

Automatisierte Verkehrsbeobachtung mithilfe eines Beschleunigungssensors

Vorgehen

Mithilfe der *Phyphox*-App wird die Beschleunigung in x,y und z-Richtung während einer Autofahrt aufgenommen.
Anschließend werden die aufgenommenen Daten programmatisch ausgewertet.

Auswertung

Die Auswertung erfolgt über eine selbst entwickeltes Anwendung. Zu Beginn werden die Daten in CSV-Format (Comma separated value) eingelesen.
Anschließend werden die Rohdaten in sog. *ValueRelations* überführt. Dabei handelt es sich um eine Relation zwischen zwei Werten aus Phyphox. Diese Relationen enthalten Daten, wie die zwei originalen Werte, die Differenz deren und ob der Wert ansteigt, abfällt oder gleich bleibt.

| relation | first | second | diff | firstTime | secondTime |
|----------|-------|--------|------|-----------|------------|
| ASC | 0.5 | 1.2 | 0.7 | 0.02 | 0.05 |

Abbildung 1: Beispielhafte ValueRelation

Nun werden in den Datensätzen bestimmte vordefinierte Muster gesucht, beispielsweise:

```
ValuePattern(  
    ValuePatternPart(  
        ValueRelationType.ASCENDING,  
        2,  
        Int.MAX_VALUE,  
        0.5,  
        Double.MAX_VALUE  
    ), ...  
)
```

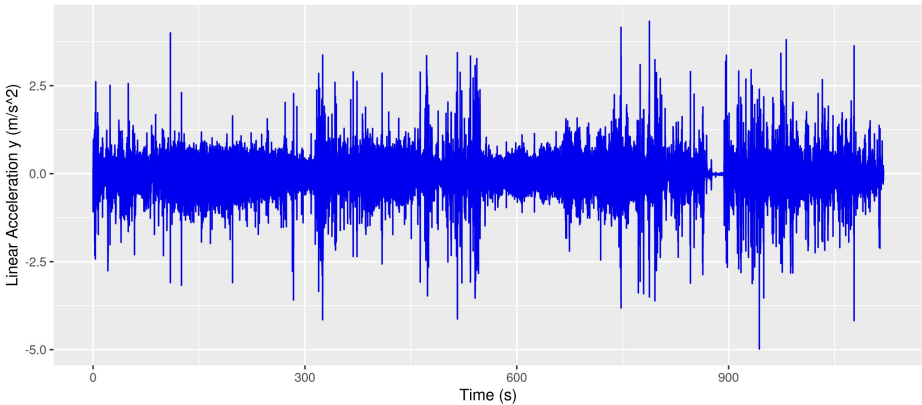
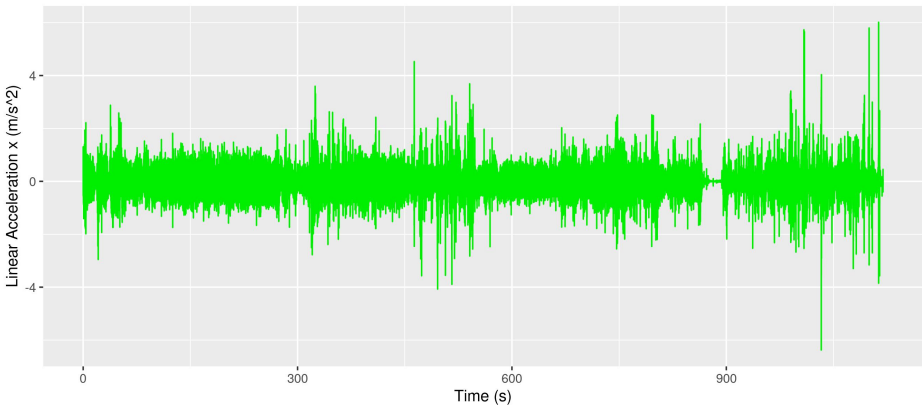
In diesem Beispiel wird ein Muster definiert, welches zwischen 2 und 2147483647 (Int.MAX_VALUE) ansteigende Datenpunkte mit einer Steigung zwischen 0.5 und $1.7 \cdot 10^{308}$ (Double.MAX_VALUE) akzeptiert



