

# BALANCER ROBOT

## Berufsmaturitätsarbeit

### Wichtig

Dieses Projekt ist im Rahmen der Berufsmaturitätsarbeit 2019 an der BMZ entstanden. Da der Abgabetermin für diese Arbeit erst am 26.11.2019 ist und auch die Information über die Stiftung Schweizer Jugend Forscht erst am 22.10.2019 an der Schule kommuniziert wurde, ist meine Arbeit noch nicht fertig.

Trotzdem möchte ich diese Arbeit gerne Einreichen, damit Sie sich ein Bild über das Projekt machen können und mir vielleicht sogar eine Teilnahme ermöglichen.  
Besten Dank!

Lacher, Markus  
markuslacher@hotmail.ch  
28.10.2018



## Abstract

### Inhalt

Abstract .....	1
Einleitung.....	2
Fragestellung .....	2
Hauptteil.....	3
Konzept.....	3
Physikalisches Grundprinzip.....	3
Software .....	4
Sensorik .....	5
Rotation .....	5
Beschleunigung .....	5
Konfiguration.....	5
Datenverarbeitung .....	6
Sensor Auswertung .....	7
Elektronik.....	8
Ansteuerung Motor.....	8
Motor Stromaufnahme .....	10
Mechanik.....	12
Regelungsalgorithmus.....	12
Schluss .....	13
Fazit .....	13
Glossar .....	13
Quellenverzeichnis .....	13
Abbildungsverzeichnis.....	13
Tabellenverzeichnis .....	13
Dank.....	13
Anhang .....	13



## Einleitung

### Fragestellung



## Hauptteil

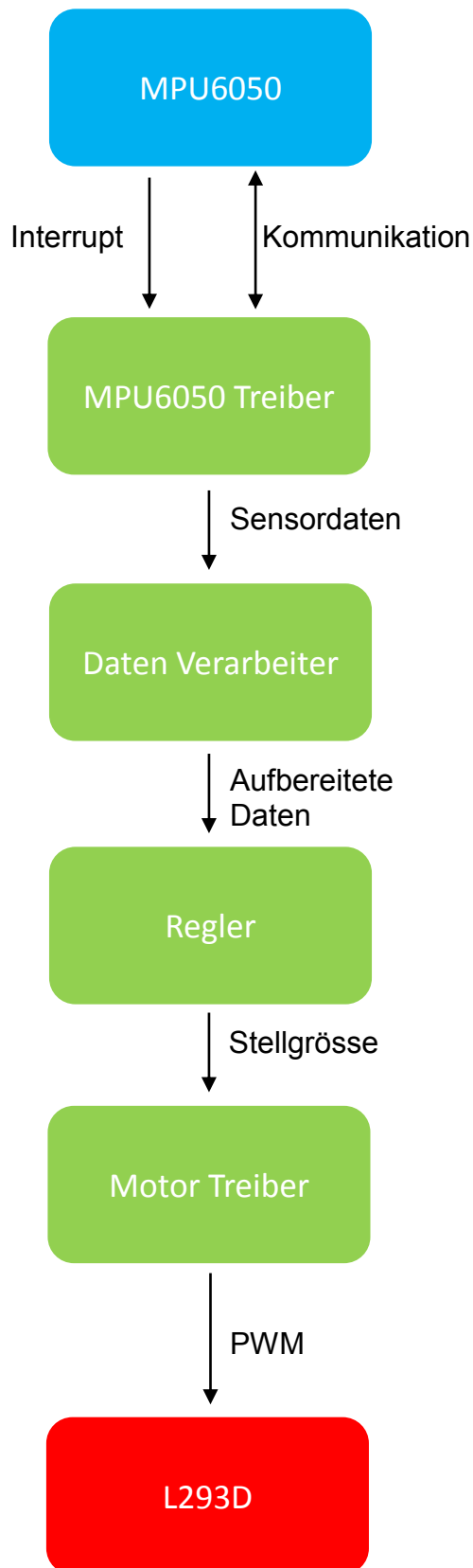
### Konzept

Der Balancer Robot ist im Prinzip nichts anderes als eine Regelungselektronik. Über Sensorik wird eine physikalische Grösse gemessen und mithilfe der Regelung wird ein Aktor angesteuert, sodass sich ein gewollter Sollwert einstellt.

Der Balancer Robot misst mithilfe eines 3-Achsen Gyrosensors und eines 3-Achsen Beschleunigungssensors die Kräfte die auf den Roboter wirken. Diese Messwerte werden von einem Mikrocontroller eingelesen. Dieser verarbeitet die Messwerte und steuert die Motoren des Roboters an, um diesen vom Umfallen zu bewahren.

