ATaCR Manual

Automated Tilt and Compliance Removal

Contents

Introduction **3**
Contact Information . . . 3
Getting Started . . . 4

Workflow **5**

Data **5**
Downloading Noise Data . . . 6
Downloading Event Data . . . 7
Preprocessing . . . 8
Input Noise Data Structure . . . 10
Input Event Data Structure . . . 10

Package Overview **11**
Setting the Main Parameters . . . 11
Running the Scripts . . . 12

Technical Details **13**
Daily Spectra Analysis . . . 13
Clean Station Spectra . . . 18
Transfer Function Calculation . . . 23
Event Correction . . . 27

Troubleshooting **30**
Limitations of the Quality Control Method . . . 30
Instrument Response . . . 30

References **30**

Introduction

The Automated Tilt and Compliance Removal (**ATaCR**, *pronounced attacker*) package is developed for characterizing and removing tilt and compliance noise from ocean bottom seismometer (OBS) instruments.

Details related to the quality control procedures used to obtain stations' noise characteristics can be found in Janiszewski et al. (2019). We employ tilt and compliance corrections based on the methods of Crawford and Webb (2000) and Bell et al. (2014).

This manual will walk a new user through the steps to produce tilt and compliance corrected seismograms from OBS data, starting with downloading the data and ending with corrected seismograms. File and folder names are written in **`teal`** and variable names are written in **`magenta`**.

Users of this code should cite: *Janiszewski, H A, J B Gaherty, G A Abers, H Gao, Z C Eilon, Amphibious surface-wave phase-velocity measurements of the Cascadia subduction zone, Geophysical Journal International, Volume 217, Issue 3, June 2019, Pages 1929-1948, https://doi.org/10.1093/gji/ggz051*

Contact Information

Have questions? Find bugs? Suggestions or contributions?
Contact author: Helen Janiszewski
E-mail: hjaniszewski@carnegiescience.edu

Getting Started

The **ATaCR** package is written in Matlab and requires the Curve Fitting, Optimization, Statistics and Machine Learning, and Image Processing toolboxes.

This manual is divided into three main sections:

Data will lead you through the steps to download and process data using the supplied scripts; or, if you prefer to use your own scripts, it provides you with the data formatting requirements to run the main package.
Package Overview reviews the sequence of codes for the tilt and compliance corrections.
Technical Details includes information for all user defined variables and descriptions of all figures produced while running this code package.

We also briefly outline some common troubleshooting techniques and common possible data issues.

New Users Start Here

Try to replicate the results included in the sample folder using your preferred data download and processing methods. Examples of the analysis for a month of data recorded at the Cascadia Initiative ocean bottom seismometers M08A and G03A are given in the folder NOISETC_SAMPLE_CI. It includes daily noise files from March 1 - March 30, and an example earthquake correction for the Mw 6.6 earthquake that occurred near Vanuatu on March 9, 2012.

Workflow

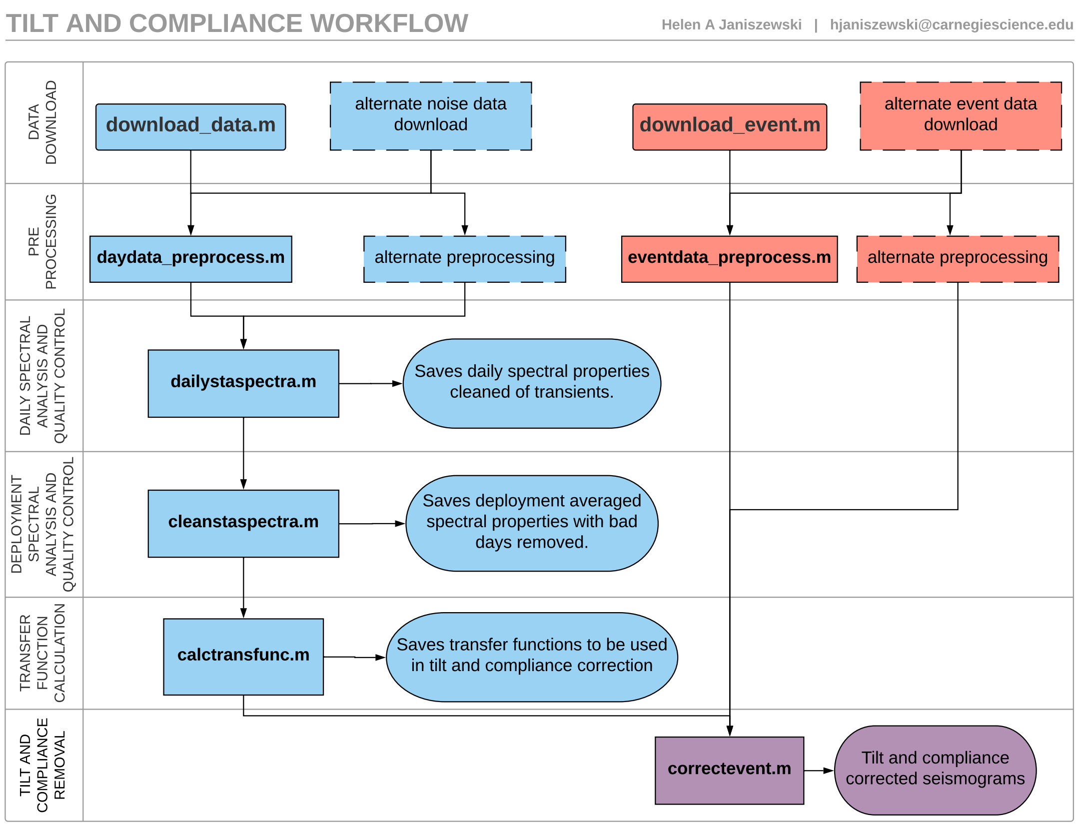

Figure 1: Flowchart for using the **ATaCR** Package.

Data

There are two types of data the **ATaCR** code requires: noise data and event data. The noise data refers to data used to calculate transfer functions for the correction. The event data refers to the data that is being corrected for tilt and compliance noise.

For both the noise and event data, we provide the user with scripts to download the data using `irisFetch.m` and to perform standard preprocessing, including downsampling, response removal, and gain adjustment. However, the user may elect to use their own methods for data download and preprocessing and subsequently format the data to be compatible with the **ATaCR** package.

Java Heap Memory

If you are using the provided Matlab scripts to download daily time series data, you may need to increase the Java Heap Memory in Matlab preferences. If you find that daily data files are not being saved, try increasing this value. A value of 2048 MB has been found to work for broadband data with a sample rate up to 125 sps.

Downloading Noise Data

Download daily noise time series files using **`download_data.m`**; the parameters are:

```
startlist = 'NOISETC_CI/starttimes_CItest.txt';
datalength = 86400;
download_networks = '7D';
download_stations ='M08A';
chz_vec = 'BHZ';
ch1_vec = 'BH1';
ch2_vec = 'BH2';
chp_vec = 'BDH';
datacache = 'NOISETC_CI/DATA/datacache_day';
```

`startlist` - location of an ASCII file listing days to download for the noise data. Each day should be formatted as **`YYYYMMDD0000`**, for example 201203070000.

`datalength` - length of data to download in seconds. In principle, the package can handle data of any length, but it has only been tested for downloading daily values so we do not recommend changing this parameter from its default value of 86400.

`download_networks` - list of networks to download from IRIS. Can list multiple following the formatting guidelines for **`irisFetch.m`**.

`download_stations` - list of stations to download from IRIS. Can list multiple following the formatting guidelines for **`irisFetch.m`**, or use * to download all stations part of a given network.

ch[z12p]_vec - list of channel names that are acceptable for the vertical, horizontal 1, horizontal 2, and pressure components respectivley. Can list multiple following the formatting guidelines for **irisFetch.m**.

datacache - location to save noise data.

