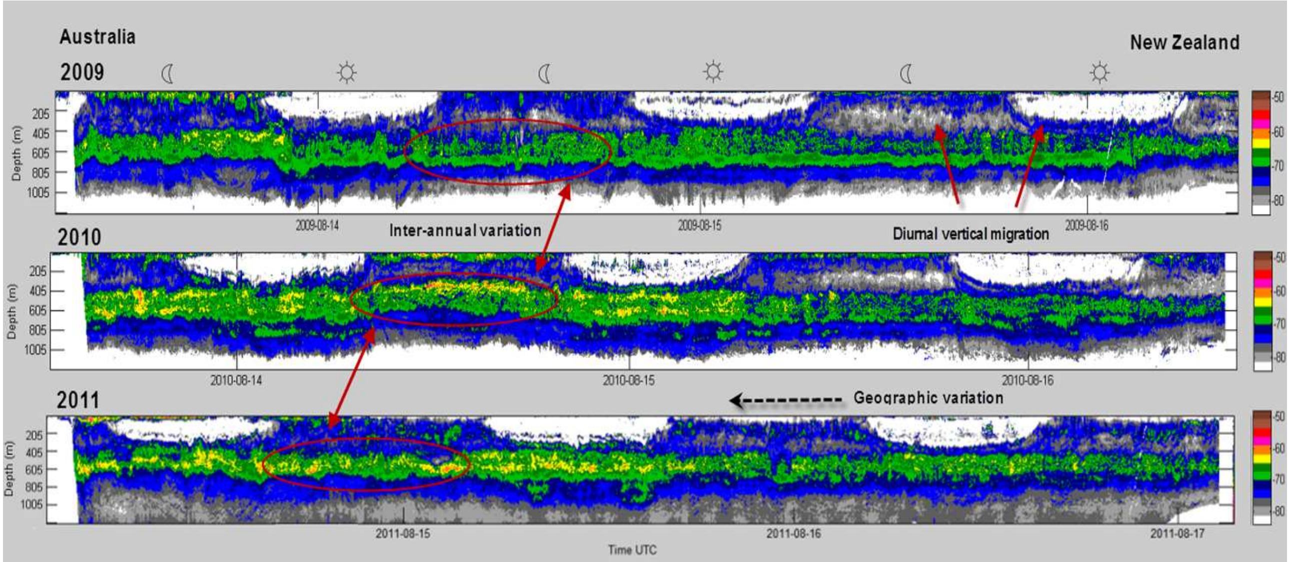

IMOS SOOP-BA NetCDF Manual

Version 1.1
September 21st 2011

Gordon Keith, Tim Ryan, Ryan Downie

Document History

Version	Date	Comments
0.1	2010-12-10	Initial draft
0.2	2011-01-13	
0.3	2011-03-01	
0.4	2011-03-04	Updated variable names
0.5	2011-03-22	Updated variable names
0.6	2011-03-25	Updated variable names, improved formatting
0.7	2011-04-04	Updated variables, changed DEPTH to RANGE added viz_sv
0.8	2011-04-06	Updated variables name, added range_reference
0.9	2011-04-11	Revert RANGE to DEPTH
1.0	2011-05-23	Changed to SI units (from dB to m^{-1})
1.1	2011-09-21	Updated Viz_sv script

Table of Contents

1 Overview

1.1 About IMOS SOOP-BA

Integrated Marine Observing System - Ships Of Opportunity – Bio-Acoustics (IMOS SOOP-BA) is an IMOS project to collect basin scale acoustic data from ships fitted with echo sounders.

IMOS is a distributed set of equipment and data-information services which collectively contribute to meeting the needs of marine climate research in Australia. The observing system provides data in the open oceans around Australia out to a few thousand kilometres as well as the coastal oceans. The IMOS Office coordinates the deployment of a wide range of equipment and assembles the data through 11 facilities distributed around the country. The data are made available to researchers through the electronic Marine information Infrastructure (eMII) located at the University of Tasmania. The IMOS infrastructure also contributes to Australia’s role in international programs of ocean observing.

Bio-Acoustic Ship Of Opportunity Sub-facility (BASOOP) is a new addition to the IMOS program that started in July 2010. BASOOP is focusing on the acoustic sensing of mid-trophic level organisms (meso-zooplanktonic and micronekton communities ~2 to 20 cm in length including small fish, crustaceans, squids and gelatinous) across the oceans basins.

1.2 About this document

This document describes the data files produced by the SOOP-BA project and made available through IMOS, including via the ocean portal http://imos.aodn.org.au/webportal/.

