Gridded COVIS Imaging-mode data

Table of Contents

This file provides a description of the COVIS gridded data format specific to the Imaging 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, "-imaging1" indicates Imaging-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: 'imaging'
      comments: 'ASHES at Axial'
          user: [1×1 struct]
         sonar: [1×1 struct]
    processing: [1×1 struct]
          grid: [1×1 struct]
       release: '2.0'
         sweep: [1×1 struct]
          ping: [1×365 struct]
         burst: [1×73 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 *-imaging1.mat, *-imaging2.mat, or *-imaging3.mat file was acquired using the Imaging mode of COVIS and should have a type of 'imaging' regardless of the sector number.

```
covis.type


ans =

    'imaging'
```

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: 2
      debug: 0
       view: [1×1 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
        heading: 0
```

- '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 degres.

Processing metadata

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

```
covis.processing
```

```
ans =

  struct with fields:

         beamformer: [1×1 struct]
          calibrate: [1×1 struct]
             filter: [1×1 struct]
   ping_combination: [1×1 struct]
             bounds: [1×1 struct]
                snr: [1×1 struct]
             oscfar: [1×1 struct]
```

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

```
covis.processing.beamformer
```

```
ans =

  struct with fields:

        type: 'fast'
          fc: 396000
           c: 1.4845e+03
          fs: 3.4483e+04
```

```
       first_samp: 1
        last_samp: 3522
      start_angle: -54
        end_angle: 54
      array_length: 0.4080
        num_beams: 256
            angle: [1×256 double]
            range: [3523×1 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: 'VSS'
```

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.processsing.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'
```

The spatial boundaries of raw data used in griding is reported in sonar centric coordinates in *covis.processing.bounds*. If the value of a field is 0, that field was not set during processing and thus assigned with a null value 0.

```
covis.processing.bounds
```

```
ans =

  struct with fields:

      pitch: [1×1 struct]
    heading: [1×1 struct]
      range: [1×1 struct]
```

- 'pitch': start and end pitch angles in degrees

```
covis.processing.bounds.pitch
```

```
ans =

  struct with fields:

    start: -20
     stop: 62
```

- 'heading': start and end headings in degrees

```
covis.processing.bounds.heading
```

```
ans =

  struct with fields:

    start: 0
     stop: 0
```

- 'range': start and end ranges in meters

```
covis.processing.bounds.range


ans =

  struct with fields:

    start: 0
     stop: 0
```

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: 0.1970
      threshold: 50
```

- 'noise_floor': noise level in uncalibrated machine units

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

Parameters in the OS-CFAR method for eliminating side-lobe noises in raw data are included in *covis.processing.oscfar.*

```
covis.processing.oscfar


ans =

  struct with fields:

        clutterp: 65
    scrthreshold: 20
```

- 'clutterp': percentile (%) corresponding to the cutoff value used to identify clutters

- 'scrthreshold': cutoff in dB for masking out data points with low signal-to-clutter ratios

The gridded data itself

covis.grid contains the output gridded data, grid coordinates, and the basic gridding parameters.

```
covis.grid


ans =
```

```
struct with fields:

          type: 'backscatter-cross-section'
         shape: 'rectangular'
         units: [1×1 struct]
    dimensions: 3
        bounds: [1×1 struct]
       spacing: [1×1 struct]
             x: [241×241×81 double]
             y: [241×241×81 double]
             z: [241×241×81 double]
            Ia: [241×241×81 double]
            Id: [241×241×81 double]
       Ia_filt: [241×241×81 double]
       Id_filt: [241×241×81 double]
            Kp: [241×241×81 double]
             w: [241×241×81 double]
          size: [241 241 81]
          axis: [-40 20 -40 20 0 20]
          name: 'COVIS-20200301T000002-imaging1'
```

- 'type': type of data value (usually 'backscatter-cross-section').

- '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.

- 'Ia','Id','Ia_filt','Id_filt': volume backscatter cross-sections, where the 'a' indicates pings were averaged, 'd' indicates pings were differenced, and 'filt' indicates that OS-CFAR filtering was applied.

- 'Kp': normalized variance of volume backscatter cross-sections.

- '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: 'imaging1'
    alpha__id: 'COVIS-20200301T000002-imaging1'
    starttime: {[1.5830e+09]  [589293]}
      endtime: {[1.5830e+09]  [763699]}
       motion: [1×1 struct]
     settings: [1×1 struct]
         path: '/tmp/covis7821to49'
         name: 'COVIS-20200301T000002-imaging1'
```

- 'mode': acquisition mode and sector no. (e.g., imaging1).

- 'alpha_id': the file name.

- 'starttime','endtime': start and end times of the scan. The foramt is: first value + second value/10^6 = 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 acquisiton.

```
covis.sweep.motion


ans =

  struct with fields:

        start: {[155]  [241.6000]  [135]}
          inc: {[0]  [1]  [0]}
        steps: 73
     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 beefore 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: 53
       txpower: 220
   pulse_width: 5.0000e-04
         range: 75
```

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

```
covis.ping


ans =

  1×365 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.

covis.burst lists the information of the burst of pings transmitted at each motor step during an scan.

```
covis.burst


ans =

  1×73 struct array with fields:

    pitch
    ping
    npings
```

- 'pitch': pitch angle in degrees at each step

- 'ping': ping no. of those within each burst.

- 'npings': number of pings within each burst.

Diagnostics

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

```
covis.bad_ping


ans =

    []
```

Published with MATLAB® R2019a