pynoise Documentation

Release v0.3.4

Marco Guerreiro

Nov 06, 2018

CONTENTS:

CHAPTER

ONE

INTRODUCTION

This package was created to add noise to signals, which can be matrices or vectors. As of October 2018, only the additive white gaussian noise is implemented. Future implementations can consider other types of noise, as listed in Colors of Noise.

1.1 Additive White Gaussian Noise

Additive White Guassian Noise (AWGN) is a model of noise that mimics the occurence of random processes. AWGN is derived from a normal distribution, having its energy spread across all spectrum.

CHAPTER

TWO

EXAMPLES

2.1 Corrupt ramp signal

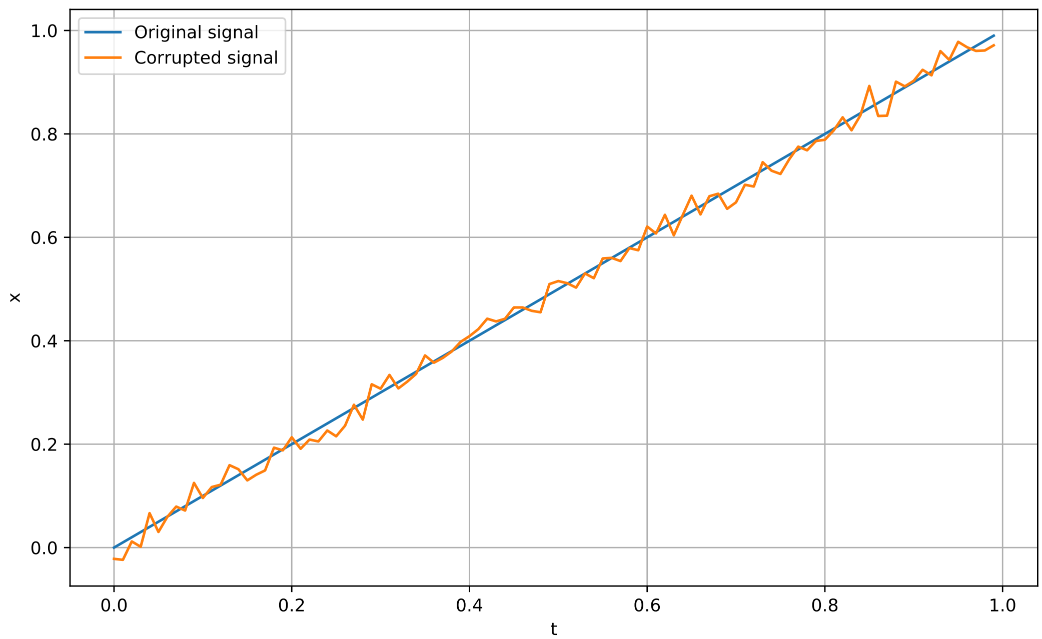
Add 30 dB of white gaussian noise to the ramp signal $x(t) = t$:

```
import numpy as np
import pynoise
import matplotlib.pyplot as plt

# --- Signal ---
t = np.arange(0, 1, 0.01)

x = t
xn = pynoise.awgn(x, 30)

# --- Plots ---
plt.figure(figsize=(10,6))
plt.plot(t, x, label='Original signal')
plt.plot(t, xn,  label='Corrupted signal')
plt.grid()
plt.xlabel('t')
plt.ylabel('x')
plt.legend()
plt.show()
```


Original signal
Corrupted signal
1.0
0.8
0.6
0.4
0.2
0.0
x
0.0
0.2
0.4
0.6
0.8
1.0
t

CHAPTER

THREE

PYNOISE

3.1 pynoise package

3.1.1 Introduction

This package was created to add noise to signals, which can be matrices or vectors. As of October 2018, only the additive white gaussian noise is implemented.

Additive White Gaussian Noise

Additive White Gaussian Noise (AWGN) is a signal generated from a normal distribution. Hence, the noise energy is spread across all spectrum.

Fig. 3.1 shows the histogram (top) and spectra (bottom) of a typical noise signal.

