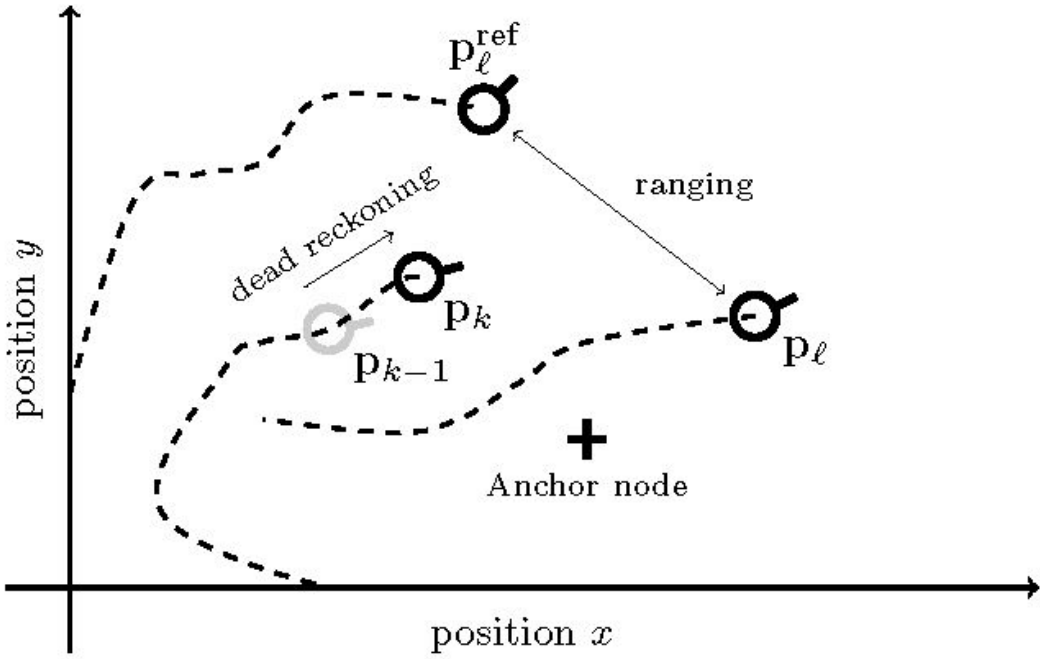


Understanding animal behaviour and location in the landscape using animal-borne technologies

User Guide and Installation

20.05.2016

Installation

1. Download and install Anaconda for python 3 (https://www.continuum.io/downloads)
2. Using Anaconda, install Python 3.4 (It ships with 3.5).
    a. Open a console/terminal window in the Anaconda installation directory. By default, this is located at "`C:\Anaconda3\`" on Windows.
    b. Run "`conda create -n py34 python=3.4 anaconda`".
3. Download and install PyQt5, by running the following command: "`conda install -c https://conda.anaconda.org/mmcauliffe pyqt5`".
        i. N.B. On a Mac, once installed, you must append to your qt.conf file in the same directory as your python binary for your environment to contain a path to your QT5 plugins, this way it can find the cocoa library i.e:
            1. Plugins = /usr/local/opt/qt5/plugins
4. Download the repository's master branch from GitHub (https://github.com/davemccormick/pyAnimalTrack), and extract the files.

User Guide

Startup

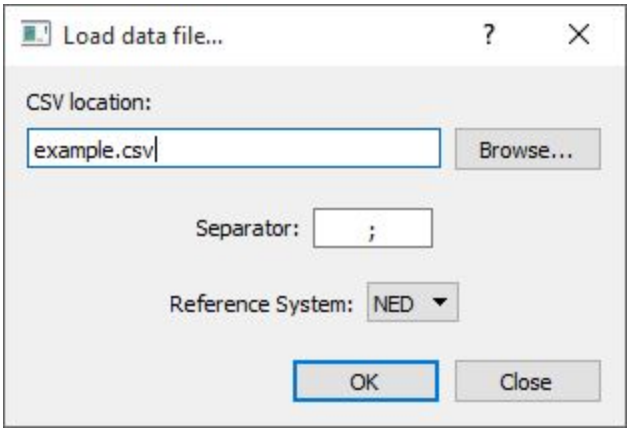
Windows

1. Open a console window to the directory you extracted pyAnimalTrack to, within this directory move into pyAnimalTrack/src/.
2. Once in the correct directory, run "`C:\Anaconda3\envs\py34\python.exe pyAnimalTrackRunner`" (Assuming you kept the default installation directories and the UI should now be open.

