

Project Data Explorer - Mohne

Projektbericht

Entwicklung eines Fahrzeugs für autonome/ ferngesteuerte Datenerfassung inklusive der Umweltbedingungen

Eingereicht bei:

**Fachhochschule Kufstein Tirol Bildungs GmbH**

**Studiengang Smart Products & Solutions**

Verfasser:

**Anna Duregger, Christian Gruber, Christoph Jungwirth, Maximillian Möbes, Ludwig Paula, Markus Wiesmüller**

Abgabedatum:

**10.06.2017**

Table of Contents

[List of Figures III](#_Toc513277774)

[Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden. III](#_Toc513277775)

[List of Tables IV](#_Toc513277776)

[List of Abbreviation V](#_Toc513277777)

[Kurzfassung VI](#_Toc513277778)

[1. Konzeptentwicklung 27](#_Toc513277779)

[1.1 Konzept der Fahrmodi 27](#_Toc513277780)

[1.1.1 autonom 27](#_Toc513277781)

[1.1.2 ferngesteuert 27](#_Toc513277782)

[1.2 Fahrzeugaufbau 27](#_Toc513277783)

[1.2.1 Elektronik 27](#_Toc513277784)

[1.2.2 Software 27](#_Toc513277785)

[2. Model Based Analytics 27](#_Toc513277786)

[2.1 Digitaler Zwilling 27](#_Toc513277787)

[3. Simulation 27](#_Toc513277788)

[3.1 Pi Simulation 27](#_Toc513277789)

[3.2 Arduino Simulation 27](#_Toc513277790)

[4. Datenübertragung 27](#_Toc513277791)

[4.1 Thinkspeak 27](#_Toc513277792)

[4.2 Auswertung 27](#_Toc513277793)

[5. Zusammenfassung 35](#_Toc513277794)

# ****List of Figures****

[Abbildung 1: Mechanisches Kräftemodell des Autos 13](#_Toc516071526)

[Abbildung 2: Diagramm für die Beschleunigung 14](#_Toc516071527)

[Abbildung 3: Diagramm für die Winkelgeschwindigkeit 15](#_Toc516071528)

# ****List of Tables****

**Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.**

# ****List of Abbreviation****

# Kurzfassung

# Abstract

# Conecpt Development

Unser Team konnte zwischen zwei verschiedenen Fahrzeugtypen entscheiden. Grundsätzlich wollen wir auf unserem Erstprojekt der mobilen Drohne das neue Projekt weiter aufbauen, jedoch stand eine weitere Idee im Raum. Die Idee lag darin, ein einfaches Auto mit deren Grundfunktionen zu entwickeln. Schlussendlich sind wir doch zur Entscheidung gekommen, dass wir unsere „Mohne“ weiterentwickeln wollen.

Für das derzeitige Projekt sind folgende Aufgabenstellungen des Lektoren Teams gestellt worden:

* Eigenschaften des Fahrzeuges: Fahrmodus, Telemetriedaten erfassen und übertragen
* Datenerfassung: Umweltbedingungen permanent erfassen
* Digitaler Zwilling in MATLAB: Fahrzeug (mechanisches Modell-Kräfte)
* Simulation: Rapsberry Pi und Arduino Simulator
* Tracks: Ferngesteuert und Autonom

Die aufgezählten Punkte werden im folgenden Projektbericht detailliert erläutert und darauf eingegangen.

## Konzept der Fahrmodi

Die beiden Konzepte der Fahrmodi bestehen aus den Modi autonom Fahren und ferngesteuertes Fahren.

Bei der Aufgabenstellung autonomes Fahren – Obstacle Detection sind die Anforderungen folgende:

* Eindeutige markierte Hindernisse müssen umfahren werden
* Die Hindernisse müssen abfotografiert werden
* Es werden keine Begrenzungswände verwenden
* Startpunkt lieg bei (0,0)
* Endpunkt wird vorgegeben.

