Worked example I - cliff coast collapses

Michael Dietze

December 20, 2017

This document is part of the supplementary materials to the article by Dietze (2017).

Introduction

During a pilot study, the cliff coast section of Germany's largest island, Rügen, has been instrumented by four Nanometrics Trillium Compact 120s broadband seismometers and Omnirecs Cube^3 ext data loggers, recording at 200 Hz and powered by 70 Ah batteries. The stations were deployed between April and May 2017 on top of the about 100 m high chalk cliffs, spaced by about one km, in hand-dug pits of about 50 cm depth. The primary purpose of this monitoring campaign was to detect and locate cliff collapses that occur as rock falls, rock avalanches, rotational slides and debris flows.

The network has detected a cliff coast collapse on March 21 2017, 04:38:45 am, which has been confirmed by the National Park staff a day later during a visit of the seismically predicted location.

The raw data used in this document is contained as `data_example_I.rda` and `dem_alos_crop.img` in these supplementary materials. The DEM has been downloaded from the ALOS data portal (http://www.eorc.jaxa.jp/ALOS/en/aw3d30/index.htm), projected to the UTM zone 33 N system, cropped (x-limits 404999.1 – 414989.1 and y-limits 6040011 – 6050001) and stored as `dem_alos_crop.img`.

Analysis script preparation

The analysis is initiated by loading all required R packages, defining the working directory and loading the station information file, which has been created earlier and is not shown in this document. For details see R file `~/Documents/projects/Ruegen/R/prepare_raw_Data.R` and code snippets in the linked article (Dietze, 2017). These also contain the code used to convert the raw cube files to sac files and organise them in the coherent structure used by the 'eseis' package.

```
## load packages
library("eseis")
library("raster")
library("magrittr")

## set working directory
setwd(dir = "~/Documents/projects/Ruegen/data/seismic/")

## load station info table
stations <- read.table(file = "station/station_info_RUEG17_utm.txt",
                       header = TRUE,
                       stringsAsFactors = TRUE)

## load DEM
dem <- raster::raster(x = "../geodata/dem/dem_alos_crop.img")
```

```
## Loading required package: sp

##
## Attaching package: 'magrittr'
```

```
## The following object is masked from 'package:raster':
##
##     extract
```

Session information

For a full picture of the R session in which the analysis takes plase, all relevant information is displayed.

```
sessionInfo()
```

```
## R version 3.4.3 (2017-11-30)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 16.04.3 LTS
##
## Matrix products: default
## BLAS: /usr/lib/libblas/libblas.so.3.6.0
## LAPACK: /usr/lib/lapack/liblapack.so.3.6.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] magrittr_1.5 raster_2.6-7 sp_1.2-5     eseis_0.4.0
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.14      lattice_0.20-35   XML_3.98-1.9
##  [4] digest_0.6.12     rprojroot_1.2     grid_3.4.3
##  [7] backports_1.1.1   evaluate_0.10.1   stringi_1.1.6
## [10] IRISSeismic_1.4.6 rmarkdown_1.8     rgdal_1.2-16
## [13] tools_3.4.3       stringr_1.2.0     yaml_2.1.15
## [16] compiler_3.4.3    htmltools_0.3.6   knitr_1.17
```

Import of the seismic data of the event

The vertical components of all four seismic stations are imported for the event with a few seconds buffer at both sides to account for processing artefacts and the different arrival times of the seismic wave at the stations. The imported datasets are simplified to a list containing only the vertical components as `eseis` objects.

```
## import data sets of the event
x <- aux_getevent(start = as.POSIXct(x = "2017-03-21 04:38:45",
                                     tz = "UTC"),
                  duration = 20,
                  station = stations$ID,
                  component = "BHZ",
                  dir = "sac/")
```

```
## simplify data structure
x <- lapply(X = x, FUN = function(x) {x[[1]]})
```

