

test_CTFgradComp_tSSS_OTP_inline

February 27, 2019

1 Test tSSS movement compensation with gradient compensations on CTF data

1.1 Steps

- Interpolate always bads
- [optional]: Remove channel-noise (incl. squid jumps) by `oversampled_temporal_projection`
[example](#), [doc](#), [commit](#), [paper](#)
> Note: “OTP can improve sensor noise levels (especially under visual inspection) and repair some bad channels. This noise reduction is known to interact with tSSS such that increasing the `st_correlation` value will likely be necessary.”
- Extract head position by `mne.chpi._calculate_head_pos_ctf`
(for Elektra files: [generating head pos](#), [reading head pos](#), using the [command line option](#))
- Movement compensation and re-alignment by tSSS (`maxwell_filter`)
[example](#), [manual](#), [doc raw](#), [doc epochs](#), [paper](#), [old paper](#)
 - Compare different combinations of gradient compensation and tSSS

1.2 Discussions

1.2.1 [github-discussion from 2015](#):

As for the reference sensor compensation, such as the 3rd order gradiometrization in CTF systems, the compatibility of the compensated data with `MaxFilter` depends on how the compensation is done. As far as I understand, the CTF compensation is done by suppressing the homogeneous fields and the 1st and 2nd order derivatives of the external fields based on the reference sensor recordings. This kind of data can be processed with `MaxFilter` without any extra manipulations and the resulting data should have any required forward-model reconstructions automatically applied, just like in the case of Elekta’s active compensation.

staulu

Your comment made me realize that it is actually not feasible to perform the reference sensor based compensation if SSS has already been applied to the whole CTF array (actual sensors + reference sensors) because all the reconstructed signals will

be of internal origin and the assumptions that justify using external Taylor's expansion for the extrapolation of reference signals don't hold true anymore. The solution would be 1) just to apply SSS on the whole array (it will be numerically more stable thanks to the reference sensors) and consider this the final interference suppression step without additional reference sensor compensation, or 2) first perform the 3rd order gradiometrization and then possibly do SSS depending on the details of the compensation. I think this topic requires some extensive thinking and planning, maybe in a future project :)
staulu

1.2.2 [github-discussion from 2019](#)

Thanks again for your answers. So, to summarize, as of the current state SSS and gradient comp cannot really be combined, right? 1. (t)SSS → gradient compensation: Problematic due to Taylor assumptions probably not met anymore. 2. gradient compensation → (t)SSS: Needs adaptations in the core of the maxwell code (applying the gradient comp also to the SSS model). Option 2 is theoretically sound and just needs to be implemented, for option 1 further theoretical insight is needed.
jeythekey

I basically agree. I have a couple of comments: 1. In the case of (t)SSS → gradient compensation, one assumes that the data contain signals of external origin that can be reconstructed from the reference sensor measurement. However, if the actual MEG channels have been processed by (t)SSS, then extrapolating and subtracting external interference based on reference sensors (gradient compensation) will most probably increase interference in the (t)SSS-processed data. 2. In principle, if the gradient compensation is applied to purely subtract homogeneous fields or low-order gradients of the external field from the measurement, then one shouldn't necessarily need to modify the SSS model because such a subtraction, if done properly, shouldn't fundamentally alter the spatial pattern of the brain signal. But whether the SSS model needs to be modified or not depends on the details of the gradient compensation. Anyway, this should be a doable option.
staulu

1.2.3 in fieldtrip:

- http://www.fieldtriptoolbox.org/example/how_to_incorporate_head_movements_in_meg_analysis/
- http://www.fieldtriptoolbox.org/reference/ft_megrealign/
- http://www.fieldtriptoolbox.org/reference/ft_regressconfound/

1.3 Info about gradient compensation from CTF manual

[...] Thus, a well designed, unshielded 3rd-order gradiometer offers a level of noise discrimination equal to, or better than, that of a shielded lower-order system. [...]

To resolve this issue, the CTF MEG System implements high-order gradiometer formation through software, as shown in Figure 7 (c). This system uses a low-order, hardware gradient detector (e.g., 1st-order) that is then processed relative to a common

reference system to form a higher-order gradient response. This technique is not adaptive. The coefficients involved are time independent, and only one reference system is needed for an arbitrary number of detecting channels. [...]

In addition to these gradiometer sensor coils (flux transformers), the system includes a number of reference coils for environmental noise cancellation. This environmental noise cancellation array supplies all of the necessary information on the magnetic field vector and the gradient tensors in the vicinity of the sensor coils to eliminate the effects of uniform field and gradient common-mode noise. **This is implemented through a proprietary digital signal processing algorithm, in effect constructing an array of well-balanced higher-order virtual gradiometers in software.** The noise rejection is discussed in “Environmental Noise Rejection” on page 50. [...]

The front-end sensors on the MEG system are already first- order axial gradiometers. The higher-gradient formation noise cancellation utilizes a set of reference channels to synthetically generate second- or third-order gradient sensors. The method works by adding a carefully calculated linear combination of reference channels to each sensor channel. The process is linear so that it may be applied or removed with no ill affects. It is possible to gain all the benefits of higher-order gradient formation during post-processing. [...]

CTF-manual technical description

1.4 Test

1.4.1 Imports

```
In [1]: # set figure size:
        # %matplotlib widget
        %matplotlib inline
        from matplotlib import pyplot as plt
        plt.rcParams['figure.figsize'] = [8, 5]

        # general:
        import numpy as np

        # mne
        import mne
        from mne.preprocessing import maxwell_filter
        from mne.chpi import _calculate_head_pos_ctf, head_pos_to_trans_rot_t
        from mne.fixes import einsum

        # my own stuff:
        from utils import loadRawEvents, loadRun
```

1.4.2 Load data, add annotations etc. (my own stuff ...)

```
In [2]: raw, ev, info, cndrows, endrows = loadRawEvents('SZ14G', 1, preload=False)[:2]
        raw = loadRun(raw, ev, cndrows, endrows, meg=True, ref_meg=True, eog=False, ecg=False, e
```

```

    # ias = headloc (hlc-chans)
    # HP-filter at .1 Hz applied (Maxwell with 10 sec windows have that effect as well.)
    raw.crop(tmin=20, tmax=30) # 10 secs of data

..._01.ds
res4 data read.
hc data read.
Separate EEG position data file read.
Quaternion matching (desired vs. transformed):
    0.68   77.75   0.00 mm <->   0.68   77.75   0.00 mm (orig :  -60.10   49.66 -268.16 mm
   -0.68  -77.75   0.00 mm <->  -0.68  -77.75   0.00 mm (orig :   49.56  -60.50 -263.99 mm
   112.66   0.00   0.00 mm <->  112.66   0.00   0.00 mm (orig :   73.23   74.56 -254.58 mm
Coordinate transformations established.
Polhemus data for 3 HPI coils added
Device coordinate locations for 3 HPI coils added
4 EEG electrode locations assigned to channel info.
4 EEG locations added to Polhemus data.
Measurement info composed.
...
Current compensation grade : 3

HP filtering at .1 Hz ...
Filtering raw data in 1 contiguous segment
Setting up high-pass filter at 0.1 Hz
l_trans_bandwidth chosen to be 0.1 Hz
Filter length of 39601 samples (33.001 sec) selected

