|  |  |  |
| --- | --- | --- |
| **בי"ס להנדסת חשמל** | | |
| פרויקט מס'16-1-1-1127  דו"ח פתיחה | | |
| שם הפרויקט: \_MEMS Battery operated projects\_ | | |
| מבצעים: | | |
|  | שם: \_מרים מישל \_\_\_\_\_ | ת.ז. \_\_324540061 \_\_ |
|  | שם: \_\_דניאל אקוע\_\_\_\_ | ת.ז. \_\_203974571\_\_\_\_ |
| מנחה: \_\_ארקדי רפלוביץ \_\_\_\_\_\_\_\_ | | |
|  |  |  |
| מקום ביצוע הפרויקט: \_\_\_\_\_האוניברסיטה\_\_\_\_\_\_ | | |

Table of Contents

1. Project Purpose
   1. Background
   2. Project description

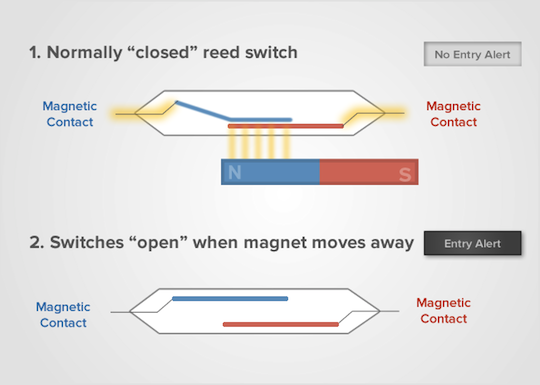
**Project Purpose**

**Background**

Entrances and exits are your home's weakest points—and to a burglar, an unsecured door or window is like a big "WELCOME" sign. That's why door sensors—also known as "entry sensors" or "window sensors”— are some of the most popular home security devices. Before technology evolved, home security systems were entirely made up of door sensors—when someone opened a door or window, it triggered a big vibrating bell in a central part of the home. Nowadays, technology has improved the market offers different choices. Here is a short description of the most popular entry sensor today.

Magnetic Sensors

Most of the door and window sensors use a "reed switch" to determine when a monitored entrance has been breached. A reed switch consists of a set of electrical connectors placed slightly apart. When a magnetic field is placed parallel to the electrical connectors, it pulls them together, closing the circuit.

****

Door sensors have one reed switch and one magnet, creating a closed circuit. If someone opens the door or window, the magnet is pulled away from the switch, which breaks the circuit and triggers an alarm.

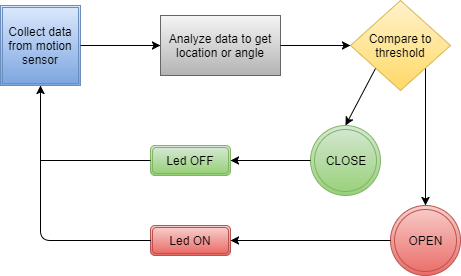
A system made of reed switch and a magnet has some disadvantages. One of them is that for some doors and windows the system may not fit, such as doors and windows with very deep frames. Another disadvantage is that metal objects can affect the sensors performance and may cause false-positive events or even miss true events. The biggest disadvantage of the magnet operated sensor is that a burglar can use a special magnet and trick the system, an unlikely event, but still possible.

**Project Description**

A window opening detector is a sensor used in electronic burglar alarms that detects if a window or door is opened. There are different products in the market which detect window opening. Most of them work with magnets. In our project, we’ll use an accelerometer and a gyroscope to detect movements of the window or door. Our product uses a micro-controller and an accelerometer from ST product catalog.

A window opening is detected when the position or angle calculated from the accelerometer’s data is above the threshold value. An accelerometer sensor can identify the acceleration and return the value to the micro-controller for analyzing process.

