Motion Sensor Latencies for Software Development

The Motion Sensor

The motion sensor collects three dimensional acceleration and angular velocity data at 64 Hz. This is a fixed sample rate that is not affected by the mode (wireless streaming or memory) or communication between the sensor and the computer.

Latency Effect of Wireless Bluetooth Communication

While the sensor's sample rate is constant, sending commands to the sensor to initiate and stop data collection is subject to a communication delay. This delay is due to the nature of wireless data transmission between the computer and the sensor. This delay is variable depending on the computer processer and Bluetooth signal strength.

A delay of up to 100ms is possible for each command sent between the computer and sensor. This will not affect the sample rate of the sensor but will effect when the sensor initiates and terminates data collection. This will create an offset between sensors, and between the sensors and the computer start time. This offset will remain constant within the data collection trial but will be different for each initiation of data collection.

Methods to Reduce the Latency Effects on Data Collection

The process for initial connection to the device, which can take multiple seconds, should be isolated from the data collection process. Additionally, if a significant delay from the 'start acquisition' function is noted, the following method is recommended.

Create an internal data saving loop

Since the data collection will be initiated at a variable time, one method of correction is creating an independent loop for saving data. Before entering the data collection loop, just as the trial is started, the sensor data buffer can be cleared. While this will reduce the effect of the time to complete the start acquisition function, there still may be some residual latency effect due to the time to receive data into the computer buffer from the sensor. An example of this method, implemented in LabVIEW with the LabVIEW sensor SDK subVIs, is shown in Fig 1.

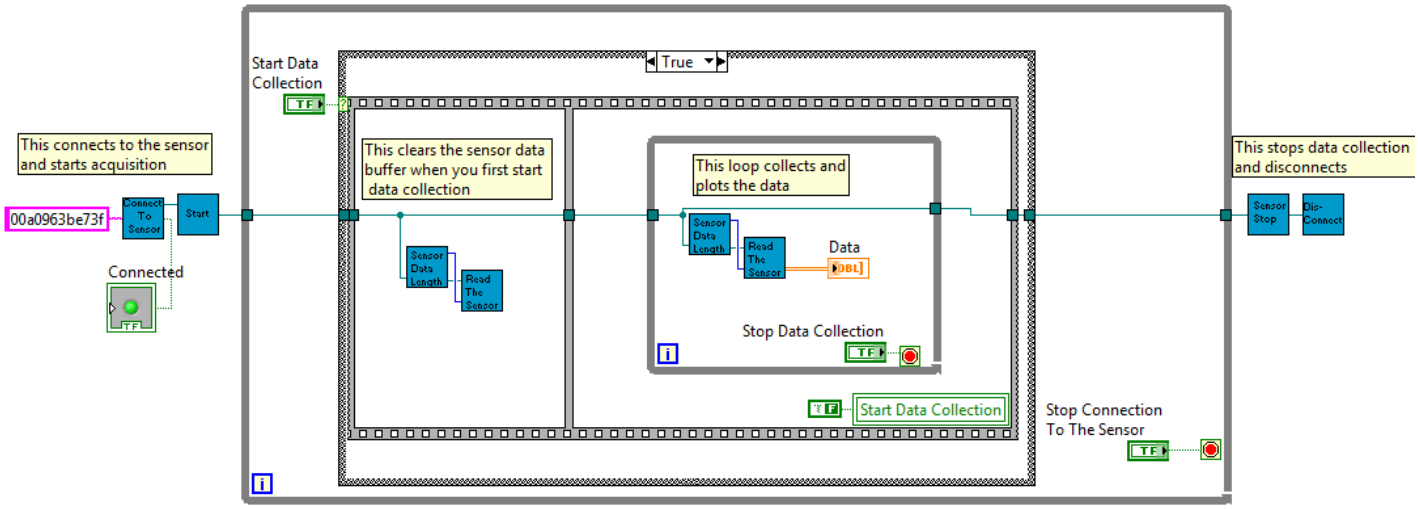

Figure 1. Example data collection loop to reduce latency effects implemented in LabVIEW. The program connects to the sensors and immediately starts sensor acquisition. The system then waits to pull any data from the buffer. Once ready to start data collection, the 'Start Data Collection' button is pressed, the buffer is cleared, and data collection starts. The 'Stop Data Collection' button is pressed to stop the data collection trial.