Downloading Event Data

Download event time series files using **download_event.m**; the parameters are:

```
startlist = 'NOISETC_CI/eventtimes_CItest.txt';
datalength = 7200;
download_networks = '7D';
download_stations = 'M08A';
chz_vec = 'BHZ';
ch1_vec = 'BH1';
ch2_vec = 'BH2';
chp_vec = 'BDH';
datacache = 'NOISETC_CI/DATA/datacache';
```

startlist - location of an ASCII file listing start times for the event data. Each event should be formatted as **YYYYMMDDhhmm**, for example 201203090709.

datalength - length of data to download in seconds. The default (7200 s) is designed for teleseismic surface wave analysis. The length of this data window is tied to the length of the window used to calculate spectra. We have not tested lengths of time other than 7200 s. We recommend using a time window similar to this length for the corrections, and subsequently cutting out a shorter window of data if your analysis does not require a long window.

download_networks - list of networks to download from IRIS. Can list multiple following the formatting guidelines for **irisFetch.m**.

download_stations - list of stations to download from IRIS. Can list multiple following the formatting guidelines for **irisFetch.m**, or use * to download all stations part of a given network.

ch[z12p]_vec - list of channel names that are acceptable for the vertical, horizontal 1, horizontal 2, and pressure components respectivley. Can list multiple following the formatting guidelines for **irisFetch.m**.

datacache - location to save event data. Make sure this directory is separate from the noise data directory.

Preprocessing

Use **daydata_preprocess.m** and **eventdata_preprocess.m** to perform basic preprocessing on the noise and event time series respectively. Currently, our codes support options to downsample data, remove instrument response, and perform gain adjustments. The scripts are designed to run for a single station.

Data Requirements

There is flexibility for the user to decide on the most appropriate preprocessing steps for their project, but **it is essential that the noise and event data are processed identically to calculate accurate corrections**. All channels must also have the same sample rate.

The parameters for **daydata_preprocess.m** and **eventdata_preprocess.m** will be the same, with the exception of **INPUTdir** and **OUTPUTdir**. An example of the user input parameters for **daydata_preprocess.m** is:

```
INPUTdir = 'NOISETC_CI/DATA/datacache_day';
OUTPUTdir = 'NOISETC_CI/DATA/datacache_day_preproc/'
pole_zero_dir='';

network = '7D';
station = 'M08A';
channels = {'BHZ','BH1','BH2','BDH'};

resprm = [1 1 1 1];
gaincorr = [1 1 1 1];
samprate = 5;
hp_filt = [0 0 0 0];

lo_corner = 0.001;
npoles=5;
```

INPUTdir - location of input seismic data. This should be the output folder from **download_day.m** or **download_event.m**, or a similarly formatted directory.

OUTPUTdir - location to save the preprocessed data.

pole_zero_dir - for use in response removal. The default uses the poles and zeros downloaded and saved in the data files through **irisFetch.m**; however, here you can specify a directory of SAC_PZ files to use instead. Unless you identify problems with the poles and zeros downloaded through **irisFetch.m**, we recommend leaving this blank.

network- network for given station.

station - station name.

channels- list of N channel names. Can accept up to four channel names. For tilt and compliance corrections, the vertical, two horizontal, and pressure channels are required.

resprm- vector of length N with each entry corresponding to a channel. Can be a value of either 1 or 0. A value of one will remove the instrument response for that channel, a value of 0 will not. Response removal is recommended unless there is a known problem.

gaincorr- vector of length N with each entry corresponding to a channel. Each entry corresponds to the factor by which the data will be multiplied for a gain correction. If no gain correction is necessary, a factor of 1 should be specified.

samprate - scalar integer indicating the final sample rate for each channel after downsampling (samples/s). We recommend using the minimum sample rate needed for your analysis to avoid large file sizes.

hp_filt - vector of length N with each entry corresponding to a channel. Can be a value of either 1 or 0. Should only be set to 1 when the response is not removed for that channel (i.e. resprm = 0). The response removal step includes a long-period high-pass filter, so this step is unnecessary if the response has already been removed.

lo_corner - the corner frequency in Hz for the long-period high-pass filter used in the response removal. Recommended to keep default value of 0.001 Hz.

npoles - number of poles for the long-period high-pass filter.

Input Noise Data Structure

Noise data are saved in matfiles. We include example folders as a guide to replicate matfile structure if you prefer to use your own downloading and preprocessing scripts.

The daily noise data are under **DATA/datacache_day_prepro**.

Files are organized under a **NETWORK/STATION/** folder structure.

Each matfile name is formated as **YYYYMMDD0000_NT_STNM.mat** where **YYYYMMDD0000** stands for the day, **NT** stands for the network code, and **STNM** stands for the station name.

Within this matfile there is structure named **traces_day**. It is a 1xN structure with N channels that has the same structure as the output data using **irisFetch.m**. Only a subset of fields downloaded via **irisFetch.m** are required for the package. Below is an example entry for the vertical channel of station M08A including only the required fields.

```
traces_day(4) =

network: '7D'
station: 'M08A'
channel: 'BHZ'
elevation: -154
data: [432000x1 double]
sampleRate: 5
startTime: 7.3494e+05
endTime: 7.3494e+05
```

Input Event Data Structure

Event data are saved in matfiles. We include example folders as a guide to replicate matfile structure if you prefer to use your own downloading and preprocessing scripts.

The event data are under **DATA/datacache_preproc**.

Files are organized under a **YYYYMMDDmmss** folder structure that corresponds to the event origin time.

Each matfile name is formated as **YYYYMMDDmmss_NT_STNM.mat** where **YYYYMMDDmmss** stands for the event origin time, **NT** stands for the network code, and **STNM** stands for the station name.

Within this matfile there is structure named **traces**. It is a 1xN structure corresponding to N channels. The required fields are:

```
traces(4) =

network: '7D'
station: 'M08A'
channel: 'BHZ'
data: [36000x1 double]
sampleRate: 5
```

Package Overview

Setting the Main Parameters

Once the data has been downloaded and preprocessed you can calculate tilt and compliance corrections. First adjust the parameters in **setup_parameter.m**. For routine tilt and compliance corrections, we recommend only changing the directory, station, and channel information:

```
WORKINGdir = 'NOISETC_CI/DATA/datacache_day_preproc/';
OUTdir = 'NOISETC_CI/DATA/NOISETC';
FIGdir = 'NOISETC_CI/FIGURES/NOISETC';

network = '7D';
station = 'M08A';

chz_vec = {['HHZ'], ['BHZ'], ['LHZ']};
ch1_vec = {['HH1'], ['BH1'], ['LHN'], ['LH1']};
ch2_vec = {['HH2'], ['BH2'], ['LHE'], ['LH2']};
chp_vec = {['HDH'], ['BDH']};
```

The remaining parameters do not need to be changed for most projects, but offer the user flexibility for more unique investigations. We provide detailed descriptions of these parameters in the **Technical Details** section.

Running the Scripts

Once the data has been downloaded, preprocessed and formatted, you are ready to run the main package. The sequence of codes is:

1. **`dailystaspectra.m`**. This runs for a single station and loops over all daily noise files. For each station, it calculates daily spectral properties. Toggles for plotting and file overwriting are at the top of the script. It is recommended to turn figures off except when troubleshooting or evaluating station quality. The first quality control step is applied in this script.
   Input: preprocessed daily time series.
   Output: **`SPECTRA/*.mat`**

2. **`cleanstaspectra.m`**. This runs for a single station. This code performs a second step of quality control, discarding spectra from low quality days. It will then calculate and save an average spectrum for each station in a new file. Toggles for figures and overwriting are located at the top of the script. It is recommended that figures are turned on to review spectral properties.
   Input: **`SPECTRA/*.mat`**
   Output: edited **`SPECTRA/*.mat`** and **`AVG_STA/*.mat`**

3. **`calctransfunc.m`**. This code loops over all stations for which spectra files exist and saves transfer functions based on the daily and station average spectral properties. Toggles for figures and overwriting are located at the top of the script. It is recommended to turn figures on.
   Input: **`SPECTRA/*.mat`** and **`AVG_STA/*.mat`**
   Output: **`TRANSFUN/*.mat`**

4. **`correctevent.m`**. This applies tilt and compliance corrections based on the transfer functions calculated in the previous step. It loops over all stations and all events. At the top of this script, specify the path to the event data and a period range for filtering the seismograms for plotting. Plotting and overwriting can be toggled on and off at the top of the script, and is recommended to be turned on.

