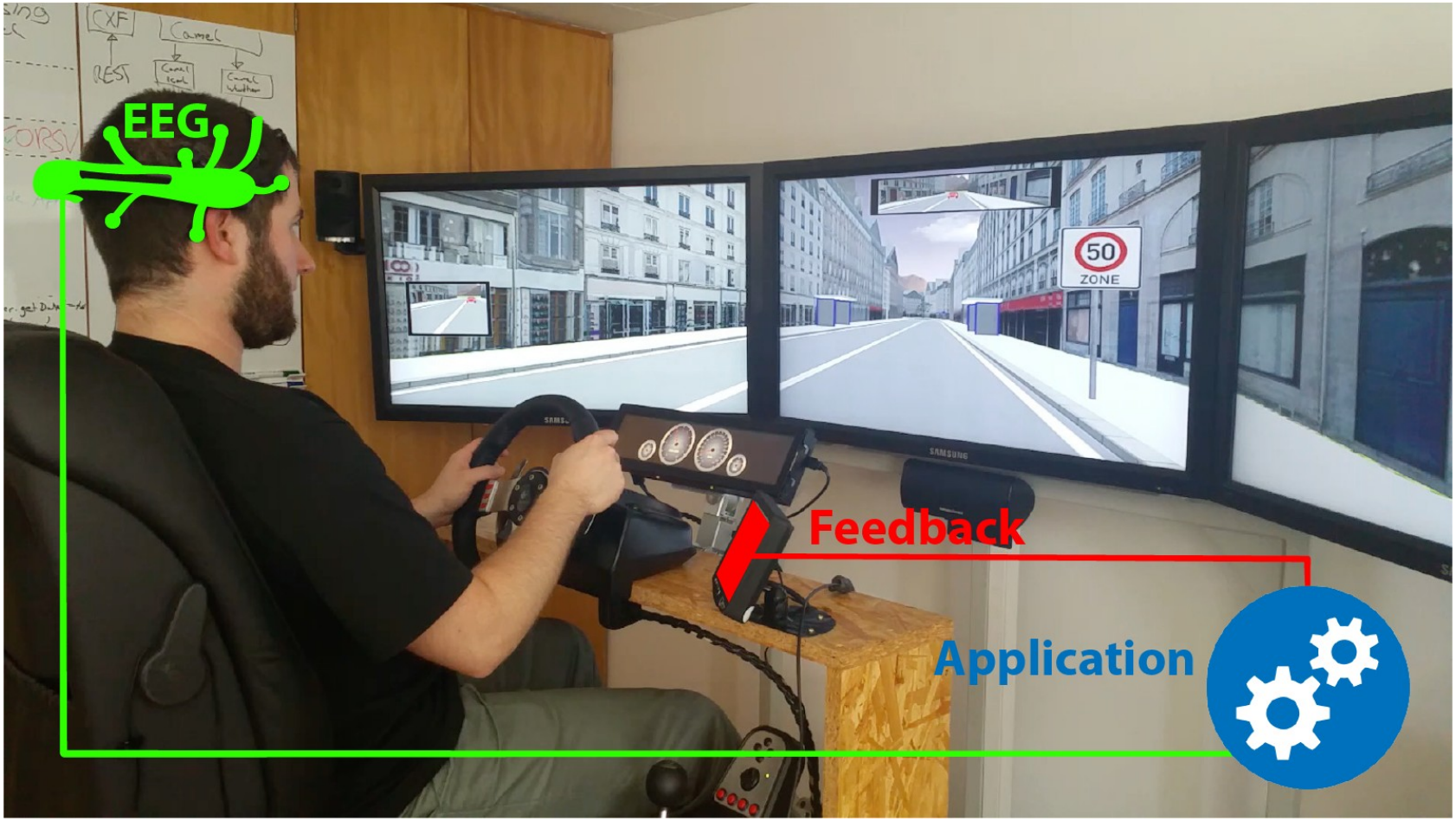
PORTABLE SYSTEM TO DETECT DRIVER DROWSINESS WITH BODY SENSORS

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

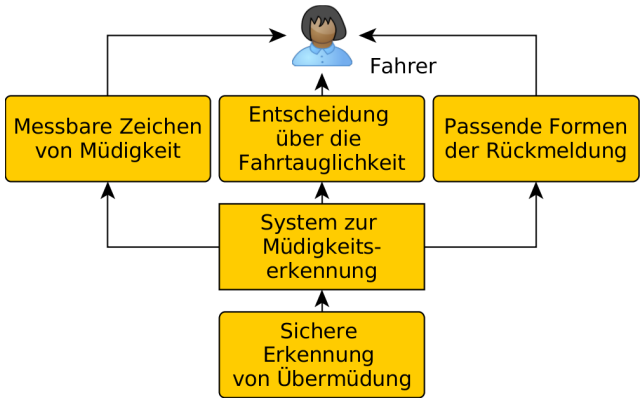


1 Motivation

- Der flächendeckende Einsatz von Fahrerassistenzsystemen könnte die Zahl schwerer Unfälle um bis zu 28% verringern [2]
- In Deutschland wurden 2015 rund 2,5 Mio. Unfälle polizeilich aufgenommen, die Zahl der Verkehrstoten liegt bei 3.450 [4]
- Jeder 5. Unfall lässt sich auf Müdigkeit zurückführen [5]

1 Problemstellung / Ziel

Für ein System zur Müdigkeitserkennung müssen
- messbare Zeichen von Müdigkeit ermittelt werden
- eine Entscheidung getroffen werden, ob der Fahrer noch in der Lage ist sein Fahrzeug zu führen
- der Fahrer auf seine aktuelle Fahrtauglichkeit hingewiesen werden



Ziel ist es die Zahl schwerer Unfälle zu verringern, indem der Fahrer rechtzeitig vor drohender Müdigkeit gewarnt wird.

Erkennung von Müdigkeit
- im Fahrzeugumfeld
- mit Körpersensoren (EEG)
- mit leicht portierbarer Hardware
- und verteilter Software-Architektur

2 Stand der Technik I

Vorgehensweisen zur Müdigkeitserkennung durch Analyse
- des Fahrverhaltens
- von optischen Parametern (CV, Computer Vision)
- von Körpersignalen

Fahrverhalten
- Eingesetztes Vorgehen in der Praxis (Bspw. Daimler, VW, Bosch [3] [7])
- Kaum zugängliche Information / Veröffentlichungen
