

Technische Universität Berlin
Institut für Werkzeugmaschinen und Fabrikbetrieb
Industrielle Automatisierungstechnik
- Automatisierungstechnisches Projekt -



ABSCHLUSSBERICHT

Automatisierter Air-Hockey-Spieler mit modularem Aufbau

23. September 2019

Gruppenmitglieder: EG
CH
SK
DM
JN
HV

Gruppenleiter: SK

Betreuer: CN
AV

Danksagung

Wir bedanken uns insbesondere bei unserem Projektbetreuer, Herrn C.N., welcher uns bei der Umsetzung des Projekts sehr helfen konnte und mit dessen Unterstützung wir die Aufgabenstellung erfolgreich bewältigen konnten. In diesem Zusammenhang sind die regelmäßigen Treffen während des Semesters zu nennen, bei welchen aktuelle Fragestellungen gut geklärt und verschiedene Ideen produktiv beleuchtet werden konnten. Trotz dieses regelmäßigen Austauschs wurde uns insgesamt ein sehr großes Vertrauen in die eigene Gruppenarbeit entgegengebracht und viel Spielraum für eigene Gestaltungen gelassen. Dadurch ist ein großes Verantwortungsgefühl innerhalb der Gruppe entstanden, welches die gesamte Gruppe motivierte.

Des Weiteren danken wir Herrn Axel Vick, der uns im Zuge der Zwischenpräsentation durch besonders konstruktive Kritik und mit guten Ideen aus einem anderen Blickwinkel zur weiteren Optimierung des Konzeptes verhelfen konnte.

Ein herzlicher Dank gilt auch an der Studentischen Selbstverwaltung des Studentenwohnheims Siegmunds Hof (WSH) in Berlin Tiergarten und ihrem stellvertretenden Vorsitzenden, der es uns ermöglicht hat, deren Räumlichkeiten für unsere Projekttreffen zu nutzen. So war insgesamt eine professionelle Projektarbeit möglich.

Abstract

Dieses Projekts widmet sich der Konzipierung und Umsetzung eines automatisierten Air-Hockey-Spielers, welcher sich insbesondere durch seine robusten Ausführungen als Portalroboter und durch die Modularität seiner Software auszeichnet. So wird es darauf aufbauenden Projekten, welche sich beispielsweise mit Machine Learning beschäftigen, erleichtert, sich mit den Kernfragen ihrer Themen auseinanderzusetzen.

Inhaltsverzeichnis

Inhaltsverzeichnis	ii
Abbildungsverzeichnis	iii
Tabellenverzeichnis	iv
1 Einleitung	1
1.1 Air-Hockey	1
1.2 Projektdefinition	1
2 Stand der Technik	3
2.1 Marktanalyse	3
2.2 Vorgängerprojekt: Mini Air-Hockey	4
3 Projektanforderungen	6
3.1 Anforderungsliste	6
3.2 Allgemeine Anforderungen	8
3.3 Mechanische Anforderungen	8
3.4 Elektronische Anforderungen	11
3.5 Softwaretechnische Anforderungen	11
4 Mechanischer Arbeitsstrang	13
4.1 Konzeptentwicklung	13
4.1.1 Konzeptmöglichkeiten	13
4.1.2 Nutzwertanalyse & Auswahl des finalen Konzeptes	16
4.2 Auslegung und Berechnung	17
4.3 Motorauswahl	20
4.3.1 Motorarten	20
4.3.2 Motorkenngrößen	21
4.3.3 Motorwahl: Schrittmotor NEMA 17	23
4.4 Ergebnisse	24
5 Softwaretechnischer Arbeitsstrang	25
5.1 Konzeptentwicklung	25
5.1.1 Hardwarekonzept	25
5.1.2 Hardwarekomponenten	28
5.1.3 Softwarekonzept	29
5.1.4 Schnittstellengestaltung und Kommunikation	30
5.2 Bildverarbeitung	31
5.2.1 Konzept	32
5.2.2 Bildaufnahme	33

5.2.3	Bildanalyse	33
5.2.4	Berechnungen zu Position und Geschwindigkeit	34
5.3	Spielstrategie	35
5.3.1	Konzept	35
5.3.2	Simulation	35
5.3.3	Bahnprognose	37
5.4	Kinematik	39
5.4.1	Kinematische Beziehungen	39
5.4.2	Motordynamik	42
5.4.3	Geradlinige Bewegung	44
5.4.4	Codeumsetzung	45
5.5	Spielmenü	46
5.6	Hochzeit der Pakete	47
6	Evaluation	48
6.1	Evaluation der Anforderungen	48
6.1.1	Allgemein	50
6.1.2	Mechanik	50
6.1.3	Elektronik	51
6.1.4	Software	52
6.2	Verbesserungsmöglichkeiten	53
6.2.1	Verbesserungsmöglichkeiten - Mechanik	53
6.2.2	Verbesserungsmöglichkeiten - Softwaretechnik	54
7	Projektorientation	57
7.1	Inhaltliche Projektplanung	57
7.2	Zeitliche Organisation	59
7.3	Realisierung der Zusammenarbeit	62
7.4	Evaluation der Projektorientation	62
8	Fazit und Ausblick	65
9	Anhang	I
Anhang		I
9.1	Modularitätsanleitung	I
9.2	Kostenübersicht	IV
9.3	Datenblätter und Produktdaten	V
Literaturverzeichnis		XI

Abbildungsverzeichnis

3.1	Anfahrbare Fläche	8
3.2	Maximaler Verteidigungsweg des Strikers	10
4.1	Konzeptmöglichkeiten	14
4.2	Nutzwertanalyse	16
4.3	Wirkende Kräfte auf die Konstruktion	17
4.4	CAD Konstruktion des Roboters	19
4.5	Abgerollte Riemenlänge bei Maximalbewegung des Strikers (links: Startposition, rechts: Endposition)	22
4.6	Konstruktion des Roboters	24
5.1	Finales Hardwarekonzept	26
5.2	Konzeptentwurf 2: Zwei Raspberry Pis	27
5.3	Konzeptentwurf 3: Datenverarbeitung mit einem PC	27
5.4	Elektronikkomponenten des Roboters	29
5.5	Kinematik-Koordinaten und Spielfeldhälften	30
5.6	Aufbau der I2C Datenübertragung [I2C]	31
5.7	Konzeptioneller Ablauf des Softwarepaketes Bildverarbeitung	32
5.8	Umwandlung des Bildes und Mittelpunktbestimmung	34
5.9	Flussdiagramm	35
5.10	Simulation Ausgangszustand	36
5.11	Simulation	36
5.12	Schnittpunktberechnung Fallunterscheidung	38
5.13	Veranschaulichung der Kinematik	39
5.14	Bewegungsachsen der Motoren	40
5.15	Drehzahlkonstellationen zur Veranschaulichung	41
5.16	Beschleunigungs- und Bremsrampe für Schrittmotoren	42
5.17	Geradlinigkeit der Bewegung	44
5.18	Konzept des Kinematik-Codes	45
5.19	Struktur des Bedienungsmenüs	46
7.1	Hauptarbeitsstränge mit Meilensteinen (gelb)	58
7.2	Detaillierungsgrad der Arbeitspakete	58
7.3	Ausschnitt des Kostentrackings	59
7.4	Ausschnitt der AP's mit Zeiten	59
7.5	Ausschnitt des Zeittrackings	60
7.6	Ganttdiagramm	61
7.7	Überblick über die Realisierung der Zusammenarbeit	62

Tabellenverzeichnis

2.1	Übernommene Werte von V1	4
3.1	Anforderungsliste	7
3.2	Geometriedaten des Spieltisches und des Strikers	9
3.3	Maximale Verteidigungswege des Strikers	10
4.1	Mögliche Lösungsalternativen eines potenziellen Air-Hockey Roboters	13
4.2	Morphologischer Kasten	14
4.3	Benötigte Größen zur Berechnung der Impulskraft	17
4.4	Massen der Bauteile	18
4.5	Benötigte Werte zur Ermittlung der Beschleunigung des Strikers . .	19
4.6	Vor- und Nachteile der Motorarten	20
4.7	Benötigte Größen zur Berechnung der Riemen geschwindigkeit . . .	21
4.8	Benötigte Größen zur Berechnung der Motordrehzahl	23
4.9	Technische Daten des ausgewählten Motors	24
5.1	Ergebnisse der Dynamiktests	43
6.1	Anforderungsliste	49
9.1	Kostenübersicht	IV

1 Einleitung

Dieser Abschlussbericht beschreibt einleitend das Air-Hockey-Spiel sowie die Projektdefinition zur Entwicklung eines automatisierten Air-Hockey-Spielers. Aber auch die Entwicklung des Roboters selbst, seine Funktionsweise und Eigenschaften auf mechanischer und softwaretechnischer Ebene werden beleuchtet. Dabei werden besonders die Konzeptentwicklung und die verwendeten Konzipierungsmethoden detailliert beschrieben. Gefolgt wird dies von den durchgeführten Berechnungen und den konstruktiven Ergebnissen sowie der Darstellung der Hard- und Softwareumsetzung. Schließlich wird die Projektorganisation im Allgemeinen beschrieben und rückblickend evaluiert.

1.1 Air-Hockey

Air-Hockey ist ein Freizeitspiel, das oft in Spielhallen gespielt wird. Dabei treten zwei bis vier Spieler gegeneinander an. Die Teams platzieren sich an den gegenüberliegenden Seiten eines speziellen Tisches. Die handelsüblichen Größen dieser Tische gehen von 400 mm x 800 mm bis 1000 mm x 2000 mm. Eine von unten mit Luft durchströmte Platte bildet eine Spielfläche mit geringem Reibungswiderstand. Jeder Spieler versucht mit einem Schläger, dem sogenannten Striker, den auf dem Luftpissen gleitenden Puck ins gegnerische Tor zu spielen. Dabei darf sich der Striker nicht über die Mittellinie bewegen. Nach einem Treffer beginnt der Spieler, in dessen Tor der Puck gespielt wurde. Gewonnen hat das Team, das zuerst sieben Tore erzielen konnte.

1.2 Projektdefinition

Das Ziel des Projektes ist die Entwicklung eines „Automatisierten Air-Hockey-Spielers mit modularem Aufbau“. Ein menschlicher Spieler soll demnach durch einen Roboter ausgetauscht werden, der die Aufgaben des Spielers übernimmt. Dabei handelt es sich um ein Folgeprojekt. Im Rahmen des ursprünglichen Projektes wurde bereits ein Spieltisch, eine Kameraaufstellung und ein erster Roboter entworfen. Es ist daher möglich, den Grundaufbau des Air-Hockey-Tisches zu übernehmen und diesen um einen zweiten, spielfähigen Roboter zu ergänzen.

Die Entwicklung des neuen Roboters beinhaltet sowohl die Konstruktion als auch die Programmierung. Dabei wird mit der existierenden Kameraaufstellung das Spielfeld erfasst. Das aufgenommene Bild soll mittels der Software ausgewertet und die Position des Pucks erfasst werden. Mit diesen Informationen kann die strategisch günstigste Reaktion des Roboters berechnet werden und die Soll-Reaktion in einen

Befehl an die Motoren des Roboters übersetzt werden. Durch diese Weiterverarbeitung kann mit dem erfassten Spielfeldbild eine mechanische Reaktion des Roboters ausgeführt werden.

Einen wichtigen Aspekt und Vorteil bei der Entwicklung des neuen Roboters stellt die Modularität dar. Sowohl die mechanische Konstruktion des Roboters als auch einzelne Softwarepakete sollen austauschbar gestaltet werden. Dafür ist es wichtig, voneinander getrennte Softwarepakete mit definierten Schnittstellen zu entwickeln. Aus diesem Grund gliedert sich der elektronische und programmietechnische Teil in Bildverarbeitung, Spielstrategie und Kinematik. Dadurch ist es beispielsweise mit geringem Aufwand möglich, die als Paket konzipierte Spielstrategie mit einer neuen, maschinell lernenden Spielstrategie auszutauschen und an den definierten Schnittstellen in diesem Roboter zu implementieren. Die maschinell lernende Spielstrategie wird in einem weiteren, gegenwärtig laufenden Projekt entwickelt.

2 Stand der Technik

Um einen Überblick über existierende, automatisierte Air-Hockey-Tische zu erhalten und Informationen über vergleichbare Projekte zu finden, wurde der Stand der Technik recherchiert und die Ergebnisse zusammengefasst. Neben einer Marktanalyse wurde der bereits bestehende Roboter anhand des Berichts und der vorhandenen Konstruktion betrachtet.

Durch dieses Vorgehen werden verschiedene Arten von Air-Hockey-Robotern näher betrachtet und Vor- und Nachteile abgewogen und dadurch eigene Umsetzungen begründet. Außerdem können einige mögliche Probleme bereits vor Beginn des Projekts erkannt werden und aus vorhergehenden Fehlern gelernt werden.

2.1 Marktanalyse

Weltweit wurden bereits verschiedene Roboter entwickelt, die Air-Hockey spielen können. Das Ziel dieser Projekte ist meist die Darstellung von Automatisierungsprozessen bzw. von Neuheiten der Industrie 4.0 und die Erprobung und Visualisierung von Machine Learning. Dieses Ziel wurde auch in diesem Projekt gesetzt, da neben der Entwicklung eines konstruktiv und steuerungstechnisch sinnvollen Roboters auch das Machine Learning und die dadurch mögliche Fähigkeitensteigerung eines automatischen Spielers von anderen Gruppen betrachtet werden.

Bei der Realisierung eines automatischen Spielerroboters sind die Bilderkennung des Pucks, die Übertragungs- und Verarbeitungsgeschwindigkeit der Daten und die Steuerungssoftware von ausschlaggebender Bedeutung.

In bestehenden Umsetzungen auf dem Markt wird zur Puckerkennung oft eine Hochgeschwindigkeitskamera eingesetzt, die 300 oder sogar mehr Bilder pro Sekunde liefern kann [JFalconer]. Diese Bilder werden in Echtzeit verarbeitet und als Input an die individuelle Steuerungssoftware des Roboters weitergegeben. Im Vergleich dazu sieht der Mensch bereits bei rund 24 Bildern pro Sekunde eine flüssige Bewegung und nimmt mehr als 50 Bilder pro Sekunde nicht mehr wahr [Motion]. Durch die hohe Anzahl an Bildern, die verarbeitet werden, kann die Puckbahn exakt bestimmt bzw. vorhergesehen werden. Bei guter Schnittstellengestaltung und optimaler Ausnutzung der Bilderkennung für die Steuerung des Roboters hat dieser das Potenzial, den Menschen als Gegner durchgängig zu besiegen. Jedoch kann auch eine Kamera mit geringerer Frequenz ausreichen, um den Robotor zu einem ernsten Gegner für den Menschen zu machen [JJRobots].

Die Spielfähigkeiten des Roboters werden stark durch die konstruktive Gestaltung des Roboters, seiner Beweglichkeit und die Antriebe beeinflusst. Bei bestehenden Projekten lassen sich verschiedene Bauweisen erkennen.

So existiert zum Beispiel die Möglichkeit, den Roboter als einen Striker zu konstruieren, welcher ausschließlich linear entlang zweier orthogonaler Achsen bewegt werden kann. Dadurch kann eine rechteckige Fläche vom Roboter bespielt werden. Angetrieben werden diese Konstruktionen durch zwei bis drei Motoren an Linearachsen welche einen jeweiligen Riemen führen. Antriebssysteme sind wie auch in diesem Projekt geplant häufig additiv gefertigt. [ITQ] [JJRobots]

Eine weitere Möglichkeit ist die Realisierung des Roboters als mehrgelenkiger Arm, wodurch eine halbkreisförmige Fläche bespielt werden kann [BBC]. Beispielsweise wurde in einem Projekt ein bereits für andere Zwecke entwickelter Roboterarm umfunktioniert, um Air-Hockey spielen zu können. [LFarlow]. Auch das Vorgängerprojekt V1 hat einen Roboterarm mit zwei Gelenken genutzt.

2.2 Vorgängerprojekt: Mini Air-Hockey

Das Vorgängerprojekt V1 dient als Grundlage dieses Projekts und macht die Umsetzung eines Automatisierten Air-Hockey-Spielers überhaupt erst möglich. Es ist ein bereits voll funktionsfähiger Air-Hockey-Tisch vorhanden, welcher von Menschen oder Robotern bespielt werden kann. Der Tisch ist 512 mm breit, 1080 mm lang und hat durch eine Kamerahalterung eine Höhe von 1304 mm. Für den Air-Hockey-Tisch wurde bereits ein Roboter entwickelt, bei dem es sich um ein Zweigelenkarm handelt. Dafür ist der Tisch mit einigen elektronischen Komponenten ausgestattet: mit einer Kamera, um das Spielfeld aufzunehmen, einem Raspberry Pi, der die Kamerabilder verarbeitet, einer Tischbeleuchtung, um gleichmäßige Lichtverhältnisse zu schaffen und einem Arduino, der die Servomotoren des Roboters ansteuert. Von den elektronischen Bauteilen wird die Kamera übernommen und der Raspberry Pi mitbenutzt. So soll es möglich sein, dass beide Roboter von derselben Kamera Bilder erhalten, somit parallel laufen und dadurch gegeneinander spielen können.

Durch den Bau des Roboters im Vorgängerprojekt, hat sich die Vorgängergruppe mit einigen Problemstellungen auseinandergesetzt und einige Annahmen getroffen, welche ebenso Einfluss auf unser Projekt haben. Bei Durchsicht des Vorgängerberichts und beim Nachvollziehen der einzelnen Arbeitsschritte, konnten einige Rückschlüsse für das gegenwärtige Projekt getroffen und manche Werte und Lösungsansätze übernommen werden.

Folgende Werte wurden vom Vorgängerprojekt übernommen:

Tabelle 2.1: Übernommene Werte von V1

Bezeichnung	Formelzeichen	Wert
Masse des Pucks	m_{Puck}	20,0 g
Zeit der Berührung (Puck-Striker)	Δt	20,0 ms
Puckgeschwindigkeit	v_{Puck}	3,2 $\frac{m}{s}$
Benötigte Zeit des Strikers: Durchquerung der Spielfeldlänge	t	250 ms

Einige der Rückschlüsse, die sich ergeben haben, werden im Folgenden kurz erwähnt. Beim Vorgängerprojekt werden die meisten Rechenoperationen auf einem externen Computer durchgeführt, der, um den Roboter spielen zu lassen, immer erst mit dem Raspberry Pi über WLAN verbunden werden muss. Daraus ergab sich für das gegenwärtige Projekt der Anspruch, einen Roboter zu gestalten, der ohne Computer funktioniert und als Gesamtkonstruktion austauschbar ist. Daraus folgt, dass die Verkabelung übersichtlich gestaltet werden muss, um einen guten Überblick zu ermöglichen und es für nachfolgende Projekte erweiterbar zu gestalten. Des Weiteren soll es möglich sein, die Geschwindigkeit der Motoren zu verändern und zu kontrollieren. Dies ist bei den im Vorgängerprojekt verwendeten Servomotoren nicht möglich.

Die übernommenen Werte und die für das Projekt gewonnenen Rückschlüsse, fließen zum Teil in die Anforderungen mit hinein.

3 Projektanforderungen

Zu Beginn der Projektarbeit wurden verschiedene Kriterien zur Bewertung des Ergebnisses herausgearbeitet, um den Roboter anhand dieser Kriterien vergleichbar zu machen. Dabei wurde sowohl das gesamte Ergebnis allgemein als auch die Bereiche Mechanik, Elektronik und Software betrachtet. Es wurde eine Anforderungsliste zusammengestellt, die relevante, möglichst auch messbare Anforderungen an den zu konstruierenden Roboter enthält. Diese Liste wurde im Verlauf der Projekts erweitert und überarbeitet.

Allgemeine Anforderungen ergaben sich aus der Projektdefinition, die die Gruppe erarbeitet hat. Konkrete Werte für die spezifischen Anforderungen konnten durch vereinfachte Annahmen und Rechnung und durch den bereits bestehenden Roboter gefunden werden. Leider gibt es in der Literatur kaum Vergleichswerte, da Air-Hockey nicht normiert ist und die Eigenschaften des Spiels auch stark tischabhängig sind.

3.1 Anforderungsliste

In der Anforderungsliste sind die Anforderungen in die Bereiche Allgemein, Mechanik, Elektronik und Hardware eingeteilt. Dabei wird unterschieden, ob die Anforderung als Forderung (F) oder als Wunsch (W) gestellt werden. In den folgenden Kapiteln wird auf die Anforderungen in den jeweiligen Bereichen genauer eingegangen.

Tabelle 3.1: Anforderungsliste

F / W	Anforderung
Allgemein	
F	Ergebnis ist ein spielfähiger Roboter
F	Einhalten des Budgets von 250€
F	Montage des Gesamtsystems in vorliegendem Bauraum möglich
F	Gesamtsystem ist modular ausgeführt
Mechanik	
F	Ausreichend steife Komponenten halten den Belastungen stand
F	Minimale Sichtbehinderung für die Kamera
F	Aktionsfläche ausreichend zur Torverteidigung (200 mm x 150 mm)
F	Ausreichend starke Aktoren um auftretende Trägheiten zu überwinden
F	Realisierung ausreichender Striker-Geschwindigkeiten
F	Realisierung gewünschter Bewegungsbahnen des Strikers
W	Größere Aktionsfläche realisiert (400 mm x 300 mm)
Elektronik	
F	Hohe Treffgenauigkeit wird ermöglicht
F	Motoren drehen flüssig, Schritte werden nicht übersprungen
F	Geradlinige Bewegungen sind möglich
F	Zeit von Erhalt der Koordinate bis Erreichen unter 500 ms
F	Betriebstemperatur der Hardware bleibt im unkritischen Bereich
F	Start-Kalibrierung der Strikerposition möglich
W	Übersichtliche und nachvollziehbare Verkabelung der Bauteile
W	Stromversorgung der Hardware ausschließlich über ein gemeinsames Netzteil
W	Keine Nutzung externer Geräte wie eines Laptops, damit keine aufwändige Einrichtung des Systems vorm Spielen notwendig ist
W	Menü am Tisch ermöglicht einfache Bedienung des Roboters
W	Zeit von Erhalt der Koordinate bis Erreichen unter 150 ms
Software	
F	Gut zugängliche Software mit standardisierten Schnittstellen
F	Genaue Puckdetektion durch Bildverarbeitung
F	Berechnung des Puck-Abprallens an den Bändern umgesetzt
F	Berechnung des notwendigen Striker-Zielpunktes um Puck zu treffen
F	Striker kann sich in alle Richtungen bewegen
W	Angriffsbewegung möglich, sodass Puck sich gezielt bewegt

3.2 Allgemeine Anforderungen

Die allgemeinen Anforderungen lassen sich aus der Art des Projekts, der Aufgabenstellung und den Projektinhalten herleiten.

Der Roboter ist ein Studentenprojekt und soll voraussichtlich in zukünftigen Studentenprojekten weiter bearbeitet werden soll. Deswegen ist die Funktionalität des Gesamtsystems und besonders auch die Modularität wichtig, damit das System einfach angepasst werden kann. Weiterhin sollte der Roboter regelkonform spielen und einen möglichst guten Gegner darstellen. Dazu trägt auch eine angemessene Reaktionszeit bei, die in 3.3 weiter erläutert wird. Außerdem müssen die äußeren Faktoren des Projekts eingehalten werden. Dazu gehört, dass die verwendeten Komponenten und Materialien insgesamt nicht mehr als 250€ kosten dürfen.

3.3 Mechanische Anforderungen

Neben allgemeinen Anforderungen an die Konstruktion werden insbesondere kinematische Anforderungen noch vor der Auslegung- und Konstruktionsarbeit festgelegt. Dazu zählen die Stabilität, eine minimale Sichtbehinderung für die Kamera, die bespielbare Fläche, Realisierung gewünschter Bewegungsbahnen der Strikers, die Reaktionszeit und die damit einhergehende Schnelligkeit des Roboters.

Die gewünschte Stabilität erreicht der Roboter, wenn keine sichtbaren Schwingungen, keine Biegung durch Trägheit bei Beschleunigung und keine Biegung bei Impuls durch den Puck auftreten. Außerdem soll der Roboter dem Eigengewichts in Erdbeschleunigungsrichtung standhalten.

Anfahrbare Fläche s_a :

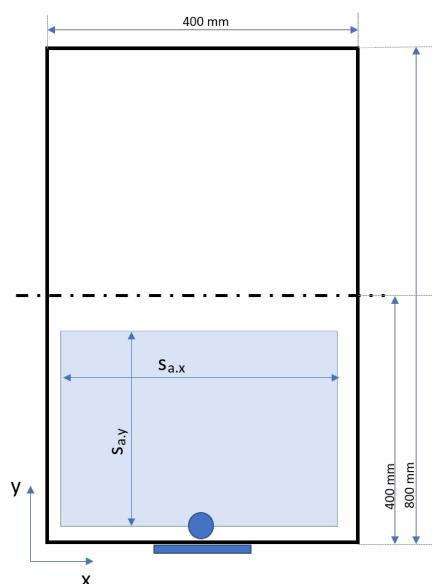


Abbildung 3.1: Anfahrbare Fläche

Tabelle 3.2: Geometriedaten des Spieltisches und des Strikers

Bezeichnung	Formelzeichen	Wert
Durchmesser des Strikers	$d_{S.alt}$	34 mm
Pufferstrecke in y-Richtung	$s_{Puffer,y}$	10 mm
Länge des Tisches	l_{Tisch}	800 mm
Breite des Tisches	b_{Tisch}	400 mm

Die anzufahrende Fläche ergibt sich aus der Regel, dass der Striker nur die eigene Hälfte bespielen darf. Es wird davon ausgegangen, dass der Durchmesser des zu konstruierenden Strikers mit dem des Vorgängerprojektes übereinstimmt. Subtrahiert man von den 400 mm in jede Richtung nun in x-Richtung den Durchmesser des Strikers und in y-Richtung sowohl den Durchmesser des Strikers als auch einen Puffer von 10 mm, erhält man folgende Werte:

Anfahrbare Strecke in x-Richtung:

$$s_{a.x} = b_{Tisch} - d_{S.alt} = 366 \text{ mm.} \quad (3.1)$$

Anfahrbare Strecke in y-Richtung:

$$s_{a.y} = \frac{l_{Tisch}}{2} - s_{Puffer,y} - d_{S.alt} = 356 \text{ mm.} \quad (3.2)$$

Reaktionszeit t_R :

Für die Reaktionszeit werden die Versuche aus dem Projekt „Mini Air-Hockey“ herangezogen. Aus diesen Versuchen lässt sich eine Zeit von 250 ms ableiten, die der Puck im Mittel benötigt, um den Tisch in y-Richtung zu durchqueren (vgl. Kap. 2.2). Davon seien weitere 50 ms für die Software reserviert. Da innerhalb dieses Projekts mit demselben Tisch gearbeitet wird, werden die Zeiten übernommen. Jedoch wird von der Gesamtzeit von 250 ms ein Puffer von 20% abgezogen, um einen etwaigen Angriffsschlag zu simulieren, welcher die mittlere Gesamtzeit unterschreiten würde. Nach Abzug der für die Software reservierten Zeit von 50 ms verbleiben 150 ms für die Reaktion des Roboters. Allerdings wird vorausgesetzt, dass der Roboter den Puck bereits auf der Hälfte seines Spielfeldes abfangen kann, um einen guten Spieldynamik des Roboters zu gewährleisten. Folglich verbleiben nur 75% von den 150 ms, nämlich 112,5 ms, für die Reaktion des Roboters.

$$t_R = (250 \text{ ms} - 0,2 \cdot 250 \text{ ms} - 50 \text{ ms}) \cdot 0,75 = 112,5 \text{ ms} \quad (3.3)$$

Geschwindigkeit $v_{S.min}$:

Aus der Reaktionszeit ergibt sich die Geschwindigkeit, die der Roboter im Extremfall fahren muss.