Input: **`TRANSFUN/*.mat`** and user preprocessed event time series.
Output: **`CORRSEIS/*.mat`** or **`CORRSEISAVTF/*.mat`**

Technical Details

Daily Spectra Analysis

The script **`dailystaspectra.m`** calculates daily spectral properties for the noise data for a given station. Each day of day of data is divided into N (determined by the window length) overlapping windows with a Hanning taper applied to each window equal to the width of the overlap.

The first step of quality control is applied. Windows that include transient events are identified (Janiszewski et al., 2019) and discarded. The spectra of all windows that pass the quality control are stacked and averaged to determine daily noise properties. Following Bell et al. (2014) the orientation of maximum coherence between the vertical and horizontal components is determined for each day.

These variables from **`setup_parameter.m`** are related to this code:

```
% Spectral Properties Windowing
T = 7200;
overlap = 0.3;

% Quality Control Parameters for Daily Windows
pb = [0.004, .2];
tolerance = 1.5;
a_val = 0.05;
minwin = 10;

% Tilt orientation
tiltfreq = [.005, .035];
```

`T` - the length of each window in seconds. Should equal event window length.

`overlap` - fractional width of the window overlap for Hanning taper.

pb - frequency range (Hz) over which quality control step is applied. Should span the frequency range where you plan to apply the corrections.

tolerance - tolerance factor for quality control; higher values will accept more windows.

a_val - confidence level for the f-test for the quality control. The value of 0.05 corresponds with a 95% confidence level.

minwin - the minimum number of windows that must pass the quality control for an average daily spectra to be calculated and saved. Days that have fewer than this number of windows are discarded.

tiltfreq - frequency range (Hz) over which maximum coherence direction between vertical and horizontal channels is calculated.

This script can produce figures (e.g. Figures 2 - 5) showing the daily noise spectra, the quality control process, and the orientations. This is helpful for troubleshooting. These are controlled by following variables in **dailystaspectra.m**. Each can be set to 0 or 1; 1 means the figure or file will be plotted or saved, and 0 means it will not.

```
% CODE OPTIONS
isfigure_spectrogram = 0;
isfigure_powerspec = 0;
isfigure_orient = 1;
issavefigure = 1;
isoverwrite = 1;
```

isfigure_spectrogram - produces plots similar to Figure 2 for each day of noise data. Recommended to be set to 0 unless troubleshooting due to script run time.

isfigure_powerspec - produces plots similar to Figure 3 for each day of noise data. Recommended to be set to 0 unless troubleshooting due to script run time.

isfigure_orient - produces plots similar to Figures 4 and 5 for each station. Recommended to be set to 1 if evaluating patterns of tilt orientation.

issavefigure - If set to 1, will save all figures generated.

isoverwrite - If set to 1, will overwrite all spectra files.

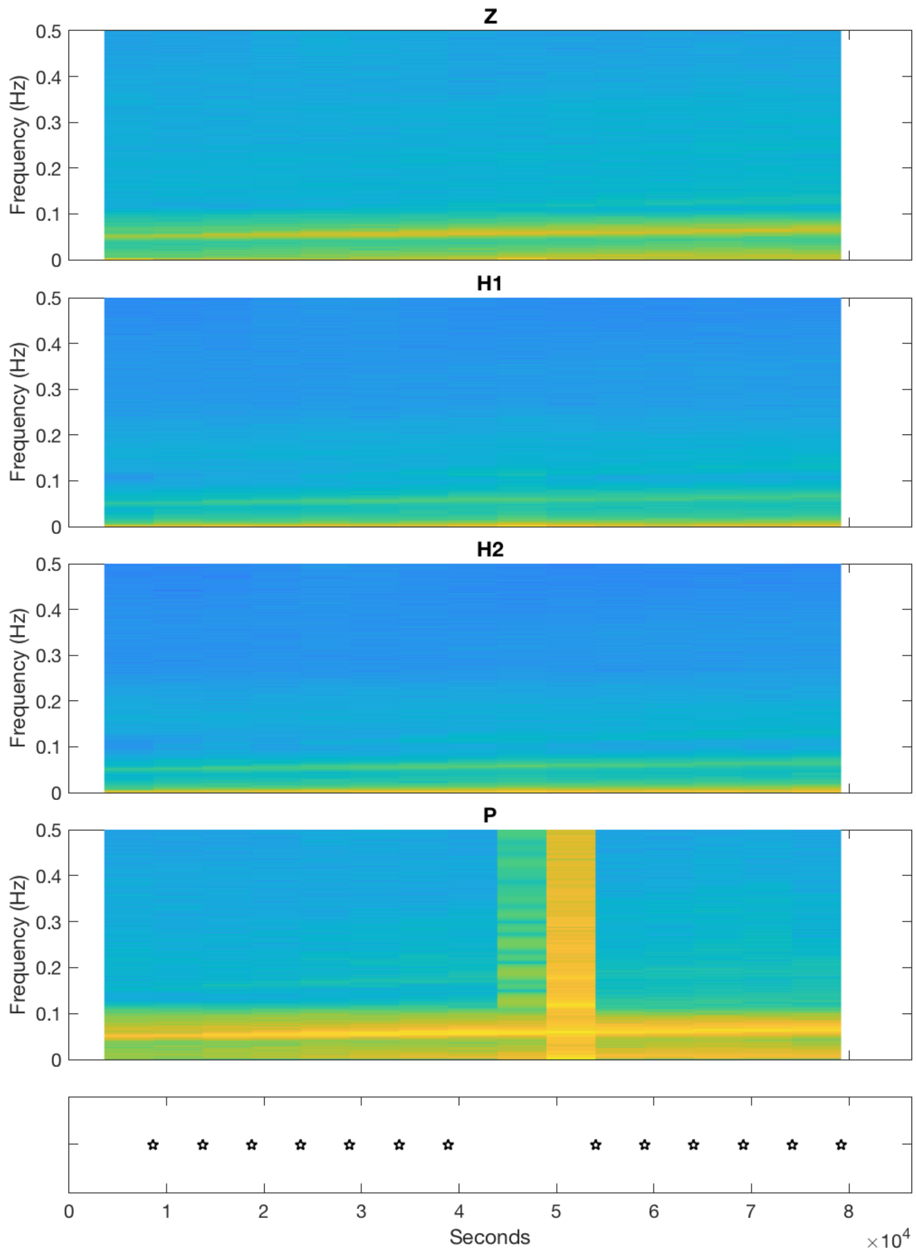

Figure 2: Daily spectrogram for the vertical (Z), horizontals (H1, H2), and pressure (P) components. The bottom panel indicates the windows that passed the quality control; a star beneath the window indicates the window is accepted, while the absence of a star indicates it is excluded from the daily average. In this example, three windows are excluded due to the anomalous signal on the P component. Example is from March 4, 2012 on station M08A.