OSX

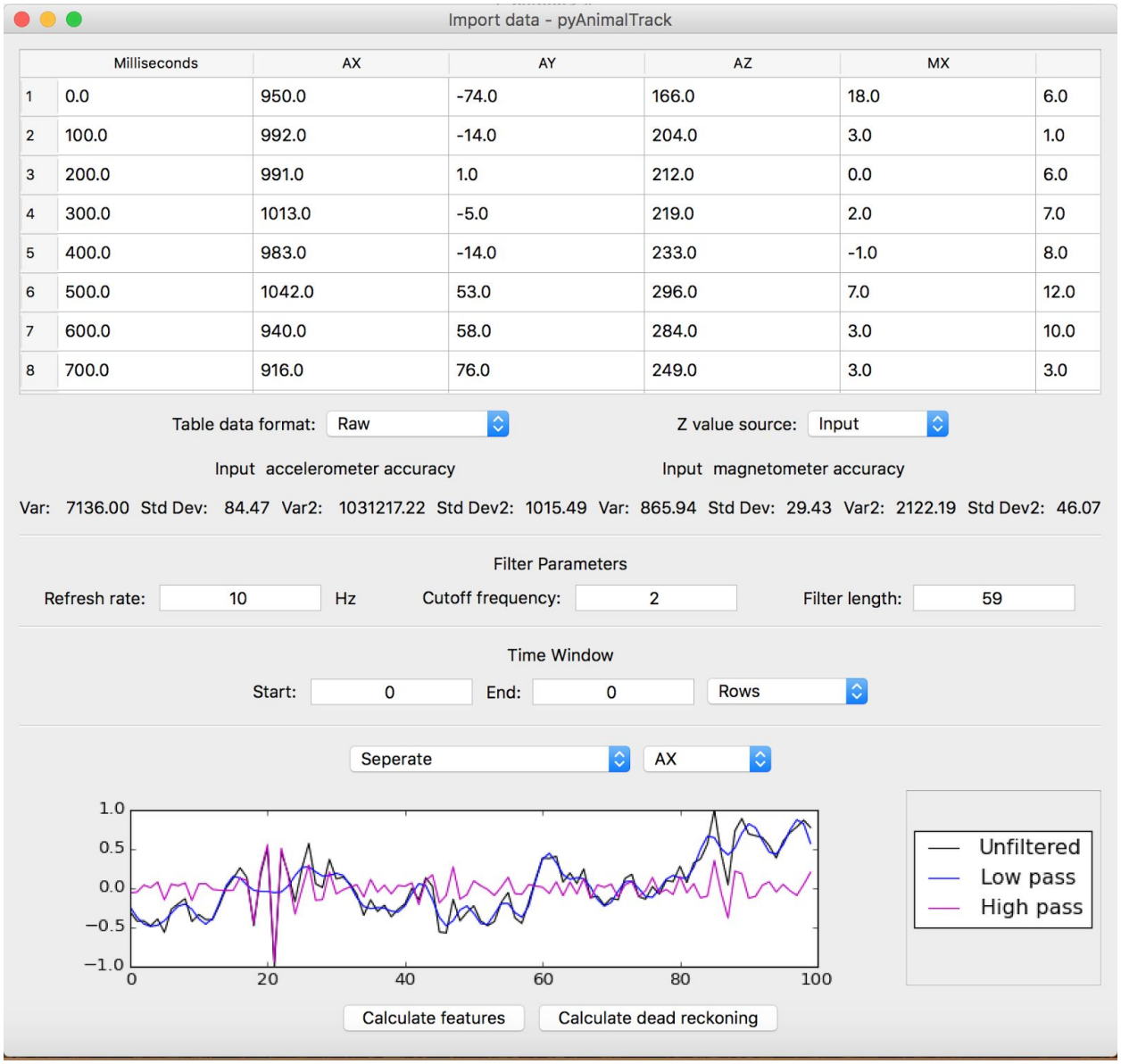
1. To begin, activate your virtual environment:
   "source activate py34"
2. Once activate, navigate to the directory you extracted pyAnimalTrack to, within this directory move into pyAnimalTrack/src/
3. Once in the correct directory, run "python pyAnimalTrackRunner" and the UI should now be open.

File open dialog:



1. File location: The data file to parse.
2. Separator: Which character has been used to deliminate each cell value.
3. Reference System: The sensor reference system that was used during recording.