### Physikalisches Grundprinzip

Kräfte beschreiben, evtl. Pendelprinzip



#### Software

Blau: Sensoren

Grün: Mikrocontroller (Software)

Rot: Aktoren

Der MPU6050-Treiber konfiguriert den Sensor, sodass dieser mit einer Frequenz von 100Hz die 3-Achsen des Gyrosensors, sowie die 3-Achsen des Accelerometers misst. Über einen Interrupt Pin wird dem Mikrocontroller signalisiert, dass die Daten bereitstehen. Der Mikrocontroller liest dann sofort die Daten aus.

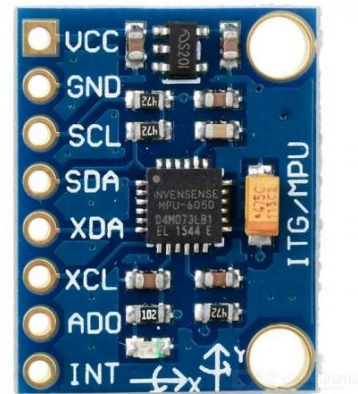
Die Sensordaten werden dem Datenverarbeiter übergeben. Dieser berechnet den dynamischen Mittelwert der Daten, um diese zu Glätten und Störungen herauszufiltern.

Die Aufbereiteten Daten werden dem Regler übergeben. Dieser errechnet mithilfe des Regelalgorithmus die Stellgrösse für die Motoren.

Die Stellgrösse wird dem Motortreiber übergeben. Aufgrund der Stellgrösse steuert der Motor Treiber die Motoren per PWM Signal an.

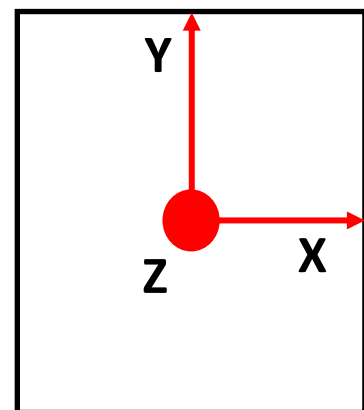
## Sensorik

Die rechts abgebildete Sensorleiterplatte ist mit dem MPU-6050, ein sehr verbreiteter Gyro- und Beschleunigungssensor, bestückt. Dieser misst in allen 3-Achsen die Beschleunigung, sowie die Rotationsgeschwindigkeit. Über die digitale I2C Kommunikationsschnittstelle kann der Sensor konfiguriert und ausgelesen werden.



## Rotation

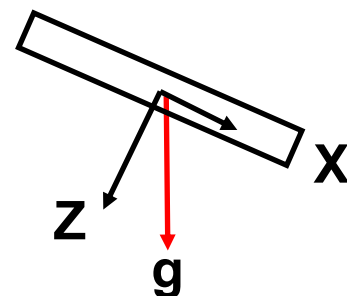
Der Gyrosensor misst die Rotation um eine Achse in Grad/s. Je nach Konfiguration kann der Sensor bis zu 1000°/s messen. Befindet sich der Sensor in Ruhelage, ist die Rotation aller Achsen gleich 0. Man könnte den Winkel bestimmen indem man die Rotationsgeschwindigkeit pro Zeiteinheit aufsummiert, jedoch driftet die Berechnung des Winkels ab, da Messfehler addiert werden und somit das Ergebnis mehr und mehr verfälscht wird.



## Beschleunigung

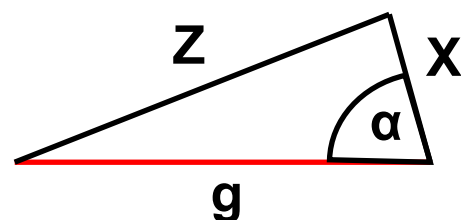
Die Beschleunigung wird in  $m/s^2$  gemessen. Je nach Konfiguration kann der Sensor bis zu  $\pm 16G$  messen, das entspricht der 16-fachen Erdbeschleunigung also ca.  $160m/s^2$ .

Liegt der Sensor still, ist die einzige Kraft, welche auf ihn wirkt, die Erdbeschleunigung. Würde der Sensor flach auf dem Tisch liegen wie oben abgebildet, dann misst man auf der Z-Achse eine Beschleunigungskraft von  $9.81m/s^2$ . Auf der X- und Y-Achse liegt keine Kraft an. Beginnt man nun den Sensor zu kippen, teilt sich die Kraft der Erdbeschleunigung auf die Achsen auf. Mithilfe von Vektorgeometrie lässt sich dann der Winkel bestimmen.



$$\alpha = \arctan\left(\frac{Z}{X}\right)$$

Um den Roboter auszubalancieren versucht die Regelung den Winkel Alpha immer bei 90° zu halten.



