

University of Rhode Island

DigitalCommons@URI

Physical Oceanography Technical Reports

Physical Oceanography

2-21-2022

PIES and CPIES Data Processing Manual

Maureen Kennelly

University of Rhode Island, mkennelly@uri.edu

Karen L. Tracey

University of Rhode Island, krltracey@uri.edu

Kathleen A. Donohue

University of Rhode Island, kdonohue@uri.edu

D. Randolph Watts

University of Rhode Island, randywatts@uri.edu

Follow this and additional works at: https://digitalcommons.uri.edu/physical_oceanography_techrpts

Creative Commons License



This work is licensed under a [Creative Commons Attribution-Noncommercial-Share Alike 3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/).

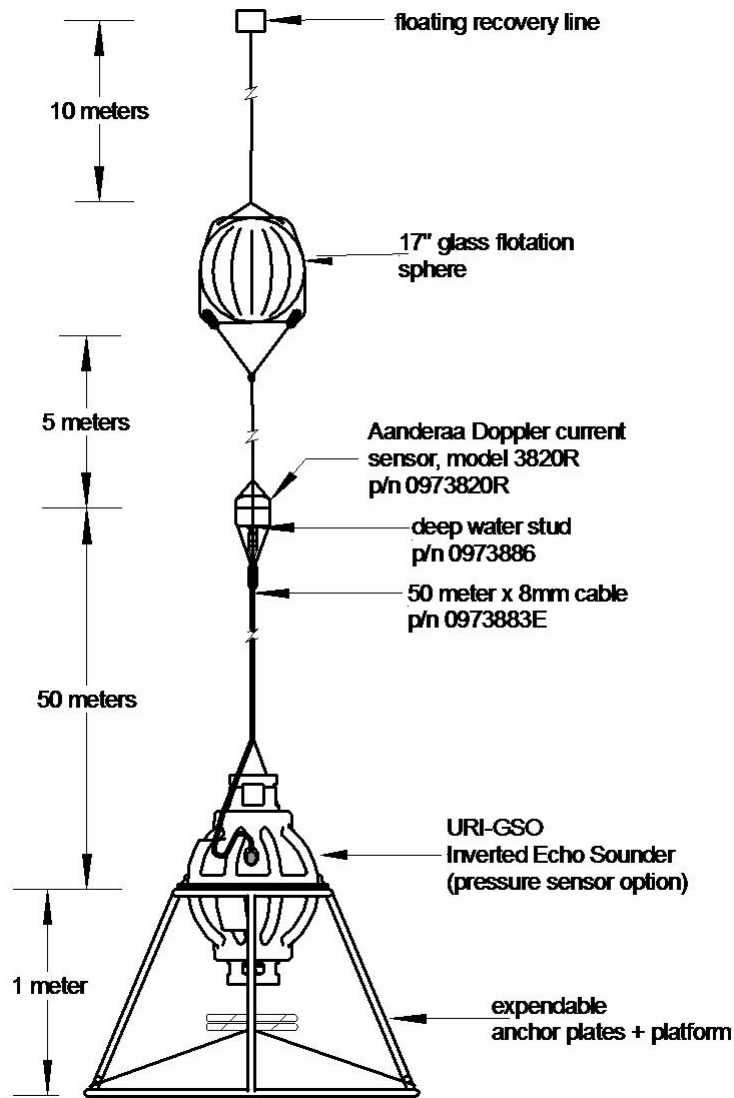
Recommended Citation

Kennelly, Maureen; Tracey, Karen L.; Donohue, Kathleen A.; and Watts, D. Randolph, "PIES and CPIES Data Processing Manual" (2022). *Physical Oceanography Technical Reports*. Paper 38.

https://digitalcommons.uri.edu/physical_oceanography_techrpts/38https://digitalcommons.uri.edu/physical_oceanography_techrpts/38

This Article is brought to you for free and open access by the Physical Oceanography at DigitalCommons@URI. It has been accepted for inclusion in Physical Oceanography Technical Reports by an authorized administrator of DigitalCommons@URI. For more information, please contact digitalcommons@etal.uri.edu.

PIES and CPIES Data Processing Manual



by
Maureen A. Kennelly, Karen L. Tracey, Kathleen A. Donohue and D. Randolph Watts

Physical Oceanography Technical Report 38
Graduate School of Oceanography
University of Rhode Island
Narragansett, Rhode Island

First edition February 21, 2022
https://digitalcommons.uri.edu/physical_oceanography_technical_reports/38

ABSTRACT

The Inverted Echo Sounder (IES) is an ocean bottom-moored instrument that measures the vertical acoustic travel time (VATT) round-trip from the sea floor to the sea surface and back. The VATT varies principally due to changes in the temperature profile of the water column, making the IES well-suited for monitoring changes in temperature structure and dynamic height (baroclinic signal). Currently, the Model 6.2, a combined IES, data-logger, and acoustic release, with measurements of bottom pressure and temperature (PIES) and optional measurements of current speed and direction (CPIES, with attached AanderaaTMDoppler current sensor) is produced at URI/GSO. Data are processed in situ and are available (optional) remotely by an acoustic telemetry link. In addition to the IES-measured baroclinic signals, barotropic near-bottom pressure variations may be measured with the optional pressure sensor. A report was written in 1991 describing IES data processing [*Fields et al.*, 1991]. Since that report, significant improvements have been made to both IES hardware and software, warranting an update of the IES data processing. The report by *Kennelly et al.* [2007] documents the standard processing steps contained in IESpkg 3, which has been used since the early 2000s, for IES/PIES/CPIES Models 6.1 and 6.2 at URI/GSO. More recently, IESpkg 4 was developed to allow more flexibility in the processing steps and data outputs, and to process the Fast PIES versions that sample 96 travel times each hour. This report documents the processing steps in IESpkg 4 and it repeats as much of the original text of *Kennelly et al.* [2007] as is still applicable. A separate document, Inverted Echo Sounder User's Manual, IES Model 6.2, describes the IES hardware and instrument configuration.

Contents

Abstract	i
List of Figures	iv
List of Tables	v
List of Acronyms	vi
Quick Start Guide	vii
1 Introduction	1
2 Overview of IES Processing	2
3 Step by Step Processing using IES_GUIDAT2MAT	5
3.1 Setting up InitGuiDat2Mat.dat	5
3.2 Running IES_GUIDAT2MAT	6
4 Step by Step Processing using IES_GUIDRIVER	9
4.1 Setting up InitGuiDriver.dat	9
4.2 Running IES_GUIDRIVER	11
4.2.1 Step 1: Windowing and Unfiltered Values	13
4.2.2 Step 2: Initial Detiding	19
4.2.3 Step 3: Despiking	24
4.2.4 Step 4: Dedrifting	27
4.2.5 Step 5: Final Detiding	29
4.2.6 Step 6: Low-pass filtering	30
4.2.7 Step 7: Plotting	30
4.3 Additional Information	31
4.3.1 Time base	31
4.3.2 No travel time data	35
4.3.3 Processing individual variables	35
4.3.4 Delete button	35
5 Acknowledgments	35
References	37
Appendices	37
A IES_GUIDAT2MAT additional figures	38
A.1 Content of .mat files created	38
A.2 Example plots generated by IES_GUIDAT2MAT for PIES and CPIES	39
A.3 Examples of plots generated by IES_GUIDAT2MAT for currents (CPIES only)	41
B Response Analysis of Tides	41
B.1 Theory	41
B.2 Tidal Prediction Options	43
B.3 Compiling Respo.for	44
C Recommended additional processing	46
C.1 Additional velocity processing recommended	46
C.2 Additional dedrifting recommended	47
D IES_GUICOUNTS2MAT	47

List of Figures

1	Recommended directory tree structure	1
2	Default IES_GUIDAT2MAT window	3
3	Default IES_GUIDRIVER window	4
4	Example InitGuiDat2Mat.dat file	6
5	IES_GUIDAT2MAT window - READY State	7
6	Example InitGuiDriver.dat file	10
7	IES_GUIDRIVER window after initialization	12
8	Full range of tau measurements	14
9	IES_GUIDATALIMITS to set travel time bounds	14
10	Examples IES_GUIDATALIMITS figures with windowed values	15
11	Histograms for initial windowing of pressure measurements	16
12	Windowed pressure time series	16
13	Windowing of Paroscientific temperature	17
14	Paros temperature equilibrium for hourly values	18
15	Paros temperature equilibrium after with user selection of points to omit	18
16	DCS temperature equilibrium	19
17	Example mat and log contents after Step 1, windowing	20
18	Example CP01.mat file contents after Step 2, initial detiding	22
19	Example CP01.mat tidal phases and amplitudes	23
20	Example processing log after Step 2, initial detiding	23
21	IES_GUIDESPIKE window before and after despiking	24
22	IES_GUIDESPIKE window showing expanded view for hand editing	25
23	IES_GUIDESPIKE window after hand editing	25
24	IES_GUIDESPIKECM window before despiking	26
25	Example processing log after Step 3, despiking	26
26	IES_GUIDEDRIFT window showing data prior to and after drift removal	28
27	IES_GUIDEDRIFT window showing expanded views of data	28
28	Example map and log contents after Step 4, dedrifting	29
29	Example mat and log contents after Step 6, lowpass filtering	31
30	Final plots of unfiltered data	32
31	Final plot of low-pass filtered data	32
32	Sampling timeline for 4 travel time pings every 10 minutes	33
33	Sampling timeline for 24 travel time pings every 60 minutes	33
34	Delete files message	36
A.1	Example TXXX_#.mat file contents	38
A.2	Example PXXX_#.mat file contents	38
A.3	Example CXXX_#.mat file contents	38
A.4	Example EXXX_#.mat file contents	38
A.5	Example KXXX_#.mat file contents	39
A.6	IES_GUIDAT2MAT Travel Time (seconds)	39
A.7	IES_GUIDAT2MAT Pressure (10kPa)	39
A.8	IES_GUIDAT2MAT Temperature (degrees C)	40
A.9	IES_GUIDAT2MAT Engineering measurements	40
A.10	IES_GUIDAT2MAT Temperature counts	41
A.11	IES_GUIDAT2MAT DCS measurements for CPIES	41
B.12	Spherical harmonics corresponding to tidal amplitude functions	43
B.13	Description of contents of Tidal Prediction Option Files	45

D.1	Default IES_GUICOUNTS2MAT window	48
D.2	Example IES_GUICOUNTS2MAT window after coefficients loaded	49

List of Tables

1	Raw data files downloaded from IES	2
2	Data files generated by IES_GUIDAT2MAT	8
3	Record Length for Tidal Prediction Options	20
4	Tidal Constituents Provided by Processing	21

List of Acronyms and Variables

Acronyms

CPIES	IES with optional current meter and pressure sensor
DCS	Doppler current sensor
IES	Inverted echo sounder
GSO	Graduate School of Oceanography
Paros	Paroscientific Digiquartz pressure sensor
PDT	Pulse Delay Telemetry
PFF	Plot File Format
PIES	IES with optional pressure sensor
PSN	IES serial number and mission configuration number
URI	University of Rhode Island
UT	Universal Time
VATT	Vertical Acoustic Travel Time

CPIES Variables

P, prs	bottom pressure
T, tmp	temperature
tau	vertical acoustic travel time
u, v	east and north current velocities

Quick Start Guide

1. Put the .DAT files downloaded from the PIES/CPIES in a 'raw_data' directory
2. Create InitGuiDat2Mat.dat and InitGuiDriver.dat files and put them in a 'mat_data' directory.
3. After starting Matlab, make mat_data the working directory using the 'cd' command.
4. Run IES_GUIDAT2MAT:
 - Enter the IES serial number and index: XXX_#
 - Press Initialize or set the parameters manually.
 - Press Begin File Conversions.
 - Output mat-files and figures will be stored in the 'Write Directory' (typically called 'mat_data')
5. Run IES_GUIDRIVER:
 - Enter the IES serial number and index: XXX_#
 - Press Initialize or set the parameters in the left panel manually.
 - Set the parameters in the right panel manually, including the processing steps to execute.
 - Press Begin Processing.
 - Output files (processed mat-files, logs, and figures) will be stored in the 'Write Directory' (typically named 'processed').
 - Notes:
 - (a) Processing steps must be executed sequentially for each variable, but variables can be processed or reprocessed separately.
 - (b) Whenever possible, execute 'Step 1' for travel time before or at the same time as pressure, since their time bases are linked.
 - (c) Output interval for travel time is always hourly. Output intervals of the other variables can either be hourly or the same as the sampling interval.
 - (d) Each variable can be reprocessed at any step without having to reprocess the other variables.
 - (e) Press 'Delete Files' button to clear out all data files, logs, figures, and workspaces to reprocess an instrument from the beginning.

1 Introduction

The Inverted Echo Sounder (IES) is an ocean bottom-moored instrument that measures the vertical acoustic travel time round-trip from the seafloor to the sea surface and back. The latest models come equipped with Paroscientific bottom pressure sensors, termed PIES. Models with optional current sensor are termed CPIES. The typical IES is configured to emit twenty-four 12 KHz pings per hour, in programmable bursts of 4, 8, 12 or 24 pings. The specialized Fast PIES is configured to emit 96 pings per hour in six bursts of 16 pings at 10 minute intervals. The time required for each ping to reach the surface and return is recorded within the instrument on removable (type ATA compact flash) memory cards (15–512 Mbyte capacity). Data are stored in ASCII files as described in the IES User’s Manual.

This report documents the processing steps carried out after data have been downloaded from a recovered instrument. IES data can be processed on a variety of platforms running either Windows, UNIX or Mac OS X operating systems with a processing package developed at URI/GSO. The current version of the processing software for Models 6.1 and 6.2 PIES/CPIES instruments is IESpkg4 which uses MATLAB Version 2019 and above. The IES processing codes have been tested with Windows 10 and Mac OS X. The MATLAB signal processing toolbox is required to run the processing programs.

Prior to Processing - Setting up the Directory Structure

To protect the integrity of raw data files, the following directory structure is recommended. Create a directory for the experiment or study. Then create subdirectories for raw, converted, and processed data. Within the raw data directory, create subdirectories for each instrument. An example directory tree is shown in Figure 1 for an experiment called “Manual”. Within directory “Manual” the subdirectories for raw and converted data are “raw_data” and “mat_data”, the read and write directories for the program IES_GUIDAT2MAT.m. In “raw_data” a subdirectory has been made for IES serial number “109” with site designator “CPO1” (i.e., CPO1_109). Do not include spaces in the directory or site designator names. Site designators are usually assigned prior to launch by the Principal Investigator. The processing code will prepend the word “PIES” to any site designator entered as numeric only. Serial numbers are assigned by URI/GSO at the time of manufacture.

The third directory shown in Figure 1 called “processed” will contain the data files created by the program IES_GUIDRIVER.m. Data from CPIES serial number 109 will be used throughout this manual to illustrate processing steps.

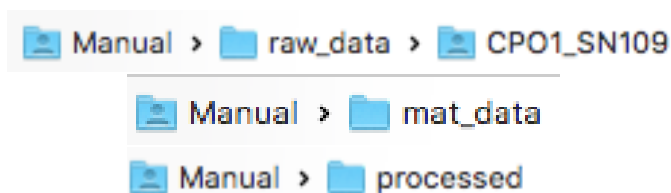


Figure 1: Recommended directory tree structure

After an IES is recovered at sea, raw data from the flash memory card are downloaded and should be put in the raw data subdirectory for that particular IES (e.g. Figure 1 Manual/raw_data/CPO1_109). Up to seven ASCII files are downloaded from each instrument and are listed in Table 1.

XXX is the instrument serial number (e.g. 109) and # is the mission configuration number (typically 1). The mission configuration number depends on the number of times the mission configuration has been modified. See the IES User’s Manual for further information about mission configuration. The

format and contents of the ASCII .DAT files are described in detail in the IES User’s Manual.

The TELEM.DAT file contains daily, rather than hourly samples, and is used with the Pulse Delay Telemetry option from a ship. The IES processing package discussed in this report does not handle daily values and these data must therefore be processed separately if necessary. The TELEM.DAT file will not be discussed further in this manual. Consult the IES User’s Manual for a description of the TELEM.DAT file.

On UNIX machines, permissions for the raw data files listed in Table 1 and the raw data directory (e.g. GOM/raw_data/CPO1_109) should be restrictive to prevent overwriting (*i.e.*, set read only permission for raw data directory and files).

Table 1: Raw data files downloaded from IES.

File	Contents
TXXX_#.DAT	Travel time data
PXXX_#.DAT	Pressure and temperature data
EXXX_#.DAT	Engineering data
KXXX_#.DAT	Raw pressure and temperature counts from the Paroscientific pressure sensor
IESXXX.LOG	System log information
TELEM.DAT	Daily values to be telemetered
CXXX_#.DAT	Acoustic Doppler current sensor data (CPIES only)

2 Overview of IES Processing

Run the two main processing programs sequentially from their GUI interfaces. IES_GUIDAT2MAT.m is the interface for IES_DAT2MAT.m. IES_GUIDRIVER.m is the interface for IES_DRIVER.m. IES_GUIDAT2MAT must be run first to create .mat files from the raw (.DAT) data files listed in Table 1. Next run IES_GUIDRIVER, the main processing program for IES Models 6.1 and 6.2. The programs are designed to process travel time, pressure, temperature, and velocity data. However, the codes will run if any of those measurements are missing. Engineering data are also plotted and stored. Two final data products are produced:

1. unfiltered data file containing
 - despiked travel time and temperature
 - despiked, detided, and dedrifted pressure
 - despiked currents and current meter temperature (CPIES only)
2. lowpass filtered, subsampled file at user-specified interval in days

Program 1: IES_GUIDAT2MAT.m

IES_GUIDAT2MAT.m is a GUI interface (Figure 2) to easily input several parameters to the program IES_DAT2MAT.m which performs the following operations:

- plots the raw data from the IES .DAT files listed in Table 1 and saves into .mat files
- gives an initial look at the data

- rescales the units to typical values

Each variable processed will be stored in a .mat file corresponding to the .DAT file name listed in Table 1.

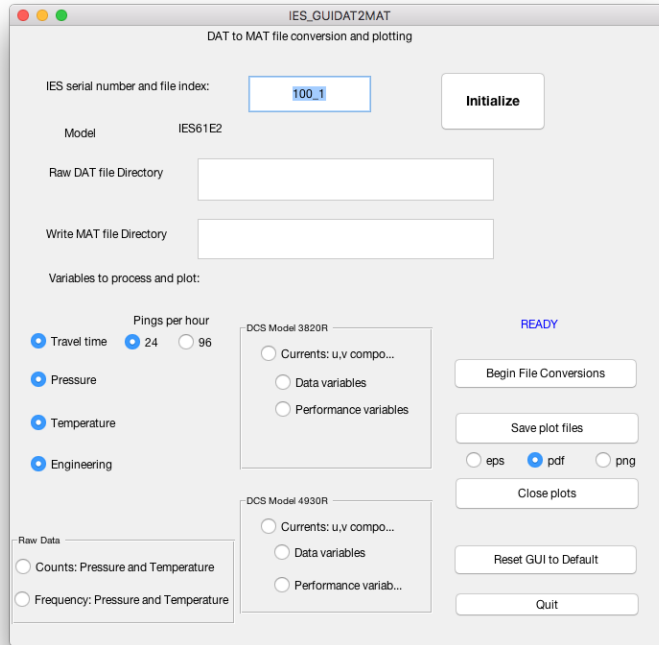


Figure 2: Default IES_GUIDAT2MAT window

Program 2: IES_GUIDRIVER.m

IES_GUIDRIVER.m is a GUI interface (Figure 3) to easily input many parameters to IES_DRIVER.m which performs up to seven processing steps from initial windowing through low-pass filtering.

