The Myoelectric Personal Training and Control Environment (MPTCE)

Martín García Carmueja – FH Campus Wien
Masterstudiengang Health Assisting Engineering

CD-Labor für Wiederherstellung von Extremitätenfunktionen

FH CAMPUS WIEN

MEDIZINISCHE UNIVERSITÄT WIEN

ZENTRUM FÜR MEDIZINISCHE PHYSIK UND BIOMEDIZINISCHE TECHNIK
CENTER FOR MEDICAL PHYSICS AND BIOMEDICAL ENGINEERING

MPBT

Überblick

- Einführung
  - Zum Auftraggeber
  - Elektromyographie
  - Biofeedback Training
  - Das Aktuelle System
  - Ziel des Projektes
- Methode
  - Mustererkennung
  - Schwellenbasierte Aktivierung
  - Entwicklungsmethode
  - Struktur der Software
  - Designlösungen
  - Testverfahren
- Ergebnisse
  - Demonstrationsvideos
  - Leistung
- Diskussion und Zukünftige Arbeit

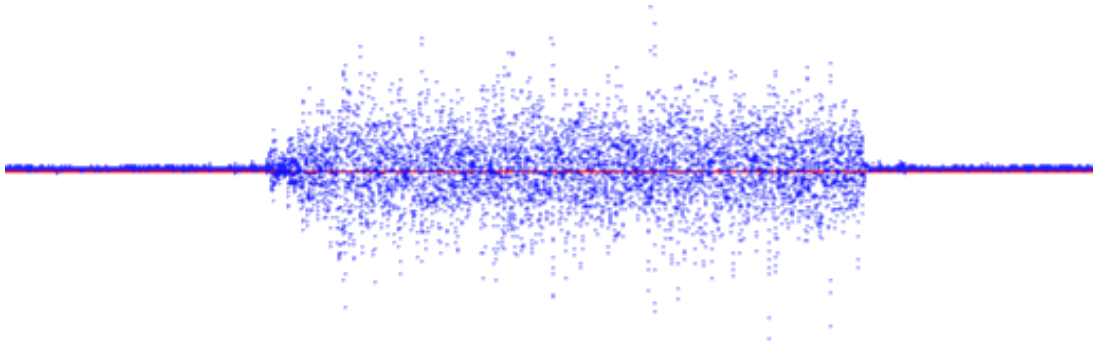
Zum Auftraggeber

Christian Doppler Labor für Wiederherstellung von Extremitätenfunktionen

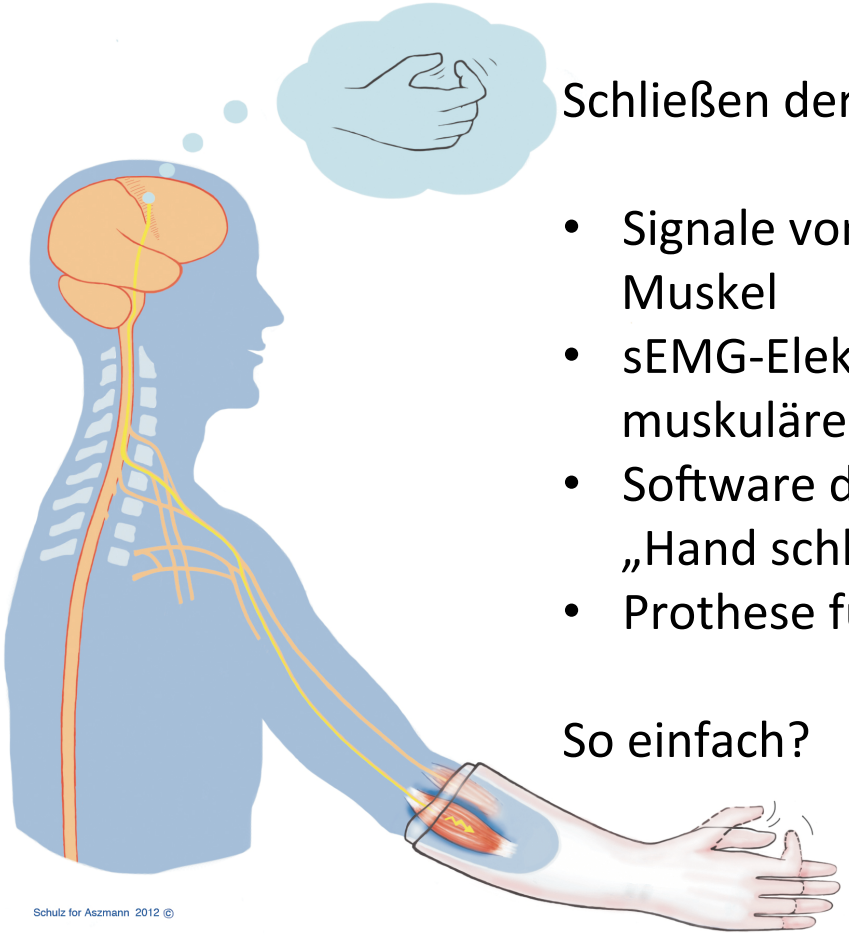
- Widerherstellung der Funktion der oberen Extremität
- **sEMG**-basierte Prothesen
- Natürliche Interaktion mit der Umgebung
  - Mehrere Freiheitsgrade
  - Simultane Ausführung mehrerer Bewegungen
- Evaluierung verschiedener **mustererkennung**sbasierten Steuerungsalgorithmen
- Forschungsergebnisse werden an Rehabilitationspatienten angewandt.

Elektromyographie (EMG)

- Erfassung der elektrischen Impulse, die durch die Aktivität von skelettalen Muskeln produziert werden.
  - Invasiv: Nadel[1]- und Fadenelektroden -> Aktivität von Muskelfasern (einzelne/kleine Gruppen).
  - Nicht invasiv: Oberflachenelektroden[2] -> Aktivität von ganzen Muskeln. **Surface Electromyography (sEMG)**



sEMG-basierte Prothese



Schließen der Hand:

- Signale vom motorischen Cortex erreichen den Muskel
- sEMG-Elektroden der Prothese erfassen die muskuläre Aktivität
- Software der Prothese erkennt die Aktivität als „Hand schließen“
- Prothese führt die Bewegung aus

So einfach?

Jede(r) PatientIn ist anders
Jede Prothese ist anders
Trainingsprozess ist notwendig!

[4]

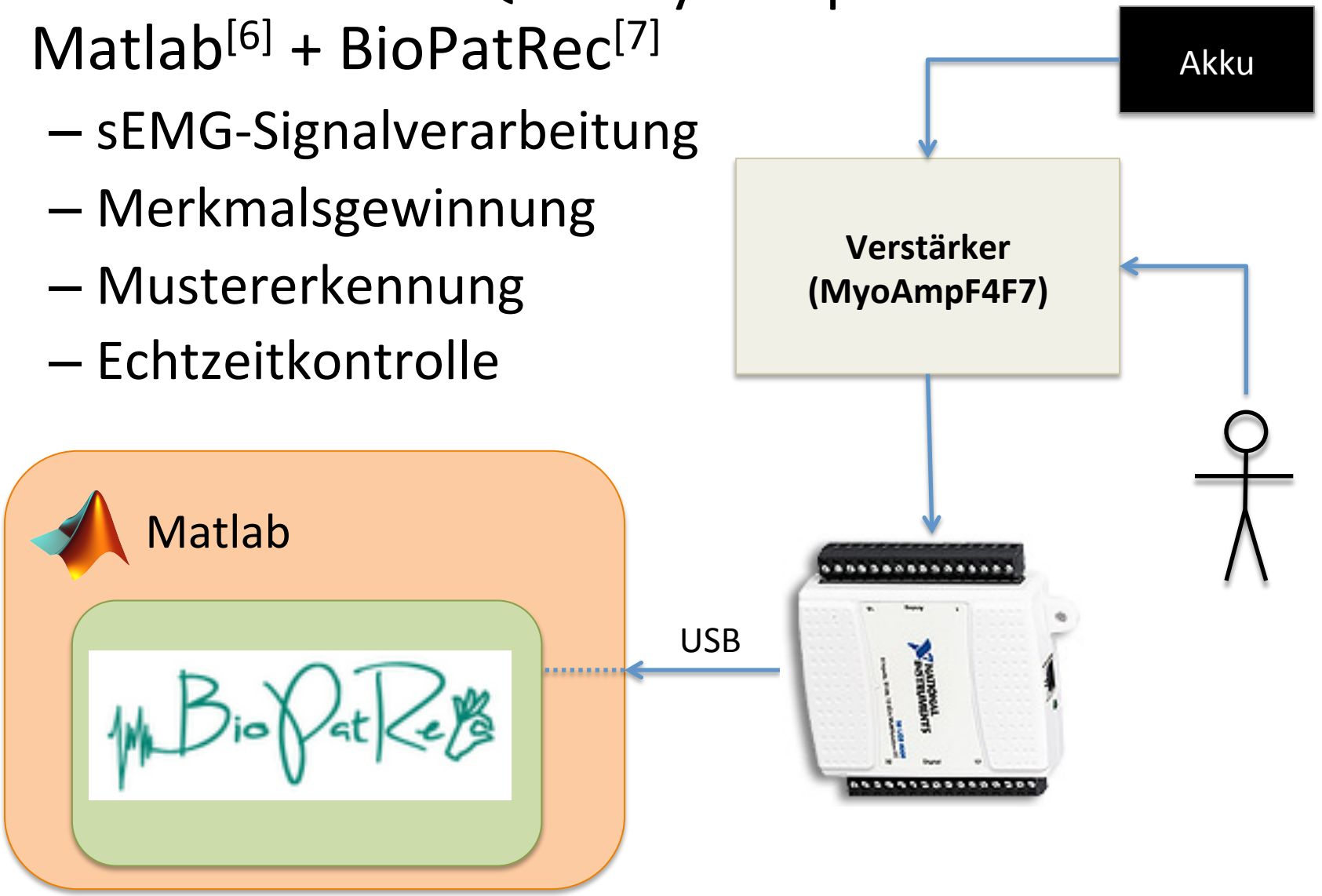
Biofeedback Training

- Lernen, Körperfunktionen zu kontrollieren.
  - Körpersignale durch Geräte aufgenommen
  - Feedback-Signale werden generiert

- Anpassung von sEMG-steuerten Prothesen
  - Neuromuskuläre Kontrolle
  - So viel Training wie möglich
    - Virtuelle Prothese
    - Videospiele