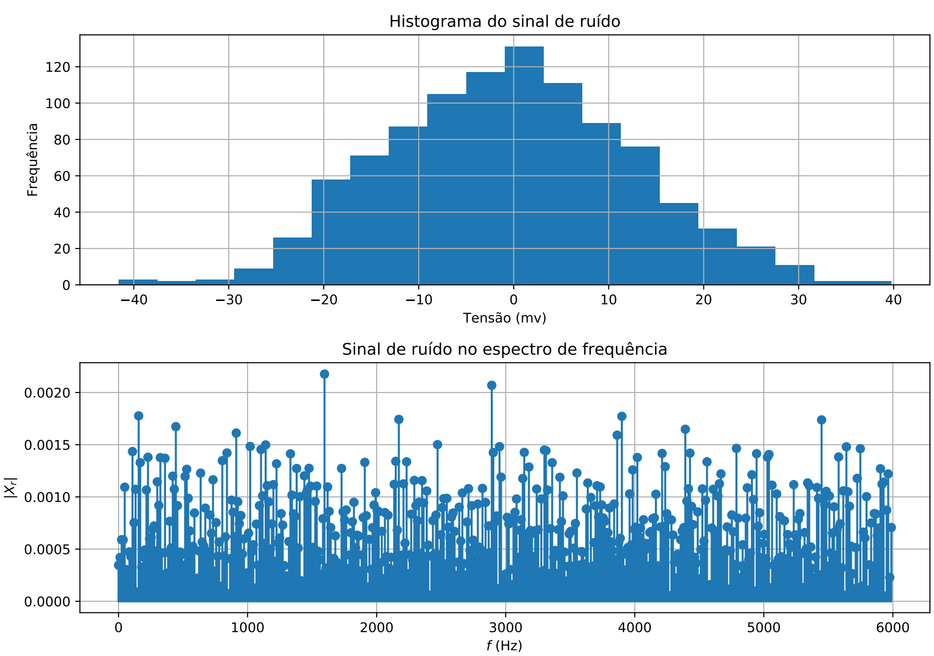


Fig. 3.1: Noise histogram (top) and spectra (bottom).

3.1.2 Submodules

noise module

This module contains functions to work with signals and noise.

`pynoise.noise.awgn` (*x*, *snr*, *out='signal'*, *method='vectorized'*)

Adds White Gaussian Noise to a signal.

The noise level is specified as a Signal-to-Noise Ratio (SNR) value, which relates to signal-to-noise energy or power.

The SNR between a signal x and a noise n is defined as:

$$\mathrm{SNR} = 10\log\left(\frac{E_x}{E_n}\right),$$

where E_x is the energy of the signal x and E_n is the energy of the signal n.

Parameters

- **x** (*np.ndarray*) – Signal, as a vector or column-matrix.
- **snr** (*int, float*) – Signal-to-Noise ration.
- **out** (*str, optional*) – Output data. If 'signal', the signal *x* plus noise is returned. If 'noise', only the noise vector is returned. If 'both', signal with noise and noise only are returned. Any other value defaults to 'signal'.
- **method** (*str, optional*) – Method to compute noise vector (matrix) to be introduced in the signal. In the 'vectorized' method, the matrix energy is computed and used to compute the noise energy. In the 'max_en' method, the energy of each column of *x* is computed and only the highest value is used to compute the noise energy. The 'vectorized' method is used by default.

Returns Corrupted signal.

Return type *np.ndarray*

Raises `ValueErrorExpection` – If *method* is not recognized.

Example

Add 30 dB of white gaussian noise to the ramp signal $x(t) = t$:

```
import numpy as np
import pynoise
import matplotlib.pyplot as plt

# --- Signal ---
t = np.arange(0, 1, 0.01)

x = t
xn = pynoise.awgn(x, 30)

# --- Plots ---
plt.figure(figsize=(10,6))
plt.plot(t, x, label='Original signal')
plt.plot(t, xn,  label='Corrupted signal')
plt.grid()
plt.xlabel('t')
plt.ylabel('x')
plt.legend()
plt.show()
```

CHAPTER

FOUR

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

INDEX