Beim ferngesteuerten Fahren - Datenerfassung sind folgende Anforderungen:

* Startpunkt bei (0,0)
* Abfolge von Messfeldern wird je Team vorgegeben
* Hindernisse müssen umfahren werden
* Explorer wird ferngesteuert (Eingabegerät offen; Eingabe von Koordinaten, Steuerbefehle mittels Joystick)

### autonom

### ferngesteuert

ludi

Wenn ich eine Taste drücke W,A,S,D

Q Taste löscht alle Befehle

X bricht das Programm ab, Strom wird abgeschaltet

I2C wird vom pi eine Zahl an Arduino geschickt und muss den motor ansteuern

## Fahrzeugaufbau

Explosionsdarstellung:



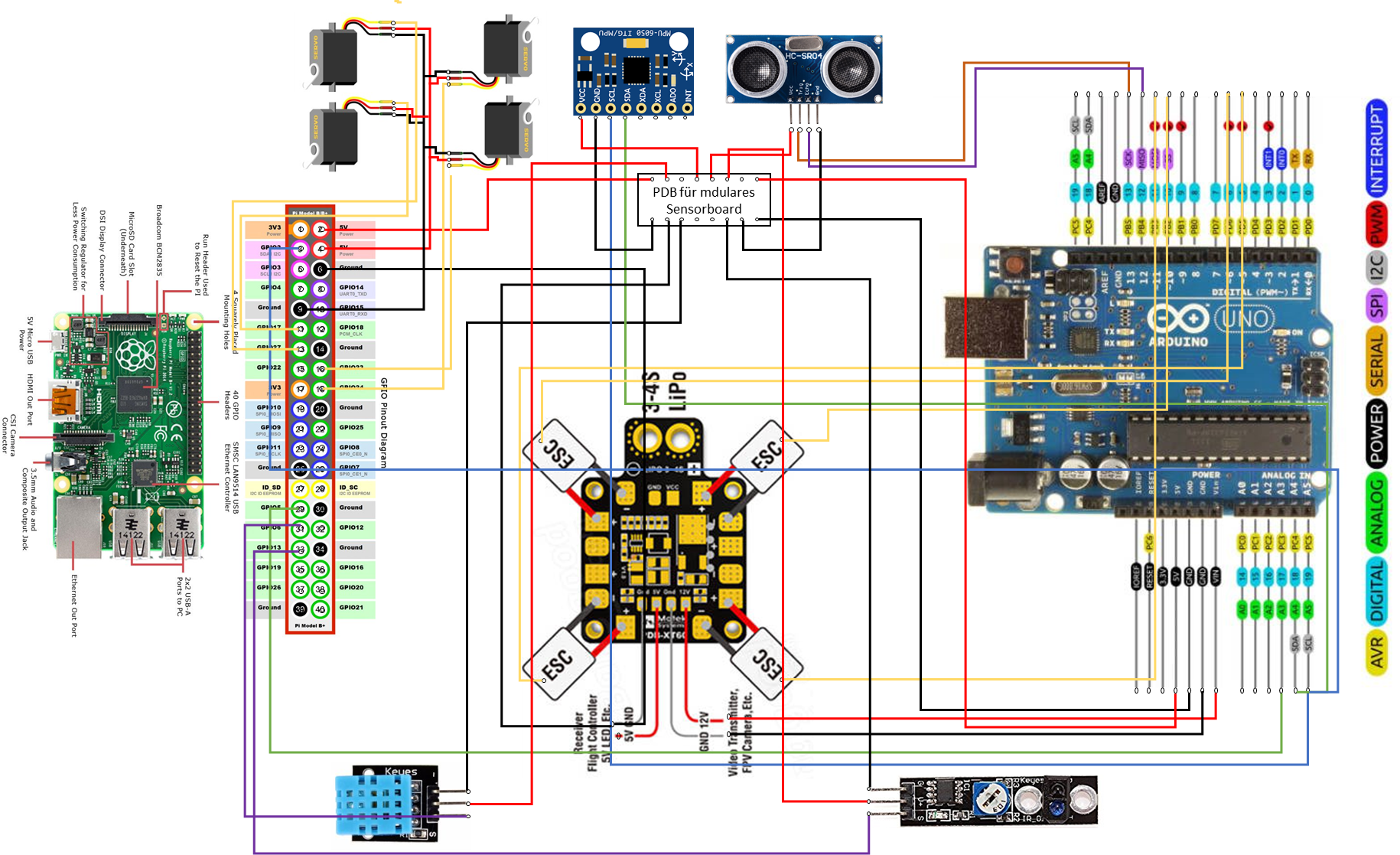
Abbildung 1: Explosionsdarstellung Mohne

