MATLAB tools for processing and plotting ASL AZFP data

1. Important notes:

The intent is for owners of AZFP's and AZFP data to share and improve scientific post-processing software free of charge. The user of the software assume responsibility for the results.

Programmers are asked to follow best practices and include lots of comments and notes in their programs. Programmers and users are asked to maintain a log of known problems, bugs and associated fixes. ASL Environmental Sciences hopes to contribute to the software as well.

2. Versions:

Version	Date	Author	Comments
1.0	Sept 30, 2016	Dave Billenness, ASL	Initial release
1.1	Oct 31, 2016	Dave Billenness, ASL	Now properly loads data if only some frequencies/channels are collected with a multi-freq instrument (eg. frequency 2 only is collected on a 4 frequency instrument). Fixed bug if #bins change over multi-freq the program gave an error.
1.2	Mar 10, 2017	Dave Billenness, ASL	Fixed loading a directory containing Echoview .evi files Minor xmlfilename changes for loading large datasets for data cubes (stop prompting for loading xml)
1.3	May 2018	Dave Billenness, ASL	Improve plots of temp/tilts and handling of plotting of noisefloor

3. Required files:

ParametersAZFP.m – a MATLAB script file that contains the user settings for loading, averaging and plotting the AZFP data.

ProcessAZFP.m – a MATLAB function that combines the hourly files that are averaged in depth and time. The data is saved to a structured array variable in MATLAB. The variable and plot settings are passed to the PlotAZFP function (if plotting is selected in the parameter settings).

LoadAZFP.m – a MATLAB function that loads an hourly binary AZFP file (*.001A, etc) and an instrument coefficient xml file. It computes hourly average temperature, sound speed and absorption coefficients (for each frequency). Based on the user input, it averages the range bins and time averages. The data is converted to Sv and Ts (volume backscatter and target strength). This function is called by ProcessAZFP.m.

PlotAZFP.m – a MATLAB function that plots the structured array variable that is output from the combined files from ProcessAZFP. This function is called by ProcessAZFP.m or it can be run on its own.

AZFPColormap.mat – colour scheme settings for the echogram plots used by PlotAZFP.m. The colour scheme can be changed using MATLAB's built in colormapeditor function and saving the new color map.

4. Steps:

a) Edit the settings in the file **ParametersAZFP.m:**

```
% Default values shown are used if a parameter is missing
%
% FILE LOADING AND AVERAGING:
% Parameters.ProcDir = 0; 1 will prompt for an entire directory to
% process, = 0 will prompt to load individual files in a directory
Parameters.ProcDir = 1;

% Parameters.datafilename = ''; % '' will prompt for hourly AZFP
% file(s) to load, example '16010100.01A'
Parameters.datafilename = '';

% Parameters.xmlfilename = ''; % prompt for XML filename if no XML file exists
% in the directory, example '15101614.XML'
Parameters.xmlfilename = '';

% Parameters.Salinity = 35; % Salinity in psu
Parameters.Salinity = 32;

% Parameters.Bins2Avg = 10; % number of range bins to average
Parameters.Bins2Avg = 10;

% Parameters.Time2Avg = 60; % number of time values to average
Parameters.Time2Avg = 10;

% Parameters.Pressure = 50; % in dbars (~ depth of instrument in meters).
% This can be approximate and is used in the soundspeed and absorption calc
Parameters.Pressure = 150;

% PLOTTING:
% Parameters.Plot = 1; % show an echogram plot for each channel
Parameters.Plot = 1;

% Parameters.Channel: freq to plot #1-4, default 1
Parameters.Channel = 1;

% Parameters.Value2Plot = 2; 1,2,3,4 = Counts, Sv, TS, Temperature+Tilts, default 2
Parameters.Value2Plot = 2;

% Parameters.NoiseFloor = 10000; % for Sv and Ts plotting only, values
% with counts < NoiseFloor will be set to -150, can use individual values
% for eash frequency, ex. 'NoiseFloor',[10000; 11000; 10500; 12500]
% Default = 10000.
Parameters.NoiseFloor = [9500; 9500; 9500; 9500];

% Parameters.Orientation = 0 instrument on bottom looking up (range bins), 1 at surface
% looking down (depth bins). This changes the ydir on the echogram plots only. Default is 1.
Parameters.Orientation = 1;

% Parameters.UseTiltCorr = 0; Use the tilt corrected ranges for the echogram plots,
% default 0. Will give a warning if the tilt magnitudes are unreasonable (> 20 deg)
Parameters.UseTiltCorr = 0;
```

