HOMEWORK SET #1

Sinusoids, Sampling, and Heart Rate Analysis

Total Points: 100 (Theory: 40 pts + Programming: 60 pts)

Published Date:	January 18, 2026
Due Date:	January 25, 2026 at 11:59 PM
Student Name:	________________________________

Required Files (download from Canvas):

• PPG_dataset.csv — Heart pulse signal data (sampling rate: 125 Hz)

Important Notes

• You may discuss with classmates but must write your own solutions.
• Show all work for full credit. Partial credit given for correct approach.
• Submit: (1) Theory solutions as PDF or clear photos of your handwritten work, (2) Jupyter notebook (.ipynb) with all cells run.

PART A: THEORY (40 points)

Problem A.1. Sinusoid Parameters (12 points)

Consider the sinusoidal signal: $x(t) = 4\cos(100\pi t + \pi/6)$

(a) (3 pts) Identify the amplitude A, angular frequency ω (rad/s), frequency f (Hz), and phase φ (radians and degrees).

(b) (3 pts) Calculate the period T in seconds and milliseconds.

(c) (3 pts) Calculate $x(0)$, $x(T/4)$, and $x(T/2)$. Show your work.

(d) (3 pts) Find the smallest positive time t when $x(t)$ reaches its maximum value.

Problem A.2. Time Delay and Phase Shift (10 points)

The relationship between time delay and phase shift is:

$$\varphi = -2\pi f \times \Delta t \quad \leftrightarrow \quad \Delta t = -\varphi \,/\, (2\pi f)$$

Two signals with frequency 200 Hz are given:

$x_1(t) = 3\cos(400\pi t)$

$x_2(t) = 3\cos(400\pi t - \pi/3)$

(a) (3 pts) Is $x_2(t)$ leading or lagging $x_1(t)$? Explain based on the sign of the phase.

(b) (4 pts) Calculate the time delay Δt between the two signals in milliseconds.

(c) (3 pts) Express the time delay as a fraction of the period T.

Problem A.3. Sampling and Discrete Signals (10 points)

A PPG (photoplethysmography) sensor samples a heart pulse signal at $f_s = 125$ *Hz*.

(a) (2 pts) Calculate the sampling period T_s in milliseconds.

(b) (3 pts) According to Nyquist theorem, what is the maximum frequency that can be captured without aliasing?

(c) (2 pts) If we record for 60 seconds, how many samples will we have?

(d) (3 pts) A person's heart rate is 72 BPM (beats per minute). What is the frequency of heartbeats in Hz? Is this frequency well below the Nyquist limit?

Problem A.4. Adding Three Sinusoids (8 points)

When sinusoids have the **same frequency**, their sum is also a sinusoid.

To find A and φ, use the **angle addition formula**: $\cos(\alpha + \beta) = \cos\alpha \cos\beta - \sin\alpha \sin\beta$

After expanding all signals and combining, convert using: $a\cos(\theta) + b\sin(\theta) = R\cos(\theta - \varphi)$

where $R = \sqrt{(a^2 + b^2)}$ and $\varphi = \arctan(b/a)$

Given three signals (f = 100 Hz):

$x_1(t) = 5\cos(2\pi \cdot 100 \cdot t)$

$x_2(t) = 4\cos(2\pi \cdot 100 \cdot t + 90°)$

$x_3(t) = 2\cos(2\pi \cdot 100 \cdot t + 180°)$

(8 pts) Use the angle addition formula to expand $x_1(t)$, $x_2(t)$, and $x_3(t)$, then add them all together. Express the result as $x(t) = A\cos(2\pi \cdot 100 \cdot t + \varphi)$. Show all steps.

PART B: PPG SIGNAL ANALYSIS (60 points)

Real-Time Heart Rate Monitoring with Sliding Window

What is PPG?

Photoplethysmography (PPG) uses light to measure blood volume changes in tissue. Each heartbeat causes a pulse in the signal.

Real-world applications: Apple Watch, Fitbit, pulse oximeters, and hospital monitors all use PPG technology to track heart rate continuously.

Your task: Build a heart rate monitor that calculates HR every 2 seconds using a 10-second sliding window — just like a real smartwatch!

Sliding Window Algorithm

Window size: **10 seconds** | Step size: **2 seconds**

```
Window 1: [0s  - 10s]  → Count peaks → HR₁ at t = 5s
Window 2: [2s  - 12s]  → Count peaks → HR₂ at t = 7s
Window 3: [4s  - 14s]  → Count peaks → HR₃ at t = 9s
... and so on
```

Output: A time series of HR values showing how heart rate changes over time.