Data preparation

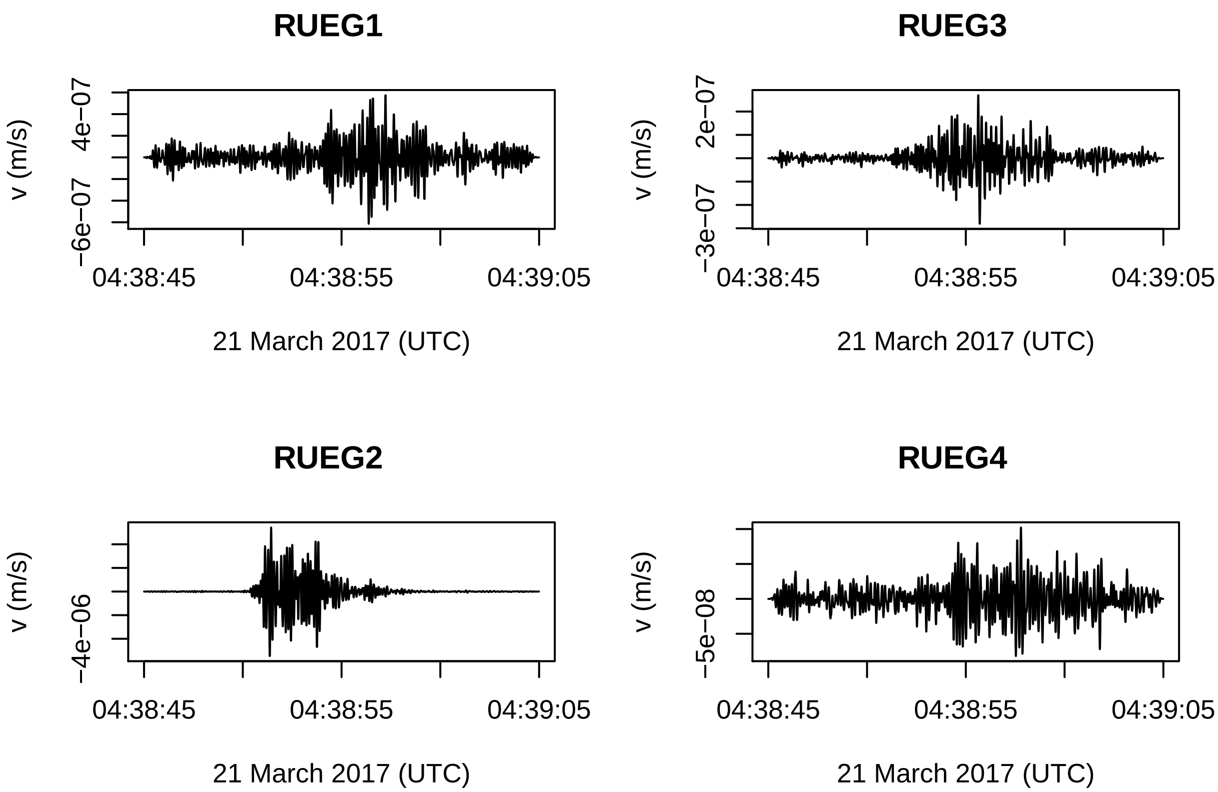
The imported files are deconvolved to account for the instrument response and yield values in m/s. For further processing it is necssary to detrend the data set. The seismic traces are then bandpass filtered between 5 and 8 Hz, using a Butterworth filter. Finally, to account for the filter effects, the signals are tapered at both ends by 300 samples.

```
## deconvolve signals
x <- signal_deconvolve(data = x,
                       sensor = "TC120s",
                       logger = "Cube3extBOB",
                       gain = 1)

## detrend signals
x <- signal_detrend(data = x)

## filter signals
x_f <- signal_filter(data = x,
                     f = c(5, 8))

## taper signals
x_f <- signal_taper(data = x_f,
                    n = 300)
```

Visual inspection of the signal waveforms

For a brief visual inspection, the four seismic waveforms are plotted with user-adjusted plot labels.

```
## setup plot area to have two rows and two columns of plots
par(mfcol = c(2, 2))

## apply the plot_signal function to all four seismic traces
plots <- lapply(X = x_f, FUN = function(X) {
  plot_signal(data = X,
              main = X$meta$station,
              xlab = "21 March 2017 (UTC)",
              ylab = "v (m/s)",
              format = "%H:%M:%S")
})
```

