

Python Term Project 提案內容

專題名稱 (Project Title): 互動式訊號產生器與頻譜分析實驗台
Interactive Signal Generator & Spectrum Analysis Workbench

1. 專題摘要 (Project Objective):

本專題旨在開發一個訊號處理模擬軟體。不同於傳統使用麥克風錄音，本程式允許使用者輸入自定義的數學波形函數（如 $\sin(2\pi ft) + \cos(2\pi ft)$ ），程式將解析函數並生成數位訊號。隨後，程式將模擬真實環境，對訊號注入高斯白雜訊 (Gaussian White Noise) 或隨機干擾。利用快速傅立葉變換 (FFT) 分析頻譜，並設計數位濾波器 (Digital Filter) 將雜訊移除，最終比較原始訊號、含噪訊號與還原訊號的差異，以視覺化方式呈現訊號處理原理。

2. 核心功能與規劃 (Key Features):

- 函數解析器 (Function Parser):
 - 設計介面讓使用者輸入波形公式（支援三角函數、指數等）。
 - 利用 Python 解析字串並生成對應的時域波形數據 (Time Domain Data)。
- 雜訊模擬 (Noise Injection):
 - 設計功能可將特定強度的隨機雜訊 (Random Noise) 疊加到使用者生成的純淨波形上，模擬受干擾的訊號。
- 頻譜分析 (Spectrum Analysis):
 - 對含噪訊號進行 FFT 運算，繪製出頻率對振幅的頻譜圖，讓使用者觀察「訊號」與「雜訊」在頻域上的分佈。
- 濾波與還原 (Filtering & Restoration):
 - 實作「頻域遮罩 (Frequency Masking)」或「巴特沃斯濾波器 (Butterworth Filter)」。
 - 讓使用者設定截止頻率 (Cutoff Frequency)，將特定頻段的雜訊濾除，並透過逆傅立葉變換 (IFFT) 還原波形。

3. 使用技術與套件 (Tech Stack):

- 數值運算: NumPy, SciPy (濾波器設計 `scipy.signal`)
- 函數解析: SymPy (符號運算, 較安全) 或 Python 內建 `eval` (需做安全限制)
- 視覺化: Matplotlib (繪製 Subplots: 原始波形 vs. 含噪波形 vs. 頻譜圖 vs. 還原波形)
- 介面: Tkinter (輸入函數框、滑桿調整雜訊強度與濾波頻率)

4. 數學原理 (Mathematical Background):

- 訊號合成 (Signal Synthesis): 理解波的疊加原理。
- 傅立葉轉換 (FFT): 時域與頻域的轉換關係。
- 訊號雜訊比 (SNR): 計算並比較降噪前後的訊號能量指標。