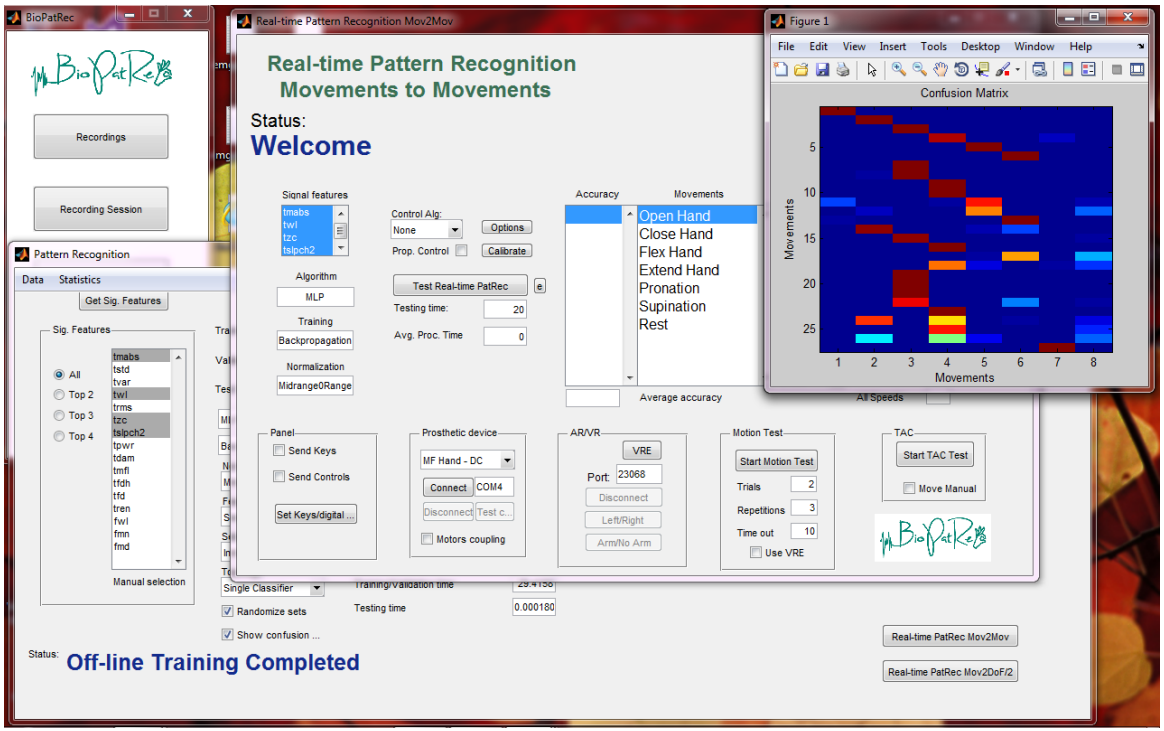
Das aktuelle System

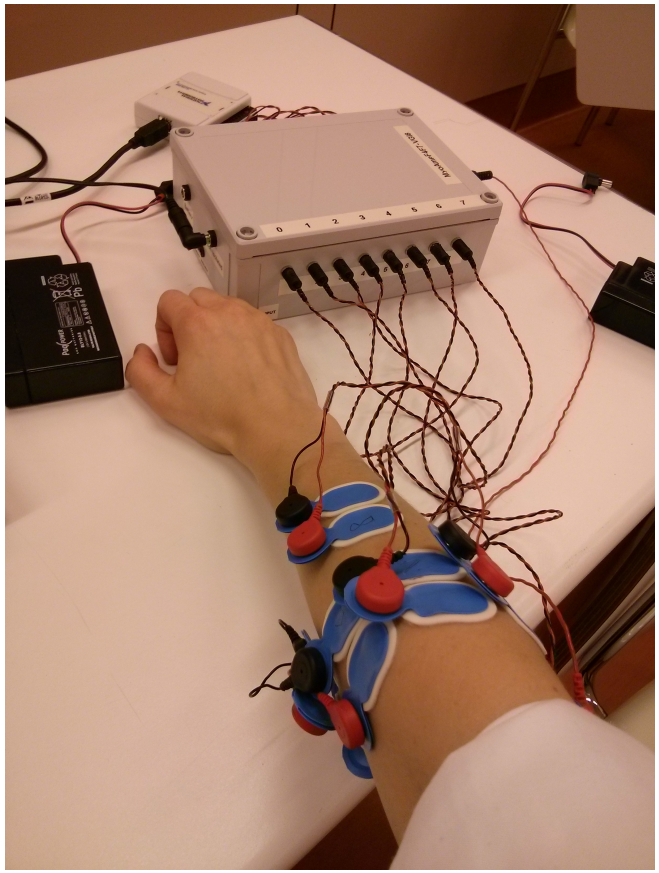
- NI USB-6009 DAQ[5] +MyoAmpF4F7-VGI8 + Matlab[6] + BioPatRec[7]
  - sEMG-Signalverarbeitung
  - Merkmalsgewinnung
  - Mustererkennung
  - Echtzeitkontrolle



Das aktuelle System

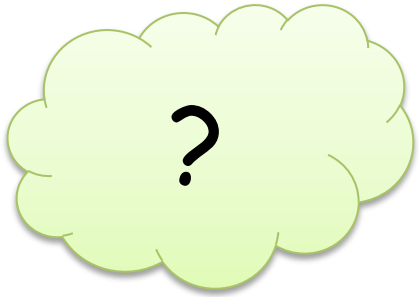
- Nicht geeignet für ein autonomes Training
  - Aufbau der Hardware ist kompliziert
  - Nicht einfach bedienbar
  - Erfordert Matlab (€€€)





Ziel des Projektes

- Entwicklung eines neuen Systems
  - Neue, kompakteren Hardware (vorhanden!)
    - TI-ADS1298-basiertes USB-Erfassungsgerät
    - Entwickelt vom Zentrum für Medizinische Physik und Biomedizinische Technik (MedUniWien)
  - Neue Software
    - Ermöglicht Training in der privaten Umgebung
    - Einfach bedienbar
    - Windows, keine weiteren Anforderungen
    - Einfach zu erweitern und modular! (C#)



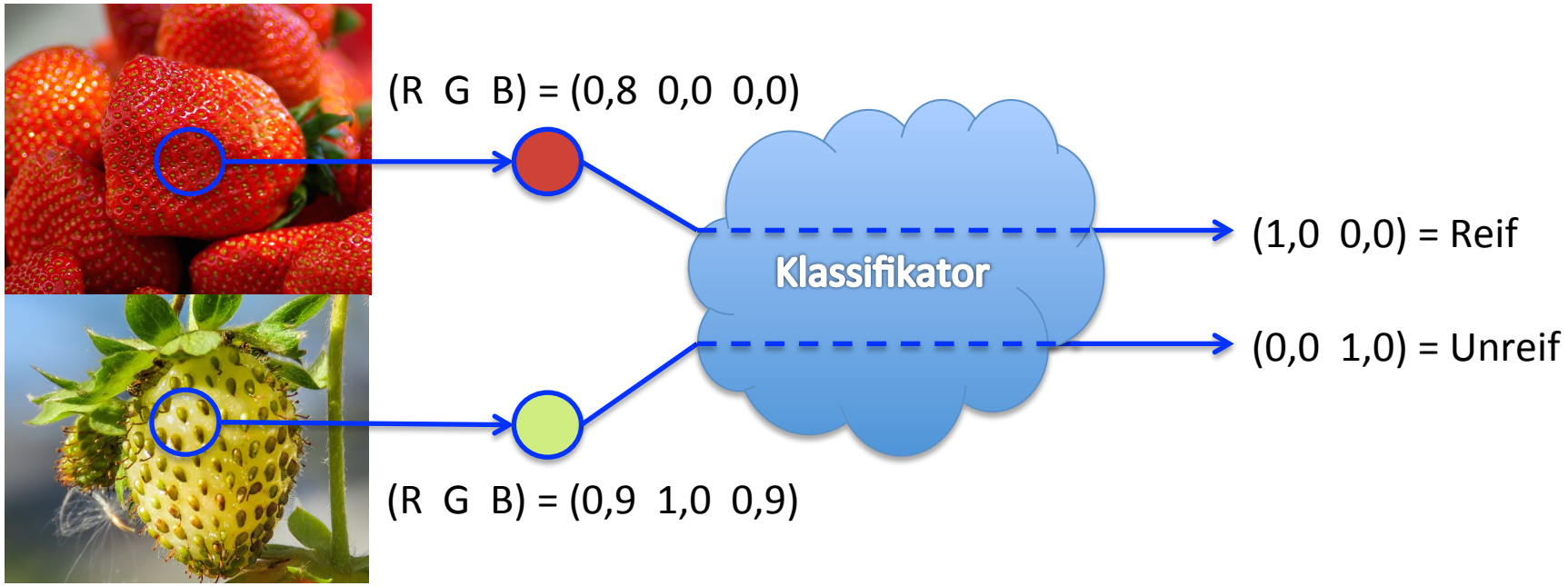
- Nicht-Ziele
  - Portierung von BioPatRec auf C#
  - Eine reine sEMG-basierte Steuerungslösung

Funktionalitäten

- Multiprozessorfähiges Design
- Aufnahme von EMG-Signalen
  - 8 Kanäle, 2 KHz Musterfrequenz
  - Automatische Speicherung nach der Aufnahme
- 6 einfache Bewegungen der Hand zu erkennen
  - open, close, flex, extend, pronation, supination
  - Mögliche Kombinationen von 2 und 3 simultane Bewegungen
- Merkmalsgewinnung und Training für mustererkennungsbasierte Bewegungserfassung
  - Lineare Diskriminanzanalyse (LDA)
  - Multilayer Perceptron (MLP)
- Schwellenbasierte Bewegungserfassung
- Echtzeitbetrieb
  - Darstellung der EMG-Aktivität
  - Generierung von Tastaturevents zur Steuerung anderer Software (z. Bsp. Spiele)

Mustererkennung

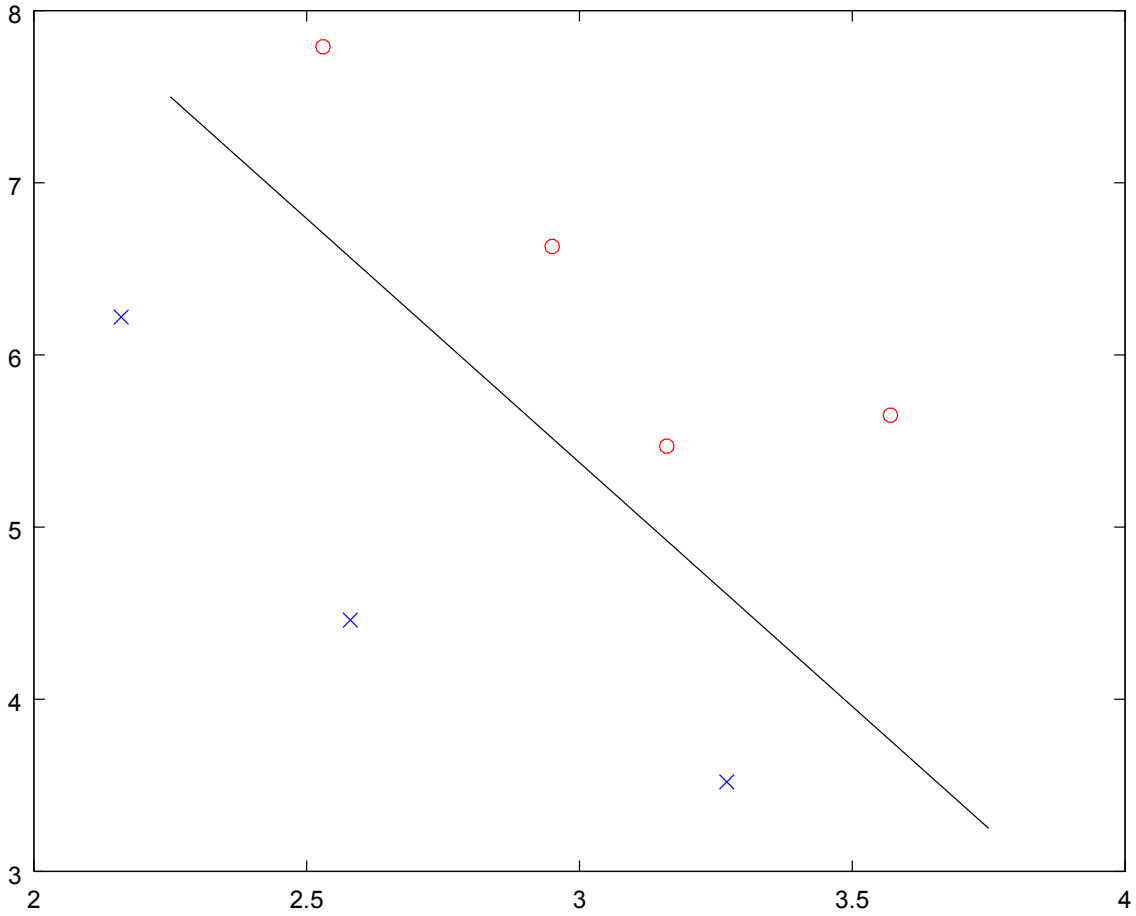
- Klassifikatoren



  - Merkmalsvektor als Eingang
  - Klassifikationsvektor als Ausgang
  - Initialisierung (Training) erforderlich

Mustererkennung: LDA

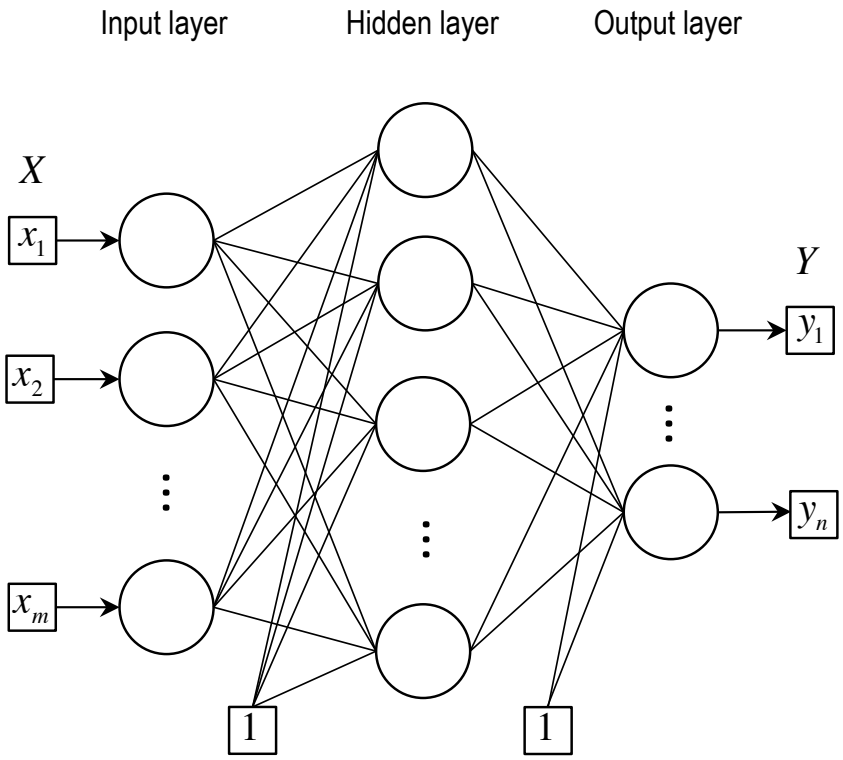
- Statistische Methode
- **Lineare** Funktion von Merkmalen als Grenze zwischen jedem Paar von Klassen

