EEG generator

```
close all;clear all;clc
%1
Fs=512;n_channel=64;t_end=20;n=0;na=1;nb=0;att=20;toler=-1;lim=-1;range=-1;
report=1;NOISE_SNR=-1;af='sin';normaliz=-1;
```

```
close all;clear all;clc
%2
Fs=512;n_channel=64;t_end=20;n=0;na=4;nb=0;att=20;toler=-1;lim=-1;range=-1;
report=1;NOISE_SNR=-1;af='sin';normaliz=-1;
```

```
close all;clear all;clc
%3
Fs=512;n_channel=64;t_end=20;n=0;na=1;nb=0;att=20;toler=-1;lim=-1;range=-1;
report=1;NOISE_SNR=-1;af='ICA';normaliz=-1;
```

```
close all;clear all;clc
%4
Fs=512;n_channel=64;t_end=20;n=0;na=4;nb=0;att=20;toler=-1;lim=-1;range=-1;
report=1;NOISE_SNR=-1;af='ICA';normaliz=-1;
```

```
close all;clear all;clc
%5
Fs=512;n_channel=64;t_end=20;n=0;na=4;nb='default';att=20;toler=-1;lim=-1;range=-1;
report=1;NOISE_SNR=-1;af='sin';normaliz=-1;
```

```
close all;clear all;clc
%6
Fs=512;n_channel=64;t_end=20;n=0;na=1;nb='default';att=20;toler=-1;lim=-1;range=-1;
report=1;NOISE_SNR=-1;af='sin';normaliz=-1;
```

```
close all;clear all;clc
%7
Fs=512;n_channel=64;t_end=20;n=0;na=4;nb='default';att=20;toler=-1;lim=-1;range=-1;
report=1;NOISE_SNR=-1;af='ICA';normaliz=-1;
```

```
close all;clear all;clc
%8
Fs=512;n_channel=64;t_end=20;n=0;na=1;nb='default';att=20;toler=-1;lim=-1;range=-1;
report=1;NOISE_SNR=-1;af='ICA';normaliz=-1;
```

orintation

```
load Dip_ori_perpendicular
D=Dip_ori_perpendicular;
if n<=size(Dip_ori_perpendicular,1) && n>0
    for i1=1:n
        fprintf('For source %d .\n',i1);
        prompt1 = "type index you want to specify it's orientation? ";
        a(i1,1) = input(prompt1);
        prompt2="type orientation you want to specify it's orientation in this' ..." + ...
            " form [x y z] in radian? ";
        D(a(i1,1),:)=input(prompt2); %D is new orientation matrix
    end
end
```

channel number

```
switch n_channel
    case 16
        load('grid16.mat')
    case 32
        load('grid32.mat')
    case 64
        load('grid64.mat')
    case 128
        load('grid128.mat')
    otherwise
        disp('the channel number is not acceptable')
end
```

eeg simulation for main source

```
amp=1;% you can change it but amp=1 is closer to real data
eeg=zeros(n_channel,t_end*Fs);
for i2=1:na
    fprintf('For main active source %d .\n',i2);
    prompt3 = "type index you want to specify as an active source? ";
    mainsourceIDX(i2,1) = input(prompt3);
    if af=='sin'
        amp=1;
        prompt4 = "type the frequency of the source signal in Hz: ";
        f1(i2) = input(prompt4);
        tt=[1/Fs:1/Fs:t_end];
        afms(i2,:)=amp*sin(2.*pi.*f1(i2).*tt);
        if normaliz==1
            afms(i2,:)=normalize(afms(i2,:));
        end
        eeg=eeg+grid.leadfield{mainsourceIDX(i2,1)}*D(mainsourceIDX(i2,1),:)'*afms(i2,:);
    end
if af=='ICA'
    load('fasticaData.mat')
    stp(i2,1)=randi(size(fasticaData{1,1},2)-t_end*Fs);%start point
    if stp(i2,1)>=(t_end*Fs)
```

```
        afms(i2,:)=fasticaData{1,randi(size(fasticaData,2))}(randi(size(fasticaData{1,1},1)), ...
            [abs(t_end*Fs-stp(i2,1)-1):stp(i2,1)]);
    else
       afms(i2,:)=fasticaData{1,randi(size(fasticaData,2))}(randi(size(fasticaData{1,1},1)), ...
[abs(stp(i2,1):t_end*Fs+stp(i2,1)-1)]);
    end
        if normaliz==1
            afms(i2,:)=normalize(afms(i2,:));
        end
        eeg=eeg+grid.leadfield{mainsourceIDX(i2,1)}*D(mainsourceIDX(i2,1),:)'*afms(i2,:);
    end
% end
rmain(i2,1)=rank(cov(eeg')) % rank that have been produced by main sources
end
```

