**Aim of the project:** Estimate the impact of available battery power in smart devices on variability of timestamps of data captured from device’s sensors.

**Research design:** Sensors are always or periodically active. They produce data at certain acquisition frequency in real time that are digitized and put it in a memory buffer for reading. However, in cases where the sensors are managed by the Android OS, the sensor raw data acquisition can be delayed causing forward shifts in the timestamps. That delay can be random and not constant but rather a variable that depends on many parameters. In this study we are concentrating on how device’s power management affects sensing data quality. While we can control the sensor data read frequency, we do not have access to the wait in a queue for the buffer read. Therefore, we do not have access to the time when sensor updates the buffer. However, sensing is much faster (100Hz) than we will read the sensor data (5Hz), hence, should not introduce significant differences in the timestamps by itself. In addition, we do have access to the Qualcomm sensing information through the QSEE API. For the QSEE supported sensors, we can compare OS (Android) reported sensing with the QSEE hardware timestamps for all supported sensors (3-4).

The timestamp variability will be used to estimate reliability of the sensors data timestamps under different conditions to understand better sensing data quality. Data quality can be assessed by evaluating the time interval between captured sensor data. We will use in-house designed mobile app to listen for specific sensor(s) at a given polling rate. App will be kept in the background during recordings. Since, the time when data are read from the sensor buffer are not going to be exactly equal to preset polling rate but dependent on OS queuing it, that ultimately will introduce a variability between the readings. We plan to quantify the sensing data quality for single and multiple sensors under normal conditions and when the device exhibits low power and high CPU utilization.

Initially, we select to monitor are linear accelerometer and gyroscope, which are most commonly used for health research.

**Methods:**

***Single sensor study.***

*Our control* experiment for sensor data collection will be comprised of the collecting data timestamps on each sensor separately on 100% charged plugged in device. This is best case scenario that we should expect. So, variability in the timestamps from the sensors data is expected to be at its minimum.

*The first condition* is 100% charged not plugged in device, this condition will provide preferable real-world conditions in the course of relatively consistent data collection. This should reveal timestamp variability that one should expect during a daily use of their device that maybe slightly greater than in control.

*The second condition* is 10% battery charge remaining not plugged in to imitate data recording under the various power conserving mechanisms utilized by the device’s hardware and OS. This must produce the greatest variabilities in the readings that will provide insight into power management control of and access to the sensor’s data. For this scenario, we will consider a device that is in normal battery power mode and low power mode.

*The third condition*, 100% charged device under the heavy nearly 100% computational load to reveal how device’s power management systems enforce task threads depending on their power consumption. In this experiment our app tasks power consumption should be negligible to the load we artificially introduce. The results will show what kind of competition profile there is implemented and how users’ activity may impact accuracy of the data coming from sensors.

**Data analyses:** Data on variability as means and standard deviations can be compiled from these experiments and statistically analyzed for significant differences.

***Multi-sensor study***

*In the next set of experiments*, we will simultaneously record data from both sensors using the same recording paradigm. The variability of timestamps between the two sensors will be assessed. This data will elucidate the impact on reliability of reading from multiple sensors in various conditions. In addition to the between sensors variability we can assess their cross-correlation (lag). If in fact it is pronounced and increased or decreased that would indicate that there is a relationship that power management introduces to multi-sensor data readings.

***Multi-device study***

The same set of experiments will be conducted using two devices from two main different manufactures, i.e. Google and Samsung, but comparable in overall specifications. That will show the impact of different hardware on the same Android platform on reliability of the sensor data. For this study, we will be using a robotic arm with 6 Degrees of Freedom (DOF).

Statistical analyses

The variability of the mean interval between sensor readings across different collection conditions will be analyzed for the significant changes using Excel 2 tailed t-test function. In the situations where t-test did not show significant changes, the distributions of the intervals between sensor readings will be analyzed using two sample Z-score test formular written in Excel. 3D sensor data converted into a simple geometrical distance were routinely tested against intervals between sensor recordings for cross correlation to ensure that there were no underlying relationships between them. Cross correlation function (CCF) vs. lag relationships were calculated with no data wrapping and plotted out in Excel using custom written macros. Throughout the text of this manuscript averages were expressed as a mean +/- standard deviation (SD) followed by number of measurements.

Results

Overall, variability of the interval between sensor data readings showed little differences in means across all the conditions. However, there were instances where it was profound, e.g. when the particular device was plugged in or asleep. This may refer to the proprietary manufacture’s algorithm or the design of the app that is accessing sensor information. CCF analyses did not show correlation between the intervals and sensor reading, normally R was less than 0.1, meaning that sensor function itself does not, as expected, impose significant stress on the device functionality.