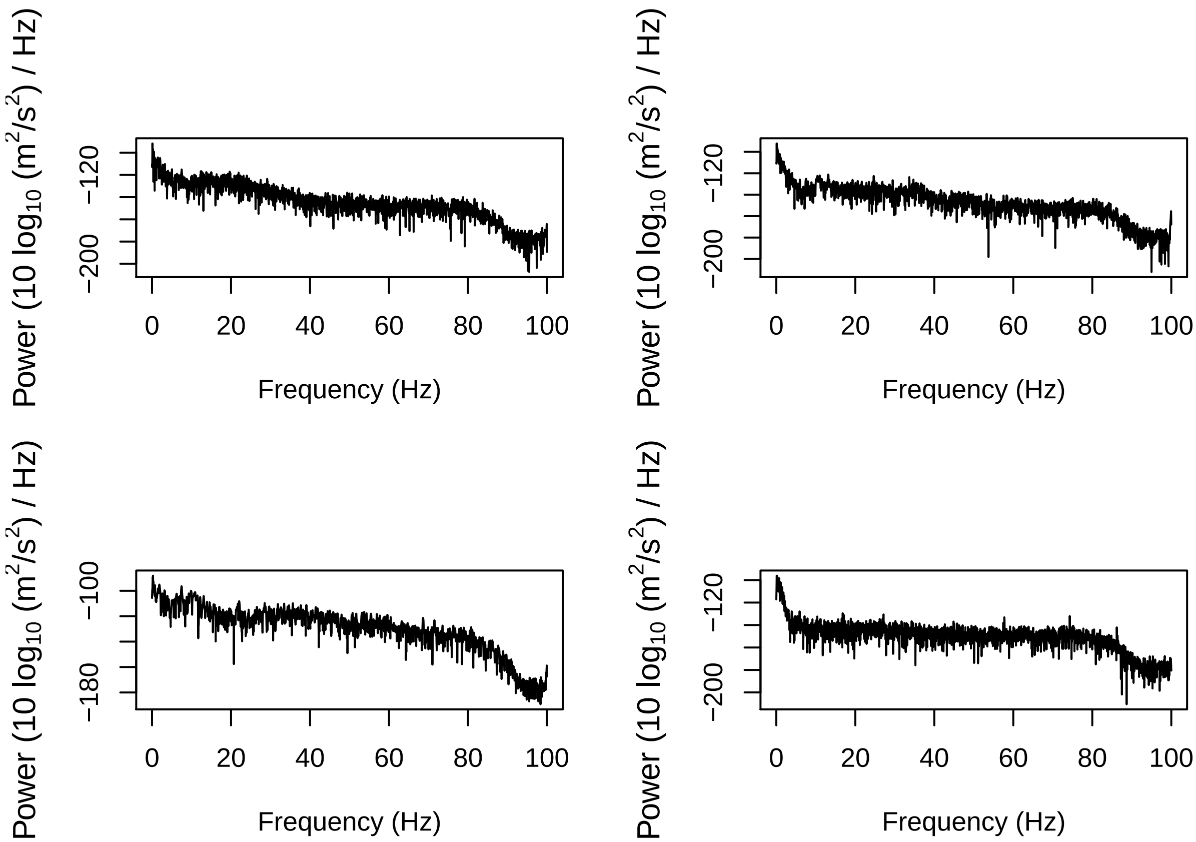


Data analysis - spectral properties

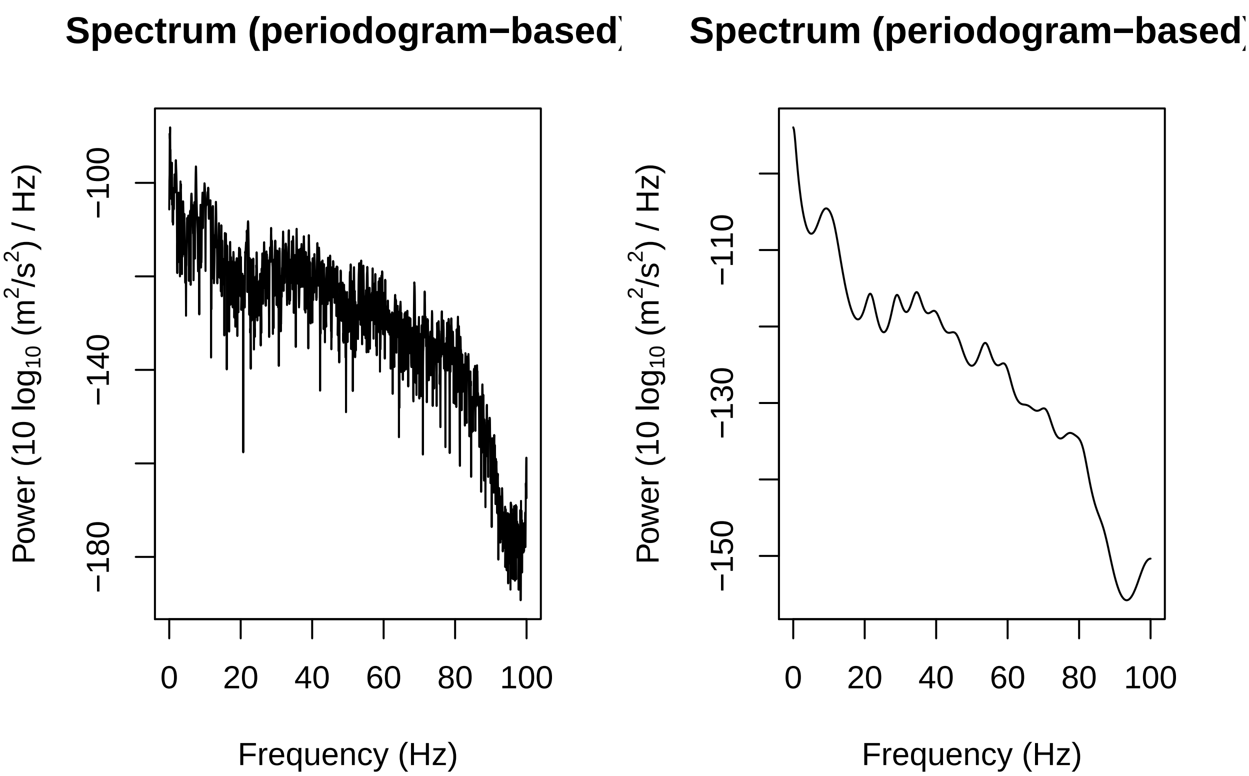
The seismic data are analysed by exploring their spectral properties, by spectra that cover the full event duration and by spectrograms that show the time evolution of the event as 95 % overlapping windows of 1 s, which are in turn calculated from overlapping averages of 0.7 s long windows, i.e., the Welch method (Welch, 1967). The results are visualised as default spectra of all four seismic stations, a comparison of the the specra types for station RUEG2, a spectrogram and a probabilistic spectrum.

```
## calculate spectra of the event with the default method
s_1 <- signal_spectrum(data = x)

## calculate spectra of the event with the autoregressive method
s_2 <- signal_spectrum(data = x,
                       method = "autoregressive")

## calculate a spectrograms of the event
s_3 <- signal_spectrogram(data = x,
                          Welch = TRUE,
                          window = 1.0,
                          window_sub = 0.7,
                          overlap = 0.95,
                          overlap_sub = 0.95)

## setup plot area
par(mfcol = c(2, 2))

## plot default spectrum of all four stations
plot_spectrum(data = s_1,
```

