Digital version

ENGS254 Signals and Systems Lab
Linear Time-Invariant Systems

Mher Saribekyan A09210183

February 18, 2026

What is it?

Linear systems are systems that satisfy two conditions: additivity and homegeneity:

$$y(x_1(t) + x_2(t)) = y(x_1(t)) + y(x_2(t))$$

$$y(c \cdot x(t)) = c \cdot y(x(t))$$

Time-invariant systems are systems that do not change the output, if the input is delayed, and they depend only on the input data and their past state.

The problem

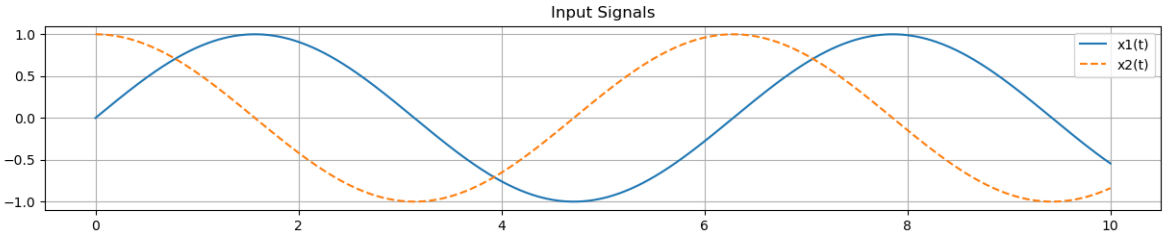
We are given three systems: $A(t) = 2 \cdot x(t)$, $B(t) = [x(t)]^2$ and $C(t) = x(t) \cdot \sin(t)$, and we need to find which of those systems are linear time-invariant systems. To achive this goal, several experiments were carried out with the help of the Python programming language.

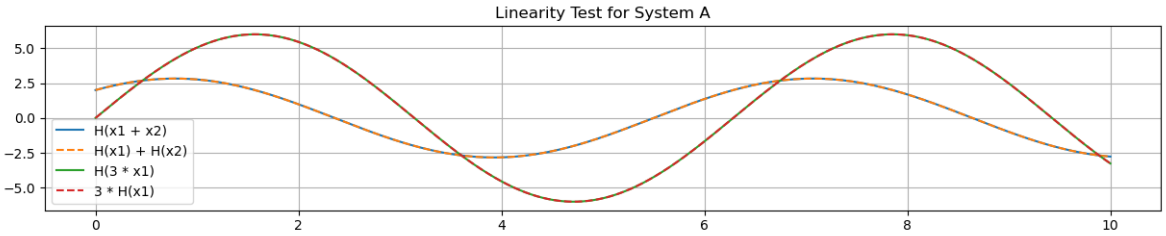
Linearity

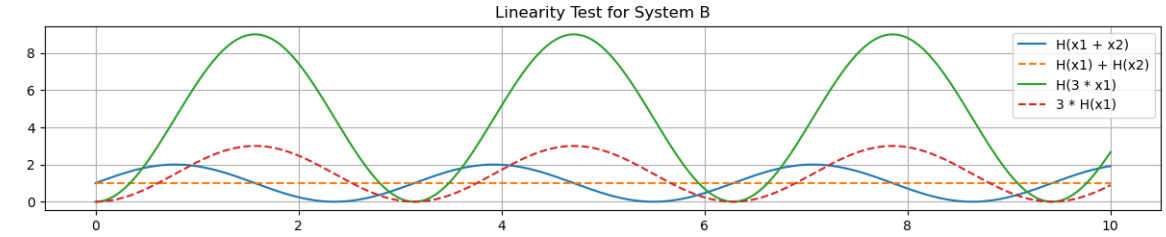
We have two properties for linear systems: additivity and homogeneity. Two functions were declared: $x_1(t) = \sin(t)$ and $x_2(t) = \cos(t)$, and were used to test linearity for systems A and B.