Google Pixel 6a

**Accelerometer** study has revealed that when Google Pixel 6a device was plugged in, for both when app was in background and foreground, the interval between instances the app accessed accelerometer sensor’s data was very close to the polling rate of 200ms, about 201.86+/-4.89 (n=891) (Fig.1, A and B). At closer inspection of the distributions (Fig.2), it was discovered that there were actually two populations with averages 200 and 218ms. Causes of this at this time are unknown and maybe investigated in the different study.

Between the rest of the cases there was no significant differences in means, so interval distributions were analyzed with Z-test. This test revealed that there was significant effect of the high CPU load (Fig.1, G, 291.44+/-0.56, n=617) on the intervals when compared with just 100% charged device (Fig.1, C and D, 291.37+/-0.67, n=617 and 291.37+/-0.71, n=617). 10% charged devices (Fig.1, E and F, 291.42+/-0.76, n=617 and 291.39+/-0.77, n=617) showed tendency but were insignificant with Z scores of 0.6 and 1.3, respectively. High CPU load significant effect seems to be due to the lower standard deviation in the intervals distribution that indicates more consistent sensor readings in these conditions.

Figure 1. Plugged in state and high CPU load has a significant impact on the intervals between accelerometer sensor’s data access.

Chart, bar chart

Description automatically generated

\*, # show significant mean differences in intervals between sensor readings (t-test, p<0.01) for 100%, 10% and 100% charged with high CPU load when compared with the plugged in, 100% charged device with sensor reading app running in foreground and background, respectively. + depicts significant difference between high CPU load and 100% charged device distributions of intervals between sensor readings (Z test, score >1.645, alpha 0.05). A - 100% charged device, plugged in, app is in foreground, and B when app is in background. C -100% charged device, app is in foreground, and D when app is in background. E - 10% charged device, app is in foreground, and F when app is in background. G - 100% charged device, CPU is in high load, app is in background.

Figure 2. Accelerometer sensor reading intervals show two modes when device is plugged in.

Graphical user interface, application

Description automatically generated

**Gyroscope** study, unlike accelerometer one, showed no significant differences in the intervals between the instances when gyroscope data was recorded across all the cases. Analysis of distributions with Z test revealed significant differences again between high CPU load intervals 291.45+/-0.57 (n=617), Fig.3, G. Namely, with 100% charged device both when data collecting app was in foreground and background 291.38+/-0.67(n=617) and 291.38+/-0.77(n=617) and with 10% charged device when the app was in background 291.38+/-0.79(n=617) (Fig.3 C,D, and F). Plugged in device in both data collecting scenarios and 10% charged device when data collecting app was in foreground (Fig. 3, A, B, E) showed high but insignificant tendency (Z scores were 1.4, 1.5, and 1.5, respectively). Similar to the accelerometer study, SD was the lowest in high CPU load case and seemed to cause the significant Z score.

Figure 3. High CPU load has a significant impact on the intervals between gyroscope sensor’s data access.

Chart, bar chart

Description automatically generated

+ depicts significant difference between high CPU load and 100% charged device distributions of intervals between sensor readings (Z test, score >1.645, alpha 0.05). A - 100% charged device, plugged in, app is in foreground, and B when app is in background. C -100% charged device, app is in foreground, and D when app is in background. E - 10% charged device, app is in foreground, and F when app is in background. G - 100% charged device, CPU is in high load, app is in background.

**Accelerometer and Gyroscope** sensors data recording were done simultaneously in this set of experiments. It has been observed that similarly to Accelerometer data, obtained during single sensor recordings, when device was plugged in, for both when data collecting app was in background and foreground, the interval between the readings was very close to the polling rate of 200ms, 202.08+/-5.13 (n=890) and 202.18+/-5.28 (n=890), respectively (Fig.4, A1 and B1). Interval distributions (Fig.5) was found to be similar to Fig.2 and had two modes. Averages for accelerometer sensor intervals in the plugged-in device were significantly lower from the rest of the accelerometer data sets (Fig.4).

Multi-sensor readings in contrast to single-sensor ones did not show significant changes when distributions of the intervals measured in the device that was not plugged in were tested with Z-test. There were tendencies, though insignificant, in Gyroscope readings between high CPU load (Fig.4, G2) with Z scores of around 1.0 for Fig. 4, A2, B2, C2, D2, and F2. Moreover, SD in high CPU load showed higher values of 2.34 and 4.2 then the rest of the measurements that had SD around 0.7. This was a complete reversal from the single-sensor experiments described above. That is, multi-sensor reading appears to introduce greater impact on the consistency of the intervals between the readings, hence it may possibly reduce the quality of the sensor data, especially at a higher CPU loads.

Figure 4. Plugged in state has a significant impact on the intervals between accelerometer sensor’s data access.

Chart, bar chart

Description automatically generated

\*, # show significant mean differences in intervals between accelerometer sensor readings (t-test, p<0.01) for 100%, 10% and 100% charged with high CPU load when compared with the plugged in, 100% charged device with sensor reading app running in foreground and background, respectively. Indices 1 and 2 refer to Accelerometer and Gyroscope interval measurements, respectively. A - 100% charged device, plugged in, app is in foreground, and B when app is in background. C -100% charged device, app is in foreground, and D when app is in background. E - 10% charged device, app is in foreground, and F when app is in background. G - 100% charged device, CPU is in high load, app is in background.