Um diesen Extremfall abzubilden, muss der maximale Verteidigungsweg verwendet werden. Von der mittleren Verteidigungsposition aus, unmittelbar vor dem eigenen Tor, muss der Roboter in einem solchen Extremfall jeden Punkt auf der Hälfte seines

Spielfeldes erreichen können. Da es sich hierbei um einen konservativ ausgelegten Extremfall handelt, wird von einer konstanten Geschwindigkeit ausgegangen und die Be- und Entschleunigung der Konstruktion vernachlässigt.

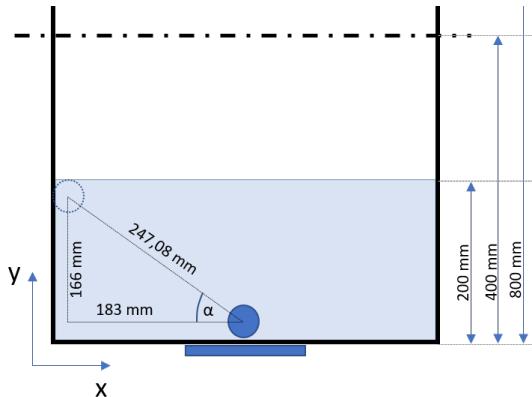


Abbildung 3.2: Maximaler Verteidigungsweg des Strikers

Tabelle 3.3: Maximale Verteidigungswege des Strikers

Bezeichnung	Formelzeichen	Wert
Max. Verteidigungsweg x-Richtung	$s_{v.x}$	166 mm
Max. Verteidigungsweg y-Richtung	$s_{v.y}$	183 mm

Damit ergibt sich der maximale Verteidigungsweg zu [TM, S. 18]:

$$s_{v.max} = \sqrt{s_{v.x}^2 + s_{v.y}^2} = 247,08 \text{ mm}. \quad (3.4)$$

Diese Strecke muss unter dem Winkel α gefahren werden können [TM, S. 19]:

$$\alpha = \arctan \left(\frac{s_{v.x}}{s_{v.y}} \right) = 42,21^\circ. \quad (3.5)$$

Die resultierende Mindestgeschwindigkeit des Strikers ergibt nach [TM, S. 30] zu:

$$v_{S.min} = \frac{s_{v.max}}{t_R} = 2,196 \frac{\text{m}}{\text{s}}. \quad (3.6)$$

3.4 Elektronische Anforderungen

Im Bereich Elektronik beziehen sich die festgelegten Anforderungen auf die Ziele, die Spielfähigkeit zu sichern, eine weitere Bearbeitung möglichst zu vereinfachen und das System modular zu gestalten.

Besonders die ausgewählten Motoren spielen eine ausschlaggebende Rolle. Sie sollen flüssig laufen, möglichst keine Schritte überspringen und durch ausreichend Dynamik ein flüssiges Spiel ermöglichen. Außerdem werden gradlinige Bewegungen des Strikers und eine gute Treffgenauigkeit gefordert. Zusätzlich soll zur Kontrolle der Position des Strikers eine Kalibrierungsmöglichkeit der Startposition vorhanden sein.

Die Reaktionszeit der Motoren von Erhalt der Koordinate bis zum Ende der Bewegung und dem Erreichen der gewünschten Position sollte unter 500 ms liegen, damit ein Spielfluss zustanden kommen kann. Optimal wäre jedoch eine Reaktionszeit des Roboters von unter 150 ms, damit dieser bei einem schnellen Spiel zuverlässig abwehrt (vgl. Kapitel 3.3).

Eine weitere Wunsch-Anforderungen ist eine übersichtliche sowie leicht demontierbare Verkabelung und Anordnung der Bauteile, um die Abläufe des Systems leichter durchschauen zu können. Dadurch bietet sich auch die Möglichkeit, andere elektronische Bauteile ohne Probleme hinzuzufügen bzw. bestehende Teile zu entfernen. Außerdem sollte die Hardware kompakt montiert werden und alle elektronischen Bauteile durch ein gemeinsames Netzteil versorgt werden.

Weiterhin wird angestrebt, dass der gesamte Roboter ein abgeschlossenes System bilden. Dazu gehört, dass sich die benötigte Hardware sowie alle anderen Bauteile am Tisch befinden. Dies ermöglicht, dass der Roboter ohne weitere Vorbereitungen aktivierbar und der Tisch bespielbar ist. Auf die Verwendung eines Laptops für mehr Rechenleistung oder schnelle Datenverarbeitung wird also verzichtet. Um zusätzlich die direkte Bedienung des Roboters zu erleichtern, sollte sich wunschweise ein Menü am Tisch befinden.

3.5 Softwaretechnische Anforderungen

Grundlegend soll auch die Software gut zugänglich sein, modular aufgebaut sein und standardisierte Schnittstellen besitzen.

Um die Puckbahn bestimmen zu können, muss es zu jedem Zeitpunkt eine aktive Bildverarbeitung geben, die den Puck genau erkennt und die Koordinaten weitergibt. Mit den Daten der Bildverarbeitung kann die Puckbahn berechnet werden, wobei Abprallen des Pucks an der Bande mit einbezogen wird. Außerdem soll der Punkt des Eintreffens des Pucks genau bestimmt werden, damit der Striker an dieser Zielposition den Puck verteidigen kann. Damit die Verteidigung überhaupt möglich ist, soll jeder Punkt innerhalb der benötigten Aktionsfläche angesteuert werden können.

Neben reiner Verteidigung ist eine Wunschanforderung das Angreifen. Dies soll realisiert werden, indem die Kinematik ermöglicht, dass von jedem Punkt aus eine Bewegung in jede beliebige Richtung durchgeführt werden kann. Dann kann auch in der Berechnung des Reaktionspunktes der jeweilige Zielwinkel mit einbezogen werden und der Punkt unter diesem Winkel angefahren werden. Eine noch bessere Zielfähigkeit wäre gegeben, wenn der Zielpunkt auf einer beliebigen Bahn, möglicherweise auch einer kurvenähnlichen Bahn angefahren werden kann.

4 Mechanischer Arbeitsstrang

Der erste Arbeitsstrang des Projekts umfasst den mechanischen Aufbau des Roboters. Dies beinhaltet die Konzeptentwicklung, die Auslegung und Berechnung der Komponenten, der Motorauswahl und schließlich die Vorstellung der Ergebnisse dieses Arbeitsstrangs.

4.1 Konzeptentwicklung

Da die Erfüllung des Projektziels maßgeblich von dem mechanisch Aufbau des Roboters abhängt, wird der Konzeptentwicklung des mechanischen Aufbaus viel Aufmerksamkeit und Zeit gewidmet. So werden zunächst fünf verschiedene Konzepte entwickelt und intern ausgewertet. Damit kann die Konstruktion auf drei verschiedene mechanische Konzepte eingegrenzt werden, die mit verschiedenen Antriebskonfigurationen vier sinnvolle, zu evaluierenden Varianten darstellen. Diese vier Varianten werden in einer umfassenden Nutzwertanalyse mit der gesamten Projektgruppe bewertet, sodass sich eine finale Variante herauskristallisiert.

4.1.1 Konzeptmöglichkeiten

In der Tabelle 4.1 werden alle Kriterien des Roboters mit ihren möglichen Lösungsvarianten aufgezeigt.

Tabelle 4.1: Mögliche Lösungsalternativen eines potenziellen Air-Hockey Roboters

Kriterium	Mögliche Lösungsvarianten		
Anzahl der Arme	1	2	3
Anzahl der Gelenke	0	1	2
Art der Bewegung	rotatorisch	translatorisch	beide
Anzahl der Motoren	1	2	3
Kraftübertragung	direkt	Riemen	Getriebe

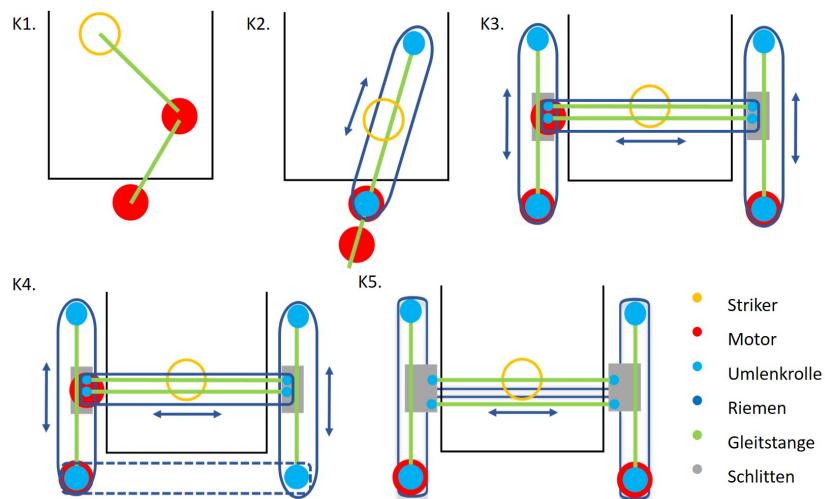
Da der Roboter den Striker ausschließlich in der 2D-Ebene auf dem Spielfeld führen muss, lässt sich der mechanische Aufbau auf einige wenige mögliche Konzepte eingrenzen.

Tabelle 4.2: Morphologischer Kasten

Kriterium	Mögliche Lösungsvarianten
Anzahl der Arme	1 2 2
Anzahl der Gelenke	0 1 2
Art der Bewegung	rotatorisch translatorisch beide
Anzahl der Motoren	1 2 3
Kraftübertragung	direkt Riemen Getriebe
	K1 K2 K5 K3 K4

So kommen folgende Konzepte in Frage:

- K1. Gelenkarm mit einem Gelenk
- K2. Schwenkarm mit Linearführung
- K3. Linearführung mit drei Motoren
- K4. Linearführung mit zwei Motoren und Übersetzung auf die zweite Linearachse
- K5. Linearführung mit H-Aufbau und einem durchgängigen Riemen

**Abbildung 4.1:** Konzeptmöglichkeiten

Nach interner Bewertung werden die Konzepte K2, K3 und K5 ausgewählt, um diese in einer Nutzwertanalyse mit der gesamten Projektgruppe zu evaluieren. Hierbei

ist zu beachten, dass das Konzept K2 in zwei verschiedenen Motorkonfigurationen bewertet wird. Während in allen anderen Konzepten ausschließlich Schrittmotoren (vgl. Kap 4.3) verwendet werden sollen, lässt das Konzept K2 auch die Verwendung von einem Servomotor zum Antrieb des Schwenkarmes zu, da dieser nie über 180° drehen wird.

So ergeben sich folgende 4 Variationen:

- V1. Schwenkarm mit Linearführung mit Schrittmotoren
- V2. Schwenkarm mit Linearführung mit Servomotoren
- V3. Linearführung mit drei Motoren
- V4. Linearführung mit H-Aufbau

V1 und V2: Schwenkarm

Von einem Schritt- oder Servomotor angetrieben kann der Schwenkarm über das Spielfeld geschwenkt werden. Der Schwenkarm selbst bildet dabei eine Linearführung für den Striker, welcher mittels eines Riementriebs mit einem weiteren Schrittmotor verbunden ist. So entsteht eine Kombination aus rotatorischer (Schwenken) und translatorischer Bewegung (entlang der Linearführung).

Am Schwenkarm entstehen hier vergleichsweise große Belastungen mit der Gefahr von Instabilität. Der Aufwand des Umsetzens der kinematischen Beziehung während der Programmierung ist überschaubar, da der Aufbau relativ simpel ist.

V3: Linearführung mit drei Motoren

Diese Variante entspricht mit seiner Kinematik dem denkbar einfachsten Aufbau. Während eine in x-Richtung ausgerichtete Linearführung mithilfe eines Schrittmotors den Striker quer über das Spielfeld bewegt, gewährleisten zwei weitere Schrittmotoren und Linearführungen die Bewegung in y-Richtung. Der Striker wird, ähnlich wie in V1 und V2, mithilfe von Riementrieben über die Linearführung geradlinig bewegt.

V4: Linearführung mit H-Aufbau

Der H-Aufbau entspricht weitgehend der Variante V3. Jedoch sind die Komponenten hier über nur einen einzigen Riemen verbunden, welcher von beiden Motoren bewegt werden kann. Bewegen sich beide Motoren mit gleicher Drehzahl entgegengesetzt, wird der Striker in Längsrichtung vor- oder zurückbewegt. Drehen die Motoren gleichsinnig, bewegt sich der Striker in Querrichtung. Relativbewegungen dazwischen resultieren in, in Längs- und Querrichtung überlagerte, Bewegungsrichtungen.

Dieser Aufbau hat den Vorteil, dass durch die geschickte Anordnung des Riementriebs und der Umlenkrollen, ein dritter Motor gespart werden kann und kein Motor bei Bewegungen mitbewegt werden muss. Dadurch wird die zu bewegende Masse in y-Richtung stark reduziert und Trägheiten während des Spielens werden weitgehend minimiert. Die mechanischen Komponenten müssen weniger aushalten

können, die Motoren können kleiner dimensioniert und die Gesamtkonstruktion folglich in Richtung Leichtbau verbessert werden.

Da durch den komplizierteren Aufbau die resultierende Bewegungsrichtung bei jeweiliger Motordrehung weniger trivial sind, ist die Umsetzung der kinematischen Beziehungen während der Programmierarbeit als schwerer im Vergleich zu V1, V2 und V3 einzuschätzen.

4.1.2 Nutzwertanalyse & Auswahl des finalen Konzeptes

In der Nutzwertanalyse werden die vier Varianten hinsichtlich der Kriterien: Komplexität, Stabilität, Anbauaufwand, Preis, bespielbare Fläche, Behinderung der Kamera, Schnelligkeit, Kinematikumsetzung im Code, Positionserkennung des Motors, Trägheit und Originalität bewertet. Dabei werden die Kriterien je nach Relevanz im Rahmen des Projekts unterschiedlich stark gewichtet. Außerdem werden den einzelnen Bewertungen der Aufbauvarianten je Kriterium Partialnutzenwerte von 1 bis 5 zugeordnet. Anschließend werden die Partialnutzenwerte mit der Gewichtung verrechnet, die Summe gebildet und diese normiert, sodass sich die optimale Variante herauskristallisieren kann. Die verschiedenen Varianten werden wie folgt bewertet:

Nutzwertanalyse								
Attribute	Wichtigkeit	Evaluationsart	Aufbau					
			V1. Schwenkarm:		V2. Schwenkarm:		V3. Linear	
			Servo + Schritt	Schritt + Schritt	nur Schritt	nur Schritt	mittelschwer	
Komplexität	3 6,25%	Bewertung PNW	einfach	einfach	schwer	mittelschwer		
			4	4	1	2		
Stabilität	8 16,67%	Bewertung PNW	nicht stabil	nicht stabil	sehr stabil	sehr stabil		
			1	1	5	5		
Anbauaufwand	3 6,25%	Bewertung PNW	mittelleicht	mittelleicht	schwerer, Kabel	mittelschwer		
			4	4	1	2		
Preis	6 12,50%	Bewertung PNW	günstig	günstig	teurer 3M	mitteleuer		
			3	3	1	2		
Bespielbare Fläche	4 8,33%	Bewertung PNW	ungünstig	ungünstig	sehr gut	sehr gut		
			2	2	5	5		
Behinderung der Kamera	4 8,33%	Bewertung PNW	doll	doll	mittel	mittel		
			1	1	4	4		
Schnelligkeit	4 8,33%	Bewertung PNW	schnell	schnell	schnell	schneller		
			4	4	4	5		
Kinematikumsetzung im Code	2 4,17%	Bewertung PNW	mittel	mittel	super leicht	schwer		
			2	2	5	1		
Positionserkennung Motor	3 6,25%	Bewertung PNW	mittel	schlecht	schlecht	schlecht		
			2	0	0	0		
Trägheit	7 14,58%	Bewertung PNW	mittelschlecht	mittelschlecht	schlecht	mittigut		
			2	2	1	4		
Originelle Lösung	4 8,33%	Bewertung PNW	fancy	fancy	intuitiv, primitiv	sehr elegant		
			3	3	4	5		
Summe:	48		2,38	2,25	2,85	3,54		

Abbildung 4.2: Nutzwertanalyse

Wie in 4.2 ersichtlich, hat die Variante V4 Linearführung mit H-Aufbau mit 3,53/5 Bewertungspunkten die beste Bewertung erhalten und kann darüber hinaus auch einen deutlichen Vorsprung vor den anderen Varianten aufweisen. Variante V4 wird deshalb für die weitere Konstruktion ausgewählt und im weiteren Verlauf ausgelegt und dimensioniert.

4.2 Auslegung und Berechnung

Die Auslegung und die Berechnung der Roboterkonstruktion müssen simultan geschehen, da diese sich gegenseitig bedingen. In der Abbildung 4.3 wird ein Überblick über die Kräfte gegeben, die auf die Konstruktion wirken. Dabei handelt es sich nicht um einen Freischnitt der Konstruktion, sondern lediglich um eine skizzenhafte Darstellung. Der Roboter muss den statischen Kräften des Eigengewichtes der Bauteile und der Vorspannkraft des Riemens standhalten. Durch die Berührung des Pucks erfährt der Striker eine Impulskraft. Bei einer Bewegung der Querkonstruktion entlang des Tisches und einer Bewegung des Strikers kommt es zu einer Krafteinwirkung in Folge der Trägheit.

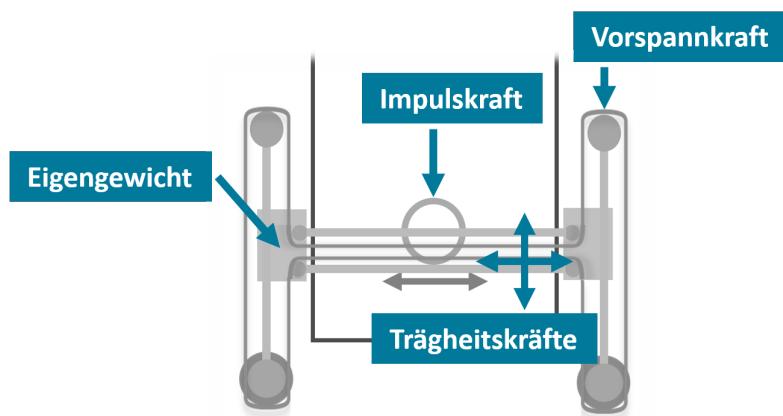


Abbildung 4.3: Wirkende Kräfte auf die Konstruktion

Das Eigengewicht der Bauteil kann aufgrund der geringe Dichte der Bauteile vernachlässigt werden. Die Riemenvorspannkraft wird bei der Dimensionierung der Wellen, auf welchen die Umlenkrollen laufen, berücksichtigt. Diese werden äußerst konservative ausgelegt, weshalb auf die Berechnung der exakten Größe verzichtet werden kann. Im Folgenden wird deshalb nur die Impulskraft und die Trägheitskraft berechnet.

Berechnung der Impulskraft F_I :

Vom Vorgängerprojekt „Mini Air-Hockey“ können folgende Daten übernommen werden (vgl. Kap. 2.2):

Tabelle 4.3: Benötigte Größen zur Berechnung der Impulskraft

Bezeichnung	Formelzeichen	Wert
Masse des Pucks	m_{Puck}	20,0 g
Zeit der Berührungszeit (Puck-Striker)	Δt	20,0 ms
Puckgeschwindigkeit	v_{Puck}	3,2 $\frac{m}{s}$

Nach Dubbel [DUB, S. B25(13) & S. B26(17)] gilt für:

$$F_I = m_{Puck} \cdot \frac{\Delta v}{\Delta t} = m_{Puck} \cdot \frac{v_{Puck.2} - v_{Puck.1}}{\Delta t}. \quad (4.1)$$

Näherungsweise kann Folgendes angenommen werden:

$$v_{Puck.1} = v_{Puck.2} \quad (4.2)$$

$$\Delta v = 2 \cdot v_{Puck}. \quad (4.3)$$

Damit ergibt sich die Impulskraft, die der Puck auf den Striker ausübt zu:

$$F_I = m_{Puck} \cdot \frac{2 \cdot v_{Puck}}{\Delta t} = 6,2 \text{ N}. \quad (4.4)$$

Berechnung der Trägheitskraft F_T :

Bei einer Be- oder Entschleunigung der Konstruktion treten Trägheitskräfte der einzelnen Komponenten auf. Dabei überwiegen die Trägheitskräfte in y-Richtung. Es wird daher im Folgenden nur auf die Trägheitskräfte in y-Richtung eingegangen.

Nach Dubbel [DUB, S. B25(13)] können die Trägheitskräfte der Bauteile nach folgender Formel berechnet werden:

$$F_T = m \cdot a. \quad (4.5)$$

Aus den Datenblättern (vgl. Kapitel 9.3 und Materialeigenschaften der Bauteile [Wiki-Stahlb] sowie den 3D-Druck Informationen können folgende Werte für die Massen der Bauteile, die einen Einfluss auf die Trägheitskräfte in y-Richtung haben, zusammengefasst werden:

Tabelle 4.4: Massen der Bauteile

Bezeichnung	Formelzeichen	Wert
Masse des Strikers	m_S	21 g
Masse der Querstange	m_Q	79 g
Masse des Verbindungsstückes	m_V	16 g
Masse der Umlenkrolle	m_U	8 g
Masse des Wälzlers	m_W	3 g
Masse des Linearlagers	m_L	17 g
Masse des Riemens	m_R	27 g

Die Masse der Querkonstruktion, die eine Beschleunigung in y-Richtung erfährt ergibt sich zu:

$$m_{QK} = m_S + 2 \cdot m_Q + 2 \cdot (m_V + m_U + 2 \cdot m_W) + 4 \cdot m_L + m_R = 333 \text{ g}. \quad (4.6)$$

Neben der Masse der Querkonstruktion fließt auch die Beschleunigung des Strikers in die Berechnung der Trägheitskraft ein.

Tabelle 4.5: Benötigte Werte zur Ermittlung der Beschleunigung des Strikers

Bezeichnung	Formelzeichen	Wert
Mindestgeschwindigkeit des Strikers	$v_{S.min}$	2,196 $\frac{\text{m}}{\text{s}}$
Zeit	t	20 ms
Winkel des max. Verfahrensweges	α	42,21 °

Mit der berechneten Geschwindigkeit des Strikers v_S (Kap. 3) lässt sich die Geschwindigkeit des Strikers in y-Richtung $v_{S,min,y}$ ableiten, welche der Geschwindigkeit der Querkonstruktion entspricht [TM, S. 19]:

$$v_{S,min,y} = v_{S,min} \cdot \sin(\alpha) = 1,476 \frac{\text{m}}{\text{s}}. \quad (4.7)$$

Damit ergibt sich die Beschleunigung a in y-Richtung näherungsweise zu [TM, S. 30]:

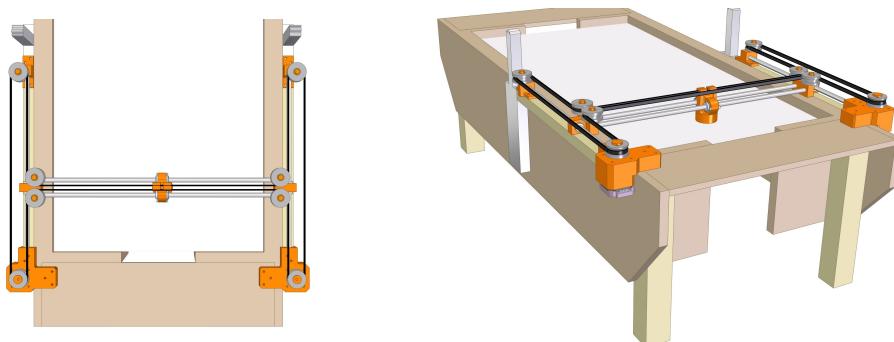
$$a = \frac{v_{S,min,y}}{t} = 73,78 \frac{\text{m}}{\text{s}^2}. \quad (4.8)$$

Mit der Masse der Querkonstruktion und deren Beschleunigung ergibt sich die Trägheitskraft in y-Richtung zu:

$$F_T = m_{QK} \cdot a = 24,59 \text{ N}. \quad (4.9)$$

Auslegung der Baugruppe:

Mithilfe der überschlägigen Berechnungen der Kräfte und der Anforderungen an die jeweiligen Bauteile können die einzelnen Bauteile dimensioniert und das ausgewählte Konzept realisiert werden. Dabei wurde besonders auf die Modularität der Konstruktion geachtet. Seitlich am Tisch wurden Holzleisten angebracht, an welchen der Roboter über Durchgangsverbindungen de-/montiert werden kann. Die Motoraufnahme wurde permanent auf einer austauschbaren Holzplatte befestigt. Dies bietet den Vorteil, dass die Konstruktion nach der Demontage weiterhin stabil ist und die Möglichkeit, diese durch einen alternativen Roboter zu ersetzen, weiterhin besteht.


Abbildung 4.4: CAD Konstruktion des Roboters

4.3 Motorauswahl

Auch wenn eine mögliche Motorisierung schon während der Konzeptfindung berücksichtigt wurde, muss die abschließende Auswahl der geeigneten Motorisierung fundiert in einem eigenständigen Arbeitspaket getroffen werden.

4.3.1 Motorarten

Zur Realisierung des ausgewählten Konzepts stehen drei gängige Elektromotoren zur Auswahl: Servomotoren, Schrittmotoren und Gleichstrommotoren.

Auf die Verwendung und Konstruktion eines zusätzlichen Getriebes soll aufgrund der zeitlichen Limitierung des Projekts verzichtet werden. Damit ergeben sich für die drei Motoren folgende Vor- und Nachteile [Adafruit-MSG]:

Tabelle 4.6: Vor- und Nachteile der Motorarten

Schrittmotor	Gleichstrommotor	Servomotor
+ Genaue Schritte	+ Hohe Drehzahlen möglich	+ Nur 180° schwenkbar
+ Unkomplizierte Positionssteuerung		+ Position sehr gut steuerbar
+ Geschwindigkeit sehr gut steuerbar		
+ Gängig in ähnlichen Anwendungen		
- Keine Erkennung bei übersprungenen Schritten	- Geschwindigkeit schlecht steuerbar	- Geschwindigkeit mäßig steuerbar
- Sehr hohe Drehzahlen selten	- Keine Positionskontrolle	- Geringe Fehlererkennung

Da die Kosten für die verschiedenen Motoren in einem vergleichbaren Rahmen liegen, wird dieser Faktor nicht weiter betrachtet. Auffällig ist jedoch, dass ein Gleichstrommotor aufgrund der schlechten Steuerbarkeit seiner Drehzahl und der fehlenden Positionskontrolle insgesamt ungeeignet für die spezielle Anwendung ist.

Ein Servomotor dagegen bildet durch seine sehr gute Positionssteuerbarkeit und die gute Steuerbarkeit seiner Drehzahl eine passende Alternative. Insbesondere die eingebaute Steuerung bei Servomotoren ist ein großer Vorteil, da es mit dieser möglich ist, den Motor zu regeln und ggf. auftretende Fehler wie z.B. Überdrehen zu erkennen und zu beheben.