Step 1) Windowing and output intervals

- windows to remove large data spikes.
- selects representative travel time for each hour using modified quartile (recommended), median or specialized methods.
- sets the output intervals for pressure and temperature and currents as either hourly (default option) or the same as the sampling intervals.
- identifies on-bottom records (between launch and release command).
- eliminates pressure and temperature records prior to the internal temperature reaching equilibrium with surroundings.
- assigns the mid-sample time for each variable in decimal days, relative to January 1 0000 UT of launch year.
- checks for missing samples (e.g. during telemetry operations) and fills the gaps with NaNs.

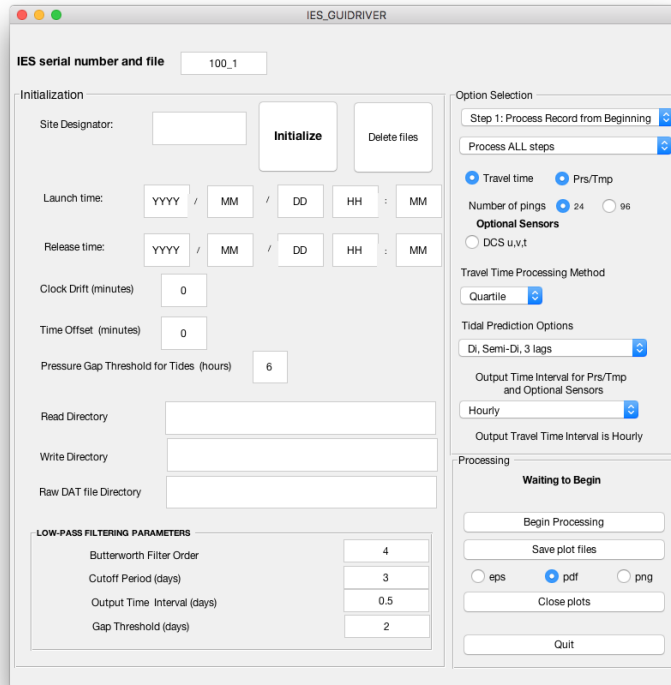


Figure 3: Default IES_GUIDRIVER window

Step 2) Initial Detiding

- calculates and removes tides from the pressure record using a FORTRAN program called RESPO.for based on the response analysis method of *Munk and Cartwright* [1966].
- calculates and removes tides from the travel time record using the tidal constituents determined for pressure with the VATT time base.

Step 3) Despiking

- identifies data spikes using Slope or Tukey methods and replaces them with interpolated values.
- slope method identifies measurements that change more rapidly from the preceding value by more than the specified rate (tolerance).
- Tukey method performs median filters on the time series and identifies values that differ from the filtered record by the specified tolerance as outliers (requires the MATLAB signal processing toolbox).
- travel times can only be despiked using the slope method; either method will work on pressure, temperature, and currents.
- hand editing of spikes can also be performed.

Step 4) Dredrifting

- fits and removes drift from the pressure record (uses MATLAB optimization toolbox if available).

Step 5) Final Detiding

- adds the tides to the dedrifted pressure record and then recalculates the tides.
- stores the amplitude and phase of the eight major tidal constituents.
- removes the updated tides from the travel times.

Step 6) Low-pass Filtering

- applies a Butterworth filter to all variables using user specified parameters (requires MATLAB signal processing toolbox).
- chops beginning and end of records to remove startup transients.
- interpolates output time series to be “on the hour”.

Step 7) Plotting

- produces 3 standard plots of final products.

NOTE: All GUI windows may be proportionally resized using the mouse or using the resizing options (“minimize” and “resize”) in the upper right (Windows) or left (Mac) corner of the GUI title bar (e.g. Figure 2). Both IES_GUIDAT2MAT and IES_GUIDRIVER can be closed using either the **Quit** button or the “close” option in the upper corner of the title bar.

3 Step by Step Processing using IES_GUIDAT2MAT

IES_GUIDAT2MAT.m is the GUI interface to easily input several parameters to IES_DAT2MAT.m. Each variable processed by this program will be output to a .mat file corresponding to the original .DAT file name listed in Table 1 (e.g. the output file for ./raw_data/CPO1_109/T109_1.DAT will be ./mat_data/T109_1.mat). For convenience, the GUI interface can be populated with the information stored in a parameter file called InitGuiDat2Mat.dat. This parameter file should be stored in the “mat_data” subdirectory and IES_GUIDAT2MAT should be executed from the same directory.

3.1 Setting up InitGuiDat2Mat.dat

For the user’s convenience, prior to running IES_GUIDAT2MAT, a parameter file called InitGuiDat2Mat.dat may be created. The parameter file can be considered a lookup table, based on IES serial number, containing information about directory structure. The parameter file is used to initialize the GUI with values so that the user does not have to enter them manually. A single parameter file can be made for all instruments in a study using your preferred editor (e.g., WordPad, vi, emacs). Default values for all instruments are listed first, followed by instrument specific entries.

An example InitGuiDat2Mat.dat file is shown in Figure 4. In this example there are sections labeled “array” and “psn”. The “array” section contains the description of your study (e.g. “Manual”) followed by entries listing the directories common to all instruments (e.g. rawdir and writedir). This array section is not required; it is provided as a convenience when multiple instruments will be processed.

The “PSN” section is mandatory if the **Initialize** button will be pressed on GUI interface. The first line of this section must contain the IES serial number and mission configuration (e.g. 109_1). Subsequent lines may contain the rawdir, writedir, or sitedir parameters. Sitedir is the directory containing the individual IES’s raw data files (.DAT files). If sitedir is specified, it will be appended to rawdir to create a full pathname in the GUI window.

Each PIES or CPIES requires its own PSN section, thus multiple PSN sections may occur in the file. In this example, the second PSN section for instrument 152_1 specifies the full path of the .DAT files using the rawdir parameter rather than specifying the sitedir parameter. When any parameter is specified in both the array and PSN sections, the value in the PSN section takes precedence.

IMPORTANT: InitGuiDat2Mat.dat needs to be located in the current working directory, otherwise MATLAB will prompt the user for the directory location.

```

% InitGuiDat2Mat.dat
%
% Directories to specify
% rawdir -- main directory containing instrument directories
% writedir -- directory to save all the .mat files
% sitedir -- instrument directory containing the .DAT files. This
will be appended
% to the rawdir to create the full pathname.
%
array Manual
rawdir /users/karen/Manual/raw_data/
writedir /users/karen/Manual/mat_data/

psn 109_1
sitedir CP01_SN109/

psn 152_1
rawdir /users/karen/Manual/raw_data/CP03_SN152/

```

Figure 4: Example InitGuiDat2Mat.dat file

3.2 Running IES_GUIDAT2MAT

Start MATLAB from the directory where you will store the .mat files, e.g. ./mat_data. At the MATLAB prompt type

```
>>IES_GUIDAT2MAT
```

The default IES_GUIDAT2MAT window will appear as in Figure 2. Carry out the following steps:

- Replace “100_1” with the 3-digit IES serial number and file index which corresponds to the mission configuration (e.g. 109_1). After you press Enter, the Model will be displayed in the GUI. Serial number 109 is a model 6.1E2 and appears as “IES61E2 ” in Figure 5.
- If the parameter file has been created (See section 3.1), press **Initialize**.
- Otherwise, set the following options manually.
 - Enter the Raw DAT file directory.
 - Enter Write MAT file directory.
 - Note the pre-selected radio buttons in the lower left: Travel time, Pressure, Temperature and Engineering. The default number of pings for travel time is 24; select 96 pings for Fast PIES.
 - Toggle the Counts or Frequency buttons: Pressure and Temperature radio button under Raw Data in the lower left. This feature is rarely, if ever, needed. See Appendix D for more information.
 - For CPIES toggle the appropriate radio buttons for your DCS model in the middle of the GUI Window. See below for further information about DCS model specific buttons.

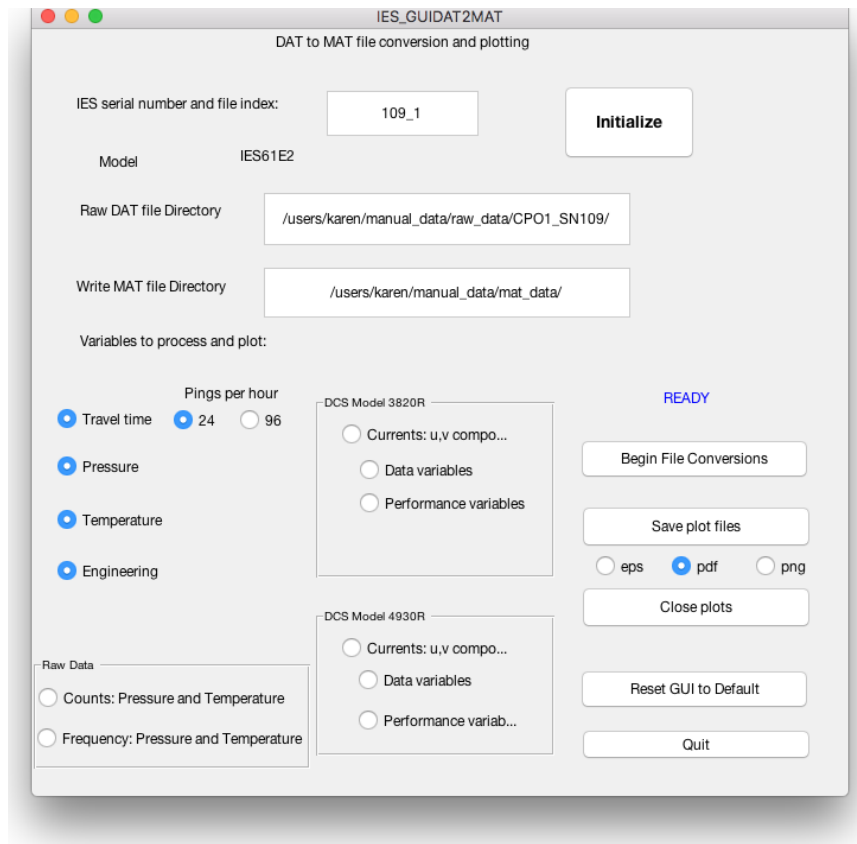


Figure 5: Default IES_GUIDAT2MAT window - READY State

- Press the button on the right side of the GUI window **Begin File Conversions**.

CPIES can be configured using different Aanderaa Doppler current sensors, models 3820 and 4930. The outputs from these models differ, so you must choose the processing options based on the model number. Examine the paperwork provided with your instrument to locate the DCS model number associated with your instrument; model 4930 sensors are sometimes referred to as ‘Z-pulse’. Specifically, on the User’s Disk provided with your instrument, navigate to the IES_SpecificInformation folder and open the Specific Information file. Scroll to the ‘Aanderaa Test & Specifications’ page where the DCS model number is listed (e.g. Product: DCS 4930). If you are unable to locate the model number, examine the seventh (7th) column in CXXX_#.DAT. For model 4930 sensors, this column will contain ‘4930R’.

Most CPIES will be configured to write the u (positive eastward), v (positive northward) components to the CXXX_#.DAT file, so select the appropriate radio button based on the model number. Most CPIES will be configured to record both data and performance variables. Data variables (u, v, and temperature) are always recorded, so this radio button should always be selected. Normally, the radio button for performance variables (such as signal strength, ping count, or tilts) should also be selected. Different versions of model 4930 sensors output a different number of performance variables. As of August 2012, the code automatically counts the number of columns in the CXXX_#.DAT file. Do not select the radio button if the performance variables were not recorded.

MATLAB will create figures and .mat files according to the selected variables. For a PIES, a maximum of 8 Figure windows and 4 .mat files (listed in Table 2) may be created. For a CPIES, the maxima are 10 figures and 5 .mat files.

Table 2: Data files generated by IES_GUIDAT2MAT

File	Contents
TXXX_#.mat	Travel time data
PXXX_#.mat	Pressure and temperature data
EXXX_#.mat	Engineering data
KXXX_#.mat	Raw pressure and temperature counts data
CXXX_#.mat	Acoustic Doppler current sensor data (CPIES only)

After the file conversions have completed, the word READY on the right side of the GUI window is replaced with FINISHED (not shown). The variables contained in the output .mat files listed in Table 2 are shown in Appendix A:

- Figure A.1: T109_1.mat travel time variables.
- Figure A.2: P109_1.mat pressure and temperature variables.
- Figure A.3: C109_1.mat current meter variables (CPIES only).
- Figure A.4: E109_1.mat engineering variables (13 variables stored in array Efrq).
- Figure A.5: K109_1.mat counts variables (pressure and temperature).

NOTE: If a .DAT file does not exist for a parameter, the remaining data can still be processed. However, a warning message will be displayed in the MATLAB window. For example, if there is no TXXX_#.DAT file, the message “*** WARNING: Could not open Travel Time” will be displayed. The rest of the files will be converted to .mat files and figures for the converted parameters will be generated.

It is recommended that you save and print the figures created by IES_GUIDAT2MAT by pressing the button **Save plot files**. Select the radio button for the plot file format (PFF). Options for PFF are .eps, .pdf, or .png. IES_GUIDAT2MAT will create a “figs” subdirectory (e.g. Manual/mat_data/figs) in which to store the plot files. The saved filenames will have the syntax PIESXXX_#_VAR_N.PFF. VAR denotes the variable plotted: travel time (tau), pressure (prs) Paroscientific temperature (tmp), engineering variables (engA, engB, and engC), temperature counts (tct), pressure counts (pct), and Doppler current sensor (dcsA and dcsB). N denotes the plot number. Typically the default scaled plot files will be written with N=1, followed by an expanded scale version with N=2. Successive saves will increment N by 1. Axis limits will reflect whatever appears in the Figure window at the time the button is pressed. Examples of the MATLAB figure windows are included in Appendix A.

To assess data quality, examine the figures generated by IES_GUIDAT2MAT.

Figure 1: Travel time (seconds). The full scale is 0-10 seconds but most of the data will be concentrated in a narrow range of approximately 0.05 seconds (e.g. Figure A.6 left). Check the travel time quality by zooming in using the toolbar in the figure window (e.g. Figure A.6 right).

- an instrument moored between 2000 and 5000 m depth records travel time data in the range 2.6

to 6.6 seconds.

- values near 10 seconds are bad and indicate no echoes.
- good quality data have small scatter.

Figure 2: Pressure (10kPa which is equivalent to dbar). The default plot will show the full range of pressure (e.g. Figure A.7 left). The signal typically looks nearly constant because it includes a wide range of pressure values in going from air (while on the deck) to the ocean bottom. Zooming in will show the bottom pressure which is typically dominated by the tidal signal (Figure A.7 right).

Figure 3: Temperature (degrees C). The default plot will show a large range (e.g. Figure A.8 left), going from air temperature values followed by a rapid decrease as the instrument equilibrates to the value at the seafloor with a linear temperature for the remainder of the time series. Zoom in on the linear portion to see the temperature stabilize approximately 24 hours after deployment (e.g. Figure A.8 right).

Figures 4-5 (CPIES) If the instrument is a CPIES, Figure 4 will contain current speed, current direction, a vector stick plot and current meter temperature (e.g. Figure A.11 left). Figure 5 will contain signal strength, compass direction, north and east tilt (deg) and ping count (Figure A.11 right).

Figures 4-6 (PIES) or 6-8 (CPIES). Engineering plots. These are primarily used to assess instrument engineering performance (e.g. Figure A.9).

Figure 7 (PIES) or 9 (CPIES). Temperature counts plotted with the default y-axis limits (e.g. Figure A.10 left).

Figure 8 (PIES) or 10 (CPIES). Pressure counts with the default y-axis limits(e.g. Figure A.10 right).

To complete this step, press the **Quit** button in the lower right corner of the IES_GUIDAT2MAT GUI window or **red circle** in the upper left corner of the menu (Figure 5).

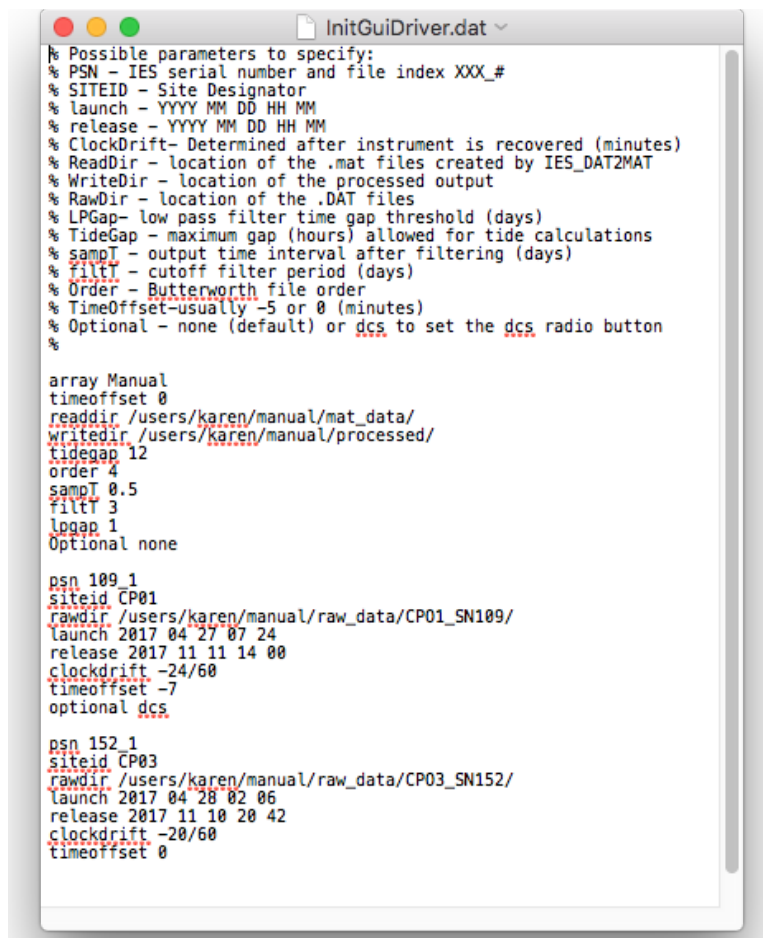
4 Step by Step Processing using IES_GUIDRIVER

4.1 Setting up InitGuiDriver.dat

Prior to running IES_GUIDRIVER, a parameter file called **InitGuiDriver.dat** may be created for the user's convenience. This parameter file can be considered a lookup table, based on IES serial number, containing information relevant to a particular instrument and study. The parameter file is used to initialize the GUI with values so that the user does not have to enter them manually. A single parameter file can be made for all instruments in a study using your preferred editor (e.g., WordPad, vi, emacs). Default values for all instruments are listed first, followed by instrument specific entries. The user will need to consult the log sheets from the deployment/recovery cruises for some values (e.g., timeoffset, clockdrift, launch and release times). **NOTE:** All times should be entered as UT. Entries in the parameter file are not case sensitive.