## Konfiguration

Der Sensor wird so konfiguriert, dass er mit einer Frequenz von 1000Hz die Messwerte der 6-Achsen aufzeichnet. Sobald diese bereit zum Auslesen sind, wird dies über den Pin INT (Interrupt) signalisiert. Die Messwerte werden dann sofort vom Mikrocontroller ausgelesen und verarbeitet.

## Datenverarbeitung

Der Sensor besitzt einen internen DMP. Richtig konfiguriert filtert dieser die Messdaten mit einem Tiefpassfilter, damit Störungen bereits intern im Sensor herausgefiltert werden können.

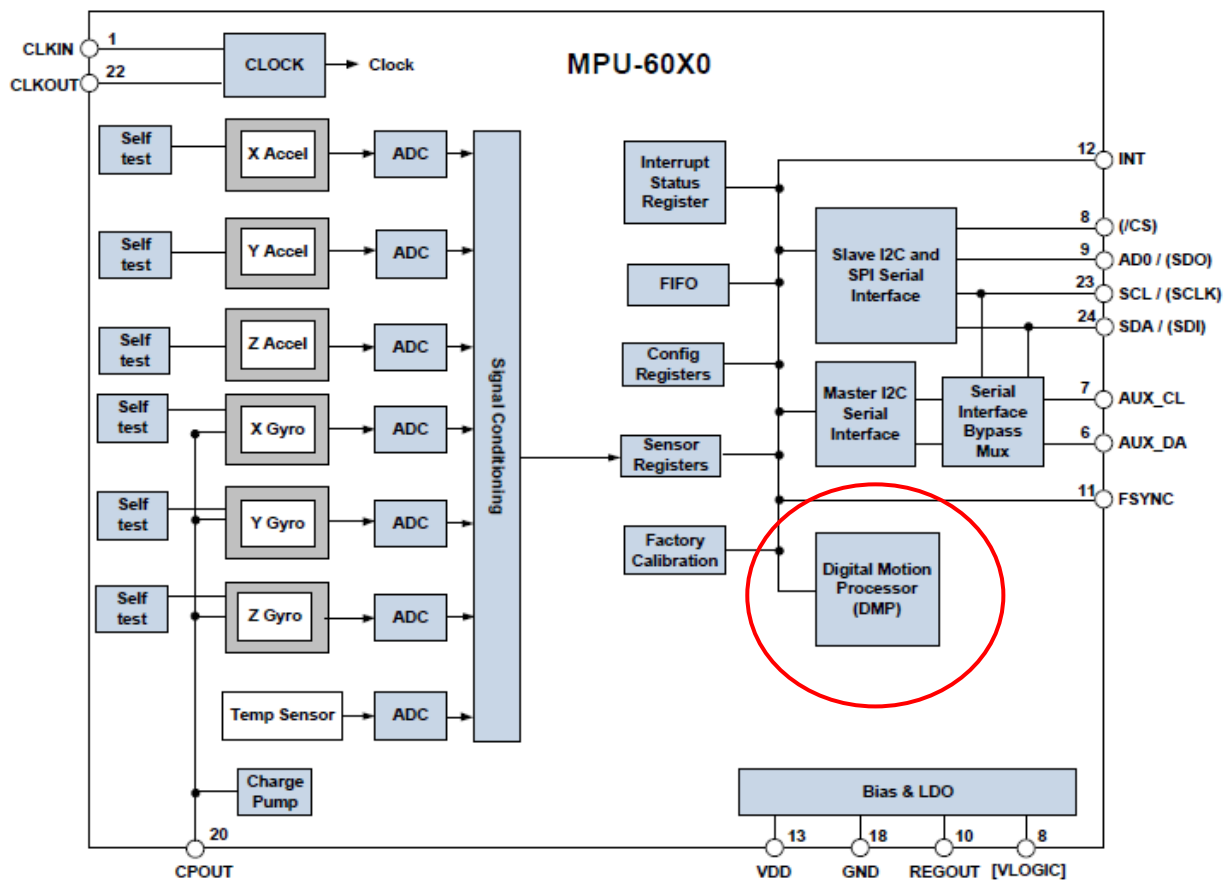


Abbildung 2: (InvenSense, 2019)

Auch per Software werden die Daten nochmals optimiert. Das Daten Verarbeiter Modul glättet die Messwerte mithilfe eines gleitenden Mittelwerts.

