

NA-Online-Toolbox: Modulbeschreibung

Datum:	12.06.2014
Autor:	Christof Pieloth
Änderungen:	2014-07-21: Head Position Estimation hinzugefügt 2014-06-12: Erstellung Dokuments

Inhaltsverzeichnis

Glossar.....	2
MNE Real-time Client.....	3
FieldTrip Real-time Client.....	4
EMM Generator.....	5
FIFF Reader.....	6
EMM Simulator.....	7
FIR Filter.....	8
Epoch Separation.....	9
Epoch Rejection.....	10
Epoch Averaging.....	11
Alignment.....	12
Leadfield Interpolation.....	13
Head Position Estimation.....	14
Source Reconstruction.....	15

Glossar

Module	Einheit aus Benutzeroberfläche, Visualisierung und Algorithmus zur Verarbeitung von elektromagn. Daten; Glied in der Signalverarbeitungskette
EMM	ElectroMagneticMeasurement; Container-Objekt für alle Daten einer EEG-/MEG-Messung
EMD	ElectroMagneticData; Rohdaten und Meta-Informationen einer Modalität; wird unterteilt in EMDEEG, EMDMEG, ...
CMD	Command, Befehl; Ein Befehlsobjekt kann Befehle (RESET, INIT, COMPUTE) und Daten (EMM) enthalten. Zwischen Modulen werden Befehle weitergereicht.
Paket/Block	Der Datenstrom eines EEG-/MEG-Systems wird blockweise verarbeitet. Zusätzlich kann eine aufgezeichnete Messung als Datenstrom blockweise verarbeitet werden. Die Bezeichnung Paket wird im Kontext von EMM verwendet. Die Bezeichnung Block wird im Kontext der Rohdaten verwendet.
Trigger	Siehe Event
Stimuli	Siehe Event
Event	Event bezeichnet ein Stimuli bzw. Reiz bei einem „evoked potential EEG/MEG“.
Trial	Siehe Epoche
Epoche	Ist ein Datenausschnitt, der ein Event enthält. Für ein festgelegtes Event sind die pre- und post-Stimuli-Zeiten i.d.R. konstant.
FIFF	Functional Image File Format, Container-Dateiformat zum Speichern elektromagn. Daten

MNE Real-time Client

Überblick:

Kurzbeschreibung:	Liest eine laufende Messung vom MNE Real-time Server über TCP/IP.
Moduleingang:	-
Modulaustrag:	1..n EMM-Pakete mit einer festen Blockgröße
Vorbedingungen:	Gestarteter MNE Real-time Server
GPU-Unterstützung:	-

Parameter:

IP:	IP-Adresse vom MNE Real-time Server
Connectors:	Auswahl verfügbarer „Connectors“, z.B. Neuromag Client
Simulation File:	Datei inkl. Pfad auf dem MNE Real-time Server zum Streaming mit dem FIFF Simulator
Block size:	Angeforderte Blockgröße beim MNE Real-time Server, wird auch für EMD-Blöcke am Modulaustrag verwendet
Apply scaling:	Rechnet das Scaling $range \cdot cal$ in die Daten ein

Beschreibung:

- Verbindungsaufbau zum MNE Real-time Server
 - Command-Port: 4217
 - Daten-Port: 4218
- Anforderung von Messinformationen:
 - verfügbare Modalitäten,
 - Anzahl der Kanäle,
 - Abtastrate, usw.
- Start der Übertragung
 - Solange Messung nicht gestoppt
 - Lesen der Daten im FIFF-/MNE-Format
 - Überführung in Objektstruktur von NA-Online (EMM, EMD)
 - Weitergabe des erstellten EMM-Pakets an Folgemodul
- ggf. Stopp der Übertragung

FieldTrip Real-time Client

Überblick:

Kurzbeschreibung:	Liest eine laufende Messung vom FieldTrip Buffer.
Moduleingang:	-
Modulaustrag:	1..n EMM-Pakete mit einer festen Blockgröße
Vorbedingungen:	Gestartete Implementierung eines FieldTrip Buffer mit festgelegter Blockgröße, z.B. rtMEG
GPU-Unterstützung:	-

