

Abschlussbericht IoT-Projekt

Reutlingen, 28.06.2016

Kenndaten des Projektes

Name:	Paul Pasler
Titel:	Portable System to Detect driver drowsiness with Body Sensors.
Zeitraum:	02.11.2015 – 28.06.2016

Zweckbestimmung

- Anwendung zur Müdigkeitserkennung im Fahrzeugumfeld
- Integration des EEG- und EKG-Sensors in die Simulationsumgebung
- Erkennung von Müdigkeit des Fahrers anhand von EEG- / EKG- Signalen
- Warnungen an den Fahrer bei erkannter Müdigkeit über ein Interface des Fahrsimulator

Bestimmungsgemäßer Gebrauch

- Der Fahrer trägt das **EEG Emotiv EPOC** auf dem Kopf.
- Der Fahrer trägt das **EKG Brustband „Zephyr Bioharnes“** um die Brust
- Die Sensoren sind mit der Anwendung über die Infrastruktur des Fahrsimulators verbunden.
- Der Fahrer fährt im Fahrsimulator und wird bei erkannter Müdigkeit gewarnt

Ergebnisse des Projektes

- Integration: Generische und offene Integration des EEGs in die Infrastruktur des Simulationsumfelds.
 - *Die bereitgestellten Schnittstellen sind ausführlich dokumentiert.*
- Testfahrt: Erstellung und Durchführung von Testfahrten im Simulator und Aufnahme von Testdaten mit übermüdeten Fahrern. Die Testfahrten werden per

Video aufgezeichnet und später manuell mit „Müdigkeitszeichen“ (Gähnen, Kopf fällt nach vorn) markiert. Weiterhin wird das Fahrverhalten zur Analyse hinzugezogen (bspw. Ruckartiges Gegenlenken). Das Erstellen einer Teststrecke im Simulator soll dokumentiert werden.

- *Die Daten können Wiederverwendet werden*
- **Datenverarbeitung:** Analyse und Aufbereitung der EEG / EKG Signale. Geeignete Verfahren sind implementiert und getestet.
 - *Die einzelnen Bausteine der Aufbereitung sind unabhängig, auch in anderen Projekten einsetzbar.*
- **Klassifikator:** Prototypische Implementierung eines Neuronalen Netzes zur Erkennung von Müdigkeit.
- **Dokumentation:** Alle Schritte die den Fahrsimulator bzw. die Sensoren betreffen, sind ausführlich dokumentiert, sodass Folgearbeiten darauf aufbauen können.
- **Support:** „Simulator-Neulinge“ wurden in den Fahrsimulator eingeführt

Ziel ist eine Anwendung zur Erkennung von Müdigkeit, die zumindest die Testdaten richtig klassifiziert und entsprechende Rückmeldung gibt.

Liste der abgegebenen Artefakte

#	Bezeichnung	Ablageort / Hinweis zum Auffinden der Daten
1	Python Anwendung <ul style="list-style-type: none"> • posdbos.py (main-Klasse) • EEG Rohdaten HTTP Server • Lose gekoppelte Util Klassen • Beispiele / Tests / Doku 	https://github.com/ppasler/current-adas/tree/master/project/code
2	Fahrsimulator <ul style="list-style-type: none"> • Testszenario • Änderungen am Sourcecode 	IoT SVN-Repo
3	CANoe (Steuergerät) <ul style="list-style-type: none"> • Einbau der EEG Signale 	IoT SVN-Repo
4	Sim2CAN	IoT SVN-Repo

	• Einbau der EEG Signale	
5	Testdaten / Videos	???
6	Präsentation	https://github.com/ppasler/current-adas/tree/master/project
7	Artikel	https://github.com/ppasler/current-adas/tree/master/project/paper

Installationsanweisung

1. IoT SVN-Repo

1. Auschecken

Das [SVN-Projekt](#) auschecken ([How-To](#)).

Funktioniert nur im Hochschulnetz oder VPN.