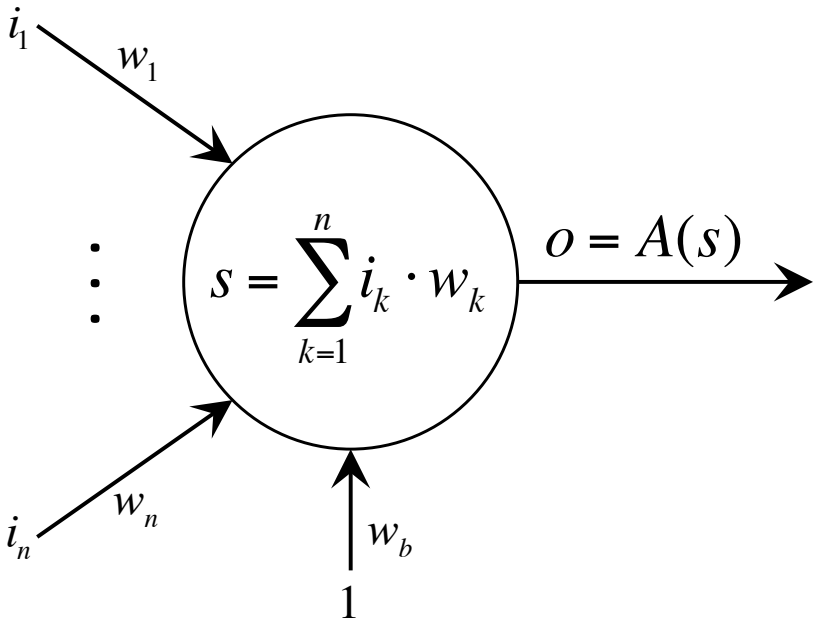


[8]

Mustererkennung: MLP (I)

- Multilayer Perceptron: Künstliches neuronales Netz
- Mind. 3 komplett miteinander verbundener Schichten künstlicher Neuronen
- Encog Machine Learning Framework 3.3 (C#, Apache public license)

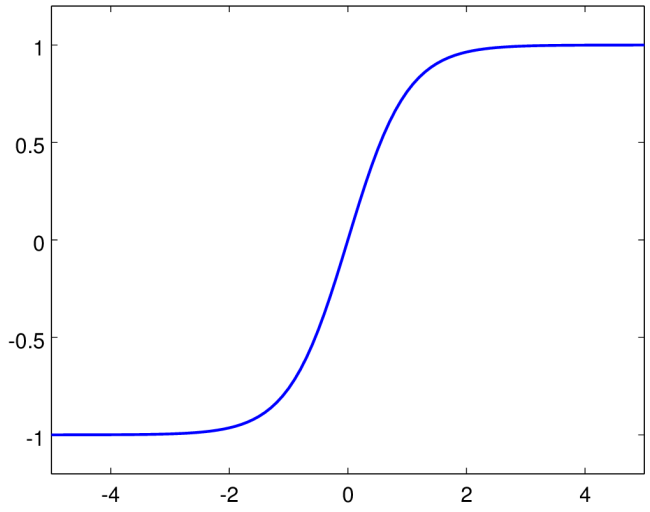




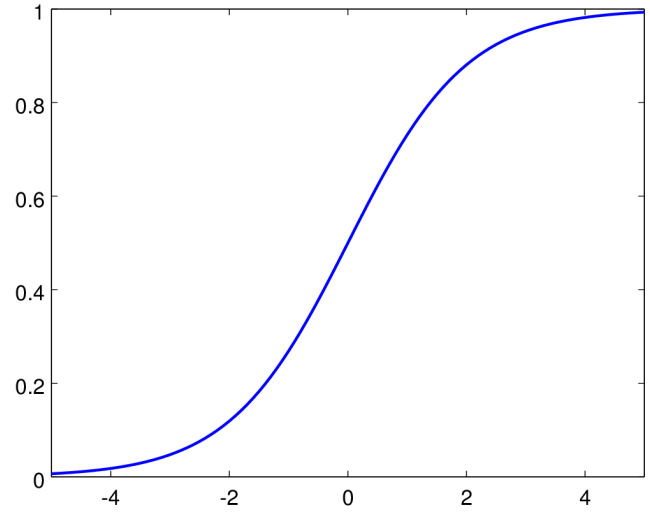
Mustererkennung: MLP (II)

Aktivierungsfuktionen *A(s)*

- Tangens hyperbolicus[9]:



- Sigmoidale[10]:

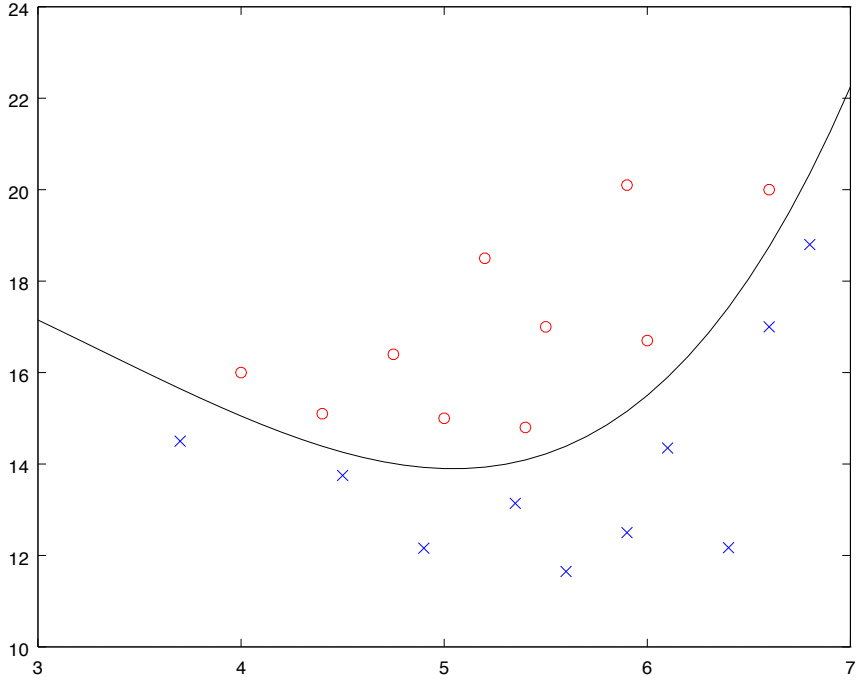


Andere: SoftMax [0,1], Logarithmische [-∞,+∞]

Mustererkennung: MLP (III)

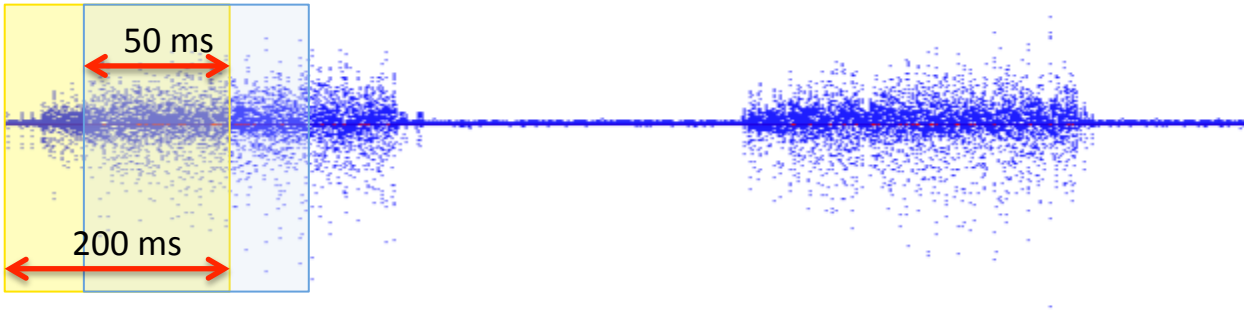
- Training:
  - MLP wird mit Training-Set am Eingang und den erwarteten Klassifikationsergebnisse konfrontiert.
  - Adjustierung der Werten von w_i in jedem Neuron, um Klassifikationsfehler zu minimieren.
  - Validierung des Trainings gegen unbekannte Mustervektoren.

- Erkennung von nicht-linearen Grenzen zwischen Klassen
- Eingangsmerkmale müssen normalisiert werden (Üblicherweise [0,1] oder [-1,1])



Mustererkennung: Merkmale

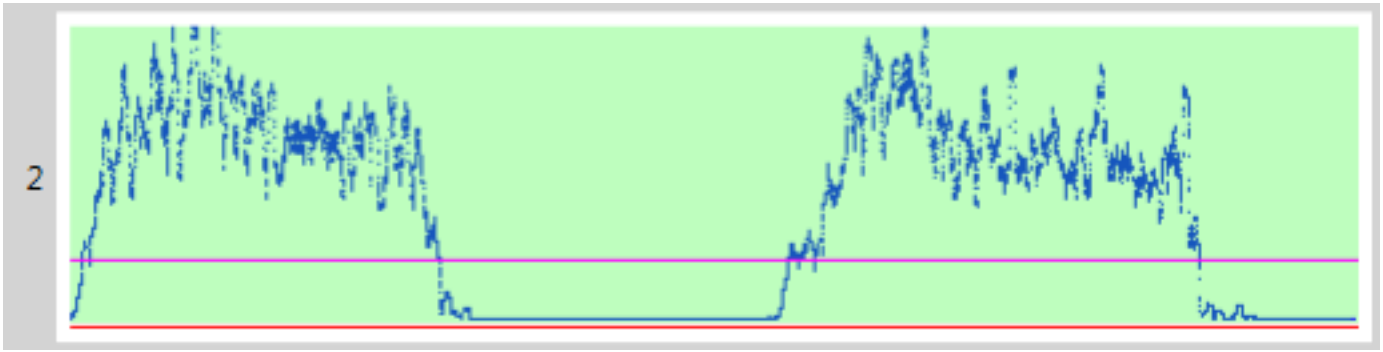
- Skalare Parameter
  - Auf überlappenden Zeitfenstern berechnet



- 5 Merkmale werden unterstützt:
  - Mittelwert (*mean*)
  - Absoluter Mittelwert (*tmabs*)
  - Anzahl der Nullachsenüberquerungen (*tzcs*)
  - Länge des Kurvenlaufes (*twl*)
  - Anzahl der Änderungen des Steigungszeichens (*tslpcs*)

Schwellenbasierte Aktivierung

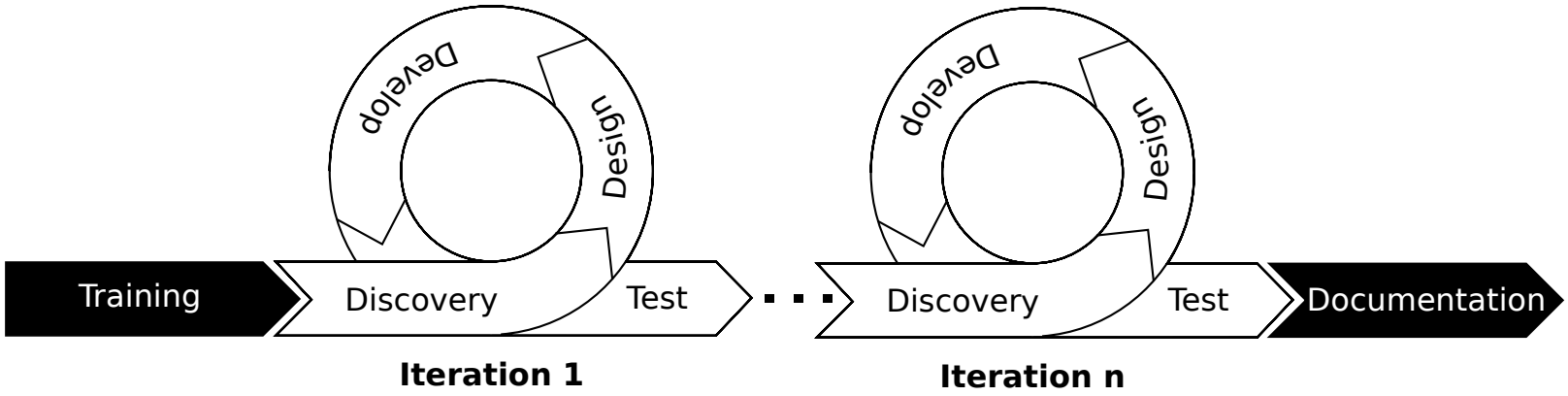
- sEMG-Signal im positiven Bereich umwandeln (Gleichrichtung)
- Glättung (Root Mean Square)
- BenutzerIndefinierte Schwelle
- Überschreitung der Schwelle generiert eine Aktivierung
- BenutzerIndefinierter Bewegungscode



