Low battery, high CPU load and a number of sensors impact the timing of the sensor data readings on mobile devices.

Boris Gafurov  
Thinksense Inc.McLean, 8200 Greensboro Dr, Suite 700, McLean, VA 22102, USA  
bgafurov@thinksense.com

Mohsen Rajabpour  
Thinksense Inc.McLean, 8200 Greensboro Dr, Suite 700, McLean, VA 22102, USA mrajabpour@thinksense.com

Angelos Stravou  
Thinksense Inc.McLean, 8200 Greensboro Dr, Suite 700, McLean, VA 22102, USA angelos@thinksense.com

*Abstract*—Sensor data reliability plays a critical role in applications that collect and uses this data. The types of effects that can invalidate the data could be physical data drift, bias, missing data or outliers, to name a few. These can arise from several factors, where power management and operating systems, that are built to conserve the power, are of the most concern. This study assessed an impact of the battery charge, functional state of the device and a number of sensors listened to on variability in the timestamps of the sensor data readings, that reflects the quality of the data acquired. It has been found that lower battery charge and high CPU loads have significantly impacted a mean interval between the sensor data readings, however at a very small scale 1-2ms but in different ways on different devices. Most importantly, interval distributions showed introduction of occasional longer intervals with up to 60ms increase in duration that decreases sampling bandwidth and may introduce aliasing distortions. The number of sensors included in experiments and device movement changed the variability too but not on all devices tested. These findings indicate that the process of the sensor data reading is highly dependent on the device’s conditions that decrease the reliability of the data.

Keywords—mobile devices, sensors, power management, data reliability.

# Introduction (*Heading 1*)

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 [1]. 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 [2]. In this study we are concentrating on how device’s power management affects sensor data quality. While we can control the sensor data reading frequency, we do not have access to the wait in a queue for the buffer reading. Therefore, we do not have access to the time when sensor updates the buffer. However, sensing is much faster (>100Hz) [3] than we will read the sensor data (5Hz), hence, should not introduce significant differences in the timestamps by itself. In addition, and for the future research, 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 sensor 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 accelerometer and gyroscope sensors, which are most commonly used for health and other research [4][5][6][7].

# Methods

For this study we chose Google Pixel 6a and Samsung S22 smart phones with Android 11 operating system. These are modern and powerful devices with variety of sensors from which we can capture the data. They also have advanced power management system and should be ideal for the experiment we are planning to conduct. Data from sensors will be acquired in two modes, when device is stationary or moving, with the later meant to provide more realistic data sets. Device's displacement will be accomplished using Elephant robotic arm myCobot 280 using the same trajectory stored in the arm's memory.

## Single-sensor study

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.

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 is expected to produce the greatest variability in the readings that will provide insight into power management control of and access to the sensor’s data.

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 are going to 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.

## Multi-sensor study

Identify applicable funding agency here. If none, delete this text box.

In the next set of experiments, we will simultaneously record data from two sensors using the same recording paradigm. The variability of timestamps in accelerometer and gyroscope sensors will be assessed. This data will elucidate the impact on reliability of reading from multiple sensors in various conditions. In addition to the sensor readings 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 a hardware and/or power management introduces to multi-sensor data readings.

## Data and Statistical analyses

Data on variability expressed as means and standard deviations can be compiled from our experiments and statistically analyzed for significant differences. The variability of the mean interval between sensor readings across different acquisition conditions will be analyzed for the significant changes using Excel 2 tailed t-test function. To show significant changes in the distributions of the intervals between sensor readings, data will be analyzed using two sample Z-score test formula written in Excel. 3D sensor data converted into a simple geometrical distance or rather acceleration or rotational speed for gyroscope absolute value would be routinely tested against the intervals between sensor readings for cross correlation to ensure that there are no underlying relationships between them. Cross correlation function (CCF) vs. lag relationships will be calculated with no data wrapping and plotted out in Excel using custom written macros. Throughout the text of this manuscript averages will be expressed as a mean ± standard deviation (SD) followed by number of measurements in parentheses.

## Application design

We used Android Studio to develop our app to collect sensor data. Briefly, app is designed to listen to single or multiple sensors without buffering with 200ms polling rate. When sensor OnSensorChanged listener receives the input from the sensor, app checks if the time, since the last data was recorded for this sensor, is greater than our default polling rate of 200ms, then x, y, z metrics of the sensor data are stored in local database. To make OS not to suspend our app, when it goes into background, or interfere with the app functionality depending on the battery status, the Wake Lock mechanism was implemented and battery optimization was disabled in the app. Experiments under each condition were run for 5min, that was determined to be the maximum time during which battery percentage did not change under the high CPU load. Middle 3min portion of that was used for further off-line analyses.