```

1.4.3 Calc and plot head-pos

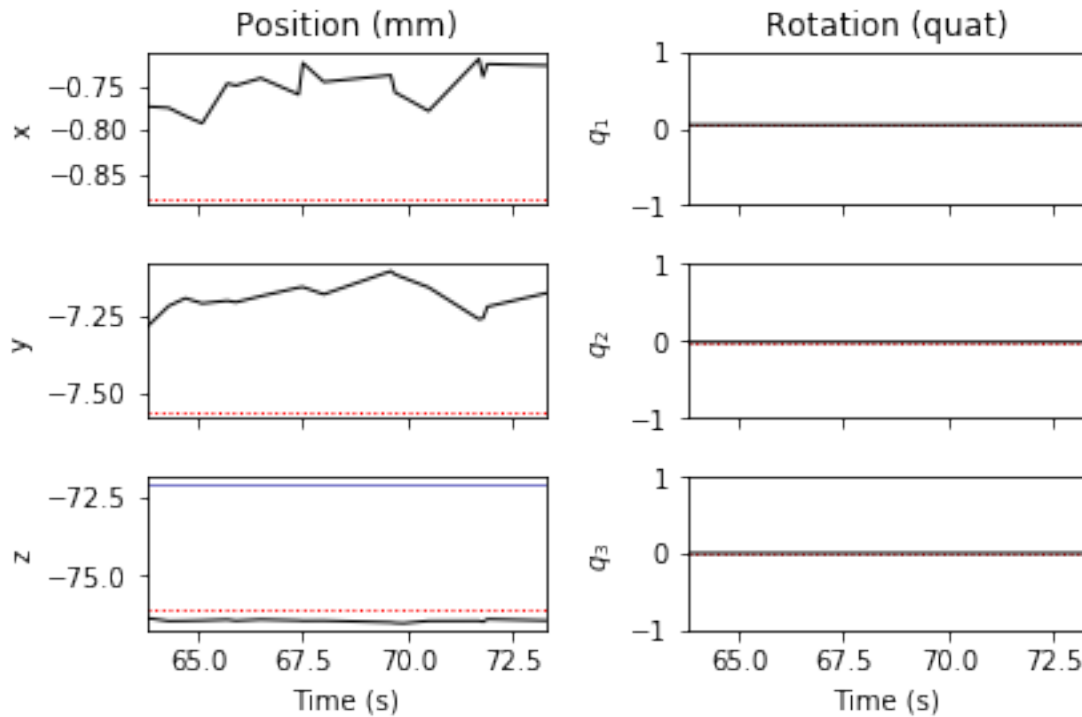
```

In [3]: head_pos = _calculate_head_pos_ctf(raw)

mne.viz.plot_head_positions(
    head_pos, mode='traces', destination=None, info=raw.info); # raw.info['dev_head_t']
# plots in device coordinates [head_dev_t] (otherwise, would start at zero)
# --> destination, given in [dev_head_t], is plotted inverted.

Coordinate transformations established.

```



```
In [4]: ## Use mean head-trans as destination later
        destination = np.mean(head_pos[..., 4:7], axis=0) # this is the trans, without rotation
```

1.4.4 A) Plot raw with different gradient compensations (no tSSS)

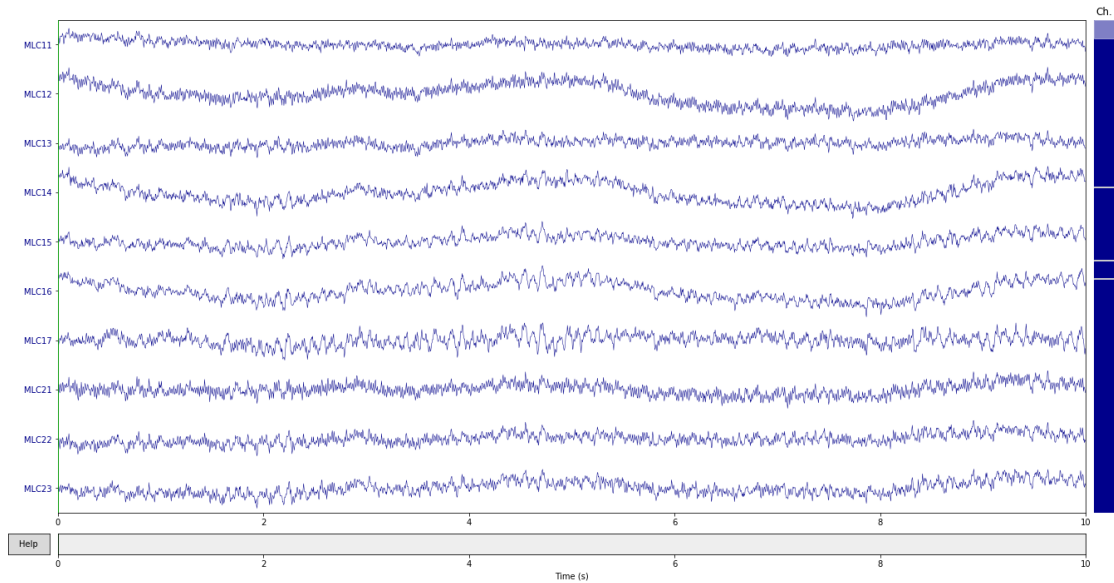
```
In [5]: ## prepare raw plot
        def myrawplot(raw, tit):
            picks = mne.pick_types(raw.info, meg=True, ref_meg=False, eeg=False, eog=False,
                                   ecg=False, stim=False, exclude=[]) #, selection=selection, ref_meg=ref_meg

            fig = raw.plot(order=picks, scalings=None, lowpass=70, highpass=0.1, #, scalings='M',
                           butterfly=False, duration=10, n_channels=10, # group_by='position', n_channels=n_channels,
                           remove_dc=True, title=tit);

            return fig

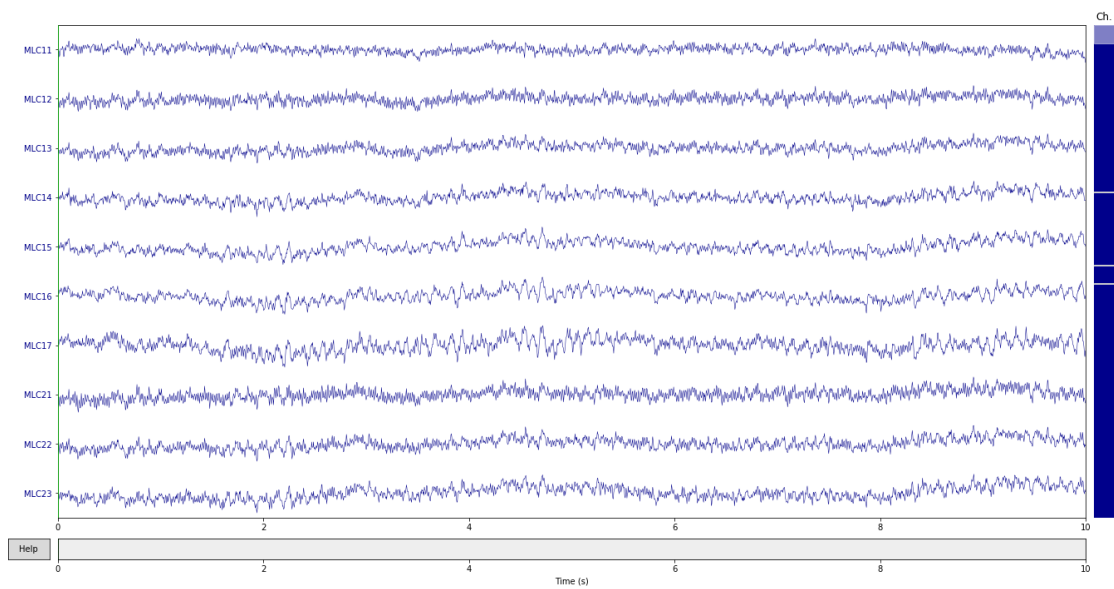
In [6]: tit = "no grad comp"
        raw_tmp = raw.copy().apply_gradient_compensation(0)
        fig = myrawplot(raw_tmp, tit)
```

