

# **Spezialisierung einer Unfallerkennung am Zweirad mittels Smartphone auf Unabhängigkeit der Trageposition**

**Master-Thesis**

des Studiengangs Medizintechnik an der  
Universität Stuttgart

**Cand. M.Sc. Oays Darwish**

**3480821**

**Prüfer:** Dr. Thomas Günther  
**Betreuer:** Dipl.- Ing. Nino Häberlen  
**Abgabedatum:** 15.12.2022





## Masterarbeit

# Spezialisierung einer Unfallerkennung am Zweirad mittels Smartphone auf Unabhängigkeit der Trageposition

## Unternehmensbeschreibung

Bosch.IO zählt weltweit über 800 Beschäftigte aus den Bereichen Digitalisierung und IoT und arbeitet mit 30.000 Fachleuten bei ganz Bosch zusammen. Bosch baut und liefert jährlich über eine Milliarde Geräte („Dinge“) an Kunden aus den unterschiedlichsten Branchen.

Bosch.IO GmbH  
Postfach 30 02 40  
70442 Stuttgart  
GERMANY  
Besucher:  
Grönerstraße 5/1  
71636 Ludwigsburg  
Telefon 0711 811-0  
[www.bosch.io](http://www.bosch.io)

18. März 2022

## Aufgabenbeschreibung

Für die breite Anwendung einer Smartphone-basierten Unfallerkennung (u.a. Beschleunigungs-, GPS-Sensor) sollen verschiedene Trage- bzw. Transportpositionen auf ihre Eignung zur Detektion von Unfällen untersucht werden.

- Analyse Unfallarten beim Motorrad aus der Unfallstatistik
- Korrelation Unfallarten zu Verletzungsschwere
- Analyse der Übertragbarkeit statistischer Daten auf bestehenden Messdatensatz
- Analyse der Merkmale verschiedener Positionen: Worin unterscheiden sich verschiedene Positionen voneinander?
- Einfluss verschiedener Messorte auf die Qualität der Erkennung von Unfallklassen
- Vergleich Messung mit Smartphone am Fahrer (z.B. Jackentasche) vs. Smartphone am Fahrzeug (z.B. Tankrucksack)
- Spezialisierung der Help Connect-Unfallerkennung zu einer robusten Multipositions-Unfallerkennung

Das Ziel ist es, eine Unfallerkennung zu entwickeln, die weitgehend unabhängig von Trage- bzw. Messpositionen an Fahrer und Zweirad funktioniert.

## Kontakt

Nino Haeberlen  
Bosch.IO GmbH, IOB/PAC2  
Telefon +49 173 1756137  
[Nino.Haeberlen@bosch.io](mailto:Nino.Haeberlen@bosch.io)



# **Kurzzusammenfassung**

DE

# **Abstract**

Englisch



# **Danksagung**

An erster Stelle möchte ich meinen Eltern, meiner Frau und Familie danken, die mir mein Studium durch ihre Unterstützung ermöglicht haben und stets ein offenes Ohr für mich hatten. Ein besonderer Dank gilt Herrn Nino Häberlen für seine fachliche Unterstützung. Herrn Dr. Thomas Günter möchte ich dafür danken, dass er die Arbeit betreut hat. Darüber hinaus gilt mein Dank allen Personen, die beim Korrekturlesen meiner Abschlussarbeit tätig waren. Abschließend möchte ich mich bei all denjenigen bedanken, die mich während der Anfertigung dieser Arbeit unterstützt und motiviert haben.

Stuttgart, im November 2022

Oays Darwish



# Inhaltsverzeichnis

<b>Kurzzusammenfassung</b>	<b>v</b>
<b>Abstract</b>	<b>v</b>
<b>Danksagung</b>	<b>vii</b>
<b>Abbildungsverzeichnis</b>	<b>xi</b>
<b>Tabellenverzeichnis</b>	<b>xiii</b>
<b>Abkürzungsverzeichnis</b>	<b>xv</b>
<b>Symbolverzeichnis</b>	<b>xvii</b>
<b>1 Einleitung</b>	<b>1</b>
<b>2 Grundlagen</b>	<b>3</b>
2.1 Unfall . . . . .	3
2.1.1 Deffinition . . . . .	3
2.1.2 Zeitliche Phasen eines Unfalls . . . . .	4
2.1.3 Mechanik und Biomechanik des Unfalls . . . . .	5
2.1.4 Statistische Zahlen über Unfälle . . . . .	9
2.2 Technische Grundlagen . . . . .	14
2.2.1 Senesoren und Signale der Smartphones . . . . .	14
2.2.2 Matlab/Simulink . . . . .	16
2.3 Mathematische Grundlagen . . . . .	18
2.3.1 FFT . . . . .	18
2.4 Methoden der Softwareentwicklung . . . . .	19
2.5 Unfallerkennungsalgorithmus . . . . .	20
2.5.1 Kalibrierung . . . . .	20
2.5.2 Übersicht der bereits erkennbaren Unfälle . . . . .	21
2.5.3 Erklärung TipOver . . . . .	21
2.5.4 Erklärung GroundHit . . . . .	21
2.5.5 Erklärung CollisionHit . . . . .	25
<b>3 Unfallerkennung im Pocket-Mode</b>	<b>27</b>
3.1 Kritische Unfallszenarien . . . . .	27

3.2 Lauferkennung . . . . .	27
3.2.1 Peaks aufzählen . . . . .	27
3.2.2 Frequenzbasierend . . . . .	30
3.3 Verifikation des Algorithmus' . . . . .	36
3.3.1 Groundtruth sammeln . . . . .	36
<b>4 Ergebnisse</b>	<b>37</b>
<b>5 Ausblick</b>	<b>39</b>
<b>Literaturverzeichnis</b>	<b>xix</b>
<b>A Anhang</b>	<b>xxi</b>
<b>Eidesstattliche Erklärung</b>	<b>xxiii</b>

# Abbildungsverzeichnis

2.1	Einfache Darstellung des Regelkreises “Fahrer-Fahrzeug-Umfeld”[11] . . . . .	4
2.2	Beispiel der zeitlichen Phasen einer kritischen Fahrsituation [11] . . . . .	5
2.3	Einleiten der Kurvenfahrt (Abb. 3-1) . . . . .	6
2.4	Vereinfachte Darstellung der Kräfte bei stationärer Kurvenfahrt (Abb. 3-2) . . . . .	7
2.5	Verschiedene Kurventechniken bei gleicher Geschwindigkeit (Abb. 3-6) . .	8
2.6	Geschwindigkeitsüberschreitungen nach PKW- und Motorradfahrer an 7 Messstationen aus der VSS-Datenbank [7] . . . . .	10
2.7	Unfallschwere über Geschwindigkeit aus der GAIDA-Datenbank[7] . . . . .	11
2.8	Anzahl der Unfälle gegenüber Unfallgegner . . . . .	12
2.9	Anteil der aktiven sowie passiven Unfälle . . . . .	13
2.10	Anzahl der Unfälle über der Art der Ursache . . . . .	14
2.11	Beispiel von einem FFT (Zeitbereich und Frequenzbereich)[19] . . . . .	18
2.12	Achsenrichtung in Sensorframe sowie in Bikeframe [23] . . . . .	21
2.13	Kalibrierungsmodul - Nachkaklibrierung deaktiviert . . . . .	22
2.14	Testfahrt ohne GroundHit - Full View . . . . .	23
2.15	Testfahrt ohne GroundHit - auf die Energie gezoomt . . . . .	23
2.16	Crash mit GroundHit - ID 2806 - Full View . . . . .	24
2.17	Crash mit GroundHit - ID 2806 - auf die Energie gezoomt . . . . .	25
3.1	Testbeispiel - Lauferkennung - Peaks Zähler . . . . .	28
3.2	Testbeispiel - Lauferkennung - Sinussignalgenerator - Spezifikationen .	29
3.3	Testbeispiel - Lauferkennung - Sinussignal . . . . .	30
3.4	Testbeispiel - Frequenzbasierte Lauferkennung - Sinussignal . . . . .	31
3.5	Testbeispiel - Frequenzbasierte Lauferkennung - Ausgabe des Spektrum- Analyzer und seine Spezifikationen . . . . .	31
3.6	Testbeispiel - Frequenzbasierte Lauferkennung - Ausgabe des Spektrum- Analyzer im Bereich zwischen 0-0.5 kHz . . . . .	32
3.7	Testbeispiel - Frequenzbasierte Lauferkennung - Ausgabe des Spektrum- Analyzer mit anderen Spezifikationen . . . . .	32
3.8	Testbeispiel - Frequenzbasierte Lauferkennung - FFT - Modell . . . . .	33
3.9	Testbeispiel - Frequenzbasierte Lauferkennung - FFT - Modell . . . . .	33
3.10	Entscheidungsskript durch mehrere Kriterien (Frequenz und Geschwin- digkeit) . . . . .	34
3.11	Testbeispiel - Frequenzbasierte Lauferkennung - Ausgangsbeispiel - ID 2488 . . . . .	35

3.12 Testbeispiel - Frequenzbasierte Lauferkennung - Ausgangsbeispiel - ID 2488 - Scope . . . . .	36
--	----

# **Tabellenverzeichnis**



# **Abkürzungsverzeichnis**

VSS	Verkehrssicherheitsscreening
GIDAS	German In-Depth Accident Study
MEMS	Mikro Elektronisch Mechanischen Systeme
FP	Frontalpanel (LabVIEW)
BD	Blockdiagramm (LabVIEW)
FFT	Fast-Fourier-Transformation
DFT	Diskrete-Fourier-Transformation



# **Symbolverzeichnis**

$f_s$	Abtastfrequenz
$B_l$	Blocklänge
$f_n$	Bandbreite
$D$	Messdauer
$d_f$	Frequenzauflösung



# 1 Einleitung

Jedes Jahr wird das Leben von etwa 1,3 Millionen Menschen durch einen Verkehrsunfall beendet. Zwischen 20 und 50 Millionen weitere Menschen erleiden nicht-tödliche Verletzungen, wobei viele infolge ihrer Verletzung eine Behinderung erleiden. Verkehrsunfälle verursachen erhebliche wirtschaftliche Verluste für Einzelpersonen, ihre Familien und Nationen insgesamt. Diese Verluste ergeben sich aus den Behandlungskosten sowie aus Produktivitätsverlusten für diejenigen, die durch ihre Verletzungen getötet oder behindert wurden, und für Familienmitglieder, die sich von der Arbeit oder Schule freinehmen müssen, um sich um die Verletzten zu kümmern. Straßenverkehrsunfälle kosten die meisten Länder 3% ihres Bruttoinlandsprodukts. Verletzungen im Straßenverkehr sind die häufigste Todesursache für Kinder und junge Erwachsene im Alter von 5 bis 29 Jahren.[13]

Motorradfahren findet in Deutschland immer mehr Zuwachs. Rund 3,8 Millionen zweirädrige Kraftfahrzeuge waren am 1. Januar 2012 in Deutschland zugelassen. Dies entspricht 7,26 % von allen zugelassenen Kraftfahrzeugen in Deutschland??.

Motorradfahrer stellen im Straßenverkehr eine besonders gefährdete Gruppe an Verkehrsteilnehmern dar. Studien aus dem Jahr 2014 zeigten, dass die Wahrscheinlichkeit in der USA ein Motorradfahrer bei einem Unfall zu sterben, 27-mal höher ist als die eines Autofahrers, und dass Verletzungen sechsmal so wahrscheinlich sind [18].

Verzögerungen bei der Erkennung und Versorgung der an einem Verkehrsunfall beteiligten Personen erhöhen die Schwere der Verletzungen. Die Versorgung von Verletzungen nach einem Unfall ist äußerst zeitkritisch. Verzögerungen von Minuten können über Leben und Tod entscheiden. Die Verbesserung der Versorgung nach einem Unfall erfordert die Sicherstellung des Zugangs zu rechtzeitiger präklinischer Versorgung und die Verbesserung der Qualität sowohl der präklinischen als auch der stationären Versorgung.[13]

Die Reaktionszeit der Rettung spielt eine besonders große Rolle dabei, Leben zu retten. -; Unfallerkennungsalgorithmus;

-; Automatische Unfallerkennung

-; kürzere Reaktionszeit; Genaue Informationen mitgeteilt; .... usw.

-; kurze Versorgungszeit -; Leben retten!!

- Geschwindigkeitsüberschreitung bei Motorrädern viel häufiger als bei Autos.

## *1 Einleitung*

---

Diese Arbeit beschäftigt sich mit der Weiterentwicklung eines Unfallerkennungsalgoritmus mittels Smartphone.

- Pocket-Mode
- Edgecases
- Mögliche Maßnahmen
-

## **2 Grundlagen**

In diesem Kapitel werden Grundlagen vorgestellt, die zum Verständnis dieser Arbeit dienen.

ODER:

In diesem Kapitel wird die Relevanz der vorliegenden Arbeit erörtert.

### **2.1 Unfall**

In diesem Unterkapitel wird ein Unfall definiert, die Ablaufphasen eines Unfalls erläutert sowie eine Unfallstatistik präsentiert.

#### **2.1.1 Deffinition**

Straßenverkehrsunfälle können in der Regel nur unter Berücksichtigung des geschlossenen Regelkreises „Fahrer-Fahrzeug-Umfeld“ erklärt, analysiert und bewertet werden. Denn die Ursachen und Folgen von Unfällen lassen sich fast nie allein auf eine Komponente des Regelkreises zurückführen, sondern sind das Ergebnis des Zusammenspiels dreier Komponenten. Unfälle werden daher fast immer durch eine Kombination von Ursachen (z.B. Blendung entgegenkommender Fahrzeuge und Fußgänger in dunkler Kleidung) und deren Auswirkung auf das Zusammenspiel mehrerer Situationen (z.B. Tragen von Schutzhelmen, Auslösen von Sicherheitsairbags, Aufpralleinwirkung) verursacht. [11]



Abbildung 2.1: Einfache Darstellung des Regelkreises „Fahrer-Fahrzeug-Umfeld“[11]

Die Abbildung 2.1 zeigt einen vereinfachten Regelkreis des Verhaltens zwischen den drei Komponenten (Fahrer-Fahrzeug-Umfeld). In dem Modell wurde die Ablenkung als ein Störgrößenbeispiel an den Fahrer und die nasse Straße als eine Störgröße ans Fahrzeug. Dieses Modell macht es leichter den Ablauf eines Unfalls zu verstehen und anschließend weitere Unfälle zu vermeiden.

Weiterbeschreiben: Durch eine Störung ändert sich das Ergebnis.

### 2.1.2 Zeitliche Phasen eines Unfalls

Nach dem zeitlichen Unfallverlauf werden folgende Unfallphasen unterschieden:

- Pre-Crash-Phase (Einlaufphase): Die Einlaufphase beschreibt den Zeitraum vom Erkennen der kritischen Situation bis zum ersten Kontakt mit dem Hindernis bzw. Unfallgegner.
- Crash (Kollisionsphase): Der Zeitraum vom ersten Kontakt zwischen den Unfallbeteiligten bis zur Lösung. Bei Mehrfachkollisionen werden mehrere Kollisionsphasen auftreten.
- Post-Crash-Phase (Folgephase): Die Folgephase ist der Zeitraum vom Lösen der Unfallbeteiligten bis zu ihrem Stillstand oder bis zu einem nachfolgenden Zusammenstoß. Bei Mehrfachkollision treten auch mehrere Post-Crash-Phasen auf.

Die Einlaufphase (Pre-Crash-Phase) ist maßgeblich vom Fahrer, der Straßenumgebung und der aktiven Sicherheit vom Fahrzeug abhängig (z.B. Bremsverhalten, Fahrzeuggeladung, gefährliche Kreuzungen, ...usw.).