  - Abkommen von der Spur mit heftigem Gegenlenken

Optisches Verfahren
- Fahrer wird von einer Kamera gefilmt und analysiert
- CV-basiert funktioniert ist, aber optischen Grenzen unterworfen [8][9]
  - Das Verhältnis von offenen zu geschlossenen Augen (PERCLOS, Percentage of Eye Closure)

2 Stand der Technik II

Körpersensoren

- Analyse nur mit EKG / EOG keine eindeutigen Ergebnisse [10][11]

- Kombinationen EKG / EEG und EOG / EEG liefern gute Ergebnisse, im Vergleich reicht jedoch auch nur das EEG aus [13]

- Verschiedene Arbeiten nur mit dem EEG
  - Subasi et al. [14] Unterscheidung von „wach", „schläfrig" und „schlafen", Künstliches Neuronale Netz (KNN), Erkennungsrate 93%
  - Vuckovic et al. [15] Suche nach bestem Algorithmus für den Aufbau eines KNN, Erkennungsrate 90%
  - Weitere Arbeiten mit EEG: Huang et al. [16] 90%, Lin et al [17] 88%

2 Analyse

- Fahrverhaltens- und CV-Ansätze grundsätzlich geeignet, werden jedoch nur zum Labeln der EEG Sequenzen herangezogen

- Das EEG eignet sich sehr gut für die Erkennung von Müdigkeit
- Entweder medizinisches EEG oder Eigenentwicklungen
- Keine Tests in echten Fahrsituationen
- KNN wird häufig verwendet (aber auch LDA, HMM, SVM)

3 Anforderungen

Must
- Präzision und Korrektheit (false positive / false negative)
- Robust gegen Fehler / Fehleingaben sein
- Performance
- Portabilität
- Einfache Handhabung und möglichst hoher Tragekomfort

