

This document describes data processing for ship-mounted ADCP (R/V Pt. Sur)

Table of Contents

Section 1. RDI Workhorse Surveyor 300khz LTA data with 300 second ensembles collected during SBCLTER cruise 05.

Chris Gotschalk (gots@lifesci.ucsb.edu) 5/22/2006

Section 2. Ocean Surveyor 75khz LTA data with 300 second ensembles collected during SBCLTER cruise 11.

Chris Gotschalk (gots@lifesci.ucsb.edu) 5/22/2006

Section 3. RDI Workhorse Surveyor 300khz LTA data with 300 second ensembles collected during SBCLTER cruise 15.

Chris Gotschalk (gots@lifesci.ucsb.edu) 5/22/2006

SECTION 1. RDI Workhorse Surveyor 300khz LTA data with 300 second ensembles collected during SBCLTER cruise 05.

Document prepared by Chris Gotschalk (gots@lifesci.ucsb.edu)
5/22/2006

This document describes all the steps performed while processing ship mounted ADCP data. This example is for RDI Workhorse Surveyor 300khz LTA data with 300 second ensembles collected during SBCLTER cruise 15.

Print out the page located at for an example of what you're going to do:
http://currents.soest.hawaii.edu/docs/adcp_doc/quickpy_doc/pingdata_commands.txt

To set up the main directory (ex. lter05_300) and processing tree, run from the command line adcptree.py in a base directory (ex. /data02/users/gots/)

```
$ adcptree.py lter05_300
```

Copy all the PINGDATA files into /lter05_300/ping/

cd into /lter05_300/ and run quick_adcp with the following switches:

```
$ quick_adcp.py --yearbase 2002 --dbname alter05 --use_refsm  
--auto (if you dont want to be prompted at each step)
```

There are a number of files that have been modified from the original versions or are unique to my processing routines. These files have been stored in the directory /data02/users/gots/newCODASmfiles/ and should be copied as follows:

```
$ cd /data02/users/gots/newCODASmfiles/
```

```
$ cp aflagit_setup.m get_udas.m apply_flags.m ../lter05_300/edit/
```

Download and print out /cal/rotate/alter05.ps to check navigational data coverage.

Bottom track calibration ...

Check for the availability of bottom track info in /cal/botmtrk/btcaluv.out.

Inspect the plot btcaluv.ps.

In Matlab cd to /cal/botmtrk/ and run the mfile btcaluv (3 times) with the following syntax (lter15_300 example) for 3 window sizes (1,2,3):

```
>> [a,pg,gd_mask]=btcaluv('alter05.ref','alter05.btm','step size = 1',1);  
>> [a,pg,gd_mask]=btcaluv('alter05.ref','alter05.btm','step size = 2',2);  
>> [a,pg,gd_mask]=btcaluv('alter05.ref','alter05.btm','step size = 3',3);
```

Results are appended to btcaluv.out. Inspect and choose phase and amplitude correction (smallest SD? see demo process.txt for help choosing).

Watertrack calibration ...

Look for water track info in /cal/watertrk/adcpal.out.

Play around with different window sizes (5,7,9) in timslip.tmp. Generate an output file for each window size and test the output stats with adcpal.m

In Matlab cd to /cal/watertrk/ and run adcpal.m like a dis:

```
>> adcpal('alter05_5.cal','window = 5');  
>> adcpal('alter05_7.cal','window = 7');  
>> adcpal('alter05_9.cal','window = 9');
```

Each run of adcpal.m will append to adcpal.out so no need to write things down. Inspect and choose a good phase and amplitude correction.

We had both bottom track and water track info for this cruise. An angle rotation of +1.83 degrees and an amplitude correction of 1.02 will be applied.

Now we generate temperature and salinity correction files using the Underway Data Acquisition System (UDAS).

