pyrotd Documentation

Release 0.6.1

Albert R. Kottke

Jun 30, 2023

Contents

CHAPTER 1

pyRotd

|PyPi Cheese Shop| |Documentation| |Build Status| |Code Quality| |Test Coverage| |License| |DOI|

Acceleration response spectrum calculations implemented in Python.

1.1 Introduction

Simple Python functions for calculating psuedo-spectral acceleration and rotated psuedo-spectral acceleration. The response of the single-degree-of-freedom oscillator is computed in the frequency domain along with frequency-domain interpolation to accurately capture the high-frequency characteristics.

The calculation of the response spectrum is performed using frequency domain transfer functions, as well as frequency domain interpolation methods to insure that the time step of the motions is greater than 10 times the natural frequency of the oscillator frequency. Two perpendicular ground motions can be rotated to compute the response at various percentiles. For example, the minimum, median, and maximum could be computed using percentiles of 0, 50, and 100. The orientation of both the minimum and maximum percentile are provided, but not orientation is provided for other percentiles because the rotate spectral acceleration is not monotonically increasing.

1.2 Installation

`pyrotd` is available from the Python Cheese Shop:

```
pip install pyrotd
```

1.3 Example

```
#!/usr/bin/env python3
import os

import matplotlib.pyplot as plt
import numpy as np

import pyrotd

pyrotd.processes = 1

# Load the AT2 timeseries
fname = os.path.join(
    os.path.dirname(__file__), "test_data/RSN8883_14383980_13849360.AT2"
)
with open(fname) as fp:
    for _ in range(3):
        next(fp)
    line = next(fp)
    time_step = float(line[17:25])
    accels = np.array([p for l in fp for p in l.split()]).astype(float)

# Compute the acceleration response spectrum
osc_damping = 0.05
osc_freqs = np.logspace(-1, 2, 91)
resp_spec = pyrotd.calc_spec_accels(time_step, accels, osc_freqs, osc_damping)

# Create a plot!
fig, ax = plt.subplots()

ax.plot(resp_spec.osc_freq, resp_spec.spec_accel)

ax.set(
    xlabel="Frequency (Hz)",
    xscale="log",
    ylabel="5%-Damped Spectral Accel. (g)",
    yscale="log",
)
ax.grid()
fig.tight_layout()
plt.show(fig)
```

A more thorough example of using the code is provided here

1.4 API

pyrotd.**calc_oscillator_resp**(*freq*, *fourier_amp*, *osc_damping*, *osc_freq*, *max_freq_ratio=5.0*, *peak_resp_only=False*, *osc_type='psa'*)

Compute the time series response of an oscillator.

Parameters

- **freq** (*array_like*) – frequency of the Fourier acceleration spectrum [Hz]
- **fourier_amp** (*array_like*) – Fourier acceleration spectrum [g-sec]
- **osc_damping** (*float*) – damping of the oscillator [decimal]
- **osc_freq** (*float*) – frequency of the oscillator [Hz]

- **max_freq_ratio** (*float, default=5*) – minimum required ratio between the oscillator frequency and then maximum frequency of the time series. It is recommended that this value be 5.
- **peak_resp_only** (*bool, default=False*) – If only the peak response is returned.
- **osc_type** (*str, default='psa'*) –

  type of response. Options are: 'sd': spectral displacement 'sv': spectral velocity 'sa': spectral acceleration 'psv': psuedo-spectral velocity 'psa': psuedo-spectral acceleration

Returns **response** – time series response of the oscillator

Return type numpy.ndarray or float

pyrotd.**calc_rotated_oscillator_resp**(*angles*, *percentiles*, *freqs*, *fourier_amps*, *osc_damping*, *osc_freq*, *max_freq_ratio=5.0*, *osc_type='psa'*, *method='optimized'*)

Compute the percentiles of response of a rotated oscillator.

Parameters

- **percentiles** (*array_like*) – percentiles to return.
- **angles** (*array_like*) – angles to which to compute the rotated time series.
- **freq** (*array_like*) – frequency of the Fourier acceleration spectrum [Hz]
- **fourier_amps** (*[array_like, array_like]*) – pair of Fourier acceleration spectrum [g-sec]
- **osc_damping** (*float*) – damping of the oscillator [decimal]
- **osc_freq** (*float*) – frequency of the oscillator [Hz]
- **max_freq_ratio** (*float, default=5*) – minimum required ratio between the oscillator frequency and then maximum frequency of the time series. It is recommended that this value be 5.
- **peak_resp_only** (*bool, default=False*) – If only the peak response is returned.
- **osc_type** (*str, default='psa'*) –

  type of response. Options are: 'sd': spectral displacement 'sv': spectral velocity 'sa': spectral acceleration 'psv': psuedo-spectral velocity 'psa': psuedo-spectral acceleration
- **method** (str, default of *optimized*) – If *optimized*, then the rotation is done on a subset of values. If *rigorous*, then all values are considered.

