

# Linear Time- Invariant System Analysis

RESPONSE TO ARBITRARY SIGNAL AND FREQUENCY RESPONSE

# Outline

- ▶ Purpose
- ▶ Tool
- ▶ Method
- ▶ GUI
- ▶ Analysis
- ▶ Demo
- ▶ Review

# Purpose

- ▶ Linear Time-Invariant System
  - ▶  $(A+B)(t) = A(t) + B(t)$  --Linear
  - ▶  $A(t) \rightarrow B(t)$  &&  $A(t-1) \rightarrow B(t-1)$  --Time-Invariant
- ▶ Response to Arbitrary Signal
- ▶ Frequency Response

# Tool

- ▶ IDE

- ▶ Eclipse IDE for Java Developers
  - ▶ Neon Release (4.6.0)

- ▶ Programming Language

- ▶ Java SE Development Kit 8u121



# Method

- ▶ Response to Arbitrary Signal
  - ▶ Convolution :  $y[n] = (x * h)[n]$
- ▶ Frequency Response
  - ▶ Basic input :  $x[n] = \sin(\omega n)$
  - ▶ Compare with Response  $y[n]$ 
    - ▶ Amplitude =  $\max(y[n]), n=0, 1, 2, \dots, \text{MAX}$
    - ▶ Phase =  $\omega * \Delta n$

# GUI

- ▶ Input & System state area
- ▶ Ts setting textfield
- ▶ Output type setting area
- ▶ Confirm area

The screenshot shows a MATLAB-style GUI window titled "System Analysis". The window contains several input fields and checkboxes. A red rectangle highlights the top section containing the input signal equation  $x(t) = 1.0 \cdot \sin(2.0 \cdot t + 1.57)$  and the system difference equation  $y[n] = 0.2 \cdot y[n-1] + 2.0 \cdot x[n-2]$ , each with a "set" button. A blue rectangle highlights the sampling time  $T_s$  field set to 0.02 seconds. A purple rectangle highlights the output type settings, which include checkboxes for "Original Signal", "Signal After Processing", "Amplitude Freq-response", and "Phase Freq-response". A green rectangle highlights the bottom section with "Clear" and "Submit" buttons.

System Analysis	
Input : $x(t) = 1.0 \cdot \sin(2.0 \cdot t + 1.57)$	set
System : $y[n] = 0.2 \cdot y[n-1] + 2.0 \cdot x[n-2]$	set
$T_s$ : 0.02	sec
<input checked="" type="checkbox"/> Original Signal	<input checked="" type="checkbox"/> Signal After Processing
<input checked="" type="checkbox"/> Amplitude Freq-response	<input type="checkbox"/> Phase Freq-response
Clear	Submit