Methode (I)

- Kleines und individuelles Entwicklungsprojekt
- Begrenzte Entwicklungszeit (< 1 Jahr)
- Relativ einfaches System
- Anforderungen sind nicht komplett bekannt am Beginn des Projektes
- Eine Expertin des CD-Labors verfügbar für Zusammenarbeit als Kundin
- Keine vorherige Erfahrung mit C# und der Entwicklungsumgebung

Methode (II)

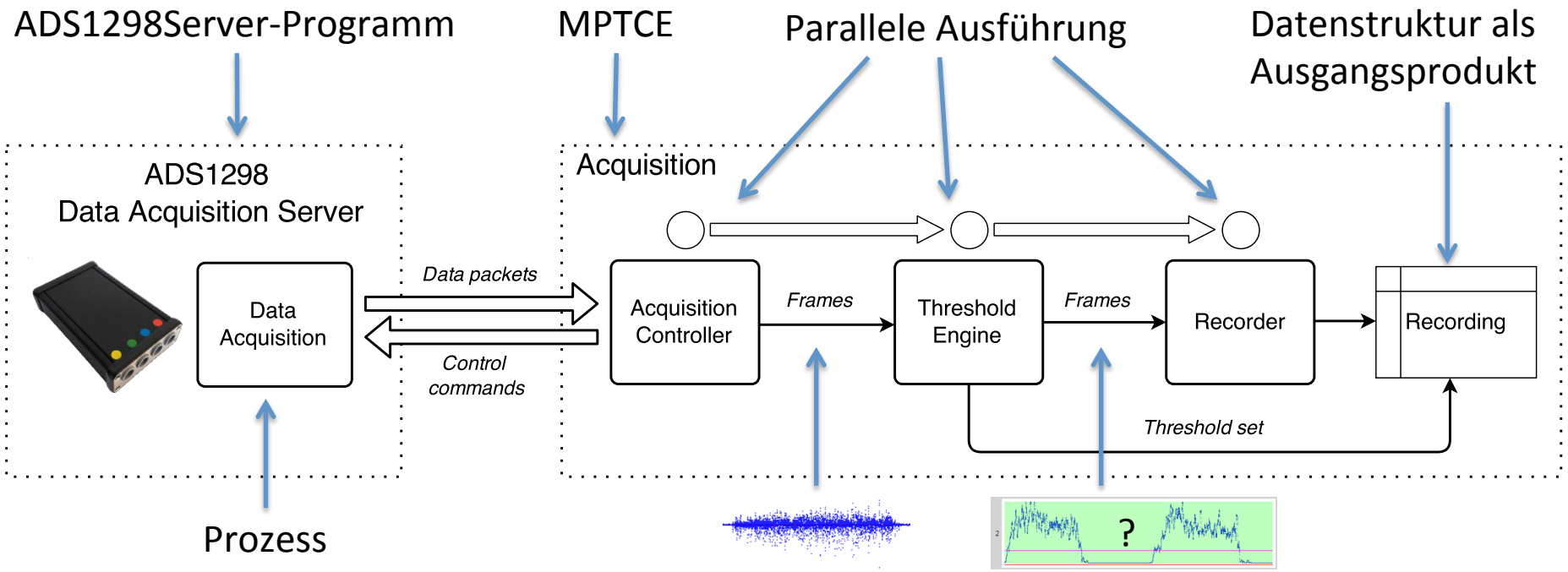


- Agiler Ansatz
  - Inkrementelle Entwicklung: Ein Release am Ende jeder Iteration. Häufige Releases.
  - Dokumentation: nur was notwendig ist
  - Einbeziehung der Kundin: periodisches Treffen
    - Bewertung der aktuellen Iteration
    - Erhebung von neuen/detaillierteren Anforderungen
    - Diskussion von Prototypen
    - Priorisierung

Struktur der Software (I)

- 4 allgemeine Funktionalitäten identifiziert
  - Datenerfassung (Acquisition)
  - Verarbeitung (Treatment)
  - Training
  - Echtzeitbetrieb (Realtime)
- 4 entsprechende Module am höchsten Abstraktionsniveau wurden definiert

Struktur der Software (II) - Acquisition



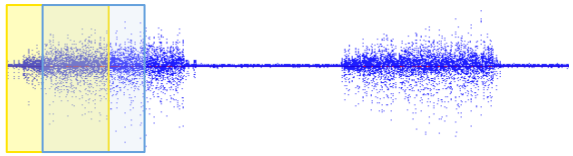
Data packet: Kommunikationseinheit vom ADS1298 Server

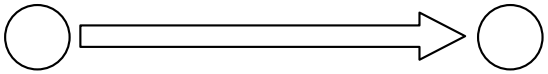
- 1 Data Packet = rohe Daten von mehreren Frames

Frame: Die kleinste Informationseinheit des Programms

- 1 Muster pro Kanal
- Muster gleichzeitig vom Gerät erfassen
- 1 Sequenznummer + 1 Zeitmarke
- Bewegungscode
- ...

