APOLLO SEISMIC DATA ANALYSIS IN PYTHON: OBSPY MODULE AND RESPONSE FUNCTION.

Y. Yamamoto[1,5], R. Yamada[2], Y. Ishihara[3], Y. Nakamura[4], and H. Ishikawa[5], [1]ISAS/JAXA, Japan (Yoshino-dai 3-1-1, Chuo, Sagamihara, Kanagawa 252-5210 Japan. E-mail: yukio@planeta.sci.isas.jaxa.jp), [2]University of Aizu, Japan, [3]NIES, Japan, [4]University of Texas Institute for Geophysics, [5]Tokyo Metropolitan University, Japan.

Introduction: During the Apollo missions in the 1970s, the seismometers were deployed on the moon as a part of the Apollo Lunar Surface Experiments Packages, ALSEP. The Apollo seismic datasets were archived in the original binary format[1] and are distributed from the DARTS website in JAXA. They still are the only datasets of seismic signals available to u after nearly 50 years have passed.

Standard seismic data formats, such as SEED, were established only after the Apollo missions, and analysis tools were also developed for terrestrial seismology. For terrestrial seismologist, it is more convenient to apply familiar tools without decoding a specific format. There are two approaches to perform this: one is to convert the original data to a standard format, and the other is to extend the analysis tool to support the Apollo original format. Converting SEED format was attempted by Null et al.[2] while we used the latter method.

ObsPy: Recently, the programming language Python is drawing attention especially in machine learning field and Python users are increasing. ObsPy is a seismic analysis tool in Python [3], and it supports many formats in seismology. We have created a module: 'obspy.io.alsep' to read the Apollo data directly. The module supports three tape formats: Passive Seismic Experiment (PSE), Work Tape in Normal-bitrate (WTN), and Work Tape in High-bitrate (WTH). Figure 1 shows the decoding flow of tape files in the module. All tape files are decoded to ALSEP words once and then remapped to each component appropriately. Figure 2 also shows the basic steps to read seismic data from the original PSE Tape files. A user can specify URI of a specific tape file directly, and extract the seismic data by selecting a channel.

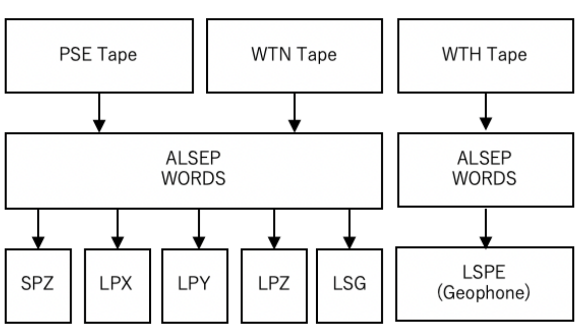

Figure 1. Decoding pf ALSEP Tapes. ALSEP WORDS are extracted from each tape, and remapped to each component.

Table 1. Channel name examples used in the module.

Instrument	Current	Recommend
Short-period	SPZ	SHZ
Long-period	LPX/LPY/LPZ TDX/TDY/TDZ	MH1/MH2/MH3 LH1/LH2/LH3
Gravimeter	LSG	SGZ
Geophone	GP1/GP2/GP3/GP4	SPZ (location: 01/02/03/04)

Stream Design: ObsPy deals with seismic data as a stream identified by a combination of network, station, location, and channel as well as the SEED format. Recently, the Apollo network was registered as 'XA' to International Federation of Digital Seismograph Networks, FDSN. The station codes were also referable in FDSN: 'S' + Apollo mission number. We adopted the same definition for the network and stations. However, we do not use the standard names for channels. The channel name convention is recommended by the SEED Manual 2.4, Appendix A. However, the Apollo seismometers have been analyzed for a long time with the traditional channel names. Therefore, the traditional names are used exceptionally as the first attempt. Table 1 lists representative channel names in the module. The location is not used currently, but it might be useful to distinguish LSPE geophones in the future.

The stream includes traces to manage a chunk of time-series data. In designing, we considered how much data one trace should hold. Each trace has one start time, and Apollo seismic data are also embedded in transfer frames with a time tag. Logically, each frame has enough information to construct a trace. However, it was difficult to make a one-to-one correspondence between a frame and a trace because too many traces would make program run slower in analysis. For this reason, we put multiple frames in the same trace if the frames were continuous.

Response function: The response functions were provided by equations during the Apollo missions, and most of users used them to remove the instrument response. Considering the easy use in ObsPy, we provided poles and zeros for the Apollo long-period seismometers (LPX/LPY/LPZ) and short-period seismometer (SPZ) in FDSN StationXML format. First, we adopted SAC-style ASCII text format because LPX/LPY/LPZ and SPZ use the same response function throughout Apollo missions. Later, we changed the format to FDSN StationXML to describe different response functions at different times, which was important for the LP seismometers because they have two operational modes: peaked mode and flat mode. Figure 3 shows the displacement response in ObsPy. FDSN StationXML was loaded as inventory instance, and it could be applied for seismic data.

Future work: This module requires a few minutes to decode a file. One of the challenges is speeding up the decoding. Also, a function should be prepared to help users search which files to use for a given set of channels and time.

References: [1] Nakamura Y. (1992), *UTIG technical report* 118. [2] Nunn C. *et al* (2017), *AGU*. [3] Beyruther M. *et al*. (2010), *SRL*.

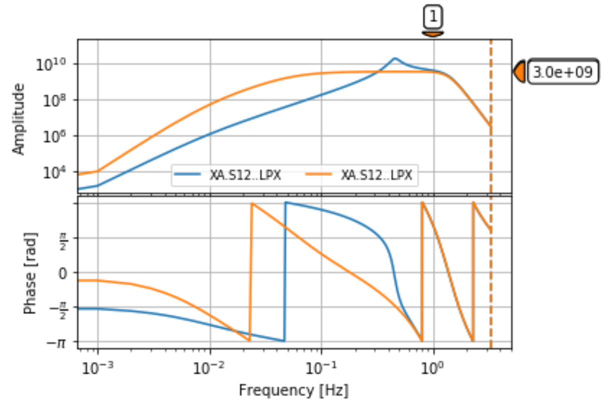

Figure 3. Apollo LP displacement response in ObsPy. Peaked and Flat modes are switched by specified time.

In [1]:
```
from obspy import read, UTCDateTime
import matplotlib.pyplot as plt
```

In [2]:
```
st = read('http://darts.isas.jaxa.jp/pub/apollo/pse/p15s/pse.a15.1.2')
```

In [3]:
```
st_spz = st.select(id='XA.S15..SPZ')
```

In [4]:
```
st_spz
```

Out[4]:
```
13 Trace(s) in Stream:
XA.S15..SPZ | 1971-08-01T18:52:00.515000Z - 1971-08-01T23:36:03.816887Z | 53.0 Hz, 903296 samples
XA.S15..SPZ | 1971-08-01T23:36:10.162000Z - 1971-08-02T01:39:05.237472Z | 53.0 Hz, 390880 samples
XA.S15..SPZ | 1971-08-02T01:39:05.468000Z - 1971-08-02T01:40:33.600075Z | 53.0 Hz, 4672 samples
XA.S15..SPZ | 1971-08-02T01:40:34.218000Z - 1971-08-02T01:40:58.953849Z | 53.0 Hz, 1312 samples
XA.S15..SPZ | 1971-08-02T01:40:59.575000Z - 1971-08-02T01:41:00.159906Z | 53.0 Hz, 32 samples
XA.S15..SPZ | 1971-08-02T01:41:00.783000Z - 1971-08-02T01:41:01.367906Z | 53.0 Hz, 32 samples
XA.S15..SPZ | 1971-08-02T01:41:02.594000Z - 1971-08-02T01:41:03.178906Z | 53.0 Hz, 32 samples
XA.S15..SPZ | 1971-08-02T01:41:04.393000Z - 1971-08-02T02:53:21.279792Z | 53.0 Hz, 229856 samples
XA.S15..SPZ | 1971-08-02T02:53:20.461000Z - 1971-08-02T05:56:59.310057Z | 53.0 Hz, 584000 samples
XA.S15..SPZ | 1971-08-02T06:01:58.797000Z - 1971-08-02T06:10:30.174358Z | 53.0 Hz, 27104 samples
XA.S15..SPZ | 1971-08-02T06:10:48.277000Z - 1971-08-02T13:35:47.126057Z | 53.0 Hz, 1415040 samples
XA.S15..SPZ | 1971-08-02T13:35:53.520000Z - 1971-08-02T13:46:52.821887Z | 53.0 Hz, 34944 samples
XA.S15..SPZ | 1971-08-02T13:46:53.409000Z - 1971-08-02T18:52:01.012774Z | 53.0 Hz, 970304 samples
```

In [5]:
```
st_spz.plot(starttime=UTCDateTime('1971-08-02T17:11:22.6'), endtime=UTCDateTime('1971-08-02T17:11:35.8'))
plt.show()
```

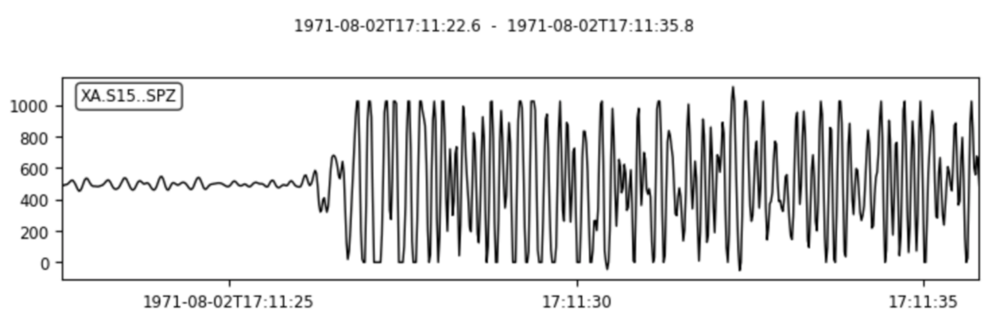

Figure 2. Data reading steps of ObsPy module: obspy.io.alsep. The PSE tape file 'pse.a.15.1.2' is loaded. It has the stream 'XA.S15..SPZ' which includes 13 traces for SPZ.