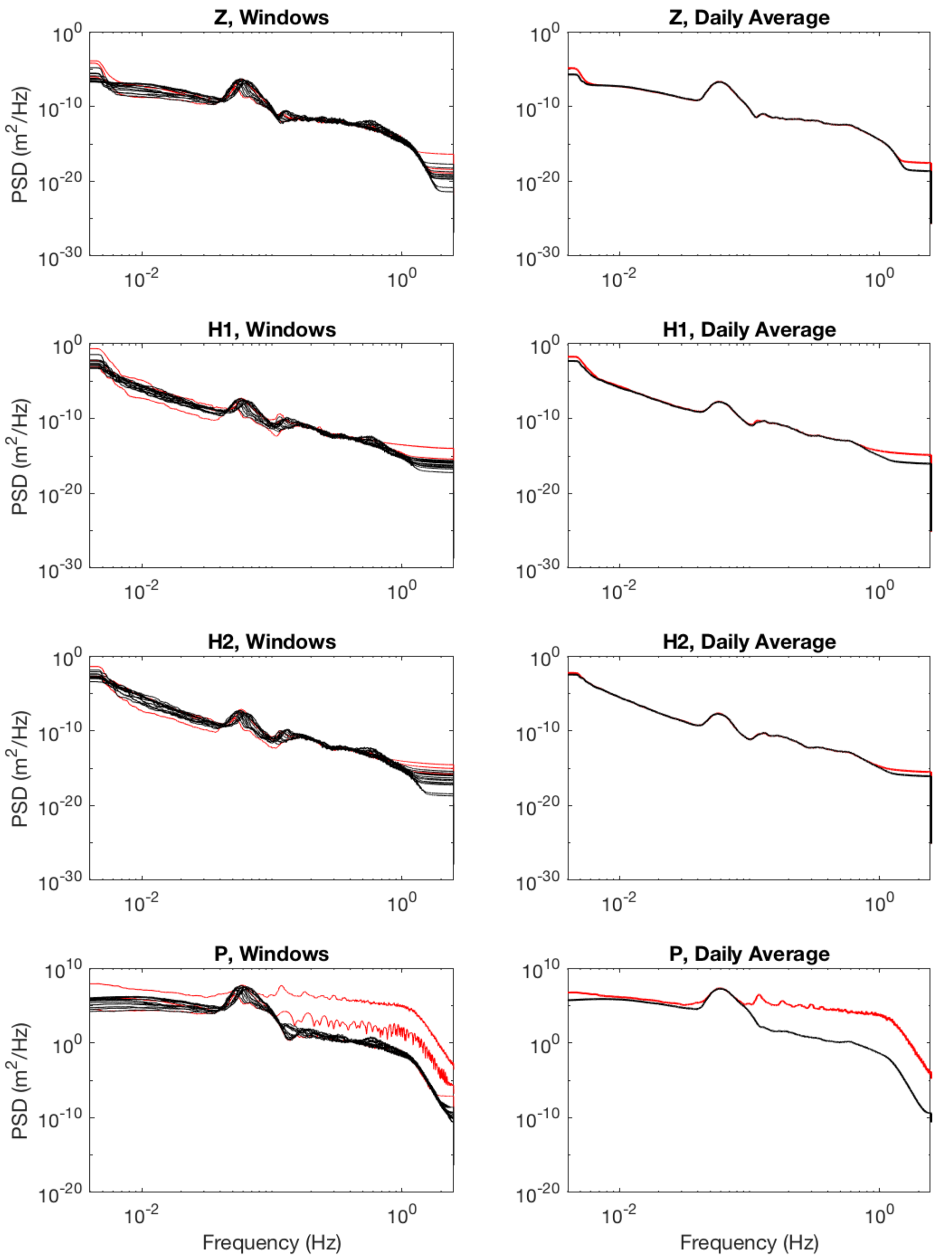

Figure 3: Power spectral density (PSD) functions for the Z, H1, H2, and P components from a single day of data (M08A, March 4, 2012, same as in Figure 2). The left column shows PSDs for each individual window; PSDs from windows that did not pass the quality control are colored red (the three windows where stars are absent in Figure 2). The right column compares the daily average PSD with (red) and without (black) the bad windows included.

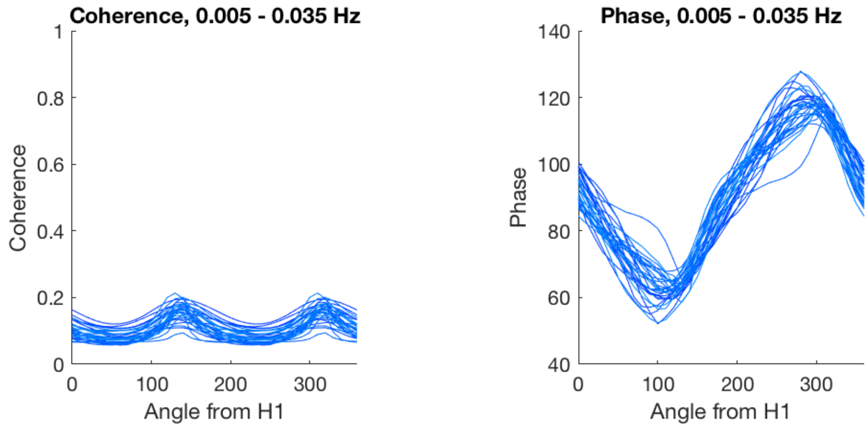

Figure 4: The orientation of maximum coherence between the vertical and the two horizontal components for M08A during March 2012. (Left) Coherence as a function of angle from the H1 component. (Right) Phase as a function of the angle. In this example, the coherence is low indicating the absence of dominant, uni-directional tilt noise.

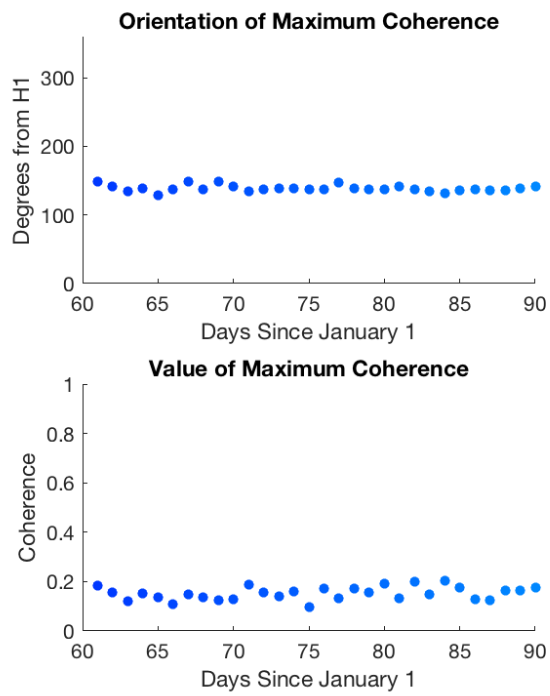

Figure 5: The orientation of maximum coherence for each day of noise data (top panel), and the value of the coherence that corresponds to that orientation (bottom panel) for March 2012 at M08A.

Clean Station Spectra

The script **cleanstaspectra.m** calculates spectral properties averaged overmultiples days during the deployment for a given station. It applies a second round of quality control to discard days with anomalous PSDs relative to the larger time of interest. Bad days are flagged and not used in subsequent analyses. All good days are then used to calculate average spectral properties for the station over the time period of investigation.

```
% Quality Control Parameters for Deployment Days
pb_dep = [0.004 .2];
tolerance_dep = 1.5;
a_val_dep = 0.05;
```

pb_dep - frequency range (Hz) over which quality control step is applied. Should span the frequency range where you plan to apply the corrections.

tolerance_dep - tolerance factor for quality control; higher values will accept more windows.

a_val_dep - confidence level for the f-test for the quality control. The value of 0.05 corresponds with a 95% confidence level.

This script can output figures showing the power spectral density functions, coherence, admittance, and phase as a function of frequency for all days at each station (for a detailed explanation on the definitions of these spectral properties see Crawford and Webb, 2000, Bell et al., 2014, and references therein). The figures highlight days that have been discarded. The options to turn plotting and saving of these figures on and off are located at the top of **cleanstaspectra.m**. Examples of these figures are shown from Figures 6 - 9.

