

Robotik-Projekt WS 2005

Verantwortlich: Prof. Hommel

Betreuer: Christian Fleischer

Raum: EN256

Tel.: 73114

fleischer@cs.tu-berlin.de

Organisatorisches I

- Projekt wird mit 6 SWS angerechnet
- keine Klausur am Ende
- Alles wird bewertet

- Anfangsvorträge zur Einarbeitung
- Zwischenbesprechung (um Weihnachten herum)
- Abschlußvorträge
- Projektbericht

Organisatorisches II

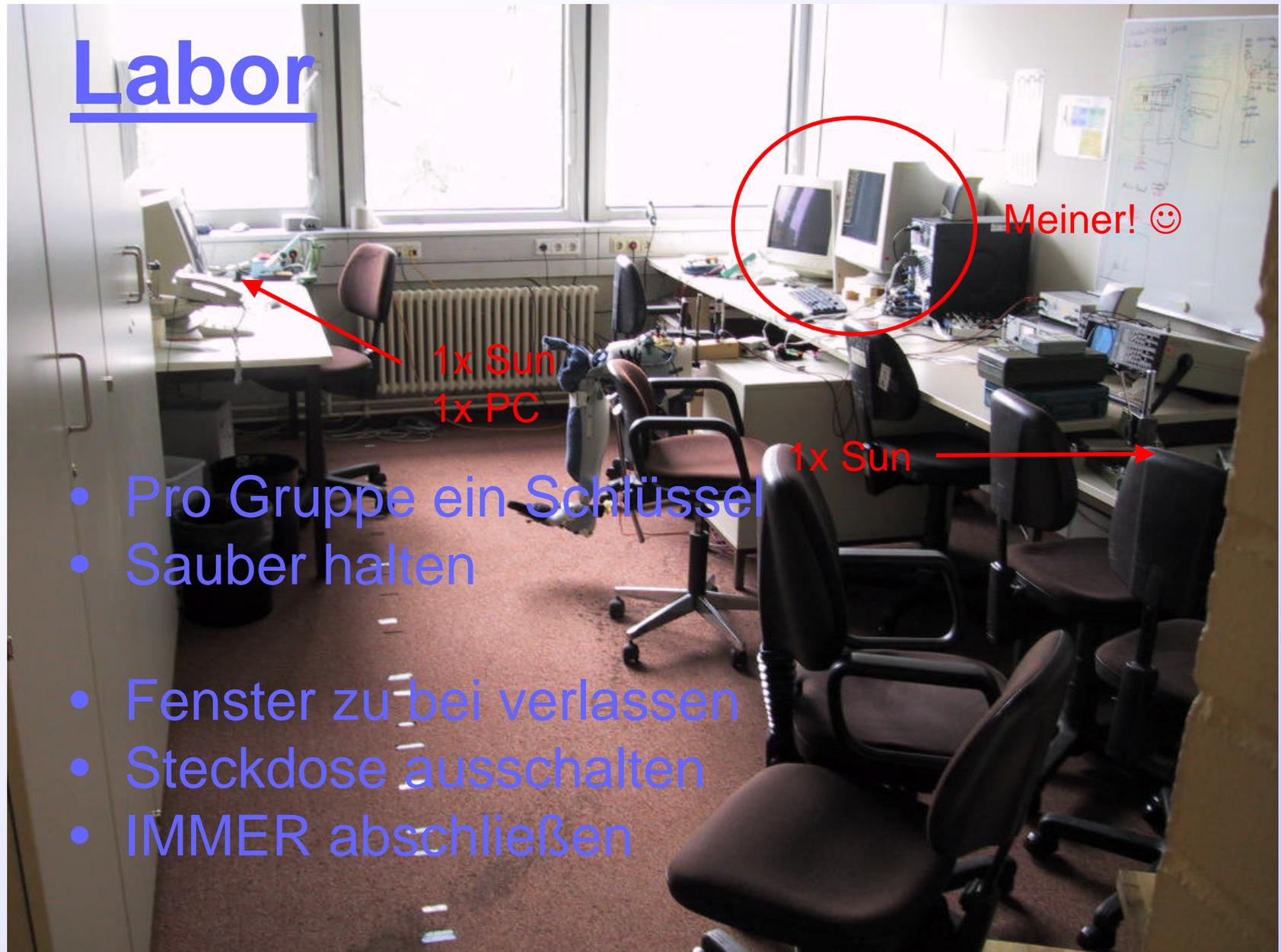
- fester Termin, alle müssen da sein
- zusätzliche Arbeitszeiten nach Belegung des Labors
- ab und zu Berichte an Dienstagen 12-14 Uhr, EN-183

- Labor: EN-242
- Telefon: 314-23408

Lernziele

- Selbständiges Aufbereiten der Aufgabenstellung, Zeitplanung
- Recherchieren nach notwendigen Informationen
- Arbeiten in der Gruppe
- Kommunikation mit anderen Gruppen, Schnittstellen
- Präsentation von Ergebnissen

