Signal Processing and Matched Filter Output

Name: Sachin Kuamr
Roll No: 220104026
Section: D
Lab No. : 4

October 19, 2024

Objective

The objective of this lab is to analyze signal processing concepts such as unit step functions, exponential signals, and matched filter output using convolution. We will plot the time-domain signals and calculate the matched filter output.

Introduction

Matched filters are widely used in signal processing to detect known signals in noisy environments. This lab focuses on implementing and visualizing unit step functions, exponential signals, and computing the matched filter output using convolution.

Code Implementation

Below is the Python code that implements the signal generation and matched filter computation.

```

import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import convolve

# Generate an array of values from -6 to 6
t = np.linspace(-6, 6, 400)
```

```
# Define unit step function u(t) for t >= 0
u = np.where(t >= 0, 1, 0)

# Define the function p(t) = exp(-2t) * u(t)
p = np.exp(-2 * t) * u

# Define unit step function u(-t) for t <= 0
u_neg = np.where(t <= 0, 1, 0)

# Define the function p(-t) = exp(2t) * u(-t)
p_neg = np.exp(2 * t) * u_neg

# Plot p(t)
plt.figure(figsize=(10, 6))
plt.subplot(3, 1, 1)
plt.plot(t, p, label='p(t)')
plt.title('p(t)')
plt.grid(True)

# Plot p(-t)
plt.subplot(3, 1, 2)
plt.plot(t, p_neg, label='p(-t)')
plt.title('p(-t)')
plt.grid(True)

# Define h(t) = p(T-t) * u(t) for T = 2
T = 2
u_T_t = np.where(t <= T, 1, 0)
h = np.exp(-2 * (T - t)) * u_T_t

# Plot h(t)
plt.subplot(3, 1, 3)
plt.plot(t, h, label='h(t) = p(T-t)*u(t)')
plt.title('h(t)=p(T-t)*u(t)')
plt.xlim(-7, 7)
plt.grid(True)
plt.tight_layout()
plt.show()

# Perform convolution for matched filter output
dt = t[1] - t[0]  # Time step
matched_filter_output = convolve(p, p_neg, mode='same') * dt
```

```
# Plot the matched filter output
plt.figure(figsize=(8, 5))
plt.plot(t, matched_filter_output, label='Matched Filter Output')
plt.title('Matched Filter Output')
plt.xlabel('t')
plt.ylabel('Output')
plt.grid(True)
plt.legend()
plt.show()
```

Results

The following figures show the signal plots and the matched filter output:

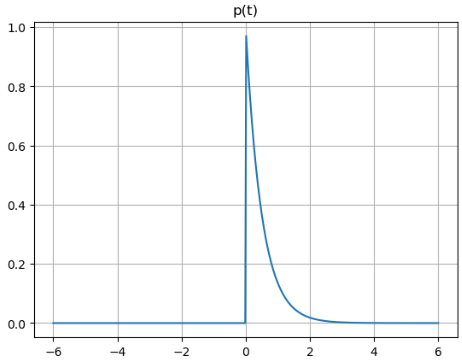

Figure 1: Plot of $p(t)$

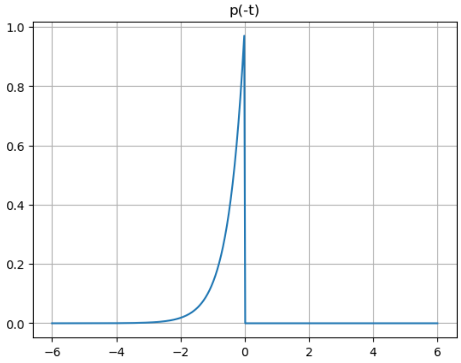

Figure 2: Plot of $p(-t)$

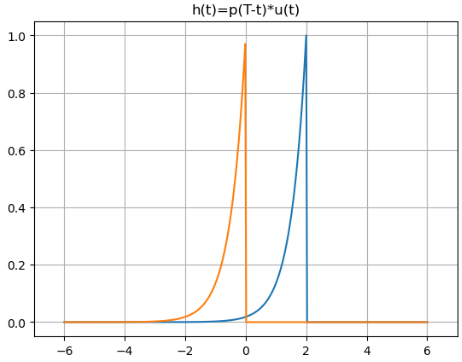

Figure 3: Plot of $h(t) = p(T - t) * u(t)$

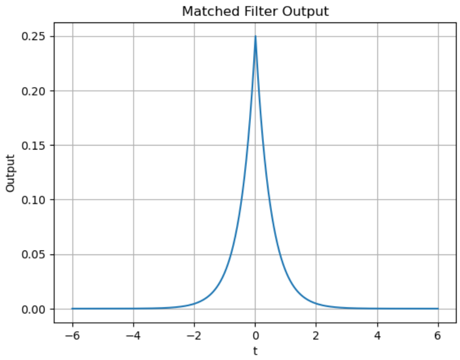

Figure 4: Matched Filter Output