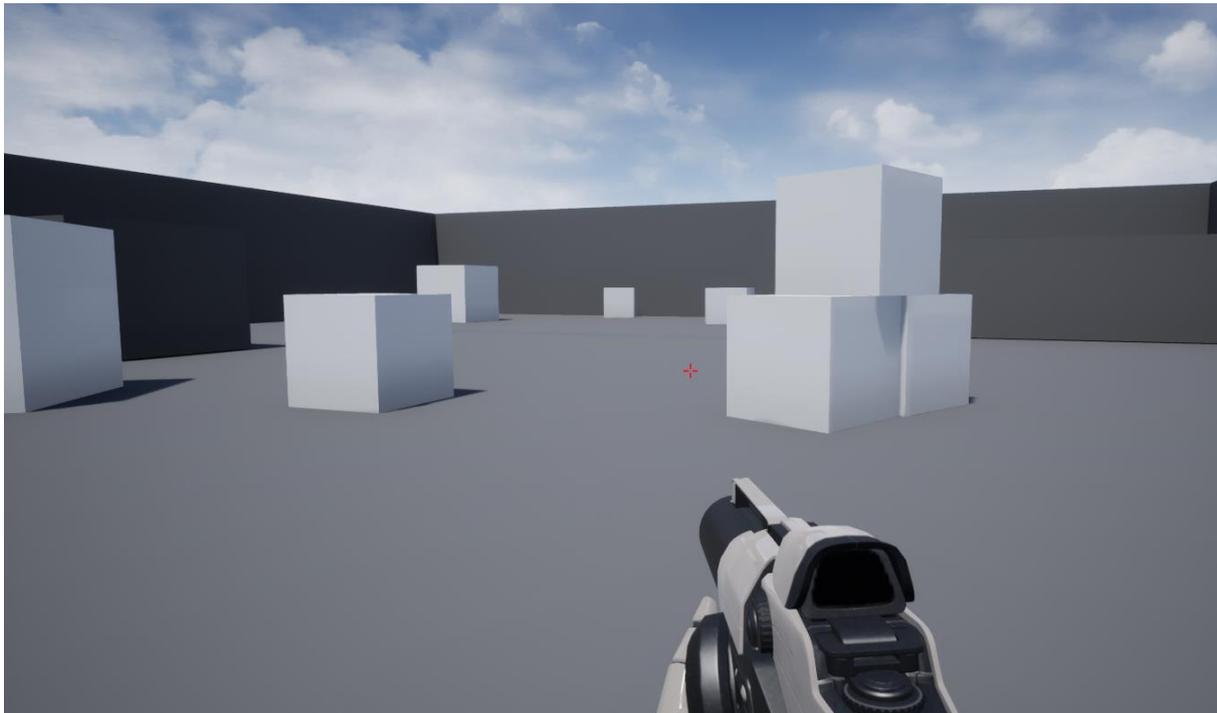


Abbildung 42 Unreal Engine Beispiel Spiel First Person Shooter

5.9.5 Programmablauf

In diesem Abschnitt werden Programmablauf und Informationsfluss dargestellt.

Inputloop: (Implementiert)

1. Erfassung Yaw der Füße. (Arduino Nano rechts und links)
2. Erfassung der Fussinformationen, ob diese abgestellt sind oder nicht. (Arduino Nano rechts und links)
3. Serielle Übertragung von Yaw und Fussinformationen an Teensy. (Arduino Nano rechts und links)
4. Linearschlittenposition von ODrive abfragen. (Teensy)
5. Yaw und virtuelle Positionsverschiebung berechnen (Teensy)
6. PC fragt Daten an. (PC / Spiel)
7. Serielle Übertragung von Yaw und Position an PC. (Teensy)
8. Virtuelle Position nullsetzen. (Teensy)
9. Umsetzung der Bewegungen in die VR Welt. (PC/ Spiel)

Feedbackloop / Aktor-Loop: (Noch nicht implementiert)

1. Interaktionen in VR Welt analysieren. (PC/ Spiel)
2. Force Feedback berechnen. (PC / Spiel)
3. Serielle Übertragung an Teensy. (PC / Spiel)
4. Fussinformationen und Bewegungsinformationen analysieren. (Arduino Mega)
5. Force Feedback berechnen (Teensy)
6. Serielle Übertragung von Force Befehlen an ODrive (Teensy)
7. Überwachung der Kraft auf den Fuss via Wägezelle (Teensy & Arduino Nano rechts und links)

6 Resultat

6.1 Stand der Entwicklung

Der Prototyp und die Demoumgebung wurden so weit fertiggestellt, dass das Konzept erfolgreich getestet werden konnte. Da der Code noch nicht optimiert wurde und der Microcontroller nur über begrenzte Leistung verfügt, läuft der Prototyp mit nur 11Hz Update Frequenz. Es darf angenommen werden, dass Hard- und Softwareoptimierungen diese um ein Vielfaches gesteigert werden könnte. Diese tiefe Update Frequenz bedeutet, dass bis zu 90ms Input lag vorhanden sind und die Bewegungen unsanft/leicht ruckelnd dargestellt werden. Trotzdem zeigte der Prototyp und die dafür programmierte Demoumgebung, den geplanten Funktionsumfang, um das Konzept zu testen.

In einem nächsten Schritt kann nun das aktive Force-Feedback implementiert werden. Dieses sollte für eine noch bessere Immersion sorgen. Force-Feedback wurde, soweit recherchiert, bisher noch nicht in VR Treadmills implementiert und ist in den meisten Systemen nicht implementierbar. Da sich diese Umsetzung als sehr komplex gezeigt hat, wurde diese Implementierung aus der Arbeit weggelassen, um den Rahmen nicht zu sprengen. Das Grundkonzept gegen Motion-Sickness, welches getestet werden sollte, funktioniert ebenfalls ohne Force-Feedback.

6.2 Ergebnisse von Tests mit Versuchspersonen

Zur Überprüfung der Funktion des Prototyps, wurden Tests mit insgesamt 8 Personen aus dem eigenen Umfeld gemacht. Die Tests wurden so ausgelegt, dass möglichst viele Erfahrungen mit dem System gesammelt werden konnten. Aus diesen Erkenntnissen lassen sich Stärken und Mängel im System ableiten. Dabei konnten auch viele Ideen für Weiterentwicklungen gesammelt werden.

Die Tests mit den Versuchspersonen sind positiv verlaufen. Bei jeder Versuchsperson wurde festgehalten, wie schnell sie sich an den Prototyp gewöhnt hat und darauf laufen konnte. Alle Probanden waren in der Lage, in unter 30 Sekunden zu laufen. Nach einer Testphase von mindestens 5 Minuten konnte bei keinem Probanden Übelkeit oder Schwindel festgestellt werden. Alle Probanden empfanden das Laufen als intuitiv, wobei die Illusion zu Laufen überzeugend wirkte.

Bei den Tests fiel auf, dass die Illusion gebrochen wird, wenn beide Füße abgestellt und parallel bewegt werden. Da die Laufbewegung den realen Boden als Referenz hat, bewegt man sich in diesem Fall unrealistisch schnell. Dieses Brechen der Illusion führte bei den Testpersonen zu Unwohlsein.

Dieser Sachverhalt lässt schlussfolgern, dass der Prototyp beim normalen Laufen eine genügend gute Illusion erzeugt, dass den Benutzern nicht übel wird. Damit hat das getestete Konzept bei den Testpersonen funktioniert.

