Bijlage 1

In- en exclusie criteria van het overkoepelende onderzoek van het AMC.

Deel A:

Inclusion criteria	Exclusion criteria
Previous thalamotomy for the treatment of tremor Presence of tremor in at least the arm ipsilateral to surgery at the time of investigation Age above 18 years	MR-incompatible implanted metal bodies, including stereotactic implant for Deep Brain Stimulation. Other contraindications for MR (Claustrophobia, obesity, etc.) Previous brain surgery other than thalamotomy Use of medicines/drugs that could influence the performance during the tasks (such as anti-epileptic drugs, neurodepressants, etc.) Pregnancy or suspected pregnancy Incapability to give informed consent

Deel B:

Inclusion criteria	Exclusion criteria
Presence of tremor in at least one arm at the time of investigation Inclusion in the waiting list for DBS Age above 18 years	MR-incompatible implanted metal bodies including stereotactic implant for Deep Brain Stimulation. Other contraindications for MR (Claustrophobia, obesity, etc.) Previous brain surgery Use of medicines/drugs that could influence the performance during the tasks (such as anti-epileptic drugs, neurodepressants, etc.) Pregnancy or suspected pregnancy Incapability to give informed consent

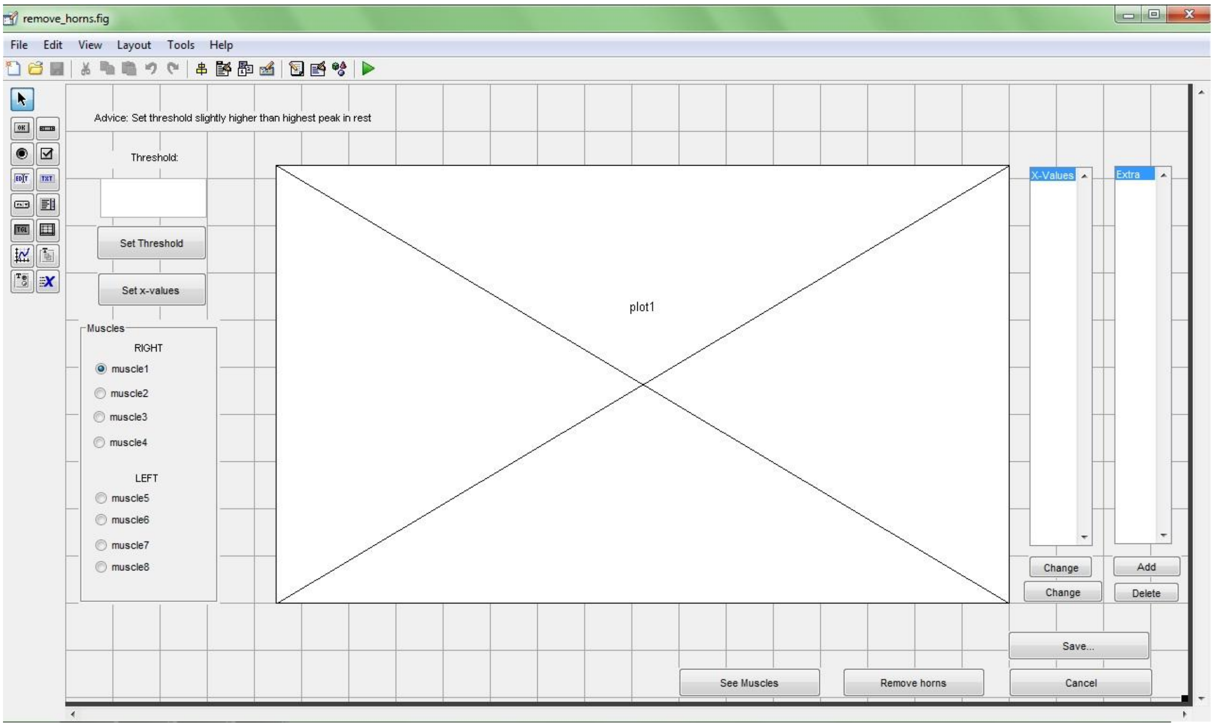
Bijlage 2

Geselecteerde spieren in stap 6 van EEGLab, bij het maken van modellen.

Patient	Protocol	Spieren bij stap 5	Spieren bij stap 6
6201	Tremor1	extensorR extensorL	2+6
6201	Tremor2	TricepsR EIPL	4+7
6202	Tremor1	ExtensorR BicepsL	1+8
6202	Tremor2	FIDR BicepsL	2+8
6203	Tremor1	ExtensorR BicepsL	2+6
6203	Tremor2	EIPR Flexor L	3+7
6204	Tremor1	extensorR extensorL	1+5
6204	Tremor2	extensorR extensorL	1+7
6205	Tremor1	extensorR extensorL	1+7
6205	Tremor2	FIDR extensorL	1+8
6401	Tremor1	extensorR extensorL	3+7
6401	Tremor2	flexorR extensorL	3+7

Bijlage 3

MatLab script van ‘peak artifact removal method’.



