Dataset description for Surface waves prediction based on acoustic backscattering

D.Razumov, A. Ermoshkin, D. Kosteev, A.Ponomarenko, M.Salin

Data collected: 15 Sep - 1 Oct, 2021; published: May, 2022

Common description. Acoustic measurements dataset is represented by three types of files “.raw”, “.mat”, ”.dat”

1. “.raw” format also represented by three types of data.
   - “...search.raw” files contain complex envelop from all hydrophones calculated at four emitted frequencies
   - “...chan.raw” files is a signal in a wide band from one of the hydrophones - for control and noise analysis.
   - the largest files are the raw wideband signal from all hydrophones. One such file was saved per eight-hour sound emission cycle.
2. Spectrogram files are saved in MATLAB format “.mat” v7 . Phasing of the antenna array (all-round view) and calculation of window spectra near each emitted pulse was carried out.
3. ”.dat” files contain features of the average spectrum of the backscattered signal.
4. Acoustic measurements were accompanied by direct measurements of surface wave characteristics using a Datawell DWR-G4 buoy.

Users should refer to the published preprint for a description of the experimental conditions and instrumentation characteristics. https://arxiv.org/abs/2204.10153

In this archive, we upload all available files of the ”dat” and ”mat” type and a limited number of ”raw” files. For a complete set of ”raw”, please contact the authors at the emails
ddrazumov@ipfran.ru or mikesalin@ipfran.ru

Buoy data format

The buoy data is recorded in the file “bouy21.xlsx” in a microsoft excel format. Each entry in the table was acquired within a 30-minute interval, which corresponds to the fragmentation of the acoustic data. Of the available columns, the study used:

Hm0 - significant wave height (cm)
Dirp - wave direction (degrees)
Tz is the period corresponding to the spectral peak
DateTime - UTC date and time

Note: Users should pay particular attention to the fact that data is in UTC and acoustic data is in UTC+3

features and spectrogram data The first folders correspond to the frequency at which the signal was emitted and processed. The next level is the data type: features or spectrogram. Inside there are folders whose names are days in September 2021. The file name is the end time of the signal recording. The duration of the recording is about an hour, for ".dat" files this hour is divided into 30min halves: half1 and half2.

mat files contain the next variables:

1. angles [1D array] - angles of phasing -90:30:90
2. F3 [1D array] - scale of Doppler frequencies
3. T3 [1D array] - scale of time delays. The reverberation signal for which the features are calculated goes on index 15:1:22 for 2s pulse duration and 33:1:40 for 8s pulse duration on 2080Hz.
4. Nk[int] - length of angles
5. N[int] - number of pulses
6. Nexcep2[2D array NxNk] - table of excluded pulses
7. P_A_N_dB[4D array size(T3)xsize(F3)x(N+1)xNk] - Power Spectral Density in logarithmic scale of received pulses depending on pulse number and phasing angle. The last (:,:,N+1,:) is averaging over pulses with exception.
8. Pm1_dB[3D array size(T3)xsize(F3)xNk] - spectrogramm averaged by pulse number over first half
9. Pm2_dB[3D array size(T3)xsize(F3)xNk] -"- over second half
10. car_freq [double] exact value of heterodyne frequency
11. t_before & t_after are times in second before and after pulse begging, Tp_ang are times of pulse beggings

dat files contain a table in ASCII format. Columns are angles. First column correspond to -90 phasing angle, second to -60 ... the last to +90. Each 8 rows correspond to a feature for increasing time delays. First 8 rows contain sk_le feature, rows from 9 to 16 - sk_ri, next follow centr1, centr2, lvl, sk_le2, sk_ri2, lv_brag_le, lv_brag_ri.

Let us remind features meaning:

1. sk_le [dB/Hz] slope from the left, the coefficient of approximation of the polynomial of the first degree by the least squares method of the line section in Fig. 1 in the range [-1.75; -0.75] Hz, must be positive.
2. sk_ri [dB/Hz] slope to the right, -"- in the range of [0.75; 1.75] Hz, -"- negative
3. centr1 [Hz] weighted average of frequency
4. centr2 [Hz] average median value of frequency
5. lvl [dB] signal level
6. sk_le2[dB/Hz] - second slope for negative Doppler frequencies in the range [-3.2; -1.45] Hz for 2kHz ; [-2.7;-1.45] for 1.3kHz ; [-3;-1.45] for 2.7 kHz.
7. sk_ri2[dB/Hz] - second slope for positive Doppler frequencies

8. lv_brag_le [dB] - signal level for negative Doppler frequencies in the range of 1Hz around the Bragg frequency -2±0.5Hz for 2kHz; [-2.2;-1.18] for 1.3kHz; [-2.92;-1.9] for 2.7kHz
9. lv_brag_ri[dB] - signal level for positive Doppler frequencies in the range of 1Hz around the Bragg frequency

description of "RAW" data format

The received acoustic signals are recorded in the RAW format, which is the de facto internal standard of the Department of Physical Acoustics of the IAP RAS. At the beginning of the file is a header, which is described in Listing 1 in C terms. The header contains information about the number of channels, sampling rate, data format, and more. In addition, the header contains the "data_offset" value, the offset from the beginning of the file in bytes, starting from which the values of the digitized signal go directly in the file. The vector of values for all channels $x_1, x_2...x_N$, obtained at one time t_m, is stored continuously. Those, all digitized data is stored as a sequence:

$p(x_1,t_1), p(x_2,t_1), ..p(x_N,t_1)$

$p(x_1,t_2), p(x_2,t_2), ..p(x_N,t_2)$

In addition, an example of a Matlab/Octave program that reads a raw file is attached, see Listing 2. Channel permutation needs to be applied to the data. The file "oldant.txt" contains a table: 1st column - hydrophone number in order, 2nd column - channel number in the file. The data folder also contains ".raw.config" files, which contain service information and are not of interest for signal processing.