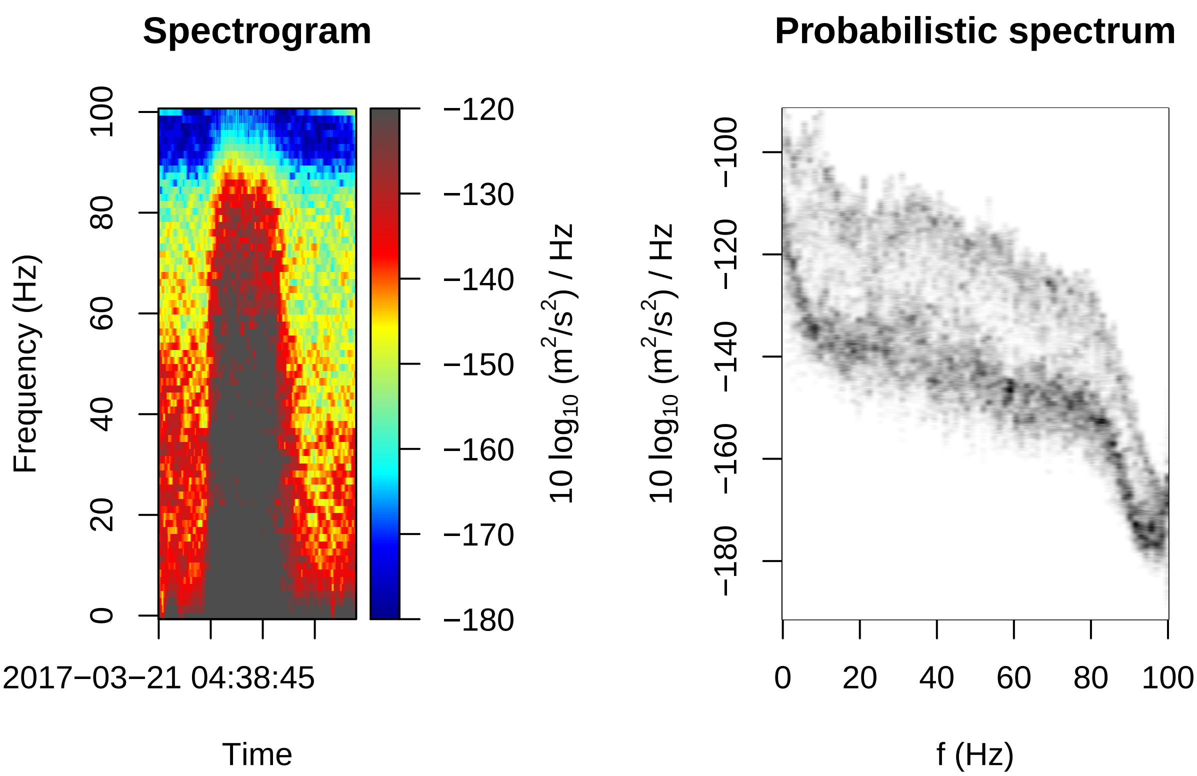
```
              main = "Spectrum (periodogram-based)")
```




```
## setup plot area
par(mfcol = c(1, 2))

## plot default spectrum of all four stations
plot_spectrum(data = s_1$RUEG2,
              main = "Spectrum (periodogram-based)")

## plot default spectrum of all four stations
plot_spectrum(data = s_2$RUEG2,
              main = "Spectrum (periodogram-based)")
```




```r
## plot spectrogram
plot_spectrogram(data = s_3$RUEG2$PSD,
                 zlim = c(-180, -120),
                 main = "Spectrogram",
                 legend = TRUE)

## plot probabilistic spectrum
plot_ppsd(data = s_3$RUEG2$PSD,
          main = "Probabilistic spectrum")
```



Seismic source location

Preparation

To use the signal migration approach it is necessary to have distance maps, lookup tables of the topography-corrected distance between a seismic station and any pixel of a grid. These are calculated from a freely available ALOS DEM for all four seismic stations of the Rügen network.

```
## create distance maps
D <- spatial_distance(stations = stations[,3:4],
                      dem = dem)
```

Also, the seismic signals must be filtered to the frequency window of interest/relevance. Based on the spectral information from above the frequency window is set to 5-10 Hz. The filtered signals are tapered afterwards to account for artefacts. The workflow is performed with the piping symbol from the 'magrittr' package.

```
## create distance maps
s_f_loc <- signal_filter(data = x, f = c(5, 10)) %>% signal_taper(n = 300)
```

Location

The prepared signals of the cliff collapse event are migrated to identify the probability density of the source location using the 60m distance maps and an apparent seismic wave velocity of 600 m/s. For illustrative reasons, the output is clipped to values above the 0.99 quantile.

```
## locate event source
xy <- spatial_migrate(data = s_f_loc,
                      d_stations = D$stations,
```

```
                      d_map = D$maps,
                      v = 600)
```