Struktur der Software (III) - Treatment





Recording

Pretreat

Frames

Window Maker

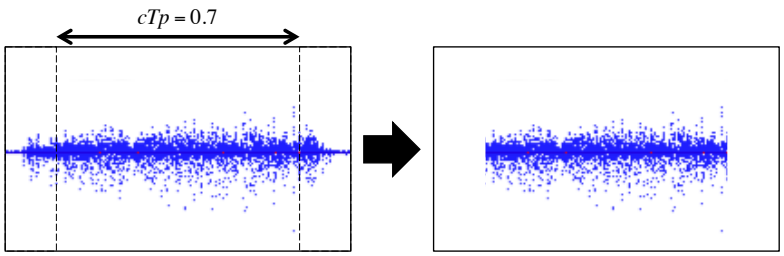
Windows

Feature Extractor

Windows + features

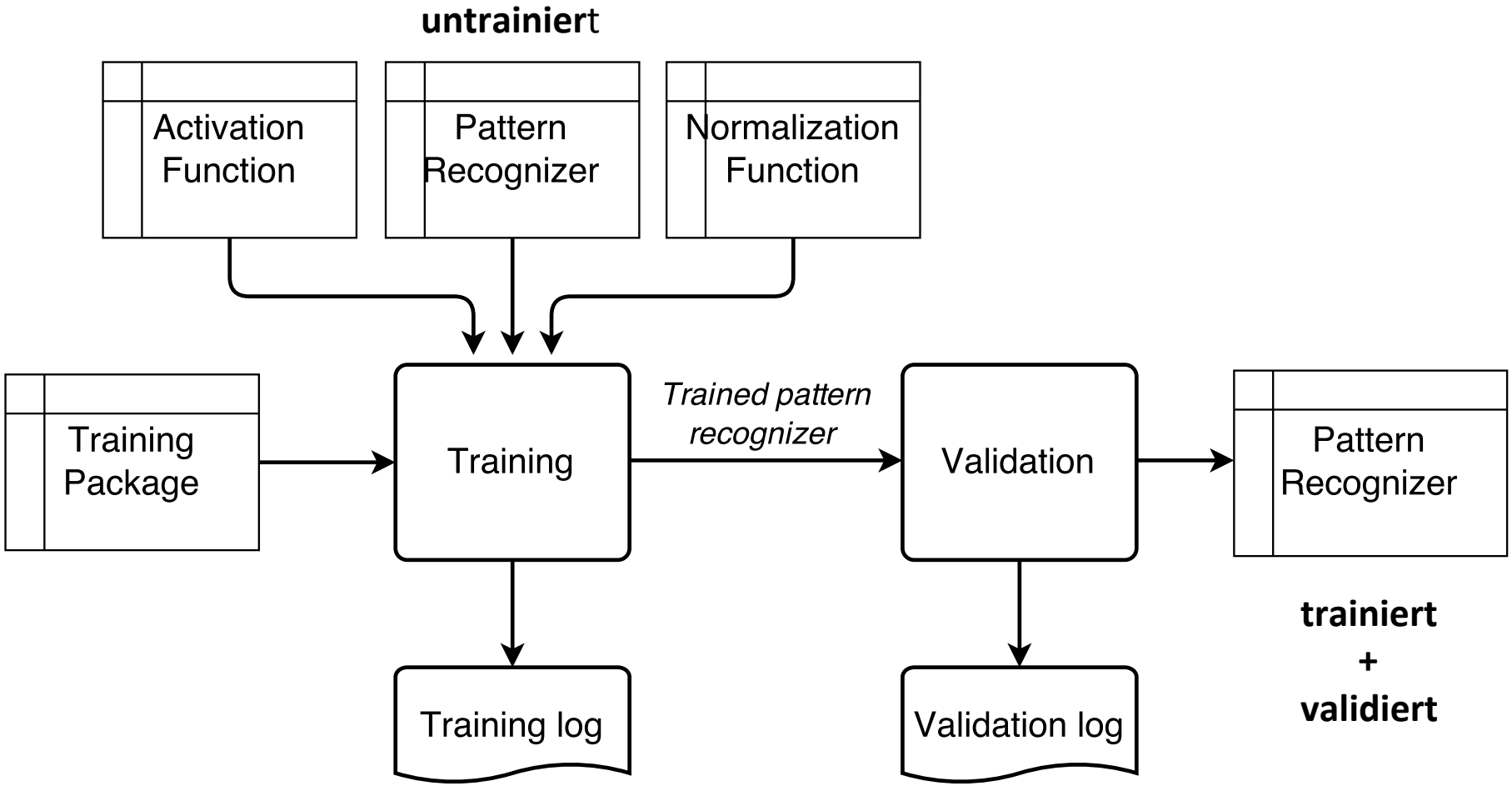
Training Package Builder

Training Package

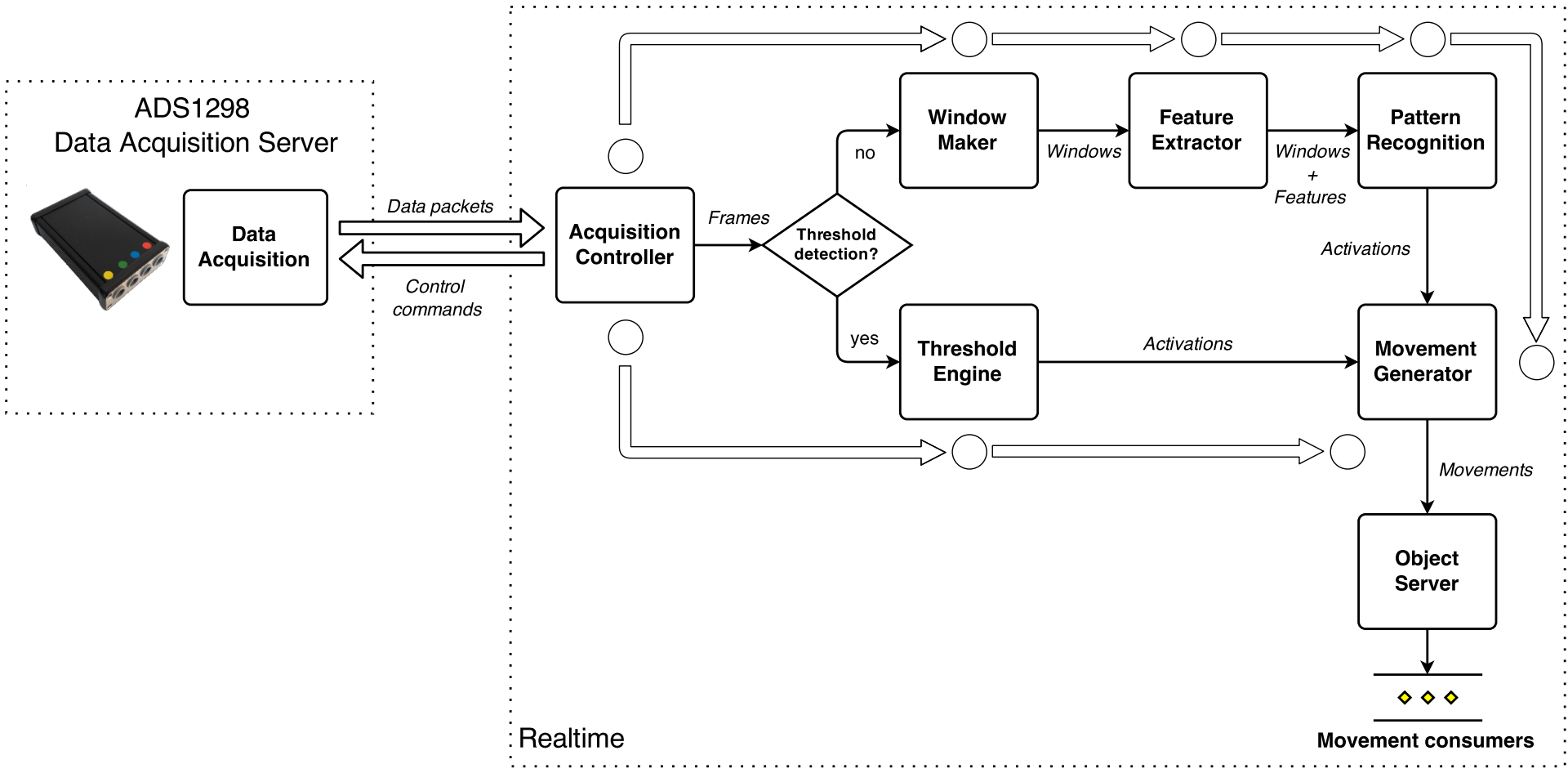


Pro Bewegung:
1 Trainingsset + 1 Validierungsset

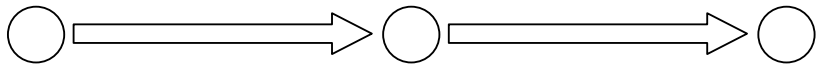
Struktur der Software (IV) - Training



Struktur der Software (V) - Realtime



Designlösungen (I) – Parallele Ausführung

Pipeline (Rohrleitung) 

- Mehrere Elemente gleichzeitig verarbeitet
- Wie eine Fabrikationsstraße: mehrere *Etagen*
  - Ein Produkt jederzeit in einer anderen Etage
- Mit mehreren Prozessorkernen:
  - Tätigkeiten zwischen Kernen verteilt
  - Pipeline nur so langsam wie die langsamste Etage
- Pipelines brauchen Kommunikationslogik! → „Schlangen“ zwischen den Etagen
  - Langsamer als ohne Pipeline in einem Einzelkerncomputer.
  - Aber Mehrkerncomputer sind schon überall! (Smartphones, Desktops, Laptops, Tablets...)

Designlösungen (II) – Erweiterbarkeit

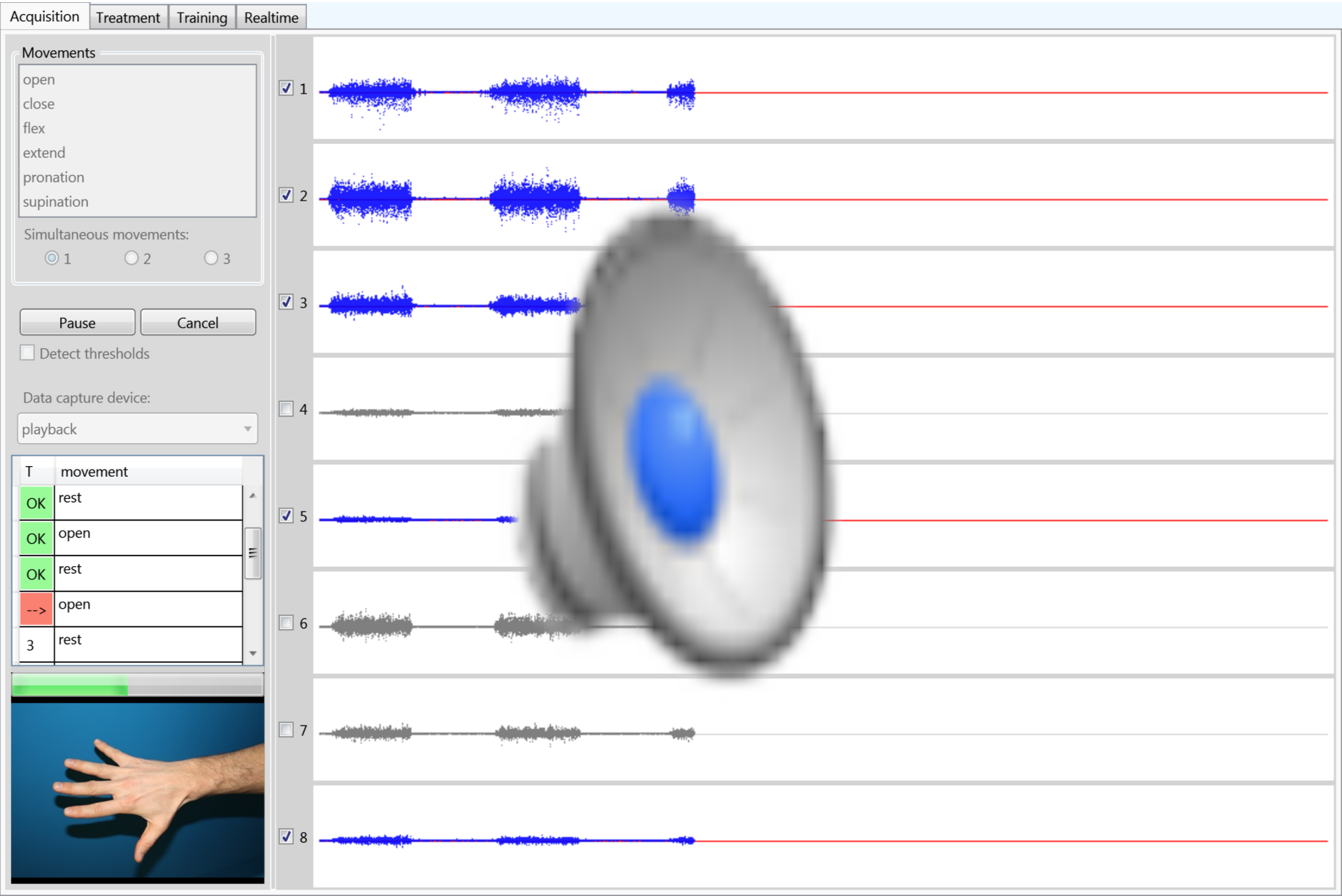
Generic Factories

- Produktion von Objekten auf Basis eines Kataloges
- Eine Generic Factory wird bei der ersten Nutzung mit einem Objekttyp verbunden
  - Aber Objekttypen können Subtypen haben!
  - Eine Factory kann dann Objekte von einer Hierarchie Objekttypen produzieren
  - Den Katalog wird in Laufzeit erzeugt
  - Der Katalog kann der Programmierer jederzeit von der Factory anfordern
  - Die Namen im Katalog können menschenfreundlich sein
  - Der/die BenutzerIn kann die Namen als Liste in der Schnittstelle bekommen und auswählen
- Beispiele: Klassifikatoren, Datenerfassungsobjekte, Aktivierungsfunktionen, Normalisierungsfunktionen, Datei-Leser/ Schreiber, usw. → alles, was austauschbar/erweiterbar sein sollte.
- Neue Subtypen werden nach ihren Definition im Code automatisch erkannt.

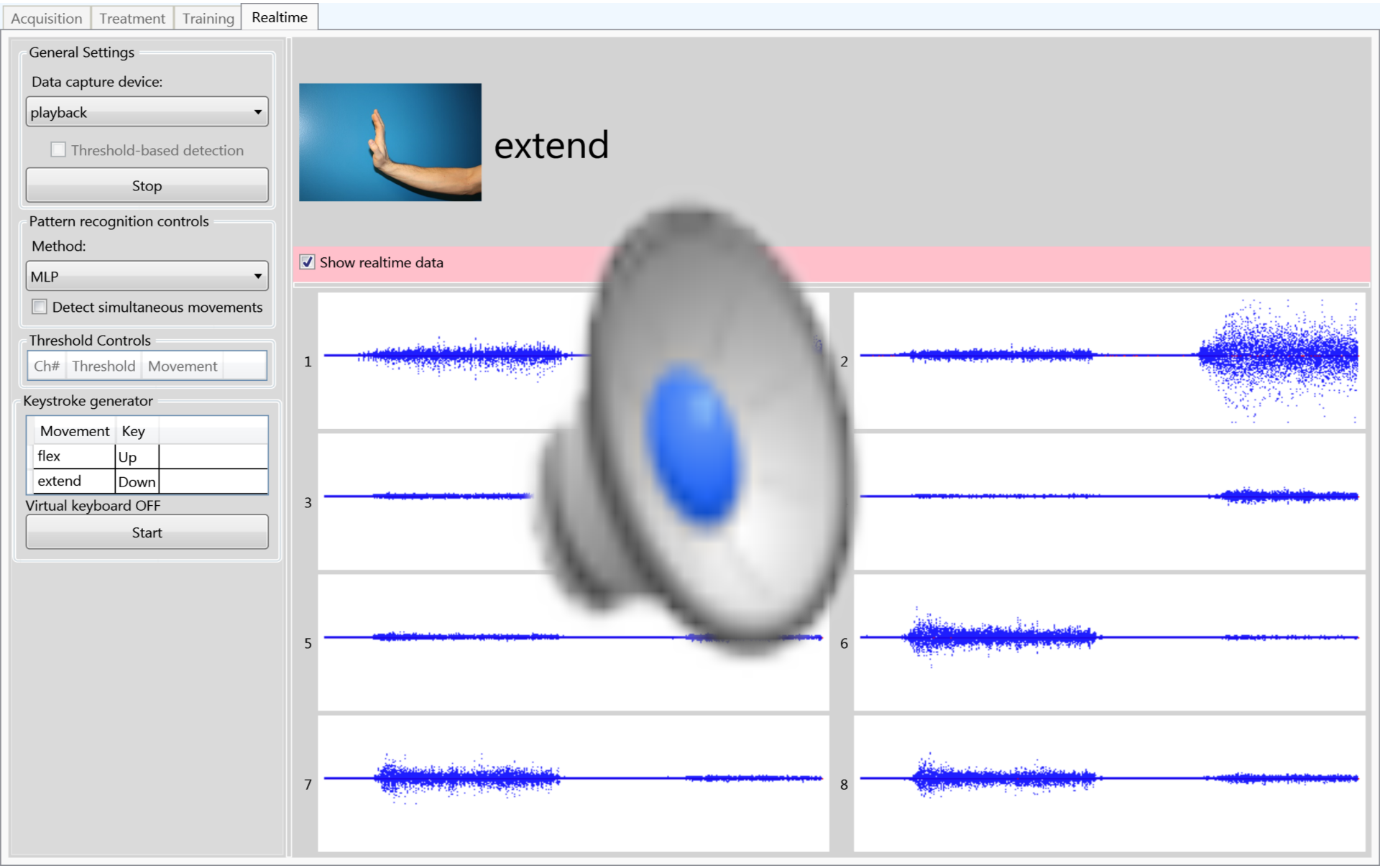
Testverfahren

- Komponententest
  - Manuell, verhaltens- statt datenorientiert
  - Eigene (einfache) Testinfrastruktur
- Integrationstest
  - Mehrere Komponenten arbeiten zusammen
  - Vereinfachtes Szenario oder direkt gegen Hauptprogramm
- Akzeptanztest
  - Nach jedem Release: Treffen mit der Kundin
  - Auch als Fortschrittsbericht
- Letzter Usability-Test
  - Geplant: Mitte/Ende Juli 2015
  - Kundin + zusätzliche Rehabilitationsexpertin