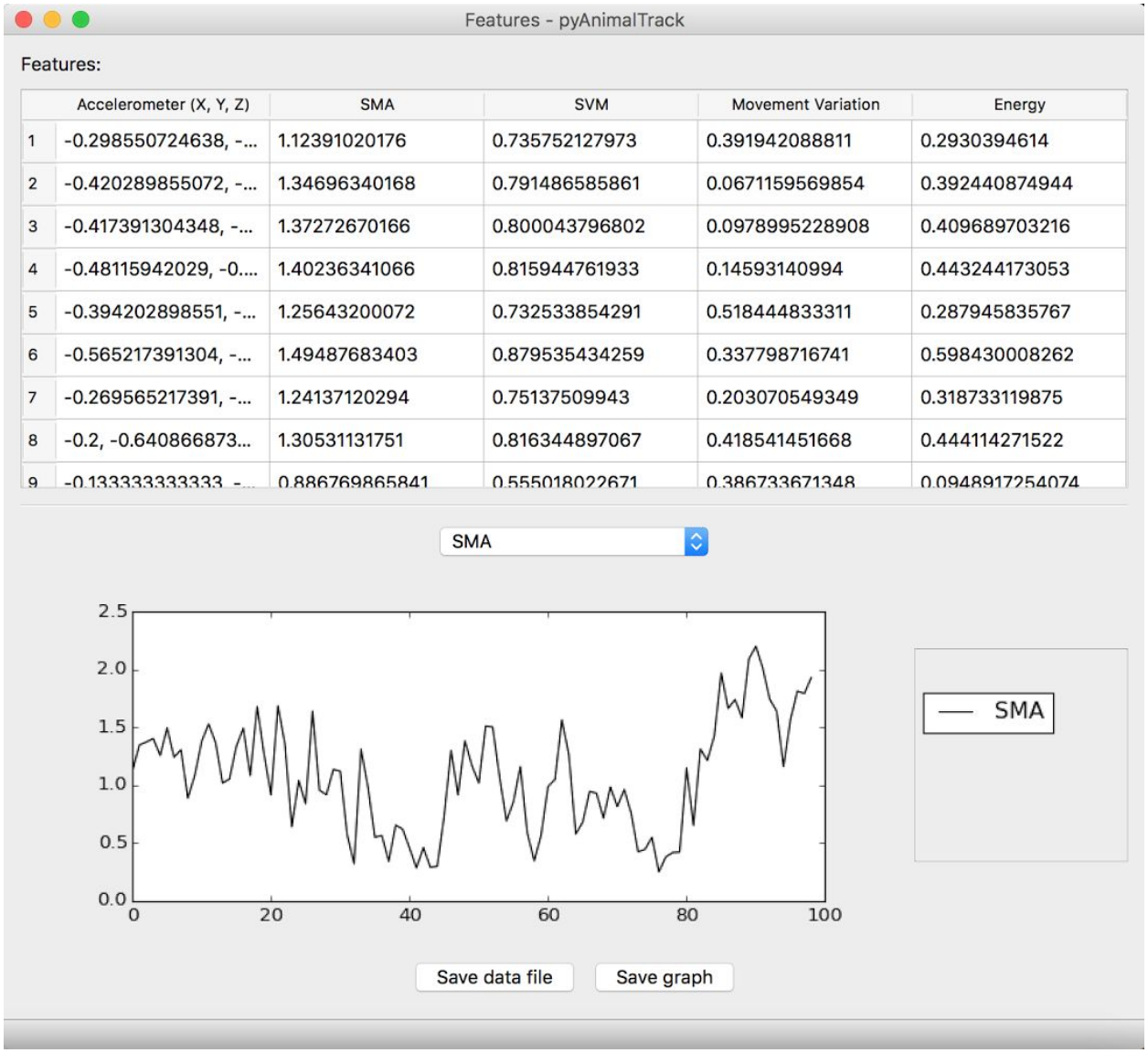
Main window:



1. Table view: The dataset shown as a table of values. This can be changed to show either the raw imported data, the calibrated data, or the filtered data (Low and High pass filters).
2. Filter parameters: These can be adjusted on a per column basis, and control the operation of the Low and High pass filters.
3. Epoch selection: This can be used to choose a subset of the data for any operations to occur upon, by either a selection of rows, or an amount of time (in milliseconds)