CAD-Render Modell:



Abbildung 2: Rendering Bild Mohne

### Elektronik



### Software

# Model Based Analytics

Model Based Analytics is a method of analysis that uses modelling to perform the analysis and capture and communicate the results. For social problems the two main forms of modelling used are [causal loop diagrams](http://www.thwink.org/sustain/glossary/CausalLoopDiagram.htm) and simulation modelling. Causal loop diagrams are used for preliminary conceptual attacks on the problem. This gives way to simulation models for the rest of the work. Thwink.org has selected [system dynamics](http://www.thwink.org/sustain/glossary/SystemDynamics.htm) as the most appropriate simulation modelling tool, due to its simplicity, power, and emphasis on [feedback loops](http://www.thwink.org/sustain/glossary/FeedbackLoop.htm).[[1]](#footnote-1)

Der Grundgedanke von Model Based Analytics ist die virtuelle bzw. digitale Darstellung eines physischen Modells. Dieses Modell wird in der digitalen Welt analysiert und interpretiert. Dies ist eine Methode gewisse Problemstellungen virtuell darzustellen und diese als Simulation zu verwenden.

## Digitaler Zwilling

Das Konzept eines digitalen Zwillings besteht aus drei Elementen: einem physischen Zwilling in der realen Welt, einem digitalen Zwilling im virtuellen Raum, der sowohl alle alten betrieblichen Daten als auch die Echtzeitdaten des physischen Zwillings abbildet, und die Interkonnektivität, die es den Zwillingen ermöglicht, die Informationen auszutauschen. Digitale Zwillingen können für Simulationen verwendet werden und für Einblicke in den arbeitenden Zwilling, um Prozesse zu überwachen und mögliche Fehler zu entdecken.[[2]](#footnote-2)

Im folgenden Abschnitt werden nähere Beschreibungen zum digitalen Zwilling angeführt.

### Allgemein

Ein digitaler Zwilling ist eine virtuelle Modellbildung zum Beispiel eines Prozesses, eines Produkts oder einer Dienstleistung, welches die reale und virtuelle Welt verbindet. Digitale Zwillinge verwenden reale Daten von installierten Sensoren, welche die Arbeitsbedingungen oder Position von Maschinen repräsentieren. Diese Kopplung der virtuellen und realen Welten ermöglicht die Analyse von Daten und die Überwachung von Systemen, um Probleme verstehen und bearbeiten zu können, bevor sie überhaupt auftreten. Ausfallzeiten können frühzeitig vermieden werden und dadurch neue Chancen entwickeln und mithilfe von Computersimulationen, damit die Zukunft besser geplant werden kann. Heutzutage werden digitale Zwillinge eine geschäftliche Notwendigkeit, weil der gesamte Lebenszyklus eines Produkts, Prozesses oder Geschäftsmodells abdeckt werden kann und somit die Grundlage für verbundene Produkte und Dienstleistungen bilden.

Für den sogenannten digitalen Zwilling sind folgende Anforderungen notwendig: das abzubildende reale Objekt, den digitalen Zwilling im virtuellen Raum und Informationen, welche die beiden miteinander verbinden. Digitale Zwillinge können dadurch die operative und auch finanzielle Leistungsfähigkeit eines Objekts wie einer Anlage oder einer Dienstleistung verbessern und in Echtzeit dargestellt werden.

In der ersten Phase „Design“ geht es um den Umgang mit komplexen Produktanforderungen, schnellen Entwicklungszyklen und strengen regulatorischen Anforderungen. In der zweiten Phase „Erstellung (manufacturing)“ kann der Zwilling helfen, bessere Effizienz, Qualität und höheren Ertrag in der Erstellung zu erreichen. In der dritten Phase „Nutzung (operate)" kann ein digitaler Zwilling u. A. eingesetzt werden, um die Verfügbarkeit von Objekten (z. B. Maschinen) zu verbessern. In der vierten Phase „Wiederverwertung (Recycling)“ kann der ein digitaler Zwilling z.B. für die Ersatzplanung oder der Eruierung von Upcycling-Potenzialen eingesetzt werden.