Parameter:

Connection Type:	Verbindung über TCP/IP oder Unix-Sockets
Host IP:	IP-Adresse vom FieldTrip Buffer
Port number:	Port des FieldTrip-Buffer-Dienstes
Max. data request timeout:	Timeout für das Warten auf neue Daten/Events, bevor das Streaming abgebrochen wird
Apply scaling:	Rechnet das Scaling $range \cdot cal$ in die Daten ein

Beschreibung:

- Verbindungsaufbau zum FieldTrip Buffer
- Start der Übertragung
- Warten auf Daten vom FieldTrip Buffer, da die Messung mit der Neuromag Acquisition Software gestartet wird (rtMEG)
 - Solange Messung nicht gestoppt
 - Lesen der Daten im FieldTrip-Format
 - Überführung in Objektstruktur von NA-Online (EMM/EMD)
 - Weitergabe des erstellten EMM-Pakets an Folgemodul
- ggf. Stopp der Übertragung

EMM Generator

Überblick:

Kurzbeschreibung:	Generiert eine EEG-Messung mit zufälligen Daten.
Moduleingang:	-
Modulaustrag:	Ein EMM-Paket mit EEG-Modalität und zufälligen Daten
Vorbedingungen:	-
GPU-Unterstützung:	-

Parameter:

Sampling Frequency:	Abtastrate in Hz
Length:	Länge der zu generierenden Daten in Sekunden
Channels:	Anzahl der Kanäle bzw. Sensoren

Beschreibung:

- Initialisierung eines leeren EMM und EMDEEG
- Berechnung der Anzahl der Samples: $sFreq * length$
- Initialisierung der Daten mit der entsprechenden Kanal- und Sample-Anzahl
- Setzen jedes Samples mit `rand()` aus der `cstdlib`
- Generierung zufälliger Sensorpositionen
- Weitergabe des EMM an Folgemodul

FIFF Reader

Überblick:

Kurzbeschreibung:	Liest eine Messung aus einer FIFF-Datei ein.
Moduleingang:	-
Modulaustritt:	Ein EMM-Paket mit allen verfügbaren Daten und Modalitäten
Vorbedingungen:	-
GPU-Unterstützung:	-

Parameter:

FIFF file:	FIFF-Datei zum Einlesen
BEM Layer (optional):	Auswahl gefundener BEM Layer
Surfaces:	Auswahl gefundener Surfaces

Beschreibung:

- Liest eine FIFF-Datei mit Hilfe der `libfiffio` ein
- Überführt die Daten in die Objektstruktur von NA-Online
- Falls die Ordnerstruktur den Konventionen der MEG-Gruppe folgt und die FIFF-Datei aus dem `rawdir` geöffnet wurde, wird nach Zusatzdaten wie BEM Layer, Source Space u.m. Gesucht und kann anschließend geladen werden
- Weitergabe des EMM an Folgemodul, nachdem *Send* gedrückt wurde

EMM Simulator

Überblick:

Kurzbeschreibung:	Simuliert ein Streaming indem eine vollständige Messung paket-/blockweise übertragen wird.
Moduleingang:	1x EMM-Paket mit Blockgröße b1
Modulaustrag:	1..n EMM-Pakete mit Blockgröße b2, ggf. angereichert mit Zusatzdaten
Vorbedingungen:	EMM-Paket am Moduleingang
GPU-Unterstützung:	-

Parameter:

Auto start:	Startet das Streaming sobald ein EMM-Paket verfügbar ist
Block size:	Blockgröße in Millisekunden

Beschreibung:

- EMM-Paket am Moduleingang bzw. die enthaltenen Modalitäten werden in Blöcke geteilt und sequenziell mit Pausen an den Modulausgang gelegt
- `Packetizer` erstellt aus dem EMM und der enthaltenen EMDs neue Objektinstanzen mit der gewünschten Blockgröße
 - Initialisierung mit Blockgröße und der zu teilenden Messung (EMM)
 - Iteration/Blockbildung mit den Methoden: `hasNext()`, `next()`
- Solange `streaming` und `Packetizer.hasNext()`:
 - Starte Zeitmessung
 - `EMM=Packetizer.next()`
 - setze ggf. Zusatzdaten wie Leadfield, Source Space, BEM Layer usw.
 - Weitergabe EMM an View und Folgemodule
 - Stoppe Zeitmessung
 - Warte **Blocklänge-benötigte Zeit**, bevor das nächste EMM-Paket erstellt wird

FIR Filter

Überblick:

Kurzbeschreibung:	Filtert EMDs mit lowpass-, highpass-, bandpass-, bandstop-Charakteristik
Moduleingang:	1x EMM-Paket
Modulaustrag:	1x neues EMM-Paket gefiltert und mit halber Phasenverschiebung
Vorbedingungen:	<ul style="list-style-type: none">• EMD-Blockgröße < Filterkoeffizienten• EMDs besitzen gleiche Abtastrate
GPU-Unterstützung:	CUDA

Parameter:

Coefficients:	Laden von exportierten Filterkoeffizienten aus dem <i>FDATool</i> von Matlab
Filter Type:	Auswahl einer Filtercharakteristik
Windows:	Auswahl einer Fenstermethode
Sampling Frequency:	Abtastrate der EMDs, benötigt für Berechnung der Filterkoeffizienten
Cutoff frequency 1:	<ul style="list-style-type: none">• Grenzfrequenz für lowpass/highpass• untere Grenzfrequenz für bandpass/bandstop
Cutoff frequency 2:	<ul style="list-style-type: none">• obere Grenzfrequenz für bandpass/bandstop• keinen Einfluss auf lowpass/highpass
Order:	Anzahl der Filterkoeffizienten

Beschreibung:

- Initialisierung/Design des Filter durch Berechnung oder Laden der Filterkoeffizienten
- Filtern aller EMDs unter Einbeziehung des vorherigen EMDs oder Nulldaten
- Speichern der gefilterten EMDs in neuen EMD-Instanzen
- Speichern des ungefilterten EMDs für nächstes Paket
- Klonen eines neuen EMMs und hinzufügen der gefilterten EMDs
- „Post-Processing“: Verschieben der Events/Trigger um $nCoefficients/2$
- Weitergabe der neuen EMM-Instanz an Folgemodul

Epoch Separation

Überblick:

Kurzbeschreibung:	Sucht nach einem definierten Event im EMM-Paket und erstellt eine Epoche.
Moduleingang:	1x EMM-Paket
Modulaustritt:	0..n neue EMM-Pakete der Blockgröße: preSamples+postSamples+1
Vorbedingungen:	Logisch aufeinander folgende EMM-Pakete am Moduleingang.
GPU-Unterstützung:	-

Parameter:

Pre-Samples:	Anzahl der Samples die vor dem Event in die Epoche übernommen werden sollen
Post-Samples	Anzahl der Samples die nach dem Event in die Epoche übernommen werden sollen
Triggers:	<ul style="list-style-type: none">• Komma separierte Liste der Event-Codes• diese werden zu einer Bit-Maske zusammengeführt
Event channel:	Index des Event-Kanal, welcher die Events enthält

Beschreibung:

- Initialisierung:
 - Berechnung der Anzahl vor zuhaltender EMM-Pakete anhand der Blockgröße, Pre- und Post-Samples
 - Erstellung eines Ringpuffers für das vorhalten vorangegangener EMM-Pakete
- Verarbeitung:
 - Füge EMM Ringpuffer hinzu
 - Vergleiche alle Samples des Event-Kanals mit der Bit-Maske
 - wenn Event gefunden, erstelle EMM/EMD aus den vorgehaltenen und verfügbaren Daten
 - wenn Post-Samples fehlen, füge neues EMM in Liste offener EMMs hinzu
 - für alle offenen EMMs, übernehme fehlende Samples aus EMM
 - für alle offenen EMMs, überprüfe ob noch Post-Samples fehlen, wenn nicht, dann entferne diese aus der Liste und füge diese EMMs in eine Liste für fertige EMMs hinzu
 - entferne ältestes EMM aus dem Ringpuffer
 - Sequenzielle Weitergabe aller fertigen EMMs an das Folgemodul

