**Notes on sonargui\_nsl\_inp.m and .fig**

This program is ETM mods starting with sonargui, written by CRS. He did all the hard math, I prettied it up some.

* The name means all the save and load commands were removed, and now it only uses guidata(hObject,handles) to save things from the callbacks in handles.
  + The save button is the only way to save tweaks you’ve made, but there is no way to load from the saved file yet.
* And that it needs input parameters to show anything more than one file from UNH
  + The files in the azdata box are determined by the contents of a directory
  + The values of the sliders are determined by the values read from the geom file for the experiment
* The axis limits edit boxes all work now
* The swap of views doesn’t just operate on some elements in the display
* When you choose a new experiment, it resets to the tripod view and defaults to not plotting azimuth data
* The “Apply this orientation” button operates on everything shown

Usage**: sonargui\_nsl\_inp (path\_names, expnames, fcns)**

Where the arguments are gotten from sonargui\_inputs.m. The user needs to edit sonargui\_inputs.m to have the names, paths and geom files correct for the experiments they want to view. Then execute it to formulate the arguments needed. This is what’s in mine to set up for 3 experiments:

addpath 'C:\Users\emontgomery\Documents\GitHub\sonargui\geom'

input\_struct(1).dpath='c:\home\data\unh\sonar\_data\';

input\_struct(1).function='unh\_geom'; % must be in the geom directory above

input\_struct(1).exname='UNH Tank';

input\_struct(2).exname='MVCO 2007';

input\_struct(2).dpath='c:\home\data\mvco\_07\sonar\_data\';

input\_struct(2).function='mvco\_geom'; % make sure this is the $MATLABPATH

input\_struct(3).exname='Fire Island 2012';

input\_struct(3).dpath='c:\home\data\FI2012\sonar\_post\Iris\_az\';

input\_struct(3).function='FI12\_geom'; % make sure this is the $MATLABPATH

%Now put the structure items into cell arrays

for ik=1:length(input\_struct)

path\_names(ik)=cellstr(input\_struct(ik).dpath);

expnames(ik)=cellstr(input\_struct(ik).exname);

fcns(ik)=cellstr(input\_struct(ik).function);

end

Note that all the geom.m files are expected to be in sonargui\geom directory now. I have one azimuth file from the UNH tank experiment in the directory with the programs to allow the default case to work, even if no arguments are given.

Even though 3 arguments are passed, once in the gui they are all part of varargin, (and must be cell arrays), so in places like the experiment chooser popup, you now just read the index from the popup and use it to index into the handle element that came from varargin (no more cases). In the program, **handles** has these elements to use by functions looking for a path or geom file to use:

fcns: {'unh\_geom' 'mvco\_geom' 'FI12\_geom'}

expnames: {'UNH Tank' 'MVCO 2007' 'Fire Island 2012'}

dpath: {1x3 cell}

Also note that in plot\_azdata2.m, the huge case statement was removed and the function now just uses the value in the box combined with the path to find the file to read and plot.

To do:

* Add a load button that will plot the saved settings
* Make color scale in plots consistent for all
* Add a colorbar

Questions:

* What is the definition of 0 when “use this compass” is on?
  + Find out how adcp beam 3 relates to the compass heading in the data
  + Learn how to compute the value to put in Hdg to get **North is UP**
* Figure how to compute azimuth yaw correctly

Handles notes:

All the stuff in the GUI is stored in a variable called handles. To allow information to be stored, the handles are persistent during operation, and the state of each GUI element is stored there. When you enter **guidata(hObject, handles)**, the state of all the handles is saved- so this needs to happen in most callbacks

Here’s how it works-when a gui thing is clicked, the callback for that element is executed, so the code you want to happen for each element has to go in its callback. Each element has its own handle entry- the box in the top left of sonargui is in handles.instrument\_popupmenu4. To find the information displayed in this element, you do:

instvals=get(handles.instrument\_popupmenu4,'String')

instvals =

'Tripod'

'ADCP (upward)'

'Fan beam sonar'

'Azimuth drive sonar'