Obwohl sich alle Testpersonen drehen und bewegen konnten, zeigte sich die Drehbewegung als noch nicht ganz ausgereift. Das Problem schien die Abweichung der Drehbewegung beim Laufen zu sein. Normalerweise wird aus der Hüfte gedreht und nicht aus den Knien wie es mit dem Prototyp gemacht werden muss. Dies macht das Drehen weniger intuitiv. Trotzdem konnten sich einige Testpersonen schnell an diese Bewegung gewöhnen und Bewegungsrichtung problemlos ändern.

Interessant zeigte sich auch bei der Drehbewegung, dass einige Personen diese Invertiert bevorzugten und andere normal. Wobei alle Testpersonen die Dreh- und Bewegungs-Rate für tief hielten. Wurden die bestehenden Dreh und Bewegungsinputs verstärkt, zeigte sich keine Beeinträchtigung der Immersion.

6.3 Technische Optimierungen für Weiterentwicklung

Die Ideen in diesem Abschnitt resultieren aus den Tests mit der Anlage. Diese wurden noch nicht genauer analysiert oder getestet. Sie dienen lediglich als Lösungsansatz für aufgetretene Probleme.

6.3.1 Drehbewegung

Die Drehbewegung zeigte sich als nicht intuitiv, sie kann jedoch auf weitere verschiedene Arten implementiert und verbessert werden.

- Momentan wird die Drehbewegung nur im abgesetzten Zustand verarbeitet. Diese könnte durch die invertierte Bewegung in der Luft erweitert werden. Die Drehbewegung würde dadurch beschleunigt und kontrollierter.
- Ein Kurvenlaufen könnte implementiert werden, wenn beide Füße bei einer Laufbewegung schräg gehalten werden. Dadurch müssten die Füße nicht andauernd gedreht werden und es würde automatisch eine Radius-Bewegung entstehen.
- Es könnte an Ort gedreht werden, indem beide Füße abgestellt und auf den Linearschienen antiparallel verschoben werden.

6.3.2 Lauf Unterstützung

Ein Proband empfand das Laufen auf dem Prototyp als anstrengend. Die Bewegung könnte erleichtert werden, indem das Momentum des Körpers mit den motorisierten Linearschienen simuliert wird. Die Füße würden dadurch beim Abstellen automatisch beschleunigt werden, wie es beim normalen Laufen der Fall ist.

6.3.3 Trägheit der Achsen

Bei Tests fiel auf, dass die Motoren mit ihrer grossen Rotationsträgheit, den Widerstand, der beim Beschleunigen der Füße auftritt, erhöhen. Dadurch fühlen sich die Füße beim Bewegen weniger frei an. Mögliche Lösungen wären:

- Eine leichtere Motorisierung zu wählen.
- Die Übersetzung zu ändern.
- Den Widerstand mit den verbauten Messzellen an den Riemen zu messen und mit den motorisierten Linearachsen zu kompensieren.

6.3.4 Referenz System

Bei gewissen Bewegungen zeigten sich Probleme im Messsystem. So bewegt man sich mit doppelter Fussgeschwindigkeit in der virtuellen Umgebung, wenn man beide Füße abstellt und parallel wegstösst und zu sich zieht. Das Problem ist auf die Verrechnung der beiden Achsen und die Referenz zum Boden zurückzuführen. Die Lösung ist, die Position und Bewegung der Füße relativ zueinander zu betrachten. Dabei würde sich eine solche Bewegung auflösen und mehr den physikalischen Gegebenheiten der Realität entsprechen.

6.3.5 Simulierung der Füße in VR

Die Tests mit den Probanden haben gezeigt, dass die Simulierung der Füße aus Orientierungsgründen hilfreich sein könnte. Da der Prototyp die Position der Füße genau erfasst, sollte dies leicht zu implementieren sein.

6.4 Technische Erweiterungen für Weiterentwicklung

Bei diesen technischen Erweiterungen geht es um Ideen, die mit den aktiven Linearachsen umgesetzt werden könnten. Diese Funktionen erhöhen die Interaktionsmöglichkeiten mit der virtuellen Umgebung. Die dadurch resultierende höhere Wechselwirkung zwischen Realität und virtueller Umgebung, lässt diese Welten mehr verschmelzen und sorgt für eine bessere Immersion.

6.4.1 Simulieren von virtuellen Objekten

Mit den Linearachsen können Kollisionen mit virtuellen Objekten simuliert werden. So wird der Fuss beispielweise gestoppt, sobald man gegen eine virtuelle Wand läuft. Ein anderes Beispiel wäre das Kicken eines virtuellen Balls, wobei die Linearachsen die Berührung simulieren.

6.4.2 Simulieren der Physik eines Bodens

Momentan können die Linearschlitten immer frei bewegt werden. Dies entspricht jedoch nicht der Physik eines Bodens, auf dem man steht. Sinnvoll könnte sein, die Trägheit des Körpers zu simulieren. Fängt man an zu laufen, muss man zuerst die simulierte Masse beschleunigen. Bei einem Stopp müsste man wieder bremsen, wie es in der Realität der Fall ist. Dieses Feature würde ebenfalls für eine flüssiger Laufbewegung sorgen.

Auch möglich wäre die Implementierung der Reibung eines Bodens. So können die Füße in der Luft einfach bewegt werden und erfahren mehr Widerstand, sobald sie auf den Platten abgestellt sind. Die Füße könnten beispielweise auch nicht mehr beide abgestellt und beliebig bewegt werden, was ebenfalls für realitätsnäheres Bewegungen sorgen würde.



Abbildung 43 Test mit Lukas



Abbildung 44 Test mit Thomas

6.5 Konklusion

Der Prototyp funktionierte auf Anhieb, wobei die Wirksamkeit des ausgearbeiteten Konzepts bei den Testpersonen bestätigt werden konnte. Obwohl der Fokus bei der Auslegung des Konzepts bei der Bekämpfung von Motion-Sickness lag, konnte das Konzept während der Arbeit immer weiter ausgebaut werden und zeigte viele zusätzliche interessante Aspekte und Möglichkeiten. Die Immersion, die mit dem Konzept auf einfache Weise erreicht werden kann und durch Systeme wie Force Feedback ausgebaut wird, macht das Konzept einzigartig. Momentan gibt es kein anderes bekanntes System auf dem Markt, das nur annähernd die Präzision oder Features unterstützen kann.

Beim Vorstellen des Projektes zeigte sich oft die Schwierigkeit ein solches System zu erklären. Personen die noch nie zuvor ein VR-System benutzt haben, verstehen meist das Bewegungsproblem nicht. Dies ist auf die Komplexität des Problems zurückzuführen, welches für uns ungewöhnlich wirkt, da es nur in Wechselwirkung mit einer virtuellen Umgebung auftritt. Da die Logik in dieser Arbeit auf diesem Problem aufbaut, ist das Verständnis dafür wichtig. Dies rüberzubringen wird leider nun noch schwieriger, da der Wettbewerb nur virtuell stattfindet. Um die Funktionsweise des Prototyps besser zeigen zu können, wurde ein Demonstrationsvideo gemacht:



<https://youtu.be/qZTvMpuDZHA>

7 Fazit

7.1 Zielüberprüfung

Das Ziel eine neuartige funktionierende VR-Treadmill zu entwickeln wurde erreicht. Auf dem Prototyp lässt sich angenehm gehen und alle Testpersonen konnten in kürzester Zeit darauf laufen. Der Prototyp ist kompakt und das Konzept lässt sich preiswert umsetzen, was das Konzept auch für den kommerziellen Heimanwender-Bereich interessant macht. Durch die genaue Bewegungsanalyse der Anlage, kann die Bewegung in VR realistisch nachgestellt werden, wobei eine überzeugende Illusion entsteht, wirklich zu laufen.