Allerdings besitzen Servomotoren nur einen Bewegungsradius von 180° und sind damit ohne Getriebe für einen Riementrieb ungeeignet. Zwar werden auch freirotierende Servomotoren angeboten, diese sind allerdings sehr selten und teuer, sodass

auch der Servomotor keine zufriedenstellende Alternative für die Wahl des Motors bildet.

Der Schrittmotor dagegen scheint die meisten Kriterien hinreichend gut zu erfüllen. Lediglich das Fehlen einer Steuerung, die es ermöglicht, Fehler wie das Überspringen von Schritten zu erkennen, ist ein großer Nachteil. Diese Herausforderung kann allerdings konstruktiv und softwaretechnisch gelöst werden. Vorteilhaft ist, dass ein Schrittmotor aufgrund seiner Bauart um eine genau definierte Anzahl von Schritten bewegt werden kann. Damit ist das Steuern der Position des Motors sehr gut möglich und auch die Steuerung der Geschwindigkeiten ist vergleichsweise einfach zu realisieren.

Aufgrund dieser Eigenschaften werden Schrittmotoren bei Anwendungen mit einem ähnlichen Anforderungsbereich oft eingesetzt. So bilden Schrittmotoren z.B. bei Riementrieben von 3D-Druckern oder bei Linearachsen von CNC-Maschinen oft den Antrieb [Wiki-SMa]. Dass Schrittmotoren, die die benötigten Drehzahlen erreichen, in der benötigten Größenkategorie vergleichsweise selten zu finden sind, wird durch die ausgesprochen große Auswahl im Internet aufgrund des breiten Anwendungsbereichs relativiert.

Folglich ist die Wahl von Schrittmotoren zur Motorisierung des Air-Hockey-Roboters zu empfehlen. Ein möglicher Schrittmotor muss dabei die nachfolgend berechneten Motorkenngrößen erfüllen.

4.3.2 Motorkenngrößen

Um einen Schrittmotor auszuwählen, bedarf es der Berechnung der benötigten Motordrehzahl und des benötigten Motordrehmomentes. Diese werden im Folgenden aufgeführt.

Berechnung der benötigten Riemengeschwindigkeit:

Tabelle 4.7: Benötigte Größen zur Berechnung der Riemengeschwindigkeit

Bezeichnung	Formelzeichen	Wert
Reaktionszeit des Roboters	t_R	112,5 ms
Bespielbare Tischbreite	s_x	183,0 mm
Bespielbare Tischlänge	s_y	166,0 mm

In Kapitel 3.3 wurde die benötigte Strikergeschwindigkeit berechnet. Um daraus die benötigte Riemengeschwindigkeit abzuleiten, wird die Riemenlänge, die je Entfernung über eine Umlenkrollen laufen muss, veranschaulicht.

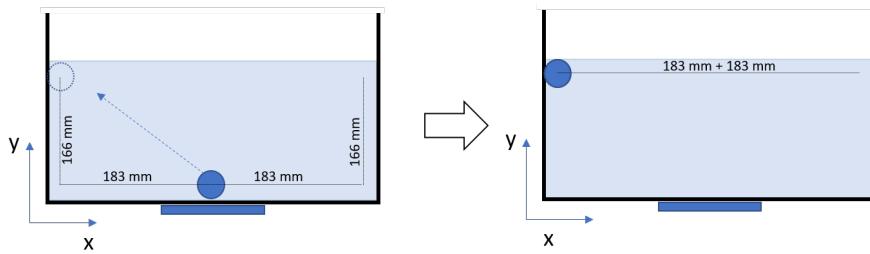


Abbildung 4.5: Abgerollte Riemenlänge bei Maximalbewegung des Strikers (links: Startposition, rechts: Endposition)

In der Startposition befindet sich der Striker vor dem eigenen Tor. In der Abbildung 4.5 ist die zu bespielende Fläche blau hinterlegt. Dabei muss sich der Striker 183 mm in positive und negative x-Richtung und 166 mm in positive y-Richtung bewegen können.

In einem „Worst Case-Szenario“ würde sich der Striker zur linken oberen Ecke der zu bespielenden Tischfläche bewegen. Dabei würden auf beiden Seiten jeweils 166 mm des Riemens in y-Richtung über die obere Umlenkrolle laufen. Außerdem würde sich der Riemen, der sich links vom Striker in x-Richtung befindet, um 183 mm verringern, um den Striker auf der Querkonstruktion zum linken Rand zu bewegen. Dabei wird die Riemenlänge, die sich rechts vom Striker auf der Querkonstruktion befindet, um 183 mm verlängern.

Diese Riemenlängen müssen an den Umlenkrollen passieren. Dabei werden die Umlenkrollen, welche dem gegnerischen Spielfeld am nächsten sind, betrachtet. Die zu passierende Riemenlänge berechnet sich aus der Differenz der Riemenlängen von Start- und Endposition.

Die zu passierende Riemenlänge $l_{R.l}$ ergibt sich an der linken, oberen Umlenkrolle zu:

$$\begin{aligned} l_{R.l} &= l_{R.l.\text{Startlänge}} - l_{R.l.\text{Endlänge}} \\ &= (166 \text{ mm} + 183 \text{ mm}) - 0 \text{ mm} = 349 \text{ mm} \end{aligned} \quad (4.10)$$

Da für beide Rollen von einem mathematisch positiven Drehsinn der Umlenkrollen ausgegangen wird und sich die Riemenlänge rechts vom Striker dadurch bereits „hinter“ der rechten Umlenkrollen befindet, muss bei dieser die Startlänge von der Endlänge abgezogen werden. Damit ergibt sich die zu passierende Riemenlänge $l_{R.r}$ an der rechten, oberen Umlenkrolle zu:

$$\begin{aligned} l_{R.r} &= l_{R.r.\text{Endlänge}} - l_{R.r.\text{Startlänge}} \\ &= (183 \text{ mm} + 183 \text{ mm}) - (183 \text{ mm} + 166 \text{ mm}) = 17 \text{ mm} \end{aligned} \quad (4.11)$$

Aus den positiven Ergebnissen der oben aufgeführten Rechnungen (4.10)(4.11) wird deutlich, dass sich beide oberen Umlenkrollen gegen den Uhrzeigersinn drehen müssen, damit der Striker von seiner Startposition zur „Worst Case–Position“ gelangt.

Dabei muss die linke Umlenkrolle mit einer höheren Geschwindigkeit als die rechte Umlenkrolle drehen, damit die komplette in (4.10) berechnete Länge passieren kann. Die rechte Umlenkrolle muss sich in der gleichen Zeit langsamer drehen und nur die in (4.11) berechnete Länge passieren lassen.

Es wird mit der Riemenlänge links aus (4.10) weitergerechnet, da es sich hierbei um die maximale Riemenbewegung handelt. Aus dieser und der Roboter-Reaktionszeit ergibt sich die minimal benötigte Riemengeschwindigkeit [TM, S. 30]:

$$v_{R,min} = \frac{l_{R,l}}{t_R} = 3,102 \frac{\text{m}}{\text{s}} \quad (4.12)$$

Berechnung der benötigten Motordrehzahl:

Tabelle 4.8: Benötigte Größen zur Berechnung der Motordrehzahl

Bezeichnung	Formelzeichen	Wert
Minimal benötigte Riemengeschwindigkeit	$v_{R,min}$	$3,102 \frac{\text{m}}{\text{s}}$
Radius der Motorumlenkrolle	r_{As}	15 mm

Nach dem Tabellenbuch Metall [TM, S. 30] ergibt sich mit der minimal benötigten Riemengeschwindigkeit und dem Radius der Motorumlenkrolle, welche sich auf der Motorwelle befindet und den Riemen führt, die minimal benötigte Drehzahl des Motors:

$$n_{min} = \frac{v_{R,min}}{2 \cdot \pi \cdot r_{As}} = 1984 \frac{1}{\text{min}}. \quad (4.13)$$

Berechnung des benötigten Drehmomentes:

Der Motor muss zur Erzeugung einer Bewegung die Trägheitskräfte der Konstruktion überwinden. Mit den zu überwindenden Trägheitskräften (vgl. Kapitel 4.2) und dem Radius der Motorumlenkrolle lässt sich das benötigte Drehmoment des Motors ableiten [TM, S. 35]:

$$M_{min} = F_T \cdot r_{As} = 0,369 \text{ J}. \quad (4.14)$$

4.3.3 Motorwahl: Schrittmotor NEMA 17

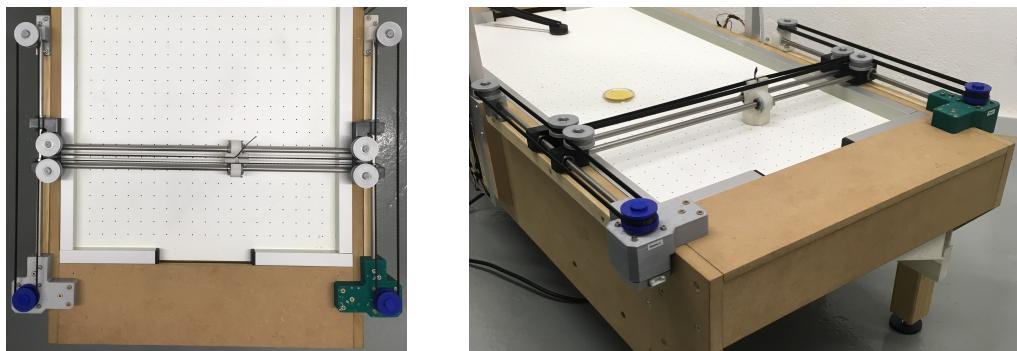
Für den mechanischen Antrieb der Konstruktion wird ein bipolarer Hybrid-Schrittmotor der Bauform NEMA 17 gewählt. Der Motor erfüllt die kinematischen Anforderungen an Drehzahl und Drehmoment und bietet ausreichende Möglichkeiten zur Ansteuerung. Nachfolgend sind die technischen Daten des Motors zusammengefasst. Eine ausführlichere Beschreibung kann dem Datenblatt im Anhang (Kap. 9) entnommen werden.

Tabelle 4.9: Technische Daten des ausgewählten Motors

Bezeichnung	Wert
Modell	JK42HS48-1684-08AF
Bauform	NEMA 17
Schrittwinkel	1,8°
Nennstrom	1,68 A
Nennspannung	2,8 V
Haltemoment	5 kg cm
Induktivität	2,8 mH
Anzahl der Leitungen	4

4.4 Ergebnisse

Durch eine intensive Konzeptentwicklung, eine ausführliche Berechnungsgrundlage und Motorauswahl konnte die Konstruktion des Roboters realisiert werden. Viele Bauteile wurden mittels additiver Fertigung (3D-Druck) angefertigt, nachbearbeitet und ggf. optimiert. Die Quer- und Seitstangen sowie Holzleisten wurden eigens zugeschnitten und die Wälz- und Linearlager zusammen mit den Motoren eingekauft. Das Ergebnis dieser langen Planungs- und Herstellungs- und Optimierungsphase ist ein einsatzbereiter Air-Hockey Roboter.

**Abbildung 4.6:** Konstruktion des Roboters

5 Softwaretechnischer Arbeitsstrang

Der zweite Arbeitsstrang des Projekts umfasst die elektronischen Einrichtungen, also die verbaute Hardware und deren Programmierung. Die Bezeichnung als „softwaretechnischer“ Arbeitsstrang ergibt sich aus der anfänglichen Einteilung der Arbeit in die Arbeitspakete Mechanik und Software, obwohl die gesamte Elektronik unter diesen Arbeitsstrang fällt.

Im Folgenden werden die jeweilige Konzepte einschließlich ihrer Umsetzung dargestellt. Außerdem wird abschließend eine Anleitung zur Modularität gegeben, welche das Austauschen einzelner Module des Air-Hockey Roboters erklärt und ein Weiterarbeiten erleichtern soll.

5.1 Konzeptentwicklung

Um den konstruierten Spieler spielfähig zu machen, wird ein Konzept für das Zusammenwirken von Software, Hardware und den jeweiligen Schnittstellen entwickelt. Die benötigte Software wurde in Pakete aufgeteilt und verschiedenen Elektronikbauteilen zugeordnet. Das Vorgehen der Konzeptentwicklung und das finale Hard- und Softwarekonzept wird im Folgenden vorgestellt.

5.1.1 Hardwarekonzept

Zu Beginn wurden drei Hardwarekonzepte erarbeitet, die sich in einzelnen Bauteilen bzw. Kommunikationswegen unterscheiden. In allen Konzepten wurden eine Pi Kamera als Kamera, mindestens ein Raspberry Pi sowie zwei Schrittmotoren verwendet.

Finales Konzept

Im ersten Schritt sendet eine Pi Kamera Bilder über eine Flachbandverbindung an einen ersten Raspberry Pi. Beide Bauteile sind bereits am Tisch vorhanden, da diese im Vorgängerprojekt genutzt wurden. Sie werden nicht zur Elektronik des aktuellen Projekts gezählt. Ziel ist es, eine parallele Nutzung des neuen und des bestehenden Roboters potenziell zu ermöglichen. Das darauffolgende Bauteil ist ein zweiter, gruppeneigener Raspberry Pi. Auf diesem findet die weitere Datenverarbeitung bestehend aus der Strategie und dem Code für das Menü statt. Außerdem wird dort die Kommunikation zu den Bedienungsknöpfen, zum LCD-Display und zur Motorsteuerung sichergestellt.

Die Berechnung der nötigen Motorbewegung findet auf dem nächsten Bauteil, einem leistungsstarken Arduino, statt. Dieser sendet Signale an zwei Motortreiber, die die Schrittmotoren steuern.

Das Konzept ist in Abbildung 5.1 übersichtlich dargestellt. In dieser Abbildung ist auch die Verortung der Software zu sehen, die in Kapitel 5.1.3 genauer beschrieben wird.

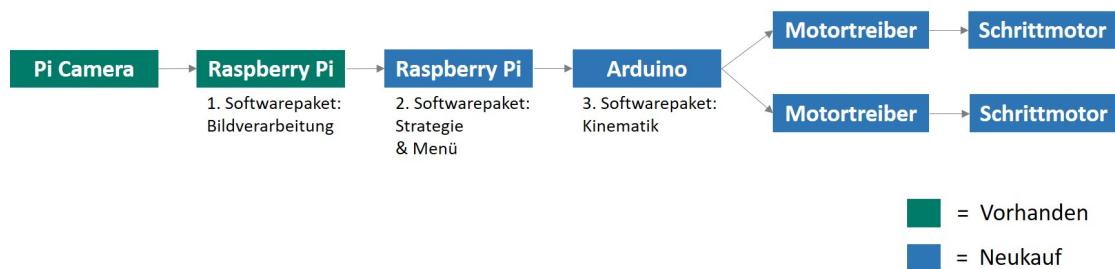


Abbildung 5.1: Finales Hardwarekonzept

Dieses Konzept wurde als beste Lösung identifiziert, da alle angestrebten Anforderungen erfüllt werden. Das System ist vollständig am Tisch verbaut, weshalb der Roboter betrieben werden kann, ohne dass zusätzliche Errichtungen für das Spiel notwendig sind. Außerdem besteht die Möglichkeit, das System um weitere Bauenteile bzw. Softwarepakete zu erweitern bzw. Teile zu verändern. Auf dem zweiten Raspberry Pi könnte beispielsweise sehr einfach ein umfangreicheres Strategieprogramm installiert werden. Wichtig ist auch, dass sich die Bildverarbeitung beider Roboter auf dem selben Raspberry Pi befindet, da dieser über das Flachbandkabel mit der Kamera verbunden ist.

Alle elektronischen Bauteilen sollen übersichtlich angeordnet werden. Lediglich der erste Raspberry Pi sowie die Pi Kamera werden an der Stelle belassen, an der diese im Vorgängerprojekt angeordnet wurden, um die Gegebenheiten nicht zu verändern. Die Motoren befinden sich direkt am Roboter.

Weitere Konzeptentwürfe

Zunächst war die Verwendung von ausschließlich zwei Raspberry Pis vorgesehen. Dadurch sollte die nötige Leistung zur Datenverarbeitung sichergestellt werden, indem die Softwarepakete auf die beiden Raspberry Pis aufgeteilt werden. Ein Raspberry Pi ist jedoch nicht echtzeitfähig, wodurch dieser für die Ansteuerung der Motortreiber nicht sinnvoll verwendet werden kann. Aus diesem Grund wurde zusätzlich zum zweiten Raspberry Pi ein Arduino eingeplant. Dadurch sollte zudem sichergestellt werden, dass die Bauteile über genug Leistung verfügen.

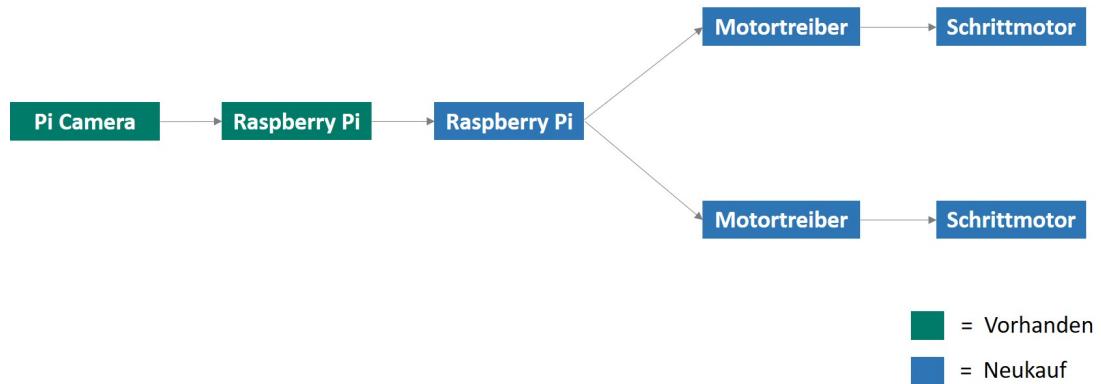


Abbildung 5.2: Konzeptentwurf 2: Zwei Raspberry Pis

Das anfänglich ausgewählte Konzept, zwei Raspberry Pis zu verwenden, stellt den zweiten Entwurf dar (vgl. Abbildung 5.2).

Das dritte Konzept schließt die Verwendung eines Laptops zur Datenverarbeitung ein. Der Datenfluss ist in Abbildung 5.3 dargestellt.

Beide Entwürfe erfüllen jedoch die festgelegten Anforderungen nicht vollständig, ergeben keinen funktionstüchtigen Roboter bzw. verkomplizieren das System unnötig. Würde beispielsweise ein Laptop verwendet werden, müsste der Benutzer des Tisches bei jeder Benutzung einen Laptop mit dem programmierten Code mitführen, was zusätzlicher Aufwand zur Benutzung des Roboters bedeutet. Dies entspricht nicht der Anforderung, dass alle Bauteile des Roboters am Tisch verbaut sind und dieser somit eine abgeschlossene Einheit bildet.

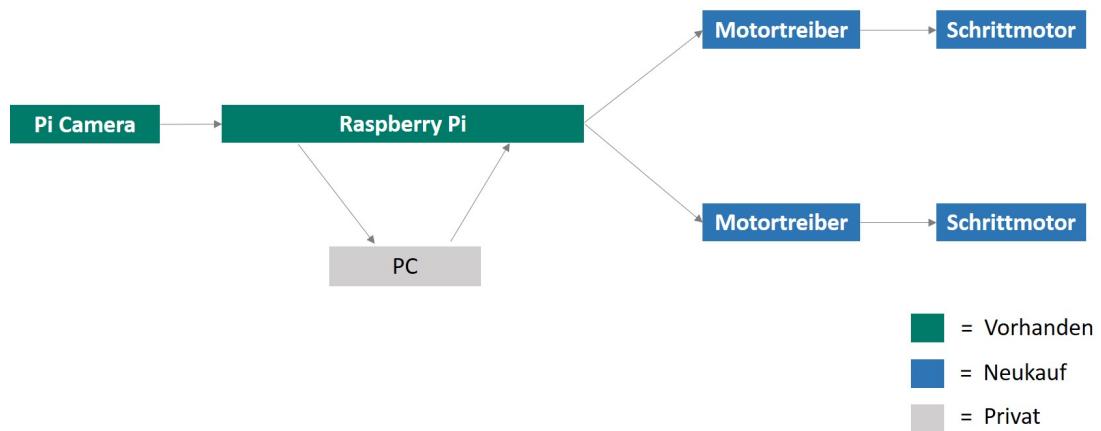


Abbildung 5.3: Konzeptentwurf 3: Datenverarbeitung mit einem PC

Bei alle entwickelten Konzepten wird davon ausgegangen wird, dass die Leistung und Schnelligkeit der Raspberry Pis in Kombination mit den Arduinos für die Datenverarbeitung ausreicht.

5.1.2 Hardwarekomponenten

Im Folgenden werden die verwendeten Komponenten aufgeführt (vgl. Abbildung 5.4) und die Wahl der Bauteile begründet.

Die Pi Kamera ist eine Raspberry Pi Kamera V1.3, die über ein Flachbandkabel zur Verbindung mit einem Raspberry Pi verfügt. Diese ist bereits am Tisch verbaut und wird auch in der eingebauten Position belassen. 60 Bilder können bei 720p pro Sekunde aufgenommen werden [RaspberryPiCamerab]. Durch diese Frequenz der Aufnahmen können genug Bilder zur Verfügung gestellt werden.

Zur Datenverarbeitung werden zwei Raspberry Pi 3+ verwendet. Dies ist ein Einplatinencomputer mit einer 64-Bit und einem Vierkern-Prozessor [RaspberryPi]. Er besitzt ein CSI camera port, an den die Raspberry Pi Kamera angeschlossen werden kann. Bei der Stromversorgung werden 5 Volt und 2.5 Ampere benötigt, was beim Anschließen des Raspberry Pis beachtet werden muss. Ein Raspberry Pi befindet sich bereits aus dem Vorgängerprojekt am Tisch, während der andere Raspberry Pi als identisches Modell hinzugekauft wurde.

Für die Motoransteuerung besonders geeignet ist ein Arduino. Für das Projekt wird daher der leistungsstärkste Arduino DUE eingesetzt. Auf diesem Microcontroller kann die für gegebene Zielkoordinaten notwendige Kinematik berechnet werden. Bei diesem Modell muss jedoch beachtet werden, dass anders als viele Arduinos dieser mit einer Betriebsspannung von 3,3V betrieben wird. Eine Betriebsspannung von sonst üblichen Spannung 5V könnte den Arduino DUE zerstören.

Sowohl der Arduino als auch die Raspberry Pis werden über weitere Hardwarekomponenten, die Step-Down-Converter, mit einer Spannung von 5V versorgt,

Die vom Arduino errechneten Steuerinformationen werden von weiteren Komponenten, den sog. Motortreibern, in konkrete Spannungswerte umgerechnet.

Wie in Kapitel 4.3 beschrieben, werden zwei Schrittmotoren der Bauform NEMA 17 als Antrieb des Riementriebs gewählt.

Das Menü, welches am Tisch verbaut wurde, wird auf einem 20x4 Zeichen LCD Display Modul angezeigt. Dieser verfügt über einen HD44780U Controller zur Ansteuerung und über einen PCF8574T I²C-Adapter zur Kommunikation. Zur Bedienung sind zusätzlich 3 Buttons installiert, die über die GPIO Pins 21, 22 und 26 mit dem Raspberry Pi verbunden sind. Damit kann im Menü hoch und runter gesteuert oder mit Enter bestätigt werden.

Die über das Menü einleitbare Kalibrierung ist hardwaretechnisch mittels einfacher Endschalter gelöst.

Alle Komponenten werden dabei von einem Netzteil mit Leistung versorgt. Dabei handelt es sich um einen *Mean Well 36V LRS-200-36*, der 36V Nennspannung liefert. Zusätzlich lässt dieser sich im Bereich von ca. 30-40V feinjustieren.

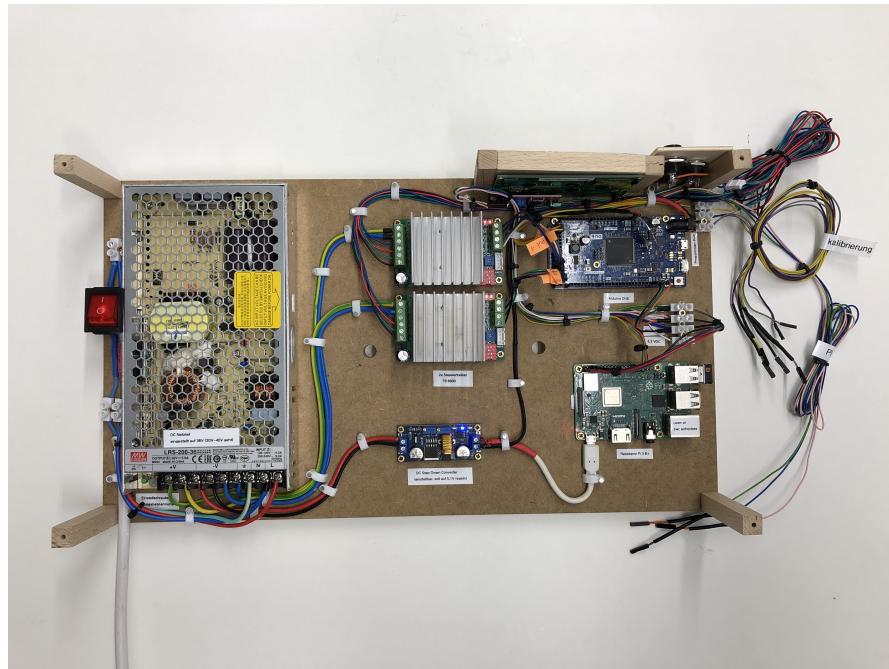


Abbildung 5.4: Elektronikkomponenten des Roboters

5.1.3 Softwarekonzept

Die benötigte Software wurde in drei Pakete eingeteilt, die unabhängig voneinander entwickelt wurden. Ein Paket kann als gesamtes Paket ersetzt oder innerhalb des Pakets bearbeitet werden. In der Abbildung 5.1 ist bereits die Abfolge der Softwarepakete und die Zuordnung zu den entsprechenden, elektronischen Bauteilen dargestellt. Neben den drei Paketen wurde zusätzlich ein Menü erstellt, welches die Steuerung des Spiels ermöglicht.

Das erste Paket „Bildverarbeitung“, welches in Kapitel 5.2 erläutert wird, setzt sich aus der Bildaufnahme, der Positionsbestimmung und zusätzlich auch der Geschwindigkeitsvektorberechnung des Pucks zusammen. Es befindet sich auf dem ersten Raspberry Pi.

Das zweite Paket „Strategie“ beinhaltet die Prognose der Puckbahn und berechnet, wann und an welcher Stelle der Puck eine festgelegte Abwehrlinie überschreitet. Daraus ergibt sich die Zielposition, zu der sich der Striker bewegen muss. Die genaue Funktionsweise ist in Kapitel 5.3 erklärt. Dieses Paket befindet sich auf dem zweiten Raspberry Pi. Dadurch läuft es parallel zur Bildverarbeitung. Ebenfalls auf dem zweiten Raspberry Pi befindet sich das Menü. Die verschiedenen Optionen des Menüs und ihre Funktionen werden im Kapitel 5.5 beschrieben.

Das dritte Paket „Kinematik“ kommuniziert mit den Motortreibern und berechnet die benötigten Motorenbewegungen für das Erreichen der Zielposition. Dafür wird die geometrische Bewegung zur Zielposition in Umdrehungen der Motoren umgewandelt.