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

t = np.linspace(0, 10, 1001)
x1 = np.sin(t)
x2 = np.cos(t)

def system_A(x):
    return 2 * x

def system_B(x):
    return x**2

# Test Linearity: H(x1+x2) vs H(x1)+H(x2)
out_combined_A = system_A(x1 + x2)
```

```
out_separate_A = system_A(x1) + system_A(x2)

# Test Linearity: C * H(x1) vs C * H(x1)
out_scaled_input_A = system_A(3 * x1)
out_scaled_output_A = 3 * system_A(x1)

if np.allclose(out_combined_A, out_separate_A) and np.allclose(out_scaled_input_A,
                                                out_scaled_output_A):
    print("System A is linear.")
else:
    print("System A is not linear.")

out_combined_B = system_B(x1 + x2)
out_separate_B = system_B(x1) + system_B(x2)

out_scaled_input_B = system_B(3 * x1)
out_scaled_output_B = 3 * system_B(x1)

if np.allclose(out_combined_B, out_separate_B) and np.allclose(out_scaled_input_B,
                                                out_scaled_output_B):
    print("System B is linear.")
else:
    print("System B is not linear.")


# This Section is just for plotting the signals and the results of the linearity tests
                                                for both systems.

fig, (ax1, ax2, ax3) = plt.subplots(3, 1, figsize=(10, 4))

ax1.plot(t, x1, label="x1(t)")
ax1.plot(t, x2, '--', label="x2(t)")

ax2.plot(t, out_combined_A, label="H(x1 + x2)")
ax2.plot(t, out_separate_A, '--', label="H(x1) + H(x2)")
ax2.plot(t, out_scaled_input_A, label="H(3 * x1)")
ax2.plot(t, out_scaled_output_A, '--', label="3 * H(x1)")

ax3.plot(t, out_combined_B, label="H(x1 + x2)")
ax3.plot(t, out_separate_B, '--', label="H(x1) + H(x2)")
ax3.plot(t, out_scaled_input_B, label="H(3 * x1)")
ax3.plot(t, out_scaled_output_B, '--', label="3 * H(x1)")

ax1.legend()
ax1.set_title("Input Signals")
ax1.grid()

ax2.legend()
ax2.set_title("Linearity Test for System A")
ax2.grid()

ax3.legend()
ax3.set_title("Linearity Test for System B")
ax3.grid()

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







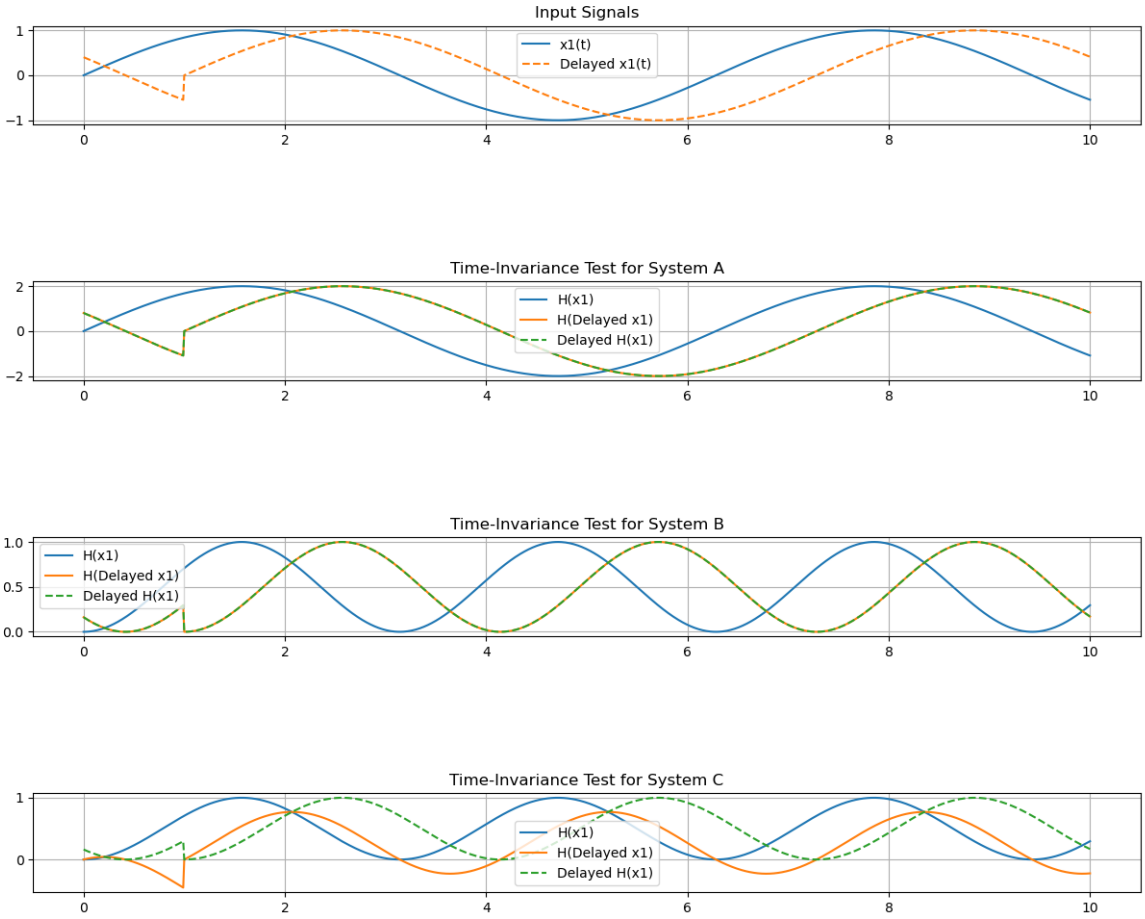
As we can see, for system A, both the additivity and homogeneity properties apply, however with system B, we see that both properties do not apply, as the plot of the system $B(x_1 + x_2)$ is not the same as the plot of $B(x_1) + B(x_2)$. We can conclude that System A is **linear** while System B is **not lienar**.