Die Folgen der Kollisionsphase werden für die betroffenen Verkehrsteilnehmer maßgeblich durch die Maßnahmen der passiven Sicherheit (z.B. Lederkleidung beim Motorradfahrer) beeinflusst. Der Ablauf der Folgephase hängt stark von den verschiedenen Parametern beim Fahrzeug, beim Insassen und bei der Umgebung (z.B. Schnelligkeit

der Rettungskräfte) abhängig.[11]

### Beispiel

Die Abbildung 2.2 zeigt für die Einlaufphase ein vereinfachtes Szenario einer kritischen Situation am Beispiel einer Kurve. Diese kritische Situation kann, muss aber nicht zwangsläufig zu einer möglichen Kollision führen. Zu einem bestimmten Zeitpunkt erkennt der Fahrer eine kritische Situation. Es ist zuerst unabhängig, ob es zu einem Unfall kommt. Nach dem Erkennen dieser Situation hat der Fahrer die Entscheidung, welche Maßnahmen zu greifen sind, um eine Unfall-Situation zu vermeiden. Dabei wird der Fahrer auf bereits vorliegende Erfahrung zugreifen und eine Zur Vermeidung dieser kritischen Situation geeignete Maßnahmen ergreifen. Das Fahrzeug reagiert auf die Fahreraktionen, was zu Fahrer-Fahrzeug-Interaktion führt, die zu Unfällen führen.[11]

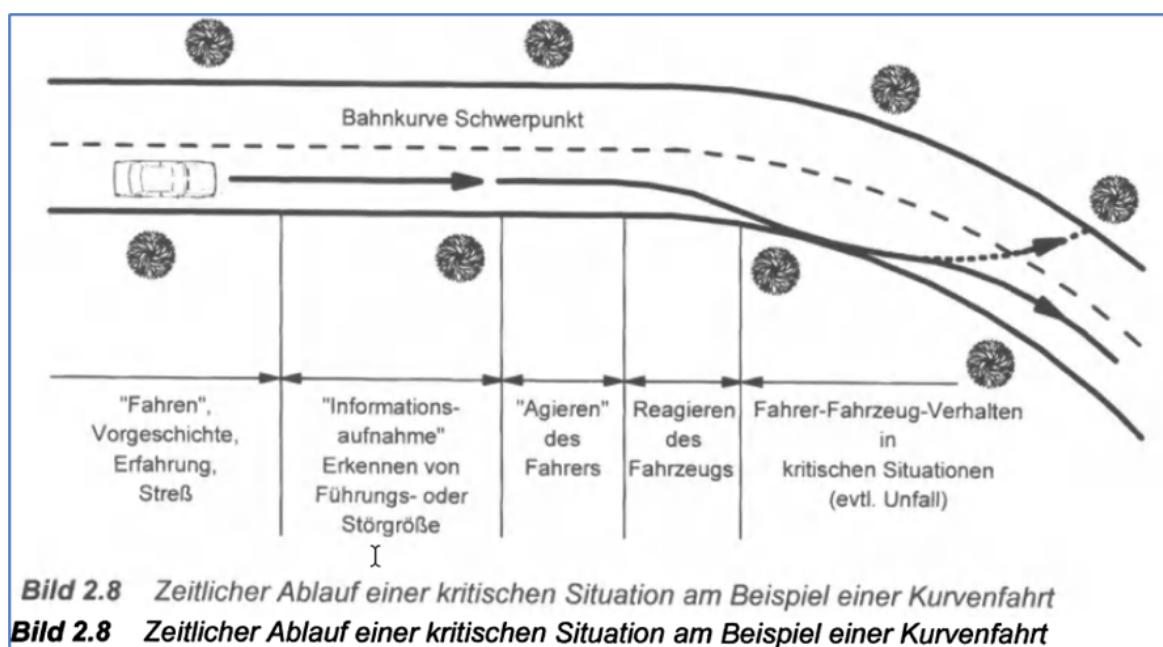


Abbildung 2.2: Beispiel der zeitlichen Phasen einer kritischen Fahrsituation [11]

### 2.1.3 Mechanik und Biomechanik des Unfalls

- Kinematik und Verletzungsbilder
- Zweiradfahrer (Bilder)

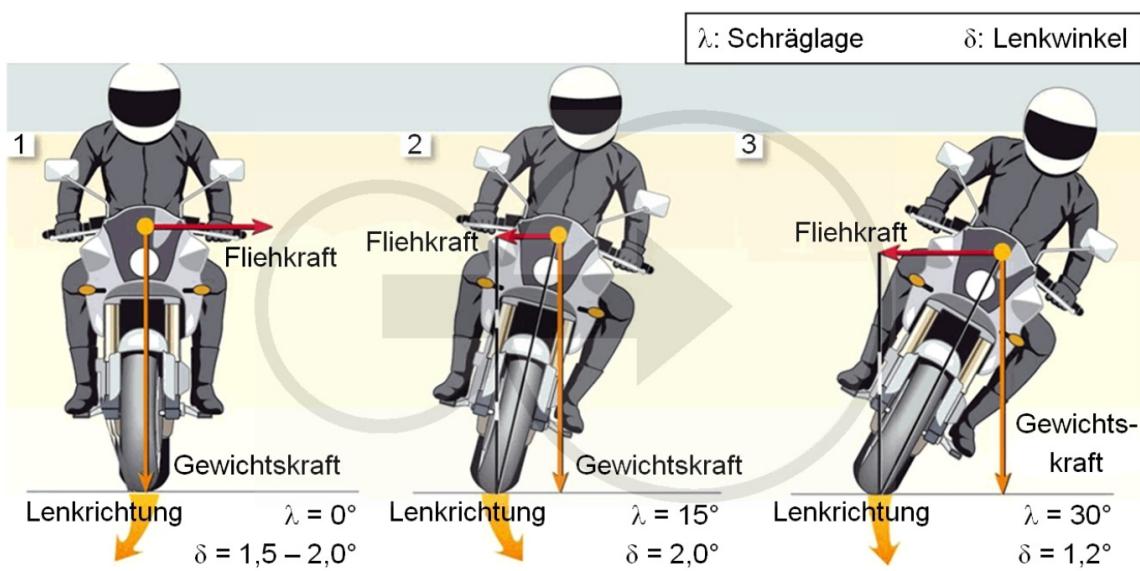
#### Kurvenfahrt

[12] Um das Motorrad durch die Kurve zu bewegen, muss sich der Fahrer mit seiner Maschine in die Kurve legen. Dieser in der Praxis automatisch ablaufende Prozess besteht

im Detail aus einer Reihe einzelner Handlungsschritte. Im Folgenden soll die Kurvenfahrt in ihrer Komplexität näher analysiert werden. Aus Gründen der Vollständigkeit wird zunächst beschrieben, auf welche Weise der Motorradfahrer sein Motorrad bei der Geradeausfahrt im Gleichgewicht halten kann. Hierfür sind zwei Bereiche zu definieren, in denen das Motorrad aus unterschiedlichen Gründen in der Senkrechten bleibt. Ab einer Geschwindigkeit von zirka  $v = 25 \text{ km/h}$  stabilisieren die Kreiselkräfte der Räder und die inneren Kreiselkräfte der rotierenden inneren Bauteile (wie zum Beispiel der Kurbelwelle) die Maschine. Fährt der Fahrer jedoch langsamer, reichen die Kreiselkräfte nicht aus, um die Maschine im Gleichgewicht zu halten. In einer Reihe von Versuchen konnte gezeigt werden, dass der Fahrer bei Geschwindigkeiten  $v < 25 \text{ km/h}$  im Sekundentakt mithilfe kleinster Lenkausschläge von ein bis zwei Grad nach links und rechts das Motorrad im Gleichgewicht hält.

#### Phasen der Kurvenfahrt:

Das Einleiten einer Kurvenfahrt ist das bewusste Herbeiführen einer Instabilität aus der Gleichgewichtslage. In der Regel lässt sich die Kurvenfahrt auf der Landstraße in verschiedene Phasen einteilen. Zuerst bremst der Fahrer die Maschine auf den Einlenkpunkt ab. Ziel einer jeden Kurvenfahrt ist es, exakt im Scheitelpunkt der Kurve die gewünschte Schräglage zu erreichen. Diese komplexe Entscheidung läuft innerhalb weniger Sekundenbruchteile ab. Am zuvor festgelegten Einlenkpunkt macht der Fahrer eine kurze Lenkbewegung in die kurvenabgewandte Richtung, Phase 1 in Abbildung 2.3.



© Ingenieurbüro Morawski + Hugemann, 2012

Abb. 3-1: Einleiten der Kurvenfahrt [IFZ10]

Abbildung 2.3: Einleiten der Kurvenfahrt (Abb. 3-1)

Der erste Lenkimpuls, ähnlich dem Ausbalancieren der Maschine bei Geradeausfahrt

bei niedrigen Geschwindigkeiten, bringt das Motorrad aus dem Gleichgewicht und die Maschine kippt in Richtung Kurvenmittelpunkt (nach links in Abb. 3-1). Verharrt der Fahrer in Phase 1, würde die Maschine schlicht und ergreifend umkippen. Zur Vermeidung des Unfalls reagiert der Fahrer auf die kippende Maschine mit einem sanften Lenkeinschlag in Richtung der Kurve, nach links in Abb. 3-1 (Phase 2). Mit dem gewählten Lenkeinschlag stellt sich im Verlauf der Kurve ein Gleichgewichtszustand ein und die Kurve kann in der gewünschten Schräglage durchfahren werden, Abb. 3-1 Phase 3. Während der gesamten Kurvenfahrt kann der Fahrer den Verlauf der Fahrlinie sowohl durch positives oder negatives Beschleunigen als auch durch die Veränderung des Lenkwinkels beeinflussen, siehe Kapitel 3.4.

### Kräfte am Fahrzeugschwerpunkt bei stationärer Kurvenfahrt:

Befährt ein Motorradfahrer eine Kurve mit konstantem Bahnradius, wirkt aufgrund des sich ständig ändernden Bewegungszustandes eine konstante Querbeschleunigung auf die Einheit Fahrer und Maschine. Die Querbeschleunigung zeigt grundsätzlich zum Momentanpol der durchfahrenen Kurve und berechnet sich wie folgt:

$$a_q = \frac{v^2}{R} = \frac{1}{D} \quad (2.1)$$

Die Querbeschleunigung wirkt auf die träge Masse der Einheit Maschine mit Fahrer und bewirkt eine Seitenkraft im Gesamtschwerpunkt, Abbildung 2.4. Diese Seitenkraft muss von den Reifenseitenkräften, welche in der Fahrbahnebene am vorderen und hinteren Reifenlatsch angreifen, im Gleichgewicht gehalten werden:

$$a_q = \frac{v^2}{R} = \frac{1}{D} \quad (2.2)$$

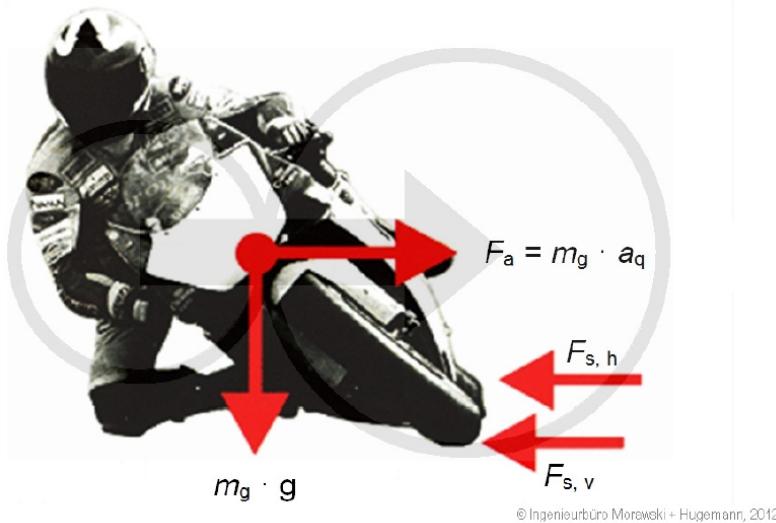


Abb. 3-2: Vereinfachte Darstellung der Kräfte bei stationärer Kurvenfahrt [ECK10a]

Abbildung 2.4: Vereinfachte Darstellung der Kräfte bei stationärer Kurvenfahrt (Abb. 3-2)

Zusätzlich zum Kräftegleichgewicht in horizontaler Richtung muss das Momentengleichgewicht um den Radaufstandspunkt hergestellt werden. Da das Moment der Fliehkraft um den Radaufstandspunkt ständig bestrebt ist, die Maschine zur Kurvenaußenseite zu kippen, würde der Fahrer ohne weiteres Handeln umfallen. Aus diesem Grund durchfahren Einspurfahrzeuge wie das Motorrad Kurven unter Schräglage. Die Schräglage bewirkt eine Verschiebung des Gesamtschwerpunktes von Maschine und Fahrer zur Kurveninnenseite. Aus dieser Verschiebung resultiert ein Moment der Gewichtskraft um den Reifenaufstandspunkt, welches dem Moment der Fliehkraft entgegen wirkt [ECK10a]. Während der stationären Kurvenfahrt stehen die resultierenden Momente aus Gewichtskraft  $F_G$  und Zentrifugalkraft  $F_a$  im ständigen Gleichgewicht. Abb. 3-3 zeigt die Kräfte und die daraus resultierenden Momente während der stationären Kurvenfahrt an einem vereinfachten Modell.

### Kurventechniken:

Die bisherigen Überlegungen und Berechnungen beziehen sich jeweils auf den Fall, dass der Fahrer während der Kurvenfahrt in einer Flucht mit der Maschinenachse bleibt. Tatsächlich gibt es jedoch verschiedene Techniken, eine Kurve zu durchfahren, vgl. (Abbildung 2.5).

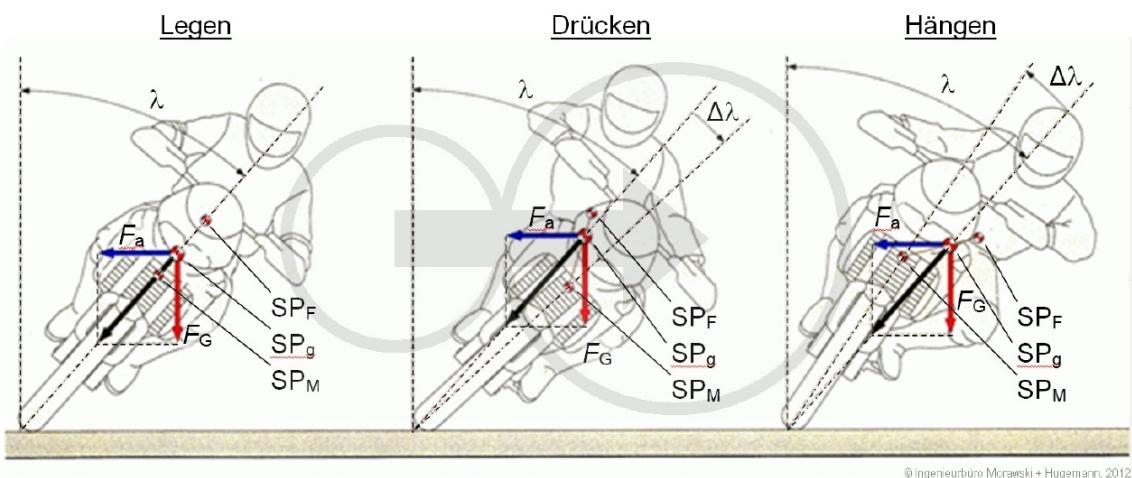


Abb. 3-6: Verschiedene Kurventechniken bei gleicher Geschwindigkeit [COC05]

Abbildung 2.5: Verschiedene Kurventechniken bei gleicher Geschwindigkeit (Abb. 3-6)

Die mit den Modellen der vorangegangenen Abschnitte berechnete Schräglage stimmt mit der Schräglage des Fahrstils „Legen“ überein. Hierbei liegen der Schwerpunkt der Maschine  $SP_M$  und der des Fahrers  $SP_F$  auf einer Achse, sodass der Gesamtschwerpunkt  $SP_g$  ebenfalls auf dieser Achse liegt, Abb. 3-6 links.

