Not for USA

Elekta Neuromag

# MaxFilter™

## User's Guide

**Software version 2.2**

October 2010

ELEKTA NEUROSCIENCE

| Printing History | Neuromag p/n | Software | Date |
|---|---|---|---|
| 1st edition | NM24057A | 2.2 | 2010-10-04 |

CE 0537

# Contents

1

CHAPTER 1 **Introduction**

# 1.1 Overview

This User's Guide gives detailed explanation of the Elekta Neuromag *MaxFilter* software for MEG data analysis.

*MaxFilter* is intended to be used with Elekta Neuromag MEG products in suppressing magnetic interference coming from inside and outside of the sensor array, in reducing measurement artifacts, in transforming data between different head positions, and in compensating disturbances due to magnetized material on the head and due to head movements.

This Chapter presents a general overview and main functionalities of the software. Chapters 2-5 describe how to use the program and to control the functionality and parameters. Chapter 6 discusses the performance and limitations of *MaxFilter*. Mathematical background and some further information are included in the Appendices.

# 1.2 Maxwell filtering

*Signal Space Separation* (SSS) is a method that utilizes the fundamental properties of electromagnetic fields and harmonic function expansions in separating the measured MEG data into three components (Figure 1.1):

$\boldsymbol{b}_{in}$: The brain signals originating inside of the sensor array (space $S_{in}$).

$\boldsymbol{b}_{out}$: External disturbances arising outside of the sensor array (space $S_{out}$).

$\boldsymbol{n}$: Noise and artifacts generated by the sensors and sources of interference located very close to the sensors (space $S_T$).

The disturbing magnetic interference is suppressed by omitting the harmonic function components corresponding to unduly high spatial frequencies, by neglecting the $S_{out}$-space component $\boldsymbol{b}_{out}$, and by reducing the $S_T$-space component $\boldsymbol{n}$. Since the method is based directly on Maxwell's equations, the operation can be called Maxwell filtering; hence the name *MaxFilter*.

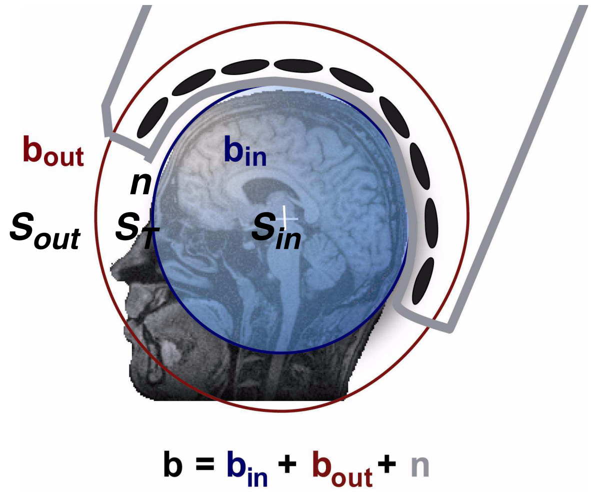


**Figure 1.1** *The geometry in Maxwell filtering. One hypothetical spherical shell inside of the sensor array encloses the subject's brain, and another one encloses all MEG sensors. The radii of the shells are determined, respectively, as the smallest and largest distances from the origin to the sensor locations.*

The basic Maxwell filtering operation can be regarded as spatial filtering, because separation of $\boldsymbol{b}_{in}$ and $\boldsymbol{b}_{out}$ is done on the basis of the spatial patterns and is independent of time. Spatial separation can suppress only *external* interference emanating from space $S_{out}$, such as electromagnetic pollution due to power lines, radio communication, traffic, elevators etc. External interference can also arise in the patient. For instance, normal cardiac and muscular activation cause fields detectable by MEG sensors, and any pieces of magnetized material in/on the body may cause very large disturbances.

Identification and suppression of the $S_T$-space components require additional knowledge of the temporal dynamics. The temporal extension of the SSS method (*tSSS*[1]) significantly widens the software shielding capability of *MaxFilter,* because *tSSS* can suppress also *internal* interference that arises in the $S_T$-space or very close to it. Such disturbances can be caused, for example, by magnetized pieces in/on the subject's head (such as dental work, braces, or magnetized left-overs in burr-holes), or by pacemakers or stimulators attached to the patient.

Maxwell filtering inherently transforms measured MEG signals into *virtual channels* in terms of harmonic function amplitudes. Because the virtual channels are independent of the device, they offer a straightforward method for estimating corresponding MEG signals in other sensor arrays. This function called *MaxMove* provides an elegant way to transfer MEG signals between different head positions and to compensate for disturbances caused by head movements during recordings.

1. In version 2.1, tSSS had a synonym MaxST.

*MaxFilter* is easy to apply and the default settings provide good results in most cases. Maxwell filtering as implemented in this program is an irreversible operation. The expansion corresponding to the space outside of the sensor array ($\boldsymbol{b}_{out}$) is discarded before saving the result. In addition, *tSSS* projects out identified sensor-space artifact waveforms.

**Note:** The original recording cannot be reconstructed from the result FIFF file. Therefore, it is very important to keep also the original data files on suitable backup media after applying *MaxFilter*.

The mathematical basis of Maxwell filtering is described briefly in Appendix A. The Signal Space Separation algorithms and their applications are discussed in detail in:

1. S. Taulu, M. Kajola, and J. Simola. Suppression of interference and artifacts by the signal space separation method. *Brain Topography* **16**(4), 269-275, 2004.
2. S. Taulu, and M. Kajola. Presentation of electromagnetic multichannel data: the signal space separation method. *Journal of Applied Physics*, **97**(12), 124905, June 2005.
3. S. Taulu, J. Simola, and M. Kajola. Applications of the signal space separation method. *IEEE Transactions on Signal Processing*, **53**(9), 3359-3372, 2005.
4. S. Taulu, and J. Simola. Spatiotemporal signal space separation method for rejecting nearby interference in MEG measurements. *Physics in Medicine and Biology,* **51**, 1759-1768, 2006.

## 1.3 Software functionality

*MaxFilter* 2.2 software provides three separate programs:

**MaxFilter**
The main command-line application for Maxwell filtering. This program includes also the *tSSS* and *MaxMove* functions.

**MaxAve**
Off-line version of the on-line *averager*, provided for convenient re-averaging of raw data before or after Maxwell filtering.

**MaxFilter_GUI**
Graphical user interface program which collects the input arguments, and then starts *MaxFilter* or *MaxAve* as a subprocess. Information of the data processing is displayed on the main display and in a log window.

The main functions of *MaxFilter* are:

**Software shielding**

By subtracting the component $\boldsymbol{b}_{out}$ from measured signals $\boldsymbol{b}$, the program performs software shielding on the measured MEG data (i.e. interference suppression).

**Automated detection of bad channels**

By comparing the reconstructed sum $\boldsymbol{b}_{in} + \boldsymbol{b}_{out}$ with measured signals $\boldsymbol{b}$, the program can automatically detect if there are MEG channels with bad data that need to be excluded from Maxwell filtering.

**Spatio-temporal suppression of $S_T$-space artifacts**

By subtracting the reconstructed *waveforms* $\boldsymbol{b}_{in}(t) + \boldsymbol{b}_{out}(t)$ from measured signals $\boldsymbol{b}(t)$, the program can identify and suppress artifact waveforms which arise in the $S_T$-space.

**Transformation of MEG data between different head positions**

By transforming the component $\boldsymbol{b}_{in}$ into harmonic amplitudes (i.e. virtual channels), MEG signals in a different head position can be estimated easily.

**Correction of disturbances caused by temporary head movements**

By extracting head position indicator (HPI) signals applied continuously during a measurement, the data transformation capability is utilized to estimate the corresponding MEG signals in a static reference head position.

## 1.4 Software safety

This manual contains important hazard information which must be read, understood and observed by all users. General limitations of the program are included in the following Chapters. For your convenience all warnings that appear in the manual are presented below.

**Warning:** It is important that the user inspects both the input and the output data visually to judge the quality of the *MaxFilter* result.

**Warning:** If the fine-calibration and cross-talk correction data are not available, the performance of *MaxFilter* may not be as good as with the fine-calibrated system.

**Warning:** Special care should be taken to ensure that right fine-calibration data are used for imported or old data for which the default calibration does not apply.

**Warning:** If the threshold of the automated bad channel detection is too small, the program may classify good channels as bads, and if it is too high, some bad channels may remain undetected.

**Warning:** If *tSSS* is applied on averaged data or if there were several saturated channels in raw data, the result must be inspected very carefully.

**Warning:** The user must judge the result carefully if the *tSSS* correlation limit is lowered from the default value.

**Warning:** *MaxMove* operations require that the initial and reference coordinate transformations are defined correctly.

**Warning:** Head position calculation errors affect the data quality after movement compensation. The user must inspect the head position fitting error and goodness before data analysis.

**Warning:** If internal active shielding was applied in the input file, the user must not perform data analysis on *MaxFilter* output files obtained with the maintenance options `-nosss` or `-ctc only`.

**Warning:** Spatiotemporal signal space separation (*tSSS*) may diminish brain signals arising from very strong, superficial sources.

**Warning:** Head movements generate physiological background signals due to muscle and motor cortex activation. These disturbances are not suppressed by *MaxFilter* and may deteriorate the source localization accuracy.

**Warning:** *MaxFilter* cannot suppress movement artifacts in data segments with rapid head movements, such as those arising during motor seizures. Movement artifacts may deteriorate the localization accuracy of the system.

2

# CHAPTER 2 Using MaxFilter

## 2.1 Background

*MaxFilter* can be applied to a FIFF-file with raw data or averaged MEG measurement results. The parameters needed in calculations are set to well-defined initial values, separately for Elekta Neuromag®, Neuromag System and Neuromag-122 data, and you can run the program without changing these values. However, sometimes it may be useful to tune the details of the Maxwell filtering operation.

In brief, you can change the origin of the expansions and the dimensions and selection of the internal and external multipole bases. Optionally, you can manually identify and set bad channels that are not taken into account in the reconstruction. In cases where the source of interference is located inside or very close to the sensor array, it is recommended to use the spatio-temporal Maxwell filtering, *tSSS*. In addition, you can transfer data between different head positions and compensate disturbances due to head movements using *MaxMove*.

**Warning:** It is important that the user inspects both the input and the output data visually to judge the quality of the *MaxFilter* result.

## 2.2 Launching the program

### 2.2.1 Graphical User Interface

The graphical user interface, GUI, is a program that collects the input parameters of *MaxFilter* or *MaxAve*, and then runs the command-line program as a subprocess. You can monitor the execution on a log window (Section 2.8). The GUI can be launched in

**HP-UX 11:**
Click the Neuromag toolbox icon labeled as *MaxFilter*.

**Linux:**
Select the application menu *Neuromag –> MaxFilter*.

**Command line:**
`/neuro/bin/X11/maxfilter_gui.`

Upon launching the program checks available licenses, and displays a welcome logo (Figure 2.1). Click *Hide* to close the window and to start using the main dialog (Figure 2.2).

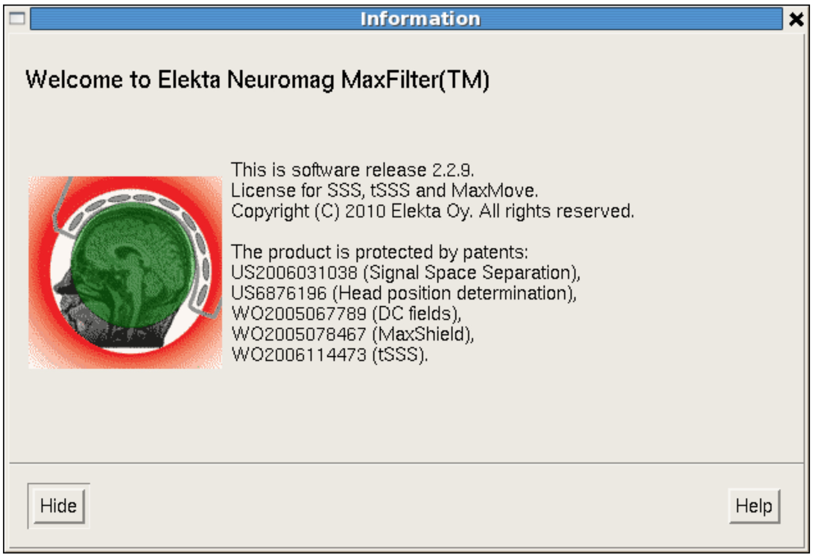


**Figure 2.1** *MaxFilter welcome window.*

### 2.2.2 Command line

Alternatively, you can start *MaxFilter* from a command line as

```
/neuro/bin/util/maxfilter -f input_file.fif
[options]
```

If no arguments are given, the program gives just a brief message:

```
usage: maxfilter -f <infile> [options]
'maxfilter -help' shows available options.
```

A comprehensive list of the options is given in Appendix E.

## 2.3 GUI main window

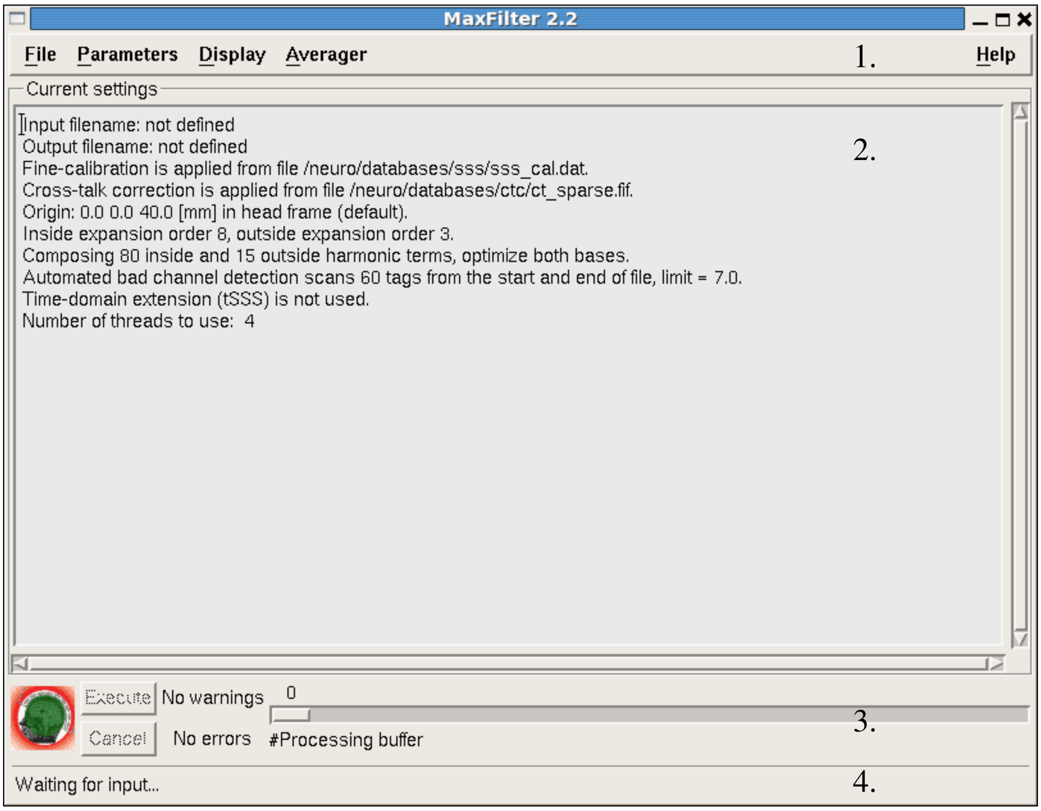


**Figure 2.2** *MaxFilter GUI main window.*

The main window consists of the following areas:

1. The menubar.
2. A text area for showing current parameter values, or optionally a drawing area for estimated head position parameter curves.
3. Display area for showing the processing status.
4. A message text label at the bottom of the window.

The menus and controls are described in the following sections.

When you have defined the input and output filenames and modified the parameters, press `Execute` to start Maxwell filtering. The progress scale bar indicates the number of processed buffers. You can press `Cancel` to cancel processing. Then you can modify the parameters and try again (press `Execute`).

The labels next to the `Execute` button indicate the number of warnings and errors reported by *MaxFilter*. During execution, the background colour of these labels is green. If there are warnings, background of the label `warnings` changes to red and the label reports the number of warnings. If *MaxFilter* is terminated due to a fatal error, background of the label `errors` turns to red.

The scale bar and the text label at the bottom indicate the status of the Maxwell filtering operation. You can view more detailed information of the *MaxFilter* output and warnings by selecting *Show log...* from the *File* menu.

When you start *MaxMove* processing, the program starts head position estimation. Estimated head positions and fitting parameters can be shown in the drawing area of the main window (see Chapter 4).

## 2.4 Menus

You can access any of the menu choices directly by first pressing the `Alt-`<*underlined letter in the menu name*> and then <*underlined letter in the menu choice*>. For example, to select the *Set directory...* item from *File* menu, press `Alt-f` followed by `d`. The same procedure applies to menus found in other windows of the program as well.