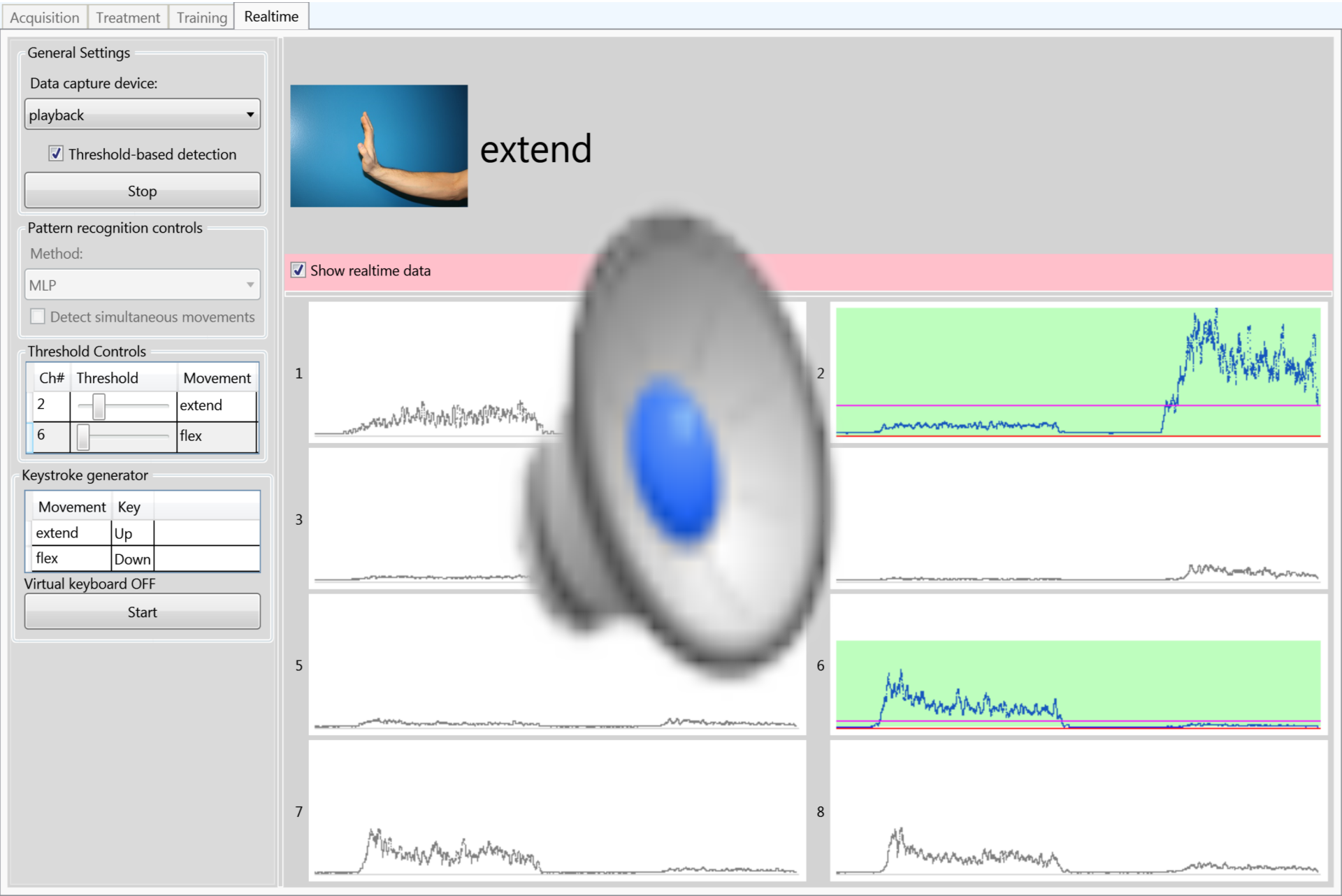
MPTCE Mustererkennung-Walkthrough



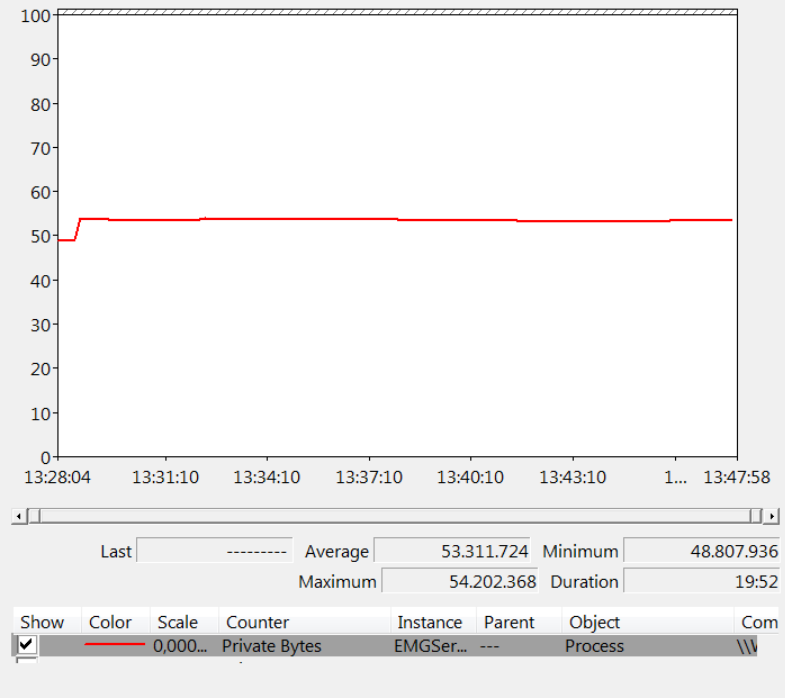
MPTCE Mustererkennung – Echtzeitkontrolle

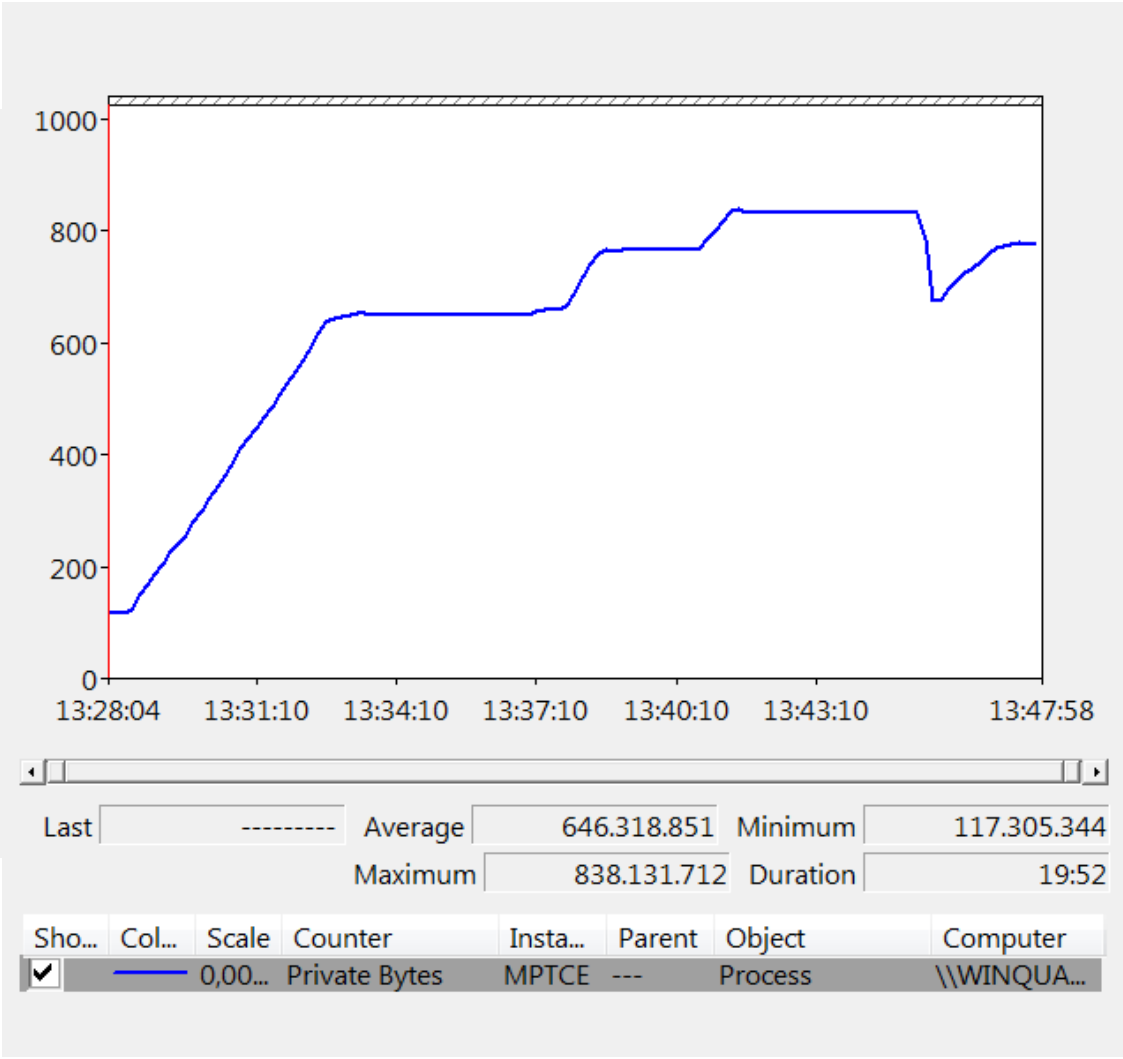


MPTCE Schwellenbasierte Aktivierung



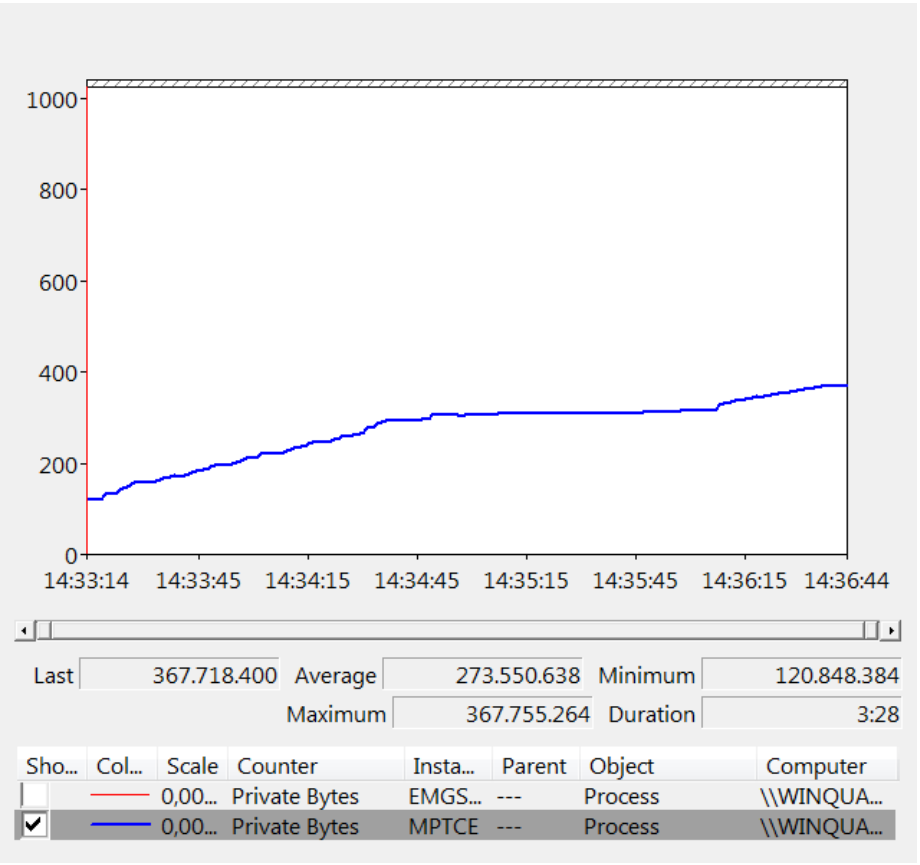
Leistung: Speicherverbrauch (I)



