Helicorder Class for GISMO

Dane Ketner
daneketner@gmail.com

April 13, 2011

1 Helicorder Class

A helicorder (or seismogram) is a multi-line display of ground motion traditionally recorded onto paper around a rotating drum. The MatLAB helicorder class is designed to generate such a multi-line display and includes a number of customizeable properties that can be defined by the user, or ignored. One can think of a helicorder object `h` as a blueprint for the construction of a helicorder figure. The `build` function is then used to create the helicorder figure based on the properties in `h`. User-defined properties can override default properties if included in a call to the heliorder constructor, though all that is needed for a basic helicorder is a single waveform object input argument. In the following example, `w` is a waveform object containing three hours of data from REF:EHZ starting from March 29, 2009 starting at 07:00 UTC. Figure 1 displays the output of the `build` function, with `fh` being the handle of the figure generated.

```
h = helicorder(w);
fh = build(h);
```

We see that the resulting helicorder figure contains 10 minutes of data per line, and that waveform data is printed in black. Also notice the amplitude scaling that occurs in the plot. This scaling sets the max datapoint value at the center of the trace two traces above (or below) the current trace. The max value in this plot appears to be the first earthquake on line 6 (Line 07:50:00) which reaches up to the center of line 4 (Line 07:30:00). Minutes per line, trace color, and scale are all properties that can be manually set. A user-defined property can be set upon initial creation of `h` in the following way: `h = helicorder(w, 'mpl', 15);`. This call to the constructor would override the default value of 10 with 15 minutes per line. Property values can also be set after `h` is constructed using the `set` function as follows: `h = set(h, 'mpl', 15);`.

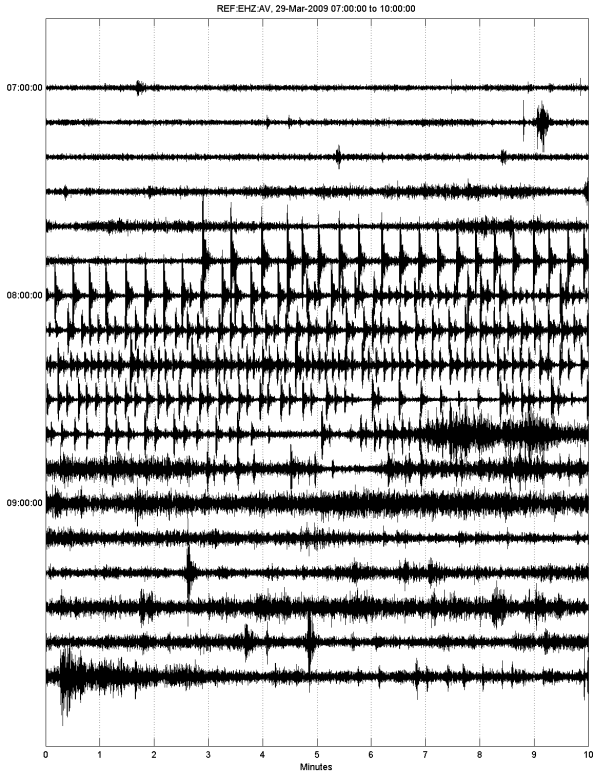

Figure 1: Default helicorder display for a single waveform

1.1 Helicorder Multi-waveform display

Next, let's test the default display created from an input `w` which contains multiple waveforms. In this case `w` is a 1x3 waveform object containing all three components from station REF for the same time span as the previous case. We know there will be more data to display than the previous example, so we set 20 minutes per line: `build(helicorder(w,'mpl',20))`. EHZ data is displayed in blue, EHE in black, and EHN in green in Figure 2. This multi-waveform display type is called 'alternate' which can be changed by the user in addition to the color scheme.