Figure 5. Accelerometer sensor reading intervals show two modes when device is plugged in, multi-sensor reading.

Graphical user interface, application

Description automatically generated with medium confidence

**Device movement**

As we mentioned above that when device was asleep, we saw the greatest increases in the intervals between the polling. Assumption was made that operating system due to device and sensors inactivity was disregarding application requests to access sensor’s data. To investigate if these numbers can be affected by the device movements, we conducted a separate set of measurements with the devices mounted on a robotic arm from Elephant Robotics and constantly moved about with frequent changes in directions to make sure accelerometer and gyroscope sensors get engaged. The same trajectory of the movement, stored in the robotic arm memory, was used for all the data collection in these series.

**Accelerometer** data from stationary or moving Google Pixel 6a phone are presented on Figure 6. 1st bar (Fig.6A) shows data from plugged in to a charger device just for the illustration purposes. We did not use plugged in device in our moving experiment due to the constrains that tethering imposes on a setup that might serve as a confounding variable. Nevertheless, distributions of between polling intervals of stationary plugged in (Fig.6A), 100% (Fig.6B), and 10% (Fig.6D) charged devices (1270.23+/-5996.30 n=117, 1498.39+/-5285.21 n=84, 2210.84+/-5257.84 n=77, respectively) were significantly different from 100% high load data (Fig.6F; 291.44+/-0.54 n=617) with Z scores of 1.76, 2.09, 3.2, respectively. Similarly, moving 100% (Fig.6C), and 10% (Fig.6E) charged devices (2191.94+/-9134.04 n=82, 1593.06+/-4258.59 n=112, respectively) were different from 100% high load (Fig.6G; 291.48+/-0.54 n=617) with Z scores of 1.88 and 3.24, respectively. These results highlight remarkable differences between sleeping and, though under a high amount of stress but still awake devices. Whether stationary or moving accelerometer data are much rarely accessed in a sleeping state.

Sleeping stationary vs. moving comparison although insignificant however still showed a tendency. Namely, 100% charged when moving showed on average longer between polling interval (Fig.6B,C, Z score=0.6). 10% charged phones when moving revealed the opposite, i.e. on average shorter polling intervals (Fig.6D,E, Z score=0.86). In the high load case, intervals were virtually identical (Fig.6F,G, Z score=1.2).

Figure 6. Moving and stationary devices showed similar mean intervals between the readings of accelerometer data.

Chart

Description automatically generated

+ shows significant difference in the distributions of intervals between sensor readings compared with (F) high CPU load when device was stationary (Z test, score >1.645, alpha 0.05). ^ shows comparisons with (G) high CPU load when device was moving. A - 100% charged plugged in, B – 100% charged, D - 10% charged device sleeping and stationary. C – 100% charged, E – 10% charged, device sleeping and moving. F and G – high CPU load, with app in background, when device was stationary or moving, respectively.

Tendencies noticed in the Figure 6 were investigated further. Figure 7 shows whisker plots of the intervals between the pollings for 100% charged and 10% charged devices. High load CPU cases did not show visible differences and were not shown. Surprisingly, device movement while it was asleep produced clusters of even longer intervals between the accelerometer data readings with averages around 40s and 15s (Fig.7Bs) when compared with stationary phones intervals of 20s and 5s (Fig.7As), for 100% and 10% charged devices, respectively. Despite that moving phones averages were close or lower than stationary, indicating that device was actually able to capture accelerometer data more often at the faster rate, approximately every 291ms.

Figure 7. Accelerometer data of moving devices shows clusters of longer intervals between sensor data reading.

Chart, scatter chart

Description automatically generated Chart, scatter chart

Description automatically generated

A – denotes whisker box plot for sleeping stationary device, when B – sleeping moving device.

**Gyroscope** data from stationary or moving Google Pixel 6a phone are presented on Figure 8. 1st bar (Fig.8A) shows data from plugged in to a charger device just for the illustration purposes. The tests showed that only the distribution of between polling intervals of stationary plugged in (Fig.8A, 1534.11+/-5221.82 n=84) was significantly different from high CPU load data (Fig.8F; 291.45+/-0.57 n=617) with Z scores of 2.18. 100% (Fig.8B), and 10% (Fig.8D) charged devices (11410.33+/-33398.14 n=91, 6237.89+/-15117.17 n=9, respectively) were not significantly different from 100% high load data (Fig.8F) though showing a tendency with Z scores of 1.00, and 1.18, respectively. However, moving 100% (Fig.8C), and 10% (Fig.8E) charged devices (821.74+/-4344.30 n=186, 1246.55+/-3223.11 n=131, respectively) were different from 100% high load (Fig.8G; 291.46+/-0.56 n=617) with Z scores of 1.67 and 3.39, respectively. These results show that different sensors may react differently when device is in a sleeping mode depending on a battery charge remaining. It appears that gyroscope data are more often accessed at lower battery charge when device is in a sleeping state and in motion.

