Compute And Plot The Convolution Y[n]=x[n]h[n], Wherex[n]=(1/3)^n U[n1] And H[n]=u[n1]

Compute And Plot The Convolution Y[n]=x[n]h[n], Wherex[n]=(1/3)^n U[n1] And H[n]=u[n1]

Introduction

Convolution is a fundamental operation in signal processing, used to determine the output of a system when the input signal and the system's impulse response are known. In discrete-time systems, convolution involves summing the products of shifted and flipped versions of the input and impulse response signals. This article explores the convolution of two discrete signals:

- $x[n] = \left(\frac{1}{3}\right)^n U[n-1]$
- $h[n] = u[n-1]$

where $U[n-1]$ and $u[n-1]$ are unit step functions shifted by 1. We will analyze these signals, compute their convolution both analytically and numerically, and visualize the results through plots.

Understanding the Signals

The Input Signal $x[n]$

- Definition: $x[n] = \left(\frac{1}{3}\right)^n U[n-1]$
- Components:
- $\left(\frac{1}{3}\right)^n$: An exponential decay function.
- $U[n-1]$: A shifted unit step function, which is 0 for $n < 1$ and 1 for $n \geq 1$.

The Impulse Response $h[n]$

- Definition: $h[n] = u[n-1]$
- Components:
- $u[n-1]$: A shifted unit step function, similar to $U[n-1]$, but typically lowercase $u[n]$ denotes the standard unit step.

Signal Support and Behavior

- Both signals are causal, starting at $n=1$.
- $x[n]$ decays exponentially after $n=1$.
- $h[n]$ is a step function starting at $n=1$, remaining 1 thereafter.

Analytical Computation of the Convolution

Convolution Definition

The convolution of two discrete signals $x[n]$ and $h[n]$ is given by:

$$
Y[n] = (x h)[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n - k]
$$

Given the finite support due to the step functions, the summation reduces to relevant indices where the signals are non-zero.

Determining the Limits of the Sum

- Since $x[k] \neq 0$ only for $k \geq 1$,
- And $h[n - k] \neq 0$ only for $n - k \geq 1 \Rightarrow k \leq n - 1$,
- The sum simplifies to:

$$
Y[n] = \sum_{k=1}^{n-1} x[k] \cdot h[n - k]
$$

for $n \geq 2$; for $n < 2$, $Y[n] = 0$.

Expressing $x[k]$ and $h[n-k]$

- $x[k] = \left(\frac{1}{3}\right)^k$ for $k \geq 1$,
- $h[n - k] = u[n - k - 1]$, which is 1 for $n - k \geq 1 \Rightarrow k \leq n - 1$.

Final Expression for $Y[n]$

$$
Y[n] = \sum_{k=1}^{n-1} \left(\frac{1}{3}\right)^k
$$

which is a geometric series.

Summation of the Geometric Series

The sum of a geometric series:

$$
\sum_{k=1}^{m} r^k = r \frac{1 - r^m}{1 - r}
$$

Applying this to our case:

$$
\

$$Y[n] = \sum_{k=1}^{n-1} \left(\frac{1}{3}\right)^k = \frac{\frac{1}{3} \left(1 - \left(\frac{1}{3}\right)^{n-1}\right)}{1 - \frac{1}{3}} = \frac{\frac{1}{3} \left(1 - \left(\frac{1}{3}\right)^{n-1}\right)}{\frac{2}{3}} = \frac{1 - \left(\frac{1}{3}\right)^{n-1}}{2}$$

Final Expression for $Y[n]$

$$
Y[n] =
\begin{cases}
0, & n < 2 \\
\frac{1 - \left(\frac{1}{3}\right)^{n-1}}{2}, & n \geq 2
\end{cases}
$$

Numerical Computation and Plotting

Implementation in Python

To visualize the convolution, we can implement the calculation numerically and generate plots using libraries such as NumPy and Matplotlib.

Step-by-Step Procedure

1. Define the range of n (e.g., from 0 to 20).
2. Generate $x[n]$ and $h[n]$ based on their definitions.
3. Use the convolution function to compute $Y[n]$.
4. Plot the signals $x[n]$, $h[n]$, and $Y[n]$.