### 2.4.1 File

**Load data...**
: Open a new file for processing and set the Maxwell filtering job (Section 2.5).

**Working directory...**
: Set the working directory. The directory is kept for output files until a new working directory is selected.

**Output options**...
: Set the output file and other output options (Section 2.7).

**Show log...**
: Show a log window to list the `stdout` and `stderr` outputs of *MaxFilter* and *MaxAve* (Section 2.8).

**Exit**
: Quit the program.

### 2.4.2 Parameters

**Origin...**
: Set the harmonic function expansion origin and coordinate frame (Section 3.2).

**Multipole selection**...
: Set the harmonic function expansion orders and optimization for component selection (Section 3.3).

**Bad channels...**
: Control the bad channel settings (Section 3.4).

**tSSS parameters...**
: Change *tSSS* parameters (Section 3.5).

**LP filter and downsample...**
Set low-pass filtering and downsampling (Section 3.6).
**Head position estimation**...
Set the parameters to estimate head positions (Section 4.3).
**Fine-calibration...**
Set the fine-calibration adjustment file (Section 3.7).
**Cross-talk compensation...**
Set the cross-talk correction file (Section 3.7).
**Parallel processing...**
Set the number of parallel threads (Section 3.8).

### 2.4.3 Display

**Load head positions...**
Open a file for viewing previously estimated head positions (Section 4.4).
**Current settings...**
Show the current *MaxFilter* settings on the main window.
**Fit parameters...**
Show the head position estimation parameters (Section 4.4).
**Head positions...**
Show the estimated head positions (Section 4.4).
**Viewing scales...**
Set scalings for head position displays (Section 4.4).

### 2.4.4 Averager

**Load raw data...**
Select the raw data file to be averaged (Section 5.3).
**Output file**...
Select the file for saving the averages (Section 5.3).
**Rejection limits**...
Change averaging rejection limits (Section 5.4).

### 2.4.5 Help

**Why the beep?**
A brief explanation why the terminal bell was rung.

**View manual...**
Start a PDF viewer program to view this manual.

**On version...**
Show current version of *MaxFilter, MaxAve* and *MaxFilter_GUI.*

## 2.5 Loading data

*MaxFilter* accepts evoked or raw-data FIFF files as input. New data are loaded by selecting *Load data...* from the *File* menu (Figure 2.3).

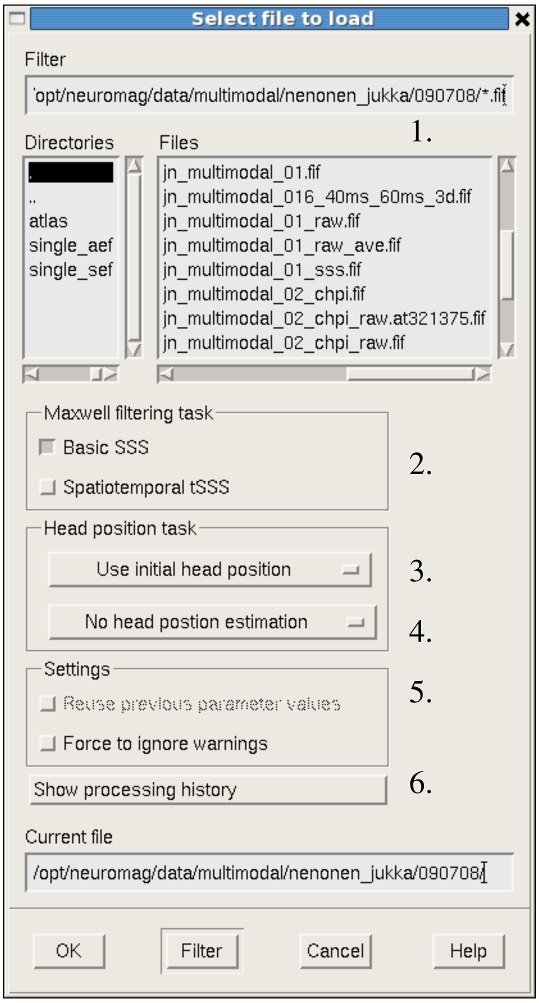


**Figure 2.3** *Data loading dialog.*

The dialog has controls for

1. Input file selection.
2. Setting the SSS or tSSS interference suppression.
3. Optional selection for head position transformation.
4. Selection for head position estimation or movement correction.
5. Reusing previously set parameters and forcing to ignore warnings.
6. Showing the processing history of the selected file.

### 2.5.1 Setting Maxwell filtering job

After selecting the file, you must select the *Maxwell filtering task* in the file loading dialog. You can select either the spatial SSS or the spatio-temporal *tSSS* for interference suppression.

The *Head position task* includes two menus. The first menu includes the choices to define the target head position:

*Use initial head position*
: Default, use the initial hpifit result as the reference position.

*Transform to default head position*
: The head and device coordinate axes are set to coincide. The origin of the device coordinate frame is used as the origin of the SSS expansions.

*Read head position from a file*
: Pops up a file selection dialog, and the coordinate transformation in the selected file serves as the reference position.

The second menu contains the options for continuous head position estimation and movement correction:

*No head position estimation*
: Default, *MaxMove* tasks are not applied. Head position is however transformed according the previous selection.

*Movement correction (SSS)*
: The head positions are estimated if they are not included in the input file. SSS interference suppression is done and the data are transformed to the reference head position (Section 4.6 on page 51).

*Movement correction (tSSS)*
: The same as above except that tSSS is employed instead of SSS interference suppression.

*Head position estimation, no SSS*
: Head positions are estimated from continuous HPI signals (see Section 4.4 on page 47). SSS interference suppression is not performed and the data are not transformed to the reference position.

*Head position estimation with tSSS*
: Head positions are estimated from continuous HPI signals and tSSS interference suppression is performed. The data are however not transformed to the reference position.

Upon selecting a new input file the program cleans all parameters values and resets the GUI dialogs. If you press the button *Reuse the previous settings*, the program applies the parameter values that were applied for the previous file. All output filenames are however cleared.

If you press *Show processing history*, the program displays processing history of the selected file in the log window (see Section 2.8).

When the input data are successfully loaded, the working directory is changed to the directory containing the file. You can change the working directory by selecting *Working directory...* from the *File* menu. Thereafter, all processed files will be located in the selected working directory automatically, until you select a new working directory. This option is useful when you don't have permission to write in the directory where the input file resides.

### 2.5.2 Ignoring warnings

It is possible to bypass the warnings and error messages using the button *Force to ignore warnings*. The program tries to continue execution even if warnings or nonfatal error messages are encountered. The program is however terminated in a fatal error case.

**Note:** Normally, the program checks if the output FIFF file already exists, and refuses to overwrite an existing file. When ignoring warnings, the program tries to overwrite an existing file without asking the user. However, overwriting cannot be done if the existing file is write-protected.

## 2.6 Time scale of raw data

*MaxFilter* automatically sets the zero time to the beginning of the measurement session, i.e., the moment when the data acquisition was started. Usually, there is a delay before recording of raw data is started and the signal time does not start from zero.

The raw data are stored in tagged blocks (called 'tags') in the FIFF file. *MaxFilter* first extracts the number of samples (*nsamp*) in a block. All blocks (except the very last one in the file) are assumed to contain the same *nsamp*. Time per block is then obtained as *tbuf = nsamp/sfreq* where *sfreq* is the sampling frequency. The program counts the number of data blocks (*nbuf*), the number of initial data skips (*nskip1*) and the number of skips in the middle of data (*nskip2*, i.e., if raw data recording was interrupted during data collection).

Thereafter, *MaxFilter* sets the start time as *t1 = nskip1 * tbuf,* and end time as *t2 = (nskip1 + nbuf + nskip2)* tbuf*. All reporting of the program is based on this time scale. As an example, *show_fiff* output from a raw data file sampled with 1000 Hz shows:

```
104 = {              102 = raw data
      301 = data skip       [22]
      300 = data buffer     [128]
      301 = data skip       [55]
      300 = data buffer     [120]
105 = }              102 = raw data
```

Here *nsamp = 1000* and *tbuf = 1 s*. Thus, *t1 = 22.0 s* and *t2 = (22 + 128 + 55 + 120) * 1 s = 325.0 s*.

Very long recordings may be splitted into continuation files. In such cases the program automatically detects if the file contains a tag labeled as `FIFF_FIRST_SAMPLE` and sets the start and end time according to it.

**Note:** If you want to browse raw data exactly with this time scale on *graph*, you need to open *Displays -> Control Panel*, double click widget `file`, click `resources`, and de-activate `compress-skips`. If you are using the program *mne_browse_raw*, you need to select *Keep the initial skip* in the file opening dialog. Otherwise, the time shown by *graph* or *mne_browse_raw* may differ from *MaxFilter* reporting.

## 2.7 Output options

After selecting the input datafile you can modify the output options by selecting *Output options...* from the *File* menu (Figure 2.4).

Besides file selection, the dialog has controls for changing the output format and for setting data skips.

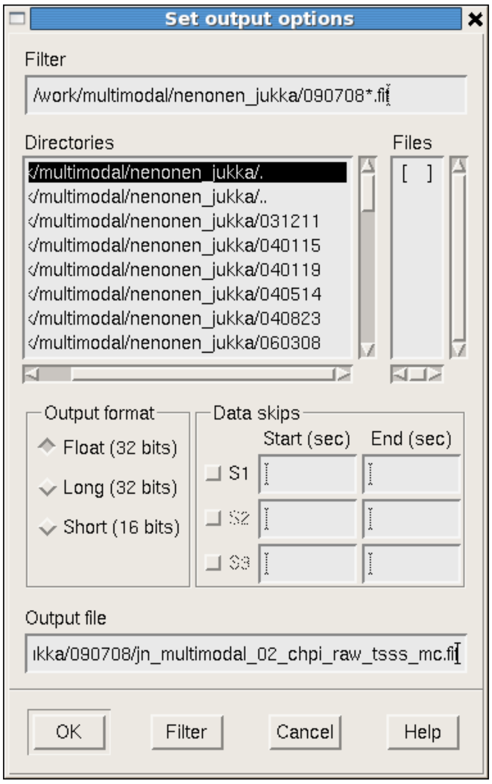


**Figure 2.4** *Output options.*

### 2.7.1 Filename

If the output filename is not set, the program creates automatically a file named as `input_file_`*job_ext*`.fif` where *job_ext* is one or a combination of the following:

- `_sss` for spatial Maxwell filtering,
- `_tsss` for tSSS,
- `_trans` for data transformed into a different head position,
- `_quat` for head position estimation,
- `_mc` for movement correction.

If you want to set the output filename manually, you can enter the desired name in the file selection dialog text field *Output file.*

Output is not generated if the output file already exists. In that case, you should either remove the old file, set a new output filename, or select *Force to ignore warnings* when loading the input file.

Normally, the output file is produced in the working directory. The program gives an error if the file or the directory is not writable. In such cases you can change the working directory, or set the output filename in another writable directory. Alternatively, you can copy or link the input file to a writable working directory.

**Note:** If the output files are written in a different directory than input files, special care should be taken to avoid mixing files of different patients.

### 2.7.2 Data packing

The result of the processing is saved in a FIFF-file. You can define the data packing by selecting 32-bit *float*, 32-bit integer (*long*), or 16-bit *short* packing.

**Note:** If you don't select data packing: 1) Raw data files are saved in *float* format. Therefore, the result file becomes twice as large as the input file if the original data were packed as *short*. 2) Evoked data are packed in the same format as the input file.

Changing of data packing may be needed, for example, in processing 32-bit data acquired with a new generation electronics. Neuromag Data Analysis Software release 3.3 (and earlier) cannot process data stored with 32-bit integer packing. Therefore, the data packing needs to be *short* or *float* when saving the result file.

The current FIFF format does not support files larger than 2 GB. If the size of a 16-bit format input file is over 1 GB, output in 32 bit format becomes too large. Therefore, the program gives an error message and refuses to continue processing if the output filesize seems to become over 2 GB. In such cases, you can apply downsampling, change the data packing to *short*, or use the data skipping to produce a file smaller than 2 GB.

### 2.7.3 Data skipping

You can select 1-3 time intervals if you want to skip some sections from the input data. In such cases the program writes a data skip in the output file instead of processing the data. Thus, the time scale of the input FIFF file is retained. All data skip intervals should be given according to the time scale setting explained in Section 2.6.

### 2.7.4 Processing history

When saving processed data, *MaxFilter* updates the processing history block (if it is found), or creates a new processing history. The block includes the Maxwell filtering parameter values and information about the cross-talk and fine-calibration correction (see Section 3.7.2).

An example of the processing history block:

```
104 = {              900 = proc. history
104 = {              901 = proc. record
103 = block ID       1.2 0x1279dec8e40000 Tue 14 Oct 2008
204 = date           Tue 14 Oct 2008 01:28:33 PM EEST
212 = scientist      jne
113 = creat.program  maxfilter 2.1.12
104 = {              502 = SSS info
264 = SSS task       5
263 = SSS crd frame  4
265 = SSS origin     3 floats
266 = SSS ins.order  8
267 = SSS outs.order 3
268 = SSS nr chnls   306
269 = SSS components 95 ints
278 = SSS nfree      61
243 = HPI g limit    0.98
244 = HPI dist limit 0.005
105 = }              502 = SSS info
104 = {              504 = MaxST info
264 = SSS task       10
272 = SSST subsp.cor. 0.98
279 = SSST buflen    3.9959
105 = }              504 = MaxST info
104 = {              501 = CTC correction
103 = block ID       1.1 0x1083f957d70000 Fri 04 May 2001
204 = date           Fri 01 Sep 2000 03:00:00 AM EEST
113 = creat.program  create_ct_matrix 1.0
800 = CTC matrix     2: 306 x 306 (5360 floats)
3417 = proj item chs MEG0113:MEG0112:MEG0111:...
105 = }              501 = CTC correction
104 = {              503 = SSS finecalib.
```

```
270 = SSS cal chnls  2: 2 x 306 (612 ints)
271 = SSS cal coeff  2: 14 x 306 (4284 floats)
105 = }              503 = SSS finecalib.
105 = }              901 = proc. record
105 = }              900 = proc. history
```

If the input file was already processed, the program exits with an error message "ERROR: SSS was already applied!" and no output is produced. The processing parameters of such files are shown on the GUI log window. You can however select *Force to ignore warnings* if you want to reprocess the input file despite the error message.

**Note:** Forcing *MaxFilter* reprocessing may distort the result if different expansion origin or order settings are used.

### 2.7.5 Degrees of freedom

When a datafile is processed with *MaxFilter*, the resulting data have the same number of degrees of freedom (NDOF) than the number of multipole amplitudes in the inside expansion.

Some data analysis programs need to know the NDOF, because decompositions like principal component analysis perfomed on the data show exactly NDOF non-zero eigenvalues, all other *nchan* - NDOF eigenvalues become zero.

Therefore, *MaxFilter* writes a integer tag (extension to the FIFF format) defined as `#define FIFF_SSS_NFREE 278`
which shows the NDOF in the file. The NDOF depends on the inside expansion order (Section 3.3.1) and optimization of multipole selection (Section 3.3.2).

## 2.8 Logging the output

You can display the output of *MaxFilter* and *MaxAve* by selecting *Show log...* from the *File* menu (Figure 2.5). The log window has three areas: the top text area lists the normal `stdout` output of the program, the middle text area is for displaying all `stderr` warnings, and the lowest text area displays the execution command which the GUI composes for running *MaxFilter* or *MaxAve*.

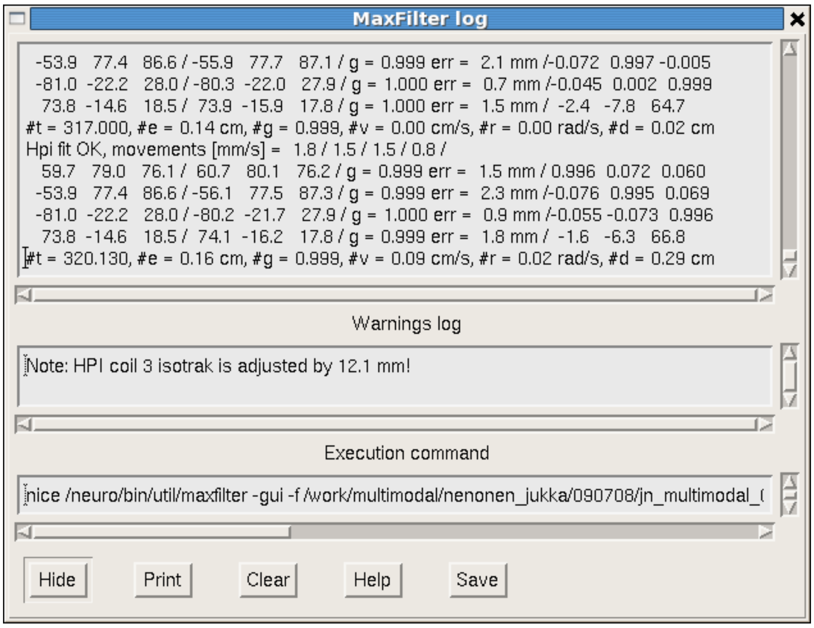


**Figure 2.5** *MaxFilter logging window.*

## 2.9 Command-line arguments

When you use directly the command-line program *MaxFilter*, you can define the name of the input file with the option `-f` *input_file.fif*, or `-f` *<input_directory_path>/input_file.fif*.