# GUI(cont')

**System Analysis**

Input :  $x(t) = 1.0 \sin(2.0 \cdot t + 1.57)$  set

System :  $y[n] = 0.2 \cdot y[n-1] + 2.0 \cdot x[n-2]$  set

Ts : 0.02 sec

☒ Original Signal ☒ Signal After Processing

☒ Amplitude Freq-response ☐ Phase Freq-response

Clear Submit

**Set DT System Input**

$x(t) = 1.0 \sin(2.0 \cdot t + 1.57)$

A	W	phi
1	2	1.57

Update Delete

Clear Set

**Set DT System Difference Equation**

$y[n] = 0.2 \cdot y[n-1] + 2.0 \cdot x[n-2]$

I/O	delay	coefficient
y	1	0.2

Update Delete

Clear Set

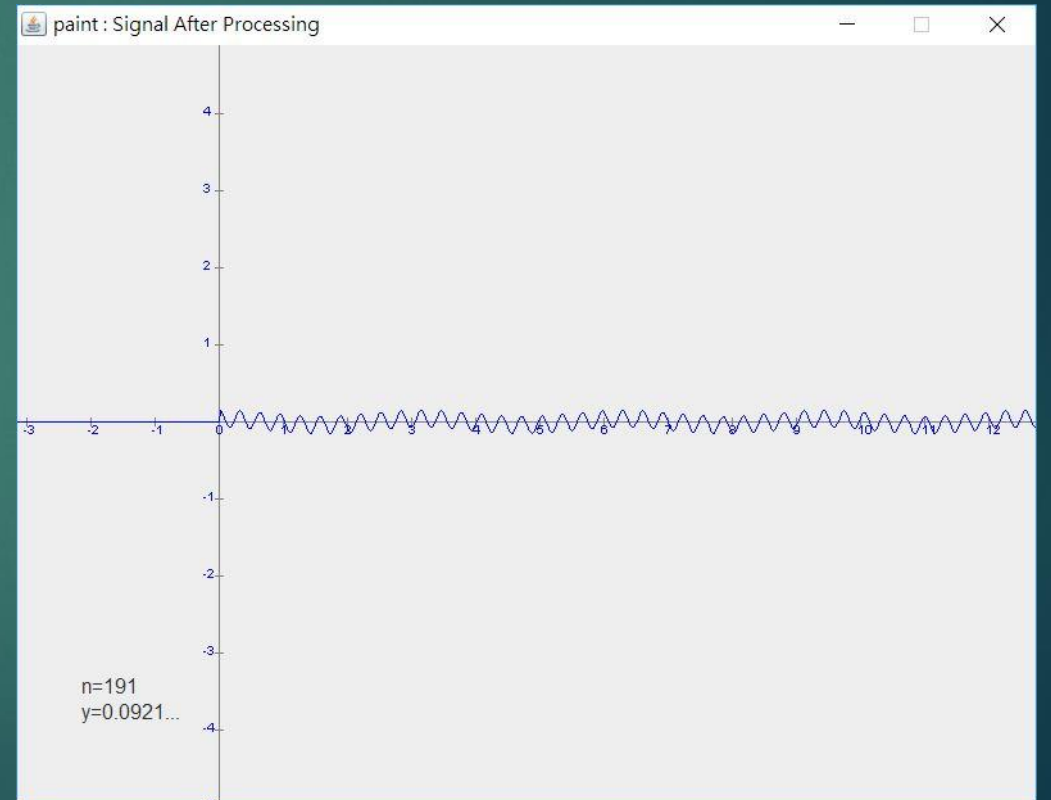
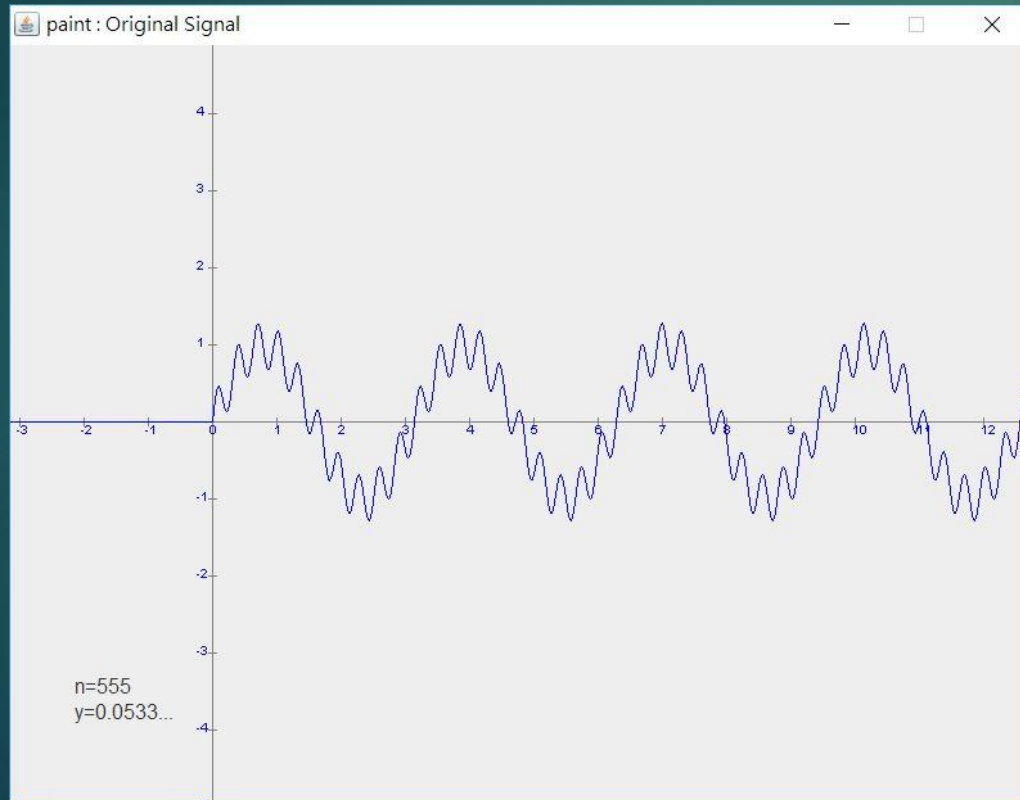
# Analysis

- ▶ High-pass filter
- ▶ Low-pass filter
- ▶ Change  $T_s$



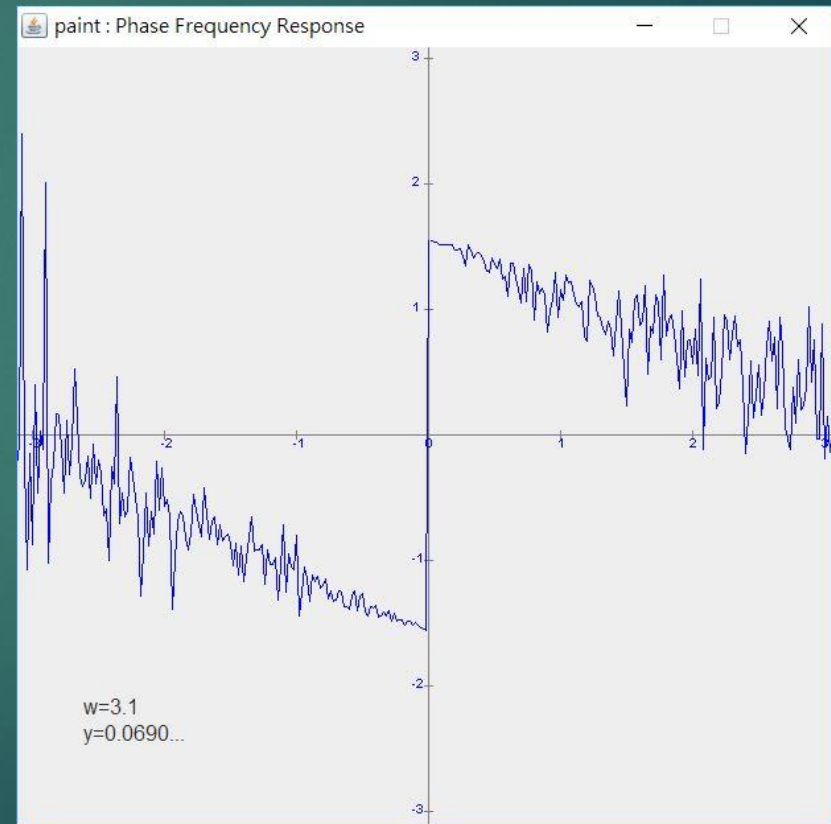
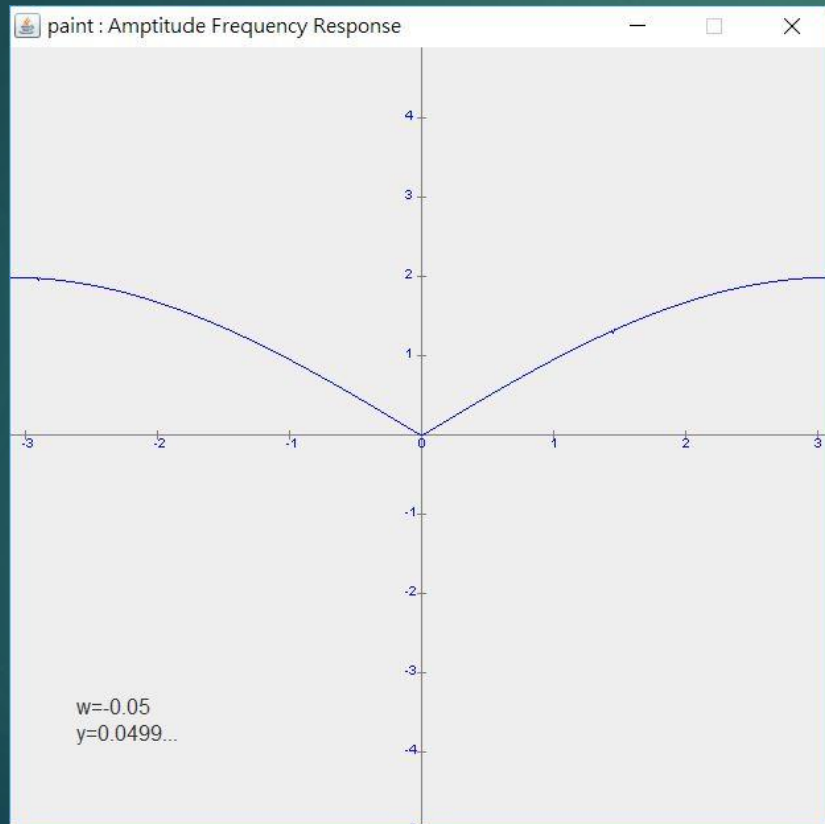
# Analysis(cont')

## ► High-pass filter



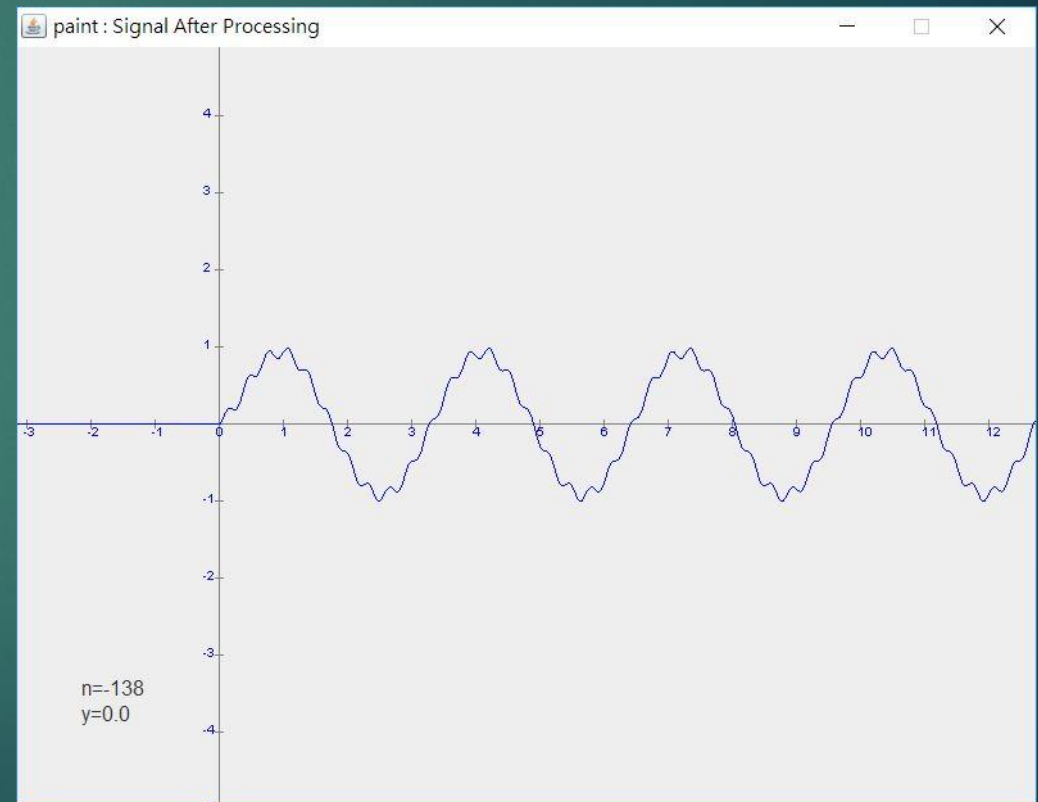
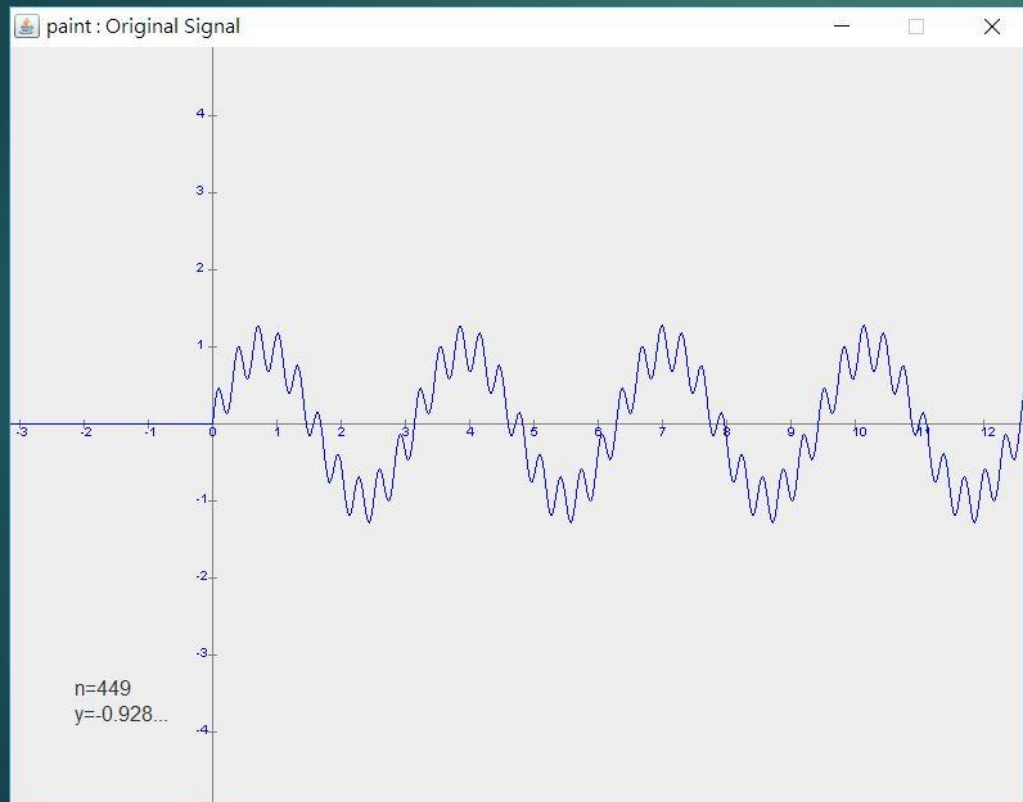
# Analysis(cont')

## ► High-pass filter



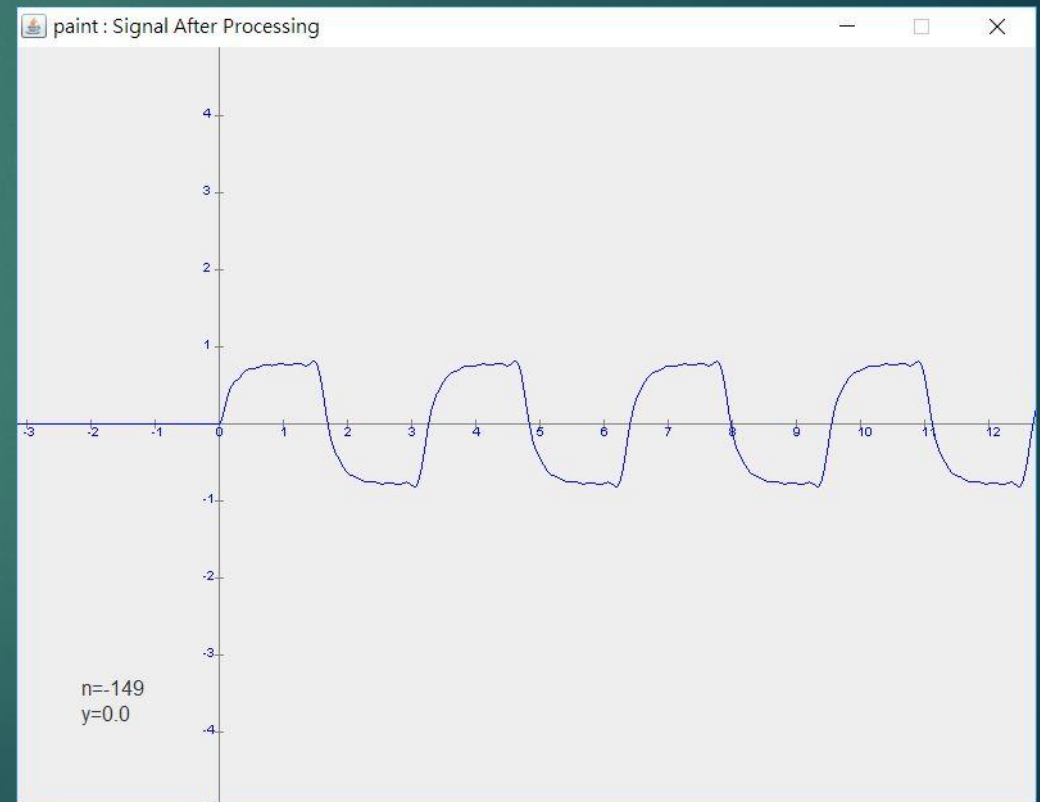
# Analysis(cont')

## ► Low-pass filter



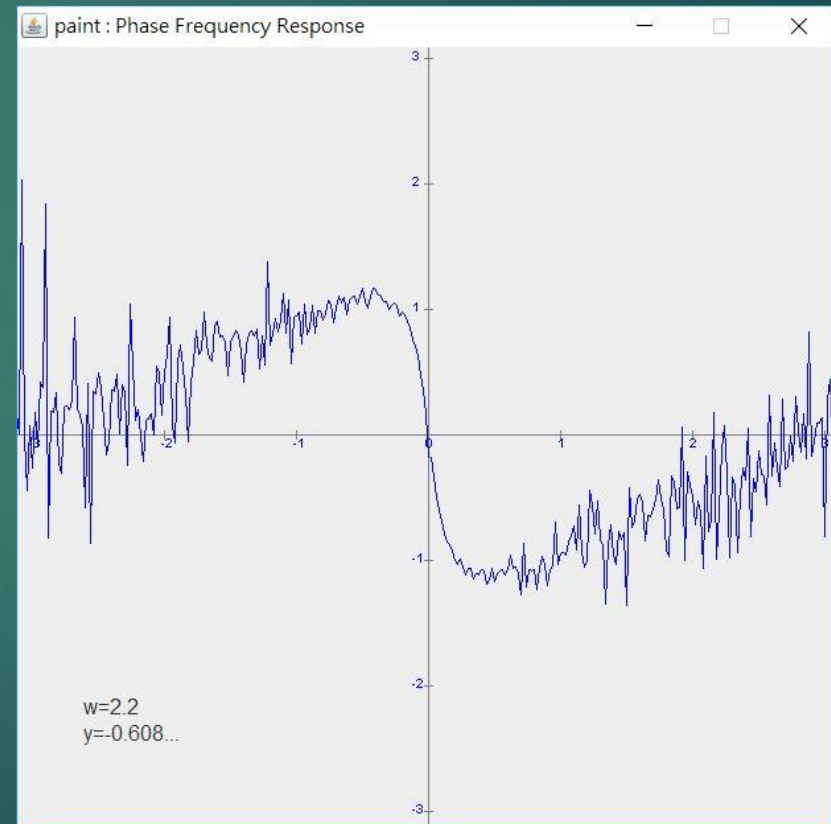
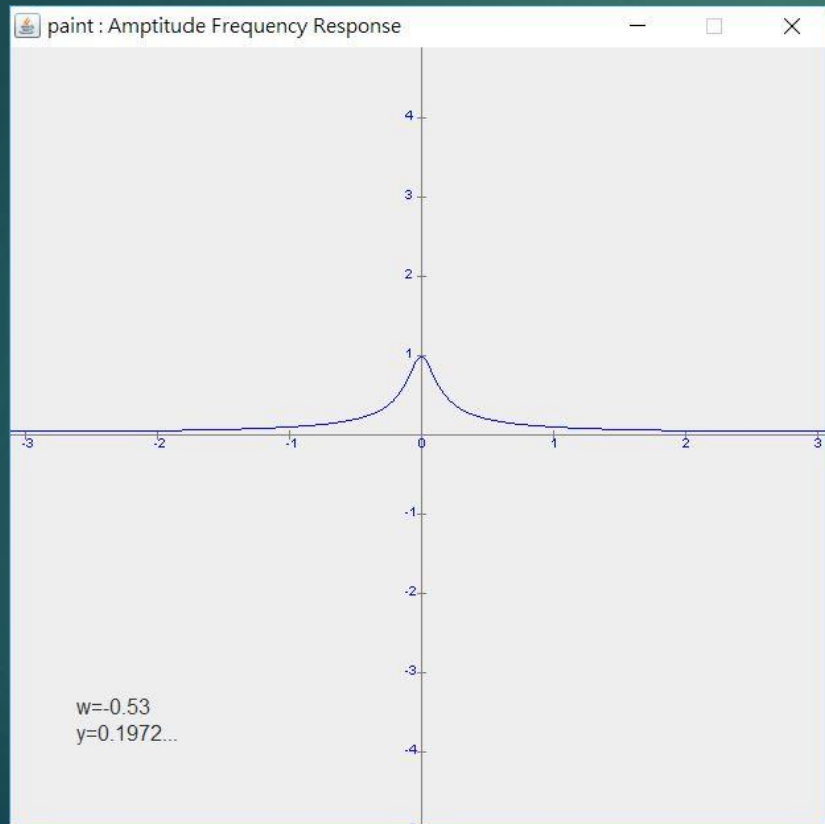
# Analysis(cont')

## ► Low-pass filter



# Analysis(cont')

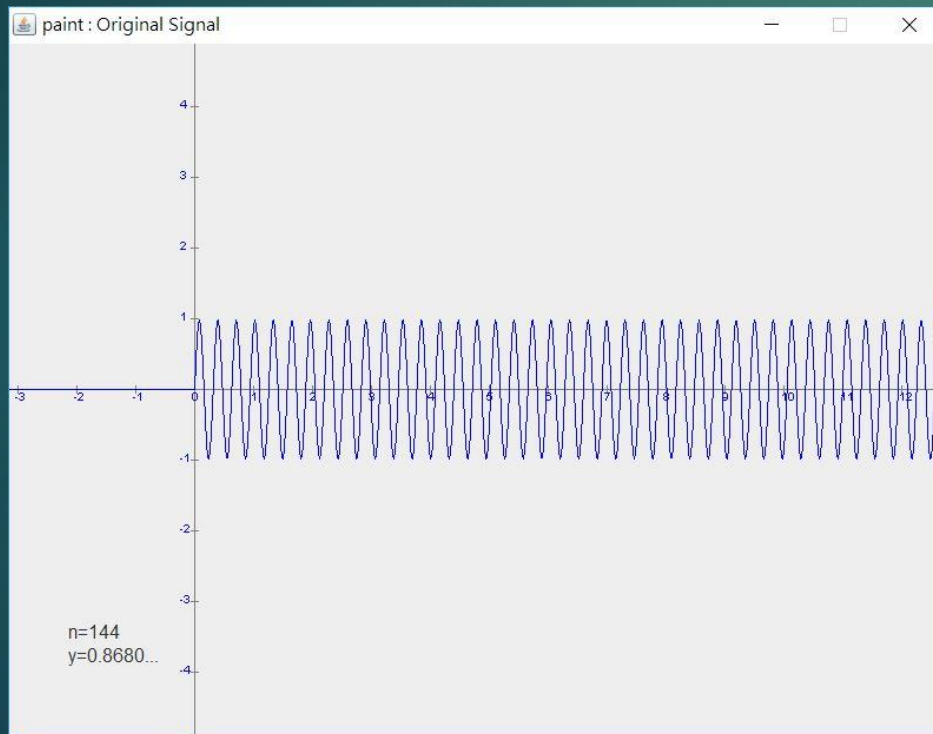
## ► Low-pass filter



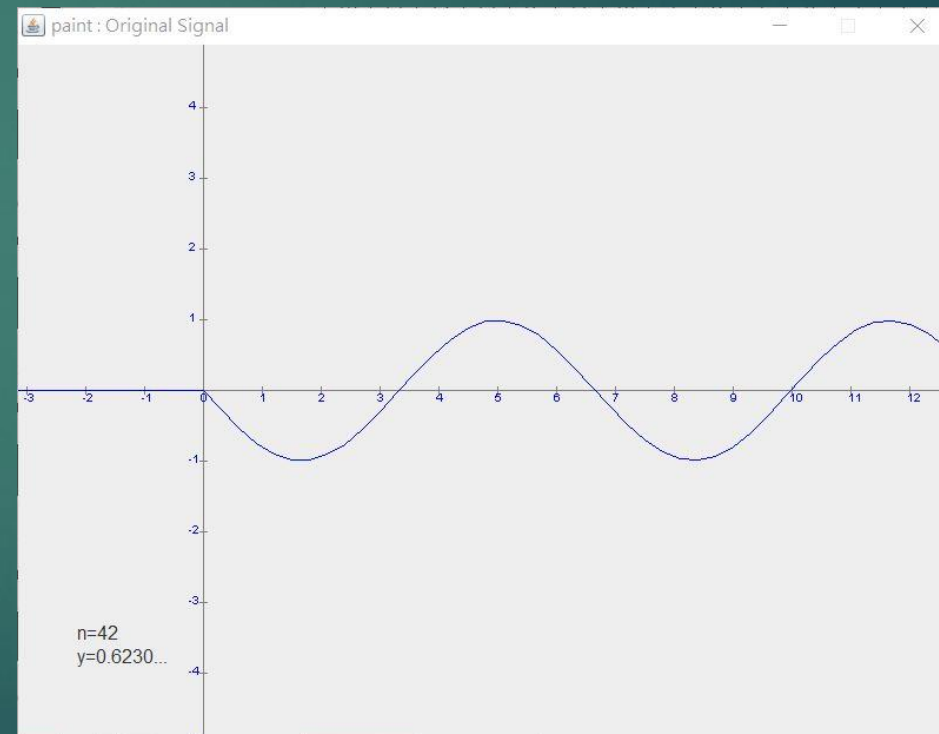
# Analysis(cont')

Input:  $x(t) = 1.0 * \sin(20.0 * t)$  set

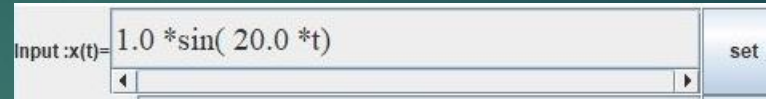
►  $T_s = 0.02$



►  $T_s = 0.3$



# Analysis(cont')



- ▶ Depend on Nyquist-Shannon sampling theorem
  - ▶ Sampling time  $< \frac{1}{2} \cdot T = \frac{\pi}{\omega}$
- ▶  $T_s < \frac{\pi}{20} = 0.157\text{sec}$



# Demo



# Review

- ▶ 此次專案製作期間，進一步使用了GUI中較為進階的部分，如：子母視窗切換及傳值等等。使我學習到更多GUI的設計技巧。
- ▶ 此外，計算相位差的部分，在實作時發現因為離散的因素而導致計算上會產生極大的誤差，這點目前可以想到的解決方式，大概僅能依靠 $|H(z)|$ 及 $\angle H(z)$ ，並利用圖形去求出相關數值。
- ▶ 總體來說，藉由此次的專案，不僅能看見波形在各種系統下的變化，還讓我更加了解信號處理的基本過程，使得我受益良多。