An example InitGuiDriver.dat file is shown in Figure 6. Descriptions of possible parameters to specify are listed in the comments section of InitGuiDriver.dat. **NOTE:** Items in the right panel of GUI window cannot be entered into the InitGuiDriver.dat file. The exception is the DCS radio button which can be set with the optional parameter.

The file consists of an optional “array” section and followed by one or more mandatory “PSN” sections.



```
InitGuiDriver.dat
% Possible parameters to specify:
% PSN - IES serial number and file index XXX_#
% SITEID - Site Designator
% launch - YYYY MM DD HH MM
% release - YYYY MM DD HH MM
% ClockDrift- Determined after instrument is recovered (minutes)
% ReadDir - location of the .mat files created by IES_DAT2MAT
% WriteDir - location of the processed output
% RawDir - location of the .DAT files
% LPGap- low pass filter time gap threshold (days)
% TideGap - maximum gap (hours) allowed for tide calculations
% sampT - output time interval after filtering (days)
% filtT - cutoff filter period (days)
% Order - Butterworth file order
% TimeOffset-usually -5 or 0 (minutes)
% Optional - none (default) or dcs to set the dcs radio button
%

array Manual
timeoffset 0
readdir /users/karen/manual/mat_data/
writedir /users/karen/manual/processed/
tidegap 12
order 4
sampT 0.5
filtT 3
loggap 1
Optional none

psn 109_1
siteid CP01
readdir /users/karen/manual/raw_data/CP01_SN109/
launch 2017 04 27 07 24
release 2017 11 11 14 00
clockdrift -24/60
timeoffset -7
optional dcs

psn 152_1
siteid CP03
readdir /users/karen/manual/raw_data/CP03_SN152/
launch 2017 04 28 02 06
release 2017 11 10 20 42
clockdrift -20/60
timeoffset 0
```

Figure 6: Example InitGuiDriver.dat file

In the “array” section specify a descriptor for your study (e.g. Manual). The lines following “array” specify any parameters common to all instruments (e.g. readdir, writedir, lpgap) or parameters for which you wish to specify a default value (e.g. time offset).

The PSN sections are used to specify parameters specific to each instrument (e.g. launch, release, clockdrift, rawdir). The keyword “psn” begins the list for a particular instrument. There should be no spaces in the value entered for siteid, although spaces will be replaced with the underscore character. Consult the deployment/recovery cruise log sheets for the values of launch, release, and clockdrift. Clock drift should be entered in minutes, however equations are allowed (e.g., for 109_1, from the recovery log sheet, a clock drift of -24 seconds can be entered as -24/60 minutes, see Figure 6). TimeOffset is typically zero and it is recommended that the default value zero be entered with the “array” parameters (Figure 6). However, if there are multiple instruments at the same site, one will have a time offset with the offset value listed on the cruise log sheets. Enter the TimeOffset for the second instrument at a site with its parameters (see timeoffset entry for psn 109_1 in Figure 6). Including “timeoffset 0” with the “array” entries prevents a nonzero value being inadvertently applied in a case of no time offset. When a parameter is specified in both the array and PSN sections, the psn

value takes precedence.

Not all values need to be specified in `InitGuiDriver.dat`. When values are not specified, any default values displayed in the GUI (e.g., Lowpass Filtering Parameters in Figure 3) will be used. The user may also enter the values in the GUI manually after initialization. **IMPORTANT:** `InitGuiDriver.dat` needs to be located in the current working directory (e.g. `Manual/mat_data/`) or in the Read Directory specified in the GUI (prior to pressing **Initialize**), otherwise MATLAB will prompt the user for the directory location.

4.2 Running IES_GUIDRIVER

This program performs the following steps:

- Step 1: process record from the beginning (windowing and unfiltered values)
- Step 2: initial detiding
- Step 3: despiking
- Step 4: dedrifting
- Step 5: final detiding
- Step 6: low-pass filtering
- Step 7: plotting

`IES_GUIDRIVER` creates two subdirectories “`wrkspc`” and “`figs`” in the “Write Directory” (e.g. `Manual/processed/wrkspc/` and `Manual/processed/figs/`). The “`wrkspc`” subdirectory holds a MATLAB workspace created by `IES_GUIDRIVER` as well as tide logs generated during detiding. The “`figs`” subdirectory holds the final figures created during plotting, Step 7.

The `.mat` files produced by `IES_GUIDAT2MAT` listed in Table 2 are the input data sets for `IES_GUIDRIVER`. At each step, MATLAB saves selected variables in a workspace mat-file (e.g. `WorkspaceXXX_#.mat`). The contents of the workspace allow you to reprocess the data starting at any step. In general, you will not need to use the workspace for other purposes. Information about the steps carried out and the processing parameters specified is also stored in the `WorkspaceXXX_#.mat` file. After processing has finished, the information is written to log file called “`PIESXXX_#.processing.log`” in the “write” directory. If you repeat an intermediate step for a single variable, the log file and the workspace will be updated for only the reprocessed variable.

The final data sets will be named based on what is specified as the “site designator” which can be entered into the `InitGuiDriver.dat` parameter file as `siteid` (Figure 6). For example, if you specify “`CP01`” as the site designator then `CP01.mat` and `CP01lp.mat` will be the final data sets created for the unfiltered and low-pass filtered data sets, respectively. Default names `PIESXXX_#.mat` and `PIESXXX_#lp.mat` are used if a site designator is not specified. We recommend using short site designators since they are also used to name the structure arrays in the final data sets.

At the MATLAB prompt, type:

```
>>IES_GUIDRIVER
```

The default `IES_GUIDRIVER` window will appear as in Figure 3. The GUI parameters can be initialized manually or from a previously created parameter file (see Section 4.1 `InitGuiDriver.dat`). To initialize with the values entered in an `InitGuiDriver.dat` file, that file must be in your current

working directory or specified in the Read Directory entry in the GUI window. Enter the IES serial number and file index (e.g. **109_1**) and press the **Initialize** button in the upper middle of the GUI. The default values in the left panel of Figure 3 will be replaced with the values associated with **109_1** from the InitGuiDriver.dat file (listed in Figure 6) as shown in Figure 7. Parameters not specified in InitGuiDriver.dat will keep the default values of Figure 3.

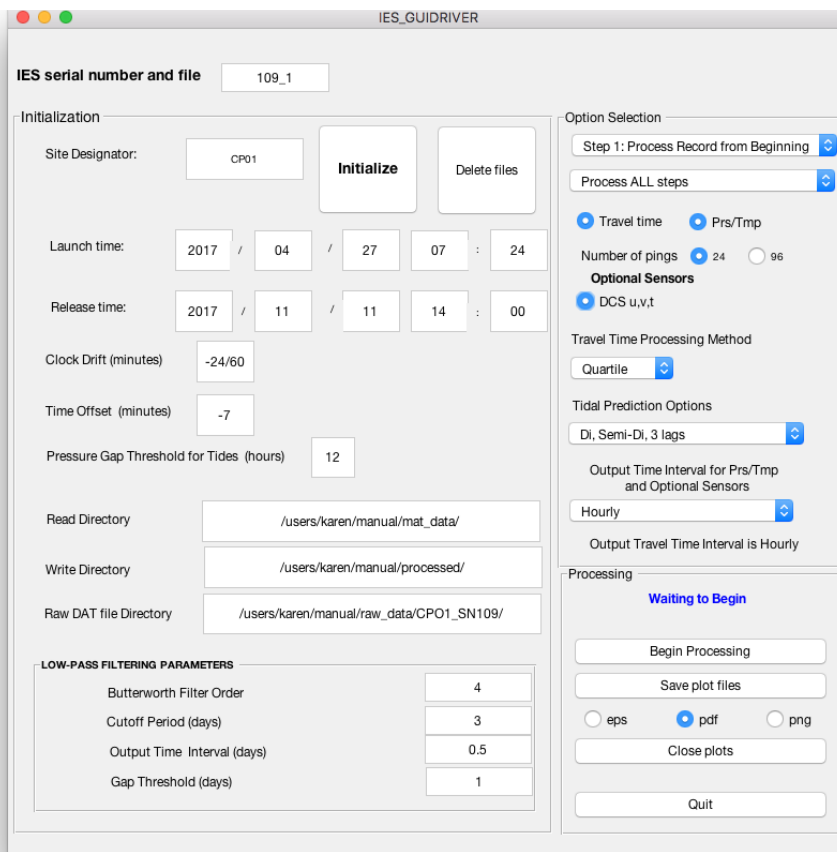


Figure 7: Example IES_GUIDRIVER window after initialization for processing ALL steps.

The values in the right panel of Figure 3 must be selected manually by the user before beginning the processing. The default processing assumes that all steps will be carried out from beginning to end (*i.e.*, Figure 7 top right pull down menu shows **Step 1: Process Record from Beginning** and second pull down menu shows **Process ALL Steps**). The user may choose to start and stop at other steps. Note that the steps must be run sequentially, they cannot be executed out of order. However, IESpkg4 does permit individual variables to be reprocessed without affecting any of the other variables.

Use the pull down menus in the right panel to:

- select the method to process the travel time measurements into hourly values.
- select the option for determining the tides.
- select the output time interval for both the Paros and DCS variables.

Additional information about these options are discussed in the following sections.

NOTE: Once initialized and the first step has been run, any value filled in (as in Figure 7) will be retained for all subsequent processing steps and need not be specified again unless changes are desired.

4.2.1 Step 1: Windowing and Unfiltered Values

This first step performs several tasks. Based on the launch and release times specified in the IES_GUIDRIVER window, it identifies which records were collected while the instrument was on the bottom. It assigns a time base for each variable in decimal days, relative to January 1 0000 UT of the launch year. It checks for missing samples (during telemetry operations) and fills the gaps with NaN values. It selects a representative travel time for each hour using modified quartile, median, or specialized methods. If the output time interval for pressure and optional sensors was selected to be hourly, then mean hourly values will be calculated for pressure, temperature and the DCS velocities and temperatures. If the output time interval was selected to be the sampling interval, no averaging will be performed. With user interaction, it removes large spikes and eliminates pressure and temperature records prior to the internal temperature reaching equilibrium with surroundings. It processes the DCS temperature in a similar manner.

Most PIES and CPIES are designed to emit 24 pings per hour, so this is the default option in the GUI. If your model is a Fast PIES, select the 96 ping per hour option.

Specify the **Travel Time Processing Method** (Quartile (default), median, specialized, or do not reprocess) using the pull down menu in the middle of the right panel of the IES_GUIDRIVER window. We recommend using the “Quartile” method which tracks the clean, sharp leading edge of the raw travel time measurements well. Using this method, each hourly burst of tau measurements are passed through two stages of windowing to eliminate outliers and to reduce the noise in the hourly values. In the first stage, the measurements in each burst are sorted and the first quartile value is calculated. These hourly values are then smoothed with a 4 hour median filter. Subsequently, the travel time measurements within each hourly burst are windowed with an asymmetric window about these fitted values. Because a histogram of the travel time measurements during a typical hourly sample is skewed with more spread to the right of the mean (*i.e.*, toward longer travel times) an asymmetric window (from minus 100 counts (~ 5 ms) to plus 50 counts (~ 2.5 ms)) is used to exclude more long taus than short taus from the data window. The n measurements which fall within this asymmetric window are passed to the second stage of processing. In the second stage, the first quartile is found for the n travel time measurements in each hourly burst. Next the $n/6$ travel times nearest the quartile value are averaged to obtain the final travel time for each hour. For example, if $n = 24$ for hour h (*i.e.*, all samples in the hour are acceptable), the hourly travel time will be an average of four measurements. This averaging reduces the noisiness of the hourly values.

The “median” option is supplied because it is an easily understood statistical method. However, the median method will shift the hourly values to longer travel times and may create some false offsets.

The “specialized” option is provided to allow users the flexibility of creating their own method for selecting the appropriate hourly tau value. The user should edit and modify IES_TTSPECIAL.m for this purpose. One example of another method is IES_TT3RD.m which selects the third earliest ping for the hourly tau value.

4.2.1.1 Travel time radio button selected

A figure window will appear (Figure 8) showing a plot of the full range of tau measurements (in seconds) as a function of record number. The first step is to specify an acceptable range of good tau

counts to eliminate early and/or late echo returns from being used during subsequent processing steps. Position the pointer below the good tau counts (near 4 seconds) and click the mouse to select the lower bound. Then reposition the pointer above the good tau counts and click the mouse again to select the upper bound.

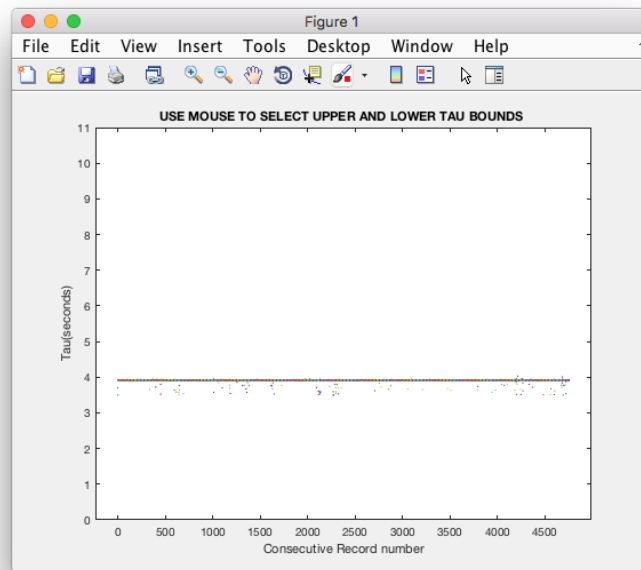


Figure 8: Example figure with full range of tau measurements.

Pick one of these two options to begin refining hourly tau values

Select bounds with crosshairs

Use mouse to select upper and lower bounds

Hit ENTER in plot when finished with crosshairs

Upper Bound 3.67201

Lower Bound 2.96647

Use Crosshairs

Enter bounds manually

Enter upper and lower bounds

then hit Apply Bounds

Upper Bound

Lower Bound

Apply Bounds

RED points show hourly tau values

Accept Hourly Values and Finish

Figure 9: IES_GUITAULIMITS is used to set the minimum and maximum values of travel time.

Next, THE IES_GUITAULIMITS window opens (Figure 9). The GUI instructs you to pick one of two options to begin refining hourly tau values. Usually you will begin with the “crosshairs” option (left panel). Press the button **Use Crosshairs**. A new figure window will open (Figure 10 left).

The y-axis limits shown are those selected from Figure 8. In Figure 10 the blue dots are the raw tau measurements and the red line shows the resulting hourly tau values. The goal is for the red curve (the calculated hourly tau values) to track the lower edge of the dense cloud of blue dots. Position the cursor below the red dots and click the mouse. Then reposition the cursor to adjust the upper bound. The hourly taus will be recalculated and the plot redrawn with narrower y-axis range (Figure 10 right). The selected tau bounds will be shown as green lines on the plot. You can repeatedly use the cursor and mouse to refine the upper and lower tau bounds; each time the hourly taus are recalculated and replotted. Additionally, the Upper and Lower Bounds in both the left and right panels of the GUI window (Figure 9) will update. Typically when good bounds are set, the lower bound is below the blue cloud since the leading edge is usually sharp/clean but the upper bound cuts through the cloud. For reference, as an approximation, a 1000 m change in thermocline depth would produce about a 0.05 second change in tau. To terminate the selection of the bounds with the cursor, press **Enter** on the keyboard while the cursor is in the plot window. Control returns to the IES_GUITAULIMITS window.

Instead of using the cursor, the right panel of IES_GUITAULIMITS allows the tau bounds to be entered manually. Manual entry is useful if you need to repeat this processing step, but were satisfied with the tau bounds recorded in the processing log. Enter the upper and lower values on the right side of the IES_GUITAULIMITS window (Figure 9) and press **Apply Bounds**. A new figure window just like Figure 10 (right) will open, but the cursor used for the crosshairs option will not be active. Repeat setting the limits manually until you are satisfied. **Note:** you can switch between the crosshairs and manual options to select tau bounds at any time.

When you are finished refining the tau bounds, press the button **Accept Hourly Values and Finish** located in the bottom of the GUI window to end the processing and exit the window.

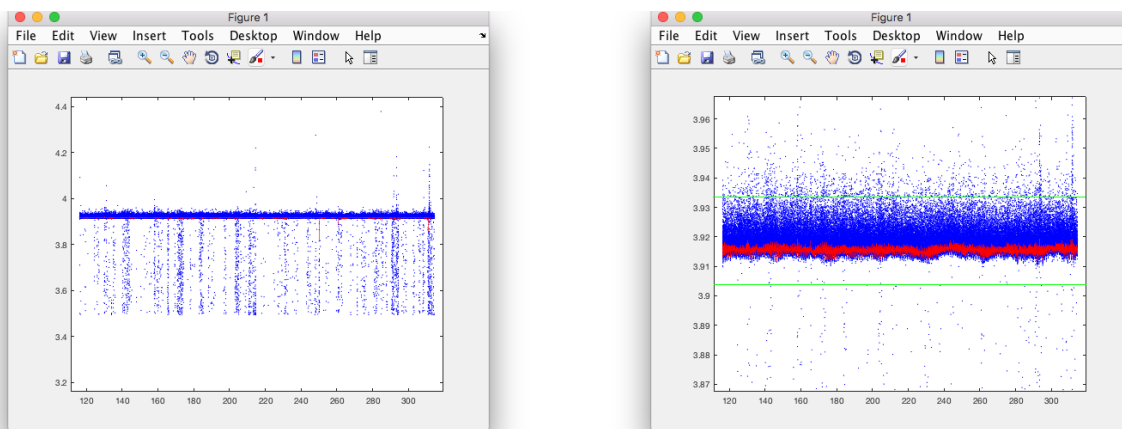


Figure 10: Examples of IES_GUITAULIMITS windows with hourly tau values for (left) wide and (right) refined limits.

