Managing DTAG-3 Sensor data

Mark Johnson
Sea Mammal Research Unit
University of St. Andrews
markjohnson@st-andrews.ac.uk
Feb 2013

This note describes the differences between the DTAG-2 and DTAG-3 (D2 and D3 for short) sensor data formats and documents the Matlab tools that are used to access D3 data. See d3tag2whale.pdf for a step-by-step protocol for reading in D3 sensor data.

Version 3 DTAGs support variable and multi-rate sensor sampling protocols. The sampling rate of each sensor can be different and accordingly the amount of data produced during a deployment by each sensor can be different. Also the sensor channels read by each tag may be different depending on what sensor suite it contains. To accomodate heterogenous data rates, a cell array must be used instead of a matrix to store sensor data in Matlab. Cell arrays are similar to matrices but allow storage of different sized objects in each cell. For D2 data, swvread.m was used to read a matrix of sensor data. It returned the matrix and the sampling rate of the data which is the same for all channels. Each column of the matrix represented a different sensor channel with a fixed mapping for all tags, e.g., accelerometer data was always in the first three columns. For D3 data, d3readswv returns a structure containing three objects: the sensor data cell array, a vector of sampling rates, and a vector of sensor channel numbers. Each cell in the data array represents a different sensor channel and contains a vector of the data recorded. The sampling rate of the cell is given in the corresponding location in the sampling rate vector and a number uniquely identifying the kind of sensor data is in the channel number vector. Thus, typing:

```
X=d3readswv(recdir,prefix,df)
```

results in a structure X containing X.x (the cell array of sensor data), X.fs (the sampling rate vector), and X.cn (the channel numbers). The nth channel of sensor data is accessed by X.x{n}. It has a sampling rate of X.fs(n) and a channel number X.cn(n). The channel numbers can be converted into information about the sensor channels using:

```
[ch_names,descr] = d3channames(X.cn)
```

This returns cell arrays of text strings with the name and description of each channel. The nth sensor channel has name ch_name{n} and description descr{n}. Although most DTAGs may have the same sensor suites and software, and so will produce the same sequence of sensor data, there will be some that differ. The automatic calibration tools (d3calpressure, d3calmag and d3calacc) support different sensor sequences and data rates. They achieve this by looking in the channel number vector for sensor data with specific types. For example, d3calpressure looks for pressure and temperature channels in the sensor structure rather than assuming that they will be in specific locations. This tool also interpolates the temperature data to match the pressure data, allowing temperature (which varies slowly) to be sampled at a lower rate than pressure.

Another point of difference between the D2 and D3 tools is the use of two strings, recdir and prefix, to define each tag deployment. The recdir defines the directory where the raw data are to be found. This replaces the settagpath('AUDIO',...) method for the D2. The prefix is the deployment name used to name the data files. For example, if the DTG files for a deployment have names like 'mn185a001.dtg', 'mn185a002.dtg' etc., the prefix is 'mn185a'. The corresponding sequence of files with different suffix numbers are automatically found and combined to produce continuous sensor data. The name used henceforward to refer to the tag deployment can either be the prefix used in the d3readswv call or you can specify a new name when calling d3deployment.m to register the tag deployment. This new naming format gives more flexibility in naming deployments as required to accomodate other applications of the D3 devices. However, it is strongly recommended to keep the D2 naming convention for animal deployments, i.e., the two-letter latin species abbreviation followed by the Julian day number (3 digits) and a letter defining the deployment sequence on the day. This will help in maintaining our and your databases of tag deployments uniform despite different tag generations.

The tool d3deployment replaces some of the initial calibration steps that had to be performed manually for D2. To register a new tag deployment, define the name that will be assigned to the deployment, and setup a calibration file, you run:

```
CAL = d3deployment(recdir,prefix,deploy_name) ;
```

where deploy_name is a string containing the name that will be used to refer to the deployment. If deploy_name is not given, prefix will be used as the deployment name. An example usage of d3deployment is:

```
[CAL,DEPLOY] = d3deployment( 'e:/data/zc10/zc10_247a', 'zc247a', 'zc10_247a') ;
```

This function will look in directory /data/zc10/zc10_247a on the e: drive for files called zc247aNNN.xml, where NNN is any 3-digit number. These files will be read in numerical sequence to extract calibration and timing information. This information will be used to generate a calibration file called zc10_247acal.xml which will be stored in the CAL directory defined on the tag path using settagpath as for D2, .i.e., D3 calibration files will be stored with the D2 calibration files. The d3deployment call also returns a CAL structure which you will use for automatic sensor calibration if it is able to locate one. It will look in any directory on the Matlab path for a .xml file that contains bench calibration information for the tag used in the deployment. Bench calibration files are usually called after the convenience name of the tag (e.g, d101.xml) and should have been provided with the tags. These files can be kept with the d3matlab tools (there is a cal subdirectory that already contains some bench cal files). If d3deployment can not find the bench cal file, look for it on the soundtags website (see below). The file name to look for is given in the ID component of the DEPLOY structure (DEPLOY.ID). You only need to run d3deployment once. If it doesn't find a bench cal, you can read one in when you have found it using d3findcal.m.