Advice: Set threshold slightly higher than highest peak in rest

Figuur 1. De gemaakte GUI in GUIDE (GUI Developer van Matlab)

```
function varargout = remove_horns(varargin)
% REMOVE_HORNS M-file for Remove_horns.fig
%
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Remove_horns

% Last Modified by GUIDE v2.5 04-Jun-2010 17:26:57
%
%=====================================================|\====
%+++++++++++++++++++++++++++++++++++++++++++++++++++++||\+++
% This script is made by: David, Nicole, Eric en Tom  ||\\
%+++++++++++++++++++++++++++++++++++++++++++++++++++++||+\\+
%=====================================================||==\\

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @remove_horns_OpeningFcn, ...
                   'gui_OutputFcn',  @remove_horns_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
```

```matlab
% --- Executes just before Remove_horns is made visible.
function remove_horns_OpeningFcn(hObject, eventdata, handles, varargin)
    emg_fmri_globals; % make sure this is the first call in the fn

% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Remove_horns (see VARARGIN)

% Choose default command line output for Remove_horns
    handles.output = hObject;
    handles.protocol = varargin{1};

% Set titles of muscles
    handles.current_muscle = 0;

    if EEG.nbchan<=0
        set(handles.txt_titel,'String','NO EMG DATA AVAILABLE!');
        %set(handles.txt_threshold,'Enable','off');
    else
        handles.current_muscle = 1;

        for iChannel=1:EEG.nbchan
            t = EEG.chanlocs(iChannel).labels;
            set(handles.(['radio_muscle' num2str(iChannel)]),'String',t);
        end
    end

% Update handles structure
    guidata(hObject, handles);

%Begin function
    EEG.remove_horns.threshold = 500;                          %standard value
    set(handles.threshold_input,'String',EEG.remove_horns.threshold);

    EEG.remove_horns.oud = EEG.data;                           %oud is temporary

    EEG.remove_horns.EMG_gem_comb = create_emg_gem_comb();
                          %create averaged signal of sum of all muscles
    EEG.remove_horns.blokjes_tijden_block = create_tijdstippen_actief_block();
                          %create the green lines (active areas according
                           to block with safe margin)
    EEG.remove_horns.blokjes_tijden_emg = create_tijdstippen_actief_emg();
                          %create the red lines (active areas according to EMG)
    EEG.remove_horns.artefacten_verwijderd = 0;
                          %boolean if artifacts are removed already or not.
    EEG.remove_horns.a = 4;
         %variable of how many times the modus should be set to modus value

    plot_tijdstippen_actief_threshold_gem(handles);         %Beginning plot

     set(handles.verwijder_horns,'Visible','off');
     set(handles.btn_group_muscles,'Visible','off');
     set(handles.btn_see_muscles,'Visible','off');
     set(handles.lst_x_waarden, 'Visible', 'off');
     set(handles.btn_change,'Visible','off');
     set(handles.btn_change2,'Visible','off');
     set(handles.btn_save,'Visible','off');
     set(handles.set_x_waarden, 'Visible', 'off');

% ===========================PLOTS=========================================

% --- Plot een grafiek met de gezette threshold in gemiddeld signaal
function plot_tijdstippen_actief_threshold_gem(handles)
    emg_fmri_globals;               % make sure this is the first call in the fn
    plot(handles.plot1, EEG.remove_horns.EMG_gem_comb);
    axis([0 EEG.pnts/EEG.remove_horns.factor 0 5000])
    title(['Average (100*less points) sum of all rectified muscles
                 Protocol: ' handles.protocol]);
    xlabel('EMG samples/100 ~ sec');
    ylabel('Power in microVolt');
    hold(handles.plot1, 'on');

    plot(ones(1,EEG.remove_horns.npnts_gem)*EEG.remove_horns.threshold);
    gridxy(EEG.remove_horns.blokjes_tijden_block,'Color','g','Linestyle',':');
    gridxy(EEG.remove_horns.blokjes_tijden_emg,'Color','r','Linestyle',':');
    % legend
    hold(handles.plot1, 'off');

% --- Plot een grafiek met de gezette threshold met signaal per spier
function plot_tijdstippen_actief_spier(handles)
    emg_fmri_globals;                % make sure this is the first call in the fn

    plot(handles.plot1, EEG.remove_horns.oud(handles.current_muscle,:));
    axis([0 EEG.pnts 0 2000])
    title(['Active areas in ' EEG.chanlocs(handles.current_muscle).labels]);
    xlabel('EMG samples');
    ylabel('Power in microVolt');
    hold(handles.plot1, 'on');

    gridxy(EEG.remove_horns.blokjes(handles.current_muscle,:),
        'Color','r','Linestyle',':');
    hold(handles.plot1, 'off');

    set(handles.lst_x_waarden,'string',
        EEG.remove_horns.blokjes(handles.current_muscle,:));
    set(handles.lst_x_waarden,'value',
        1:length(EEG.remove_horns.blokjes(handles.current_muscle,:)));

        % --- Plot een grafiek met de begin en eindwaarden van actieve stukjes;
                 signaal per spier
function plot_spier_met_verwijderd_artefact(handles)
    emg_fmri_globals;      % make sure this is the first call in the fn

    plot(handles.plot1, EEG.data(handles.current_muscle,:),'Color','r');
    axis([0 EEG.pnts 0 2000])
    title([EEG.chanlocs(handles.current_muscle).labels '           Erased
          movement artifacts in red       Protocol: ' handles.protocol]);
    xlabel('EMG samples');
    ylabel('Power in microVolt');
    hold(handles.plot1, 'on');
    plot(handles.plot1, EEG.remove_horns.oud(handles.current_muscle,:));
    hold(handles.plot1, 'off');


% =========================FUNCTIES========================================

% --- Gemiddelden maken en sommeren
function EMG_gem_comb = create_emg_gem_comb()
    emg_fmri_globals; % make sure this is the first call in the fn
    factor = 100;
    EEG.remove_horns.factor=factor;
    npnts_gem = int32(round(EEG.pnts/factor));
         %size matrix moet integer zijn, daarom zo expliciet opgeschreven
    EEG.remove_horns.npnts_gem=npnts_gem;
    EMG_gem = zeros(8,npnts_gem);
         %matrix met alle 8 spieren, 100 keer minder waarden -> beginwaarden 0
    EMG_gem_comb = zeros(1,npnts_gem);
         %matrix met de som van de 8 spieren (van EMG_gem, dus 100 keer minder
          waarden) -> beginwaarden 0

    for k=1:8
```

```
        for i=0:EEG.pnts/factor-1
            for j=1:factor
                EMG_gem(k,i+1) =
                     EMG_gem(k,i+1)+abs(EEG.remove_horns.oud(k,i*factor+j));
                     %Spier k, i-de element is abs som van 100 waarden in
                      EEG.remove_horns.oud
            end
            EMG_gem(k,i+1) = EMG_gem(k,i+1)/factor;
                 %Spier k, i-de element is nu gemiddelde van 100 waarden in
                  EEG.remove_horns.oud
        end
        EMG_gem_comb = EMG_gem_comb + EMG_gem(k,:);
                 %EMG_gem_comb is som van alle spieren
    end

% --- Achterhaal tijden van de blokjes zoals bekend in BlockEMG
function blokjes_tijden_block = create_tijdstippen_actief_block()
    emg_fmri_globals; % make sure this is the first call in the fn
    npts_blokje = EEG.srate*29;
         %in beide protocollen is het 29 secs per blokje
    npts_blokje_gem = round(npts_blokje/EEG.remove_horns.factor);
         %in gemiddelde signaal

    aantal_blokjes =
         floor(length(EEG.remove_horns.EMG_gem_comb)/npts_blokje_gem);
    blokjes_tijden_block = [];
    for i=1:aantal_blokjes
        if mod(i,2) == 0                             %als even getal...
            blokjes_tijden_block = [blokjes_tijden_block,
                 i*npts_blokje_gem+10000/EEG.remove_horns.factor];
                         %10000 waarden rechts ervan
        else                                         %als oneven ...
            blokjes_tijden_block = [blokjes_tijden_block, i*npts_blokje_gem-
                 6000/EEG.remove_horns.factor];      %6000 waarden links ervan
        end
    end
    blokjes_tijden_block = [blokjes_tijden_block,
         length(EEG.remove_horns.EMG_gem_comb)];  %de laatste waarde nog toevoegen
    aantal_blokjes=aantal_blokjes+1;
    EEG.remove_horns.aantal_blokjes = aantal_blokjes;

% --- eerste/laatste x-waarde met y-waarde boven threshold aanwijzen als begin en
eindpunt actieve gebiedje
function blokjes_tijden_emg = create_tijdstippen_actief_emg()
    emg_fmri_globals;              % make sure this is the first call in the fn
    blokjes_tijden_emg = EEG.remove_horns.blokjes_tijden_block;
    for i=1:EEG.remove_horns.aantal_blokjes
        if mod(i,2) == 0          %als even getal...van rechts naar links lopen
            j=0;
            while EEG.remove_horns.EMG_gem_comb(blokjes_tijden_emg(i)-j) <
                         EEG.remove_horns.threshold
                j=j+1;
            end
            blokjes_tijden_emg(i) = blokjes_tijden_emg(i)-j;
        else                      %als oneven getal...van links naar rechts lopen
            j=0;
            while EEG.remove_horns.EMG_gem_comb(blokjes_tijden_emg(i)+j) <
                         EEG.remove_horns.threshold
                j=j+1;
            end
            blokjes_tijden_emg(i) = blokjes_tijden_emg(i)+j;
        end
    end


% --- reken de modus van actieve en rust stukjes uit.
    function modi = create_modi(handles)
        emg_fmri_globals;         % make sure this is the first call in the fn
        modi = zeros(8,EEG.remove_horns.aantal_blokjes);
        for i=1:8                                    %per spier
            for j=1:EEG.remove_horns.aantal_blokjes  %voor alle blokjes
                begin = int32(EEG.remove_horns.blokjes(i,j))+3000;
                eind = int32(EEG.remove_horns.blokjes(i,j+1))-3000;

                ac=abs(EEG.remove_horns.oud(i,begin:eind));
                          %ac zijn waarden van activiteitsgebiedje gerectificeerd
                A = zeros(1,length(ac)); %lege array met goede lengte
                H = imag(hilbert(ac));   %imaginaire gedeelte van hilbert
                                            transformatie van actieve gebiedje
                for k=1:length(ac)    %voor alle waarden van het actieve gebiedje
                    A(k) = sqrt(ac(k)^2 + H(k)^2);  %pythagoras van reëel en
                          imaginaire gedeelte van hilbert is momentane aplitudes
                end

                histogram = hist(A,100);     %momentane amplitudes in 100 bins
                                                   verdeeld in histogram
                maximum = max(A);
                bin_grootte = maximum/100;

                modus = find(max(histogram) == histogram);%modus is de x waarde
                                           waar het max van histogram zit (vaakst
                                           voorkomende momentane amplitude)
                modi(i,j) = modus(1)*bin_grootte;   %in geval van meerdere
                                  maxima, neemt hij eerste en slaat op in array
            end
        end



% --- Outputs from this function are returned to the command line.
function varargout = remove_horns_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


%========================BUTTONS==========================================

 % --- Executes on button press in btn_set_threshold
 function btn_set_threshold_Callback(hObject, eventdata, handles)
    emg_fmri_globals; % make sure this is the first call in the fn
% hObject    handle to btn_set_threshold (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
    EEG.remove_horns.threshold =
         str2double(get(handles.threshold_input,'String'));
    EEG.remove_horns.blokjes_tijden_emg = create_tijdstippen_actief_emg();
    plot_tijdstippen_actief_threshold_gem(handles);
    set(handles.set_x_waarden, 'Visible', 'on');


% --- Executes on button press in btn_see_muscles.
function btn_see_muscles_Callback(hObject, eventdata, handles)
  emg_fmri_globals; % make sure this is the first call in the fn

  EEG.remove_horns.blokjes_tijden_emg =
                 EEG.remove_horns.blokjes_tijden_emg*EEG.remove_horns.factor;
  EEG.remove_horns.blokjes = ones(8,EEG.remove_horns.aantal_blokjes+1);
  for i=1:8
    EEG.remove_horns.blokjes(i,2:end) = EEG.remove_horns.blokjes_tijden_emg;
  end                     %kollom 1 zijn alleen enen
  handles.current_muscle = 1;
  plot_tijdstippen_actief_spier(handles);
  set(handles.set_x_waarden,'Visible','off');
  set(handles.btn_set_threshold,'Visible','off');
  set(handles.btn_group_muscles, 'Visible', 'on');
  set(handles.btn_change2, 'Visible', 'on');
  set(handles.btn_change, 'Visible', 'off');
  set(handles.verwijder_horns, 'Visible', 'on');
  set(handles.threshold_input, 'Visible', 'off');
  set(handles.text3, 'Visible', 'off');
  set(handles.text6, 'Visible', 'off');
  set(handles.btn_see_muscles,'Visible','off');