Sleeping stationary vs. moving gyroscope data comparison, in contrast to accelerometer data, gyroscope’s between readings intervals distributions showed significant differences between stationary and moving devices when they were 100% charged and sleeping (Fig.8 B vs C, Z score = 3.01). 10% charged devices data distributions were insignificant but showed a tendency (Fig.8D vs E, Z score=0.99). Both cases when moving revealed that on average there were shorter polling intervals between the readings. In the high CPU load case, intervals were virtually identical (Fig.8F vs G, Z score=0.3).

Figure 8. 100% charged sleeping but moving device reads gyroscope data more often.

Chart, box and whisker chart

Description automatically generated

+ shows significant difference in the distributions of intervals between sensor readings compared with (F) high CPU load when device was stationary (Z test, score >1.645, alpha 0.05). ^ shows above comparisons but with (G) high CPU load when device was moving. & shows significant difference in distribution of between sensor reading intervals between stationary and moving devices under the same conditions (100%, 10% charged, and high CPU load). A - 100% charged plugged in, B – 100% charged, D - 10% charged device sleeping and stationary. C – 100% charged, E – 10% charged, device sleeping and moving. F and G – high CPU load, with app in background, when device was stationary or moving, respectively.

Closer look at the distributions of the intervals between gyroscope sensor data reading (Fig.9) for 100% and 10% charged devices revealed opposite relationships when compared to accelerometer (Fig.7). Figure 9 shows whisker plots of the intervals between the readings for 100% charged and 10% charged devices. Device movement while it was asleep (Fig.Bs) produced more compact clusters of shorter intervals between the gyroscope data readings with averages around 10s (Fig.9Bs) when compared with stationary phones intervals of 50s and 20s (Fig.9As), for 100% and 10% charged devices, respectively. Ultimately, that made moving phones averages lower than stationary, that was similar to the effect observed in accelerometer data, however displaying different or opposing dynamics.

Figure 9. Gyroscope data of moving devices shows clusters of shorter intervals between sensor data reading.

Chart, scatter chart

Description automatically generated Chart, scatter chart

Description automatically generated

A – denotes whisker box plot for sleeping stationary device, when B – sleeping moving device.

**Accelerometer and Gyroscope** sensors data recording while device was moving, or stationary were done simultaneously in this set of experiments. Averages and SDs of the between reading intervals are presented on Figure 10. Figure 10 A1 and A2 illustrate tethered device data while stationary and asleep, similar to Figures 6 and 8. Analyses did not reveal significant changes in averages or distributions between the moving and stationary devices. However, like accelerometer alone data, accelerometer and gyroscope data showed significant changes in distributions of intervals between reading on plugged in, 100% and 10% charged devices when compared with devices at high CPU load (Accelerometer: Fig.10A1,B1,D1 vs F1, 2284.50+/-6314.87 n=69, 1841.26+/-7085.69 n=214 with Z score=2.03, 1583.70+/-5122.55 n=220 with Z score=2.22, vs 291.37+/-2.38 n=617; Gyroscope: Fig.10A2,B2,D2 vs F2, 3093.29+/-7200.04 n=51, 1862.41+/-7147.52 n=217 with Z score=2.03, 1527.34+/-2898.29n=84 with Z score=2.21 vs 291.19+/-4.18 n=618). Analogous results were seen in the distributions when device was moving (Accelerometer: Fig.10C1,E1 vs G1, 2495.22+/-8213.45 n=54 with Z score=1.97, 1248.40+/-2898.29 n=144 with Z score=3.96 vs 291.54+/-3.42 n=617, Gyroscope: Fig.C2,E2 vs G2, 2496.83+/-8217.36 n=54 with Z score=1.97, 1231.32+/-2877.65 n=146 with Z score=3.95).

Figure 10. Moving and stationary devices showed similar mean intervals between the readings of simultaneously acquired accelerometer and gyroscope data.

Chart, bar chart

Description automatically generated

+ shows significant difference in the distributions of intervals between sensor readings compared with (F) high CPU load when device was stationary (Z test, score >1.645, alpha 0.05). ^ shows above comparisons but with (G) high CPU load when device was moving. Indices 1 and 2 refer to Accelerometer and Gyroscope interval measurements, respectively. A – depicts 100% charged plugged in device data. For 100% charged device B - shows stationary, and C – moving device data. For 10% charged device D – shows stationary, and E – moving device data. For 100% charged device at high CPU load F - shows stationary, G – moving device data.

