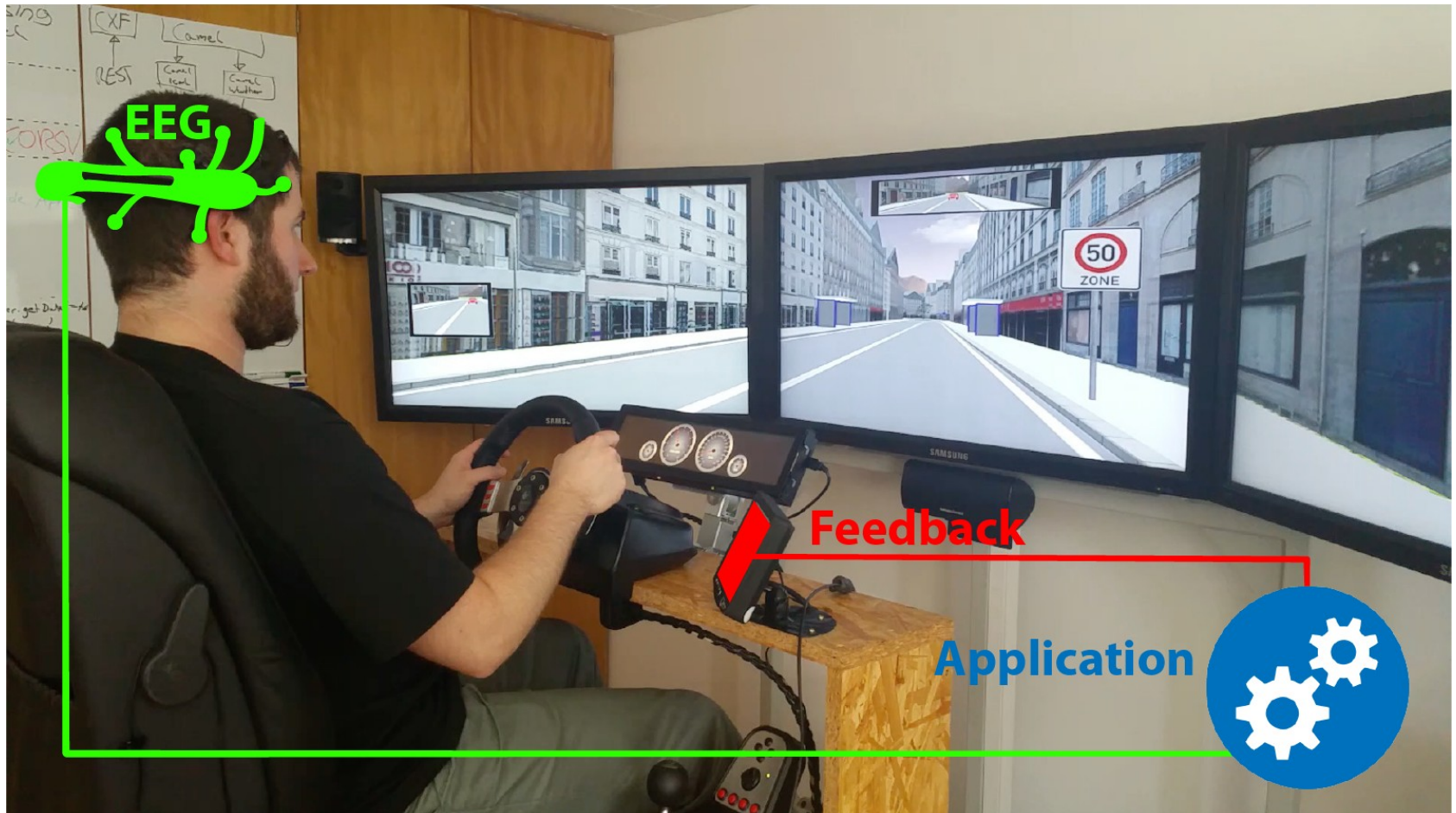


PORTABLE SYSTEM TO DETECT DRIVER DROWSINESS WITH BODY SENSORS

PAUL PASLER - paul.pasler@student.reutlingen-university.DE



Motivation und Kurzbeschreibung

- Jeder 5. Unfall lässt sich auf Müdigkeit zurückführen [EVEo8]
- Der flächendeckende Einsatz von Fahrerassistenzsystemen könnte die Zahl schwerer Unfälle um bis zu 28% verringern [MAA15]

Ziel ist es diese Zahl zu verringern,
indem der Fahrer rechtzeitig vor drohender
Müdigkeit gewarnt wird