Durch das Tiefpassfilter und den gleitenden Mittelwert werden Störungen, Messfehler und Rauschen des Sensor minimiert.

Einziger Nachteil: Durch das Verarbeiten der Daten wird die Zeitverzögerung der tatsächlichen Messung bis zu deren Verarbeitung durch Regler grösser. Dies kann das Verhalten des Reglers negativ beeinflussen.

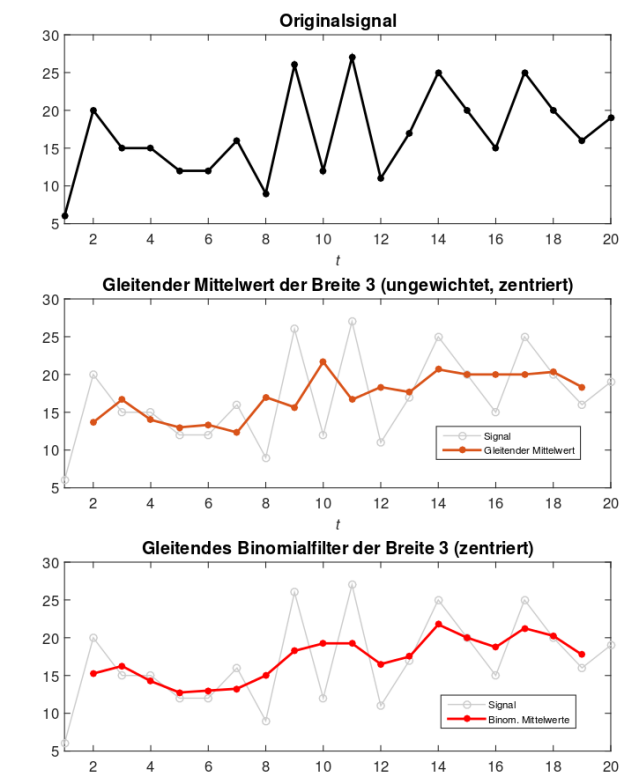


Abbildung 1: (Wikipedia, 2019)

## Sensor Auswertung

Um den Sensor zu testen wurde eine erste Testsoftware für den Mikrocontroller geschrieben, welche die gemessenen Werte über die UART-Schnittstelle des Mikrocontrollerboards an einen angeschlossenen PC sendet. Zusätzlich wurde eine Python Software entwickelt, welche die gesendeten Daten verarbeitet und diese in eine Excel Tabelle schreibt. Dadurch konnte das Verhalten des Sensors analysiert werden.

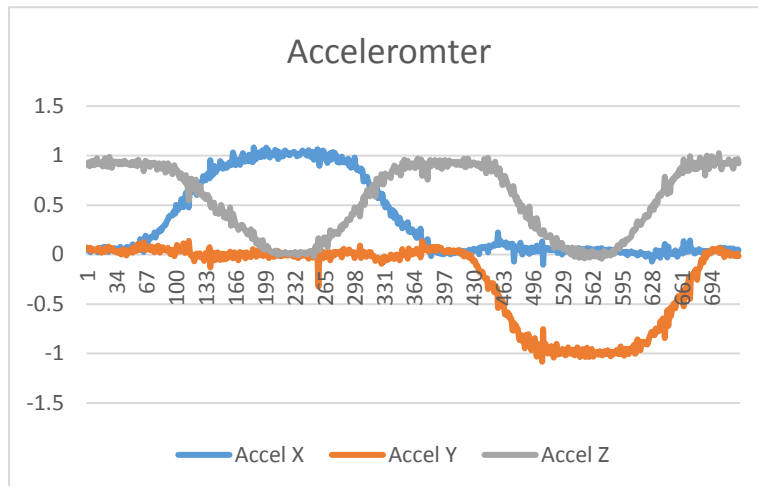


Abbildung 3: Messung des Accelerometers

### Kippen von Hand

Der Sensor wurde von Hand um alle drei Achsen gedreht. Die Messwerte sind stark verrauscht, da einerseits die Handbewegungen nicht exakt sind, zudem ist das digitale Tiefpassfilter des Sensors nicht eingeschaltet.

Die Y-Achse entspricht der Kraft in G und die X-Achse der Nummer des Messwerts.