Der Code befindet sich auf dem Arduino. Das kinematische Konzept, die theoretischen Überlegungen und die Umsetzung werden in Kapitel 5.4 beschrieben.

Damit alle Softwarepakete miteinander Daten austauschen können, wurde ein einheitliches Koordinatensystem gewählt, in dem die Koordinaten an den Schnittstellen weitergegeben werden. Die Lage dieses Koordinatensystems ist in Abbildung 5.5 dargestellt. Der Nullpunkt liegt im Strikermittelpunkt, wenn sich dieser in der linken, unteren Ecke des Tisches befindet. Hier ist der Radius des Strikers von 39mm berücksichtigt. Ebenfalls sind in der Abbildung die Grenzen des bespielbaren Bereichs angetragen. Diese liegen bei $x = 361$ (Tischbreite abzüglich Striker-Durchmesser) und $y = 215$ (durch den mechanischen Aufbau begrenzte Fläche abzüglich des Striker-Durchmessers).

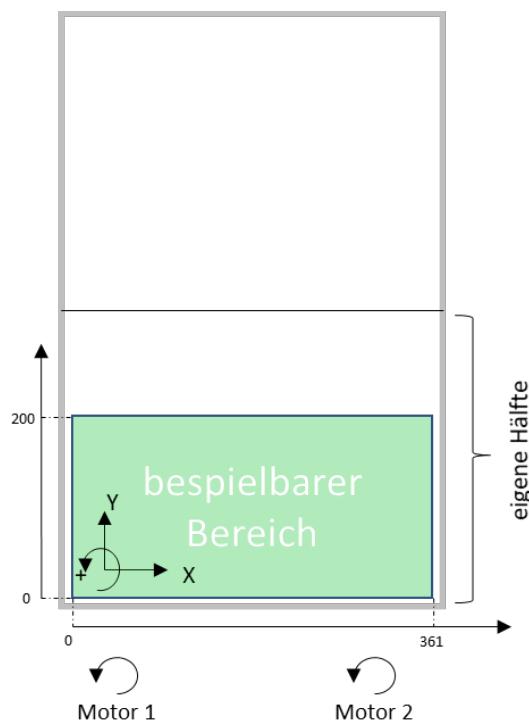


Abbildung 5.5: Kinematik-Koordinaten und Spielfeldhälften

5.1.4 Schnittstellengestaltung und Kommunikation

Die Datenkommunikation zwischen den einzelnen Komponenten verläuft über den I²C Datenbus. I²C steht für inter-integrated circuit und ist eine verbreitete Schnittstelle, sodass sowohl für Raspbian als auch für Arduino viel Nachschlagewerk und mehrere Bibliotheken vorliegen. Eine Alternative dazu wäre die Serial Peripheral Interface (SPI) Schnittstelle, jedoch hat I²C den Vorteil, dass weniger Leitungen verlegt werden müssen.

In einem I²C-Netzwerk werden alle Geräte an zwei Signalleitungen angeschlossen, der Takteleitung SCL und der Datenleitung SDA. Die Komponenten können entweder als Master oder Slave agieren. Allein der Master darf in dem Netzwerk

die Versendung der Daten initiieren, er kann die Daten entweder von einem Slave anfordern oder einem Slave zuschicken. Jede angeschlossene Komponente besitzt eine Adresse, mit der eine eindeutige Zuordnung besteht, welches Gerät vom Master angesteuert wird.

Der Aufbau der Datenübertragung über I²C wird im Folgenden anhand der Abbildung 5.6 beschrieben. Der Master sendet ein Startbefehl, gefolgt von der 7-bit Adresse des Slaves und einem sogenannten Read/Write Bit. Dieser kommuniziert, wie der Name sagt, ob die Übertragung vom Slave zum Master oder umgekehrt verläuft. Danach beginnt die Datenübertragung, bei der abwechselnd Daten und Bestätigungsbits verschickt werden. Abschließend sendet der Master ein Stopbefehl.

In diesem Projekt fungiert der zweite, gruppeneigene Raspberry Pi als Master, dabei sind der erste Pi, der Arduino und das Display Slaves. Die Adresse des Displays ist 0x27, die Adressen des Arduinos und des ersten Pis sind frei einstellbar im Code und werden zu 0x07 für den Arduino und 0x17 für den Pi gewählt.

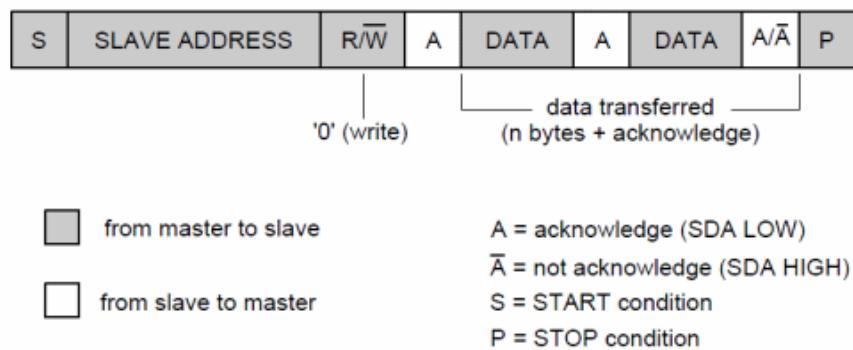


Abbildung 5.6: Aufbau der I2C Datenübertragung [I2C]

5.2 Bildverarbeitung

Die Bildverarbeitung lässt sich in mehrere Teilschritte einteilen. Innerhalb dieser Projektarbeit wird zur Bildverarbeitung nicht nur die Phase der Bilderfassung und -verarbeitung bis zum Auslesen der Koordinate des Pucks gezählt, sondern auch eine erste Betrachtung dieser Koordinate und Berechnung der Geschwindigkeit.

Es ergeben sich damit die Teilschritte: Aufnahme des Bildes durch die Pi Kamera (Kapitel 5.2.2), Umwandlung und Analyse des Bildes zur Bestimmung der Koordinaten (Kapitel 5.2.3) und Berechnung der Geschwindigkeit aus der bestimmten Position unter Beachtung von Bandenkontakt (Kapitel 5.2.4). Im Folgenden wird zunächst das Konzept (Kapitel 5.2.1) und der angestrebte Ablauf dargestellt. Danach werden die einzelnen Teilschritte detailliert beschrieben.

5.2.1 Konzept

Die Bildverarbeitung wird hardwaretechnisch, wie in 5.1 beschrieben, durch eine Pi Kamera und einem mit dieser verbundenen Raspberry Pi ausgeführt. Der Raspberry Pi erhält bei Betrieb dauerhaft Bilder, wandelt diese um, erkennt die Koordinate und führt erste Berechnungen durch. Eine Schleife umfasst dabei die Aufnahme eines Bildes bis zur berechneten Geschwindigkeit des Pucks. Mit jedem neuen Bild startet eine neue Schleife. Die Position sowie die Geschwindigkeit des Pucks werden nach Durchlauf einer Schleife als Schnittstelle an das zweite Softwarepaket, Strategie, für weitere Berechnungen übergeben.

Im Flussdiagramm 5.7 sind die Funktionsweise, die Verarbeitungsschritte und die Weitergabe der Daten dargestellt.

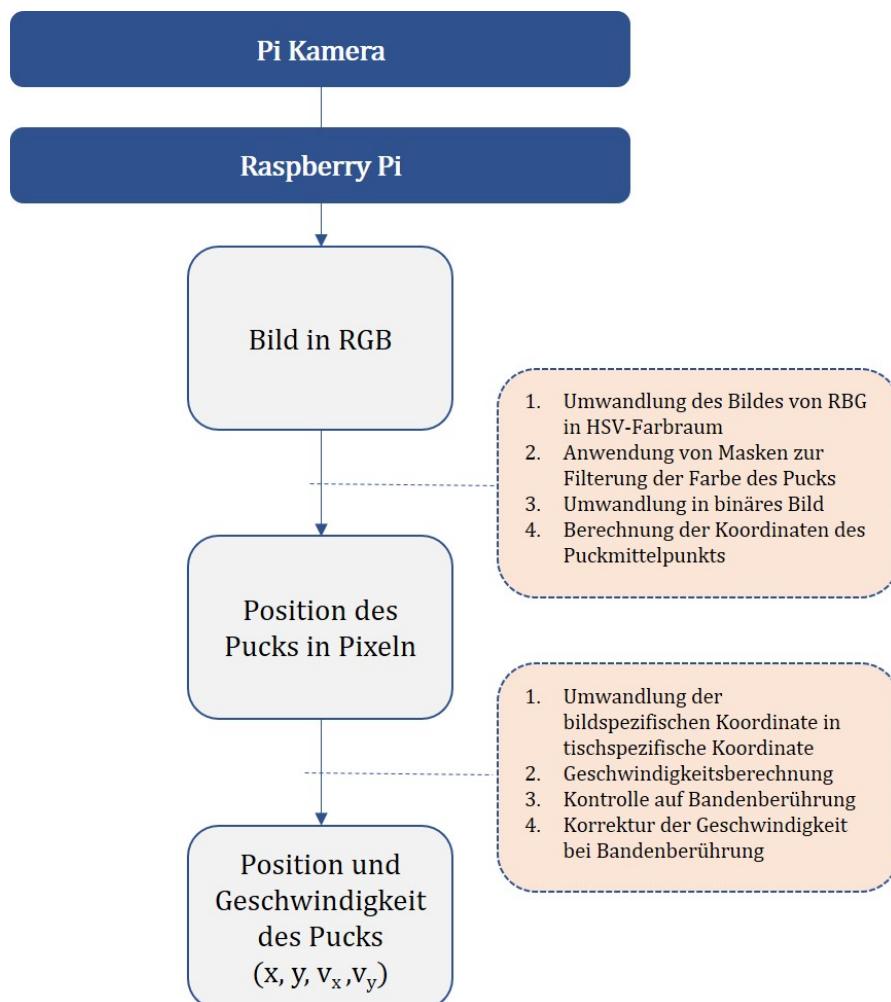


Abbildung 5.7: Konzeptioneller Ablauf des Softwarepaketes Bildverarbeitung

5.2.2 Bilddurchlauf

Der erste Teilschritt des Softwarepaket Bildverarbeitung ist die Bilderfassung. Die Pi Kamera befindet sich an einer bereits vorhandenen Halterung mittig über dem Tisch. So ist eine Aufnahme des gesamten Air-Hockey-Tisches möglich. Die Kamera wird hochgefahren und macht dann durchgängig mit einer Frequenz von 30 fps Fotos, die dann weiter verarbeitet werden. Angestrebt war eine Frequenz von 60 fps, jedoch konnte dies bisher nicht realisiert werden. Damit die Kamera den gesamten Tisch aufnimmt, sollte diese in einem Probeflug aufgerichtet werden.

Die Bilder werden im Farbraum RGB aufgenommen und haben die Abmaße 1280 x 640 Pixeln. Diese Abmaße ergeben sich aus der Form des Air-Hockey-Tisches, dessen Länge mit 800 mm doppelt so lang ist wie die Weite mit 400 mm. Die Pixel können dann in Millimeter umgerechnet werden.

5.2.3 Bildanalyse

Ein aufgenommenes Bild wird im nächsten Teilabschnitt des Codes in verschiedene Farbräume umgewandelt und gefiltert, um den Puck zu erkennen und die Koordinaten des Pucks auszulesen.

Zunächst wird das originale RGB-Bild in den HSV-Farbraum umgewandelt. Die gelbe Farbe des Pucks wird mithilfe festgelegter Werte für den Farbton (Hue), die Sättigung (Saturation) und die Helligkeit (Value) im Bild erkannt und herausgefiltert. Mit weiteren Filtern werden weitere kleinere erkannte Farbflächen herausgefiltert. Bei der Filterung wurden die entsprechenden Parameter an das Bild angepasst. Sollten sich beispielsweise die Lichtverhältnisse durch eine Veränderung der Beleuchtung des Tisches ändern, sollte die Filterung auf mögliche Anpassungen kontrolliert werden.

Schließlich wird ein Binärbild erzeugt, auf dem nur die erkannte gelbe Farbe in weiß dargestellt ist, wodurch der Puck deutlich zu erkennen ist. Der Mittelpunkt des Pucks wird bestimmt und als Variable zur weiteren Berechnung gespeichert. Die verwendeten Funktionen in diesem Teil stammen zu großen Teilen aus OpenCV.

Der Ablauf der Bildverarbeitung wird in Abbildung 5.8 dargestellt. Das Originalbild von der Pi Kamera wird im RGB-Farbraum aufgenommen, in den HSV-Farbraum umgewandelt, gefiltert und ein binäres Bild wird erzeugt. In diesem binären Bild wird der Mittelpunkt des Pucks bzw. der noch sichtbaren Fläche ermittelt. In Abbildung 5.8 im Bild unten rechts wird zur besseren Veranschaulichkeit zusätzlich der erkannte Umriss angezeigt.

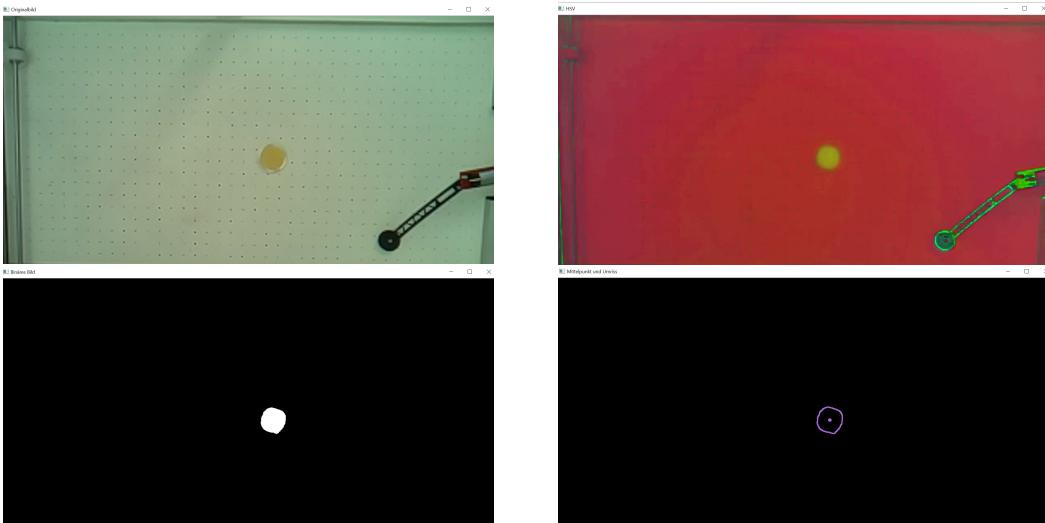


Abbildung 5.8: Umwandlung des Bildes und Mittelpunktbestimmung

Damit im Folgenden mit Millimetern anstelle von Pixeln gerechnet werden kann, müssen die Koordinaten in mm weitergegeben werden. Dazu werden die 1280 bzw. 640 Pixel auf 800 bzw. 400 Millimeter umgerechnet. Da ein einheitliches Koordinatensystem für alle weiteren Berechnungen in den Softwarepaketen gewählt wurde, müssen an dieser Stelle Änderungen vorgenommen werden, wenn die Aufnahme der Kamera und damit das kameraspezifische Koordinatensystem verändert wird (vgl. 5.1.3).

Sobald das Bild an den Raspberry Pi gesendet wird, wird außerdem die Zeit ausgelesen und abgespeichert. Damit kann die zeitliche Differenz zwischen den Bildern erkannt werden und die Differenz für die Geschwindigkeitsberechnung genutzt werden.

5.2.4 Berechnungen zu Position und Geschwindigkeit

Im folgenden Schritt werden die Koordinaten für weitere Berechnungen genutzt. Aus den Positionen im letzten und im aktuellen Foto kann eine Gerade gezogen werden und mithilfe der Zeitdifferenz zum letzten Foto die Geschwindigkeit errechnet werden. Es wird ein Geschwindigkeitsvektor in x und y-Richtung aufgestellt.

Bei dieser Berechnung muss jedoch beachtet werden, ob möglicherweise die Bande berührt worden ist. Dazu wird der letzte Geschwindigkeitsvektor genutzt, um zu bestimmen, wo sich der Puck bei dieser Geschwindigkeit im aktuellen Foto befinden müsste. Wenn dieser errechnete Punkt in y-Richtung außerhalb der Tischfläche liegt, wurde eine Bande zwischen dem letzten und dem aktuellen Bild berührt. Es wird also eine Berechnung zum Bandenkontakt durchgeführt, bei der der Kontaktspunkt in x- und y-Richtung bestimmt wird. Dieser Punkt lässt sich mithilfe der letzten Geschwindigkeit und der Zeitdifferenz berechnen. Vom Kontaktspunkt aus wird der neue Geschwindigkeitsvektor berechnet, indem eine lineare Bewegung zur aktuellen Position in der verbliebenen Zeit angenommen wird.

Die Berechnung der Geschwindigkeit des Pucks wird mit jedem neuen Bild durchgeführt. Daten in Form von (x, y, v_x, v_y) sind als Schnittstelle vorgesehen: Position in x-, Position in y, Geschwindigkeit in x- und Geschwindigkeit in y-Richtung. In 5.3 wird erläutert, wie daraus innerhalb des Pakets Strategie eine Zielposition zur Abwehr des Pucks bestimmt wird.

5.3 Spielstrategie

Um den Roboter in seinen grundlegenden Funktionen zu testen und ihn zu einem funktionsfähigen Gegenspieler für den Menschen oder andere Roboter zu machen, wurde ein einfaches Verteidigungsprogramm geschrieben, welches Gegentore verhindern soll.

5.3.1 Konzept

Die Strategie läuft zusammen mit der Bildverarbeitung auf dem Raspberry Pi. Mit jedem Bild, das die Kamera macht, wird an die Strategie die aktuellen Koordinaten (x,y) und die aktuelle Geschwindigkeit des Pucks (v_x,v_y) übergeben. Daraus wird die zukünftige Bahn des Pucks vorhergesagt und ein Schnittpunkt mit der Aktionslinie berechnet. In Abbildung 5.9 werden die einzelnen Schritte des Strategieprogramms als Flussdiagramm dargestellt.

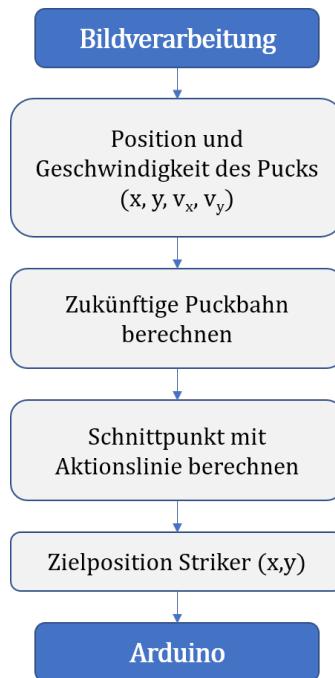


Abbildung 5.9: Flussdiagramm

5.3.2 Simulation

Die Strategie kann nur an einem fertigen Roboter und mit einer funktionierenden Bildverarbeitung ausgeführt werden. Deshalb wurde, um effizient an einer Stra-

tegie arbeiten zu können undum sie testen zu können, ein Simulationsprogramm geschrieben, welches auf den Funktionen der Bibliothek SFML (Simple and Fast Multimedia Library) basiert. Die Simulation erstellt ein Fenster, welches das Spielfeld des Air-Hockey-Tisches darstellt und jeweils den Striker und den Puck als Kreise darin abbildet. Das Fenster wurde so erstellt, dass ein Pixel in der Simulation genau einem Millimeter auf dem Tisch entspricht.



Abbildung 5.10: Simulation Ausgangszustand

Die Abbildung 5.10 zeigt die Simulation in ihrem Ausgangszustand. Der Striker ist der blaue Kreis und befindet sich auf der linken Seite des Feldes. Der Puck ist als weißer Kreis dargestellt und hat seinen Ausgangszustand auf der gegenüberliegenden Seite. Der Ursprung des Koordinatensystems befindet sich in der oberen linken Ecke des Spielfeldes. Allerdings wurde der Koordinatenursprung um den Radius des Strikers in die Mitte versetzt, um ausschließlich die bespielbare Fläche zu simulieren. So ergibt sich eine simulierte Spielfeldgröße von 361 mmx761 mm.

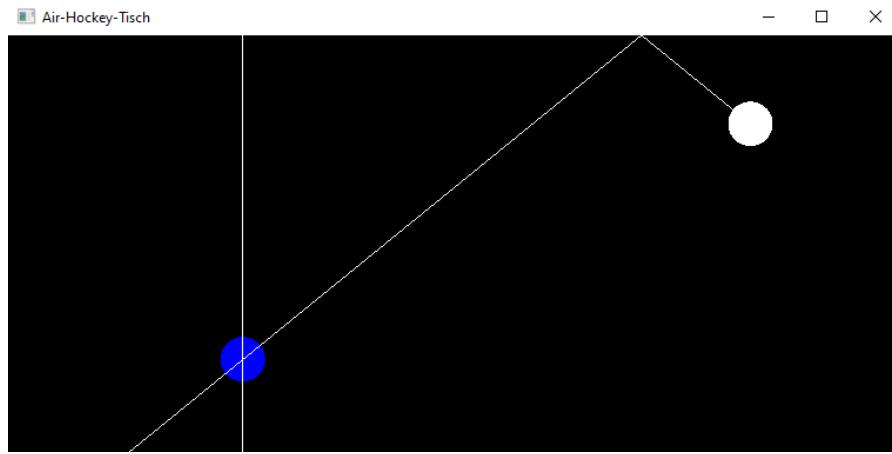


Abbildung 5.11: Simulation

Die Simulation wird wie folgt gesteuert: Wenn mit der linken Maustaste in das Fenster geklickt wird, bewegt sich der Puck in diese Richtung. Damit wird ein Angriffsschlag eines Gegenspielers simuliert. Bewegt der Puck sich auf den Striker zu,

wird wie in Abbildung 5.11 zu sehen, der weitere Verlauf der Puckbahn durch eine dünne Linie dargestellt. Es wurde außerdem darauf geachtet, dass die Bewegungen des Pucks ungefähr der Realität entsprechen. Dazu werden u.a. die Winkel beim Abprallen vom Striker und der Banden berechnet.

Die Aufgabe der Simulation ist es auch, die Bildverarbeitung zu simulieren. So gibt sie bei jedem Durchlauf (ähnlich zu den Frames der Pi Kamera) die Daten der Position und Geschwindigkeit des Pucks aus, welche für die Berechnungen der Strategie essenziell sind.

5.3.3 Bahnprediktion

Um den Puck erfolgreich abwehren zu können, ist es notwendig, die ungefähre Bahn des Pucks vorhersagen zu können. Voraussetzung für die Berechnung der Bahnprediktion ist, dass der Puck sich in Richtung des Tors bewegt. Tritt dieser Fall ein, wird aus den von der Bildverarbeitung übermittelten Daten (x-, y-Position und x-, y-Geschwindigkeit) eine Geradengleichung aufgestellt.

$$y = a \cdot x + b \quad (5.1)$$

Dafür wird zuerst die Steigung a der Puckbahn aus dem Verhältnis der Geschwindigkeiten v_x und v_y berechnet.

$$a = \frac{v_y}{v_x} \quad (5.2)$$

Der Wert b , um den die Gerade der Bahn in y-Richtung verschoben ist, kann mit Hilfe der aktuellen Koordinaten x_{akt} , y_{akt} und der schon davor errechneten Steigung a bestimmt werden.

$$b = x_{akt} - y_{akt} \cdot a \quad (5.3)$$

Die Verteidigungsstrategie ist sehr rudimentär gehalten, um späteren Gruppen Raum für ausgefeilte, selbstlernende Strategien zu geben. Sie funktioniert nach dem Prinzip, eine Kollision zwischen Puck und Striker zu verursachen. Dazu wurde eine Aktionslinie festgelegt, die sich auf der Hälfte der bespielbaren Fläche des Roboters bei $y = 100$ befindet. Um den Schnittpunkt mit der Aktionslinie zu erhalten, wird $y = 100$ in die Formel (5.1) eingesetzt.

Im nächsten Schritt wird überprüft, ob der Schnittpunkt innerhalb zu bespielbaren Fläche liegt. Wenn dies der Fall ist, wird der Puck, ohne vorher eine Bande zu berühren, die Aktionslinie schneiden. Der Schnittpunkt ist damit bestimmt und die Zielposition, zu der sich der Roboter bewegen soll, wird an die Akteure weitergegeben.

Bewegt sich der Puck jedoch nicht auf direktem Weg Richtung Aktionslinie, also liegt der zunächst berechnete Schnittpunkt nicht in dem Feld, kommt es zwangsläufig zu einem oder mehreren Bandenkontakten. Deshalb muss ein neuer Schnittpunkt

berechnet werden, welcher davon abhängt, wie viele Bandenkontakte noch ausstehen und mit welcher Bande der Puck als nächstes kollidieren wird. Dazu wird die in Abbildung 5.12 dargestellte Fallunterscheidung getroffen. Denn der neue Schnittpunkt hängt von der geraden oder ungeraden Anzahl an Bandenkontakten, aber auch von der Seite des Kontaktes ab und wird entsprechend unterschiedlich berechnet.



Abbildung 5.12: Schnittpunktberechnung Fallunterscheidung

Die Strategie gibt nach jeder Berechnung einen Zielpunkt für den Striker an den Arduino DUE weiter. Wenn es zu einer Kollision zwischen Striker und Puck gekommen ist und der Puck sich daher vom Striker wieder entfernt, wird kein neuer Schnittpunkt berechnet. Stattdessen wird als Zielkoordinate eine Verteidigungsposition vor dem eigenen Tor an den Arduino DUE weitergegeben, bis der Puck sich schließlich auf das eigene Tor zubewegt und ein neuer Schnittpunkt mit der Aktionslinie berechnet werden kann.

5.4 Kinematik

Die „Kinematik“ hat primär die Aufgabe, die abstrakten Bewegungsanweisungen der Strategiesoftware in entsprechende, für die Motortreiber zugeschnittene Signale umzuwandeln, welche dann die von der Strategie gewünschte Bewegung mithilfe der Motoren umsetzen.

Dafür ist einerseits ein großes Verständnis für die verwendete Hardware, deren Eigenschaften und Ansteuerungsmöglichkeiten erforderlich. Andererseits müssen die kinematischen Beziehungen des mechanischen Aufbaus analysiert werden, um eine Beziehung von Größen des abstrakteren, kartesischen Koordinatensystems zur jeweiligen Motorpositionen und Drehzahlen herzustellen. Dies ist der Grund, warum die Namensgebung als *Kinematik*-Paket sinnvoll ist.

5.4.1 Kinematische Beziehungen

Aufgrund des im Vergleich zu üblichen Portalrobotern [Wiki-Rob] viel komplizierteren *H-Aufbaus*, anstatt des simplen Standardaufbaus mit geraden Achsen, wird sich an erster Stelle mit den mechanischen Zusammenhängen befasst.

Hierzu ist Abbildung 5.13 zu betrachten, um die Bewegungsmöglichkeiten genauer zu untersuchen.