4.2.1.2 Prs/Temp Radio Button Selected

Histograms are used to determine the acceptable ranges of good Paroscientific pressure and temperature measurements. The next figure window to open shows two histograms of pressure (Figure 11). Working with the histogram in the lower plot, set limits for the minimum and maximum pressures. Following the directions (in red) in the Figure window, click in the lower plot to select left and right limits of the histogram. The histograms are then replaced by a plot of the full pressure time series with the selected limits (Figure 12). Again directions in the Figure window guide the user. To change

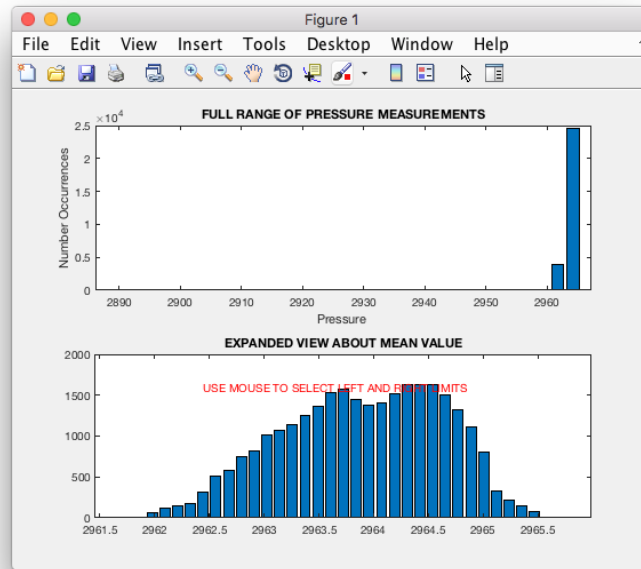


Figure 11: Histograms generated by Step 1 for initial windowing, full range of pressure measurements (top) and expanded view about mean value (bottom).

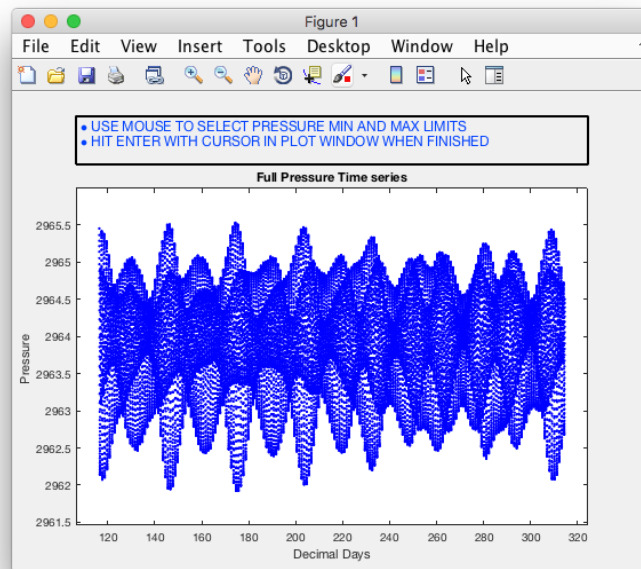


Figure 12: Pressure time series after initial windowing with the tidal signals included.

the bounds click minimum and maximum limits. Repeat as many times as necessary to bound the pressure measurements. If you make a mistake and need to expand the axes range of the plot, click somewhere outside the plot and instruction boxes. When you are satisfied with the range of pressure, while the cursor is in the figure window press **Enter** to finish with pressure. Next, repeat these steps for temperature (see Figure 13).

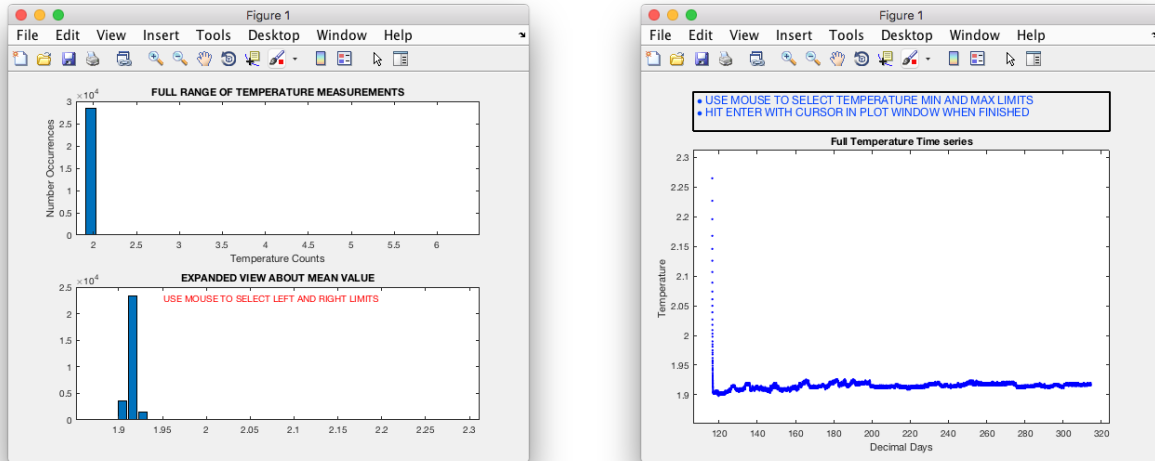


Figure 13: (left) Histograms of Paroscientific temperature generated by Step 1 for initial windowing, (top) full range of temperature measurements and (bottom) expanded view about mean value. (right) Temperature time series after initial windowing.

NOTE: The pressure and temperature time series are typically very clean and may only need bounds changed once, if at all. You can press **Enter** at any point with the cursor in the figure window; you don't have to select new bounds in either histogram or time series figure for either variable.

After the temperature bounds selection is completed, the next figure is a time series of hourly temperature from the beginning of the record (Figure 14). This plot allows the user to identify when the internal temperature of the IES reached equilibrium with the surroundings. Pressure and temperature data prior to the equilibrium time will be discarded. The internal temperature of the IES is measured by a Paroscientific sensor located within the IES glass sphere (*i.e.*, NOT in direct contact with seawater) and is needed for the conversion of pressure counts to decibars. When the figure opens, a four day long segment of temperature is plotted with values up to 12 hours after launch circled in red. These points will be eliminated unless the user selects a different value. Follow the directions in the Figure window. Position the crosshairs and then click the mouse on the last point to omit. The revised points will be replotted; Figure 15 shows examples for temperature time series with both the hourly and sampling time intervals. When finished, hit **Enter** with the cursor in the window.

4.2.1.3 DCS Radio Button Selected

If the output time interval for pressure and optional sensors was selected to be hourly, then hourly averages of the velocity components and the DCS temperature will be calculated next. Otherwise, no averaging will be performed and the time series will have the same increments as the sampling interval. No figures of the velocity components are generated during this step. However, the user must determine when the current meter temperature has reached equilibrium with the surroundings. Equilibration

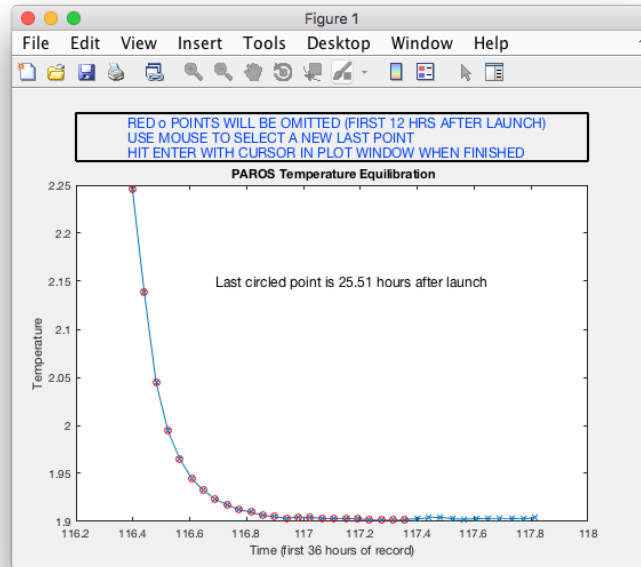


Figure 14: Initial window generated by Step 1 showing beginning of hourly Paros temperature time series with points to omit (12 hours after launch) shown in red.

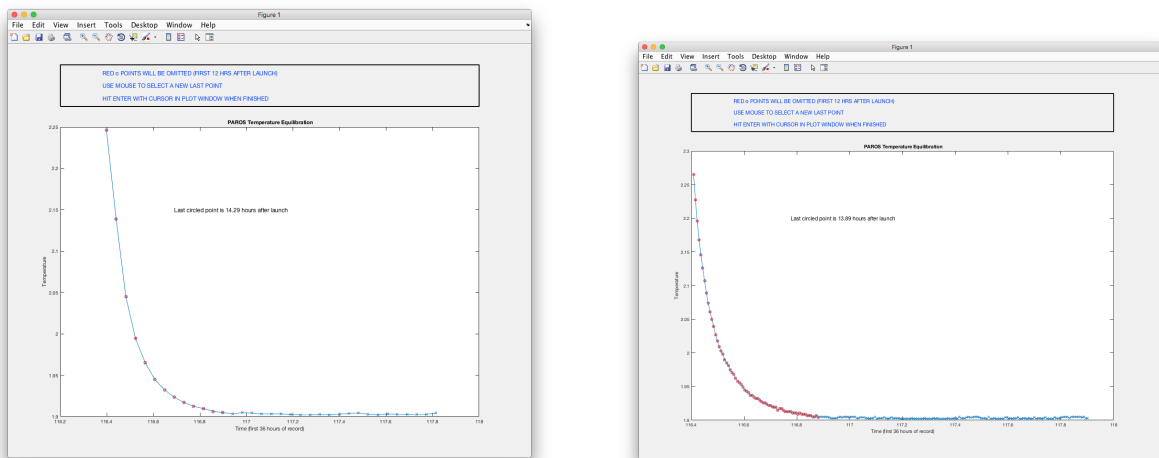


Figure 15: Examples of beginning of the Paros temperature time series with user selected points to omit shown in red for (left) hourly and (right) sampling interval time series.

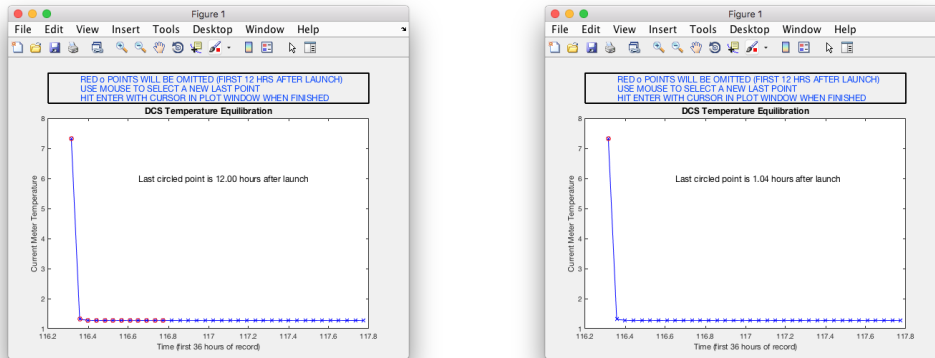


Figure 16: DCS temperature equilibrium. Points circled in red are flagged to be eliminated. (left) Initial state. (right) After user selection.

occurs more rapidly than for the Paroscientific sensor because the current meter temperature sensor is on the circuit board inside the metal DCS rather than in the IES glass sphere. Follow the directions in the Figure window (see Figure 16) to select the last temperature record to omit prior to equilibrium with surroundings and hit **Enter** in the plot window.

4.2.1.4 Completion of Step 1

At the completion of this step, a CP01.mat file is created with the structure array contents shown in Figure 17. The output file and structure array names are derived from the site designator set in the IES_GUIDRIVER window (see Figure 7). The CP01.comments structure should read ‘Created after windowing’ for the variables processed. The log file (PIES109_1_processing.log) should have entries similar to Figure 17, listing launch and release times, time base information, number of gaps filled, the method for tau processing, and based on the interactive windowing, the bounds for tau, pressure and temperature, and when both the Paros and DCS temperature equilibria were achieved. If a time offset had been used in processing, a line would appear in the log preceding the tau processing method stating “Time offset of XX minutes was included”.

4.2.2 Step 2: Initial Detiding

This step removes the tidal contribution from the pressure record using a FORTRAN program called RESPO.for (Response Analysis of Tides), based on the work of *Munk and Cartwright* [1966]. Response analysis constructs and applies a predictive filter which represents the ocean’s response to gravitational forcing. Unlike the related harmonic analysis, the response analysis assumes nothing about which frequencies are present, because the input function is derived directly from Newtonian-Keplerian orbital motions; the input function contains all the variations of the astronomic forcing regardless of size. The oceanic response is considered distinctly from the astronomic forcing. The method also has a more physical basis than harmonic analysis since it treats the ocean as a dynamical system. Details of the response method are given in Appendix B.

Before running this step it will be necessary for you to create an executable version for your computer using a FORTRAN compiler (such as g77 or gfortran). Guidance on how to perform this task is provided in Appendix B. (Please contact us at URI if you require assistance.)

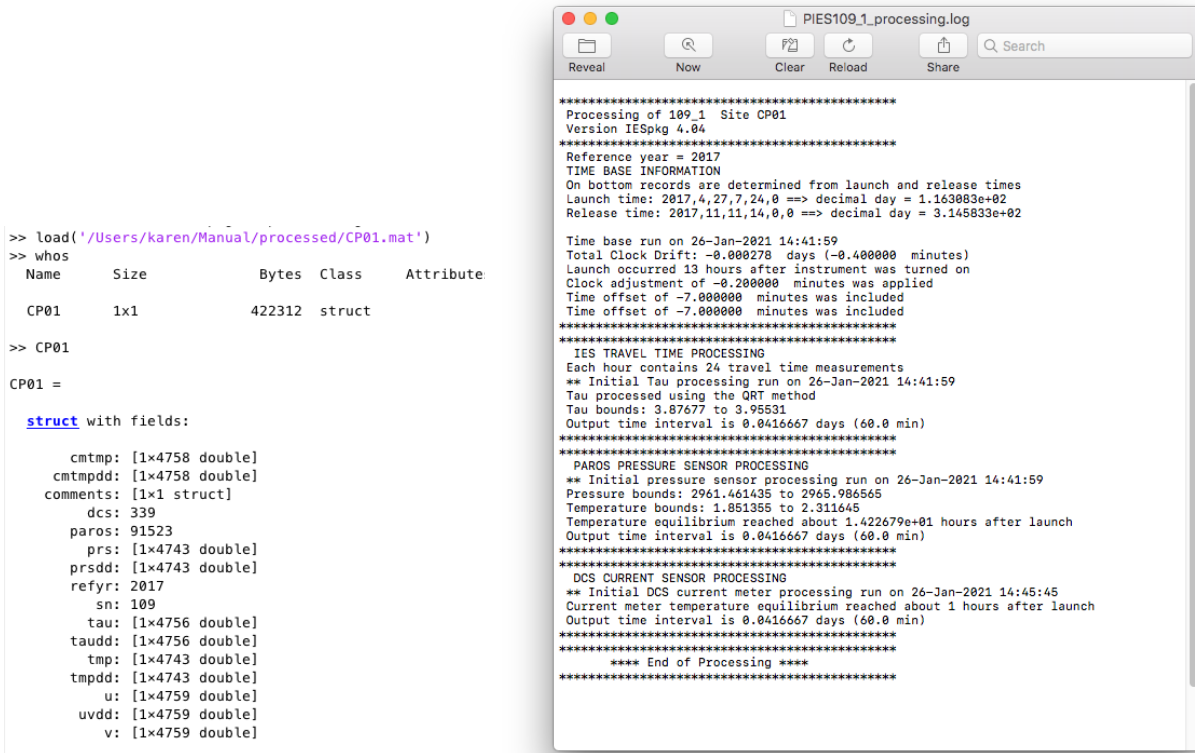


Figure 17: Example (left) CP01.mat file and (right) processing log contents after Step 1, windowing.

A pull down menu provides two tidal prediction options in the IES_GUIDRIVER window:

- Di, Semi-Di, 3 lags (default)
- Di, Semi-Di, 5 lags

where Di refers to diurnal and Semi-Di to semi-diurnal. The tidal prediction option to use depends on the length of the pressure time series. Table 3 provides recommended guidelines for choosing the tidal prediction option from the pull down menu. Currently, these options calculate the total gravitational

Table 3: Record Length for Tidal Prediction Options.

Record length	Tidal Prediction Option
<1 year	Di, Semi-Di, 3 lags
> 1 year	Di, Semi-Di, 5 lags

potential for the lunar and solar tidal constituents (Table 4). In the future, an option may be added to calculate the sun's radiational potential. Listings of the files associated with the two tidal prediction options are included in Appendix B.2 as well as a description of the file contents.

The value for the pressure gap threshold for tides (maximum allowable data gap size that can be linearly interpolated before performing the tidal analysis) must be specified (see middle of the left panel

Table 4: Tidal Constituents Provided by Processing

Name	Symbol	Approximate Period (Solar Hours)
Semi-diurnal:		
Principal lunar	M2	12.4
Principal solar	S2	12.0
Larger lunar elliptic	N2	12.7
Luni-solar semi-diurnal	K2	11.97
Diurnal:		
Luni-solar diurnal	K1	23.9
Principal lunar diurnal	O1	25.8
Principal solar diurnal	P1	24.1
Larger lunar elliptic	Q1	26.9

of the IES_GUIDRIVER window in Figure 7). The default value in the GUI is 6 hours, but this can be reset manually or during GUI parameter initialization.

NOTE: If processing is specified to begin at any step other than Step 1, the parameter fields in the left panel of IES_GUIDRIVER will be replaced with blanks. The original values are saved in the workspace. However, the parameters in the processing program retain the values that were set in Step 1. If you wish to change a value, you can enter the new one at this time. Note that the tau processing method updates to “Do not reprocess” which refers to the processing that occurred in Step 1.

When Step 2 is executed, the following message will appear in the MATLAB command window:

Entering IES_DETIDE. If an error occurs, create a new FORTRAN executable of RESPO.for

If an executable compatible with your operating system exists, a time series of the tide will be calculated and removed from the pressure series. The following message will be displayed in the MATLAB command window:

```
Existing executable code runs. Exiting IES_DETIDE.
*** End of Processing ***
```

If the existing executables are not compatible with your operating system, one or more error messages will appear in the MATLAB command window.

The RESPO.for program creates a detailed log file with the filename derived from the site designator ending with .tidelog, (e.g., CP01.tidelog) which is written to the wrkspc subdirectory.

If your data contains gaps larger than the threshold (default = 6 hours), this step performs the tidal analysis on the longest segment of the time series. Then the calculated weights are passed to RESPO.for and the tide is predicted for the full record. A second log file is created with “_fullrecord.tidelog” as part of the name.

The tidal coefficients determined for the pressures are subsequently used to remove the tides from the travel time measurements using IES_GENERATETIDE.m. The tide is generated using the travel

time time base, and this is subsequently scaled to travel time units (e.g. seconds). The scale factor is calculated as

$$\text{scale factor} = \frac{2}{cP_{\text{fact}}}$$

where

$$P_{\text{fact}} = 1 + (\bar{P})(1.1e^{-5}) \quad \text{and} \quad 1.1e^{-5} = \left(\frac{1}{c}\right) \left(\frac{\delta c}{\delta \bar{P}}\right)$$

and where $c = 1500 \text{ m s}^{-1}$ is the nominal sound speed and \bar{P} is the mean bottom pressure at the instrument location. The scale factor is recorded in the processing log. The corresponding tide log file (e.g. CP01.tau.tidelog) is also written to the wrkspc directory. The travel time record will only be detided if the pressure record was detided.