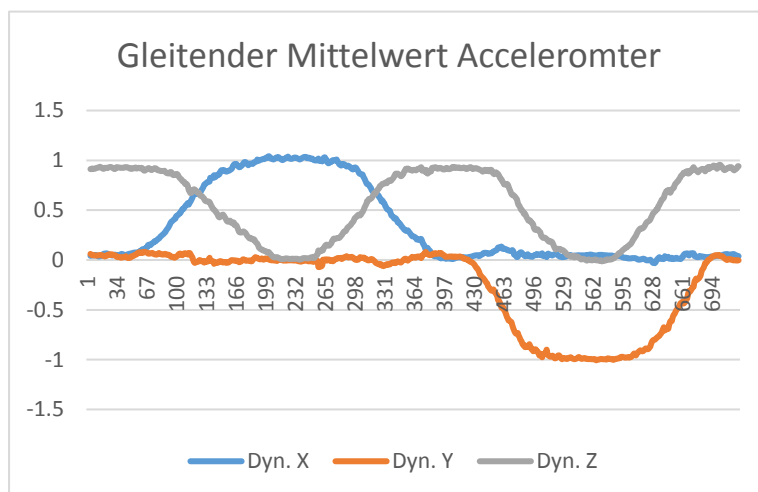


Abbildung 4: Messung mit gleitendem Mittelwert

### Gleitender Mittelwert

Im Excel wurde der gleitende Mittelwert aus den letzten 5 Werten gebildet. Das Rauschen wurde gedämpft und Störungen bzw. Spitzen wurde geglättet.

Grafiken des Gyrosensors anfügen



## Elektronik

### Ansteuerung Motor

Ein DC-Motor besitzt zwei Anschlüsse. Ändert die Richtung des Stromes, wechselt auch die Drehrichtung des Motors. Um dieses Wechseln des Stromes elektronisch umzusetzen wurde eine H-Brücke, auch Vierquadrantensteller genannt, verwendet.

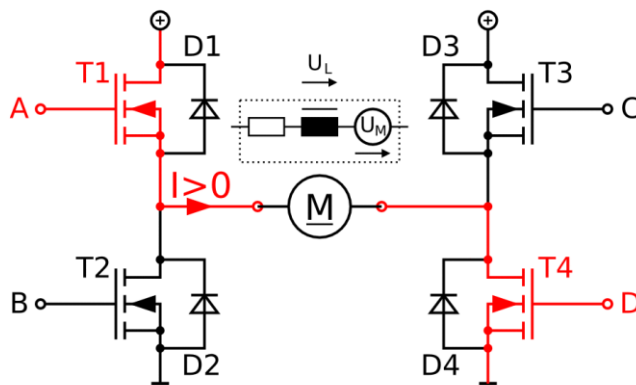


Abbildung 5: (Wikipedia, 2019)

Prinzip: Die Schaltung besitzt eine Motorspannung  $U_+$  und einen GND. Schaltet der Transistor T1 und T4 während T2 und T3 sperren, dann liegt vom linken zum rechten Anschluss des Motors eine positive Spannung an.

Sperrt T1 und T4 und leiten dafür T2 und T3, dann liegt eine negative Spannung vom linken zum rechten Anschluss an. Die Drehrichtung ändert.

Da ich die Anschlüsse A-D der H-Brücke mit CMOS-Logik (0 – 3.3V, gegeben durch Mikrocontroller) ansteuern möchte, die Motorspannung  $U_+$  jedoch bis zu 16V beträgt, müsste für den Anschluss A und C noch zusätzliche Komponenten für die Spannungstransformation verwendet werden. Ausserdem bräuchten die Bauteile einzeln verbaut mehr Platz. Deswegen wird der IC **L293D** verwendet.

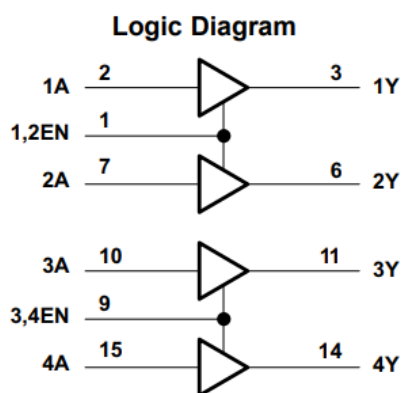


Abbildung 6: (Texas Instruments, 2019)

Der Baustein besitzt 4 High-Voltage Treiber. Über den Eingang A kann mit 0V oder 3.3V der Ausgang Y auf 0V oder auf den Level der Speisespannung geschaltet werden. Die Speisespannung kann bis zu 36V betragen. So kann an 1Y und 2Y der erste Motor und über 3Y und 4Y der zweite Motor angeschlossen werden.

Die Ausgänge können bis zu 1.2A liefern und auch „schlucken“. Mit den beiden Signalen 1,2EN und 3,4EN können die Treiber aktiviert werden.