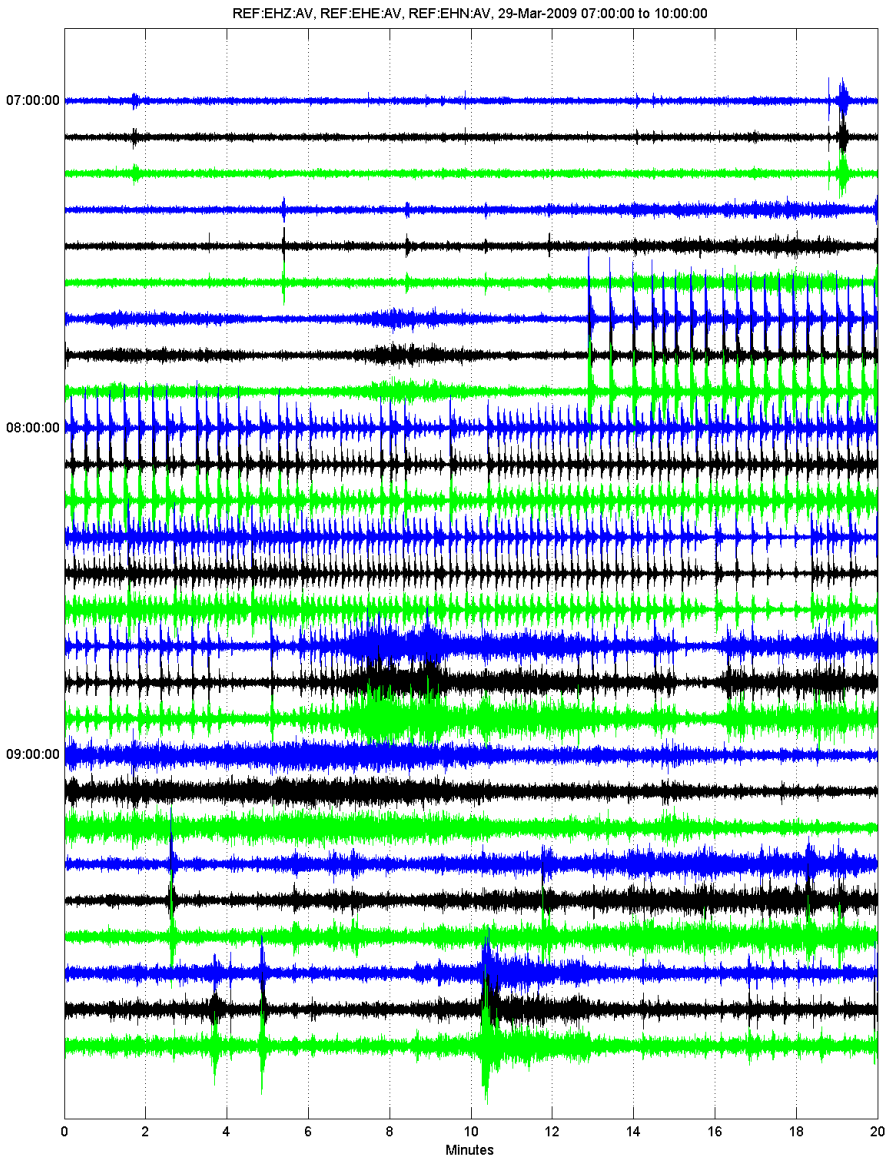

Figure 2: Default helicorder display for a 1x3 waveform

The multi-waveform display type can be changed from 'alternate' to 'group' or 'stack'. Setting display to 'group' will plot all trace data from the first element in `w` before starting the next. Figure 3 displays the output from `build(helicorder(w,'mpl',20,'display','group'))`.

