NFWF Data Analysis:

March 22, 2020

(1) A number of global variables are sent in the file globalParameters.py

```
################################################################ GLOBAL
PARAMETERS
workingDirectory = "/home/val/Documents/NFWF_Files/2020_Analysis/"   #  Note Bene:
This MUST be set IN .py file BEFORE imports
modelDeltaSecs = 10   # interval between successive time steps
tortDamping = 6.0
minNumWhaleObs = 4    # need at least this many whale observations to constitute a
passby

maxObsGapMins = 15                # any larger gap between whale sitings will start
a new whale passby
maxPriorOrPostMinsForBoats = 5   # accept any boat obs from 5 min before focal
animal's first fix to 5 min after last one

whaleCVSfileName = "csvFiles/2003_2005_whalePassby.csv"
boatCVSfileName  = "csvFiles/2003_2005_boatPassby.csv"

minNumWhaleObs = 5   # don't save passby with less than 5 whale observations
ff_fileName =  "csvFiles/Whale Activity 2003-2005_EA quality
check_FINAL_all_sorted_tabbed.csv"
boatFileName = "csvFiles/All 2003-2005 boat data_EA
Quality_Final_sorted_tabbed.csv"

spreadingLaw = -18
backgroundNoiseLevel = 50

condenseIfDtLessThan = 10   #  any obs of same target closer together than this
parameter are averaged or one is deleted
                            #   15 secs produces 27 boats
                            #   10 secs produces 12 boats
backgroundDb = 100

clickSourceLevel = 170
clickOutwardSpreading = -10
targetCrossSection = 20

callSourceLevel = 150

# parameters for plotting noise from individual boats

boat_dBlevels = [130, 120, 110, backgroundDb]   # change colors for display at
ranges where these spreading source levels fall
```

(2) Run Python script NFWF_parser.py

This reads input Oceans Initiative csv files (see globalParameters.py)for whale passby and for all boat passbys.
Find each passby and create whale and boat objects containing observations.
Save all this as:

```
save_CVS_format(whalePassbyList, boatsPassbyList)
helpers.save_obj(whalePassbyList,"whalePassbys_2003_2005")
helpers.save_obj(boatsPassbyList,"boatsPassbys_2003_2005")
helpers.save_obj(tracksList,"tracksList_2003_2005")
```

(3) Run NFWF_passbyModel.py to build step by step interpolated paths and write new output files:
Read input from " tracksList_2003_2005"
Plots multiple views of each passby
Summarized whale and boat objects in file:
"tracksModel_2003_2005"

(4) Script Analysis_Rls.py reads in output of previous program, constructs source level for each boat at each location (speed dependent) and calculates received levels at each whale modeled location.

This program outputs whale and boat objects now in file
"tracksModel_Rls_2003_2005"

(5) Script movieFrames.py reads input from previous program and generates frames for movies.

(6) Turn frames into video with ffmpeg: -r 2 will use 2 images per second of video, size is specified as 1152x1152

```
ffmpeg -r 2 -i img%03d.png -c:v libx264 -vf fps=25 -s 1152x1152 -pix_fmt yuv420p
passby_80__2.mp4
```

(8) Script movieFrames_detail.py makse a series of images that are centered on a small region around the focal animal. Size is secified as 288x288

```
ffmpeg -r 2 -i img%03d.png -c:v libx264 -vf fps=25 -s 288x288 -pix_fmt yuv420p
passby_80_detail_2.mp4
```

(9) Liam uses his video editing skills to embed the detailed movie into the larger scale movie.