Erkennung von Müdigkeit

- im Fahrzeugumfeld
- mit Körpersensoren (EEG)
- mit leicht portierbarer Hardware



[MAA15]

Xavier Mosquet, Michelle Andersen and Aakash Arora. „A road map to safer driving through advanced driver assistance systems.“

[Eveo8]

Claudia Evers. „Unterschätzte Risikofaktoren Übermüdung und Ablenkung als Ursachen für schwere LKW-Unfälle.“



Meilensteine

M2.2: April

Aufbereitete Testdaten die zum Training des Klassifikators geeignet sind

Vorbereiteter Klassifikator Körper der auf „echte Daten“ wartet

M3: Juni

Experiment Teil 1: Aufgenommene Testdaten (EEG) und Videos mit übermüdetem Fahrer (~3 Teilnehmer)

Analysierte und gelabelte Testdaten aus dem Experiment

Trainierter und getesteter Klassifikators zur Erkennung von Müdigkeit

M4: Juli

Portierung und Test des Systems in einem echten Fahrzeug

Derzeitiger Stand

- EEG ist kaputt und immer noch beim Emotiv Support
- Codeänderungen im Simulator wegen Problemen beim Lenkverhalten für hohen Geschwindigkeiten (Lenkwinkel zu stark → Überschlag)
- Implementierung der EEG Rohdaten Verarbeitung
 - EEG-Frequenzbänder mit einem Bandpass-Filter
 - Emotiv EEG Bibliothek in Python
- Anbindung eines Klassifikators
- Dokumentation des Simulators / Einführung für Juniors

Feinschliff des Testszenarios im Simulator

Codeänderungen im Simulator wegen
Problemen beim Lenkverhalten für hohen
Geschwindigkeiten
(Lenkwinkel zu stark → Überschlag)

- Funktion die den Lenkwinkel an die Geschwindigkeit anpasst
- Simulator *.jar bauen und deployen



