Triton User Guide

A software package for evaluating acoustic data sets

Version 1.92

Contents

1 Quick Setup 3
1.1 Required Software 3
1.2 Install Triton 3
1.3 Start Triton 4
1.4 Open, Display and Play Example Sound File 5
2 Triton Overview 8
2.1 Summary 8
2.2 Features 8
2.3 Long Term Spectral Average (LTSA) 9
2.4 Remoras 10
2.4 Transfer Function 11
3 Triton Windows 12
3.1 Plot Window 12
3.2 Control Window 14
3.2.1 Band Pass, Axis Limits, and Play Sound pop-up windows 17
3.2.2 File Menu 18
3.2.3 Settings Menu 18
3.2.4 Tools Menu 19
3.2.4.1 Convert Single HARP Raw File 19
3.2.4.2 Load and Apply Transfer Function 20
3.2.4.3 Decimate 21
3.2.4.4 Create LTSA 21
3.2.5 Remoras Menu 23
3.2.5.1 Add, Execute, and Remove Remoras 23
3.2.5.2 Remora Requirements 25
3.2.5.3 Best Practices 25
3.2.5.4 Example Remora: Hello World 26
3.2.5.5 HARP Raw File Remora: HRP 26
3.2.6 Help Menu 27
3.3 Message Window 28
3.3.1 File Menu 29
Appendix A1- HARP Raw Disk Format 30
Non-Compressed, Single Channel 30
Non-Compressed, Four Channel 31
Compressed, Single Channel 32
Appendix A2- XWAV File Format 33
Appendix A3- Software Routines (*.m files) 35

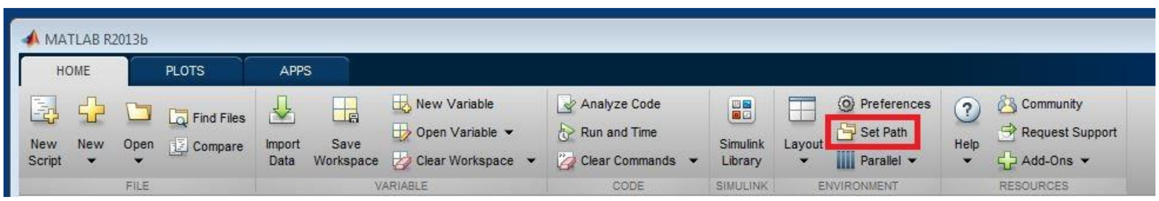
1 Quick Setup

1.1 Required Software

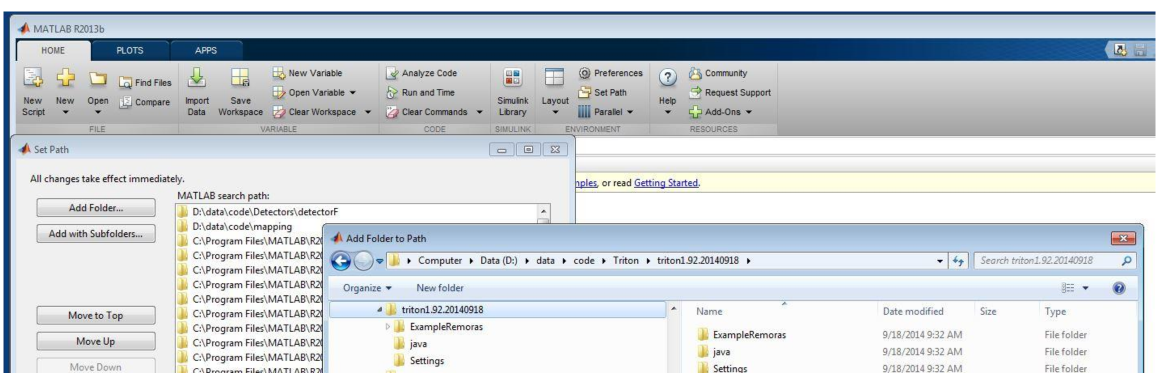
The Triton software package was developed using MATLAB (www.mathworks.com), a high-level technical computing language. Triton is typically executed from within MATLAB running on a Windows operating system. Triton may function on other MATLAB-supported operating systems such as Mac OS and Linux, and can be compiled to run as a stand-alone executable; however, additional testing and development may be needed for these unsupported cases.

1.2 Install Triton

- Download the most recent release of Triton as a compressed file (Triton1.92.20YYMMDD.zip) from the Scripps Whale Acoutic Laboratory's web site: http://cetus.ucsd.edu/technologies_Software.html
- Extract the *.zip file into a folder or directory, such as D:\Triton\
- Start MATLAB and set path to include Triton's folders by using the **Set Path** option in MATLAB's Home tab:

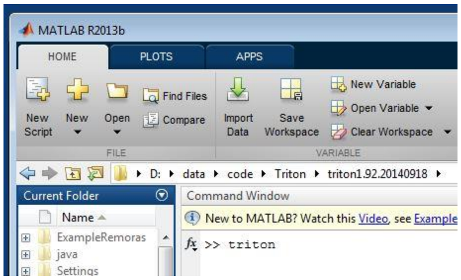


- In the *Set Path* window, remove folders from other versions of Triton with the **Remove** button.
- Click **Add Folder…** button, browse and select the folder containing the new extracted version of Triton.
- Click **Save** and then **Close** buttons at bottom of Set Path Window.

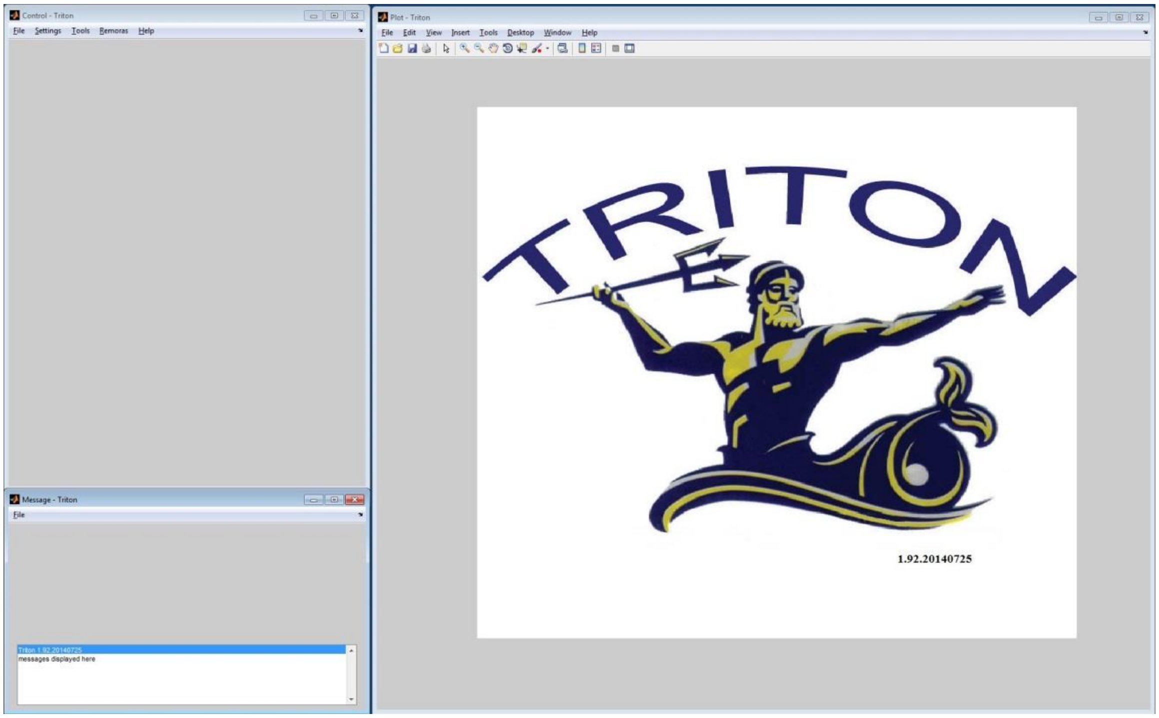


1.3 Start Triton

- At the MATLAB command prompt >> type **triton** to start Triton:



- Three windows will be displayed: *Plot, Control, and Message*:



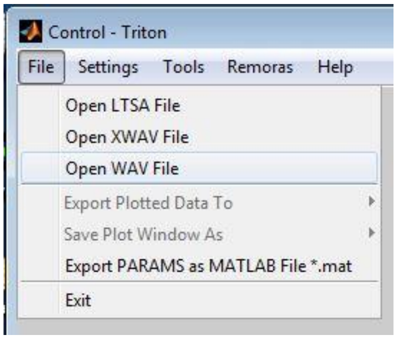
- ***Plot Window*** (Right) **-** Displays plots of acoustic time series, spectra, spectrograms and Long Term Spectral Average (LTSA).

- ***Message Window*** (Lower Left) **–** Displays information and warnings from users' actions such as *Plot Window* cursor location along with selected data values.

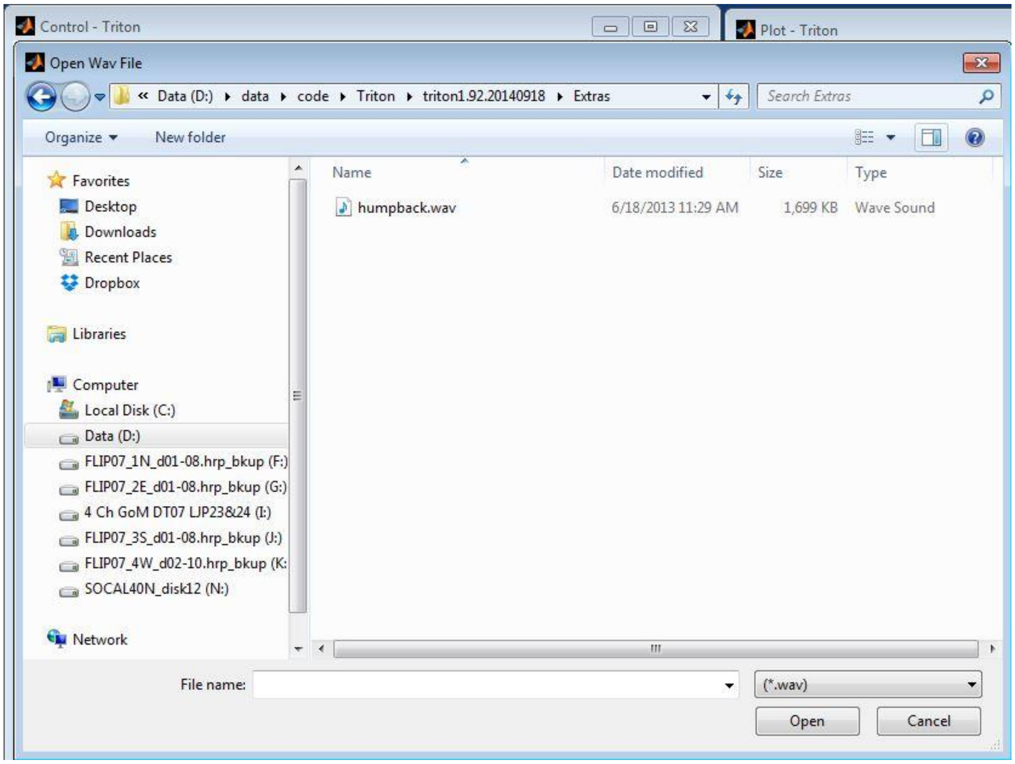
- ***Control Window*** (Upper Left) – Controls plotting parameters for *Plot Window* and various file functions.