# Results

In general, intervals between the reading were very close to the polling rate of 200ms. Despite that, variability of the intervals between sensor data readings showed significant differences in means across all the conditions, albeit a small one, on a scale of 0.1-0.2ms. At times, under certain conditions, intervals showed greater up to 60ms but intermittent increase that, ultimately, changed statistics between the interval distributions. 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 the values from sensor readings, normally R was less than 0.1, meaning that sensor activity itself did not, as expected, impose significant stress on sensor functionality [1].

## Google 6a phone

Accelerometer study has revealed that when Google Pixel 6a device was 100% charged and stationary intervals between the sensor data reading were significantly different from 10% and 100% charged with CPU working at a high load (Fig.1 A vs C and E). These results were based on the comparisons of both interval means with t-test and interval distributions with Z-test. Moving device showed significant difference only between 100% charged and high CPU load cases, again in both interval means and distributions (Fig.1 B vs F). Moving vs. stationary comparisons were significant only in 10% charged device and only in interval distributions (Fig.1 C vs D). We must acknowledge the absolute differences between the means were small on a scale of the 0.1-0.2ms (e.g., Fig.1 A, C, and E showed 200.35±0.58ms, 200.46±0.6ms, and 200.43±0.52ms, respectively) but highly significant due to the great n=898.

A graph with blue and orange bars

Description automatically generated

1. 100% charged device shows lower intervals between the sensor readings. \* - significant mean difference obtained by t-test (\* - p<0.05, \*\* - p<0.01, \*\*\* - p<0.001, n=898). # - significant difference in interval distributions obtained by Z-test (score > 1.645, alpha < 0.05). Blue A - 100% charged battery, C – 10% charged battery, E – 100% charged battery with CPU at high load when device was stationary. Orange B, D, F – same conditions but when device was moving.

Closer inspection of the sample interval distributions (Fig.2) showed faster rise and decay in the 100% charged device however with occasional high duration intervals. Whisker box plot inset presents a better view for the high duration intervals (Fig.2 inset).

The general tendency in Google 6a accelerometer data shows that 10% charged battery and high CPU load create a little bit higher though significant mean interval. And, interestingly, it seems to diminish longer intervals between the sensor data readings.

A graph of different colored lines and numbers

Description automatically generated

1. Interval distributions of accelerometer sensor data access in stationary devices show decrease in long intervals at lower battery charge and high CPU load. Representative interval distributions from stationary devices for above analyses. Order in the bar graph and whisker box plot from left to right is as follows: 100% charged (green), 10% charged (blue), and 100% charged at high CPU load (red). Dotted lines represent a moving average of the bar graphs. Inset shows whisker box plot for corresponding data distribution.

Gyroscope study, unlike accelerometer one, mostly showed significant differences in moving devices not the stationary ones (Fig.3). Stationary devices showed no significant differences in means between different conditions. The only statistical difference was detected between interval distributions in 100% charged vs high CPU load comparison (Fig.3 A vs E). However, there were differences between stationary and moving devices in 100% charged and high CPU load cases both in means and interval distributions (Fig.3 A vs B, E vs F). On the other hand, moving devices revealed that intervals between the sensor readings in 10% charged device were significantly different from 100% and high CPU load cases (Fig.3 D vs B, D vs F). That was opposite to the Accelerometer study.

A graph of a graph with blue and orange bars

Description automatically generated

1. Gyroscope data reading intervals were the lowest at high CPU load. \* - significant mean difference obtained by t-test (\* - p<0.05, \*\* - p<0.01, \*\*\* - p<0.001, n=898). # - significant difference in interval distributions obtained by Z-test (score > 1.645, alpha < 0.05). Blue A - 100% charged battery, C – 10% charged battery, E – 100% charged battery with CPU at high load when device was stationary. Orange B, D, F – same conditions but when device was moving.

To get an insight into low battery charge impact on the intervals we plotted out the distributions for all experimental cases while device was moving (Fig.4). It is clearly can be seen that high SDs in 100% charged and high CPU load were caused by occasional long intervals, i.e., Figure 3B, D, and F showed 200.7±3.4ms, 200.4±0.9ms, and 200.7±3.0ms, respectively. On the other hand, low battery charge seemed to be beneficial for acquiring sensor data more consistently.