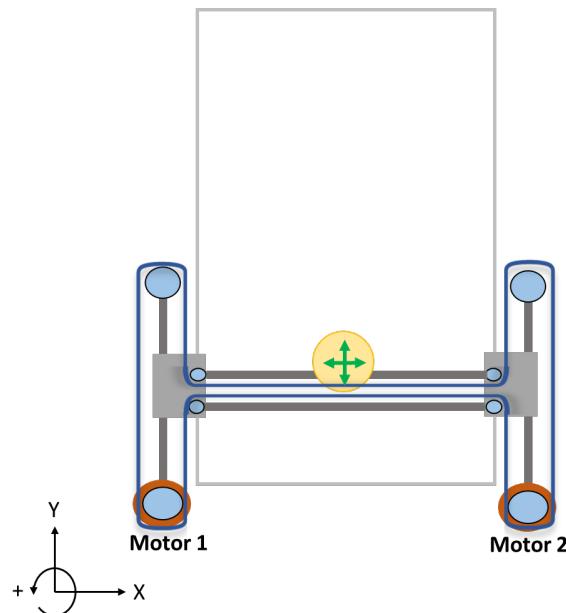


Abbildung 5.13: Veranschaulichung der Kinematik

Wie bereits in Unterkapitel 4.1.1 Konzeptmöglichkeiten erklärt, kann sich der Striker bei gegensinniger Motordrehung in Y-Richtung und bei gleichsinniger Motordrehung in X-Richtung bewegen. Konstellationen dazwischen führen zu überlagerter Bewegung.

Bei genauerer Betrachtung wird klar, dass die einzelnen Motoren die in Abbildung 5.14 abgebildeten Bewegungsachsen besitzen und die Gesamtbewegung aus der Kombination dieser Bewegungsachsen resultiert. Die Achsen bilden den Zustand bei jeweils positivem Drehsinn ab, nach Koordinatensystem aus Abbildung 5.13.

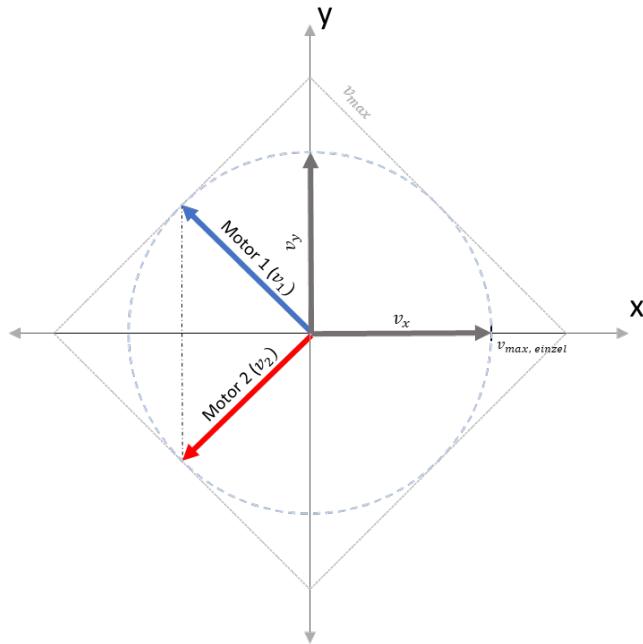


Abbildung 5.14: Bewegungsachsen der Motoren

In blau und rot die Einzelgeschwindigkeiten der Motoren, Einheitslänge zur Darstellung der maximalen Geschwindigkeit. Durch Vektoraddition ergibt sich insgesamt das grau umrandete, quadratische Geschwindigkeitsfeld, welches die Gesamtheit der möglichen Geschwindigkeitsvektoren abbildet.

Es ist erkennbar, dass die absolute Geschwindigkeit in den Hauptachsen am größten ist, ($\sqrt{2}$ -fache der Einzelbeschleunigungen) und dass in Diagonalrichtung am langsamsten beschleunigt werden kann.

Um den Bewegungszusammenhang noch weiter zu verdeutlichen dient Abbildung 5.15, welche erneut den Bereich der möglichen Striker-Geschwindigkeiten zeigt, sowie einer Tabelle zur Veranschaulichung der zugehörigen Drehzahl-Konstellationen.

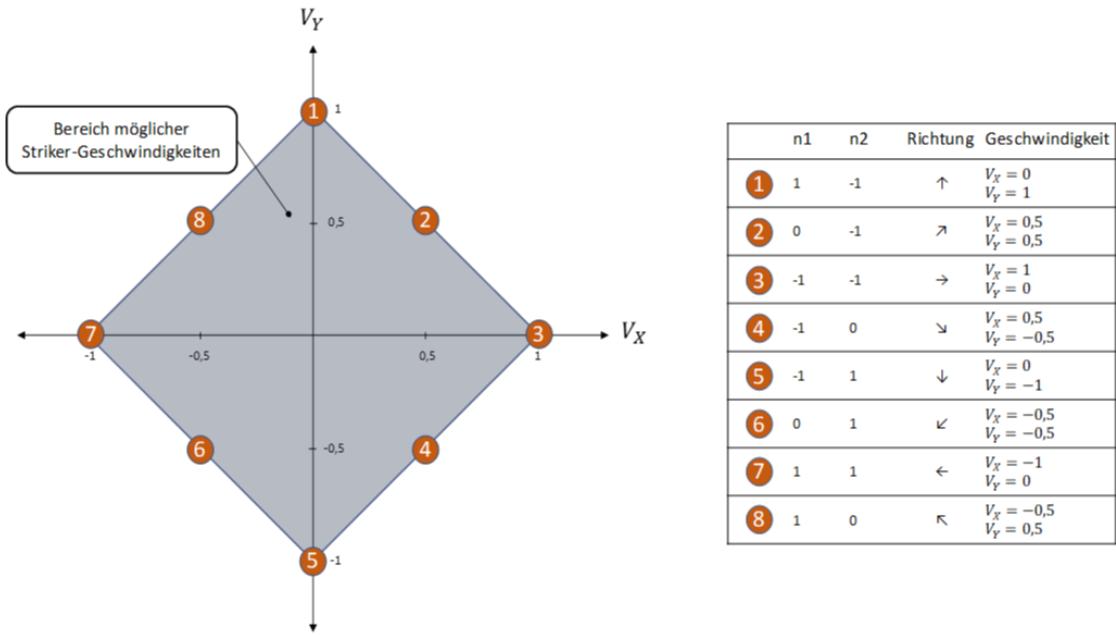


Abbildung 5.15: Drehzahlkonstellationen zur Veranschaulichung

Aus dem Zusammenhang zwischen translatorischer- und Winkelgeschwindigkeit, bekannt aus dem Tabellenbuch Metall [TM, S. 30] (Umfang U beträgt 94,248 mm)

$$v = 2 \cdot \pi \cdot r \cdot n = U \cdot n \quad (5.4)$$

ergibt sich für die Drehzahl n :

$$n = \frac{v}{U}. \quad (5.5)$$

Die Einzelgeschwindigkeiten v_1 und v_2 (siehe 5.14) lassen sich wie folgt mit Hilfe von v_x und v_y abbilden:

$$v_1 = \cos 45^\circ \cdot (-v_x) + \sin 45^\circ \cdot v_y \quad (5.6)$$

$$v_2 = \cos 45^\circ \cdot (-v_x) + \sin 45^\circ \cdot (-v_y). \quad (5.7)$$

Eingesetzt in (5.5) ergibt sich:

$$(n_1 \mid n_2) = (\frac{v_1}{U_{Scheibe}} \mid \frac{v_2}{U_{Scheibe}}) = \left(\frac{-v_x + v_y}{\sqrt{2} \cdot U_{Scheibe}} \mid \frac{-v_x - v_y}{\sqrt{2} \cdot U_{Scheibe}} \right), \quad (5.8)$$

welche die Geschwindigkeits-Transformation

$$(v_x \mid v_y) \rightarrow (n_1 \mid n_2) \quad (5.9)$$

darstellt. Durch Integration ergibt sich daraus nun die Orts-Transformation.

$$(x \mid y) \rightarrow (Umdr_1 \mid Umdr_2) = \left(\frac{-x+y}{\sqrt{2} \cdot U_{Scheibe}} \mid \frac{-x-y}{\sqrt{2} \cdot U_{Scheibe}} \right) \quad (5.10)$$

Damit ist es möglich, Positionen des X-Y-Koordinatensystems in Umdrehungs-Koordinaten umzuwandeln. Mit diesen können die Motoren angesteuert werden.

5.4.2 Motordynamik

Um die Dynamik des Roboters abschätzen zu können, werden zuerst konkrete Bewegungsfälle untersucht.

Im ersten Fall wird angenommen, der Striker müsste sich von der unteren linken Ecke (0|0) bis zur oberen rechten Ecke (361|215) des befahrbaren Bereichs bewegen. Diese Bewegung bildet den „Worst-Case-Fall“ ab, da es der längste zu fahrende Weg für einen der Motoren darstellt. Nach (5.10) ist der abzufahrende Weg in diesem Fall **4,32 Umdrehungen**.

Im zweiten Fall wird angenommen, der Striker müsste sich von der Tor-Position (180, 5|0) zur äußersten Seite einer gedachten Verteidigungslinie bewegen (361|100), welche sich etwa Handbreit vor der Torlinie befindet. Dann ergeben sich für diesen Fall nach (5.10) **2,1 Umdrehungen**, welche ein Motor zurücklegen muss.

Da der zweite Fall realistischer ist, wird mit diesem weitergearbeitet.

Um der Wunschanforderung aus 3.1 zu entsprechen, müssen die Zielumdrehungen in 125 ms zurückgelegt werden. Das ergibt eine nötige Durchschnittsgeschwindigkeit für diese Bewegung von etwa 17 Umdrehungen pro Sekunde.

Schrittmotoren müssen mit Hilfe einer (in der Regel linearen) Rampe beschleunigt und gebremst werden, welche in Abbildung 5.16 [Step-Speed] schematisch dargestellt ist.

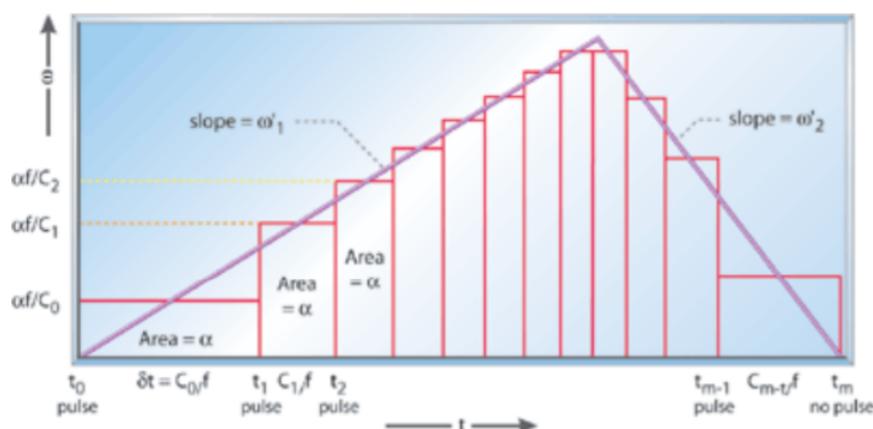


Abbildung 5.16: Beschleunigungs- und Bremsrampe für Schrittmotoren

Es fällt dabei auch auf, dass die Bremsrampe steiler ist als die Beschleunigungsrampe. Bezieht man nun die beiden Rampen in den Bewegungsvorgang mit ein, muss geschätzt eine Maximalgeschwindigkeit von 50 Umdrehungen pro Sekunde erreicht werden, damit die nötige Durchschnittsgeschwindigkeit eingehalten wird.

Grundlegend bewegen sich Schrittmotoren abhängig von der Frequenz des Steuerungssignals. Für jede Periode des Ein-Aus-Signals macht ein Motor genau einen Schritt. Die verwendeten Motoren benötigen 200 Schritte pro Umdrehung und werden im Vollschrittmodus betrieben. Dies kann am Treiber eingestellt werden.

Dadurch können logischerweise die höchsten Geschwindigkeiten erreicht werden. Je höher die Ansteuerungs-Frequenz des Kinematik-Programms, desto höher ist natürlich die Dynamik der Motoren, jedoch neigen Schrittmotoren dann auch dazu, Schritte zu überspringen und damit ungenau zu werden, sich mehr zu erhitzen und einen lauteren Betrieb zu haben. Hier muss zwangsläufig abgewogen werden, um jeweilige Nachteile zu minimieren.

Durch Versuche wurden mehrere Einstellungen für die maximale Beschleunigung und maximale Geschwindigkeit getestet. Die Ergebnisse sind in Tabelle 5.1 dargestellt. Die maximale Geschwindigkeit v_{max} und die maximale Beschleunigung a_{max} sind in Umdrehungen pro Sekunde angegeben. SGU steht für „Striche gegen Uhrzeigersinn“ und gibt die Position der Einstellschrauben für den Motorstrom an den Treibern an.

Die Einstellschrauben haben eine Pfeildarstellung, welche bei genauem Hinsehen erkennbar ist. Diese Einstellung muss passend zu den Dynamikeinstellungen im Kinematik-Programm sein, die jeweilige Stellung muss durch Versuche bestimmt werden. Zu hoher Motorstrom erhöht die Last auf den Motoren und damit das Moment, welches für höhere Dynamik höher sein muss, damit die Motoren stark genug sind. Andererseits erhöht ein höherer Motorstrom den Geräuschpegel, lässt die Motoren wärmer werden und die Genauigkeit sinken.

Die Spalte Zeit gibt an, in welcher Zeit die im zweiten Fall beschriebene Strecke absolviert wird.

Tabelle 5.1: Ergebnisse der Dynamiktests

v_{max}	a_{max}	SGU	Zeit	Kommentar
120	120	1,25	0,21 s	laut, Erwärmung schlecht, ungenau
100	100	1,5	0,25 s	laut, Erwärmung ok, ab und zu Ruckler
80	80	2,4	0,30 s	leiser, Erwärmung gut. Genauigkeit ok, selten Aussetzer
40	40	3,5	0,45 s	leise, Erwärmung top, Genauigkeit besser

Es ist erkennbar, dass die Zeit für die Bewegung deutlich über der Wunschanforderung von 125 ms liegt. Jedoch werden bei weiterem Erhöhen der Dynamik die auftretenden Nachteile zu groß, sodass mit diesen Motoren keine dynamischeren Bewegungen machbar sind.

Bei den Dynamik-Versuchen ist aufgefallen, dass die dritte Einstellung aus 5.1 mit v_{max} und a_{max} bei $80 \frac{U_{mdr.}}{s}$ und $80 \frac{U_{mdr.}}{s^2}$ die beste Wahl ist. Diese Einstellung ist im Kinematik-Code jedoch leicht global anzupassen.

5.4.3 Geradlinige Bewegung

Ein weiteres Thema bei der Bewegung des Strikers ist die Art, wie er sich bewegt. Hier ist eine möglichst geradlinige Bewegung wünschenswert, da diese am meisten der Intuition entspricht und sich somit am Leichtesten damit auseinandergesetzt werden kann. Wird jedoch, wie in der aktuellen Strategie-Kinematik Schnittstelle, nur eine Zielposition übermittelt und diese dann auch noch mit jeweils Höchstgeschwindigkeit angefahren, ergibt sich keine geradlinige Bewegung. Dies ist in Abbildung 5.17 veranschaulicht.

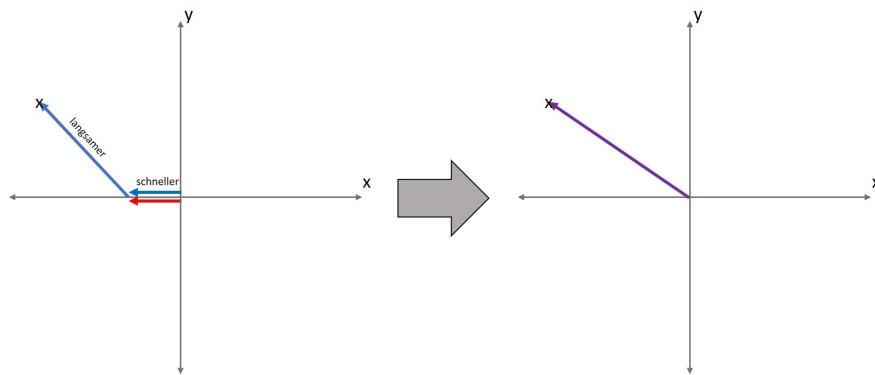


Abbildung 5.17: Geradlinigkeit der Bewegung

Da die Motoren in den meisten Fällen unterschiedlich große Wege zurückzulegen haben, kommt ein Motor zwangsläufig vor dem Anderen ans Ziel. Solange die Motoren noch gleichzeitig laufen, bewegt sich der Striker nur auf den Hauptachsen, also nur in x- oder y-Richtung. Ist ein Motor am Ziel, wird das letzte Stück des Striker-Weges in Diagonalrichtung zurückgelegt, wie ebenfalls in der Abbildung zu erkennen.

Um aus dieser Bewegungsart eine geradlinige Bewegung zu erzeugen, wird durch das Kinematik Programm die Geschwindigkeit und die Beschleunigung des Motors mit dem kürzeren Weg entsprechend verringert, damit beide Motoren gleichzeitig ihr Ziel erreichen.

Dadurch entsteht, abgesehen von kleineren Abweichungen während der unterschiedlich langen Beschleunigungsphasen, eine annähernd geradlinige Bewegung. Dieses Verfahren ist im Kinematik-Code implementiert.

5.4.4 Codeumsetzung

Der Code der Kinematik ist nach dem in Abbildung 5.18 dargestellten Konzept aufgebaut. Dieses beinhaltet der groben Ablauf des Programms, welches in der Loop-Schleife des Arduino Codes wiederholt durchgeführt wird.

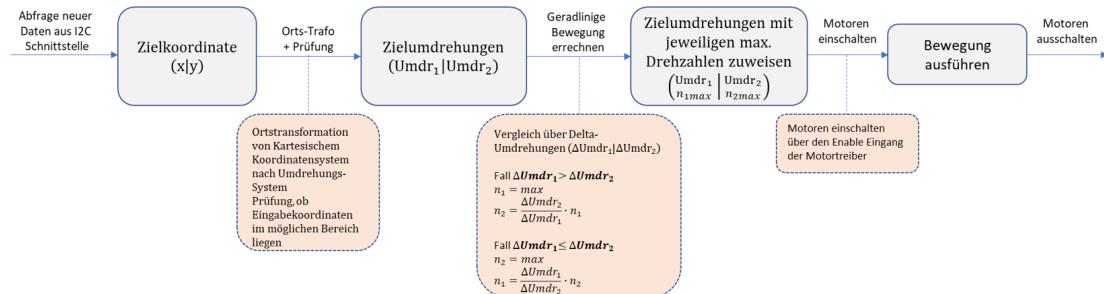


Abbildung 5.18: Konzept des Kinematik-Codes

Hierbei ist besonders hervorzuheben, dass das Ein- und Ausschalten der Treiber und der Motoren über den Enable-Eingang der Treiber wichtig ist. Dadurch werden die Motoren nur unter Last gesetzt, wenn diese auch wirklich verwendet werden. Hierdurch wird die thermische Belastung minimiert und ein Überhitzen der Bauteile ist weitestgehend ausgeschlossen.

Würden die Motoren dauerhaft aktiviert bleiben, würden sie ein Haltemoment aufbringen und sich somit weiter erhitzen.

Zusätzlich zur Grundaufgabe des Kinematikpaketes, die Ansteuerung der Motoren zu realisieren, ist eine Kalibrierung der Motoren als zusätzliche Funktion vorgesehen. Diese soll ebenfalls in der Kinematik-Programmierung implementiert werden.

Dabei soll eine eigentlich ungültige, ganz bestimmte Zielangabe an den Arduino DUE gesendet werden, welcher genau diese Eingabe erkennt.

Dann wird mit sehr langsamer Geschwindigkeit in Richtung des Koordinatenursprungs verfahren, bis die jeweiligen Endschalter, welche mit dem Arduino verkabelt sind, auslösen und ein Erreichen der Nullposition in der jeweiligen Achse übermitteln. Wenn beide Endschalter betätigt sind und der Striker sich somit in der Nullposition befindet, ist die Kalibrierung abgeschlossen und neue Werte können über die Schnittstelle empfangen werden.

Der gesamte Arduino Code ist umfangreich kommentiert, sodass sich möglichst leicht in diesen eingearbeitet und mit der Kinematik weitergearbeitet werden kann.

5.5 Spielmenü

Ein Bedienungsmenü auf dem zweiten, gruppeneigenen Pi vereint die drei Softwarepakete. Das Menü ist nach der Automatentheorie aufgebaut und beschreibt mehrere Zustände, zwischen denen mit den Buttonbefehlen „hoch“, „runter“ und „auswählen“ gewechselt werden kann. Insgesamt gibt es 13 unterschiedliche Zustände.

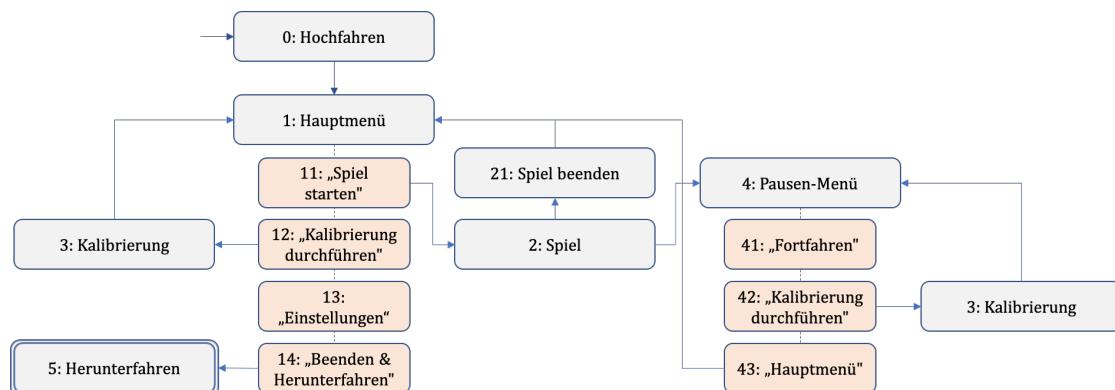


Abbildung 5.19: Struktur des Bedienungsmenüs

In Abbildung 5.19 wird die Struktur des Bedienungsmenüs dargestellt. Das Programm startet in Zustand 0 und wird in Zustand 5 beendet. Nach dem Hochfahren wird das Hauptmenü erreicht, welches in mehrere Zustände aufgeteilt wird: 11, 12, 13 und 14. Es kann zwischen den Optionen „Spiel starten“, „Kalibrierung durchführen“, „Einstellungen“, „Beenden und Herunterfahren“ gewählt werden.

Wird 11 gewählt, wird ein Spiel gestartet und der Zustand 2 erreicht. Während des Spiels werden die Programme Bildverarbeitung, Strategie und Kinematik angesprochen. Außerhalb dieses Zustandes werden diese Programme im Code nicht aktiviert. Aus dem Spielzustand kann das Spiel entweder pausiert oder beendet werden. Nach dem Beenden erscheint erneut das Hauptmenü. Der Zustandswechsel vom Spiel zum Spielende wurde noch nicht implementiert, da noch keine Bedingung dafür vorliegt. Die geplante Bedingung ist das Erreichen einer bestimmten Anzahl an Toren durch einen Spieler. Das Überprüfen dieser Bedingung erfordert jedoch eine Verbindung zum Punktezähler, die noch nicht besteht.

Wird das Spiel durch gleichzeitiges Drücken der "hoch und "runter" Tasten pausiert, gelangt der Nutzer in das Pausen-Menü, welches analog zum Hauptmenü aufgebaut ist. Dabei stehen die Optionen 41: „Fortfahren“, 42: „Kalibrieren“, 43: „Main Menu“ zur Verfügung.

Wird im Hauptmenü bzw. im Pausemenü „Kalibrieren“ gewählt, soll der Roboter zum Nullpunkt des Koordinatensystems fahren und somit seine Position abgleichen. Unter dem Menüpunkt Einstellungen gibt es bisher keine Einstellungsmöglichkeiten. Dies könnte jedoch entsprechend erweitert werden.

5.6 Hochzeit der Pakete

Nachdem die einzelnen Softwarepakete fertiggestellt wurden, begann das Zusammenfügen der Software und Hardware im Zuge der Hochzeit. Dafür wurde im ersten Schritt geplant, wie die Hardware anzutragen ist und diese schließlich auf einer Holzplatte strukturiert angeordnet. Das Ergebnis ist in Abbildung 5.4 dargestellt. Eine gute Zugänglichkeit der Anschlüsse sowie das Ziel, die Verkabelung übersichtlich und auf kurzen Wegen zu realisieren, wurde beachtet. Die Bedienungselemente wurden an der Oberseite der Holzplatte befestigt um eine möglichst benutzerfreundliche Bedienung des Roboters zu ermöglichen. Alle Bauteile wurden montiert und die nötigen Verbindungen zwischen den Bauteilen gesteckt, verlegt oder verlötet. Das Netzteil versorgt alle Bauteile mit Strom. Auch ein leuchtender Hauptschalter für den Stromanschluss des Netzteils wurde installiert. Die verbauten elektronischen Komponenten sind in Kapitel 5.1.2 genauer beschrieben. Das Holzbrett ist mit zwei M10 Schrauben an der Seite des Tisches befestigt, sodass die gesamte Elektronik des Roboters nur durch Lösen der beiden Schrauben vom Tisch demontiert werden kann.

Die unabhängig voneinander entwickelten Softwarepakete wurden auf die entsprechenden, elektronischen Bauteile geladen und dort einzeln getestet. Das Menü auf dem Raspberry Pi wurde mit dem Display und den Buttons verbunden und die Anzeige der Menüoptionen wurde getestet. Das gesamte Strategiepaket wurde als getrennte Klasse implementiert, sodass diese ohne Aufwand ausgetauscht werden kann. Die im Softwarepaket SStrategie entwickelte Simulation wurde ebenfalls hinzugefügt, um ein wertvolles Werkzeug zum Debuggen und Testen des Gesamtaufbaus inklusive der Puckbewegung zu bieten. Anhand der Simulation kann man die aktuell erkannte Puckposition, die Puckprognose der Strategie und die Reaktion des Strikers mitverfolgen.

Die Kommunikation und Weitergabe der Schnittstellendaten zwischen den Raspberry Pis sowie dem Arduino sollte wie in Kapitel 5.1.4 beschrieben realisiert werden. Das Einrichten der I²C-Verbindung konnte jedoch aufgrund zeitlicher Knappheit nur teilweise realisiert werden. Die Verkabelung der Bauteile zur Kommunikation ist fertiggestellt, aber die nötige Implementierung im Code ist nicht fertiggestellt.

6 Evaluation

Nach Abschluss der Gruppenarbeit wird das gesamte Projekt evaluiert, wobei einerseits das entstandene Produkt und andererseits die Projektarbeit bewertet wird. In diesem Kapitel wird nur auf den Roboter eingegangen, während im Kapitel 7 die Projektorganisation und die Zusammenarbeit in der Gruppe betrachtet und evaluiert wird.

Zuerst wird der erarbeitete Air-Hockey-Roboter hinsichtlich der Anforderungen untersucht. Die gestellten Anforderungen an den mechanischen Teil, den Software bzw. Hardware-Teil und das gesamte Ergebnis sind in der Anforderungsliste (3.1) aufgeführt. Diese Anforderungen werden im Folgenden auf Erfüllung bzw. auf den Grad der Erfüllung untersucht. Dabei wird beachtet, ob die Anforderung als Wunsch oder Forderung festgelegt worden ist.