1.4 Open, Display and Play Example Sound File

- From the *Control Window*, select **File, Open WAV File**:



- Browse to Triton\Extras\ folder, select *humpback.wav* file, and click **Open**:



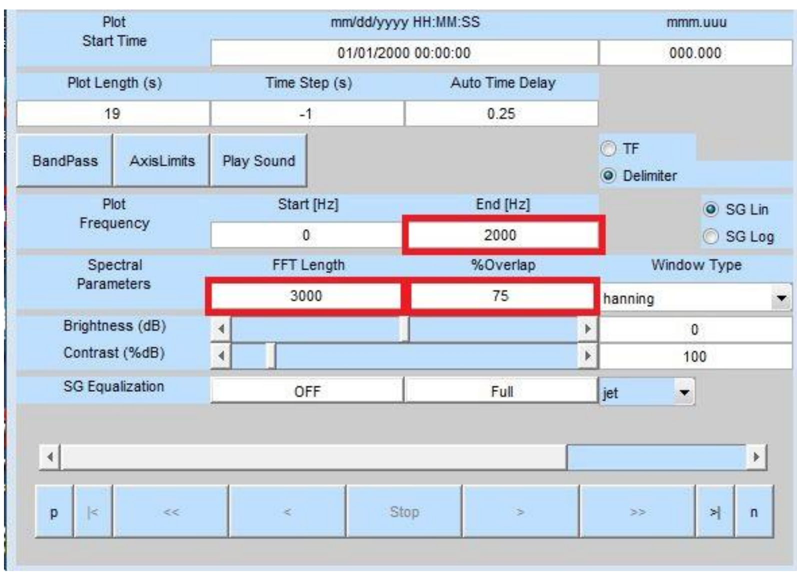
- When the *Set Start Time Window* pops up, select **OK**:

- The first second of a recorded humpback call will be displayed as a time series waveform in the *Plot Window*. Show the full recording by changing the **Plot Length** in the *Control Window* to **20** and the plot will adjust to length of the recording (19s):

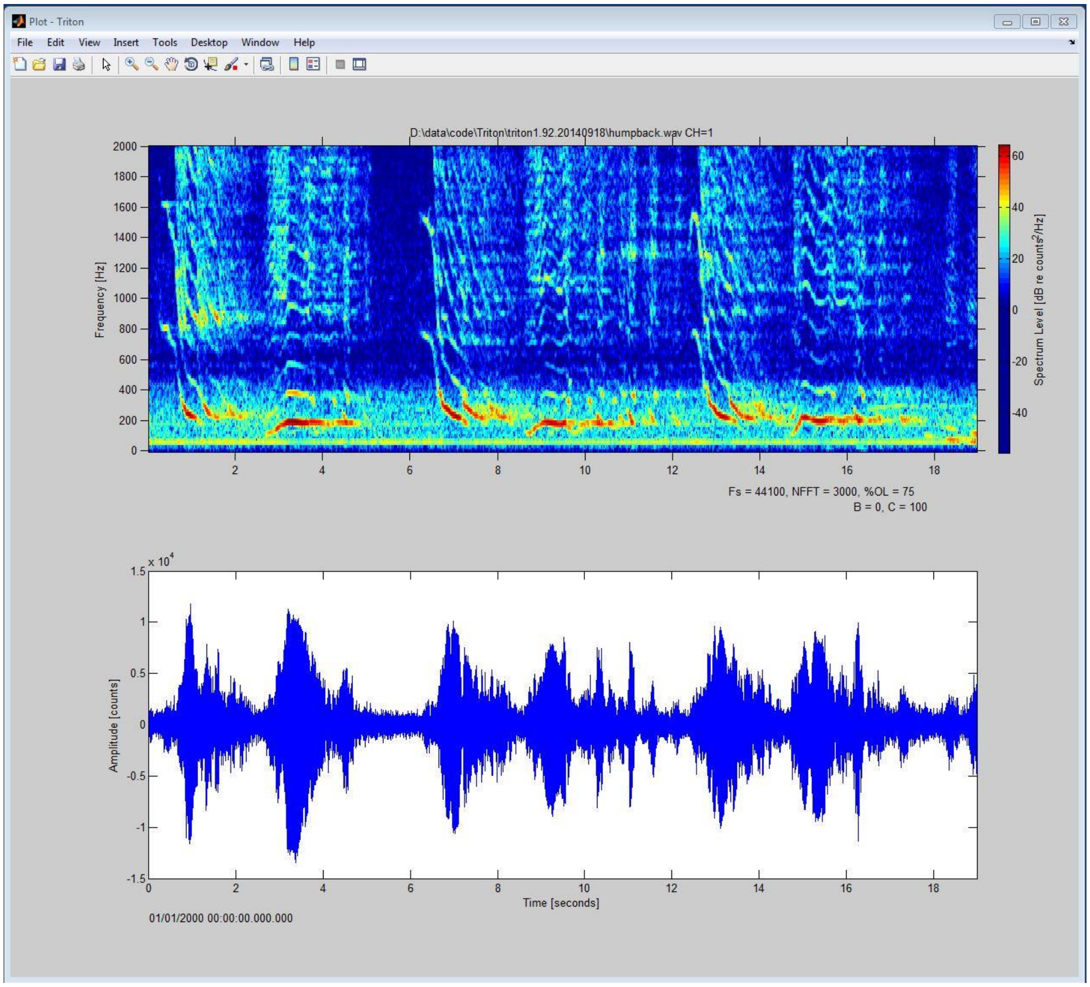
- Select the **Spectrogram** button in the *Control Window* to display the time-frequency representation of the recording in the *Plot Window*:

- Adjust the *Plot Window* parameters by modifying the following *Control Window* values:

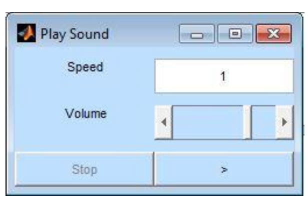
  - ***Plot Frequency End:* 2000**
  - ***FFT Length:* 3000**
  - ***%Overlap:* 75**



- Your *Plot Window* should appear as:



- If your computer has a sound card with speakers or headphones attached, you can play the sound by selecting **Play Sound** button in the middle of the *Control Window* which will open a pop up window:



- Push the play button  to play the sound.

2 Triton Overview

2.1 Summary

Triton has had various development cycles since it was started ~2000, but over the years it has retained its two primary goals. The first is to provide a tool which can be used to evaluate marine acoustic recordings from Acoustic Recording Packages (ARPs)[1], more recently from High-frequency Recording Packages (HARPs) [2], and other devices with WAV-formatted files. These data are typically single or multi- channel, long duration (up to one year or longer), continuous and scheduled duty-cycle sound pressure time series. The time series data are often transformed to the spectral domain for evaluation as power spectra, spectrograms and Long-Term Spectral Averages (LTSA). Triton provides the necessary tools to quickly review a large data set via an easy to use graphical user interface (GUI).

The second goal behind the development of Triton is to allow for additional features and enhancements to be added by users familiar with programming with MATLAB. For example, event detection, classification, logging, and localization algorithms along with other processing tools are currently being developed for use with Triton, utilizing Triton's data management tools, GUI, and most recently, the Remora add-in capabilities.

2.2 Features

- Read raw HARP data and convert into XWAV files. XWAV files are similar to WAV files, but also may contain additional header information such as instrument location, depth, name, and most importantly, time and date (see Appendices A1 & A2 below).
- Display, modify, and advance through plot frames of time series, spectrogram, and spectra from single and multi- channel WAV and XWAV files
- Create and display LTSAs from a collection of WAV or XWAV files providing long-term spectrograms with quick and easy linking to the finer-scale originating WAV or XWAV files by simply clicking on an event of interest in the LTSA plot frame (see LTSA section below)
- Save displayed data as mat-file, WAV, XWAV, JPEG, or other graphical file types
- Decimate high sample rate data files and save as lower sample rate XWAVs to allow for easier analysis of low frequency sounds, for example, baleen whale call analysis from ultra-sonic data.
- Easily add user-developed features with the Remora tool

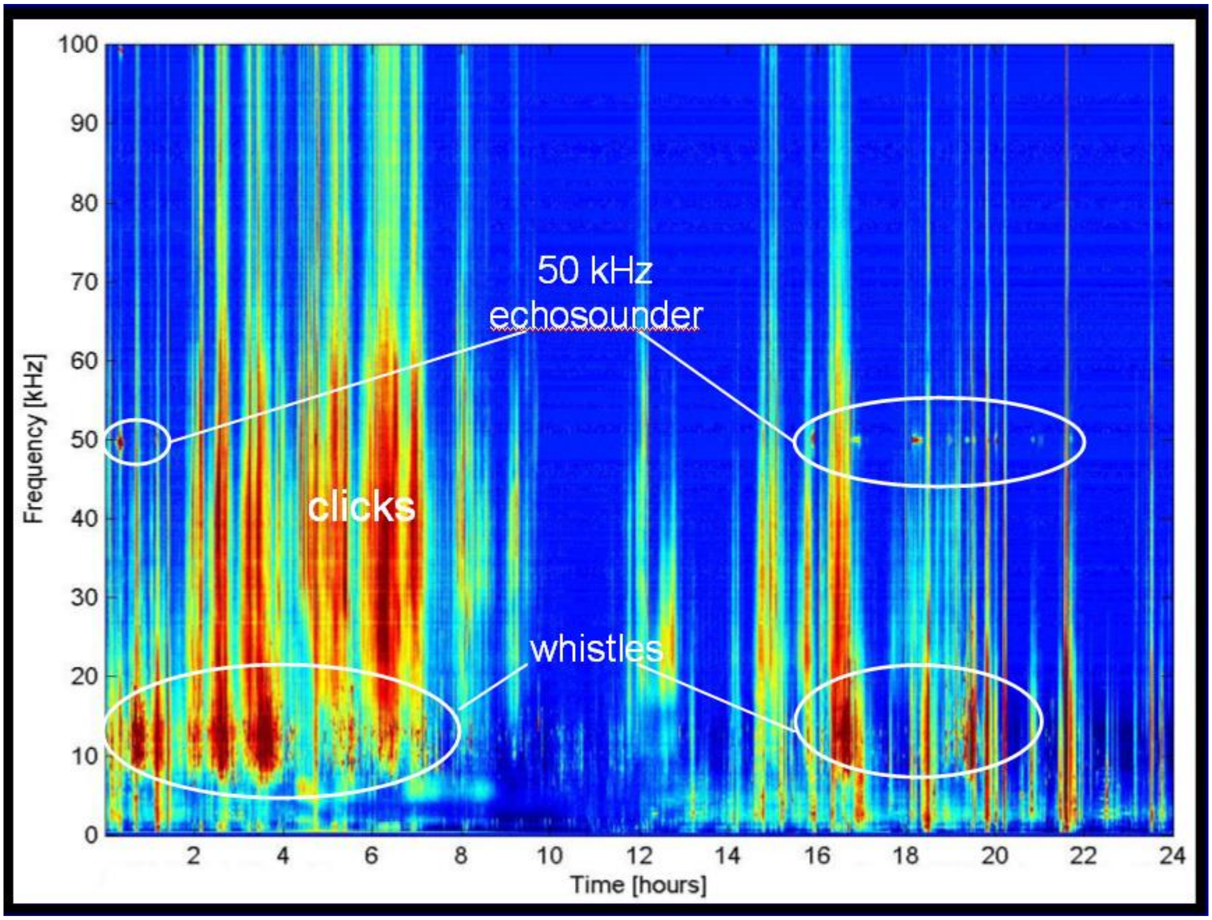
[1] Wiggins, S. M., Autonomous acoustic recording packages (ARP's) for long-term monitoring of whale sounds, Marine Technology Society Journal, vol. 37(2), pp. 13-22, 2003.