This document should be read as an adjunct to the IMOS NetCDF User's Manual: NetCDF Conventions and Reference Tables version 1.2, April 2009 http://imos.org.au/fileadmin/user_upload/shared/emii/IMOS_netCDF_usermanual_v1.2.pdf. For ease of cross-referencing, section headings in this document correspond to the headings in the IMOS NetCDF User’s manual. Sections omitted or with minimal content apply directly as written in the IMOS NetCDF User's manual.

The IMOS NetCDF convention manual is based on the one prescribed by the OceanSITES User’s Manual, version 1.1. The OceanSITES program is the global network of open-ocean sustained time series reference stations that have been implemented by an international partnership of researchers. More information about this project is available at http://www.oceansites.org.

2 SOOP-BA data format

SOOP-BA uses the IMOS NetCDF conventions, based on the NetCDF Climate and Forecast (CF) Conventions (http://cf-pcmdi.llnl.gov/). Keywords are taken from the GCMD vocabulary (Olsen et.al., 2007). The GCMD keywords list can be downloaded from http://gcmd.nasa.gov/Resources/valids/archives/keyword_list.html

SOOP-BA uses the NetCDF format with CF, COARDS an Udunits conventions as described in section 2 of the IMOS NetCDF User's manual.

For more information on CF, COARDS, NetCDF, Udunits, and ISO8601 see:

NetCDF:	http://www.unidata.ucar.edu/software/netcdf/docs/BestPractices.html
Udunits:	http://www.unidata.ucar.edu/software/udunits/
CF:	http://cf-pcmdi.llnl.gov/
COARDS:	http://ferret.wrc.noaa.gov/noaa_coop/coop_cdf_profile.html
ISO8601:	http://en.wikipedia.org/wiki/ISO_8601

3 NetCDF file structure

3.1 *Global Attributes*

3.1.1 Definition

The global attribute section of a NetCDF file contains metadata that describes the overall contents of the file and allows for data discovery. All fields should be human-readable and can be of either ‘character’ or ‘numeric’ type. IMOS recommends that all listed attributes be used and contain meaningful information unless there are technical reasons rendering this impossible (for example, information not available for historical data). Files must at least contain the attributes listed as “mandatory”.

Global attributes are derived from five sources:

- Institution (the same for all SOOP-BA files)
- Platform (the same for a ship)
- Instrument (the same for a transducer)
- Dataset (different for each file)
- Data (different for each file)

Institution, platform and instrument attributes may change over time but are generally static. Dataset and data attributes describe the particular dataset and will change with each data file.

3.1.2 List of the global attributes

The following tables of global attributes supersede Table 2 of the IMOS NetCDF User's manual. They include all the mandatory fields and some of the optional fields of the IMOS manual with a number of additional optional fields added (see section 3.1.4 of the IMOS NetCDF User's Manual). The tables have been divided differently than the IMOS manual.

They list all the global attributes used to define an IMOS SOOP-BA dataset.

Mandatory fields are marked with an asterisk (*).

Fields marked with *italics* do not appear in the IMOS NetCDF User's manual.

The “Type” values are S for string, N for numeric (byte, short, long, integer, float or double), D for the type of the data variable and Q for the type of the corresponding quality control data variable

3.1.2.1 Project attributes

The values provided in the table below will appear in all SOOP-BA NetCDF files.

Name	Type	Value	Definition
* project	S	Integrated Marine Observing System – Ships of opportunity - Bio-	The scientific project that produced the data

Name	Type	Value	Definition
		Acoustics	
* conventions	S	IMOS version 1.2	Name of the format convention used by the dataset
* institution	S	CSIRO Marine and Atmospheric Research	Name of the institute or facility where the original data was produced
institution_address	S	Castray Esplanade Hobart TAS 7000 Australia	Address of the institute or facility where the original data was produced
institution_postal_address	S	GPO Box 1538 Hobart TAS 7001 Australia	Postal address of the institute or facility where the original data was produced
source	S	Echosounder	Method of production of the original data
* keywords	S	Oceans > Ocean Acoustics > Acoustic Scattering, Oceans > Aquatic Sciences > Fisheries, ACOUSTIC SOUNDERS, ECHO SOUNDERS,	A comma separated list of key words and phrases. To be consistent with the MEST we recommend using the GCMD vocabulary (Olsen et.al., 2007). The GCMD keywords list can be downloaded from: http://gcmd.nasa.gov/Resources/valids/archives/keyword_list.html Non-GCMD keywords may be used at your discretion
references	S		Published or web-based references that describe the data or the methods used to produce the data. Include a reference to IMOS and a project-specific reference if appropriate. Multiple references should be separated with a semicolon “;”
netcdf_version	N	3.5	NetCDF version used for the dataset
naming_authority	S	IMOS	This will always be “IMOS”
cdm_data_type	S	Trajectory	This will always be “Trajectory”
data_centre	S	eMarine Information Infrastructure (eMII)	Data centre in charge of the data management or party who distributed the resource
* data_centre_email	S	info@emii.org.au	Data Centre contact e-mail address
author_email	S	Ryan.Downie@csiro.au	NetCDF file author contact e-mail address
author	S	Ryan Downie	Name of the person responsible for the creation of the dataset