Listing 1. C++ example

```
#pragma pack(push,1)
struct Raw_head
        {public:
        char Version[16];    /* file tag RAWDATA\0              */
        char Date[20];       /* date/time of file begging       */
        char comment[476];   /* comment(text)                   */
        unsigned int dos_ftime; /* date/time in DOS format   */
        short date[3],time[3]; /* date/time YY/MM/DD, HH/MM/SS */
        short nchan;           /* number of channels            */
        short ADCoffset;       /* shift of AD converter         */
        unsigned short ADCmask;  /* mask of ADC                 */
        float Ftakt;         /* sample rate (Hz/channel)      */
        float F0;            /* heterodyne frequency (Hz)   */
        float ADCscale;      /* scale factor, reduction to V*/
        short type;            /* 1 - wide band,
                        > 1 - number of quadratures, 0 - sweep  */
        short word;            /* 2 - short, 3 - long,        */
                             /* 4 - float, 5 - double;            */
        int data_offset;    /* header length                */
        short min,max;
        short f0[5];
        };

#pragma pack(pop)
```

Listing 2. Matlab/Octave example, reading wideband raw file

```
function read_raw_file

fname='D:\RAW\2016_06_16\vert_rezh1_first_hour.raw';

f = fopen(fname, 'rb');
if (f==-1); error ('Can''t␣open␣RAW␣file!'); end

%---header---
%nchan - number of channels
fseek(f,528,'cof');
nchan=fread(f,1,'int16');
%ftakt - sample rate
fseek(f,4,'cof');
ftakt=fread(f,1,'float32');
%data_offset - sets a pointer in the file to the beginning of the data.
data_offset=fread(f,1,'int32');
fseek(f,data_offset,'bof');

% choose a reasonable length of data portion for reading
n_read_pack = 1024000;

%---Read data until file ends---
while(1)
    data = fread(f, [nchan, n_read_pack], 'float32');
    if isempty(data); return; end
end
```

Listing 3. Python example, reading mat-files and reproducing fig. 8 in paper

```
import math
import numpy as np
import matplotlib.pyplot as plt
import scipy.io as ios

# Hint: check the path
home = '/put/your/path/here/2020Hz/curspec_mat/'

def spec_from_file(fname, ofs, linestyle, label_, dest=None):
    global fcar
    if dest is None: dest=plt.gca()
    print(fname)
    data = ios.loadmat(fname)
    P_A_N_dB = data['P_A_N_dB']
    F3 = data['F3']
    car_freq = data['car_freq']
    fcar=2020
    for vf in [1320, 2080, 2720]:
        if abs(car_freq - vf) < 5:
```

```
            fcar = vf
    dest.plot(F3 - (fcar-car_freq), P_A_N_dB[ofs, :, -1, 1],
              linestyle, label=label_)

ofs03355 = 14

plt.rcParams.update({'font.size': 7})
cm = 1/2.54  # centimeters in inches
[fig, ax] = plt.subplots(1,2, figsize=(19*cm, 8*cm))

spec_from_file(home + '22/20_00.mat', ofs03355, 'r--', '-0.2␣Hz', ax[0])
spec_from_file(home + '28/12_00.mat', ofs03355, 'k-',  '0.0␣Hz', ax[0])
spec_from_file(home + '23/07_00.mat', ofs03355, 'g--', '0.2␣Hz', ax[0])
spec_from_file(home + '21/10_00.mat', ofs03355, 'b-.', '0.5␣Hz', ax[0])
ax[0].legend()
ax[0].set_title('(a)␣various␣values␣of␣centr1')

ax[0].set_xlim(-6,6)
ax[0].set_ylim(-70,-30)
fBragg = math.sqrt(4*math.pi*fcar/1450*10) / (2.*math.pi)
ax[0].plot([-fBragg,-fBragg], [-65,-40], 'k--')
ax[0].plot([fBragg,fBragg], [-65,-40], 'k--')
ax[0].set_xlabel('Doppler␣frequency␣(Hz)')
ax[0].set_ylabel('Backscattering␣signal␣PSD␣(dB)')

spec_from_file(home + '22/22_13.mat', ofs03355, 'r--', '-14␣dB/Hz', ax[1])
spec_from_file(home + '23/23_00.mat', ofs03355, 'k-',  '-18␣dB/Hz', ax[1])
spec_from_file(home + '24/04_00.mat', ofs03355, 'b-.', '-24␣dB/Hz', ax[1])
ax[1].legend()
ax[1].set_title('(b)␣various␣double_skew')

ax[1].set_xlim(-6,6)
ax[1].set_ylim(-70,-30)
fBragg = math.sqrt(4*math.pi*fcar/1450*10) / (2.*math.pi)
ax[1].plot([-fBragg,-fBragg], [-65,-40], 'k--')
ax[1].plot([fBragg,fBragg], [-65,-40], 'k--')
ax[1].set_xlabel('Doppler␣frequency␣(Hz)')

plt.show()

# uncomment above and comment below to save as pdf

#fig.savefig('extreme_and_mean_spectra.pdf', format='pdf',
#            pad_inches=0.1, bbox_inches='tight')
```