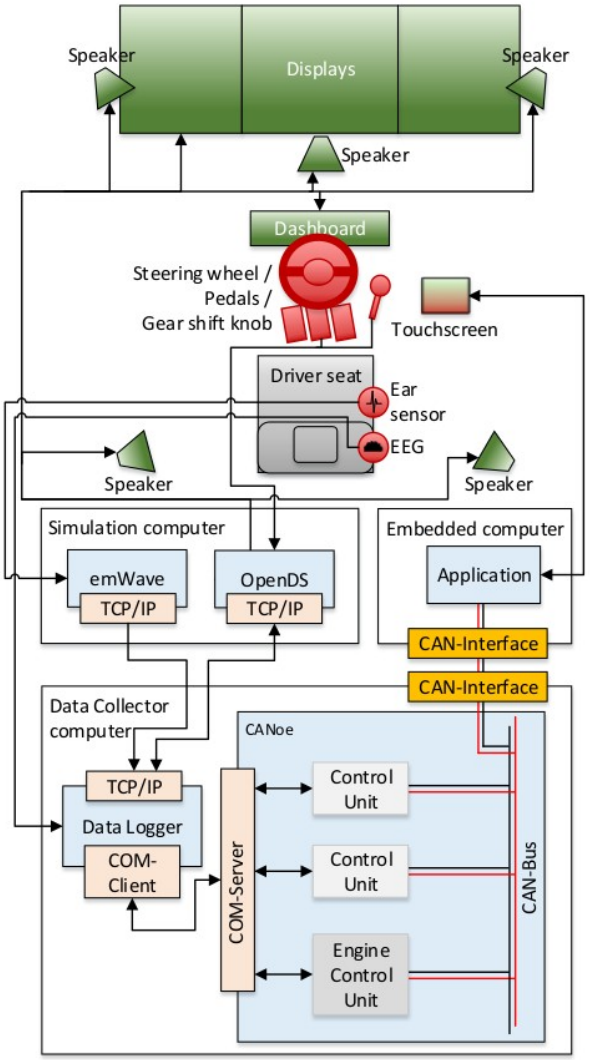
Should
- Flexibilität
  - Verschiedene Betriebssysteme und Gerätetypen
  - Verteilte Anwendung
- Eindeutige Kommunikation mit dem Fahrer

Qualität
- Dokumentation
- Testabdeckung

4 Infrastruktur

- Anwendung muss in der Fahrsimulator Umgebung laufen

- Anwendung wird in Python 2.7 geschrieben
  - Leicht zu lernen und zu verstehen
  - Bibliotheken für wissenschaftliche
  - MachineLearning Bibliotheken
  - Portierbar auf Win / Mac / Linux, Tablet / Smartphone
  - Ausreichende Performance



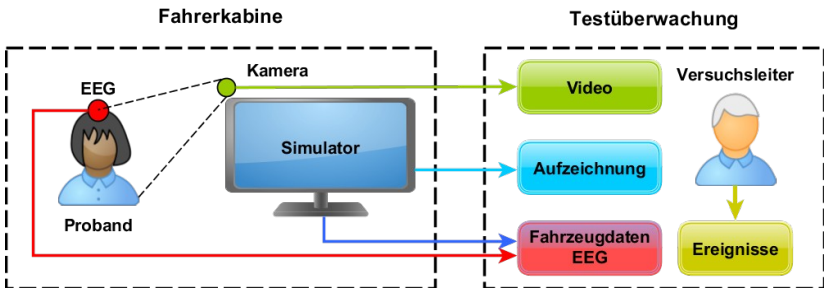
4 Testdatenbeschaffung / Experiment

Ziel ist es, den Testfahrer möglichst müde zu machen, sodass er deutliche Anzeichen für Müdigkeit zeigt

Szenario im Fahrsimulator

Dafür werden folgende Parameter gesetzt

- Experiment findet zwischen 14:00 und 16:00 statt [24]
- Schlafmenge vor dem Experiment < 6h [23]
- Simulierte Nachtfahrt (Dunkel)
- Eintönige Geradeausstrecke ohne Abwechslung
- Keine anderen Verkehrsteilnehmer



Aufbau des Experiments

Erhobene Daten

- EEG Rohdaten
- Video (Fahrer / Simulation)
- Fahrzeugdaten (Geschwindigkeit)
- Besondere Ereignis (Abkommen von der Spur)