### Projektbezogen

Unser PlanB des Projektes Data Explorer wurde am Tag zu einem Auto umgebaut und deren Funktionen erweitert bzw. ergänzt. Am Auto herrscht ein mechanisches Kräftemodell, welches im folgenden Kapitel dargestellt wird. Dadurch haben wir entschieden, dass wir die Kraft des Motors mit der Beschleunigung als Digitalen Zwilling modellieren bzw. darstellen wollen. Interessant an der Darstellung wird, wie sich die Beschleunigung „a“ während der Datenerfassung verhält. Die Erfassung der Daten erfolgt mithilfe der Matlab Software und dessen Mobile App. Durch die Kommunikation zwischen den beiden Anwendungen kann eine Erfassung der Daten mit verschiedenen Sensoren wie Beschleunigungssensor erfasst werden. Es kann analysiert werden, ob bei der Absolvierung des vorgegebenen Parcours eine Verminderung der Beschleunigung eintritt, wenn ein Hindernis kommt oder eine Kurve nach links oder rechts gemacht wird. Der digitale Zwilling wird in einem Plot der Matlab Software dargestellt, damit das Diagramm entsprechend erläutert und interpretiert werden kann.

### Material und Methoden

Für die Erfassung des digitalen Zwillings unseres Sommerprojektes haben wir uns entschieden, dass wir die Anwendung der Matlab App verwenden werden, da wir kein Logfile über den Raspberry Pi erstellen. Für die Anwendung ist die Matlab Software und der Download der dazugehörigen Matlab Mobile App notwendig. Zu Beginn wird die Matlab Mobile App am Smartphone installiert, egal ob Android oder iOS Betriebssystem. Am Smartphone können die unterschiedlichsten Sensoren wie Beschleunigungs-, Magnetfeld-, Positions-, Winkelgeschwindigkeits- und Orientierungssensor aktiviert werden. In der Software ist eine Installation eines Add-Ons zu erledigen. Um eine Verbindung zwischen der Software und der App herzustellen, wird in der Kommandozeile der Software der Konnektor aktiviert. Es wird nun die funktionierende IP-Adresse angezeigt, welche im Smartphone eingegeben wird, um eine Kommunikation herstellen zu können. Als nächsten Schritt wird der Link von PC und Smartphone erzeugt und es können sowohl am Smartphone und in der Kommandozeile durch bestimmte Matlabbefehle die Sensoren aktiviert werden. Für eine korrekte Datenerfassung werden durch Befehle in der Kommandozeile die Datenaufnahme der Messwerte aktiviert und auch wieder beendet. Mit den erfassten Sensordaten des Smartphones können die Daten auf verschiedensten Arten analysiert und auch dargestellt werden. Die Daten können in einem Diagramm inklusiver Legende in Matlab dargestellt werden.

### Mechanisches Modell des Fahrzeuges

Im folgenden Bild wird das mechanische Modell unseres Fahrzeuges abgebildet. Das Kräftemodell des Fahrzeuges besteht aus der Reibungskraft, die Erdanziehungskraft, die Kraft des Motors in der Vorwärtsbewegung und die Kraft des Autos.