Das „Drücken“ stammt aus dem Geländesport und eignet sich besonders für langsame und enge, unübersichtliche Kurven, die einen blitzartigen Kurswechsel erfordern. In der Mitte der Abb. 3-6 ist das Fahrverhalten eines Fahrers mit diesem Fahrstil gezeigt. Mithilfe von Gl. 3-3 kann gezeigt werden, dass eine Kurve mit einer definierten Geschwindigkeit mit nur einem einzigen Schräglagenwinkel durchfahren werden kann,

wobei die Schräglage als gedachte Linie durch den Radaufstandspunkt und den Gesamtschwerpunkt aus Fahrer und Maschine SPg definiert ist. Die Schräglage durch den Gesamtschwerpunkt ist die gleiche, wie bei dem Fahrstil „Legen“. Vergleicht man die Schräglagen durch den Schwerpunkt der beiden Motorräder SPM, so erreicht man durch das „Drücken“ eine deutlich größere Schräglage, als durch das in die Kurve „Legen“. Der Fahrstil „Drücken“ bietet verglichen mit dem Fahrstil „Legen“ keinen Zugewinn an Grip auf der Straße. Es ist durchaus vorstellbar, dass mit zunehmender Schräglage die Reifenaufstandsfläche, auch Reifenlatsch genannt, abnimmt.

Fahrer von Renn- und Supersport-Maschinen bedienen sich der Tatsache des sich verändernden Reifenlatsches und „Hängen“ sich während der Kurvenfahrt von der Maschine. Der Schwerpunkt des Fahrers SPF liegt unterhalb des Schwerpunktes der Maschine SPM, so dass das Motorrad die Kurve mit deutlich geringerer Schräglage durchfahren kann, als beim „Legen“, Abb. 3-6. Bezogen auf den Gesamtschwerpunkt wirken bei gleichem Bahnradius und gleicher Geschwindigkeit die identischen Zentrifugal- und Gewichtskräfte, wodurch der Schräglagenwinkel durch Radaufstandspunkt und Gesamtschwerpunkt auch derselbe ist. Durch die größere Radaufstandsfläche können jedoch wesentlich höhere Seitenkräfte übertragen werden.

## 2.1.4 Statistische Zahlen über Unfälle

Das baden-württembergische Verkehrsministerium stellt ein Portal zur Verfügung, über das einzelne Verkehrsmessstellen abgefragt werden können. Die Messstationen wurden nach zwei Kriterien ausgewählt. Einerseits müssen Unfallschwerpunkte in unmittelbarer Nähe zu Messstationen sein, um zuverlässige Aussagen über Verkehr und Störstellen zu treffen. Andererseits muss darauf geachtet werden, dass es keine Abzweigungen zwischen der Messstation und dem Unfallschwerpunkt gibt, da sonst das richtige Verkehrsaufkommen nicht erfasst werden kann.

Die Abbildung 2.6 stellt dar, wie oft die Geschwindigkeit von Motorradfahrer sowie von PKW-Fahrer an verschiedenen Stationen überschritten wurde. Aus der Abbildung 2.6 ist deutlich zu erkennen, dass an sechs der sieben Stationen das Geschwindigkeitslimit regelmäßig überschritten wird. Die Motorradfahrer missachten die Geschwindigkeitsbegrenzung häufiger als die PKW-Fahrer. [7]

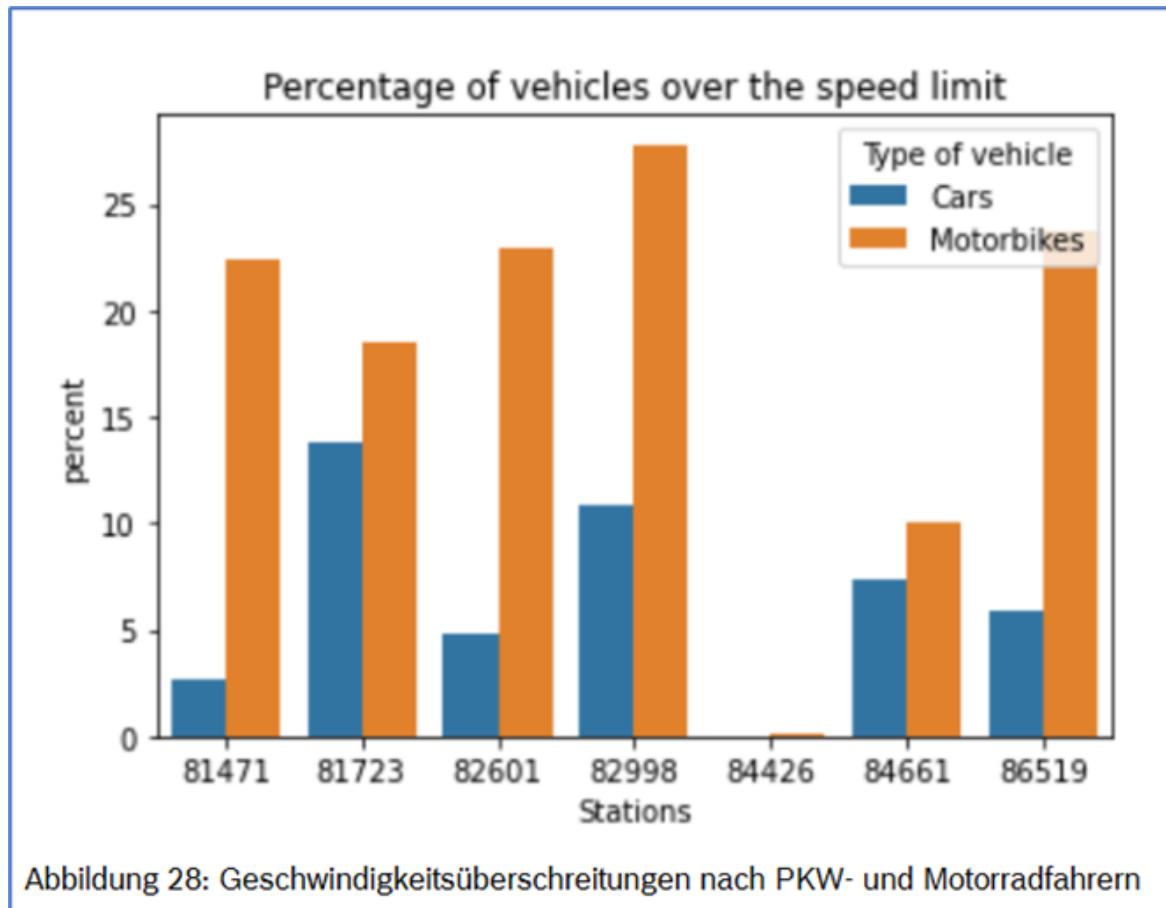
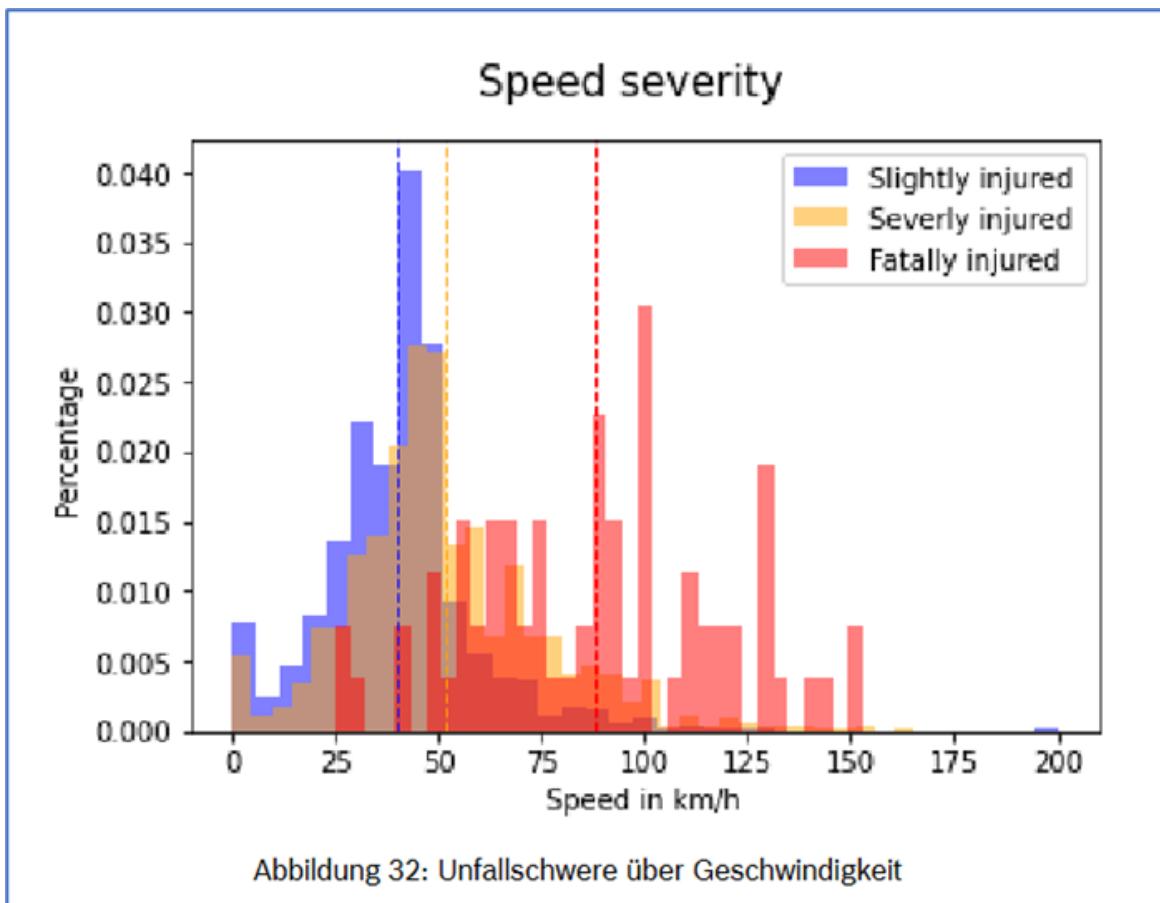


Abbildung 2.6: *Geschwindigkeitsüberschreitungen nach PKW- und Motorradfahrer an 7 Messstationen aus der VSS-Datenbank [7]*

Die GIDAS-Daten verfügen über die Ausgangsgeschwindigkeit des Motorradfahrers, was die Ermittlung des Einflusses dieser Geschwindigkeit auf die Unfallfolgen ermöglicht. In der Abbildung 2.7 sind die Unfallschwere nach Motorradgeschwindigkeit als Histogramm dargestellt. Die Unfälle werden dabei nach Unfallschwere (leicht verletzt, schwer verletzt, tödlich verwundet) unterteilt. Die gestrichelten Linien repräsentieren die Mittelwerte der Histogramme. Die Grafik zeigt, dass die Verletzungsschwere stark von der Geschwindigkeit abhängig ist. Bei einer 0km/h gibt es keine tödliche Unfälle, wobei die Unfälle bei einer Geschwindigkeit von 100 fast immer mit einer schweren Verletzung oder tödlichen Verkehrsteilnehmer enden. Die hohe Geschwindigkeit kommt immer mit hohen Kräften zusammen, welche bei einem Unfall dem Fahrer bewirken. Im Fall eines Motorradfahrers ist das besonders wichtig zu betrachten, da diese Kräfte dem Fahrer direkt übertragen werden. [7]

Abbildung 2.7: *Unfallschwere über Geschwindigkeit aus der GAIDA-Datenbank[7]*

## Youtube-Statistik

Wie oft kommt welcher Unfall vor? (analog zu meiner BA)

Im Sinne der Verifizierung des angepassten Unfallerkennungsalgorithmus muss erstmals bekanntgegeben werden, welche Unfälle bzw. Unfallarten am häufigsten vorkommen, damit diese tief betrachtet werden. Dazu wurden mehrere Videos von Motorradunfälle auf dem Plattform "Youtube" stichpunktartig angeschaut und die vorgestellten Unfälle statistisch analysiert. Es wurden insgesamt 32 Unfallsituationen ausgewertet. In der Auswertung wurden die Unfallgegner und den Ablauf des Unfalls betrachtet. [3] [2] [4] [1] [5]

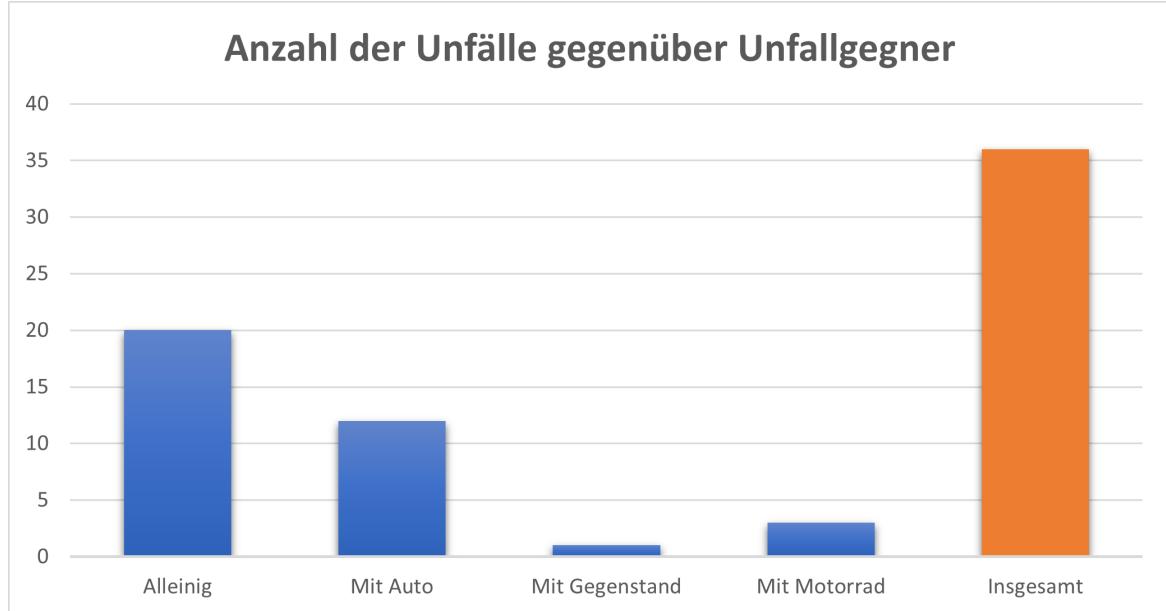


Abbildung 2.8: Anzahl der Unfälle gegenüber Unfallgegner

In der Abbildung 2.8 stellt die Grafik die Anzahl der Motorradunfälle gegenüber der Unfallgegner dar. Es wurde hier zwischen alleiniger Unfall, Unfall mit einem Gegner (Auto, Motorrad) oder Unfall wegen eines Gegenstands unterschieden. Diese Unterscheidung ist wegen der Verhaltensunterschied während eines Unfalls. Von insgesamt 36 Unfälle waren die alleinige Unfälle am meisten gefolgt von den Unfällen mit einem Auto. Die Unfälle mit einem anderen Motorrad oder wegen eines Hindernis sind am wenigsten. Unter alleinige Unfälle werden die zwei Szenarien (An einer Kurve rutsche und Kontrolle verloren) am meisten aufgetreten. Das kann sehr gut in der Abbildung 2.10 sichtbar sein. Das Szenario, in dem das Motorrad an ein Auto von hinten angestoßen hat, hatte die dritte Stelle besetzt. Die Hauptgründe der Unfällen mit einem Gegner waren v.A. die hohe Geschwindigkeit und das schlechte Wetter (z.B. Nässe, Schnee), was zu Schwierigkeiten geführt hat, das Motorrad in kritischen Situationen zu kontrollieren.