Ob mit dem Konzept Motion-Sickness komplett verhindert wird, konnte noch nicht evaluiert werden. Es wurden Tests mit verschiedenen Versuchspersonen gemacht. Dabei wurde eine Wirksamkeit gegen Motion-Sickness festgestellt.

Der Vergleich zwischen CAD und der fertigen Anlage zeigt, wie gut die Planung und Umsetzung in die Realität funktioniert hat. (Abbildung 46)



Abbildung 46 3D Modell aus CAD



Abbildung 46 Projektumsetzung

7.2 Verbesserungspotential

Bei erneuter Planung einer VR-Treadmill wäre es sinnvoll, diese aus verschiedenen Gründen ohne Motorisierung zu planen. Einerseits ist das haptische Feedback in Spielen schwierig zu integrieren und braucht viel Zeit. Des Weiteren würde es die Anlage und Dokumentation enorm vereinfachen. Zudem hätte das Projekt sehr wahrscheinlich nur ein Bruchteil gekostet. Es wäre auch viel einfacher gewesen, das Projekt termingerecht fertigzustellen, wobei es automatisch mit den meisten Spielen kompatibel gewesen wäre.

7.3 Organisieren der Teile

Die Beschaffung der nötigen Teile für dieses Projekt in der zur Verfügung stehenden kurzen Zeit, stellte eine Herausforderung dar. Es wurden verschiedene Verkaufsplattformen gewählt, um möglichst schnell und günstig die richtigen Komponenten zu beschaffen. Die ODrive-Platine musste in Amerika bestellt werden, da der Europäische Online-Shop zu diesem Zeitpunkt ausverkauft war. Aus Qualitätsgründen und Unterstützungszwecken wurde beim Original-Hersteller eingekauft, was kostspieliger war. Die Aluminium Profile wurden bei einem lokalen Händler eingekauft, um die Transportkosten zu sparen. Es war geplant, einen grossen Teil der Komponenten bei Bangood direkt aus China zu bestellen. Obwohl die Teile innert 24 Stunden verschifft hätten werden sollen, waren sie nach einer Woche immer noch nicht auf dem Weg. Darauf wurde die Bestellung storniert und auf der Plattform Aliexpress bestellt. Bis auf die Motoren trafen fast alle Teile von Aliexpress nach zwei bis drei Wochen ein. Diese gingen leider bei der DHL verloren. Daraufhin wurde entschieden, sofort nochmals zusätzliche Motoren zu bestellen, welche nach einigen Tagen ankamen. Diese stellten sich jedoch als qualitativ schlecht heraus, wodurch viel Zeit bei der Fehlersuche verloren ging. Sie erwiesen sich als untauglich, da beide Motoren einen Wackelkontakt in deren Wicklung aufwiesen. Fünf Bestellungen waren nötig, bis endlich brauchbare Motoren vorhanden waren.

Die Kosten des Projektes belaufen sich auf etwa SFr. 1'200.--. Dies entspricht der Obergrenze, welche für dieses Projekt gesteckt wurde. Das Projekt wird jedoch als wertvolle Umsetzung betrachtet und zeigt sich beim Abgleich von Aufwand und Ertrag als wertig.

7.4 Sicherheit

Bei der Konzeptionierung und Umsetzung der Anlage wurden viele Überlegungen im Bereich von möglichen Gefahren gemacht, welche eine solche Anlage für den Nutzer mit sich bringen kann. Die Motorisierung ist sicherlich der grösste Risikofaktor. Mit der Reduzierung auf 75 Watt wurde die Leistung so stark herabgesetzt, dass die Anlage als sicher eingestuft werden kann. Die Motoren können bei voller Leistung problemlos von Hand gestoppt werden.

7.5 Zufriedenheit

An diesem Projekt zu arbeiten hat mir persönlich sehr viel Spass gemacht. Mit der Breite, in der mich das Projekt forderte, arbeitete ich noch nie zuvor. Die Verbindung von Mensch, Physik, Hardware und Software war interessant und abwechslungsreich. Dabei war ein grosser Teil meiner Interessen und Fähigkeiten in dem Projekt gefordert. Ich musste neue Fähigkeiten, wie ein Spiel zu programmieren, lernen, um dieses Projekt zu können. Das Arbeiten mit VR hat mir ebenfalls Spass gemacht. Ich werde die Technologie sicherlich weiterverfolgen. Das Funktionieren des innovativen Konzepts stimmt mich zufrieden und stolz. Trotz des grossen Arbeitsaufwandes bin ich glücklich dieses Projekt gestartet und umgesetzt zu haben.

7.6 Ausblick

Die schnell wachsende VR-Community macht ein System welches Motion-Sickness zuverlässig umgeht sicherlich gefragt. Leider konnte bis zu diesem Zeitpunkt kein Unternehmen gefunden werden, mit dem ein Produkt entwickelt oder das Konzept weiterverfolgt werden könnte. Das Potenzial des Konzeptes bestätigte sich bei Tests mit dem Prototyp. Ich erhoffe mir dieses Projekt in einer Community weiter entwickeln zu können oder ein Unternehmen zu finden, das sich für die Idee interessiert.

8 Danksagung

An dieser Stelle möchte ich mich nochmals bei allen beteiligten Personen bedanken, die mich unterstützten. Ohne ihre Inputs und Unterstützung würde ich ein solches Projekt nicht umsetzen wollen. Ich möchte folgenden Personen speziell danken:

- Meinen Freunden, die ebenfalls großartige Ideen haben und mich immer wieder inspirieren und unterstützen.
- Meinem Experten Markus Zank, der sich Zeit für mich nahm und mich durch den Wettbewerb von SJF begleitete.
- Frau Jingming LI SALINA beim IGE für die zeitaufwändige Patentrecherche
- Bei Schweizer Jugend forscht, für die Unterstützung und Organisation des Wettbewerbes.
- Meiner Familie, die mich moralisch und bei der Korrektur der Arbeit unterstützen.
- Zudem bei all den Personen, die mich bei Laune hielten, an mich glaubten und das Arbeiten zu einem Vergnügen machten.

9 Eigenständigkeitserklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbstständig und ohne unerlaubte, fremde Hilfe erstellt habe und dass alle Quellen und Hilfsmittel wahrheitsgetreu verwendet und deklariert sind.