The name of the output file is set with the option `-o` *output_file.fif*, or `-o` *<output_directory_path>/output_file.fif*

Note that if you don't specify the output file or give only the plain name *output_file.fif*, the program automatically writes the output file into the current working directory even if the input file is read from another directory.

You can define the data packing with the option `-format` *type*, where *type* can be *short* (16-bit short packing), *float* (32-bit float packing) or *long* (32-bit integer packing).

Data skip intervals can be set with the option `-skip` *<t1 t2> ...<t1 t2>* , where *<t1 t2>* is a pair of times (*t1* < *t2*) indicating a time interval to be skipped from processing.

If you want to see the processing history of a file which was already processed, select `-f` *input_file.fif* `-history`.

Finally, you can bypass the warnings and error messages using the option `-force`.

# CHAPTER 3 MaxFilter parameters

## 3.1 Coordinate systems

There are two coordinate systems which can be used: the *head* coordinates and the *device* coordinates. Both are right-handed Cartesian coordinate systems. The direction of $x$ axis is from left to right, that of $y$ axis to the front, and the $z$ axis thus points up.

The origin of the device coordinate system is located at the center of the posterior spherical section of the helmet with $x$ axis going from left to right and $y$ axis pointing front. The $z$ axis is, again normal to the $xy$ plane with positive direction up.

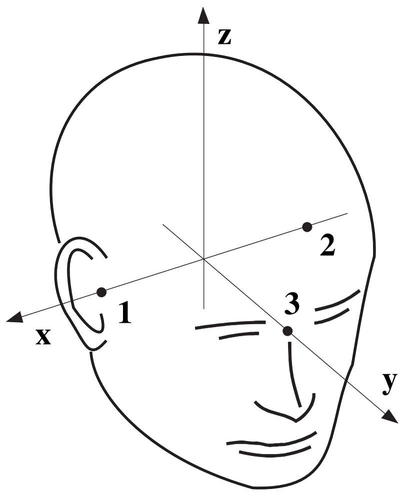


**Figure 3.1** The head coordinate system.

The $x$ axis of the head coordinate system passes through the two periauricular or preauricular points digitized before acquiring the data with positive direction to the right. The $y$ axis passes through the nasion and is normal to the $x$ axis. The $z$ axis points up according to the right-hand rule and is normal to the $xy$ plane.

## 3.2 Expansion origin

The outcome of the expansions for $\boldsymbol{b}_{in}$ and $\boldsymbol{b}_{out}$ depends on the location of the harmonic function expansion origin. In general, best results of Maxwell filtering are expected when the origin is defined so that the $S_{in}$-space in Figure 1.1 on page 6 covers the whole brain.

### 3.2.1 Default setting

**Note:** *MaxFilter* versions 2.1 and 2.2 utilize a different default origin setting than version 2.0.

The default origin setting follows the Source Modelling program, Xfit; see NM20568A Source Modelling Software User's Guide section 5.4: "The MEG sphere model". The origin is set to point (0, 0, 40 mm) in the *head* coordinate frame. If the input file does not have a coordinate transformation, the origin is set to point (0, 0, 0) in the *device* coordinate frame.

It is also possible to customize the default origin setting: 1) the coordinates of the fixed origin can be changed, or 2) the program can be set to always fit the origin to isotrak data. See Section 3.10.2 on page 41 for details how to customize the default settings.

### 3.2.2 Changing the origin and frame

For changing the origin, select *Origin...* from the *Parameters* menu (Figure 3.2) .

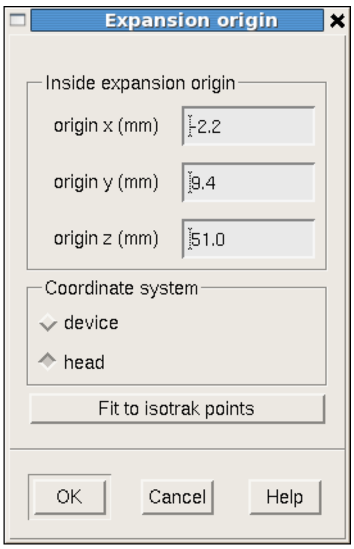


**Figure 3.2** *Setting the expansion origin and coordinate frame.*

Enter the desired coordinate values (in mm) and select the *head* or *device* coordinate frame. Switching between *head* and *device* selection converts the origin values shown in the dialog to the selected coordinate frame. For example, you can inspect how far the origin set in the head coordinates is from the device origin (0, 0, 0).

**Note:** If the origin is set outside of the sensor array or closer than 5 cm to the nearest sensor the program reports an error and terminates.

### 3.2.3 Fit to isotrak or anatomy

If you press *Fit to isotrak points*, all digitized points (including cardinal landmark points) are searched, and a sphere is fitted to these points. The fitted values are shown on the dialog in the *head* coordinate frame.

If the input file does not contain a suitable coordinate transformation, the program fits a sphere to all sensor locations. The fitted point is shown in the *device* coordinate frame. In the case of Elekta Neuromag®, the optimal device origin is at (0, 13, -6 mm).

**Note:** If the sphere fitted to isotrak points extends outside of the sensor array, for example due to isotrak points that were digitized outside of the head surface or due to bad coordinate transformation, the program reports an error and stops execution.

Following NM20568A Source Modelling Software User's Guide section 5.4: "The MEG sphere model" and NM20419 Mrilab User's Guide section 3.3.3 "Fitting spheres", you can fit a sphere on MRI data and enter the fitted values on the dialog. By using the middle button of the mouse, you can also drag the fitted result from the Mrilab dialog onto the *MaxFilter* origin dialog.

## 3.3 Harmonic basis functions

### 3.3.1 Order selection

*MaxFilter* evaluates the harmonic basis functions for all MEG sensors. The default orders of the harmonic expansions are set to $L_{in} = 8$ and $L_{out} = 3$ which have turned out sufficient in most practical applications (see the publications listed on page 7).

You can change the expansion orders by selecting *Multipole selection...* from the *Parameters* menu (Figure 3.3).

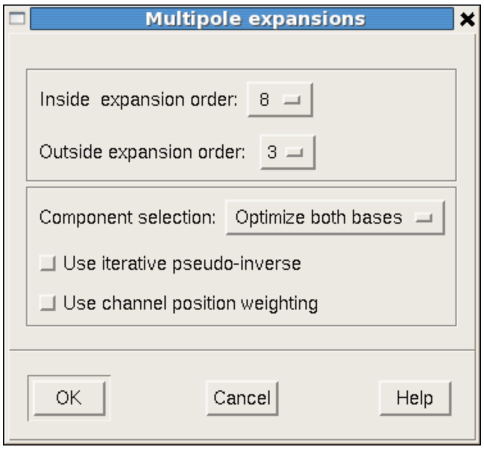


**Figure 3.3** *Setting the expansion orders and optimization of the multipole component selection.*

The total number of multipole components becomes

$$M = (L_{in} + 1)^2 + (L_{out} + 1)^2 - 2,$$

and must not exceed the number of good MEG channels. In practice, the largest values of $L_{in}$ and $L_{out}$ are limited for avoiding numerical instabilities:

$$5 \le L_{in} \le 11, 1 \le L_{out} \le 5.$$

### 3.3.2 Optimization of the SSS bases

The lower part of the dialog controls the component selection options. The program estimates the theoretical signal to noise ratio (SNR) of each multipole component in the inside and outside expansions. To avoid increasing of the noise, virtual channels with smallest SNRs can be neglected (see Appendix A.4 on page 67 for details).

The menu *Component selection* has three choices:

*Use all multipoles*
Switch the component selection off, use all virtual channels.

*Optimize inside basis*
Optimize SNRs of the inside expansion multipole components.

*Optimize both bases*
Optimize SNRs of both inside and outside expansion multipole components (default).

The SNRs of the harmonic components depend on the head position. If the head is in the middle of the helmet, $L_{in} = 8$ and $L_{out} = 3$, the optimal selection typically involves 65 of 80 inside expansion amplitudes and 13 of 15 outside expansion amplitudes.

The dialog has also toggle buttons for two special options:

*Use iterative pseudo-inverse*
The program switches to an iterative method for determining the multipole amplitudes. This option is meant for systems with gradiometer channels only, and it is slower than the normal *direct pseudo-inverse* method (see Appendix A.3).

*Use channel position weighting*
When processing recordings from an empty magnetically shielded room, Maxwell filtering reduces overall interference levels. Gradiometer sensor noises are also decreased but magnetometer sensor noise levels are slightly increased. This option calculates channel-specific Wiener filtering coefficients for the multipole amplitudes which suppress the magnetometer noise level increase (see Appendix A.5).
Note that is option is effective only if the file contains raw data recorded in an empty magnetically shielded room.

## 3.4 Bad channels settings

Successful Maxwell filtering requires that channels with artifacts or very poor data quality need to be excluded from the reconstructions. The channels marked in the bad channel tag of the input FIFF-file or manually marked bad in starting *MaxFilter* are treated as static bad channels, i.e. they are automatically excluded.

You can also use the utility program `/neuro/bin/util/mark_bad_fiff` to mark permanently the channels in the input file that need to be excluded.

If you need to change bad channel detection settings, select *Bad channels...* from the *Parameters* menu (Figure 3.4). You can enter the logical channel numbers separated by space for setting manual bad channels. You can also set the automated bad channel detection parameters.

Independently from bad channels settings, *MaxFilter* always scans for saturated channels. If the signal variation in a 30-ms window (or at least 20 consequtive samples) becomes zero, the channel is ‘quarantined’ for a period of 10 seconds, i.e., it is excluded from all Maxwell filtering calculations.

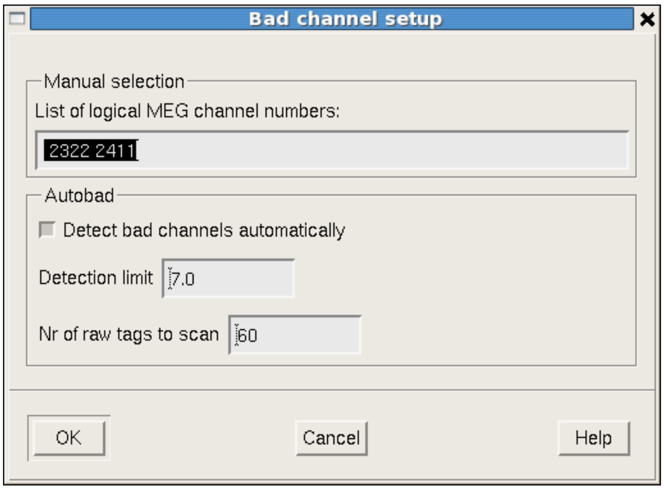


**Figure 3.4** *Setting bad channels.*

### 3.4.1 Autobad

The program can determine automatically if there are MEG channels with spurious sensor artefacts. Bad channel detection is performed by reconstructing the inside and outside expansion signals $\hat{\boldsymbol{b}} = \boldsymbol{b}_{in} + \boldsymbol{b}_{out}$ and by the difference from measured data, $\boldsymbol{b}_s = \boldsymbol{b} - \hat{\boldsymbol{b}}$, which in an ideal case should contain only white noise of the SQUID sensors.

Bad channels typically exhibit large values in $\boldsymbol{b}_s$, which apparently originate in the sensor space $S_T$. The amplitude range is then calculated for each channel as $\boldsymbol{d}_k = \boldsymbol{b}_{s,k}(max) - \boldsymbol{b}_{s,k}(min)$, $k = 1,...,M$. Average and standard deviation values $\boldsymbol{d}_{ave}$, $\boldsymbol{d}_{SD}$ are calculated separately over magnetometer and planar gradiometer channels. A channel is determined bad if $\boldsymbol{d}_k > \boldsymbol{d}_{ave} + r \cdot \boldsymbol{d}_{SD}$. The default threshold value in *MaxFilter* is $r = 7$.

You can enable or disable autobad with the toggle button *Detect bad channels automatically*, and set the threshold value in the *Detection limit* field. The number in *Nr of raw tags to scan* field (here *nraw*=60) means that in the case of raw data the first and last 60 data buffers are scanned (see Section 3.4.3).

**Warning:** If the threshold of the automated bad channel detection is too small, the program may classify good channels as bads, and if it is too high, some bad channels may remain undetected.

### 3.4.2 Evoked data

Each evoked response set is treated separately, i.e., the channel selection may vary from set to set. If the program finds more than 12 bad channels, a warning is printed and the execution terminates (unless you have selected *Force to ignore warnings*). If the fine-calibration is not in use, it may happen that the autobad detection produces too many bad channels for any threshold values. In such cases you should examine the input data to determine the bad channels, and repeat the program by disabling autobad and setting bad channels manually.

### 3.4.3 Raw data

For raw data FIFF files the program scans bad channels from the first and last *nraw* buffers. Artifacts on bad channels are especially clear on low frequencies. To speed up the scanning the program copies the data to a temporary buffer which is filtered to frequency band 0.1...50 Hz and downsampled before determining bad channels. The channels that appear bad in more than five buffers are treated as static bad channels throughout the whole raw data file.

If the default settings are not satisfying, you can:

- Set the bad channels manually, and disable autobad, or
- Set *nraw=1* and select *Force to ignore warnings*; the program tries then to detect bad channels from each raw data buffer separately, or
- Apply *tSSS* (Section 3.5).

### 3.4.4 Sensor artifacts

Sometimes the MEG data may contain saturated channels, or interference that originates in the SQUID sensors or the electronics. Such disturbances are usually manifested in few channels as spurious artifacts in the signal, and the program can detect and discard such artifacts automatically.

If the sensor artifacts are present in a larger number of channels (e.g., due to a strong interference coupled via the electronics), the result may still contain unwanted contributions. In such cases you can apply *tSSS* to reduce the interference (see Section 3.5).

**Note:** If the raw data has segments where there are too many artifact or saturated channels, the program may not be able to do Maxwell filtering properly. You may still be able to process the segments which show acceptable data. The program gives a warning if there are such bad data segments; they are indicated by setting the output data of all channels to zero.

## 3.5 Temporal SSS settings

The temporal extension of SSS, *tSSS*, can be regarded as a four-dimensional filter: besides the three spatial dimensions it also checks the time domain. First, normal spatial Maxwell filtering is applied to the data, typically in blocks of four seconds. The program reconstructs the waveforms $\boldsymbol{b}_{in}(t)$ and $\boldsymbol{b}_{out}(t)$, and subtracts them from the measured data $\boldsymbol{b}(t)$:

$$\boldsymbol{b}_s(t) = \boldsymbol{b}(t) - (\boldsymbol{b}_{in}(t) + \boldsymbol{b}_{out}(t)) \ .$$

If the interference is located very close to the sensor array, residual waveforms exhibit very large disturbances, and remaining disturbance is also evident in $\boldsymbol{b}_{in}(t)$.

The insight of the temporal extension is that if there are similar waveforms in $\boldsymbol{b}_{in}$ and $\boldsymbol{b}_s$, they must be artifacts. Such waveforms can be easily recognized by computing correlations of the temporal subspaces. Correlation values close to 1 indicate intersecting waveforms which should be projected out of the data. Mathematical basis of the temporal subspace projection is described in Appendix C.

**Note:** *tSSS* is recommended to use on raw data files in order to ensure optimal interference suppression.

The *tSSS* method can be applied to all FIFF data files with sufficiently long data for adequate statistics. The program then reports the number of components which are projected out from each data block. If the length of data is shorter than 500 samples, the program reports an error and terminates.

When *tSSS* is used the automated bad channel detection is automatically switched off. The program however detects saturated channels from each buffer separately. Such channels and static bad channels are excluded before estimation of the intersecting waveforms.

**Warning:** If *tSSS* is applied on averaged data or if there were several saturated channels in raw data, the result must be inspected very carefully.

In order to change *tSSS* parameters, select *tSSS parameters...* from the *Parameters* menu (Figure 3.5). You can define the processing buffer length and the correlation limit for the subspace intersection. You can also save intersecting waveforms in a separate FIFF file.

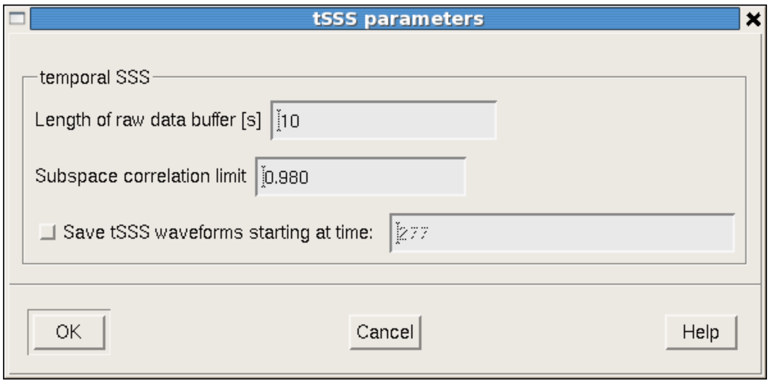


**Figure 3.5** *Setting tSSS parameters.*

The default length of data buffering in *tSSS* is 10 seconds. Offsets and slow-frequency variations are often seen also in the sensor space ($S_T$) signals. Therefore, *tSSS* suppresses DC and very slow frequency components, and acts as a high-pass filter. The cut-off frequency is related to the buffer length; 4-second buffering corresponds to the cut-off frequency of 0.25 Hz. Longer buffers can account for slower background variations.

