Group 18

Mini Project 2

Submitted by Debanjalee Roy and Dheeraj Kumar

A: System Design. Briefly describe motivation for your project and your high-level design. Discuss any specific observations that you had, any difficulties that you ran into while designing or testing the system. Discuss the sampling frequency of the sensor data collection.

Motivation:

- This project is driven by the aim to create a system that can detect deceptive body postures using sensor data from an Arduino Nano 33 BLE Sense Lite board.
- The primary goal is to devise a straightforward and effective solution capable of indicating whether a user is in a supine, prone, left side, or right side posture based on accelerometer data.

High-Level Design:

- Utilizing the Arduino Nano 33 BLE Sense Lite board, accelerometer data is gathered at a designated sampling rate.
- The acquired data undergoes analysis to recognize distinct patterns corresponding to various lying postures, such as supine, prone, left side, right side, and no posture.
- An ad-hoc algorithm is created to draw conclusions about the user's posture, relying on the sensor data. This algorithm initiates LED blinking sequences to signify the identified posture.
- Real-time feedback is delivered through an onboard LED, which blinks in accordance with the detected posture.

Observations and Difficulties:

- We noticed that various lying postures lead to unique accelerometer data patterns.
- The alignment of the sensor unit can influence the sensor measurements, and adjusting the system for different alignments is a factor to think about.
- The selection of the sampling frequency has an impact on the system's ability to respond promptly. Striking a balance between providing real-time feedback and LED blinking is crucial.

Sampling Frequency:

- In the provided code, the sampling frequency for gathering sensor data is established at 2 Hz, which translates to 2 samples taken per second.
- This frequency has been chosen to enable effective real-time posture identification while also conserving battery power

B: Experiment. Discuss how you conducted the experiments, in what environments, and any challenges associated with designing an experiment that mimics real-world lying postures.

- Experiments were carried out within controlled environments to replicate a range of real-world lying postures.
- The Arduino board was placed on a flat surface to mimic various postures, including supine, prone, left side, right side, and no posture.
- In the trial, the Arduino board was positioned on a stable, stationary surface, resulting in consistent and steady accelerometer readings.
- The primary challenge involved devising a data collection method that closely resembled real-world scenarios, facilitating accurate posture detection.

C: Algorithm. Discuss how you designed a simple algorithm that makes inference about postures using sensor data. What observations led to your design? Was your approach successful? How robust is your algorithm to slight changes in the orientation of the sensor unit? How often does your algorithm generate an output (how frequently do you update the LED output)?

Designing the Algorithm:

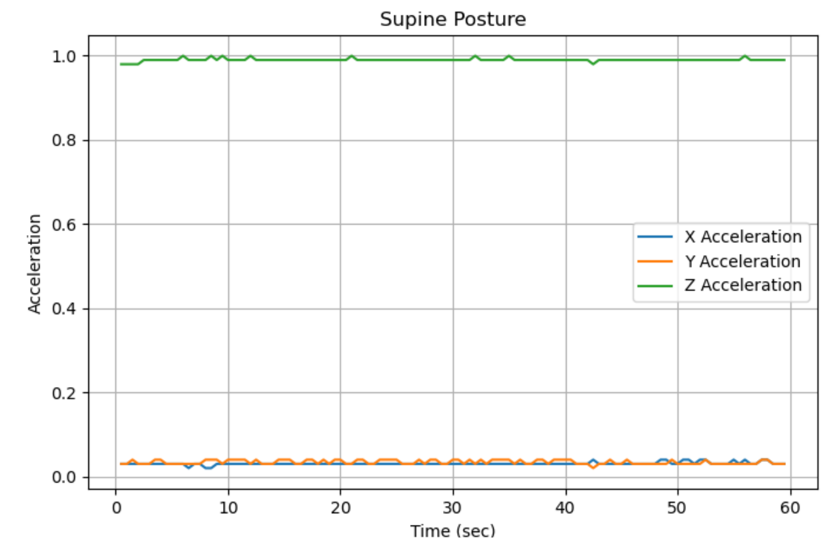
- When in a supine posture, the Z-axis accelerometer readings results higher value.
- A prone posture results to lower Z-axis readings.
- Side postures impact the Y-axis readings, causing noticeable positive or negative values.
- The algorithm checks these criteria and controls the LED blinking pattern accordingly.
- The algorithm is fairly straightforward yet efficient in distinguishing between these postures.