Login gibt's beim Inf-Service

2. Software

Dort befindet sich

- Der Fahrsimulator
- Der Sim2CAN Client
- Die CANoe Konfiguration
- Der EmotivePOC Control Panel

2. Anwendung

1. Auschecken

Das [git-Projekt](#) auschecken ([How-To](#))

2. Python installieren

Python Version 2.7.x und folgende Module werden benötigt:

gevent (1.0.2), pybrain (0.3), pycrypto (2.6.1), pygame (1.9.1), scikit-learn (0.17.1), matplotlib (1.5.1), numpy (1.10.4), pywinusb (0.4.0)

Für Windows gibt es Builds hier: <http://www.lfd.uci.edu/~gothke/pythonlibs/>

Oder mit pip: <https://docs.python.org/2.7/installing/index.html>

Die Anwendung nutzt zu diesem Zeitpunkt Testdaten und benötigt kein EEG, im Server-Log taucht die Meldung: „Using 4096 dummy datasets“ auf.

Als leichtgewichtige Entwicklungsumgebung empfiehlt sich [IDLE](#)

3. Testen

In den Ordner current-adas/project/code/src/example oder /test wechseln

Wenn alles geklappt hat, sollten sich Klassen ohne Fehler öffnen lassen.

Dann in src wechseln und Server und Client ausführen. Der Client sollte 3 Calls ansetzen und loggen. Der Server loggt den Antwort-Header.

3. Emotiv EEG installieren

1. Treiber installieren

Das Bluetooth USB-Dongle einstecken und Plug-and-Play

2. Software installieren

Im IoT SVN-Repo das „Emotiv EPOC Control Panel“ installieren

3. EEG vorbereiten

[Kurz-Anleitung](#)

4. Testen

Headset anmachen und Control Panel starten. Die Farbe der Pins sollte sich verändern (Schwarz = kein Signal, Grün = gute Signal). Solange die Position der Sensoren verändern, bis alle Sensoren Grün sind (Das erfordert u.U. etwas Geduld).

Bedienungsanleitung

Anwendung komplett auf einen Rechner

Im Code-Ordner muss die posdbos.py gestartet werden. Die Anwendung beginnt, nach der Initialisierung, damit die Fahrtauglichkeit des Fahrers zu prüfen.

Verteilte Anwendung

Die EEG-Rohdaten werden auf einen andern Rechner, als die Berechnung und Klassifizierung ausgeführt. Dann muss auf dem Roh-Datenrechner der Server gestartet werden. In der Config auf dem anderen Rechner muss die richtige Quellen eingetragen werden und wiederum die posdbos.py gestartet werden.

FAQs

1. IoT SVN-Repo

- **Warum bekomme ich keine Verbindung zum SVN-Repo?**
Funktioniert nur im Hochschulnetz oder VPN
- **Woher bekomme ich einen Login zum SVN-Repo?**
Beim Inf-Service der Hochschule
- **Warum sind im Repo so viele Projekte und wozu sind sie gut?**
Beim Aufbau des Fahrtrainers wurde alle relevanten Projekt in diesem Repo gepflegt. Leider existieren nicht mehr für alle Projekte Dokumentationen.

2. Anwendung

- **Warum startet meine Anwendung nicht?**
Evtl. sind nicht alle Module installiert. Wenn Klasse aus IDLE gestartet wird (öffnen und F5), bleibt die Fehlermeldung erhalten und muss dann näher analysiert.
- **Wo befindet sich die „main“-Klasse für die Anwendung? Welche Klassen müssen gestartet werden?**
Läuft die gesamte Anwendung auf einem Rechner muss lediglich die posdbos.py gestartet werden. Laufen Roh-Datensammlung und Berechnung auf verschiedenen Rechnern, muss die passende Quelle und Empfänger gestartet werden (bspw. Client und Server).

3. Emotiv EEG

- **Warum sind nicht alle Sensoren Grün?**
Das Signal ist dann zu schwach. Die Signalqualität schwankt von Schwarz (= kein Signal) über Rot und Orange bis zu Grün (gutes Signal). Siehe auch „Trouble Shooting“

Trouble shooting

„Have you tried turning it off and on again?“

- *Das Einstellen der EEG Sensoren (alles Grün) erfordert ein wenig Geduld. Oftmals hilft es die betreffenden Sensor-Filze mit mehr Flüssigkeit zu versehen (richtig nass!). Bei dichtem Haar müssen die Sensoren evtl etwas zu Seite geschoben werden. Positionsveränderungen und Druck auf den Sensoren sollten zu einer Veränderung führen.*