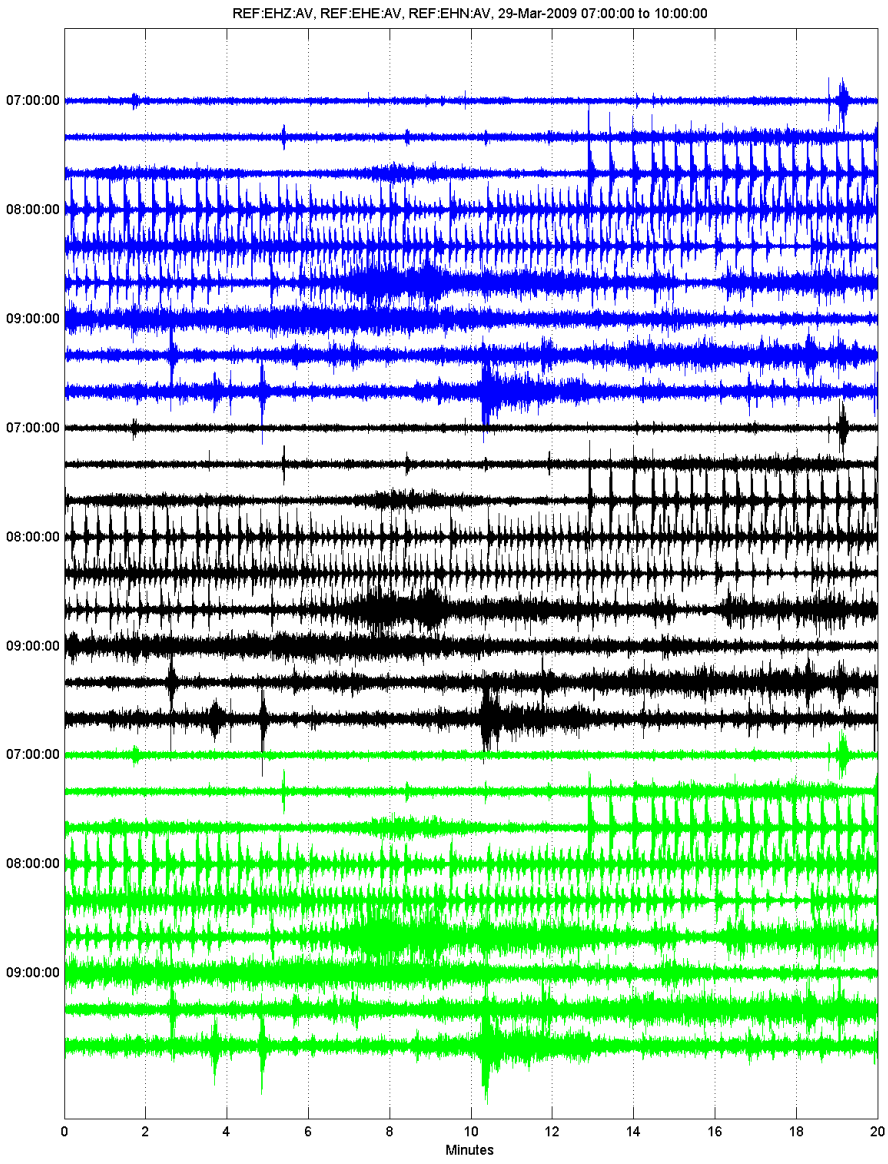

Figure 3: Multi-waveform 'group' display

The final multi-waveform display type is 'stack' which will plot waveforms one on top of the other. It should be noted that the default amplitude scaling of all waveforms is determined by the first waveform plotted in 'stack' mode, in this case `w(1)` is REF:EHZ. If separate amplitude scaling is desired for waveforms in w, it must be defined by the user. Figure 4 displays the output from
`build(helicorder(w,'display','stack'))`.

