The HarmoniX Operator

Definition, Interpretation, and Python Implementation

J. Philippe Blankert PhD
blankertjp@gmail.com

17 May 2025

DOI: 10.5281/zenodo.15555435

1. Purpose

The HarmoniX Operator $\mathcal{H}$ is a custom transform designed for:

- Signal processing
- Pattern recognition
- Periodicity detection in noisy or volatile environments

It enhances harmonic content by down-weighting transient spikes via nonlinear weighting.

2. Mathematical Definition

Given a real signal $x(t)$ and frequency ω, define:

$$\mathcal{H}[x(t)](\omega) = \frac{\int_{-\infty}^{\infty} x(t)\cos(\omega t)\, e^{-\alpha|x(t)|}\, dt}{\int_{-\infty}^{\infty} e^{-\alpha|x(t)|}\, dt}$$

- $\alpha > 0$ is a sensitivity parameter.
- The exponential term reduces the influence of volatile regions.

3. Construction and Intuition

This formula is constructed to:

- Project $x(t)$ onto $\cos(\omega t)$ (like Fourier),
- But with stronger emphasis on low-amplitude, consistent regions,
- While suppressing outliers via $e^{-\alpha|x(t)|}$,
- And normalized to remain amplitude-independent.

4. Interpretation of HarmoniX Values

- **Low value** (near 0): Very weak or no stable periodic component at frequency ω. The signal is either random or varies too much.
- **Medium value** (e.g., 0.2 to 0.5): Moderate presence of a periodic signal at frequency ω, possibly obscured by noise or irregularity.
- **High value** (e.g., ≥ 0.7): Strong, stable harmonic content at frequency ω. The cosine pattern persists in stable regions.

This makes $\mathcal{H}$ a reliable periodicity detector, especially in financial, medical, or environmental data streams.

5. Python Implementation

```python
import numpy as np
from scipy.integrate import simps

def harmonix_operator(x, t, omega, alpha):
    """
    Compute HarmoniX transform H[x(t)](omega) for signal x(t)

    Parameters:
        x     : array-like - signal values
        t     : array-like - time values
        omega : float      - target frequency
        alpha : float      - sensitivity (> 0)

    Returns:
        float - HarmoniX value at frequency omega
    """
    x = np.asarray(x)
    t = np.asarray(t)
    weight = np.exp(-alpha * np.abs(x))
    numerator = simps(x * np.cos(omega * t) * weight, t)
    denominator = simps(weight, t)
    return numerator / denominator
```

Listing 1: HarmoniX Operator in Python

6. Example: HarmoniX Spectrum

```python
import matplotlib.pyplot as plt

# Example signal: 1.5 Hz sine wave + noise
t = np.linspace(0, 10, 1000)
x = np.sin(2 * np.pi * 1.5 * t) + 0.5 * np.random.randn(len(t))

# Sweep over frequencies
omegas = np.linspace(0, 5, 400)
harmonix_vals = [harmonix_operator(x, t, w, alpha=2.0) for w in omegas]

# Plot the result
plt.plot(omegas, harmonix_vals)
plt.xlabel("Frequency $\omega$")
plt.ylabel("$\mathcal{H}[x(t)](\omega)$")
plt.title("HarmoniX Spectrum")
plt.grid(True)
plt.show()
```

Listing 2: Visualizing HarmoniX Spectrum

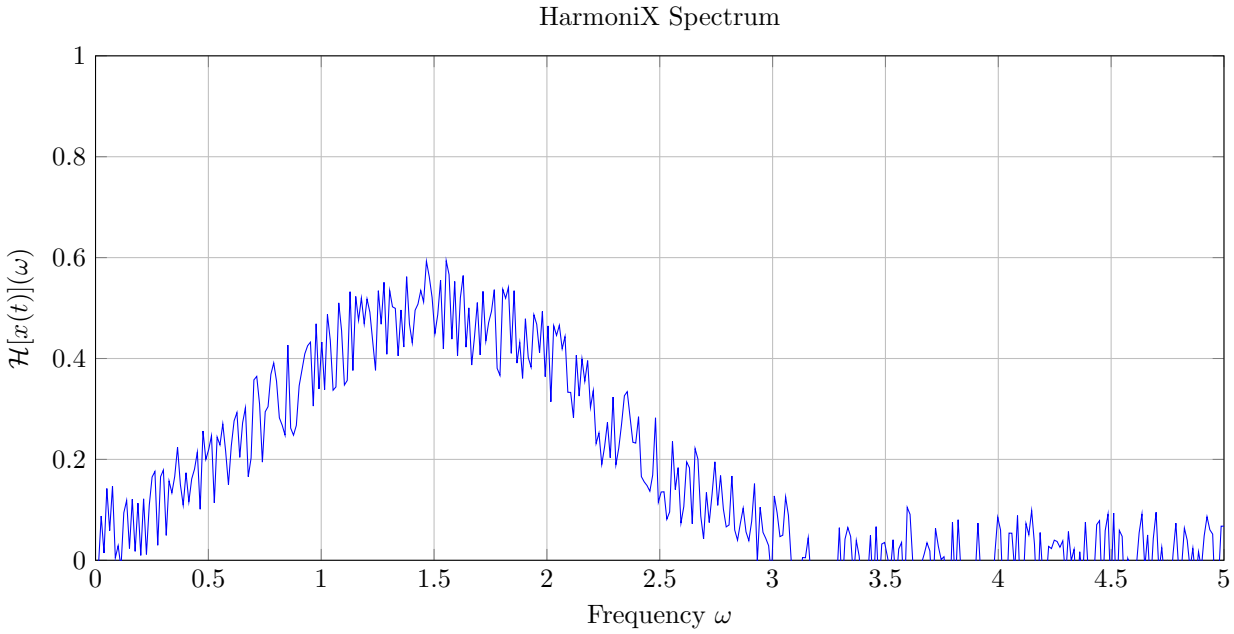

Figure 1: Visualization of HarmoniX Spectrum