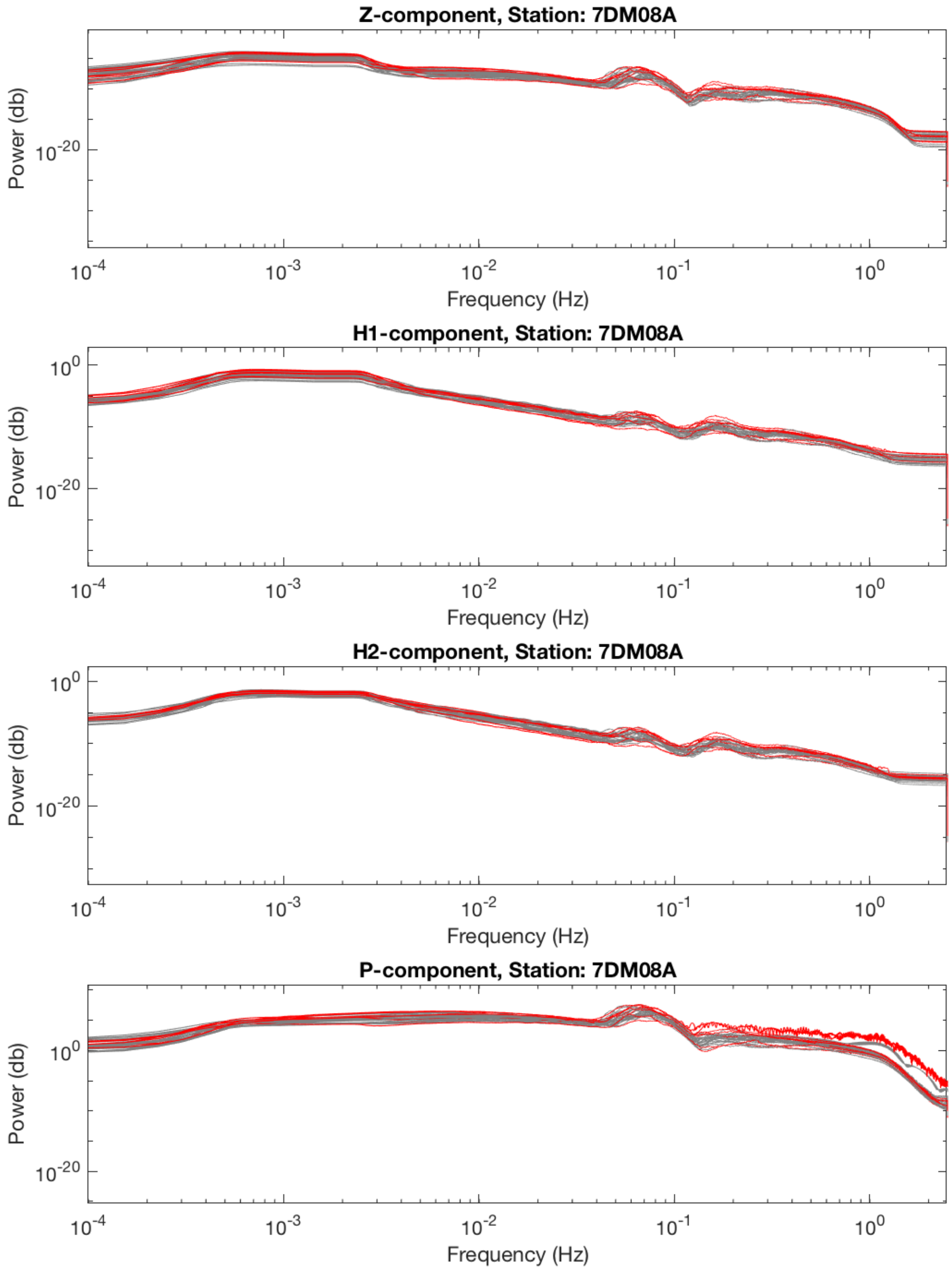

Figure 6: The daily PSDs plotted for the vertical (Z), horizontal (H1, H2), and pressure (P) components for March 2012 at station M08A. Each line is a daily PSD. Gray colors indicate days that were accepted by the second quality control step, while the red colors indicate days that were discarded.

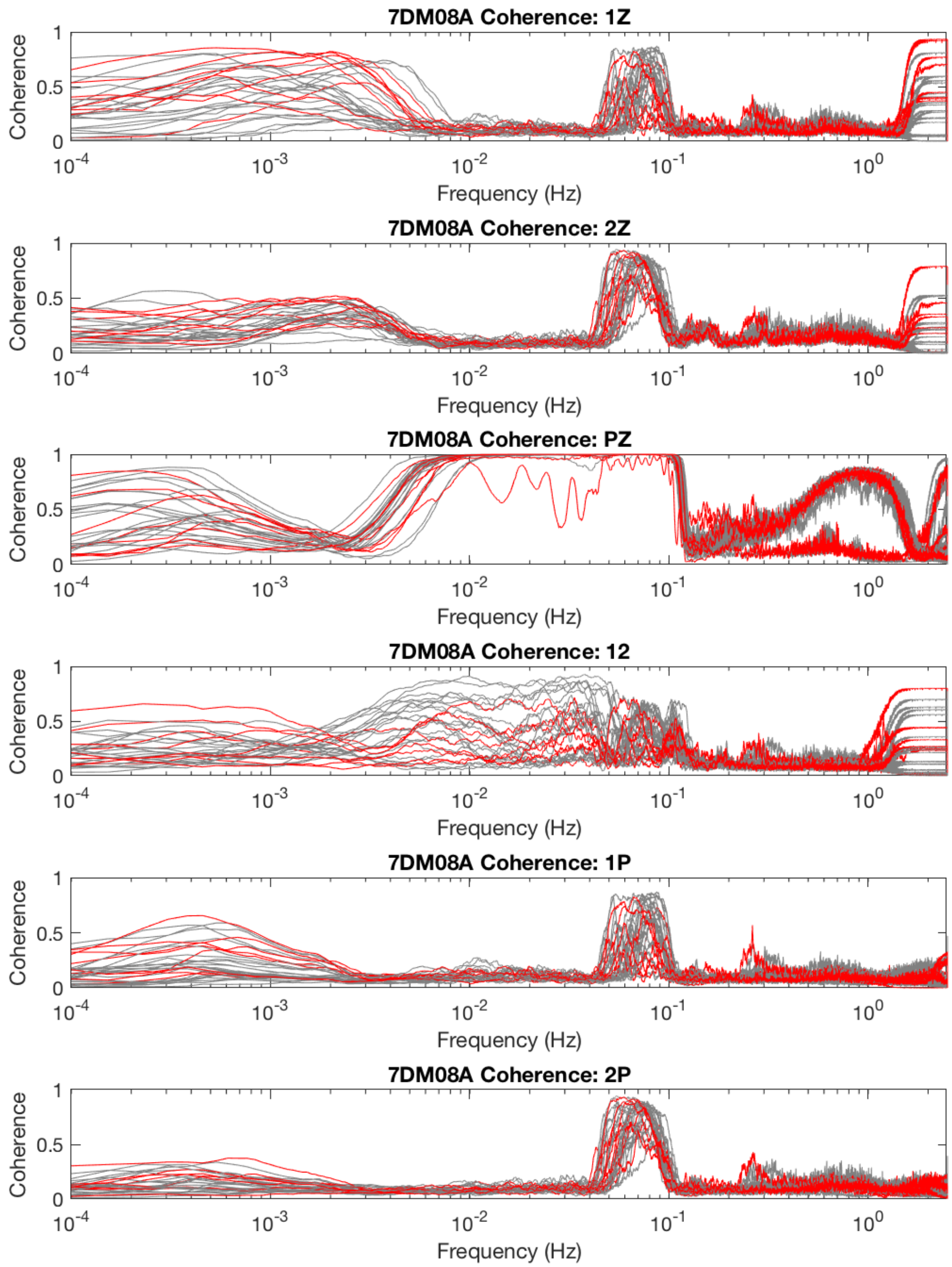

Figure 7: The daily coherences between pairs of components as indicated above each subplot for March 2012 at station M08A (e.g. 1Z - coherence between Horizontal 1 and Vertical). Each line represents average coherence for a single day in the data set. Gray colors indicate days that were accepted by the second quality control step, while the red colors indicate days that were discarded.

