
Gridded COVIS Diffuse-mode data

Table of Contents

The content of a COVIS data file: the COVIS structure	1
Walking through the fields of the covis structure	1
COVIS location and related metadata	2
Processing metadata	3
The gridded data itself	6
Version control	7
Information on sonar setup and data acquisition	7
Diagnostics	9

This file provides a description of the COVIS gridded data format specific to the Diffuse mode data. The names of COVIS data files contain useful metadata, most of which is also inside the file. The time of data collection can be found after the prefix 'COVIS-' in the format: YearMonthDayTHourMinuteSeconds. The suffix following the time stamp is a code word indicating the data acquisition mode (e.g., imaging, diffuse). The number at the end of the modes indicates the sector to which the sonar head was directed. For example, "-diffuse1" indicates Diffuse-mode data collected from sector 1.

The content of a COVIS data file: the COVIS structure

inside the COVIS structure is all the data and metadata needed for analysis and visualization

```
covis
```

```
covis =
```

```
    struct with fields:
```

```
        type: 'diffuse'
        comments: 'ASHES at Axial'
        user: [1x1 struct]
        sonar: [1x1 struct]
        processing: [1x1 struct]
        grid: {1x7 cell}
        release: '2.0'
        sweep: [1x1 struct]
        ping: [1x159 struct]
        bad_ping: []
```

Walking through the fields of the covis structure

This and following sections cover general information about COVIS, background information on raw data acquisition and processing as well as the output grid values and coordinates.

covis.type includes the acquisition mode (e.g., 'imaging', 'diffuse'); this information is also in the filename: a *-diffuse1.mat, *-diffuse2.mat, or *-diffuse3.mat file was acquired using the Diffuse mode of COVIS and should have a type of 'diffuse' regardless of the sector number.

```
covis.type
```

```
ans =
```

```
    'diffuse'
```

covis.comments lists the site at which the data was acquired (e.g., ASHES at Axial)

```
covis.comments
```

```
ans =
```

```
    'ASHES at Axial'
```

covis.user contains information about user-related settings, such as whether to report maximum (verbose=2), medium (verbose=1), or minimum (verbose=0) amount of information during raw data processing; The 'debug', 'view', and 'outpath' fields are obsolete and will be removed in the next release.

```
covis.user
```

```
ans =
```

```
    struct with fields:
```

```
    verbose: 1
    debug: 0
    view: [1x1 struct]
    outpath: 'nan'
```

COVIS location and related metadata

The positioning of COVIS at the time of data acquisition is reported in *covis.sonar.position*. If the value of a field is 0, that field was not set during processing and thus assigned with a null value 0.

```
covis.sonar.position
```

```
ans =
```

```
    struct with fields:
```

```
    easting: 0
    northing: 0
    depth: -1500
    altitude: 4.2000
```

declination: 16

- 'easting' and 'northing': nominal location of COVIS in UNM coordinates in meters.
- 'depth': nominal depth of the feet of the platform in meters.
- 'altitude': height of the sonar heads above the feet in meters.
- 'declination': magnetic declination at COVIS's location in degrees.
- 'heading': direction of the central axis of the sonar in degrees.

Processing metadata

covis.processing reports the metadata used to process raw data into gridded data.

```
covis.processing
```

```
ans =
```

```
struct with fields:
```

```
    beamformer: [1x1 struct]
    calibrate: [1x1 struct]
    filter: [1x1 struct]
    ping_combination: [1x1 struct]
    correlation: [1x1 struct]
    averaging: [1x1 struct]
    snr: [1x1 struct]
```

The inputs to the beamforming process are listed in *covis.processing.beamformer*.

```
covis.processing.beamformer
```

```
ans =
```

```
struct with fields:
```

```
    type: 'fast'
    fc: 200000
    c: 1.4845e+03
    fs: 3.4483e+04
    first_samp: 1
    last_samp: 1644
    start_angle: -64
    end_angle: 64
    array_length: 0.4080
    num_beams: 128
    angle: [1x128 double]
    range: [1644x1 double]
```

- 'type': beamforming method used.
- 'fc': central frequency (Hz)
- 'c': sound speed (m/s)
- 'fs': sampling rate (Hz)
- 'first_samp', 'last_samp': no. of first and last samples
- 'start_angle', 'end_angle': azimuthal range for beamforming (degree)
- 'array_length': physical length of the receiver array of the sonar (m)
- 'num_beams': number of beams formed
- 'angle': azimuths of formed beams (radian)
- 'range': sonar range (m)

covis.processing.calibrate lists the calibration method used.

`covis.processing.calibrate`

`ans =`

```
struct with fields:
    mode: 'TS-Wide'
    filt: 1
    filt_bw: 2
```

covis.processing.filter indicates whether filtering was done and lists the method of filtering and its inputs.

`covis.processing.filter`

`ans =`

```
struct with fields:
    status: 'on'
    type: 'butterworth'
    bw: 2
    order: 4
    decimation: 1
```

covis.processing.ping_combination indicates the basic method of handling the combination of pings. 'diff' refers to a ping-to-ping differencing technique.