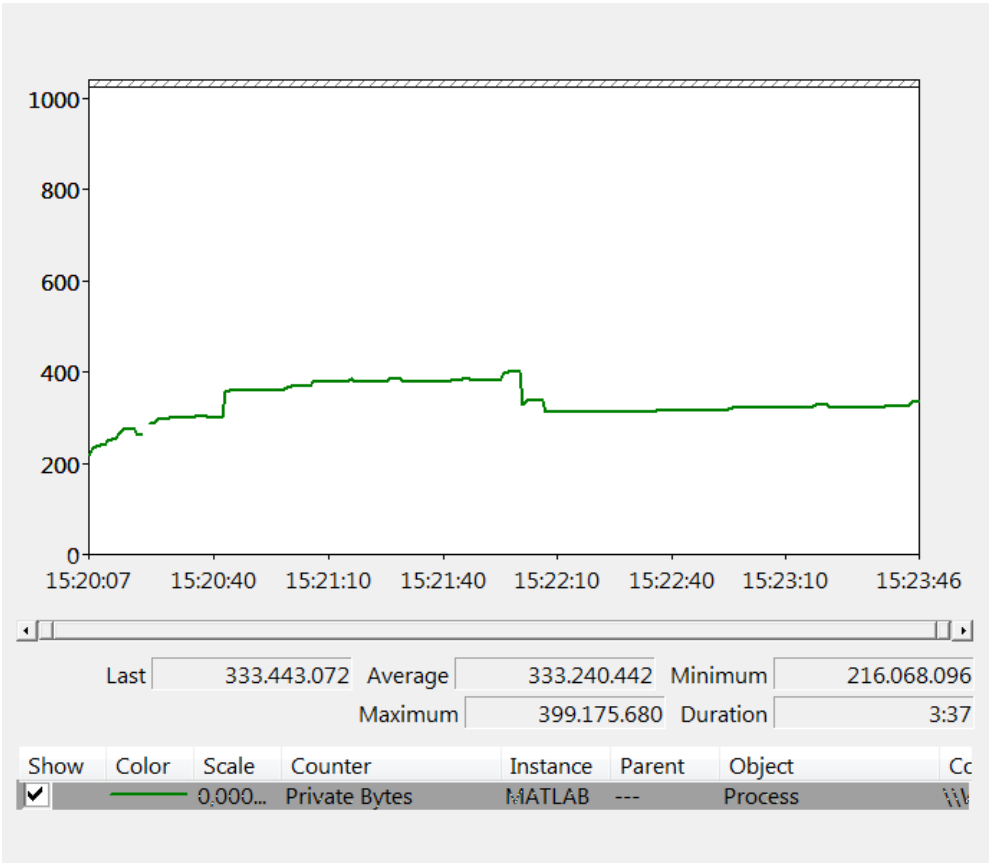


Leistung: Speicherverbrauch (II)

MPTCE

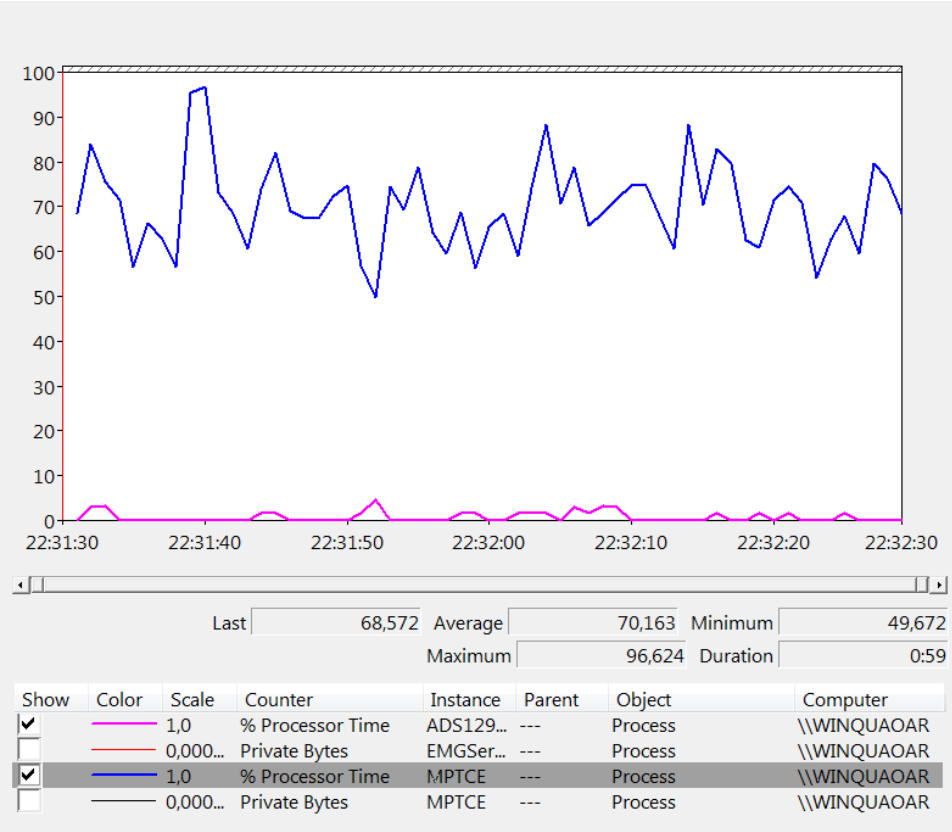


Matlab + BioPatRec

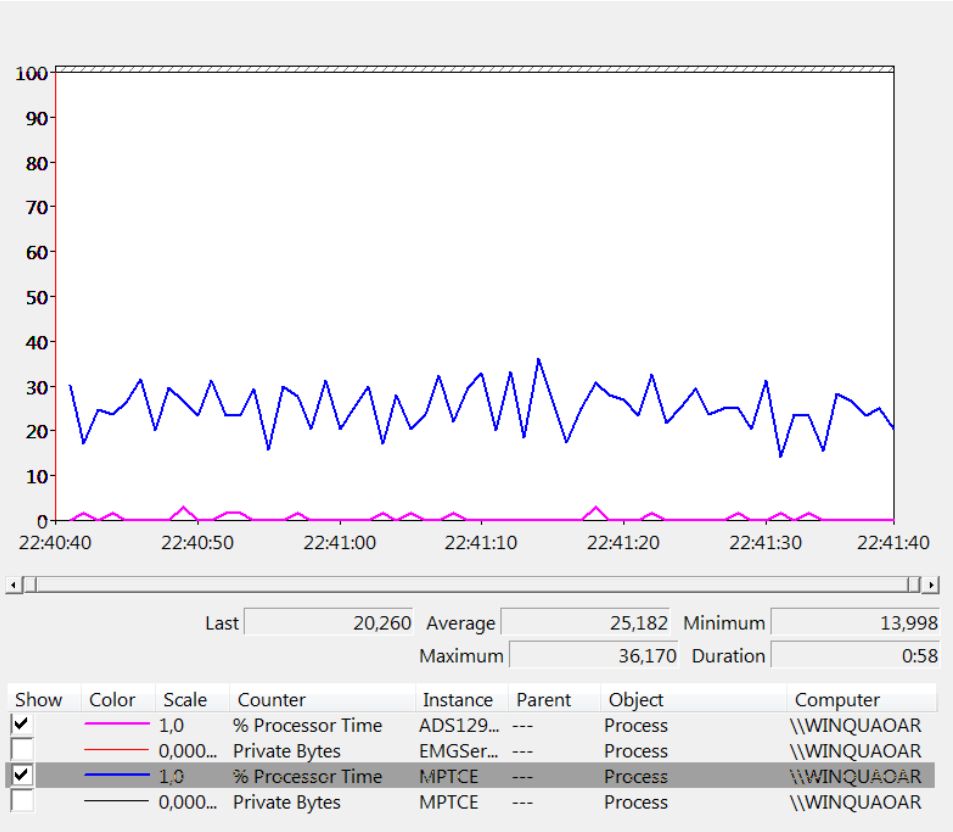


Leistung: Prozessorverbrauch (MLP)

Eine Minute Echtzeitbewegungserfassung



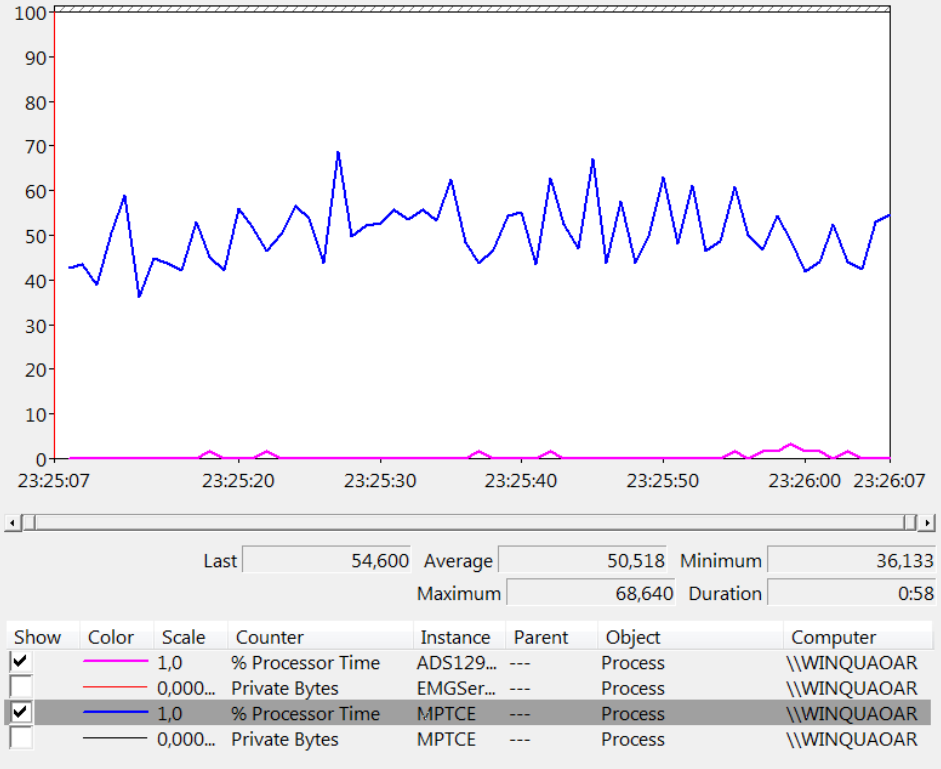
Mit Echtzeitgrafiken



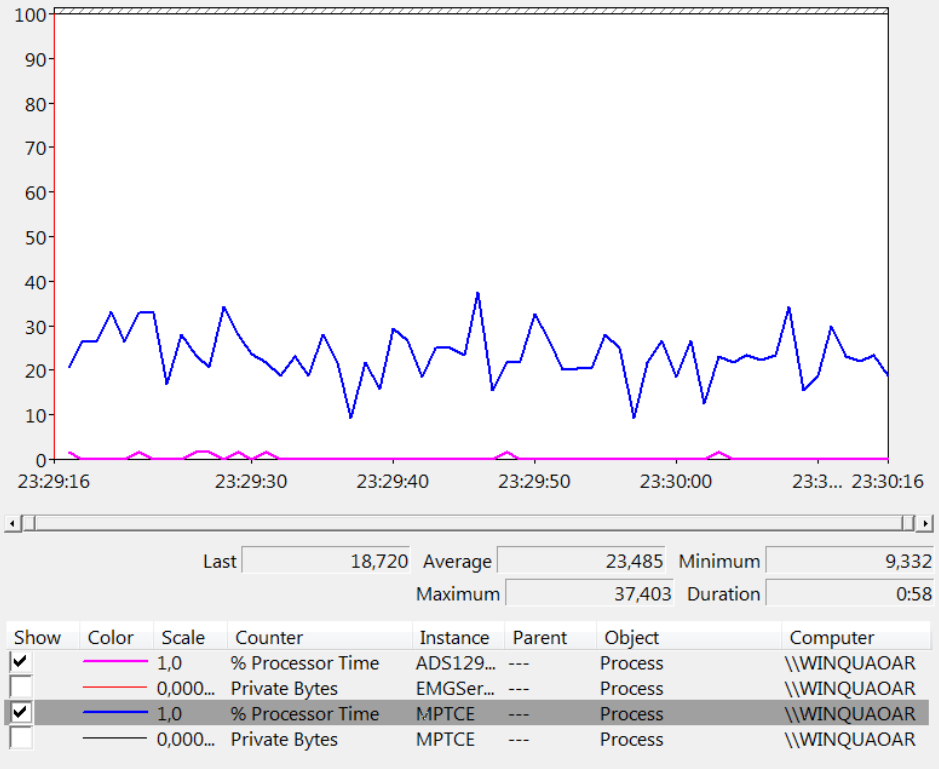
Ohne Echtzeitgrafiken

Leistung: Prozessorverbrauch (LDA)