A graph of different colored lines

Description automatically generated

1. Gyroscope data reading interval distributions at high CPU load had the lowest width. Representative interval distributions from stationary devices and moving device when CPU was at high load for above analyses. Order in the bar graph and whisker box plot from left to right is as follows: 100% charged (green), 10% charged (blue), and 100% charged at high CPU load (red). Dotted lines represent a moving average of the bar graphs. Inset shows whisker box plot for the whole data range to illustrate the histogram distributions shown.

Accelerometer and Gyroscope study, when two sensors were simultaneously recorded from, revealed no significant differences in means in the contrast to the single sensor recordings (Fig.5). However, there were significant differences between the distributions of the intervals. Firstly, there was significant difference in interval distributions between stationary and moving devices when they were 100% charged for both accelerometer and gyroscope, and at high CPU load but only for accelerometer. There was no difference between the cases when device was stationary. However, there were differences when device was moving but only between 10% charged device and both 100% charged and high CPU load for the accelerometer reading intervals. For gyroscope, the difference was detected only when compared with high CPU load. The later observation was reminiscent of the one that we saw in gyroscope alone study, although not for the accelerometer. Somehow, accelerometer data acquisition began acting similar to gyroscope, or perhaps it was influenced by it. But this observation needs further investigation.

A graph of a graph with blue and orange bars

Description automatically generated

1. Mean intervals between data reading from accelerometer and gyroscope simultaneously. ± - significant difference in interval distributions obtained by Z-test (score > 1.645, alpha < 0.05). Blue A - 100% charged battery, C – 10% charged battery, E – 100% charged battery with CPU at high load when device was stationary. Orange B, D, F – same conditions but when device was moving. Indexes 1 or 2 denote Accelerometer or Gyroscope sensor readings, respectively.

A graph of different colored bars

Description automatically generated

1. Sample interval distribution from accelerometer and gyroscope that were accessed simultaneously. Sample distribution histograms for the Accelerometer and Gyroscope study above. Order in the bar graph and whisker box plot from left to right is as follows: 100% charged stationary accelerometer (blue), 100% charged stationary gyroscope (orange), 100% charged moving accelerometer (grey), and 100% charged moving gyroscope (yellow). Dotted lines represent a moving average of the bar graphs. Inset shows whisker box plot for the whole data range to illustrate the histogram distributions shown.

Since in this study we collected pairs of data sets, i.e. intervals between sensor data access for both gyroscope and accelerometer, in the course of one experiment, we can investigate if there is any degree of relationship between them. Fig.7 shows cross-correlation vs lag plot for each condition for moving or stationary device. The highest and very distinctive, though low to be considered as significant (r=0.2), correlation peak appeared for stationary device when battery was at 10%. Thus, there seems to be a distinct interplay between the sensors, however statistically insignificant.

A graph on a paper

Description automatically generatedA graph on a white background

Description automatically generatedA graph of a stationery

Description automatically generated

A graph with a line graph

Description automatically generatedA graph with a line graph

Description automatically generatedA graph of a graph

Description automatically generated with medium confidence

1. Cross-correlation between gyroscope and accelerometer reading intervals showed weak interplay when device was 10% charged and stationary. Cross-correlation analyses of the Accelerometer vs Gyroscope intervals in stationary and moving conditions.

So, it is clearly can be seen that multi-sensor reading introduced changes in the data collected on Google Pixel 6a device. Whether it is due to the application design or hardware implementation will require further investigation.

## Samsung S22 phone

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 for Google Pixel 6a.

Accelerometer study revealed that sensor data collected when phone was 100% charged and operating at high CPU load were highly and significantly different from both 100% and 10% charged phones judging by the means and interval distributions (Fig.8). Mean values were different by up to 2ms, as seen on a Fig.8 (e.g., Fig. 8 A, C, and E showed 201.9±1.8ms, 202.0±2.2ms, and 200.9±0.9ms, respectively). On the Figure 9, it can be seen that the distribution of the intervals between the sensor data that had been read for high CPU load was much narrower than the others too. These results were strikingly different from the Google Pixel 6a phone but supported the reports from other authors about differences between the sensor data obtained on the devices from different manufactures.

A graph of a graph with a bar graph

Description automatically generated with medium confidence

1. Accelerometer data reading intervals were the lowest at high CPU load. \* - significant mean difference obtained by t-test (\* - p<0.05, \*\* - p<0.01, \*\*\* - p<0.001, n=898). # - significant difference in interval distributions obtained by Z-test (score > 1.645, alpha < 0.05). Blue A - 100\% charged battery, C – 10\% charged battery, E – 100\% charged battery with CPU at high load when device was stationary. Orange B, D, F – same conditions but when device was moving.