For the consistency with previous data presentation, whisker box plots were used for simultaneously recorded data from two sensors too (Fig.11), even though there was not significant differences and also reliable tendencies. Since we recorded data simultaneously, we have an opportunity to check recordings for cross-correlation and see if there is a similarity between two sensors timing. Cross-correlation functions were calculated on intervals between sensor readings as accelerometer vs gyroscope for each experimental condition (Fig.12). 100% and 10% charged devices revealed robust correlation between the sensors around zero moment with +/-1ms lag and with some longer positive and negative lags. The most interesting result was when devices were at the high CPU load. In this case when device was stationary there was a little cross-correlation between the sensors, and when device was moving it even introduced great negative correlation. These observations can be supported by judging the outliers on Figure 11. It can be noticed that there is a similarity in the data points between accelerometer and gyroscope in 100% and 10% charged devices. Taken together it appears to show that accessing multi-sensor data could be affected in positive or negative way by itself and by the functional state of the device. That may impact data reliability in unwanted way.

Figure 11. Simultaneously obtained Accelerometer and Gyroscope data of stationary and moving devices.

Chart, scatter chart

Description automatically generated Chart, scatter chart

Description automatically generated Chart, scatter chart

Description automatically generated Chart, scatter chart

Description automatically generated

A – denotes whisker box plot for sleeping stationary device, when B – sleeping moving device.

Figure 12. Functional state (high CPU load) of the device showed greatest desynchronizing impact on between sensor cross-correlation.

A picture containing text, screenshot, line, number

Description automatically generated Chart, line chart

Description automatically generated A picture containing text, number, screenshot, font

Description automatically generated Chart

Description automatically generated Chart, line chart

Description automatically generated Chart, line chart

Description automatically generated

Figure Sample Accelerometer and Gyroscope interval alignments

A picture containing text, screenshot, font, line

Description automatically generatedA picture containing text, screenshot, font, line

Description automatically generated

A graph with blue and orange lines

Description automatically generated with low confidenceA picture containing text, screenshot, font, line

Description automatically generated

Squares represent intervals between accelerometer and circles represent intervals between gyroscope simultaneous data readings. Left axis displays scales for the accelerometer and right, for the gyroscope intervals. Bottom axis shows the index of the recording. Dashed rectangles show misalignment or change in the relationship in the intervals.

Samsung S22

To investigate differences between manufactures, we have collected the same as above data sets on a different form Google Pixel 6a but comparable by its specs Samsung S22 device. Results as described below appeared to be quite different between the different manufactures despite that we made all the efforts to keep environment as similar as possible, i.e. same OS, same set of default applications, etc. Following presents the data for Samsung S22 in the same fashion as Google Pixel 6a.

**Accelerometer** Samsung S22 study has revealed only high CPU load (Fig. 13G) had significant impact on the intervals between sensor data readings. That was in contrast to Google Pixel 6a device where plugged in device, for both when app was in background and foreground, had significant differences too (Fig.1A,B) and was very close to the polling rate of 200ms. S22 intervals averages were all within a similar range of 374.23+/-50.19 n=481 and 372.64+/52.59 n=483, 372.99+/-51.90 n=482 and 374.55+/-49.53 n=480, 372.6908714+/-52.50 n=482 and 376.96+/-45.85 n=477, for plugged in, 100% charged, and 10% charged, in fore-, and back- ground, (Fig.13A-F) respectively. Moreover, their distributions showed significant Z-test score around 7.0 when compared to the 100% charged at high CPU load device that had average interval between accelerometer sensor readings of 389.23+/-1.00 n=462. It is worth mentioning that all but high CPU load data obtained from Samsung S22 showed modest variability (SD) that was much greater, 10 to 50-fold, that from Google Pixel 6a.

Closer look at the intervals distributions of plugged in, 100%, and 10% charged devices (Fig.14A), revealed that they were similar and had two populations with averages of 200 and 390ms. 100% charged but at a high CPU load showed only one population with average of 390ms (Fig.14B), indicating that the differences seen were caused by omission of shorter intervals. To characterize distributions between foreground and background application activity Z tests were conducted. There were no significant differences detected, although the tendency was noticed in the 100% charged devices with Z score of 0.78.

Figure 13. High CPU load has a significant impact on the intervals between accelerometer sensor’s data access.

Chart

Description automatically generated

+ shows significant difference between high CPU load distributions of intervals between sensor readings (Z test, score >1.645, alpha 0.05). A - 100% charged device, plugged in, app is in foreground, and B when app is in background. C -100% charged device, app is in foreground, and D when app is in background. E - 10% charged device, app is in foreground, and F when app is in background. G - 100% charged device, CPU is in high load, with app collecting accelerometer data being in background.

Figure 14. High CPU load eliminates modality in accelerometer reading intervals distribution.

Chart, histogram

Description automatically generated Chart, bar chart

Description automatically generated

A – representative dual mode distribution histogram of the intervals between acccelerometer data readings for 100% charged device, app is in background. B – single mode distribution histogram of the intervals for 100% charged and at high CPU load device.