Rapperswil-Jona, 20. März 2021



Marvin Hotz

10 Verzeichnisse

10.1 Abbildungsverzeichnis

Abbildung 1 Oculus Quest VR Headset.....	8
Abbildung 2 Visualisierung Bewegungsproblem.....	9
Abbildung 3 Visualisierung Haptisches Feedback.....	11
Abbildung 4 Achsen Visualisierung.....	12
Abbildung 5 Laufanalyse Visualisierung.....	12
Abbildung 6 ODrive Steuerplatine.....	13
Abbildung 7 BLDC Motor mit Encoder für ODrive.....	13
Abbildung 8 Treadmill KAT Walk C.....	14
Abbildung 10B Cybershoes System mit Rollen.....	15
Abbildung 10A Cybershoes Paket.....	15
Abbildung 11 Bild des Aufbaus (von Vorne).....	17
Abbildung 12 Laufbewegung auf Anlage.....	17
Abbildung 14 Fuss in der Luft.....	18
Abbildung 14 Fuss bei Bodenkontakt.....	18
Abbildung 15 Bild des Aufbaus (von Hinten).....	19
Abbildung 16 Zeichnungsbereich Fusion 360.....	20
Abbildung 18A Selbstgebauter 3D Drucker.....	21
Abbildung 18B 3D Drucker Ender 3 V2.....	21
Abbildung 19 Einpressen von Kugellager.....	21
Abbildung 21A Aluminium Profil.....	22
Abbildung 21B V-Slot Rollen.....	22
Abbildung 22 Schnitt eines Schlittens.....	22
Abbildung 23 Linearschlitten.....	23
Abbildung 25A Riemensystem Antriebsseite.....	24
Abbildung 25B Riemensystem mit 3D-gedrucktem Pully.....	24
Abbildung 26 Übertragung mit Riemensystem.....	25
Abbildung 28A Kraftmessung V1.....	25
Abbildung 28B Kraftmessung V2.....	25
Abbildung 29 Haptisches Force Feedback.....	26
Abbildung 31A Untere Position Gelenkarm.....	27
Abbildung 31B Obere Position Gelenkarm.....	27
Abbildung 32 Verbinder zwischen Armen.....	27
Abbildung 35B Fusshalterung / Fussverbinder.....	28
Abbildung 35C Aufrauen der Oberfläche für Klebevorgang.....	28
Abbildung 35A Sandalen von Adidas.....	28
Abbildung 36 Schnitt vordere Fussplatte mit Spannmehanismus.....	29
Abbildung 37 Abrollen des Fusses.....	29
Abbildung 38 Elektronik in der hinteren Fussplatte.....	29
Abbildung 39 Teensy 3.2.....	30
Abbildung 40 ArduinoMega oberhalb von ODrive.....	30
Abbildung 41 Schema.....	31
Abbildung 42Unreal Engine Beispiel Spiel First Person Shooter.....	33
Abbildung 43 Test mit Lukas.....	37
Abbildung 44Test mit Thomas.....	37
Abbildung 46 3D Modell aus CAD.....	39
Abbildung 46 Projektumsetzung.....	39

10.2 Literaturverzeichnis

- [1] C. Berger, «VR Nerds \ Was hilft gegen Motion Sickness in VR?,» VR Nerds GmbH, 20 06 2016. [Online]. Available: <https://www.vrnerds.de/was-hilft-gegen-motion-sickness-in-vr/>. [Zugriff am 28 10 2020].
- [2] omnia360 GmbH, «OMNIA 360 \ Motin Sickness: 6 Tipps, damit VR nicht zum Kotzen wird,» omnia360 GmbH, 07 10 2019. [Online]. Available: <https://omnia360.de/blog/motion-sickness-in-virtual-reality/>. [Zugriff am 25 10 2020].
- [3] F. Koslucher, E. Haaland, A. Malsch, J. Webeler und T. A. S. regen, «Sex Diff erences in the Incidence of Motion Sickness Induced by Linear Visual Oscillation,» Aerospace Medical Association, University of Minnesota, Minneapolis, MN, USA, 2015.
- [4] L. Jiwon, K. Mingyu und K. Jinmo, «A Study on Immersion and VR Sickness in Walking,» MDPI Basel, Department of Software, Catholic University of Pusan, Busan 46252, Korea, 2017.
- [5] T. Dixon, «iFixit \ Good Vibrations: How Apple Dominates te Touch Feedback Game,» 20 5 2019. [Online]. Available: <https://de.ifixit.com/News/16768/apple-taptic-engine-haptic-feedback>. [Zugriff am 24 10 2020].
- [6] haptix, «haptix \ Industrial-grade haptic technology,» HaptX Inc, [Online]. Available: <https://haptx.com/technology/>. [Zugriff am 28 10 2020].
- [7] S. Loveridge, «gamesradar \ What is haptic vedback and how will it change the way we paly?,» Future US Inc, 17 3 2020. [Online]. Available: <https://www.gamesradar.com/haptic-feedback-explained/>. [Zugriff am 28 10 2020].
- [8] O. Weigl, «ODriverobotics \ Hobby Motors For Robotics,» ODriverobotics, [Online]. Available: <https://odriverobotics.com/>. [Zugriff am 28 10 2020].
- [9] Kikstarter, «Kickstarter \ Search \ VR Treadmill,» [Online]. Available: https://www.kickstarter.com/discover/advanced?ref=nav_search&term=vr%20treadmill. [Zugriff am 23 10 2020].
- [10] Cybershoes GmbH, «Cybershoes Businessplan,» [Online]. Available: https://wordpress-upload-conda-bucket-xbg8r4800mm5.s3.eu-central-1.amazonaws.com/uploads/2019/08/business_plan-cybershoes.pdf. [Zugriff am 28 10 2020].
- [11] Nathie, «YouTube \ 6:05 This VR Treadmill Bring Us Closer To Making Ready Player One A Reality (First Look: Kat Walk C),» Nathie, 23 06 2020. [Online]. Available: <https://youtu.be/jABB85ZTcF8?t=365>. [Zugriff am 28 10 2020].
- [12] W. Alex, Interviewee, *Auslandbeauftragter Manager bei KATVR*. [Interview]. 27 10 2020.
- [13] Conrad, «Conrad \ Cybershoes,» [Online]. Available: https://www.conrad.ch/de/p/cybershoes-2000-l-virtual-reality-bewegungsunterstuetzung-schwarz-2299518.html?gclid=CjwKCAjw0On8BRAGeIwAincsHO2A7S-YitoNwFR6bL_mSt6WF1AaCw7ILGDzkTIFcYWAb0G7Nh-yQxoCVcsQAvD_BwE&utm_source=google-shopping-de&utm_medium=search&utm. [Zugriff am 28 10 2020].

- [14] INFINADECK, «INFINADECK \ Move beyond,» [Online]. Available: <https://infinadeck.com/>. [Zugriff am 28 10 2020].
- [15] Arduino, «Arduino \ What is Arduino?,» 05 02 2018. [Online]. Available: <https://www.arduino.cc/en/Guide/Introduction>. [Zugriff am 26 10 2020].
- [16] Arduino, «Arduino \ SoftwareSerial Library,» 24 12 2019. [Online]. Available: <https://www.arduino.cc/en/Reference/SoftwareSerial>. [Zugriff am 10 27 2020].
- [17] T. Grohgan, «VR Nerds \ Sony: Neues VR-System gegen Motion Sickness patentiert,» VR Nerds gmbH, 19 09 2018. [Online]. Available: <https://www.vrnerds.de/sony-neues-vr-system-gegen-motion-sickness-patentiert/>. [Zugriff am 28 10 2020].

11 Anhang

- Vergleich mit bestehenden Systemen
- Testversuche
- Elektroschema
- Code
- Datenblätter / Spec-sheets