Name	Type	Value	Definition
*principal_investigator	S	Rudy Kloser	Name of the principal investigator in charge of the platform
principal_investigator_email	S	Rudy.Kloser@csiro.au	Principal Investigator e-mail address
institution_references	S		References that describe the data provider institution, the place to find all information on the dataset (web-based, i.e. give URLs). Multiple references should be separated with a semicolon “;”.
distribution_statement	S	Data, products and services from IMOS are provided "as is" without any warranty as to fitness for a particular purpose.	Statement describing data distribution policy: Re-packagers of IMOS data should include a statement that information about data quality and lineage is available from the metadata record and a statement that data, products and services from IMOS are provided "as is" without any warranty as to fitness for a particular purpose
acknowledgement	S	Any users of IMOS data are required to clearly acknowledge the source of the material in the format: "Data was sourced from the Integrated Marine Observing System (IMOS) - an initiative of the Australian Government being conducted as part of the National Collaborative Research Infrastructure Strategy and and the Super Science Initiative.”	Any users (including re-packagers) of IMOS data are required to clearly acknowledge the source of the material in this format
project_acknowledgements	S		Statement acknowledging funding sources, collaborating institutes and industry stakeholders supporting this project

3.1.2.2 Platform attributes

Each ship will have a set of attributes from the table below. These attributes will change only rarely.

Attribute names in *italics* are non-IMOS attributes.

Name	Type	Example	Definition
platform_code	S	VNAA	Ship's callsign (Australian callsign if the ship has multiple callsigns)
vessel_name	S	Aurora Australis	Name of the ship
callsign	S	VNAA	Ship's callsign (Australian callsign if the ship has multiple callsigns)
alt_callsign	S		Alternative callsign if the ship has more than one
imo	S	8717283	Ship's International Maritime Organisation ship identification number
home_port	S	Hobart	Home port of the vessel
operator	S	Australian Antarctic Division	Name of organisation which operates the ship
platform_acknowledgement	S		Statement acknowledging owners and operators of vessels supporting collection of this data

3.1.2.3 Instrument attributes

The following table of attributes will occur for each transducer on the vessel. The <id> in the variable name will be replaced by the frequency of the transducer (e.g. 38).

Name	Type	Example	Definition
frequency_<id>	N	38	Frequency of the transceiver/transducer combination in kHz
transceiver_manufacturer_<id>	S	Simrad	Transceiver manufacturer
transceiver_model_<id>	S	ES60	Transceiver model
transceiver_serial_<id>	S	GPT 38 kHz 009072048f0b 1 ES38B	Transceiver serial number
transceiver_firmware_<id>	S		Transceiver firmware version
transceiver_power_<id>	N		Transceiver power (watts)
transceiver_pulse_<id>	N		Transceiver pulse length

Name	Type	Example	Definition
			(milliseconds)
transceiver_gain_<id>	N		Transceiver gain value
transceiver_sa_correction_<id>	N		Sa correction value (Simrad transceivers)
transceiver_absorption_<id>	N		Absorption value in dB/m
transceiver_sound_speed_<id>	N		Sound speed value in m/s
transducer_location_<id>	S	Keel	Location of installed transducer
transducer_depth_<id>	N		Depth of transducer-face beneath the water-surface (m)
transducer_orientation_<id>	N	0	Direction of main transducer beam with respect to water surface (degrees). A downward pointing transducer will have an orientation of 0 degrees, an upward pointing transducer an orientation of 180 degrees
transducer_aperture_<id>	S	single	Transducer aperture, either single or split
transducer_manufacturer_<id>	S		Transducer manufacturer
transducer_model_<id>	S		Transducer model
transducer_serial_<id>	S		Transducer serial number
transducer_psi_<id>	N		Transducer equivalent beam angle
acquisition_software_version_<id>	S	2.20	Name of acquisition software and its version number
processing_script_version_<id>	S	R402	Version number of the processing scripts used as per the version control

Name	Type	Example	Definition
			system.
processing_software_version_<id>	S	V4.90.70.17641	Name of post processing software and its version number
processing_template_<id>	S	REH2010_2048ms.ev	Source template upon which all processing was based
processing_template_version_<id>	S	R401	Version of the source template in the version control system.
calibration_date_<id>	S		Date of nearest calibration
calibration_accuracy_<id>	N	0.5	Estimated accuracy of the calibration

3.1.2.4 Dataset attributes

The attributes in this table will vary with each data file and will be provided by the author of the data file.