Die Software soll die Leistungsabgabe des Motors steuern können, ansonsten ist eine Regelung nicht möglich. Um diese Steuerung zu erreichen wird die H-Brücke mit einem **PWM** Signal angesteuert. PWM steht für **P**ulse **W**ide **M**odulation.

Die Spannung über dem Motor soll reguliert werden. Da die Hardware jedoch nur zwei Zustände einstellen kann, (siehe H-Brücke) wird mit ständigem Ein- und Ausschalten die Spannung über dem Motor reguliert.

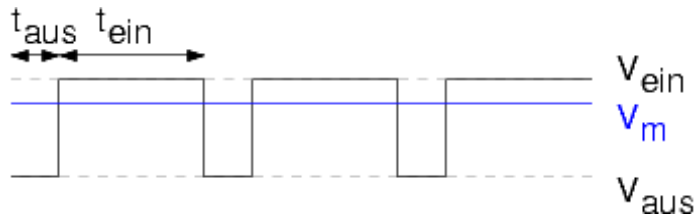


Abbildung 7: (Mikrocontroller.net, 2019)

$$T = t_{ein} + t_{aus}$$

$$f = \frac{1}{T}$$

$$Tastgrad = \frac{t_{ein}}{T}$$

## Motor Stromaufnahme

### Problem

Die ersten Tests mit dem L293D mit PWM Ansteuerung verliefen gut. Die Motoren liessen sich in beide Richtungen drehen. Der L293D wurde aber relativ heiss. Doch als ich den Roboter das erste Mal zusammengebaut hatte und diverse Probleme nach einer Zeitintensiven Fehlersuche endlich beheben konnte, wollte ich die komplette Software mit einem einfachen Regelungsalgorithmus testen.

Der L293D Baustein wurde noch heisser als sonst und ein Motor arbeitete nicht richtig. Ich vermutete der Baustein sei beschädigt, da ich in Anfangs falsch angeschlossen hatte. Als ich den Baustein austauschte in der Hoffnung die Ansteuerung der Motoren würde funktionieren, ging dieser direkt bei Einschalten der Motoren kaputt, der Mikrocontroller funktionierte auch nicht mehr.

### Theorie

Im Projekt werden Gleichstrom- Bürstenmotoren eingesetzt. Diese erzeugen durch die Rotation der Achse eine Gegeninduktion in der Spule, welche den Strom ausbremst. Je höher die Drehzahl des Motors, desto grösser ist die Gegenspannung und dementsprechend wird der Strom umso kleiner. Dies ist auch der Grund für die Tatsache, dass das Drehmoment bei Elektromotoren bei steigenden Drehzahlen abnimmt.