Die Abbildung 2.9 zeigt den Anteil der aktiven sowie passiven Unfälle. Wenn das Motorrad angestoßen wird, wird von einem passiven Unfall gesprochen, da der Motorradfahrer keinen Einfluss darauf hat. Im Vergleich dazu könnte er bei einem aktiven Unfall das Ergebnis beeinflussen, z.B. durch langsamer fahren oder mehr Abstand mit den anderen Verkehrsteilnehmer.

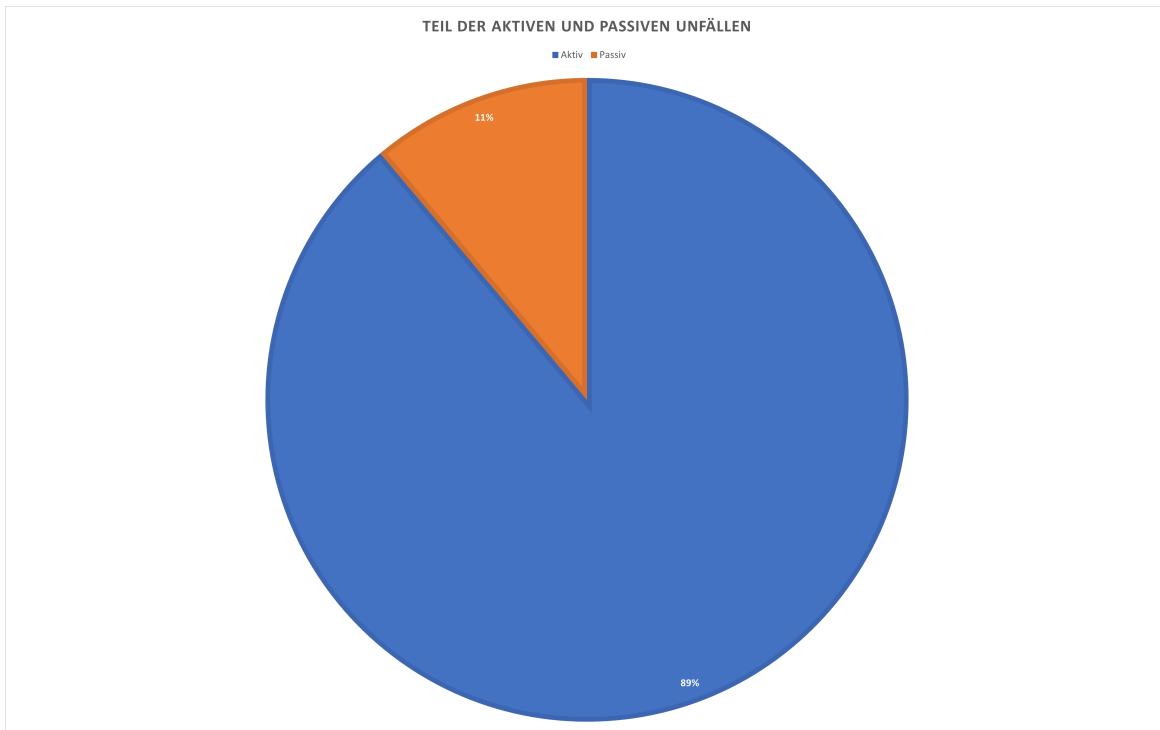


Abbildung 2.9: *Anteil der aktiven sowie passiven Unfälle*

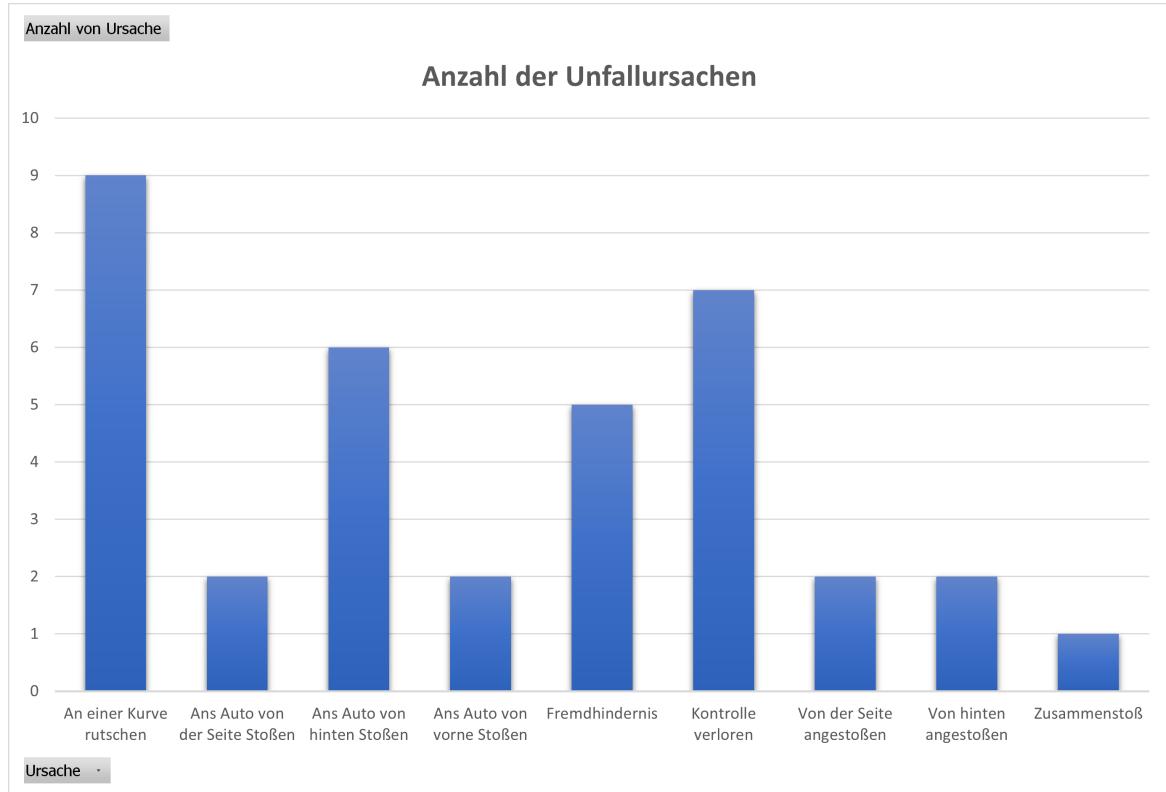


Abbildung 2.10: *Anzahl der Unfälle über der Art der Ursache*

## 2.2 Technische Grundlagen

### 2.2.1 Sensores und Signale der Smartphones

Die meisten Smartphones haben einen Beschleunigungsmesser, und viele enthalten jetzt ein Gyroskop. Je nach Gerät können die softwarebasierten Sensoren ihre Daten entweder vom Beschleunigungs- und Magnetometer oder vom Gyroskop beziehen. Bewegungssensoren sind nützlich zum Überwachen von Gerätebewegungen wie Neigung, Schütteln, Drehung oder Schwingen. Die Bewegung spiegelt normalerweise die direkte Benutzereingabe wider, kann aber auch die physische Umgebung widerspiegeln, in der sich das Gerät befindet (z.B. Das Smartphone bewegt sich mit der Person, die es am Körper hat und sich selbst bewegt).[9]

#### Beschleunigungssensoren

- Aufbau kurz erläutern
- Beispieldaten

Der Beschleunigungssensor ist ein elektromechanisches Gerät, das die Beschleunigungskraft misst, die durch Bewegung, Schwerkraft oder Vibration verursacht wird.

Diese Kräfte können statisch wie Schwerkraft, dynamische Bewegungsempfindungen oder Vibrationen sein. Mathematisch gesehen ist die Beschleunigung ein Maß für die zeitliche Geschwindigkeitsänderung (bzw. Geschwindigkeit geteilt durch die Zeit). Der Beschleunigungssensor im Smartphone misst die lineare Beschleunigung des Geräts. In der Ruheposition stellt die Figur die auf das Gerät wirkende Schwerkraft dar und misst gleichzeitig auch die Beschleunigung auf der X- und Y-Achse, die Null sein wird. Die meisten Smartphones verwenden heutzutage Beschleunigungssensoren, um die Bildschirmanzeige abhängig von der Position auszurichten, in der das Gerät gehalten wird. Mit den eingebauten Beschleunigungssensoren können Benutzer ein besseres Anzeigerlebnis erzielen. [24]

Der Beschleunigungssensor im mobilen Gerät liefert die XYZ-Koordinatenwerte, die zum Messen der Position und der Beschleunigung des Geräts verwendet werden. Die XYZ-Koordinate stellt die Richtung und Position des Geräts dar, an dem eine Beschleunigung aufgetreten ist. Die Drehrichtung und -position werden mit Gyroskopsensoren gemessen. Die vom Gerät bereitgestellten Beschleunigungsmesserwerte enthalten normalerweise auch die Schwerkraft. Das Signal des Beschleunigungssensor wird in die Tief-/Hochpassfilter geleitet, um das Ergebnis basierend auf der verwendeten Anwendung zu verfeinern. [21]

- Wird das Gerät auf die linke Seite geschoben (bewegt sich also nach rechts), ist der x-Beschleunigungswert positiv.
- Wenn das Gerät auf seinen Boden gedrückt wird, ist der y-Beschleunigungswert positiv.
- Wenn das Gerät mit einer Beschleunigung von  $A \text{ m/s}^2$  in den Himmel geschoben wurde, ist der Wert der z-Beschleunigung gleich  $A + 9,81$ , da die Schwerkraft ( $9,81 \text{ m/s}^2$ ) mitberechnet wird.

Im Allgemeinen ist der Beschleunigungssensor ein guter Sensor, wenn Sie die Bewegung des Geräts überwachen. [9]

### Gyroskop

Gyroskop ist ein Gerät, das ein sich schnell drehendes Rad oder einen umlaufenden Lichtstrahl enthält. Gyroskop wird verwendet, um die Abweichung eines Objekts von seiner gewünschten Ausrichtung zu erkennen. Gyroskope werden zur automatischen Lenkung und zur Korrektur der Dreh- und Nickbewegung in Marschflugkörpern und ballistischen Flugkörpern verwendet.[20] Der Gyroskopsensor im MEMS ist winzig (zwischen 1 und 100 Mikrometer, die Größe eines menschlichen Haars). Wenn der Gyro gedreht wird, wird eine kleine Resonanzmasse bei einer Winkelgeschwindigkeitsänderung verschoben. Diese Bewegung wird in elektrische Signale mit sehr geringem Strom umgewandelt, die verstärkt und von einem Host-Mikrocontroller gelesen werden können.[25]

### **Global Positioning System (GPS)**

GPS besteht aus drei Teilen: Satelliten, Bodenstationen und Empfängern. Die Position der Satelliten ist jederzeit bekannt. Die Bodenstationen verwenden Radar, um sicherzustellen, dass die Satelliten sich tatsächlich dort befinden, wo sie sich befinden sollen. Ein Empfänger in dem Smartphone oder im Auto wartet ständig auf ein Signal von diesen Satelliten und findet heraus, wie weit er von einigen von diesen Satelliten entfernt ist. Sobald die Entfernung zwischen einem Empfänger und vier oder mehr Satelliten berechnet wurde, ist genau bekannt, wo der Empfänger sich befindet. Der Basis-GPS-Dienst bietet Benutzern eine Genauigkeit von etwa 7,0 Metern, 95% der Zeit. GPS-Empfänger zeigen die Geschwindigkeit an und berechnen die Geschwindigkeit mithilfe von Algorithmen im Kalman-Filter. Die meisten Empfänger berechnen die Geschwindigkeit durch eine Kombination aus Bewegung pro Zeiteinheit und Berechnung der Dopplerverschiebung in den Pseudoentfernungssignalen von den Satelliten.[17][10][8]

### **2.2.2 Matlab/Simulink**

Matlab ist eine Hochleistungssprache für technisches Rechnen. Matlab integriert Berechnung, Visualisierung und Programmierung in einer benutzerfreundlichen Umgebung, in der Probleme und Lösungen in einer vertrauten mathematischen Notation ausgedrückt werden.

Simulink ist ein grafisches Softwarepaket zur Modellierung, Simulation und Analyse dynamischer Systeme und basiert auf Matlab. Die Software hat sich in den letzten Jahren zum weitesten verbreiteten Softwarepaket in Wissenschaft und Industrie entwickelt. Simulink unterstützt lineare und nichtlineare Systeme, die in kontinuierlicher Zeit, gesampelter Zeit oder einer Mischung aus beiden modelliert sind. Für die Modellierung bietet Simulink eine grafische Benutzeroberfläche (GUI) zum Erstellen von Modellen als Blockdiagramme. Mit dieser Schnittstelle können die gewünschten dynamischen Systeme einfach aufgebaut werden. Mithilfe von Scopes und anderen Anzeigeblocken können die Simulationsergebnisse während der Simulation analysiert werden. Die Simulationsergebnisse können zur Nachbearbeitung und Visualisierung in den MATLAB-Arbeitsbereich gestellt werden. [6][14]

### **Simulink und LabVIEW**

LabVIEW ist eine von "National Instruments" entwickelte Software. Sie wird häufig von Ingenieuren, Wissenschaftlern und Studenten für die Datenerfassung, Instrumentensteuerung und industrielle Automatisierung verwendet. Die LabVIEW-Umgebung besteht aus zwei Hauptkomponenten: Frontpanel (FP) und Blockdiagramm (BD). Ein FP stellt die grafische Benutzeroberfläche bereit, während ein BD die Bausteine eines Systems enthält und einem Flussdiagramm ähnelt. LabVIEW-Systeme werden als virtuelle Instrumente (VIs) bezeichnet und ihr FP erscheint als Instrumententafel, die aus verschiedenen Bedienelementen und Anzeigen besteht.

Ähnlich wie LabVIEW bietet Simulink einen blockbasierten Programmieransatz für die Simulation, den Entwurf und die Analyse dynamischer Systeme. Es bietet eine interaktive grafische Umgebung zusammen mit einer Reihe von Bibliotheken zum Entwerfen und Simulieren von Systemen, einschließlich DSP-Systemen. Simulink-Blöcke werden als Modelle bezeichnet, und im Gegensatz zu LabVIEW werden die Codeimplementierung und Eingabe-/Ausgabeeinheiten in Simulink nicht explizit unterschieden. Simulink ist in MATLAB integriert und kann daher auf die Funktionen und Tools zugreifen, die in der MATLAB-Umgebung verfügbar sind. [16] [15]

Wenn komplexe Simulationen ausgeführt werden sollen oder komplexe Simulationsmodelle von Steuerungen oder Anlagen zu erstellen/debuggen sind, wird Simulink verwendet, da LabVIEW keine effizienten Codegeneratoren für die dynamische Simulation hat. Simulink konzentriert sich hauptsächlich auf Simulation und Modellierung, was bei LabVIEW sicherlich nicht der Fall ist.