You can decrease the cut-off frequency by increasing the data buffer length. Long buffers also increase the memory usage. In the case of 306-channel Elekta Neuromag® data sampled with 1 kHz, 4-second buffers typically take about 50 MB of memory, while increasing buffer length to 30 seconds expands the memory usage to about 400 MB.

**Note:** In *MaxFilter* 2.2 the default *tSSS* buffer length was changed from 4 to 10 seconds for more efficient suppression of internal interference.

If the interference suppression seems inadequate, you can try to lower the subspace correlation limit. However, you should be careful if you set a lower limit or if the length of *tSSSS* buffer is shorter than 4 seconds to avoid mixing brain signal components with disturbance waveforms.

**Warning:** The user must judge the result carefully if the *tSSS* correlation limit is lowered from the default value.

Optionally, you can store tSSS waveforms into an evoked format FIFF file (default name *tsss_waves.fif*). In case of raw data, the integer number in the text field *Starting at time* refers to the timepoint (in seconds) to start the waveform writing (see Section 2.6).

For viewing the *tSSS* waveforms in the plotting program, *xplotter*, a suitable layout is provided in the file */neuro/setup/maxfilter/tsss_waves.lout*.

In addition, *MaxFilter* creates an ascii file (e.g. *tsss_waves.fif.cor*) which lists the subspace correlation for each waveform and shows how many of them were projected out. This option is useful when you try to change the correlation limit for optimal interference suppression.

## 3.6 Low-pass filtering and downsampling

You can apply finite impulse response (FIR) low-pass filtering by selecting *LP filter and downsample...* from the *Parameters* menu (Figure 3.6). You can toggle the filter on or off and set the corner frequency.

You can also select a downsampling *factor* if you want to decrease the output file size. FIR filters are employed, and the low-pass frequency is determined as *sfreq / (3*factor)*.

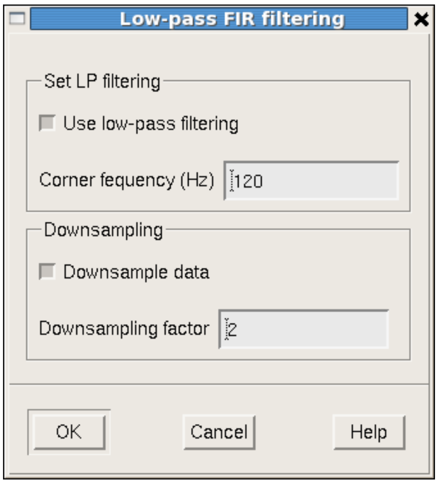


**Figure 3.6** *Controlling low pass filtering and downsampling.*

If head position estimation is performed, *MaxFilter* extracts continuous HPI signals and calculates head positions first (Section 4.4). Thereafter, low-pass filtering and downsampling are applied to data buffers before any Maxwell filtering operations.

## 3.7 MEG sensors

### 3.7.1 Sensor types

Maxwell filtering can process all Elekta Neuromag sensor types using numerical integration over the pickup coils. Details of the sensor types are collected in Appendix B (Table B.1).

The Elekta Neuromag® system combines two types of sensors, and an appropriate scaling between them has to be applied before combining the data for Maxwell filtering. The default scaling factor between magnetometer and gradiometer channels is 100 (see Appendix B.2).

Often sensor noise levels are used in source modelling studies. The noises can be defined as standard deviations from the baseline during a specified time window. In the absence of brain signals (empty room data), the noise values are typically uncorrelated and obey normal distribution.

Maxwell filtering however modifies the sensor noise properties, and the baseline noises may become correlated. Therefore, statistical parameters such as confidence intervals and volumes are incorrect if analysis software uses sensor noises estimated from the baselines.

**Note:** Correlations of the sensor noises must be taken into account if the Maxwell-filtered sensor noises are applied in source modelling.

### 3.7.2 Calibration adjustment and cross-talk correction

Maxwell filtering can be applied to improve the standard calibration of MEG systems. The adjustment includes accurately defined sensor orientations and magnetometer calibration factors, and imbalance correction for the planar gradiometers. In addition, *cross-talk correction* can be applied to reduce mutual interference between overlapping magnetometer and gradiometer loops of a sensor unit.

Currently, these options are available only for 306-channel Elekta Neuromag® systems. Fine-calibration and cross-talk matrix files are prepared and installed by the Elekta Neuromag service personnel.

Fine-calibration adjustment is not performed if the fine-calibration file is not found, or the processing history of *input_file.fif* already includes the fine-calibration. Likewise the cross-talk correction is not done if the cross-talk matrix file is not found, or the processing history of *input_file.fif* already includes the correction.

After opening the FIFF-file the program attempts to load the channel cross-talk correction and fine-calibration data files. The default files are, respectively, `$(NEUROMAG_ROOT)`*/databases/ctc/ct_sparse.fif* and `$(NEUROMAG_ROOT)`*/databases/sss/sss_cal.dat*.

The root directory can be defined via the environmental variable `NEUROMAG_ROOT`. By default, it points to directory */neuro* (or */opt/neuromag*). For other locations, you should set `NEUROMAG_ROOT` to desired directory path before running *MaxFilter.*

If you analyze data only from one Elekta Neuromag® system, the default cross-talk correction and fine-calibration data files installed by the Elekta Neuromag service are sufficient.

**Warning:** If the fine-calibration and cross-talk correction data are not available, the performance of *MaxFilter* may not be as good as with the fine-calibrated system.

### 3.7.3 Changing the fine-calibration and cross-talk correction

You can change fine-calibration and cross-talk correction files by selecting *Fine-calibration...* or *Cross-talk compensation...* from the *Parameters* menu (Figure 3.7).

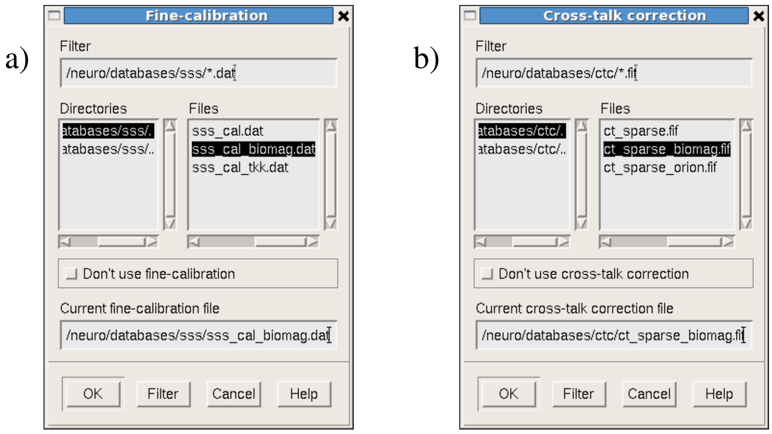


**Figure 3.7** *Selecting a) the calibration adjustment file, b) the cross-talk correction file.*

**Note:** The program reports an error and terminates if the selected files are not found, or if they do not contain appropriate data.

By default, the fine-calibration and cross-talk correction are always attempted if suitable files are found. Sometimes they however need to be switched off, for example with simulated data or evoked FIFF files exported from the *graph* program when the adjustments were already applied in the original raw data file.

The current version of *MaxFilter* does not check automatically if the selected fine-calibration or cross-talk correction files are consistent with the input data file. Therefore, you should be very careful when:

- Changing the cross-talk or fine-calibration correction files from the default ones.
- Processing data that were recorded with other measurement devices.

**Warning:** Special care should be taken to ensure that right fine-calibration data are used for imported or old data for which the default calibration does not apply.

## 3.8 Parallel processing

Both the Linux and HP-UX versions of *MaxFilter* 2.x utilize high-performance mathematical library packages which support parallel processing. If you are using a workstation with more than one processor, the *MaxFilter* GUI automatically requests the number of processors and displays it in the dialog *Parameters -> Parallel processing...* (Figure 3.8).

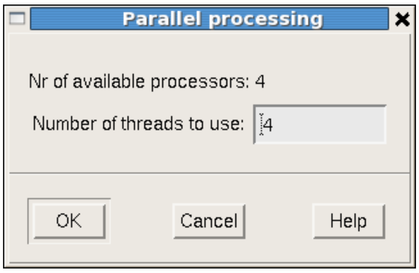


**Figure 3.8** *Setting the number of threads.*

If needed, you can modify the number of parallel threads the program uses. The GUI sets an environment variable to the desired value: `OMP_NUM_THREADS` on Linux or `MLIB_NUMBER_OF_THREADS` on HP-UX.

The run-time priority is automatically set lower by running the program *maxfilter* via the command *nice*.

You can also set the number of threads in advance, e.g., in a 4-processor workstation:

**Linux:** `export OMP_NUM_THREADS=4`
**HP-UX:** `export MLIB_NUMBER_OF_THREADS=4`

If you try to set a larger value than the number of processors in the workstation, the program displays an error message. The maximum number of threads is limited to 4 in HP-UX and to 32 in Linux (see also Section 3.10 on page 41).

## 3.9 Command-line options

When you use the command-line program *MaxFilter* directly, you can define the origin with the option `-origin` *x0 y0 z0* where the coordinates *x0 y0 z0* are given in mm. You also need to define the coordinate frame with the option `-frame` *device* | *head.* If you want to fit the origin to isotrak data, select `-origin` *fit*.

You can change the orders of the harmonic function expansions with the options `-in` $L_{in}$ and `-out` $L_{out}$. Apply the option `-regularize` *in* | *both* | *off* if you want to optimize the component selection for inside or both bases, or apply all virtual channels.

You can set bad channels using the option `-bad` *ch_1 ... ch_n*. Automatic bad channel detection can be set with the option `-autobad` *on* | *off* | *nraw*. Argument *on* indicates that the bad channel detection is done separately for each data block, while argument *off* means that automated detection is not applied. Argument *nraw* scans first *nraw* data buffers. Value *nraw* = 1 is equivalent with `-autobad` *on*. You can also define the threshold with the option `-badlimit` *r*.

To control *tSSS*, you can use the options `-st` [*time*] and `-corr` *limit*. Optional argument *time* gives the starting time in a raw data file to write the waveforms.

The option `-lpfilt` *freq* sets low-pass FIR filtering, where *freq* is the low-pass corner frequency. The default corner frequency value is 120 Hz. You can invoke the downsampling with the option `-ds` *factor*, where *factor* is the downsampling ratio.

Cross-talk correction and file-calibration file options `-ctc` and `-cal` are needed if you need to analyze data that were recorded with different measurement devices. For convenience, *MaxFilter* includes option `-site` *sitename*, which tries to load the files
`$(NEUROMAG_ROOT)`*/databases/ctc/ct_sparse_sitename.fif* and
`$(NEUROMAG_ROOT)`*/databases/sss/sss_cal_sitename.dat*.

You can display the default settings with the option `-def` without any other arguments.

Iterative pseudo-inverse can be selected with the option `-iterate [n|off]` (see Appendix A.3). Special command-line options `-magbad`, `-T2`, `-T3`, `-reconst` and `-mem` are described in Appendix B.

## 3.10 Default parameters

### 3.10.1 Default values

Command-line option `-def` shows the listing of default values:

```
Default parameter values:
         frame = head
         origin: (0 0 40) mm
         in = 8
         out = 3
         autobad 60
         badlimit = 7
         regularize both
         MaxMove not applied
         HPI amp window 200 ms
         HPI amp step 10 ms
         tSSS not applied
         tSSS buflen 10 sec
         tSSS correlation limit 0.980
         Calibration adjustment file =
         /neuro/databases/sss/sss_cal.dat
         Cross-talk correction file =
         /neuro/databases/ctc/ct_sparse.fif
```

### 3.10.2 Customizing default settings

*MaxFilter* 2.2 includes a possibility to adjust some default parameter values. They are contained in file */neuro/setup/maxfilter/maxfilter.defs*:

```
#
# This file contains default values for some parameters.
# Note: items must be blank- or tab-separated (" \t")
#
# expansion origin [x0 y0 z0 in mm] in head coordinates
origin       0  0  40
# automatic fit to isotrak points?
fitorigin    0
# output raw data format (float/short/long)
format       float
# autobad scan length from start and end of raw data files
autobad       60
# tSSS default buffer length in seconds
buflen       10
# HPI amplitude extraction window in ms
hpiwin       200
# HPI amplitude estimation step in ms
hpistep      10
# Fine-calibration adjustment file
calfile      /neuro/databases/sss/sss_cal.dat
# Cross-talk correction file
ctcfile      /neuro/databases/ctc/ct_sparse.fif
# starting directory (GUI)
dir          /neuro/data
# max nr of threads to use per run
maxthreads   32
```

This file is automatically created when *MaxFilter* 2.2 is installed. Only users *neuromag* and *root* can modify the file.

For instance, if the default origin is wanted at location (0, 0, 50) instead of (0, 0, 40), change the origin line to:

```
origin 0 0 50
```

Alternatively, if the origin is always to be fitted to isotrak data (such as in *MaxFilter* software version 2.0), change the fitorigin line to:

```
fitorigin 1
```

Finally, the maximum number of threads per process can be limited by changing the value of `maxthreads`.

4

# CHAPTER 4 MaxMove

## 4.1 Data transformation

This chapter describes the *MaxMove* functionality and controls for the parameters. The performance and limitations of the methods are discussed in more detail in Chapter 6 which also gives brief guidelines for head movement correction.

Direct comparison of different MEG measurements is very difficult, even if the data were acquired with the same device. This applies both to data from the same subject measured several times or data from several subjects. In addition, movements of the patient's or subject's head during the recording cause distortions in the MEG signals.

As described in Section 1.2 and Appendix A.2, *MaxFilter* transforms the MEG data into harmonic function amplitudes which can also be interpreted as virtual channels. *MaxMove* utilizes the virtual channels in estimating the MEG signals corresponding to a different head position or sensor array.

The data transformation of *MaxMove* can be applied in:

- Correction of disturbances due to temporary movements of the patient's or subject's head during continuous recordings.
- Conversion of data acquired during multiple recording sessions into a reference head position.

**Warning:** Head movements generate physiological background signals due to muscle and motor cortex activation. These disturbances are not suppressed by *MaxFilter* and may deteriorate the source localization accuracy.

You need to define the head position estimation, movement correction or data transformation already when loading the input file (see Section 2.5.1 on page 17). The reference head position is also selected then.

The data transformation does not require continuous head position indicator (HPI) signals during recordings. Thus, it can be applied to all Elekta Neuromag systems even if continuous head position tracking was not utilized. You can also apply the mean head position in a file containing previously estimated head positions; see Section 4.5.2.

## 4.2 Initial head position

**Warning:** *MaxMove* operations require that the initial and reference coordinate transformations are defined correctly.

Before running *MaxFilter* it is important to judge the accuracy of the initial head position determination. You can view the coordinate transformation in a FIFF file by typing `show_fiff -vt 222` <filename.fif> (see Section 6.2.2 on page 60 for an example).

Before starting head position estimation, *MaxFilter* evaluates the consistency between fitted isotrak points and initial HPI fitting results. The fitted coil positions are transformed to head coordinates, and the distances between them and the isotrak points are calculated. If the average distance of accepted coil fits is smaller (larger) than 3 mm, the program reports that "HPI consistency of isotrak and hpifit is OK (poor)".

If a coil fit was rejected by the hpifit program, *MaxFilter* gives a warning about mismatch between the digitized and fitted coil positions.

## 4.3 Continuous head position tracking

During the recording, the head position tracking has to be enabled by feeding continuous sinusoidal signals to 4-5 head position indicator (HPI) coils (see Neuromag Data Acquisition User's manual section "Head position indicator").

Head position estimation can be done only if the continuous HPI was applied during the recording. Old hardware (e.g. Neuromag 122) may however not support the continuous HPI.

If the input file does not contain previously estimated head position parameter channels, the program estimates and subtracts sinusoidal signals of the HPI coils. Head positions are fitted from the estimated HPI amplitudes and saved as new quaternion parameter channels. Details of these operations are presented in Appendix D.

### 4.3.1 HPI parameters

When you want to set parameters for estimating the head positions, select *Head position estimation...* from the *Parameters* menu (Figure 4.1).

The selection *Adjust initial hpifit consistency* tries to reduce the initial mismatch. If the HPI coil fitting was accepted during data acquisition, the isotrak point of the mismatching coil is adjusted to the fitted location. This option is useful, for example, if the coil digitization was erroneous or the coil moved after digitization.