[2] Wiggins, S. M. and Hildebrand, J. A., High-frequency Acoustic Recording Package (HARP) for broad-band, long-term marine mammal monitoring. International Symposium on Underwater Technology 2007 and International Workshop on Scientific Use of Submarine Cables & Related Technologies 2007, Tokyo, Japan, Institute of Electrical and Electronics Engineers, pp. 551-557, 17-20 April, 2007.

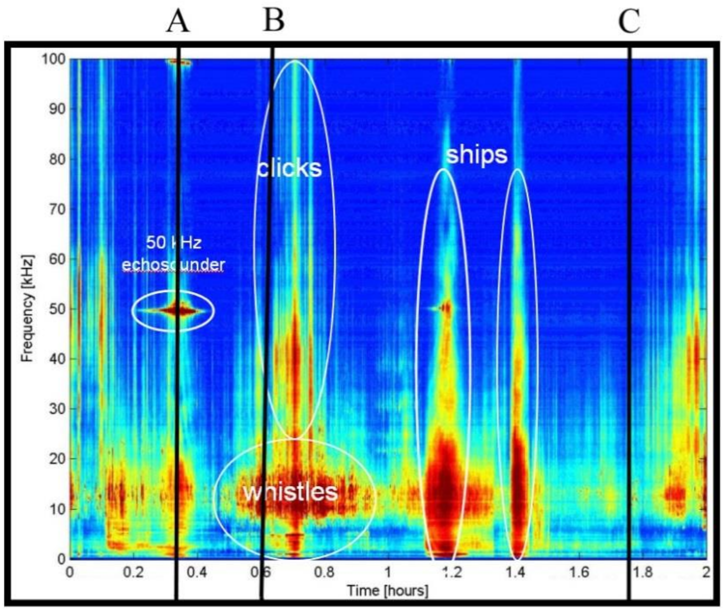
2.3 Long Term Spectral Average (LTSA)

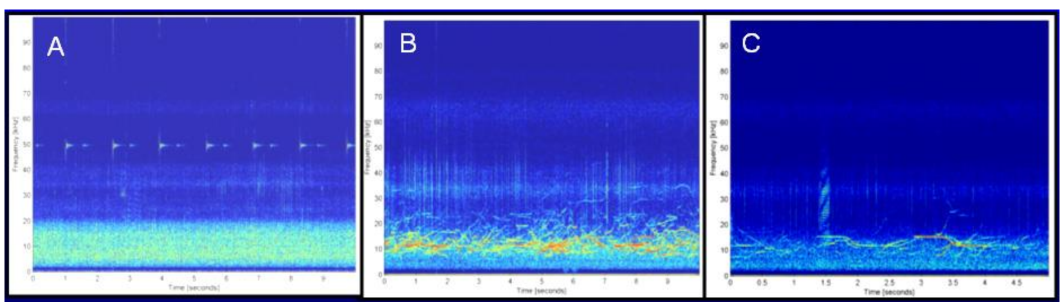
LTSA (Long-Term Spectral Average) plots allow large time-series data sets to be viewed and analyzed, including searching for, noting, and evaluating acoustically significant events, such as dolphin and whale calls or anthropogenic sources. In 2010, each HARP (High-frequency Acoustic Recording Package) deployment was capable of producing over 11,000 XWAV files (~10 TB total for 0.9 GB XWAV files), but viewing and analyzing each one of these XWAV files is not practical. LTSA plots provide a means of presenting an overview of these large data sets in a compressed format and allow quick linking to noteworthy events in the finer time scale XWAV data, which were originally used to generate the LTSA plots.

An LTSA of time-series data is essentially a spectrogram (three dimensional time-frequency-energy plot) where each power spectra plotted along time is averaged over a longer period than for a typical spectrogram (i.e., great than one windowed frame of a Fast-Fourier Transform (FFT)). The quantization time of an LTSA is defined as the duration over which consecutive single-window spectra are averaged. The averaged-spectra are then plotted sequentially with energy shown as color.



Above is an example of a 24 hour LTSA from 35 GB of 16-bit sample data sampled at 200 kHz off the shore of Southern California. Notice how well ship echosounders and dolphin clicks and whistles can be identified, and periods or frequencies that are relatively quiet.





The LTSA plot above is from the first two hours of the previous figure, and the three plots A, B, and C below it are spectrograms from the XWAVs used to generate the LTSA at the corresponding times A, B, and C in the LTSA plot. The three spectrograms were displayed by using the **Expand** function and selecting the events of interest in the LTSA plot. The 50 kHz tonal from a ship's echo sounder in the LTSA appears at pulses in the 10s spectrogram (A), broadband stripes in the LTSA are dolphin echolocation clicks (B), and dolphin whistles appear as narrow-band pulses in the LTSA (B & C). See sections below on how to generate and use LTSAs.

2.4 Remoras

Remoras are user-developed MATLAB software routines that can be added in to Triton. Remoras provide a means to allow users to customize Triton to fit their own specific needs without modifying Triton's core routines. Two sample Remoras are included with Triton in the \ExampleRemoras\ folder: HRP and the HelloWorld. The HRP Remora allows users to work with data and headers of raw HARP files (*.hrp), including header values and raw file timing directory list display and time check, and converting raw files to XWAVs. The HelloWorld Remora plots a spectrogram of the *humpback.wav* example file and in its coding, provides details on how to use the mouse click button and how to set up user-defined hot-keys.

2.4 Transfer Function

Transfer functions (TF) are used to convert data from one unit or metric to another potentially more meaningful unit. In the case of a HARP system, the recorded data are in units of the analog-to-digital converter (ADC i.e., [counts]). These counts are related through a frequency dependent TF to pressure [µPa] at the hydrophone. Two main conversions occur within this system. The first is at the hydrophone where time-varying pressure is converted to time varying voltage, amplified and filtered providing a frequency dependent response to pressure. The second conversion is in the data logger where the hydrophone preamplifier analog voltage signal is digitized and stored on digital media such as hard disk drives. The details of these conversions are often applied in reverse to correct the recorded data back into meaningful physical units.

Since XWAV data files units are in [counts] representing the physical units of pressure [µPa], the TF is described as the inverse sensitivity or 1/sensitivity [dB re µPa/counts] and is frequency dependent. Spectra and TFs levels are often described in logarithmic base-10 decibel [dB] units because of a large dynamic range of the sound signals. Currently in Triton, the transfer function is only applied to spectra plots, not LTSA, spectrogram nor time series plots. The TF is applied to spectra plots via a user generated TF file consisting of two columns: one with Frequency [Hz] and the other with 1/sensitivity [dB re µPa/counts]. The TF file is an ascii text file with extension *.tf.

Transfer Function Calculation:

The inverse sensitivity for a TF file is calculated in via the negative of the sum, in dB, of the following three TF components:

1) Sensor sensitivity (Ceramic/PZT/hydrophone – can be frequency dependent or flat response) [dB re V/µPa] Open Circuit Received Response

2) Preamplifier + Filter Board Gain (usually frequency dependent i.e., gain varies with frequency in TF file) [dB]

3) Analog-to-digital converter (ADC – usually flat response)   [dB re counts/V]

Sensitivity = [dB re counts/µPa] = [dB re V/ µPa] + [dB] + [dB re counts/V]

Example HARP TF calculation:

ITC-1042: ~ -202 dB re V/ µPa
600 series preamplifier: + 50 dB gain
ADC (16-bit, 0-5.0V range): $20*\log_{10}(2^{16}\text{counts}/5.0\text{V})$ = +82 dB re counts/V
Inverse sensitivity = - (-202 +50 +82) = +70 dB re µPa/counts

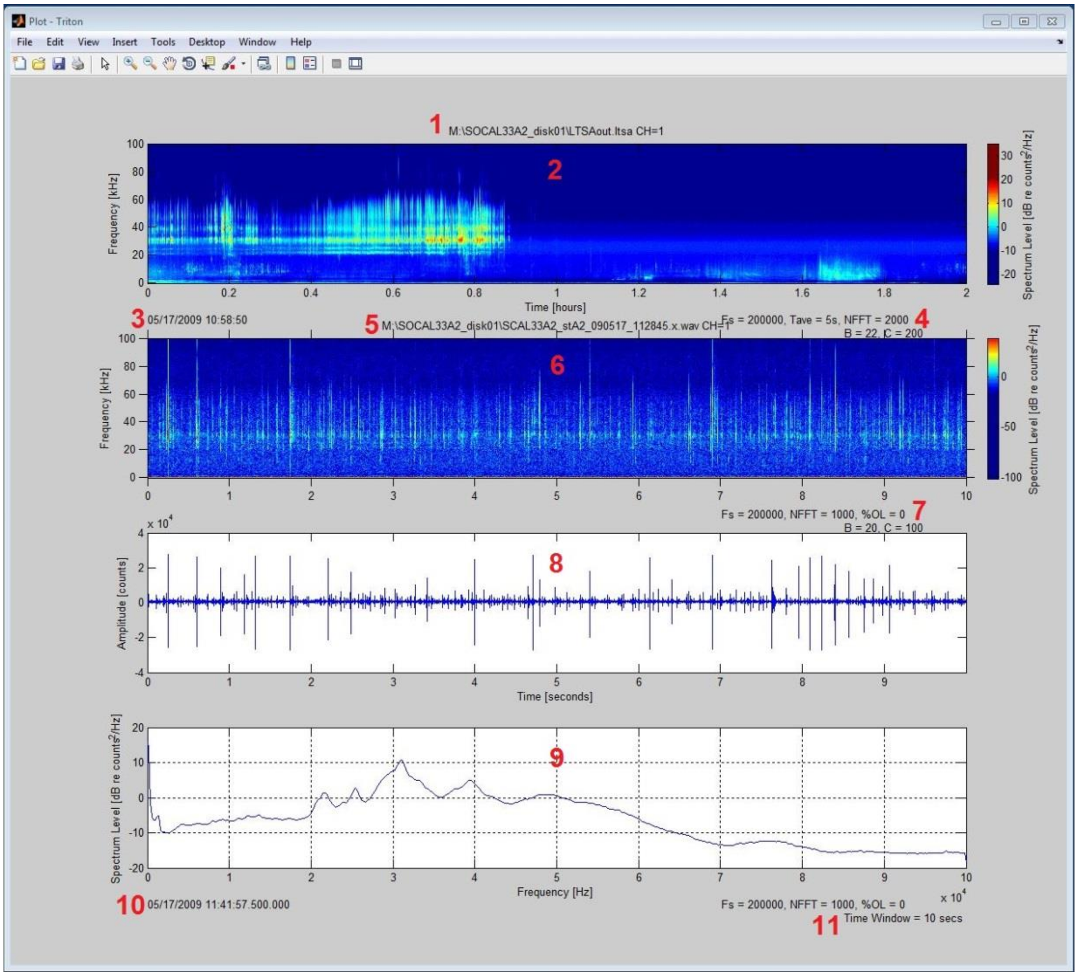
Important: The inverse sensitivity is the *negative* of the sum, in dB, of the sensor sensitivity, preamp gain and ADC voltage quantization.

For more detailed description of TF calculations see Triton subroutine loadTF.m.