Compensator constructed to change 3 -> 0
Applying compensator to loaded data



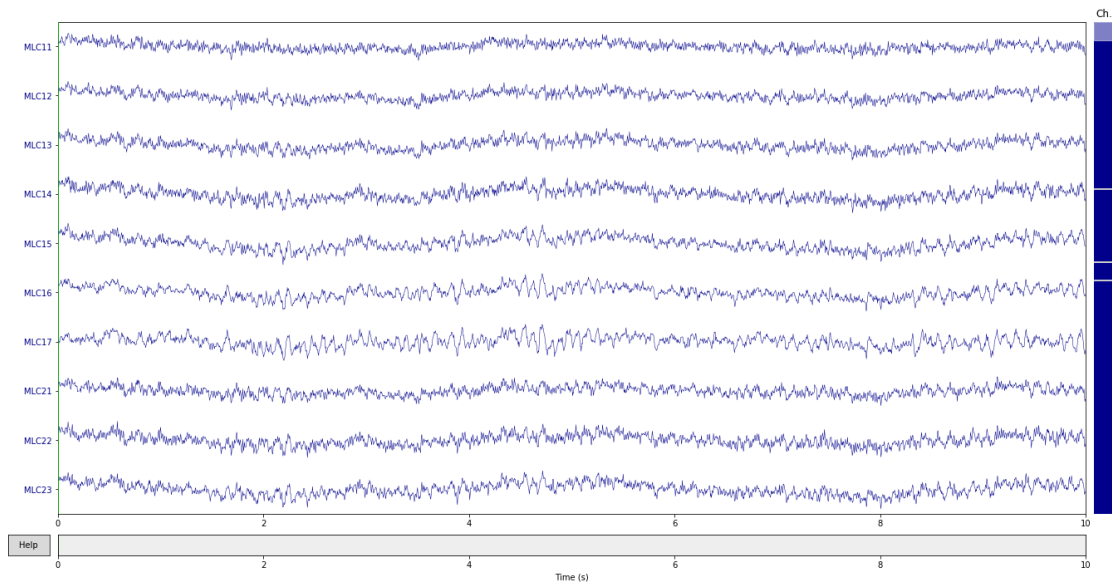
```
In [7]: tit = "1st order grad comp"
        raw_tmp = raw.copy().apply_gradient_compensation(1)
        fig = myrawplot(raw_tmp, tit)
```

Compensator constructed to change 3 -> 1
Applying compensator to loaded data

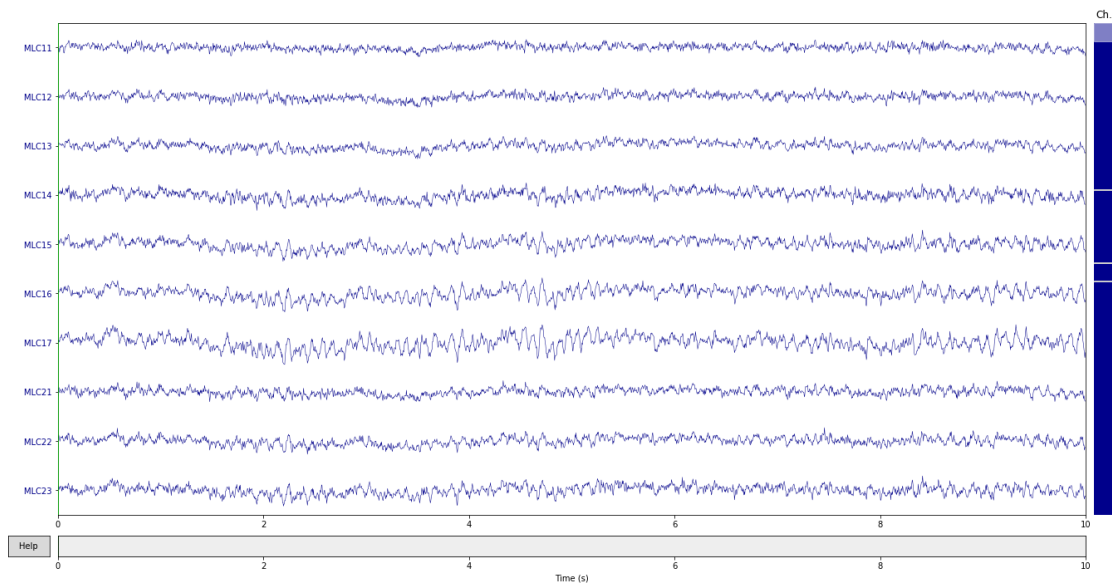


```
In [8]: tit = "2nd order grad comp"
raw_tmp = raw.copy().apply_gradient_compensation(2)
# not recommended to change grad-comp several times --> copy
fig = myrawplot(raw_tmp, tit)
```

Compensator constructed to change 3 -> 2
Applying compensator to loaded data



```
In [9]: tit = "3rd order grad comp"
fig = myrawplot(raw, tit) # is already applied
```



Summary As expected, the default 3rd order gradient compensation looks cleanest. Most of the slow drifts got removed.

1.4.5 B) Plot raw with grad-comp (different orders) -> tSSS

```
In [10]: def mymaxwell(raw, destination, head_pos):
        raw_sss = maxwell_filter(raw, st_duration=10,
                                destination= destination, # in head coordinates
                                st_correlation=.98,
                                ignore_ref=True,
                                coord_frame='head',
                                head_pos=head_pos, # head_pos or None
                                st_fixed=True,
                                st_only=False,
                                skip_by_annotation=('bad', 'edge', 'bad_acq_skip'))

        return raw_sss
```

```
In [11]: # Maxwell only
        tit = "no grad comp --> tSSS +mvmntComp +destMean -ref"

        raw_sss = raw.copy().apply_gradient_compensation(0)
        raw_sss = mymaxwell(raw_sss, destination, head_pos)
        fig = myrawplot(raw_sss, tit)
```

Compensator constructed to change 3 -> 0

Applying compensator to loaded data

Maxwell filtering raw data

Appending head position result channels and copying original raw data

Removing 5 compensators from info because not all compensation channels were picked.

Bad MEG channels being reconstructed: ['MLP53', 'MRC12', 'MRC25']

Processing 0 gradiometers and 269 magnetometers (of which 269 are actually KIT gradiometers)

Automatic origin fit: head of radius 86.7 mm

Using origin 1.0, -23.9, 38.5 mm in the head frame

Processing data using tSSS with st_duration=10.0

Using 82/95 harmonic components for 0.000 (70/80 in, 12/15 out)

Processing 1 data chunk

```
<ipython-input-10-eed8a0bf3da7>:10: RuntimeWarning: Only 4 head digitization points of the speci
skip_by_annotation=('bad', 'edge', 'bad_acq_skip'))
```

```
<ipython-input-10-eed8a0bf3da7>:10: RuntimeWarning: (X, Y) fit (1.0, -23.9) more than 20 mm from
skip_by_annotation=('bad', 'edge', 'bad_acq_skip'))
```

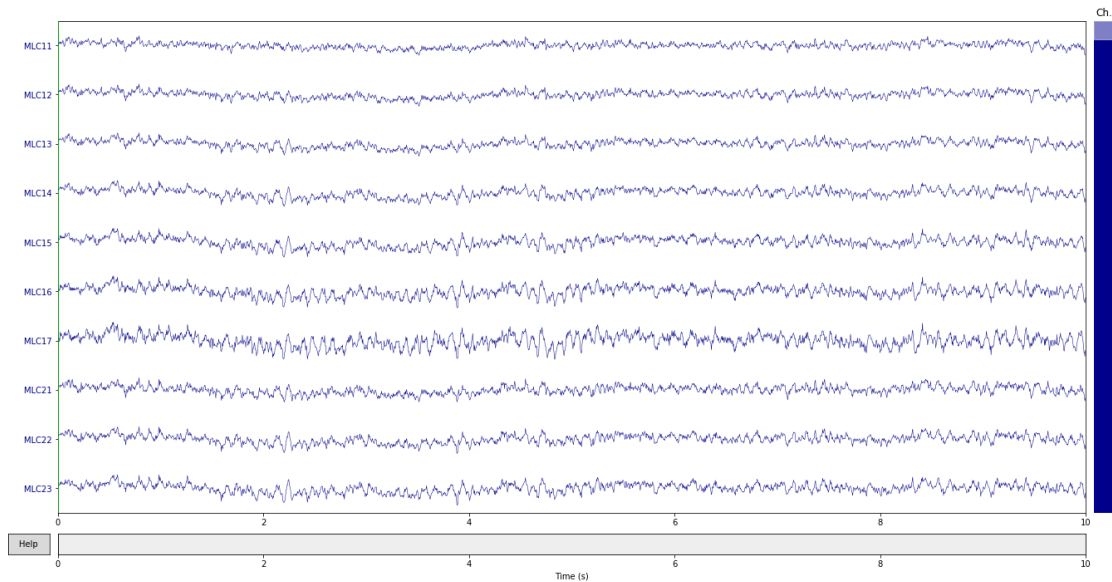
Using 83/95 harmonic components for 0.000 (71/80 in, 12/15 out)