**Gyroscope**, intervals between reading the data from the sensor showed very similar profiles to the accelerometer in Samsung S22 device (Fig.13). Moreover, they were similar to Google Pixel 6a too (Fig.3), however, showed significant changes in the distributions across all of them when compared to 100% charged, high CPU load device. Namely, 100% charged plugged in, 100% and 10% charged in fore-, and back-ground interval distributions with averages of 374.20+/-50.29 n=480 and 373.81+/-50.80 n=481 (Fig.15A and B), 377.34+/-45.07 n=477 and 375.34+/-48.41 n=479 (Fig.15C and D), 374.21+/-50.06 and n=480 372.66+/-52.57 n=482 (Fig.15E and F), respectively, showed significant Z test scores of around 6 against 100% charged at high CPU load distribution that had average of 388.81+/-8.86 n=462 (Fig.15G). Distributions between foreground and background similarly did not show significant differences, but there was a Z score of 0.66 in the 100% charged devices.

Gyroscope sensor reading intervals distributions showed similar dual mode shapes with averages of 200 and 390ms for all conditions except when device was at high CPU load with an average of 390ms. Figure 16 shows (A) representative sample of the dual mode and (B) high CPU load distributions of the between reading intervals. Since analogous omission can be seen that may be pointing at the fact that different sensors operate under the same constrains imposed by the device or operating systems.

Figure 15. High CPU load has a significant impact on the intervals between gyroscope sensor’s data access.

Chart

Description automatically generated

+ shows significant difference between high CPU load distributions of intervals between sensor readings (Z test, score >1.645, alpha 0.05). A - 100% charged device, plugged in, app is in foreground, and B when app is in background. C -100% charged device, app is in foreground, and D when app is in background. E - 10% charged device, app is in foreground, and F when app is in background. G - 100% charged device, CPU is in high load, with app collecting accelerometer data being in background.

Figure 16. High CPU load eliminates modality in gyroscope reading intervals distribution.

Chart, histogram

Description automatically generatedChart, bar chart

Description automatically generated

A – representative dual mode distribution histogram of the intervals between gyroscope data readings for 100% charged device, app is in background. B – single mode distribution histogram of the intervals for 100% charged and at high CPU load device.

**Accelerometer and Gyroscope** simultaneous sensor data reading on Samsung S22 device showed fairly similar averages and statistics (Fig.17) when compared to the separate sensor data reading (Fig.12 and 15). The same situation was observed with Google Pixel 6a device. Distribution histograms for multi-sensor data readings (data not sown) revealed the same dual mode shapes as we have seen before in single sensor situations (Fig.13, 16).

Based on just two vs. single sensor reading and two different devices, we can state that there were no noticeable changes in how sensor’s data were accessed by the data acquisition application. There might be discrepancies when substantially more numbers of sensors are accessed though, that may be an aim for the future research projects.

Figure 17. High CPU load has a significant impact on the distribution of intervals between sensor data reading.

Chart, bar chart, box and whisker chart

Description automatically generated

+ show significant differences in the distributions of intervals between accelerometer or gyroscope sensor readings (Z test, score>1.65, p<0.05) when compared with high CPU load with sensor reading app running in foreground and background, respectively. Indices 1 and 2 refer to Accelerometer and Gyroscope interval measurements, respectively. A - 100% charged device, plugged in, app is in foreground (1- 372.32+/-53.01 n=483, Z score=7.01, 2- 373.45+/-51.20 n=481, Z score=6.65), and B when app is in background (1- 371.11+/-54.64 n=484, Z score=7.30, 2- 375.76+/-47.53 n=478, Z score=6.08). C -100% charged device, app is in foreground (1- 384.14+/-29.91 n=468, Z score=3.69, 2- 383.32+/-32.25 n=469, Z score=3.80), and D when app is in background (1- 378.90+/-42.15 n=475, Z score=5.35, 2- 382.12+/-35.29 n=471, Z score=4.21). E – 10% charged device, app is in foreground (1- 378.14+/-43.72 n=476, Z score=5.54, 2- 378.33+/-43.13 n=476, Z score=5.38), and F when app is in background (1- 378.94+/-42.16 n=475, Z score=5.33, 2- 379.75+/-40.16 n=475, Z score=5.01). G - 100% charged device, CPU is in high load, app is in background (1- 389.25+/-0.90 n=462, 2- 389.03+/-4.61 n=463).

**Device movement**

When Samsung S22 device, similar to Google Pixel 6a described above, was asleep, we documented the greatest increases in the intervals between the readings. Since there were differences found between stationary and moving Pixel 6a device, we repeated the same experiments using the same trajectory with the help of robotic arm from Elephant Robotics and acquired accelerometer or/and gyroscope sensor data.