A graph of a function

Description automatically generated with medium confidence

1. Accelerometer data reading interval distributions at high CPU load had the lowest width. Representative interval distributions from stationary devices for above analyses. Order in the bar graph and whisker box plot from left to right is as follows: 100% charged (green), 10% charged (blue), and 100% charged at high CPU load (red). Dotted lines represent a moving average of the bar graphs. Inset shows whisker box plot for the whole data range to illustrate the histogram distributions shown.

Gyroscope study on Samsung S22 phone revealed similar tendencies, i.e. the high CPU load showed the lowest and highly significant values by means (around 1ms, i.e., Fig.10 A, C, and E showed 202.0±1.8ms, 202.0±1.9ms, and 201.0±2.0ms, respectively) and interval distributions (Fig.10). Moreover, there was a significant difference between stationary and moving devices in high CPU load case (Fig.10 E vs F). Interval distributions for gyroscope sensor data reading were similar to accelerometer (Fig.11). It can be seen that there is similar tight distribution for stationary high CPU load (although with one very long 252ms outlier) and much wider distributions for the others.

A graph with blue and orange bars

Description automatically generated

1. Gyroscope data reading intervals were the lowest at high CPU load. \* - significant mean difference obtained by t-test (\* - p<0.05, \*\* - p<0.01, \*\*\* - p<0.001, n=898). # - significant difference in interval distributions obtained by Z-test (score > 1.645, alpha < 0.05). Blue A - 100% charged battery, C – 10% charged battery, E – 100% charged battery with CPU at high load when device was stationary. Orange B, D, F – same conditions but when device was moving.

A graph of different colored lines and numbers

Description automatically generated

1. Gyroscope data reading interval distributions at high CPU load had the lowest width. Representative interval distributions from stationary devices and moving device when CPU was at high load for above analyses. Order in the bar graph and whisker box plot from left to right is as follows: 100% charged (green), 10% charged (blue), 100% charged at High CPU load (red), and 100% chargedat high CPU load moving (dark blue). Dotted lines represent a moving average of the bar graphs. Inset shows whisker box plot for the whole data range to illustrate the histogram distributions shown.

Accelerometer and Gyroscope study in general produced very analogous results as single sensors studies (Fig.12). Namely, data obtained for the high CPU load, however only for stationary devices, were highly significant for 100% and 10% charged devices. Moreover, data from 100% charged moving device showed significant change in the distribution when compared with high CPU load (Fig. 13).

A graph of a bar graph

Description automatically generated

1. Accelerometer and Gyroscope data reading intervals were the lowest at high CPU load stationary devices. ± - significant difference in interval distributions obtained by Z-test (score > 1.645, alpha < 0.05). \* - significant mean difference obtained by t-test (p<0.001, n=898) between E1,2 and A1,2 or C1,2. # - significant difference in interval distributions obtained by Z-test (score > 1.645, alpha < 0.05) between E1,2 and A1,2 or C1,2. Blue A - 100% charged battery, C – 10% charged battery, E – 100% charged battery with CPU at high load when device was stationary. Orange B, D, F – same conditions but when device was moving. Indexes 1 or 2 denote Accelerometer or Gyroscope sensor readings, respectively.

A graph of different colored bars

Description automatically generated

1. Accelerometer and Gyroscope reading interval distributions. Sample distribution histograms for the Accelerometer and Gyroscope study above. Order in the bar graph and whisker box plot from left to right is as follows: 100% charged stationary accelerometer (blue), 100% charged stationary gyroscope (orange), 100% charged at high CPU load stationary accelerometer (grey), and 100% charged at high CPU load stationary gyroscope (yellow). Dotted lines represent a moving average of the bar graphs. Inset shows whisker box plot for the whole data range to illustrate the histogram distributions shown.

Clearly, Samsung S22 phone showed similarities in multi and single sensor data acquisitions, which was on a contrary to the Goggle Pixel 6a phone, that may indicate that the differences that we saw with Google 6a could be attributed mainly to the hardware design, since Samsung S22 acquisitions were mostly similar.