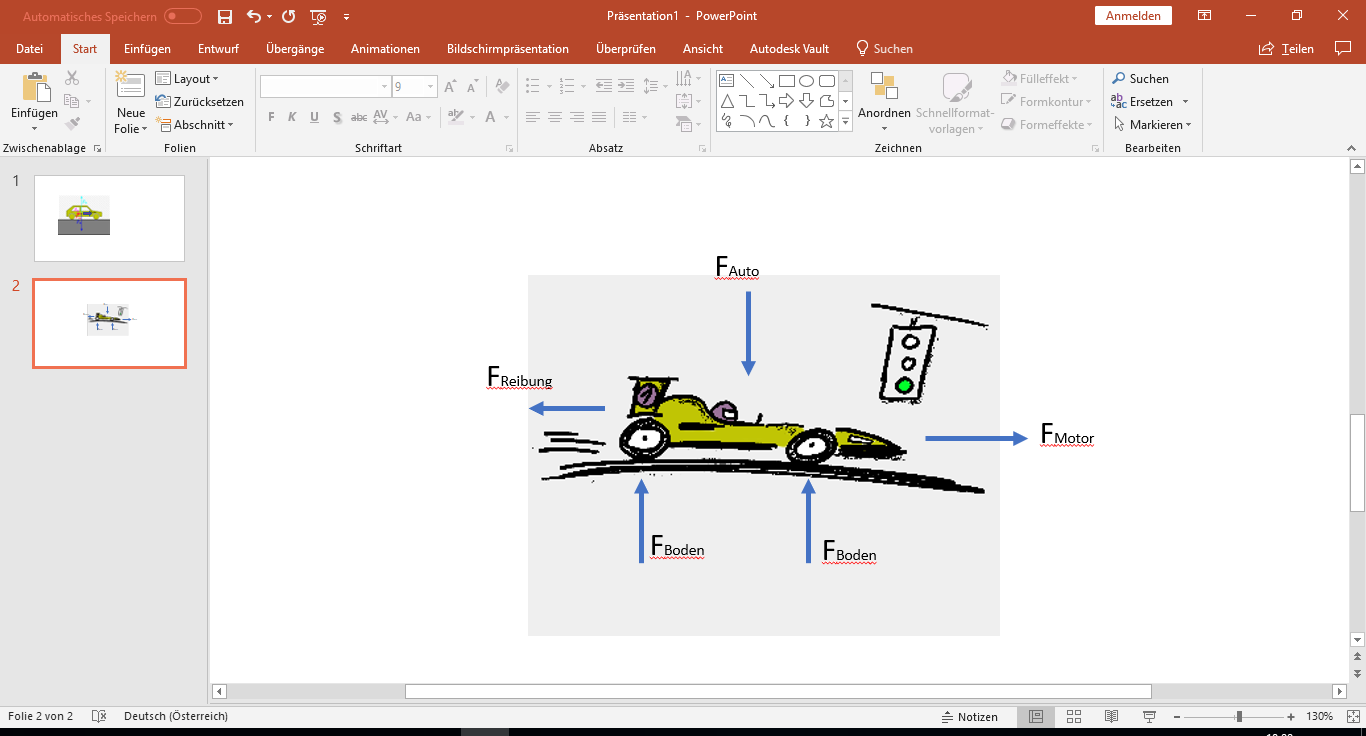


Abbildung 3: Mechanisches Kräftemodell des Autos

### Ergebnisse

Aus der Datenerfassung der Matlab App konnte die Kraft des Motors erfasst und gemessen werden. Die Daten wurden während unseres autonomen Fahrens, welches leider in der kurzen Zeit nicht möglich war fertig zu programmieren, aufgenommen. Anhand der Daten, die auf die Software von Matlab übertragen wurden kann herausgelesen werden, dass unser Programm eine relativ kontinuierliche Geschwindigkeit durchfährt. Daraus resultiert sich, dass unser Fahrzeug, wegen den nicht fertigen ferngesteuertem Programm, nicht immer Objekte bzw. Hindernisse erkennen kann und somit sich nicht mit einer kontinuierlichen Geschwindigkeit fortbewegt.

Mit der Darstellung der Beschleunigung ist keine kontinuierliche Beschleunigung ersichtlich. Der programmierte Code enthält leider keine konkrete Struktur bzw. Abfolge und somit resultiert sich ein relativ hektisches Beschleunigungsdiagramm. In der Mitte des Diagramms kann man erkennen, dass das Fahrzeug einen Stillstand hat und womöglich ein Hindernis erkennen konnte.