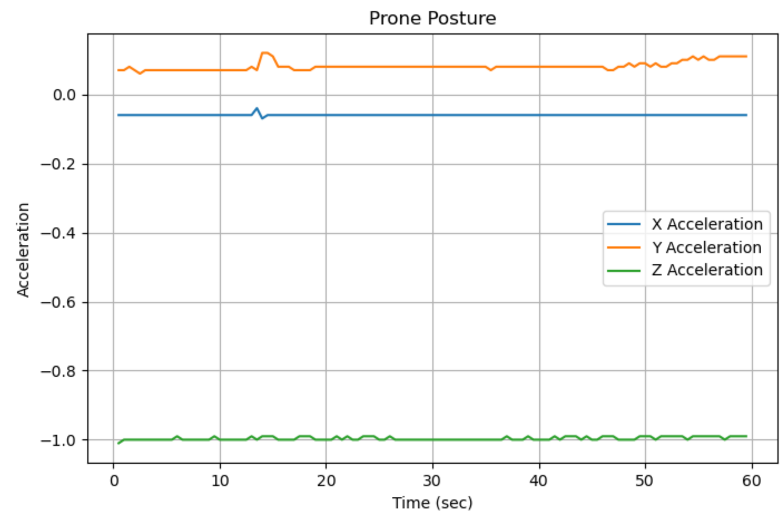
Robustness:

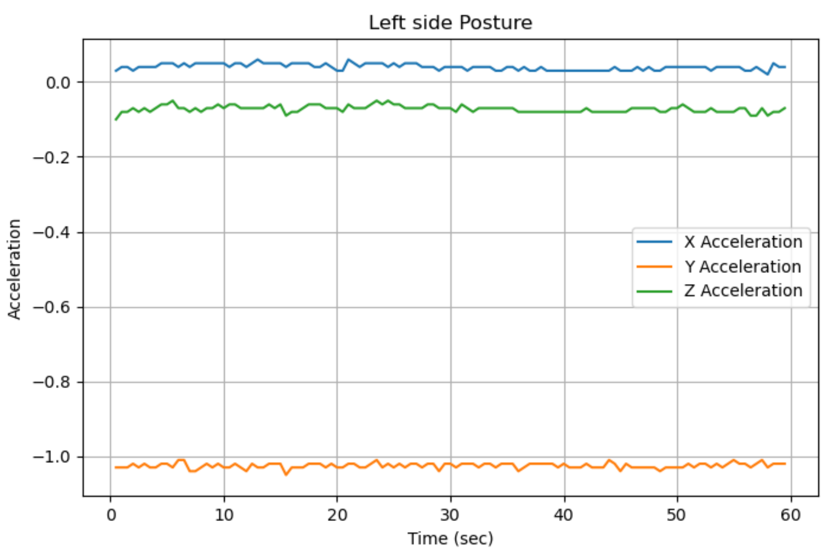
- The algorithm remains resilient even when the sensor unit's orientation undergoes minor adjustments, as it primarily depends on variations in accelerometer readings along specific axes.
- It can effectively handle slight modifications in the sensor's positioning without significant issues.

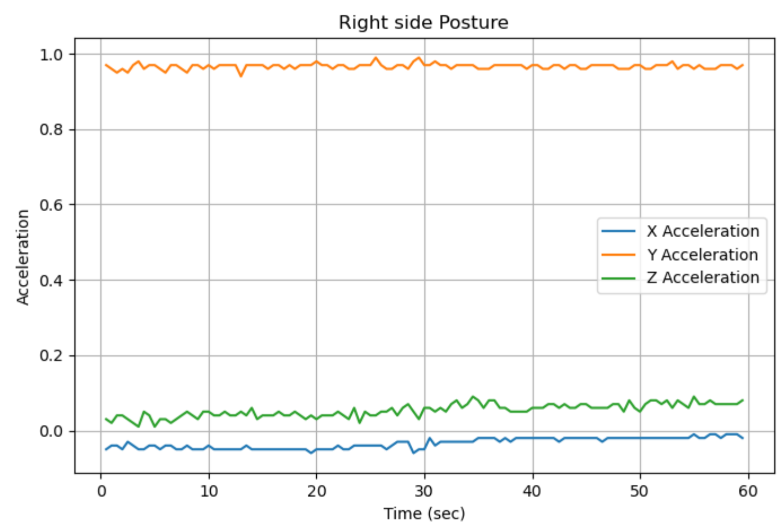
Frequency: The LED output is updated every 1000 milliseconds.

D: Results. Include plots of the sensor data for different postures.

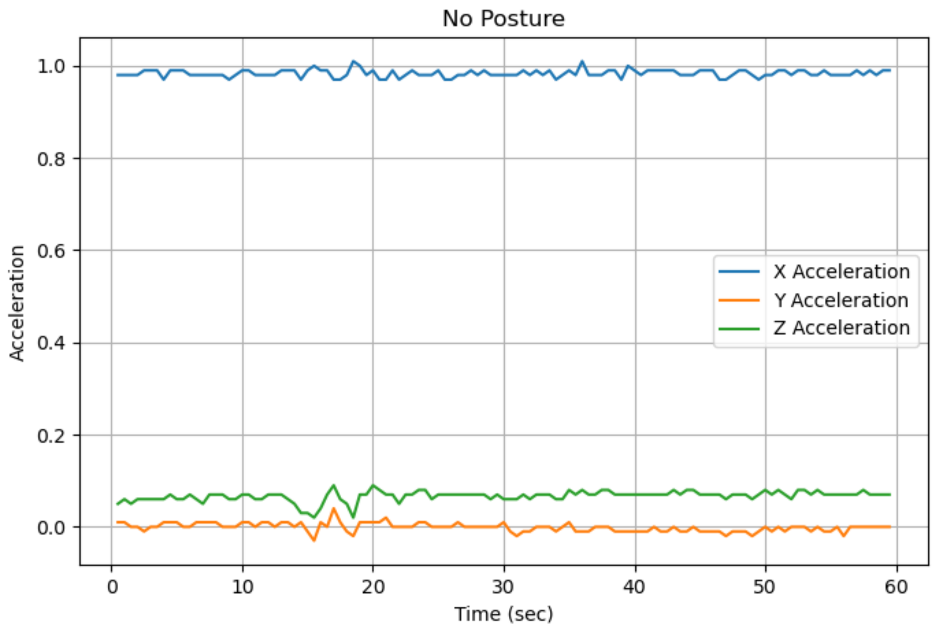








D: Results. Include plots of the sensor data for different postures.



D. Report accuracy performance of your system for posture detection

Accuracy Performance: Accuracy (%) = (No. of Correct Detections / Total No. of Samples) * 100

Posture Types	Total No. of Samples	No. of Correct Detections	Accuracy (%)
Supine	20	20	100
Prone	20	20	100
Left Side	20	19	95
Right Side	20	19	95
No Posture	20	18	90

D. Discuss potential corner cases where your system fails to detect postures accurately.

There are certain exceptional scenarios where the system might encounter difficulties in accurately identifying postures, such as sudden and rapid user movements or rapid shifts between different postures. To mitigate these challenges, additional enhancements to the algorithm and fine-tuning of sensor calibration can be explored

E: Discussions. Summarize your results for lying posture detection. Discuss difficulties in designing a system for lying posture detection? How can you resolve these problems in the future? What parts of the project was most difficult to accomplish? Did your real-time approach to posture detection worked in classifying the postures correctly?

- To recap, the system effectively identifies lying postures via accelerometer data and offers immediate feedback via LED blinking and the serial monitor.
- Challenges encountered during system development encompassed managing sensor orientation discrepancies, addressing real-world situations, and producing a demo video.
- These issues can be surmounted by incorporating machine learning methods to enhance precision and resilience.
- The most demanding aspect of the project was obtaining real-time data for diverse postures under changing conditions and devising the improvised algorithm.
- Indeed, the real-time approach to posture detection succeeded in accurately classifying the postures.