5 Systemübersicht / Datenfluss I

Rohdaten kommen über freie Lib „Emokit“

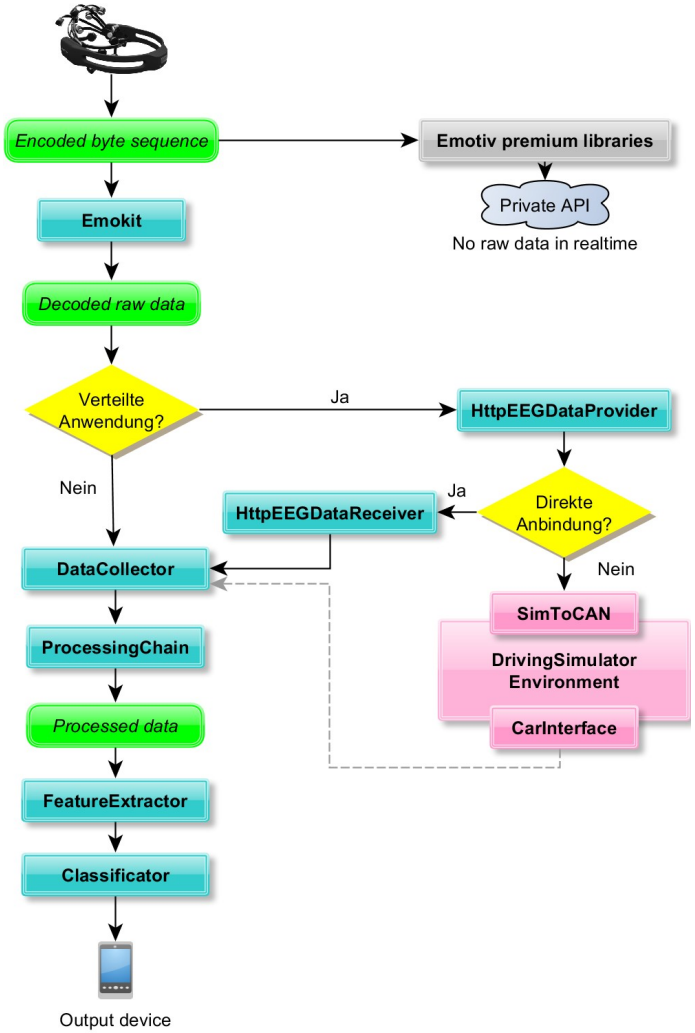
Die Rohdaten können
- direkt (Anwendung läuft auf einem System)
- via http (Holen und Verarbeitung auf verschiedenen Systemen)

übertragen werden

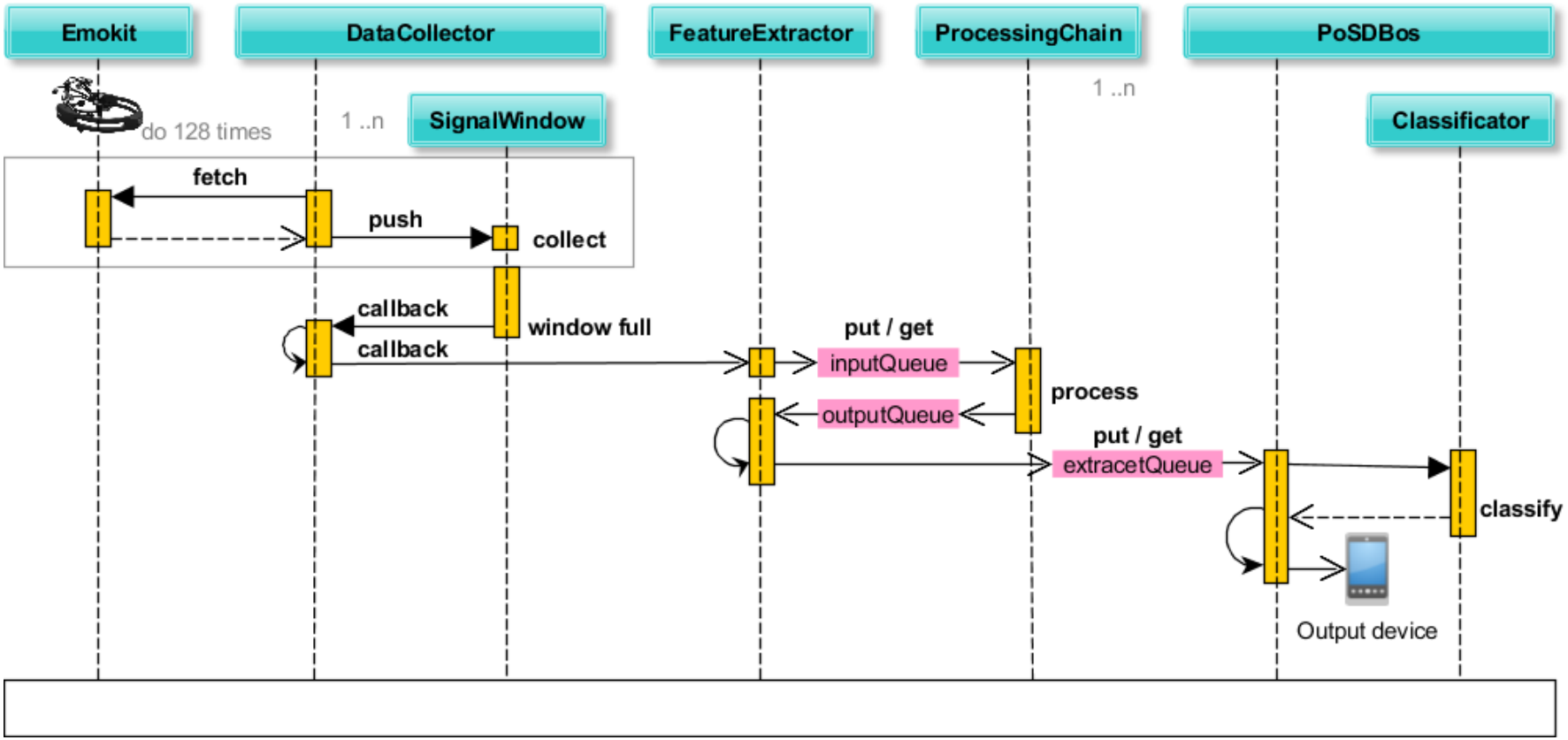
Über den http Weg können die Daten
- von einem http-Client
- mit CAN über die Simulator Infrastruktur