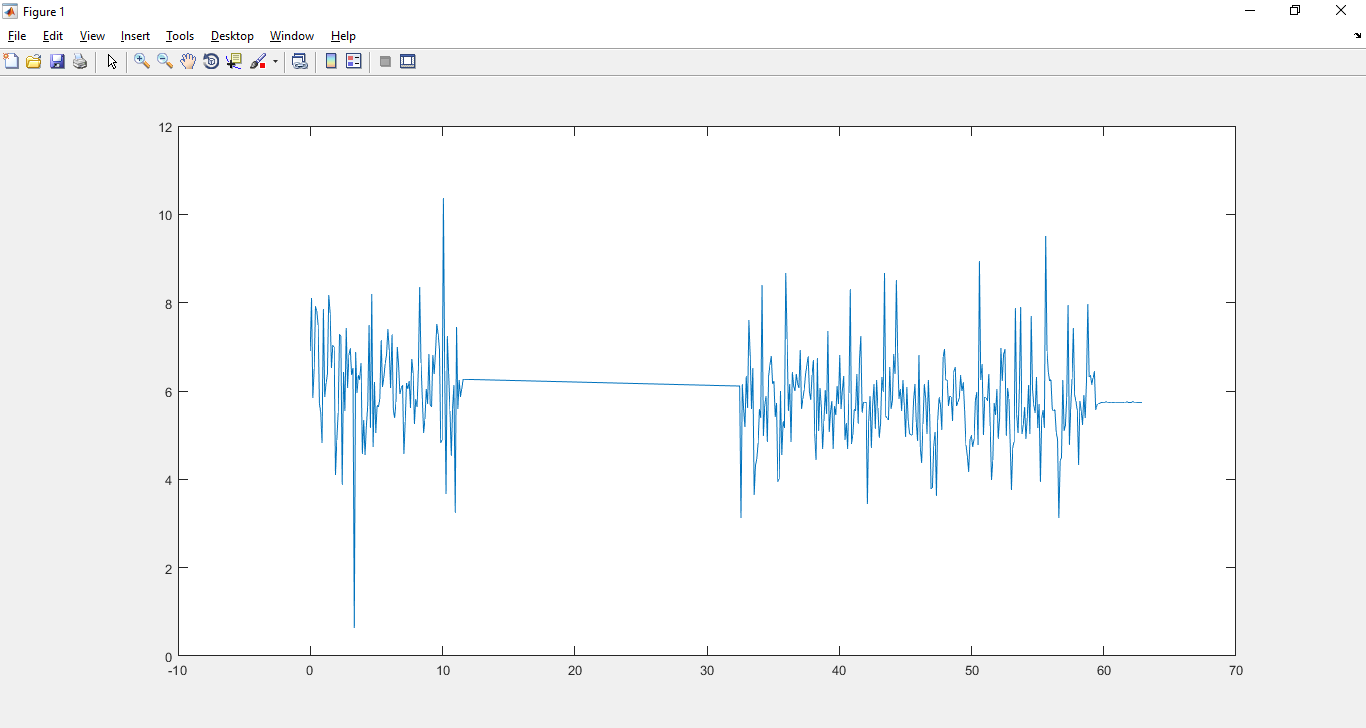


Abbildung 4: Diagramm für die Beschleunigung

Im folgenden Plot von Matlab ist die Winkelgeschwindigkeit dargestellt. Auf dieser Darstellung erkennt man wie sich das Fahrzeug in den Richtungen verändert. Wie schon vorher erläutert haben wir den Fernsteuerungsmodus leider nicht bis zum Ende programmieren können. Im Diagramm ist schön zu sehen, wie oft und schnell das Fahrzeug die Richtung ändert. Das ist der Grund, dass das Programm nicht in der Zeit fertig geworden ist. Deswegen fährt das Fahrzeug willkürlich durch den Raum und in der Gerade des Diagramm hat es an gehalten, weil es womöglich ein Hindernis erkannt hat. Ursprünglich ist es gedacht, dass das Fahrzeug einem Hindernis ausweicht, jedoch nach dem Diagramm sind auf unserem vorgegebenen Parcour sehr viele Hindernisse platziert worden.