Time-Invariance

To test time-invariance, we can give a bit of delay to the function, then pass it through the system and compare the result to the function passed to the system, then delayed. The delay was achieved with the roll funciton in Python.

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

t = np.linspace(0, 10, 1001)
x1 = np.sin(t)

def system_A(x):
    return 2 * x
```

```
def system_B(x):
    return x**2

def system_C(x, t_vec):
    return x * np.sin(t_vec)

# 1. Shift input then system
x_delayed = np.roll(x1, 100) # Shift by 100 samples

A_1 = system_A(x_delayed)
B_1 = system_B(x_delayed)
C_1 = system_C(x_delayed, t)

# 2. System then shift output
A_x = system_A(x1)
B_x = system_B(x1)
C_x = system_C(x1, t)

A_2 = np.roll(A_x, 100)
B_2 = np.roll(B_x, 100)
C_2 = np.roll(C_x, 100)

# 3. Compare out1 and out2. If they are different, system is Non-TI.
fig, (ax1, ax2, ax3, ax4) = plt.subplots(4, 1, figsize=(10, 4))

ax1.plot(t, x1, label="x1(t)")
ax1.plot(t, x_delayed, '--', label="Delayed x1(t)")

ax2.plot(t, A_x, label="H(x1)")
ax2.plot(t, A_1, label="H(Delayed x1)")
ax2.plot(t, A_2, '--', label="Delayed H(x1)")

ax3.plot(t, B_x, label="H(x1)")
ax3.plot(t, B_1, label="H(Delayed x1)")
ax3.plot(t, B_2, '--', label="Delayed H(x1)")

ax4.plot(t, C_x, label="H(x1)")
ax4.plot(t, C_1, label="H(Delayed x1)")
ax4.plot(t, C_2, '--', label="Delayed H(x1)")

ax1.legend()
ax1.set_title("Input Signals")
ax1.grid()

ax2.legend()
ax2.set_title("Time-Invariance Test for System A")
ax2.grid()

ax3.legend()
ax3.set_title("Time-Invariance Test for System B")
ax3.grid()

ax4.legend()
ax4.set_title("Time-Invariance Test for System C")
ax4.grid()

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



From the plots, we clearly see that systems A and B are time invariant, as both delayed functions are the same, while for system C, the functions do not match, therefore it is not time-invariant.

Conclusions

System A is linear and time-invariant. System B is non-linear time-invariant. System C is not time-invariant.