Projecting 1 intersecting tSSS component for 0.000 - 10.000 sec (#1/1) (across 18


```

Using 82/95 harmonic components for    0.000  (70/80 in, 12/15 out)
...
Using 82/95 harmonic components for    9.516  (70/80 in, 12/15 out)
[done]

```



```
In [12]: tit = "1st order grad comp --> tSSS +mvmntComp +destMean -ref"
```

```

raw_sss = raw.copy().apply_gradient_compensation(1)
raw_sss = mymaxwell(raw_sss, destination, head_pos)
fig = myrawplot(raw_sss, tit)

```

Compensator constructed to change 3 -> 1

Applying compensator to loaded data

Maxwell filtering raw data

Appending head position result channels and copying original raw data

Removing 5 compensators from info because not all compensation channels were picked.

Bad MEG channels being reconstructed: ['MLP53', 'MRC12', 'MRC25']

Processing 0 gradiometers and 269 magnetometers

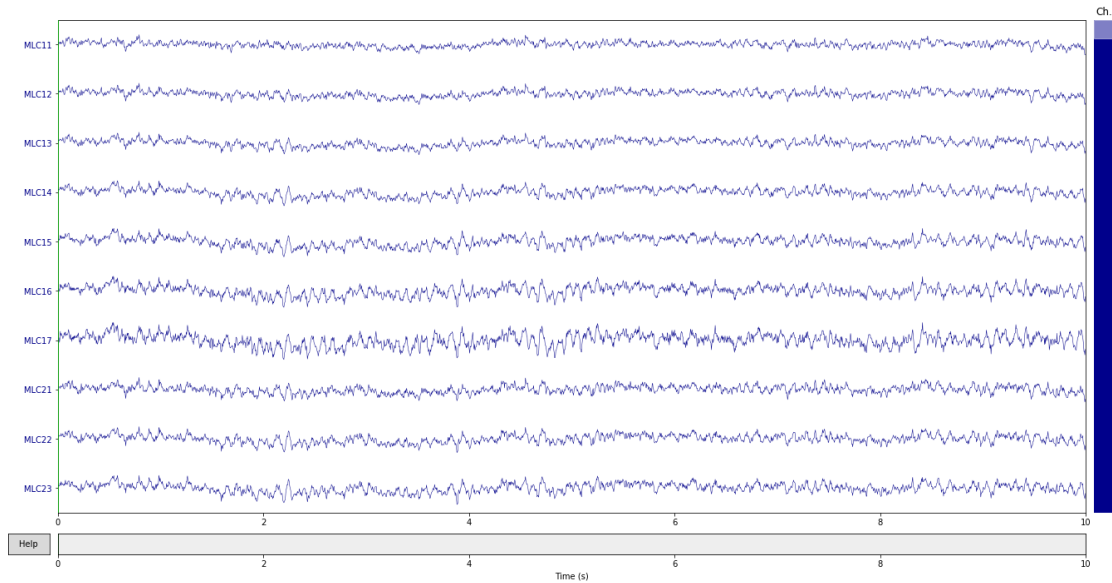
Automatic origin fit: head of radius 86.7 mm

Using origin 1.0, -23.9, 38.5 mm in the head frame

Processing data using tSSS with st_duration=10.0

Using 85/95 harmonic components for 0.000 (70/80 in, 15/15 out)

Processing 1 data chunk



```
In [13]: tit = "2nd order grad comp --> tSSS +mvmntComp +destMean -ref"
```

```
raw_sss = raw.copy().apply_gradient_compensation(2)
raw_sss = mymaxwell(raw_sss, destination, head_pos)
fig = myrawplot(raw_sss, tit)
```

Compensator constructed to change 3 -> 2

Applying compensator to loaded data

Maxwell filtering raw data

Appending head position result channels and copying original raw data

Removing 5 compensators from info because not all compensation channels were picked.

Bad MEG channels being reconstructed: ['MLP53', 'MRC12', 'MRC25']

Processing 0 gradiometers and 269 magnetometers

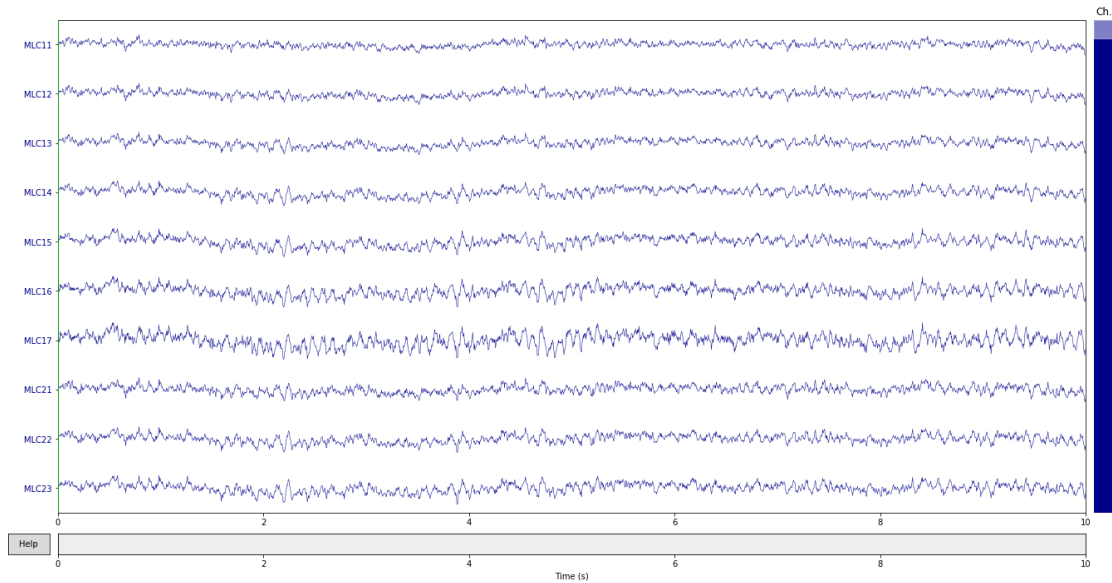
Automatic origin fit: head of radius 86.7 mm

Using origin 1.0, -23.9, 38.5 mm in the head frame

Processing data using tSSS with st_duration=10.0

Using 85/95 harmonic components for 0.000 (70/80 in, 15/15 out)

Processing 1 data chunk



```
In [14]: tit = "3rd order grad comp --> tSSS +mvmntComp +destMean -ref"
```

```
raw_sss = mymaxwell(raw, destination, head_pos)
fig = myrawplot(raw_sss, tit)
```

Maxwell filtering raw data

Appending head position result channels and copying original raw data

Removing 5 compensators from info because not all compensation channels were picked.