% --- Executes when selected object is changed in button_group_muscles.
function btn_group_muscles_SelectionChangeFcn(hObject, eventdata, handles)
    emg_fmri_globals;
    str = get(eventdata.NewValue,'Tag'); % Get Tag of selected object.
    handles.current_muscle  = str2double(str(end));
                                           % last character should be muscle #
    % Update handles structure
    guidata(hObject, handles);

    if EEG.remove_horns.artefacten_verwijderd == 0
        plot_tijdstippen_actief_spier(handles);
    else
        plot_spier_met_verwijderd_artefact(handles);
    end

% --- Executes on button press in btn_cancel.
function btn_cancel_Callback(hObject, eventdata, handles)
% hObject    handle to btn_cancel (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
    close;


% --- Textbox met een waarde voor de threshold
function threshold_input_Callback(hObject, eventdata, handles)
% hObject    handle to threshold_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of threshold_input as text
%        str2double(get(hObject,'String')) returns contents of threshold_input as a
double


% --- Executes during object creation, after setting all properties.
function threshold_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to threshold_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
         if ispc && isequal(get(hObject,'BackgroundColor'),
                          get(0,'defaultUicontrolBackgroundColor'))
             set(hObject,'BackgroundColor','white');
         end


% --- Executes on selection btn_change in lst_x_waarden.
function lst_x_waarden_Callback(hObject, eventdata, handles)
% hObject    handle to lst_x_waarden (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: contents = cellstr(get(hObject,'String')) returns lst_x_waarden contents
as cell array contents{get(hObject,'Value')} returns selected item from
lst_x_waarden


% --- Executes during object creation, after setting all properties.
function lst_x_waarden_CreateFcn(hObject, eventdata, handles)
% hObject    handle to lst_x_waarden (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
        if ispc && isequal(get(hObject,'BackgroundColor'),
                         get(0,'defaultUicontrolBackgroundColor'))
           set(hObject,'BackgroundColor','white');
        end


% --- Executes on button press in verwijder_horns.
function verwijder_horns_Callback(hObject, eventdata, handles)
% hObject    handle to verwijder_horns (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
    emg_fmri_globals; % make sure this is the first call in the fn

    %PLEASE WAIT WHILE CALCULATING nog inbouwen
    EEG.remove_horns.modi = create_modi(handles);

    for i=1:8                                      %per spier
        for j=1:EEG.remove_horns.aantal_blokjes    %voor alle blokjes
            begin = int32(EEG.remove_horns.blokjes(i,j));
                                  % j = nummer van actief blokje (max = 24 of 16)
            eind = int32(EEG.remove_horns.blokjes(i,j+1));
            if mod(j,2) == 0    %als actief stukje, dus niet rust
                for m=begin-3000:begin+3000
                                  %for alle waarden rond begin van actief blokje
                    if abs(EEG.remove_horns.oud(i,m)) >  %als groter dan a*modus
                                  EEG.remove_horns.a*EEG.remove_horns.modi(i,j)
                        EEG.remove_horns.oud(i,m) = EEG.remove_horns.modi(i,j);
                                           %dan krijgt het waarde van modus
                    end
                end
                if j<EEG.remove_horns.aantal_blokjes
                                           %als niet laatste blokje, dan...
                    for n=eind-3000:eind+3000
                                  %for alle waarden rond eind huidige blokje
                        if abs(EEG.remove_horns.oud(i,n)) >
                                  EEG.remove_horns.a*EEG.remove_horns.modi(i,j)
                                                            %als groter dan a*modus
                            EEG.remove_horns.oud(i,n)=EEG.remove_horns.modi(i,j);
                                           %dan krijgt het waarde van modus
                        end
                    end
                else                      %als laatste blokje, dan...
                    laatste_stukje = EEG.pnts-eind;
                          %veiligheid, zodat we niet meer dan EEG.pnts bereiken
                    for n=eind-3000:eind+laatste_stukje
                        if abs(EEG.remove_horns.oud(i,n)) >
                                  EEG.remove_horns.a*EEG.remove_horns.modi(i,j)
                            EEG.remove_horns.oud(i,n)=EEG.remove_horns.modi(i,j);
                        end
                    end
                end
%             else                            %voor alle rust stukjes
%                 for m=begin:begin+3000
%                                 for alle waarden rond begin van actief blokje
%                     if abs(EEG.remove_horns.oud(i,m)) >
%                                 EEG.remove_horns.a*EEG.remove_horns.modi(i,j)
%                                                  %als groter dan 6*modus
%                         EEG.remove_horns.oud(i,m) = EEG.remove_horns.modi(i,j);
%                                          %dan krijgt het waarde van modus
%                     end
%                 end
%                 for n=eind-3000:eind
%                     if abs(EEG.remove_horns.oud(i,n)) >
%                                 EEG.remove_horns.a*EEG.remove_horns.modi(i,j)
%                         EEG.remove_horns.oud(i,n)=EEG.remove_horns.modi(i,j);
%                     end
%                 end
            end
        end
    end

    handles.current_muscle = 1;
    EEG.remove_horns.artefacten_verwijderd = 1;

    guidata(hObject, handles);

    plot_spier_met_verwijderd_artefact(handles);
    EEG.remove_horns.extra_peaks = [];

    set(handles.radio_muscle1,'Value',1.0);
    set(handles.verwijder_horns,'Visible','off');
    set(handles.lst_x_waarden, 'Visible', 'off');
    set(handles.btn_change2, 'Visible', 'off');
    set(handles.btn_save, 'Visible', 'on');
    set(handles.lst_extra_values, 'Visible', 'on');
    set(handles.btn_add_peak, 'Visible', 'on');

% --- Executes on button press in set_x_waarden.
function set_x_waarden_Callback(hObject, eventdata, handles)
% hObject    handle to set_x_waarden (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
    emg_fmri_globals; % make sure this is the first call in the fn
    set(handles.lst_x_waarden,'String', EEG.remove_horns.blokjes_tijden_emg);
    set(handles.lst_x_waarden,'value',
                 1:length(EEG.remove_horns.blokjes_tijden_emg));
    set(handles.lst_x_waarden, 'Visible', 'on');
    set(handles.btn_change,'Visible','on');
    set(handles.btn_see_muscles,'Visible','on');
    set(handles.set_x_waarden,'Visible','off');

% --- Executes on button press in btn_change.
function btn_change_Callback(hObject, eventdata, handles)
% hObject    handle to btn_change (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
    emg_fmri_globals;
    ongewenste_x_waarde = get(handles.lst_x_waarden,'value');
    [u,v]=getpts;
    EEG.remove_horns.blokjes_tijden_emg(ongewenste_x_waarde) = round(u(1));
    EEG.remove_horns.blokjes_tijden_emg =
                 sort(EEG.remove_horns.blokjes_tijden_emg);
    set(handles.lst_x_waarden,'string', EEG.remove_horns.blokjes_tijden_emg);
    set(handles.lst_x_waarden,'value',
                 1:length(EEG.remove_horns.blokjes_tijden_emg));
    plot_tijdstippen_actief_threshold_gem(handles);

% --- Executes on button press in btn_change2.
function btn_change2_Callback(hObject, eventdata, handles)
% hObject    handle to btn_change2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
    emg_fmri_globals;

    ongewenste_x_waarde = get(handles.lst_x_waarden,'value');
    [u,v]=getpts;
    EEG.remove_horns.blokjes(handles.current_muscle,ongewenste_x_waarde) =
                                  round(u(1));

    %set(handles.lst_x_waarden,'string',
                 EEG.remove_horns.blokjes(handles.current_muscle,:));
    %set(handles.lst_x_waarden,'value',
                 1:length(EEG.remove_horns.blokjes(handles.current_muscle,:)));
    EEG.remove_horns.blokjes = sort(EEG.remove_horns.blokjes,2);
    plot_tijdstippen_actief_spier(handles);


% --- Executes on button press in btn_save.
function btn_save_Callback(hObject, eventdata, handles)
% hObject    handle to btn_save (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
    emg_fmri_globals;

    EEG.data = EEG.remove_horns.oud;
    clear EEG.remove_horns.oud;

    ppdir = fullfile(EMG_fMRI_study_dir,'pp',EMG_fMRI_patient);
    emgdir = fullfile(ppdir,handles.protocol,'emg');

     filepath = fullfile(emgdir,'emg_corrected2.mat');
     [filename pathname ] = uiputfile('*.mat','Save corrected data as',filepath);

     if filename~=0
        filepath = fullfile(pathname, filename);
        message = ['saving ' filepath];
        disp(message);
     end
     save(filepath, 'EEG')
     close;

% --- Executes on selection change in lst_extra_values.
function lst_extra_values_Callback(hObject, eventdata, handles)
% hObject    handle to lst_extra_values (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns lst_extra_values
contents as cell array contents{get(hObject,'Value')} returns selected item from
lst_extra_values

% --- Executes during object creation, after setting all properties.
function lst_extra_values_CreateFcn(hObject, eventdata, handles)
% hObject    handle to lst_extra_values (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
    if ispc && isequal(get(hObject,'BackgroundColor'),
                 get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

% --- Executes on button press in btn_add_peak.
function btn_add_peak_Callback(hObject, eventdata, handles)
% hObject    handle to btn_add_peak (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
    emg_fmri_globals;
    EEG.remove_horns.extra_peaks = [];

    k = handles.current_muscle;

    [u,v]=getpts;
    nieuwe_pieken = round(u);

    for i = 1:length(nieuwe_pieken)
        tijdelijk = sort([EEG.remove_horns.blokjes(k,:) nieuwe_pieken(i)]);
        actief_gebied = find(tijdelijk == nieuwe_pieken(i))-1;
        if actief_gebied(1) > length(EEG.remove_horns.modi(k,:))
            actief_gebied = length(EEG.remove_horns.modi(k,:));
        end
        modus = EEG.remove_horns.modi(k,actief_gebied(1));
        for j = nieuwe_pieken(i)-1000:nieuwe_pieken(i)+1000
            if abs(EEG.remove_horns.oud(k,j)) > EEG.remove_horns.a*modus
               EEG.remove_horns.oud(k,j)=modus;
            end
        end
    end

    EEG.remove_horns.extra_peaks = sort([EEG.remove_horns.extra_peaks
                 nieuwe_pieken]);

    set(handles.lst_extra_values,'string', EEG.remove_horns.extra_peaks);
    set(handles.lst_extra_values,'value',
                         1:length(EEG.remove_horns.extra_peaks));

    % Update handles structure
    guidata(hObject, handles);

    plot_spier_met_verwijderd_artefact(handles);



% --- Executes on button press in btn_delete.
function btn_delete_Callback(hObject, eventdata, handles)
% hObject    handle to btn_delete (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

Bijlage 4

EEGLab EMG_fMRI Manual

Based on:

- -Day of the Scan
  by Robin Huizing
- -Guide to Artifact Cleaning of the EMG
  by Johan van der Meer

Caroline Mattijssen
Marjanne Bom
Bauke Kogelman
Merijn Eskes

modifications by:
Paul Groot
Eric Schat
Nicole van Klink
David Meijer
Tom Konert

Scanning Procedure

Before the scan

- Prepare the EMG and bring your laptop.
- Test EMG and batteries
- Check the fMRI room (beamer, mirror, screen)

Attach the EMG wires and check the signal. Replace the wires if the signal is not good enough.

After the scan

All the original data must be copied to the local network in the following directory and subfolders. For each new patient you have to create multiple files and a directory, take e.g. patient 6666 and study Tremor.
Create directories:

- x:\onderzoek\fMRI\Tremor\ruw\6666
- x:\onderzoek\fMRI\Tremor\ruw\6666\parrec
- x:\onderzoek\fMRI\Tremor\ruw\6666\trc

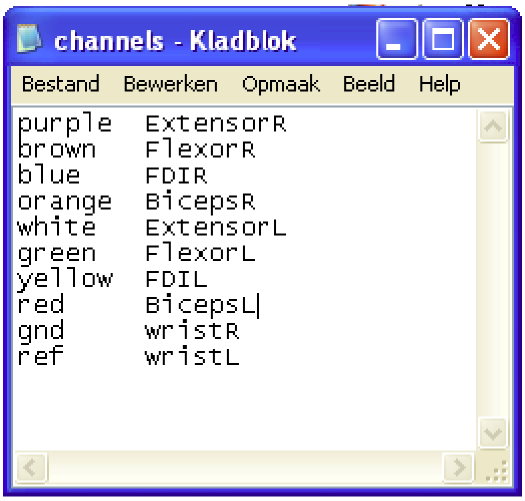
Create files in the directory x:\onderzoek\fMRI\Tremor\ruw\6666:

- channels.txt
- files.txt
- notes.txt

The last thing to do is to copy the data from the MRI computer and EMG computer:

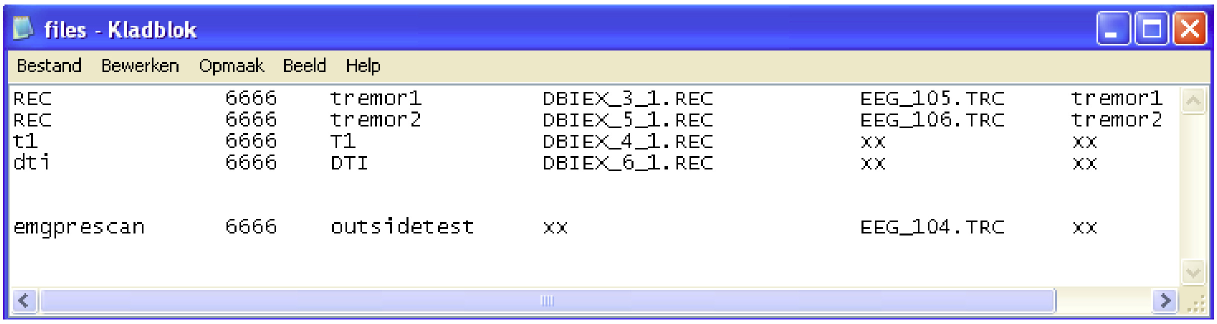
- MRI computer → parrec files → in x:\onderzoek\fMRI\Tremor\ruw\6666\parrec
- EMG computer → trc files → in x:\onderzoek\fMRI\Tremor\ruw\6666\trc

In x:\onderzoek\fMRI\Tremor\ruw there are examples of what the channels.txt, the notes.txt and the files.txt must look like. Below are some screenshots of the files.:



```
purple  ExtensorR
brown   FlexorR
blue    FDIR
orange  BicepsR
white   ExtensorL
green   FlexorL
yellow  FDIL
red     BicepsL
gnd     wristR
ref     wristL
```

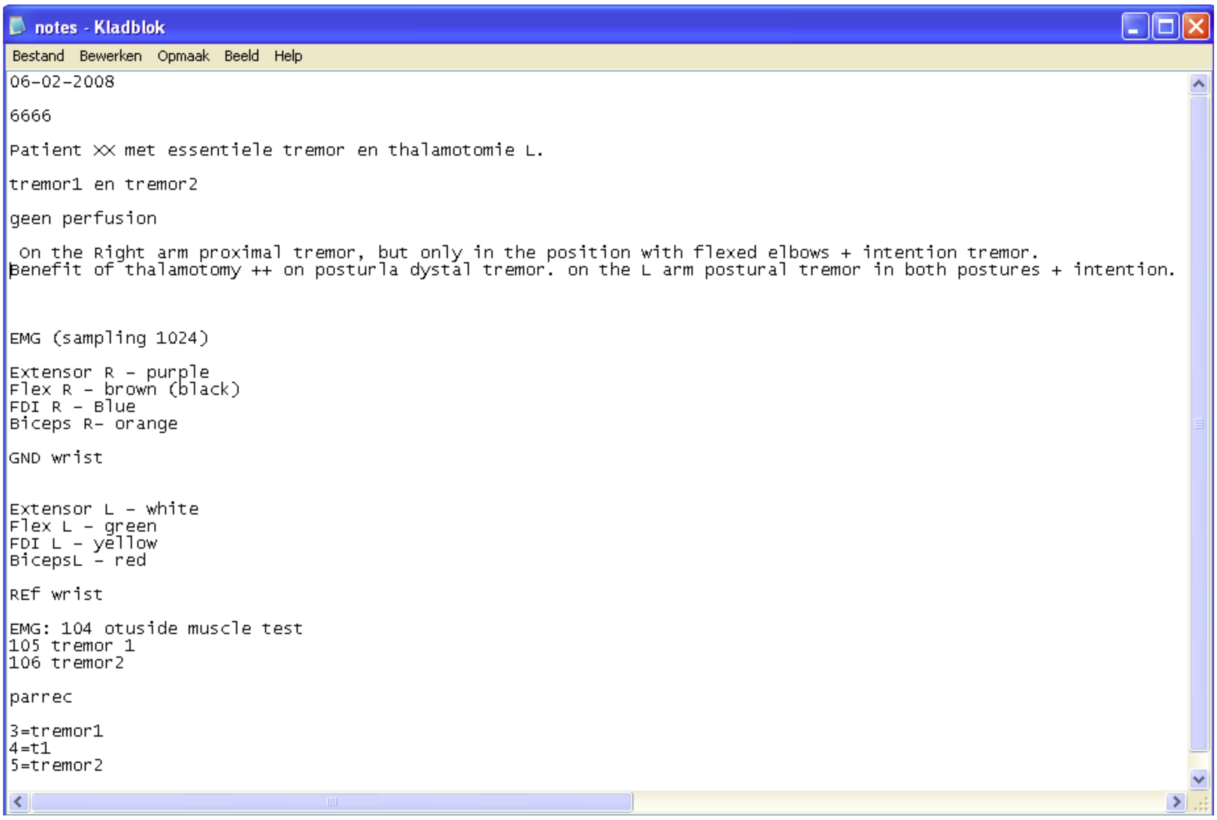
Figuur 1, Example of notes.txt