First step is to concatenate all the raw .txt files.

```
$ cat *.txt>lter05_all_udas.txt
```

Each file will have a one line header so we have to go into the file and:

---> print out a list of column designations for this cruise and then

---> remove all the header lines

---> comma separate the date mm/dd/yyyy -> mm,dd,yyyy,
and time HH:MM -> HH,MM,

---> check for odd time stamp issues especially around midnight or when a new file has been generated. Search and remove the dates 0/0/00 (these regularly exist).

Move this file to the /edit/ subdirectory. In the mfile get_udas.m, edit the column designations for time, date, temp and

salinity, and run `get_udas.m` in matlab to generate the correction files `corrsal.dat` and `corrtemp.dat`.

Edit `fix_temp.cnt` to include proper database name and file names for `corrsal.dat` and `corrtemp.dat` corrections:

```
/******  
dbname: ../adcpdb/alter05  
original_soundspeed= 1500.0 /* set to this value in DAS */  
true_temperature= corrtemp.dat /* using UDAS to correct the ADCP record */  
true_salinity= corrsal.dat  
end  
time_ranges:  
all  
*****
```

and run on command line:

```
$ fix_temp fix_temp.cnt
```

Ok. We've got all our ducks in a row and can now run through `quick_adcp` for the second time to adjust the rotation. For this example we want to rotate the database phase +1.83 degrees and amplitude by 1.02.

On the command line `cd` to the main directory (ex. `Iter05_300`) and enter:

```
$ quick_adcp.py --yearbase 2002 --use_refsm --rotate_angle 1.83 --rotate_amplitude 1.02 --steps2rerun  
rotate:navsteps:calib --auto
```

If all runs smoothly we can move on to edit the database in `gautoedit`.

Gee! Autoedit (`gautoedit.m`)

After the second pass through `quick_adcp` we're ready to go through `gautoedit` processing. Here we edit bad bins, profiles, and check for realistic bottom flagging. The mfiles `aflagit_setup.m` and `asetup.m` are provided automatically from `adcp`tree with default values. A modified version of `aflagit_setup.m` can be moved from `/newCODASmfiles/` and will keep us consistent from cruise to cruise. The mfile `asetup.m` needs to be modified in `/edit/` prior to the `gautoedit` session to update yearbase, some file names and paths.

Very nice instructions for using `gautoedit` exist at:

http://currents.soest.hawaii.edu/docs/adcp_doc/edit_doc/edit_html/autoedit.html

After the `gautoedit` session the edits must be applied to the database with a 3rd pass of `quick_adcp` with the following switches:

```
$ quick_adcp.py --yearbase 2002 --use_refsm  
--steps2rerun apply_edit:navsteps:calib:matfiles --auto
```

So. Now what?

Here's what you have to do to get the 5 minute averaged data and all the bins (flagged as good) into a .mat file.

Instructions can be found at:

[http://currents.soest.hawaii.edu/docs/adcp_doc/UHDAS_webdoc/...
Processing/uhdas_matlab_access.html](http://currents.soest.hawaii.edu/docs/adcp_doc/UHDAS_webdoc/...Processing/uhdas_matlab_access.html)

but here are the highlights.

In Matlab the following will extract all the 5 minute profiles:

```
>> [alldata,config]= run_agetmat('ddrange', [-60 390],...  
    'editdir', '/data02/users/gots/lter05_300/edit');
```

Then apply the flags from gautoedit with apply_flags.m which you've copied into the /edit/ subdirectory earlier.

```
>> data = apply_flags(alldata, alldata.pflag); %edit bad points
```

The definitions for all the structure fields can be found at:

[http://currents.soest.hawaii.edu/docs/adcp_doc/UHDAS_webdoc/...
Processing/agetmat_output.txt](http://currents.soest.hawaii.edu/docs/adcp_doc/UHDAS_webdoc/...Processing/agetmat_output.txt)

You're done :) Save out the structures [data,config]

```
>> save('/data02/users/gots/lter05_300/lter05_300_final.mat','data','config');
```

and move entire directory to a secure spot.