At the completion of this step, the CP01.mat file (Figure 18) contains the detided pressure data (CP01.prs), the time series of the predicted tide (CP01.tide), and the tidal amplitudes (CP01.amplitude) and phases (CP01.phase) extracted from the separate log file created by the RESPO program (CP01.tidelog). The mat-file also contains the detided travel times (CP01.tau) and predicted tide removed (CP01.tau_tide). Figure 19 lists amplitudes (in dbar) and phases (in degrees) calculated for the CP01 example. The CP01 comments structure for pressure will consist of two lines containing ‘Created after initial detiding’ and ‘prs=prs-tide’. The comments for travel time will be similar. You may want to write down the values of CP01.amplitude and CP01.phase for later comparison with the values calculated after Step 5 (final detiding).

The processing log file (Figure 20) will document if initial detiding has been performed for both travel time and pressure. Under the pressure processing the log will list the gap threshold, the tidal constituents removed (diurnal and semi-diurnal), and the maximum admittance control value (default: 3 lags).

```
>> CP01

CP01 =

    struct with fields:

    amplitude: [1x1 struct]
    cmtmp: [1x4758 double]
    cmtmpdd: [1x4758 double]
    comments: [1x1 struct]
    dcs: 339
    paros: 91523
    phase: [1x1 struct]
    prs: [1x4743 double]
    prsdd: [1x4743 double]
    refyr: 2017
    sn: 109
    tau: [1x4756 double]
    taudd: [1x4756 double]
    tide: [1x4743 double]
    tmp: [1x4743 double]
    tmpdd: [1x4743 double]
    u: [1x4759 double]
    uvdd: [1x4759 double]
    v: [1x4759 double]
    tau_tide: [1x4756 double]
```

Figure 18: Example CP01.mat file contents after Step 2, initial detiding.

```

>> CP01.amplitude

ans =

   struct with fields:

    O1: 0.2635
    K1: 0.4190
    Q1: 0.0478
    P1: 0.1375
    M2: 0.8245
    K2: 0.0550
    N2: 0.1659
    S2: 0.2265

>> CP01.phase

ans =

   struct with fields:

    O1: 219.6060
    K1: 235.2030
    Q1: 212.8490
    P1: 234.1620
    M2: 225.0500
    K2: 251.4120
    N2: 200.7040
    S2: 250.2380

```

Figure 19: Example CP01.mat tidal amplitudes and phases.

```

Each hour contains 24 travel time measurements
** Initial Tau processing run on 27-Jan-2021 11:04:54
Tau processed using the QRT method
Tau bounds: 3.90815 to 3.93177
Output time interval is 0.0416667 days (60.0 min)

** Tau detiding run on 27-Jan-2021 11:05:38
Tau was detided using initial pressure constituents
Constituent amplitudes were scaled by 0.001321 for tau
*****
*****
PAROS PRESSURE SENSOR PROCESSING
** Initial pressure sensor processing run on 27-Jan-2021 11:04:54
Pressure bounds: 2961.461435 to 2965.986565
Temperature bounds: 1.851355 to 2.311645
Temperature equilibrium reached about 1.451607e+01 hours after launch
Output time interval is 0.0416667 days (60.0 min)

** Initial pressure detiding run on 27-Jan-2021 11:05:38
Data gaps less than 12 hours will be interpolated before detiding
Diurnal and semidurnal tides removed
Maximum Lags for Admittance: 3
*****

```

Figure 20: Snippet of processing log after Step 2, initial detiding, showing both the travel time and pressure entries.

4.2.3 Step 3: Despiking

Step 3 is done to identify and remove data spikes in the time series of all the variables. Processing is accomplished with two similar GUI windows. The first GUI is used to despike the travel time, and the Paroscientific temperature and pressure records. The second GUI is used to despike the DCS velocity components and temperature. Either a slope or Tukey method are used to replace the spikes with interpolated values. The slope method identifies outliers as data points that change more rapidly from the preceding value by more than the specified rate (tolerance). The Tukey method runs a median filter on the time series and identifies outliers as values that differ from the filtered record by the specified tolerance. Travel times can only be despiked using the slope method. Either method can be used to despike the other variables.

4.2.3.1 Travel time, pressure and temperature despiking

A GUI window appears with a toolbar (Figure 21) showing the tau time series before despiking in the upper panel. On the right side of the GUI you can select the averaging length (default, 5), the tolerance (default, 0.0008 seconds) and the despiking method (travel times can only be despiked with the slope method). Using the GUI defaults, press the button **Run Despike**.

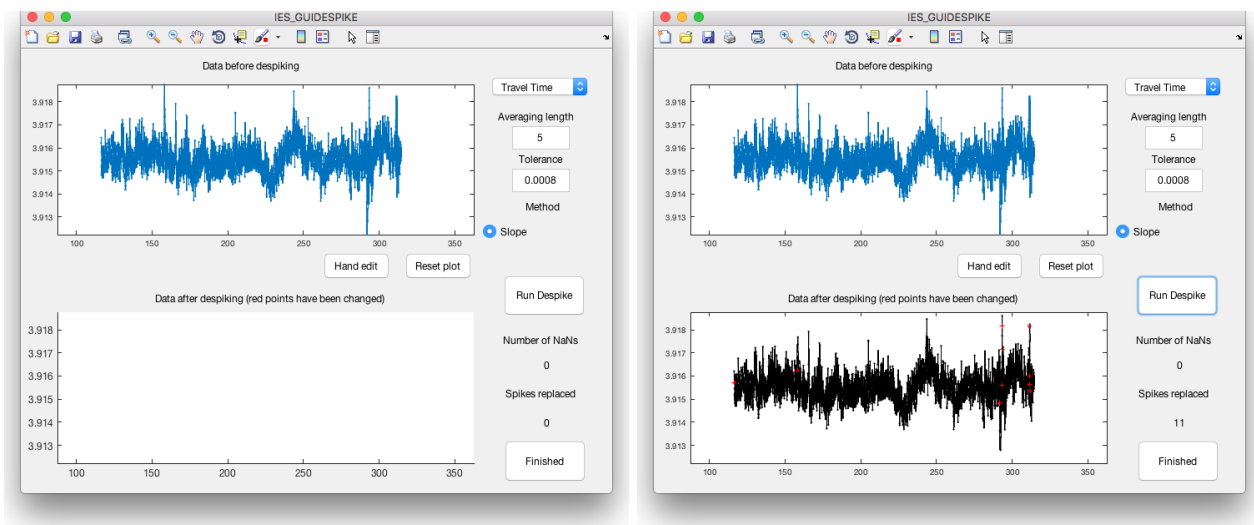


Figure 21: Example IES_GUIDESPIKE window showing data (left) before and (right) after running the despike program. Red plus markers flag the data points that have been replaced.

The GUI displays the number of spikes replaced and the number of NaNs in the record at the bottom right. The new despiked time series appears in the lower plot of the GUI (Figure 21). The replaced values are shown by red plus symbols. If you are not satisfied with the despiked tau time series you can change the averaging length or tolerance. Decreasing the tolerance removes more data spikes from the time series.

After running the **Despike** program, you may wish to manually edit the time series. To edit points manually, it is best to first zoom in using the MATLAB toolbar to show the area to be edited in more detail. Press the **Hand Edit** button between the two plots. The upper panel plot now shows the data before hand editing (Figure 22). A dialog box appears (Figure 23, left) directing the user how to proceed. **NOTE:** this dialog box only appears the first time you select the **Hand edit** button. Click **OK** to dismiss the dialog box. Click on the left end point first, and then the right end point.

The two points you selected with the mouse will then be circled in red in the upper panel. The new interpolated values will appear between the two circled red points in the lower plot (Figure 23, right). A dialog box will ask if you want to keep this interpolation. Click **Yes** if satisfied. Click **No** to reject the interpolation and restore to the values shown in the upper plot. Click **Replace with NaNs** if you want the interpolated values changed to NaNs.

If you wish to despiking the pressure and or temperature time series **DO NOT** click **Finished**. Rather, go up to the top right pull down menu and select pressure or temperature. The default averaging lengths and tolerances are 5 and 0.5 for pressure and 5 and 0.1 for temperature. Repeat the

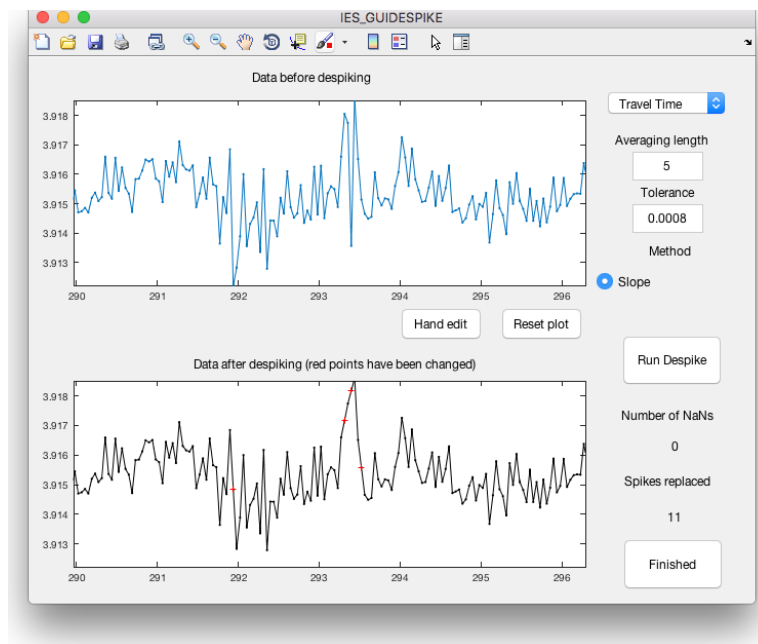


Figure 22: Example IES_GUIDESPIKE window showing expanded view for hand editing

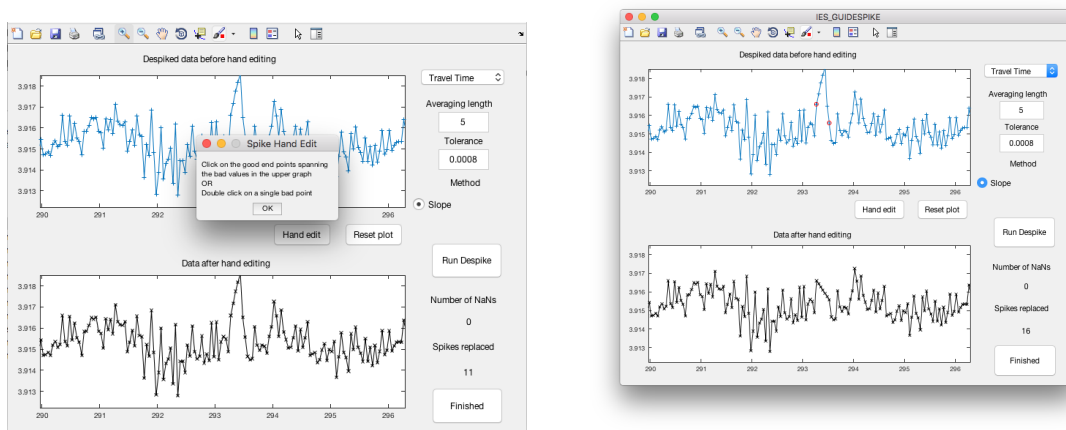


Figure 23: (left) Hand editing dialog box providing instructions. (right) Example IES_GUIDESPIKE window showing data before (top) and after hand editing (bottom).

steps as for tau. Click **Finished** when satisfied with *all three* time series.

4.2.3.2 DCS velocity and temperature despiking

Next the despiking GUI window for the DCS variables appears (Figure 24). The variables u, v, and current meter temperature can be despiked using either the slope or Tukey methods. The default

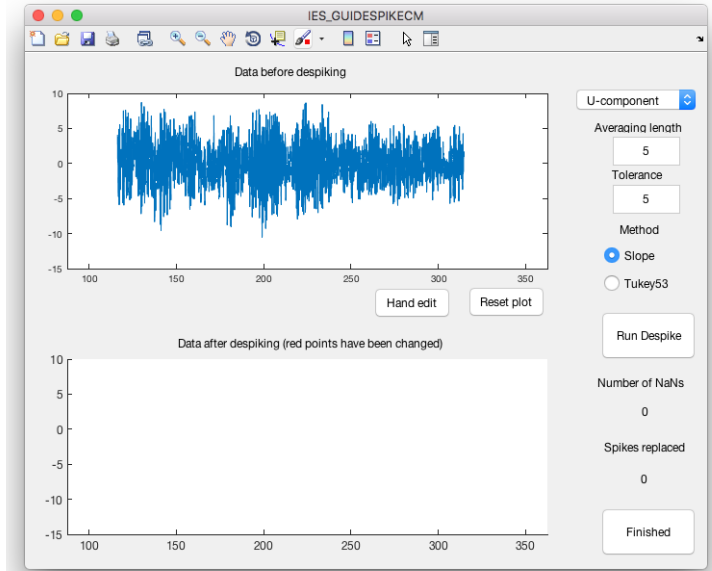


Figure 24: Example of IES_GUIDESPIKECM window showing data before despiking

averaging lengths and tolerances for u and v are 5 and 5. For the DCS temperature the defaults are 5 and 0.1, respectively. Change the values and rerun **Despiking** until you are satisfied with the results. Subsequently, additional changes can be made manually with **Hand Edit**. It is not uncommon to need to edit a few points at the beginning and end of the velocity records. Changing points at the beginning or end of the record of either component to NaNs results in both time series being shortened, since they share the same time base. When points of either velocity component are changed to NaNs, the corresponding points of the other component are also changed. However, if points of one component are changed to values other than NaNs, the other component is not affected. DCS temperature has a different time base than u and v and is also treated separately.

Click **Finished** in the IES_GUIDESPIKECM window when satisfied with *all three* time series. The

```
** Tau despiking run on 28-Jan-2021 09:11:12
Method = Slope
tolerance = 0.00080
averaging length = 5
Hand editing performed = Yes
Total NaNs in record = 0
Total records changed: 16
```

Figure 25: Example processing log after Step 3, despiking, shown for travel time record. Similar information is logged for the other variables.

CP01.mat file variables will be updated with the despiked data and the contents of the comments structure will display ‘Created after despiking’ for all the variables. A summary of the despiking processing is appended to each variable’s section of the log file. An example of the travel time section is shown in Figure 25.

Note: The velocities require two additional corrections for magnetic declination and local deep sound speed. Because they are site-specific, they are not performed here. Refer to Appendix C.1 for additional guidance.

4.2.4 Step 4: Dedrifting

This step fits and removes drift from the pressure record. Drifts are typically associated with variations in the properties of the pressure sensor crystal over long timescales or slight imperfections in the IES master clock. The pressure records are dedrifted by a linear-exponential function. The linear exponential function method was recommended by *Watts and Kontoyiannis* [1990] who have examined sensor drift and performance where the rate of drift decays with time and is best approximated by a function of the form

$$\text{Drift} = Ae^{Bt} + Ct + D$$

The processing code calculates the initial estimates for the coefficients A, B, C and D using one of two MATLAB m-files. If the MATLAB optimization toolbox is available, `fmincon.m` is used to calculate these parameters. If the optimization toolbox is not available, the program `fminsearchbnd.m` is used instead and a message appears in the GUI window.

The IES_GUIDEDRIFT window opens showing the data prior to fitting drift in the left plot. Below the plot the drift equation is shown with initial guesses for the coefficients A, B, C and D. The user can modify the OPTIMSET parameters on the right panel of the GUI (see Figure 26) and/or specify constraints on the coefficients A-D in the GUI. The OPTIMSET parameters are defined in MATLAB (type `help optimset` for more information) as follows:

- MaxFunEvals – Maximum number of function evaluations allowed (positive integer)
- MaxIter – Maximum number of iterations allowed (positive integer)
- TolFun – Termination tolerance on the function value (positive scalar)
- TolX – Termination tolerance on X (positive scalar)

Press the **Fit Drift** button. The drift curve (in red) is superimposed on the data in the “Prior to Fitting Drift” plot and the dedrifted data (data minus drift) is plotted in the “After Drift Removed” plot (Figure 26).

Note, the Initial Guesses have been replaced with Fitted Values (Figure 26) and at the bottom of the OPTIMSET box “Converged” should appear. In general, the OPTIMSET parameters should be appropriate. However, if the GUI seems to hang (or stops responding), decrease the value of MaxIter and increase TolX (in the OPTIMSET box). Use the toolbar to zoom in on the beginning of the pressure record in the left panel to examine the fit (Figure 27). You can constrain the values of the A, B, C and D coefficients to fall within certain limits. For each coefficient you can specify upper and lower limits. For example, if you believe that a linear fit better describes the entire record, change the

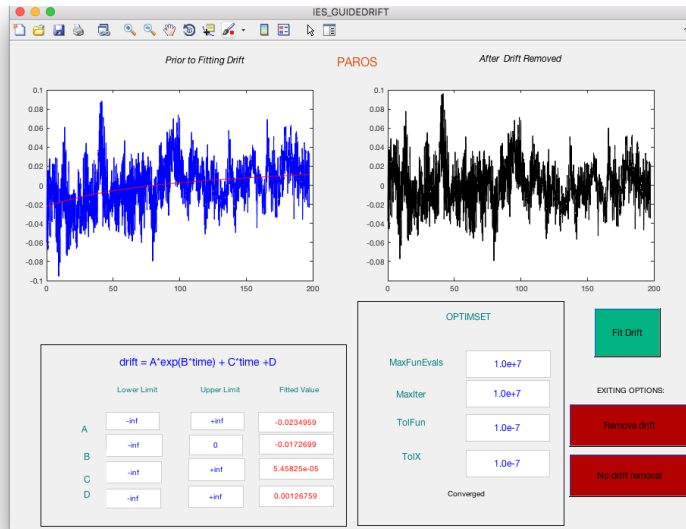


Figure 26: Example IES_GUIDEDRIFT window showing data prior to fitting drift (left) and after drift removal (right). The fitted drift is shown by the red line.

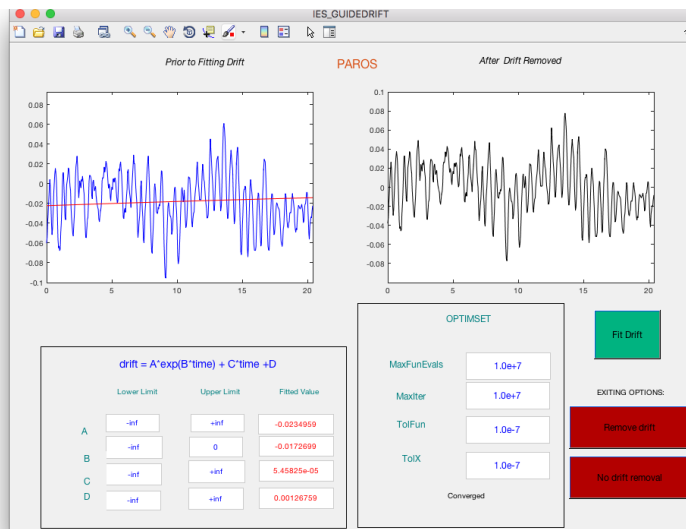


Figure 27: Same as Figure 26, except showing expanded views of data

upper and lower limits of A or B to zero in the IES_GUIDEDRIFT window and press **Fit Drift** again. The Fitted Values should then be zero.