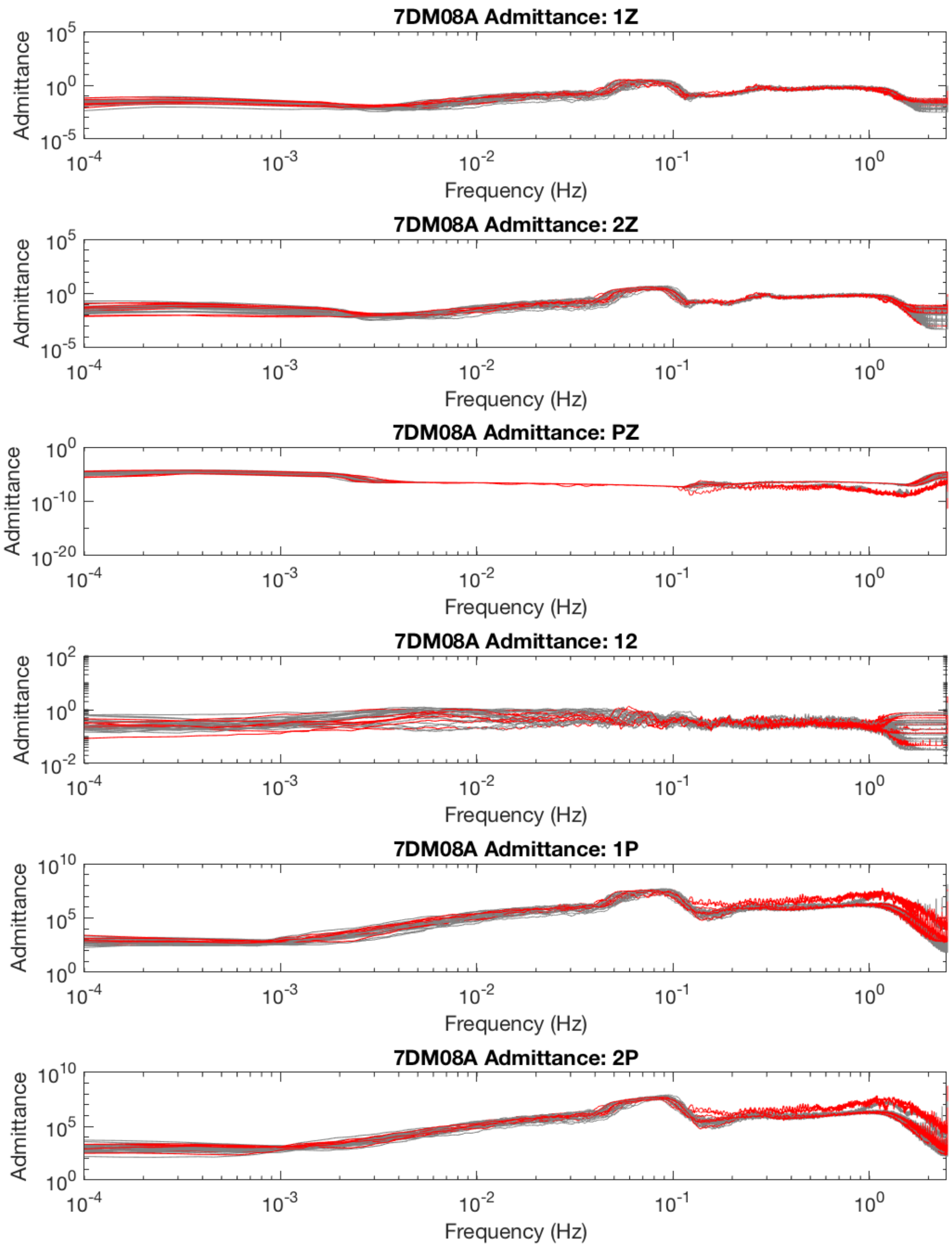

Figure 8: The daily admittances between pairs of components as indicated above (see Figure 7) each subplot for March 2012 at station M08A. Each line is a daily admittance. Gray colors indicate days that were accepted by the second quality control step, while the red colors indicate days that were discarded.

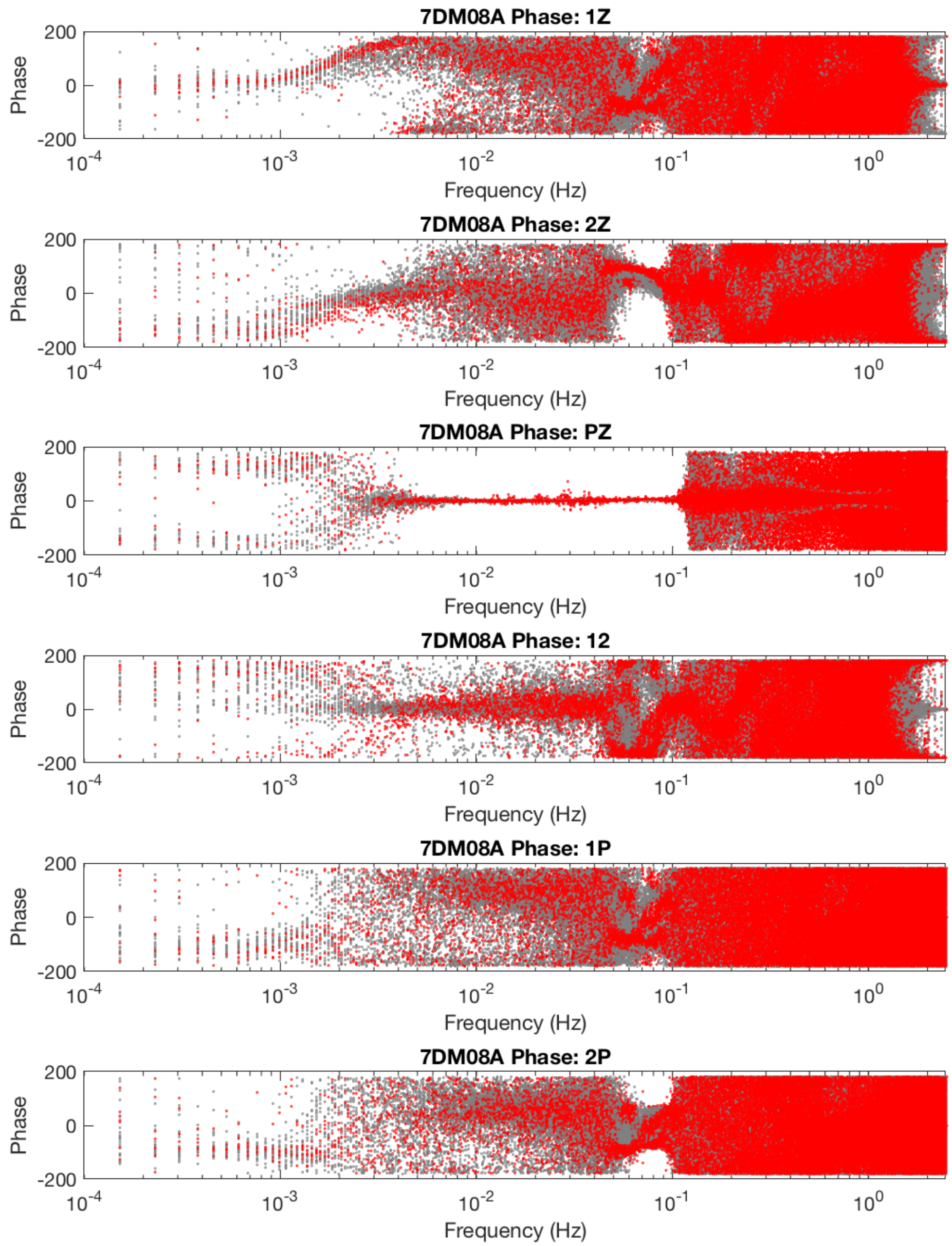

Figure 9: The daily phases between pairs of components as indicated above (see Figure 7) each subplot for March 2012 at station M08A. Each line is a daily phase. Gray colors indicate days that were accepted by the second quality control step, while the red colors indicate days that were discarded.

Transfer Function Calculation

The script **calctransfunc.m** calculates the transfer functions between the components of interest (specified in **TF_list**). It calculates daily transfer functions from the spectra of all days that have passed the quality control, and station average transfer functions from the spectra calculated in **cleanstaspectra.m**.

At this point in the ATaCR package, the user must decide what correction method(s) they wish to use to remove tilt and compliance noise from the event data; different methods will require different transfer functions to be calculated in this step. There are several commonly used methods:

1. Compliance correction (e.g. Crawford and Webb, 2000). This is appropriate if you want to only remove compliance noise from the vertical component. This correction requires the transfer function between Z and P to be calculated. An example of a situation where this correction would be appropriate is if one or more of the horizontal components of the seismometer failed, but it still had a working pressure and vertical component.

   ```
   TF_list = {'ZP'};
   ```

2. Component-wise tilt and compliance correction (e.g. Crawford and Webb, 2000). This is appropriate if you want to remove tilt and compliance noise from the vertical component. This correction requires a series of transfer functions to be calculated. First a transfer function between Z and H1 is calculated. Then the transfer function between Z with H1 removed and H2 is calculated. Finally, the transfer function between Z with both H1 and H2 removed and P is calculated. An example of a situation where this correction would be appropriate is if all three components of the seismometer and the pressure gauge are working, and you want to remove both tilt and compliance noise, but you are not sure if the tilt noise is a strong, unidirectional signal.