Labor



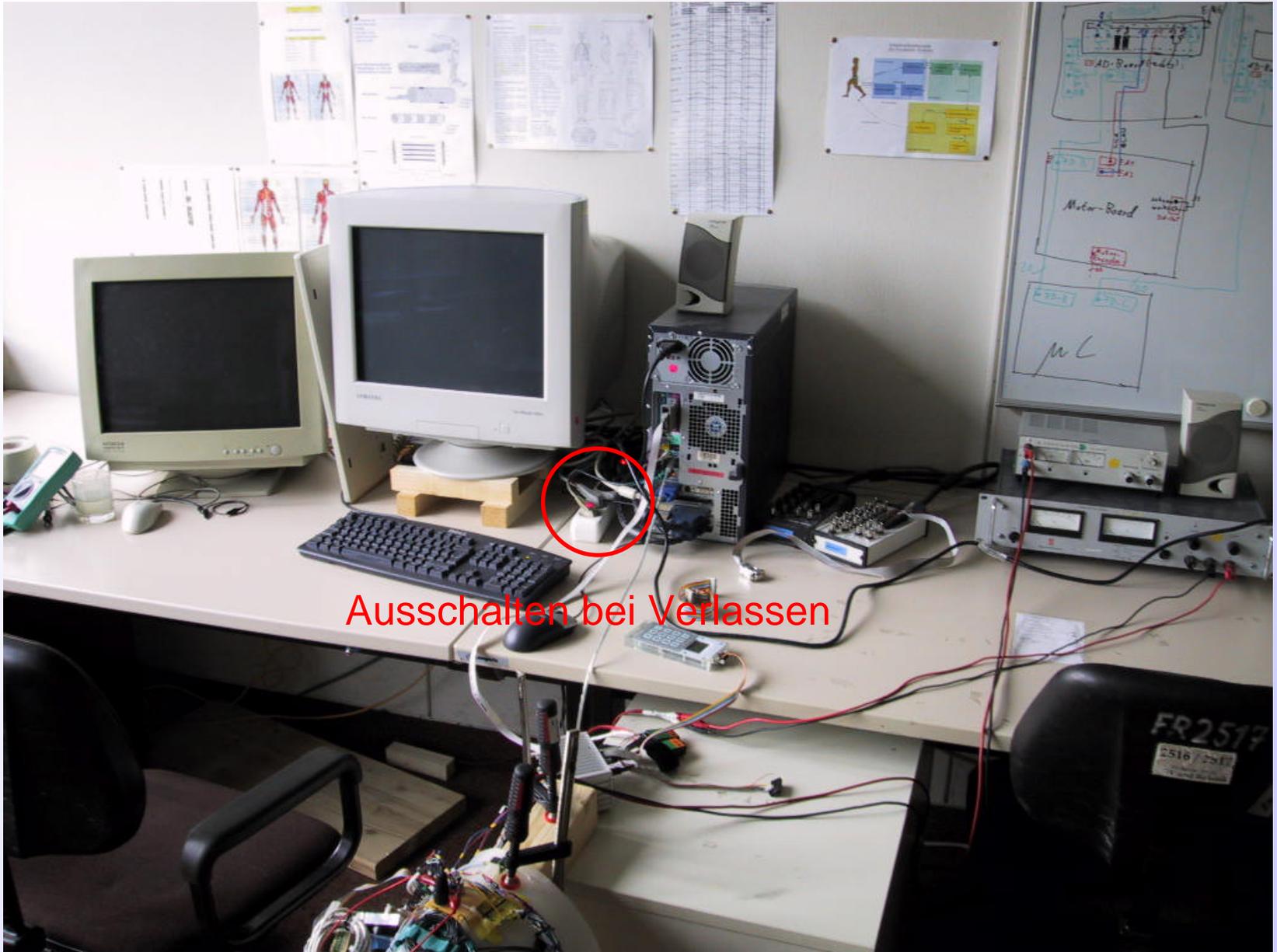
- Pro Gruppe ein Schlüssel
- Sauber halten
- Fenster zu bei verlassen
- Steckdose ausschalten
- IMMER abschließen

1x Sun
1x PC

1x Sun

Meiner! 😊

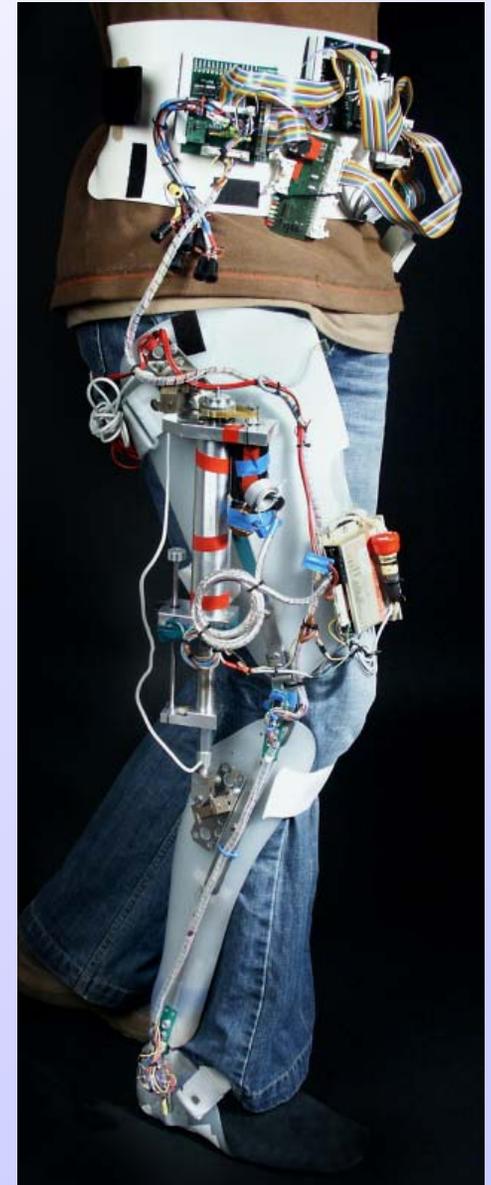




Ausschalten bei Verlassen

Hintergrund

- Steuerung einer Orthese mit Muskelsignalen
- Hilfe zur Rehabilitation
- Unterstützung durch Kraftzufuhr
- Fehlbewegungen verhindern