SECTION 2. Ocean Surveyor 75khz LTA data with 300 second ensembles collected during SBCLTER cruise 11.
Chris Gotschalk (gots@lifesci.ucsb.edu) 5/22/2006

Document prepared by Chris Gotschalk (gots@lifesci.ucsb.edu)
5/22/2006

This document describes all the steps performed while processing ship mounted ADCP data. This example is for

Ocean Surveyor 75khz LTA data with 300 second ensembles collected during SBCLTER cruise 11.

Create a scratch directory to deposit the raw data files:

/data02/users/gots/temp11_75/ for example.

Copy the raw 75khz data files with suffix LTA, ENS, N2R, and VMO to this scratch directory,

Print out the page located at:

http://currents.soest.hawaii.edu/docs/adcp_doc/quickpy_doc/LTA_commands.txt

To set up the main directory (ex. lter11_75) and processing tree, run from the command line adcptree in a base directory (/data02/users/gots/ = /root/) with the following switches:

```
$ adcptree.py lter11_75 --instclass os --datatype lta
```

Copy all the .LTA files into /root/lter11_75/ping/, cd into /lter11_75/ and run quick_adcp with the following switches:

```
$ quick_adcp.py --yearbase 2004 --dbname alter11 --use_refsm --instname os75 --instclass os --datatype lta --
datafile_glob *.LTA
--auto (if you dont want to be prompted at each step)
```

There are a number of files that have been modified from the original versions or are unique to my processing routines. These files have been stored in the directory /data02/users/gots/newCODASmfiles/ and should be copied as follows:

```
$ cd /data02/users/gots/newCODASmfiles/
```

```
$ cp aflagit_setup.m asetup.m get_udas.m apply_flags.m ../lter11_75/edit/
```

```
$ cp get_headcorr.m lookinatENSfiles.m mk_avgdh.m codaspaths.m ../lter11_75/cal/rotate/
```

Open Matlab in a VNC window and cd to /cal/rotate/.

Run codaspaths.m,

Edit and run the mfile lookinatENSfiles.m

```
%%%%%%%%%
```

```
figure
hold on
```

```
ensfiles = dirs(fullfile(ensdir,'*.ENS'))
```

```
for filei = 1:length(ensfiles)
    filename = ensfiles(filei).name
```

```
    % NOTE: use os below for Ocean Surveyor 75khz
    %       or bb for WH 300 khz
    data=read(os,[ensdir,'/',filename], 'vars','all');
    dd=restruct(os,data);
```

```
    plot(dd.dday, dd.heading)
end
```

%%%%%%%%%

Take a look at the gyro data plot. Data should range from 0-360 degrees and there shouldn't be any odd time stamp issues. If all seems well, move on

The next several steps are required to generate the gyro heading correction angle file (ex. LTER11_head_corr.ang)

Create a directory called rbin in /cal/rotate/ and cd into it.

Run the Python script serasc2bin.py with the following switches. Note that for some reason this script isn't in my path (and I cant add it???) so I have to use the entire path at the command line:

```
$ /usr/local/pkgs/codas/logging/parsing/serasc2bin.py ...
-rv      ...
-y 2004  ...
-t vmdas ...
-c 'last' ... (note: with quotes)
-m adu   ...
/data02/users/gots/temp11_75/*.N2R
```

The output Ashtech headingt files (*.adu.rbin) will be deposited in the /rbin/ directory.

Next we need to run a short section of code to generate the gyro .ehd.rbin files. In Matlab run this portion of get_headcorr.m. I'll probably make this into it's own little mfile someday.

From the /rbin/ directory run:

```
%%%%%%%%%
ensdir='/data02/users/gots/temp11_75';

ensfiles = dirs(fullfile(ensdir,'*.ENS'));

% convert each file into rbins
for filei = 1:length(ensfiles)
    filename = ensfiles(filei).name
    ensgyro2rbin(os,fullfile(ensdir,filename));
end
```

%%%%%%%%%

Comment out this portion of get_headcorr.m and from the /cal/rotate/ directory run the remaining lines. A few plots as well as the heading correction angle file lter11_head_corr.ang will be output. Inspect the output visually by plotting /cal/rotate/gyrodh_stats.ps.