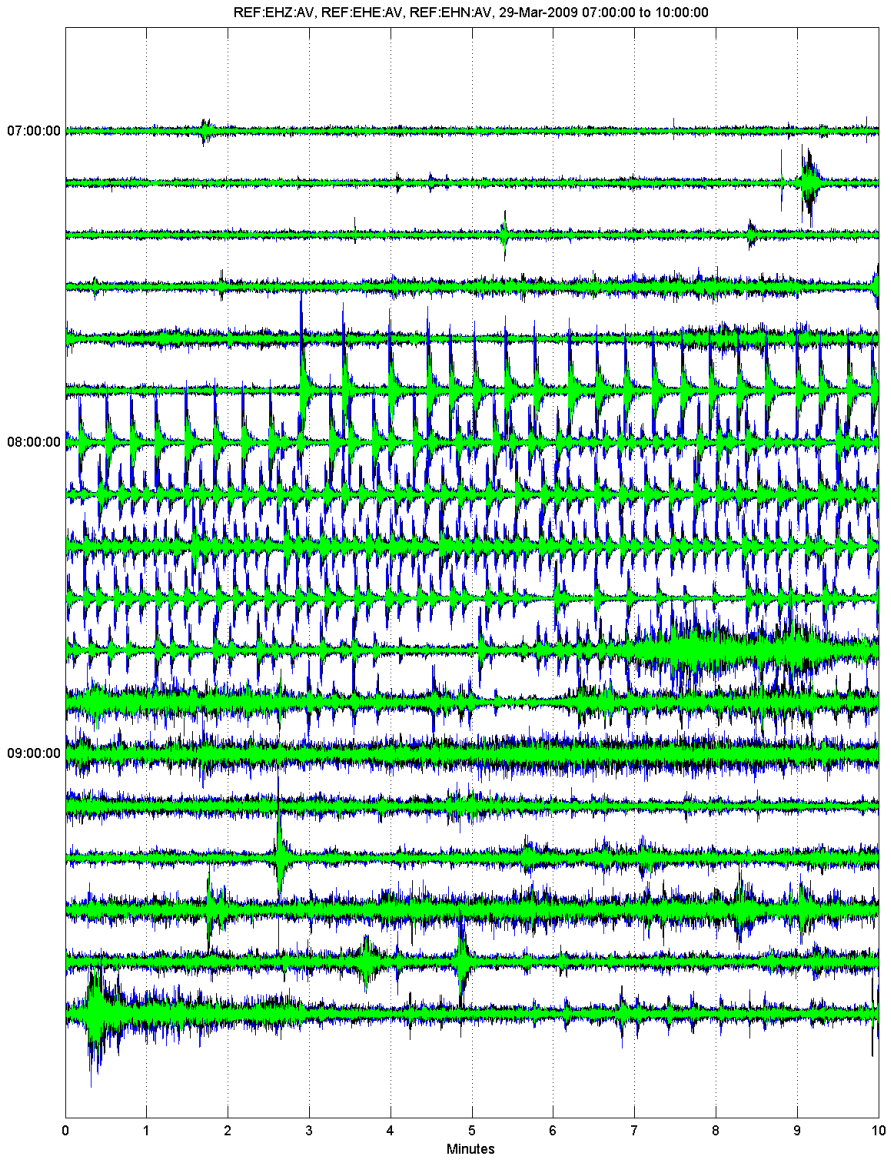

Figure 4: Multi-waveform 'stack' display

1.2 Helicorder event overlay

Helicorder can also be used to display discrete events from a waveform object. To do this, a set of event start/stop times (`sst1`) is required. Start/stop times can be manually picked, or can be the result of an auto-detection algorithm such as STA/LTA. 212 events are detected from the waveform data in Figure 5. In our case, `sst1` is a 212x2 array of MatLAB times defining the begining and end of each event. Figure 5 displays the output from `build(helicorder(w,'mpl',15,'e_sst',sst1))` where `w` is broadband data from RDWB:BHZ from March 27, 2009 that has been band-pass filtered between 1 Hz and 10 Hz.