```
REC           6666    tremor1       DBIEX_3_1.REC        EEG_105.TRC    tremor1
REC           6666    tremor2       DBIEX_5_1.REC        EEG_106.TRC    tremor2
t1            6666    T1            DBIEX_4_1.REC        xx             xx
dti           6666    DTI           DBIEX_6_1.REC        xx             xx


emgprescan    6666    outsidetest   xx                   EEG_104.TRC    xx
```

Figuur 2, Example of files.txt



```
06-02-2008

6666

Patient XX met essentiele tremor en thalamotomie L.

tremor1 en tremor2

geen perfusion

 On the Right arm proximal tremor, but only in the position with flexed elbows + intention tremor.
Benefit of thalamotomy ++ on posturla dystal tremor. on the L arm postural tremor in both postures + intention.


EMG (sampling 1024)

Extensor R - purple
Flex R - brown (black)
FDI R - Blue
Biceps R- orange

GND wrist


Extensor L - white
Flex L - green
FDI L - yellow
BicepsL - red

REf wrist

EMG: 104 otuside muscle test
105 tremor 1
106 tremor2

parrec

3=tremor1
4=t1
5=tremor2
```

Figuur 3, Example of notes.txt

Conversion of raw files

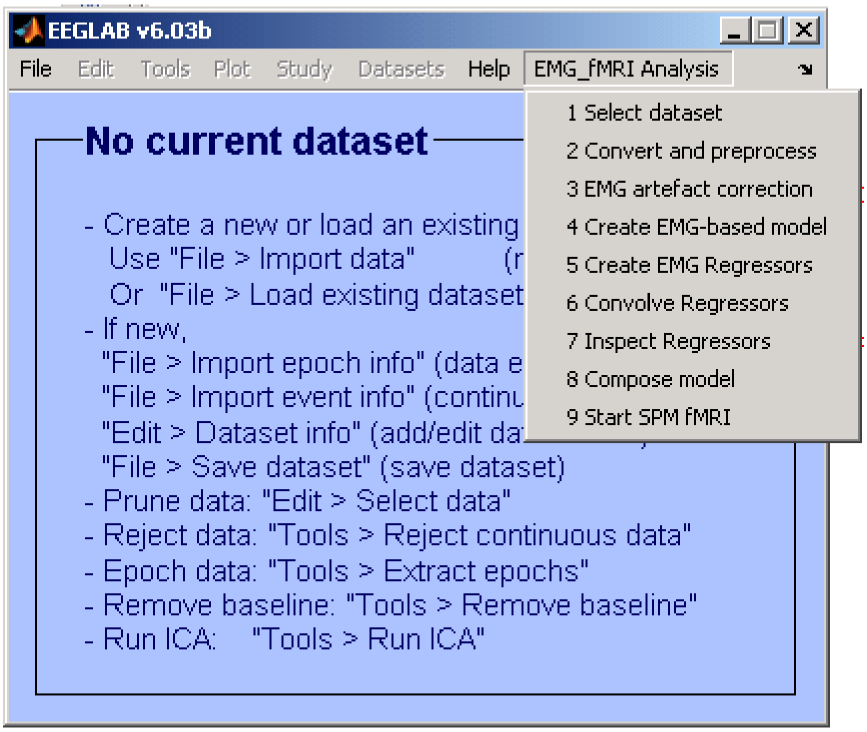
Before the data can be processed in Matlab the raw files must be converted. This can be done directly after the scanning procedure or on another day.

[obsolete] Cygwin: Transformation of parrec files (MRI)

Conversion of PAR/REC files is now included in the EMG_fMRI GUI by means of calling functions of the r2agui application (http://r2agui.sourceforge.net/).

EMG_fMRI as EEGLab plug-in in Matlab

The EMG data will be managed in Matlab using a graphical user interface (GUI). The GUI can be started by running startemgfmri.m from within matlab. A shortcut to this script can be created on the Matlab shortcut's toolbar for convenience. The start-up script will add the required directories to the Matlab path and start EEGLab (http://sccn.ucsd.edu/eeglab). The EMG_fMRI GUI will be loaded as plug-in and add a new submenu to the right of EEGLab menu.



No current dataset

- Create a new or load an existing
  Use "File > Import data"   (r
  Or "File > Load existing dataset
- If new,
  "File > Import epoch info" (data e
  "File > Import event info" (continu
  "Edit > Dataset info" (add/edit da
  "File > Save dataset" (save dataset)
- Prune data: "Edit > Select data"
- Reject data: "Tools > Reject continuous data"
- Epoch data: "Tools > Extract epochs"
- Remove baseline: "Tools > Remove baseline"
- Run ICA: "Tools > Run ICA"

Select Dataset

The following steps are necessary for all that follows! If Matlab is closed and restarted again, this step needs to be done! So keep in mind that every time you are going to work with the EMG_fMRI plug-in execute the following:

- In EEGLab
- Click on the menu 'EMG_fMRI analysis'
- Click on 'Select Dataset'

A new popup window will occur. In this screen the study (Tremor) and patient identifier must be filled in and the study directory must be chosen (x:\Onderzoek\fMRI\Tremor). Furthermore the protocols must be filled in, e.g. tremor1 & tremor2. Up to 5 protocols can be defined. (Just leave unused protocol boxes empty.)

Transformation of trc and Log files and fMRI preprocessing

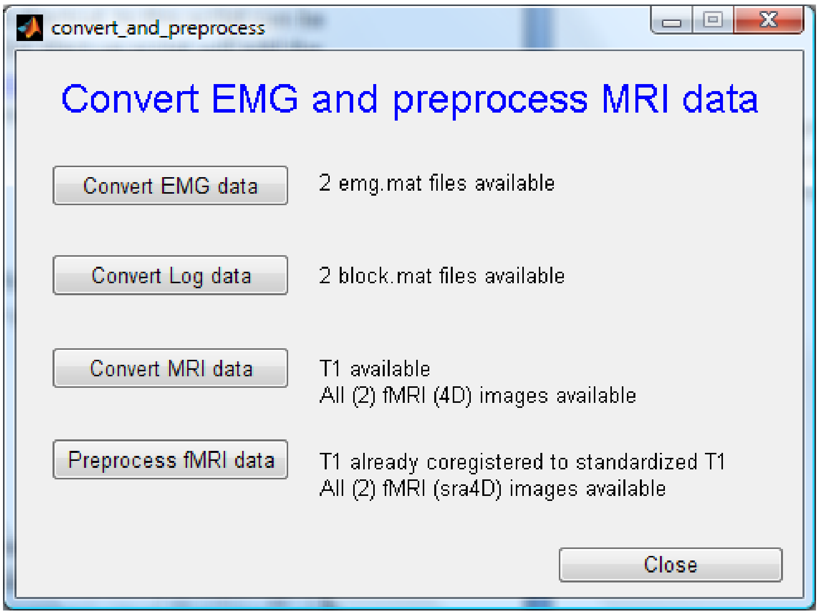
Make sure the previous step ‘Select Dataset’ has been done! Then the following steps will transform trc files to emg.mat files in all the folders and create two extra files inside the EMG folder (only if an emg measurement outside the scanner was done):

- In EEGLab
- Click on the menu ‘EMG_fMRI analysis’
- Click on the menu ‘Convert and pre-process’

The pop up that will appear has three conversion options and one pre-processing option.

- Convert EMG Data → starts batch_emg_to_mat → transforms trc into emg.mat files
- Convert Log Data → starts batch_log_to_mat → transforms log files to block.mat and event.mat (if there are no log files, this script still has to be run for creating block.mat).
- Convert MRI Data → starts batch_parrec_to_analyze→ transforms PAR/REC into analyze (hdr/img) files
- Preprocess fMRI Data → starts SPM5 and batch_fmri_preprocessing → this will execute a bunch of spatial preprocessing methods in SPM. (If you choose this option, eeglab will be closed and preprocessing of the fMRI data will be done in SPM fMRI. After SPM fMRI is done you need to restart EEGLab with the following command in Matlab: “eeglab redraw”.(The command window is the place where you can execute commands.) These steps require time to complete:
    - Slice timing correction
    - Reorienting image
    - Coregistration
    - Normalization
    - Smoothing

The pre-processing of EMG data will take approximately 10 minutes, the pre-processing of log data will take less then 20 seconds and pre-processing of fMRI data will take approximately 90 to 120 minutes.



Convert EMG and preprocess MRI data

2 emg.mat files available

2 block.mat files available

T1 available
All (2) fMRI (4D) images available

T1 already coregistered to standardized T1
All (2) fMRI (sra4D) images available

EMG artefact correction

The EMG signal is full of artefacts which basically consists of two major components:

- From moving wires in the 3Tesla field. The greater the velocity of the wires, the bigger the artefact. This one is the easiest to remove by simply using a filter.
- From the EPI sequence used by the scanner to record images. This artefact originates from the fast-changing 10 mTesla field. Although it is a relatively small field, the artefact is huge, because of the fast changes. This artefact consists of two types:
    - Volume artefact, once in every TR (e.g. 2.9 secs)
    - Slice artefact, once every slice (~60msec)

Correction procedure

The data will be corrected in EEGLab, using the EMG_fMRI plug-in. This plug-in uses EEGLab functions and the FMRIB plug-in developed by the University of Oxford Centre. It also uses some Matlab scripts created by Johan van der Meer.

- In EEGLab
- Click on the menu ‘EMG_fMRI analysis’
- Click on ‘EMG artefact correction’