There are two options for exiting: “Remove drift” or “No drift removal” (Figure 27). The “Remove drift” button should be pressed after the “Fit Drift” button when the user is satisfied with the fitted drift shown in the GUI window. Pressing the “No drift removal” button sets all the drift coefficients (A–D) to zero and allows the user to continue to the next step of processing without explicitly setting the upper and lower limits of the coefficients to zero in the GUI window. **NOTE:** if no drift is removed the second figure generated in Step 7, plotting, (e.g. Figure 30) will show a straight line drift with $y=0$ for the “Detided with drift” pressure time series.

Upon exiting, the CP01.mat file (Figure 28) contains the dedrifted pressure data (CP01.prs), the time series of the drift (CP01.drift) and the drift coefficients (CP01.driftcoef). The CP01 comments structure for pressure will read ‘Created after dedrifting’, ‘prs=prs-tide-drift’ and the values of the drift coefficients are appended to the log file (Figure 28).

```

>> CP01

CP01 =

  struct with fields:

    amplitude: [1x1 struct]
    cmtmp: [1x4758 double]
    cmtmpdd: [1x4758 double]
    comments: [1x1 struct]
    dcs: 339
    drift: [1x4743 double]
    driftcoef: [1x1 struct]
    paros: 91523
    phase: [1x1 struct]
    prs: [1x4743 double]
    prsave: 2.9639e+03
    prsdd: [1x4743 double]
    refr: 2017
    sn: 109
    tau: [1x4756 double]
    tau_tide: [1x4756 double]
    taudd: [1x4756 double]
    tide: [1x4743 double]
    tmp: [1x4743 double]
    tmpdd: [1x4743 double]
    u: [1x4759 double]
    uvdd: [1x4759 double]
    v: [1x4759 double]

    ** Pressure dedrifting run on 28-Jan-2021 11:02:45
    Mean pressure removed before dedrifting: 2963.90
    Drift equation: A*exp(B*time) + C*time + D
    Drift coefficients A = -2.349590e-02
    Drift coefficients B = -1.726988e-02
    Drift coefficients C = 5.458246e-05
    Drift coefficients D = 1.267590e-03

```

Figure 28: Example (left) CP01.mat file and (right) processing log contents after Step 4, dedrifting

Note: This step fits the best curve to the pressure data. However, some of the fitted drift might actually be real oceanic signals. See Appendix C.2 for additional guidance on how to distinguish them.

4.2.5 Step 5: Final Detiding

After the pressure record has been dedrifted, it is customary to detide the record again to improve the tide prediction. This step adds the tides removed by initial detiding (Step 2) to the dedrifted pressure record and then recalculates the tides. The travel time record is also updated by restoring the initial tides, recalculating the tides with the updated pressure coefficients, and removing the final scaled tidal signal.

The CP01.mat file will contain the dedrifted pressure data (CP01.prs) and the travel time (CP01.tau) after final detiding. The CP01 comment structure for pressure will read ‘Created after final detiding’,

‘prs=prs-tide-drift’. CP01.tide and CP01.tau_tide now contain the recalculated time series of the tide. The amplitudes (CP01.amplitude) and phases (CP01.phase) of the tidal constituents have been overwritten with the recalculated values. The amplitudes and phases after final detiding will likely show only slight differences from those calculated initially.

The processing log file will contain updated information for both pressure and travel time. For pressure, it states that the tidal coefficient amplitudes and phases were recalculated after drift removal. For travel time, it states that the final pressure coefficients were used.

4.2.6 Step 6: Low-pass filtering

Residual pressure, temperature, travel time, and DCS velocities are filtered (using a Butterworth filter from MATLAB’s Signal Processing Toolbox) and subsampled. The Butterworth filter design is known for its characteristic sharp monotonic transition between flat pass and stop bands with a minimum of coefficients. The Butterworth is also well known and used regularly in the oceanographic field. The filtering is done twice, once forward and once in the reverse direction. Three parameters control low-pass filtering:

- Butterworth Filter Order default = 4
- Cutoff Period (days) default = 3 days
- Output Time Interval (days) default = 0.5 days

Transients at the records’ ends are reduced by removing a linear ramp generated from the first and last points of the series before filtering. The same linear ramp is added after filtering. To avoid contamination by startup transients, data at each end of the filtered series are discarded. The amount of data discarded is one third of the cutoff period. After filtering, the time series is interpolated onto the output time interval.

Parameters used by this step are those listed in the GUI as LOWPASS FILTERING PARAMETERS: Butterworth Filter Order, Cutoff Period, Output Time Interval, and Gap Threshold. Gaps less than this threshold will be interpolated before filtering (default = 2 days). These parameters can be specified when initializing the GUI or reset by entering in new values manually. **NOTE:** If no values for the LOWPASS FILTERING PARAMETERS were ever specified a message will appear in the MATLAB window “No lowpass filtering will be done”.

After this step is executed, a file called CP01lp.mat is created that contains the low-pass filtered and subsampled records (Figure 29). The processing log file will contain information about the filtering for each variable (Figure 29).

4.2.7 Step 7: Plotting

This step plots the unfiltered and low-pass filtered data that have been processed. The data are loaded from the two mat-files created during processing (e.g. CP01.mat and CP01lp.mat) and plotted. All final variables stored in the mat-files are plotted. Three Figure windows will be generated. The Figure 1 window (Figure 30) for a PIES contains the hourly tau, pressure and temperature time series from the final CP01.mat file. If the instrument is a CPIES, the figure window also contains the u and v and DCS temperature time series. The Figure 2 window (Figure 30) is the same for both PIES and CPIES and provides a chronology of the pressure data processing, showing from top to bottom, the despiked pressure, the tides, the detided pressure with drift, and the dedrifted pressure time series. For

```

>> CP01lp
CP01lp =
  struct with fields:
    cmtmp: [1×393 double]
    cmtmpdd: [1×393 double]
    comments: [1×1 struct]
    dcs: 339
    paros: 91523
    prs: [1×392 double]
    prsave: 2.9639e+03
    prsdd: [1×392 double]
    refyr: 2017
    sn: 109
    tau: [1×393 double]
    taudd: [1×393 double]
    tmp: [1×392 double]
    tmpdd: [1×392 double]
    u: [1×393 double]
    uvdd: [1×393 double]
    v: [1×393 double]
** Tau low-pass filtering run on 04-Feb-2021 11:04:27
Butterworth lowpass filter: Order = 4, Cutoff period = 3 days
Filtered record has sampling interval of 0.5 days
Data gaps less than 1 days will be interpolated before filtering

```

Figure 29: Example (left) CP01lp.mat file and (right) processing log contents after Step 6, lowpass filtering. The log entry is shown for travel travel; other variables have similar log entries.

a PIES, the Figure 3 window contains the lowpass filtered, subsampled tau, pressure and temperature time series from the CP01lp.mat file. For a CPIES, u, v and DCS temperature are also plotted in the Figure 3 window (Figure 31).

It is recommended that you save (and later print) the three figures by pressing the button **Save plots** in the center bottom of the GUI window. Following our example, the saved plot filenames will have the syntax CP01_finalfigX.PFF for 109_1, with PFF either .pdf, .eps, or .png.

4.3 Additional Information

4.3.1 Time base

4.3.1.1 Sampling schedule effects on time base.

The IES wakes up from low-power sleep every 10 minutes and performs all data collection tasks that are scheduled. During IES mission setup, three independent data acquisition schedules are configured on multiples of this 10 minute schedule: travel time measurements, Paros pressure and temperature measurements, and DCS current and temperature measurements (if the optional DCS is installed). The IES makes these measurements in the following sequential order, travel time, Paros pressure and temperature, and DCS currents and temperature. The resulting time base associated with each measurement depends on the acquisition schedule of the other measurements. To obtain consistent time bases for all measurements without the need for interpolation, care should be taken when choosing data acquisition schedules. For example, the travel time schedule should be the same as or more frequent than the Paros and DCS measurements. Likewise, the Paros measurements should be the same as or more frequent than the DCS measurements.

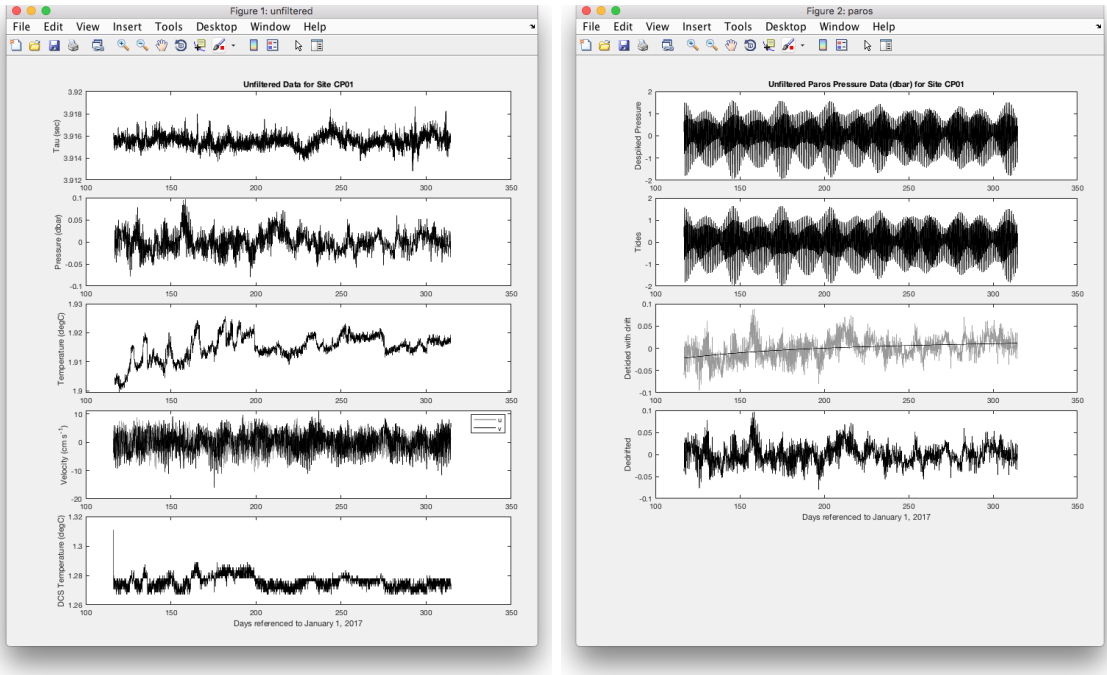


Figure 30: Final plots of unfiltered data. (left) Example Figure 1 window showing the unfiltered travel time, pressure, temperature, and velocity records. The output interval is hourly for travel time, and set by the user for the Paros and DCS variables. (right) Example Figure 2 showing the despiked pressure, tides, detided pressure with drift and dedrifted pressure time series.

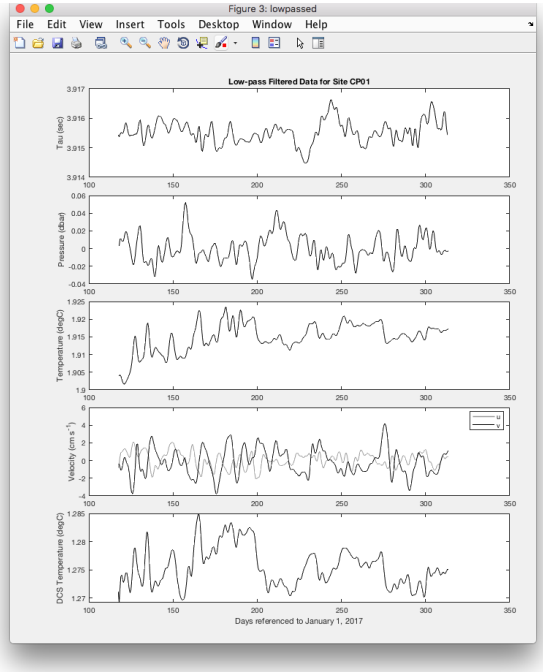


Figure 31: Example Figure 3 window showing the low-pass filtered and subsampled tau, Paros pressure and temperature, DCS u, v and temperature records for a CPIES.

Two examples illustrate IES data acquisition schedules:

Example 1: Consistent time base acquisition schedules (Figure 32)

- travel time measurement schedule: 4 pings every 10 minutes
- pressure & temperature schedule: P&T measurement every 10 minutes
- DCS schedule: DCS measurement every 10 minutes

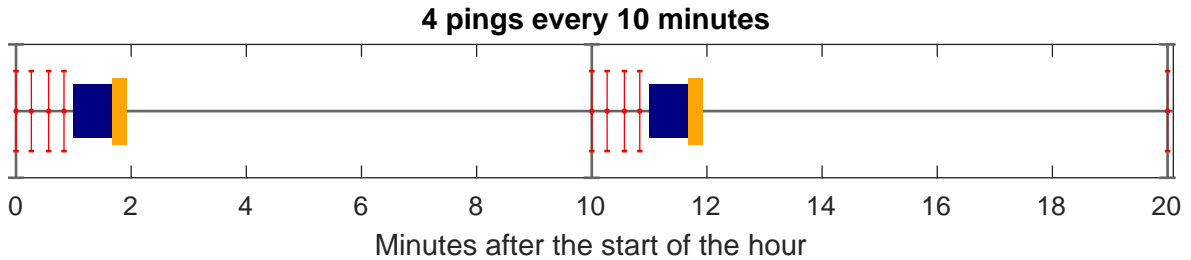


Figure 32: Time line illustrating the dependence of the Paros and DCS sampling times on the travel time sampling scheme. All variables are set up to sample every 10 minutes. The first 20 minutes of a one hour sampling period are illustrated. During each 10-minute sampling period, 4 travel time pings (red bars) occur first, followed by Paros sampling during a 40-second window (blue box), and subsequently, by DCS model 4930 sampling during a ~15-second window (orange box).

On this schedule, travel time will be sampled precisely at ten minute increments of the real time clock. Next, Paros pressure and temperature are measured at the top of the minute following the travel time burst of 4 pings (i.e., at the one minute mark of each 10 minute increment). The DCS measurement immediately follows the Paros measurements. These data acquisition schedules result in time bases that are consistent throughout the hour.

Example 2: Inconsistent time base acquisition schedules (Figure 33)

- travel time measurement schedule: 24 pings every 60 minutes
- pressure & temperature schedule: P&T measurement every 10 minutes
- DCS schedule: DCS measurement every 10 minutes

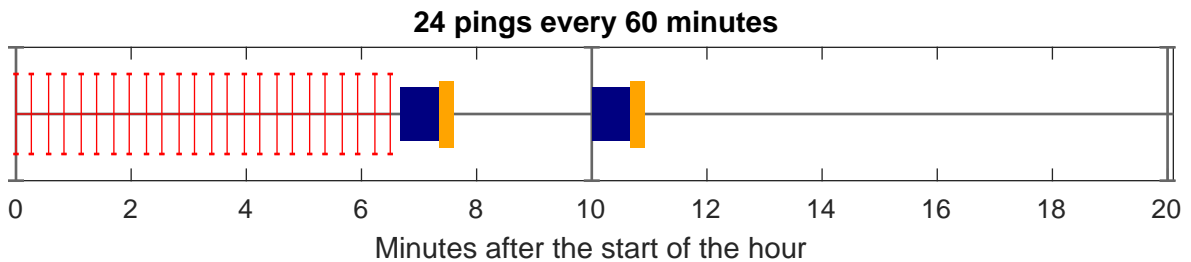


Figure 33: Same as Figure 32, except travel time is set to sample 24 pings every 60 minutes.

On this schedule, all 24 pings will be sampled at the top of each hour. At the top of the next minute following the 24 ping burst, the Paros pressure and temperature are measured. This moves the time associated with the first Paros pressure and temperature measurement much later for the first 10 minute

increment. Subsequent pressure and temperature measurements within each hour occur without delay at the 10 minute marks. The DCS measurement immediately follows the Paros measurements. These data acquisition schedules result in unevenly measured data. Low-pass filtering, which is routinely done for IES data, requires input data to be on uniform time intervals. Therefore, the processing code checks data acquisition schedules to determine if pressure and currents were measured more frequently than travel time. If so, pressure and current measurements are interpolated onto uniform time intervals.

4.3.1.2 Clock drift

The model 6 CPIES and PIES make adjustments to the real time clock oscillator frequency throughout the deployment to minimize clock drift. The number of clock corrections during the mission is recorded in the system log file (IESXXX.LOG) in the REL ('Release') record. See the IES User's Manual Section 4.3.1 for more information. Under ideal conditions these adjustments would maintain a clock accuracy of 10 seconds/year. However, clock drift rates of 20 seconds/year are more commonly observed, and this processing code makes adjustments to the timing of the variables to account for the observed drift. The primary reason for making these adjustments is to obtain more accurate estimates of the phases of the tides.

Clock drift is determined at sea when the instrument is recovered. The recommended method is to determine the difference between the IES's internal clock and actual UTC (δt) before the instrument is turned off. Alternatively, the offset can be determined by observing the timing of the first ping of a sample burst before the IES is recovered. However δt calculated this way will include transmission delays of a few seconds.

If the clock drift rate is less than 1 minute/year, its affect on the tidal phase is deemed small and a simple adjustment to the time base is made. The start time of the first on-bottom ping is adjusted by $\delta t/2$, and the time base is recalculated using the original sampling interval. Note the sign of the adjustment depends on whether the IES clock is early or late.

If the clock drift rate is greater than 1 minute/year, its affect on the phase is deemed more problematic. For this case, the original start time of the first on-bottom ping is used and the sampling interval is changed. The adjustment to the sampling interval is based on δt and the total elapsed on-bottom time. For example, if after these corrections the time base uncertainty is ± 0.5 minute during the year, then the semidiurnal tide phase uncertainty is $\pm(0.5 \text{ min})/(12 \text{ hr} \times 60 \text{ min/hr}) \times (360 \text{ deg/cycle}) = \pm 0.25$ degrees. For the model 6 instruments, this type of clock drift correction is rarely needed.

4.3.1.3 Time offset

Time offsets are typically introduced when there is concern that there may be interference with the travel time measurements of two closely spaced neighboring instruments. In this case, the IES real time clock is purposely offset from UTC, and the offsets are always less than the 10-minute sampling interval. The time base correction is made by subtracting this constant offset from the internally-recorded times.

Additionally, this parameter can be used to correct for IES clock errors introduced when the instrument was manually set up. For example, the internal clock may have been set to local time rather than UTC. The correction must be specified in minutes with the appropriate sign.

