### Understanding Time Invariance and Time Shifting in Systems

\*\*Brief:\*\*

Time invariance in a system means that shifting the input signal results in an identical shift in the output without altering its form. Time shifting involves applying a transformation to a time-shifted input and comparing it with the directly computed output.

### MATLAB Code Examples

#### Example: Checking Time Invariance

\*\*Original System:\*\*

\[ y(t) = t \cdot e^{-t} \cdot x(t) \]

\*\*Script:\*\*

```matlab

Clc; clear; close all;

% Define time range

T = -2:0.01:10;

T0 = 2;

% Define input signal x(t)

X = heaviside(t) – heaviside(t – 5);

% Compute original system output y(t)

Y = t .\* exp(-t) .\* x;

% Shift input signal x(t – t0)

X\_shifted = heaviside(t – t0) – heaviside(t – t0 – 5);

% Compute system output for shifted input y’(t)

Y\_shifted\_input = t .\* exp(-t) .\* x\_shifted;

% Compute time-shifted output y(t – t0)

Y\_time\_shifted = (t – t0) .\* exp(-(t – t0)) .\* x\_shifted;

% Plot results

Figure;

Subplot(3,1,1);

Plot(t, y, ‘b’, ‘LineWidth’, 1.5);

Title(‘Original Output y(t)’);

Xlabel(‘t’); ylabel(‘y(t)’);

Grid on;

Subplot(3,1,2);

Plot(t, y\_shifted\_input, ‘r’, ‘LineWidth’, 1.5);

Title(‘Output of Shifted Input y’’(t)’);

Xlabel(‘t’); ylabel(‘y’’(t)’);

Grid on;

Subplot(3,1,3);

Plot(t, y\_time\_shifted, ‘g’, ‘LineWidth’, 1.5);

Title(‘Time-Shifted Output y(t-t0)’);

Xlabel(‘t’); ylabel(‘y(t-t0)’);

Grid on;

% Check if y\_shifted\_input == y\_time\_shifted

If isequal(y\_shifted\_input, y\_time\_shifted)

Disp(‘The system is Time-Invariant.’);

Else

Disp(‘The system is Time-Variant.’);

End

```

### Understanding Linearity in Systems

\*\*Brief:\*\*

A system is linear if it satisfies both additivity (the response to a sum of inputs is the sum of their individual responses) and homogeneity (scaling the input scales the output by the same factor).

### MATLAB Code Examples

#### Example: Checking Linearity in Discrete-Time

\*\*System 1: Exponential Transformation (Nonlinear)\*\*

\[ y[n] = 2^{x[n]} \]

\*\*System 2: Multiplication with Index (Linear)\*\*

\[ y[n] = n \cdot x[n] \]

\*\*Script:\*\*

```matlab

Clc; clear; close all;

% Define discrete time range

N = 0:5;

X1 = 0.8 .\* n;

X2 = cos(n);

A1 = 2;

A2 = 3;

X\_combined = a1\*x1 + a2\*x2;

% System 1: Exponential Transformation (Nonlinear)

Y1\_combined = 2.^x\_combined;

Y1\_individual = a1\*2.^x1 + a2\*2.^x2;

Figure;

Subplot(2,1,1);

Stem(n, y1\_combined, ‘r’, ‘filled’); grid on;

Title(‘System 1: Applying y = 2^x on Combined Input’);

Subplot(2,1,2);

Stem(n, y1\_individual, ‘b—‘, ‘filled’); grid on;

Title(‘System 1: Applying y = 2^x on Scaled Inputs then Adding’);

% System 2: Multiplication with Index (Linear)

Y2\_combined = n .\* x\_combined;

Y2\_individual = a1 \* (n .\* x1) + a2 \* (n .\* x2);

Figure;

Subplot(2,1,1);

Stem(n, y2\_combined, ‘r’, ‘filled’); grid on;

Title(‘System 2: Applying y = n\*x on Combined Input’);

Subplot(2,1,2);

Stem(n, y2\_individual, ‘b—‘, ‘filled’); grid on;

Title(‘System 2: Applying y = n\*x on Scaled Inputs then Adding’);

```

### Understanding Static and Dynamic Signals

\*\*Brief:\*\*

Static signals do not change with time or remain proportional to the input signal without any transformation. Dynamic signals undergo time-dependent modifications such as shifts or delays.

### MATLAB Code Examples

#### Example: Static Signal

\*\*Script:\*\*

```matlab

Clc; clear; close all;

T = -5:0.01:5;

X = heaviside(t) – heaviside(t-1);

% Static transformation (Scaling)

Y\_1 = 3 \* x;

Subplot(311);

Plot(t, x);

Title(‘Original Signal x(t)’);

Subplot(312);

Plot(t, y\_1);

Title(‘Scaled Signal y\_1(t) = 3x(t)’);

```

#### Example: Dynamic Signal

\*\*Script:\*\*

```matlab

% Dynamic transformation (Time Shift)

Y\_2 = heaviside(t-1) – heaviside(t-2);

Subplot(313);

Plot(t, y\_2);

Title(‘Shifted Signal y\_2(t) = x(t-1)’);

```

### Understanding Time Shifting in Systems

\*\*Brief:\*\*

Time shifting involves applying a transformation to a time-shifted input and comparing it with the directly computed output to determine if a system is time-invariant or time-variant.

### MATLAB Code Example

\*\*Script:\*\*

```matlab

Clc; clear; close all;

% Define time range

T = -2:0.01:10;

T0 = 2;

% Define input signal x(t)

X = heaviside(t) – heaviside(t – 5);

% Compute original system output y(t)

Y = t .\* exp(-t) .\* x;

% Shift input signal x(t – t0)

X\_shifted = heaviside(t – t0) – heaviside(t – t0 – 5);

% Compute system output for shifted input y’(t)

Y\_shifted\_input = t .\* exp(-t) .\* x\_shifted;

% Compute time-shifted output y(t – t0)

Y\_time\_shifted = (t – t0) .\* exp(-(t – t0)) .\* x\_shifted;

% Plot results

Figure;

Subplot(3,1,1);

Plot(t, y, ‘b’, ‘LineWidth’, 1.5);

Title(‘Original Output y(t)’);

Xlabel(‘t’); ylabel(‘y(t)’);

Grid on;

Subplot(3,1,2);

Plot(t, y\_shifted\_input, ‘r’, ‘LineWidth’, 1.5);

Title(‘Output of Shifted Input y’’(t)’);

Xlabel(‘t’); ylabel(‘y’’(t)’);

Grid on;

Subplot(3,1,3);

Plot(t, y\_time\_shifted, ‘g’, ‘LineWidth’, 1.5);

Title(‘Time-Shifted Output y(t-t0)’);

Xlabel(‘t’); ylabel(‘y(t-t0)’);

Grid on;

% Check if y\_shifted\_input == y\_time\_shifted

If isequal(y\_shifted\_input, y\_time\_shifted)

Disp(‘The system is Time-Invariant.’);

Else

Disp(‘The system is Time-Variant.’);

End

```

By analyzing these simplified examples, you can better understand the key concepts of time invariance, linearity, and the distinction between static and dynamic signals in signal processing systems.