This script can be renamed, for example “ParametersAZFP_Bin10Time60.m”. If a parameter is omitted, then the default value will be used.

b) Run the processing function **ProcessAZFP.m:**

In MATLAB:

```
>> ParametersAZFP; %(load the parameters into the MATLAB workspace using the
                     saved script filename)
>> [Output,Par] = ProcessAZFP(Parameters); % pass in the variable Parameters
```

Depending on the settings in the script file from step 1, the user will be prompted to load a directory of hourly AZFP files (`Parameters.ProcDir = 1`) or individual hourly files, and also an instrument configuration xml file. If a single xml file exists in the data directory it will be loaded automatically.

The function averages an hourly file using the range bin and time averaging information in the parameter file. Hourly sound speed and absorption values are computed using the hourly averaged temperature and the input value of Salinity (psu) and Pressure (db). Sv and Ts are computed using the instrument settings in the xml file and the computed sound speed and absorption values.

An Sv offset is applied to compensate for the effects of the finite response times of both the receiving and transmitting parts of the instrument. The offset value ranges from +0.3 to +1.1, as described on page 89 of the AZFP Operator's Manual.

The distance to the centre of the range bins are calculated using the formula (11) given in the Operator's Manual on page 86. Range bins are calculated every hour based on the computed sound speed.

The structured array variable `Output` in the function **ProcessAZFP.m** is stored in the MATLAB workspace. Each hourly file is loaded, averaged, combined and then stored in the workspace. If no averaging is done, the `Output` variable will become very large and, depending on the amount of RAM in the PC, an Out of Memory error will be reported by MATLAB. When working with large datasets (8 GB+), it is recommended to use bin and time averaging, or load and plot smaller segments of the data.

The structured array variable `Output` is size N, where N is the number of AZFP frequencies collected. For example, the results after processing raw 4 frequency AZFP data with 3 second pings using a 10-ping time average and loading in 4 hours of data:

```
>> Output(1)
         Date: [480×1 double]
           Tx: [480×1 double]
           Ty: [480×1 double]
            T: [480×1 double]
     filename: {'12022316.01A' '12022317.01A' '12022318.01A' '12022319.01A'}
HourlyAvgTemp: [4×1 double]
   SoundSpeed: [4×1 double]
            N: [480×689 double]
        Range: [4×689 double]
TiltCorrRange: [4×689 double]
           Sv: [480×689 double]
           TS: [480×689 double]
       seaAbs: [4×1 double]
         Freq: 125
     Bins2Avg: 10
     Time2Avg: 10
     BurstInt: 3
PingPerProfile: 1
   NumAcqPings: 1
      DataType: [4×1 double]

>> whos
  Name              Size                Bytes  Class     Attributes

  Output            1x4             317828208  struct
  Parameters        1x1                  2400  struct
```

This results in a 31.9 MB `Output` variable (for all 4 frequencies) after loading in 4 hours of data.

For multi-frequency data sets, fields in the structured array are shown as empty for the fields that are the same for each frequency, for example:

```
>> Output(2)
              Date: []
                Tx: []
                Ty: []
                 T: []
          filename: []
     HourlyAvgTemp: []
        SoundSpeed: []
                 N: [480×689 double]
             Range: [4×689 double]
     TiltCorrRange: [4×689 double]
                Sv: [480×689 double]
                TS: [480×689 double]
            seaAbs: [4×1 double]
              Freq: 200
          Bins2Avg: []
          Time2Avg: []
          BurstInt: []
    PingPerProfile: []
       NumAcqPings: []
          DataType: []
```

The `Output` variable can be saved to a MATLAB binary *.mat file using:

```
>> save('AZFPdata','Output');
```

c) Plot the data:

The data can be plotted using the `Parameters.Plot = 1;` parameter during step b), or by passing the `Output` variable and the loaded Parameters from step a) to the plotting function:

```
>> PlotAZFP(Output,Parameters);
```

The AZFP frequency to plot (channels 1-4, depending on the instrument) and the value to plot (counts, Sv, Ts or tilts/temperature) can be set in the parameter file.