`covis.processing.ping_combination`

`ans =`

```
struct with fields:
```

```
mode: 'nan'
```

Parameters used to calculate ping-to-ping correlation

```
covis.processing.correlation
```

```
ans =
```

```
struct with fields:
```

```
window_size: 1.0000e-03  
window_overlap: 0.5000  
windthresh: 1.0000e-03  
nlag: 4  
tlag: 2
```

- 'window_size': length of the time window (in milliseconds)
- 'window_overlap': overlap ratio between two adjacent windows
- 'windthresh': this parameter is obsolete
- 'nlag': this parameter is obsolete
- 'tlag': time lag between pings in seconds

length of the time window for calculating time-averaged variables in seconds

```
covis.processing.averaging.win
```

```
ans =
```

```
4
```

Parameters used to mask out raw data points with low signal-to-noise ratios (SNR) are included in *covis.processing.snr*.

```
covis.processing.snr
```

```
ans =
```

```
struct with fields:
```

```
noise_floor: 0.6400  
threshold: 45
```

- 'noise_floor': noise level in uncalibrated machine units

- 'threshold': cutoff in dB for masking out data points with low SNRs (<threshold)

The gridded data itself

covis.grid contains the output gridded data, grid coordinates, and the basic gridding parameters. There are currently seven types of gridded data, each stored in an element of the grid cell array. Different elements share the same grid coordinates and gridding parameters.

```
covis.grid
```

```
ans =
```

```
1x7 cell array
```

```
Columns 1 through 4
```

```
{1x1 struct} {1x1 struct} {1x1 struct} {1x1 struct}
```

```
Columns 5 through 7
```

```
{1x1 struct} {1x1 struct} {1x1 struct}
```

```
covis.grid{1}
```

```
ans =
```

```
struct with fields:
```

```

    type: 'decorrelation intensity'
    shape: 'rectangular'
    units: [1x1 struct]
dimensions: 2
    bounds: [1x1 struct]
    spacing: [1x1 struct]
        x: [121x121 double]
        y: [121x121 double]
        v: [121x121 double]
        w: [121x121 double]
    size: [121 121]
    axis: [-30 30 -30 30 0 0]
    name: 'COVIS-20200417T003002-diffuse1'
```

- 'type': type of data value (e.g., 'decorrelation intensity').
- 'shape': shape of the grid (rectangular).
- 'units.spatial': units of grid coordinates.
- 'units.value': units of data values (1/meters for volume backscattering cross-section).
- 'dimensions': number of grid dimensions.

- 'bounds': lower and upper bounds of each coordinate axis.
- 'spacing': spacing between grid points on each coordinate axis.
- 'x','y','z': coordinates of grid points in each dimension.
- 'v': data value,
- 'w': weight function for interpolation of raw data onto the grid.
- 'size': number of grid points on each coordinate axis.
- 'axis': vectors containing the lower and upper bounds of coordinate axes.
- 'name': name of the data file.

Version control

covis.release gives the version no. of the source code used to create the gridded data

```
covis.release
```

```
ans =
```

```
    '2.0'
```

Information on sonar setup and data acquisition

covis.sweep describes how and when the data were acquired.

```
covis.sweep
```

```
ans =
```

```
struct with fields:
```

```
    mode: 'diffuse1'  
    alpha__id: 'COVIS-20200417T003002-diffuse1'  
    starttime: {[1.5871e+09] [738566]}  
    endtime: {[1.5871e+09] [84268]}  
    motion: [1x1 struct]  
    settings: [1x1 struct]  
    path: 'C:\COVIS\Axial\COVIS_data\raw\raw_data_combine  
2020\04\17'  
    name: 'COVIS-20200417T003002-diffuse1'
```

- 'mode': acquisition mode and sector no. (e.g., imaging1).
- 'alpha_id': the file name.

- 'starttime','endtime': start and end times of the scan. The format is: first value + second value/10⁶ = seconds since 1/1/1970 (or Unix Time).
- 'motion': the motion of the sonar head during acquisition.
- 'settings': the settings of the sonar during acquisition.
- 'path' and 'name': temporary path on the upstream server and file name.

covis.sweep.motion gives the motion of the sonar head during acquisition.

```
covis.sweep.motion
```

```
ans =
```

```
struct with fields:
```

```
start: {[155] [241.6000] [135]}
inc: {[0] [0] [0]}
steps: 1
roundtrip: 0
level: 0
```

- 'start': starting roll, pitch, and yaw in terms of the motor positions.
- 'inc': increments to change roll, pitch and yaw at each scan step ('0' when the corresponding motor position remains unchanged during a scan).
- 'steps': number of scan steps.
- 'roundtrip': '0' indicates the driving motor moves in one direction during a scan (e.g., pitch up or down); '1' indicates the driving motor moves in both directions during a scan (e.g., pitch up and down).
- 'level': '0' indicates the sonar head was leveled before starting the scan. '1' indicates no leveling was done before the scan.

covis.sweep.settings lists the changeable sonar settings, including the gain, power level, pulse width, and clipping range.

```
covis.sweep.settings
```

```
ans =
```

```
struct with fields:
```

```
rxgain: 30
txpower: 200
pulse_width: 3.0000e-04
range: 35
```

covis.ping contains all the stored metadata in raw data. For each ping, it lists the following:

```
covis.ping
```



```
ans =  
  
1x159 struct array with fields:  
  
    num  
    sec  
    rot_pitch  
    sen_pitch  
    rot_roll  
    sen_roll  
    rot_yaw  
    sen_head  
    hdr
```

- 'num': ping number.
- 'sec': time stamp (fractional seconds since 1/1/1970 or Unix Time)
- 'rot_pitch', 'rot_roll', 'rot_yaw': motor positions.
- 'sen_pitch', 'sen_roll', 'sen_head': the sensor-reported position of the sonar head in degrees.
- 'hdr': a detailed list of sonar information.

Diagnostics

covis.bad_ping contains the ping no. of those that are not processable

```
covis.bad_ping
```

```
ans =  
  
[]
```

Published with MATLAB® R2019a