**Accelerometer** data from stationary or moving Samsung S22 phone showed no significant differences (Fig.18). However, when compared to Google Pixel 6a phone (Fig.6) showed opposing trends, that is averages of 100% and 10% charged stationary (Fig.18B, 25581.17+/-38216.21 n=6 and Fig.18D, 2210.84+/-5257.84 n=77, respectively) vs moving (Fig.18.C, 51221 n=1 and Fig.18E, 5701.9+/-9977.66 n=10) tend to increase in S22 vs decrease in Pixel 6a. These tendencies are supported by Z test scores, though calculated only for 10% charged devices, of 1.09. Distributions of between polling intervals of stationary plugged in (Fig.18A 1609.51+/-6281.53 n=105), and 10% charged (Fig.18D) devices were significantly different from 100% high load data (Fig.18F; 389.22+/-1.00 n=462) with Z scores of 2.00, 3.04, respectively. Similarly, moving 10% (Fig.18E) charged device intervals distribution was different from 100% high load (Fig.18G; 389.29+/-1.22 n=462) with Z score of 1.68. Data from 100% charged device (Fig.18C) could not be compared because there was just one recording when device was moving. It appears as accelerometer data are much rarely accessed in a sleeping state and even less when moving.

Figure 18. Moving vs stationary devices showed similar mean intervals between the readings of accelerometer data.

Chart, bar chart

Description automatically generated

+ shows significant difference in the distributions of intervals between sensor readings compared with (F) high CPU load when device was stationary (Z test, score >1.645, alpha 0.05). ^ shows above comparisons but with (G) high CPU load when device was moving. A - 100% charged plugged in, B – 100% charged, D - 10% charged device sleeping and stationary. C – 100% charged, E – 10% charged, device sleeping and moving. F and G – high CPU load, with app in background, when device was stationary or moving, respectively.

**Gyroscope** data from stationary or moving Samsung S22 phone are presented on Figure 19. 1st bar (Fig.18A) shows data from plugged in to a charger device just for the illustration purposes. The tests showed that only the distribution of between polling intervals of stationary plugged in (Fig.19A, 2659.63+/-4322.46 n=65) was significantly different from high CPU load data (Fig.19F; 388.81+/-8.86 n=462) with Z scores of 4.24. 100% (Fig.19B) stationary and moving (Fig.19C) charged devices (180000, 123373, respectively) had only one point in the data collected and were excluded from analyses. Distributions of 10% charged stationary and moving devices (Fig.19D, 6237.89+/-15117.17 n=9, Fig.19E, 11087+/-28023.67 n=12, respectively) were not significantly different from 100% high load (Fig.19F; 388.81+/-8.86 n=462, Fig.G,388.85+/-9.13 n=462) but showed a trend with Z scores of 1.16 and 1.32, respectively. Comparison with the Google Pixel 6s device was difficult to conduct due to the very small datasets. Though, Google Pixel 6a had more significant differences in the distributions of the intervals, specifically when device was 100% and 10% charged and moving.

Figure 19. Moving device reads gyroscope data similarly to stationary.

Chart, bar chart

Description automatically generated

+ shows significant difference in the distributions of intervals between sensor readings compared with (F) high CPU load when device was stationary (Z test, score >1.645, alpha 0.05). A - 100% charged plugged in, B – 100% charged, D - 10% charged device sleeping and stationary. C – 100% charged, E – 10% charged, device sleeping and moving. F and G – high CPU load, with app in background, when device was stationary or moving, respectively.

**Accelerometer and Gyroscope** sensors simultaneous data recording while device was moving, or stationary are shown on Figure 20. Averages and SDs of the between reading intervals are similar to the single sensor data recording (Fig.18,19). Analyses did not reveal significant changes in averages or distributions between the moving and stationary devices, but both accelerometer and gyroscope data showed significant changes in distributions of intervals between reading on plugged in, 100% in stationary position and 10% charged devices when moving when compared with devices at high CPU load (Accelerometer: Fig.20A1,B1 vs F1, 1403.58+/-6400.56 n=125 Z score=1.77, 1841.26+/-7085.69 n=86 Z score=1.90 vs 389.25+/0.90 n=462; Gyroscope: Fig.10A2,B2 vs F2, 1403.58+/-6400.56 n=125 Z score=1.77, 1862.41+/7147.52 n=85 Z score=1.90 vs 389.03+/-4.61 n=463). 10% charged stationary distributions were excluded from the analyses because of the very low number of points in their datasets (Fig.20D1,D2, 89170.5+/-21540.59 n=2, 89170+/-21535.64, n=2). Analogously, only the distributions when device was 10% charged and moving were analyzed (Accelerometer: Fig.20E1 vs G1, 1489.84+/-4312.97 n=94 Z score=2.47 vs 389.24+/0.98 n=462, Gyroscope: Fig.20E2 vs G2, 1460.81+/-4271.58 n=96 Z score=2.46 vs 389.04+/-4.62 n=463) but stationary has just too few points (Fig.20C1,C2 30739.6+/-46569.1 n=5, 30739.6+/-46567.47 n=5).