Edit /cal/rotate/rotate.tmp for the proper year_base and time_angle_file name.

```
/*
DB_NAME:    ../adcpdb/alter11
LOG_FILE:   rotate.log
TIME_RANGE: 2004/09/08 21:07:28 to 2004/09/09 05:22:31 /* only from scanned files */

OPTION_LIST:
```

```

water_and_bottom_track:
  year_base=      2004
  time_angle_file: lter11_head_corr.ang
  amplitude=      1
  angle_0=        0
end
end

```

/*****/

This will be used during the next pass with quick_adcp.py.

Bottom track calibration ...
 Check for bottom track info in /cal/botmtrk/btcaluv.out.
 Inspect the plot btcaluv.ps.

In Matlab cd to /cal/botmtrk/ and run the mfile btcaluv (3 times) with the following syntax (lter11_75 example) for 3 window sizes (1,2,3):

```

>> [a,pg,gd_mask]=btcaluv('alter11.ref','alter11.btm','step size = 1',1);
>> [a,pg,gd_mask]=btcaluv('alter11.ref','alter11.btm','step size = 2',2);
>> [a,pg,gd_mask]=btcaluv('alter11.ref','alter11.btm','step size = 3',3);

```

Results are appended to btcaluv.out. Inspect and choose phase and amplitude correction (smallest SD? see demo process.txt for help choosing).

Watertrack calibration ...
 Look for water track info in /cal/watertrk/adcpcal.out.

Play around with different window sizes (5,7,9) in timslip.tmp. Generate an output file for each window size and test the output stats with adcpcal.m

In Matlab cd to /cal/watertrk/ and run adcpcal.m like a dis:

```

>> adcpcal('alter11_5.cal','window = 5');
>> adcpcal('alter11_7.cal','window = 7');
>> adcpcal('alter11_9.cal','window = 9');

```

Each run of adcpcal.m will append to adcpcal.out so no need to write things down. Inspect and choose a good phase and amplitude correction.

Now we generate temperature and salinity correction files using the Underway Data Acquisition System (UDAS).

First step is to concatenate all the .txt files.

```
$ cat *.txt>lter11_all_udas.txt
```

Each file will have a one line header so we have to go into the file and:

```

---> print out a list of column designations for this cruise and then
---> remove all the header lines
---> comma separate the date mm/dd/yyyy -> mm,dd,yyyy,
      and time    HH:MM -> HH,MM,

```

--> check for odd time stamp issues especially around midnight or when a new file has been generated. Search and remove the dates 0/0/00 (these regularly exist).

Move this file to the /edit/ subdirectory. In the mfile get_udas.m, edit the column designations for time, date, temp and salinity, and run get_udas.m in matlab to generate the correction files corrsal.dat and corrtemp.dat.

NOTE: The output file lter11_all_udas.txt is used for both the 75 and 300khz data but the mfile get_udas.m needs to be run for both data sets and will generate different corr*.dat files based on the availability of adcp ensemble times.

Edit fix_temp.cnt to include proper database name and file names for corrsal.dat and corrtemp.dat corrections:

```
/*-----*/
dbname: ../adcpdb/alter11
original_soundspeed= 1500.0 /* set to this value in DAS */
true_temperature= corrtemp.dat /* using UDAS to correct the ADCP record */
true_salinity= corrsal.dat
end
time_ranges:
all
/*-----*/
```

and run on command line:

```
$ fix_temp fix_temp.cnt
```

Ok. We've got all our ducks in a row and can now run through quick_adcp for the second time to adjust the rotation. For this example we want to rotate the database -0.76 degrees. No amplitude adjustment is needed for this dataset.