To be consistent with Google Pixel 6a phone study we investigated if there was any degree of relationship between gyroscope and accelerometer data. Figure 14 shows cross-correlation vs lag plot for each condition for moving or stationary device. In contract to Google Pixel 6a that had highly distinctive peak with r=0.2 and various others around 0.1, Samsung S22 did not show the same or anything close to that. All cross-correlations vs lag plot were rather unimpressive. That may account for the lack of any relationships between gyroscope and accelerometer (as it should be) in Samsung, however Google might have displayed some background interplay between the sensors (but it should not have).

A graph on a white background

Description automatically generatedA graph on a paper

Description automatically generatedA graph on a paper

Description automatically generated

A graph on a white sheet

Description automatically generatedA graph with a line graph

Description automatically generatedA graph with a line graph

Description automatically generated

1. Cross-correlation between gyroscope and accelerometer reading intervals showed they are accessed mostly independently. Cross-correlation analyses of the Accelerometer vs Gyroscope intervals in stationary and moving conditions.

# Discussion

## Stationary 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. [8][9][10]. 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. Many factors can interfere with that. Acquisition frequency is one of them. The higher frequency the better information we can get about the device and/or owner activities [11]. However, higher frequencies will introduce higher storage and most importantly battery consumption that by itself may discourage end users from it and also may have an impact on a data acquisition from the sensors.

Aside from the battery charge level, sensor data acquisition has been reported to have a great deal of drift. The data has to be cleaned out by using Fourier transform and band pass filtering mostly to attenuate low frequency noise, as described for example here [12]. This indicates an existence of signals with longer duration that affect the sensor readings.

In our results, although we saw significant changes in average times between sensor readings, they were no more than 2ms. However, sometimes reading from the sensors was strongly delayed up to 60ms that had been noticed by the distributions. Despite that these delays were very rare and present in both devices that may be, most likely, caused by the operating system. There is a need for more research in this direction to find out exactly why, when and what caused them.

Surprisingly, lower battery charge did not introduce great deviations in the intervals between the readings. However, high CPU load did on a contrary, i.e., it reduced the intervals leading to more consistent sensor data access. One explanation can be because data were accessed from the cache and not actually from the memory buffer that is managed by PMI and OS or it was something similar to the jitter effect in smartphones [13][14].

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 low power procedure.

## Moving devices

When mobile device was moving during the experiments it introduced several significant changes. They were not across all the cases, i.e. 100%, 10% charged or high CPU load and both sensors, but mostly for 100% charged and high CPU load and for both sensors. There must be something specific about the sensors that makes them fire on change event differently when moving at these conditions. Another player in these changes could be operating system that sporadically delays the sensor data access.

## Multi-sensor reading

In our simultaneous reading from two sensors, we noticed that on Google Pixel 6a phone interval differences between different conditions changed. E.g., accelerometer reading at 10% battery charge became significantly different from both 100% charged and high CPU load cases, etc. Intervals between the sensors readings on Samsung S22, on a contrary, stayed mostly the same, i.e. high CPU load case produced mostly the lowest intervals when compared to both of the others. Thus, Samsung’s sensors work separately and accessing their data is an independent process, when Google phone has variability in this process. Cross correlation analyses between simultaneously read sensors showed some correlation in intervals between data access on a Google phone, though insignificant but maybe pointing on the some interplay between gyroscope and accelerometer. Samsung phone showed almost no cross-correlation at all.

## 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 just by judging the multi-sensor readings. That has to be taken into account when collecting of the sensor data is done on nonidentical devices. In our experiments we documented that one device consistently showed same relationships in the data, i.e., Samsung S22 high CPU load was the lowest across the board. Google phone showed differences between accelerometer and gyroscope in single sensor acquisition mode, and also when they were simultaneously recorded from.

## Data acquisition limitations

As it has been discovered our data reading from sensors showed random gaps in timing, up to around 60ms. Whether actual x, y, and z metrics of accelerometer and gyroscope were reflecting real device movement could not be answered by this study, however, that brought a question if, or to what extent this intermittent paucity can affect the researches that rely on data consistency, e.g. signaling, continuous monitoring, etc. From the Nyquist sampling theorem to avoid aliasing one has to ensure at least 2 fold acquisition frequency over the possible bandwidth of the signals. Discovered 60ms gaps, or decrease in sampling frequency, may drastically shift an upper cut off limit down to 8.3Hz, however intermittent. This limitation may impact high frequency data analyses where the bandwidth of the sampled signals may drop suddenly all the way down to about 8Hz and lead to a missing signals or activity, and essentially introduce erroneous lower frequency activity.