If you wish to save the head position parameters in a separate ascii file, choose *Save head position parameters in a separate file* (the format is shown in Appendix D.4).

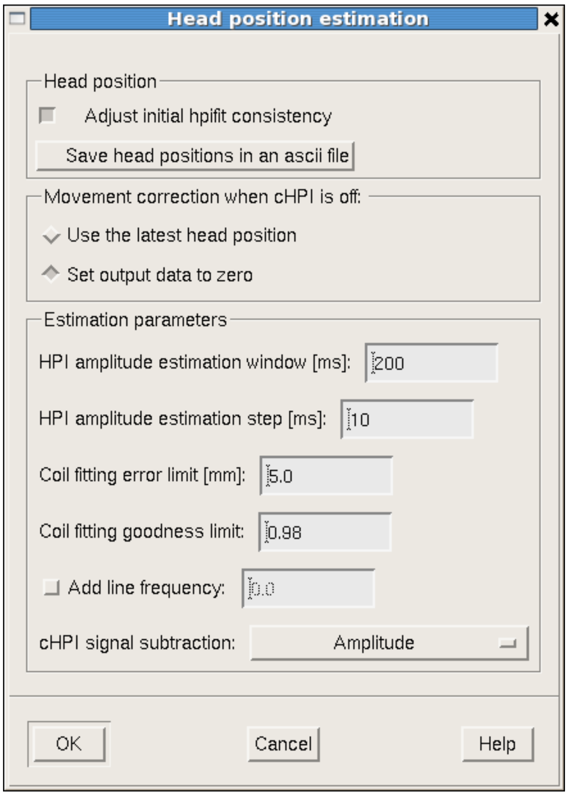


**Figure 4.1** *Setting head position estimation parameters.*

The frame *Estimation parameters* includes controls for the parameters :

*HPI amplitude estimation window*
: The length of the buffer (in ms) to model sinusoidal terms, default 200 ms.

*HPI amplitude estimation step*:
: The step (in ms) to construct HPI amplitudes, default 10 ms.

*Coil fitting error limit*
: Accuracy limit for fitted HPI coil positions, default 5 mm.

*Coil fitting goodness limit*
: Goodness of fit limit for HPI coil position fit, default 0.98.

*Add line frequency*
: Include line frequency and harmonics. These terms are automatically included if the tag FIFF_LINE_FREQ has been written in the raw data file.

*cHPI signal subtraction:*
: Control how the estimated continuous HPI signals are subtracted.

In the last case, the choices are:

*Amplitude*
: When movement correction is performed, the program subtracts the estimated cHPI signals from raw data before SSS operations.

*Amplitude + line frequencies*
: Subtract also the line frequency terms; the subtraction affects the line frequency and its harmonics similarly to a notch filter.

*No cHPI subtraction*
: Do not subtract continuous HPI nor line frequency and its harmonics.

**Note:** Subtraction of estimated cHPI signals suppresses them significantly. The remaining cHPI signals can be removed only by applying low-pass filtering.

### 4.3.2 Intermittent HPI

The selection *Movement correction when cHPI is off* defines what to do in the data segments where continuous HPI was not active:

*Use the latest position*
: Sometimes the continuous HPI may be needed only periodically, e.g. for few seconds every 1 or 2 minutes to check the head position during a long recording. This option transforms the data into the last defined head position.

*Set output data to zero*:
: No SSS operations, output data are set to zero when continuous HPI was not active.

Note that this selection is activated only if you have selected the Maxwell filtering task *Movement correction* when loading the input file.

## 4.4 Head position estimation

**Note:** *MaxFilter* assumes that the normal HPI fitting is done successfully during the data acquisition. The program reports an error and stops head position estimation if it does not find the HPI result from the input file.

The program estimates first the HPI signal amplitudes (see Appendix D.1 for details). According to the default settings, the HPI signal amplitudes are reconstructed 100 times per second.

Next the program estimates the location parameters (*x, y, z*) of each HPI coil with non-linear Simplex minimization. Time steps for fitting new coil positions are determined dynamically from the extracted HPI signal amplitudes (see Appendix D.1). The success of HPI fitting is judged in terms of

**Goodness of fit, g-value:**
Measures the match of measured and modelled HPI amplitude data, ranges between 0 and 1.

**Estimation error**:
The distance of the isotrak point and the fitted point when fitted HPI coil positions are transformed from the device to the head frame.

HPI fitting is considered successful if the g-value exceeds the goodness limit and error is less than the error limit for at least three HPI coils. In such cases the program reports that the fit was OK. HPI fit fails if the acceptance criteria are met with less than three HPI coils. In such cases the coordinate transformation cannot be defined, and the program gives a warning.

The program subtracts the continuous HPI signals before Maxwell filtering. Matching of the fitted and digitized HPI coil positions is performed using *quaternion* parameters, resulting in a coordinate transformation from the device to the head frame. The quaternion parameters are saved as extra channels in the result FIFF file. Details of the quaternion matching are presented in Appendix D.2.

## 4.5 Viewing the head positions

The parameters and results of the head position estimation can be viewed in the main display during processing a datafile. You can choose the main display view (Figure 2.2) from menu *Display* selections:

*Current settings...*
Show the current setting values, Figure 2.2.
*Fit parameters...*
Show head position estimation parameters, Figure 4.2.
*Head positions...*
Show the estimated positions, Figure 4.3.
*Viewing scales...*
Set scalings for the displays, Figure 4.4.

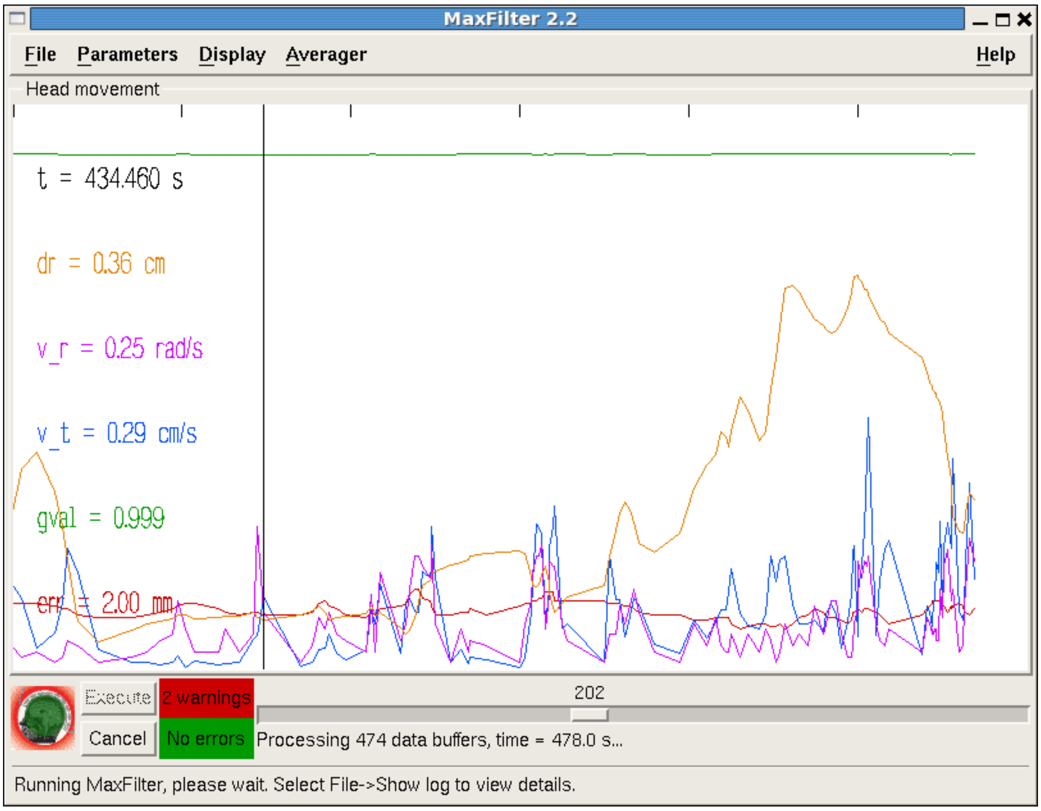


**Figure 4.2** *Head position fitting parameters: fitting error* [*mm*] *(red), goodness of fit (green), translational velocity* [*cm/s*] *(blue), rotation velocity* [*rad/s*] *(magenta), and drifting from the initial position* [*cm*] *(orange).*

The velocities are smoothed by averaging the fitted positions within 200 ms windows.

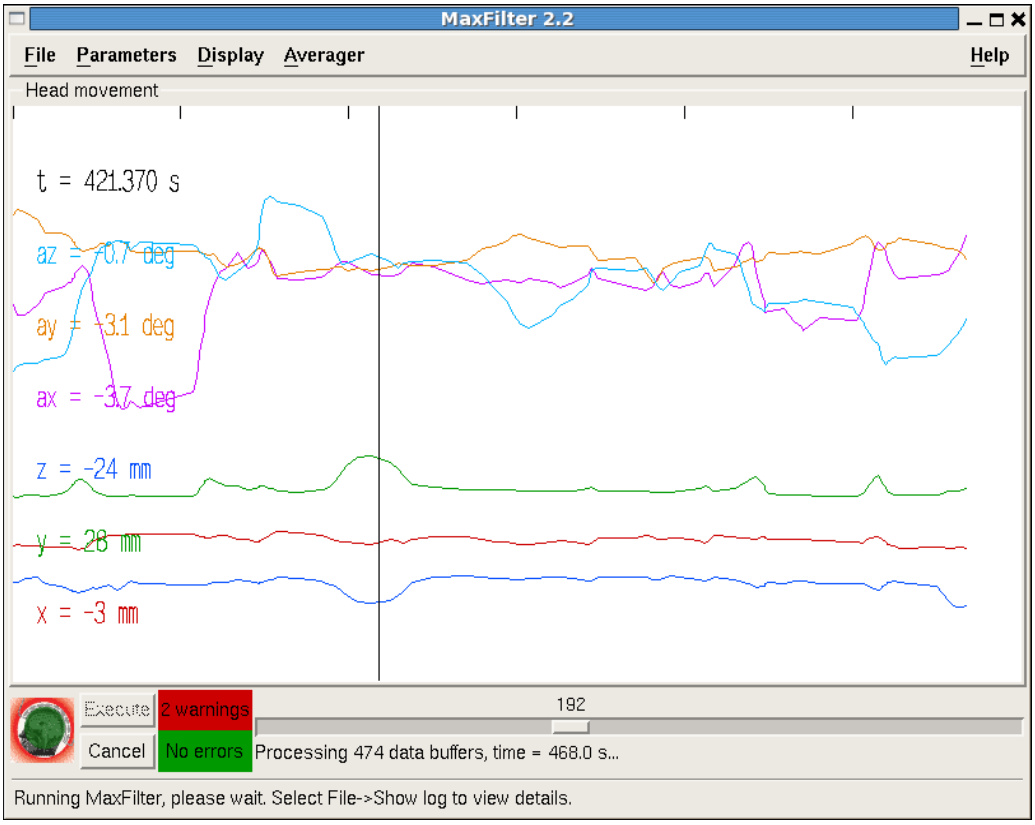


**Figure 4.3** *Fitted head positions: x-coordinate (red), y-coordinate (green), z-coordinate (blue), rotation angles around the x-axis (magenta), y-axis (orange), and z-axis (light blue).*

The *x*, *y* and *z* coordinates $[mm]$ represent positions of the SSS expansion origin (*r0*) in respect to the device coordinate point (0, 0, 0) and *ax, ay, az* values represent rotation angles $[deg]$ of the head coordinate system axes (see Figure 3.1 on page 25):

| | |
|---|---|
| *x > 0:* | *r0* is to the subject's right from the device origin, |
| *y > 0:* | *r0* is anteriorly from the device origin, |
| *z > 0:* | *r0* is upward from the device origin, |
| *ax > 0:* | subject's *y*-axis has rotated counterclockwise around the *x*-axis, i.e., the nose points upwards, |
| *ay > 0:* | subject's *x*-axis has rotated counterclockwise around the *y*-axis, i.e., the head is tilted to the right, |
| *az > 0:* | subject's *xy*-plane has rotated counterclockwise around the *z*-axis, i.e., the head is rotated to the left. |

You can define scaling of the curves by selecting *Viewing scales...* from the *Display* menu. If you are viewing *Fit parameters*, the scaling dialog (Figure 4.4 left) has controls to adjust the visibility and scaling of the five curves in Figure 4.2. When you view *Head positions*, the scaling dialog (Figure 4.4 right) has controls to set the ranges for position (three lower curves) and angle parameters (three upper curves) in Figure 4.3.

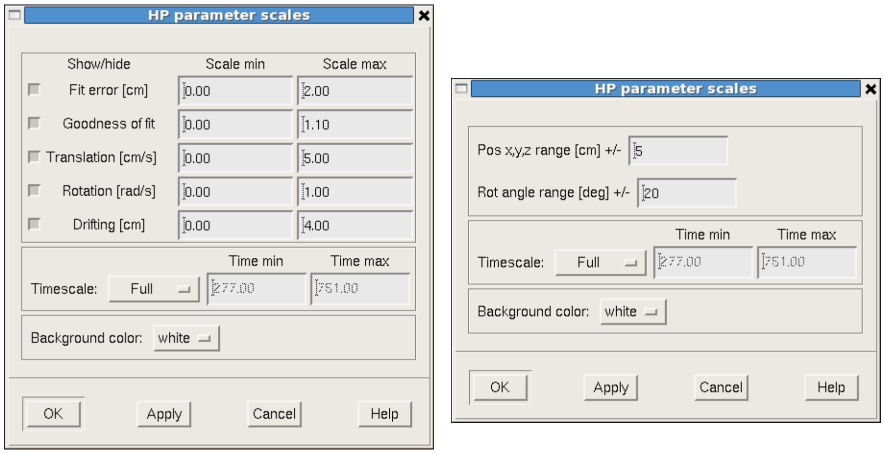


**Figure 4.4** *Scaling controls for the fit parameter (left) and head positions (right) displays.*

Both dialogs have also controls to set the time scale and background color (black or white). The time scale has the choices:

| | |
|---|---|
| *Full* | Show the full time scale in the file (see Section 2.6). |
| *Last 60 s* | Automatically adjust the time scale during processing to show the latest 60 seconds. |
| *Last 20 s* | Show only the latest 20 seconds. |
| *Edit* | Set manually *Time min* and *Time max* values (in seconds). |

### 4.5.1 Exporting graphics

The head positions display has a drag-and-drop capability: the graphics can be transferred to *Cliplab*, the clipboard application from the *Graphics Clipboard* icon. The transfer is accomplished by pressing the middle mouse button down on the curve display and by dragging the pointer on top of one of the *Cliplab* viewports.

You can also make hardcopies of the curves by pressing the right mouse button on the curve display. A popup menu is shown, including the choices:

**Print (bw)**
Print the display on a monochrome Postscript printer through the spooling system.

**Print (color)**
Print the display on a color Postscript printer through the spooling system.

**Illustrator output**

Produce an Encapsulated Postscript file compatible with Adobe Illustrator, version 5.0, or later. You will be prompted for the name of the file to hold the output using a file selection dialog.

**TIFF output**

Produce a TIFF (Tagged Image File Format) file. The image will have 1024 by 1024 pixels. The background of the image will be white regardless of the color of the background on the screen. The filename is entered in the same way as for the Illustrator output.

### 4.5.2 Transformation to mean head position

It is also possible to load previously estimated quaternion channels for viewing by selecting *Load head positions...* from the *Display* menu. The file selection dialog has a button *Save mean head position transformation*. By clicking it when loading a file, you can save the coordinate transformation corresponding to the mean position in a FIFF file.

If you want to convert raw data to the mean position, you should first process the file using the Maxwell filtering task *Head position estimation* (see *Section 2.5.1*). Thereafter, you can load the result file containing the quaternion channels and select the *Movement correction* task. Then load the file containing the mean transformation as the reference head position (see Section 2.5.1) before pressing `Execute`. Note that you also need to select *Force to ignore warnings.*

## 4.6 Movement correction

When the task *Movement correction* is selected, the program searches first if the quaternion data channels are already included in the input file (i.e., if the head positions were estimated earlier). If they are not found, the head positions are estimated as presented above. Then *MaxFilter* performs the normal SSS or tSSS software shielding and thereafter transforms the data to a static reference head position.

When processing a raw data buffer, the program  performs the data transformation only if the HPI fitting was successful. If the HPI fit failed, a warning is given and the output data during failed HPI fits shows zero values in all MEG channels.

The program recognizes from each raw data block if the continuous HPI is on or off. In the latter case, the program reports that continous HPI was off and the output data block either shows zero values on all MEG channels or transforms the data to the latest head position (see Section 4.3.2).

Often head movements may induce other disturbances, such as slow-frequency contributions due to magnetized material on the head or in the EEG cables. Disturbances on MEG channels are especially evident during rapid head movements. Therefore, they may deteriorate the SSS software shielding. In such cases you may need the capabilities of *tSSS* together with movement correction.