Problem B.1. Load and Visualize PPG Data (10 points)

Download **PPG_dataset.csv** from Canvas. The sampling rate is $fs = 125\ Hz$.

Hint - Load CSV file:

```
df = pd.read_csv('...')
col_t = df.columns[...]          # Time column (first)
col_ppg = df.columns[...]        # PPG column (second)
t = df[...].values               # Time as numpy array
ppg_signal = df[...].values
```

(a) (3 pts) Load the CSV file using pandas. Print the total number of samples and calculate the recording duration in seconds.

(b) (4 pts) Plot the full PPG signal with time (seconds) on the x-axis. Add appropriate title and axis labels.

(c) (3 pts) Create a zoomed plot showing exactly 10 seconds of data (e.g., from t=0 to t=10s). You should be able to see individual heartbeat peaks clearly.

Problem B.2. Peak Detection (15 points)

Use *scipy.signal.find_peaks()* to detect heartbeat peaks. You need to choose appropriate parameters.

Hint - find_peaks function:

```
from scipy.signal import find_peaks
peaks, _ = find_peaks(signal, height=____, distance=____)
```

(a) (10 pts) Set **distance** by observing your zoomed plot: estimate the inter-peak interval (in seconds), then convert to samples using *distance_samples = distance_seconds* × *fs*.

Important: Do not set distance too large, or you will miss peaks! Use about **50%** of your observed interval to be safe (e.g., if you observe ~0.8s between peaks, use distance ≈ 0.4 × 125 = 50 samples). For **height**, try using *np.mean(ppg_signal)*. Apply find_peaks to detect all heartbeat peaks.

(b) (5 pts) Plot the 10-second zoomed view with detected peaks marked as red dots. Report how many peaks were detected in this window.

Problem B.3. Heart Rate Calculation — Single Window (15 points)

Calculate the heart rate for a fixed 10-second window (t = 0 to 10 seconds).

Hint - Calculate HR from peaks:

```
peak_times = t[peaks_in_window]          # Times of peaks
intervals = np.diff(peak_times)          # Time between consecutive peaks
avg_interval = np.mean(intervals)        # Average interval
hr = 60 / avg_interval                   # HR in BPM
```

(a) (5 pts) Extract peaks from the first 10-second window. Report the number of peaks detected.

(b) (7 pts) Calculate the average inter-peak interval and the heart rate using: *HR = 60 / avg_interval*

(c) (3 pts) Is your result physiologically reasonable? Normal resting HR is 40-180 BPM. Report your calculated HR.

Problem B.4. Sliding Window Heart Rate (15 points)

Calculate the heart rate over time using a sliding window approach with **window size = 10 seconds** and **step size = 2 seconds**.

Hint - Sliding window algorithm:

```
hr_list = []
time_list = []
peak_times = peaks / FS  # Convert peak indices to seconds

for start_time in range(0, int(duration - 10) + 1, 2):
    end_time = start_time + 10
    # Filter peaks in window [start_time, end_time]
    mask = (peak_times >= start_time) & (peak_times < end_time)
    peaks_in_window = peaks[mask]
    # Calculate HR (same method as B.3)
    # ...
    hr_list.append(hr)
    time_list.append(start_time + 5)  # Center of window
```

(a) (10 pts) Implement the sliding window algorithm. For each window, filter peaks and calculate HR using the method from Problem B.3.

(b) (5 pts) Apply your algorithm to the full PPG signal. Report: How many HR values were calculated?

Problem B.5. Visualize Heart Rate Trend (5 points)

Create a plot of Heart Rate vs. Time. Include: HR values (y-axis) vs. window center time (x-axis), a horizontal line showing the mean HR, proper title, axis labels, and legend.

Submission Requirements

Part A: Theory solutions (Problems A.1-A.4) as PDF. File: *HW1_YourName_Theory.pdf*
Part B: Jupyter notebook with all code and outputs. File: *HW1_YourName_PPG.ipynb*
Deadline: ________________, 2026 at 11:59 PM on Canvas

Grading Summary

Part	Problem	Topic	Points
A	A.1	Sinusoid Parameters	12
A	A.2	Time Delay & Phase	10
A	A.3	Sampling	10
A	A.4	Adding Sinusoids	8
		Part A Subtotal	**40**
B	B.1	Load & Visualize PPG	10
B	B.2	Peak Detection	15
B	B.3	HR Single Window	15
B	B.4	Sliding Window HR	15
B	B.5	HR Visualization & Statistics	5
		Part B Subtotal	**60**
		TOTAL	**100**