Calibrating DTAG-3 sensor data

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

This document defines the sequence of operations needed for DTAG-3 tags to verify deployment data and convert the sensor data into engineering units. Before you start, make sure that you have the latest version of the DTAG Matlab tools and that the tools directory is on the Matlab path (use File -> Set Path in the pull-down menu at the top of the Matlab screen to change the path). Be sure to add the directory to the path including sub-directories and to save the path before closing the Set Path window. You should only have to do this once; the path is remembered on subsequent uses of Matlab.

1. Notify Matlab of where processed data should be stored. Type the following at the Matlab command prompt:

*settagpath('cal',**caldir,**'prh'**,prhdir**) ;*

where: ***caldir*** is the directory to put the calibration files e.g., *'c:/tag/tag2/matlab/cal'* and ***prhdir*** is the directory for the prh files. You can add this instruction to the Matlab startup.m file so that it runs automatically everytime you open Matlab (see help startup in Matlab).

2. Read the sensor data:

*X = d3readswv(**recdir**,**prefix**,**df**);*

Example:

X = d3readswv('/tag/data/berardius','bb214a',10);

recdir is the full path to the directory where the raw sensor data is stored. This directory must contain both the .swv and .xml files generated when you run d3read.exe on the .dtg archives. You do not need the audio .wav files for the sensor calibration. **prefix** is the common first part of the data filenames. Exclude the last three digits which identify different files within the deployment. In the example, the data are stored in files with names like *bb214a001.xml*, *bb214a002.xml* etc, in the directory */tag/data/berardius*.

df specifies an optional integer decimation factor. The DTAG-3s can sample sensors at high rates and so the full sensor data set for a long deployment may be very large. d3readswv can automatically downsample the data to a size more suitable for analysis. For example, specifying df=4 returns sensor data at 1/4 the rate at which it was originally recorded. Aim for a sampling rate of about 10-20 Hz for calibrating sensor data. Most D3s collect acceleration data at 200 Hz so df=10 or 20 is a good choice.

3. Register the deployment:

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

Example:

CAL = d3deployment('/tag/data/berardius','bb214a','bb10_214a') ;

Where **deploy_name** is the name that you want to call the deployment. For animal data, this should conform to the D2 9-character standard of 2 latin species initials, two digit year of deployment, underbar, 3 digit Julian day (use *jday.m* to convert a date to Julian day), and a letter to indicate the animal of the day. You no longer have to follow this format for non-animal data, e.g., glider or buoy deployments. For these, choose your own naming convention.

If CAL contains information, go on to step 4. If no bench calibration file is found, CAL will return empty. If this happens, you need to find a bench cal file for the tag and put it in a directory on the Matlab path (a good place is the directory called *cal* in the *d3matlab* tools directory). You should have received the bench cal file when you got the tag but if not, look on the soundtags website (see below). To do this, you will need the name or ID code of the tag used for the deployment. You can find the ID code by re-running d3deployment with two output arguments, e.g.,

[CAL,D] = d3deployment('/tag/data/berardius','bb214a','bb10_214a') ;

Type D and look for an ID field with the tag 8-character code. Once you have the right bench cal file, read it into Matlab using:

CAL = d3findcal(id) ;

where id is the 8-character code in a string e.g.,

CAL = d3findcal('e303b342') ;

4. Optimize the pressure calibration:

[p,CAL]=d3calpressure(X,CAL,'full');

This opens a dive profile figure in which you can select a subset of the deployment over which to perform the calibration. To move the time bars, position the cursor in the figure and type *l* or *r* to move the left or right bar. When you are satisfied, press the enter key. The same graphical interface as for the DTAG-2 will open in a new figure allowing you to select pressure points that correspond to surfacings. When you have completed selecting points, you are given the opportunity to accept the calibration, return to the dive profile window to select a different time interval or to quit without changing the calibration.

5. Optimize the acceleration calibration:

[A,CAL,fs] = d3calacc(X,CAL,'full',**min_depth**) ;

As for d3calpressure, you will be shown a dive profile window allowing you to select a subset of the data. Use the *l* and *r* keys to move the time bars and then press *enter* to continue. Automatic calibration will proceed over the time interval you have selected. The optional **min_depth** argument allows you to restrict automatic calibration to when the animal is below the surface. This reduces noise and results in a better calibration. For animals that routinely dive to 10's of meters, use a **min_depth** of 10 (meters). For very shallow diving animals, try a **min_depth** of 5.

6. Optimize the magnetometer calibration:

[M,CAL] = d3calmag(X,CAL,'full',**min_depth**) ;

As for d3calacc, you will be shown a dive profile window to select a subset of the data and then automatic calibration will proceed. The optional **min_depth** argument allows you to restrict automatic calibration to when the animal is below the surface.

As a result of the last three steps, you will have *A* and *M* matrices (3-columns each) and a *p* vector, all of the same length and sampled at *fs* Hz. These data are now compatible with the remainder of the DTAG-2 tools.

7. Save the calibration information so far to the deployment CAL file:
*d3savecal(**deploy_name**,'CAL',CAL)*

Example:
d3savecal('bb10_214a','CAL',CAL)

8. Estimate the orientation of the tag on the whale in the same way as for DTAG-2, i.e., using *prhpredictor* to generate an orientation table (OTAB). See *tag2whale.pdf* for information on these functions. Also type *help prhpredictor* and *help tag2whale* in Matlab.

PRH = prhpredictor(p,A,fs,[TH,METHOD,DIR])
[Aw,Mw] = tag2whale(A,M,OTAB,fs)

9. Save the OTAB for the deployment.
*d3savecal(**deploy_name**,'OTAB',OTAB)*

10. Add other optional information to the deployment calibration file, e.g.:

d3savecal('mn12_198a','LOCATION','Greenland')
d3savecal('mn12_198a','TAGON.POSITION',[67.4567 -15.3456])
d3savecal('mn12_198a','DECL',-30.5)

Use standard names for this metadata - see the suggestions in the document *d3sensor_data.pdf*.

11. Apply the calibration at different sensor sampling rates as required.
*d3makeprh(**deploy_name**,**df**)*

Example:
d3makeprh('mn12_198a',10)

where df is the decimation factor to apply. This will read the raw sensor data, decimate it by **df** and apply the calibration constants in the deployment calibration file. A *prh* file will be generated with the name ***deploy_name****prh****nn****.mat*, where **nn** is the sampling rate in Hz e.g., *mn12_198aprh5.mat*. The prh file will be stored in the directory that you defined with *settagpaths.m* for prh data.

Usage

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. The tools are in constant development and are not guaranteed to perform any particular function. To ensure that you have the latest tools and are aware of any short-comings, check on the soundtags website: *http://soundtags.st-andrews.ac.uk*. Contact the author (markjohnson@st-andrews.ac.uk) to report any errors or modifications.