Idee zur Problemlösung

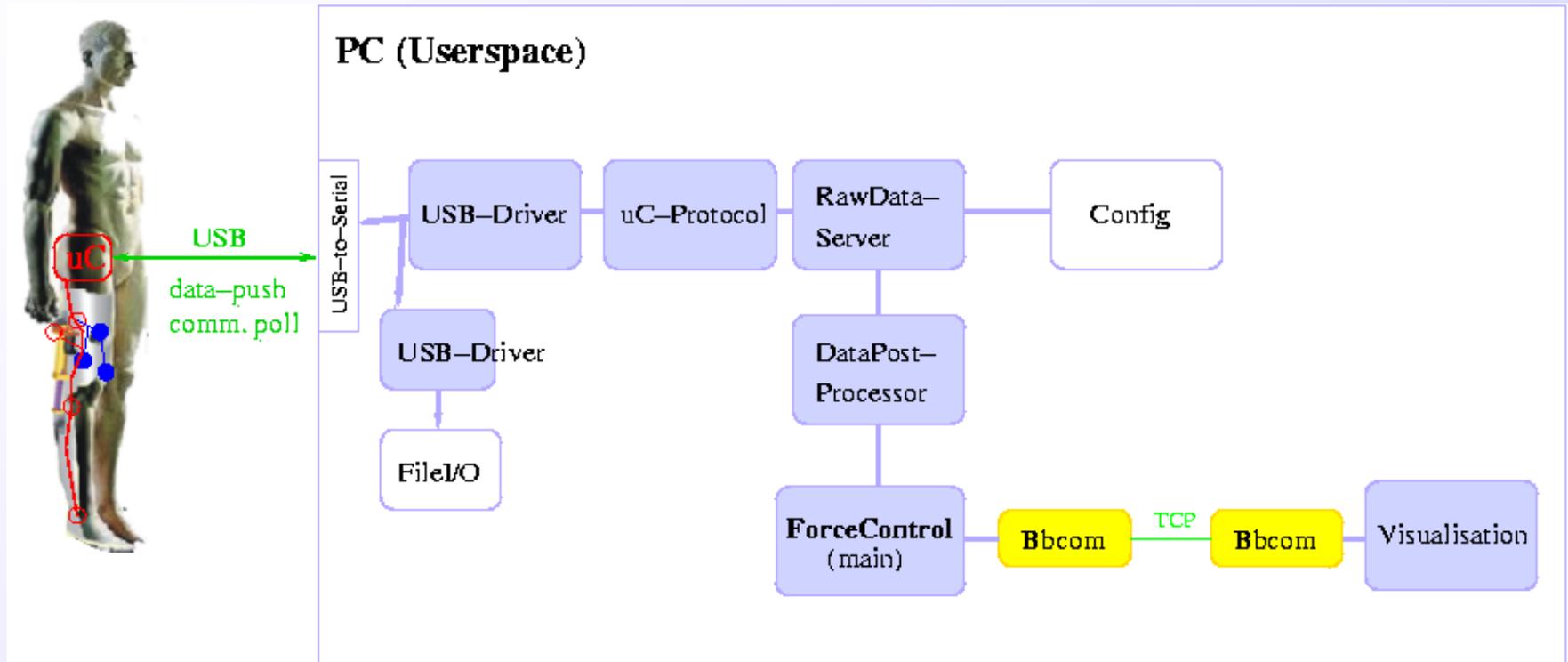
- Aufnahme der Muskelsignale
- Berechnung des Drehmoments im Knie (durch eigene Kraft)
- Hinzufügen von Drehmoment durch Aktuator
- Kraftsensor in Reihe mit Antrieb
- Misst Aktuator-Beitrag
- Kraftregler mit Kraftsensor

Stand der Dinge

- Orthese kann durch EMG-Signale bewegt werden
- Microcontroller erfaßt Sensoren und regelt
- PC führt Rechnungen durch
- Kommunikation über USB

- Kraftunterstützung nicht optimal
- Viele Bewegungen noch ungetestet
- Nicht autark

Übersicht



Hardware-Schwerpunkte

- **HW1:** Aufbau der zweiten Orthese
 - Verkabelung
 - Verbesserung der Aktuatorbefestigung
 - Beispielmessungen
- **HW2:** Displayansteuerung mit Benutzer-API
- **HW3:** Sicherheitsschaltungen, Überwachung
 - Spannungsüberwachung, Strommessung
 - Watchdog für Microcontroller

Software-Schwerpunkte

- **SW1:** Daten-Visualisierung mit QT, QWT, BBcom
- **SW2:** Konfigurationssystem zur Verwaltung aller einstellbaren Größen
- **SW3:** Blinde Quellentrennung von EMG-Signalen
- **SW4:** Patienten- und Meßdatenverwaltung
- **SW5:** Verbesserung der EMG-Signale durch Filter

HW1:

Orthesen-Aufbau

Ziele:

- **zweite Orthese für parallele Tests**
- **jetzige nicht geeignet für verschiedene Patienten**
- **einfaches an- und ablegen**



Inhalte im Einzelnen I:

- **Orthese vorhanden**
- **Sensorplatinen bestücken, anbringen**
- **Verkabelung**

- **flexible Anpassung an verschiedene Personen**
- **EMG-Sensoren einbetten**

- **bessere Befestigung des Aktuators
(Bewegungsbereich, Drehmoment)**
- **Durchführung von Beispielmessungen mit
verschiedenen Personen**

Inhalte im Einzelnen II:

- **Neuer Bauchgurt:**
 - Clip für Microcontroller
 - Akkus
 - Steuerung



HW2:

Displayansteuerung, Benutzer-API

Ziele:

- **Benutzerschnittstelle für Einstellungen, wenn System autark**
- **Tastatur ist vorhanden, funktioniert (Anpassung)**
- **Display in Hardware vorhanden, ungetestet**



Inhalte:

- **Test der Hardware, ggf. Änderung**
- **Microcontroller-Ansteuerung programmieren**

- **kleine API auf uC-Seite**
- **Erweiterung des Kommunikations-Protokolls**

- **komfortablere API auf PC-Seite**
- **Anbindung an QT**

HW3:

Sicherheitsschaltungen

Ziele:

- **Schaltungen entwerfen, die Fehler abfangen**
- **Zum Beispiel Motor abschalten, bei uC-Fehler**
- **Endabschalter einbauen**

Inhalte:

- **Spannungsüberwachung für alle Sensorgruppen**
- **Stromverbrauch messen, Aktuator separat**
- **Sicherheitsschaltung für Motor
(watchdog-Schaltung)**