Bad MEG channels being reconstructed: ['MLP53', 'MRC12', 'MRC25']

Processing 0 gradiometers and 269 magnetometers

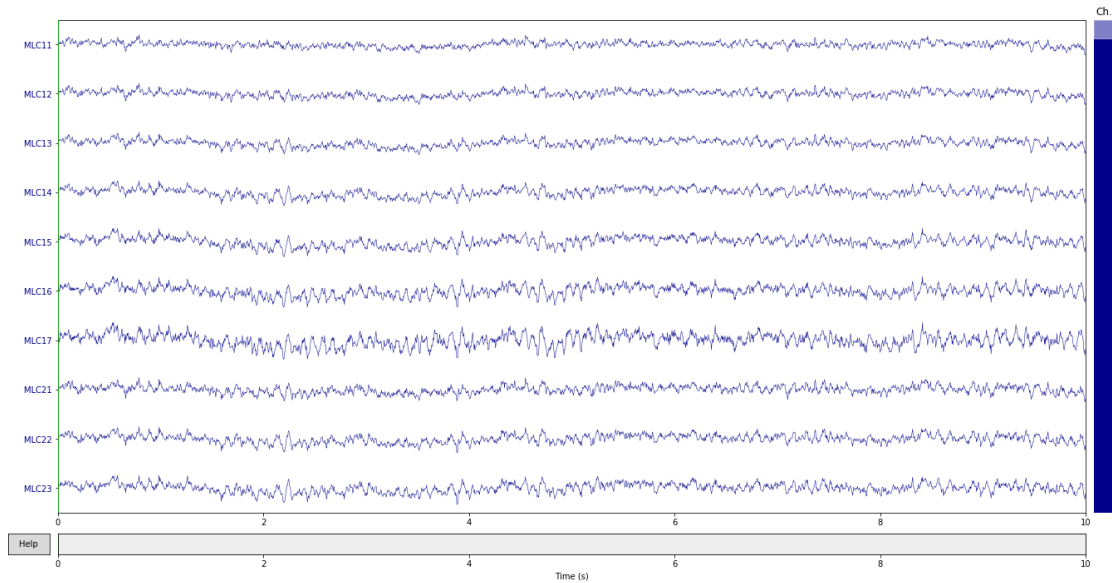
Automatic origin fit: head of radius 86.7 mm

Using origin 1.0, -23.9, 38.5 mm in the head frame

Processing data using tSSS with st_duration=10.0

Using 85/95 harmonic components for 0.000 (70/80 in, 15/15 out)

Processing 1 data chunk



Summary

- tSSS alone performs similarly to gradient compensation, but seems to suppress more higher frequency noise (at least for most channels, see below ...)
- Additional gradient compensation before tSSS removes slow drifts (0th order vs. >1)
- No difference between 1st order and 2nd order gradient comp (but there is definitely a difference without SSS, see above.)
- Subtle difference between 1st order and 3rd order gradient comp (3rd order a little less slow drifts).

1.4.6 C) Plot raw with tSSS (also on ref_meg) -> grad-comp

In [15]: tit = "no grad comp --> tSSS +mvmtComp +destMean +ref"

```
raw_sss = raw.copy().apply_gradient_compensation(0)
raw_sss = maxwell_filter(raw_sss, st_duration=10,
                        destination= destination, # in head coordinates
                        st_correlation=.98, # default
                        ignore_ref=False, # !! include them (otherwise completely m
                        coord_frame='head', # default
                        head_pos=head_pos, # head_pos or None
                        st_fixed=True,
                        st_only=False,
                        skip_by_annotation=('bad', 'edge', 'bad_acq_skip'))

fig = myrawplot(raw_sss, tit)
```

Compensator constructed to change 3 -> 0

Applying compensator to loaded data

Maxwell filtering raw data

Appending head position result channels and copying original raw data

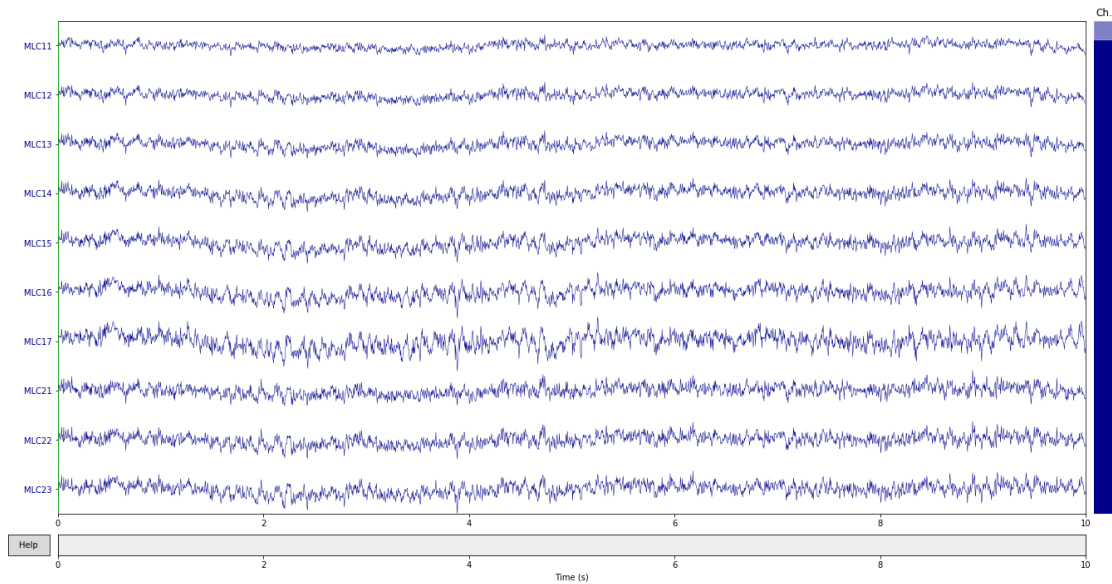
Bad MEG channels being reconstructed: ['MLP53', 'MRC12', 'MRC25']

Processing 0 gradiometers and 297 magnetometers (of which 288 are actually KIT gradiometers)

Automatic origin fit: head of radius 86.7 mm

Using origin 1.0, -23.9, 38.5 mm in the head frame

Processing data using tSSS with st_duration=10.0

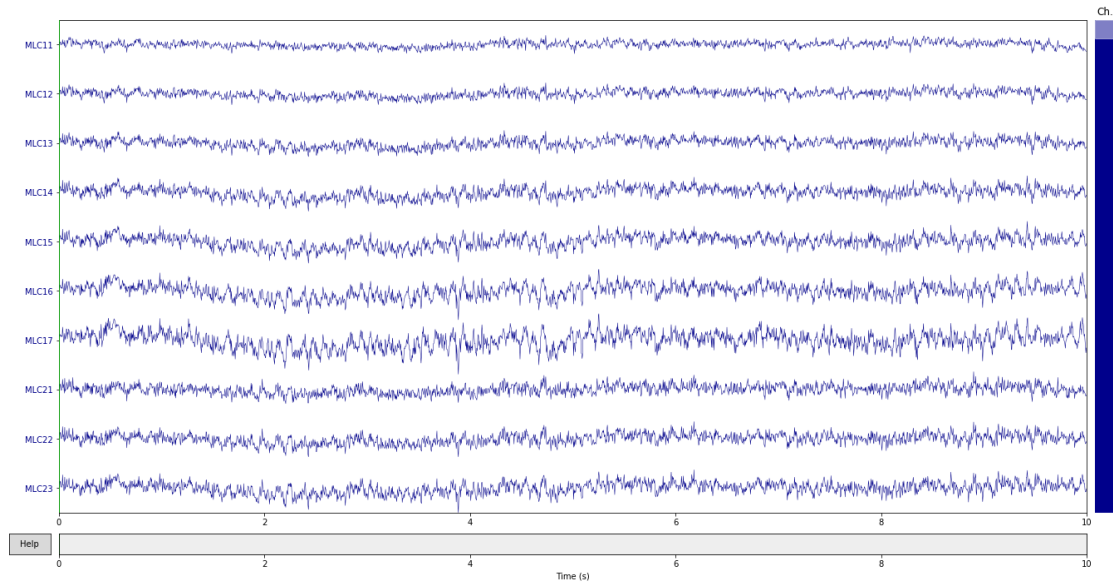


```
In [16]: tit = "no grad comp --> tSSS +mvmtComp +destMean +ref --> 3rd order grad comp"
```

```
# re-apply ctf grad comp:  
raw_sss.apply_gradient_compensation(3)  
fig = myrawplot(raw_sss, tit)
```

Compensator constructed to change 0 -> 3

Applying compensator to loaded data



Summary tSSS → gradient compensation is, as expected, not usable.