4.3.1.4 Timing of hourly-averaged variables

The midpoint times associated with hourly-averaged variables depend on their sampling intervals; they are not set to be near 30 minutes after the hour. For a 10-minute sampling interval with samples

occurring between 0 and 50 minutes after the hour, the midpoint times are close to 25 minutes after the hour. For example, for a travel time measurement schedule of 4 pings every 10 minutes, the centered transmit time is $(16 + 18 + 16 + \text{mean tau (4.67 sec for an average depth of 3500 m)})/2 = 27.33$ sec after the start of the burst. The time assigned by the processing program for the hourly average of the 6 bursts at 0, 10, 20, 30, 40 and 50 min is HH:25:27.33 (HH:MM:SS). For a P&T measurement every 30 minutes, each centered between 60 -100s = 80s after the start of the burst, the hourly average is at 15 min + 80 sec = HH:16:20. For a 60-minute sampling interval, the midpoint times are close to the top of the hour. For current data, the processing code uses the time stamps in the CXXX_#.DAT file to calculate the hourly averages of the measurements within the hour. Note that the times listed in the examples will also be adjusted for any clock drift and time offsets.

4.3.2 No travel time data

If the travel time file (TXXX_#.mat) is missing, the remaining data can still be processed with IES_GUIDRIVER. During Step 1, pressure and temperature limit selection will proceed as described in Chapter 4 starting at Figure 11. Without travel time data, the program will read the instrument log file (IESXXX.LOG) to find the travel time measurement interval. Time bases for the other measurements will then be calculated using this travel time measurement interval. The user will be prompted for the travel time measurement interval if it could not be read from the instrument log file. The processing log will contain entries regarding the assumptions made for the time base. Subsequent processing will proceed as described earlier in Chapter 4.

4.3.3 Processing individual variables

While IESpkg 4 requires that the seven processing steps be performed sequentially, it does allow flexibility in processing the variables. The variables may be processed as the whole group, smaller subsets, or individually. However because the time bases of the pressure and travel time records are linked, it is recommended that travel time be executed at **Step 1** either before or at the same time as pressure.

Additionally, variables can be reprocessed at any step without affecting the other processed variables. After initializing the GUI, select the desired variables to process using the radio buttons and the processing steps to execute with the pull down menus. The mat-files, workspace and processing log will update only the variables and steps selected when the GUI is run. Warning messages about the time base will appear in the MATLAB command window and the processing log if pressures are processed before travel time.

4.3.4 Delete button

If you need to reprocess all the variables from the beginning, or if something goes awry, the existing workspace and mat-files should be deleted. Additionally, it is desirable to delete all figures and logs. While these can be deleted manually, it is more convenient to do so with the GUI. Pressing the **Delete files** button will eliminate all files created by IES_GUIDRIVER for the instrument. A pop-up window will ask the user to confirm before the files are deleted. The names of the deleted files will be listed in the MATLAB command window (e.g. Figure 34).

5 Acknowledgments

The MATLAB IES processing codes described in this manual have evolved from earlier versions created by Douglas Mitchell and Charles James. Mark Wimbush wrote the Respo.for program used for tidal analysis and helped implement it for our application. We gratefully acknowledge these pro-

```
Command Window
>> IES_GUIDRIVER
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Running: IESpkg 4.05
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Deleting files for PIES109_1 designated CP01 in directory /users/karen/manual/processed/
Deleted: /users/karen/manual/processed/PIES109_1_processing.log
Deleted: /users/karen/manual/processed/CP01.mat
Deleted: /users/karen/manual/processed/CP01lp.mat
Deleted: /users/karen/manual/processed/wrkspc/Workspace109_1.mat
Deleted: /users/karen/manual/processed/wrkspc/CP01.tidelog
Deleted: /users/karen/manual/processed/wrkspc/CP01_tau.tidelog
>>
```

Figure 34: Files deleted are listed in the MATLAB command window.

gramming efforts.

This research program has been sponsored by the National Science Foundation under EAGER award 1728060 and grant OCE1358470, and by the National Academies of Sciences, Engineering, and Medicine award 2000009943.

References

- Donohue, K. A., D. R. Watts, K. Tracey, A. D. Greene, and M. Kennelly (2009), Mapping circulation in the Kuroshio Extension with an array of Current and Pressure recording Inverted Echo Sounders, *J. Atmos. Oceanic Technol.*, 27, 507–527, doi:10.1175/2009JTECHO686.1.
- Fields, E., K. Tracey, and D. R. Watts (1991), Inverted echo sounder data processing report, *GSO Technical Report 91-3*, University of Rhode Island.
- Kennelly, M. A., K. L. Tracey, K. A. Donohue, and D. R. Watts (2007), Inverted echo sounder data processing manual, *GSO Technical Report 2007-02*, Graduate School of Oceanography, University of Rhode Island, Narragansett, RI.
- Munk, W. H., and D. E. Cartwright (1966), Tidal spectroscopy and prediction, *Phil. Trans. Roy. Soc. London*, 259, 533–581.
- Watts, D. R., and H. Kontoyiannis (1990), Deep-ocean bottom pressure measurement: drift removal and performance, *J. Atmos. Oceanic Technol.*, 7, 296–306.

A IES_GUIDAT2MAT additional figures

A.1 Content of .mat files created

```
>> load('T109_1.mat')
>> whos
Name              Size              Bytes  Class  Attributes

Tau              114264x1          914112 double
theadr           5x10              100    char
tieshrs         114264x1          914112 double
```

Figure A.1: Example TXXX_#.mat file contents

```
>> load('P109_1.mat')
>> whos
Name              Size              Bytes  Class  Attributes

P                28566x1           228528 double
Tmp             28566x1           228528 double
pheader         23x10             460    char
pieshrs         28566x1           228528 double
```

Figure A.2: Example PXXX_#.mat file contents

```
>> load('C109_1.mat')
>> whos
Name              Size              Bytes  Class

cieshrs          28554x1           228432 double
compass_direction 28554x1           228432 double
dcs              1x1               8      double
direction        28554x1           228432 double
ping_count       28554x1           228432 double
signal_strength  28554x1           228432 double
speed            28554x1           228432 double
temperature      28554x1           228432 double
tilt_east        28554x1           228432 double
tilt_north       28554x1           228432 double
u                28554x1           228432 double
v                28554x1           228432 double
```

Figure A.3: Example CXXX_#.mat file contents

```
>> load('E109_1.mat')
>> whos
Name              Size              Bytes  Class  Attributes

Efrq            198x13            20592 double
eheader         4x10              80    char
eieshrs         198x1             1584 double
```

Figure A.4: Example EXXX_#.mat file contents

```

>> load('K109_1.mat')
>> whos
  Name           Size           Bytes  Class  Attributes
  ----           -
  kP             28566x1          228528 double
  kTmp          28566x1          228528 double
  kheader        23x10             460    char
  kieshrs       28566x1          228528 double

```

Figure A.5: Example KXXX_#.mat file contents

A.2 Example plots generated by IES_GUIDAT2MAT for PIES and CPIES

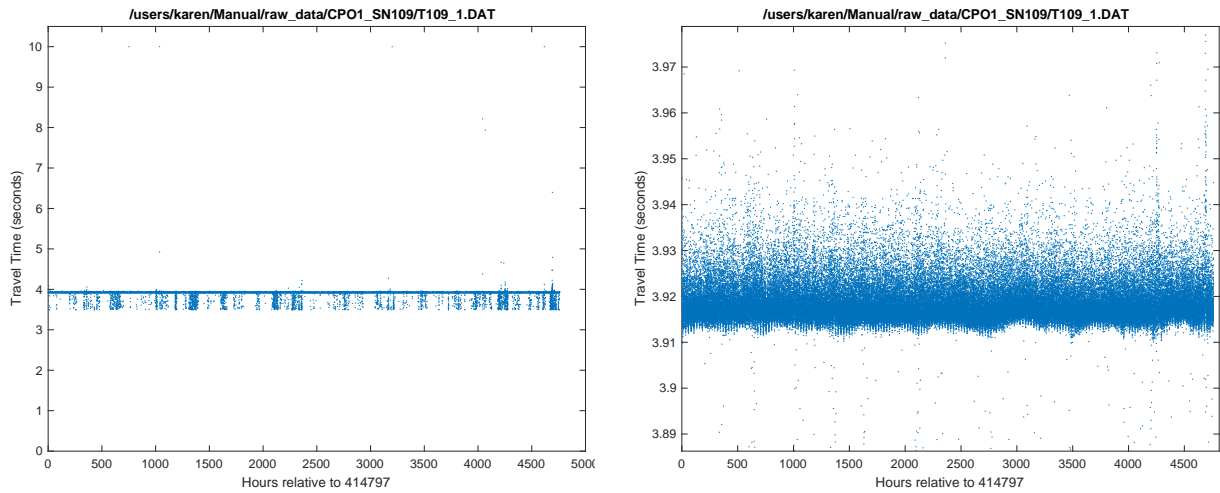


Figure A.6: IES_GUIDAT2MAT Travel Time (seconds) with the default (left) and reduced (right) y-axis limits.

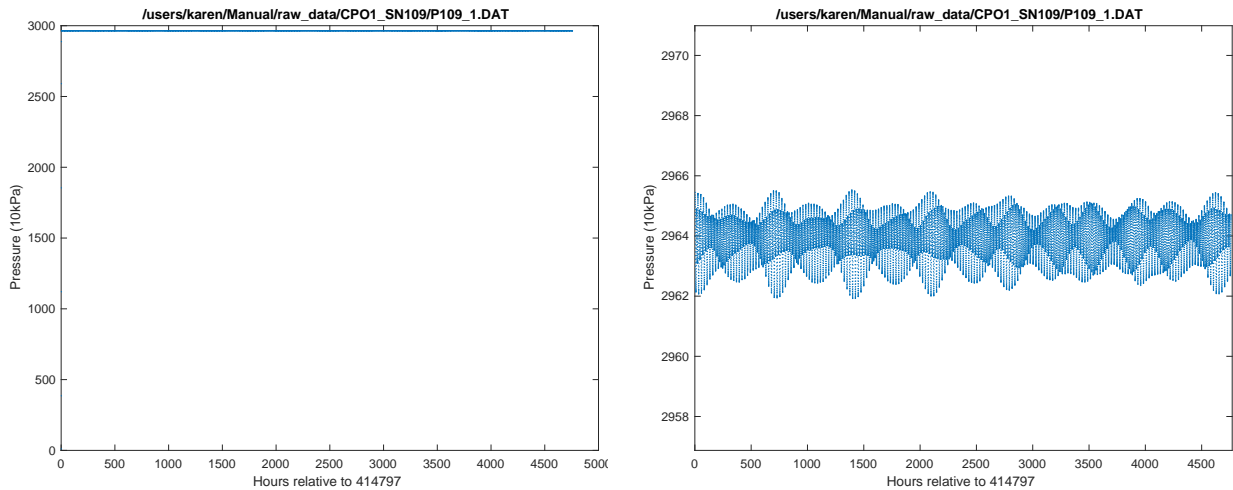


Figure A.7: IES_GUIDAT2MAT Pressure (10kPa) with the default (left) and reduced (right) y-axis limits.

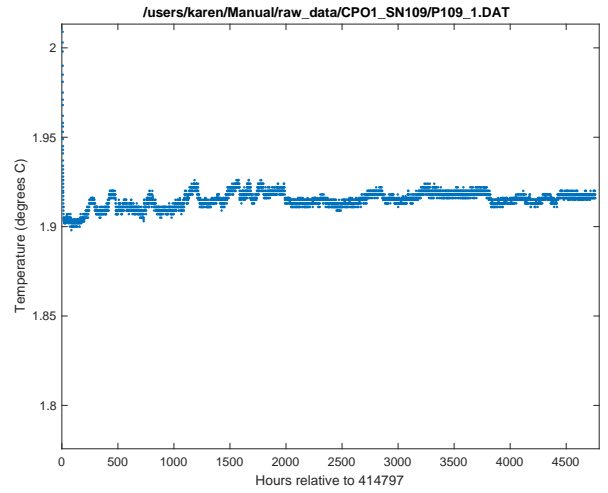
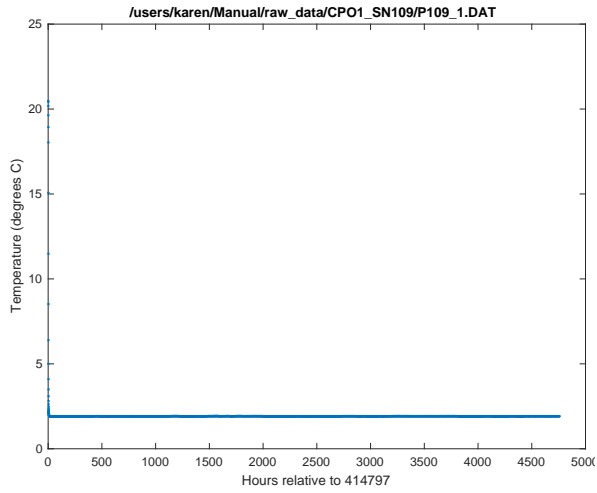


Figure A.8: IES_GUIDAT2MAT Temperature (degrees C) with the default (left) and reduced (right) y-axis limits.

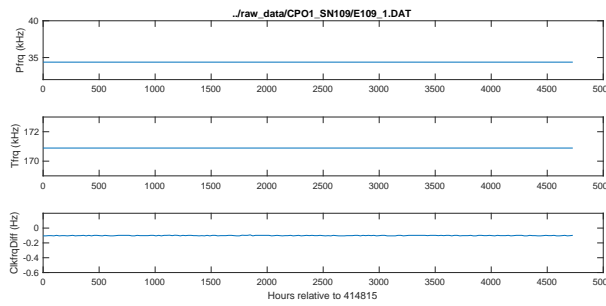
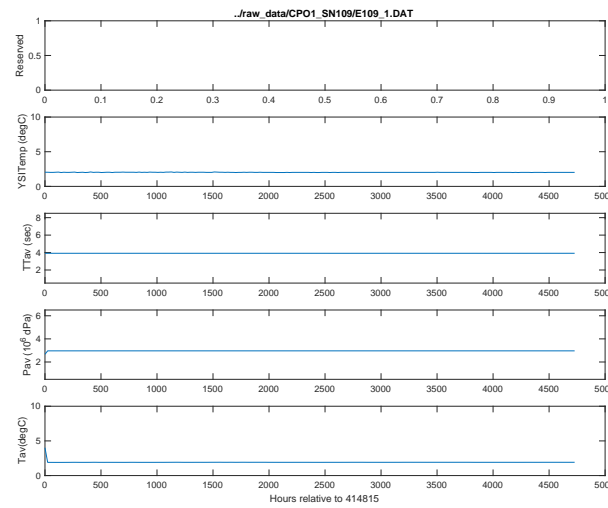
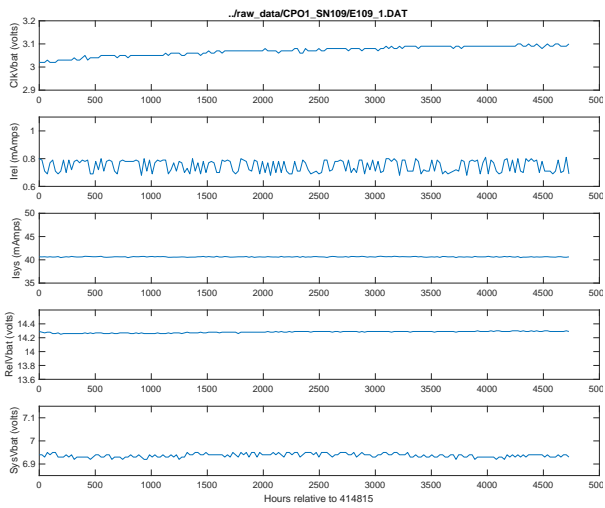


Figure A.9: IES_GUIDAT2MAT three sets of Engineering measurements

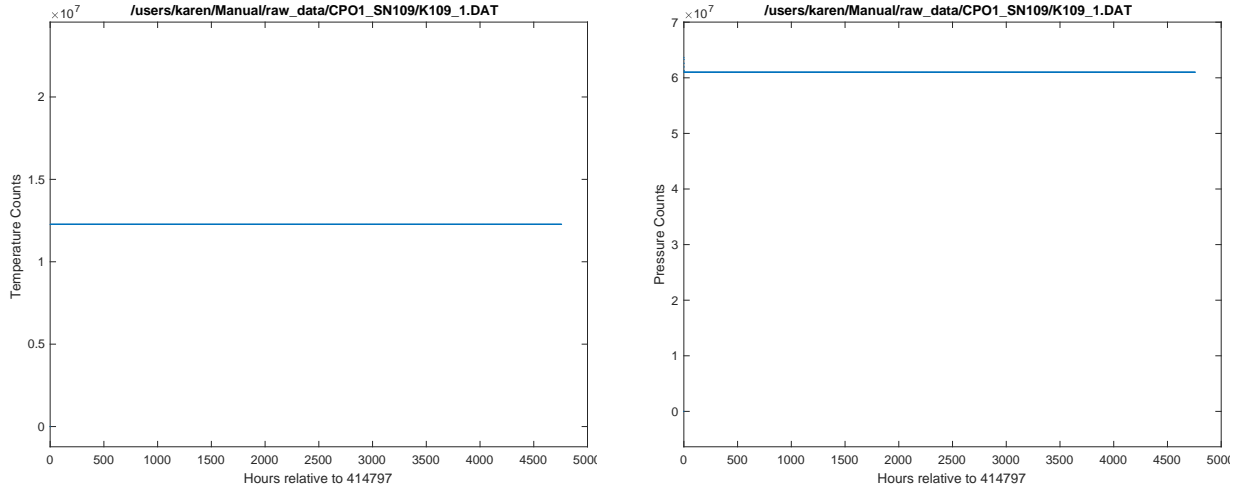


Figure A.10: IES_GUIDAT2MAT Temperature (left) and Pressure (right) counts with the default y-axis limits.

A.3 Examples of plots generated by IES_GUIDAT2MAT for currents (CPIES only)

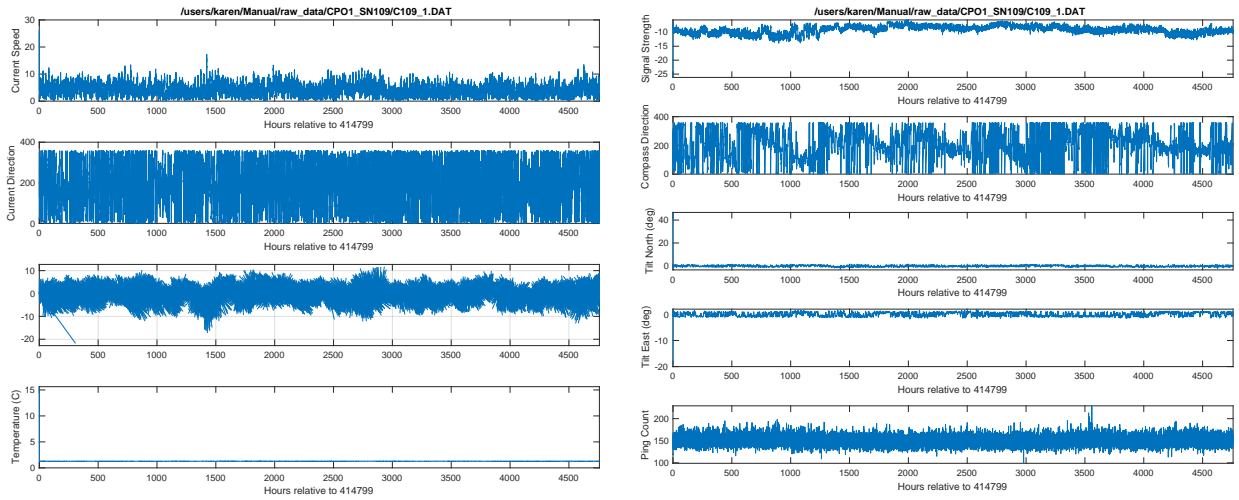


Figure A.11: IES_GUIDAT2MAT two sets of DCS measurements for CPIES.