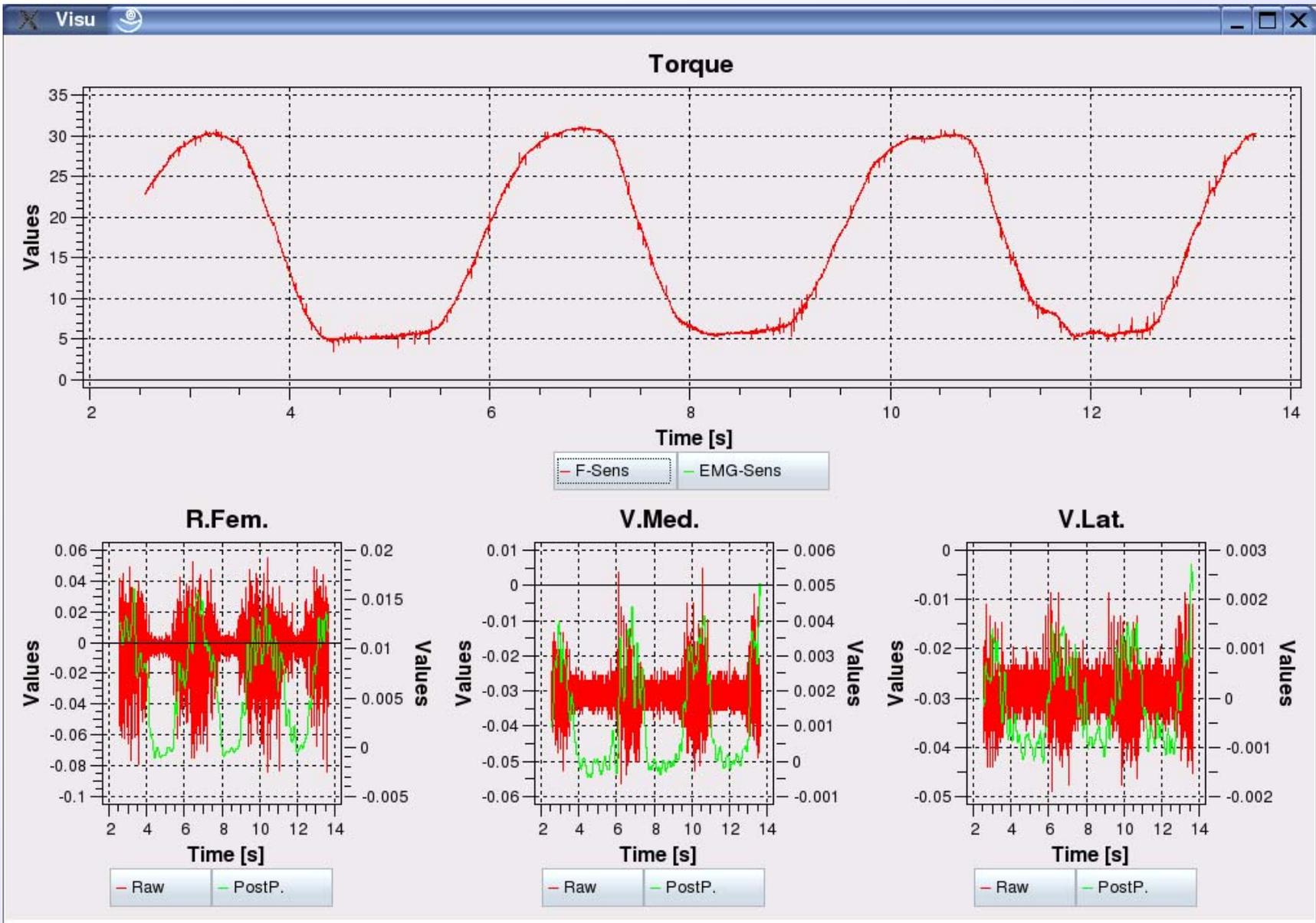
- **Endabschalter einbauen bzw. verbessern**
- **Hardware- und Softwareabfrage**

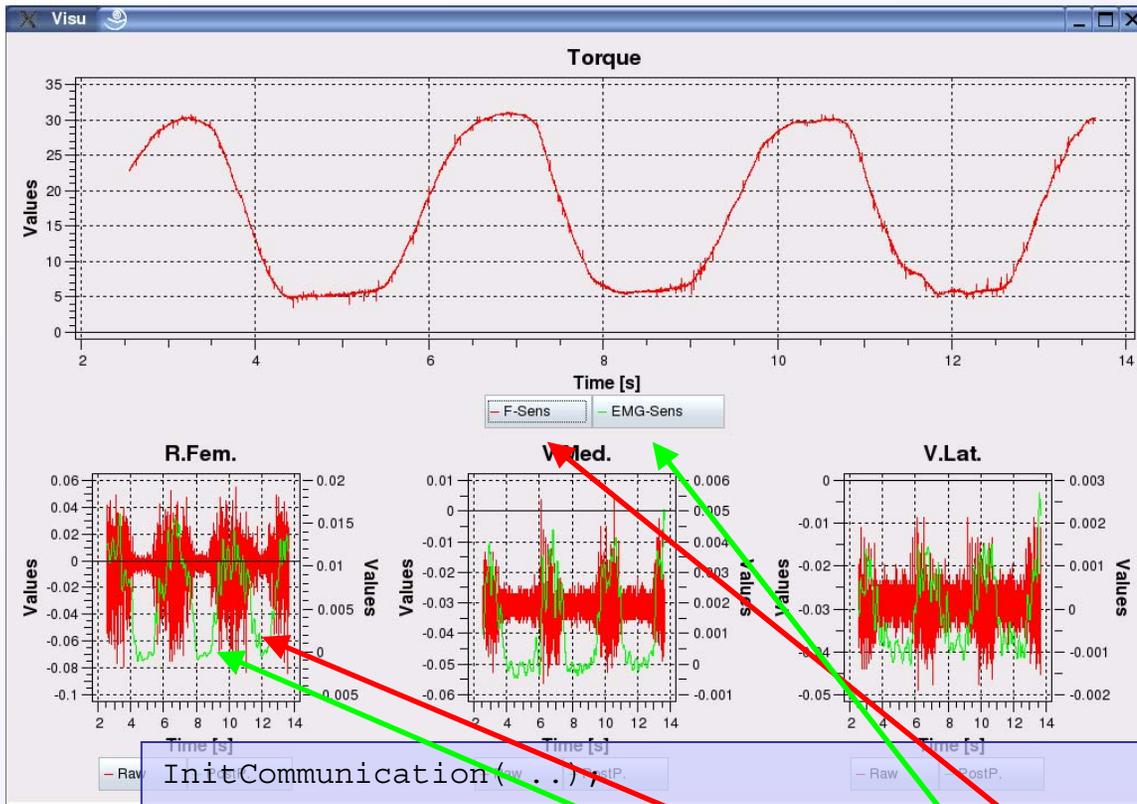
SW1:

Visualisierung der Meßdaten

Hintergrund:

- **Es gibt bereits eine Visualisierung**
- **Client-Server Architektur: Visualisierung ist Server**
- **Einfachste Handhabung**
- **Einzeiler für Signaldarstellung**
- **Demo-Visualisierung: optisch schön, 3D-Grafik der Orthese**





```
InitCommunication(...);
```

```
while (...)
{ // main loop
  ...
  CommunicateVisuSend(TransferBuffer("Torque", "F-Sens", Time, ...));
  CommunicateVisuSend(TransferBuffer("Torque", "EMG-Sens", Time, ...));

  CommunicateVisuSend(TransferBuffer("Torque|1,0#R.Fem.", "Raw", Time, ...));
  CommunicateVisuSend(TransferBuffer("Torque|1,0#R.Fem.", "PostP.>", Time, ...));
  ...
}
```

Inhalte:

- **Erarbeitung eines stimmigen Konzeptes und Übertragungsprotokolls**
- **Diagramme besser konfigurierbar (Achsen, Titel, Farben, Linienart...)**
- **Diagramme scroll-bar**
- **Darstellungsbreite einstellen**
- **Fenster für Textmeldungen**
- **Statuslämpchen**
- **3D-Visualisierung: intuitive Darstellung; für den Laien verständlich**