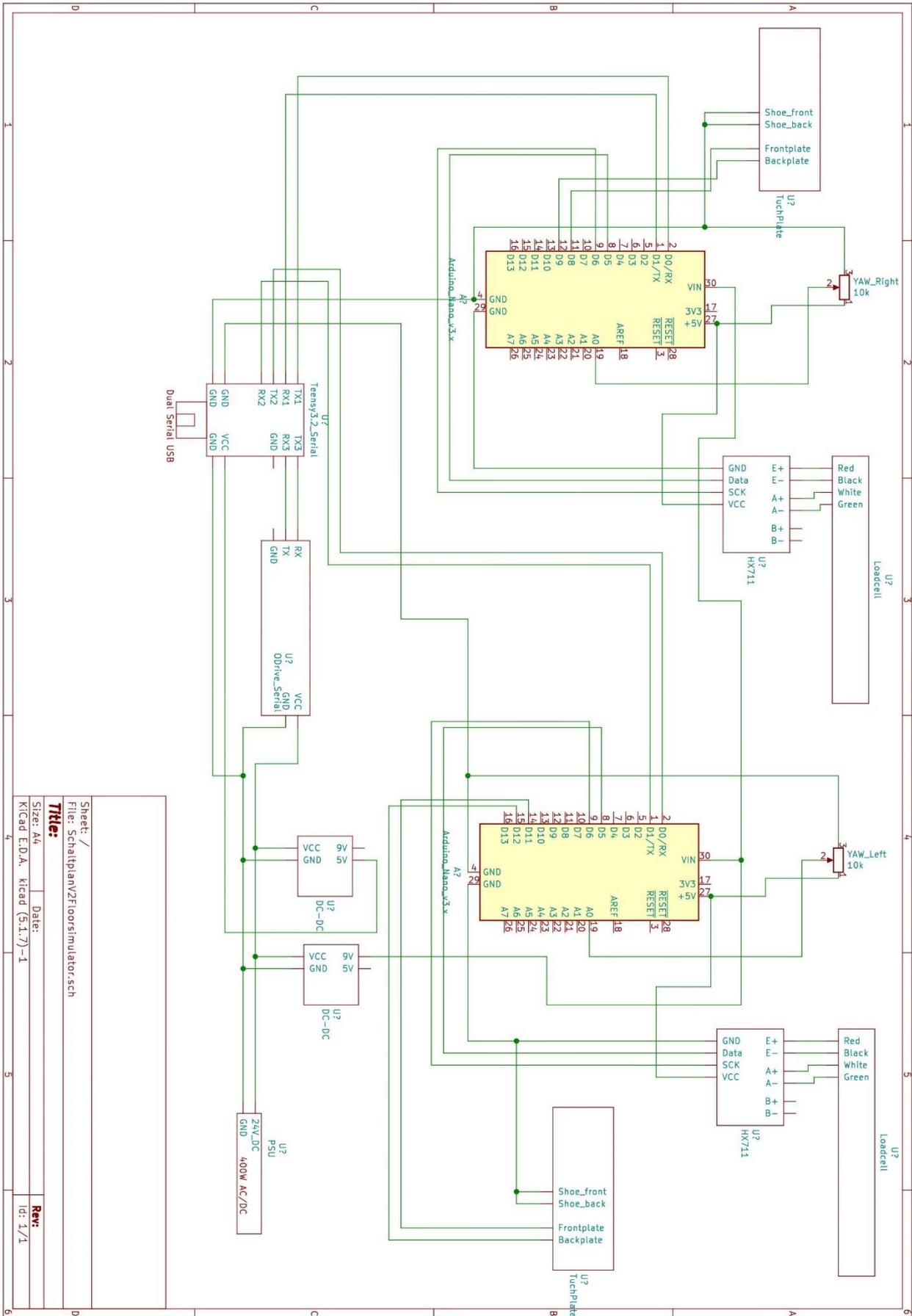
(Folgt auf folgenden Seiten)

Eigenschaft\System	Rutsch-Treadmill	Cybershoes	Entwickelte VR Treadmill
Preis	>CHF 1500	CHF 400	CHF 1200 (Entwicklung) Konzept unter CHF 500 umsetzbar
Komplexität	Mittel	Tief	Mittel
Grösse	Gross	Mittel	Mittel
Force Feedback	Nicht möglich	Nicht möglich	Möglich
Tracking Genauigkeit	Mittel	Tief	Hoch

11.1 Versuchsergebnisse mit Probanden

- a. Überprüfung auf Motion-Sickness nach <5min
 - b. Anfälligkeit auf Motion-Sickness 1-10
1 = Noch nie vorgekommen
10 = kommt häufig bei vielen Tätigkeiten vor
 - c. Zeit benötigt, um Laufbewegung zu lernen / machen können
 - d. Verhalten von Drehbewegung
 - e. Vorschläge / Anmerkungen
-
1. Marvin Hotz
 - a. Keine Anzeichen
 - b. 8 (Auto, Bus, Videogames, Skifahren)
 - c. <20s
 - d. Nach einigen Minuten, machbar nicht intuitiv (normal)
 - e. -
 2. Lukas Reitemeier
 - a. Keine Anzeichen
 - b. 3 (Fast nie, VR)
 - c. <20s
 - d. Nach einigen Minuten, machbar nicht intuitiv (normal / invertiert)
Beim Stillstehen invertiert besser, beim Laufen normal besser
 - e. Rotationsträgheit
Kabelführung
Pfeil / Problem wohin laufe ich?
Beide Füße gleichzeitig am Boden bewegen bricht Illusion
 3. Patrick Haas
 - a. Leichte Kopfschmerzen (allgemeine Nebenwirkung von VR?)
 - b. 8 (Oft, Schaukel)
 - c. <20s
 - d. Etwas mühe, Drehen im Stillstand (normal)
 - e. Füße in VR anzeigen / simulieren
Übelkeit bei gleichzeitigem Bewegen von Füßen am Boden
Bewegung Füße Relativ zueinander?
 4. Marcel Hotz
 - a. Keine Anzeichen
 - b. 4 (Selten)
 - c. <20s
 - d. Drehbewegung hauptsächlich im Stillstehen (Invertiert)
 - e. Laufbewegung verstärken / multiplizieren
Drehbewegung verstärken / multiplizieren

5. Belinda Hotz
 - a. Keine Anzeichen
 - b. 3 (fast nie)
 - c. <20s
 - d. Drehbewegung funktionierte auch im Laufen (invertiert)
 - e. Drehbewegung auch in der Luft?
Füsse in VR anzeigen
6. Fabian Steiner
 - a. Keine Anzeichen
 - b. 3 (fast nie)
 - c. <20s
 - d. Nach einigen Minuten, nicht intuitiv (normal)
 - e. Füsse in VR anzeigen
7. Mischa Wolf
 - a. Keine Anzeichen
 - b. 2 (nie)
 - c. <20s
 - d. Nach kurzer Zeit (normal / invertiert)
 - e. Drehbewegung optimieren
8. Thomas Wyser
 - a. Keine Anzeichen
 - b. 2 (fast nie)
 - c. <20s
 - d. Drehbewegung schnell gelernt (invertiert)
 - e. Füsse schräg, um Kurve zu laufen
Füsse in VR anzeigen / simulieren
anstrengend
Taktfrequenz zu gering



Sheet: /
 File: SchatplanV2Floorsimulator.sch
Title:
 Size: A4
 Date: / /
 Kicad E.D.A. kicad (5.1.7)-1
Rev:
 Id: 1/1


```

45
46 // Teensy 3 and 4 (all versions) - Serial1
47 // pin 0: RX - connect to ODrive TX
48 // pin 1: TX - connect to ODrive RX
49 // See https://www.pjrc.com/teensy/td_uart.html for other options on Teensy
50 HardwareSerial& odrive_serial = Serial1;
51
52
53
54
55 // ODrive object
56 ODriveArduino odrive(odrive_serial);
57
58
59
60 //Timer object
61 IntervalTimer CalcTimer;
62 IntervalTimer USB_Serial2_Timer;
63
64 //<<<<<<<< Variablen Namen-Management (von oben nach unten immer mit _
    dazwischen) >>>>>>>>
65 //G --> Global
66 //K --> Konstante
67 //U --> Unsigned (Nur positive zahlen!) (mehr platz im positiven
    zahlenbereich)
68 //DV --> defined Value
69 //DP --> defined Pin
70 //
71 //i --> Int (-32768 bis 32767)
72 //f --> Float (3.4E+38 bis -3.4E+38)
73 //l --> Long (-2147483648 bis 2147483647)
74 //c --> char (-128 bis 127)(Zeichen)
75 //S --> string (kette von char's)
76 //by --> Byte (0 bis 255)
77 //bo --> Bool (0 bis 1)
78
79
80
81 //Inputs right foot (Serial2)
82 bool R_FrontTuch = 1;
83 bool R_BackTuch = 1;
84 int R_Angle = 512;
85 bool R_Force = 0;
86
87
88 //Inputs left foot (Serial3)
89 bool L_FrontTuch = 1;
90 bool L_BackTuch = 1;
91 int L_Angle = 512;
92 bool L_Force = 0;
93
94