Neben der Bewertung der Anforderungen werden auch Verbesserungsmöglichkeiten dargestellt. Diese wurden teilweise bereits zu Beginn als Zusatzoptionen definiert oder haben sich im Laufe des Projekts als Optimierungsmöglichkeiten herausgestellt. Aus Zeit- oder Kostengründen wurden die Ideen nicht in diesem Projekt umgesetzt. Weitere Verbesserungen ergeben sich aus den Anforderungen, die nicht erfüllt werden konnten. Bei einer Weiterarbeit mit dem Roboter könnten die Vorschläge jedoch relevant sein und sollten für ein verbessertes Ergebnis umgesetzt werden. Die Verbesserungsvorschläge werden in Mechanik und Software bzw. Hardware unterschieden.

6.1 Evaluation der Anforderungen

Im Folgenden ist die Erfüllung der einzelnen Anforderungen dargestellt. Dafür ist innerhalb der Anforderungsliste das „F“ bzw. das „W“ je nach Einteilung der Forderung mit rot oder grün markiert oder wurde offen gelassen. Der Erfüllungsgrad bzw. die Markierung aller Anforderungen wird in den darauffolgenden Kapiteln genauer begründet. Grün deutet auf umgesetzte Forderungen hin, während in Rot alle Forderungen markiert sind, die bisher nicht erfüllt werden konnten. Offen gelassen wurden die Anforderungen, die bisher noch nicht getestet werden konnten, jedoch als sehr realistisch eingeschätzt werden.

Tabelle 6.1: Anforderungsliste

F / W	Anforderung
Allgemein	
F	Ergebnis ist ein spielfähiger Roboter
F	Einhalten des Budgets von 250€
F	Montage des Gesamtsystems in vorliegendem Bauraum möglich
F	Gesamtsystem ist modular ausgeführt
Mechanik	
F	Ausreichend steife Komponenten halten den Belastungen stand
F	Minimale Sichtbehinderung für die Kamera
F	Aktionsfläche ausreichend zur Torverteidigung (200 mm x 150 mm)
F	Ausreichend starke Aktoren um auftretende Trägheiten zu überwinden
F	Realisierung ausreichender Striker-Geschwindigkeiten
F	Realisierung gewünschter Bewegungsbahnen des Strikers
W	Größere Aktionsfläche realisiert (400 mm x 300 mm)
Elektronik	
F	Hohe Treffgenauigkeiten wird ermöglicht
F	Geradlinige Bewegungen sind möglich
F	Zeit von Erhalt der Koordinate bis Erreichen unter 500 ms
F	Motoren drehen flüssig, Schritte werden nicht übersprungen
F	Betriebstemperatur der Hardware bleibt im unkritischen Bereich
F	Start-Kalibrierung der Strikerposition möglich
W	Übersichtliche und nachvollziehbare Verkabelung der Bauteile
W	Stromversorgung der Hardware ausschließlich über ein gemeinsames Netzteil
W	Keine Nutzung externer Geräte wie eines Laptops, damit keine aufwändige Einrichtung des Systems vorm Spielen notwendig ist
W	Menü am Tisch ermöglicht einfache Bedienung des Roboters
W	Zeit von Erhalt der Koordinate bis Erreichen unter 150 ms
Software	
F	Gut zugängliche Software mit standardisierten Schnittstellen
F	Genaue Puckdetektion durch Bildverarbeitung
F	Berechnung des Puck-Abprallens an den Bändern umgesetzt
F	Berechnung des notwendigen Striker-Zielpunktes um Puck zu treffen
F	Striker kann sich in alle Richtungen bewegen
W	Angriffsbewegung möglich, sodass Puck sich gezielt bewegt

6.1.1 Allgemein

Wie bereits in Kapitel 9.1 beschrieben, war die Modularität des Roboters ein zentrales Ziel im Projekt, welches wie angestrebt erreicht wurde. Auch konnte die Montage aller Bauteile im vorliegenden Bauraum gut realisiert werden. Weiterhin wurde das Budget von 250€ eingehalten, wie in Kapitel 7.1 erklärt wird.

Aufgrund des Mangels an Zeit, welcher durch die Unterschätzung des Arbeitsaufwandes und durch das Fehlen von geeigneter Kompetenz der Gruppenmitglieder im Bereich der Informatik, aber auch durch unvorhersehbare Herausforderungen entstanden ist, konnte der Roboter nicht spielfähig fertiggestellt werden. Die Gründe für diese Problematik werden im Rahmen der Analyse und Evaluation der Projektorganisation, Kapitel 7.4, ausführlich erläutert.

6.1.2 Mechanik

Die Konstruktion des Roboters ist so ausgelegt, dass möglichst die größtmögliche Aktionsfläche realisiert werden konnte. So waren 200mm x 150mm das Minimum, das festgelegt wurde, damit eine Verteidigung des Pucks überhaupt möglich ist. Als Wunsch ist die Bespielbarkeit von der doppelten Fläche 400mm x 300mm formuliert worden. Final konnte eine bespielbare Fläche von 400mm x 254mm realisiert werden. Dies konnte durch Abfahren der gesamten Fläche unter Führung des Strikers mit der Hand getestet werden. Die Wunschanforderung konnte also Auch diese Anforderungen konnte erfüllt werden.

Durch händisches Antreiben der Motorumlenkrollen konnte außerdem getestet werden, dass alle Bewegungsbahnen, die theoretisch möglich sind, auch praktisch abgefahren werden können. Durch unterschiedlich schnelle Drehung der Motorumlenkrollen scheint es, als würde das Erreichen der gewünschten Geschwindigkeit bzw. Beschleunigung konstruktiv ermöglicht.

Bei den Bewegungstests des Roboters hat sich jedoch gezeigt, dass die Linearführung auf den Gleitstangen durch zu hohe Toleranzen nicht die nötigen Bewegungsfreiheit ermöglicht. Wie z.B. bei 3D-Druckern, wurde auch beim Air-Hockey-Roboter ursprünglich nur ein Lager im Schlitten der Querkonstruktion verwendet [Ulti]. Dadurch musste jedoch eine leichten Bewegungsspiel der Querkonstruktion festgestellt werden, wodurch die Querkonstruktion nicht mehr senkrecht zu den seitlichen Gleitstangen liegen muss. Bei Betrieb kann der Roboter an dieser Stelle verkeilen und müsste von einem Menschen gestoppt und erneut in waagerechte Position gebracht werden. Daher wurde dieses Bauteil durch eine Schlitten ausgetauscht, der durch zwei Lager geführt wird. Damit wird zwar das Verkeilen verhindert, allerdings wurde aufgrund der Toleranzen beim 3D-Druck die Reibung in den Linealagern erhöht. Denn durch Spannungen, die entstehen, da die Lager nicht vollkommen in einer Flucht liegen, sind die Lager einer erhöhten Querkraft ausgesetzt.

Eine weitere Schwierigkeit wird durch die Festigkeit der 3D-gedruckten Bauteile

hervorgerufen. Bereits bei der Herstellung musste festgestellt werden, dass einige Bauteile nicht die gewünschte Festigkeit haben. Die Qualität der Bauteile wurde durch erneutes Drucken mit angepassten Einstellungen erhöht.

Die Anforderungen, eine möglichst geringe Sichtbehinderung zu erzeugen, konnte durch das mechanische Konzept erfüllt werden. Der Roboter kann sich bis an die eigene Torlinie zurückziehen und bedeckt dort kaum Fläche. In der Programmierung des Roboters wird zudem angestrebt, dass der Puck immer auf der halben Länge der Tischhälfte des Roboters abgewehrt wird. Dadurch sollte der Puck gar nicht unter die Querkonstruktion gelangen. Dies wurde im Zuge der Evaluation der Konstruktion getestet. Eine problematische Sichtbehinderung konnte also konstruktiv verhindert werden.

Durch eine durchdachte Auswahl geeigneter Motoren ist auch die Anforderung ausreichend starker Aktoren erfüllt.

6.1.3 Elektronik

Da leider der Roboter nicht final fertiggestellt werden konnte, konnte die reale Treffgenauigkeit des Roboters nicht getestet werden. Ebenso konnte die Reaktionszeit der Motoren von Erhalt der anzufahrenden Koordinate bis zum Erreichen dieser Zielposition nicht gemessen werden.

Die Motoren konnten bei Probedurchläufen der Kinematik auf ihre Fähigkeiten getestet werden. Dabei wurde die Motoren so eingestellt, dass diese maximale Beschleunigung bzw. Geschwindigkeit bei ebenso maximaler Genauigkeit erreichen. Neben der Abwägung dieser beiden variablen Faktoren wurde zusätzlich darauf geachtet, dass die Motoren flüssig laufen und kaum Schritte übersprungen werden. Wie sich dies im Spiel verhält, konnte nicht getestet werden.

Auch die Betriebstemperatur der Hardware konnte nicht geprüft werden. Hierbei lässt sich jedoch sagen, dass sich weder die Motoren noch die anderen Hardwarebauteile bei unzähligen Testdurchläufen der Softwarepakete stark erhitzt haben.

Damit die Strikerposition kalibriert werden kann, falls Schritte übersprungen wurden und die Position des Striker nicht mehr stimmt, sollte eine Möglichkeit der Kalibrierung beim Start eingebaut werden. Diese konnte jedoch durch Zeitknappheit nicht mehr realisiert werden. Welche Möglichkeiten dafür bestanden hätten, werden in Kapitel 6.2.2 erläutert.

Die Montage der Hardware sowie das Hardwarekonzept erfüllen jedoch alle übrigen Anforderungen. Die gesamte Hardware befindet sich auf einer Holzplatte übersichtlich angeordnet und auch die Kabel wurden strukturiert und leicht nachvollziehbar verlegt. Alle Bauteile wurden mit eindeutigen Beschriftungen versehen, damit es nicht zu Unklarheiten kommt. Aus der Anordnung der Hardware ist ersichtlich, dass alle Bauteile durch ein gemeinsames Netzteil versorgt werden.

Lediglich der Raspberry Pi der Vorgängergruppe, der für die Bildverarbeitung

genutzt wird, fällt dabei raus, da dieser nicht zu den eigenen Bauteilen des neuen Roboters gehört. Er befindet sich an einer anderen Stelle am Tisch.

Darüber hinaus gibt es jedoch keine weiteren Bauteile, die sich nicht am Tisch befinden. Dadurch wird sichergestellt, dass keine aufwendige Einrichtung des Systems vor dem Spiel notwendig ist.

Auch konnte ein Menü am Tisch installiert werden. Ein LCD Display wurde so an der Holzscheibe befestigt, dass es bei Nutzung des Tisches gut sichtbar ist. Auf diesem Display wird das Menü anzeigen. Zusätzlich befinden sich drei Buttons neben dem Display, mit denen das Menü bedient werden kann. Die Funktionsfähigkeit des Menüs wurde bereits erfolgreich getestet.

6.1.4 Software

Wie in Kapitel 9.1 beschrieben, ist die Elektronik übersichtlich auf der Holzplatte angeordnet. Dabei sind alle Anschlussmöglichkeiten an die Bauteile am Rand angeordnet, wodurch diese gut zugänglich sind und einfach auf die Bauteile und die darauf laufende Software zugegriffen werden kann.

In Testdurchläufen der Bildverarbeitung ist ersichtlich, dass die Position des Pucks in einem aufgenommenen Bild gut erkannt wird. Weiterhin wurde im Code eine Bandenkontaktberechnung eingefügt. Innerhalb der Strategie wird der notwendige Zielpunkt des Strikers zur Abwehr des Pucks berechnet. Diese strategischen Berechnungen konnten mithilfe der in Kapitel 5.3.2 beschriebenen Simulation getestet werden.

Das Kinematikpaket hat ermöglicht, dass sich die Motoren wie gewünscht bewegen und dadurch den Striker theoretisch in alle Richtungen bewegen können. Wie dies berechnet und realisiert wurde, ist in Kapitel 5.4 beschrieben. Eine Angriffsbewegung konnte aus Zeitknappheit nicht mehr programmiert werden.

Insgesamt konnten die Ergebnisse der Software jedoch nicht am realen Roboter getestet werden, da das gesamte System nicht spielfähig realisiert werden konnte. Die Finalisierung der Hochzeit und eine Debugging- bzw. Optimierungsphase steht noch aus. Die Anforderungen konnten also nur innerhalb des Codes auf Erfüllung getestet werden. Mögliche Verbesserung werden im Folgenden aufgeführt.

6.2 Verbesserungsmöglichkeiten

Verbesserungsmöglichkeiten wurden sowohl für die Konstruktion als auch die Hard- und Software gefunden werden.

Wie sich aus der Evaluation ergibt, konnten einige Anforderungen nicht erfüllt werden. Dies sind Forderungen oder als Wunsch formulierte Anforderungen, welche im begrenzten Rahmen des Projekts nicht realisiert werden konnten. Hier greifen einige der Verbesserungsvorschläge, die diese Forderungen erfüllen.

Weitere Verbesserungen sind an Stellen möglich, die nicht in den Anforderungen an das Projekt erwähnt wurde, jedoch für ein optimales Ergebnis förderlich wären.

6.2.1 Verbesserungsmöglichkeiten - Mechanik

Die momentane Konstruktion bietet Möglichkeiten zur Verbesserung:

Um die Gefahr des Verkeilens der Querkonstruktion weiter zu minimieren, sollte der Schlitten zwischen Seit- und Querstangen erneuert werden. An dieser Stelle sollten weiterhin zwei Linearlager hintereinander verbaut werden, damit die zur Verkeilung beitragenden Kräfte abgefangen werden können. Allerdings muss sichergestellt werden, dass die Bohrungen für die Linearlager präzise in einer Flucht liegen.

Des Weiteren wäre auch eine denkbare Möglichkeit nur ein Lager im Schlitten zu verwenden, auftretende Spannungen zu vermeiden. Allerdings müssen dann hochpräzise Führungsstangen und Linearlager verwendet werden, damit ein Verkeilen durch zu großes Bewegungsspiel verhindert wird.

Aus Kostengründen wurde bei der Konstruktion auf die Verwendung von Präzisionsstangen verzichtet. Dadurch kommt es zu Reibungskräften zwischen Linearlager und Stange. Diese tragen zu einem Geschwindigkeitsverlust bei. Um die Reaktionszeit des Roboters zu verbessern, sollten die montierten Stangen mit Präzisionsstangen aus Aluminium ausgetauscht werden. Die geringe Dichte des Aluminiums trägt zusätzlich zu einer Minimierung der Trägheitskräfte bei, wodurch die Leistung des Roboters ebenfalls verbessert werden könnte.

Einen ähnlichen Effekt hätte der Austausch der Linearlager sowie der Wälzläger innerhalb der Umlenkrollen. Auch hier wurden aus Kostengründen günstigere Modelle eingesetzt. Hier würden bessere Modelle sowohl die Präzision der Bewegungen verbessern als auch die Reibungskräfte verringern.

Eine weitere Verbesserungsmöglichkeit bildet der Austausch der Holzleisten am Tisch. Während diese aufgrund der Materialeigenschaften von Holz nicht allzu präzise und gut zu bearbeiten sind, wäre es sinnvoll die Holzleisten durch Leisten aus z.B. Aluminium auszutauschen.

Zwar bilden die Durchgangslöcher in den Holzleisten eine gute Möglichkeit, den Air-Hockey-Roboter einfach mithilfe von Durchsteckverbindungen zu de-/montieren, allerdings wäre die Verwendung von Klemmverbindungen zwischen dem Air-Hockey-Roboter und den Holz- bzw. Aluleisten der einfachen Entnahme des Roboters und damit der Modularität der Gesamtkonstruktion zuträglich.

6.2.2 Verbesserungsmöglichkeiten - Softwaretechnik

Neben mechanischen Verbesserungsmöglichkeiten gibt es Verbesserungsvorschläge für jedes der Softwarepaketes sowie die Hardware.

Bildverarbeitung

Die Bildverarbeitung könnte um die Detektion des Strikers erweitert werden, damit dieser ebenfalls verfolgt wird und die Bewegungen des Striker kontrolliert werden können. Dadurch kann abgesichert werden, dass der vom Striker angesteuerte Punkt mit gewünschter Genauigkeit erreicht wird und keine Probleme wie Schrittfehler des Motors die Bewegungen des Strikers verfälschen.

Die detektierten Koordinaten des Pucks oder möglicherweise auch des Strikers können zur genaueren Positionsbestimmung weiter gefiltert werden. Dadurch kann der Einfluss von Messungenauigkeiten z.B. durch die Kamera oder die Mittelpunktbestimmung des Pucks verringert werden. Die erkannte Puckbahn kann von ausschlagenden Messungen geglättet werden und eine genauere Puckbahn prognostiziert werden.

Um eine ausreichende Schnelligkeit im Spiel zu ermöglichen, wäre die Bildaufnahme mit 60 oder sogar 90 fps optimal. Dadurch würden mehr Positionsdaten in der Zeit von Abschluss des Gegners bis zur nötigen Verteidigung durch den Roboter zur Verfügung stehen. Die Genauigkeit der Vorhersage könnte steigen und vereinzelte, stärkere Messfehler geringere Auswirkungen haben.

Strategie

Die Strategie kann hinsichtlich des Verteidigungsprogramms verbessert werden, aber auch durch eine Angriffsstrategie komplett ersetzt werden. Durch die verbesserte Filterung nach der Bildverarbeitung würde der Schnittpunkt mit der Aktionslinie, der berechnet wird, genauer bestimmt werden können und mit jeder neuen Berechnung weniger vom tatsächlichen Kollisionspunkt abweichen.

Aus einer Strecke und einer Geschwindigkeit kann der Zeitpunkt des Schnittpunktes mit der Aktionsleiste berechnet werden. Ist die verbleibende Zeit bis zur Ankunft bekannt, könnte der Roboter sein Verhalten daran anpassen. Zum Beispiel müsste dann der Roboter nicht mit maximaler Geschwindigkeit direkt zum Zielpunkt fahren, sondern könnte je nach verbleibender Zeit entweder warten oder nur mit geringer Geschwindigkeit die Koordinaten anfahren.

Für eine Angriffsstrategie muss bedacht werden, dass nicht nur eine Kollision zwischen dem Puck und dem Striker initiiert wird, sondern versucht wird, den Puck gezielt zu treffen. Dafür muss der Roboter seine Anfahrkurve entsprechend der Puckbahn wählen, sodass der Puck möglichst so abprallt, dass er nach einer Kollision einen Geschwindigkeitvektor in Richtung des gegnerischen Tores hat. Somit wäre der Roboter in der Lage nicht nur durch Zufall ein Tor zu erzielen, sondern würde jeden seiner Schläge gezielt auf das Gegentor ausrichten. Durch die

Schrittmotoren ist es möglich, dies zu realisieren und den Roboter auf einer bestimmten Kurve mit variierender Geschwindigkeit zu einer Position fahren zu lassen.

Würde der Striker von der Kamera detektiert werden, ist die exakte Position des Strikers immer bekannt. Dies wäre für eine Angriffstrategie sehr hilfreich, da dadurch kaum Abweichungen, welche bei einem Schlag große Auswirkungen auf die Richtung hätten, von der realen Position entstünden.

Die Regelung der Geschwindigkeit des Roboters könnte auch von Vorteil sein, um so zu vermeiden, dass der Puck zu stark geschlagen wird und so ein unkontrolliertes Herausfliegen aus dem Spielfeld vermieden wird.

Kinematik

Eine wichtige Verbesserung ist der Ausbau der Schnittstelle zwischen der Strategie und der Kinematik. Bisher wird die anzufahrende x- und y-Zielposition weitergegeben, welche in Bewegungssignale für die Motoren umgewandelt werden, um die gewünschte Position zu erreichen. Während der Ausführung der Bewegung können die Motoren keine weiteren Informationen verarbeiten. Dadurch kann die anzufahrende Position erst aktualisiert werden, wenn die zuerst weitergegebene Position erreicht wurde. Um die Motoren besser steuern zu können, sollte das Intervall zwischen der Signalweitergabe an die Motoren verkleinert werden. Optimal wäre es, wenn der Motor in kleinen, schnellen Zügen zur gewünschten Position fährt und dadurch die gewünschte Position jederzeit angepasst werden kann.

Eine weitere Möglichkeit, diese Schnittstelle auszubauen, wäre die zusätzliche Weitergabe einer Zielgeschwindigkeit zur weitergegebenen Zielposition. Dies wäre nötig, um die Stärke der Schläge bei einer Angriffsstrategie variieren zu können. Mit dem Weitergabe der Geschwindigkeit wäre also auch ein langsamerer Schlag nötig. Bisher wird die Zielposition immer schnellstmöglich angefahren.

Menü

Der Menüpunkt „Einstellungen“ hat bisher keine Funktion. Mögliche Einstellungen wie z.B. die Art der Spielstrategie (ob Abwehr oder Angriff) oder die Geschwindigkeit der Motoren könnte in Zukunft hinzugefügt werden. Eine weitere Verbesserung ist die Umgestaltung der Datenkommunikation, sodass diese auf dem zweiten, gruppeneigenen Pi nicht mehr seriell läuft, sondern als paralleler Thread. Dies würde die Zykluszeit minimieren und eventuell eine schnellere Reaktionszeit des Roboters ermöglichen.

Elektronik

Da nicht vermieden werden kann, dass die Schrittmotoren Schritte überspringen, ist eine Kalibrierung notwendig. Diese wird durch zwei Endschalter realisiert und der entsprechende Code ist auf dem Arduino innerhalb des Kinematikpaketes vorgesehen. Damit wird jedoch lediglich eine einmalige Kalibrierung realisiert, die nur auf Aktivierung den Striker bewegt und dadurch kalibriert.

Optimal wäre folglich die Einrichtung einer permanenten Kalibrierung. Diese sollte

sich außerhalb des Kinematikpaket befinden und als eigenständiges Paket realisiert werden. Eine Möglichkeit, wie ein permanenter Abgleich der realen Strikerposition mit der intern berechneten Position eingerichtet werden könnte, wäre die zusätzliche Detektion des Strikers durch die Kamera. Eine weitere Verbesserungsmöglichkeit wäre ein noch dynamischerer Motor, beispielsweise ein Motor des Typs NEMA 23. Ein solcher Motor könnte höhere Ströme aushalten, ohne zu überhitzen. Dies würde ein höheres Moment, eine bessere Beschleunigung, höhere Maximalgeschwindigkeiten und weniger übersprungene Schritte bedeuten. Die Motortreiber sowie die Taktung des Arduino DUEs sind für derartige Geschwindigkeiten geeignet.

7 Projektorganisation

Eine gute Projektorganisation wurde als ausschlaggebende Voraussetzung für die Realisierung des Projekts gesehen. Da weder inhaltlich noch zeitlich detaillierte Festlegungen vor Beginn des Projekts bestanden, nahm die anfängliche Projektplanung eine wichtige Rolle ein.

Im Folgenden wird in inhaltliche (7.1) und zeitliche Planung (7.2) unterschieden und das Vorgehen genauer erläutert (7.3). Im Abschluss wird die Projektorganisation evaluiert (7.4) und auf Verbesserungsmöglichkeiten geprüft.

7.1 Inhaltliche Projektplanung

Im ersten Schritt wurde das übergreifende, inhaltliche Ziel, einen neuen Air-Hockey-Spieler zu konstruieren, in detaillierte Teilarbeitsschritte herunterbrochen. Dadurch konnte sich ein Überblick über alle nötigen Arbeiten verschafft werden. Zum Überblick, zur besseren Koordination und zur vereinfachten Aufteilung auf die Gruppenmitglieder wurden die Teilarbeitsschritte in übergreifende Arbeitspakete zusammengefasst.

Das Projekt wurde in „Vorarbeit und Planung“, die Hauptarbeitsstränge „Software“ (inklusive der Elektronik) und „Mechanik“ sowie die Finalisierung des Projektes aufgeteilt. Dazu kommen die Arbeitspakete „Exposé“, „Projektbericht“ und die Zwischen- sowie Abschlusspräsentation, die den formalen Teil des Projekts umfassen. Eine Übersicht darüber bietet die Abbildung 7.1.

Die Arbeitsstränge bestehen aus mehreren Arbeitspaketen (AP), die durch die Teilarbeitsschritte jeweils sehr stark detailliert beschrieben sind (vgl. Abbildung 7.2). Damit ist jederzeit klar, welche Inhalte und Ziele zu diesem Arbeitspaket gehören. Weiterhin konnte durch diese Einteilung auch die zeitliche Planung einfacher durchgeführt werden. Außerdem konnten einzelne Schritte sowie auch gesamte Arbeitspakete einzelnen Gruppenmitgliedern zugeordnet werden.

Für einen guten Überblick über den Projektfortschritt wurden sechs Meilensteine definiert, welche sich jeweils auf die schaffende Arbeit am Air-Hockey-Roboter beziehen.

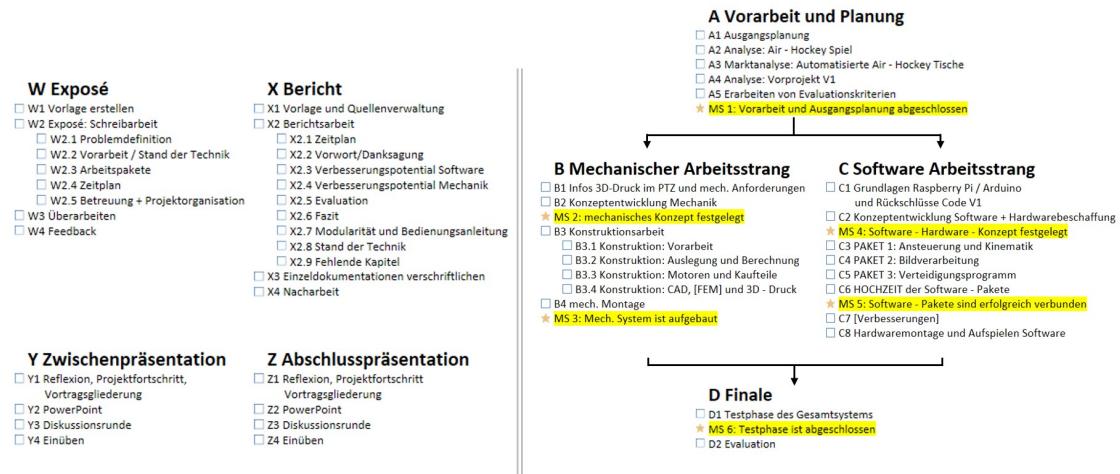


Abbildung 7.1: Hauptarbeitsstränge mit Meilensteinen (gelb)

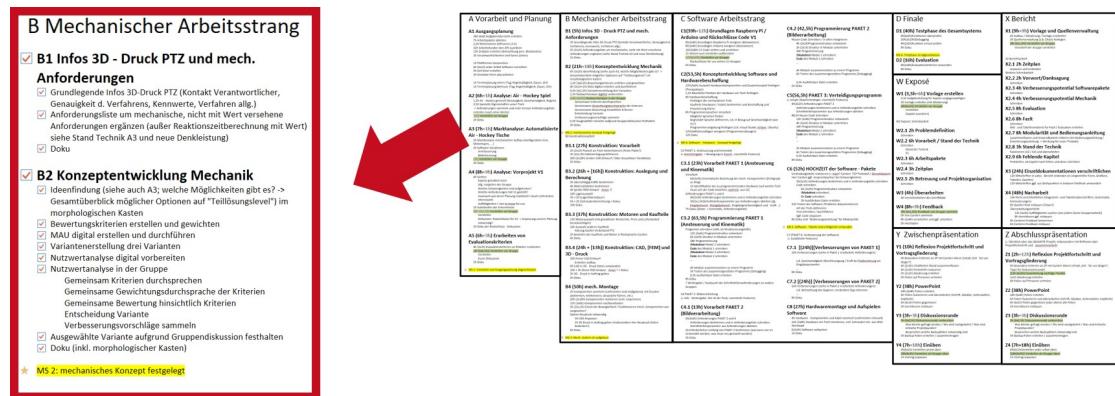


Abbildung 7.2: Detaillierungsgrad der Arbeitspakete

Diese detaillierte Planung der Arbeit wurde von der Gruppe gemeinsam durchgeführt, damit alle Gruppenmitglieder einen guten Eindruck vom Gesamtumfang des Projektes haben. Außerdem kann jedes Gruppenmitglied schnell nachvollziehen, woran die anderen Gruppenmitglieder gerade arbeiten und welche Schritte noch fehlen.