After selecting the desired protocol a file selection dialog will be shown to select the raw EMG file. This will normally be the emg.mat file. After loading the selected file, the data will be made bipolair (only required for channels 5-12). Next, a dialog window will appear that has two choices, Personalised and New. These will be described below.

The New method

If the new method is selected, a fully automated artefact removal script (emg_corr) will be run. This is the script J.N. van der Meer has made to remove the artefacts. The script will perform the following steps:

- load emg, make bipolar, add muscle names, add slice triggers (using channel 1-4 with max value) and save as emg_added_slicetriggers.mat
- remove EMG before first and after last volume and save as emg_added_slicetriggers_revised
- apply high-pass, call do_new_slice_timing3 (using channel 1-4 with max value) and save as state_after_slice_timing.mat
- call do_volume_correction and save as state_after_volume_correction.mat
- call do_pca and save as state_after_pca.mat
- apply high pass, call do_anc, apply low pass, insert burst segment events, remove outside measurement data, remove slice events, add model events and save as emg_corrected.mat

The Personalised method

Using the personalised method most steps can be manually executed and values can be manually assigned. The EMG data is processed per protocol. The processed EMG signal can be inspected and saved at any point. It is also possible to load, edit and save a profile containing parameter values. With the inspect data button two plots will show up. One plot showing the power spectrum and one plot showing the EMG data. The save button saves the EMG data into the ../protocol/emg directory (e.g. ../tremor1/emg/). The filename will be: emg_oude_methode_xxx_year.month.day.time. The following functions are available:

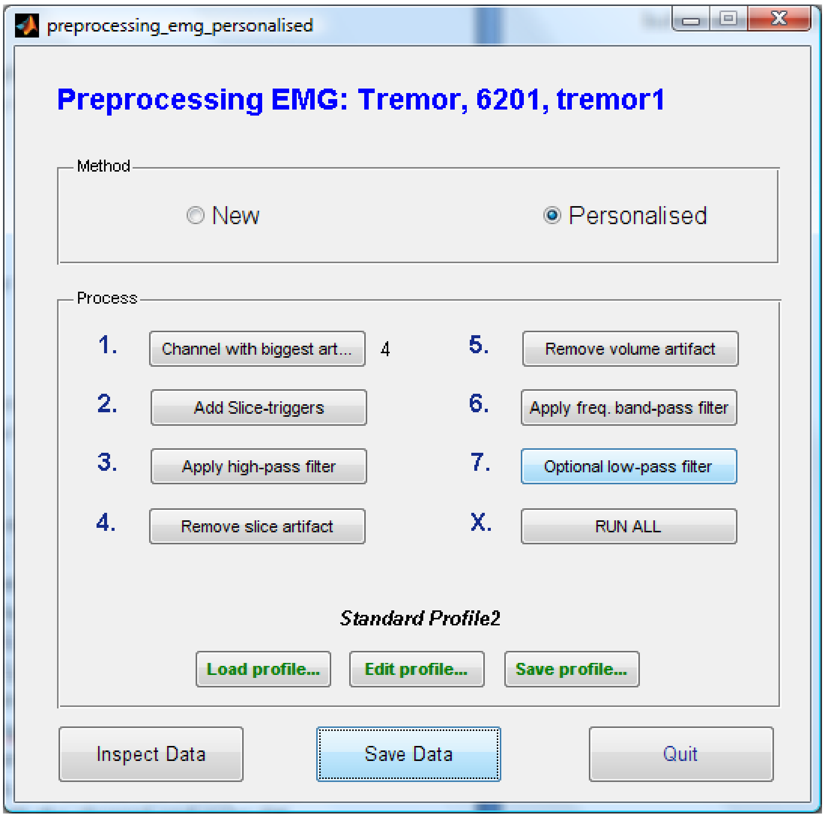
1. Choose Channel: will make the data bipolar using emg_make_bipolar. After this, a plot will be displayed with the bipolar EMG data. Then a popup will be displayed where a channel can

be chosen. Look at the channels and choose the one with the greatest artefact and click on the bullet consistent with the channel and press OK. This channel will be used for the placing of the slice triggers. To have a good view fill in a value of 20000 and go to e.g. slice 80.

2. Add Slice-triggers: runs the emg_add_slicetriggers script which add slice triggers to the data.
3. Apply Highpass filter: will apply a highpass filter on the data with the desired cutoff frequency. The standard values is 25.
4. Remove slice artifact: will start FMRIB Tools → FASTR. This will remove slice artifacts. The standard values are 0, 10, 10 and 0.07. Use dots, not commas! This will take ~10 min.
5. Remove volume artifact: will start FMRIB Tools → FASTR. This will remove volume artifacts. The standard values are 0, 10, 10 and 0.03. Use dots not commas! This will take ~10 min.
6. Apply freq. bandpass filter: runs emg_filter_bandpass. The standard values are 20 and 250.
7. Optional lowpass filter: is not needed, and does not have standard values.

X. RUN ALL. This will execute all of the above steps with the current profile.

Now it is done and the data can be inspected, if it is OK, save the data using the save data button and do not forget to save your profile if you want to use the settings again.


preprocessing_emg_personalised

Preprocessing EMG: Tremor, 6201, tremor1

Method

○ New ● Personalised

Process

1. Channel with biggest art... 4

Standard Profile2

Remove peak artefacts

In 2010 students of the Universeit Twente were asked to write a new method for artefact correction. The main target were the peaks in the beginning and the end of the activation periods seen on the EMG corrected signal. An activation period is defined as the time from the initiation till the termination of an action which is done by the patient during a fMRI-EMG co-registration. Removing these peaks would make the EMG data more 'clean' for analysis. See also 'Verbetering van analysetechnieken van gecombineerde EMG-fMRI techniek voor het bepalen van de doelstructuur voor diepe herselektroden bij essentiële tremor' written by:

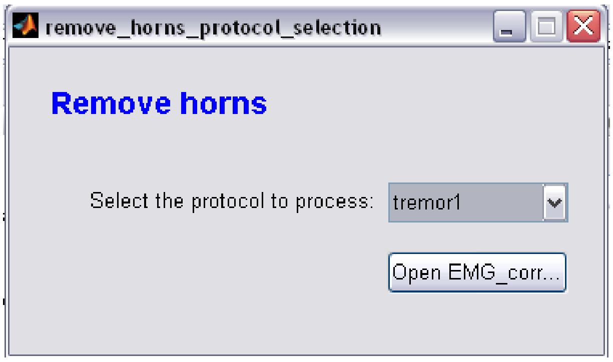
- Eric Schat (e.schat@student.utwente.nl)
- Nicole van Klink (n.e.c.vanklink@student.utwente.nl)
- David Meijer (d.meijer-1@student.utwente.nl)
- Tom Konert (t.konert@student.utwente.nl)

Their method for artefact correction is based on targeting the peaks and giving 1000 samples left and right from the peaks the value of the envelope-mode of the activation period after a Hilbert transformation.

To remove the artefacts in the EMG corrected signal, in EEGLab do the following:
-In EEGLab
-Click on the menu 'EMG_fMRI Analysis'
-Click on the 'Remove Artefacts'

A window will pop up with the following option:

- Open EMG_corrected

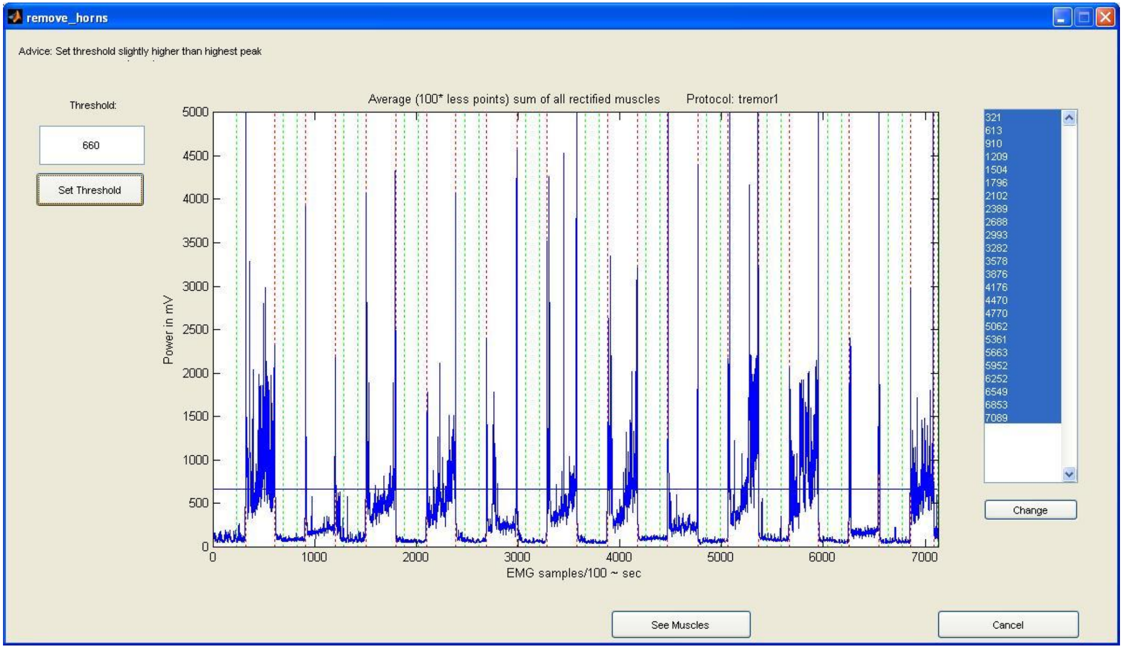


Remove horns

Select the protocol to process: tremor1

Open EMG_corrected (remove artefacts)

- Choose the EMG corrected data
- An EMG signal shows up which consists of the sum of the signal of all eight muscles together



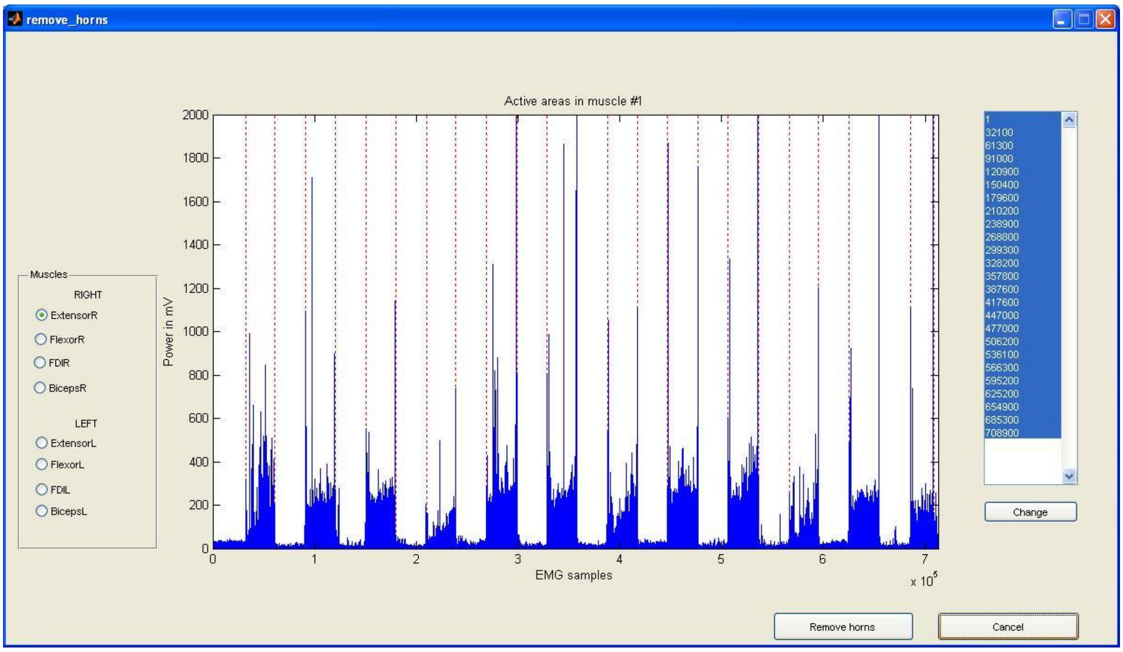
Advice: Set threshold slightly higher than highest peak

Threshold: 660

Protocol: tremor1

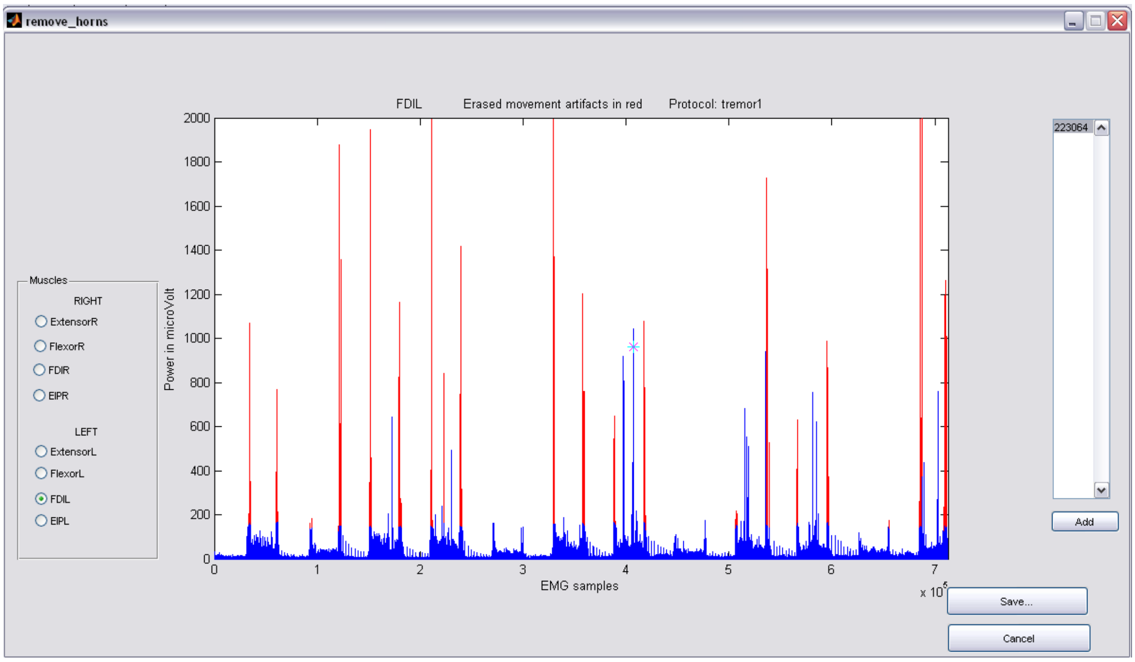
321
613
910
1209
1504
1796
2102
2389
2688
2993
3282
3578
3876
4176
4470
4770
5062
5361
5663
5952
6252
6549
6853
7089

- Determine the threshold so that the peaks at the beginning and the end of each activation period are included. The red vertical lines show the beginning and end of an activation period. Advice: Set threshold slightly higher than the highest peak in a rest period.
- If you're satisfied with the threshold click 'Set x-values'. All the x-values of the beginning and end of each activation period will show up in a list.
- Click a value in the list and then click the 'change' button if you want to change a x-value. After that click the place in the EMG signal that you want as your new x-value and press 'Enter'. After this your selected x-value will change in the list.
- If you're satisfied click 'See muscles' to view the beginning and end of an activation period per muscle.