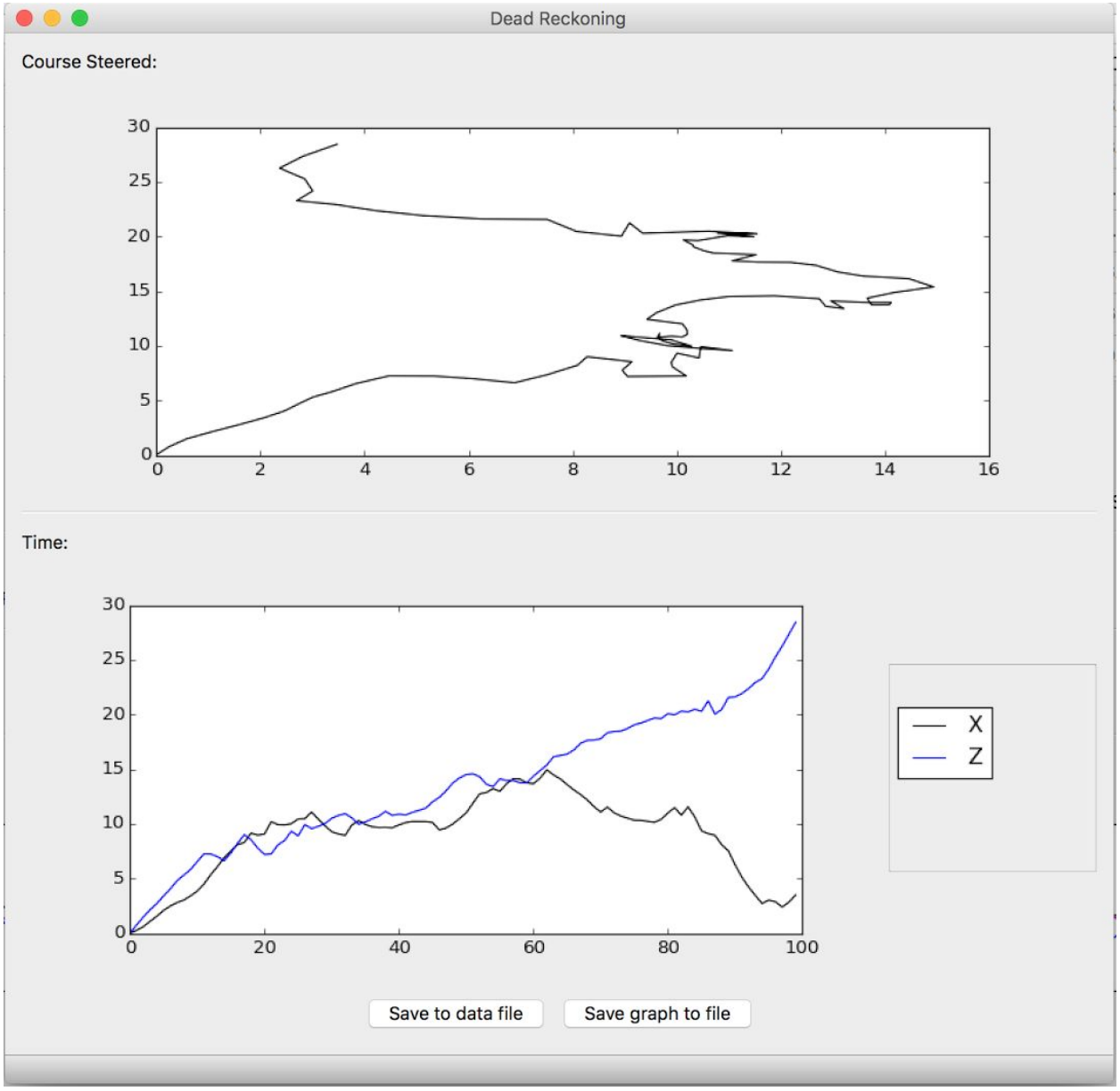
4. Graph: A graph that can be used to display a combination of data sets, such as the three different filtered values for a column, or all values for a sensor.

Features:



1. Table view: The complete feature based dataset is shown, including the accelerometer values, and all calculated features.
2. Feature graph: A graph of the currently selected feature.
3. Saving: Use these buttons to export the calculated feature information, both as a CSV, and saving the graph to file.

Dead Reckoning



1. Course Steered: This graph shows the path which the sensor has moved, plotting X and Y against each other.
2. Time: This graph shows how the X and Y coordinates have separately changed over time, independently of each other.
3. Saving: Use these buttons to export the calculated dead reckoning information, both as a CSV, and saving the graphs to file.

External settings

After the software has been run once, a settings.json configuration file will be created at "`pyAnimalTrack/src/pyAnimalTrack/settings.json`". This file allows for the end user to choose between a selection of colours to be used by the program's graphing functionality.


```
{
  "graph_SaveFormats": [
    "png"
  ],
  "ground_reference_frame_options": [
    "NED",
    "ENU"
  ],
  "csv_separator": ";",
  "filter_parameters": {
    "SampleRate": 10,
    "CutoffFrequency": 2,
    "FilterLength": 59
  },
  "lines": [
    "k",
    "b",
    "m"
  ],
  "scaling": {
    "az": 1,
    "ay": -1,
    "my": 1,
    "ax": -1,
    "gy": 1,
    "mx": 1,
    "gz": 1,
    "gx": 1,
    "mz": 1
  },
  "data_SaveFormats": [
    "csv",
    "txt"
  ]
}
```

Under the “lines” value, each colour can be adjusted, with the choices listed below:

b	Blue	m	Magenta
g	Green	y	Yellow
r	Red	k	Black
c	Cyan		

Walkthrough

For a full walkthrough, please refer to the GitHub page which links to a video based demonstration of all features included in the software. The GitHub page also links to the (technically orientated) Sphinx generated documentation.