Once the data has been loaded, the other channels can be plotted using:

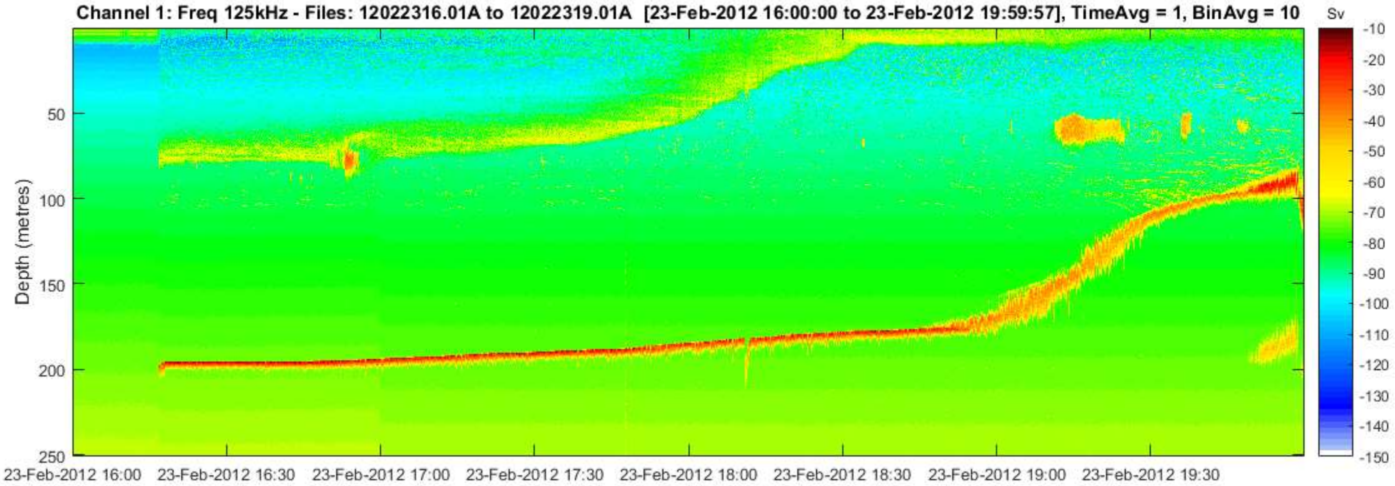
```
>> Parameters.Channel = 3;PlotAZFP(Output,Parameters);
```

A noise floor value (see page 88 of the Operator's Manual) can be set for each AZFP frequency in the parameter file. Values with counts less than the noise floor are set to -150 in the Sv and Ts echogram plots (the data is not changed).

The instrument orientation can be set in the parameter file (1=instrument on bottom looking up, 0 at surface looking down). This changes the y-axis direction on the echogram plots.

A tilt correction can be applied to the ranges using the parameter file setting. The hourly average of the cosine of the tilt magnitude is used to correct the ranges. The tilt corrected ranges are given in the `Output` variable regardless of the parameter file setting. The parameter file setting only changes the echogram plot to display the tilt corrected ranges. If the tilt magnitudes are unreasonable (>20 deg), a warning is given if the parameter is set to use the tilt magnitudes for the plots.

An Sv plot for channel 1 (125 kHz) using the sample data:


Channel 1: Freq 125kHz - Files: 12022316.01A to 12022319.01A [23-Feb-2012 16:00:00 to 23-Feb-2012 19:59:57], TimeAvg = 1, BinAvg = 10
Depth (metres)
23-Feb-2012 16:00
23-Feb-2012 16:30
23-Feb-2012 17:00
23-Feb-2012 17:30
23-Feb-2012 18:00
23-Feb-2012 18:30
23-Feb-2012 19:00
23-Feb-2012 19:30
Sv