However, also only tSSS on the whole array (as suggested by staulu), including ref_meg introduces a lot of noise to the data channels.

1.4.7 D) Plot some spectra

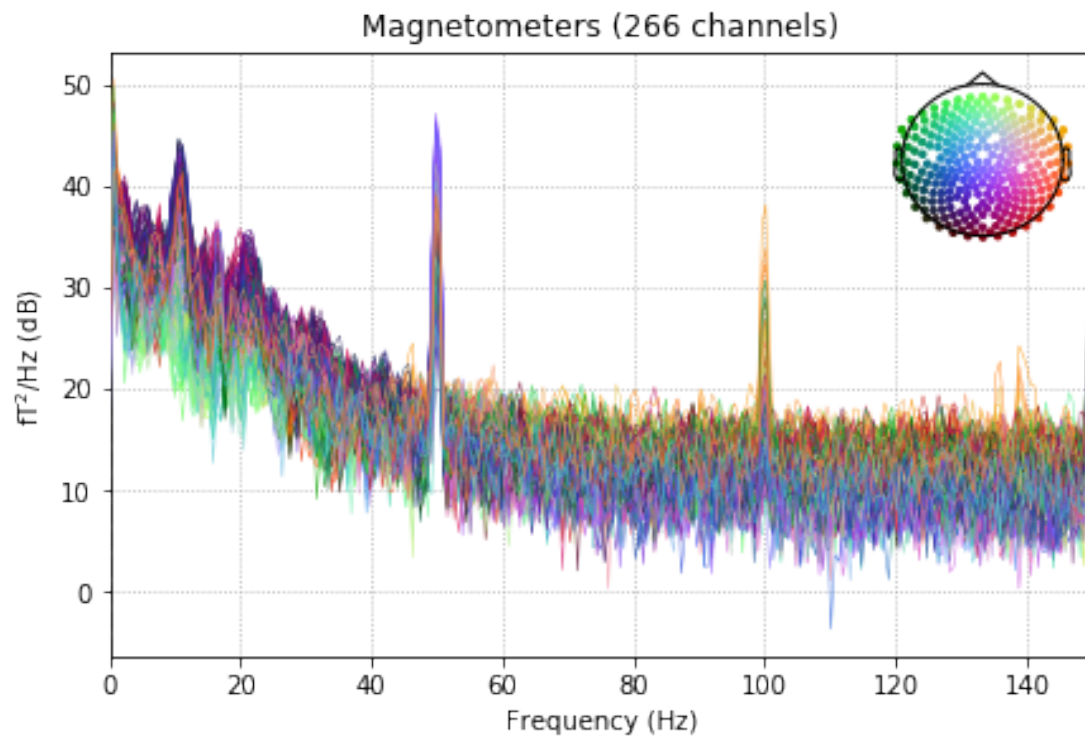
In [17]: `tit = "no grad comp"`

```
picks = mne.pick_types(raw.info, meg=True, ref_meg=False)
raw_tmp = raw.copy().apply_gradient_compensation(0)
fig = raw_tmp.plot_psd(fmax=150, picks=picks)
```

Compensator constructed to change 3 → 0

Applying compensator to loaded data

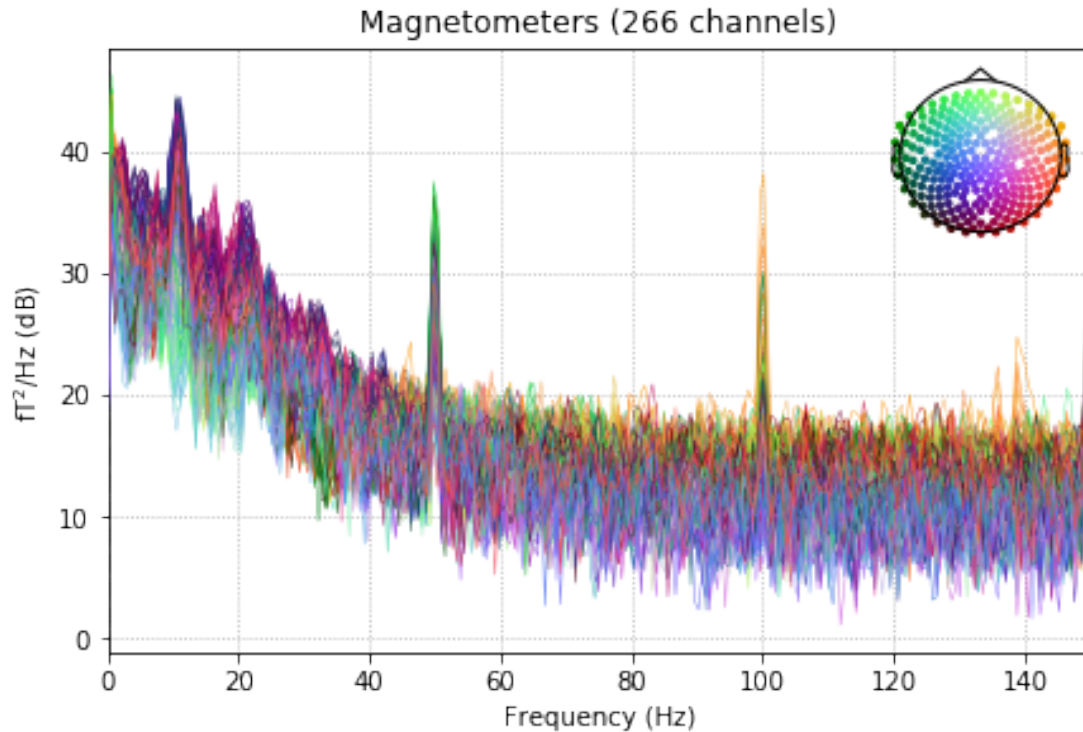
Effective window size : 1.707 (s)



```
In [18]: tit = "3rd order grad comp"
```

```
fig = raw.plot_psd(fmax=150, picks=picks)
```

```
Effective window size : 1.707 (s)
```



```
In [19]: # Maxwell only
         tit = "no grad comp --> tSSS +mvmntComp +destMean -ref"

         raw_sss = raw.copy().apply_gradient_compensation(0)
         raw_sss = mymaxwell(raw_sss, destination, head_pos)
         fig = raw_sss.plot_psd(fmax=150, picks=picks)
```

Compensator constructed to change 3 -> 0

Applying compensator to loaded data

Maxwell filtering raw data

Appending head position result channels and copying original raw data

Removing 5 compensators from info because not all compensation channels were picked.

Bad MEG channels being reconstructed: ['MLP53', 'MRC12', 'MRC25']

Processing 0 gradiometers and 269 magnetometers (of which 269 are actually KIT gradiometers)