3 Triton Windows

3.1 Plot Window

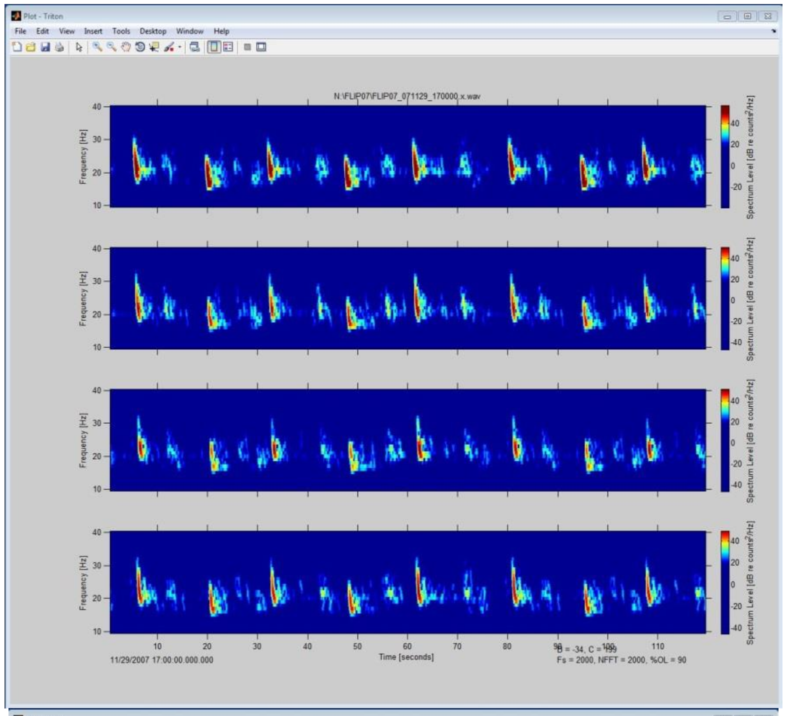
The Triton *Plot Window* is a standard MATLAB figure window, allowing it to be modified, saved or exported in various formats, and printed. All pull-down menus and tools are the same as typical MATLAB figure windows except the zoom in/out tools which will modify the **Plot Start Time** after zooming in and deselecting the tool.

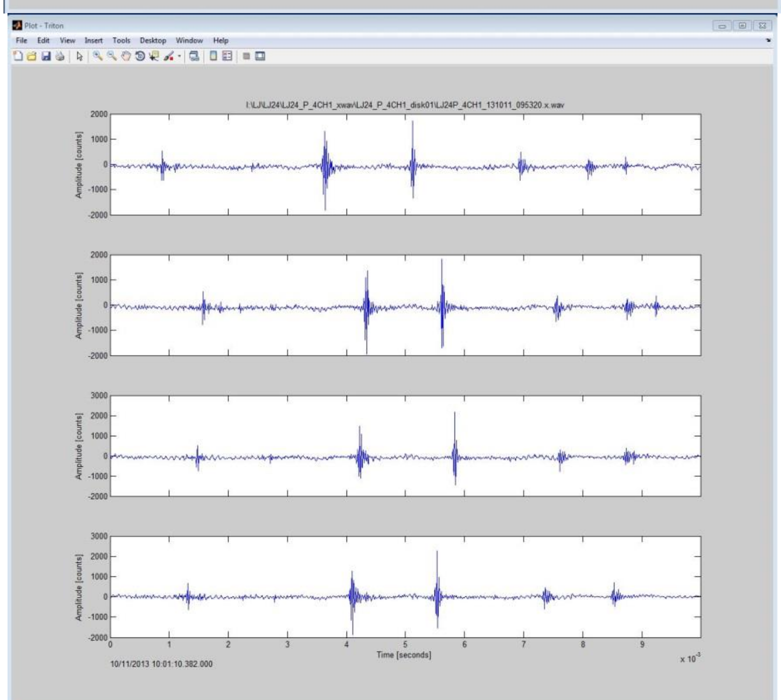


Components of the *Plot Window*:

1) LTSA file name
2) LTSA plot
3) LTSA **Plot Start Time**
4) LTSA parameters
5) XWAV/WAV file name
6) Spectrogram plot
7) Spectrogram parameters
8) Time series plot
9) Spectra plot
10) Spectrogram/Time Series/Spectra **Plot Start Time**
11) Spectra parameters

Triton can also display multichannel data as spectrograms, time series or spectra. For example, the two *Plot Windows* below show spectrogram of fin whale calls from a km-scale array on the top and time series of dolphin clicks from meter-scale array on the bottom.

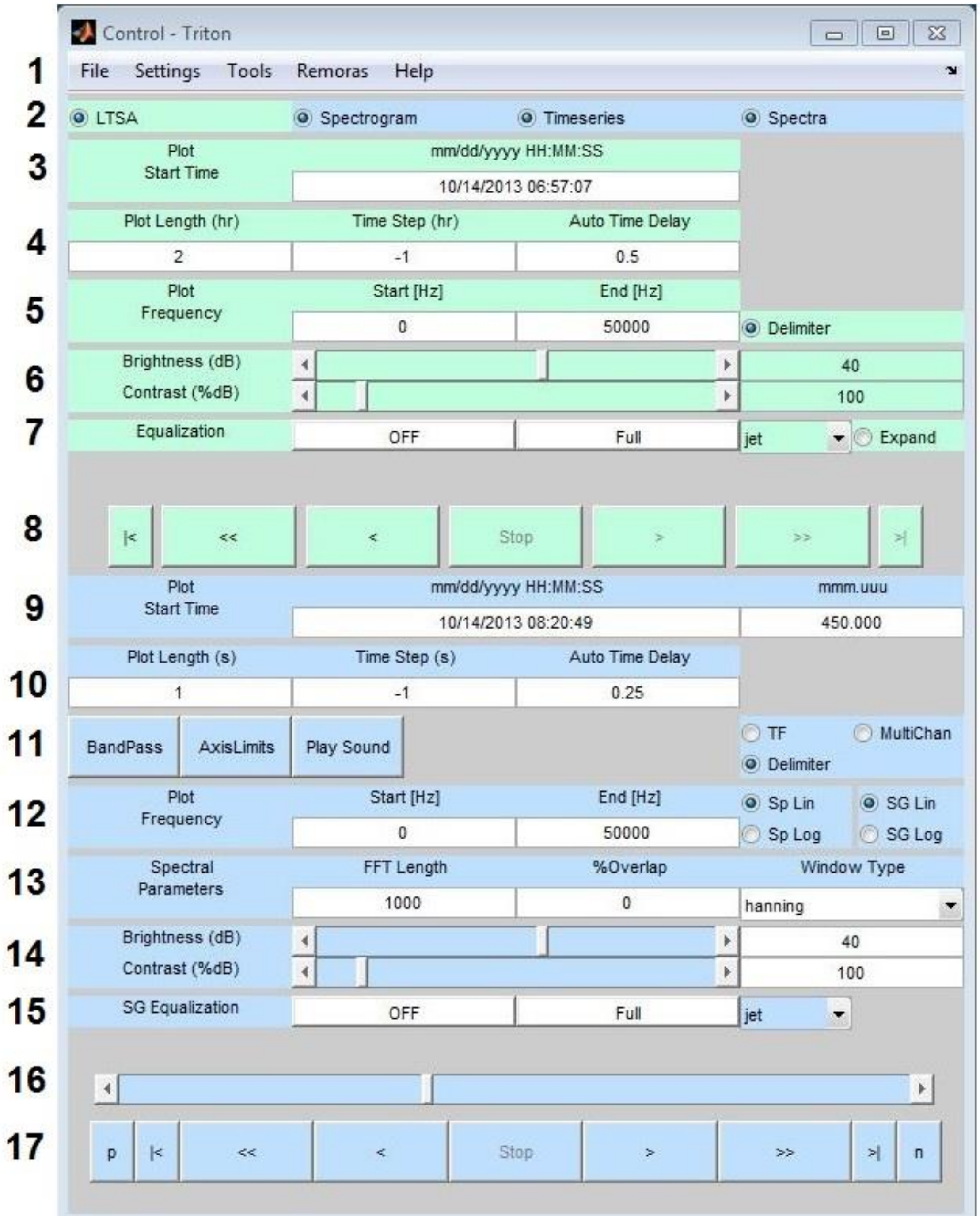




This feature has only been tested with 4 channel data, but multichannel functionality should work for any number of channels up to some practical display limit, for example, ~ 8 channels.

3.2 Control Window

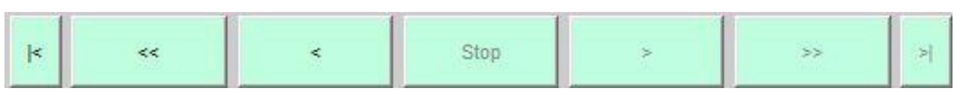
The *Control Window* allows users to open, convert, and plot data files in addition to controlling various plotting parameters. The *Control Window* has three main sections: pull-down menus at the top, light green LTSA plot control in the upper half, and the light blue XWAV/WAV plot control in the lower half.



1) **File**, **Settings, Tools**, **Remoras,** and **Help** pull-down menus – see next section for details.
2) Radio buttons to toggle on/off the different plot types: **LTSA**, **Spectrogram**, **Timeseries**, and **Spectra**. Note the *Control Window* color scheme: LTSA plot control is light green, and the XWAV/WAV plot controls are light blue.

LTSA Plot Control – Light Green

3) **Plot Start Time** – Display and change LTSA plot start time [mm/dd/yyyy HH:MM:SS]
4) **Plot Length** – LTSA plot frame display length [hr]
   Time Step – LTSA advancement size between start of successive plot frames [hr].
   Set to **-1** for sequential plot advancement of one **Plot Length** frame of data.
   Set to **-2** for *next* **Plot Start Time** = *current* **Plot Start Time** + **Plot Length**.
   Settings **-1** and **-2** are different only for non-continuous data.
   Auto Time Delay – Time delay in seconds between displaying plot frames during auto forward and auto rewind functions
5) **Plot Frequency** – LTSA plot **Start** and **End** frequencies [Hz], and delimiter button for the LTSA plot (shows breaks in non-continuous data).
6) **Brightness** and **Contrast** adjustments for LTSA plot. Adjustments can be made with the sliders, arrows, or by entering values.
7) **Equalization** – **ON/OFF** button to subtract average LTSA levels from plotted LTSA. Average LTSA is from either **Full** plotted frame or shorter user **Pick**ed window.
   Dropdown list to select LTSA plot color map.
   Expand button activates cross hairs to pick an event time in the LTSA plot and open the originating XWAV/WAV file at the corresponding picked time (see LTSA selection below).
8) LTSA plot frame motion step buttons:



Button	Description	Keyboard Hot Key
|<	Go to start of LTSA file	
<<	Auto Rewind	Ctrl down arrow
<	Rewind one step	Ctrl left arrow
Stop	Stop Auto Rewind or Forward	Ctrl space bar
>	Forward one step	Ctrl right arrow
>>	Auto Forward	Ctrl up arrow
>|	Go to end of LTSA file	

XWAV/WAV Plot Control – Light Blue

9) **Plot Start Time -** Display and change XWAV/WAV plot start time. [mm/dd/yyyy HH:MM:SS]
10) **Plot Length** – XWAV/WAV plot display length [s]
    Time Step – XWAV/WAV plot step size between the start of plot frames [s].