   ```
   TF_list = {'Z1', 'Z2-1', 'ZP-21'};
   ```

3. Component-wise tilt correction (e.g. Crawford and Webb, 2000). This is appropriate if you want to only remove tilt noise from the vertical component. To perform this correction, you should calculate the series of transfer functions used in the component-wise tilt and compliance correction - you will just ignore the last transfer function in the correction step. An example of a situation

where this correction would be appropriate is if only the seismometer is working, not the pressure gauge, and you want to remove tilt noise, but you are not sure if the tilt noise is a strong, unidirectional signal.

```
TF_list = {'Z1', 'Z2-1', 'ZP-21'};
```

4. Rotational tilt and compliance correction (e.g. Bell et al., 2014). This is appropriate if you want to remove tilt and compliance noise from the vertical component. This correction requires a series of transfer functions to be calculated. First a transfer function between Z and the horizontal direction of maximum coherence, H, is calculated. Then the transfer function between Z with H removed and P is calculated. An example of a situation where this correction would be appropriate is if all three components of the seismometer and the pressure gauge are working, you want to remove both tilt and compliance noise, and the tilt noise is a strong unidirectional signal.

```
TF_list = {'ZH', 'ZP-H'};
```

5. Rotational tilt correction (e.g. Bell et al., 2014). This is appropriate if you want to only remove tilt noise from the vertical component. To perform this correction, you should calculate the series of transfer functions used in the rotational tilt and compliance correction - you will just ignore the second transfer function in the correction step. An example of a situation where this correction would be appropriate is if only the seismometer is working, not the pressure gauge, you want to remove tilt noise, and the tilt noise is a strong, unidirectional signal.

```
TF_list = {'ZH', 'ZP-H'};
```

```
% Transfer Function Options
TF_list = {'ZP','Z1','Z2-1','ZP-21','ZH','ZP-H'};
% convention:
% Z1=transfer function between Z and H1
% Z2-1=transfer function between Z with H1 removed, and H2
% ZP-21=transfer function between Z with H1 and H2 removed,
  and P
% ZH=transfer function between Z and rotated max horizontal
  noise direction (H)
```

TF_list - The list of components for which to calculate transfer functions. The naming conventions are explained in the comments, and see above for the appropriate inputs for common correction methods. For the more advanced user, other combinations should be available (e.g. ZP, Z1-P, Z2-1P to remove the compliance and then the two component tilt); these have not been thoroughly tested so proceed with caution.

This script can output figures showing the amplitudes of the transfer functions (i.e. the admittance) for all days at each station and the average over all days (Figure 10). Options to turn plotting and saving of figures on and off are at the top of **calctransfunc.m**. You can also choose to overwrite saved transfer functions here.

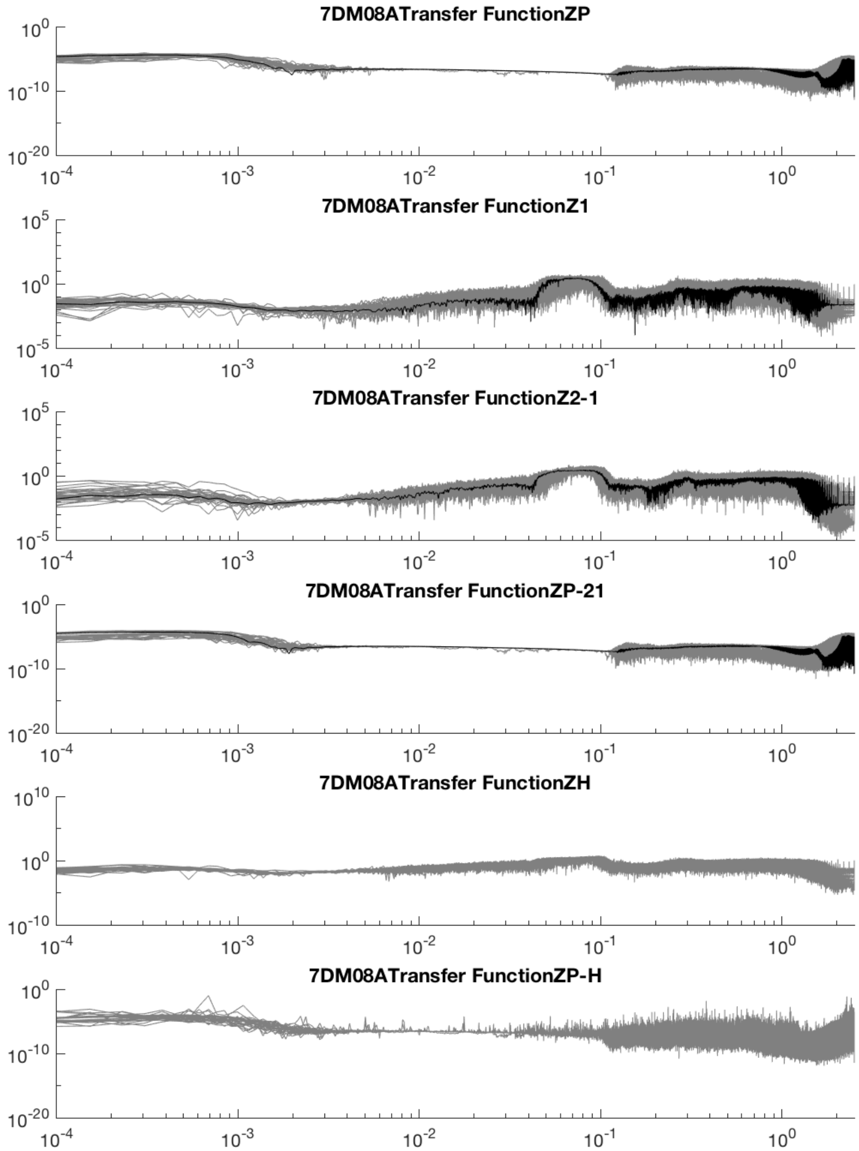

Figure 10: Transfer function amplitudes for the component combinations of interest, as indicated in the title of each subplot. This example is for the month of March 2012 for station M08A. The daily transfer functions are shown in grey and the average calculated for the whole month is shown in black.

Event Correction

The script **correctevent.m** uses the transfer functions from **calctransfunc.m** to correct event data for tilt and compliance noise. It will save time series for all events with these corrections applied.

```
% Correction Options
taptime = 0.075; % taper for seismogram correction step

tf_op = 1; %option for using either daily (1) or station
   average (2) TF in correction

filop = 2; %how to filter TF
% 1 - user specified constant high pass and low pass
% 2 - %lowpass - 0.005+freqcomp, adaptive to the
   infragravity wave cutoff, no high pass;
tap_width = 0.01; %width in percent of frequency vector of
   cosine taper
taper_lim = [0 1]; % upper and lower frequency in Hz limit
   for taper if user specified (option 1); zero means not
   applied
```

taptime - fractional width of taper applied to ends of event time series.

tf_op - transfer function option specification; you can use either the daily transfer functions or the station average for the time period of interest. The options are detailed in the comments.

filop - options for filtering the transfer function. This filters out frequencies where you don't want the data to be corrected, applying the transfer function only in the region of interest. Options are are detailed in comments.

tap_width - fractional width of taper applied to filtering the transfer function around the frequencies specified by **taper_lim**.

taper_lim - frequency range where transfer function is bandpassed in Hz. Only used in the case where **filop** is 1.

This script can output figures showing original event data (Figure 11) and the event data with the corrections applied for all the transfer functions of interest (Figure 12). Options to turn plotting and saving of figures on and off are at the top of **correctevent.m**. The bandpass filter limits for plotting the seismograms are also specified here. You can also choose to overwrite saved corrected seismograms here.