On the command line cd to the main directory (ex. lter11_75) and enter:

```
$ quick_adcp.py --yearbase 2004 --use_refsm --instname os75 --rotate_angle -0.76 --steps2rerun
rotate:navsteps:calib --auto
```

If all runs smoothly we can move on to edit the database in gautoedit.

Gee! Autoedit (gautoedit.m)

After the second pass through quick_adcp we're ready to go through gautoedit processing. Here we edit bad bins, profiles, and check for realistic bottom flagging. The mfiles aflagit_setup.m and asetup.m are provided automatically from adcp tree with default values. A modified version of aflagit_setup.m can be moved from /newCODASmfiles/ and will keep us consistent from cruise to cruise. The mfile asetup.m needs to be modified in /edit/ prior to the gautoedit session to update yearbase, some file names and paths.

Very nice instructions for using gautoedit exist at:

http://currents.soest.hawaii.edu/docs/adcp_doc/edit_doc/edit_html/autoedit.html

After the gautoedit session the edits must be applied to the database with a 3rd pass of quick_adcp with the following switches:

```
$ quick_adcp.py --yearbase 2004 --use_refsm --instname os75 --steps2rerun apply_edit::navsteps:calib:matfiles --auto
```

So. Now what?

Here's what you have to do to get the 5 minute averaged data and all the bins (flagged as good) into a .mat file.

Instructions can be found at:

[http://currents.soest.hawaii.edu/docs/adcp_doc/UHDAS_webdoc/...
Processing/uhdas_matlab_access.html](http://currents.soest.hawaii.edu/docs/adcp_doc/UHDAS_webdoc/...Processing/uhdas_matlab_access.html)

but here are the highlights.

In Matlab the following will extract all the 5 minute profiles:

```
>> [alldata,config]= run_agetmat('ddrange', [-60 390],...  
    'editdir', '/data02/users/gots/lter11_75/edit');
```

Then apply the flags from gautoedit with apply_flags.m which you've copied into the /edit/ subdirectory earlier.

```
>> data = apply_flags(alldata, alldata.pflag); %edit bad points
```

The definitions for all the structure fields can be found at:

[http://currents.soest.hawaii.edu/docs/adcp_doc/UHDAS_webdoc/...
Processing/agetmat_output.txt](http://currents.soest.hawaii.edu/docs/adcp_doc/UHDAS_webdoc/...Processing/agetmat_output.txt)

You're done :) Save out the structures [data,config]

```
>> save('/data02/users/gots/lter11_300/lter11_75_final.mat','data','config');
```

and move entire directory to a secure spot.

Section 3. RDI Workhorse Surveyor 300khz LTA data with 300 second ensembles collected during SBCLTER cruise

15.

Chris Gotschalk (gots@lifesci.ucsb.edu) 5/22/2006

Document prepared by Chris Gotschalk (gots@lifesci.ucsb.edu)

5/22/2006

This document describes all the steps performed while processing ship mounted ADCP data. This example is for RDI Workhorse Surveyor 300khz LTA data with 300 second ensembles collected during SBCLTER cruise 15.

Create a scratch directory to deposit the raw data files:

/data02/users/gots/temp15_300/ for example.

Copy the raw 300khz data files with suffix LTA, ENS, N2R, and VMO to this scratch directory. Most of these files won't be used during processing but some may be needed should problems arise. We'll use the N2R files to check the Ashtech heading data coverage.

Print out the page located at:

http://currents.soest.hawaii.edu/docs/adcp_doc/quickpy_doc/LTA_commands.txt

To set up the main directory (ex. lter15_300) and processing tree, run from the command line adcpree in a base directory (/data02/users/gots/ = /root/) with the following switches:

```
$ adcpree.py lter15_300 --instclass bb --datatype lta
```

Copy all the .LTA files into /root/lter15_300/ping/

cd into /lter15_300/ and run quick_adcp with the following switches:

```
$ quick_adcp.py --yearbase 2006 --dbname alter15 --use_refsm --instname wh300 --instclass bb --datatype lta --
datafile_glob *.LTA
--auto (if you dont want to be prompted at each step)
```