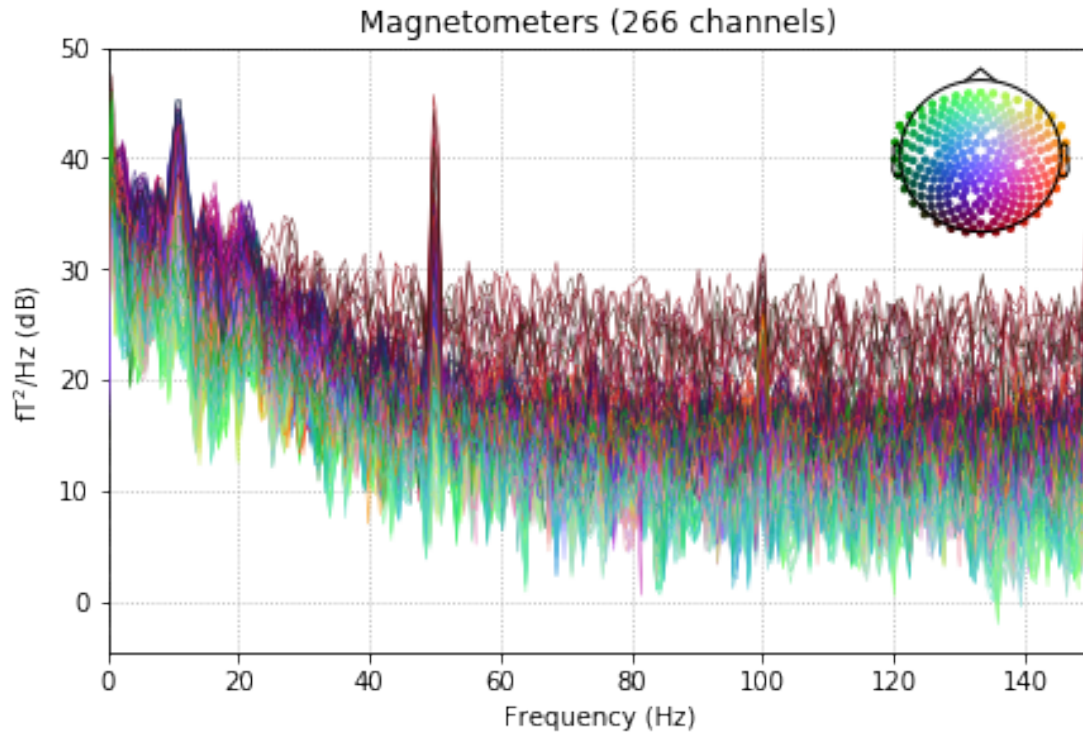
Automatic origin fit: head of radius 86.7 mm

Using origin 1.0, -23.9, 38.5 mm in the head frame

Processing data using tSSS with st_duration=10.0

Using 82/95 harmonic components for 0.000 (70/80 in, 12/15 out)

Processing 1 data chunk

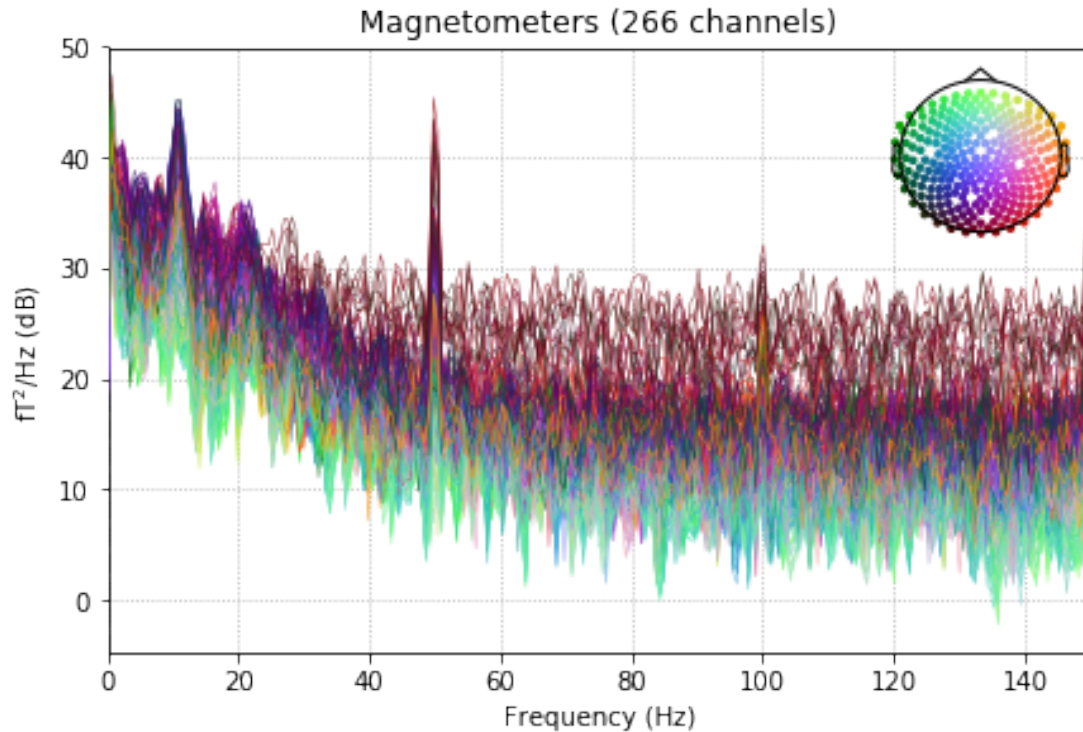


```
In [20]: # 3rd order grad compo + Maxwell
         tit = "3rd order grad comp --> tSSS +mvmntComp +destMean -ref"

         raw_sss = mymaxwell(raw, destination, head_pos)
         fig = raw_sss.plot_psd(fmax=150, picks=picks)
```

Maxwell filtering raw data

```
Appending head position result channels and copying original raw data
Removing 5 compensators from info because not all compensation channels were picked.
Bad MEG channels being reconstructed: ['MLP53', 'MRC12', 'MRC25']
Processing 0 gradiometers and 269 magnetometers
Automatic origin fit: head of radius 86.7 mm
Using origin 1.0, -23.9, 38.5 mm in the head frame
Processing data using tSSS with st_duration=10.0
    Using 85/95 harmonic components for    0.000 (70/80 in, 15/15 out)
Processing 1 data chunk
```



```
In [21]: tit = "3rd order grad comp --> tSSS +mvmntComp -destMean -ref"
```

```
raw_sss = maxwell_filter(raw, st_duration=10,
                        destination=None, # in head coordinates
                        st_correlation=.98,
                        ignore_ref=True,
                        coord_frame='head',
                        head_pos=head_pos, # head_pos or None
                        st_fixed=True,
                        st_only=False,
                        skip_by_annotation=('bad', 'edge', 'bad_acq_skip'))

fig = raw_sss.plot_psd(fmax=150, picks=picks)
```

Maxwell filtering raw data

Appending head position result channels and copying original raw data

Removing 5 compensators from info because not all compensation channels were picked.

Bad MEG channels being reconstructed: ['MLP53', 'MRC12', 'MRC25']

Processing 0 gradiometers and 269 magnetometers

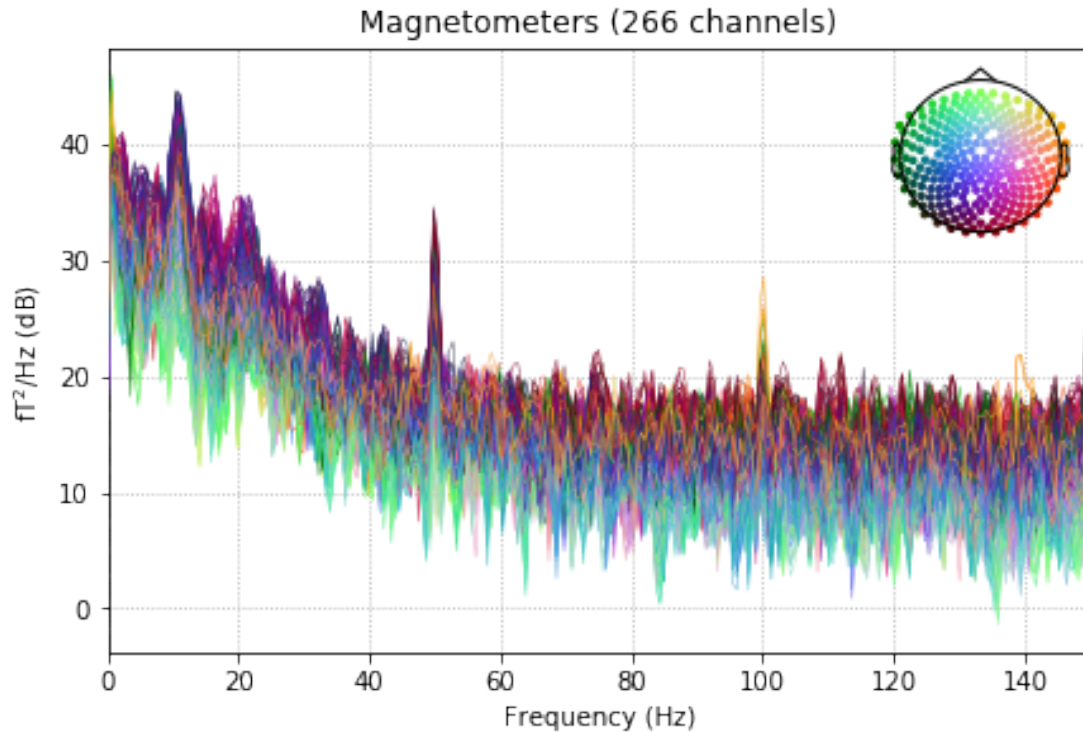
Automatic origin fit: head of radius 86.7 mm