Sample Python Code

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

Define the range of n
n = np.arange(0, 21)

Define x[n]
x = np.zeros_like(n, dtype=float)
x[n >= 1] = (1/3) n[n >= 1]

Define h[n]
h = np.zeros_like(n, dtype=float)
h[n >= 1] = 1

Compute the convolution
Y = np.convolve(x, h)
```

```
Adjust the length of Y for plotting
n_conv = np.arange(0, len(Y))

Plot x[n]
plt.stem(n, x, linefmt='b-', markerfmt='bo', basefmt='k-', label='x[n]')
plt.title('Input Signal x[n]')
plt.xlabel('n')
plt.ylabel('Amplitude')
plt.legend()
plt.grid()

plt.show()

Plot h[n]
plt.stem(n, h, linefmt='r-', markerfmt='ro', basefmt='k-', label='h[n]')
plt.title('Impulse Response h[n]')
plt.xlabel('n')
plt.ylabel('Amplitude')
plt.legend()
plt.grid()

plt.show()

Plot Y[n]
plt.stem(n_conv, Y, linefmt='g-', markerfmt='go', basefmt='k-', label='Y[n]')
plt.title('Convolution Result Y[n]')
plt.xlabel('n')
plt.ylabel('Amplitude')
plt.legend()
plt.grid()