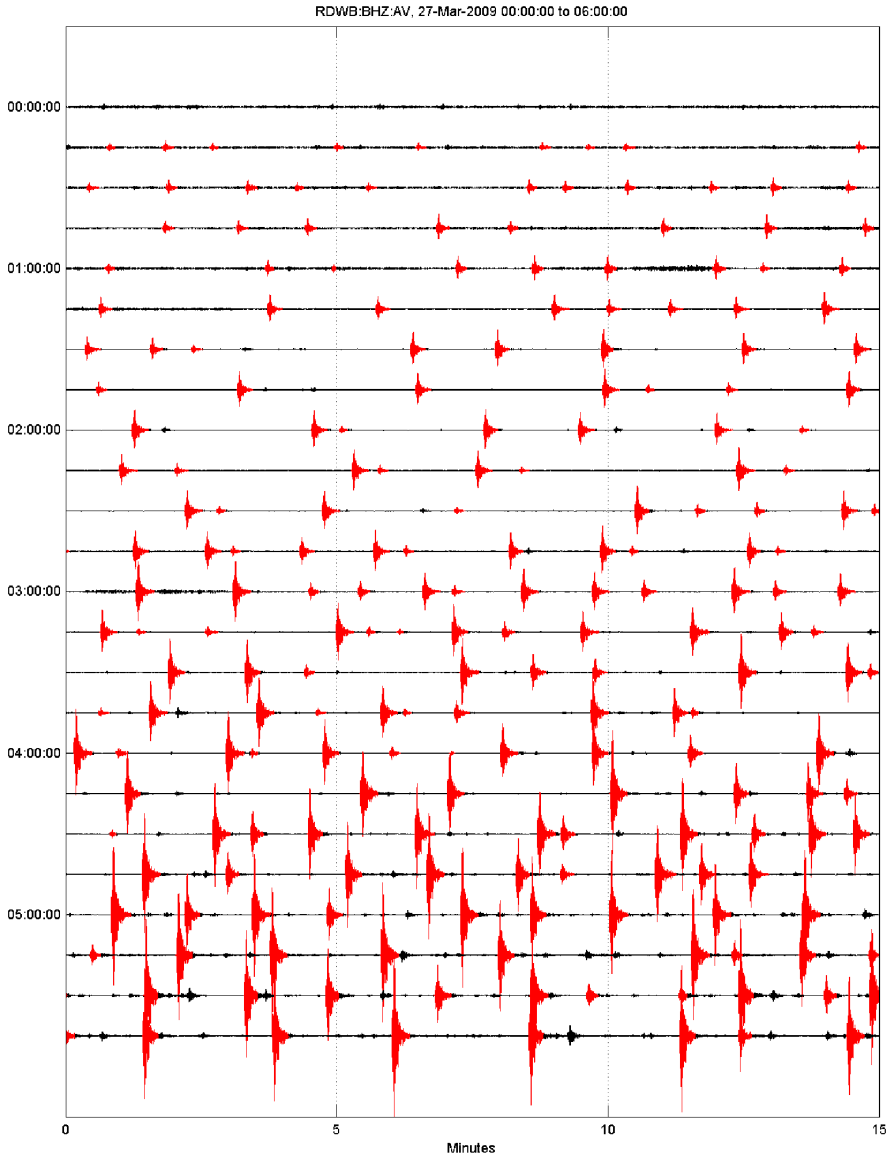

Figure 5: Helicorder with event overlay

1.3 Helicorder VLP overlay example

In the next example, `w` contains the unfiltered data from the previous event overlay example. Calling `build(helicorder(w,'mpl',15))` returns the following display:

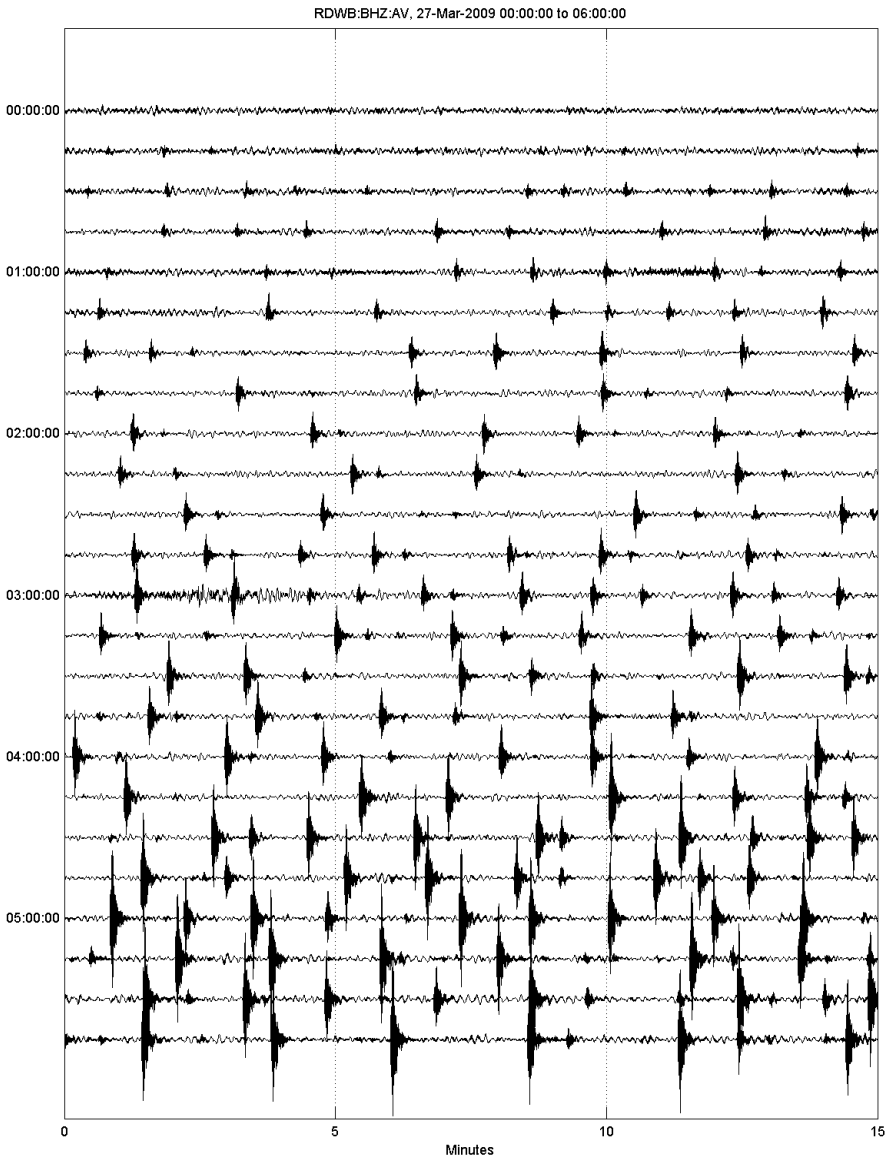

Figure 6: RDWB:BHZ helicorder

We suspect that there may be VLP energy in this data set, so we integrate `w` data to move from velocity to displacement. A second waveform `w_vlp` is created by filtering `w` over the VLP range (whatever you believe that might be). Next we combine `w` and `w_vlp` as follows: `w = [w w_vlp]`. This guarantees that VLP data is plotted on top of displacement data as seen in Figure 7. We aren't satisfied with the default trace colors so we set them ourselves:

```
color = {[0 0 0],[1 0 0]}; % Normalized RGB value for black and red
build(helicorder(w,'mpl',15,'trace_color',color))
```