SW2:

Konfigurationssystem zur Verwaltung aller einstellbaren Größen

Hintergrund:

- **verschiedene Orthesen, Patienten etc. haben unterschiedliche Parameter**
- **Getrennte Speicherung**
- **Laden und Speichern von Konfigurationssätzen**
Speicherung in Textform, einfach zu editieren

Inhalte:

- Erarbeitung eines flexiblen Softwarekonzeptes
- Programmierung des Konfigurationssystems
- Einbettung in die vorhandene Software

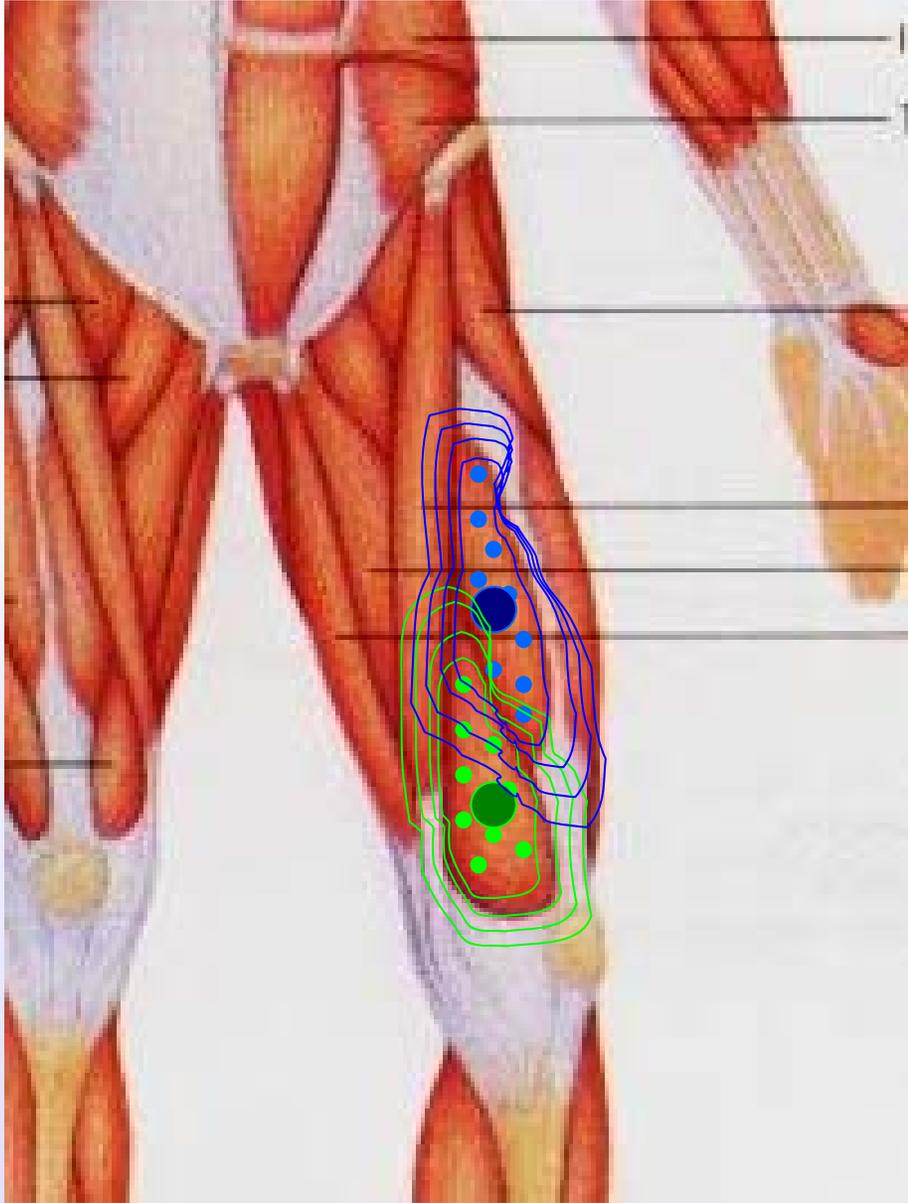
- *Soll zu Weihnachten fertig sein!*
- *Danach gibt es ein anderes Thema!*
(Was gerade aktuell ist)

SW3:

Blinde Quellentrennung von EMG-Signalen

Hintergrund:

- Sensoren teilweise dicht beieinander
- Muskeln groß
- Ein Sensor misst mehrere Muskeln
- Signale nach Muskeln trennen



Inhalte:

- **Recherche über vorhandene Algorithmen**
- **Was wurde bereits erreicht?**
- **Implementierung einer Variante**

- **Evaluierung der Qualität des Verfahrens**

SW4:

Patienten- und Meßdatenverwaltung

Hintergrund:

- Einfaches Benutzerinterface für Laien
- Interaktion mit Tastaturfeld und Display

Inhalte:

- **Ausarbeiten eines stimmigen Softwarekonzeptes**
- **soll Verwaltung von Patienten und Testläufen ermöglichen**
- ***Sehr* flexibel konfigurierbare Schnittstelle zur eigentlichen Software**
- **soll ein Framework sein, das Steuerungssoftware einbettet**
- **Kommunikation mit Steuerungssoftware**
- **Implementierung**