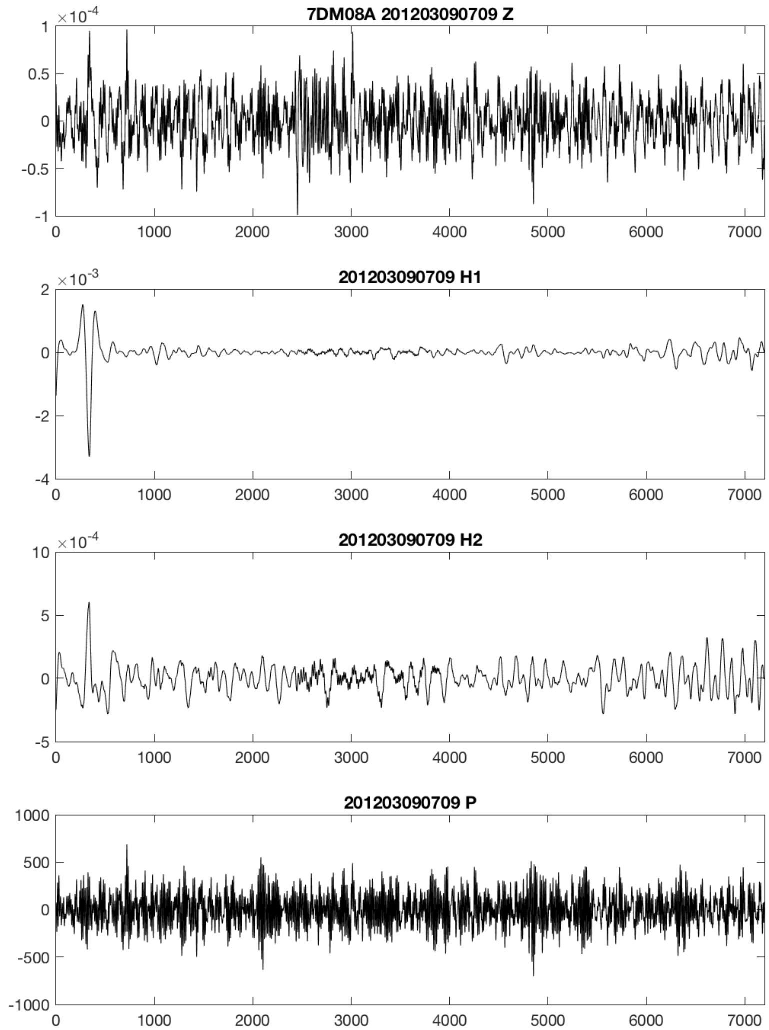

Figure 11: Event time series for the vertical (Z), horizontal 1 (H1), horizontal 2 (H2), and pressure (P) components. No corrections have been applied. The data is for station M08A for the Mw 6.6 earthquake that occurred near Vanuatu on March 9, 2012 and has been bandpass filtered from 10 - 150 s.

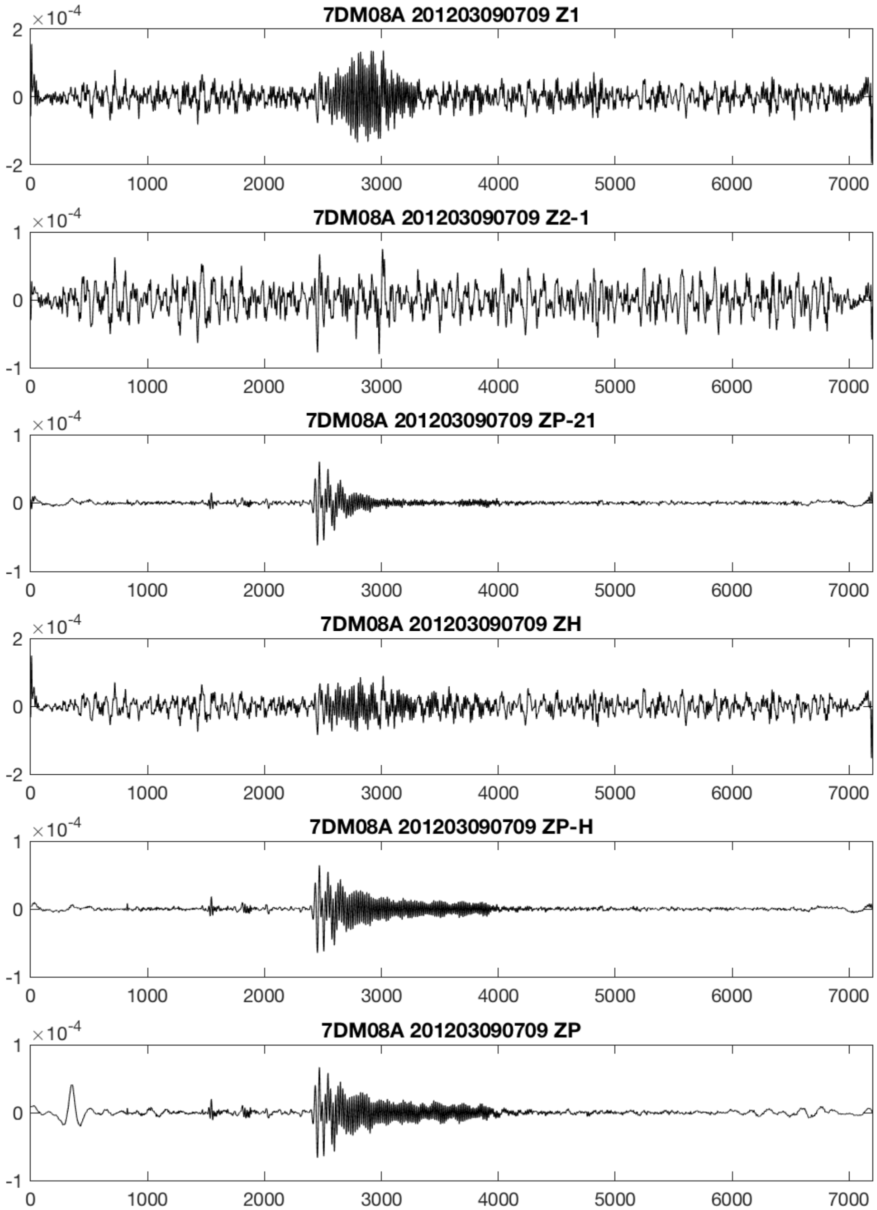

Figure 12: Event time series for the vertical (Z) components after each of the transfer functions of interest have been applied. The corrections are specified in the titles of each subplot. The data is for station M08A for the Mw 6.6 earthquake that occurred near Vanuatu on March 9, 2012 (same as Figure 11) and has been bandpass filtered from 10 - 150 s.

Troubleshooting

Limitations of the Quality Control Method

The quality control steps are based on finding sections of data that have anomalous spectral properties relative to either the rest of the day in the first step, or the rest of the deployment or subset of deployment for the second quality control step. Embedded in this is an assumption noise systematics of each station are stationary such that it represents the spectral properties that can be used to calculate the corrections accurately. This assumption is invalid if components of the station have failed for either the entire deployment or a large section of the deployment. For example, if the instrument fails for half the deployment, the first quality control step will not pick up any bad windows for the daily spectral property calculations, and that bad data will be included in the deployment quality control step and likely not be flagged to be excluded. To avoid these problems, the user should ensure that the stations operate properly over the time period of interest and discard data from times where it was broken.

Instrument Response

While you do not have to remove the instrument response for these calculations, incorrect instrument response corrections can lead to faulty spectra and thus tilt and compliance corrections. We have experienced incorrect phase information in the response files of some OBS that made tilt and compliance corrections impossible. The solution was to turn off the response removal for such stations.

References

Bell, S. W., D. W. Forsyth, and Y. Ruan (2014), Removing noise from the vertical component records of ocean-bottom seismometers: Results from year one of the Cascadia Initiative, Bull. Seismol. Soc. Am., 105, 300-313.

Crawford, W.C., Webb, S.C., (2000). Identifying and removing tilt noise from low-frequency (0.1 Hz) seafloor vertical seismic data, Bull. seism. Soc. Am., 90, 952-963.

Janiszewski, H A, J B Gaherty, G A Abers, H Gao, Z C Eilon, Amphibious surface-wave phase-velocity measurements of the Cascadia subduction zone, Geophysical Journal International, Volume 217, Issue 3, June 2019, Pages 1929-1948, https://doi.org/10.1093/gji/ggz051