Als grobe Einteilung wurden zwei Gruppenmitglieder für den Arbeitsstrang „Mechanik“ eingeteilt und drei Gruppenmitglieder innerhalb des Arbeitsstrangs „Software“ den drei Softwarepaketen „Bildverarbeitung“, „Strategie“ sowie „Kinematik“ zugeordnet. Das sechste Gruppenmitglied agierte als Springer und war für die Zusammenführung der einzelnen Arbeitspakete zuständig.

Eine Anforderung an das Projekt war die Limitierung des Budgets auf 250 €. Zur Verwaltung dieser Ressource wurden alle Ausgaben in einem speziellen Kosten-tracking (Abbildung 7.3) gesammelt. Dadurch konnten alle Gruppenmitglieder benötigte Teile kaufen, bzw. Geld dafür reservieren und diese Beträge jederzeit aktualisieren.

KOSTENÜBERSICHT							
Stand	Bereich	Objekt	Anzahl	Stück	Versand	Gesamt	von Rechnung
Final	Mechanik	Motor	2	17,56 €	4,50 €	39,62	C ja
Final	Mechanik	Alustangen	4	2,29 €		9,16	J ja
Final	Mechanik	Wälzlager	12	0,62 €	2,47 €	9,91	C ja
Final	Mechanik	Linearlager	2	1,95 €		3,90	C ja
Final	Mechanik	Linearlager 2	1	1,95 €	1,99 €	3,94	C ja
Final	Mechanik	Holzleiste	1			0,00	J ja
Final	Mechanik	Kunststoffbauteile	1			0,00	
Final	Mechanik	Riemen	1	4,47 €	2,99 €	7,46	J ja
Final	Software	Quadratleiste	1	2,30 €		0,00	D ja
Final	Software	Kabelklemmen mit Nagel	1	3,89 €		3,89	D ja

Abbildung 7.3: Ausschnitt des Kostentrackings

7.2 Zeitliche Organisation

Aufgrund des hohen Detaillierungsgrades der Projektplanung konnte die voraussichtliche Arbeitszeit für jeden Teilarbeitsschritt abgeschätzt werden und dadurch möglichst genau die benötigte Arbeitszeit für die Arbeitspakete errechnet werden. Außerdem konnte eine möglichst ausgeglichenen zeitliche Belastung der Gruppenmitglieder geplant werden.

B Mechanischer Arbeitsstrang

- B1 (5h) Infos 3D - Druck PTZ und mech.
Anforderungen
 - 2h Grundlegende Infos 3D-Druck PTZ (Kontakt Verantwortlicher, Genauigkeit d. Verfahrens, Kennwerte, Verfahren allg.)
 - 2h (2x1h) Anforderungsliste um mechanische, nicht mit Wert versehene Anforderungen ergänzen (siehe Stand Technik A3 und neue Denkleistung)
 - 1h Doku

- B2 (39h) Konzeptentwicklung Mechanik
 - 6h (3x2h) Ideenfindung (siehe auch A3; welche Möglichkeiten gibt es? -> Gesamtübersicht möglicher Optionen auf "Teillösungsebene") im morphologischen Kasten
 - 1,5h (3x0,5h) Bewertungskriterien erstellen und gewichten
 - 4h (3x1+1h) MAU digital erstellen und durchführen
 - 4,5h (3x1,5h) Variantenerstellung drei Varianten
 - 1,5h Nutzwertanalyse digital vorbereiten
 - (18h (6x3h)) Nutzwertanalyse in der Gruppe
 - Gemeinsam Kriterien durchsprechen
 - Gemeinsame Gewichtungsdurchsprache der Kriterien
 - Gemeinsame Bewertung hinsichtlich Kriterien
 - Entscheidung Variante
 - Verbesserungsvorschläge sammeln
 - 0,5h Ausgewählte Variante aufgrund Gruppendiskussion festhalten
 - 3h Doku

★ MS 2: mechanisches Konzept festgelegt

Abbildung 7.4: Ausschnitt der AP's mit Zeiten

Um eine Übersicht über die zur Verfügung stehende Zeit zu erhalten, wurden Abwesenheiten bzw. Blockzeiträumen durch z.B. Klausuren von allen Gruppenmitgliedern in einen Kalender eingetragen.

Die Laufzeit des Projekts wurde vom 25.04.2019 bis zum 14.08.2019 angesetzt. Mithilfe der Übersicht der Arbeitszeiten wurde unter Zunahme des Kalenders das Ganttdiagramm entwickelt (siehe Abbildung 7.6). Dabei wurde das Ganttdiagramm von hinten nach vorne entwickelt und den Arbeitspaketen Pufferzeiten zugeordnet. Die Monate Juli, August und teilw. September konnten aufgrund von Klausurvorbereitungen und Urlaubsplanung nicht als Arbeitszeit angesehen werden. Es war jedoch eine Verlängerung des Projekts bis zum 23.09.2019 nötig. Die Evaluation

der Planung und die Gründe der Verschiebung werden in 7.4 dargestellt.

Nach der Entscheidung für die Verschiebung wurden die zeitliche Planung angepasst und der neue Termin gewählt.

Übersicht der Arbeitszeiten												
AP	Titel	Start	Ende	angesetzte Zeit	Hauptverantwortung	Charlotte	Dennis	Emmi	Henni	Julian	Simon	Arbeitsstunden
B	Mechanischer Arbeitsstrang			05:00	Julian							00:00
B1	Infos 3D-Druck PTZ und mech. Anforderungen			02:00	Julian							00:00
	Grundlegende Infos 3D-Druck PTZ			02:00	Julian	01:00						02:10
	Anforderungsliste um mechanische Anforderungen ergänzen			01:00	Julian							02:00
	Doku			39:00	Julian							02:45
B2	Konzeptentwicklung Mechanik			06:00	Julian	03:00						00:00
	Ideenfindung			01:30	Julian	00:30						06:00
	Bewertungskriterien erstellen und gewichten			04:00	Julian	01:30						01:00
	MAU digital erstellen und durchführen			04:30	Julian	02:00						03:00
	Variantenerstellung drei Varianten			01:30	Julian							02:30
	Nutzwertanalyse digital vorbereiten			03:00	Julian	03:00	03:00	03:00	03:00	03:00		01:30
	Nutzwertanalyse in der Gruppe			00:30	Julian	00:05	00:05	00:05	00:05	00:05		18:00
	Ausgewählte Variante in Gruppendiskussion festhalten			03:00	Julian							00:30
	Doku			03:00	Julian							04:00
B3.1	Konstruktion: Vorarbeit			23:00	Charlotte							00:00
	Flansch an Tisch konkretisieren			02:00	Charlotte	01:00						01:00
	Kabelverlegung definieren			03:00	Charlotte							00:00
	Grober CAD Entwurf / Oder brauchbare Handskizze			20:00	Charlotte	08:40						09:50
	2h Doku			02:00	Charlotte							00:00
B3.2	Konstruktion: Auslegung und Berechnung			26:00	Julian							00:00
	2h überschlägig Kräfte bestimmen			02:00	Julian	04:45						11:35
	4h Materialstärken bestimmen / Dimensionierung			04:00	Julian	10:35						11:35
	4h [grobe FEM Analyse - Ansys ?]			04:00	Julian							00:00
	10h Lagerauswahl			10:00	Julian	01:55						01:55
	3h + 2h [Lagerlebensdauer]			05:00	Julian							00:00
	5h + 2h [Schraubenberechnung + Doku]			07:00	Julian							00:00
	10h Doku			10:00	Julian							00:00

Abbildung 7.5: Ausschnitt des Zeittrackings

Zur Verwaltung der Zeitressource wurde eine Datei zum Zeittracking erstellt (Abbildung 7.5), in die jeder seine gearbeiteten Stunden je nach Teilarbeitsschritt einträgt. Dies wurde vor allem zum Überblicken der benötigten Zeit geführt, um den Zeitaufwand rückblickend realistisch zu bewerten. Außerdem konnte damit mögliche Mehrarbeit einzelner Gruppenmitglieder erkannt und daraufhin ausgeglichen werden.

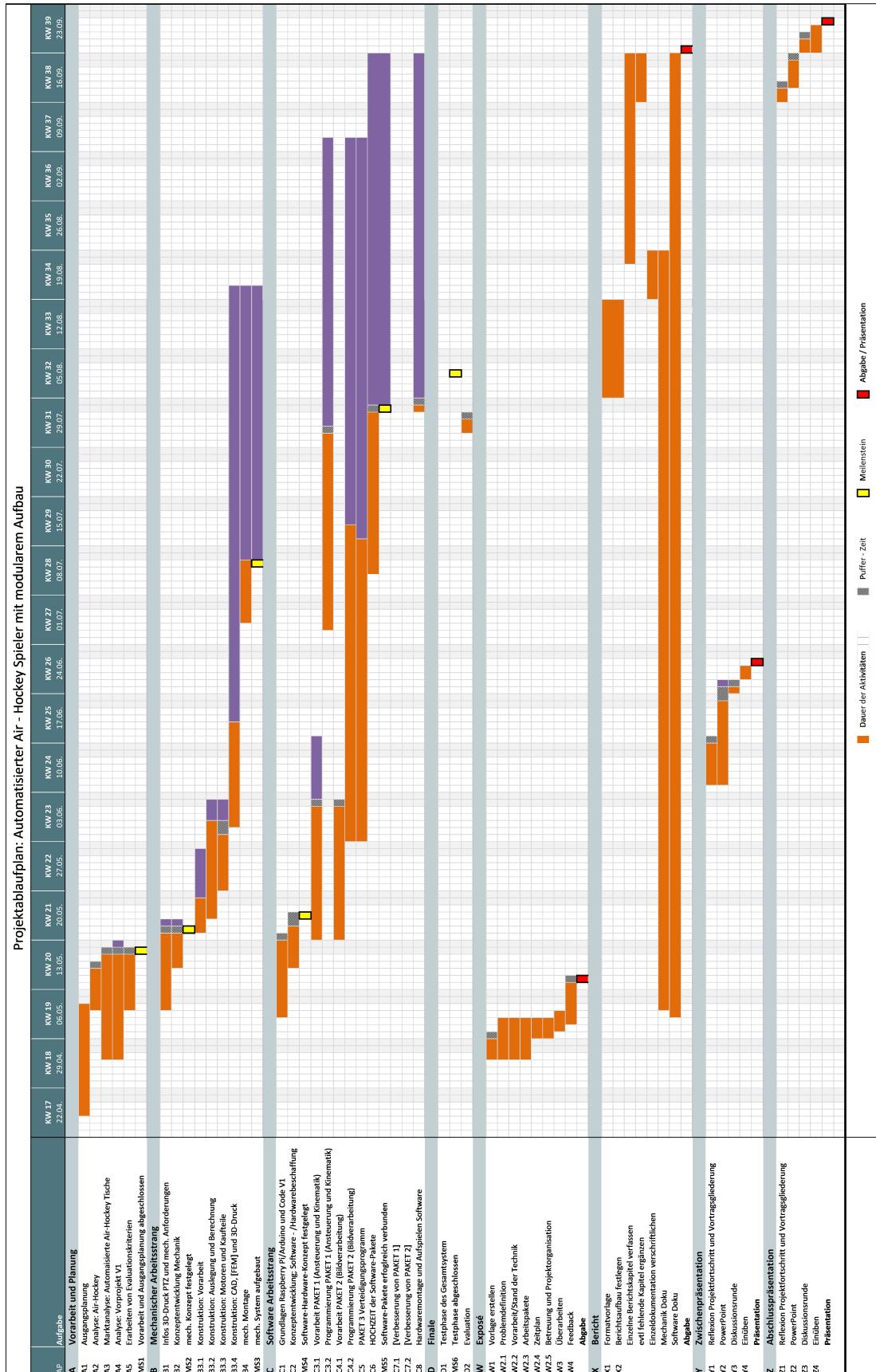


Abbildung 7.6: Ganttdiagramm

7.3 Realisierung der Zusammenarbeit

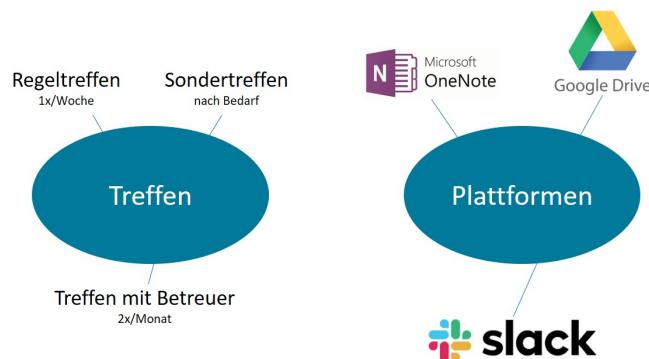


Abbildung 7.7: Überblick über die Realisierung der Zusammenarbeit

Regelmäßige Treffen und vielseitiger Austausch wurden als wichtiger Bestandteil der Gruppenarbeit angesehen. Dazu wurde ein wöchentliches Treffen terminiert. Bei diesem präsentierten zuerst alle Gruppenmitglieder ihren derzeitigen Stand. Danach wurden aufgetretene Fragen und Probleme diskutiert oder Entscheidungen getroffen. Bei Bedarf haben sich einzelne Gruppenmitglieder zu Sondertreffen getroffen. Zusätzlich wurde ungefähr zweimal monatlich ein Treffen mit dem Betreuer abgehalten, um das aktuelle Vorgehen zu präsentieren und aufgekommene Fragen zu klären.

Außerdem wurde die Plattform Slack als Kommunikationsplattform eingerichtet. Alle Gruppenmitglieder haben parallel in OneNote gearbeitet, wodurch jederzeit von überall auf die eigenen Notizen bzw. die Notizen anderer Gruppenmitglieder zugegriffen werden konnte. Weitere Dateien wurden in Google Drive für alle zugänglich gesammelt.

7.4 Evaluation der Projektorganisation

Neben der Evaluation des Air-Hockey-Roboters und der Anforderungen soll außerdem die Projektarbeit kritisch betrachtet werden. Dabei wird darauf eingangen, wie realistisch der Zeitplan war, wie gut die Kommunikation funktionierte und inwiefern das Projektziel erreicht werden konnte. Es wird bewertet, ob die Aufgaben intern gut formuliert und strukturiert und die Arbeit sinnvoll aufgeteilt war.

Inhaltliche Arbeit

Dank der detaillierten Aufschlüsselung der Arbeitspakete waren die Aufgaben jederzeit verständlich. Zusätzlich hat dies auch eine hohe Flexibilität der Gruppenmitglieder ermöglicht, da einzelne Aufgaben ohne aufwendiges Einlesen begonnen werden konnten, wenn ein Mitglied bei bestimmten Arbeitsschritten Hilfe benötigt hat. Die Verantwortlichkeiten für die Arbeitspakete waren insgesamt gut verteilt, sodass alle Gruppenmitglieder ähnlich ausgelastet waren.

Der Planung- und Koordinationsaufwand während der Projektlaufzeit war durch die ausführliche vorhergehende Projektplanung sehr gering und die Effizienz dadurch hoch.

Im Laufe der Projektdurchführung zeigte sich, dass meist die einzelnen Arbeitspakte genau so durchgeführt werden konnten, wie sie geplant wurden. Lediglich im „Softwaretechnischen Arbeitsstrang“ kamen unerwartete Arbeitsschritte hinzu. Wie der Name des Hauptarbeitstranges vermuten lässt, sollte sich hier hauptsächlich mit Software/Programmierung beschäftigt werden. Die damit verbundene Elektronik zur Ansteuerung der Motoren wurde, aufgrund der mangelnden Erfahrung der Gruppenmitglieder in diesem Bereich, mit zu geringem Arbeitsaufwand eingeschätzt und teilweise nicht beachtet.

Das Ressourcenverwaltungssystem des Kostentrackings zeigte sich als sehr hilfreich. Mit Hilfe dessen konnten die notwendigen Ausgaben schon früh geplant werden und das Budget wurde nicht überschritten.

Zeitplanung

Die eingeplante Arbeitszeit für die einzelnen Arbeitspakte war, wie aus dem Zeittracking erkennbar, für die meisten Arbeitspakte gut angesetzt. Jedoch wurde die reale, verfügbare Zeit der einzelnen Gruppenmitglieder überschätzt. Umfangreiche Arbeitsschritte wurden im Ganttdiagramm für einen zu kurzen Zeitraum eingeplant. Dies war größtenteils dadurch begründet, dass das gesetzte Projektende am 14.08.2019 eingehalten werden sollte. Da jedoch alle Gruppenmitglieder nach einigen Wochen Arbeitszeit mit den jeweiligen Arbeitspaketen in Zeitverzug waren, war eine Verschiebung des Endtermins nötig.

Bei einem Arbeitspaket im „Mechanischen Arbeitsstrang“, welches hauptsächlich erst nach der Zwischenpräsentation bearbeitet wurde, stellte sich heraus, dass hier die geplante Zeit zu optimistisch angesetzt wurde. Die Anfertigung der Bauteile mittels 3D-Drucker entwickelte sich als problematisch. Hier konnte zu Beginn kein Kontakt mit einer, für den Drucker verantwortlichen, Person hergestellt werden. Die Verantwortlichen waren ungünstigerweise unpässlich oder selbst nach einigen Versuchen nicht kontaktierbar. Auch die häufige Belegung des Druckers durch weitere Projekte oder den hauseigenen Aufträgen konnte nicht vorhergesehen werden. Die äußerst optimistisch angesetzten sieben Tage zur Beschaffung der Bauteile zog sich auf zwei Monate hinaus.

Aufgrund weiterer Verzögerungen durch ungünstige Umstände stellte sich heraus, dass einige Arbeitsschritte nicht zum gewünschten Grad fertiggestellt werden konnten. Der Grund dafür ist in den oben beschriebenen Anpassungen der anfänglichen Planung zu finden. Die Ergänzung der inhaltlichen Planung um einige unvorhergesehene Arbeitsschritte und der zeitliche Aufwand für die Beschaffung der Bauteile mittels 3D-Drucker führten zu zusätzlichen Arbeitsstunden.

Grundsätzlich wurde die angesetzte Zeit als reine Arbeitszeit definiert und jedem Gruppenmitglied aufgrund der hohen Arbeitsbereitschaft zu einem erfolgreichen Projektabschluss über 180 Arbeitsstunden zugeordnet. Die real benötigte Zeit überschritt die geplante reine Arbeitszeit. Dies lässt sich ebenfalls in dem Arbeitsaufwand von Vorlesungen in der Universität erkennen. Die reine Vorlesungszeit beträgt 90 Minuten, für eine Vorlesung werden jedoch 2 Stunden eingeplant. Durch die Vernachlässigung dieses realen Mehraufwandes, wurde die Arbeit, die jedes Gruppenmitglied in einer gewissen Zeitspanne schaffen kann, überschätzt.

Abgesehen von den genannten Schwierigkeiten kann die zeitliche Planung dennoch als Erfolg gewertet werden. Die zeitliche Planung geschah sehr fundiert, größtenteils realistisch und die Anzahl der weiteren noch notwendigen Schritte zum einwandfreien Funktionieren des Roboters sind minimal. Dennoch kann die Projektgruppe aus der Zusammenarbeit schlussfolgern, dass trotz großzügiger, realistischer Zeitplanung der reale Zeitaufwand aufgrund von unvorhersehbaren Umständen den geplanten Zeitaufwand um einen gewissen Faktor übertrifft.

Kommunikation

Die Kommunikation innerhalb der Gruppe war über die gesamte Projektlaufzeit sehr gut. Dank der regelmäßigen Treffen war jeder über die Arbeit der anderen informiert. Probleme wurden offen besprochen und bei Fragen konnte jederzeit geholfen werden. Auch war es jedem bekannt, wenn Gruppenmitglieder phasenweise in Lernphasen, im Urlaub oder bei anderen Verpflichtungen waren und nicht weiterarbeiten konnten.

Wichtig war auch, dass für das Projekt relevante Entscheidungen mit allen Gruppenmitgliedern getroffen wurden. So wurde beispielsweise die Wahl des konstruktiven Konzepts mit der gesamten Gruppe getroffen, damit alle Vor- und Nachteile gefunden und miteinander abgewogen werden konnten.

8 Fazit und Ausblick

Der Bau eines automatisierten Air-Hockey-Spielers im Rahmen des Automatisierungstechnischen Projektes bot der gesamten Projektgruppe einen hervorragenden Einblick in die Arbeitsweisen und den Ablauf innerhalb einer Projektarbeit.

Bereits bei der Ausformulierung der Aufgabenstellung gemeinsam mit dem Projektbetreuer und der anfänglichen Planung der zu bearbeitenden Pakete, bis hin zur Erstellung des Roboters und der Präsentation der Arbeit in Form des Berichtes und in Form von Präsentationen, war immer die produktive und effiziente Arbeit als Projektgruppe im Vordergrund.

So wurden bereits in den ersten Projektmeetings Plattformen und Strukturen zur erfolgreichen Projektplanung festgelegt und eingeführt. Anschließend wurde, soweit dies schon möglich war, jeder nötige Arbeitsschritt zur Erstellung des Air-Hockey-Roboters geplant, mit Zeiten versehen und in Arbeitspakete eingeteilt.

Während der folgenden Arbeitsphase in den getrennten Untergruppen (Softwarepakete und Mechanik) lief die Kommunikation und die Überwachung des Fortschritts über die eingerichtete Infrastruktur zur Projektplanung. Einzelne intensive Arbeitsphasen wie z.B. vor den Präsentationen oder vor dem Erreichen einzelner Meilensteine wurden dagegen in gemeinsamer Zusammenarbeit durchgestanden.

Damit war es insgesamt möglich, eine hervorragende Basis für einen automatisierten, modularen und spielfähigen Air-Hockey-Spieler zu schaffen. Zwar ist der Roboter ohne weitere aufgewendete Arbeit aufgrund der fehlenden Kommunikation zwischen den Softwarepaketen noch nicht einsatzbereit, allerdings sind die verschiedenen Komponenten im Einzelnen funktionsfähig und durch ihre Modularität leicht erweiterbar. So ist die mechanische Konstruktion des Air-Hockey-Roboters abgeschlossen, die einzelnen Softwarepakete funktionieren getrennt voneinander und die Bedienung der Software ist über ein Display und Buttons sichergestellt. Dabei wurde insbesondere darauf geachtet, die Austauschbarkeit sowohl des Roboters als auch von den einzelnen Softwarepaketen zu gewährleisten. Beispielsweise wurde die Verbindung des Roboters zum Tisch möglichst einfach gehalten und die Softwarepakete, aber auch die dazugehörige Elektronik im Schaltkasten, austauschbar gestaltet wurden.

In der Zukunft sollte der Fokus zunächst auf die Fertigstellung der Funktionsfähigkeit des Roboters gelegt werden. Dafür muss lediglich die Kommunikation der Softwarepakete untereinander mittels I²C fertiggestellt und eine Testphase durchgeführt werden. Anschließend könnten die mechanischen, aber auch die softwaretechnischen Verbesserungsmöglichkeiten (vgl. 6.2.1 und 6.2.2) umgesetzt werden, um einen optimierten, spielfähigen Air-Hockey-Roboter zu erschaffen.

Damit eignet sich der Air-Hockey-Spieler hervorragend für ein Nachfolgeprojekt. Durch die gute Dokumentation der Arbeit, die leicht nachvollziehbare Elektronik und den modularen Aufbau der Software und der mechanischen Konstruktion wäre es somit möglich, innerhalb eines Nachfolgeprojektes den Air-Hockey-Spieler so zu optimieren, dass er vielleicht sogar menschlichen Spielern überlegen ist.

9 Anhang

9.1 Modularitätsanleitung

Im Fokus der gesamten Projektarbeit stand die Modularität. Sowohl bei der Konstruktion, als auch in der Konzeptionierung und Montage der Hardware und bei den Softwareschnittstellen wurde auf die Modularität großen Wert gelegt.

Demontage des Gesamtsystems

Zuerst ist zu nennen, dass der Roboter als ganzes System leicht montierbar und demontierbar ist, was eine anderweitige Nutzung der Tischhälfte ermöglicht. Die Konstruktion ist auf zwei Holzleisten befestigt und zusätzlich an den Motorhalterungen mit einer entnehmbaren Tischplatte verschraubt (vgl. Abbildung 4.4). Indem die Schrauben zur Befestigung der Holzleiste am Tisch gelöst werden, kann die gesamte Konstruktion vom Tisch gehoben werden. Damit ist die gesamte Tischhälfte wieder im Ausgangszustand, es bleiben lediglich wenige, kaum sichtbare Löcher zurück und die Tischhälfte ist zur anderweitigen Nutzung bereit.

Die gesamte Elektronik befindet sich auf einer Holzplatte, die ebenfalls leicht gelöst werden kann. Dafür müssen lediglich zwei Schrauben zum Tisch gelöst werden und die Kabel zu dem vom Vorgängerprojekt am Tisch verbauten Raspberry Pi gelöst werden. Somit können Hardware und Konstruktion innerhalb kürzester Zeit abgenommen werden.

Anpassbarkeit der Elektronik

Durch die strukturierte Anordnung der Hardware auf der Holzplatte hinter einer Plexiglasscheibe sind die Wege der Informations- bzw. Stromflüsse leicht erkennbar (siehe Kapitel 5.1.2). Ein unbeabsichtigte Beschädigung wird jedoch durch die Plexiglasscheibe verhindert. Zusätzlich sind wichtige Bauteile zum Verständnis eindeutig beschriftet. Dadurch wird das mögliche Entfernen, Ersetzen oder Hinzufügen einzelner Bauteile erleichtert.

Auch sind die Bauteile, insbesondere der Raspberry Pi und der Arduino, auf der Holzplatte so angeordnet, dass leicht auf diese zugegriffen werden kann. Soll beispielsweise der Raspberry Pi per HDMI-Kabel mit einem Desktop verbunden werden, ist dies durch die leichte Zugänglichkeit der Stecker möglich, ohne die Glasscheibe zu lösen.

Zugriff auf die Raspberry Pis

Neben einer möglichen Verbindung per HDMI kann auch über die Remote-Desktop-Verbindung auf beide Raspberry Pis zugegriffen werden. Der Zugriff wird erleichtert, wenn ein mobiler Hotspot erzeugt wird, mit dem sich die Raspberrys automatisch verbinden können. Als Hotspot wird dasselbe Netzwerk wie auch im Vorgängerprojekt verwendet.

Hotspot Netzwerkname: test
Hotspot Netzwerkkennwort: 12345678

Login-Daten für den Raspberry 1 (Bildverarbeitung):

Benutzername: pi
Passwort: raspberry

Login-Daten für den Raspberry 2 (Menü, Strategie und Datenkommunikation):

Benutzername: pi
Passwort: airhockey

Zugriff auf den Arduino DUE

Um den Arduino DUE zu programmieren, verbinde man diesen via USB-Kabel mit einem Computer, auf welchem die Arduino-IDE installiert ist. Es sei zu berücksichtigen, dass vorher noch das „ATMEL SAMD Core“ Paket in die Arduino-IDE eingebunden werden muss. Hierzu gibt es viele Anleitungen auf der offiziellen Arduino Webseite.