There are a number of files that have been modified from the original versions or are unique to my processing routines. These files have been stored in the directory /data02/users/gots/newCODASmfiles/ and should be copied as follows:

```
$ cd /data02/users/gots/newCODASmfiles/
```

```
$ cp aflagit_setup.m asetup.m get_udas.m apply_flags.m ../lter15_300/edit/
```

```
$ cp get_headcorr.m lookinatENSfiles.m lookinatN2Rfiles.m mk_avgdh.m codaspaths.m ../lter15_300/cal/rotate/
```

!!!!NOTE!!!!!!!!!!!!!!

The processing for the 300 khz adcp differs here from the os75 due to the absence of any gyro data. For some reason the Ashtech is used as the primary heading device. I've created an mfile lookinatN2Rfiles.m to plot up these headings and allow us to check for coverage during the cruise. (we may want to add a routine to look at N2R files from the os75 and compare to the 300)

!!!!!!!!!!!!!!!!!!!!!!

Create a directory rbin (/cal/rotate/rbin/) and cd into it.

Run the Python script serasc2bin.py with the following switches. Note that for some reason this script isn't in my path (and I cant add it???) so I have to use the entire path at the command line:

Enter:

```
$ /usr/local/pkg/codas/logging/parsing/serasc2bin.py ...  
-rv ...  
-y 2006 ...  
-t vmdas ...  
-c 'last' ... (note: with quotes)  
-m adu ...  
/data02/users/gots/temp15_300/*.N2R
```

The output Ashtech headingt files (*.adu.rbin) will be deposited in the /rbin/ directory.

Next, in Matlab, cd to the /cal/rotate/ directory and run the mfile lookinatN2Rfiles.m. This mfile will plot up the Ashtech data within the .adu.rbin files generated above. Inspect to see that the range looks ok (0-360) and note any gaps (red data coverage line). If all looks ok, move on to the bottom track and water track calibrations.

Bottom track calibration ...

Check for the availability of bottom track info in /cal/botmtrk/btcaluv.out.
Inspect the plot btcaluv.ps.

In Matlab cd to /cal/botmtrk/ and run the mfile btcaluv (3 times) with the following syntax (lter15_300 example) for 3 window sizes (1,2,3):

```
>> [a,pg,gd_mask]=btcaluv('alter15.ref','alter15.btm','step size = 1',1);  
>> [a,pg,gd_mask]=btcaluv('alter15.ref','alter15.btm','step size = 2',2);  
>> [a,pg,gd_mask]=btcaluv('alter15.ref','alter15.btm','step size = 3',3);
```

Results are appended to btcaluv.out. Inspect and choose phase and amplitude correction (smallest SD? see demo process.txt for help choosing).

Watertrack calibration ...

Look for water track info in /cal/watertrk/adcpcal.out.

Play around with different window sizes (5,7,9) in timslip.tmp. Generate an output file for each window size and test the output stats with adcpcal.m

In Matlab cd to /cal/watertrk/ and run adcpcal.m like a dis:

```
>> adcpcal('alter15_5.cal','window = 5');  
>> adcpcal('alter15_7.cal','window = 7');  
>> adcpcal('alter15_9.cal','window = 9');
```

Each run of adcpcal.m will append to adcpcal.out so no need to write things down. Inspect and choose a good phase and amplitude correction.

We had both bottom track and water track info for this cruise. An angle rotation of -1.68 degrees and 0 amplitude correction will be applied.

Now we generate temperature and salinity correction files using the Underway Data Acquisition System (UDAS).

First step is to concatenate all the .txt files.

```
$ cat *.txt>lter15_all_udas.txt
```

Each file will have a one line header so we have to go into the file and:

---> print out a list of column designations for this cruise and then

---> remove all the header lines

---> comma separate the date mm/dd/yyyy -> mm,dd,yyyy,
and time HH:MM -> HH,MM,

---> check for odd time stamp issues especially around midnight or when a new file has been generated. Search and remove the dates 0/0/00 (these regularly exist).