Epoch Rejection

Überblick:

Kurzbeschreibung:	Epochenzurückweisung anhand von Schwellwerten.
Moduleingang:	1x EMM-Paket
Modulaustritt:	0..1 EMM-Paket
Vorbedingungen:	Sollte in der Signalverarbeitungskette zwischen Epoch Separation und Epoch Averaging platziert sein.
GPU-Unterstützung:	-

Parameter:

Epoch Buffer Size:	Die Anzahl der Epochen, die maximal für eine erneute Prüfung zurückgehalten werden.
CFG file:	Datei mit Schwellwerten
EEG threshold:	Der Schwellwert für EEG
EOG threshold:	Der Schwellwert für EOG
MEG gradiometer threshold:	Der Schwellwert für das MEG Gradiometer
MEG magnetometer threshold:	Der Schwellwert für MEG Magnetometer

Beschreibung:

- Total Channel Rejection:
 - Prüfung jeder Modalität im EMM Paket einzeln
 - Die Differenz aus dem totalen Maximum und dem totalen Minimum über alle Kanäle wird gegen den Schwellwert der Modalität verglichen.
 - Besteht eine Modalität den Test nicht, wird die Epoche auf den Zwischenspeicher verschoben, ansonsten erfolgt die Weitergabe.
- Single Channel Rejection:
 - Prüfung jeder Modalität im EMM Paket einzeln
 - Die Differenz aus dem totalen Maximum und dem totalen Minimum wird einzeln für jeden Kanal gegen den Schwellwert der Modalität verglichen.
 - Besteht ein Kanal den Test nicht, wird die Epoche auf den Zwischenspeicher verschoben, ansonsten erfolgt die Weitergabe.
- Zwischenspeicher wird in zukünftigen Entwicklungen verwendet, um false-positives aufgrund von Bad-Channels zurück in die Signalverarbeitungskette zu geben

Epoch Averaging

Überblick:

Kurzbeschreibung:	Mittelung aller eingehenden EMDs bzw. genauer Epochen.
Moduleingang:	1x EMM-Paket
Modulaustritt:	1x neues und gemittelttes EMM-Paket
Vorbedingungen:	<ul style="list-style-type: none">• EMDs benötigen gleiche Blockgröße• EMDs müssen „Event“-synchron sein, dies wird nicht überprüft!
GPU-Unterstützung:	-

Parameter:

Average Type:	Auswahl zwischen totaler Mittelung oder gleitender Mittelung
TBase	Anzahl der Samples die für die Baseline-Korrektur verwendet werden
Size of Moving Average:	Anzahl der EMM-Pakete die gemittelt werden sollen

Beschreibung:

- Total Average:
 - Initialisierung: erstelle EMM/EMDs und Zähler für laufende Summe
 - Baseline-Korrektur auf eingehenden EMD-Blöcken
 - Addiere Daten auf laufende Summe und erhöhe Zähler
 - erstelle neue Instanzen aus laufender Summe und teile Daten durch Zähler
 - übernehme Event-Kanäle aus eingehenden EMM
- Moving Average:
 - Initialisierung: erstelle Ringpuffer und Zähler
 - Baseline-Korrektur auf eingehenden EMD-Blöcken
 - füge EMM Puffer hinzu
 - erstelle ausgehendes EMM/EMDs
 - addiere alle EMMs bzw. dessen EMDs aus dem Puffer
 - teile Daten durch die Anzahl der enthaltenen EMMs
 - übernehme Event-Kanäle aus eingehenden EMM

Alignment

Überblick:

Kurzbeschreibung:	Berechnet halbautomatisch eine Transformationsmatrix zwischen Kopfkoordinatensystem (EEG) und AC-PC-Koordinatensystem (BEM-Skin-Layer).
Moduleingang:	1x EMM-Paket
Modulaustrag:	1x EMM-Pakete mit gesetzter Transformationsmatrix
Vorbedingungen:	<ul style="list-style-type: none">• Digitization/Isotrak-Punkte mit Fiducials vorhanden• BEM-Outer-Skin- oder BEM-Head-Layer vorhanden
GPU-Unterstützung:	-

Parameter:

LPA (AC-PC):	Fiducial-Punkt linkes Ohr im AC-PC-Koordinatensystem
Nasion (AC-PC):	Fiducial-Punkt Nasenwurzel im AC-PC-Koordinatensystem
LPA (AC-PC):	Fiducial-Punkt rechtes Ohr im AC-PC-Koordinatensystem
Iterations:	Maximal Anzahl der Iterationen für ICP-Algorithmus

Beschreibung:

- Weitere Details in eigener Dokumentation
- Point Cloud Library zur Schätzung der initialen Transformation und Bestimmung der finalen Transformation mittels Iterative Closest Point
- Zusammentragen der Fiducial-Punkte im Kopfkoordinatensystem und AC-PC-Koordinatensystem
 - Kopfkoordinaten aus Digitization/Isotrak-Punkte
 - AC-PC-Koordinaten aus Benutzereingabe (manuell)
- Bestimmung einer initialen Transformationsmatrix
 - Registrierung bzw. Bildung der Punktpaare: $\text{lpaHead} \leftrightarrow \text{lpaACPC}, \dots$
 - Schätzung einer Transformation für die 3 bzw. 6 Punkte mittels Singulärwertzerlegung
- Bestimmung einer finale Transformationsmatrix
 - Minimierung des Abstands zwischen EEG-Haube und BEM-Outer-Skin-/Head-Layer
 - Entfernung der unteren Punkte („Hals“) vom BEM-Layer
 - Iterative Closest Point mit initialer Transformationsmatrix, EEG-Sensorpositionen und den „oberen“ Punkten des BEM-Layer
- Wenn ICP konvergiert, setze Transformationsmatrix im EMM und reiche dieses an Folgemodule weiter

Leadfield Interpolation

Überblick:

Kurzbeschreibung:	Interpoliert aus einer hoch aufgelösten Leadfield eine geringer aufgelöste Leadfield für EEG.
Moduleingang:	1x EMM-Paket
Modulaustrag:	1x EMM-Pakete gesetzter EEG-Leadfield
Vorbedingungen:	<ul style="list-style-type: none">• EEG-Sensorpositionen• Transformationsmatrix Kopfkoordinaten/AC-PC-Koordinaten• hoch aufgelöste Leadfield (HD-Leadfield)
GPU-Unterstützung:	-

Parameter:

Sensor file:	Optional: EEG-Sensorpositionen, Digitization/Isotrak
Leadfield file:	Hoch aufgelöste Leadfield

Beschreibung:

- Weitere Details in eigener Dokumentation
- Point Cloud Library zur Nearest-Neighbor-Suche
- Erstellung der HD-Leadfield
 - Extraktion virtueller Sensoren aus dem BEM-Head-Layer mit dem Tool: `eeg_sensor_generator`
 - Erstellung einer FIFF-Datei mit einer „EEG-Hilfsmodalität“, die nur die virtuellen Sensoren bzw. die EEG-Sensorpositionen enthält
 - ACHTUNG: diese EEG-Sensorpositionen sind im AC-PC-Koordinatensystem
 - Berechnung der HD-Leadfield für EEG mit Hilfe dieser FIFF-Datei und `mne_forward_solution`
- Interpolation:
 - Transformiere „tatsächliche“ EEG-Sensorpositionen in AC-PC-Koordinaten
 - Suche für jede EEG-Sensorposition die 4 nächsten Nachbarn in der HD-Leadfield
 - Berechne für jeden der 4 nächsten Punkte einen Wichtungsfaktor anhand des Abstands zur EEG-Sensorposition
 - Erstelle für jede EEG-Sensorposition eine Interpolation aus den 4 Leadfield-Vektoren und den zugehörigen Wichtungsfaktor
- Setze interpolierte EEG-Leadfield im EMM und reiche dieses an Folgemodule weiter