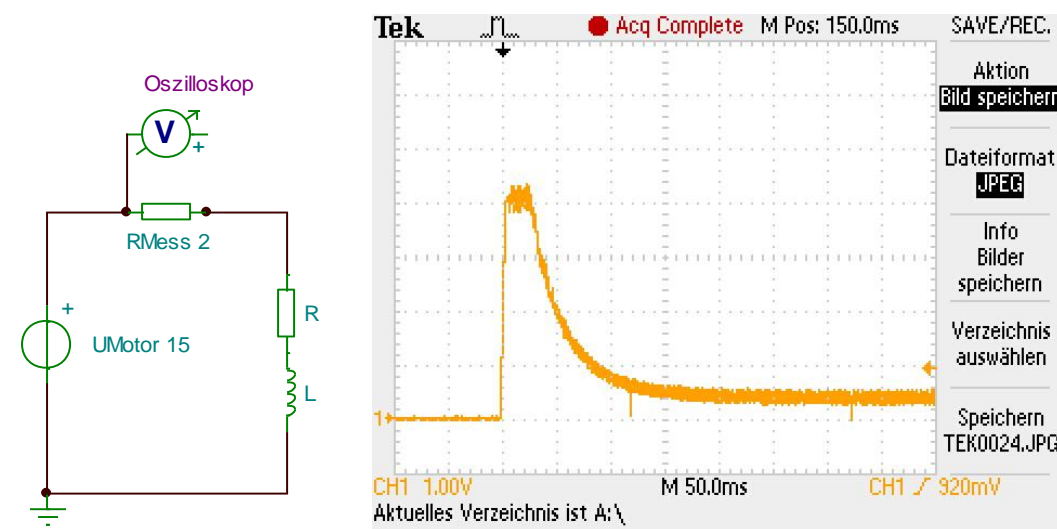


Abbildung 8: Messaufbau

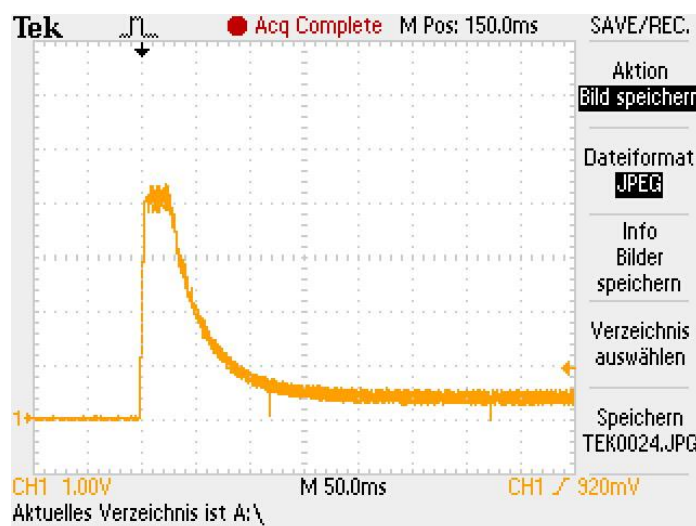


Abbildung 9: Messung 1, Start des Motors

### Analyse

Um das Problem zu analysieren wurde ein Messversuch durchgeführt. Über einen 2 Ohm Messwiderstand wurde mithilfe eines Oszilloskops die Stromaufnahme des Motors aufgezeichnet. Dieser ist nicht ideal, da der Ohm'sche Widerstand der Motorspule 3Ohm beträgt. Dies führt bei grossen Strömen zu Verminderung der Spulenspannung über dem Motor, da über dem Messwidertand eine zu grosse Spannung abfällt.

Trotzdem war der Fehler bereits nach der ersten Messung offensichtlich. Im Stillstand induziert die Motorwelle in der Spule noch keine Gegenspannung. Beim Einschalten wirkt somit ganz kurz die Impedanz der Spule, danach nur noch der Ohm'sche Widerstand von 3Ω. Bei 15V entsteht dann über einen Zeitbereich von 40ms ein Strom von über 4 Ampere. Danach sinkt der Strom innerhalb von 60ms auf 200mA. Dies entspricht ziemlich genau den Angaben des Datenblatts, welches bei keiner mechanischen Belastung einen Strom von 180mA angibt.

Der Motortreiber L293D kann pro Ausgang einen konstanten Strom von 1.2A Ampère liefern, was für den Normalbetrieb des Motors ausreichen würde. Doch der kurzzeitige Stromspitz beim Anfahren genügt um den Motortreiber zu zerstören.

Um der hohen Stromanforderung gerecht zu werden, wurde ein neues IC getestet. Der L298 ist ähnlich aufgebaut wie der alte Treiber L293D, jedoch kann dieser einen Strom von 2A liefern und kurzzeitige Spitzen bis zu 3A Ampere. Dieser wird im Betrieb auch sehr warm, funktioniert jedoch.

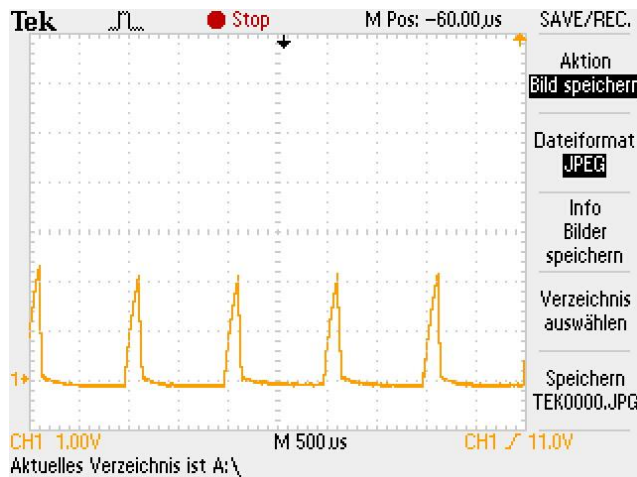


Abbildung 10: Messung des Stroms bei Anfahren mit PWM

Strommessung bei Anfahren mithilfe von PWM. Man sieht die PWM Frequenz von 1kHz sowie das Stromverhalten eines LC Gliedes, da die Kurve einer e-Funktion ähnelt. Da die Spannung nur kurzzeitig eingeschaltet ist, steigt der Strom nicht über 1.5A, weshalb der L293D intakt bleibt.

Gefährlich wird es, wenn der Motor mit einem hohen PWM Wert anfährt.