zur Verarbeitungsschicht geleitet werden

Flexible Datenübertragung

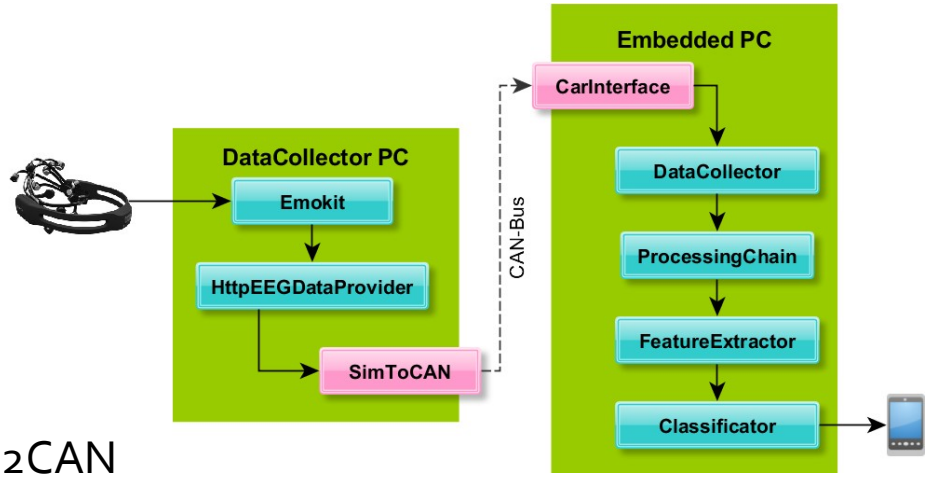


5 Systemübersicht / Datenfluss II



5 Systemintegration

Weg der Daten in die Anwendung



DataCollector PC

- via http werden die Rohdaten an Sim2CAN
- von dort auf den CAN-Bus gelegt

Embedded PC

- das CarInterface holt die relevanten Daten vom CAN Bus
- die Datenverarbeitung wird angestoßen
- bei erkannte Müdigkeit bekommt der Fahrer eine entsprechende Meldung

Dies simuliert das EEG als Teil des Fahrzeugs,
sodass es vom virtuellen Steuergerät verwaltet wird.