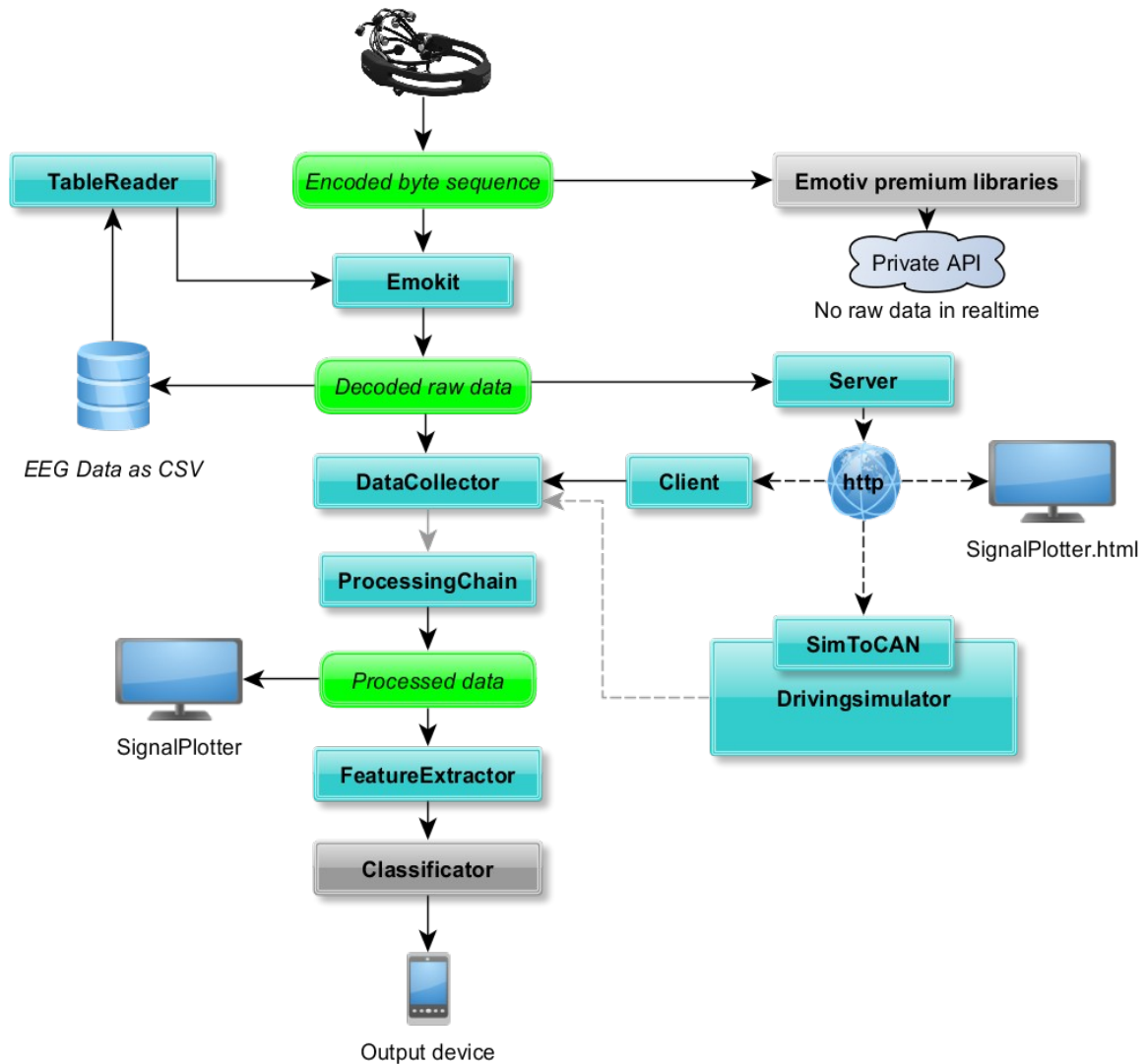
Systemübersicht

EEG Rohdaten können über mehrere Kanäle geholt werden:

- Direkt (Emokit)
- Per http (Server)
- Via CAN-Bus (Sim2Can)

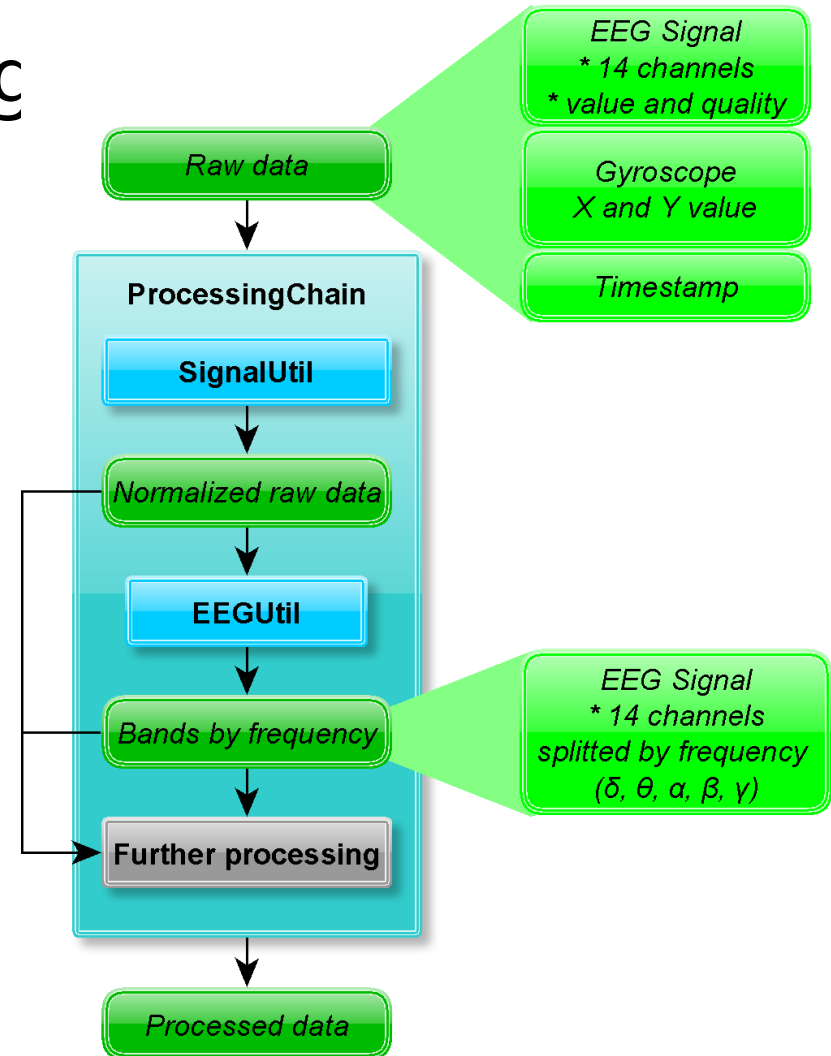
Maximale Flexibilität für Folgeprojekte.