```
1
32100
61300
91000
120900
150400
179600
210200
238900
268800
299300
328200
357800
387600
417600
447000
477000
506200
536100
566300
595200
625200
654900
685300
708900
```

- You will notice a sidebar with all the muscles named. You can go through each muscle and you can change x-values per muscle. The changes you make per muscle won't affect the others muscles. If you want to change x-values repeat the same actions as you did in the fifth step.
- If you've seen all the muscles and you're satisfied click 'remove horns'.
- After this the peaks are removed and you can see what is removed by the red color in the signal.
- To remove the horns in the middle of an activation period click 'Add', then click on the peak you want to remove and press 'Enter'. You will see immediately what you removed.



- If you're done, click 'Save..' and your data will be saved as 'emg_corrected2.m'

Create EMG-based model

In 2009 students of the Universiteit Twente were asked to write a new block design used in SPM. This block design is not based on the protocol information, but on EMG measurements. So it respresents the real activity of the patiënt. See also "Bepaling van de doelstructuur voor plaatsing van diepe herseneIektroden bij essentiële tremor met gecombineerde EMG-fMRI techniek" written by:

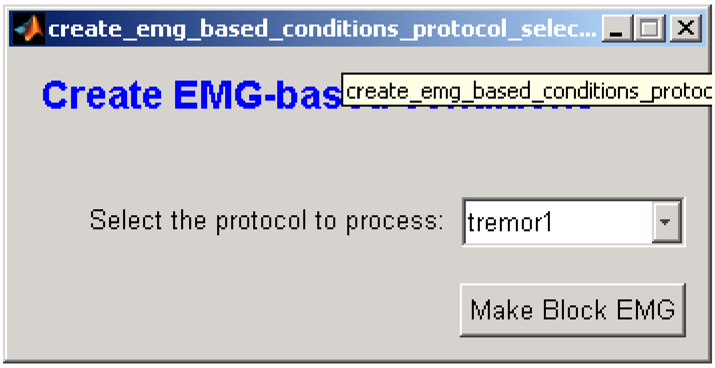
- Caroline Mattijssen (c.k.h.m.mattijssen@student.utwente.nl)
- Marjanne Bom (m.j.bom@student.utwente.nl)
- Merijn Eskes (m.eskes@student.utwente.nl)
- Bauke Kogelman (bauke@kogelman.net)

To make a block design based on the EMG, in EEGLab do the following:
- In EEGLab
- Click on the menu 'EMG_fMRI Analysis'
- Click on 'Create EMG-based conditions'

A window will pop up with two options:
- Make Block EMG
- Make regressor

Make Block EMG

- Choose the filtered EMG data
- Determine the threshold so that the activation of the muscles is included


create_emg_based_conditions_protocol_selec...
Create EMG-bas
create_emg_based_conditions_protoc
Make Block EMG

Select the protocol to process: tremor1

- Repeat the previous step 7 times (8 muscles). The thresholds will be saved in a separate file: thresholds.txt.
- Choose a right muscle (1-4) and a left one (5-8) in which you can recognize the tasks best. The model will be saved as ‘block_emg.mat’ in the map ‘regressor’.
- After you have done this, you have two options: close all windows by hand or close all windows including EEGLab. You can restart EEGLab by the following command in the command window: eeglab redraw

Make the regressors

If you like to make the block design based on the artefact free EMG data, you will have to do the following:
- In EEGLab
- Click on the submenu 'EMG_fMRI Analysis'
- Click on 'Create Regressors'

This step is optional. A popup window will appear in which you can enter values for a frequency band filter. If you choose 1-250 Hz the whole EMG is used for the regressor, if you choose e.g. 3-7 Hz only the EMG with frequencies around the tremor frequency will be used as regressor (for essential tremor). Standard is 3-7 Hz. In the popup window also 5 buttons are visible, which represents the protocol which you entered at the first step of the EMG-fMRI tool. If you press a button a new popup window will appear in which you can select the EMG data This EMG will be used for the regressor and the bandpass filter of the previous step will be applied to this EMG. After that the EMG is loaded in Matlab (duration: + 1 minute) and the bandpass filter is applied, a new window will popup. In this window you can select the muscles you like to use for your new regressor by clicking on the checkboxes. You can also inspect the EMG-data with the inspect button. You can only select two muscles, but for one model you mostly need one muscle. What you will see in this window is a picture of the values of the regressors in the freqreg matrix, that is the matrix after the bandpass filter. So if you had chosen 3-7 Hz, you see the EMG in the tremor area. You can view the different muscles by clicking on the next and previous buttons. After selecting the muscles, you press the 'Ready & Make regressors' button. Then you will be asked to select a .mat file to orthogonalize with, this can be the block_emg.mat (created with the previous method), but also the block.mat (the other one). You can find these files in the regressor directory. The new EMG regressors will be made and window with several plots will popup:
- bandpass filtered frequency regressors left and right
- an orthoganalized version of the bandpass filtered frequency regressors left and right
- EMG-based block design (4 conditions) and chipped version (4 conditions)
- Orthogonalized EMG regressors split by condition (4 conditions for left and right)

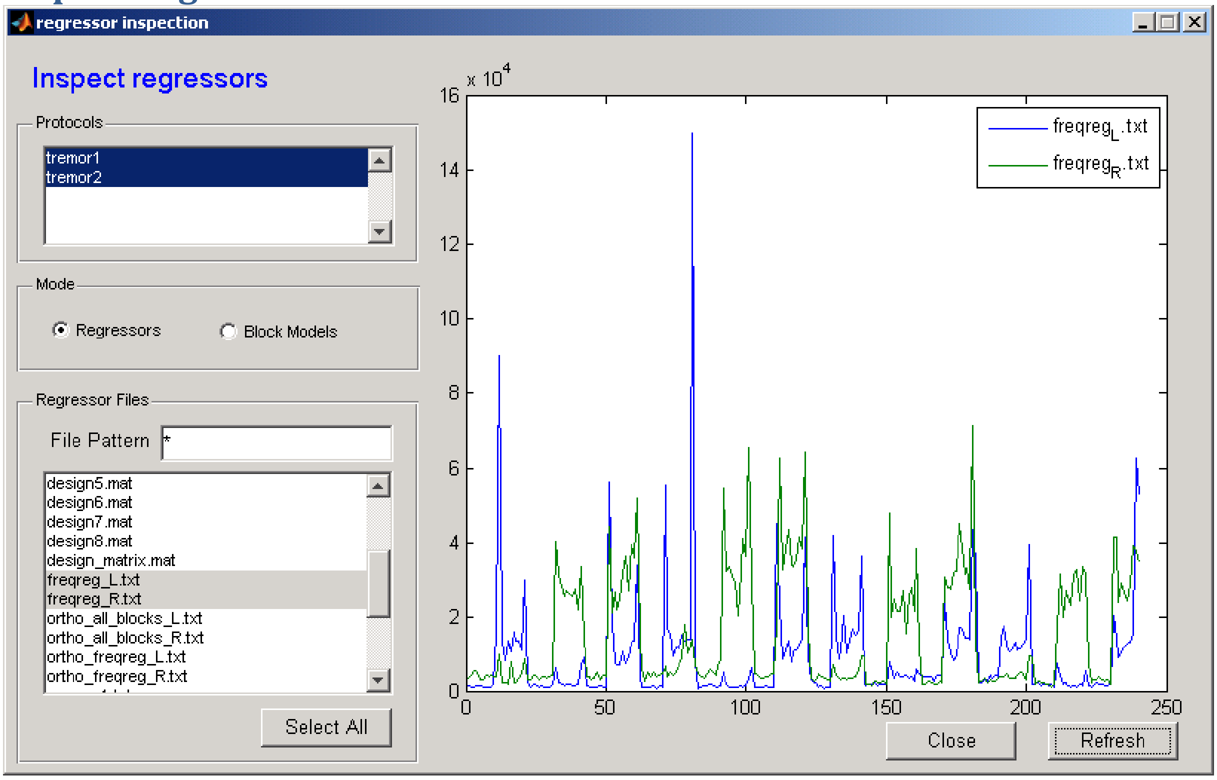
After the last window appeared you can inspect the data and close the windows. After that you can process the other protocols.
The following regressors will be made:
- frequency regressor: The EMG regressor with the bandpassfilter (e.g. 3-7 Hz) : freqreg_{L,R}.txt
- orthogonalized frequency regressor: ortho_freqreg_{L,R}.txt
- orthogonalized regressor; The above regressor orthogonalized with the normal model (block or block_emg), and divided by it's standard deviation. Separate files will be created for all 4 conditions (links strekken, rechts strekken, beide strekken, rust) for both left and right selected muscles: ortho_emg_{L,R}_{ls,rs,bs,r}.txt

Convolved version can be created in one of the final steps of the GUI. Orthogonalizing will be done with the selected Block model: block.mat or block_emg.mat. The 'frequency regressor' means that this is the EMG with the bandpass filter.

Convolve Regressors

Use this menu item to apply one of the predefined hemodynamic convolution functions to the previously created regressors.

Inspect Regressors



regressor inspection

Inspect regressors

Protocols

tremor1
tremor2

Mode

● Regressors ○ Block Models

Regressor Files

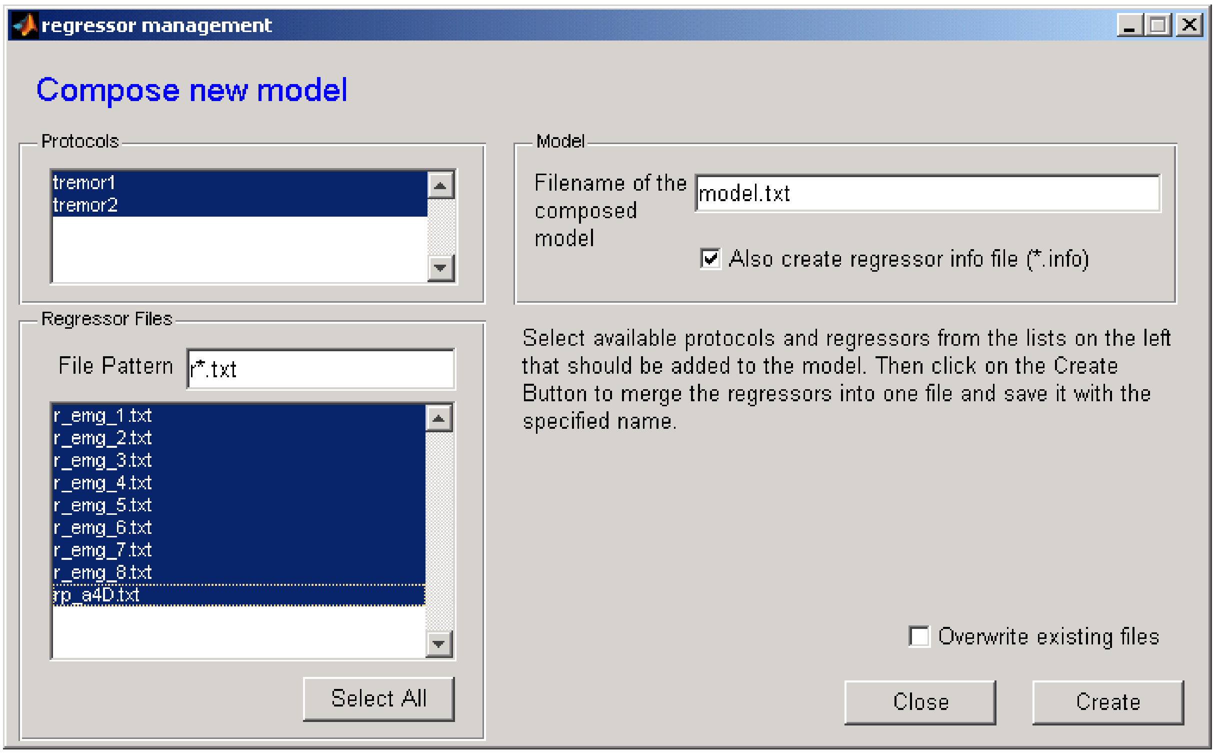
File Pattern *

design5.mat
design6.mat
design7.mat
design8.mat
design_matrix.mat
freqreg_L.txt
freqreg_R.txt
ortho_all_blocks_L.txt
ortho_all_blocks_R.txt
ortho_freqreg_L.txt
ortho_freqreg_R.txt

Make the model for SPM

After you made the new regressors for the protocols which you have processed in the previous step, you can make a model for SPM fMRI by combining several regressors. This step is only needed when you want to create a model with several regressors, e.g. a model that contains the ‘normal EMG-regressor’ but also included additional regressors such as movement parameters. To make the new model for SPM you will have to do the following:
- In EEGLab
- Click on the menu ‘EMG_fMRI Analysis’
- Click on ‘Compose Model’