```

```
95 //Bewegungsvariablen
96 int move = 0;
97 int turn = 0;
98 bool jump = 0;
99
100
101 //CodeManagement Variablen
102 bool SerialOUT_for_debugging = false;
103
104
105 //IN OUTput Management
106 #define onboardLED 13
107
108
109
110
111 void setup() {
112
113     //Setting Pins
114     pinMode(onboardLED, OUTPUT);
115
116     // ODrive uses 115200 baud
117     odrive_serial.begin(115200);
118     odrive_serial.setTimeout(8);
119
120     // Serial to PC
121     Serial.begin(115200);
122     while (!Serial) // wait for Arduino Serial Monitor to open
123     {
124         delay(30);
125         toggleLED();
126     }
127
128     //Serial info
129     //Serial.setTimeout(8);
130     Serial.println("USB-Serial1 started\nEnter help for info");
131
132     //
133     SendUSB2("USB-Serial2 started\nEnter T for turnrate\n");
134
135
136     //Serial to right foot
137     Serial2.begin(115200);
138     Serial2.setTimeout(12);
139
140     //Serial to left foot
141     Serial3.begin(115200);
142     Serial3.setTimeout(12);
143
144
145
146
```

```
147
148 //Info für Serielle "Comandline" senden
149 if (Send_SerialInfo_in_startup) Send_Serial_Info();
150
151
152
153
154
155
156 //Setup von ODrive settings
157 for (int axis = 0; axis < 2; ++axis) {
158     odrive_serial << "w axis" << axis << ".controller.config.vel_limit " << 10.0f << '\n'; //ODrive set vel limit
159     odrive_serial << "w axis" << axis << ".motor.config.current_lim " << 3.0f << '\n'; //ODrive set current limit
160 }
161
162 //Setting Interrrupts
163 CalcTimer.begin(read_and_calc_move, 22000); //Timerinterrupt der jede 10ms abgerufen wird um bewegungen zu brerrechnen
164 USB_Serial2_Timer.begin(USB_Serial2_Abfrage, 1000); //Abfragefrequenz von USB-Serial2 1khz
165
166
167 }
168
169
170
171 void loop() {
172     //blinkLED(100); //Heartbeat
173     //delay(10);
174
175     //LOL
176 }
177
178
179
180 void read_and_calc_move()
181 {
182     read_and_calc_move();
183     //Auslesen und berrechnen des rechten Fusses
184     static int last_pos_R = 0;
185     static bool last_tuch_R = false;
186
187     int odrive_readval_R = read_ODrive_pos(0);
188
189     if (!R_FrontTuch || !R_BackTuch) //Auslesen ob Fuss Boden berührt
190     {
191         if (last_tuch_R) move = move + (last_pos_R - odrive_readval_R); //Bewegte Strecke zu move rechnen (nur bei Bodenkontakt)
192         last_pos_R = odrive_readval_R;
193         last_tuch_R = true;
194     }
```



```

...e\Dokumente\Arduino\FloorSimulatorV5\FloorSimulatorV5.ino 6
244
245 }
246
247
248
249 #ifndef USB_DUAL_SERIAL
250 void SendUSB2(String USB2_TX)          //Code um mit dem 2en Seriellen USB Port ↗
    zu kommunizieren
251 {
252     byte USB_Serial_TX_length = USB2_TX.length();
253
254     for (int i = 0; i < USB_Serial_TX_length; i++)
255     {
256         usb_serial2_putchar(USB2_TX[i]);
257     }
258
259     usb_serial2_flush_output();
260 }
261
262
263
264 void SendUSB1(String USB2_TX)          //Code um mit dem 2en Seriellen USB Port ↗
    zu kommunizieren
265 {
266     byte USB_Serial_TX_length = USB2_TX.length();
267
268     for (int i = 0; i < USB_Serial_TX_length; i++)
269     {
270         usb_serial_putchar(USB2_TX[i]);
271     }
272
273     usb_serial_flush_output();
274
275 }
276
277
278
279 #endif // USB_DUAL_SERIAL
280
281
282
283 #define AnzahlMarker 5
284 char Marker[AnzahlMarker] = { 'F','B','K','P','X' }; //Die einzelnen Zeichen ↗
    nach denen der Parser sucht
285 int ReceiveVal[AnzahlMarker];
286 bool Receive[AnzahlMarker];
287
288
289 void serialEvent()
290 {
291     toggleLED();          //Toggle LED to show alive
292

```

```

...e\Dokumente\Arduino\FloorSimulatorV5\FloorSimulatorV5.ino 7
293 //String inPCString = Serial.readString(); //liest String von PC und
    schreibt sie in var
294 //char inPC = inPCString[0];
295
296 char inPC = Serial.read();
297
298 //Senden der Inputs an PC nachdem diese Abgefragt werden
299 if (inPC == 'M')
300 {
301     Serial.println(move);
302     move = 0;
303 }
304
305 if (inPC == 'T')
306 {
307     if (turn <= 2 && turn >= -2) turn = 0; //Deadband
308     Serial.println(turn);
309     turn = 0;
310 }
311
312
313 if (inPC == 'b') read_and_S_print_ODrive_busvoltage();
314 if (inPC == 'p') read_and_S_print_ODrive_position();
315 if (inPC == 's') { Serial.println(read_ODrive_speed(0)); Serial.println
    (read_ODrive_speed(1)); }
316 if (inPC == 'D') { SerialOUT_for_debugging = true; Serial.println("Debuging
    enabled"); }
317
318 /*
319 if (inPCString == "help") Send_Serial_Info();
320
321 if (inPCString == "Reboot")
322 {
323     odrive_serial << "sr";
324     Serial.println("Rebooting ODrive\nThis can take a second");
325 }
326 */
327 Serial.flush();
328 }
329
330
331
332
333 void Send_Serial_Info()
334 {
335     Serial.println("Send the character 'M' to read movment output");
336     Serial.println("Send the character 'T' to read turn output");
337
338     Serial.println("Send the character 'C' to calibrate and start the Motors");
339     Serial.println("Send the character 's' to read Axis speed");
340     Serial.println("Send the character 'b' to read bus voltage");
341     Serial.println("Send the character 'p' to read Axis Position");

```



```
486
487 //ODrive gibt Pos und Speed in Umdrehungen an
488 //1000mm = 13.9 Umdrehungen
489 int read_ODrive_speed(byte Motornum)
490 {
491     odrive_serial << "r axis" << Motornum << ".encoder.vel_estimate\n";
492     return 100 * odrive.readFloat();
493 }
494 }
495
496 int read_ODrive_pos(byte Motornum)
497 {
498     odrive_serial << "r axis" << Motornum << ".encoder.pos_estimate\n";
499     return (1000 * odrive.readFloat()) / -13.9;
500     odrive_serial.flush();
501 }
502
503 void calibration_sequence()
504 {
505
506     int motornum = 0;
507     int requested_state;
508
509     requested_state = ODriveArduino::AXIS_STATE_MOTOR_CALIBRATION;
510     Serial << "Axis" << 0 << ": Requesting state " << requested_state << '\n';
511     if (!odrive.run_state(motornum, requested_state, true)) return;
512
513     requested_state = ODriveArduino::AXIS_STATE_ENCODER_OFFSET_CALIBRATION;
514     Serial << "Axis" << 0 << ": Requesting state " << requested_state << '\n';
515     if (!odrive.run_state(motornum, requested_state, true, 25.0f)) return;
516
517     requested_state = ODriveArduino::AXIS_STATE_CLOSED_LOOP_CONTROL;
518     Serial << "Axis" << 0 << ": Requesting state " << requested_state << '\n';
519     if (!odrive.run_state(motornum, requested_state, false /*don't wait*/))
520         return;
521
522     motornum = 1;
523
524     requested_state = ODriveArduino::AXIS_STATE_MOTOR_CALIBRATION;
525     Serial << "Axis" << 1 << ": Requesting state " << requested_state << '\n';
526     if (!odrive.run_state(motornum, requested_state, true)) return;
527
528     requested_state = ODriveArduino::AXIS_STATE_ENCODER_OFFSET_CALIBRATION;
529     Serial << "Axis" << 1 << ": Requesting state " << requested_state << '\n';
530     if (!odrive.run_state(motornum, requested_state, true, 25.0f)) return;
531
532     requested_state = ODriveArduino::AXIS_STATE_CLOSED_LOOP_CONTROL;
533     Serial << "Axis" << 1 << ": Requesting state " << requested_state << '\n';
534     if (!odrive.run_state(motornum, requested_state, false /*don't wait*/))
535         return;
536 }
```