Move this file to the /edit/ subdirectory. In the mfile get_udas.m, edit the column designations for time, date, temp and salinity, and run get_udas.m in matlab to generate the correction files corrsal.dat and corrtemp.dat.

!!!!!!!!!!!!!!!!!!!!!!!!!!!!

NOTE: The output file lter11_all_udas.txt is used for both the 75 and 300khz data but the mfile get_udas.m needs to be run for both data sets and will generate different corr*.dat files based on the availability of adcp ensemble times.

!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Edit fix_temp.cnt to include proper database name and file names for corrsal.dat and corrtemp.dat corrections:

```
/*-----*/  
dbname: ../adcpdb/alter15  
original_soundspeed= 1500.0 /* set to this value in DAS */  
true_temperature= corrtemp.dat /* using UDAS to correct the ADCP record */  
true_salinity= corrsal.dat  
end  
time_ranges:  
all  
/*-----*/
```

and run on command line:

```
$ fix_temp fix_temp.cnt
```

Ok. We've got all our ducks in a row and can now run through quick_adcp for the second time to adjust the rotation. For this example we want to rotate the database -1.68 degrees. No amplitude adjustment is needed for this dataset.

On the command line cd to the main directory (ex. lter15_300) and enter:

```
$ quick_adcp.py --yearbase 2006 --use_refsm --instname wh300 --rotate_angle -1.68 --steps2rerun  
rotate:navsteps:calib --auto
```

If all runs smoothly we can move on to edit the database in gautoedit.

Gee! Autoedit (gautoedit.m)

After the second pass through quick_adcp we're ready to go through gautoedit processing. Here we edit bad bins, profiles, and check for realistic bottom flagging. The mfiles aflagit_setup.m and asetup.m are provided automatically from adcp tree with default values. A modified version of aflagit_setup.m can be moved from /newCODASmfiles/ and will keep us consistent from cruise to cruise. The mfile asetup.m needs to be modified in /edit/ prior to the gautoedit session to update yearbase, some file names and paths.

Very nice instructions for using gautoedit exist at:

http://currents.soest.hawaii.edu/docs/adcp_doc/edit_doc/edit_html/autoedit.html

After the gautoedit session the edits must be applied to the database with a 3rd pass of quick_adcp with the following switches:

```
$ quick_adcp.py --yearbase 2006 --use_refsm --instname wh300 --steps2rerun apply_edit:navsteps:calib:matfiles --auto
```

So. Now what?

Here's what you have to do to get the 5 minute averaged data and all the bins (flagged as good) into a .mat file.

Instructions can be found at:

[http://currents.soest.hawaii.edu/docs/adcp_doc/UHDAS_webdoc/...
Processing/uhdas_matlab_access.html](http://currents.soest.hawaii.edu/docs/adcp_doc/UHDAS_webdoc/...Processing/uhdas_matlab_access.html)

but here are the highlights.

In Matlab the following will extract all the 5 minute profiles:

```
>> [alldata,config]= run_agetmat('ddrange', [-60 390],...  
    'editdir', '/data02/users/gots/lter15_300/edit');
```

Then apply the flags from gautoedit with apply_flags.m which you've copied into the /edit/ subdirectory earlier.

```
>> data = apply_flags(alldata, alldata.pflag); %edit bad points
```

The definitions for all the structure fields can be found at:

[http://currents.soest.hawaii.edu/docs/adcp_doc/UHDAS_webdoc/...
Processing/agetmat_output.txt](http://currents.soest.hawaii.edu/docs/adcp_doc/UHDAS_webdoc/...Processing/agetmat_output.txt)

You're done :) Save out the structures [data,config]

```
>> save('/data02/users/gots/lter15_300/lter15_300_final.mat','data','config');
```

and move entire directory to a secure spot.