Name	Type	Example	Definition
* title	S	Trans Tasman transect October 2010	Short description of the dataset
* abstract	S		A paragraph describing the dataset: type of data contained in the dataset, how the data was created, the creator of the dataset, the project for which the data was created, the geospatial coverage of the data, the temporal coverage of the data
history	S		Provides an audit trail for modifications to the original data. It should contain a separate line for each modification, with each line beginning with a timestamp and including user name, modification name and modification arguments
comment	S		Miscellaneous information about the data or methods used to produce it. Any free-format text is appropriate
metadata	S		URL to the metadata record corresponding to the netCDF file

Name	Type	Example	Definition
citation	S	The citation in a list of references is: IMOS, [year-of-data-download], [Title], [data-access-URL], accessed [date-of-access]	The citation to be used in publications using the dataset should follow the format: “IMOS. [year-of-data-download], [Title], [Data access URL], accessed [date-of-access]”
voyage_id	S		Voyage id where one exist, common only on research vessels.
local_time_zone	N	10	Local time zone. See chapter 3.1.3 on time format in the NetCDF User's manual. If local time does not fall into one zone for the full dataset, do not use this attribute
integration_interval	S	“1 km” or “10 min” or “50 pings”	Integration cell width measured in the unit used to determine the cell, e.g. distance, time or pings
integration_height	N	10	Integration cell height in m used to determine the cell
absorption_correction	S	Constant	Method used for correcting for sound absorption. “Constant” implies a single value was used for absorption for the whole dataset
sound_speed_correction	S	Constant	Method used for correcting for sound speed. “Constant” implies a single value was used for absorption for the whole dataset
motion_correction	S	None	Method used for correcting for platform motion

3.1.2.5 Dataset attributes – automatically generated

The attributes in this table will vary with each file, but may be derived automatically from the data.

Name	Type	Example	Definition
* date_created	S	2010-12-15T14:30:00Z	The date on which the data was created
date_modified	S	2011-02-17T15:45:00Z	The date on which the data was modified
* geospatial_lat_min	N		The southernmost latitude, a value between -90 and 90 degrees
* geospatial_lat_max	N		The northernmost latitude, a value between -90 and 90 degrees
* geospatial_lon_min	N		The westernmost longitude, a value between -180 and 180 degrees

Name	Type	Example	Definition
* geospatial_lon_max	N		The easternmost longitude, a value between -180 and 180 degrees
* geospatial_vertical_min	N		Minimum depth for measurements
* geospatial_vertical_max	N		Maximum depth for measurements
* time_coverage_start	S		Start date of the data in UTC
* time_coverage_end	S		Final date of the data in UTC

3.2 Data File Dimensions

SOOP-BA NetCDF files will have two dimensions, DEPTH which is fixed and TIME which is unlimited.

3.3 Variables

3.3.1 Coordinate variables

3.3.1.1 Time

Double TIME(TIME) ;

TIME:long_name = “time” ;

TIME:standard_name = “time” ;

TIME:units = “days since 1950-01-01T00:00:00Z” ;

TIME:axis = “T” ;

TIME:valid_min = 0 ;

TIME:valid_max = 90000.0 ;

TIME:_FillValue = 99999.0 ;

TIME:calendar = “gregorian”

TIME:comments = “mean time of pings in processing cell” ;

TIME:quality_control_set = 1

TIME:quality_control_indicator = 1

TIME:uncertainty = 0.0003

3.3.1.2 Location (X-Y horizontal space)

LATITUDE

Float LATITUDE(TIME) ;

LATITUDE:long_name = “latitude” ;

LATITUDE:standard_name = “latitude” ;

LATITUDE:units = “degrees_north” ;

LATITUDE:axis = “Y” ;

LATITUDE:valid_min = -90 ;

LATITUDE:valid_max = 90 ;

LATITUDE:_FillValue = 999.0 ;

LATITUDE:comment = “mean latitude of ping in the processing cell” ;

LATITUDE:quality_control_set = 1

LATITUDE:quality_control_indicator = 1

LATITUDE:uncertainty = 0.001

LATITUDE:reference-datum = “geographical coordinates, WGS84 projection“

LONGITUDE

Float LONGITUDE(TIME) ;

LONGITUDE:long_name = “longitude” ;