11.3 Arduino Nano Code für Fusschlitten

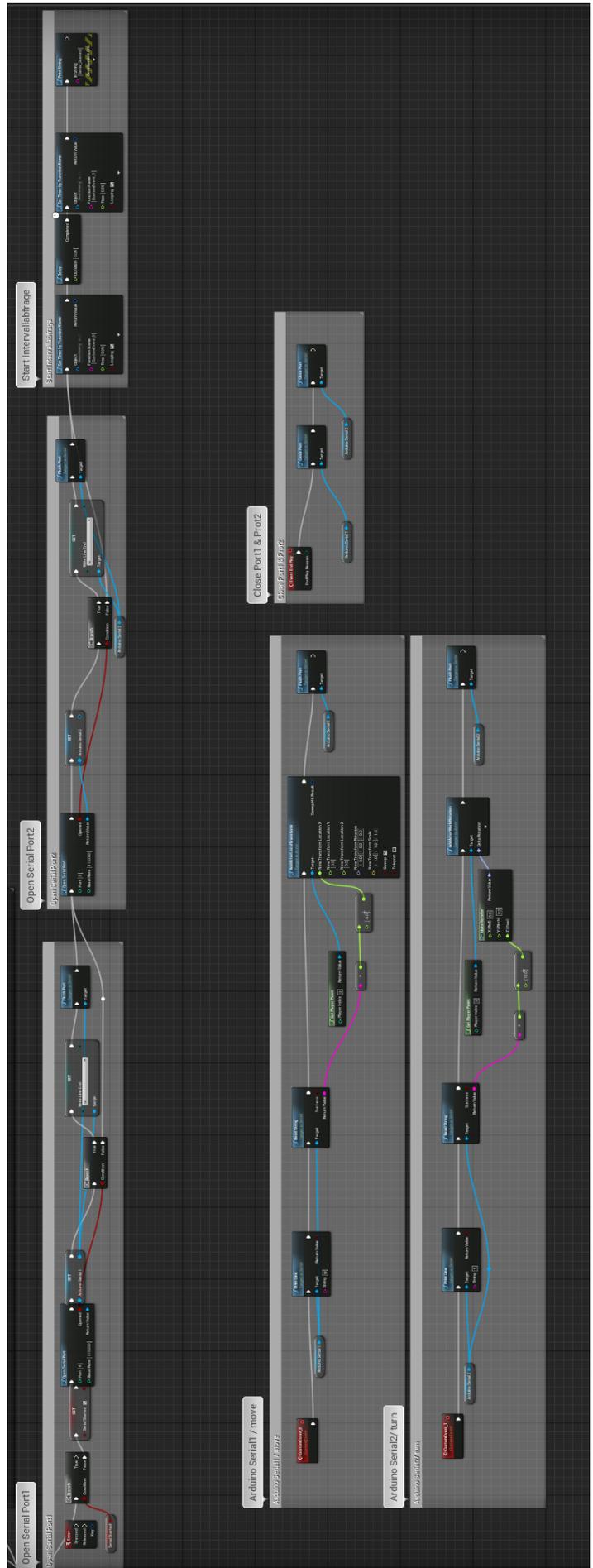
```
C:\Users\marvin\OneDrive\Dokumente\Arduino\FussCode\FussCode.ino 1
1 // Visual Micro is in vMicro>General>Tutorial Mode
2 //
3 /*
4     Name:      FussCode.ino
5     Created:   03.01.2021 22:07:50
6     Author:    ARBEITSPC\marvin
7 */
8
9 #include "HX711.h"
10
11 HX711 scale;
12
13
14 #define Smooth 15
15 uint8_t dataPin = 5;
16 uint8_t clockPin = 6;
17 float force[Smooth];
18 int counter = 0;
19 bool loadcell = 0;
20
21 bool wait = 1;
22 bool debug = 0;
23 bool offset = 0;
24
25 void setup()
26 {
27     Serial.begin(115200);
28
29     scale.begin(dataPin, clockPin);
30
31     //Fussplattensensoren initialisieren
32     pinMode(11, INPUT_PULLUP);
33     pinMode(12, INPUT_PULLUP);
34
35     // loadcell factor 20 KG
36     scale.set_scale(127.15);
37     scale.tare();
38
39     //Array leeren
40     while (counter <= (Smooth - 1))
41     {
42         force[counter] = 0; counter++;
43     }
44     counter = 0;
45 }
46
47
48 // Add the main program code into the continuous loop() function
49 void loop()
50 {
51     //Drehpotentiometer auslesen
52     Serial.print("X");
```

```
53     Serial.println(analogRead(0));
54
55
56     //Fussplatten Auslesen
57     Serial.print("F");
58     Serial.println(digitalRead(11));
59     Serial.print("B");
60     Serial.println(digitalRead(12));
61
62
63     //Force / Scale auslesen (Wägezelle)
64     if (loadcell)
65     {
66         if (debug) Serial.println("Loadcell lesen = "); //debug
67
68         //Werte lesen und in array schreiben
69         force[counter] = scale.get_units();
70
71         if (debug) Serial.println(force[counter]); //debug
72
73         //Durchschnitt bilden
74         float summ = 0;
75         for (byte i = 0; i < Smooth; i++) { summ = summ + force[i]; }
76         summ = summ / Smooth;
77
78         //Werte senden
79         Serial.print("K");
80         Serial.println(summ);
81
82         if (offset) //Falls Angefragt Offsetval senden
83         {
84             if (debug) Serial.println("Offsetval"); //debug
85             Serial.print("P");
86             Serial.println(summ+10000);
87
88         }
89
90         //Counter management
91         counter++;
92         if (counter == Smooth)counter = 0;
93     }
94
95     //Delay
96     if (wait)
97     {
98         delay(20);
99         if (debug) Serial.println("Mit Delay");
100    }
101 }
102
103 void serialEvent()
104
```