5 EEG Rohdaten Verarbeitung

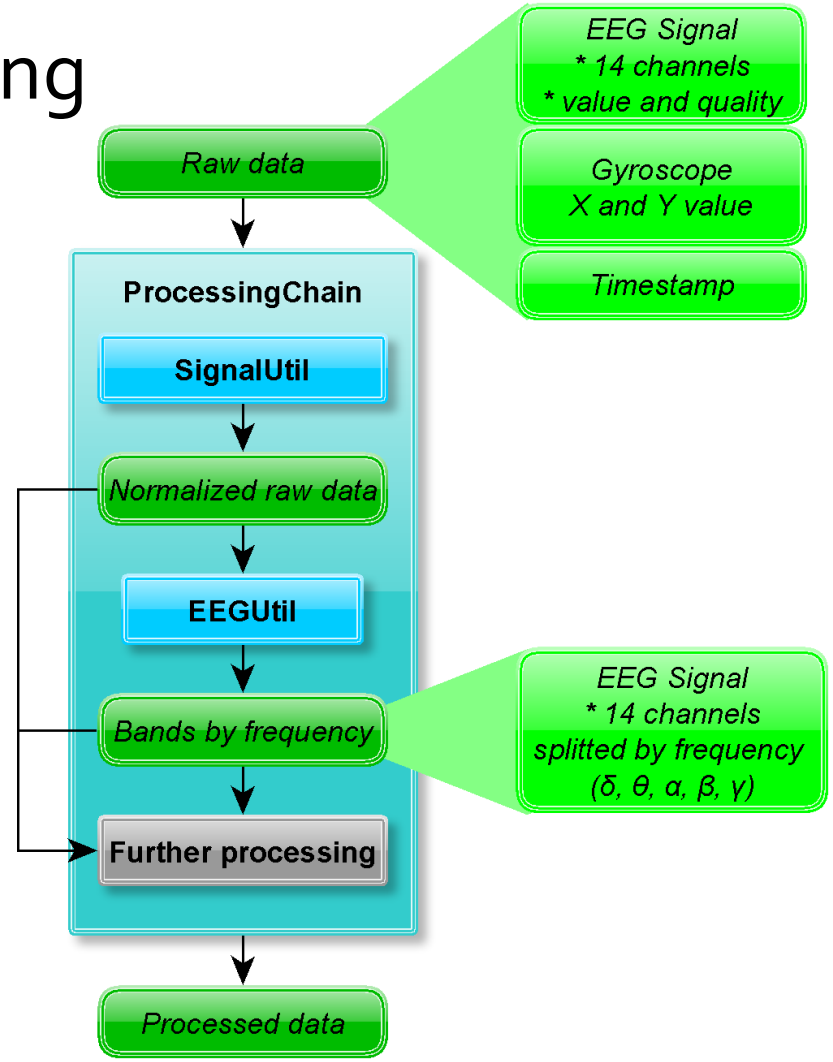
- Rohdaten bestehen aus
  - 14 EEG Kanälen mit Signal-Wert und - Qualität
  - Gyroskopinformationen (X und Y)
  - Zeitstempel

- Signal wird normalisiert (-1 und 1)

- Bandpassfilter (Frequenzbereich)

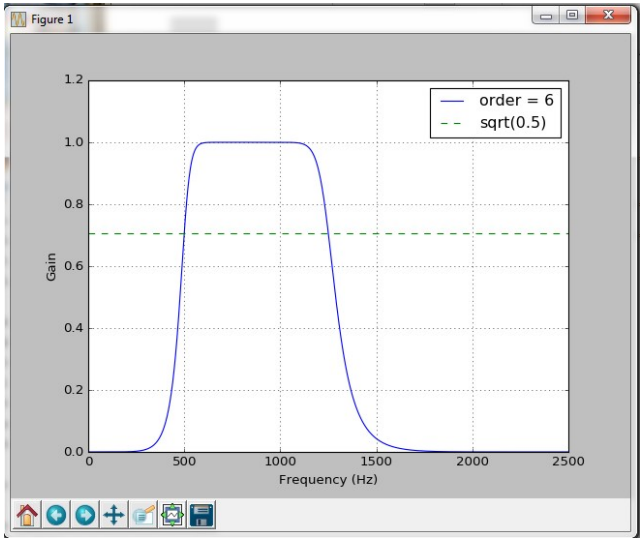
- Frequenzbänder teilen
  - δ, θ, α, β, γ - Wellen