Abbildung 2: Beispielhafte Verläufe, welche vom obigen Muster akzeptiert werden

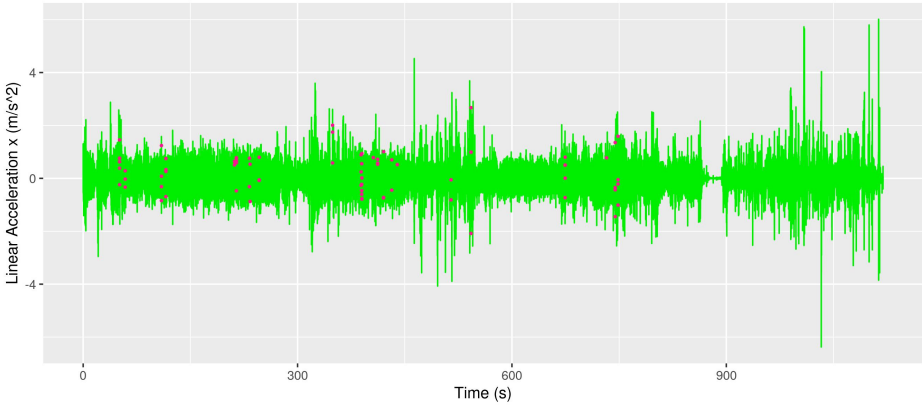
Ergebnisse

Die hier betrachtete Autofahrt dauerte ~ 18 Minuten.
Die Rohdaten sehen wie folgt aus:



Kurvenerkennung

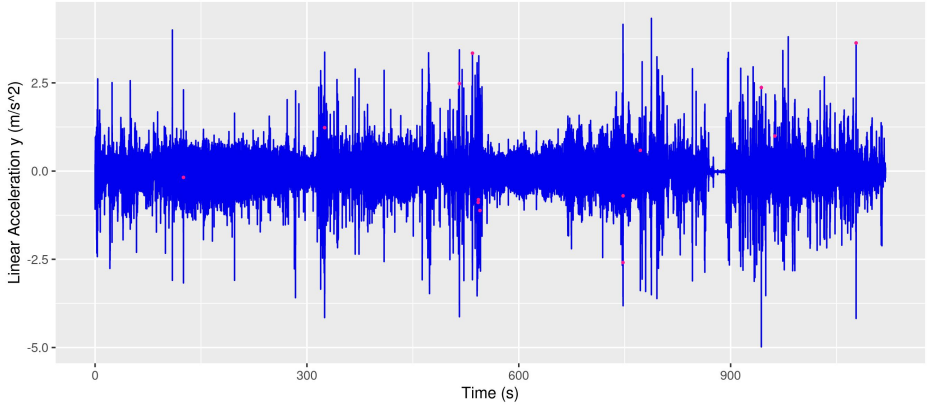
Um Kurven zu erkennen wird die Beschleunigung in x-Richtung betrachte. Nach Anwendung des passenden Musters ergibt sich folgendes Bild:



Hier entspricht jeder Punkt einer gefahrenen Kurve.

Abruptes Beschleunigen / Bremsen Erkennung

Um ein abruptes Beschleunigen / Abbremsen festzustellen werden die Daten in y-Richtung betrachtet. Dabei wird nach kurzen, starken Anstiegen gefiltert:



Jeder Punkt hier entspricht einer Beschleunigung / Bremsen $\leq 4m/s^2$

Technisches

Die Software ist in *Kotlin* realisiert, einer Java-ähnlichen Sprache, die auch auf der *JVM* (Java Virtual Machine) läuft.
Also Buildsystem dient *Make* und *Maven*.
Zum einlesen und verarbeiten der Rohdaten dient die Library *krangl*, zur Visualisierung *kravis*
Also Frontend für die Daten dient *Jupyter notebook* mit dem *Kotlin kernel*.
Als IDE kommt *IntelliJ Ultimate* zum Einsatz.
Git wurde zur *Versionsverwaltung* verwendet.
Das Plakat wurde in *LaTeX* erstellt.
Die Diagramme wurden aus *Jupyter notebook* exportiert.
Bilder wurden mit *Tikz* gezeichnet.
Das volle Projekt inklusiv Rohdaten, Sourcecode für das Programm und Poster und Bilder sind unter <https://github.com/MohrJonas/DIYPhysicsExperiment> verfügbar.
Scannbarer QR-Code:

