**EC5011**

**DIGITAL SIGNAL PROCESSING THEORY AND APPLICATION**

BANDARA H.G.T.D.

2022/E/048

GROUP CG04

SEMESTER 5

01 MAY 2025

**PART1: SAMPLING, TIME DOMAIN & FREQUENCY DOMAIN REPRESENTATION**

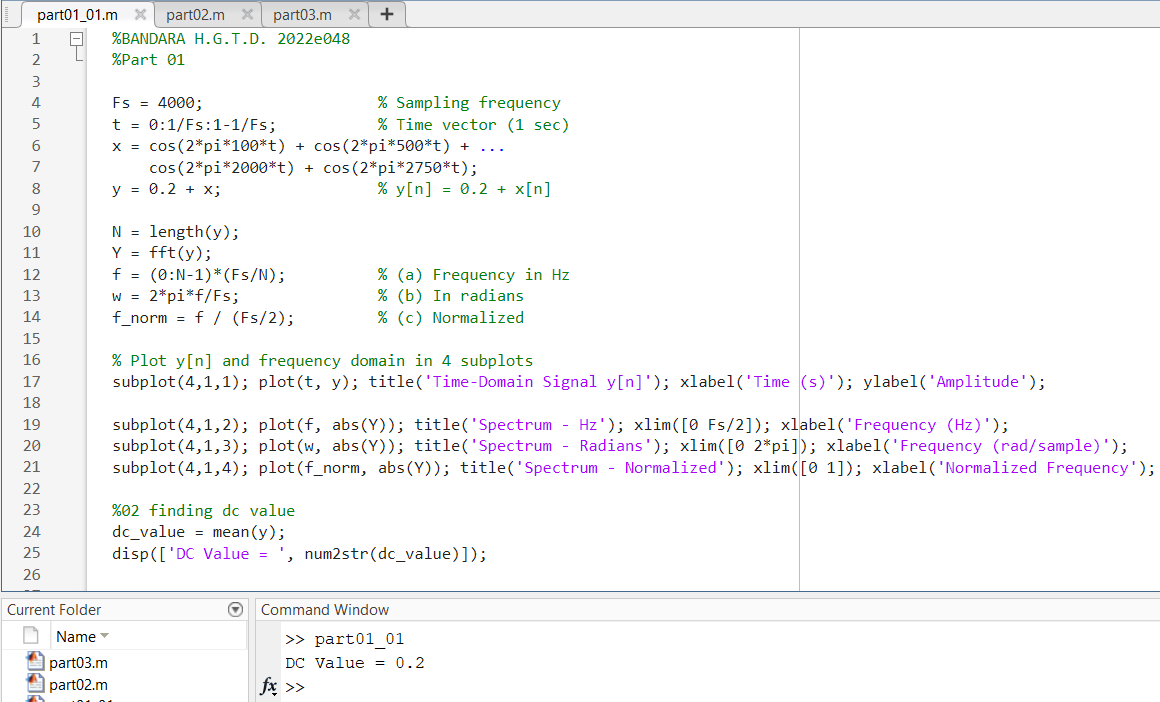
****

Figure 01:Matlab Code for SAMPLING, TIME DOMAIN & FREQUENCY DOMAIN REPRESENTATION

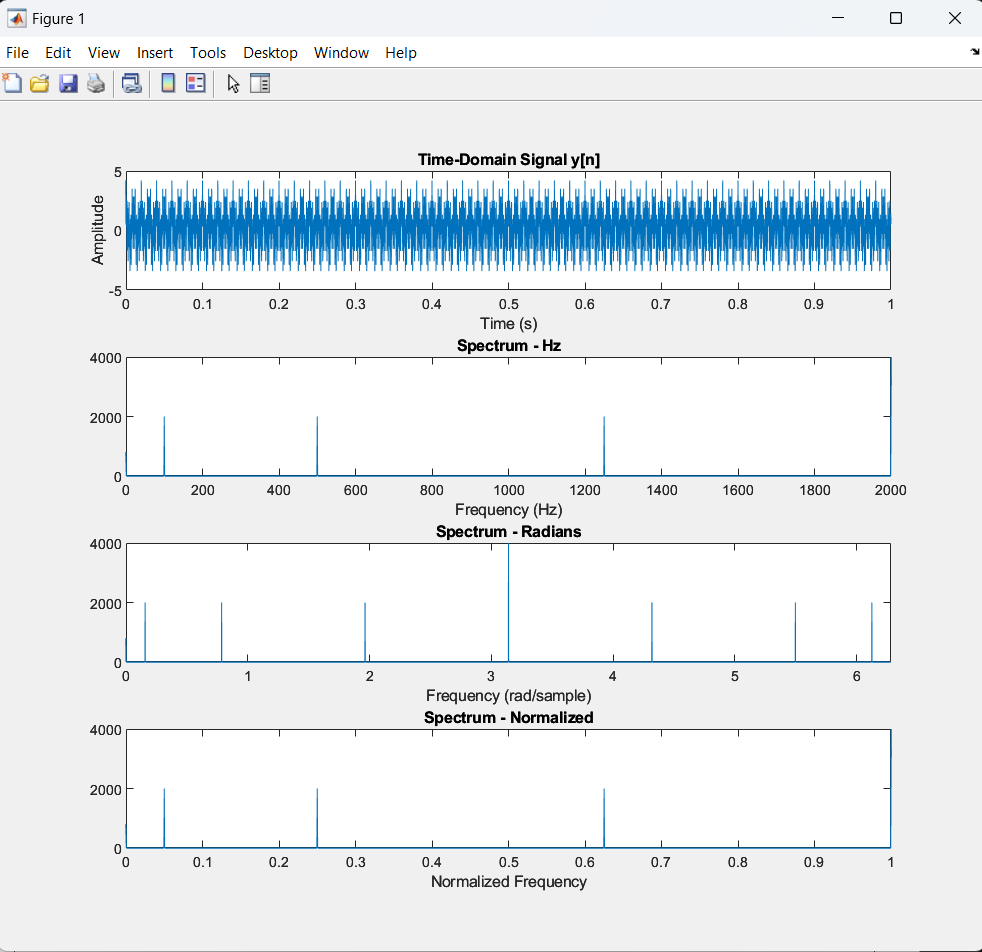
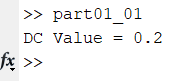


Figure 02:Output

2)

DC value of y[n]=0.2+x[n] is:

DC=0.2+mean of x[n]



3)Aliased frequency components =2750 Hz

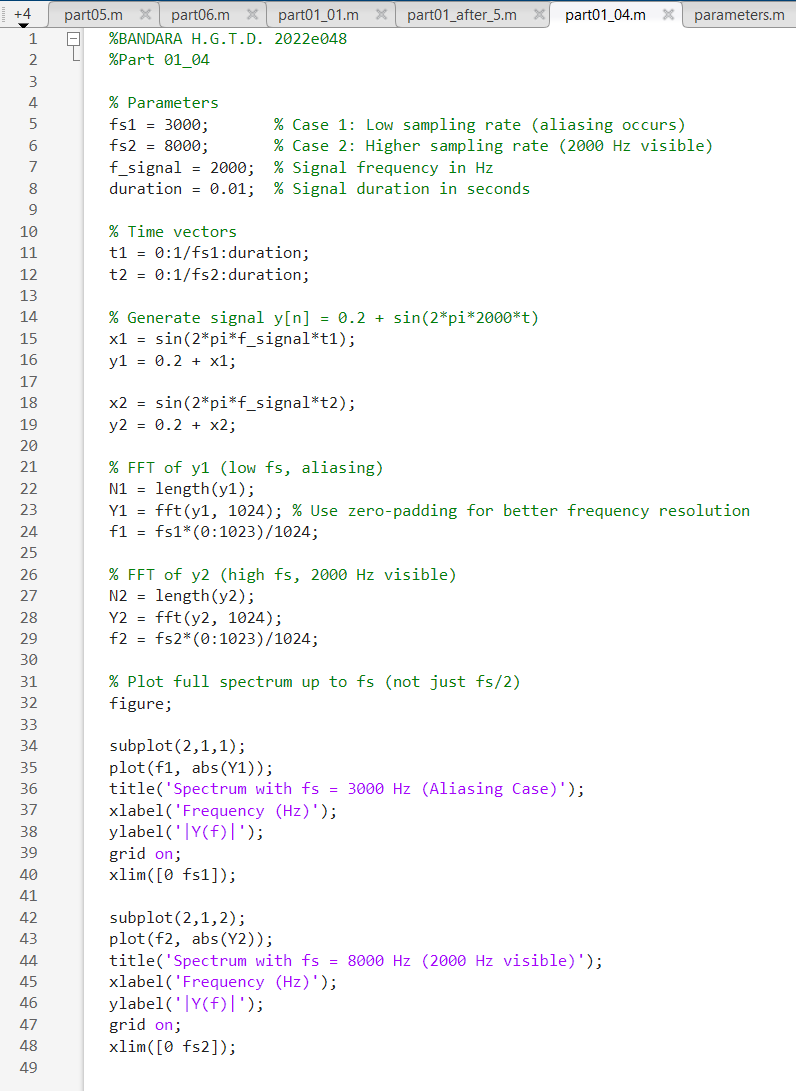
 4)

FIGURE 03: MATLAB CODE

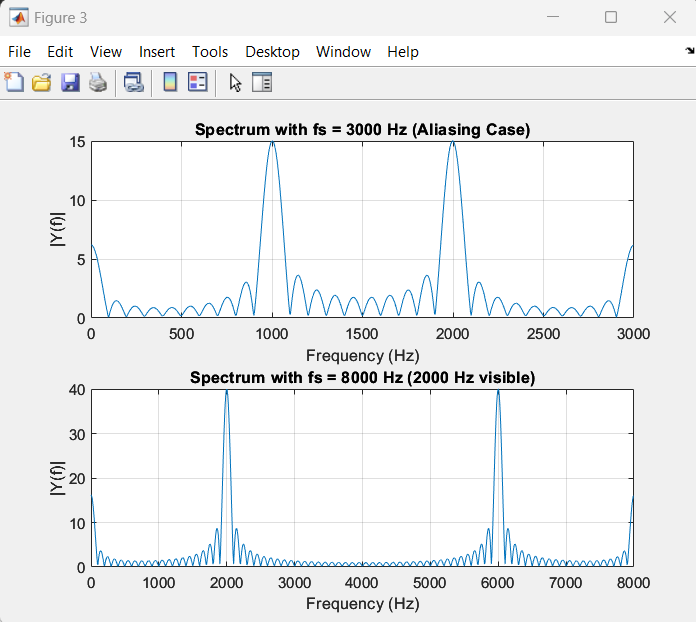


FIGURE04: OUTPUT

The 2000Hz frequency component is not detectable because it aligns with the folding frequency at 2000Hz, causing it to remain unnoticed. Thus, to integrate this frequency into the spectrum, the sampling frequency must surpass 4000Hz.

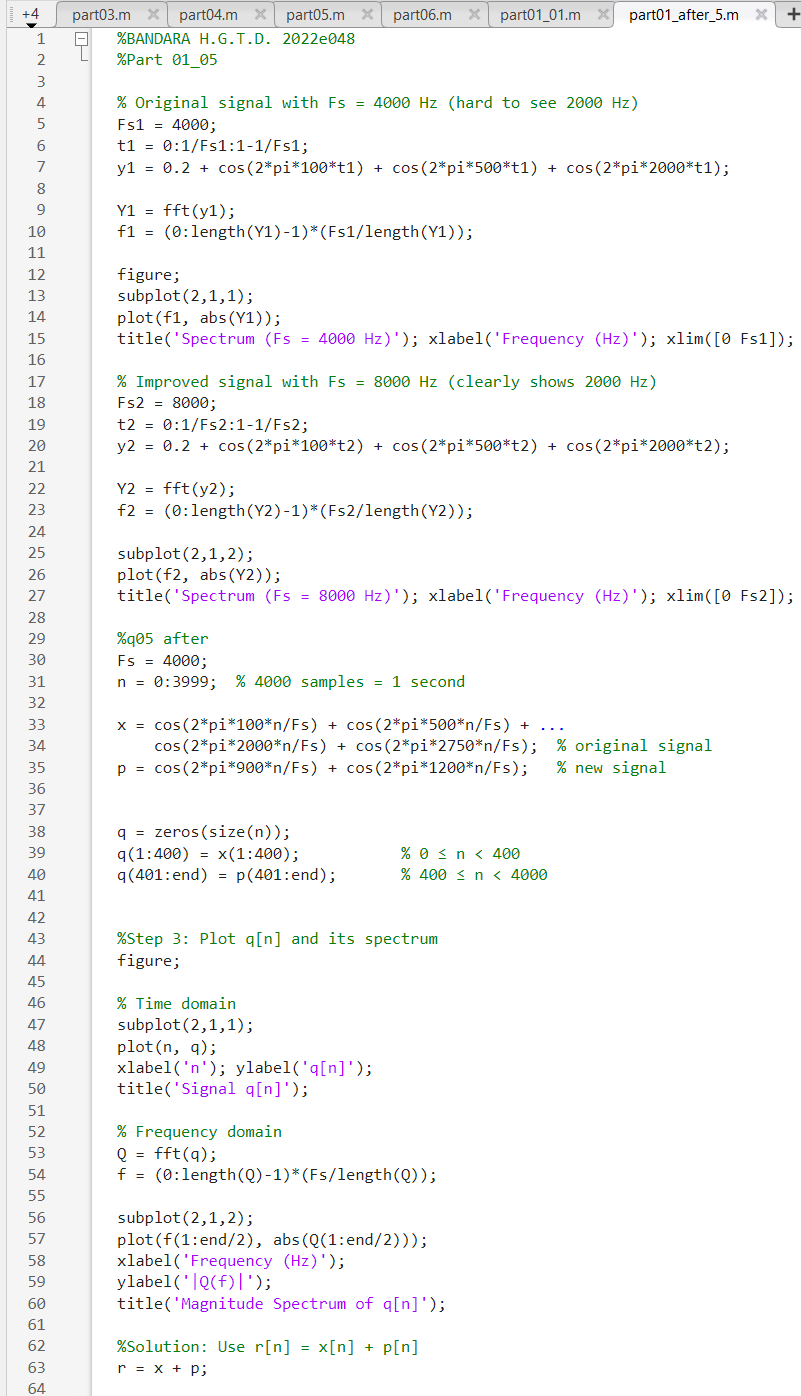
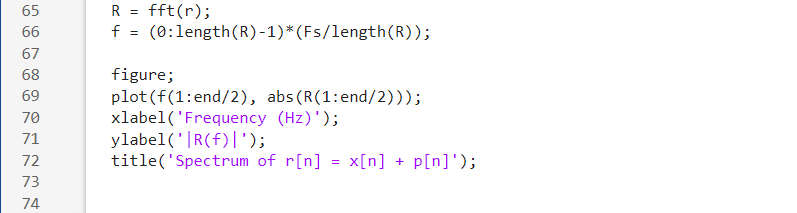
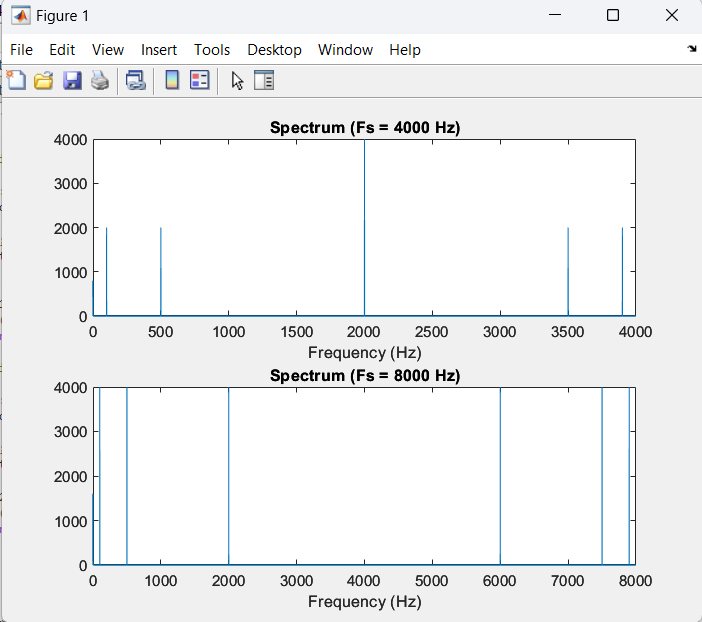
5)

FIGURE 05: MATLAB CODE

Figure 06:OUTPUT 4000Hz Vs 8000 Hz

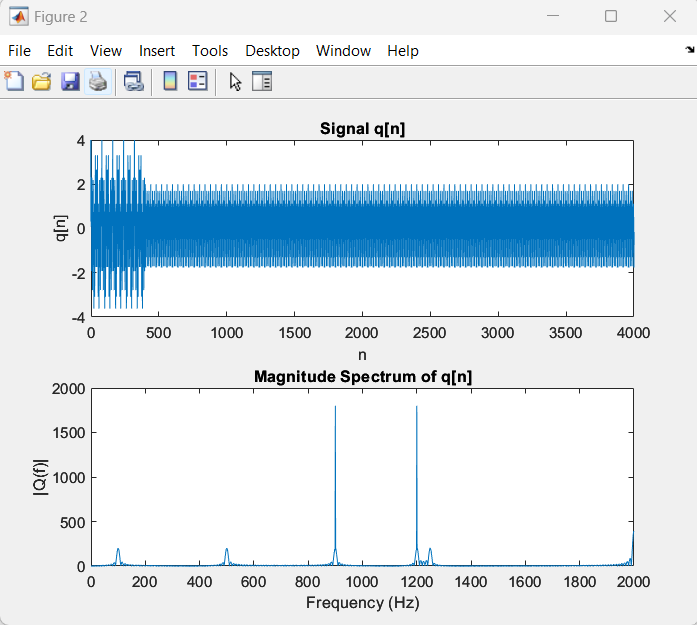


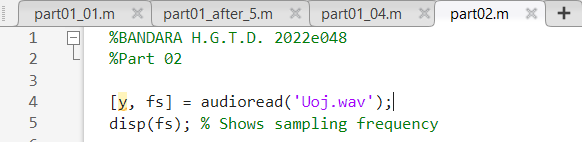
FIGURE 07: OUTPUT MAGNITUDE SPECTRUM

A screenshot of a graph

AI-generated content may be incorrect.

FIGURE 08: OUTPUT (SPECTRUM r[n] +p[n])

**PART2: TIME DOMAIN PROCESSING OF AUDIO SIGNALS**

1. ****

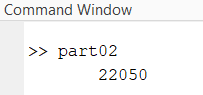
Figure 06: Matlab code for find the sampling frequency

FIGURE 09: OUTPUT

Sampling frequency = 22050 Hz

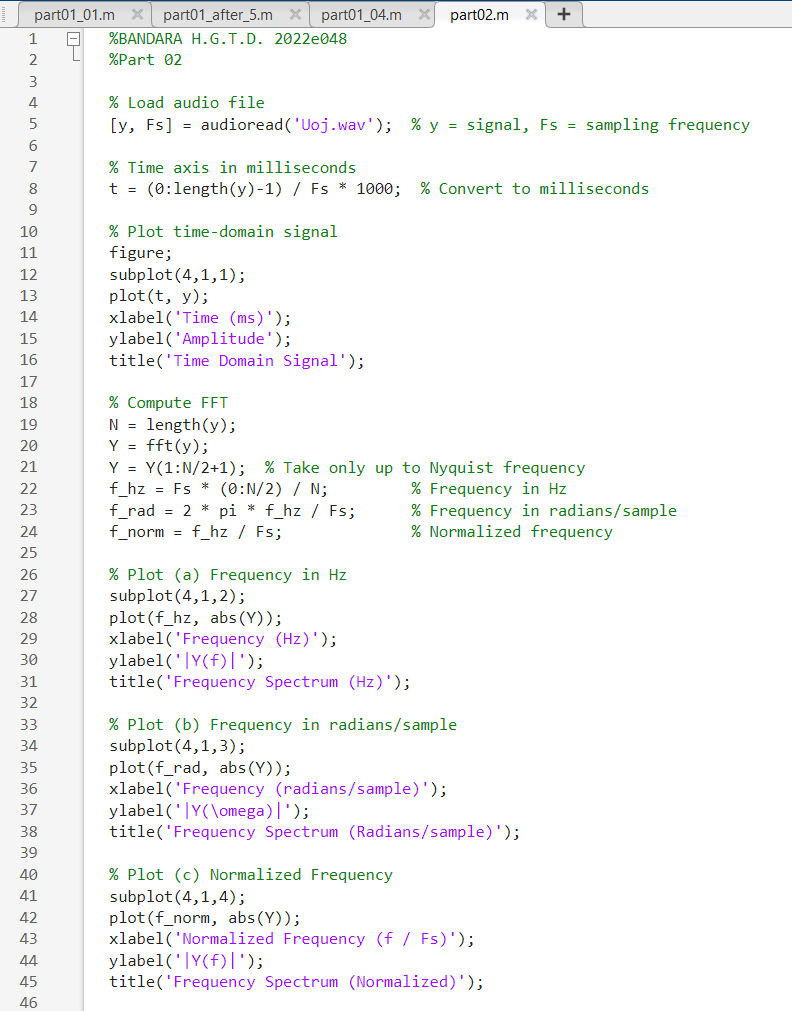
1. 

FIGURE 10:MATLAB CODE

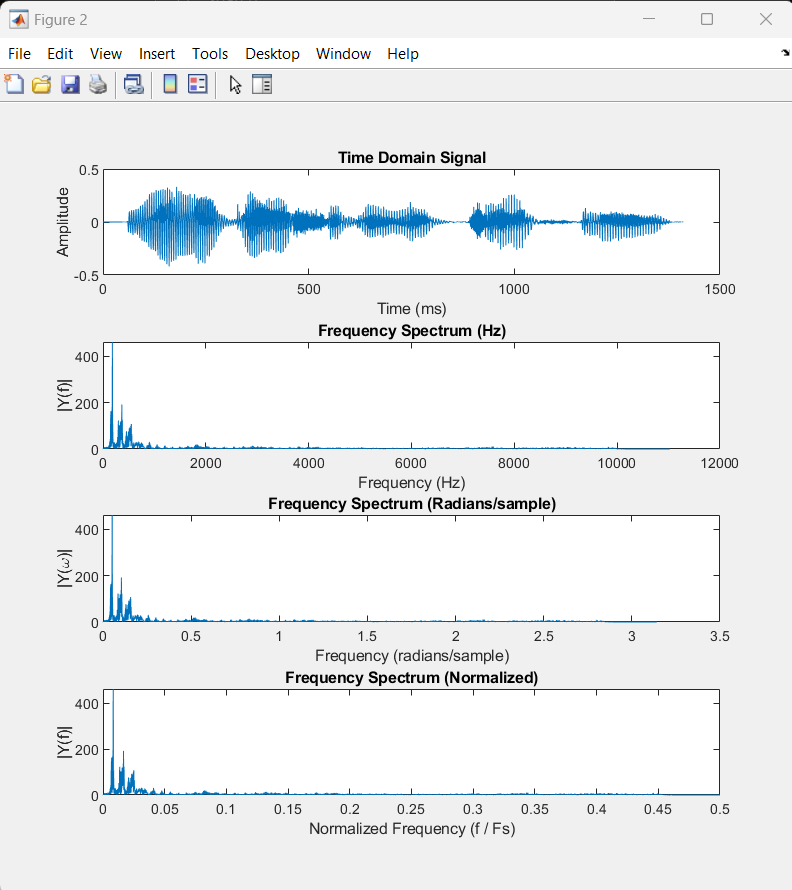


FIGURE 11:OUTPUT

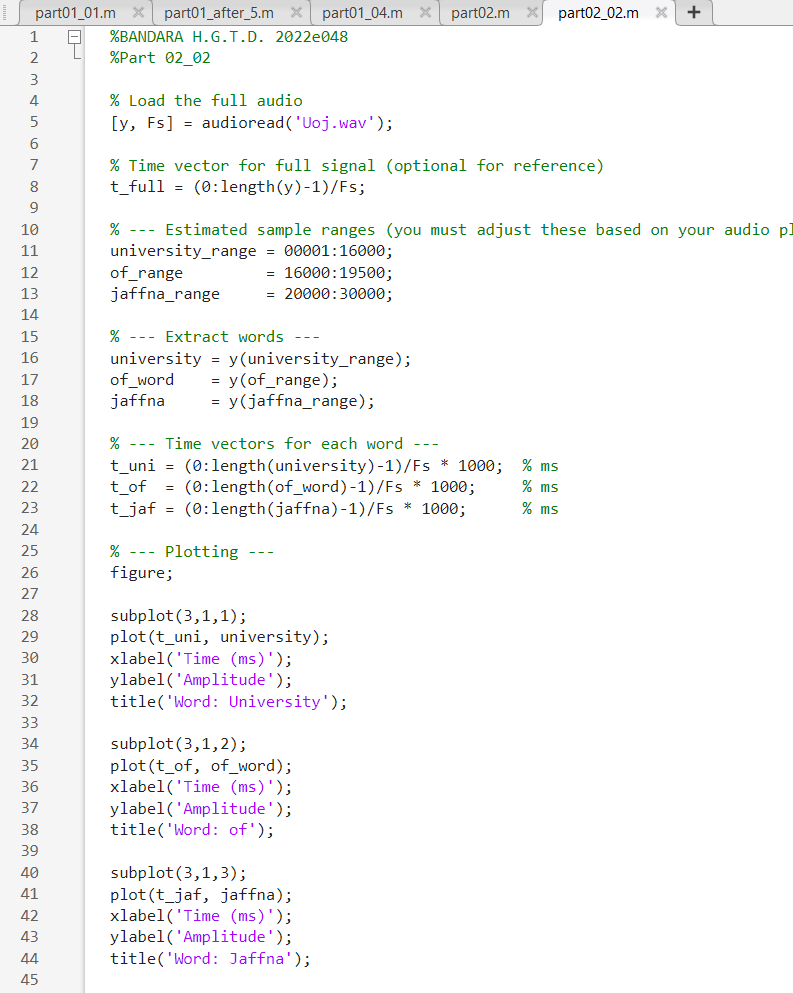
3)

FIGURE 12: MATLAB CODE

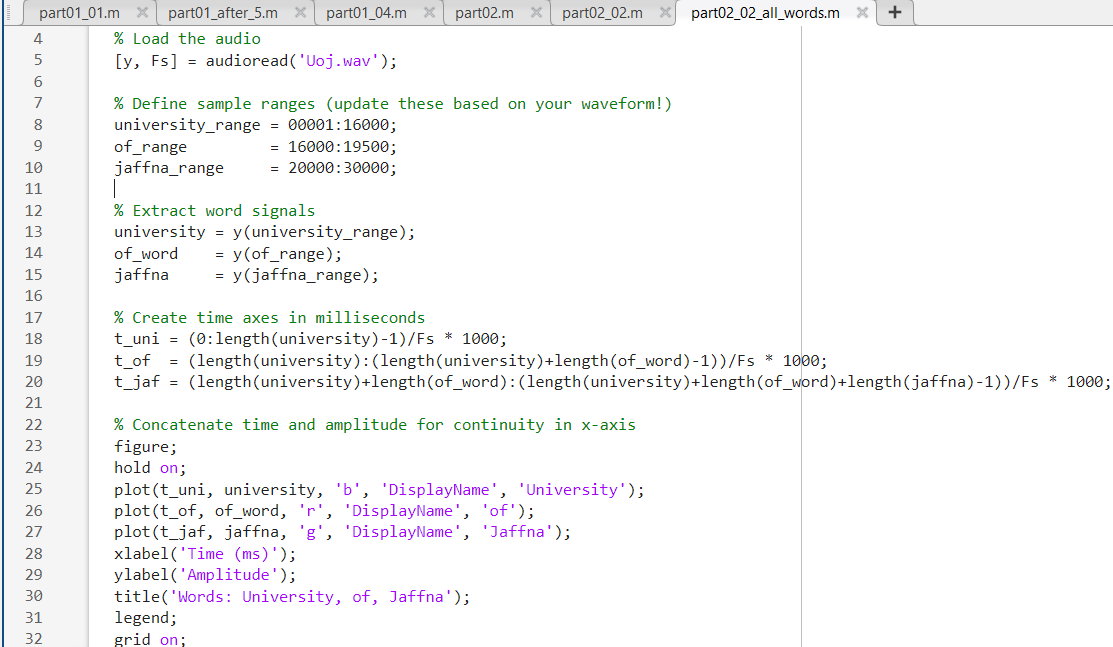
****

FIGURE 13:MATLAB CODE FOR GET ALL WORDS IN ONE PLOT

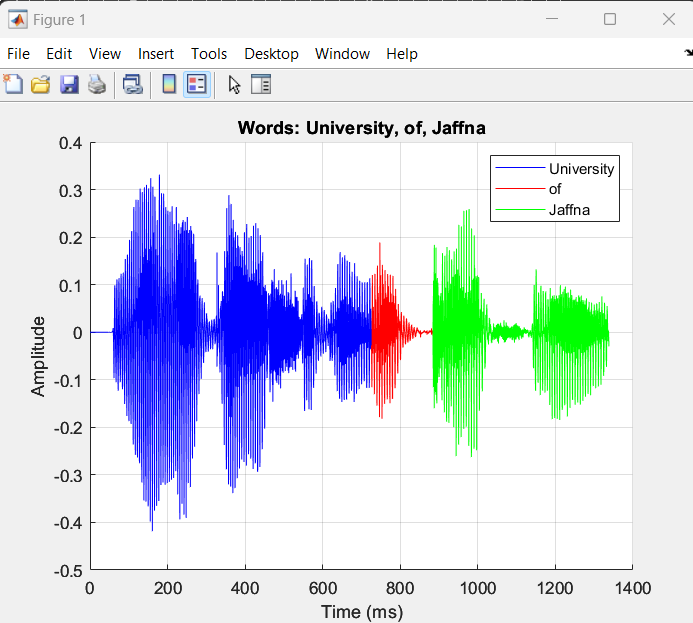
****

FIGURE 14:OUTPUT

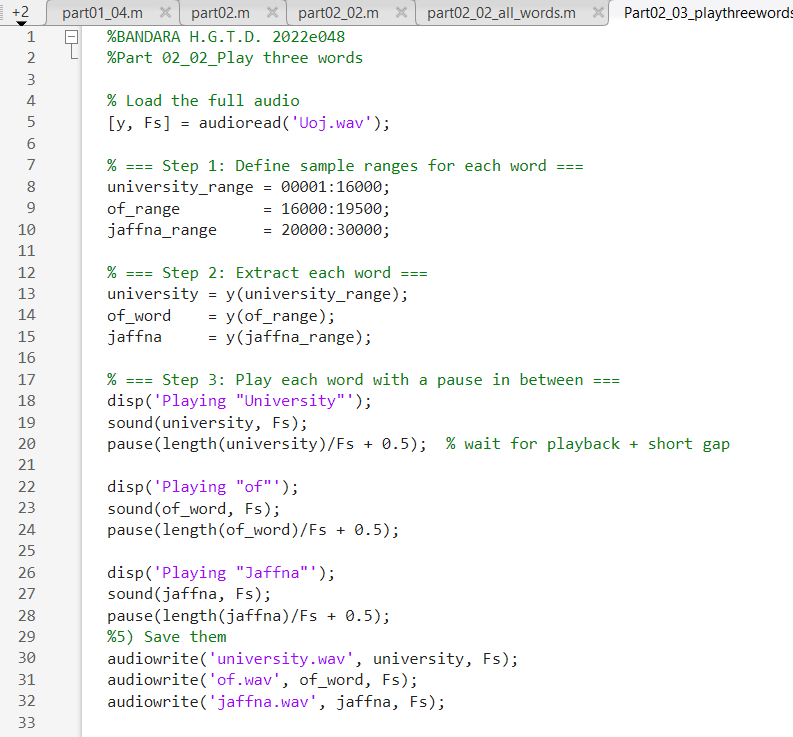
**4) 5)**

FIGURE 15: MATLAB CODE FOR PLAY ONE BY ONE SEPARATELY AND SAVE EACH PARTS

**PART3: CONVOLUTION AND FILTERING IN LTI SYSTEM**

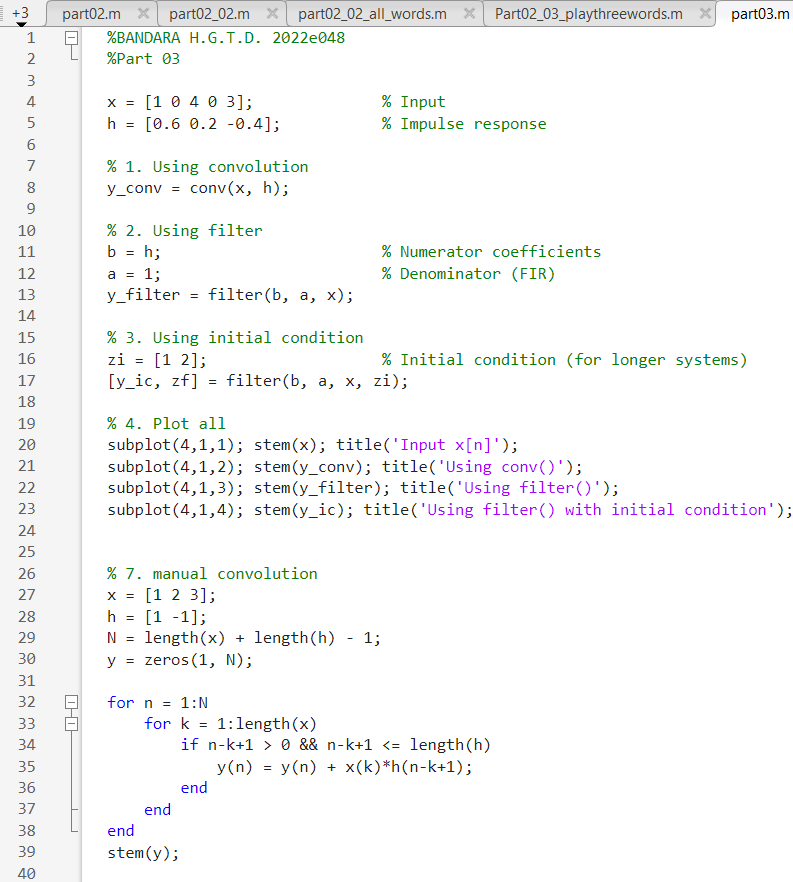
****

FIGURE 16:FUL MATLAB CODE FOR ALL SUBPARTS IN PART 03

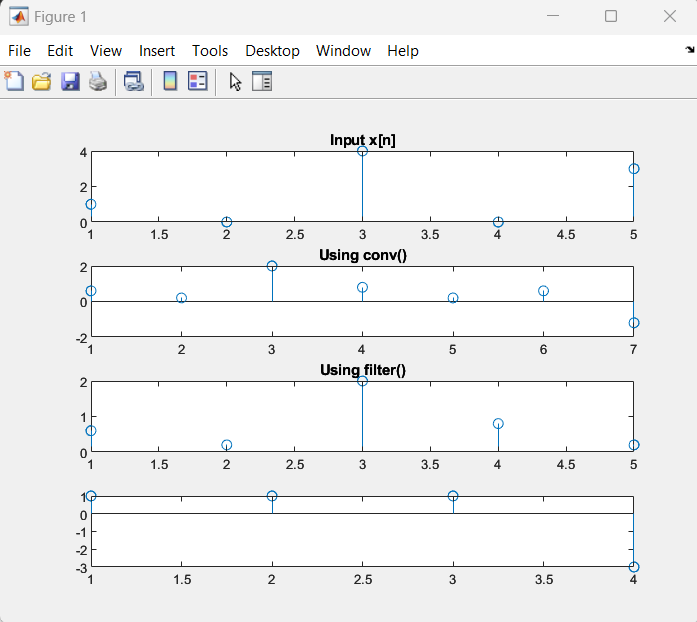


FIGURE 17: OUTPUT

1. **What it does**: Implements convolution assuming causal system starting at n=0n=0n=0.

**Length of result**: Same as x.

filter is based on the difference equation of the system:

y[n]=h[0]x[n]+h[1]x[n−1]+h[2]x[n−2]+...y[n] = h[0]x[n] + h[1]x[n-1] + h[2]x[n-2] + ...y[n]=h[0]x[n]+h[1]x[n−1]+h[2]x[n−2]+...

04) **Initial Condition (zi)**

* Represents the **memory of the filter** before the current input is applied.
* Useful when We're processing signals in **blocks** — it ensures **continuity** between blocks.

**Final Condition (zf)**

* Captures the **final internal state** of the filter **after** processing a block.
* We can use zf as the zi for the **next block** to simulate a continuous filter state.

06)

* **conv(x, h):**
* **Full convolution**, includes all output samples from n=0n = 0n=0 to n=N+M−2n = N + M - 2n=N+M−2
* **Used for theoretical analysis** or non-causal systems.
* **filter(h, 1, x):**
* Used in **real-time or causal systems**.
* Output length = length of input x
* Computation starts with assuming **zero past values**, unless given via zi.
  + **Final Condition (zf):**
* Stores the **last internal filter state**.
* Important for **processing signals in blocks** — allows continuity between blocks.

**CONCLUSION**

In this lab, we explored the foundational concepts and practical applications of digital signal processing through three major sections: sampling and frequency analysis, time-domain processing of audio signals, and convolution/filtering in LTI systems.

**🔹 Part 1: Sampling and Frequency Domain Analysis**

We began by generating and analyzing sampled signals using both time and frequency representations. By applying the FFT, we visualized how aliasing occurs when the signal frequency exceeds half the sampling rate (Nyquist limit). We learned that increasing the sampling frequency allows us to accurately capture higher-frequency components (e.g., 2000 Hz), and we demonstrated this effect with clear MATLAB plots. This part reinforced our understanding of signal reconstruction, aliasing, and frequency scaling in Hz, radians/sample, and normalized units.

**🔹 Part 2: Audio Signal Time-Domain Processing**

We processed an actual audio file by plotting its waveform and identifying individual words (“University”, “of”, “Jaffna”). Using sample indices determined from the waveform, we segmented, played, and saved each word separately. This hands-on experience emphasized the importance of sampling frequency, audio segmentation, and MATLAB’s audio functions for practical signal manipulation.

**🔹 Part 3: Convolution and Filtering in LTI Systems**

We implemented linear time-invariant (LTI) system responses using the conv and filter commands. By comparing these methods and manually coding convolution with loops, we gained deep insights into how filtering operations affect signals. We also explored the significance of initial and final conditions in the filter command, which are essential for block-wise or real-time processing to preserve system memory across signal segments.