LONGITUDE:standard_name = “longitude” ;

LONGITUDE:units = “degrees_east” ;

LONGITUDE:axis = “X” ;

LONGITUDE:valid_min = -180 ;

LONGITUDE:valid_max = 180 ;

LONGITUDE:_FillValue = 999.0 ;

LONGITUDE:comment = “mean latitude of ping in the processing cell” ;

LONGITUDE:quality_control_set = 1

LONGITUDE:quality_control_indicator = 1

LONGITUDE:uncertainty = 0.001

LONGITUDE:reference_datum = “geographical coordinates, WGS84 projection”

3.3.1.3 DEPTH

The DEPTH dimension measures the depth below the sea surface of the centre of the processing layer. For upward facing transducers and other orientations on moorings and other platforms depth is not meaningful and RANGE may be used instead.

Float DEPTH(DEPTH) ;

DEPTH:long_name = “mean depth of integration layer” ;

DEPTH:standard_name = “depth” ;

DEPTH:units = “metres” ;

DEPTH:axis = “Z” ;

DEPTH:positive = “down” ;

DEPTH:valid_min = 0 ;

DEPTH:valid_max = 12000 ;

DEPTH:_FillValue = -99999.0 ;

DEPTH:comment = “mean depth of processing layer derived from echo travel time and constant sound velocity” ;

DEPTH:quality_control_set = 1

DEPTH:quality_control_indicator = 1

DEPTH:uncertainty = 0.001

DEPTH:reference_datum = “sea surface”

3.3.2 Data variables

Each data variable will appear once for each transducer with the transducer frequency appended to the variable name. If more than one transducer of the same frequency is present (not expected) an additional suffix is added to distinguish the transducer.

SOOP-BA has one data variable per transducer: Sv_<id>

Example:

float Sv_38(TIME,DEPTH);

Sv_38:units = “m^{-1}”;

Sv_38:_FillValue = 9999;

Sv_38:long_name = “Mean volume backscatter coefficient 38kHz”;

Sv_38:valid_min = 0.;

Sv_38:valid_max = 1.;

Sv_38:ancillary_variables = “Sv_38_pcnt_good Sv_38_quality_control” ;

Sv_38:observation_type = “calculated”;

Sv_38:cell_methods = “mean”;

Sv_38:quality_control_set = 1;

3.3.3 Ancillary variables

For each Sv_ variable the following ancillary variables are defined:

Name	Long name	Description
Sv_pcnt_good_<id>	percent_Sv_samples_included_<id>kHz	Percentage of the original data that was used in calculating the Sv value (not rejected by data cleaning)
mean_depth_<id>	mean_cell_depth_<id>kHz	Mean depth of the cell for which the Sv mean is calculated
mean_height_<id>	mean_cell_height_<id>kHz	Mean height of the cell. This can be used to calculate NASC (Nautical Area Scattering Coefficient – m^2/nmi^2). $NASC = 4\pi \times 1852^2 \times mean\_height \times 10^{(Sv/10)}$ NASC = 4 * π * 1852 * 1852 * mean_height_<id> * 10 ^ (Sv_<id> / 10)

In addition to the above the following ancillary variables may also be included. These are the statistics for the data prior to rejection of bad data and may be useful in assessing the quality of the data.

Name	Long name	Description
Sv_sd_<id>	standard_deviation_volume_backscatter_<id>kHz	Standard deviation of the Sv
Sv_skew_<id>	skewness_volume_backscatter_<id>kHz	Skewness
Sv_kurt_<id>	kurtosis_volume_backscatter_<id>kHz	Kurtosis
Sv_<id>_quality_control		Indication of quality of the Sv_mean
Sv_unfilt_<id>	mean_volume_backscatter_including_bad_data_<id>kHz	Sv mean of the full data set, including rejected data
Sv_unfilt_sd_<id>	standard_deviation_volume_backscatter_including_bad_data_<id>kHz	Standard deviation of the uncleaned Sv
Sv_unfilt_skew_<id>	skewness_volume_backscatter_including_bad_data_<id>kHz	Skewness of the uncleaned Sv
Sv_unfilt_kurt_<id>	kurtosis_volume_backscatter_including_bad_data_<id>kHz	Kurtosis of the uncleaned Sv