Code ist Dokumentiert und getestet (Unittests).



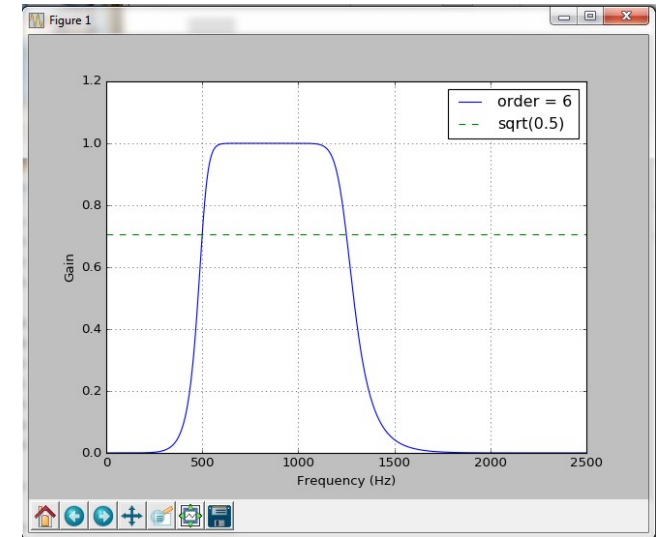
EEG Rohdaten Verarbeitung

- Rohdaten bestehen aus
 - 14 EEG Kanälen mit Signal-Wert und - Qualität
 - Gyroskopinformationen (X und Y)
 - Zeitstempel
- Signal wird normalisiert (-1 und 1)
- Bandpassfilter (Frequenzbereich)
- Frequenzbänder teilen
 - δ , θ , α , β , γ - Wellen
- Weitere Verarbeitungsschritte
 - Literaturrecherche



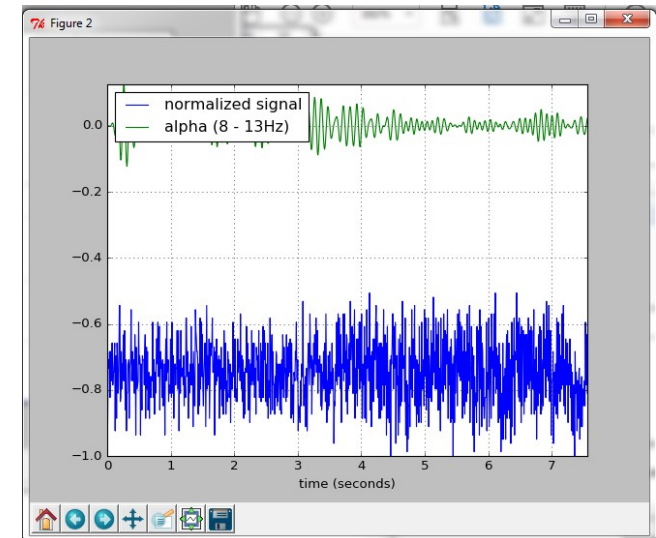
Bandpassfilter: Butterworth

- Kontinuierlicher Filter für bestimmte Frequenzbereiche
- Idealerweise 1 in den gewollten Frequenzen, sonst 0. Sprung hat aber unschöne Eigenschaften, darum kontinuierliche Funktion
- Durchgeführt für die Frequenzbänder: δ , θ , α , β , γ



Oben: Butterworth frequenzfilter

Unten: Rohes EEGSignal und gefilterte alpha Wellen



Nächste Schritte

- Weitere Verarbeitungsschritte implementieren
- Fehlende Klassen vorbereiten / implementieren
 - Klassifikator
- Fahrsimulatorcode fertigstellen und Testszenario starten
- Abgreifen der CAN-Bus Signale

Projektkennblatt 4.0: Meilensteine

- M2 wird aufgeteilt und bis April verlängert
- Aufgaben die ohne EEG erledigt werden können, werden aus M3 vorgezogen
 - Aufbereitung der Daten
 - Prototypische Implementierung des Klassifikators
- Experiment „Teil 1“ verschiebt sich in M3, Experiment „Teil 2“ wird nicht durchgeführt

Fazit

- Es wird spannend, ohne EEG kann das Projekt nicht abgeschlossen werden
- Liefert der Versuch nicht die benötigten Daten, ist keine Zeit für einen zweiten Durchlauf
- Plan B:
 - Ausführliche Dokumentation des Simulators
 - How-To's für Folgeprojekte