**Warning:** Head position calculation errors affect the data quality after movement compensation. The user must inspect the head position fitting error and goodness before data analysis.

## 4.7 Command-line options

On command line you can select coordinate transformations with the option `-trans` *name*, where *name* is the FIFF file defining the coordinate transformation of the reference head position, or *name* = *default*.

You can select the option `-headpos` to estimate the head positions without movement correction. You can set the HPI amplitude estimation window length and amplitude estimation step with the options `-hpiwin` *win* and `-hpistep` *n* where *win* and *n* are given in milliseconds. The line frequency and its harmonics are included with the option `-linefreq` *lf*.

Select the option `-movecomp` to perform the movement correction. If the transformation is not specified, the program uses the transformation stored in the original file when acquiring the data.

Option `-movecomp` *inter* does head position estimation and movement correction during the periods where continuous HPI is on. Instead of skipping data, the program assumes that the head position 'freezes' when the HPI signals are switched off, and performes Maxwell filtering using the latest head position.

Options `-hpie` *errlimit* and `-hpig` *glimit* can be used to change the HPI coil fitting acceptance limits.

Option `-hpicons` tries to reduce the mismatch between initial HPI coil fitting and digitization. Option `-hpsubt` *amp* | *line* | *off* sets the continuous HPI signal subtraction approach.

You can save the estimated head positions in a separate ascii file with the option `-hp` *filename*.

5

# CHAPTER 5 MaxAve

## 5.1 Off-line averaging

The program *MaxAve* is an off-line version of the on-line averager (see Neuromag Data Acquisition User's manual section "On-line averaging"). *MaxAve* can be applied to raw data files either before or after Maxwell filtering. The program reads the acquisition parameters from the input FIFF-file and repeats averaging in off-line mode.

*MaxAve* can be started from the GUI or from a terminal window command-line. The GUI has also controls for setting and changing the rejection limits. Other parameter values set during data acquisition can be changed by setting and editing a parameter file.

## 5.2 Launching the program

You can access *MaxAve* by starting the GUI as explained in Section 2.2 on page 12. You can select the input and output filenames and set rejection limits from the menu *Averager:*

**Load raw data...**
Select the raw data file to be averaged.

**Output file**...
Select the file for saving the averages.

**Rejection limits**...
Change averaging rejection limits.

When you have selected the input filename, the GUI main display appearance changes according to Figure 5.1. Instead of showing *MaxFilter* parameter values and head position parameter drawing area, the dialog is reserved for showing the progress of averaging.

After you have defined the output filename and optionally modified the parameters, press `Execute` to start averaging. The scale bar indicates the number of processed data buffers. The text area shows how many epochs were found and were they added or rejected. You can press `Cancel` to cancel averaging if the program rejects too many (or too few) epochs. Then you can modify the rejection limits and try again (press `Execute`).

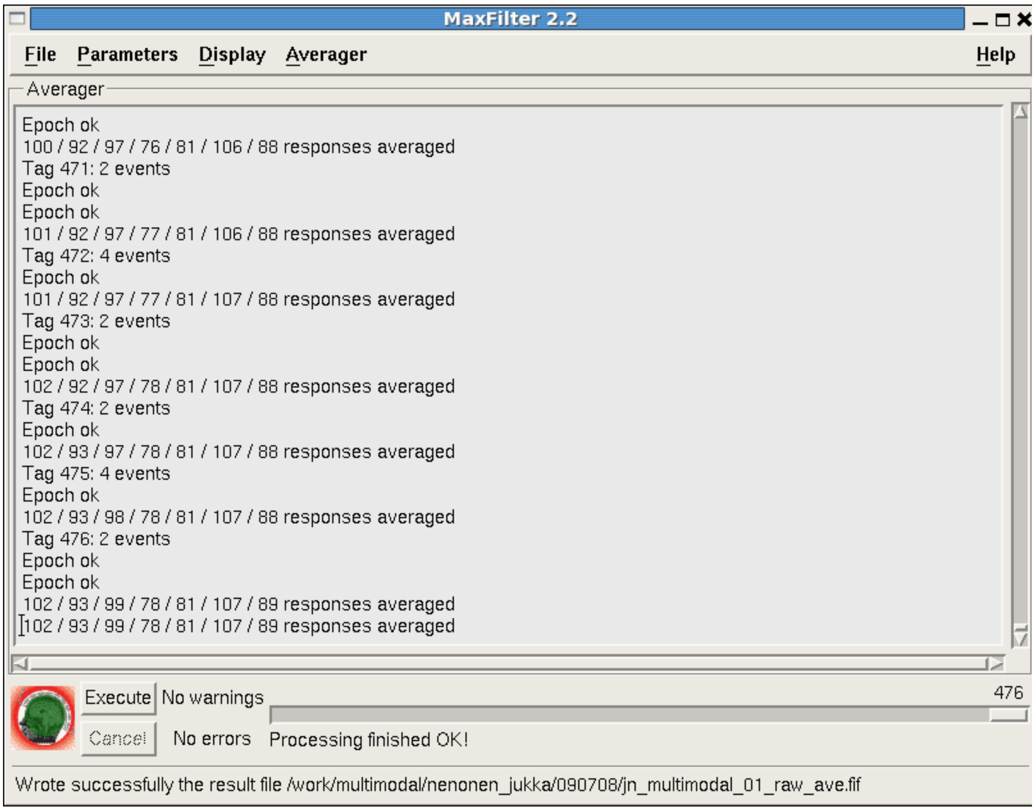


**Figure 5.1** *Averaging dialog.*

You can view more detailed information of the *MaxAve* output by selecting *Show log...* from the *File* menu.

On command line, you can run the program as

```
/neuro/bin/util/maxave [-v] -i input_file.fif
 -o output_file.fif [-p par_file.dat]
```

where:

`-v`
: Switches on verbose logging.

`-i` *input_file.fif*
: Defines the FIFF-file where the raw data are read.

`-o` *output_file.fif*
: Defines the FIFF-file where the averaged results are written.

`-p` *par_file.dat*
: Optional ascii-file where the averaging parameters are read.

## 5.3 Selecting files

*MaxAve* can be applied to FIFF files containing raw data. Off-line averaging can however be performed only if the on-line averaging parameters were set during data acquisition. Otherwise, the program reports an error and exits. New data are loaded by selecting *Load raw data...* from the *Averager* menu (Figure 5.2). Besides file selection, the dialog has a control button for optional saving of the averaging parameters.

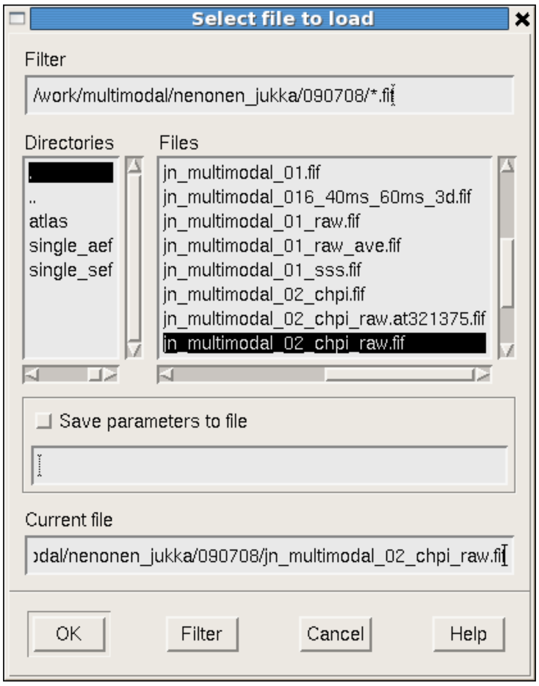


**Figure 5.2** *Data selection for averaging.*

You can set the name of the output file by selecting *Output file...* from the *Averager* menu. If you do not specify an output filename, the program tries to save the results in the file named as *input_file_ave.fif,* where *input_file.fif* is the name of the raw data file.

## 5.4 Averaging parameters

Averaging variables are defined in the file `/neuro/setup/maxave/maxave.vars`. Normally, the program extracts the parameters from the input FIFF-file and saves them in a temporary file. Averaging is performed with these parameter values, and the temporary parameter file is automatically cleaned after saving the averaged data.

If you press the button *Save parameters to file,* you can enter a filename (e.g. *ave.par*) where the program writes in ascii format all data acquisiton parameters extracted from the input file. You can then open and edit this file in a text editor, and save it before running *MaxAve*.

Thereafter you can apply the changed parameter values using

**GUI** The program automatically loads the file and reads the modified values before execution.

**Command-line**
Apply option `-par` *ave.par*.

## 5.5 Artefact rejection

If you need to change the rejection limits, you can select *Rejection limits...* from the *Averager* menu instead of saving the parameter file (Figure 5.3).

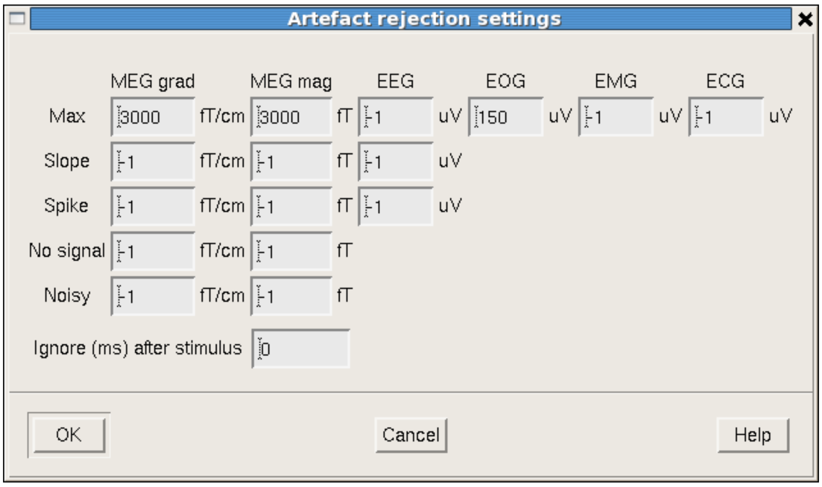


**Figure 5.3** *Setting averaging rejection limits.*

There are three kinds of artefact rejection criteria available for MEG and EEG channels. Each of these can be turned off by entering a negative number into their control field.

**Amplitude (Max)**
The peak-to-peak amplitude within an epoch must not exceed this value. The recommended value is 3000 fT/cm for gradiometers and 3000 fT for magnetometers.

**Slope**
The epoch is subdivided into four equally long pieces. The averages over each piece is calculated. None of the three differences between subsequent partial averages must not exceed this value. By default, this criterion is not in use.

**Spike**
The absolute value of the difference between any sampled value in an epoch and the average of 20 previous values must not be larger than this. By default, this criterion is not in use.

The above criteria are applied to all MEG and EEG channels. If a channel not meeting the criteria throughout the epoch is found, the epoch is rejected. Channels marked bad in the file are excluded from the artefact rejection.

EOG, EMG, and ECG rejection is only possible on the basis of the amplitude criterion.

In addition to the above criteria, an epoch up to a given time interval after the reference event can be excluded from the rejection (*Ignore (ms) after stim*). This is useful if a strong stimulus artefact is expected.

The above artefact criteria are applied in a transient manner. Every epoch is checked against the above criteria to see whether the conditions can be met. However, it sometimes happens that few MEG channels are either showing no signal or are very noisy. There are two additional parameters to check these channels: **MEG no signal** and **MEG noisy**.

If the peak-to-peak amplitude of a channel is less than the **MEG no signal** limit or larger than the **MEG noisy** limit, the channel will be omitted from further artefact checking during the current and all subsequent epochs. The noisy and silent channels so detected will be marked ‘bad’ in the resulting evoked-response data files.

The signals of “noisy” or “silent” channels are stored in the very same way as those of any other channels; the automatic detection only affects rejection checking.

**Note:** Data acquisition software release 3.3 or earlier do not contain sufficient parameter settings to separate gradiometer and magnetometer channels. Thus, *MaxAve* may reject all epochs due to large amplitudes on magnetometer channels. In such cases you need to set the rejection limits in the dialog (Figure 5.3) before running the averager.

6

# CHAPTER 6 Performance and limitations

*MaxFilter* is easy to apply and the default settings provide good results in most cases. There are however some general limitations that the user should bear in mind when applying the program.

**Warning:** It is important that the user inspects both the input and the output data visually to judge the quality of the *MaxFilter* result.

## 6.1 Clinical validation study

*MaxFilter* software version 2.1 was evaluated by processing evoked response data measured from twenty healthy adult and ten child volunteers. The study was designed to assess the effectiveness of *tSSS* and *MaxMove* by comparing the data processed with them against reference data.

Each subject was recorded four times: (1) A reference recording with a stationary head position. (2) The subject was asked to change the head position twice during a continuous recording. (3) The subject was asked to move the head all the time during a continuous recording. (4) Two magnetized pieces were attached to the scalp and the subject was asked to move the head and mouth.

During recordings, interleaved auditory and somatosensory (tactile) stimuli were delivered to the subject. The reference recordings were processed with the basic SSS, while all other recordings were processed with *MaxMove* and *tSSS*. The data were also converted into a reference head position. Finally, evoked responses were selectively averaged with *MaxAve*.

Neuronal activity was modelled with equivalent current dipoles whose locations and strengths were fitted from the somatosensory and auditory evoked fields (SEF and AEF, respectively). Dipole localizations, amplitudes and latencies fitted from the data processed with *MaxMove* and *tSSS* were compared to those obtained from the reference responses.

The following conclusions were made according to the analysis results:

- The responses from the reference recordings were unaffected by *tSSS* processing.
- When the head moved slowly and randomly, average dipole localization difference of both SEF and AEF was below 10 mm. Response waveforms and amplitudes were equivalent with the reference responses.
- In all head movement recordings, *MaxMove* could track the head positions reliably and *tSSS* suppressed interference caused by magnetized impurities.
- Continuous and regular movements caused rhythmic activation in the motor cortex and muscles moving the head. This physiological background activation affected the responses. Consequently, dipole localization differences from the AEFs were larger than 10 mm and amplitudes were 40% smaller than the references. The SEF results were, however, equivalent with the reference results.

## 6.2 Limitations of MaxFilter

### 6.2.1 Saturated MEG channels

Any magnetized objects in/on the head of the subject cause large low-frequency disturbances in the MEG signals when the head moves, but *tSSS* can generally compensate such artifacts well. However, the interference may occasionally become strong enough to exceed the dynamic range of some MEG sensors. Typically, the sensors recover within seconds from such saturations, but these channels must be excluded from Maxwell filtering (see Section 3.4.4).

If the number of remaining good MEG channels becomes smaller than the number of SSS expansion terms, Maxwell filtering cannot be performed any more. In practice, however, saturation of several channels can reduce the data quality of *MaxFilter*. The user should inspect the *MaxFilter* logging to find out if there were data segments with a large number of bad channels, and view the resulting data to judge its quality.

### 6.2.2 Bad coordinate transformation

The success of *MaxMove* operations depends on how accurately the reference head position was determined. If there is a large error in the initially determined head position, the program may set the SSS expansion origin too close to some sensors and, correspondingly, too far from some other sensors. The program stops execution if the origin is placed outside of the sensor array or closer than 5 cm to nearest sensor (Section 3.2.2).

Otherwise, a bad coordinate transformation causes too large head position translations and some MEG channels show drastically increased noise (see the next section).

When you start the head position estimation, the program reports if the initial HPI fitting accuracy was OK or poor. In the latter case, you should examine the head position carefully. You can use

```
/neuro/bin/util/show_fiff -vt 222 <filename.fif>
```

to show the coordinate transformation between the device and head coordinates. The output looks like:

```
222 = transfrom device->head
0.999432 -0.014128 -0.030580
0.018061  0.991028  0.13426
0.028435 -0.132903  0.990721
-2.800296 -1.600493 39.493006
```

The first three rows indicate the rotation matrix *R* and the last row the translation vector *T* in the coordinate transformation (see Appendix D.2 on page 77). The diagonal elements of *R* are close to 1 when the head is in a normal position. The translation vector indicates the position of the device coordinate origin in the head coordinate frame (see also Section 3.2.2) . Often *T* is fairly close to (0, 0, 40 mm) and therefore, this point is also set as the default origin value (Section 3.2.1).

The initially determined head position may be improper if the rotation *R* deviates significantly from a unity matrix, or the translation *T* is far from the device coordinate origin.

To check the initial position,  you can display a standard head or boundary-element meshes inside the sensor helmet by loading an averaged file on the Source Modelling program, Xfit version 5.5 or later (see NM20568A Source Modelling Software User's Guide section 5.1). You can also view more information about the initial HPI coil fitting status (i.e., errors, goodness of fit and acceptance).