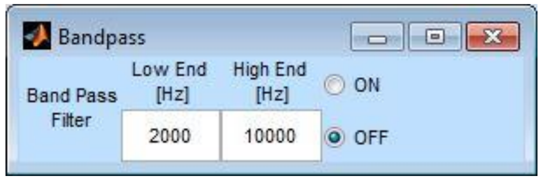
Set to -1 for steps equal to one Plot Length frame of data.
Auto Time Delay – Time delay in seconds between plot frames during auto forward and auto rewind (see below)

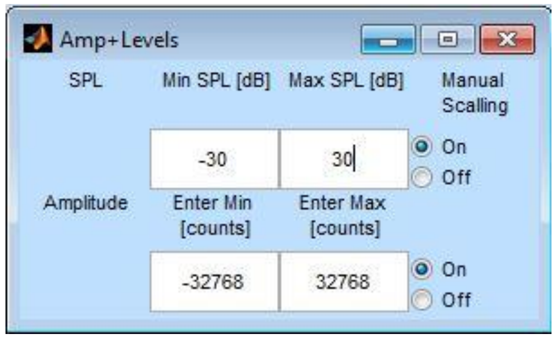
11) **Band Pass Filter** – button brings up bandpass filter window (see below)
Axis Limits – button brings up timeseries amplitude and spectra level amplitude limits (see below).
Play Sound – button brings up play sound window (see below, and *Quick Start*).
TF - radio button to turn on/off transfer function correction for *Spectra* plot only (See transfer function section). Need to load transfer function from *Control Window* **Tool** pull-down menu (see next section).
Delimiter – on/off radio button for *Timeseries* and *Spectrogram* plot vertical line delimiter between 'raw files'
MultiChan – if multiple channel XWAV/WAV file is loaded, radio button to switch between single channel and multichannel mode which affects how data are presented in the *Plot Window*.
12) **Plot Frequency** – XWAV/WAV plot **Start** and **End** frequencies [Hz].
Sp Lin/Log – radio buttons to toggle between linear and logarithmic frequency axes for *Spectra* plots
SG Lin/Log – radio buttons to toggle between linear and logarithmic frequency axes for *Spectrogram* plots
13) **FFT length** – Length of Fast Fourier Transform (FFT) window in samples
%Overlap – Percent of Window overlap
Window Type – Hanning is currently the only available window type.
14) **Brightness** and **Contrast** adjustments for *Spectrogram* plot. Adjustments can be made with the sliders, arrows, or by entering values.
15) **SG Equalization** – Toggle **On/Off** removal of average *Spectrogram* levels. Toggle between **Full** plot frame average or shorter user **Pick**ed window to be subtracted from *Spectrogram* levels.
Dropdown list to select *Spectrogram* color map.
16) Time slider – Display and adjust the relative position of the **Plot Start Time** of the loaded XWAV/WAV file. Adjust start time by selecting arrows or moving slider.
17) XWAV/WAV plot frame motion step buttons:

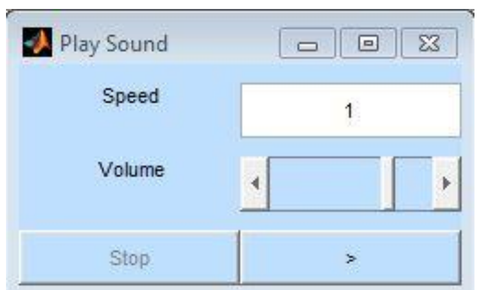
Button	Description	Keyboard Hot Key
p	Load previous file in directory	
|<	Go to beginning of loaded file	
<<	Auto rewind	down arrow
<	Rewind one step	left arrow
Stop	Stop Auto Rewind or Forward	space bar
>	Forward one step	right arrow
>>	Auto forward	up arrow
>|	Go to end of loaded file	
n	Load next file in directory	

3.2.1 Band Pass, Axis Limits, and Play Sound pop-up windows

Band Pass, Axis Limits and **Play Sound** pop-up windows are accessed via the buttons on line 11 of the XWAV/WAV section above in the *Control Window*:







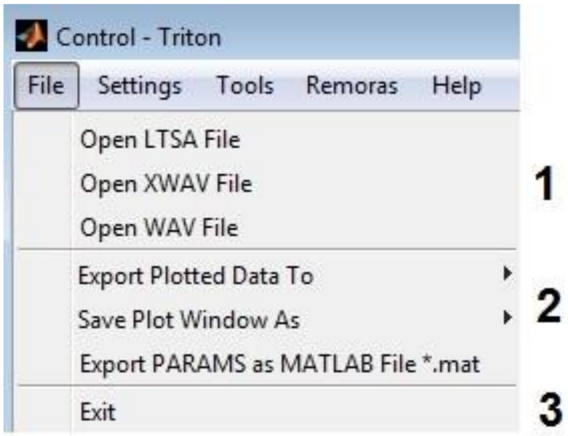
Bandpass: Filter of loaded data is toggled **ON/OFF** with radio buttons and **Low** and **High** corner frequencies for filter are set via the editable fill-in text boxes.

Amp+Levels: *Timeseries* plot amplitude and *Spectra* plot level limits are set via editable fill-in text boxes. MATLAB autoscaling is used when **Manual Scaling** buttons are toggled **Off**.

Play sound: Displayed data sound is played through computer sound card and speakers (see *Quick Start* section above).

> **Speed** factor of sound playback: 0.1 to 10 times loaded file's sample rate. Speed up low frequency sounds and slow down high frequency sounds.
>
> **Volume** of sound playback, also adjust through computer operating system.
>
>  play sound through computer sound system.
>
> **Stop** sound play back.
>
> Hint: Use the bandpass filter to remove dominate low frequency sounds (noise) before playing higher frequency sounds.

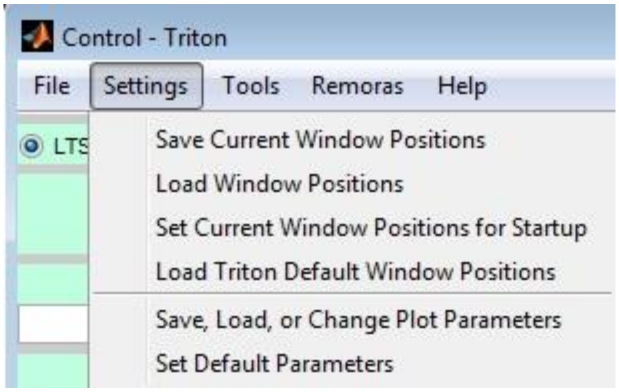
3.2.2 File Menu



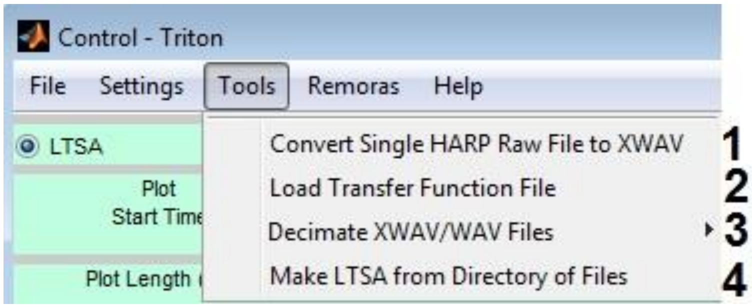
1) **Open LTSA**, **XWAV** or **WAV** files.
2) **Export** plotted data as **WAV**, **XWAV**, **or MAT-** files
   Save plotted data as **JPEG** or MATLAB **FIG** files
   Export Triton program parameters **PARAMS** to **MAT** file
3) **Exit-** Exit Triton

3.2.3 Settings Menu

When Triton is started the three windows, *Plot, Control, and Message,* are placed to fill a single monitor's full screen. These windows can be resized and repositioned as is typical for MATLAB windows. If a different window configuration than the default is desired, for example, moving the *Plot Window* to a second monitor at full screen, the new configuration can be saved, loaded or set for the next Triton start up with the **Settings** pull-down menu. Likewise, plotting parameters such as length, spectral parameters, brightness and contrast, etc., can be saved, loaded or set for startup within **Settings**.



3.2.4 Tools Menu

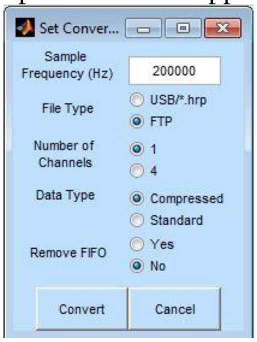


1) **Convert Single HARP FTP/USB File to XWAV** (used for lab and deck tests)
2) **Load** Transfer Function **TF** File
3) **Decimate** XWAV or WAV Files
4) **Make LTSA from Directory of Files**

3.2.4.1 Convert Single HARP Raw File

As part of standard HARP instrument lab or deck tests, single raw HARP files are converted, loaded, and plotted to evaluate system performance:

- Select **Convert Single HARP Raw File to XWAV** from the **Tools** pull-down menu, the **Set Conversion Parameters** pop-up window will appear:

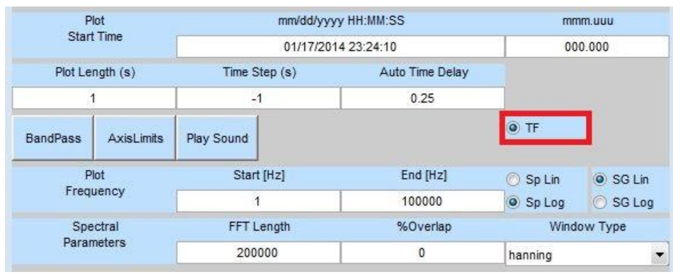


- Since HARP raw files do not have embedded conversion parameters, set conversion parameters as needed:
  Sample Frequency (Hz)
  File Type (whether FTP'ed via HARP or dd'ed via PC)
  Number of Channels
  Data Type
  Remove FIFO noise
- Select **Convert** button and an **Open Raw HARP File to convert to XWAV format** window will appear, choose file, select **Open** button.
- **Save XWAV File** window will appear, choose folder to save converted file, change save file name if needed, select **Save** button.
- The converted data file will be loaded into Triton and appear as a 10 second time series in the *Plot Window*.

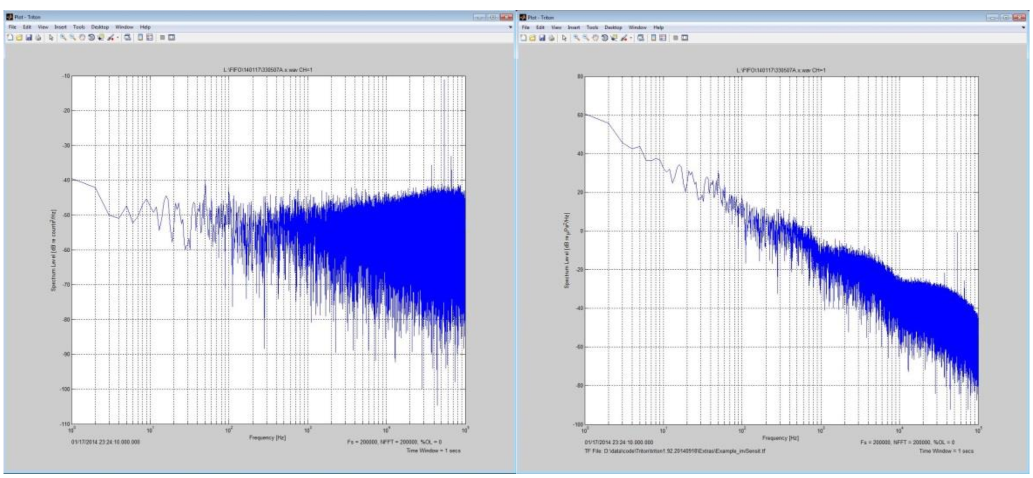
3.2.4.2 Load and Apply Transfer Function

For details about Transfer Functions (TFs), see Transfer Function section in *Triton Overview* above. TFs are only applied to *Spectra* and *Spectrogram* plots in Triton, not *Timeseries* nor LTSAs; however, Remoras could be developed to do so if needed.