Returns **response** – time series response of the oscillator

Return type numpy.ndarray or float

pyrotd.**calc_rotated_peaks**(*time_step*, *accel_a*, *accel_b*, *peak_type='pga'*, *percentiles=None*, *angles=None*, *method='optimized'*)

Compute the rotated peak values of input accelerations.

Parameters

- **time_step** (*float*) – time step of the time series [s]
- **accel_a** (*array_like*) – acceleration time series of the first motion [g]
- **accel_b** (*array_like*) – acceleration time series of the second motion that is perpendicular to the first motion [g]
- **peak_type** (*str*) – Either: pga, pgv, or pgd

- **percentiles** (*array_like or None*) – percentiles to return. Default of [0, 50, 100],
- **angles** (*array_like or None*) – angles to which to compute the rotated time series. Default of np.arange(0, 180, step=1) (i.e., 0, 1, 2, .., 179).
- **method** (str, default of *optimized*) – If *optimized*, then the rotation is done on a subset of values. If *rigorous*, then all values are considered.

Returns **rotated_peak** – If *pga*, then the units are in *g*. Otherwise, they are reported in *cm/s* or *cm* for *pgv* or *pgd*, respectively.

Return type float

pyrotd.**calc_rotated_percentiles**(*accels*, *angles*, *percentiles=None*, *method='optimized'*)

Compute the response spectrum for a time series.

This function computes the period-dependent rotated percentiles. If *optimized* is *True*, then the percentiles are only computed at times greater than a threshold. This optimized procedure is described in Equations 2.4 and 2.5 of Stewart et al. (2017) PEER report [1]_.

Parameters

- **accels** (*list of array_like*) – pair of acceleration time series
- **angles** (*array_like*) – angles to which to compute the rotated time series
- **percentiles** (*array_like or None*) – percentiles to return
- **method** (str, default of *optimized*) – If *optimized*, then the rotation is done on a subset of values. If *rigorous*, then all values are considered.

Returns **rotated_resp** – Percentiles of the rotated response. Records have keys: 'percentile', 'spec_accel', and 'angle'.

Return type np.recarray

pyrotd.**calc_rotated_spec_accels**(*time_step*, *accel_a*, *accel_b*, *osc_freqs*, *osc_damping=0.05*, *percentiles=None*, *angles=None*, *max_freq_ratio=5*, *osc_type='psa'*, *method='optimized'*)

Compute the rotated psuedo-spectral accelerations.

Parameters

- **time_step** (*float*) – time step of the time series [s]
- **accel_a** (*array_like*) – acceleration time series of the first motion [g]
- **accel_b** (*array_like*) – acceleration time series of the second motion that is perpendicular to the first motion [g]
- **osc_freqs** (*array_like*) – natural frequency of the oscillators [Hz]
- **osc_damping** (*float*) – damping of the oscillator [decimal]. Default of 0.05 (i.e., 5%)
- **percentiles** (*array_like or None*) – percentiles to return. Default of [0, 50, 100],
- **angles** (*array_like or None*) – angles to which to compute the rotated time series. Default of np.arange(0, 180, step=1) (i.e., 0, 1, 2, .., 179).
- **max_freq_ratio** (*float, default=5*) – minimum required ratio between the oscillator frequency and then maximum frequency of the time series. It is recommended that this value be 5.

- **method** (str, default of *optimized*) – If *optimized*, then the rotation is done on a subset of values. If *rigorous*, then all values are considered.

Returns **rotated_resp** – computed pseudo-spectral acceleration [g] at each of the percentiles. Records have keys: 'osc_freq', 'percentile', 'spec_accel', and 'angle'

Return type np.recarray

pyrotd.**calc_spec_accels**(*time_step*, *accel_ts*, *osc_freqs*, *osc_damping=0.05*, *max_freq_ratio=5*, *osc_type='psa'*)

Compute the psuedo-spectral accelerations.

Parameters

- **time_step** (*float*) – time step of the time series [s]
- **accel_ts** (*array_like*) – acceleration time series [g]
- **osc_freqs** (*array_like*) – natural frequency of the oscillators [Hz]
- **osc_damping** (*float*) – damping of the oscillator [decimal]. Default of 0.05 (i.e., 5%)
- **max_freq_ratio** (*float, default=5*) – minimum required ratio between the oscillator frequency and then maximum frequency of the time series. It is recommended that this value be 5.
- **osc_type** (*str, default='psa'*) –

  type of response. Options are: 'sd': spectral displacement 'sv': spectral velocity 'sa': spectral acceleration 'psv': psuedo-spectral velocity 'psa': psuedo-spectral acceleration

Returns **resp_spec** – computed pseudo-spectral acceleration [g]. Records have keys: 'osc_freq', and 'spec_accel'

Return type np.recarray

1.5 Contributing

Python Module Index

p

Index

C

P