Figure 20. Moving and stationary devices showed similar mean intervals between the readings of simultaneously acquired accelerometer and gyroscope data.

Chart, bar chart

Description automatically generated

+ shows significant difference in the distributions of intervals between sensor readings compared with (F) high CPU load when device was stationary (Z test, score >1.645, alpha 0.05). Indices 1 and 2 refer to Accelerometer and Gyroscope interval measurements, respectively. A – depicts 100% charged plugged in device data. For 100% charged device B - shows stationary, and C – moving device data. For 10% charged device D – shows stationary, and E – moving device data. For 100% charged device at high CPU load F - shows stationary, G – moving device data.

Due to the scarcity of the data we do not show here the distribution histograms or whisker box plots, however, no significant differences were documented there. Checking recordings for cross-correlation between Gyroscope and Accelerometer simultaneously obtained data is presented on Figure 21. Only high CPU load from stationary data and 100% and 10% charged and moving devices had enough points to do these calculations. Nevertheless, as compared to Google Pixel 6a (Fig.12) cross-correlation lags showed the same relationships for the available data. The only big difference was missing 15s lag in 100% charged stationary Samsung S22 device when compared to Google Pixel 6a. We refrain from drawing any conclusions from these comparisons due to the not complete picture of the data.

Figure 21. Cross-correlation between Accelerometer and Gyroscope data simultaneously obtained from stationary and moving devices.

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generatedChart, line chart

Description automatically generated

**Discussion**

**Stationary awake devices**

Sensors data reliability is very important in various scenarios of smartphone usage, whether from user perspectives or research requirements. Utilization of the sensors data is still limited and resorted to specific and easily identifiable phenomena, step, fall, jump etc. In general approach to the data easily identifiable activity may not be present, thus, it is imperative for the data to be reliable and usable, especially when trying to develop learning algorithms on the sensors data one has to be sure in their accuracy and consistency. Acquisition frequency is also a great factor in usability of the sensors data, the general rule is the higher frequency the better. This will introduce higher battery consumption and may discourage end users from it. Moreover, it has been reported that there is a great deal of drift in the data that has to be cleaned out by using Fourier transform and band pass filtering mostly to attenuate low frequency noise. This indicates an existence of signals with longer durations that affect the sensor readings.

Similarly, in our results, although we saw that overall there was no significant changes in average times between sensor readings, but there was noticeable change in distributions with longer times between the readings. Especially, it was pronounced when devices were sleeping. So, the most reliable conditions for sensor data acquisition would be when devices are active. However, even in this state there was about 90ms delay from the polling rate. This was most likely introduced by the operating system itself and would be hard to overcome in normal settings. Surprisingly, lower battery charge did not introduce significant deviations in the intervals between the readings. High CPU load was also similar with the rest of the averages. Taken together, it may indicate that our devices, that were the latest models of Android phones from Google and Samsung, were able to handle the tasks that we programmed them to do without much of the effort. So, accessing the sensors data by itself is pretty low power procedure.

**Sleeping and moving devices**

When mobile device enters a sleep mode all nonessential threads get suspended. Thus, our data acquisition was basically quickly stopped, however triggered from time to time. One fact should be mentioned here is that lower battery charge showed beneficial impact on the intervals between sensors reading. Namely, sensors were read 1.5-2 times more often. This trend even improved when we introduced movement to the sleeping devices. The fact that lower battery charge has benefits will require further exploration to understand what drives this behavior.

**Multi-sensor reading**

Overall whether we accessed single or multiple sensors at the same time, the averages and distributions of intervals between the readings were similar. That allows us to conclude that all sensors work separately and accessing their data is an independent process. Cross correlation analyses between simultaneously read sensors without high CPU load revealed hi positive correlation with the major lag in the middle (around ms) and minor lags at 5ms, 17ms. Meaning that greater intervals between the readings of one sensor are followed by greater intervals in the other. That fits the assumption that the system is controlling the access to all sensor data. However, high CPU loads revealed no, when stationary, or negative correlation with the lag peak value -1ms, when device was moving. Thus indicating that greater intervals between the reading one sensor lead to shorter ones in the other. Perhaps high CPU load imposed enough strain on the system itself or on the processing power that lead to this effect when one sensor is favored when compared to the other. That potentially may introduce a weakness in the data set of one of the sensors.

**Different device manufacturers**

Comparison across two different devices allowed as to conclude that there were substantial differences introduced by the hardware implementations by different manufactures. That has to be taken into account when collecting of the sensor data is done on nonidentical devices. In our experiments we documented appearance of dual mode distributions of the intervals between sensor readings in Samsung S22 device. Google 6a phone also showed similar behavior but only when it was tethered and only in one sensor, i.e. accelerometer. So, this duality seems to be inheriting but less of the concern in one device then the other.