- From *Control Window* **File** pull-down menu: **Open XWAV/WAV File** to load a data file.
- From *Control Window* **Tools** pull-down menu: **Load Transfer Function File**
  Browse and select the transfer function file (*.tf) to load. (An example *.tf file is in the Triton\Extras\ folder)
  Click **Open** button, the transfer function is loaded, but not yet applied to the *Spectra*
- From *Control Window*, select **Spectra** radio button.
- In the *Control Window* set spectral parameters as needed for *Spectra* plot and select the **TF** radio button:



- A comparison between uncorrected (left) and corrected (Example TF applied, right) spectra:

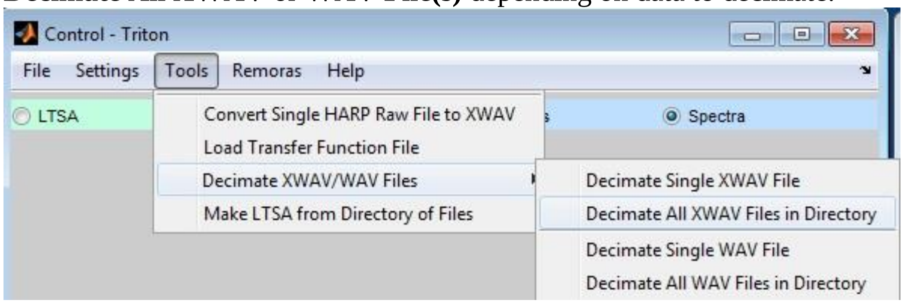


- Example_invSensit.tf, used in this example, is a two column file with the first column frequency (Hz) in steps increasing by a factor of 10 (i.e., 1, 10, 100, … 100000) and the second column is inverse sensitivity in dB re µPa/counts decreasing each step by 20 (i.e., 100, 80, 60 … 0) which is added to the uncorrected spectra.

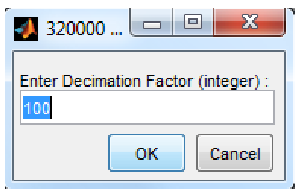
3.2.4.3 Decimate

Decimation can be useful in reducing XWAV or WAV file size and effective sample rate for quicker processing and analysis of low frequency sounds. For example, recordings with 200 kHz sample rate can be reduced by a factor of 100 to 2 kHz sample rate for evaluating baleen whale calls.

- Select **Decimate** from the **Tools** pull-down menu and choose to **Decimate Single** or **Decimate All XWAV** or **WAV File(s)** depending on data to decimate.



- Select **Open** or **Select Folder** button**.**
- A decimation factor pop-up window will appear, enter decimation factor and select **OK** button.



- A window will pop-up to choose where to save single or all files. Change name and folder as needed.

3.2.4.4 Create LTSA

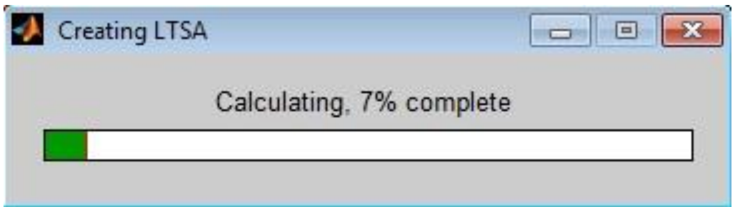
Long-Term Spectral Averages (LTSAs) are a powerful approach for evaluating long duration acoustic data sets (see *Triton Overview/LTSAs* section above). In addition to scrolling through large data sets quickly, LTSAs provide a graphical index and link into the fine-scale data used to make LTSAs so that events of interest can be investigated in fine detail.

- Select **Make LTSA from Directory of Files** from **Tools** pull-down menu in *Control Window*.
- Select file type, Type '1' for WAV files or '2' for XWAV files, select **OK**.
- Select folder of WAV or XWAV files to be used to make LTSA.
- Set Long-Term Spectrogram Parameters
    - **Time Average** – length of time for each spectral average (i.e., time bin size or Tave).
    - **Frequency Bin Size** for LTSA [Hz]

LTSA parameters are typically data sample rate dependent. The default parameters (5s averages and 200 Hz frequency bins) are for broad-band HARP data sampled at 200 kHz. For lower sample rate such as ARP data (eg. 500 Hz or 1000 Hz), longer time averages and smaller frequency bins may be useful to keep LTSA files small (eg., 120s and 1 Hz).

- Select folder to save LTSA – *typically* choose the same folder as the XWAVs or WAV files so that the LTSA can provide a link and index to these fine scale data files.

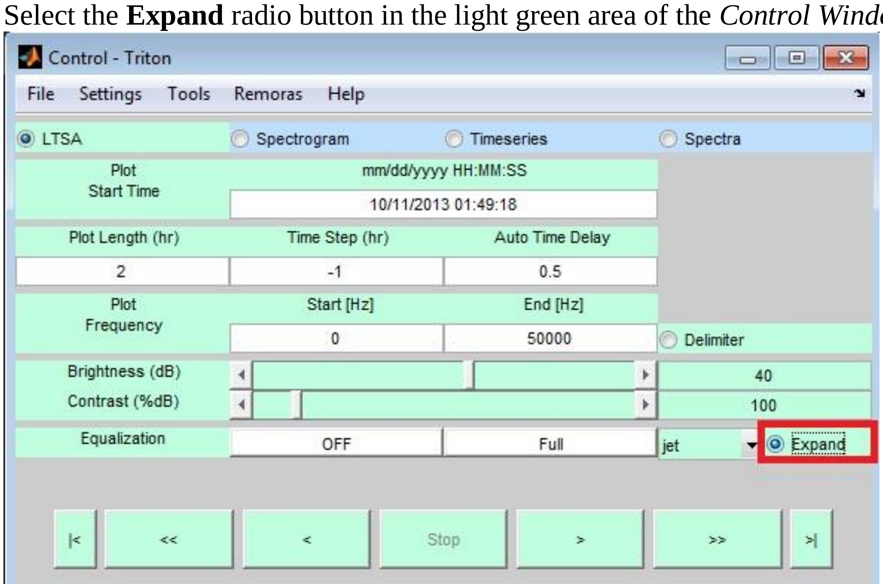
The amount of time required to create the LTSA depends upon the LTSA parameters, the type and quantity of data to be processed, and the speed of the computer. Typical processing time can be over one hour. After choosing the LTSA output folder, generation of the LTSA is started and a pop-up dialog box showing the progress of the processing is displayed:



After the LTSA has been created, it is displayed in the *Plot Window.*

To investigate a sound of interest from the LTSA *Plot Window*:

- Select the **Expand** radio button in the light green area of the *Control Window.*



- Place the mouse cross-hairs cursor on the LTSA plot at the time of interest, left click the mouse button to select, XWAV or WAV file *Spectrogram* plot will be displayed along with XWAV/WAV *Control Window* parameters (light blue).

- Select **Timeseries** and/or **Spectra** radio buttons to display these plot types.

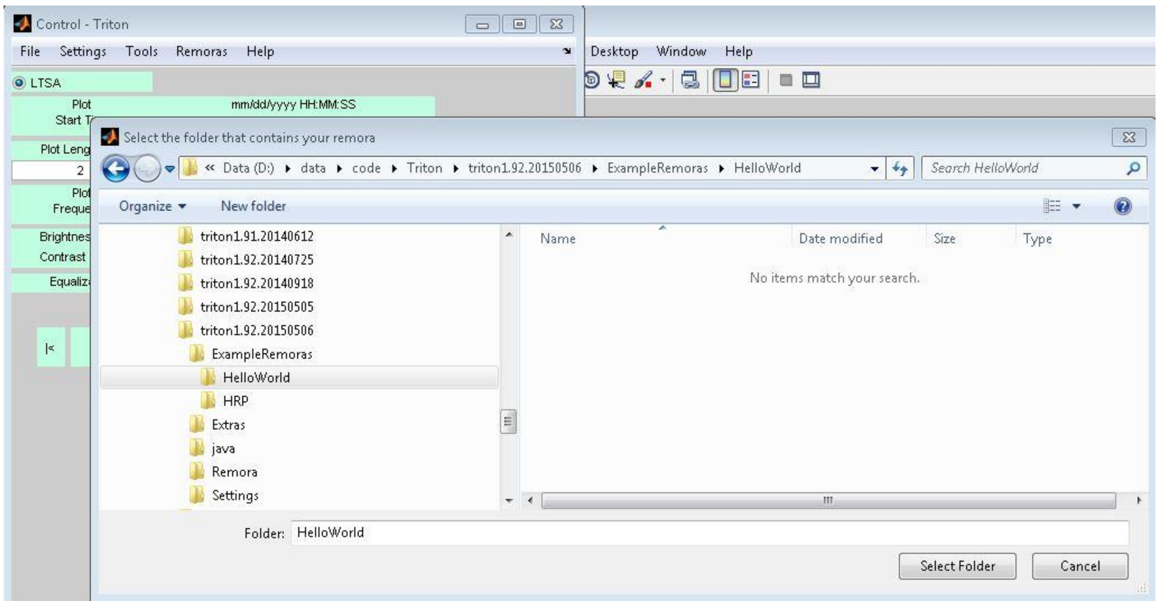
- XWAV/WAV plots can be modified as usual, or updated when another event is chosen via the **Expand** button.

3.2.5 Remoras Menu

Remoras are user-developed MATLAB m-files that can be added to Triton to increase its functionality without modifying the core functions of Triton.

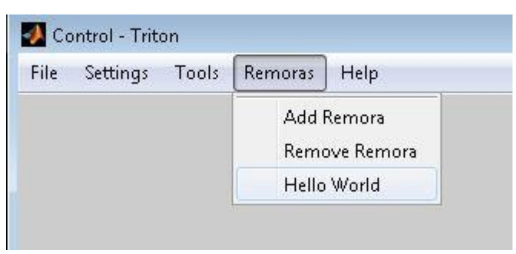
3.2.5.1 Add, Execute, and Remove Remoras

- In the *Control* Window, **Remoras** pull-down menu, select **Add Remora**.
- Choose the folder that has the files of your Remora:

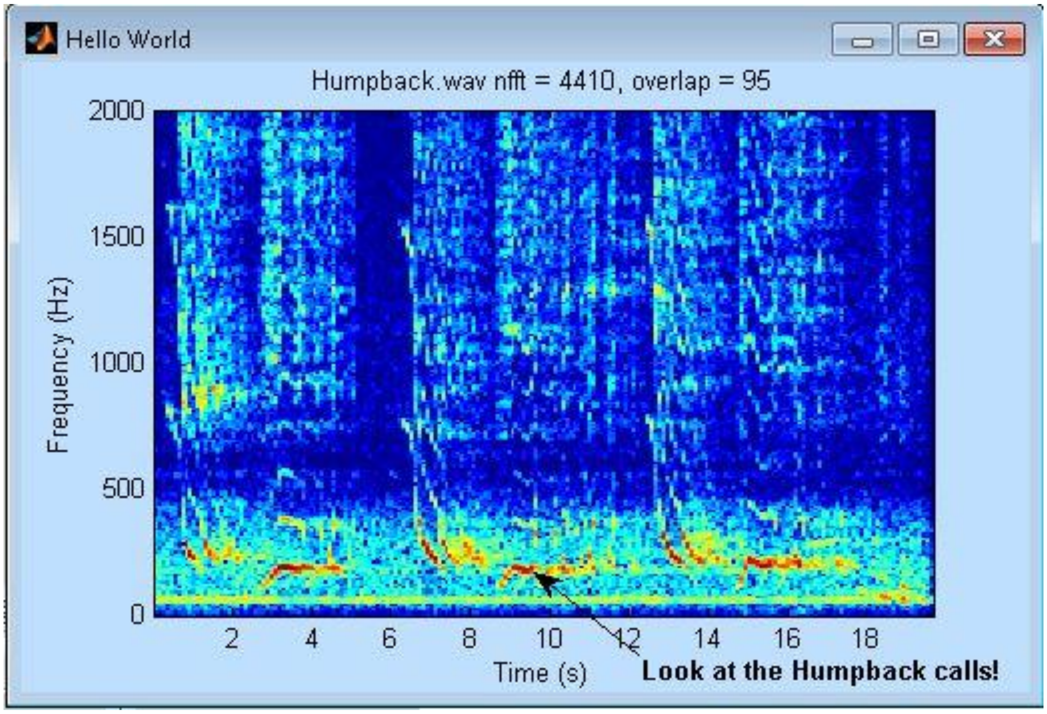


- Select **Yes** from pop-up window to restart Triton:

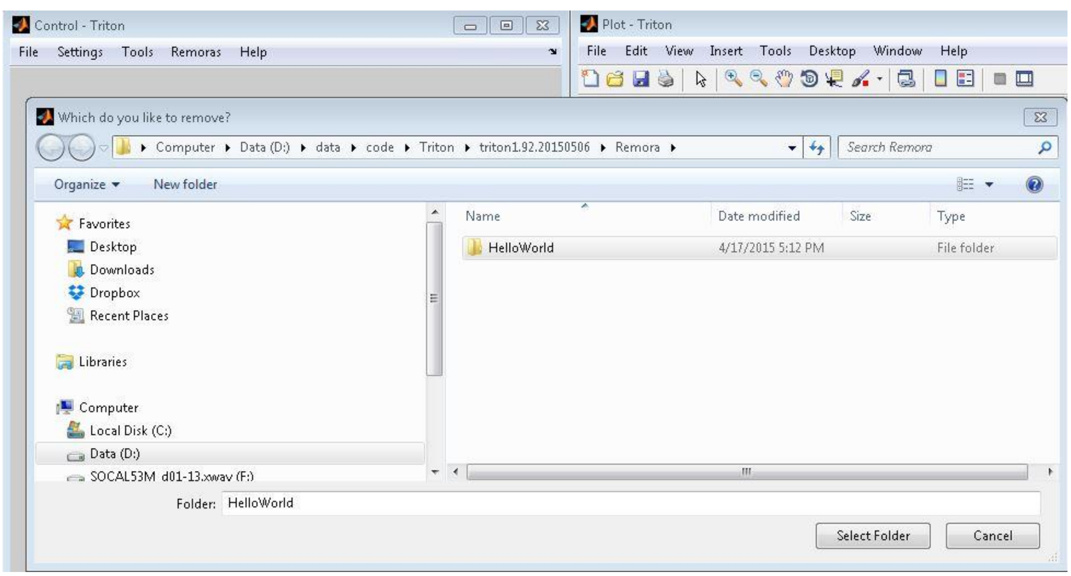
- Execute the added Remora via the **Remora** pull-down menu:



- Hello World Remora figure window:



- To remove a Remora, *Control Window*, **Remoras** pull-down menu, **Remove Remora**, select Remora folder to remove:



- From the next pop-up window, select **Remove** to remove it from the **Remoras** pull-down menu and Remora folder. Select **Backup** to also remove it from the same pull-down menu and folder, but additionally save the Remora to another folder for future work or backup:

- Select **Yes** from pop-up window to restart Triton:

3.2.5.2 Remora Requirements

To function as a Remora, there are three minimum requirements for the Remora code set:

- Remora folder - Upon adding a Remora, Triton will copy your Remora folder to the *Remora*\ folder and add it to MATLABs path. The name should be unique and the contents should only be Remora code or supplemental files.
- initialize.m – After a Remora is added to Triton, this file initializes the Remora code, creating interfaces and executing desired callback functions. The file must be named “initialize.m” for the Remora to function.
- YourFunction.m – One or more files containing the function callbacks for your Remora. Naming conventions are not required as long as they are not the same as Triton’s built in functions.

3.2.5.3 Best Practices

- Use intialize.m to populate the **Remora** pull-down menu and create new windows with Remora-specific options and execute desired callback functions.
- Remoras should not populate existing *Control*, *Plot* or *Message Windows* with new buttons or objects as these could interfere with Triton’s built in functions. Any new controls should be created in a Remora specific window.
- Large Remoras may have many callback functions. If a Remora uses an additional control window, then it should be created in initialize.m with a handles stored in the global structure REMORA.
- The REMORA structure should be used for all Remora specific variables, parameter and handles unless specifically manipulating shared data, parameters, and handles (i.e., Triton global structures DATA, PARAMS, HANDLES).

3.2.5.4 Example Remora: Hello World

An example Remora, *hello_world* (see *3.2.5.1* above), is included in the *ExampleRemoras*\ folder and can be copied and modified for user-specified data processing or presentation needs.

Two Triton 'hooks' are included in the hello_world *initialize.m* Remora:

- Access to plot xyz data via a mouse click (i.e., WindowButtonDownFcn *Plot Window* figure object)
- Set user-defined hot-keys via xml definition file.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% initialize.m
% If the remora has been added through the Triton interface, This
% function
% is called at the start of every triton session. This file populates
% toolbars with remora specific options and callbacks.
%
% A best practice for remoras that have multiple m-files containing
% callbacks
% would be initializing new control windows and buttons here rather
% than in
% child m-files.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This should be in ever initialize function. HANDLES is the global
% varible that holds all the graphical buttons and windows of triton
global HANDLES REMORA PARAMS

% our "Hello World" button is added to the tool menu
REMORA.hello = uimenu(HANDLES.remmenu,'Label','Hello World', ...
                       'Callback', 'hello_world');

% allow "Hello World" Remora to use the mouse click down button in the
% main
% Plot Window (not the Hello World Window)
REMORA.pick.value = 1;
% define what function to run after picking in the main Plot Window
% put m-file name in REMORA.pick.fcn cell array in order of execution
REMORA.pick.fcn{1} = {'hello_pick'};

% Function for adding hotkey commands to the plot figure
xmlFile = which('keymapHello.xml');
PARAMS.keypress = xml_read(xmlFile);
set(HANDLES.fig.main,'KeyPressFcn',@handleKeypress)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

3.2.5.5 HARP Raw File Remora: HRP

A Remora for HARP raw file (format-specific) data processing operations is included in the *ExampleRemoras*\ folder to work with raw HARP data and headers providing directory listing timing checks and a means of converting raw files to XWAVs.
Also, in the *Extras*\ folder is *dd.exe* (see http://www.chrysocome.net/dd) which can be used to make disk images of raw disks form HARPs or other devices.

3.2.6 Help Menu

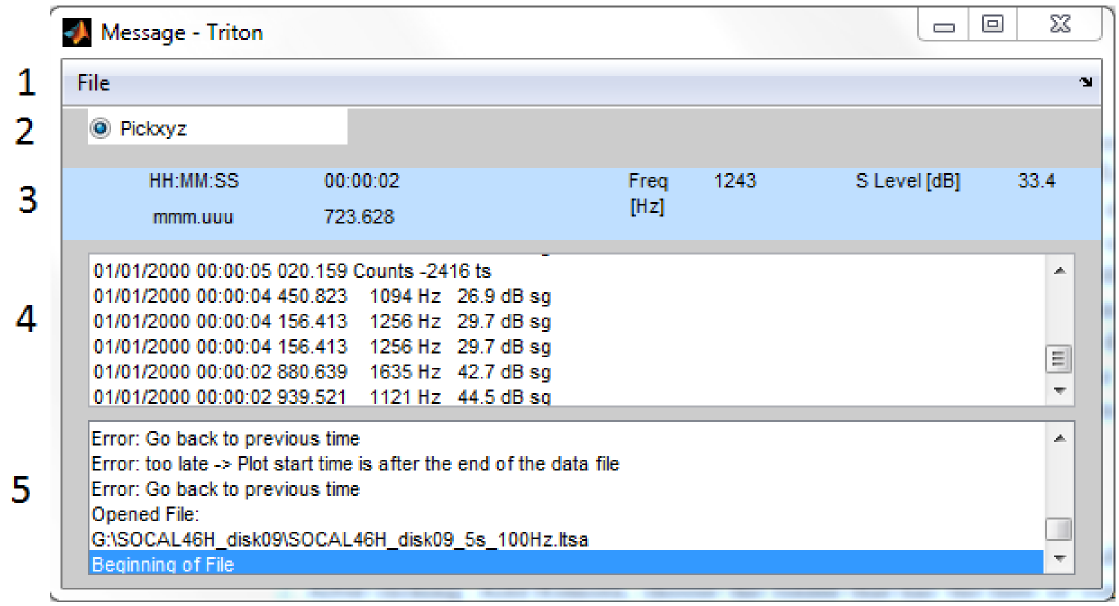
- In the *Control* Window, use **Help** pull-down menu to display Triton version number and User Manual (this document):

  - Select **About** to display pop-up window showing current Triton version number, web site to download the most-recent versions of Triton, and an email address to report bugs or request support/enhancements:

  - Select **User Manual** to view PDF of Triton User Manual included with the version of Triton currently being used.

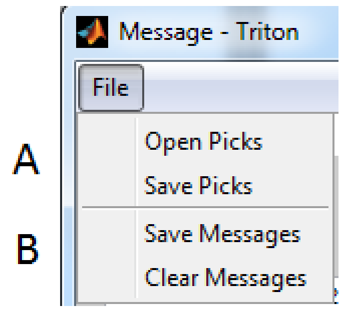
3.3 Message Window

The third standard Triton window is the *Message Window* which displays useful information such as mouse cursor location and associated data values, provides a simple means of logging of mouse cursor selections from the *Plot Window*, and provides standard messages from Triton including error messages. The mouse cursor values can be saved to a text file and the messages can be saved to a different text file.



1) **File** pull-down menu – see next section.
2) **Pickxyz** function radio button toggle. When on, mouse cursor cross-hairs are activated and left button mouse clicks in *Plot Window* are displayed in *Pickxyz Display* (section 4).
3) Displays the current mouse cursor information when in the *Plot Window*. Activated by right or left button mouse click in specific *Plot Window* (i.e., LTSA, spectrogram, timeseries, or spectra). Background color is based on what plot the cursor is in: light green for LTSA's, and light blue for spectrograms, spectra and time series.
4) *Pickxyz Display* - Pickxyz data are displayed in a scrollable, editable display. Information can be deleted, modified or added to including user defined notes. The information in this window can be saved to output text files (*.pik) and previous saved *.pik files can be uploaded to the display for additional Pickxyz's selections and modifications. In this way, it provides for a simple logging tool.
5) *Message Display* - Information based on various operations including errors and warnings are displayed to the user in scrollable window. Information in this display is not editable, but it can be saved to text files (*.msg) and used for session logging or software bug reporting.

3.3.1 File Menu



A) **Open Picks** - Open previously saved Pickxyz files (*.pik). All data in existing *Pick Display* will be lost. A warning message will pop up to inform the user of overwritten pick data.
Save Picks - Save current *Pickxyz Display* data to a *.pik ascii text file.

B) **Save Messages -** Saves current *Message Display* to a *.msg ascii text file.
Clear Messages - Clears the current *Message Display* of all information.

Appendix A1- HARP Raw Disk Format

HARP data loggers and their firmware have evolved over the years. Originally, HARPs used Motorola's 68300-series CPU and firmware version 1.17 to record non-compressed single channel

Non-Compressed, Single Channel

<table>
<tr><td rowspan="10">1 Raw HARP disk 120 GB</td><td rowspan="5">Header</td><td rowspan="3">Disk Header</td><td>Sector 0</td></tr>
<tr><td>Sector 1</td></tr>
<tr><td>Sector 2</td></tr>
<tr><td rowspan="2">Directory Listing</td><td>Sector 3-7</td></tr>
<tr><td>Sector 8-x</td></tr>
<tr><td rowspan="5">Data</td><td rowspan="5">1 Raw HARP file</td><td>Sector x+1</td></tr>
<tr><td>Sector x+2</td></tr>
<tr><td>. . .</td></tr>
<tr><td>Sector x+60,000</td></tr>
</table>

Raw Disk Size	X	N
120 GB	263	3873
100 GB	222	3207
80 GB	181	?