All metadata for D3 tags and deployments are stored in XML format files. The bench calibration file for each tag contains a unique ID (an 8-digit hexadecimal number) which is also repeated in the XML files that are generated when expanding .dtg archives using d3read.exe so there is no excuse for using the wrong calibration data. Running d3deployment to register a tag deployment generates a new XML file which contains both the bench cal information and the metadata from the dtg XML files. This deployment cal file is therefore all that is needed to interpret sensor data from then on. When you have run the automatic calibration functions, you will save the optimized calibration information to the deployment XML file using d3savecal. Likewise, OTAB will be save to this file when you generate it using prhpredictor. Finally, you can add your own metadata to the deployment calibration file following whatever naming format seems best to you. However, as a minimum, it is recommended to add the

following fields using d3savecal:
AUTHOR name of the data owner/curator (a string)
DECL magnetic declination in degrees (west is negative)
EMAIL string with the email of the data owner/curator
LOCATION place name where the animal was tagged as a string
EXPEDITION expedition name as a string
TAGON.POSITION 2-element vector [latitude longitude] in decimal degrees
TAGON.TIME 6-element vector [year month day hour min sec]
TAGON.PLACEMENT string describing where on the whale's body the tag was initially attached., e.g., 'left-ant-high'. Use 'ant' (anterior), 'dorsal', and 'post' (posterior) to indicate if the tag was well forward of the dorsal fin, about at the position of the dorsal fin, or well posterior, respectively. Use 'high', 'mid' and 'low' for the height up the body of the tag. High means well above the longitudinal mid-line of the animal (i.e., on the dorsal surface), mid means about at the mid-line on the side of the animal, and low mean well below the mid-line (i.e., on the ventral surface).
TAGOFF.POSITION 2-element vector [latitude longitude] in decimal degrees
TAGOFF.TIME 6-element vector [year month day hour min sec]
UTC2LOC number of hours between UTC time and local time
PHOTO name(s) of files containing photos of the tag on the animal (a string)

Note that lower or upper case names can be entered but all field names are turned into upper case by d3savecal.

The metadata can be added at any time after the d3deployment call by using d3savecal calls, e.g.:

```
d3savecal('zc10_247a', 'TAGON.POSITION', [38.3456,-123.456]);
```

You might also think about adding the following metadata if it is available:
ANIMAL.ID catalog id of the tagged animal (a string)
ANIMAL.GENDER 'male' or 'female'
ANIMAL.AGE 'calf','juvenile,'subadult' or 'adult'

Being text files, XML files are searchable and so any metadata you add to the file will enable you to perform more efficient searches through the DTAG deployment database.

The automatic calibration tools have been improved for the D3 data. Often tag deployments contain intervals in which the tag is not attached to the animal either because it has come off the animal and has not yet been recovered and stopped or because the tag started before it was attached. These intervals cause problems for the automatic calibrations tools and should be eliminated from the data. In D2, you had to do that manually. The D3 tools (d3calpressure, d3calmag and d3calacc) all give you the opportunity to restrict the calibration focus to a subset of the dive profile. When you run each tool, it shows a figure of the dive profile with a vertical green and red bar. You move these bars to where you want them by positioning the cursor and typing either 'l' or 'r' (for left or right). When you are satisfied, press the enter key to continue with the calibration. You can specify a different time restriction for each tool and the time restriction is stored in the CAL structure so you can use the same one if you call a tool multiple times. Both d3calmag and d3calacc allow an optional minimum depth to be entered as well. This restricts automatic calibration to when the animal is below this depth and so avoids noisy surface data. Because all the tools rely on good depth information, it is important to run d3calpressure first before you call d3calmag or d3calacc.

The D3 tags collect sensor data at a much higher rate than did the D2 tags so the sensor data structure, X, can be very large. The calibration tools automatically decimate the data to a low rate to minimize processing time. But you may have troubles even reading in the entire sensor data especially on a laptop without much memory. To avoid slow operation or memory problems, specify a decimation factor when you read in the sensor data for calibration. You do this by passing a third argument to d3readswv. The accelerometers are usually sampled at 200 Hz but 20 Hz data is plenty for automatic calibration so specify a decimation factor of 10.

A slowly growing resource of information on DTAGs can be found on the SOUNDTAGS website: *http://soundtags.st-andrews.ac.uk*. There you will find the latest versions of the Matlab tools along with documents and calibration files. Check there first if you are having problems. The D3 matlab tools are provided under a GNU General Public Licence and so are free to copy and change under the terms of this licence: see *http://www.gnu.org/licenses/gpl.html*. Please be a considerate colleague and acknowledge usage of these tools as appropriate in your papers and talks. A lot of evening hours were spent writing them when you were having fun! Finally, be warned: the tools are constantly in development and are not guaranteed to perform any particular function. Contact the author if in doubt.