Example:

```
double Sv_pcnt_good_38(TIME, DEPTH) ;
        Sv_pcnt_good_38:name = "Sv_pcnt_good_38" ;
        Sv_pcnt_good_38:long_name = "percent_Sv_samples_included_38kHz" ;
        Sv_pcnt_good_38:units = "%" ;
        Sv_pcnt_good_38:_FillValue = 9999. ;
        Sv_pcnt_good_38:valid_min = 0. ;
        Sv_pcnt_good_38:valid_max = 100. ;
        Sv_pcnt_good_38:ancillary_variables = "Sv_pcnt_good_38_quality_control" ;
        Sv_pcnt_good_38:quality_control_set = 1. ;
double mean_height_38(TIME, DEPTH) ;
        mean_height_38:name = "mean_height_38" ;
        mean_height_38:long_name = "mean_cell_height_38kHz" ;
        mean_height_38:units = "m" ;
        mean_height_38:_FillValue = 9999. ;
        mean_height_38:valid_min = 0. ;
        mean_height_38:valid_max = 100. ;
        mean_height_38:ancillary_variables = "mean_height_38_quality_control" ;
        mean_height_38:quality_control_set = 1. ;
double mean_depth_38(TIME, DEPTH) ;
        mean_depth_38:name = "mean_depth_38" ;
        mean_depth_38:long_name = "mean_cell_depth_38kHz" ;
```

```
        mean_depth_38:units = "m" ;
        mean_depth_38:_FillValue = 9999. ;
        mean_depth_38:valid_min = 0. ;
        mean_depth_38:valid_max = 10000. ;
        mean_depth_38:ancillary_variables = "mean_depth_38_quality_control" ;
        mean_depth_38:quality_control_set = 1. ;
double Sv_unfilt_38(TIME, DEPTH) ;
        Sv_unfilt_38:name = "Sv_unfilt_38" ;
        Sv_unfilt_38:long_name =
            "mean_volume_backscatter_including_bad_data_38kHz" ;
        Sv_unfilt_38:units = "m-1" ;
        Sv_unfilt_38:_FillValue = 9999. ;
        Sv_unfilt_38:valid_min = 0. ;
        Sv_unfilt_38:valid_max = 1. ;
        Sv_unfilt_38:ancillary_variables = "Sv_unfilt_38_quality_control" ;
        Sv_unfilt_38:quality_control_set = 1. ;
```

3.3.4 Data variable attributes added by user

User defined data variables will be included in future versions of the NetCDF document.

3.3.5 Quality Control (QC)

QC variables are described in full in section 3.3.5 of the IMOS NetCDF User's Manual.

TIME, DEPTH, LATITUDE, LONGITUDE and Sv_<id> variables will all have corresponding _quality_control variables as required by the IMOS conventions.

For TIME, DEPTH, LATITUDE and LONGITUDE the quality control flag will always be 1 – good data.

For the Sv_<id> variables the quality control flag will be derived from the percent_good variable. If the percent_good is above a certain threshold the data is flagged good, otherwise it is flagged bad. The threshold level is yet to be determined and may change.

3.3.6 Uncertainty

Uncertainty variables are described in section 3.3.6 of the IMOS NetCDF User's Manual and refers to http://imos.org.au/fileadmin/user_upload/shared/IMOS General/documents/IMOS/IMOS_Data_streams_and_Their_Uncertainties_3.1.pdf.

The term uncertainty is here defined as "the parameter, associated with the result of a measurement that characterises the dispersion of the values that could reasonably be attributed to the measurand" (Underwood, 2008).

All of the SOOP-BA variables are means of data values (TIME, DEPTH, LATITUDE, LONGITUDE, Sv_mean).

For the dimension variables the standard deviation of measurements is not provided by the processing software and it's not clear to me what other measurement would make sense. Note that DEPTH is calculated from echo time and a fixed sound speed, which in general will not be accurate, so some indication of range uncertainty may need to be derived from a model of likely sounds speed profiles.

For the Sv_ variable the standard deviation, skewness and kurtosis ancillary variables may be included to describe the uncertainty. No additional variables are provided.

4 Metadata

All IMOS data will be available online through an IMOS/eMII data portal.

IMOS metadata requirements comply to international standards and particularly to the Marine Community Profile of ISO19115 (Reed 2008).

A link to the eMII MEST (Metadata Entry and Search Tool) can be found online at: http://imos.org.au/emii_data.html

A fuller discussion of IMOS metadata is in section 4 of the IMOS NetCDF User's Manual.

5 IMOS file naming convention

The SOOP-BA file name format is:

IMOS_SOOP-BA_A_<Start-date>_<Platform_Code>_FV02__END-<End-date>_C-<Creation_date>.nc

- Platform_Code is the callsign of the ship that acquired the data.
- Start-date is the time of the first data point in the data.
- End-date is the time of the last data point in the data.
- Creation_date is the time at which the NetCDF file was created.