- TBD

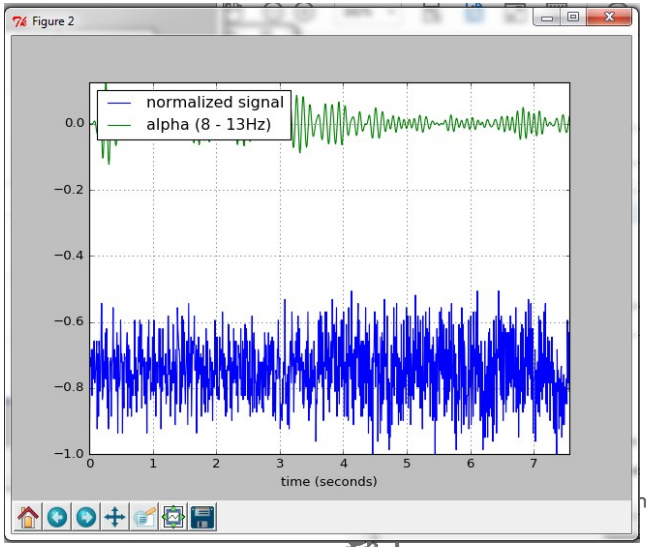


5 Bandpassfilter: Butterworth

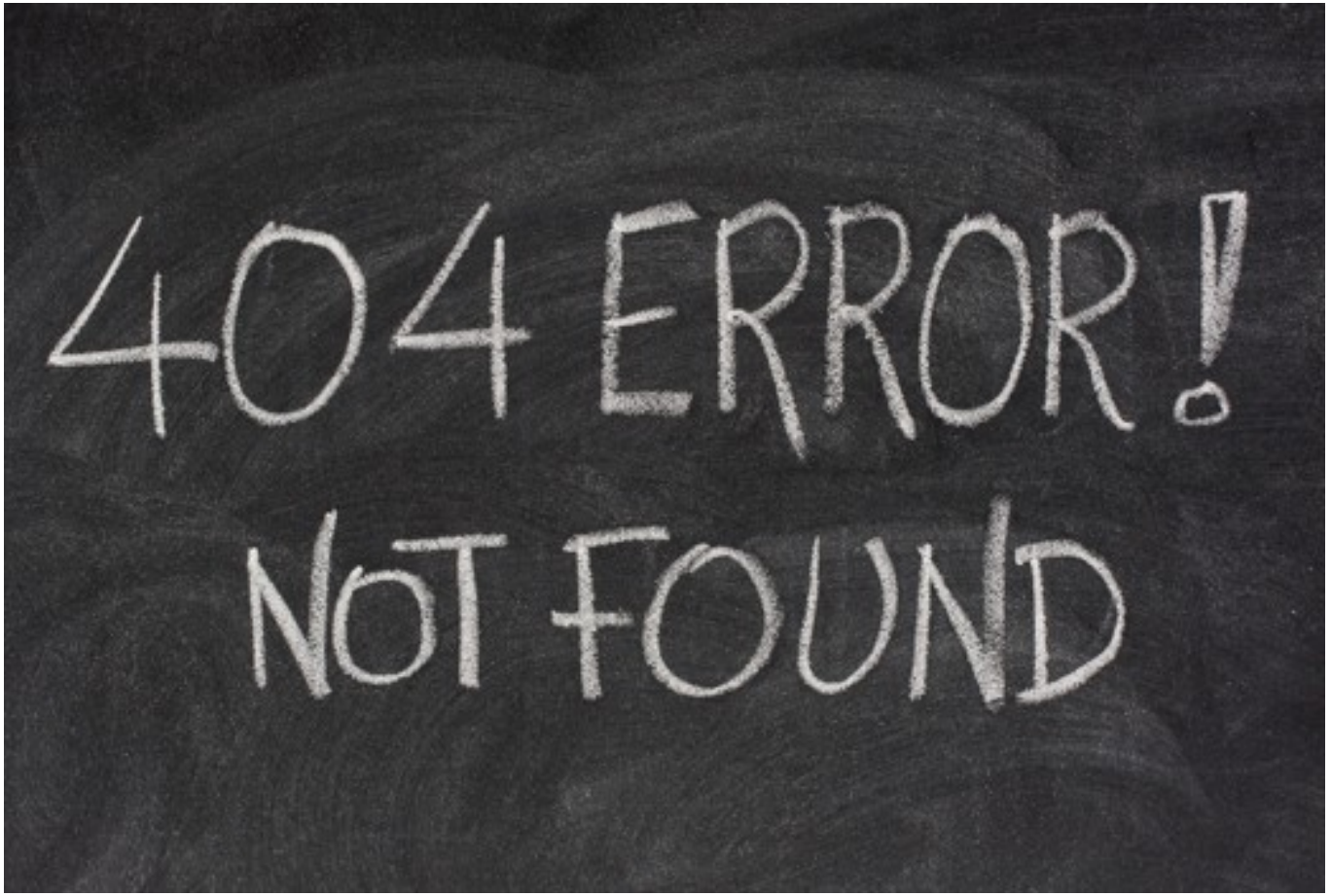
- Kontinuierlicher Filter für bestimmte Frequenzbereiche
- Idealerweise 1 in den gewollten Frequenzen, sonst 0. Sprung hat aber unschöne Eigenschaften, darum kontinuierliche Funktion
- Durchgeführt für die Frequenzbänder: δ, θ, α, β, γ
- Alternativ kann das Signal auch mit einer Fourier-Transformation in den Frequenzbereich überführt werden



Oben: Butterworth frequenzfilter
Unten: Rohes EEGSignal und gefilterte alpha Wellen