Head Position Estimation

Überblick:

Kurzbeschreibung:	Extrahiert HPI-Signale aus den Magnetometerdaten und berechnet die Transformation von Kopf-/EEG-Koordinaten in MEG-Device-Koordinaten.
Moduleingang:	1x EMM-Paket
Modulaustrag:	1x neues EMM-Paket mit HPI-Modalität (EMDHPI).
Vorbedingungen:	<ul style="list-style-type: none">• Frequenzen der HPI-Spulen angegeben• Position der HPI-Spulen bekannt (isotrak)• Positionen & Orientierung (e_z) der Magnetometer bekannt
GPU-Unterstützung:	-

Parameter:

HPI Frequency 1..n	Frequenzen der HPI-Spulen in Hz
Windows Size	Fensterlänge zur HPI-Signalextraktion in ms (kein Sliding Windows)
Max. Iterations	Maximal Anzahl der Iteration des Minimierungsalg. bis Abbruch
Epsilon	Schwellwert für Minimierungsalg. bis Abbruch
R_z, R_y', R_z''	Initiale Winkel für Rotation in z-y-z-Euler-Konvention
T_x, T_y, T_z	Initiale Translation/Verschiebung in x-, y-, z-Richtung

Beschreibung:

- Weitere Details in eigener Dokumentation
- Signal der HPI-Spulen werden vom MEG zusammen mit den Nutzdaten gemessen
- jede HPI-Spule sendet in einer bekannten Frequenz, i.d.R. $> 150\text{Hz}$
- HPI-Signalextraktion
 - FIR-Lowpass-Filter mit $\frac{1}{2} \cdot \min(f_1, f_2, \dots, f_n)$ (HPI-Spulenfrequenzen)
 - Berechnung des Amplitudenvektors aller HPI-Spulen an jedem Magnetometersensor mit Hilfe der bekannten HPI-Spulenfrequenzen
 - Es wird kein Sliding-Windows genutzt (Fenstergröße ist gleich der Schrittweite), daher sollte die Fenstergröße ein ganzzahliges Vielfaches der Blockgröße sein
- Head Position Estimation
 - Optimierung der Transformationsmatrix für die Transformation der HPI-Spulenpositionen von dem Kopf-/EEG-Koordinatensystem in das MEG-Device-Koordinatensystem
 - Minimierung von $B_{\text{Dipol}} - B_{\text{Amplitudenvektor}}$
 - Optimierung mit der Downhill-Simplex-Methode
 - 6 Parameter: 3 Rotation + 3 Translation \rightarrow Transformationsmatrix

Source Reconstruction

Überblick:

Kurzbeschreibung:	Rekonstruktion verteilter Quellen mit dem MN- oder WNM-Verfahren.
Moduleingang:	1x EMM-Paket
Modulaustrag:	1x neues EMM-Paket mit Quellaktivität (EMDSource)
Vorbedingungen:	Leadfield verfügbar
GPU-Unterstützung:	CUDA

Parameter:

Weighting Type:	Auswahl zwischen Minimum Norm und Weighted Minimum Norm
SNR:	Signal-Rausch-Verhältnis

Beschreibung:

- Initialisierung:
 - Rausch- & Datenkovarianzmatrix sind z.Z. Einheitsmatrix
 - Berechnung des inversen Operators \mathbf{G} , z.Z. einmalig bzw. bei manueller Änderung des SNR
- Rekonstruktion:
 - Berechne Average Reference über alle Kanäle der Modalität
 - Subtrahiere die Average Reference von jedem Kanal der Modalität
 - Matrixmultiplikation: `sourceActivity=G*data`
 - Erstelle EMDSource, setze die Quellaktivität und das zugrunde liegende EMD
 - Erstelle neues EMM, setze Modalitäten, Weitergabe an Folgemodule