Der implementierte Algorithmus enthält einen verbreiteten und komplexen Entscheidungsbaum, welcher sich mit Simulink sowohl übersichtlicher als auch einfacher darstellen lässt als mit LabVIEW.

## **App-Entwicklung**

Die Entwicklung mobiler Apps ist der Prozess zur Erstellung von Software für Smartphones und digitale Assistenten. Die Software kann auf dem Gerät vorinstalliert oder aus einem mobilen App Store heruntergeladen werden. Eine der bekannten Sprachen in der App-Entwicklung ist C. C ist eine leistungsstarke Programmiersprache, mit der Anwendungen in mehreren Bereichen erstellt werden können, von einfachen Taschenrechnern und Apps bis hin zu Videospielen. Sie ist eine Sprache auf niedriger Ebene, dies bietet Geschwindigkeit und eine weitaus bessere Möglichkeit zur Speicherverwaltung.

Für die Generierung eines C-Codes aus Simulink-Modellen wird der in Matlab/Simulink integrierte C/C++-Coder verwendet. \*\*\*\*\*Quelle\*\*\*\*\*

## **C/C++-Coder**

Der C/C++-Code-Generator wird für Rapid Prototyping, Hardware-in-the-Loop-Tests, Simulationsbeschleunigung oder einfach als ausführbare Datei zur Ausführung außerhalb von MATLAB und Simulink verwendet. Diese Codegenerierung ist der Prozess der Generierung von Low-Level-Code direkt aus einer High-Level-Programmiersprache oder Modellierungsumgebung.

Der C/C++-Coder ist ein weiterer Vorteil von Simulink gegenüber LabVIEW und hierfür ist Simulink die richtige Entscheidung für die Implementierung der Software bzw. des Unfallerkennungsalgorithmus'.

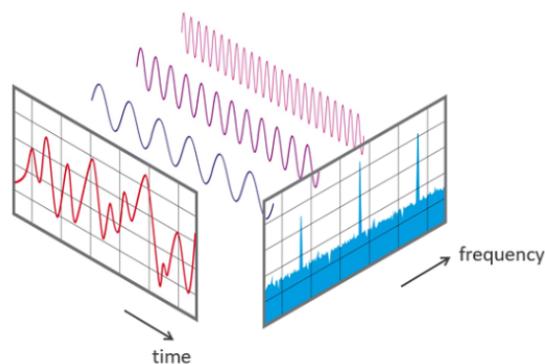
## 2.3 Mathematische Grundlagen

In dieser Arbeit wird eine FFT eingesetzt, für welche das folgende Hintergrundwissen zum Verständnis benötigt wird.

### 2.3.1 FFT

[19][26]

Die „Fast Fourier Transform“ (FFT) ist ein wichtiges Messverfahren und wurde erstmals von Cooley und Tukey 1965 diskutiert, obwohl Gauß den kritischen Faktorisierungsschritt bereits 1805 beschrieben hatte. Dieses Verfahren wandelt ein Signal in einzelne Spektralkomponenten um und liefert dadurch Frequenzinformationen über das Signal. FFT wird zur Fehleranalyse, Qualitätskontrolle und Zustandsüberwachung von Maschinen oder Anlagen eingesetzt. Dieser Abschnitt erläutert die Funktionsweise einer FFT, die relevanten Parameter und deren Auswirkungen auf das Messergebnis. Die FFT ist ein optimierter Algorithmus zur Umsetzung der „Discrete Fourier Transformation“ (DFT). Ein Signal wird über einen Zeitraum abgetastet und in seine Frequenzkomponenten zerlegt. Diese Komponenten sind einzelne sinusförmige Schwingungen mit unterschiedlichen Frequenzen, jede mit ihrer eigenen Amplitude und Phase. Diese Transformation ist an einem Beispiel im folgenden Diagramm (Abbildung 2.11) dargestellt. Über den gemessenen Zeitraum enthält das Signal drei unterschiedliche dominante Frequenzen.



*View of a signal in the time and frequency domain*

Abbildung 2.11: Beispiel von einem FFT (Zeitbereich und Frequenzbereich)[19]

### Schritt für Schritt

Im ersten Schritt wird ein Ausschnitt des Signals abgetastet und zur weiteren Verarbeitung im Speicher abgelegt. Zwei Parameter sind hier relevant:

1. Die Abtastrate bzw. Abtastfrequenz  $f_s$  des Messsystems (z.B. 48 kHz). Dies ist die durchschnittliche Anzahl von Abtastungen, die in einer Sekunde erhalten

werden (Abtastungen pro Sekunde)

2. Die Blocklänge  $B_L$  ist die ausgewählte Anzahl von Proben (Samples). Dies ist immer eine ganzzahlige Potenz zur Basis 2 (z.B.  $2^{10} = 1024$  Samples)

Aus den beiden Grundparametern  $f_s$  und  $B_L$  können weitere Parameter der Messung bestimmt werden.

**Bandbreite:**  $f_n$  (= Nyquist-Frequenz). Dieser Wert gibt die theoretische maximale Frequenz an, die durch die FFT bestimmt werden kann.

$$f_n = \frac{f_s}{2} \quad (2.3)$$

Beispielsweise können bei einer Abtastrate von 100 Hz Frequenzanteile bis 50 Hz bestimmt werden.

**Messdauer:**  $D$  ergibt sich aus der Abtastrate  $f_s$  und der Blocklänge  $B_L$  wie folgt:

$$D = \frac{B_L}{f_s} \quad (2.4)$$

**Frequenzauflösung:**  $d_f$  gibt den Frequenzabstand zwischen zwei Messergebnissen an.

$$d_f = \frac{f_s}{B_L} = \frac{1}{D} \quad (2.5)$$

In der Praxis ist die Abtastrate  $f_s$  meist eine vom System vorgegebene Größe. Durch die Auswahl der Blocklänge  $B_L$  kann jedoch die Messdauer  $D$  und Frequenzauflösung  $d_f$  definiert werden. Es gilt:

- Eine kleine Blocklänge  $B_L$  führt zu schnellen Messwiederholungen mit grober Frequenzauflösung.
- Eine große Blocklänge  $B_L$  führt zu langsameren Messwiederholungen mit feiner Frequenzauflösung.

### Spiegelfrequenzen

Wird die Nyquist-Frequenz (Gleichung 2.3) überschritten, wird das Signal an dieser gedachten Grenze reflektiert und fällt wieder in das Nutzfrequenzband zurück. Diesen unerwünschten Spiegelfrequenzen wird vor der Abtastung mit einem analogen Tiefpassfilter (Anti-Aliasing-Filter) entgegengewirkt. Der Filter sorgt dafür, dass Frequenzen oberhalb der Nyquist-Frequenz unterdrückt werden.

## 2.4 Methoden der Softwareentwicklung

Agile; Sprinte;... usw. Beschreiben je nachdem, wie ich meine Arbeit präsentieren möchte. z.B. Ich hatte eine Idee, Ein Modell gebaut, getestet. Danach habe ich das Problem entdeckt und musste das Modell anpassen. (Sprinte von 2 Wochen jeweils)

## 2.5 Unfallerkennungsalgorithmus

Der aktuelle Unfallerkennungsalgorithmus war eine Doktorarbeit bei Bosch-eBike.

- Der aktuelle Algo geht von der Energie raus. D.h. eine Unfallerkennung erfolgt durch die Berechnung der Energie und der Schwellwert. ((((((Paper von Jan)))))))

Die Modellierung der Merkmale GroundHit und Collision erfordert weitere unabhängige vorverarbeitete Signale. Zu diesem Zweck verwenden wir den ANOVA-Ansatz (Analysis of Variance) (Dalgaard, 2008), um verschiedene statistische Eigenschaften (z. B. Mittelwert, Standardabweichung, Varianz, Extrema und Integral Figo et al., 2010) über verschiedene Fenstergrößen des zu analysieren IMU-Beschleunigungsdaten. Der ANOVA-Ansatz erfordert, dass die vorverarbeiteten Daten normalverteilt sind. Wir verwenden den Anderson-Darling-Test (Scholz und Stephens, 1987), um diese notwendige Bedingung zu überprüfen. Abb. 2 zeigt ein beispielhaftes Histogramm zur Darstellung der Normalverteilung für die mittlere Beschleunigung über eine definierte Fenstergröße  $u$ . Das Ergebnis der ANOVA-Analyse liefert die spezifische Energie als optimalen Indikator unter denen aus den untersuchten statistischen Ansätzen zur Klassifizierung der Kollisions- und Bodentreffer-Ereignisse wie folgt: (\*\*\*\*\*\*) [22]

wobei  $\Delta_{e,xy}$ ,  $u$  die Änderung der massenspezifischen kinetischen Energie darstellt. Es beschreibt ein Ereignis in der xy-Ebene im Fahrradkoordinatensystem (also dem Fahrradrahmen (BF)) während des Zeitfensters  $u$  durch die Integration der Beschleunigungssignale  $a_{Bf,x}$  und  $a_{Bf,y}$ . Da Kollisions- und Bodentrefferereignisse nur in dieser Ebene stattfinden, werden die Auswirkungen in der vertikalen z-Achse auf die Fahrbahnoberfläche oder Sprünge des Fahrradsystems zurückgeführt. Zusätzlich bietet die Varianz  $\sigma_{e,xy}$ ,  $u_G H$  der x- und y-Beschleunigungssignale über ein Zeitfenster  $u_G H$  weitere Trennmöglichkeiten für GroundHit-Ereignisse. [22]

Die Unfallerkennung ist wichtig wegen folgenden Gründen.

- Der grobe Ablaufplan.

### 2.5.1 Kalibrierung

Die Kalibrierung dient dazu, die Ausrichtung des Motorrad zu erkennen, damit die Richtung der Fahrt sowie diesbezügliche Bewegungen (Beschleunigung, Bremsen, Neigung in einer Kurve) richtig erkennen und diese gut auswerten.

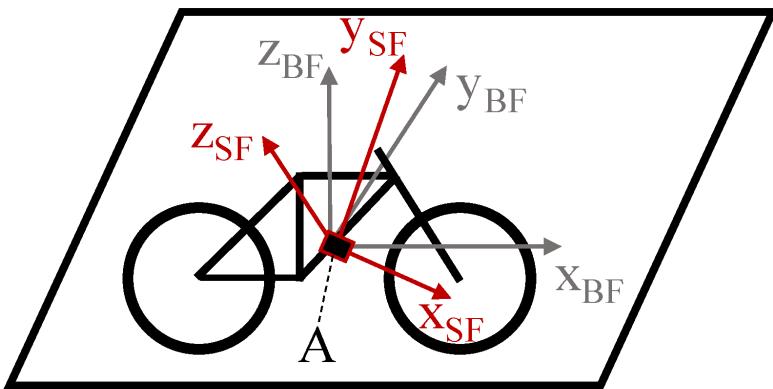


Abbildung 2.12: Achsenrichtung in Sensorframe sowie in Bikeframe [23]

- Die Achsenrichtungen im Sensorframe (Abbildung 2.12)
- Die Achsenrichtungen im Bikeframe (Abbildung 2.12)
- Nachkalibrierung erwähnen: Es wird einmal kalibriert durch ein starkes Beschleunigen sowie Bremsen. Eine Nachkalibrierung läuft immer mit.

## 2.5.2 Übersicht der bereits erkennbaren Unfälle

- Erkannte Szenarien/Fälle
- Zusammengefasste Tabelle von Jan (Vollständige Tabelle im Anhang)

## 2.5.3 Erklärung TipOver

Bei Motorradunfällen auf Straßen, die sich im Anschluss an eine Kurve ereignen, stellt sich immer wieder die Frage nach der maximalen Geschwindigkeit, mit der die Kurve auf einem Motorrad durchfahren werden konnte. Um das Motorrad durch die Kurve zu bewegen, muss sich der Fahrer mit seiner Maschine in die Kurve legen.

## 2.5.4 Erklärung GroundHit

$$AccEnergyStXYintern = AccImpulseSTX^2 + AccImpulseSTY^2$$

$$GH_{scale}xGHEnergyThreshold = GroundHitEnergyThreshold * GroundHitScaleFactor$$

Wobei *GroundHitEnergyThreshold* bei hohen  $TO_{curAngle}_{deg}$  sinkt und *GroundHitScaleFactor* bei hohen *GyrVarSum* sinkt.

*GH* wird erkannt (*GroundHitFire* = 1), wenn:

- (*RideFire* = 2 — *RideFire* = 0) UND

*AccEnergyStXYintern* >= *ScalexGHEnergyThreshold* UND

$$TO_{curAngledeg} \geq 45deg$$

- RideFire = 1 UND
  - AccEnergyStXYintern >= ScalexGHEnergyThreshold UND
- $$TO_{curAngledeg} \geq 40deg$$

### Beispiele 1: Kein GroundHit

Beim Nachsimulieren der Fahrt wurde entdeckt, dass die tatsächliche Kalibrierung nicht richtig war, und wurde dazu eine Kalibrierungsmatrix manuell erstellt und verwendet. Um eine (falsche) Nachkalibrierung zu verhindern, wurde das Modul, was für die Nachkalibrierung zuständig ist, deaktiviert (Abbildung 2.13). Es ist wichtig zu wissen, dass die Position des Geräts während der Fahrt kaum verändert wurde.

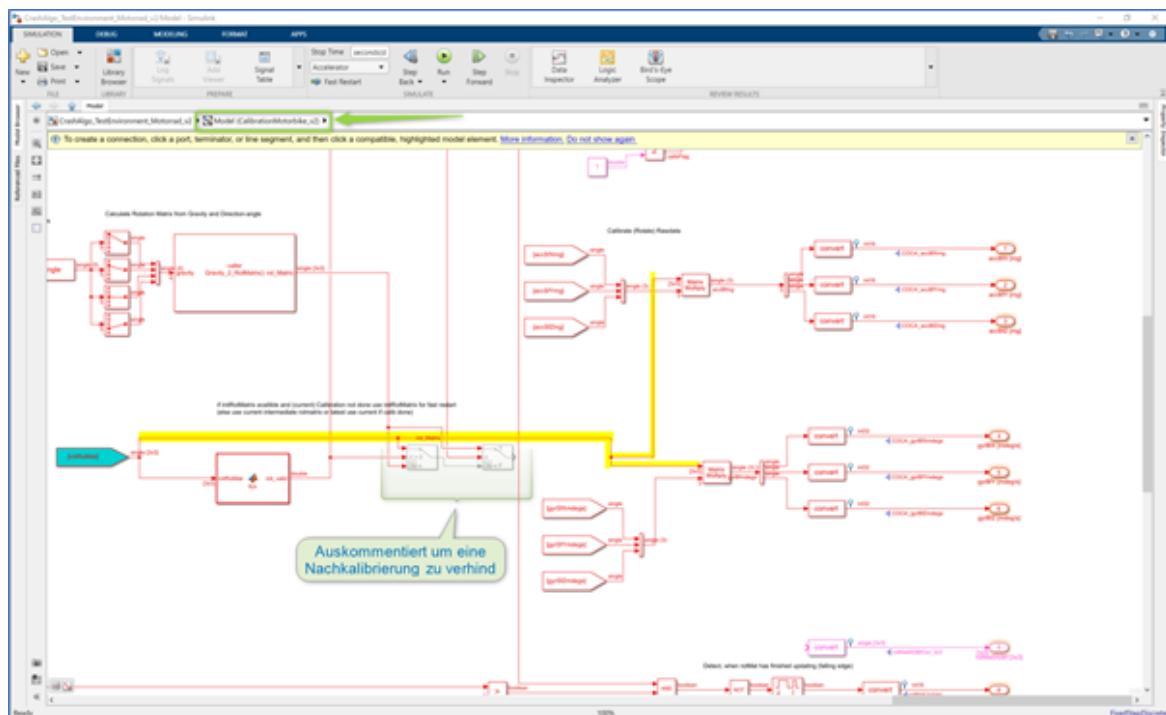


Abbildung 2.13: Kalibrierungsmodul - Nachkalibrierung deaktiviert

In der Abbildung 2.14 und Abbildung 2.15 ist der Fall vorgestellt, in dem kein Groundhit erkannt wurde. Während dieser Fahrt hat kein Unfall stattgefunden. Es wurde allerdings mehrere Szenarien getestet, die im Pocketmode vorkommen können. Z.B. laufen mit einem aktivierten Unfallerkennung oder beim Fahren an der Ampel bremsen und den Fuß auf dem Boden runterstellen mit dem Smartphone in der Hosentasche.