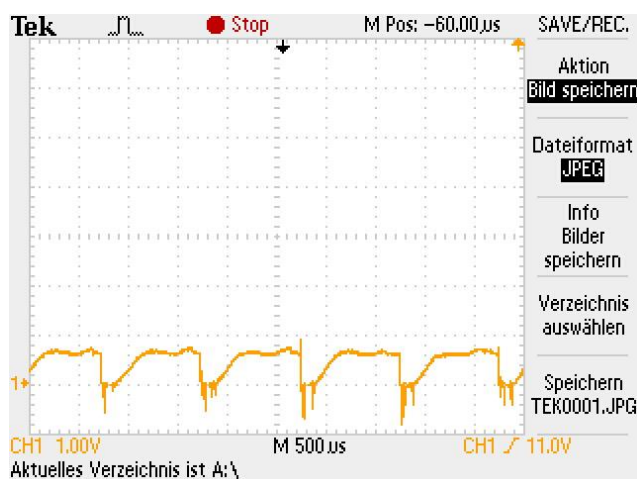


Abbildung 11: Messung des Stroms bei PWM 60%

Messung des Stroms bei laufendem Motor mit einem PWM von 60%. Man sieht gut, dass der Motor im Betrieb weniger Leistung benötigt als beim Anfahren. In der Messung sieht man auch gut wie der PWM arbeitet. Die negativen Stromspitzen werden beim Abschalten der Spule durch deren Selbstinduktion erzeugt.

## Mechanik

Die Mechanik spielt eine entscheidende Rolle im Projekt, um den Erfolg zu garantieren. Der Roboter sollte möglichst leicht sein, damit das benötigte Drehmoment um den Roboter wieder aufzurichten möglichst klein bleibt. Ausserdem muss auf die Gewichtsverteilung geachtet werden, um den Schwerpunkt tief zu halten.

Für den ersten Prototypen habe ich eine Grundplatte gezeichnet, welche von einem Mechaniker angefertigt wurde. Auf der Unterseite wurden die Motoren befestigt. Die Oberseite ist bestückt mit den Akkualtern und den Distanzbolzen, um weiter oben die Leiterplatte zu montieren.

## Mechanische Zeichnung einfügen

## Regelungsalgorithmus

Regelungstechnik ist ein grosser Teil des Projekts. Im Grundprinzip funktioniert jede Regelungsschaltung, ob digital oder analog, auf dem gleichem Grundprinzip. Die Differenz zwischen der Führungsgrösse, also dem gewollten Sollwert und der Rückführgrösse, also die gemessene tatsächliche Grösse, wird einem Regler übergeben. Dieser errechnet die Stellgrösse für den Aktor, welcher in die Regelstrecke eingreift. Ein Sensor misst die Regelstrecke aus. Über die Messeinrichtung wird die Rückführgrösse erzeugt und als Rückführgrösse rückgekoppelt.

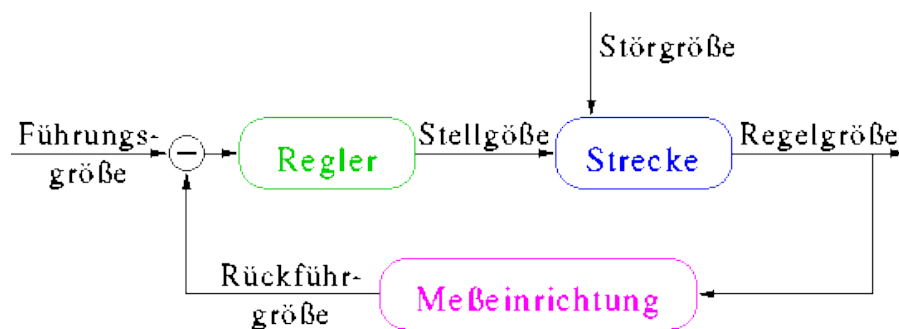


Abbildung 12: (Heuler, 2019)

Konkret auf den Balancer Robot angewendet lässt sich die Regelstrecke folgendermassen beschreiben: Abhängig von der errechneten Stellgrösse wird die Spannung über den Motoren reguliert. Durch äussere Störgrössen wie z.B. die Erdanziehungskraft kippt der Roboter in eine Richtung. Der Sensor detektiert die Veränderung des Winkels und somit die Abweichung vom Sollwert. Der Regler berechnet eine neue Stellgrösse um den falschen Winkel wieder zu korrigieren.

## PID-Regler Prinzip



Schluss

Fazit

Glossar

Quellenverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Dank

Anhang