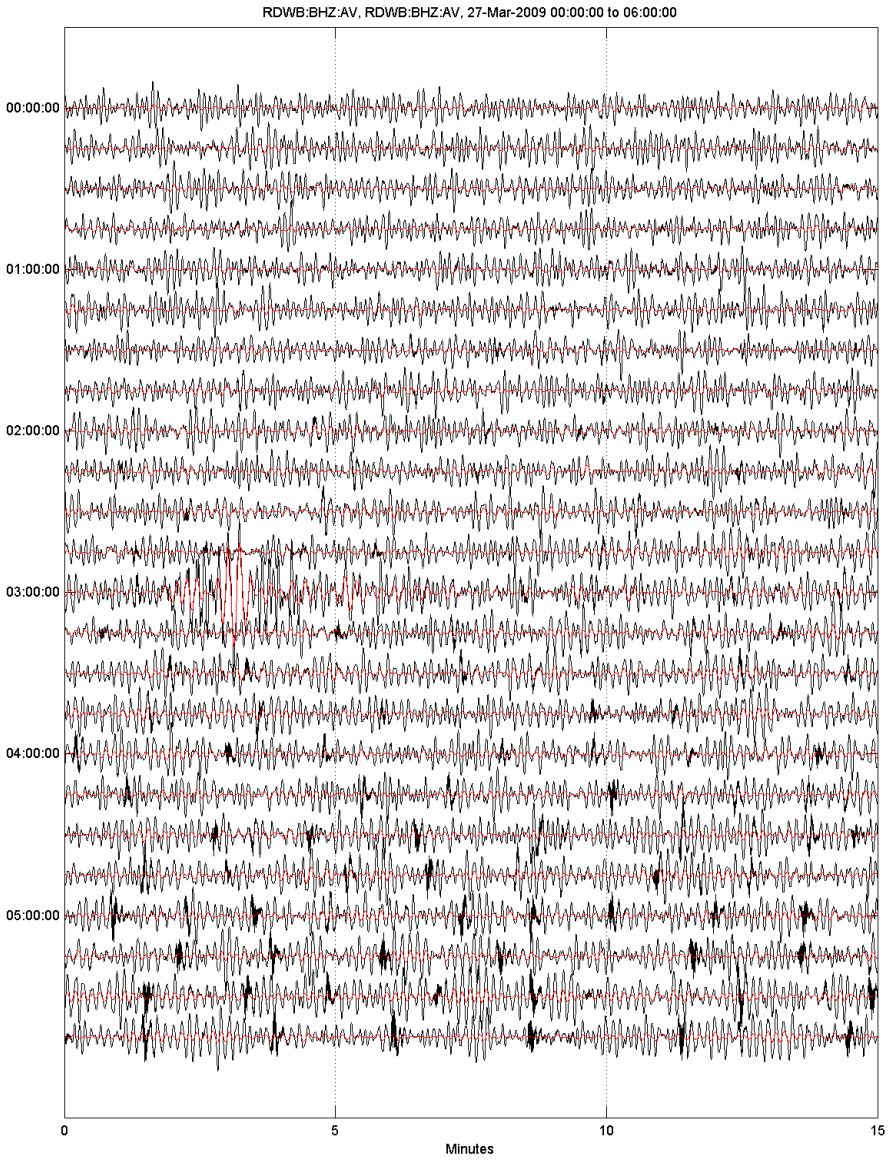

Figure 7: Helicorder VLP overlay

1.4 Valid helicorder property name/value combinations

- `e_sst` - Event Start/Stop Times
  If wave contains only one wavefrom object, `e_sst` can be entered as an Nx2 array of matlab times. If there is more than one waveform in wave (M waveforms), `e_sst` must be a 1xM cell array, each cell containing Nx2 array of start/stop times. A single waveform can also have multiple sets of start/stop times as in a waveform with multiple event families. In this case `e_sst` should be entered as a 1xL cell array for a particular waveform (with L separate groups to be highlighted). EXAMPLE:
  `h = helicorder(w,'e_sst',A)` where `w` is a 1x2 waveform object
  `A` is a 1x2 cell array
  `A(1,2)` is a 1x3 cell
  `A(1,2){1}` is a Nx2 numeric array
  `A(1,2){2}` is a Nx2 numeric array
  `A(1,2){3}` is a Nx2 numeric array
  DEFAULT = [] (No events)

- `mpl` - Minutes Per Line
  single numeric value specifying number of minutes per helicorder line. (This applies to all waveforms in the helicorder)
  DEFAULT = 10

- `ytick` - Y-Axis Ticks
  single numeric value specifying number of minutes between y-axis tick marks and labels. 'ytick' depends on 'mpl', for example, if 'mpl' is 20, an attempt to set 'ytick' to 10 will be result in 'ytick' rounded to nearest multiple of 'mpl' (20 in this case). Note also that manually setting 'mpl' will result in an automatic adjustment of 'ytick'.
  DEFAULT = 30

- `trace_color` - Color Scheme of Trace Data
  If `w` contains only one wavefrom object, `trace_color` can be entered as a 1x3 array of RGB values (between 0 and 1). For wave arguments longer than 1, `trace_color` should be entered as a 1xN cell array, each containing a 1x3 array of RGB values.

- `event_color` - Color Scheme of Highlighted Event Data
  Only use this property when `e_sst` property exists. `event_color` follows the same structure as `e_sst`. If wave contains only one waveform object, `event_color` can be entered as a 1x3 array of RGB values (between 0 and 1). For wave arguments longer than 1, `event_color` should be entered as a 1xN cell array, each containing a 1x3 array of RGB values. If multiple event sets exist for a single waveform, `event_color` must be specified accordingly.
  EXAMPLE: `h = helicorder(w,'e_sst',A,'event_color',B)`
  `A` is the same as the previous example
  `B(1,1)` is a 1x3 RGB array
  `A(1,2)` is a 1x3 cell
  `A(1,2){1}` is a 1x3 RGB array
  `A(1,2){2}` is a 1x3 RGB array
  `A(1,2){3}` is a 1x3 RGB array
  DEFAULT = [] (No events)

- `display` - Multi-Waveform Display Type
  `'single'` - Used when there is only one waveform in wave. None of the multi-waveform display types can be set unless multiple waveforms are passed.
  `'stack'` - Plots multiple waveforms over top of each other. (Multiple motivations could exist for wanting to display data in this way such as overlaying VLP energy).
  `'alternate'` - Alternate through waveforms (grouped by time).

'group' - Plot all trace data in a waveform before starting the next (grouped by waveform source). DEFAULT = 'single', 'alternate' (1 or multiple waveforms)

1.5 Interacting with a helicorder figure

The build fuction also provides a degree of interactivity with the helicorder figure. Clicking the cursor on any trace in the helicorder figure will call the function traClick from within build. The user can define what takes place after a helicorder trace line, or a highlighted event is clicked. By default, the GUI sst_pick is opened for the clicked waveform generating a minute-long waveform and spectral display.

Note: Helicorder interactivity is shut off by default as of this writing. If interested parties care to turn this on in the helicorder function build.m, they are welcome to.