Eine Minute Echtzeitbewegungserfassung



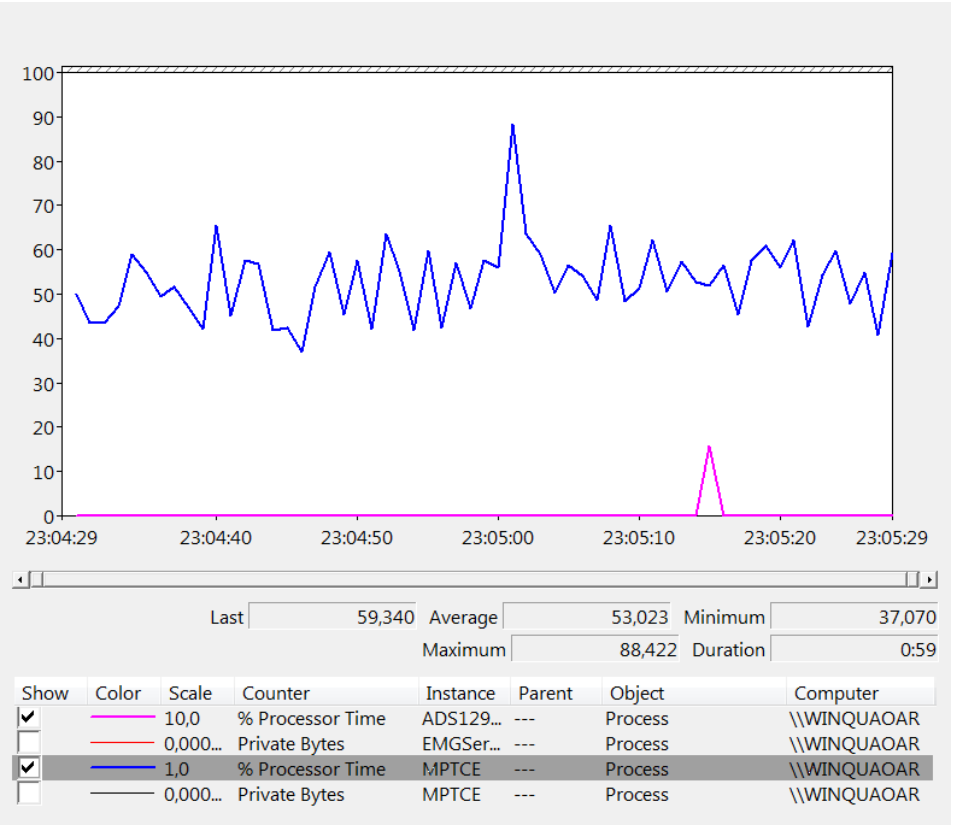
Mit Echtzeitgrafiken



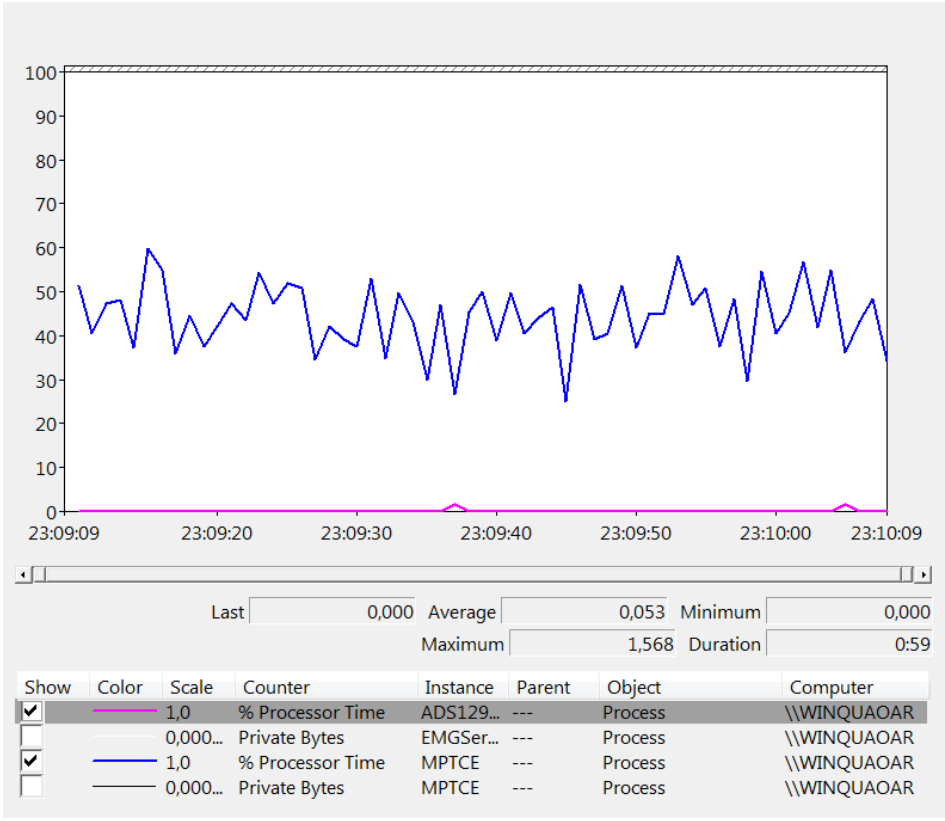
Ohne Echtzeitgrafiken

Leistung: Prozessorverbrauch (Schwellenbasierte Aktivierung)

Eine Minute Echtzeitbewegungserfassung



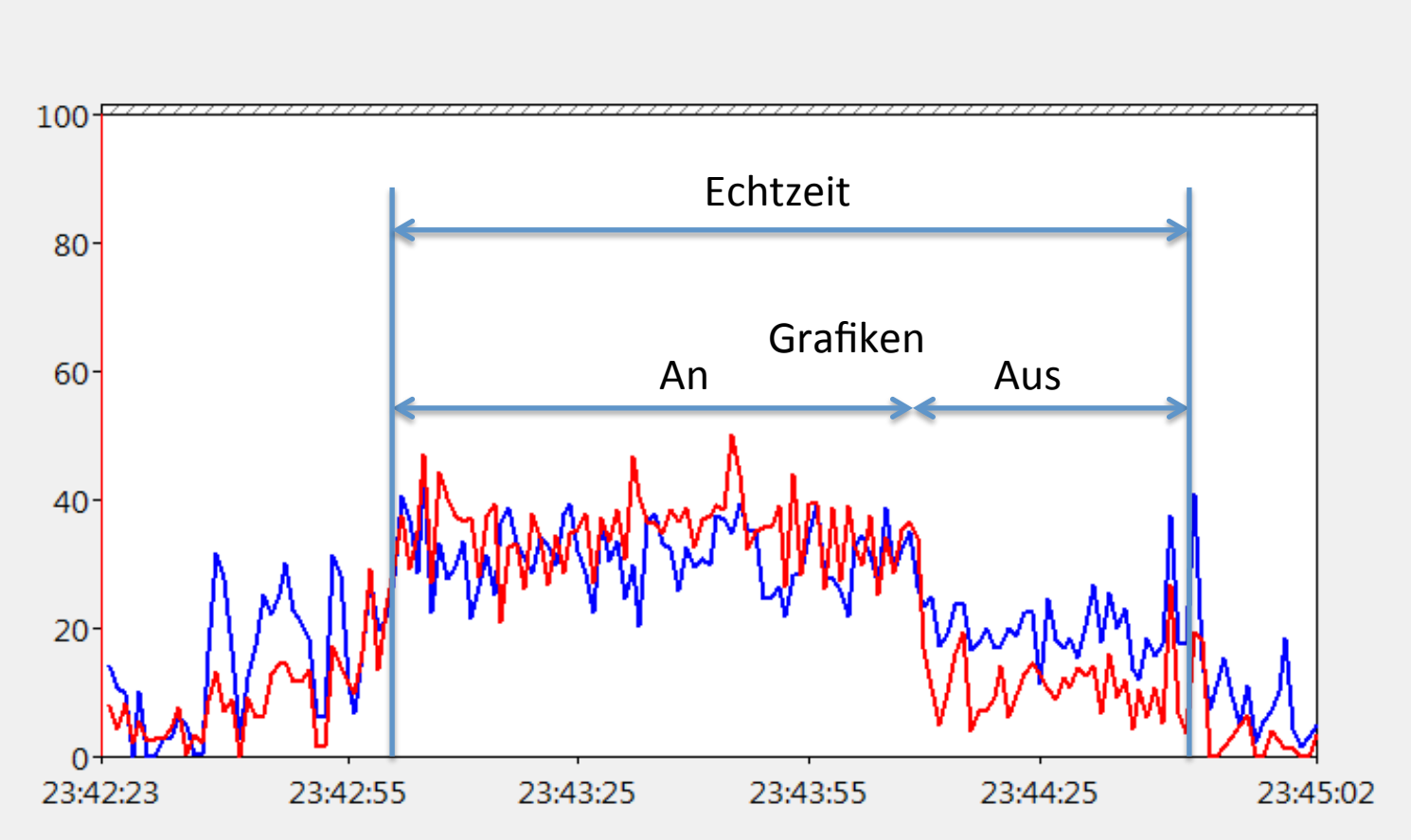
Mit Echtzeitgrafiken



Ohne Echtzeitgrafiken

Leistung: Distribution der Prozessorverbrauch zwischen zwei Kernen

Echtzeit MLP, 2 Bewegungen, ADS1298



Diskussion

- Merkmale
  - Erwünschte Funktionalitäten erfüllt
  - Erweiterbarkeit und Multiprozessorfähigkeit
  - Angemessene Nutzung von Ressourcen
- Einschränkungen / Probleme:
  - Aufwändiger Entwicklungsplan
    - Schulungszeit wurde unterschätzt
  - Testen des ADS1298-Geräts begrenzt
    - Nur Konnektoren für 2 von 8 Kanäle
    - Kabel zu kurz (30 cm)
    - Störungen, schlechtes Signal-Rausch-Verhältnis
  - Reaktionszeit der Schnittstelle nicht immer optimal
  - Erkennen von nicht vortrainierten simultanen Bewegungen war nicht erfolgreich

Zukünftige Arbeit

- Neue/bessere Mustererkennungsalgorithmen
  - Unterstützung von Simultanen Bewegungen
- Verbesserung der graphischen Schnittstelle
  - Effizienteres Grafiksystem
- Internationalisierung (Deutsch!)
- Optimierung des Speicherverbrauchs

Testen des ADS1298-Erfassungsgeräts mit längeren Kabeln bzw. genügenden Konnektoren für alle 8 Kanäle

- Verkleinerung / Integrierung
  - Erfassungshardware + Trainingsoftware in einem Gerät?

Danke

Noch Fragen?

mgcarmueja{ät}gmail{Punkt}com

Referenzen

[1] Ambu. (2015). Ambu Neuroline Inoject EMG needle electrodes. Retrieved July 6, 2015, from http://www.ambu.com/corp/products/patient_monitoring_and_diagnostics/product/neuroline_inoject-prod306.aspx

[2] Biometrics Ltd. (2015). Surface EMG Sensors. Retrieved July 5, 2015, from http://www.biometricsltd.com/semg.htm

[3] Ottobock. (2014). Michelangelo Produktbroschüre. Retrieved July 3, 2015, from http://www.leben-mit-michelangelo.de/fileadmin/downloads/techniker/deutsch/produktbroschuere_techniker.pdf

[4] Aszmann, O. C., Roche, A. D., Salminger, S., Paternostro-Sluga, T., Herceg, M., Sturma, A., … Farina, D. (2015). Bionic reconstruction to restore hand function after brachial plexus injury: a case series of three patients. Lancet, 385(9983), 2183–9.

[5] National Instruments. (2015). NI USB-6009. Retrieved July 3, 2015, from http://sine.ni.com/nips/cds/print/p/lang/en/nid/201987

[6] MathWorks. (2014). MATLAB - the language of technical computing. Retrieved September 29, 2014, from http://www.mathworks.com/products/matlab/

[7] Ortiz-Catalan, M., Brånemark, R., & Håkansson, B. (2013). BioPatRec: A modular research platform for the control of artificial limbs based on pattern recognition algorithms. *Source Code for Biology and Medicine*, *8*(1), 11.

[8] mlpy Developers. (2015). Linear Methods for Classification — mlpy v3.5.0 documentation. Retrieved July 6, 2015, from http://mlpy.sourceforge.net/docs/3.5/lin_class.html

[9] Heaton Research. (2015). Hyperbolic Tangent Activation Function - Encog Machine Learning Framework. Retrieved July 6, 2015, from http://www.heatonresearch.com/wiki/Hyperbolic_Tangent_Activation_Function

[10] Heaton Research. (2015). Sigmoid Activation Function - Encog Machine Learning Framework. Retrieved July 6, 2015, from http://www.heatonresearch.com/wiki/Sigmoid_Activation_Function