If one wants tracking the movements and visualizing it via world coordinate system transformation for acceleration, they will drift of course because of those longer accelerations at certain times. As reported by authors [12] acquisition ranges from 20Hz-200Hz for detecting subject movement. With 60ms extra, it easily may, even for the lowest 20Hz frequency, double the time subject spent ac/de-celerating at selected instances that would quadruple the displacements. Authors in [12] actually reported over estimation limitations in their research that might as well have been caused by those extra gaps. High-pass digital filter was proposed as a remedy to attenuate lower part of the spectrum but it only would minimize it. So, technically, one needs to discard this point as a continuous and devise a reset point for trajectory integration routine.

Another approach to this limitation would be switching to collecting raw timestamps from the sensor itself and not from the OS. This task is technically difficult because raw sensor timestamps are not uniform and counted from the specific time epoch, so it does not specifically reflect the time when the event happened. That feature is set by the manufacturer, so it too may vary from device to device.

# Conclusion

Low battery charge, functional state of the device and different hardware may introduce unwanted data inconsistencies when capturing sensor information. Especially, this data may be vulnerable when accessing several sensors at the same time or at higher than 8Hz frequency and using devices from different manufactures. Although, question remains whether inconsistently recorded data are valid and still usable. So, certain data validation procedures have to be put in place or special acquisition algorithms in app design must be devised before data can be deemed clean and usable for the research.

##### References

1. N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, “A survey of mobile phone sensing,” *IEEE Communications magazine*, vol. 48, no. 9, pp. 140–150, 2010.
2. Stisen A., Blunck H., Bhattacharya S., Prentow T.S., Kjærgaard M.B., Dey A., Jensen M.M. Smart devices are different: Assessing and mitigating gmobile sensing heterogeneities for activity recognition; Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems; Seoul, Republic of Korea. 23 November 2015; pp. 127–140.
3. G. Grouios, E. Ziagkas, A. Loukovitis, K. Chatzinikolaou, E. Koidou. Accelerometers in Our Pocket: Does Smartphone Accelerometer Technology Provide Accurate Data? Sensors (Basel). 2023 Jan; 23(1): 192.
4. J.R. Kwapisz, G.M. Weiss, and S.A. Moore. Activity recognition using cell phone accelerometers. SIGKDD Explorations Newsletter, 12(2):74–82, 2011.
5. T. Brezmes, J.L. Gorricho, and J. Cotrina. Activity recognition from accelerometer data on a mobile phone. Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living, pages 796–799, 2009.
6. W. Wu, S. Dasgupta, E.E. Ramirez, C. Peterson, and G.J. Norman. Classification accuracies of physical activities using smartphone motion sensors. Journal of Medical Internet Research, 14(5), 2012.
7. Douangphachanh V., Oneyama H. Exploring the use of smartphone accelerometer and gyroscope to study on the estimation of road surface roughness conditions; Proceedings of the 11th International Conference on Informatics in Control, Automation and Robotics; Vienna, Austria. 1–3 September 2014; pp. 783–787.
8. Zhai Y., Nasseri N., Pöttgen J., Gezhelbash E., Heesen C., Stellmann J. Smartphone accelerometry: A smart and reliable measurement of real-life physical activity in multiple sclerosis and healthy individuals. Front. Neurol. 2020;11:688.
9. Culhan K., O’Connor M., Lyons D., Lyons G. Accelerometers in rehabilitation medicine for older adults. Age Ageing. 2005;34:556–560.
10. Chinrungrueng J., Sartsatit S., Intarapanich A. My act: An automatic detection of daily physical activity and calorie expenditure using smart phones. J. Assist. Rehabil. Ther. Technol. 2014;2:23187.
11. Yurur, O., Labrador, M. & Moreno, W. Adaptive and energy efficient context representation framework in mobile sensing. IEEE Trans. Mob. Comput. 2014; 13, 1681–1693.
12. Straczkiewicz, M., Huang, E.J. & Onnela, JP. A “one-size-fits-most” walking recognition method for smartphones, smartwatches, and wearable accelerometers. npj Digit. Med. 2023; 6, 29.
13. E. Peguero, M. Labrador and B. Cook, Assessing Jitter in Sensor Time Series from Android Mobile Devices. 2016 IEEE International Conference on Smart Computing (SMARTCOMP), St. Louis, MO, USA, 2016, pp. 1-8.
14. W. Ramadan, E. Ozer. Modal analysis under jittering and kernel clock distribution: single-output identification. 2023. Proceedings of the Institution of Civil Engineers - Smart Infrastructure and Construction.