5 Merkmalsextraktion



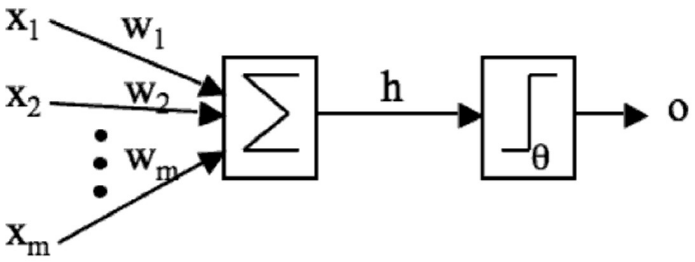
5 Klassifikator

Künstliches Neuronales Netz
- Häufig gewählter Klassifikator [14] [15] [18] [19]
- Gut erforscht
- Sehr gute Toolunterstützung (PyBrain)

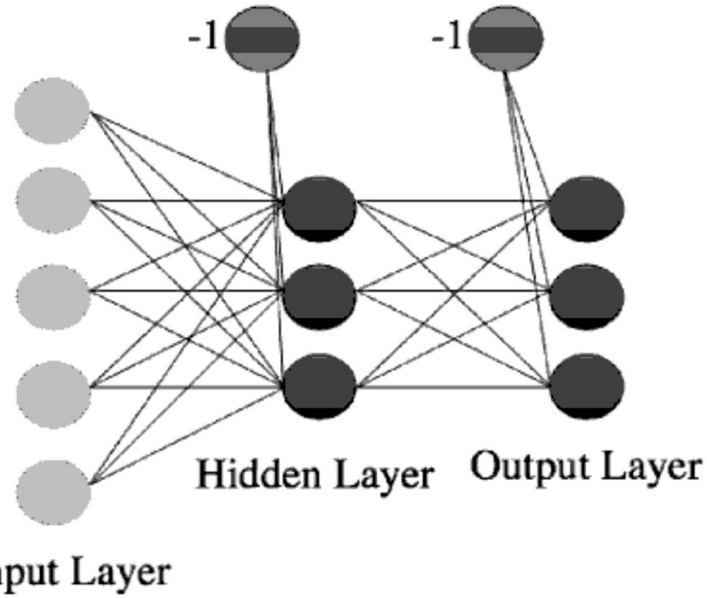
Funktionsweise
X : Eingabevektoren (Merkmalsmenge)
W : Gewichtsvektor
Σ : Skalarprodukt von X und W
θ : Schwellwert
Wenn h > θ feuert Perceptron (o = 1)

Beim Lernen werden die Gewichte immer wieder angepasst, bis die Erfolgsrate stimmt.

Das Perceptron kann einfach Aufgaben (AND) lösen. Doch bei einem logischen XOR klappt es nicht mehr und es müssen weitere Schichten eingezogen werden



Oben: Einfaches KNN: Perceptron
Unten: Erweitertes KNN: Multi Layer Perceptron



6 Ergebnis

- EEG Headset seit 4 Monaten in Reparatur,
  also keine Ergebnisse bei der Müdigkeitserkennung (Anforderungen 1-4)
- Das System portabel (5) und
  durch das Headset auch komfortabler als ein medizinisches EEG (6)
- Die Anwendung ist flexibel und lässt sich in die Fahrsimulatorumgebung einbetten
- Die Anwendung (Module, Klassen, Methoden) ist komplett dokumentiert und
  durch UnitTests abgesichert

7 Ausblick

Sobald das EEG wieder da ist

- werden Testdaten aufgenommen
- Merkmale extrahiert und klassifiziert

Mit dem Ziel Müdigkeit in den Testdaten richtig zu erkennen

Die Ergebnisse werden zu Beginn des WS16 vorgestellt

Im September wird das Projekt im Rahmen der Masterthesis weitergeführt und mit EKG Sensoren erweitert.

8 Master Projekt

- Pflichtveranstaltungen: WVK.15, InfoInside, Studientag
- Erstellen einer Latexvorlage für die Ausarbeitung
- Einführung / Dokumentation des Fahrsimulators
- Support mit CANoe, OpenDS und Emokit
- Kommunikation zwecks EEG Headset Reparatur
  - Februar: Defekt festgestellt, Nachfrage bei Emotiv
  - März: Gerät wurde eingeschickt (Probleme mit dem Zoll)
  - April: Gerät beim TechCenter angekommen
  - Mai: Fehler anscheinend behoben, auf Nachfrage wurde erneut geprüft
  - Juni: Emotiv entschied ein neues Headset zu schicken
  - Juli: Das EEG soll am Montag ankommen

IOT
INTERNET OF THINGS

INF
Informatik

Vielen Dank!
Noch Fragen?