# -\*- coding: utf-8 -\*-

"""

Created on Sat Sep 6 10:34:43 2025

@author: seshu

"""

import numpy as np

def time\_shift(signal, k):

"""

Shifts the signal by k units.

Positive k shifts the signal to the right, negative k shifts the signal to the left.

"""

return np.roll(signal, k)

def time\_scale(signal, k):

"""

Scales the time axis of the signal by a factor of k.

If k > 1, the signal is stretched; if 0 < k < 1, the signal is compressed.

"""

n = len(signal)

scaled\_signal = np.interp(np.linspace(0, n-1, int(n/k)), np.arange(n), signal)

return scaled\_signal

def signal\_addition(signal1, signal2):

"""

Performs element-wise addition of two signals.

"""

if len(signal1) != len(signal2):

raise ValueError("Signals must have the same length")

return signal1 + signal2

def signal\_multiplication(signal1, signal2):

"""

Performs element-wise multiplication of two signals.

"""

if len(signal1) != len(signal2):

raise ValueError("Signals must have the same length")

return signal1 \* signal2

# Example usage:

if \_\_name\_\_ == "\_\_main\_\_":

t = np.linspace(0, 10, 1000)

signal1 = np.sin(t)

signal2 = np.cos(t)

shifted\_signal = time\_shift(signal1, 100)

scaled\_signal = time\_scale(signal1, 2)

added\_signal = signal\_addition(signal1, signal2)

multiplied\_signal = signal\_multiplication(signal1, signal2)

print("Shifted Signal: ", shifted\_signal[:10])

print("Scaled Signal: ", scaled\_signal[:10])

print("Added Signal: ", added\_signal[:10])

print("Multiplied Signal: ", multiplied\_signal[:10])