## 2.5 Unfallerkennungsalgorithmus

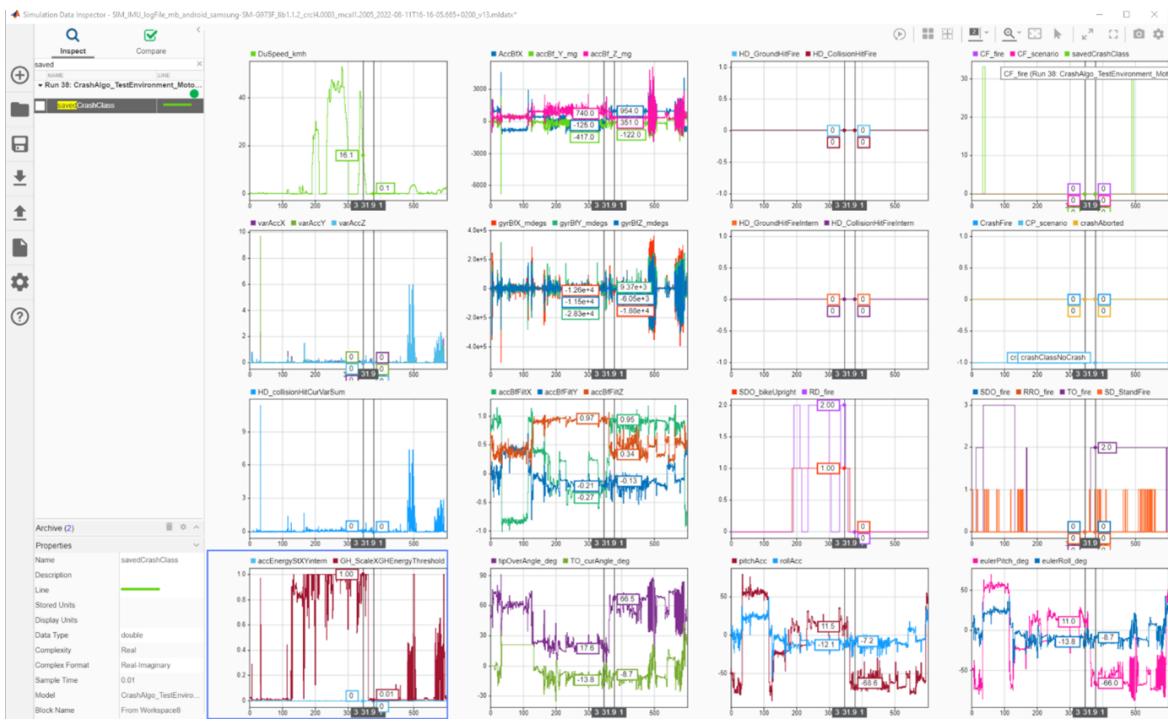


Abbildung 2.14: Testfahrt ohne GroundHit - Full View

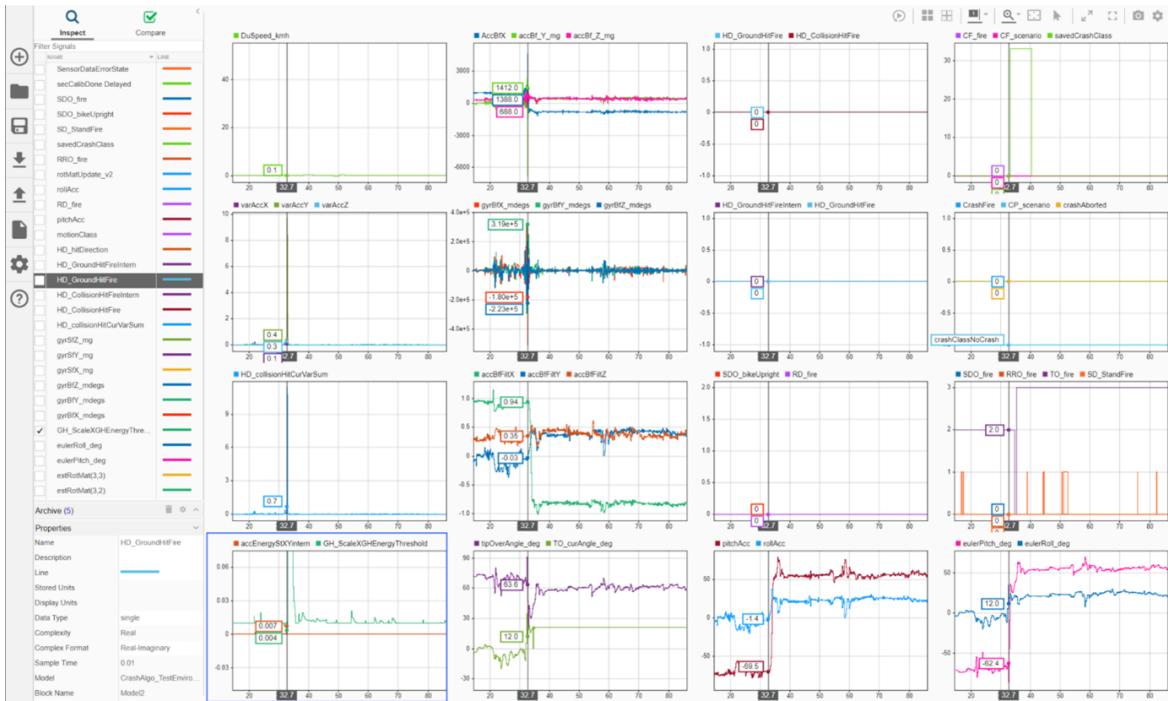


Abbildung 2.15: Testfahrt ohne GroundHit - auf die Energie gezoomt

In der Abbildung 2.15 ist zum Zeitpunkt 32,7 s Folgendes abzulesen:

- $RideFire = 0$ , keine Fahrt
  - $accEnergyStXYintern = 0,007$
  - $GHscalexGHEnergythreshold = 0,004$
  - $TOcurAngledeg = 12$
  - $eulerPitchdeg$  springt von ca. -60 auf ca. +60 (Das handy wurde wahrscheinlich um 180 gedreht. Das könnte auch durch die Acc-Daten bestätigt werden)

An dieser Stelle ist  $accEnergyStXYintern > GHscalexGHenergythreshold$ , was eigentlich einen GroundHit-Alarm auslösen soll. (Während des Testens hat das Smartphone an der Stelle einen falsch-postiven Alarm erkannt) Grund dafür ist, dass die Kalibrierung nicht 100% richtig war. In den simulierten Daten wurde kein Unfall erkannt, da die Kalibrierung richtig war, und der TO-Winkel ( $TO_{curAngledeg}$ ) nicht über 45 war.

## Beispiele 2: Unfall mit GroundHit

In der Abbildung 2.16 sind die simulierten Daten einer echten Fahrt mit einem Unfall dargestellt. Es ist zu bemerken, dass die  $GHSaleXGHEnergythreshold$  von dem Winkel ( $tipOverAngle_{deg}$ ) abhangig ist. Bei einem hohen Winkel sinkt die  $SaleXGHEnergythreshold$  (die linke unterste Darstellung der Abbildung 2.16).



Abbildung 2.16: Crash mit GroundHit - ID 2806 - Full View



Abbildung 2.17: Crash mit GroundHit - ID 2806 - auf die Energie gezoomt

In der Abbildung 2.17 sind die Daten eines Unfalls (ID: 2806) dargestellt und auf die Unfallphase gezoomt. In der Abbildung ist zu erkennen, dass die  $GH_{SaleXGH}Energythreshold$  bei einem hohen Winkel ( $tipOverAngle_{deg}$ ) sinkt. An der Stelle wo die  $accEnergyStXYintern > GH_{SaleXGH}Energythreshold$  ist (ca. 591,35 s), ist der Winkel fast gleich 45 - ; dadurch löst ein Alarm der GroundHit aus. Das entspricht die Erwartungen.

## 2.5.5 Erklärung CollisionHit



# **3 Unfallerkennung im Pocket-Mode**

- Unterschied zum aktuellen Ziel:
- Smartphone wird momentan am Lenker befestigt

Konkreter Unterschied zu meiner MA

## **3.1 Kritische Unfallszenarien**

Liste der Edge- und usecases mit einer Erklärung, warum diese kritisch sind und einen Vorschlag, was man dagegen tun kann.

## **3.2 Lauferkennung**

### **3.2.1 Peaks aufzählen**

1. Idee:

Einen Zähler zu implementieren. Das soll alle Peaks aufzählen mit der Hoffnung, dass es Unterschied zwischen Laufen und Fahren zu erkennen ist. Das LaufFrequenz ist zwischen 0.5-4 Hz. Das Fahren ist über 20 Hz. Es wurde einen Testbeispiel mit einem Sinussignal gebaut und das Prinzip getestet.

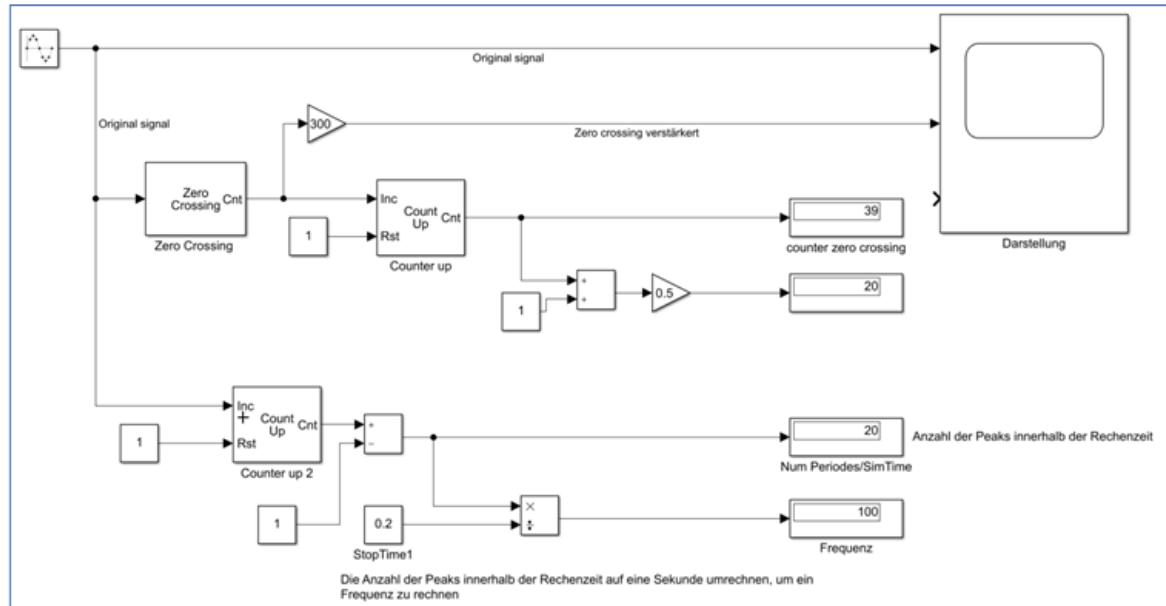


Abbildung 3.1: Testbeispiel - Lauferkennung - Peaks Zähler

In der Abbildung 3.1 ist das Beispielmodell zur Lauferkennung basiert auf das Zählen jedes Peak des Signals. Im Modell wurde ein Sinussignal generiert (Abbildung 3.2) und dieses bearbeitet und das Prinzip dadurch getestet.

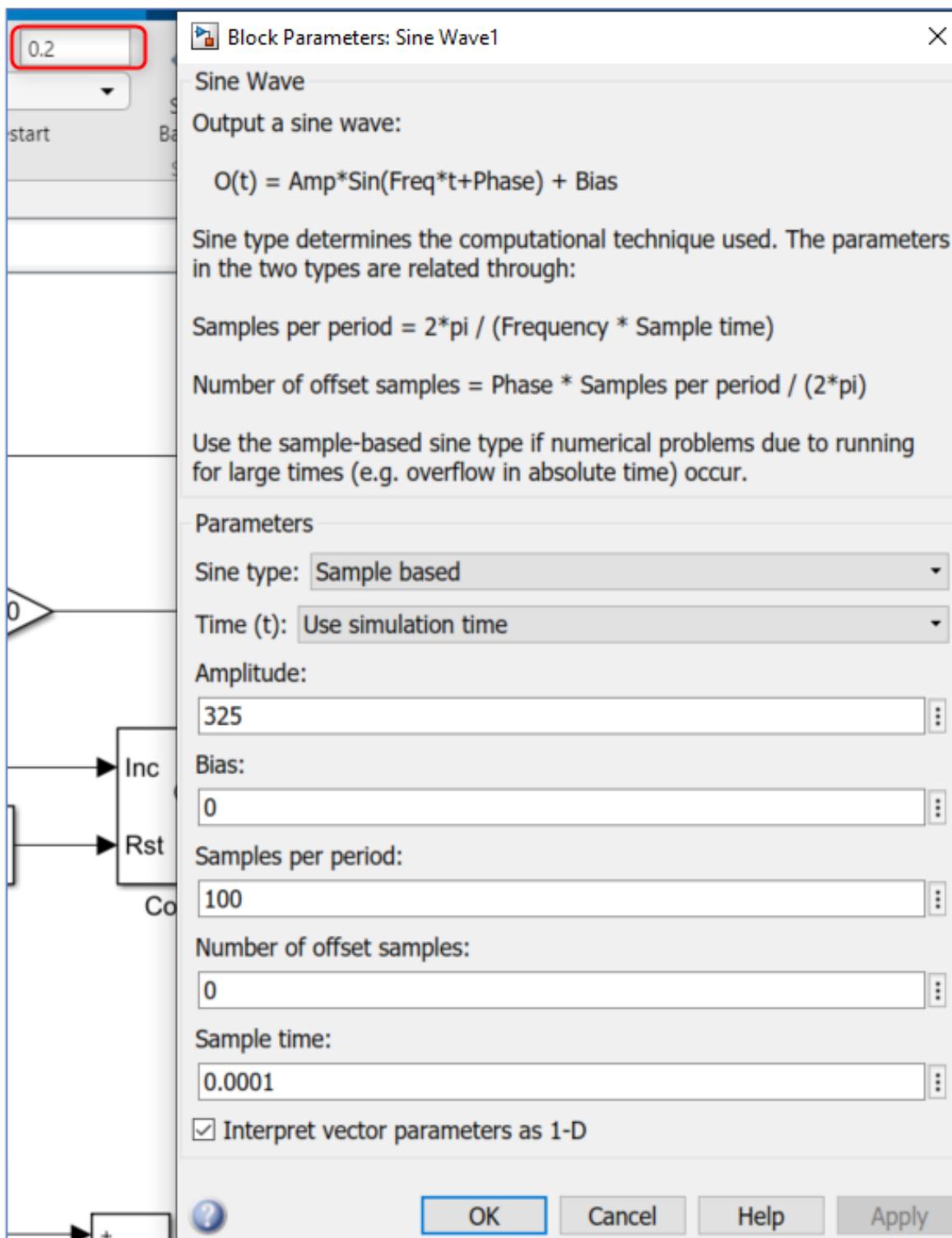


Abbildung 3.2: Testbeispiel - Lauferkennung - Sinussignalgenerator - Spezifikationen