B Response Analysis of Tides

B.1 Theory

A simple filter may be expressed as,

$$y(t) = \sum_k l(\tau_k)x(t - \tau_k) \quad (1)$$

where y is the predicted tide, x is the input function, and l is the response of the ocean to a unit of impulse of x at time zero.

The tidal prediction illustrated by Equation 1 depends only on the input's temporal variation at that particular location. As a refinement, the forcing at other locations may be included in the prediction:

$$y(t) = \sum_l \sum_k l_i(\tau_k) x_i(t - \tau_k) \quad (2)$$

where 'i' represents forcing at neighboring locations that might influence sea level at the site of interest. Response analysis systematically includes spatial dependence by expanding x in surface spherical harmonics. The predicted tide is expressed as a filter acting on the complex-valued, time-varying amplitudes of the spherical harmonic functions representing the equilibrium-tidal potential.

$$y(t) = Re \sum_k \sum_{n=2}^{\infty} \sum_{m=0}^n w_n^m(\tau_k) C_n^m(t - \tau_k) \quad (3)$$

The indices n and m are the degree and order of the surface spherical harmonic functions. $C_n^m(t)$, which replaces x , is the set of time-varying amplitudes of the corresponding spherical functions. The w_n^m , which replace l , are the complex weights associated with $C_n^m(t)$. Using the data to be detided ($\eta(t_j)$), the weights (w_n^m) are found by solving the over determined set of equations such that the difference between the data and the predicted tide ($\eta(t_j) - y(t_j)$) is minimized in a least squares-sense.

The equilibrium-tidal potential for a mass M , whose center of mass is at distance from the point of observation, is:

$$\frac{V(t)}{g} = \frac{GM}{g\rho} \quad (4)$$

Where $V(t)$ is the gravitational potential due to mass M , G is the gravitational constant, and g is local gravity. Typically, ninety-nine percent of the gravitational tidal variance can be explained with the equilibrium-tidal potential (due to the masses of the moon and sun) represented by just the C_2^1 and C_2^2 { $n=2, m=1, 2$ } amplitude functions.

The spherical harmonics corresponding to C_2^1 and C_2^2 are illustrated as viewed down the axis of rotation in Figure B.12. The plus sign represents bulging relative to the geoid, the minus sign flattening. From this illustration, it is apparent that C_2^1 and C_2^2 are associated with the diurnal and semi-diurnal species of the harmonic analysis. RESPO is set up to use these functions (C_2^1 and C_2^2) as input at time-lags of $\tau_k = k * 48hr$, $k = -1, 0, 1$. Thus, with the default case of $n=2$, Equation 3 is then truncated to:

$$y(t) = Re \sum_{k=-1}^1 \sum_{m=1}^2 w_2^m(k * 48) C_2^m(t - k * 48) \quad (5)$$

The 6 weights, $w_2^m(k * 48hr)$, $m = 1, 2, k = -1, 0, 1$ are found from the overdetermined set of equations.

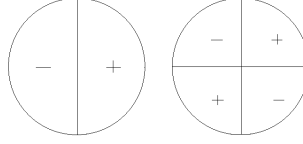


Figure B.12: Spherical harmonics corresponding to tidal amplitude functions, C_2^1 (left), C_2^2 (right)

$$\eta(t_j) = \text{Re}(C(t_j) * W) \quad (6)$$

Where

$$C(t_j) = [C_2^1(t_j + 48)C_2^1(t_j)C_2^1(t_j - 48)C_2^2(t_j + 48)C_2^2(t_j)C_2^2(t_j - 48)] \quad (7)$$

And

$$W = \begin{bmatrix} w_2^1(48) \\ w_2^1(0) \\ w_2^1(-48) \\ w_2^2(48) \\ w_2^2(0) \\ w_2^2(-48) \end{bmatrix} \quad (8)$$

These weights are applied to C_2^m to generate the predicted tide. This is subtracted from the original data $\eta(t_j)$ to give the residual tide.

B.2 Tidal Prediction Options

Currently, RESPO is set up to calculate the total gravitational potential (LGAMMA=3) for the lunar and solar tidal constituents.

Di, Semi-Di, 3 lags (Diurnal, Semi-Diurnal, 3 lags): BASECTRL_DS3.txt

```
2,
3, 1, 2,
3, 2, 2,
2, 1, 2,
3, -48, 0, 48,
2, 0, 2, 0, 3,
1, 1.0, -1, 1, 1, 1, 0, -1, 1, -1, 1,
'tidedata.txt',
'(f12.6)',
'tidedataout.txt',
'(3f15.6)',
```

Di, Semi-Di, 5 lags (Diurnal, Semi-Diurnal, 5 lags): BASECTRL_DS5.txt

```

2,
3, 1, 2,
3, 2, 2,
2, 1, 2,
5, -96, -48, 0, 48, 96
2, 0, 2, 0, 5,
1, 1.0, -1, 1, 1, 1, 0, -1, 1, -1, 1,
'tidedata.txt',
'(f12.6)',
'tidedataout.txt',
'(3f15.6)',

```

B.3 Compiling Respo.for

Here are various tidbits of information about compiling RESPO.for to create an executable compatible with your operating system.

1. Download a gfortran compiler from the web, if you don't have it on your system. The website <https://gcc.gnu.org/wiki/GFortran> provides links to unofficial binaries of gfortran and provides detailed instructions.

- Scroll down the webpage to the section 'Quick Links' and click on the 'Binaries' link under 'Downloads'.
- On the website <https://gcc.gnu.org/wiki/GFortranBinaries>, scroll to the appropriate operating system and choose one of the options. For Mac OS, we recommend using the package provided by Homebrew (<https://brew.sh>). The website provides detailed instructions for installing gcc and gfortran. After installation on Mac OS, the binary will be located in `/usr/local/gfortran`.
- Mac OS users will also need Xcode from the Apple Developer Tools. It can be downloaded from <https://developer.apple.com/develop/>. To install Xcode on Mac OS, in a Terminal window type the command:

```
xcode -select --install
```

2. Create the executable

- Simplest way to compile Respo.for to create an executable, in a Terminal window or a command line type

```
gfortran Respo.for -o Respo.exe
```

This will produce a long list of warning messages, but if Respo.exe is created then most likely it is a valid executable.

- Using the legacy option will eliminate the Hollerith warning messages.

```
gfortran -std=legacy Respo.for -o Respo.exe
```

3. After performing a system upgrade on a Mac, it may be necessary to uninstall and reinstall gcc and gfortran:

- uninstall gcc and gfortran

```

NUMGMN
LGAMMA, MORDER, NDEGRE
LGAMMA, MORDER, NDEGRE
KP (JP(K), K=1,KP)
KH (HH(K), K=1,KH)
NPHGP1 (NP(K), K=1,nphgp1) (NH(K), K=1,NPHGP1)
IADJD, CALIB, IPRIND, IDWTS, IDADM,IDRSP, KOMPLX, IPRINP, IRESID,
IPRINR, IOUT

NUMGMN = Number of gamma, M, N triplets
LGAMMA = 1 for moon's gravitational potential
         = 2 for sun's gravitational potential
         = 3 for total gravitational potential
         = 4 for sun's radiational potential
MORDER = value of M in ith gamma, M, N triplet (order of spherical
harmonic)
NDEGRE = value of N in ith gamma, M, N triplet (degree of spherical
harmonic)
KP = Dimension of array JP
JP = Array of P Values (X component numbers) to be used in forming the
     prediction weights
KH= Dimension of array HH
HH = Array of H values (X time leads) to be used in forming the
     prediction values
NPHGP1 = Dimension of array NP and of array NH
         NP(I)    NP(1) = 0
         NP(I .GT. 1) is the number of the last term in the
         (I-1)th group of array jp
         NH(I)    NH(1) = 0
         NH(I .GT. 1) is the number of the last term in the
         (I-1)th group of array HH
IADJD (if EQ 1) subtract mean from series and multiply by calib
CALIB calibration
IPRIND (if LT 0) print only partial list of normalized input data
IDWTS (if LT 0) weights are not to be computed, they are given
IDADM (if EQ 1) compute admittance
IDRSP (if EQ 1) create predicted tide series
KOMPLX (if EQ 0) only real part of predicted series is to be retained
IPRINP (if LT 0) print only partial list of predicted tide
IRESID (if EQ 1) compute residual series
IPRINR (If LT 0) print only partial list of residual series
IOUT (if LT 0) do not write output time-series file

```

Figure B.13: Description of contents of Tidal Prediction Option Files

- Get Homebrew if it isn't installed on your system
- Type these 2 command in a terminal window:


```
brew install gcc
brew reinstall gfortran
```

The following suggestions worked on older versions of Mac OS or Linux, but have not worked with newer Mac OS X systems (High Sierra and above):

1. If the executable is created, but Matlab gives an error about a library version gfortran 1.4 not found, use either

```
gfortran -static Respo.for -o Respo.exe
```

or

```
gfortran -static -libgfortran Respo.for -o Respo.exe
```

2. For some operating systems (Ubuntu Linux) and some versions of Matlab (2013a) including the static doesn't complete resolve the problem. Success was achieved by also adding the following lines of code to the IES_DETIDE.m and IES_GENERATETIDE.m files:

```
setenv('GFORTRAN_STDIN_UNIT', '5')
setenv('GFORTRAN_STDOUT_UNIT', '6')
setenv('GFORTRAN_STDERR_UNIT', '0')
eval(['!' tideprog ' > ' tidelogname])
setenv('GFORTRAN_STDIN_UNIT', '-1')
setenv('GFORTRAN_STDOUT_UNIT', '-1')
setenv('GFORTRAN_STDERR_UNIT', '-1')
```

3. Note that the static and legacy options can be used together:

```
gfortran -static -std=legacy Respo.for -o Respo.exe
```

C Recommended additional processing

C.1 Additional velocity processing recommended

After the velocity processing with IES_GUIDRIVER is complete, it is recommended that two data corrections be applied. The first correction is for magnetic declination, and the second for local mean deep sound speed. The IESpkg codes do not apply these corrections as they are site specific. It is left to the end user to apply their own corrections.

For magnetic declination correction to the u, v components, some users prefer to apply a constant correction if the time variation is small. Others apply a time-dependent magnetic declination.

The u, v components should also be corrected for local mean deep sound speed. The DCS uses a fixed sound speed of 1500 m s⁻¹ when it makes its measurements. The bottom sound speed at CPIES sites usually differs from the default value, so the measured speeds must be scaled:

$$\text{current speed}_{\text{corrected}} = \text{current speed}_{\text{measured}} \times \text{scale factor} \quad (9)$$

where

$$\text{current speed}_{\text{measured}} = \sqrt{u^2 + v^2} \quad (10)$$

and scale factor = $\text{sound speed}_{\text{bottom}} \text{ m s}^{-1} / 1500 \text{ m s}^{-1}$. A mean sound speed profile as a function of depth can be calculated from a set of historical CTDs near your region. The mean profile can be used as a look up table to get a nominal sound speed at the depth of the CPIES. Typically, for application of these corrections, the u , v components are converted to current speed and direction, the corrections are made and then the corrected current speed and directions are converted back to u , v components.

C.2 Additional dedrifting recommended

Real ocean signals may appear like instrumental drift. For preliminary drift removal allow MATLAB to select the drift coefficients. Subsequently, comparisons with coincident current measurements should be made to determine if the preliminary drift curve contains any ocean signal. If so, it will be necessary to dedrift the pressure record again. *Donohue et al.* [2009] describes a technique for dedrifting/leveling pressure time series with current meter data. This paper consolidates and documents the many advances that have taken place over the past 30 years to interpret IES measurements. Although many of the details in *Donohue et al.* [2009] are specific to the KESS array, the techniques described are pertinent for analyzing inverted echo sounder, pressure and current measurements in other regions.

D IES_GUICOUNTS2MAT

IES_GUICOUNTS2MAT is the GUI interface to easily input parameters to IES_COUNTS2MAT.m. IES_COUNTS2MAT is provided to calculate pressure in dbar and temperature in degrees C from the recorded counts using the conversion coefficients supplied by Paroscientific, Inc. and Bliley Technologies, Inc. Calibration sheets listing the calibration coefficients for the Bliley reference oscillator and the Paroscientific pressure sensor are supplied by URI/GSO in the IES Documentation Folder provided with each IES. IES_GUICOUNTS2MAT will not normally be needed in routine processing since IESs are configured to internally convert pressure and temperature counts to scientific units which are then downloaded to the PXXX_#.DAT file listed in Table 1. However there may be circumstances where you need to run this program:

- the wrong Paroscientific calibration coefficients were loaded into the IES when the instrument was set up.
- the temperature sensor fails and you want to recover pressure data.

To execute this program, change directory to the location of the InitGuiDat2Mat.dat file and type the following at the MATLAB prompt:

```
>>IES_GUICOUNTS2MAT
```

The default IES_GUICOUNTS2MAT window will appear as in Figure D.1.

Enter:

- IES Serial number and file index.
- Raw DAT and Write MAT file Directories or press the “Initialize directories” button.

To load the Paroscientific conversion coefficients and the Bliley coefficients entered into the IES at set-up press the button

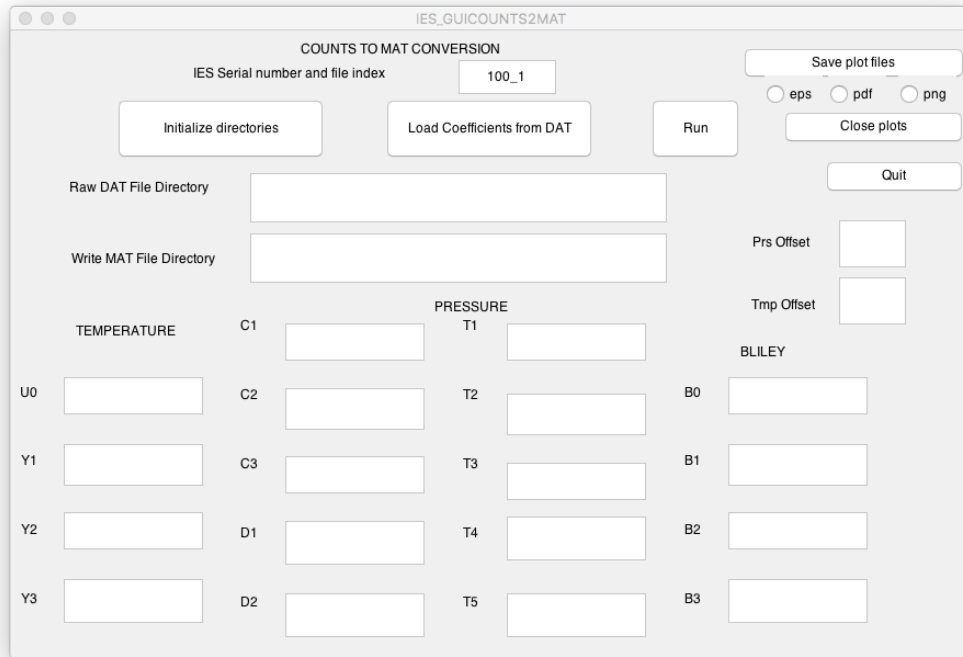


Figure D.1: Default IES_GUICOUNTS2MAT window

- **Load Coefficients from DAT**

Bliley coefficients B0, B1, B2 and B3 are then loaded from the file BLILEYLIB. The coefficients U0, Y1, Y2, Y3, C1, C2, C3, D1, D2, T1, T2, T3, T4, T5, and Prs and Tmp Offset are read from the KXXX_#.DAT file. Values can then be verified against the coefficients provided by Paroscientific and edited if necessary in the GUI window. Otherwise manually enter these conversion coefficients. Example values for 360_1 are shown in Figure D.2.

- Press **Run**

The program plots the converted temperature and pressure data, and then saves the data to a file called PXXX_#.mat which can be used by IES_GUIDRIVER. Save the plots by pressing the **Save Plots files** button. The plots will be equivalent to Figures 2 (pressure) and 3 (temperature) output by IES_GUIDAT2MAT and shown in Appendix A (Figure A.7 and Figure A.8).

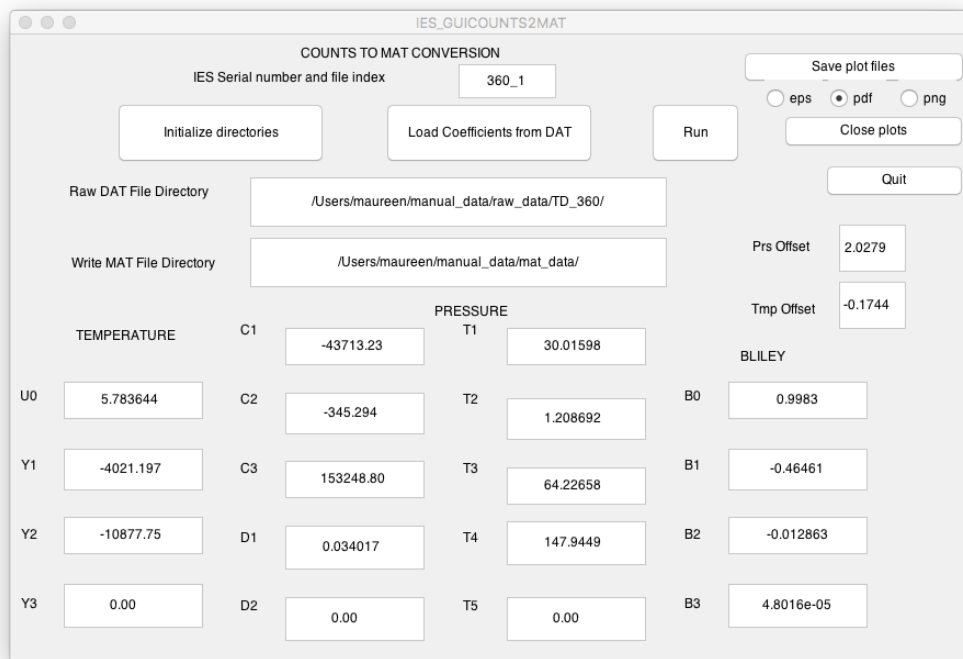


Figure D.2: Example IES_GUICOUNTS2MAT window after coefficients loaded