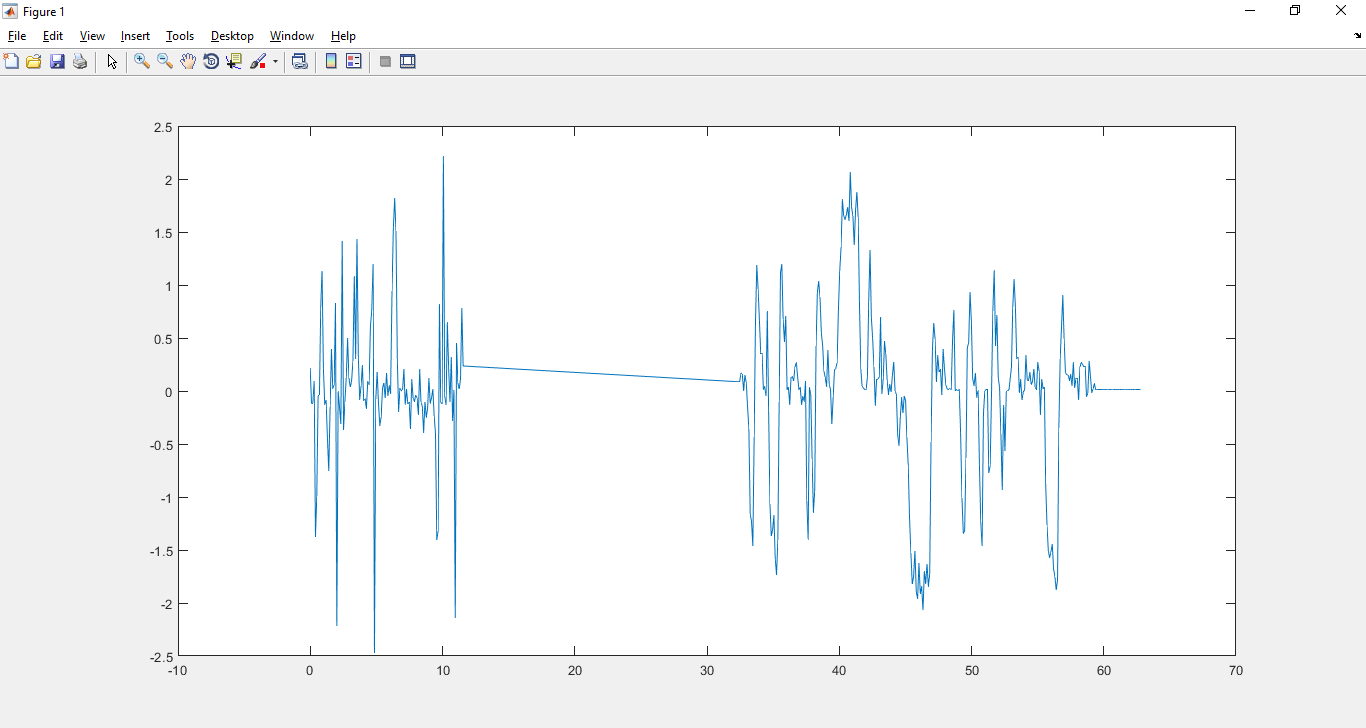


Abbildung 5: Diagramm für die Winkelgeschwindigkeit

### Konklusion

Am Beispiel der beiden Diagramme kann man erkennen, dass unser programmierter Quellcode leider nicht dem Fernsteuerungsmodus entspricht. Das Fahrzeug kann durch die Bilderkennung nicht kontinuierlich die Hindernisse erkennen und daraufhin die Geschwindigkeit reduzieren bzw. nach dem Stillstand wieder beschleunigen. In der Mitte des Diagramms ist ein Stillstand zu erkennen, weil hier womöglich ein Hindernis erkannt wurde. In beiden Abbildungen ist die Beschleunigung sowohl die Winkelgeschwindigkeit sehr hektisch, da das Programm aus zeitlichen Gründen nicht fertig geschrieben werden konnte.

# Simulation

## Pi Simulation

## Arduino Simulation

# Datenübertragung

## Thinkspeak

## Auswertung

# Rohdaten/Datalog

## Timestamp+Daten

# Zusammenfassung

1. Vgl. (Harich, 2014) [↑](#footnote-ref-1)
2. Vgl. (Hewlett Packard Enterprise Development LP, 2018) [↑](#footnote-ref-2)