After that, a window will popup which allow you to select the protocols and existing regressors. You can look for regressors which show tremor. Optional you can select the rp_a4D (realignment) parameters. When the parameters are ‘high’ (above 0.5) but not block like you can select them, but if they are low it is not needed. When they are ‘low’ and ‘block like’ you can select them, but if they are ‘high’ and block like you have a problem. There is some discussion about what to do then. (if you are not sure if they are block like, you can also try and make 2 models, one with and one without the movement parameters and compare the design orthogonalities. If they are mostly independent of task, you can choose for the model with the parameters)
After you‘ve selected the regressors, the new model will be made if you press the ‘Make model’ button. A dialog will ask you for a filename and it is also possible to change the folder, but the default folder is recommended to use.



Protocols: tremor1, tremor2

File Pattern: r*.txt

r_emg_1.txt
r_emg_2.txt
r_emg_3.txt
r_emg_4.txt
r_emg_5.txt
r_emg_6.txt
r_emg_7.txt
r_emg_8.txt
rp_a4D.txt

Filename of the composed model: model.txt

☑ Also create regressor info file (*.info)

Select available protocols and regressors from the lists on the left that should be added to the model. Then click on the Create Button to merge the regressors into one file and save it with the specified name.

☐ Overwrite existing files

Analysis

Model Making: model of the block design

The following steps will be done manually in SPM and will quit EEGLab.
- In EEGLab
- Click on the menu 'tools'
- Click on the submenu 'Start SPM fMRI'
- A confirmation popup will occur, click OK.

The first thing to do now is to create two directories in \pp\6666\protocol\ (protocol is e.g. tremor1).
Create a results folder:
- x:\onderzoek\fMRI\Tremor\pp\6666\tremor1\results

Create a subfolder with the name of your model in the results folder:
- x:\onderzoek\fMRI\Tremor\pp\6666\tremor1\results\name of your model

In SPM do the following:

- ➢ specify first level
- ➢ fMRI model specification
- ➢ directory
  specify files: …\pp\6666\tremor1\results\name of your model

- ➢ timing parameters
  - units for design = seconds
  - interscan interval = 2.8889
  - microtime resolution = 16
  - microtime onset = 1 (this depends on the slice timing)
- ➢ data and design
- ➢ new subject\session
  - scans
  - specify files (which files must be analysed)
    - \pp\6666\tremor1\fmri\swa4D (wrapped)
    - Or \pp\6666\tremor1\fmri\sra4D (not-wrapped)
  - If you type sra.* (ore swa) and in the place beneath 1:1000 you are sure you can select all sra-files. For selecting you can use Control A.
- ➢ Multiple conditions
  - Specify file = x:\Onderzoek\fMRI\Tremor\pp\6666\tremor1\regressor\block.mat
    - or block_emg.mat, block_emg_without_rest.mat or block_without_rest.mat (this are the regressors made with the UT method, or the other method)
- ➢ Multiple regressors
  - Specify file = x:\Onderzoek\fMRI\Tremor\pp\6666\tremor1\regressor\name of the model you've made. This is for example the convolved orthogonolized frequency regressor of the Extensor Left together with the movement parameters.
- ➢ Save as Job.mat (in the pp\6666\tremor1\results\name of your model)
- ➢ Run
  - This will get the SPM.mat file, to check the file
    - Review
    - Select spm.mat

        - Click design
        - Design matrix and/or Design orthogonality
- Estimate
    - This takes approximately 10 min.
    - Model estimation
        - Specify files = x:\Onderzoek\fMRI\Tremor\pp\6666\tremor1\name of your model\SPM.mat
    - Method = classical
    - Run

Contrasts

To look at the activation of the brain at different conditions, use contrasts in SPM:

- Results
    - Select SPM.mat in the directory:
        - X:\Onderzoek\fMRI\Tremor\pp\6666\tremor1\name of your model\SPM.mat
- Define the contrasts you are interested in
    - Mask with other contrasts = 'NO'
    - Name of your contrast
    - None
    - Threshold = 0.001
    - Extend threshold = 0
        - Overlays
            - Sections = select the t1 (you can find this in pp/patient's number
        - Goto global maxima (ore another place you're interested in)
    - You can save figures as e.g. jpg-pictures.

Bijlage 5

Resultaten van de uitgevoerde analyses.

Samenvatting van welke activaties er in de thalamus en het cerebellum zichtbaar zijn met bijbehorende p-waarde, per patient, per protocol, per contrast.

Patient 6201

Tremor1

Contrast	1-1			-11		
	t	c	p	t	c	p
model 1	⬛	⬛	0.1	⬛	🟦	0.001
model 2	🟦	🟦 2	0.01	🟦 2	🟥 2	0.00001
model 3	🟥 2	⬛	0.001	⬛	⬛	0.0001
model 4	⬛	🟦 2	0.001	🟥	🟦	0.001
verwacht	🟥	🟦		🟦	🟥	

Legenda:

Activatie:	
Links	🟦
Rechts	🟥
Bdz	🟨
Geen resultaat (te veel of te weinig activatie)	⬛
Niet uitgevoerd	⬜

Volgens verwachting	2 ptn
Bijderzijds	0,5 ptn

Afkorting	Betekenis
t	Thalamus
c	Cerebellum
p	p-waarde

Patient 6201

Tremor2

Contrast	1-1			-11		
	t	c	p	t	c	p
model 1	⬛	🟨 0,5	0.001	🟥	🟨 0,5	0.001
model 2	🟦	🟥	0.0001	🟥	🟦	0.0001
model 3	🟥 2	⬛	0.001	🟥	⬛	0.0001
model 4	⬛	🟨 0,5	0.001	🟥	🟥 2	0.001
verwacht	🟥	🟦		🟦	🟥	

Patient 6202

Tremor1

Contrast	1-1			-11		
	t	c	p	t	c	p
model 1	⬛	🟦 2	0.01	🟥	⬛	0.001
model 2	⬛	🟦 2	0.0001	🟥	⬛	0.0001
model 3	🟥 2	⬛	0.00001	🟦 2	⬛	0.000001
model 4	🟥 2	🟦 2	0.001	🟨 0,5	🟦	0.001
verwacht	🟥	🟦		🟦	🟥	

Patient 6202

Tremor2

Contrast	1-1			-11		
	t	c	p	t	c	p
model 1	⬛	🟨 0,5	0.01	🟥	🟦	0.01
model 2	⬛	🟦 2	0.001	🟥	⬛	0.001
model 3	🟥 2	⬛	0.001	🟦 2	⬛	0.00001
model 4	⬛	🟦 2	0.001	🟨 0,5	🟥 2	0.001
verwacht	🟥	🟦		🟦	🟥	

Patient 6204

Tremor1

Contrast	1-1			-11		
	t	c	p	t	c	p
model 1	▩	🟦 2	0.001	🟦 2	🟥 2	0.0001
model 2	🟥 2	🟦 2	0.01	🟦 2	▩	0.01
model 3	🟥 2	▩	0.1	🟥	▩	0.001
model 4	▩	🟦 2	0.001	🟧 0,5	🟥 2	0.001
verwacht	🟥	🟦		🟦	🟥	

Patient 6204

Tremor2

Contrast	1-1			-11		
	t	c	p	t	c	p
model 1	▩	🟥	0.001	🟧 0,5	🟥 2	0.0001
model 2	▩	🟦 2	0.1	🟦 2	▩	0.0001
model 3						
model 4	▩	🟦 2	0.001	🟧 0,5	🟥 2	0.0001
verwacht	🟥	🟦		🟦	🟥	

Patient 6205

Tremor1

Contrast	1-1			-11		
	t	c	p	t	c	p
model 1	▩	🟧 0,5	0.001	🟦 2	🟥 2	0.001
model 2	🟧 0,5	🟥	0.001	🟥	🟥 2	0.01
model 3	🟥 2	▩	0.01	▩	▩	0.001
model 4	🟥 2	🟦 2	0.01	🟥	🟧 0,5	0.001
verwacht	🟥	🟦		🟦	🟥	

Patient 6205

Tremor2

Contrast	1-1			-11		
	t	c	p	t	c	p
model 1	▩	🟦 2	0.001	🟥	🟦	0.001
model 2	▩	🟦 2	0.001	🟥	🟥 2	0.001
model 3						
model 4						
verwacht	🟥	🟦		🟦	🟥	

Patient 6401

Tremor1

Contrast	1-1			-11		
	t	c	p	t	c	p
model 1	🟦	0,5	0.001	0,5	2	0.001
model 2	⬛	2	0.01	⬛	⬛	0.001
model 3	2	⬛	0.01	🟥	⬛	0.001
model 4	⬛	2	0.001	⬛	⬛	0.001
verwacht	🟥	🟦		🟦	🟥	

Patient 6401

Tremor2

Contrast	1-1			-11		
	t	c	p	t	c	p
model 1	⬛	2	0.001	⬛	2	0.001
model 2	🟦	🟥	0.01	⬛	🟦	0.01
model 3						
model 4						
verwacht	🟥	🟦		🟦	🟥	

Resultaten	Totaal ptn	Aant. Metingen	Gemiddeld
Model 1	25,5	10	2,6
Model 2	28,5	10	2,9
Model 3	18	7	2,6
Model 4	29	8	3,6