### 6.2.3 Transformation distance

Due to the multipole component selection (Section 3.3.2) *MaxFilter* performance is quite robust to differing head positions. However, sometimes the subject or patient may keep the head in too low or frontal position and some parts of the brain remain far from the sensors. In such cases measured signals may be small and all multipole components cannot have good signal to noise ratio due to large distance between the brain and the MEG sensors. If the data are transformed to a reference position closer to the sensors, such poorly detected components will be amplified to the strength they would have in the target head position.  As a result, some

MEG channels show significantly increased noise. Typically, this situation occurs in topmost MEG channels when the true head position was low and the transformation brings the position upwards toward the sensors. Generally, such noise increase starts to appear when the head position changes more than 25 mm.

**Note:** Transforming the head position more than 25 mm shows increased noise on some MEG channels.

### 6.2.4 Physiological background activation

Sometimes the subject or patient can generate strong background activation which complicates the analysis of other brain signals. For example, continuous regular movements generate rhythmic activation in the muscles moving the head and in the motor cortex. Also, if the subject or patient is drowsy or has eyes closed, the recording displays alpha-waves typically in the sensors above occipital cortex.

Amplitudes of background oscillations can be stronger than the interesting brain signals, and disturbing components of the background activity may be prominent even after signal averaging. *MaxFilter* cannot suppress the rhythmic background activation signals because they originate in the brain in a manner similar to other brain signals. For the same reason, *MaxFilter* cannot suppress disturbance signals caused by head muscles or eye movements.

### 6.2.5 Limitations of tSSS

The *tSSS* method is designed to identify and suppress artifact waveforms optimally when the signal-to-noise ratio (SNR) is low. However, in some exceptional cases with high SNR the SSS reconstruction may result in leakage from the internal signal space into the sensor space ($S_{in}$ and $S_T$ in Figure 1.1 on page 6). If the source strenght is over 500 nAm and the source location in very superficial, more than 65 mm from the expansion origin, *tSSS* may diminish such brain signals.

**Warning:** Spatiotemporal signal space separation (*tSSS*) may diminish brain signals arising from very strong, superficial sources.

## 6.3 Guidelines for head movement correction

The common practice in MEG recordings is to instruct the patient or subject to keep the head still (see NM20215A Elekta Neuromag® System Hardware User's Manual). In cases where he/she cannot maintain a static head position, *MaxFilter* provides tools to detect the head position changes and to correct disturbances caused by temporary movements.

Continuous head movements, however, can cause strong background signals which affect the brain signals of interest. Therefore, continuous head movements may limit the performance of head movement correction. This limitation is not cause by the software, but it is inherent to the physiological changes related to head motion.

**Warning:** Head movements generate physiological background signals due to muscle and motor cortex activation. These disturbances are not suppressed by *MaxFilter* and may deteriorate the source localization accuracy.

**Warning:** *MaxFilter* cannot suppress movement artifacts in data segments with rapid head movements, such as those arising during motor seizures. Movement artifacts may deteriorate the localization accuracy of the system.

A

# APPENDIX A Maxwell filtering in a nutshell

## A.1 Signal space separation

MEG devices comprised of more than 300 signal channels provide generous oversampling of both biomagnetic and external disturbance magnetic fields. Because the sensor array is located in a source-free volume between the volume of interest (inside of the helmet) and the volume containing all sources of external interference (outside of the helmet), it turns out that the magnetic signal space can be split into two separate, linearly independent subspaces:

$$\boldsymbol{B}(\boldsymbol{r}) = -\mu_o \sum_{n=1}^{\infty} \sum_{m=-n}^{n} \alpha_{nm} \frac{\nu_{nm}(\theta, \varphi)}{r^{n+2}} - \mu_o \sum_{n=1}^{\infty} \sum_{m=-n}^{n} \beta_{nm} r^{n-1} \omega_{nm}(\theta, \varphi) .$$

The first sum (amplitude coefficients $\alpha_{nm}$) represents signals of interest emanating from the head surrounded by the sensors, and the second sum (amplitude coefficients $\beta_{nm}$) represents signals from sources outside of the array. As the former volume contains the biomagnetic signal sources and the latter volume contains the external disturbance sources, any measured signal can be uniquely decomposed into two magnetic subspaces with separate coefficients ($\alpha_{nm}$, $\beta_{nm}$), corresponding to the subspace spanning the biomagnetic signals and to the subspace spanning the external disturbance signals. The basis functions $\nu_{nm}$, $\omega_{nm}$ are expressed in terms of the vector spherical harmonic (VSH) functions:

$$\nu_{nm}(\theta, \varphi) = -(n+1) Y_{nm} \boldsymbol{e}_r + \frac{\partial Y_{nm}}{\partial \theta} \boldsymbol{e}_\theta + \frac{im Y_{nm}}{\sin\theta} \boldsymbol{e}_\varphi ,$$
$$\omega_{nm}(\theta, \varphi) = n Y_{nm} \boldsymbol{e}_r + \frac{\partial Y_{nm}}{\partial \theta} \boldsymbol{e}_\theta + \frac{im Y_{nm}}{\sin\theta} \boldsymbol{e}_\varphi ,$$

where $Y_{nm}$ are the ordinary spherical harmonic functions, and $i$ is the imaginary unit.

Maxwell filtering is a process where 1) the expansions are terminated to the limit where the spatial frequencies of the *n,m* components become too high (such components are buried in the sensor noise), and 2) the expansion of $\beta_{nm}$ corresponding to external magnetic interference is omitted after estimating both $\alpha_{nm}$ and $\beta_{nm}$.

## A.2 Harmonic amplitudes

MEG signals can be expressed in matrix form as

$$b = Sx = \begin{bmatrix} S_{in} & S_{out} \end{bmatrix} \begin{bmatrix} x_{in} \\ x_{out} \end{bmatrix},$$

where

$$\begin{array}{ll} S_{in} = \begin{bmatrix} \nu_{1,-1} & \dots & \nu_{N,N} \end{bmatrix} & x_{in} = \begin{bmatrix} \alpha_{1,-1} & \dots & \alpha_{N,N} \end{bmatrix}^T \\ S_{out} = \begin{bmatrix} \omega_{1,-1} & \dots & \omega_{M,M} \end{bmatrix} & x_{out} = \begin{bmatrix} \beta_{1,-1} & \dots & \beta_{M,M} \end{bmatrix}^T \end{array}.$$

Here the coefficients $\nu_{nm}$ and $\omega_{nm}$ represent the harmonic basis function values, $N = L_{in}$, $M = L_{out}$, and the vectors $x_{in}, x_{out}$ contain the harmonic amplitudes.

The amplitudes $x_{in}, x_{out}$ can be estimated from measured signals $b$ as

$$x = \begin{bmatrix} x_{in} \\ x_{out} \end{bmatrix} = S^{\dagger} b,$$

where $S^{\dagger}$ is the pseudoinverse of $S$ (see the next Section). The signals $b$ can then be separated as $b = b_{in} + b_{out}$ where

$$\begin{array}{l} b_{in} = S_{in} x_{in} \\ b_{out} = S_{out} x_{out} \end{array}.$$

Suppression of external interference can be performed by leaving out the contribution $b_{out}$.

The harmonic amplitudes $\alpha_{nm}$ of the inside expansion can be interpreted as *virtual* channels which are independent of the sensor array and head position. Optimization of the virtual channel selection is described in Section A.4.

## A.3 Pseudoinverse

The pseudoinverse $S^{\dagger}$ can be calculated with standard numerical procedures:

$$S^{\dagger} = (S^T S)^{-1} S^T .$$

The condition number of $S^{\dagger}$ describes the stability. Typically, the condition number for the Elekta Neuromag® sensor array is below 500. It may however increase dramatically, e.g., if the magnetometer channels are omitted. Therefore, we have developed an iterative method to improve the stability. Instead off all expansion terms together, the iterative pseudoinverse is composed of separate block matrices

$$S_k^{\dagger} = (S_k^T S_k)^{-1} S_k^T ,$$

where subscript *k* indicates that the submatrix contains only the expansion terms of the *k*th order, $k = 1 \ldots max(L_{in}, L_{out})$.

Estimation of the moments $x_{in}, x_{out}$ is then performed iteratively. On each iteration round, all expansion orders are processed one by one by setting the moments of the *k*th order to zero, and by estimating the moments of other orders from the sub-blocks of the pseudo-inverse matrix. The iteration converges typically in less than 10 rounds. The iterative method is always applied as default to 122- and 204-channel gradiometer system data (Neuromag System, Neuromag-122).

## A.4 Optimization of virtual channel selection

The celebrated Shannon's theory of information transmission can be applied in MEG as well. A single magnetometer can be regarded as a noisy channel conveying information from the sources in the brain. Its output, $b(t)$, is the sum of the signal, $s(t)$, and noise, $n(t)$. When $b(t)$ and $n(t)$ are normally distributed and independent, the information gained per one sample is $I = (1/2)\log_2(P + 1)$ , where $P$ is the power signal-to-noise ratio.

In the case of multichannel arrays, the channels need to be orthogonalized to make them independent of each other before the total information of the sensor array is calculated. Because Maxwell filtering transforms the measured signals $b(t)$ into orthogonal virtual channels $\alpha_{nm}$, we can directly utilize this property in evaluating the total information:

$$I_{tot} = \frac{1}{2} \sum_{n, m} \log_2 \left( \frac{\alpha_{nm}}{\eta_{nm}} \right)^2 ,$$

where $\eta_{nm}$ represents the sensor noise $n(t)$ converted into the VSH amplitude noises.

In general, the number of virtual channels can vary significantly. If $L_{in} = 5$, there are 35 components $\alpha_{nm}$, while the number of components increases to 143 for $L_{in} = 11$. Correspondingly, the number of external components $\beta_{nm}$ varies from 3 ($L_{out} = 1$) to 35 ($L_{out} = 5$). Virtual channels corresponding to highest spatial frequencies may become undetectable and therefore increase the noise in Maxwell filtering. Therefore, they need to be excluded.

*MaxFilter* utilizes the total information in determining the most optimal selection of virtual channels: maximal total information also indicates best signal to noise ratio (SNR). Total information is maximized iteratively using a hypothetical random current density of distributed over a spherical volume. Current density value $0.6\mu A/m^2$ per unit frequency band produces RMS magnitude of 100 fT in a radial point magnetometer at 3 cm from the surface.

Either inside expansion or both inside and outside epxansions together can be included in the optimization. At each iteration round, the virtual channel with smallest SNR ($\alpha_{nm}^2/\eta_{nm}^2$ or $\beta_{nm}^2/\eta_{nm}^2$) is neglected, new pseudo-inverse $S^\dagger$ is obtained, and amplitudes $\alpha_{nm}$, $\beta_{nm}$ and $\eta_{nm}$ are recalculated. Iteration is stopped when maximal $I_{tot}$ is found.

## A.5 Channel position weighting

Virtual channels $\alpha_{nm}$ with noises $\eta_{nm}$ have variable SNRs. Channels with small SNR can also be suppressed by applying Wiener filtering with the coefficients

$$w_{nm} = \frac{E[\alpha_{nm}{*}\alpha_{nm}]}{E[\alpha_{nm}{*}\alpha_{nm}] + E[\eta_{nm}{*}\eta_{nm}]},$$

where the noise estimates $E[\eta_{nm}{*}\eta_{nm}]$ can be determined from the sensor noises, and signal estimates $E[\alpha_{nm}{*}\alpha_{nm}]$ are evaluated using a random current density of distributed over a spherical volume (as above).

Increasing of magnetometer sensor noises can be prevented by modifying the Wiener filtering coefficients according to the channel positions. The outcome is an $N \times M$ weighting matrix $W$, where $N$ is the number of MEG channels and and $M$ is the number of all virtual channels $\alpha_{nm}$. The SSS reconstruction is thus performed as $b_{in} = \bar{S}_{in}x_{in}$ where $(\bar{S}_{in})_{nm} = W_{nm}(S_{in})_{nm}$.

B

# APPENDIX B Elekta Neuromag MEG sensors

## B.1 Sensor types

The detection coil geometry in *MaxFilter* is similar to the one described in NM20568A Source Modelling Software User's Guide Appendix B: "Coil geometry information".

The flux is an integral of the magnetic field component normal to the coil plane. Thus, the basis matrix elements $S_{nm}$ for $b_{in}$ are approximated by:

$$S_{nm}^{in} = \sum_{p=1}^{N_k} w_{kp} \nu_{nm}(\boldsymbol{r}_{kp}) \cdot \boldsymbol{n}_{kp},$$

where $\boldsymbol{r}_{kp}$ are a set of $N_k$ integration points covering the pickup coil loops of the sensor, $\nu_{nm}(\boldsymbol{r}_{kp})$ is the value of the inside VSH function at $\boldsymbol{r}_{kp}$, $\boldsymbol{n}_{kp}$ are the coil normal directions at these points, and $w_{kp}$ are the weights associated to the integration points. This formula essentially corresponds to numerical integration of the magnetic field over the pickup loops of sensor $k$. The VSH terms for $b_{out}$ are approximated in a similar way.

Table B.1 lists the parameters of the coil geometry descriptions employed in the software. The colums of the table contain the following data:

1. The number identifying the coil type. This number is used in the coil descriptions found in the FIFF files.
2. Description of the coil.
3. Number of integration points.
4. The locations of the integration points in coil coordinates.
5. Weights assigned to the field values at the integration points.

| Type | Description | $N_k$ | $r_k$ | $w_k$ |
|---|---|---|---|---|
| 2 | Neuromag-122<br>planar gradiometer | 8 | $(\pm 5.44, \pm 7.68, 0)$mm<br>$(\pm 11.1, \pm 7.68, 0)$mm | $\pm 1/(4 \cdot 16.2\text{mm})$ |
| 3012<br>3013<br>3014 | Type T1,<br>Type T2,<br>Type T3<br>planar gradiometer | 8 | $(\pm 5.89, \pm 6.71, 0.3)$mm<br>$(\pm 10.8, \pm 6.71, 0.3)$mm | $\pm 1/(4 \cdot 16.69\text{mm})$ |
| 3022<br>3023 | Type T1,<br>Type T2<br>magnetometer | 9 | $(0.0, 0.0, 0.3)$mm<br>$(\pm 9.99, \pm 9.99, 0.3)$mm<br>$(0.00, \pm 9.99, 0.3)$mm<br>$(\pm 9.99, 0.0, 0.3)$mm | $16/81$<br>$25/324$<br>$10/81$<br>$10/81$ |
| 3024 | Type T3<br>magnetometer | 9 | $(0.0, 0.0, 0.3)$mm<br>$(\pm 8.13, \pm 8.13, 0.3)$mm<br>$(0.0, \pm 8.13, 0.3)$mm<br>$(\pm 8.13, 0.0, 0.3)$mm | $16/81$<br>$25/324$<br>$10/81$<br>$10/81$ |

**Table B.1** MEG coil descriptions.

## B.2 Scaling between magnetometers and gradiometers

The Elekta Neuromag® data contains both magnetometer and planar gradiometer channels which have different SI units (T and T/m, respectively). Therefore, some scaling between the channel data and basis functions has to be applied before combining them for Maxwell filtering. *MaxFilter* utilizes the inherent RMS noise levels of the sensors:
$n_m$ = 3 fT/sqrt(Hz) for magnetometers,
$n_g$ = 3 fT/cm/sqrt(Hz) for gradiometers.

The basic modelling equation ($b = Sx$, see Appenix A.2), is presented in the form

$$diag(w)b = diag(w)Sx,$$

where the weight for the *k*th channel is $w_k = n_g/n_k$. Thus, the gradiometer channel weight becomes 1, while the magnetometer channel weight is 100.

Optionally, the weight $w_k$ for magnetometers can be set with the command-line option `-magscale` *mult*; the gradiometer weight is still one.

## B.3 Manipulation of sensor types

Most Elekta Neuromag® arrays consist of type T3 sensors (Table B.1). Data acquisition software may however mark the channels as type T1 or T2 sensors. In practice, the only difference between types T2 and T3 is that the magnetometer sizes are different (25.8 mm for T2, 21 mm for T3). *MaxFilter* applies the correct magnetometer size on the basis of the calibration factors even if sensors with smaller magnetometer size are marked as T1 or T2 sensors. Data analysis (source modelling) software may however utilize too large dimension for magnetometers, but such an error does not have a significant effect on the analysis results. *MaxFilter* option `-T3` was included to correct the sensor types (if needed) and to remove further modelling inaccuracies due to wrong sensor size.

Furthermore, older version of the source modelling programs (*xfit, mce*) may not recognize sensor types T3. Therefore, *MaxFilter* provides the option `-T2` for convenience: it marks all magnetometer and gradiometer channels as type T2 sensors. Note that this option may cause a small numerical error in the source modelling programs as explained above.

## B.4 Reconstruction of sensor signals

Maxwell filtering transforms measured MEG data inherently to harmonic function amplitudes which can be interpreted as *virtual* channels (see Appendix A.2 on page 66). The virtual channels are not stored in the output file, but they are instead utilized in Maxwell filtering operations, such as in composing interference-free brain signals and in transforming data between different head positions. Normally, the program applies the virtual channels to convert the input data to *idealized* sensors. Besides interference suppression, *MaxFilter* removes the distortions caused by imperfect calibration and gradiometer imbalance.