Anpassbarkeit der Software

Durch die Zugänglichkeit der Elektronik ist auch die Software gut anpassbar und neue Softwarepakete können schnell auf das jeweilige Bauteil gespielt werden.

Die Bildverarbeitungssoftware befindet sich auf dem bereits bestehenden Pi unter der anderen Tischhälfte und hat als Schnittstelle die Position und die Geschwindigkeit des Pucks. Die Kommunikation der Bauteile funktioniert über I²C. Das Strategiekopf läuft auf dem zweiten Raspberry Pi, der auf der Holzplatte sitzt, und gibt die Zielposition des Roboters zur Verteidigung an. Diese Position wird von dem Kinematikpaket, welches sich auf dem Arduino befindet, in Bewegungssignale für die Motoren umgewandelt. Diese Signale gehen über die Motortreiber an die Motoren.

Das Hauptprogramm auf dem zweiten Pi verbindet die Strategie, die Simulation und die Datenkommunikation in dem übergeordneten Menüprogramm. Diese unterschiedlichen Funktionen wurden in den folgenden Klassen untergebracht, um strukturiertes und übersichtliches Programmieren zu ermöglichen.

Die Klassen "i2cControl.cpp" und "lcdDriver.cpp" wurden aus externen Quellen bezogen und dienen der Datenkommunikation über I²C und der Herausgabe von Text auf dem LCD Display. Die Klasse "grafik.cpp" ermöglicht eine Erstellung

eines Fensters auf dem Desktop des Pis, worauf die Simulation mitverfolgt werden kann. Dabei besitzt diese Klasse die Unterklassen "striker.cpp" und "puck.cpp", die zum Zeichnen der Figuren in dem Fenster benötigt werden. "puck.cpp" beinhaltet dabei die physikalischen Regeln des Abprallens von der Bande im Rahmen der Simulation. Die letzte Klasse in dem Programm ist "strategie.cpp", welche, wie der Name sagt, die notwendige Reaktion des Strikers bestimmt. Die Eingangsvariablen hierbei sind die x- und y-Werte der Puckposition sowie der Puckgeschwindigkeit als float-Variablen, die sowohl von der Simulation als auch von der Bildverarbeitung kommen können.

Die Berechnung der Strategie befindet sich aktuell in der Methode "reagiere()". Ausgangsvariablen sind dabei der x- und der y-Sollwert des Strikers, ebenfalls als float-Werte. Solange die Eingangs- und Ausgangsvariablen gleich bleiben, kann diese gesamte Klasse ausgetauscht werden. Dies ist der Hauptansatzpunkt für eine weitere Arbeit an dem Roboter.

Mit diesem Wissen können Teile der Softwarepakete passend zu den Schnittstellen verändert oder gesamte Softwarepakete und Bauteile ausgetauscht werden. Der gesamte Software Code wurde in einem GitHub Repository gespeichert und kann unter folgender Adresse abgerufen werden.

GitHub Repository:

<https://github.com/AirHockeyV2/AirHockeyV2.git>

9.2 Kostenübersicht

Tabelle 9.1: Kostenübersicht

Bezeichnung	Betrag
Motor	39,62 €
Wälzlager	9,91 €
Linearlager	7,84 €
Metallstangen	0,00 €
Holzleiste	0,00 €
Riemen	7,46 €
Kunststoffbauteile	0,00 €
Quadratleiste	0,00 €
Kabelklemmen mit Nagel	3,89 €
Filzplatte	4,15 €
Plexiglas	3,57 €
Mitteldichte Faserplatte	1,34 €
Micro SD Karte	6,99 €
Raspberry Pi	34,99 €
Display	8,94 €
Kabel	0,00 €
Netzteil	36,45 €
Step Down Converter	5,90 €
TB6600 Motortreiber	31,98 €
Stoppschalter 6er Pack	7,99 €
Arduino Due	24,00 €
Button für Display	5,49 €
2. Step Down Converter	5,90 €
Gesamtkosten	246,41 €

Anmerkung: Bauteile, welche mit dem Betrag 0 € versehen sind, wurden von der Universität bzw. WSH entgeldlos bereitgestellt.

9.3 Datenblätter und Produktdaten

- Motordatenblatt
- Produktdaten: Kugellager
- Produktdaten: Linearlager
- Produktdaten: Riemen

1.8° 42mm High Torque Hybrid Stepper Motor

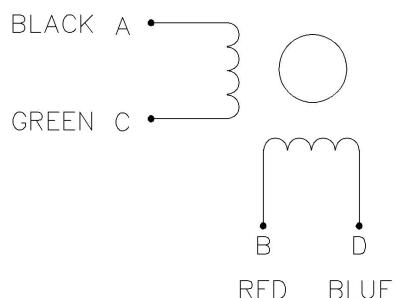
Item	Specifications
Step Angle	1.8°
Step Angle Accuracy	±5%(full step,no load)
Resistance Accuracy	±10%
Inductance Accuracy	±20%
Temperature Rise	80°C max
Ambient Temperature	-20°C ~ +50°C
Insulation Resistance	100 MΩ Min., 500VDC
Dielectric Strength	500VAC for 1 minute
Shaft Radial Play	0.02Max. (450g-load)
Shaft Axial Play	0.08Max. (450g-load)
Max. radial force	28N (20mm from the flange)
Max. axial force	10N



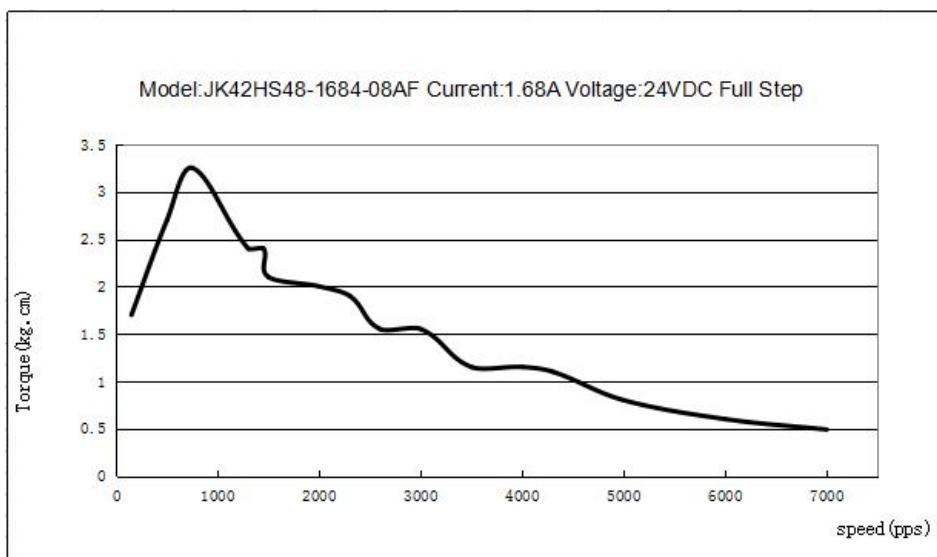
42mm Hybrid Stepper Motor Specifications:

Model No.	Rated Voltage	Current /Phase	Resistance /Phase	Inductance /Phase	Holding Torque	# of Leads	Detent Torque	Rotor Inertia	Motor Length	Weight
	V	A	Ω	mH	kg.cm	No.	g.cm	g.cm ²	(L)mm	kg
JK42HS48-1684-08AF	2.8	1.68	1.65	2.8	5.0	4	260	68	48	0.38

Wiring Diagram:



Pull out Torque Curve:





Charly Hase <charlyyyyhase@gmail.com>

**AW: Frage zu: Miniatur Kugellager 686 ZZ 6x13x5 mm
(charlyyyyhase@gmail.com)**

1 Nachricht

Verkauf | Kugellager-Express GmbH <sales@kugellager-express.de>

Antwort an: sales@kugellager-express.de

An: Charlotte Hasenauer <charlyyyyhase@gmail.com>

23. September 2019 um 07:57

Sehr geehrte Frau Hasenauer,

vielen Dank für das Interesse an unseren Produkten und Ihrer Nachfrage bzgl. der Belastungsdaten.

Anbei senden wir Ihnen die Richtwerte zu den Belastungsdaten der angefragten Produkttype/n.

Produkttype:	dynamische Tragzahl, radial (N)	statische Tragzahl, radial (N)	Grenzdrehzahl (1/min)
686-ZZ	ca. 1105	ca. 410	ca. 41.000

Das Artikelgewicht beträgt 0,0018 kg.

Bei weiteren Rückfragen stehen wir Ihnen gerne zur Verfügung!

Mit freundlichen Grüßen

i. A. Isabel Pepperl

Kugellager-Express GmbH

[Grenzweg 14a](#)

[33758 Schloß Holte-Stukenbrock](#)

Tel: +49 (0) 5207/92917-0

Fax: +49 (0) 5207/92917-28

E-Mail: i.pepperl@kugellager-express.de

Internet: <http://www.kugellager-express.de>

Registergericht

Amtsgericht Bielefeld, HRB 41312

Umsatzsteuer ID: DE293181333

Geschäftsführer: Benjamin Küpper

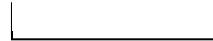
Der Inhalt dieser Nachricht ist vertraulich und nur für den angegebenen Empfänger bestimmt. Jede Form der Kenntnisnahme oder Weitergabe durch Dritte ist unzulässig. Sollte diese Nachricht nicht für Sie bestimmt sein, so bitten wir Sie, sich mit uns per E-Mail oder telefonisch in Verbindung zu setzen. Vielen Dank.

Von: Kugellager-Express GmbH <sales@kugellager-express.de>

Gesendet: Sonntag, 22. September 2019 16:38

An: Kugellager-Express GmbH <sales@kugellager-express.de>

Betreff: Frage zu: Miniatur Kugellager 686 ZZ 6x13x5 mm (charlyyyyhase@gmail.com)



Shop: Kugellager-Express GmbH

Frage zu Miniatur Kugellager 686 ZZ 6x13x5 mm
folgendem
Produkt:

E-Mail-Adresse charlyyyyhase@gmail.com
des Kunden:

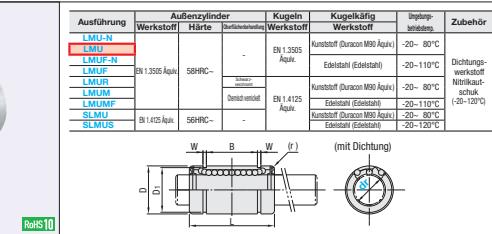
Frage: Sehr geehrtes Kugellager-Express-Team, im Rahmen eines Universitätsprojektes wurde das miniature-deep-groove-ball-bearing-686-zz-6x13x5-mm verbaut. Für weitere Berechnungen und Dokumentationsarbeit wird ein Datenblatt dessen inkl. Gewichtsangabe benötigt. Ich wäre Ihnen sehr dankbar, wenn Sie mir dieses zukommen lassen könnten. Mit freundlichen Grüßen Charlotte Hasenauer

Linearkugellager eine Nut

= Für Kunden, die Artikel gemäß Industriestandards wählen =
Das ist im roten Rahmen dargestellt den Standard Industriestandards Außenzyliner Bt 1.355 Aquak. Kunststofffahrtzug.
Beachten Sie diese Spezifikationen bei der Artikelauswahl.

■ Merkmale: Die beliebteste Linearkugellager-Ausführung.

Industriestandard	Ausführung	Außenzyylinder		Kugeln		Kugelkäfig		Ungleich- schleifung	Zubehör
		Werkstoff	Härte	Durchmesserhülse	Werkstoff	Werkstoff	Werkstoff		
LMU-N	LMU	EN 1.355 Aquak.	58HRC-	-	Kunststoff (Baracan M10 Aquak.)	Edelstahl (Edelstahl)	Edelstahl (Edelstahl)	-20 - 80 °C	Dichtungs- werkstoff
LMU-N	LMUW	EN 1.355 Aquak.	58HRC-	-	Kunststoff (Baracan M10 Aquak.)	Edelstahl (Edelstahl)	Edelstahl (Edelstahl)	-20 - 110 °C	Dichtungs- werkstoff
LMUR	LMUR	EN 1.355 Aquak.	58HRC-	-	Kunststoff (Baracan M10 Aquak.)	Edelstahl (Edelstahl)	Edelstahl (Edelstahl)	-20 - 80 °C	Nitrilkaut- schuk
LMUR	LMUWF	EN 1.355 Aquak.	58HRC-	-	Kunststoff (Baracan M10 Aquak.)	Edelstahl (Edelstahl)	Edelstahl (Edelstahl)	-20 - 110 °C	Nitrilkaut- schuk (-20 - 120 °C)
SLMU	SLMUS	EN 1.415 Aquak.	58HRC-	-	Kunststoff (Baracan M10 Aquak.)	Edelstahl (Edelstahl)	Edelstahl (Edelstahl)	-20 - 80 °C	Nitrilkaut- schuk
SLMU	SLMUS	EN 1.415 Aquak.	58HRC-	-	Kunststoff (Baracan M10 Aquak.)	Edelstahl (Edelstahl)	Edelstahl (Edelstahl)	-20 - 120 °C	Nitrilkaut- schuk



RoHS10

Teilenummer	Toleranz D	L		B	Toleranz	W	D1	(r)	Radial- sitz (Max.)	Kugel- reihen	Tragzahl (dyn/mm²) (C0 statisch N)	Gewicht (g)
		Ausführung	dr	Toleranz								
LMU-N (ohne Dichtung)	3 0	7	0	0	10	0	-	-	69	105	1.4	
LMU	4 -0.008	10	-0.009 -0.015	12	-0.12	8	-	-	88	127	2.0	
LMU	5 0	10	-0.009 -0.015	12	-0.12	8	0.6	-	107	145	3.0	
LMUF-N (ohne Dichtung)	6 0	12	0	0	19	11.3	1.1	11.5	206	265	8.5	
LMUF	8 0	15	-0.011 -0.018	24	0.2	15.3	14.3	0.4	265	380	17	
LMUR	10 0	19	-0.009	21	0	30	19.4	0.4	372	549	31	
LMUM	12 -0.008	23	-0.013 -0.021	32	0.2	20	20	0.4	412	598	41	
LMUMF	16 0	25	-0.012	29	0.2	27	22	0.8	510	784	46	
SLMU	25 0	32	-0.016 -0.025	59	0.2	37.3	30.5	0.4	582	1370	98	
SLMUS	30 -0.010	40	0	0	64	42	37.3	0.4	980	1570	236	
LMU	35 0	52	-0.016 -0.025	70	0.3	45.3	43	1.5	1570	2740	262	
LMU	40 -0.012	60	-0.019 -0.030	80	0.3	56.3	49	1.5	1670	3140	425	
LMU	45 0	70	-0.019 -0.030	90	0.3	68.8	57	1.5	2160	4020	654	
LMU	50 0	80	-0.019 -0.030	100	0.3	-	-	-	3820	7940	1700	

kfg=N0.101972

* Für Gleitlager zu Referenzsiehe S.303.

* Für LMU-N und LMUF-N siehe S.303.

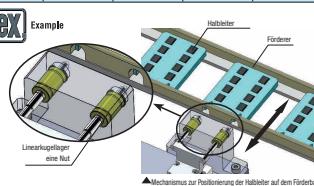
* Artikel mit d=1,4 sind ohne Dichtung. Die Ausführung ohne Dichtung hat einen geringeren Gleitwiderstand (0.4 - 1.26). Es sind gesonderte Schutzmaßnahmen erforderlich, um die gleitenden Teile vor eindringendem Staub zu schützen.

* Für LUBC-Beschichtung siehe S.304.

* Distanzheller und Ansätze für Linearkugellager siehe S.330.

dr	Stückpreis								
	LMU-N	LMU	LMUF-N	LMUF	LMUR	LMUM	LMUMF	SLMU	SLMUS
3	-	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-
25	-	-	-	-	-	-	-	-	-
30	-	-	-	-	-	-	-	-	-
35	-	-	-	-	-	-	-	-	-
40	-	-	-	-	-	-	-	-	-
50	-	-	-	-	-	-	-	-	-

Ordering Example	Teilenummer	Teilebeschreibung	
		LMU20	LMU-N (ohne Dichtung)
	LMU20	LUBC-Beschichtung	
	LMUR20	(Typ L mit Schmierstoff)	
	LMUF20	(Typ G mit Schmierstoff)	
	LMUM20	(Typ G mit Schmierstoff)	
	LMUR20H	(Typ H mit Schmierstoff)	
		Alternative Schmierstoffarten verfügbar.	
		Für Lieferzeit, Preis und Leistung, siehe S.304.	



▲ Mechanismus zur Positionierung der Halbleiter auf dem Förderband

* Für Gleitlager zu Referenzsiehe S.303.

* Für LMU-N und LMUF-N siehe S.303.

* Artikel mit d=1,4 sind ohne Dichtung. Die Ausführung ohne Dichtung hat einen geringeren Gleitwiderstand (0.4 - 1.26). Es sind gesonderte Schutzmaßnahmen erforderlich, um die gleitenden Teile vor eindringendem Staub zu schützen.

* Für LUBC-Beschichtung siehe S.304.

* Distanzheller und Ansätze für Linearkugellager siehe S.330.

Linearkugellager zwei Nuten

= Für Kunden, die Artikel gemäß Industriestandards wählen =
Das ist im roten Rahmen dargestellt den Standard Industriestandards Außenzyliner Bt 1.355 Aquak. Kunststofffahrtzug.
Beachten Sie diese Spezifikationen bei der Artikelauswahl.

■ Merkmale: Die gebräuchlichste Spezifikation bei Linearkugellagern. Verwenden Sie die doppelte Ausführung, wenn eine Momentlast aufgebracht werden soll.

Industriestandard	Ausführung	Außenzyylinder		Kugeln		Kugelkäfig		Ungleich- schleifung	Zubehör
		Werkstoff	Härte	Durchmesserhülse	Werkstoff	Werkstoff	Werkstoff		
LMUW-N	LMUW	EN 1.355 Aquak.	58HRC-	-	Kunststoff (Baracan M10 Aquak.)	Edelstahl (Edelstahl)	Edelstahl (Edelstahl)	-20 - 80 °C	Dichtungs- werkstoff
LMUW	LMUW	EN 1.355 Aquak.	58HRC-	-	Kunststoff (Baracan M10 Aquak.)	Edelstahl (Edelstahl)	Edelstahl (Edelstahl)	-20 - 110 °C	Dichtungs- werkstoff
LMUWR	LMUWR	EN 1.355 Aquak.	58HRC-	-	Kunststoff (Baracan M10 Aquak.)	Edelstahl (Edelstahl)	Edelstahl (Edelstahl)	-20 - 80 °C	Dichtungs- werkstoff
LMUWM	LMUWM	EN 1.355 Aquak.	58HRC-	-	Kunststoff (Baracan M10 Aquak.)	Edelstahl (Edelstahl)	Edelstahl (Edelstahl)	-20 - 110 °C	Dichtungs- werkstoff
LMUWF	LMUWF	EN 1.415 Aquak.	58HRC-	-	Kunststoff (Baracan M10 Aquak.)	Edelstahl (Edelstahl)	Edelstahl (Edelstahl)	-20 - 80 °C	Dichtungs- werkstoff
SLMWS	SLMWS	EN 1.415 Aquak.	58HRC-	-	Kunststoff (Baracan M10 Aquak.)	Edelstahl (Edelstahl)	Edelstahl (Edelstahl)	-20 - 120 °C	Dichtungs- werkstoff

■ Merkmale: Die gebräuchlichste Spezifikation bei Linearkugellagern. Verwenden Sie die doppelte Ausführung, wenn eine Momentlast aufgebracht werden soll.

Industriestandard	Ausführung	Außenzyylinder		Kugeln		Kugelkäfig		Ungleich- schleifung	Zubehör
		Werkstoff	Härte	Durchmesserhülse	Werkstoff	Werkstoff	Werkstoff		
LMUW-N	LMUW	EN 1.355 Aquak.	58HRC-	-	Kunststoff (Baracan M10 Aquak.)	Edelstahl (Edelstahl)	Edelstahl (Edelstahl)	-20 - 80 °C	Dichtungs- werkstoff
LMUW	LMUW	EN 1.355 Aquak.	58HRC-	-	Kunststoff (Baracan M10 Aquak.)	Edelstahl (Edelstahl)	Edelstahl (Edelstahl)	-20 - 110 °C	Dichtungs- werkstoff
LMUWR	LMUWR	EN 1.355 Aquak.	58HRC-	-	Kunststoff (Baracan M10 Aquak.)	Edelstahl (Edelstahl)	Edelstahl (Edelstahl)	-20 - 80 °C	Dichtungs- werkstoff
LMUWM	LMUWM	EN 1.355 Aquak.	58HRC-	-	Kunststoff (Baracan M10 Aquak.)	Edelstahl (Edelstahl)	Edelstahl (Edelstahl)	-20 - 110 °C	Dichtungs- werkstoff
LMUWF	LMUWF	EN 1.415 Aquak.	58HRC-	-	Kunststoff (Baracan M10 Aquak.)	Edelstahl (Edelstahl)	Edelstahl (Edelstahl)	-20 - 80 °C	Dichtungs- werkstoff
SLMWS	SLMWS	EN 1.415 Aquak.	58HRC-	-	Kunststoff (Baracan M10 Aquak.)	Edelstahl (Edelstahl)	Edelstahl (Edelstahl)	-20 - 120 °C	Dichtungs- werkstoff

■ Merkmale: Die gebräuchlichste Spezifikation bei Linearkugellagern. Verwenden Sie die doppelte Ausführung, wenn eine Momentlast aufgebracht werden soll.

Industriestandard	Ausführung	Außenzyylinder		Kugeln		Kugelkäfig		Ungleich- schleifung	Zubehör
		Werkstoff	Härte	Durchmesserhülse	Werkstoff	Werkstoff	Werkstoff		
LMUW-N	LMUW	EN 1.355 Aquak.	58HRC-	-	Kunststoff (Baracan M10 Aquak.)	Edelstahl (Edelstahl)	Edelstahl (Edelstahl)	-20 - 80 °C	Dichtungs- werkstoff
LMUW	LMUW	EN 1.355 Aquak.	58HRC-	-	Kunststoff (Baracan M10 Aquak.)	Edelstahl (Edelstahl)	Edelstahl (Edelstahl)	-20 - 110 °C	Dichtungs- werkstoff
LMUWR	LMUWR	EN 1.355 Aquak.	58HRC-	-	Kunststoff (Baracan M10 Aquak.)	Edelstahl (Edelstahl)	Edelstahl (Edelstahl)	-20 - 80 °C	Dichtungs- werkstoff
LMUWM	LMUWM	EN 1.355 Aquak.	58HRC-	-	Kunststoff (Baracan M10 Aquak.)	Edelstahl (Edelstahl)	Edelstahl (Edelstahl)	-20 - 110 °C	Dichtungs- werkstoff
LMUWF	LMUWF	EN 1.415 Aquak.	58HRC-	-	Kunststoff (Baracan M10 Aquak.)	Edelstahl (Edelstahl)	Edelstahl (Edelstahl)	-20 - 80 °C	Dichtungs- werkstoff
SLMWS	SLMWS	EN 1.415 Aquak.	58HRC-	-	Kunststoff (Baracan M10 Aquak.)	Edelstahl (Edelstahl)	Edelstahl (Edelstahl)	-20 - 120 °C	Dichtungs- werkstoff



GT2 Riemen 6mm 1m



Art.Nr.: 000078.111141

Lieferzeit: ● Ab Lager - Sofort Lieferbar (Ausland abweichend)

Lagerbestand: 256 Meter

Versandgewicht: 0,01 kg je Meter

1,49 EUR

inkl. 19% MwSt. zzgl. [Versand](#)

Meter:

-	1	+
---	---	---

[IN DEN WARENKORB](#)

[Auf den Merkzettel](#)

[Frage zum Produkt](#)

Direkt zu

Literaturverzeichnis

- [Adafruit-MSG] *Motor Selection Guide - Motor Assemblies.* 11.08.2019. URL: <https://learn.adafruit.com/adafruit-motor-selection-guide/motor-assemblies>.
- [BBC] BBC. 'Unbeatable' air hockey robot invented in Japan. 3.09.2019. URL: <https://www.bbc.co.uk/newsround/22926592>.
- [DUB] Karl-Heinrich Grote. *Dubbel: Taschenbuch für den Maschinenbau.* 22. Aufl. s.l.: Springer-Verlag, 2007. ISBN: 978-3-540-49714-1. URL: <http://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=261907>.
- [I2C] *I2C Info – I2C Bus, Interface and Protocol.* URL: <https://i2c.info/i2c-bus-specification>.
- [ITQ] ITQ GmbH und AMK Holding GmbH & Co. KG. *Smart Airhockey.* 3.09.2019.
- [JFalconer] Jason Falconer. *This Robot Wants To Beat You At Air Hockey.* 3.09.2019. URL: <https://spectrum.ieee.org/automaton/robotics/artificial-intelligence/japanese-air-hockey-robot>.
- [JJRobots] JJRobots. *Air Hockey Robot (a 3D printer hack).* 3.09.2019. URL: <https://www.jjrobots.com/air-hockey-robot-a-3d-printer-hack/>.
- [LFarlow] Lindsay Farlow. *Outrageous Engineering Challenge – Air Hockey Robot.* 29.08.2019. URL: <https://www.nuvation.com/resources/article/outrageous-engineering-challenge-air-hockey-robot>.
- [Motion] Paul Read und Mark-Paul Meyer. *Restoration of Motion Picture Film.* Elsevier, 2000.
- [RaspberryPia] Raspberry Pi Foundation. *Buy a Raspberry Pi 3 Model B+.* 16.09.2019. URL: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>.
- [RaspberryPiCamerab] Raspberry Pi Foundation. *Camera Module - Raspberry Pi Documentation.* 16.09.2019. URL: <https://www.raspberrypi.org/documentation/hardware/camera/>.

- [Step-Speed] EE Times-India. *Generate stepper-motor speed profiles in real time*. 1.01.2005. URL: http://web.archive.org/web/20140705143928/http://fab.cba.mit.edu/classes/MIT/961.09/projects/i0/Stepper_Motor_Speed_Profile.pdf.
- [TM] Roland Gomeringer u. a. *Tabellenbuch Metall: Mit Formelsammlung*. 47., neu bearbeitete und erweiterte Auflage, korrigierter Nachdruck 2017. Europa-Fachbuchreihe für Metallberufe. Haan-Gruiten: Verlag Europa Lehrmittel - Nourney Vollmer GmbH & Co. KG, 2017. ISBN: 9783808517277.
- [Ulti] *Anatomy of an Ultimaker 2+*. 21.09.2019. URL: <https://ultimaker.com/en/resources/22131-anatomy-of-an-ultimaker-2>.
- [Wiki-Rob] Wikipedia. *Portalroboter*. 14.05.2019. URL: <https://wiki.induux.de/Portalroboter>.
- [Wiki-SMa] *Schrittmotor*. 14.07.2019. URL: <https://de.wikipedia.org/wiki/Schrittmotor#Anwendungsbereiche>.
- [Wiki-Stahlb] *Stahl*. 12.09.2019. URL: <https://de.wikipedia.org/wiki/Stahl>.