Using origin 1.0, -23.9, 38.5 mm in the head frame

Processing data using tSSS with st_duration=10.0

Using 85/95 harmonic components for 0.000 (70/80 in, 15/15 out)

Processing 1 data chunk



Summary Applying any destination other than None (even using `raw.info['dev_head_t']['trans'][:3, 3]`) during Maxwell filtering makes some - mainly occipital - sensors extremely noisy > 25 Hz (broad band). `head_pos` alone, with `destination=None` does not have this effect.

-> WHAT IS GOING ON HERE?

1.4.8 Try OTP

```
In [22]: tit = '3rd order grad comp --> OTP --> tSSS +mvmntComp +dest -ref'
raw_clean = mne.preprocessing.oversampled_temporal_projection(raw)
raw_sss = maxwell_filter(raw_clean, st_duration=10,
                        destination= destination, # in head coordinates
                        st_correlation=.98,
                        ignore_ref=True,
                        coord_frame='head',
                        head_pos=head_pos, # head_pos or None
                        st_fixed=True,
                        st_only=False,
                        skip_by_annotation=('bad', 'edge', 'bad_acq_skip'))

# OTP again:
# raw_sss = mne.preprocessing.oversampled_temporal_projection(raw_sss)
myrawplot(raw_sss, tit)
fig = raw_sss.plot_psd(fmax=150, picks=picks)
```

Processing MEG data using oversampled temporal projection

Processing 1 data chunk of (at least) 10.0 sec with hann windowing

The final 0.001 sec will be lumped into the final window

Denoising 0.00 - 10.00 sec

Maxwell filtering raw data

Appending head position result channels and copying original raw data

Removing 5 compensators from info because not all compensation channels were picked.

Bad MEG channels being reconstructed: ['MLP53', 'MRC12', 'MRC25']

Processing 0 gradiometers and 269 magnetometers

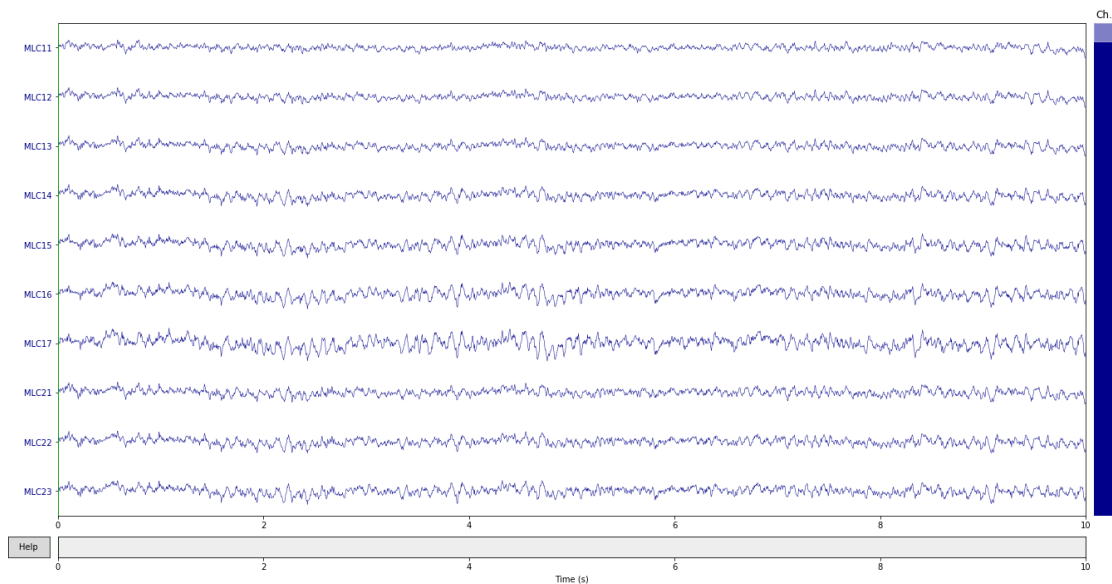
Automatic origin fit: head of radius 86.7 mm

Using origin 1.0, -23.9, 38.5 mm in the head frame

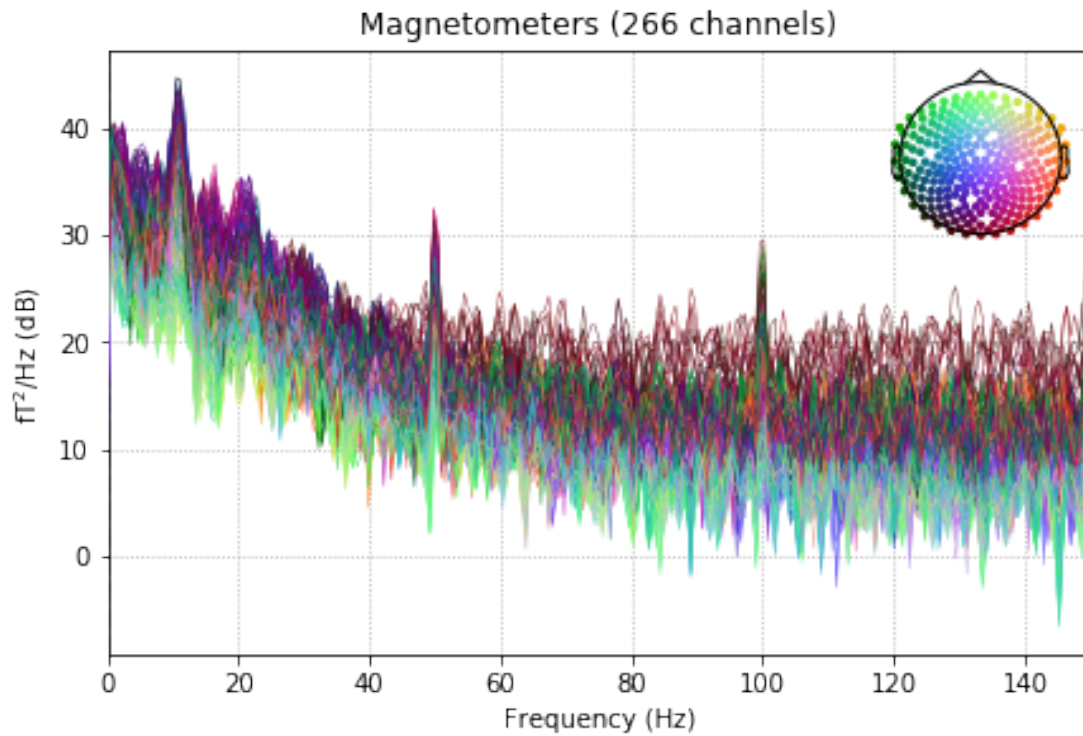
Processing data using tSSS with st_duration=10.0

Using 85/95 harmonic components for 0.000 (70/80 in, 15/15 out)

Processing 1 data chunk



Effective window size : 1.707 (s)



Summary OTP before Maxwell filtering alleviates the broad-band-noise problem a lot, and completely until 45 Hz. Cool! :)

(A second run of OTP after maxwell_filter does give any obvious improvement.)

Question: How should we adapt st_correlation to OTP? What is the criterion to select this parameter?