NetCDF files are named according to the IMOS NetCDF File Naming Convention (Mancini et. al 2008 http://imos.org.au/fileadmin/user_upload/shared/emii/IMOS_netCDF_filenaming_convention_v1.3.pdf), see Appendix 2 of the IMOS NetCDF User's Manual .

Appendix A – Matlab visualisation

The following script can be used to open and visualise the data from a SOOP-BA NetCDF file conforming to the specifications in this document.

```
function data = viz_sv(ncfile, imagefile)
% viz_sv
% function to read and visualise IMOS-SOOP-BA Sv data
% usage: object = viz_sv(ncfile)
%
%   ncfile - IMOS-SOOP-BA NetCDF file name
%   imagefile - if present a copy of the Sv figure is written to a file.
%           imagefile may be a full file name, empty, a directory or
%           an extension (e.g. '.jpg')
%
% Author:   Tim Ryan <tim.ryan@csiro.au>
%           Gordon Keith <gordon.keith@csiro.au>
% Version:  1.3
% Date:     2011-09-05

    if nargin == 0 || isempty(ncfile)
        [filename pathname] = uigetfile(...
            {'*.nc' '*.nc NetCDF file'; '*' 'All files'}, ...
            'IMOS-SOOP-BA NetCDF file');
        if isequal(filename,0)
            error('NetCDF filename required');
        end
        ncfile = fullfile(pathname,filename);
    end

    if isdir(ncfile)
        [filename pathname] = uigetfile(...
            {'*.nc' '*.nc NetCDF file'; '*' 'All files'}, ...
            'IMOS-SOOP-BA NetCDF file', ncfile);
        if isequal(filename,0)
            error('NetCDF filename required');
        end
        ncfile = fullfile(pathname,filename);
    end

    if nargin < 2
        imagefile = [];
    end

% open the netcdf file
    if isempty(ls(ncfile))
        fprintf('\n-----------------------------------------------------\n');
        fprintf('\nnetcdf file %s \ncannot be found, check location\n',ncfile);
        fprintf('\n-----------------------------------------------------\n');
    else
        ncid = netcdf.open(ncfile, 'NC_NOWRITE');
        data.file = ncfile;
        [~, filename, ~] = fileparts(ncfile);

        % read data
        lat_id = netcdf.inqVarID(ncid, 'LATITUDE');
        data.latitude = netcdf.getVar(ncid, lat_id);
```

```
lon_id = netcdf.inqVarID(ncid, 'LONGITUDE');
data.longitude = netcdf.getVar(ncid, lon_id);

try
depthid =  netcdf.inqVarID(ncid, 'DEPTH');
catch
depthid =  netcdf.inqVarID(ncid, 'RANGE');
end
depth = netcdf.getVar(ncid, depthid);
data.depth = depth;
timeid =  netcdf.inqVarID(ncid, 'TIME');
time50 = netcdf.getVar(ncid, timeid);
time = time50 + datenum('1950 01 01');
data.time = time;

svid = netcdf.inqVarID(ncid, 'Sv_38');
data.sv = netcdf.getVar(ncid, svid);
qcid = netcdf.inqVarID(ncid, 'Sv_38_quality_control');
data.qc = netcdf.getVar(ncid, qcid);
pgid =  netcdf.inqVarID(ncid, 'Sv_pcnt_good_38');
data.pg = netcdf.getVar(ncid, pgid);
svrawid = netcdf.inqVarID(ncid,'Sv_unfilt_38');
data.svraw = netcdf.getVar(ncid,svrawid);

% Ignore bad data
data.sv(data.qc>2)=NaN;

% convert to dB
data.Sv = 10 * log10(data.sv);
data.Svraw = 10 * log10(data.svraw);

% calculate NASC
data.mean_Sv = 10*log10(nmean(data.sv));
data.NASC = 4*pi*1852*1852*10.^(data.mean_Sv./10)*1200;

data.mean_Svraw = 10*log10(nmean(data.svraw));
data.NASCraw = 4*pi*1852*1852*10.^(data.mean_Svraw./10)*1200;

% plot track
figure(1)
hold on
plot(data.longitude,data.latitude,'r.');

xlabel('Longitude');
ylabel('Latitude');
title({ filename; 'Voyage Track'}, 'Interpreter', 'none');

lat = median(data.latitude);
set(gca, 'DataAspectRatio', [1 cos(lat/ 180 * pi) 1 ]);

% plot the sv data
min_sv = -84; range = 36; max_sv = min_sv + range;

echogram(data.Sv, { filename ;  'Sv mean (dB re 1 m-1)' }, ...
    [min_sv max_sv], 2);

% If second argument was given write Sv image to file
if ischar(imagefile)
    write_echogram(imagefile, ncfile);
```

```
        end

         echogram(data.Svraw, { filename ;  'Sv mean (dB re 1 m-1), Unfiltered
data' }, ...
             [min_sv max_sv], 1)

         echogram(data.pg, { filename ;  'percent accepted' }, ...
             [0 100], 0);

         % echointegration plots
         figure
         hold on
         plot(data.NASC,'g');
         plot(data.NASCraw,'r');

         legend('NASC', 'raw NASC');
         xlabel('Intervals');
         ylabel('NASC');