eeg simulation for background source

```
if toler==-1
    toler=1.4023e-06;
end
if range==-1
    range=10;
end
if lim==-1
    lim=10;
end
```

```
%manual mode
if nb~='default'
    load('fasticaData.mat')
    if nb~=0
        for i3=1:nb
            fprintf('For background source %d .\n',i3);
            prompt4 = "type index of source you want to specify as an background source? ";
            bs(i3,1) = input(prompt4);
            stp2(i3,1)=randi(size(fasticaData{1,1},2)-t_end*Fs);%start point
            if stp2(i3,1)>(t_end*Fs)
                afbs(i3,:)=(1/att)*fasticaData{1,randi(size(fasticaData,2))} ...
                    (randi(size(fasticaData{1,1},1)), ...
                    [(abs(t_end*Fs-stp2(i3,1)-1)):stp2(i3,1)]);
                eeg=eeg+grid.leadfield{bs(i3,1)}*D(bs(i3,1),:)'*afbs(i3,:);
                bsoriIDX=D(bs(i3,1),:);
            else
                afbs(i3,:)=(1/att)*fasticaData{1,randi(size(fasticaData,2))} ...
                    (randi(size(fasticaData{1,1},1)),...
                    [abs(stp2(i3,1):t_end*Fs+stp2(i3,1)-1)]);
                eeg=eeg+grid.leadfield{bs(i3,1)}*D(bs(i3,1),:)'*afbs(i3,:);
                bsoriIDX=D(bs(i3,1),:);
            end
            rbg(i3,1)=rank(cov(eeg')) %rank that have been produced by main & background source
            rtbg(i3,1)=rank(cov(eeg'),toler) %rank that have been produced by main & background
        end
        bsIDX=bs;
```

```
    end
end
difrenceS=abs(grid.pos(:,2)-grid.pos(:,3));
for i5=1:size(difrenceS,1)
        if difrenceS(i5,1)<lim
            difrenceS(i5,1)=0;
        end
  end
```

```
%default mode
if nb=='default'
    i6=1;i8=1;
    rbg(i6,1)=rmain(end,1);
    rtbg(i6,1)=rmain(end,1);
    i12=1;
    for i5=1:size(difrenceS,1)
        if difrenceS(i5,1)~=0
            bsdidx(i6,1)=i5;
            stp3(i6,1)=randi(size(fasticaData{1,1},2)-t_end*Fs);%start point
            for i7=1:na
                if bsdidx(i6,1)==mainsourceIDX(i7,1)
                    bsdidx(i6,1)=0;
                end
            end
            if rtbg(i8,1)<n_channel && bsdidx(i6,1)~=0
                if stp3(i6,1)>=(t_end*Fs)
                    afbs(i12,:)=(1/att)*fasticaData{1,randi(size(fasticaData,2))} ...
                        (randi(size(fasticaData{1,1},1)),...
                        [abs(t_end*Fs-stp3(i6,1)-1):stp3(i6,1)]);
                    eeg=eeg+grid.leadfield{bsdidx(i6,1)}*D(bsdidx(i6,1),:)'*afbs(i12,:);
                    i12=i12+1;
                    i8=i8+1;
            else
                afbs(i12,:)=(1/att)*fasticaData{1,randi(size(fasticaData,2))} ...
                    (randi(size(fasticaData{1,1},1)), ...
                    [abs(stp3(i6,1):t_end*Fs+stp3(i6,1)-1)]);
                eeg=eeg+grid.leadfield{bsdidx(i6,1)}*D(bsdidx(i6,1),:)'*afbs(i12,:);
                i12=i12+1;
                i8=i8+1;
                end
            rbg(i8,1)=rank(cov(eeg'))  %rank that have been produced by main
            % & background sources
            rtbg(i8,1)=rank(cov(eeg'),toler) %rank that have been produced by main &
            % background sources with toler
            if report==1
                bsIDX(i12,:)=bsdidx(i6,1);
                bsoriIDX=D(bsdidx(i6,1),:); %correspond to bsIDX
            end
            end

        end
        i6=i6+range;
    end
```

```
end

% end
```

report

```
if report==1
    for i11=1:size(mainsourceIDX,1)
        orimainIDX(i11,:)=D(mainsourceIDX(i11,:),:);%correspond to IDX in mainsourceIDX
    end
end
```

white noise

```
if NOISE_SNR~=-1
    eeg=awgn(eeg,NOISE_SNR);
    rankwithoise=rank(eeg,toler)
end
```

if you don't want background

```
if nb==0
    rtbg=0;rmain=0;afbs=0;bsIDX=0;bsoriIDX=0;bsdidx=0;
end
```