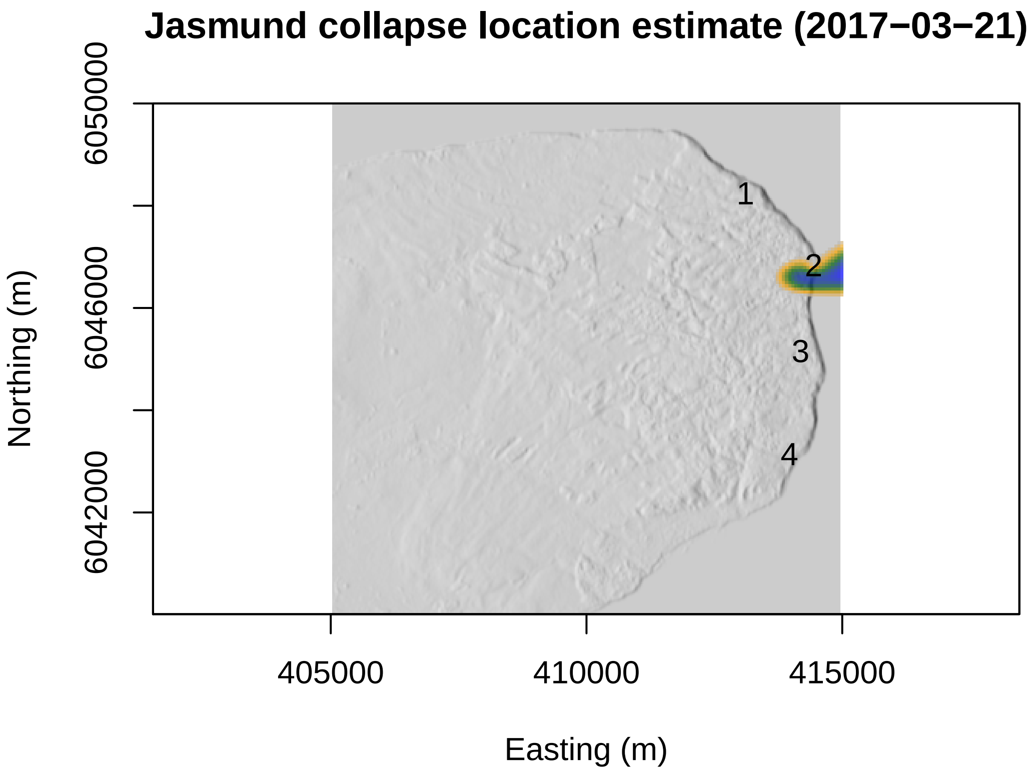
```
## [1] "No snr given. Will be calculated from signals"
```

```
## clip estimate to 0.99 percentile
xy_clip <- spatial_clip(data = xy,
                        quantile = 0.99)
```

Visualisation

For graphical output the DEM is used to generate a hillshade model. Over this hillshade, the clipped location estimate is draped with semitransparent colours. The seismic station locations are added, as well. For plotting a user-defined colour palette is used.

```
## plot hillshade and location estimate on top
hs <- hillShade(
  slope = terrain(x = dem, opt = "slope"),
  aspect = terrain(x = dem, opt = "aspect"),
  angle = 40,
  direction = 270
)

## define colour palette
gygb <- colorRampPalette(c("grey",
                           "orange",
                           "darkgreen",
                           "blue"))

## plot hillshade
raster::plot(hs,
     col = grey.colors(200),
     legend = FALSE,
     xlab = "Easting (m)",
     ylab = "Northing (m)",
     main = "Jasmund collapse location estimate (2017-03-21)")

## overlay seismic location estimate
raster::plot(xy_clip,
     add = TRUE,
     col = adjustcolor(col = gygb(200),
                       alpha.f = 0.7),
     legend = FALSE)

## add seismic station locations
points(x = stations$x,
       y = stations$y,
       pch = as.character(1:4))
```



References

Dietze, M., 2017. The R package ‘eseis’ – a comprehensive toolbox for environmental seismology. Submitted to Earth Surface Dynamics. [December 2017].

Welch, P., 1967. The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms, IEEE T. Acoust. Speech, 15, 70–73.