Sometimes it is however useful to reconstruct the signals $\boldsymbol{b}_{in}$ and $\boldsymbol{b}_{out}$ without correcting the above mentioned non-idealities, e.g., for comparison with the recorded signals. This kind of comparison is inherently utilized in bad channel detection (Section 3.4) and in *tSSS* (Section 3.5).

You can apply the command-line option `-reconst` to compose the non-idealized signals corresponding to spaces $S_{in}$ and $S_{out}$. The program applies fine-calibration data in determining the harmonic function amplitudes. The program estimates the signals $\boldsymbol{b}_{in}$ and $\boldsymbol{b}_{out}$ using the standard calibration extracted from the input file. Note that the cross-talk correction is however applied. Thus, in order to compare the original and reconstructed data you should run:

```
maxfilter -f infile.fif -reconst both -o rec.fif
maxfilter -f infile.fif -ctc only -o orig.fif
```

Thereafter you can overlay the output files and compare the differences.

C

# APPENDIX C Temporal subspace projection

The spatio-temporal extension to Maxwell filtering, *tSSS*, utilizes first spatial reconstructions of $\boldsymbol{b}_{in}$ and $\boldsymbol{b}_{out}$ for each sensor and each sample, and subtracts them from the measured data:

$$\boldsymbol{b}_s(t) = \boldsymbol{b}(t) - (\boldsymbol{b}_{in}(t) + \boldsymbol{b}_{out}(t)) \ .$$

The data $\boldsymbol{b}_{in}$ and $\boldsymbol{b}_s$ are packed in $n \times m$ matrices $B_{in}$ and $B_s$ (*m* sensors and *n* samples), and decomposed with the singular value decomposition:

$$\begin{array}{ll} B_{in} = U_{in}S_{in}V_{in}^T & S_{in} = diag(\sigma_k^{in}) \\ B_s = U_sS_sV_s^T & S_s = diag(\sigma_k^s) \end{array} ,$$

where $k = 1, \ldots, m$.

The columns of $V_{in}$ and $V_s$ span the waveforms of $\boldsymbol{b}_{in}(t)$ and $\boldsymbol{b}_s(t)$. Subspace intersection between the waveforms can be found with the QR decomposition:

$$V_{in} = Q_{in}R_{in} \ ; V_S = Q_sR_s \ ,$$

where $Q^TQ = I_m$ and $R \in \Re^{m \times m}$. An $m \times m$ matrix $C$ is composed from the $n \times m$ matrices $Q_{in}$, $Q_s$ as:

$$C = Q_{in}^TQ_s .$$

Singular value decomposition leads in

$$C = YS_CZ^T ,$$

where the diagonal matrix $S_C$ contains the singular values $\sigma_k$ of matrix $C$. The singular values define the principal angles $\theta_k$ between the two subspaces: $\cos(\theta_k) = \sigma_k$.

The intersection of the subspaces contains the waveforms corresponding to $\sigma_k^C = 1$. In practice, the subspace correlation limit in *MaxFilter* is set to 0.98.

If there are $p$ principal values exceeding the correlation limit, the program composes a *projection* operator $(I - LL^T)$ where $L(n \times p)$ contains the intersecting waveforms (first $p$ columns of the matrix $Q_s Z$). These waveforms are finally projected out as:

$$\hat{\boldsymbol{b}}_{in} = (I - LL^T)\boldsymbol{b}_{in}.$$

D

# APPENDIX D Head position estimation

## D.1 HPI signals

During the recording, the head position has to be tracked by feeding continuous sinusoidal signals to 4-5 head position indicator (HPI) coils. Typical frequencies of the signals are 154, 158, 162, 166 and 170 Hz for the sampling rate of 600 Hz (low-pass filter at 200 Hz), or 293, 307, 314, 321 and 328 Hz for higher sampling rates.

The resulting magnetic field on sensor $k$ is then

$$b_k(t) = \sum_{j=1}^{P} a_{k,j} \sin(\omega_j t + \varphi_j) + n_k ,$$

where $P$ is the number of coils, $a_{k,j}$ is the amplitude at sensor $k$ from the $j$th coil, $\omega_j$, $\varphi_j$ are, respectively, the angular frequency and phase of the $j$th sine signal, and $n_k$ is the noise. The phase can be represented as a linear combination of a sine and a cosine term. When $N$ data samples have been collected with sampling frequency $f = 1/T$, the data can be presented in matrix form as

$$\boldsymbol{b}_k = A\boldsymbol{x}_k + n_k .$$

The signal amplitudes fed to the coils are represented by the $N \times 2P$ matrix $A$:

$$A_{ij} = \begin{cases} \sin(\omega_j iT) \\ \cos(\omega_{j-P} iT) \end{cases},$$

and $x_k$ is the amplitude vector of the coils. The amplitudes can be estimated in the least squares sense as

$$\boldsymbol{x}_k = (A^T A)^{-1} A^T \boldsymbol{b}_k .$$

Besides sin and cos terms for the $P$ coils, this model can be augmented to include the contribution of interfering magnetic fields. Line frequency and its harmonics can be easily included, e.g., $\omega_j = 50, 100, \ldots, 300$ if the basic line frequency is 50 Hz and low-pass filtering frequency is 300 Hz.

Before estimating sinusoidal HPI signals, the program automatically performs high-pass filtering to remove slow background disturbances from the HPI signals. A finite impulse response (FIR) filter with the corner frequency of half of the lowest cHPI frequency is applied.

HPI signals are then fitted in windows of *N* samples (default 200 ms). The window slides forward in small steps (default 10 ms), and the HPI signal amplitudes are thus reconstructed 100 times per second.

The program judges the need of new HPI coil fitting according to correlations of spatial patterns between each time instant and the last fitted pattern. A new head position fit is needed when the correlation drops below 0.98 which corresponds to a move of about 3 mm. If the head does not move, a new position fit is typically done once per second. During rapid movements, head positions are fitted up to 100 times per second.

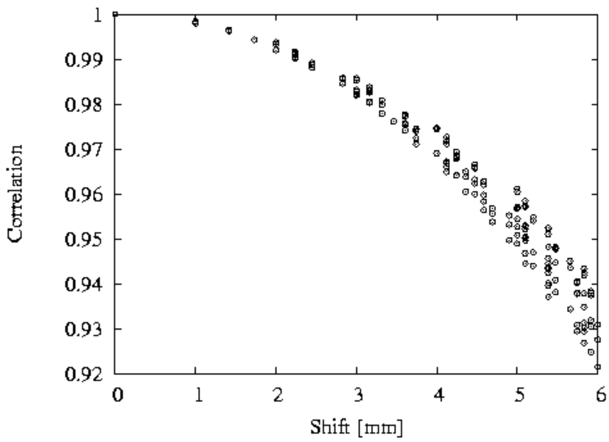


**Figure D.1** *Correlations of HPI signals when HPI coil is moved.*

## D.2 Coordinate matching

Let the point set $y_j, j = 1, \ldots, P$ represent the digitized head frame coordinates of the $M$ HPI coils, and $x_j, j = 1, \ldots, P$ represent the same coordinates in the device frame. The relation between the point sets can be expressed as

$$y_j = Rx_j + T,$$

where $R$ is a $3 \times 3$ rotation matrix and $T$ is a $3 \times 1$ translation vector.

Matching of the point sets is done by using the analytic solution based on the *quaternions*, presented in P.J. Besl and N.D. McKay, A Method for Registration of 3-D Shapes, IEEE Trans. Patt. Anal. Machine Intell., 14, 239 - 255, 1992.

Altogether seven quaternion parameters $(q_0 \ldots q_6)$ are needed to define the transformation between the coordinate systems. Using unit quaternions reduces the number of independent parameters to six, because $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$. Parameters $q_0 \ldots q_3$ define the rotation matrix, and parameters $q_4, q_5, q_6$ give the translation. The relation between the quaternion parameters and coordinate transformations is

$$R = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 + q_2^2 - q_1^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 + q_3^2 - q_1^2 - q_2^2 \end{bmatrix}$$

$$T = \begin{bmatrix} q_4 & q_5 & q_6 \end{bmatrix}^T$$

## D.3 HPI channels

When *MaxMove* options are in use, the program estimates the head position parameters and saves them as 9 new raw data channels with the names:

**QUAT001, CHPI001** quaternion parameter q1
**QUAT002, CHPI002** quaternion parameter q2
**QUAT003, CHPI003** quaternion parameter q3
**QUAT004, CHPI004** quaternion parameter q4
**QUAT005, CHPI005** quaternion parameter q5
**QUAT006, CHPI006** quaternion parameter q6
**QUAT007, CHPI007** goodness of fit
**QUAT008, CHPI008** HPI estimation error
**QUAT009, CHPI009** estimated movement velocity

Channel names QUATXXX indicate that movement compensation has not been applied, while the channel names are changed to CHPIXXX after the data have been transformed to the static reference head position.

## D.4 Head position file format

When the head positions are saved to an ascii file, the file contains a header line and one row for each fitted time interval:

| Time | q1 | q2 | q3 | q4 | q5 | q6 | g-value | error | velocity |
|---|---|---|---|---|---|---|---|---|---|
| 0.000 | 0.04553 | -0.00785 | 0.05864 | -0.00058 | -0.00372 | -0.00306 | 0.99959 | 0.00065 | 0.00000 |
| 0.200 | 0.04424 | -0.00772 | 0.06070 | -0.00089 | -0.00330 | -0.00317 | 0.99959 | 0.00064 | 0.00006 |
| 0.400 | 0.04433 | -0.00769 | 0.06066 | -0.00089 | -0.00331 | -0.00316 | 0.99959 | 0.00064 | 0.00005 |
| 0.599 | 0.04446 | -0.00773 | 0.06067 | -0.00090 | -0.00329 | -0.00317 | 0.99959 | 0.00065 | 0.00014 |
| 0.799 | 0.04442 | -0.00766 | 0.06070 | -0.00089 | -0.00329 | -0.00316 | 0.99958 | 0.00065 | 0.00008 |
| 0.999 | 0.04438 | -0.00765 | 0.06068 | -0.00090 | -0.00329 | -0.00316 | 0.99959 | 0.00065 | 0.00008 |
| ... | | | | | | | | | |

# APPENDIX E Command-line arguments

This section lists all options of the command-line program *maxfilter:*

```
/neuro/bin/util/maxfilter -f input_file.fif
[options]
```

***Common options:***

`-version`
Shows the version number of the program.

`-help`
Shows brief information of available options.

`-v`
Switches on verbose logging. This option also displays detailed processing history of the input file.

`-f` *input_file.fif*
Defines the FIFF-file where the evoked or raw data are read.

`-o` *output_file.fif*
Defines the FIFF-file where the results are written.

`-origin` *x0 y0 z0*
Sets the origin of the expansions to the point (*x0, y0, z0*) in the selected coordinate frame; *x0, y0, z0* must be given in mm.

`-origin` *fit*
Fits the origin to isotrak points or sensor locations.

`-frame` *device* | *head*
Sets the coordinate frame to *device* or *head.* The origin coordinates are given in this frame.

`-in` $L_{in}$
Sets the order of the expansion for $\boldsymbol{b}_{in}$.

`-out` $L_{out}$
Sets the order of the expansion for $\boldsymbol{b}_{out}$ .

`-bad` *bad_ch1 bad_ch2 ... bad_chn*

Marks static bad channels; *bad_chn* refers to the logical channel number, e.g., 0741, 1842, 2623.

`-autobad` *on* | *off* | *nraw*

Switches the automated bad channel detection *on* or *off,* or sets the number of tags to be scanned from the beginning of raw data file.

`-badlimit` *value*

Sets the standard deviation threshold of the automated bad channel detection.

`-skip` *t1_start t1_end ... tn_start tn_end*

Skips segments of raw data. The skip intervals are given as pairs of time points (in seconds) from the start of the file.

`-format` *short* | *float* | *long*

Sets the data packing format in the output file.

`-force`

Bypasses the warnings and error messages.

`-def`

Lists the default parameter values.

`-maint`

Lists special maintenance options.

***tSSS options:***

`-st` [*buflen*]

Applies the spatiotemporal *tSSS*, optionally sets the raw data buffer length in seconds.

`-corr` *limit*

Changes the subspace correlation threshold for *tSSS*, *limit* should be between 0 and 1.

`-waves` [*t*] *filename.fif*

Writes 42 first subspace waveforms into a FIFF file. Optional *t* defines the start time of writing in a raw data file. Also an ascii file *filename.fif.cor* is produced for listing the subspace correlation of each waveform.

***MaxMove options:***

`-trans` *trans_file.fif* | *default*
Transforms MEG data in *input_file.fif* into the sensor array defined by the channel info and transformation in *trans_file.fif*, or into default head position.

`-movecomp` [*inter*]
Estimates head positions and compensates head movements in a continuous raw data file. Option *inter* defines what to do when continuous HPI is off.

`-headpos`
Estimates head positions and stores head position parameter data, but does not compensate head movements.

`-hp` *pos_file.txt*
Stores the estimated head position parameters in a separate ascii file.

`-hpiwin` *n*
Sets the head position signal amplitude extraction window in ms.

`-hpistep` *n*
Sets the step size (ms) for moving the amplitude extraction window.

`-hpisubt` *amp* | *line* | *off*
Selects the continuous HPI signal subtraction approach.

`-hpicons`
Adjusts consistency of initial HPI coil fit and digitization.

`-hpie` *err_limit*
Sets the error limit for HPI coil fit acceptance.

`-hpig` *g_limit*
Sets the goodness limit for HPI coil fit acceptance.

`-linefreq` *lf*
Sets the basic line interference frequency (50 or 60 Hz).

***Other options:***

`-history`
Prints the processing history of the file and exits.

`-lpfilt` *corner*
Applies low-pass FIR filtering.

`-ds` [*factor* ]
Applies down-sampling and low-pass FIR filtering. Default down-sampling factor is 2.

`-site` *sitename*
Tries to load the fine-calibration and cross-talk correction data from the default directory (e.g. */neuro/databases/sss, /neuro/databases/ctc*) files *sss_cal_sitename.dat* and *ct_sparse_sitename.dat*.

`-cal` *sss_cal_file.dat* | *off*
Applies the fine-calibration data defined in file *sss_cal_file.dat*, or switches the fine-calibration off.

`-ctc` *ct_matrix_file.fif* | *off*
Applies cross-talk correction matrix in *ct_matrix_file.fif*, or switches the cross-talk correction off.

`-regularize` *both* | *in* | *off*
Applies multipole component selection using signal to noise ratios (maximized total information), or applies all components with the option *off*.

`-wchan`
Applies position-weighted Wiener filtering.

***Special options***

Furthermore, *MaxFilter* has some special options which are not meant for regular data analysis.

`-iterate [`*n*`]`
Applies iterative pseudo-inverse and multipole amplitude estimation, set *n* iteration rounds (default = 10). Value *n* = 0 forces direct pseudo-inverse.

`-reconst` *in* | *out* | *both*
Reconstructs inside, outside or both field components.

`-magbad`
Marks all magnetometer channels bad, uses only gradiometer channels in Maxwell filtering.

`-magscale` *mult*
Applies scaling between magnetometer and gradiometer channels. Default factor is 100.

`-T2`
Changes all Elekta Neuromag® sensors of type T3 to type T2.

`-T3`

Corrects Elekta Neuromag® type T2 magnetometers to T3 if needed.

`-list`

Shows more detailed output than option `-v`, mainly for tracing possible problems in processing the datafile.

`-mem`

Shows the total usage of the memory after prcessing each data block.

`-nosss`

Just copies input data to output. Default output filename is *input_file_nosss.fif*.

`-ctc` *only*

Applies cross-talk correction but does not do any Maxwell filtering operations. Default output filename is *input_file_ctc.fif*.

**Warning:** If internal active shielding was applied in the input file, the user must not perform data analysis on *MaxFilter* output files obtained with the maintenance options `-nosss` or `-ctc only.`

F

# APPENDIX F Revision history

This Appendix lists the changes made to the *MaxFilter* program and User's Guide.

**NM24057A**

- New software version 2.2 replaces *MaxFilter* 2.1.
- Job selection in the file loading dialog was improved, Section 2.5.1 on page 17.
- Temporal SSS default buffer length was changed, Section 3.5 on page 34.
- Continuous HPI signal subtraction choices, Section 4.3.1 on page 45.
- Notes were added about cleaning remaining cHPI signals, Section 4.3.1 on page 45, and averager rejections of large magnetometer channel values, Section 5.5 on page 56.
- Chapter 6 was renamed as "Performance and limitations". New sections were added: "6.1 Clinical validation study", "6.2.5 Limitations of tSSS" and "6.3 Guidelines for head movement correction".
- A warning was added that *tSSS* may diminish brain signals arising from very strong, superficial sources (page 62).
- Warnings were added about disturbances caused by continuous head movements (page 63).
- Several bug fixes (see the Release Note of *MaxFilter* 2.2).

Elekta Oy
Siltasaarenkatu 18-20
FIN-00531 Helsinki, Finland
Tel: +358 9 756 2400
Fax: +358 9 756 24011
Web: www.elekta.com