	Field Name *PARAMS.head.*	Length in Bytes	Start	End	Format	Example
Disk Header						
Sector 0	disktype	4	0	3	uint8	'HARP'
	unused	7	4	10	uint8	
	disknumberSector0	2	11	12	uint8	16
	unused	499	13	511	uint8	
Sector 1	unused	512	0	511	uint8	
Sector 2	nextFileSector	4	0	3	uint8	234420263
	(write_byte)	2	4	5	uint8	0
	unused	4	6	9	uint8	
	unused	2	10	11	uint8	
	firstDirSector	4	12	15	uint8	8
	maxFile	4	16	19	uint8	3908
	currDirSector	4	20	23	uint8	252
	nextFile	4	24	27	uint8	3907
	unused	4	28	31	uint8	
	unused	4	32	35	uint8	
	unused	4	36	39	uint8	
	unused	4	40	43	uint8	
	unused	4	44	47	uint8	
	unused	4	48	51	uint8	
	unused	4	52	55	uint8	
	unused	4	56	59	uint8	
	firstFileSector	4	60	63	uint8	263
	samplerate	4	64	67	uint8	200000
	disknumberSector2	2	68	69	uint8	16
	firmwareVersion	10	70	79	uint8	1.17
	description	80	80	159	uint8	"No Description"
	unused	2	160	161	uint8	
	unused	2	162	163	uint8	
	unused	2	164	165	uint8	
	unused	2	166	167	uint8	
	unused	2	168	169	uint8	

	unused	2	170	171	uint8	
	disksizeSector	4	172	175	uint8	234441643
	unusedSector	4	176	179	uint8	21383
	unused	2	180	181	uint8	
	unused	2	182	183	uint8	
	unused	328	184	511	uint8	

	Field Name *PARAMS.head.*	Length in Bytes	Start	End	Format	Example
DIR List						
Sector 8 *these are PARAMS. head.dirlist*	(year)	1	0	0	uint8	7
	(month)	1	1	1	uint8	15
	(Day)	1	2	2	uint8	31
	(hour)	1	3	3	uint8	23
	(min)	1	4	4	uint8	59
	(secs)	1	5	5	uint8	59
	(msecs)	2	6	7	uint8	999
	(blk_number)	4	8	11	uint8	120263
	(num_blocks)	4	12	15	uint8	60000
	(rec_length)	4	16	19	uint8	30720000
	(sample_rate)	4	20	23	uint8	200000
	unused	2	24	25	uint8	
	spare	6	26	31	uint8	
DATA Repeated for each raw file, usually 60,000X						
firstFile Sector	Year	1	0	0	uint8	7
	Month	1	1	1	uint8	15
	Day	1	2	2	uint8	31
	Hour	1	3	3	uint8	23
	Min	1	4	4	uint8	59
	Secs	1	5	5	uint8	59
	Msecs	2	6	7	uint8	999
	Unused	2	8	9	uint8	
	Num_samples	2	10	11	uint8	250
	Data samples*	500	12	511	uint16	

Notes

- 1 byte = 8 bits
- 1 sector=512 bytes
- 1 Rawfile = 60,000 sectors
- for additional information see comments in: read_rawHARPdata.m, read_rawHARPdir.m, read_rawHARPhead.m, and hrp2xwav.m
- HARP Data are recorded with Motorola CPU (little Endian) and some 2 byte, 4 byte words need to be swapped at byte level

Non-Compressed, Four Channel

<Insert table here>

Compressed, Single Channel

<Insert table here>

Appendix A2- XWAV File Format

1 XWAV file	Standard WAV header	RIFF header
		Format Chunk
	Additional XWAV header	HARP Chunk
	XWAV directory	HARP dir subchunk 1
		HARP dir subchunk 2
		. . .
		HARP dir subchunk 30
	Data	Data Chunk 1
		Data Chunk 2
		. . .
		Data Chunk 30

	Field Name *PARAMS.xhd. precedes each*	Length in Bytes	Start	End	Format	# of Elements	Example
Standard WAV header							
Riff Header	ChunkID	4	0	3	uchar	4	“RIFF”
	ChunkSize	4	4	7	uint32	1	filesize-8
	Format	4	8	11	uchar	4	“WAVE”
Format Chunk	fSubchunkID	4	12	15	uchar	4	“fmt “
	fSubchunkSize	4	16	19	uint32	1	16
	AudioFormat	2	20	21	uint16	1	1
	NumChannels	2	22	23	uint16	1	1
	SampleRate	4	24	27	uint32	1	200000
	ByteRate	4	28	31	uint32	1	400000
	BlockAlign	2	32	33	uint16	1	2
	BitsPerSample	2	34	35	uint16	1	16
SUBTOTAL		**36**	**0**	**35**			
Additional XWAV Header							
HARP Chunk	hSubchunkID	4	36	39	uchar	4	“harp”
	hSubchunkSize	4	40	43	uint32	1	56+30*32
	WavVersionNumber	1	44	44	uchar	1	0
	FirmwareVersionNumber	10	45	54	uchar	10	1.xxxyyyzz
	InstrumentID	4	55	58	uchar	4	“01 “
	SiteName	4	59	62	uchar	4	“ABCD”
	ExperimentName	8	63	70	uchar	8	“EXP12345”
	DiskSequenceNumber	1	71	71	uchar	1	1
	DiskSerialNumber	8	72	79	uchar	8	12345678
	NumOfRawFiles	2	80	81	uint16	1	1
	Longitude	4	82	85	uint32	1	-17912345
	Latitude	4	86	89	uint32	1	8912345
	Depth	2	90	91	uint16	1	5555
	Reserved	8	92	99	uchar	8	00000000
SUBTOTAL		**64**	**36**	**99**			

	Field Name *PARAMS.xhd. precedes each*	Length in Bytes	Start	End	Format	# of Elements	Example
Additional XWAV Header cont. *Repeated for each RawFile(k) n=32(k-1), usually 30*							
HARP dir Subchunks	year(k)	1	100+n	100+n	uchar	1	7
	month(k)	1	101+n	101+n	uchar	1	12
	day(k)	1	102+n	102+n	uchar	1	31
	hour(k)	1	103+n	103+n	uchar	1	23
	minute(k)	1	104+n	104+n	uchar	1	59
	secs(k)	1	105+n	105+n	uchar	1	59
	ticks(k)	2	106+n	107+n	uint16	1	999
	byte_loc(k)	4	108+n	111+n	uint32	1	1066
	byte_length(k)	4	112+n	115+n	uint32	1	30000000
	write_length(k)	4	116+n	119+n	uint32	1	60000
	sample_rate(k)	4	120+n	123+n	uint32	1	200000
	gain(k)	1	124+n	124+n	uint8	1	1
	padding	7	125+n	131+n	uchar	7	0000000
SUBTOTAL		**32+n**	**100**	**131+n**			
Data Chunk							
Data Chunk	dSubchunkID	4	132+n	135+n	uchar	4	“data”
	dSubchunkSize	4	136+n	137+n	uint32	1	datasize
	DATA		138+n				

For additional information see comments in rdxwavhd.m, wrxwavhd.m, and initdata.m in Triton.

For typical full HARP XWAV files, file size will be 900,001,068 bytes or 30 raw files (n) 30,000,000 bytes/rawfiles & 1068 byte header.

Appendix A3- Software Routines (*.m files)

<Needs Updating>

Name	Description
audvidplayer	Play sound of DATA vector (ie plotted data only)
bin2xwav	Convert *.bin (ARP binary) files into *.x.wav files
calc_ltsa	Calculate spectral averages and save to ltsa file
check_dirlist_times	Get recording parameters and run difftime_dirlist for one file or whole directory
check_ltsa_time	Check to see if plot time is within file limits
check_time	Check time of start of plot based on PARAMS.plot.dvec time
ck_ltsaparams	Check user defined ltsa parameters and adjusts/gives suggestions of better parameters so that there is integer number of averages per xwav file, called by mk_ltsa.
control	Toggle on/off control window pull-down menus and buttons set and implement newtime, newtseg, newstep, coordinate display
control_ltsa	Toggle on/off control window pull-down menus and buttons set and implement newtime, newtseg, newstep, coordinate display
coorddisp	Display cursor values from Plot window in message window
decimatexwav	Decimate XWAV file
decimatexwav_dir	Decimate all XWAV files in directory
difftime_dirlist	This function reads raw HARP files disk directory and compares times between directory entries, used for data quality checking
disp_headSummary	Display PARAMS.headall structure in useful format
disp_msg	Display messages in message window
disp_pick	Display pickxyz in message window
displaybut	Display button operation
editHeader_psds	Used to change time or other header values for psds files (not used)
editHeader_xwav	Used to change time or other header values for XWAV files
filepd	File pull-down menu options/operations
get_headers	Open data files and read headers
get_ltsadir	Get directory of WAV/XWAV files
get_ltsaparams	Get parameters needed for generating LTSA’s from user
get_recordingparams	Get recording parameters needed for checking dirlist times
getIndexBin	Get time bin index in LTSA plot
helppd	Help pull-down menu
hrp2wav	Convert *.hrp files into *.wav files
hrp2xwav	Convert *.hrp files into *.x.wav files
init_coorddisp	Initializing cursor values
init_ltsadata	Initializing ltsa data stuff
init_ltsaparams	Initializing ltsa parameters
initcontrol	Initializing control window GUI
initdata	Initializing data and timing info

Continued on next page

Name	**Description**
initparams	Initializing parameters
initpulldowns	Generate figure pull-down menus
initwins	Initialize plot, control and command (display) windows
little2big_2byte	Reads N x 2 array (a) and converts the 2 values from each row from little endian format to big endian format – array (b)
little2big_4byte	Reads N x 4 array (a) and converts the 4 values from each row from little endian format to big endian format – array (b)
logfmap	Return a matrix for premultiplying spectrograms to map the rows into a log frequency space
make_multixwav	Script to run write_hrp2xwavs over many hrp files to make xwavs
mk_headSummary	Make Summary of header values
mk_ltsa	Make LTSAs from XWAV files in a directory
mkspecgram	Make spectrogram plot from DATA
motion	Control motion of DATA plot with push button in control window
motion_ltsa	Control motion of DATA plot with push button in control window
multibin2xwav	Convert multiple *.bin files to *.x.wav files
obs2xwav	Convert *.obs files into *.x.wav files
pickxwav	Turn on picking time in LTSA files and open corresponding xwav file (i.e. zoom in)
pickxyz	Pick x, y, z cursor data from plot window
plot_ltsa	Plot LTSA data in plot window
plot_specgram	Plot spectrogram of data in plot window
plot_spectra	Plot spectra of data in plot window
plot_timeseries	Plot time series of data in plot window
plot_triton	Checks to see which plots are to be plotted and plots them
rdxwavhd	Reads XWAV or *.wav file header
read_ltsadata	Read LTSA data
read_ltsahead	Read LTSA header and directories
read_rawHARPdata	Reads one raw HARP disk data file, ~60000 sectors the first timing header will be read and used.
read_rawHARPdir	Calls the function read_rawHARPhead to get disk header values then reads the directory list, rearranges values and puts values in global PARAMS variable
read_rawHARPhead	Read raw HARP disk header info from raw HARP datafile (*.hrp) and put disk header info into global variable structure PARAMS
read_xwavHDRfile	Hdrfile for hrp2xwavs input for generating xwavs from hrp files
readseg	Read a segment of data from opened file
timenum	Convert string time in format made from timestr.m (used to solve rounding problems created by datestr.m and outputs msecs and usecs)

Name	**Description**
timestr	Can be used instead of datestr.m to solve the rounding problems created by datestr.m and outputs msecs and usecs
toolpd	Tools pull-down menu operation
triton	MAIN
write_hrp2wavs	Writes multiple XWAV (*.x.wav) files(~1GB) from single raw HARP disk file (*.hrp). The HRP files are essentially images if raw disks
write_ltsahead	Setup values for ltsa file and write header & directories for new ltsa file
write_XWAVhead	Write XWAV header values to output files
wrxwavhd	Write XWAV header values to output files