plt.show()
```

Interpretation of the Plots

- The input signal $x[n]$ shows an exponential decay starting at $n=1$.
- The impulse response $h[n]$ is a step function starting at $n=1$.
- The convolution $Y[n]$ exhibits a cumulative effect of $x[n]$ passing through $h[n]$, resulting in a smoothed, increasing function that approaches a limit as n increases.

Properties of the Convolution Result

Causality and Support

- $Y[n]$ is causal, starting at $n=2$.
- It exhibits a finite support for practical purposes, as the influence of the exponentially decaying input diminishes over time.

Long-Term Behavior

- As $n \to \infty$:

$$
Y[n] \to \frac{1}{2}
$$

since $\left(\frac{1}{3}\right)^{n-1} \to 0$.

- The convolution approaches a steady-state value, indicating the system's response stabilizes over time.

Significance in Signal Processing

Convolution provides insights into how systems respond to inputs. Here, the exponential decay input passing through a step system results in a cumulative sum that stabilizes. This process models real-world phenomena such as filter responses, system stability, and signal integration.

Summary

In this article, we:

- Analyzed the signals $x[n] = \left(\frac{1}{3}\right)^n U[n-1]$ and $h[n] = u[n-1]$.
- Derived the analytical formula for their convolution $Y[n]$.
- Recognized that the convolution sum reduces to a geometric series.
- Implemented a numerical example to visualize the signals and their convolution.
- Interpreted the behavior and properties of the convolution output.

Understanding the convolution of such signals is crucial in designing and analyzing digital filters and systems, enabling engineers and scientists to predict system responses and optimize performance.

References

- Oppenheim, A. V., & Willsky, A. S. (1997). Signals and Systems. Prentice Hall.
- Lyons, R. G. (2010). Understanding Digital Signal Processing. Pearson

Frequently Asked Questions

What is the main objective when computing and plotting the

convolution y[n] = x[n] h[n]?

The main objective is to determine the output sequence y[n] obtained by convolving the input sequence x[n] with the impulse response h[n], and then visualize the resulting sequence through plotting.

Given x[n] = (1/3)^n U[n-1], what is the support (non-zero range) of x[n]?

Since U[n-1] is a unit step starting at n=1, x[n] is non-zero for n ≥ 1, so the support is n ≥ 1.

What is the form of the impulse response h[n] = u[n-1], and how does it affect the convolution?

h[n] = u[n-1] is a step function starting at n=1, meaning it is 0 for n<1 and 1 for n≥1. This shifts the convolution result, starting the accumulated sum from n=1.

How do you set up the convolution sum for these sequences?

The convolution y[n] = sum_{k=-∞}^{∞} x[k] h[n - k]. Given the supports, the sum simplifies to the ranges where both x[k] and h[n - k] are non-zero, typically for k ≥ 1 and n - k ≥ 1.

What is the explicit expression for y[n] = x[n] h[n] in this case?

For n ≥ 2, y[n] = sum_{k=1}^{n-1} (1/3)^k, since both x[k] and h[n - k] are non-zero only for k ≥ 1 and n - k ≥ 1; for n<2, y[n]=0.

How do you compute the convolution sum for these sequences analytically?

The sum becomes a finite geometric series: y[n] = sum_{k=1}^{n-1} (1/3)^k = [(1/3)(1 - (1/3)^{n-1})] / (1 - 1/3) = (1/3) [1 - (1/3)^{n-1}] / (2/3) = (1/2) [1 - (1/3)^{n-1}].

What is the closed-form expression of y[n] for n ≥ 2?

For n ≥ 2, y[n] = (1/2) [1 - (1/3)^{n-1}]. For n < 2, y[n] = 0.

How would you plot y[n] to visualize the convolution result?

Create a discrete sequence for n starting from 1 or 2, compute y[n] using the closed-form expression, and plot these points to observe how the convolution evolves with increasing n.

What are some common pitfalls to avoid when computing and

plotting convolution in this context?

Common pitfalls include incorrect limits in the summation due to support considerations, forgetting to account for the shift in h[n], and not verifying the sequence supports before plotting. Ensuring the correct range of n and proper handling of zero values is essential.

Additional Resources

Convolution: Unlocking Signal Processing's Core Technique

Introduction

In the realm of digital signal processing (DSP), convolution stands as a fundamental operation, underpinning many applications—from filtering and system analysis to image processing and communications. At its core, convolution combines two signals to produce a third, revealing how one modifies or influences the other. This article delves into a specific convolution scenario, exploring the mathematical intricacies and visual insights of the operation:

$$ Y[n] = x[n] h[n] $$

where

$$
x[n] = \left(\frac{1}{3}\right)^n U[n - 1], \quad h[n] = u[n - 1]
$$

with $U[n - 1]$ and $u[n - 1]$ representing unit step functions shifted by 1.

Through this exploration, we will compute and plot the convolution result, offering an expert-level understanding of the signals involved, the convolution process, and the resulting insights, which are crucial for anyone aiming to master DSP fundamentals.

Signal Definitions and Foundations

Understanding $x[n]$

The signal:

$$
x[n] = \left(\frac{1}{3}\right)^n U[n - 1]
$$

has two key components:

- Exponential Decay $\left(\frac{1}{3}\right)^n$: For $n \ge 0$, this decays exponentially, reflecting a decreasing amplitude as n grows.

- Unit Step $U[n - 1]$: This ensures the signal starts at $n = 1$, meaning:

$$
x[n] = 0 \quad \text{for} \quad n < 1
$$

Thus, explicitly,

$$
x[n] = \begin{cases}
0, & n < 1 \\
\left(\frac{1}{3}\right)^n, & n \ge 1
\end{cases}
$$

Graphically, $x[n]$ is a decreasing exponential starting from $n=1$.

Understanding $h[n]$

The impulse response:

$$
h[n] = u[n - 1]
$$

is a shifted unit step function:

$$
h[n] = \begin{cases}
0, & n < 1 \\
1, & n \ge 1
\end{cases}
$$

This is a simple step function that turns on at $n=1$.

Significance of the Signals

- $x[n]$: Represents a causal, exponentially decaying input starting from $n=1$.
- $h[n]$: Acts as a causal step function, "turning on" at $n=1$.

Their convolution, $Y[n]$, effectively computes the output of a system with impulse response $h[n]$ when driven by input $x[n]$.

Mathematical Approach to Convolution

The Convolution Sum

The discrete-time convolution is defined as:

$$
Y[n] = (x\ h)[n] = \sum_{k=-\infty}^{\infty} x[k] \, h[n - k]
$$

Given the signals' causal nature (non-zero only for $n \ge 1$), the limits simplify considerably. Specifically:

- $x[k] \neq 0$ only for $k \ge 1$,
- $h[n - k] \neq 0$ only for $n - k \ge 1 \Rightarrow k \le n - 1$.

Thus, the summation reduces to:

$$
Y[n] = \sum_{k=1}^{n-1} x[k] \, h[n - k]
$$

for $n \ge 2$. For $n < 2$, the sum is empty, and $Y[n] = 0$.

Simplifying the Sum

Since $h[n - k] = u[n - k - 1]$, which is 1 when $n - k \ge 1$, and 0 otherwise, the sum becomes:

$$
Y[n] = \sum_{k=1}^{n-1} x[k] \times 1 = \sum_{k=1}^{n-1} \left(\frac{1}{3}\right)^k
$$

for $n \ge 2$.

Final Expression

The convolution reduces to a finite geometric series:

$$
Y[n] = \sum_{k=1}^{n-1} \left(\frac{1}{3}\right)^k
$$

for $n \ge 2$, with $Y[n] = 0$ for $n < 2$.

This sum can be evaluated explicitly using the geometric series formula:

$$
\sum_{k=1}^{m} r^k = r \frac{1 - r^{m}}{1 - r}
$$

where $r = \frac{1}{3}$.

Thus,

$$
Y[n] = \frac{1/3 \times (1 - (1/3)^{n-1})}{1 - 1/3} = \frac{\frac{1}{3}(1 - (1/3)^{n-1})}{\frac{2}{3}} = \frac{1}{2} \left(1 - \left(\frac{1}{3}\right)^{n-1} \right)
$$

\]

for \(n \ge 2 \). For \(n < 2 \), \(Y[n] = 0 \).

Visualizing the Convolution: Step-by-Step Plotting

1. Plotting \(x[n] \)

- Range: \(n \) from 0 to 10 for clarity.
- Behavior: Zero for \(n < 1 \), exponential decay starting at \(n=1 \).

2. Plotting \(h[n] \)

- Range: \(n \) from 0 to 10.
- Behavior: Zero for \(n < 1 \), then constant 1.

3. Plotting \(Y[n] \)

- Range: \(n \) from 1 to 10.
- Behavior: Starts from 0 at \(n=1 \), then approaches 0.5 asymptotically.

Analytical and Numerical Validation

The explicit formula:

\[
Y[n] = \frac{1}{2} \left(1 - \left(\frac{1}{3}\right)^{n-1} \right), \quad n \ge 2
\]

provides an accurate benchmark for numerical calculations and plotting.

Key observations:

- As \(n \to \infty \), \(\left(\frac{1}{3}\right)^{n-1} \to 0 \), so \(Y[n] \to \frac{1}{2} \).
- The convolution sum converges to 0.5, reflecting the steady-state response of the system to the input.

Practical Implementation: Step-by-Step Computing and Plotting

Using Python and Matplotlib

Here's a sample code snippet that computes and plots the signals:

```python
import numpy as np
```

```
import matplotlib.pyplot as plt

Define range
n = np.arange(0, 15)

Define x[n]
x = np.zeros_like(n)
x[n >= 1] = (1/3) n[n >= 1]

Define h[n]
h = np.zeros_like(n)
h[n >= 1] = 1

Compute convolution manually
Y = np.zeros_like(n)
for i in range(len(n)):
for k in range(1, i):
if k >= 0 and (i - k) >= 1:
Y[i] += (1/3) k

Alternatively, use the explicit formula for n >= 2
Y_analytic = np.zeros_like(n, dtype=float)
for i in range(2, len(n)):
Y_analytic[i] = 0.5 (1 - (1/3) (i - 1))

Plotting
plt.figure(figsize=(12, 8))

plt.subplot(3, 1, 1)
plt.stem(n, x, basefmt=' ')
plt.title('Input Signal x[n]')
plt.xlabel('n')
plt.ylabel('x[n]')
plt.grid()

plt.subplot(3, 1, 2)
plt.stem(n, h, basefmt=' ')
plt.title('Impulse Response h[n]')
plt.xlabel('n')
plt.ylabel('h[n]')
plt.grid()

plt.subplot(3, 1, 3)
plt.stem(n, Y, basefmt=' ', label='Numerical Convolution')
plt.plot(n, Y_analytic, 'r--', label='Analytical Formula')
plt.title('Convolution Result Y[n]')
plt.xlabel('n')
plt.ylabel('Y[n]')
plt.legend()
plt.grid()
```

```
plt.tight_layout()
plt.show()
```

This code computes the convolution via double summation and validates it against the explicit formula, providing a comprehensive visual understanding.

Real-World Applications and Significance

Understanding this convolution process exemplifies how systems respond to exponential inputs, which are prevalent in many engineering contexts:

- Filtering: Exponentially decaying signals often model transient responses.
- System Analysis:

Compute And Plot The Convolution Y N X N H N Wherex N 1 3 N U N1 And H N U N1

Compute And Plot The Convolution Y N X N H N Wherex N 1 3 N U N1 And H N U N1

Related Articles

- If K.E. Of An Object Moving With Velocity 4 M/s Is 72 J. Calculate Mass Of The Object.
- What Is An Object's Mass If It Accelerates At 5 M/s2 When A Force Of 0.5 N Is Applied?
- Determine If The Following Vectors Are Collinear: = (3,5, -2) And B = [12, -20,8]

Back to Home