         title({ filename ; 'echointegration' }, 'Interpreter', 'none');

         zoom(gcf, 'reset');
         ylim([0, max(data.NASC)]);
     end

     % plot data in time v depth
     function echogram(dataset, ttle, range, position)
         figure
         imagesc(dataset, range)
         colormap(EK500colourmap())

         set(zoom,'ActionPostCallback',@ticks)
         set(pan,'ActionPostCallback',@ticks)

         xlabel('Time UTC')
         ylabel('Depth (m)')
         title(ttle, 'Interpreter','none');

         colorbar;

         % calculate image size
         [dheight, dwidth] = size(dataset);
         left = 80;
         bot  = 50;
         width = dwidth + 2 * left;
         height = dheight + 2 * bot;
         base = 100 + floor(position * 1.4 * height);
         set(gcf, 'Position', [0, base, width, height ] );
         pos = get(gcf, 'Position');

         % If the figure is too wide to fit on a single screen
         % force it over screen boundary
         % only works with dual screen display.
         if pos(3) < width
             pos(1) = pos(1) + 1;
             pos(3) = width;
             set(gcf, 'Position', pos);
             get(gcf, 'Position');
             set(gcf, 'Position', pos);
         end
```

```
        % set image to one pixel per cell
        set(gca, 'Units', 'pixels');
        set(gca, 'Position', [left, bot, dwidth, dheight]);
        set(gca, 'Units', 'normalized');

        % update axis
        ticks(gcf)
    end

    % Colour map used by EK500
    function [EK500cmap]=EK500colourmap()

        EK500cmap = [255   255   255   % white
                     159   159   159   % light grey
                      95    95    95   % grey
                       0     0   255   % dark blue
                       0     0   127   % blue
                       0   191     0   % green
                       0   127     0   % dark green
                     255   255     0   % yellow
                     255   127     0   % orange
                     255     0   191   % pink
                     255     0     0   % red
                     166    83    60   % light brown
                     120    60    40]./255;  % dark brown
    end

    % redraw time and depth ticks on plot
    function ticks(figure,~)
        axes=get(figure,'CurrentAxes');

        % depth ticks
        ytick=get(axes,'YTick');
        set(axes,'YTickLabel',depth(floor(ytick)));

        % time ticks
        % xtick=get(axes,'XTick');
        % set(axes,'XTickLabel',datestr(time(floor(xtick))));
        xlim=get(axes,'XLim');
        start=time(ceil(xlim(1)));
        finish=time(floor(xlim(2)));

        len=finish-start;
        if len > 2
            format=29;      % 'yyyy-mm-dd'
            tock=1;         % day
        elseif len > .5
            format = 'yyyy-mm-dd HH:MM';
            tock=4;         % 6 hr
        elseif len > .1
            format = 15;    % 'HH:MM'
            tock=24;        % hour
        else
            format = 13;    % 'HH:MM:SS'
            tock=96;        % quarter hour
        end

        xtock=(ceil(start*tock):1:finish*tock)/tock;
        xtick(length(xtock))=0;
```

```
        for i=1:length(xtock)
            xtick(i)=find(time>xtock(i),1);
        end

        set(axes,'XTick',xtick);
        set(axes,'XTickLabel',datestr(time(xtick), format));
    end

    % write current figure to file
    function write_echogram(imagefile, file)
        % is an empty character string
        if isempty(imagefile)
            imagefile = [ file '.png' ];
        end
        % is an extension only
        if imagefile(1) == '.' && min(imagefile ~= filesep)
            imagefile = [ file imagefile ];
        end
        % is a directory
        if isdir(imagefile)
            [~, name, ext] = fileparts(file);
            imagefile = fullfile(imagefile, [ name ext '.png' ]);
        end

        drawnow;
        frame = getframe(gcf);
        [X,~] = frame2im(frame);
        imwrite(X, imagefile);
    end

    % calculate mean ignoring NaN values
    function nmean = nmean(data)
        nan = isnan(data);
        data(nan) = 0;
        count = sum(~nan);
        count(count==0) = NaN;      % prevent divide by 0
        nmean = sum(data)./count;
    end

end
```