Eine zeitliche Frequenz wird folgendes berechnet:  $Freq(Hz) = Freq(rad/sec)/(2*\pi)$ ;  $\pi = 3,14$

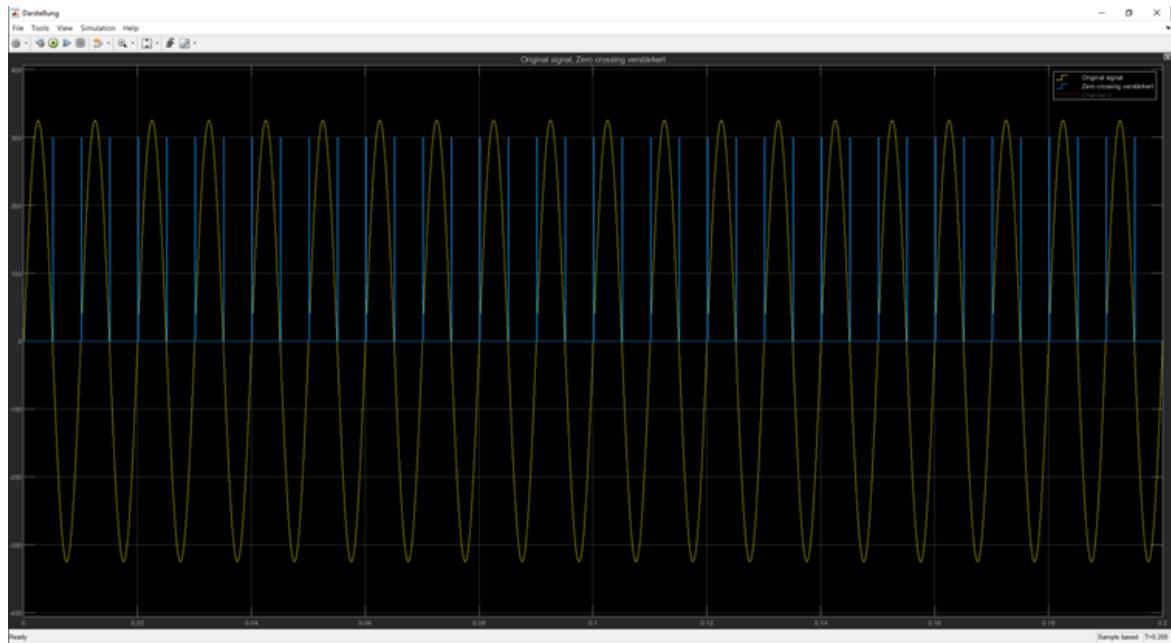


Abbildung 3.3: Testbeispiel - Lauferkennung - Sinussignal

Abbildung 3.3 zeigt die Ausgabe der Scope-Funktion. In der Grafik ist das generierte Sinussignal (gelb) sowie wie of das Signal die Nulllinie überschneidet (blau). In der Scope-Darstellung sind das originale Sinussignal und ein Zähler sichtbar. Der Zähler hat jeden „Zero-crossing“ aufgezählt. Das Testmodell hat die richtige (erwartete) Ergebnisse geliefert: 100 Hz als Frequenz. Beim Einsetzen des gleichen Vorgehens auf das richtige Signal ( $AccBfX_{mg}$ ) (das kalibrierten Signal) wurde eine Frequenz von ca. 11 Hz beim Laufen und eine Frequenz zwischen 20 – 30 Hz ausgerechnet. Das hat die Erwartungen nicht entsprochen. Daraus kann folgendes extrahiert werden: Beide Signale (Szenarien) (Laufen und Fahren) haben die gleiche Menge von Störungen (Rauschen). Der Unterschied ist die Amplitude. Es kann keine Amplitude „Hartkodiert“ entdeckt werden, da die Amplitude sich ständig ändern kann. \*\*\*\*\* Grafiken von den zwei Signalen darstellen und die Art der Rauschen näher analysieren (was könnte die Ursache sein?); Welche mögliche Lösungen gäbe es dafür? \*\*\*\*\*

**Ergebnis:** Die Frequenz muss allerdings doch gesucht und schließlich verglichen werden

### 3.2.2 Frequenzbasierend

Auf die FFT beziehen.

zum Berechnen der Frequenz: FFT (Fast Fourier Transformation)

Nochmal das gleiche mit dem Testmodel durchgeführt.

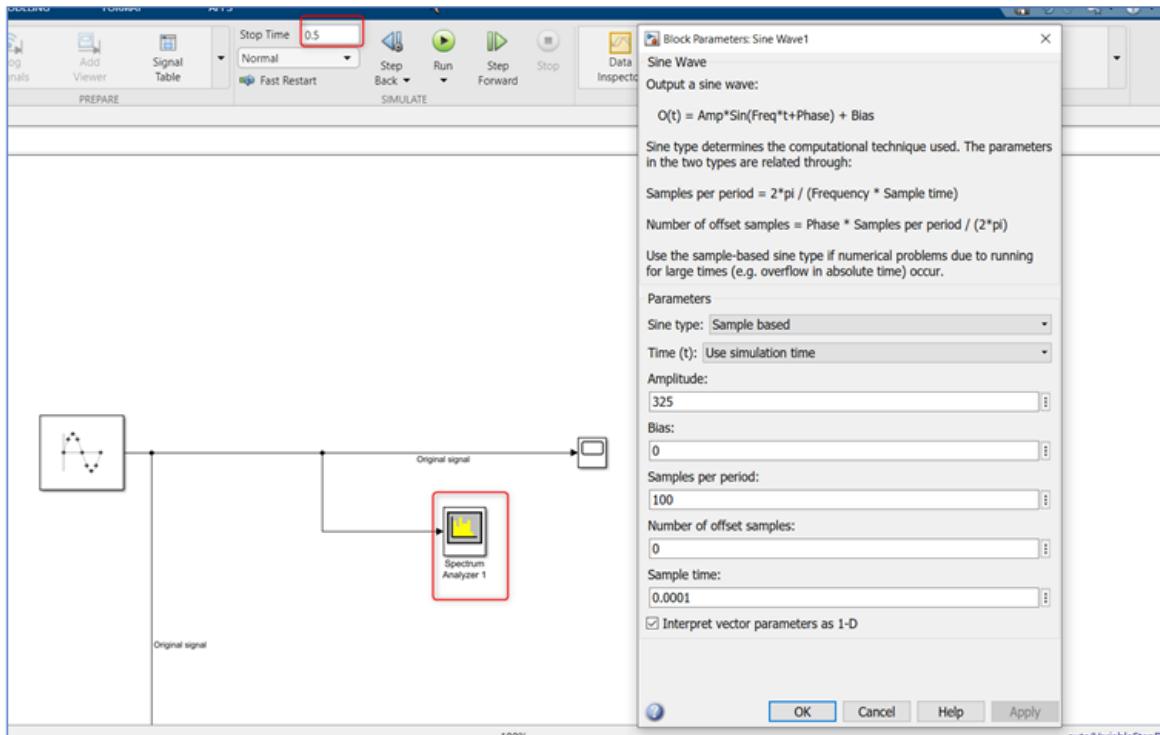


Abbildung 3.4: Testbeispiel - Frequenzbasierte Lauferkennung - Sinussignal

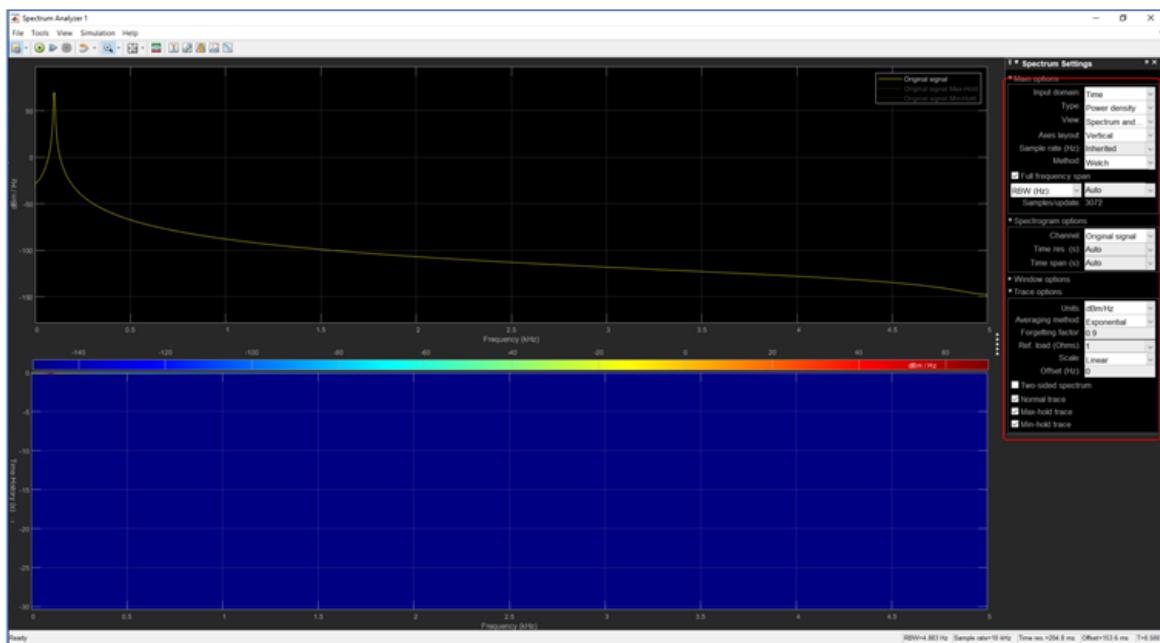


Abbildung 3.5: Testbeispiel - Frequenzbasierte Lauferkennung - Ausgabe des Spektrum-Analyser und seine Spezifikationen

### 3 Unfallerkennung im Pocket-Mode

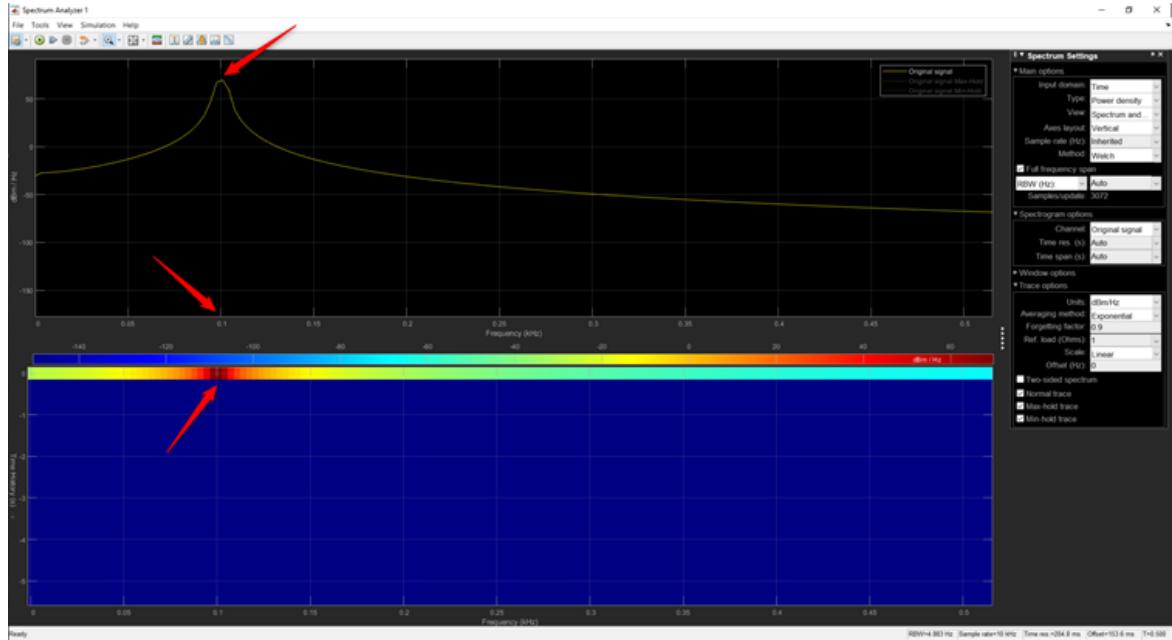


Abbildung 3.6: Testbeispiel - Frequenzbasierte Lauferkennung - Ausgabe des Spektrum-Analyzers im Bereich zwischen 0-0.5 kHz

Abbildung 3.6 Es ist schön zu bemerken, dass die Frequenz (100 Hz) sehr gut sichtbar ist. Der obere Teil zeigt die Häufigkeit einer Frequenz (Spektrum) unabhängig von der Zeit. Der untere Teil zeigt die Häufigkeit einer Frequenz mit der Zeit (Spektrogramm).

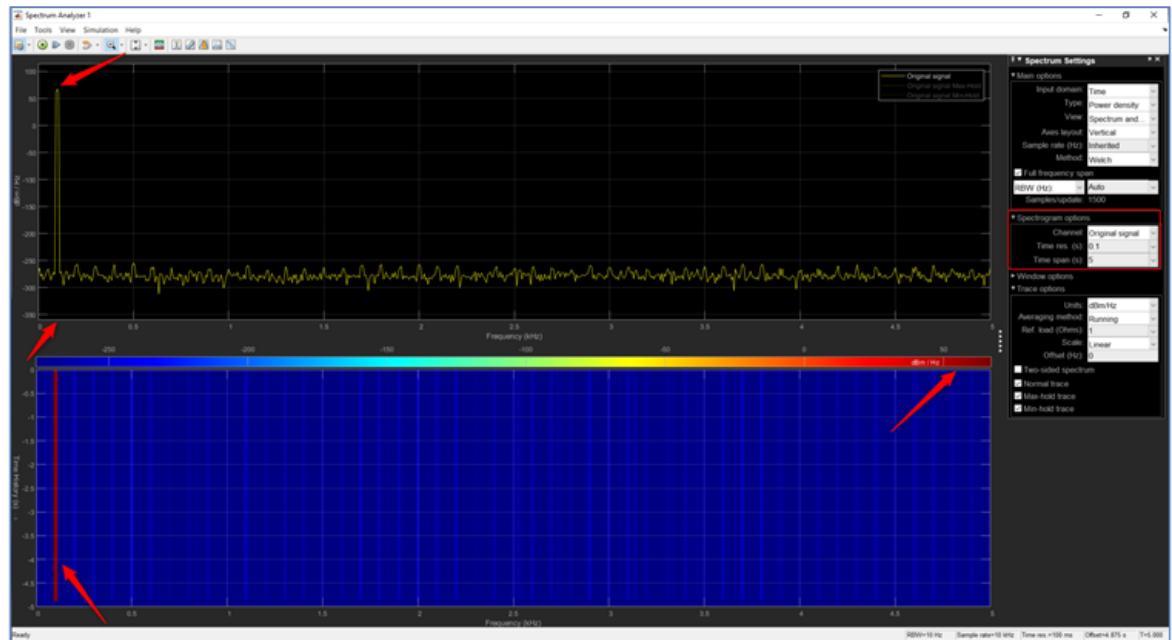


Abbildung 3.7: Testbeispiel - Frequenzbasierte Lauferkennung - Ausgabe des Spektrum-Analyzers mit anderen Spezifikationen