Synchronization between sensors

For time sensitive applications, we recommend using a synchronization signal to align your sensor data. This will allow for the correction of shifted sensor time data. For example, all of the sensors may be placed on a board (Fig 2.) and perturbed together in order to align the sensor times. While it is recommended that this synchronization process be completed at the beginning of the study, there will be no change in the time difference between sensors throughout the single data collection period. However, each connection will result in different latency times between sensors.

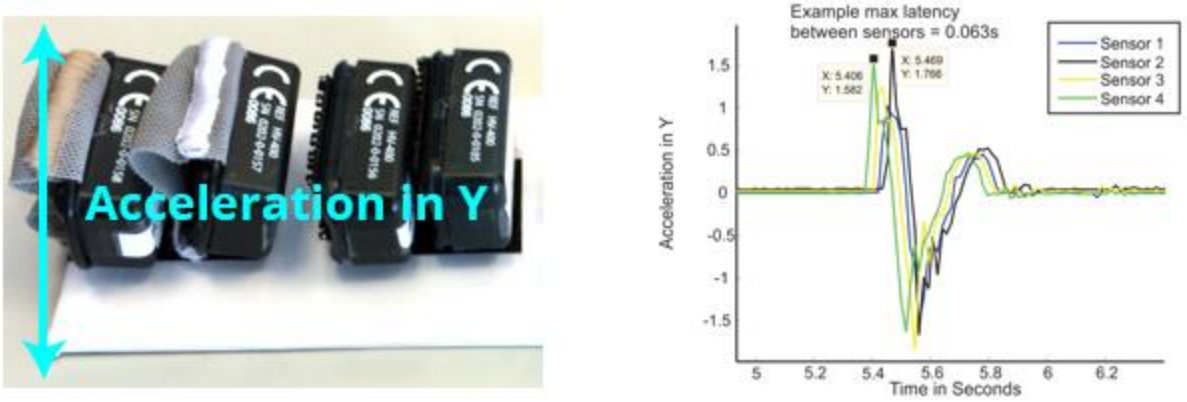

Figure 2. Example of the shake/perturbation test with the potential setup shown on the left and collected data on the right. The time of the peaks can be used to align the times of the sensor data.

Synchronization to the computer

Synchronization between the computer and a finger worn sensor can be achieved by comparing sensor and key press or mouse click data. The settled sensor data time relative to the computer keyboard or mouse state change will describe the communication delay as shown in the Fig 3. Please note that keyboards with high travel distance will potentially change state before the finger stops moving and may not be useful for this evaluation.

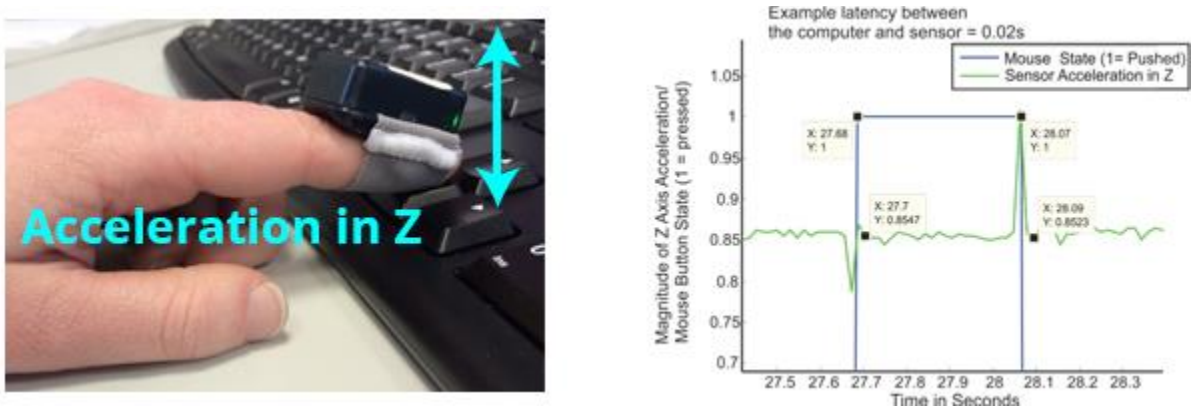

Figure 3. Example of the computer synchronization test with the potential setup for a keyboard shown on the left and collected mouse and sensor data on the right. The settled sensor acceleration time can be compared to the time of the state change to evaluate the delay between the sensor and computer.