''

Then to see which was selected do this:

idx=get(handles.instrument\_popupmenu4,'Value');

And you could see which element was selected by:

inst\_chosen=instvale(idx);

Another thing to be aware of is that we’ve stored a bunch of non-handle information in the handles variable. Ig you look at handles, the real handle things will have a number, and things we’ve added to store bits and pieces won’t. Also, only a real handle will work with a get or set command. The top things in the list are handle elements associated with gui things. Everything after “output” (blue background at the bottom) are things we’ve added for storage. You can find out the status of handles.experiment\_popupmenu6 using get and set operations, but that won’t work with an element like **expnames**.

handles =

figure1: 175.0038

text24: 221.0024

text23: 220.0024

load\_pushbutton6: 219.0037

plot\_z\_axis\_edit15: 218.0037

plot\_y\_axis\_edit14: 217.0037

text21: 216.0038

text20: 215.0038

text19: 214.0038

plot\_x\_axis\_edit13: 213.0038

text18: 212.0038

text17: 211.0038

tindex\_edit12: 210.0038

text16: 209.0038

experiment\_popupmenu6: 208.0038

usecompass\_radiobutton8: 207.0038

datetime\_edit11: 206.0038

showfloor\_radiobutton7: 205.0038

magvar\_edit10: 204.0038

text15: 203.0038

compass\_roll\_edit9: 202.0038

text13: 201.0038

text12: 200.0038

azdata\_popupmenu5: 199.0038

reset\_pushbutton5: 198.0038

save\_pushbutton4: 197.0038

azdrive\_radiobutton5: 196.0038

fanbeam\_radiobutton4: 195.0038

adcp\_radiobutton3: 194.0038

tripod\_radiobutton2: 193.0038

instrument\_popupmenu4: 192.0038

text11: 191.0038

text9: 190.0038

compass\_pitch\_edit8: 189.0038

text8: 188.0038

compass\_hdg\_edit7: 187.0038

text7: 186.0038

view\_popupmenu3: 185.0038

northing\_edit6: 184.0038

text6: 183.0038

northing\_slider6: 182.0038

easting\_edit5: 181.0038

text5: 17.0065

easting\_slider5: 16.0065

axes2: 11.0065

height\_text: 10.0065

edit\_height: 9.0065

edit\_roll: 8.0065

edit\_pitch: 7.0065

edit\_yaw: 6.0065

height\_slider: 5.0065

roll\_text: 4.0065

pitch\_text: 3.0065

yaw\_text: 2.0065

roll\_slider: 1.0065

pitch\_slider: 0.0090

yaw\_slider: 180.0038

FileMenu: 176.0038

CloseMenuItem: 179.0038

PrintMenuItem: 178.0038

OpenMenuItem: 177.0038

output: 175.0038

fcns: {'unh\_geom' 'mvco\_geom' 'FI12\_geom'}

expnames: {'UNH Tank' 'MVCO 2007' 'Fire Island 2012'}

dpath: {1x3 cell}

inst: 1

old\_inst: 1

experiment: {'UNH Tank'}

expname: {'UNH Tank'}

instloc: [1x5 struct]

tripod: [1x1 struct]

cmpss: [1x1 struct]

instval: {6x1 cell}

But if you want to know the content of handles.compass, you access it as you would any other structure.

hdg=handles.cmpss.pyy(3)

However, if you try to save handles outside matlab and bring it back using a command like this, you find the String and Value attributes are gone, even though there are still fields for all the elements you expect. If you

Hand\_save=handles;

save sonargui\_handles hand\_save; % note I avoid using the handles name

clear hand\_save

load sonargui\_handles %loads hand\_save

get(hand\_save..instrument\_popupmenu4,'Value');

gnereates an error

This syntax didn’t work either, so I think you probably have to be explicit button by button, or reformat it to save to a new geom file and re-load from there.

guidata(hObject,handles)

hand\_save=guidata(hObject);

save sonar\_nsl\_output.mat hand\_save

Here’s an annotated version of the GUI with the elements marked:

