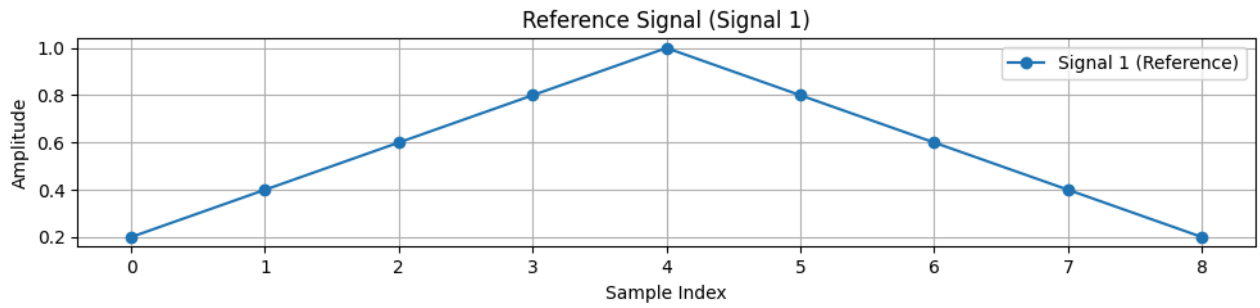
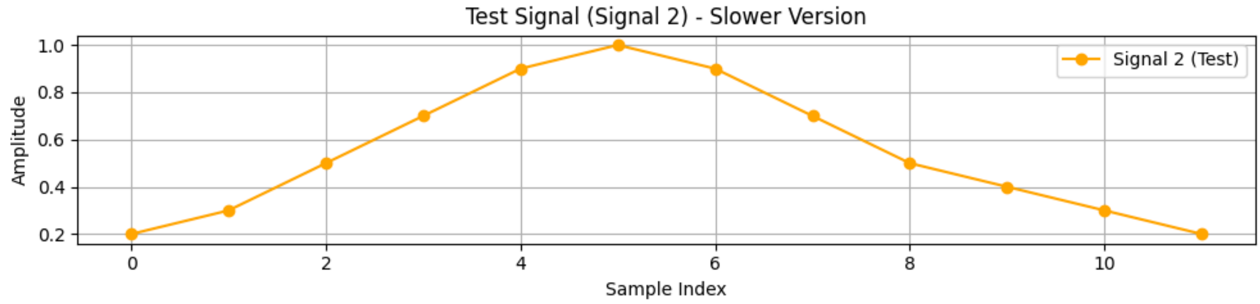
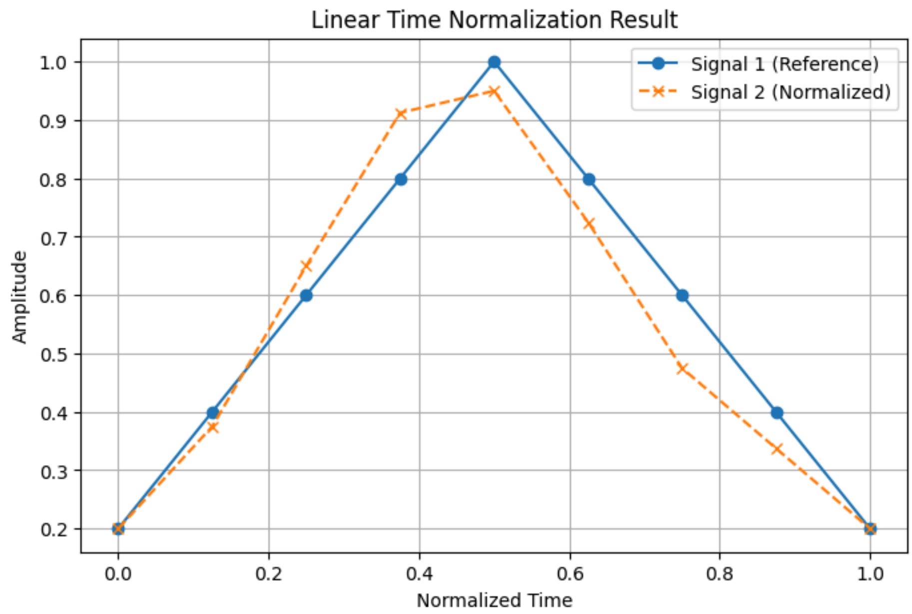
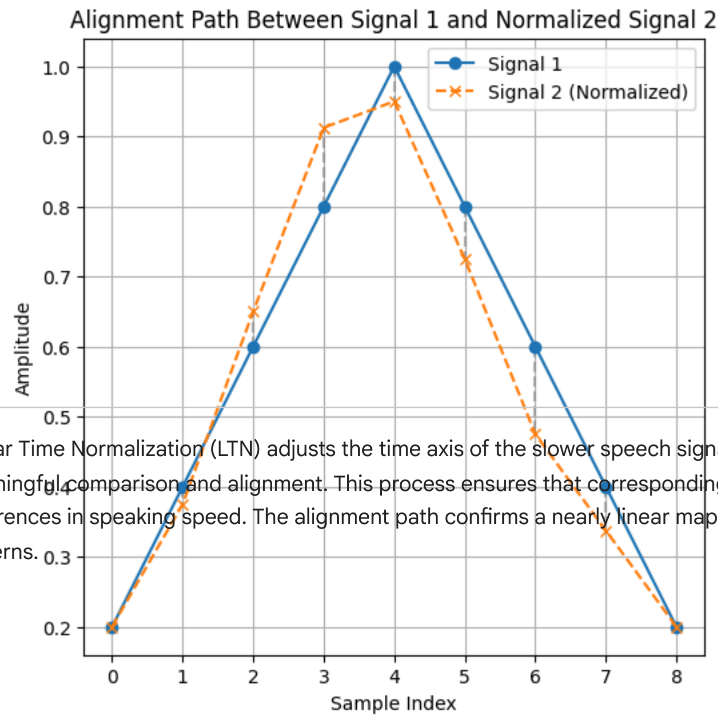
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.interpolate import interp1d

# Given Data
signal1 = np.array([0.2, 0.4, 0.6, 0.8, 1.0, 0.8, 0.6, 0.4, 0.2])  # Reference signal
signal2 = np.array([0.2, 0.3, 0.5, 0.7, 0.9, 1.0, 0.9, 0.7, 0.5, 0.4, 0.3, 0.2])  # Test signal (slower)

# Step 1: Plot both speech signals (original)
plt.figure(figsize=(10,5))
plt.subplot(2,1,1)
plt.plot(signal1, marker='o', label='Signal 1 (Reference)')
plt.title("Reference Signal (Signal 1)")
plt.xlabel("Sample Index")
plt.ylabel("Amplitude")
plt.legend()
plt.grid(True)

plt.subplot(2,1,2)
plt.plot(signal2, marker='o', color='orange', label='Signal 2 (Test)')
plt.title("Test Signal (Signal 2) - Slower Version")
plt.xlabel("Sample Index")
plt.ylabel("Amplitude")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

# Step 2: Perform Linear Time Normalization (Resample Signal 2)

# Create normalized time axes
t1 = np.linspace(0, 1, len(signal1))  # time axis for Signal 1
t2 = np.linspace(0, 1, len(signal2))  # time axis for Signal 2

# Interpolation function to resample Signal 2
interp_func = interp1d(t2, signal2, kind='linear')
signal2_normalized = interp_func(t1)  # normalized Signal 2 (same length as Signal 1)

# Step 3: Plot normalized Signal 2 vs Signal 1
plt.figure(figsize=(8,5))
plt.plot(t1, signal1, 'o-', label='Signal 1 (Reference)')
plt.plot(t1, signal2_normalized, 'x--', label='Signal 2 (Normalized)')
plt.title("Linear Time Normalization Result")
plt.xlabel("Normalized Time")
plt.ylabel("Amplitude")
plt.legend()
plt.grid(True)
plt.show()


# Step 4: Compute Alignment (Index Correspondence)
alignment_indices = list(zip(range(len(signal1)), range(len(signal2_normalized))))

# Plot alignment path
plt.figure(figsize=(6,6))
for i, j in alignment_indices:
    plt.plot([i, j], [signal1[i], signal2_normalized[j]], color='gray', linestyle='--', alpha=0.6)

plt.plot(range(len(signal1)), signal1, 'o-', label='Signal 1')
plt.plot(range(len(signal2_normalized)), signal2_normalized, 'x--', label='Signal 2 (Normalized)')
plt.title("Alignment Path Between Signal 1 and Normalized Signal 2")
plt.xlabel("Sample Index")
plt.ylabel("Amplitude")
plt.legend()
plt.grid(True)
plt.show()
```









Linear Time Normalization (LTN) adjusts the time axis of the slower speech signal so that both signals have equal lengths, enabling meaningful comparison and alignment. This process ensures that corresponding parts of the speech waveform align in time, despite differences in speaking speed. The alignment path confirms a nearly linear mapping, indicating effective normalization of temporal patterns.