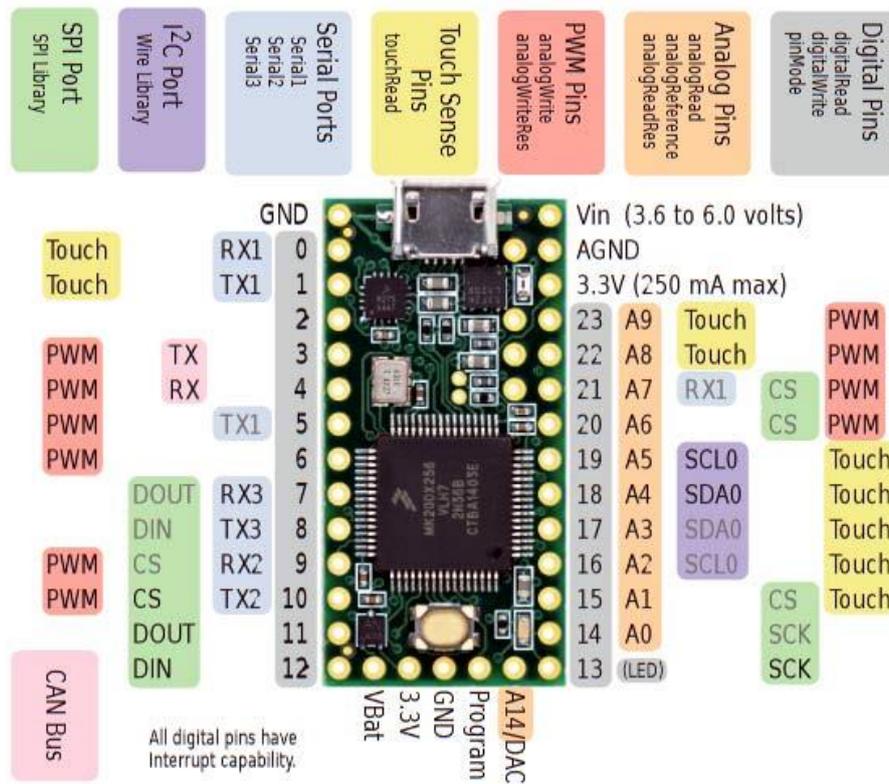
```
105 {
106   char inchar = Serial.read();
107   if (inchar == '1') scale.tare(); //loadcell null setzen
108   if (inchar == '2') loadcell = 1; //loadcell einstellen
109   if (inchar == '3') loadcell = 0; //loadcell ausstellen
110   if (inchar == '4') wait = 1; //delay ein ausstellen
111   if (inchar == '5') wait = 0;
112   if (inchar == '6') debug = 1; //debugging einstellen
113   if (inchar == '7') debug = 0;
114   if (inchar == '8') offset = 1;
115 }
```

11.4 Unreal Engine Pawn Code

Dies ist der Code, der die Verbindung mit dem Teensy herstellt und die nötigen Bewegungsinformationen einholt und umsetzt. Der Code wurde Blueprint-Format von Unreal Engine selbst im geschrieben.



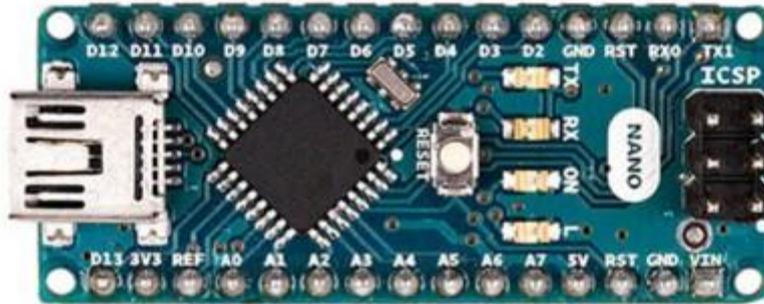
11.5 Teensy 3.2



Microcontroller	MK20DX256
Operating voltage	3.3V
Recommended input voltage for V_{IN} pin	3.6V to 6.0V
Analog inputs	21
Digital I/O pins	34
DC source/sink from I/O pins	10mA
Flash memory	256KB
SRAM	64KB
EEPROM	2KB
Frequency (clock speed)	72MHz
Communication	I ² C, SPI, UART, CAN, USB

<https://www.pjrc.com/store/teensy32.html> (abgerufen am 17.03.2021)

11.6 Arduino Nano



Microcontroller	ATmega328
Architecture	AVR
Operating Voltage	5 V
Flash Memory	32 KB of which 2 KB used by bootloader
SRAM	2 KB
Clock Speed	16 MHz
Analog IN Pins	8
EEPROM	1 KB
DC Current per I/O Pins	40 mA (I/O Pins)
Input Voltage	7-12 V
Digital I/O Pins	22 (6 of which are PWM)

<https://store.arduino.cc/arduino-nano> (Abgerufen am 17.03.2021)

11.7 Verbaute Encoder (CUI AMT102-V)



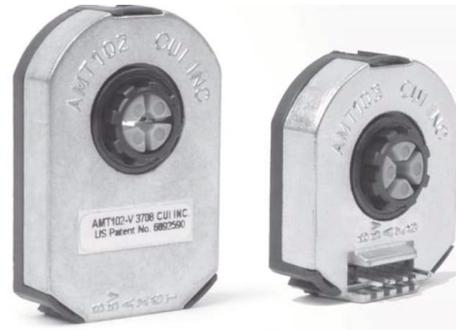
date 11/2010
page 1 of 8

SERIES: AMT10X

DESCRIPTION: INCREMENTAL ENCODER

FEATURES

- patented capacitive technology
- 16 dip switch programmable resolutions
- straight (radial) and right-angle (axial) versions
- 9 different mounting hole options for AMT102
- 6 different mounting hole options for AMT103
- low power consumption
- line driver output available (CUI-10XE-10)
- -40°C to 100°C operating temperature
- TTL voltage output
- modular package



ELECTRICAL SPECIFICATIONS

output phase difference	90° (A ch leads B ch in CW direction viewed from front)
output current	5 mA max.
output waveform	TTL voltage square wave
output signals	A, B, Z' phase (\bar{A} , \bar{B} , \bar{C} line driver available with CUI-10XE-10)
current consumption	6 mA typ., 10 mA max.
supply voltage	3.6 ~ 5.5 V dc
output resolution (ppr) ²	48, 96, 100, 125, 192, 200, 250, 256, 384, 400, 500, 512, 800, 1000, 1024, 2048
frequency response	250 kHz max.
accuracy ³	±15 arcmin (at 192, 384, 400, 500, 800, 1000, 1024, 2048 ppr) ±30 arcmin (at 96, 200, 250, 512 ppr) ±60 arcmin (at 48, 100, 125, 256 ppr)
max. rotational speed	7500 rpm (at 2048, 1000, 800, 384 ppr) 15000 rpm (at 1024, 500, 400, 192 ppr) 30000 rpm (at 48, 96, 100, 125, 200, 250, 256, 512 ppr)

1. Some stepper motors may leak a magnetic field causing the AMT index pulse to not function properly.
2. All resolutions stated are before quadrature decoding. (example: 1000 ppr x 4 = 4000 counts)
3. Based on full production testing standards and includes all electronic and mechanical based errors, not a computed estimate.

MECHANICAL SPECIFICATIONS

mounting hole options		A) 2 each M1.6 holes on 16 mm (0.63") bolt circle B) 2 each #4 holes on 19.05 mm (0.75") bolt circle C) 2 each M1.6 or M2 holes on 20 mm (0.787") bolt circle D) 3 each M1.6 or M2 holes on 20.9 mm (0.823") bolt circle E) 3 each M1.6 or M2 holes on 22 mm (0.866") bolt circle F) 4 each M1.6 or M2 holes on 25.4 mm (1") bolt circle G) 2 each #4 holes on 15.75 mm (0.62") x 20.96 (0.825") hole layout* H) 2 each #4 holes on 32.43 mm (1.277") bolt circle* I) 2 each #4 holes on 46.03 mm (1.812") bolt circle*
weight	AMT102 AMT103	20.5 g 14.0 g
vibration proof		50 m/s ² , 10 ~ 200 Hz, 2 hours each on XYZ
shock resistance		147 m/s ² , 11 ms, 2 hours each on XYZ

20050 SW 112th Ave. Tualatin, Oregon 97062 phone 503.612.2300 fax 503.612.2380 www.cui.com