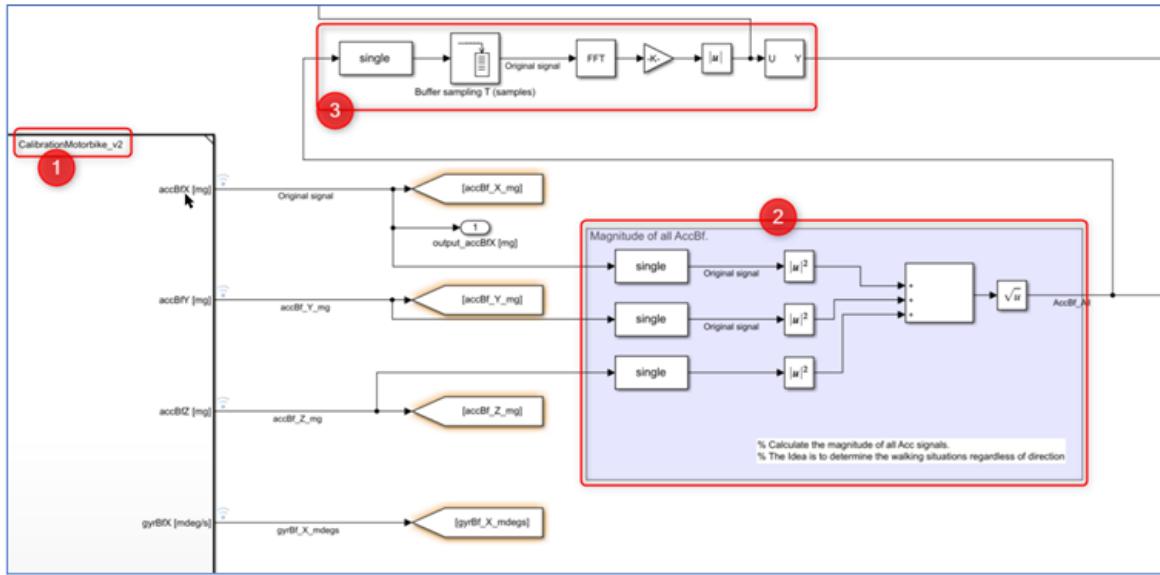


Abbildung 3.8: Testbeispiel - Frequenzbasierte Lauferkennung - FFT - Modell

Die AccBfX, AccBfY und AccBfZ sind die Ausgänge des Modells *CalibrationsMotorbike\_v2* und sie sind die Kalibrierte Signale. Diese Werte werden für die Berechnung der Betrag mit der Formel (\*\*\*\*\*\*) verwendet (Abbildung 3.8- Nummer 2). Das Ziel ist die Frequenzermittlung richtungsunabhängig zu stellen. Danach wurde eine FFT an der Variablen *AccBfAll* durchgeführt, um die Frequenzen zu ermitteln (Abbildung 3.8- Nummer 2).

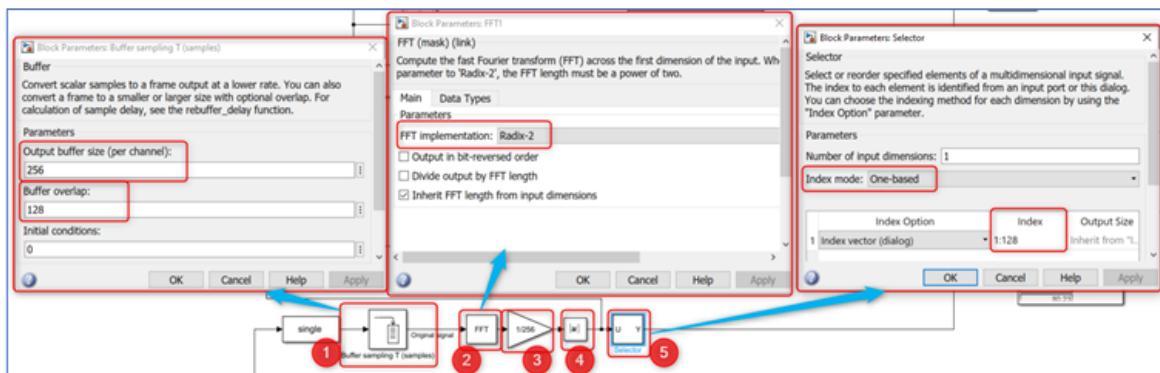


Abbildung 3.9: Testbeispiel - Frequenzbasierte Lauferkennung - FFT - Modell

In der Abbildung 3.9 wird eine FFT durchgeführt. Die Einstellparameter jedes Element ist ersichtlich.

- 1- In der Buffer sind 256 Samples zu betrachten (d.h. ca. 2,5 Sekunden des Signals, da die Abtastrate der Sensor 100 Hz ist). Eine Überlappung von ca. 1 Sekunde wurde auch eingestellt, damit die Zwischen Frequenzen nicht übersehen werden.
- 2- Der FFT-Typ ist eine Radix-2.
- 3- Die Ausgabenwerte der FFT durch 265 dividieren. (warum?)

4- Betrag des FFT-Ausgangs bilden (warum?)

5- Da FFT ein gespiegelter Ausgang liefert wird nur die Hälfte der Matrix angenommen (1:128)

## Entscheidungskriterien - Matlabskript

Screenshot vom Skript mit einer Erklärung (Abbildung 3.10)

```

1  function [motionClass, lastMotionClassTime] = fcn(Intensity, freq, DuSpeed, lastMotionClass, lastMotionClassTime)
2  % status = -1;
3  % for iFq = 1:5
4  if Intensity(iFq) > 100
5  lastMotionClassTime = 0;
6  % lastClassTime = 0;
7  if freq(iFq) >= 7
8  % high frequency (most likely driving), check with speed
9  if DuSpeed < 5
10 % high frequency but very low speed => sitting on motorbike and standing still
11 motionClass = 0;
12 else
13 % there is movement
14 motionClass = 2;
15 end
16 break
17 elseif freq(iFq) >= 0.3906 && freq(iFq) <= 2
18 % walking, (most likely walking) check with speed
19 if DuSpeed <= 7
20 % walking speed
21 motionClass = 1;
22 break
23 else
24 % speed > 7 kmh => not walking => conflict => check next frequency
25 motionClass = -1;
26 continue;
27 end
28 elseif freq(iFq) > 2 && freq(iFq) < 7
29 % undeclared. Maybe walking and maybe driving. Double check with the speed. If the speed is bigger than 7 km/h than he must be driving.
30 % If the speed's smaller than 7 km/h it should stay undeclared, he might be driving with no GPS-Signal (e.g. no speed) or walking
31 if DuSpeed >= 7
32 motionClass = 2;
33 break;
34 else
35 motionClass = -1;
36 continue;
37 end
38
39 elseif freq(iFq) < 0.3906 % undeclared f < 0.3906 Hz
40 % not intended to get here
41 % (sensor couldn't sense frequencies smaller than 0.396)
42 motionClass = -2;
43 break
44
45 else
46 % not intended to get here either; all usecases were defined in the
47 % previous elseifs
48 motionClass = lastMotionClass;
49 break
50 end
51 else
52 if DuSpeed >= 7
53 motionClass = 2;
54 break;
55 else
56 if lastMotionClassTime <=100
57 lastMotionClassTime = lastMotionClassTime + 1;
58 motionClass = lastMotionClass;
59 else
60 lastMotionClassTime = 0;
61 motionClass = -1;
62 end
63 continue
64 end
65 end
66 end
67 end
68

```

Abbildung 3.10: Entscheidungsskript durch mehrere Kriterien (Frequenz und Geschwindigkeit)

## Beispiel

Mögliche Konflikte: ID = 2488; CrashNoPSAP;

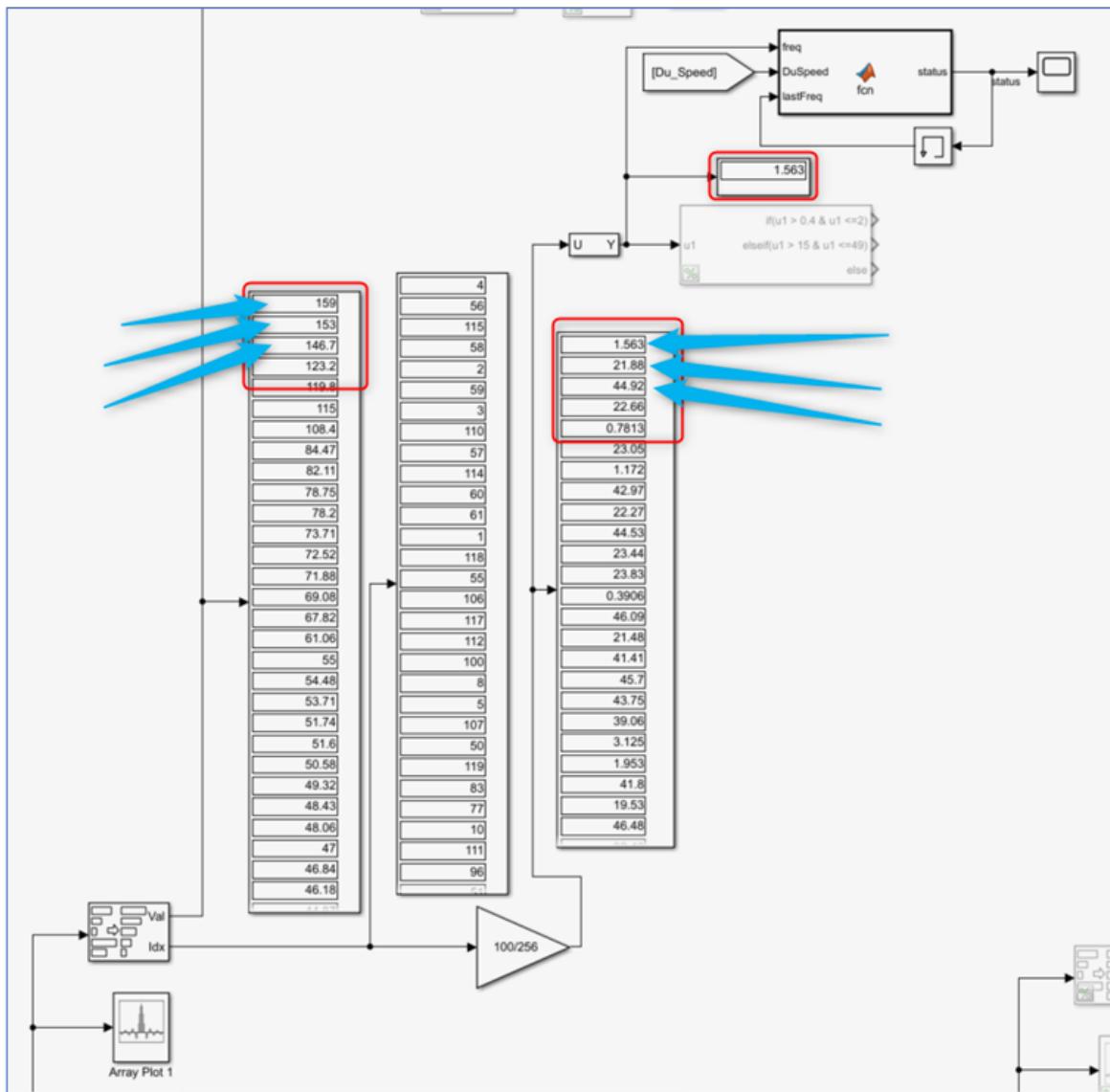


Abbildung 3.11: Testbeispiel - Frequenzbasierte Lauferkennung - Ausgangsbeispiel - ID 2488

Da hier (Abbildung 3.11 und Abbildung 3.12) die maximale Intensität der Frequenz 1,5, wird diese als das Maximum übernommen und weiterbearbeitet. Die nächste größte Intensität liegt sehr nah dazu und hat die Frequenz 21,88 Hz, was eigentlich richtiger ist.

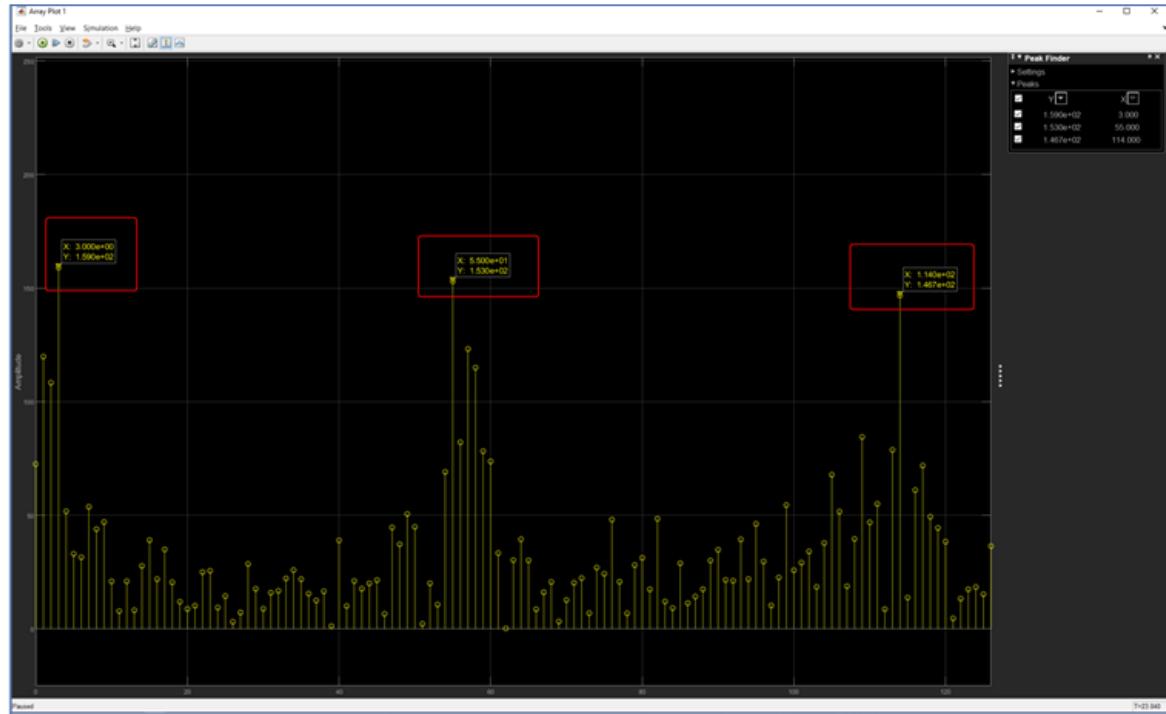


Abbildung 3.12: Testbeispiel - Frequenzbasierte Lauferkennung - Ausgangsbeispiel - ID 2488  
- Scope

#### Werte Sinnvoll auswählen

- gültige Intensität
- Frequenzbereiche
- Gültige Geschwindigkeit
- Ausschlusskriterien

## 3.3 Verifikation des Algorithmus'

### 3.3.1 Groundtruth sammeln

- Videos mit den Daten synchronisieren (Das Tool erläutern? oder vllt. nur erwähnen?)
- Groundtruth labels nachdenken und erläutern
- Videos labeln

# **4 Ergebnisse**



## **5 Ausblick**

- Aktivität immer erkennen und beim Motorradfahren Unfallerkennung automatisch aktivieren (App im Hintergrund laufen)
- Zuverlässiger machen (andere Messwerte miteinnehmen und auswerten)
- Phone-lifting-Funktion einbauen und nutzen -; Flase alarm unterdrücken, wenn das Handy benutzt wird.



# **A Anhang**

- Tabelle von Jan



# **Eidesstattliche Erklärung**

Ich erkläre an Eides statt, dass ich die vorliegende Master-Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Ort und Datum: Stuttgart, den \_\_\_\_\_ Unterschrift: \_\_\_\_\_