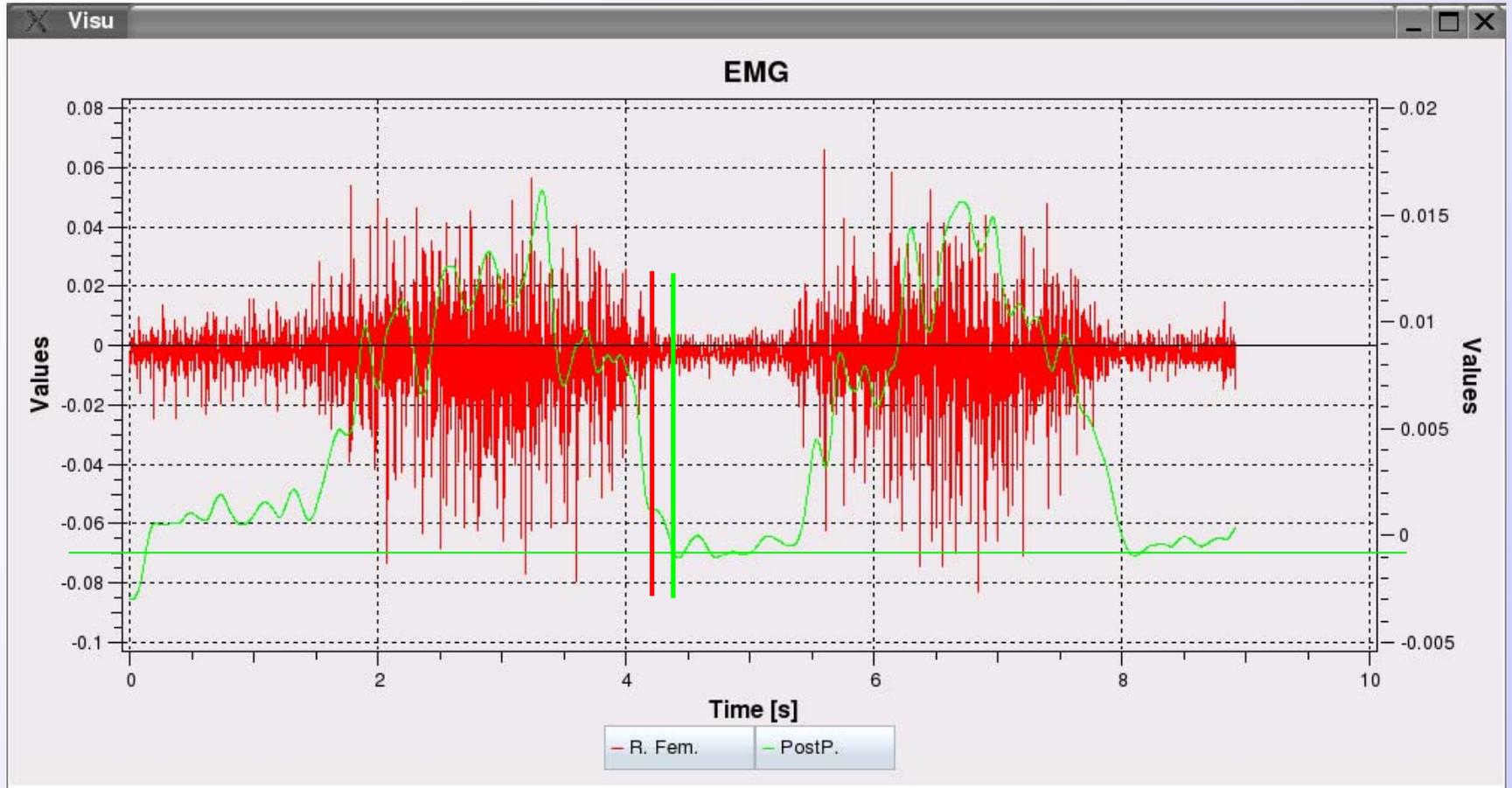
SW5:

Verbesserung der EMG-Signalverarbeitung

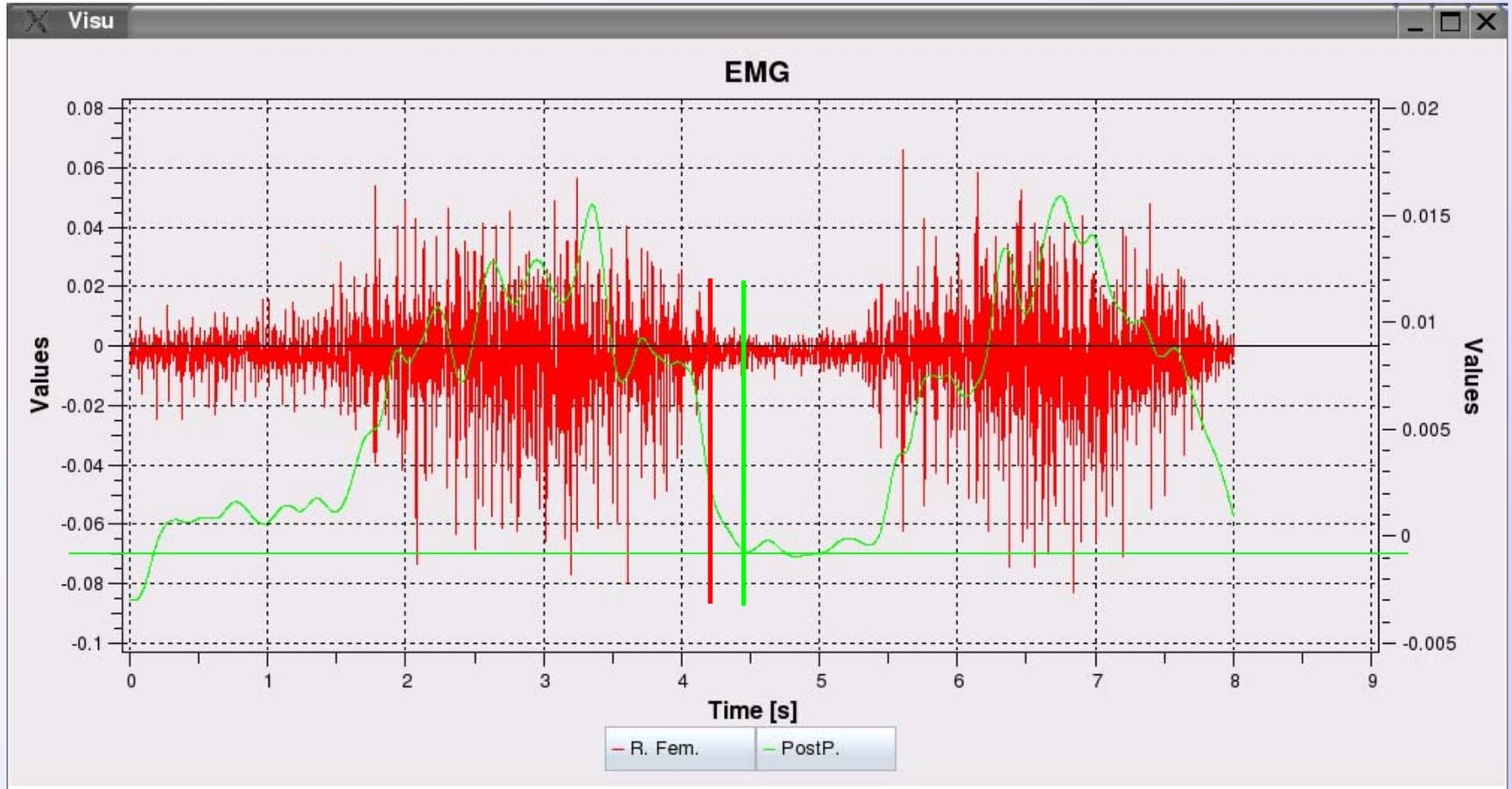
Hintergrund:

- **EMG-Signale sind sehr unstetig**
- **es wird immer nur ein Teil eines Muskels gemessen**
- **führt zu ruckeligen Bewegungen**
- **Tiefpaßfilter mit niedriger Grenzfrequenz hat hohe Verzögerung**

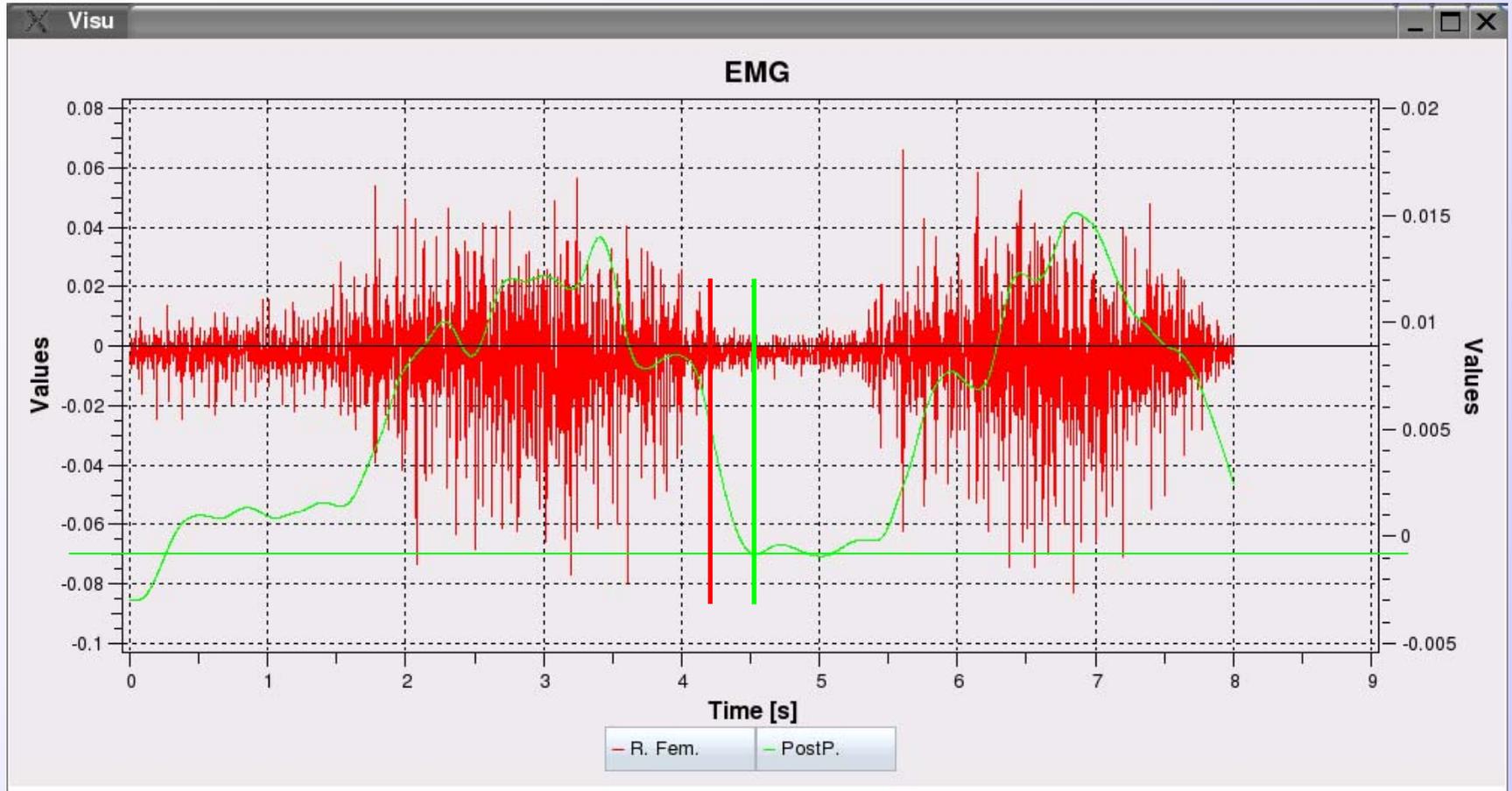
4 Hz

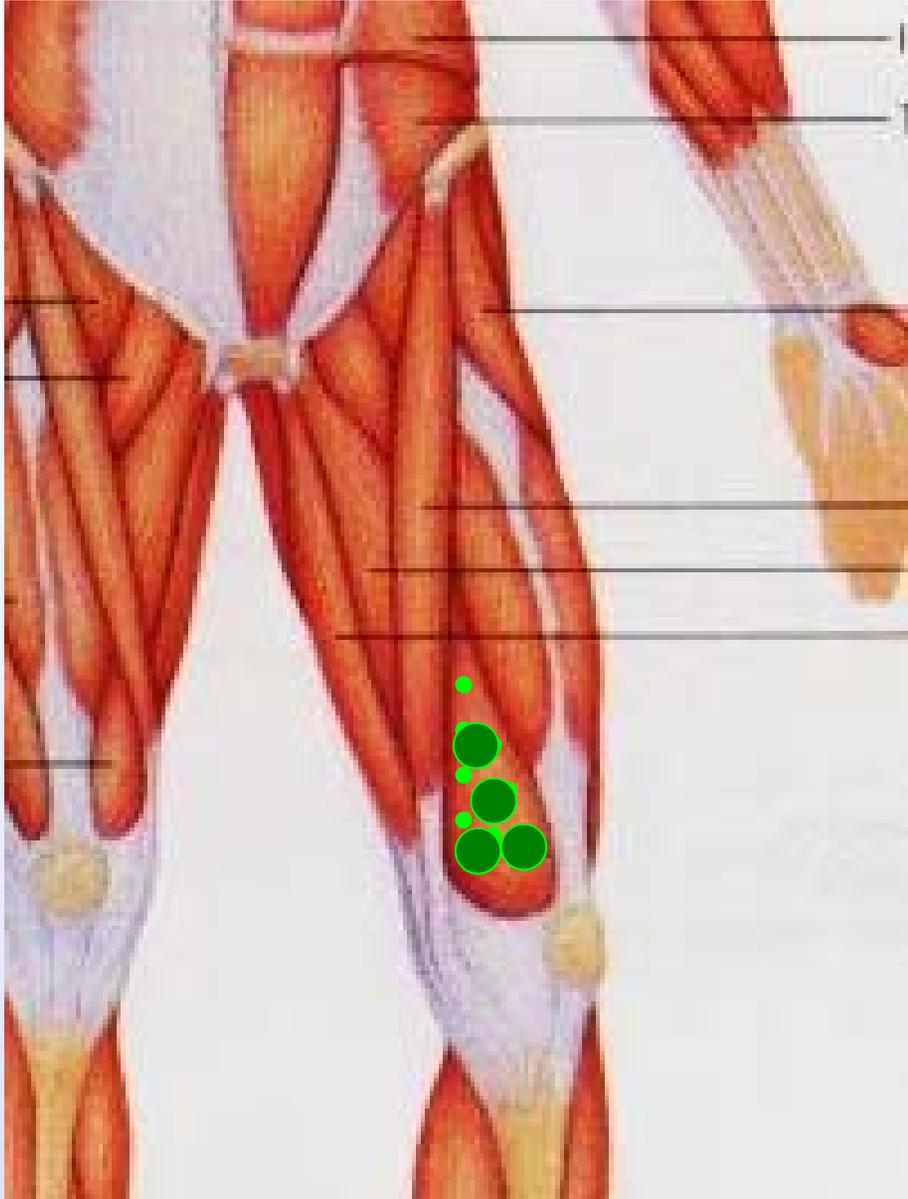


3 Hz

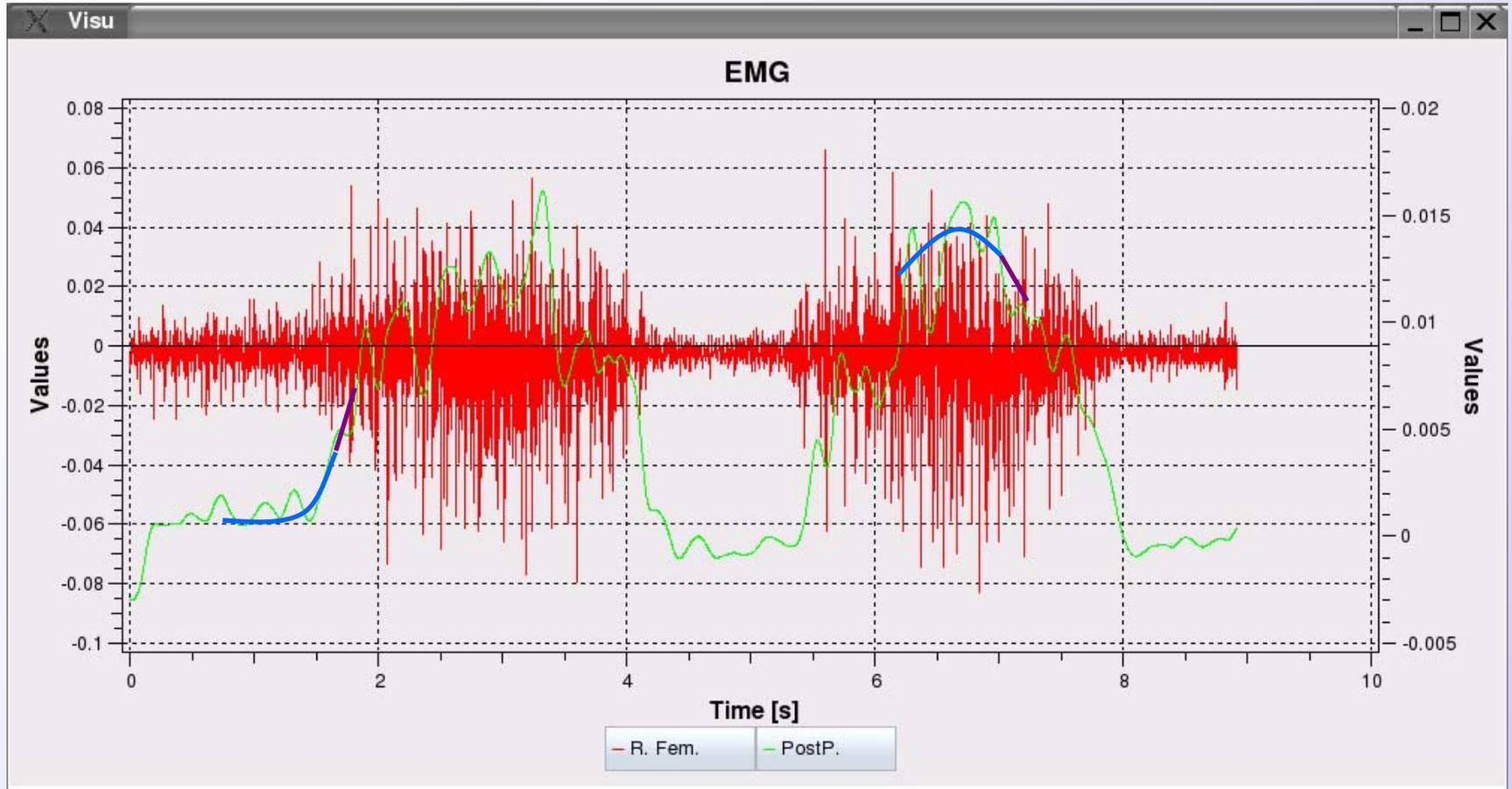


2 Hz





4 Hz



Inhalte:

- **Recherche nach verschiedenen Glättungsfiltern**
- **Test der Glättung durch mehrfach örtlich versetztes Samplen**
- **Approximation der EMG-Signale durch Polynome (einstellbare Anzahl von Maxima in einem bestimmten Intervall)**
- **Evaluierung der Filterung**

Themenvergabe

- **HW1:** Aufbau der zweiten Orthese
- **HW2:** Displayansteuerung mit Benutzer-API

- **SW1:** Daten-Visualisierung
- **SW2:** Konfigurationssystem
- **SW3:** Blinde Quellentrennung für EMG-Signalen
- **SW4:** Patienten- und Meßdatenverwaltung
- **SW5:** Verbesserung der EMG-Signale

Zeitplan in den nächsten Wochen

18.10. (KW42)	Recherche, Themenbesprechungen (alle Gruppen!)
25.10. (KW43)	Recherche
01.11. (KW44)	Vorträge: HW1, HW2, SW1, SW2
08.11. (KW45)	Vorträge: HW3, SW3, SW4, SW5
15.11. (KW46)	
...	
...	
13.12. (KW50)	Zwischenvorträge

! Viel Spaß !