The product will work as following:



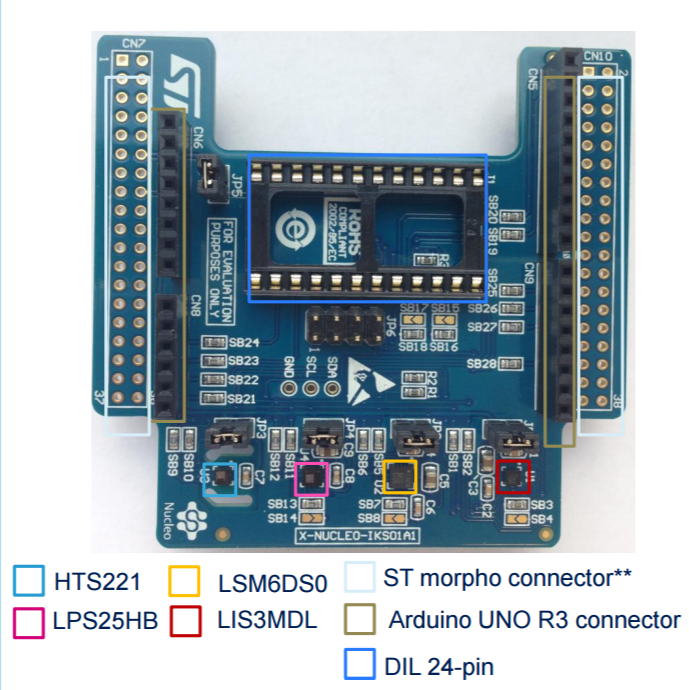
For making this algorithm work, we need to identify the threshold acceleration when a window moves. Therefore, when we get a value equal or higher from the threshold value, a moving event can be detected.

**Hardware Description**

**X-NUCLEO-IKS01A1 - motion MEMS and environmental sensor**

The X-NUCLEO-IKS01A1 is a motion MEMS and environmental sensor evaluation board system. It is compatible with the Arduino UNO R3 connector layout, and is designed around STMicroelectronics’ LSM6DS0 3-axis accelerometer + 3-axis gyroscope, the LIS3MDL 3-axis magnetometer, the HTS221 humidity and temperature sensor and the LPS25HB\* pressure sensor. The X-NUCLEO-IKS01A1 interfaces with the STM32 microcontroller via the I²C pin, and it is possible to change the default I²C port.

Features:

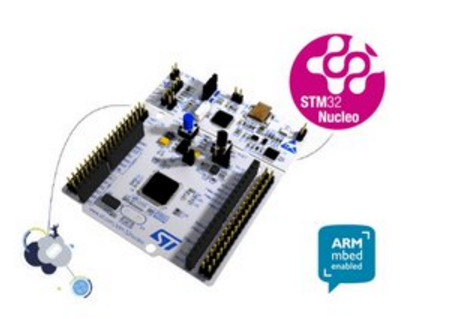
* LSM6DS0: MEMS 3D accelerometer (±2/±4/±8 g) + 3D gyroscope (±245/±500/±2000 dps)
* LIS3MDL: MEMS 3D magnetometer (±4/ ±8/ ±12/ 16 gauss)
* LPS25HB\*: MEMS pressure sensor, 260-1260 hPa absolute digital output barometer
* HTS221: capacitive digital relative humidity and temperature
* DIL 24-pin socket available for additional MEMS adapters and other sensors (UV index)
* Free comprehensive development firmware library and example for all sensors compatible with STM32Cube firmware
* Compatible with STM32 Nucleo boards
* Equipped with Arduino UNO R3 connector
* RoHS compliant

**NUCLEO-F411RE - Microcontroller**

A platform to ease prototyping using a STM32F411RET6 microcontroller. The STM32 Nucleo board provides an affordable and flexible way for users to try out new ideas and build prototypes with any STM32 microcontroller line, choosing from the various combinations of performance, power consumption and features.

The Arduino™ connectivity support and ST Morpho headers make it easy to expand the functionality of the STM32 Nucleo open development platform with a wide choice of specialized shields.

The STM32 Nucleo board does not require any separate probe as it integrates the ST-LINK/V2-1 debugger/programmer.

Microcontroller features

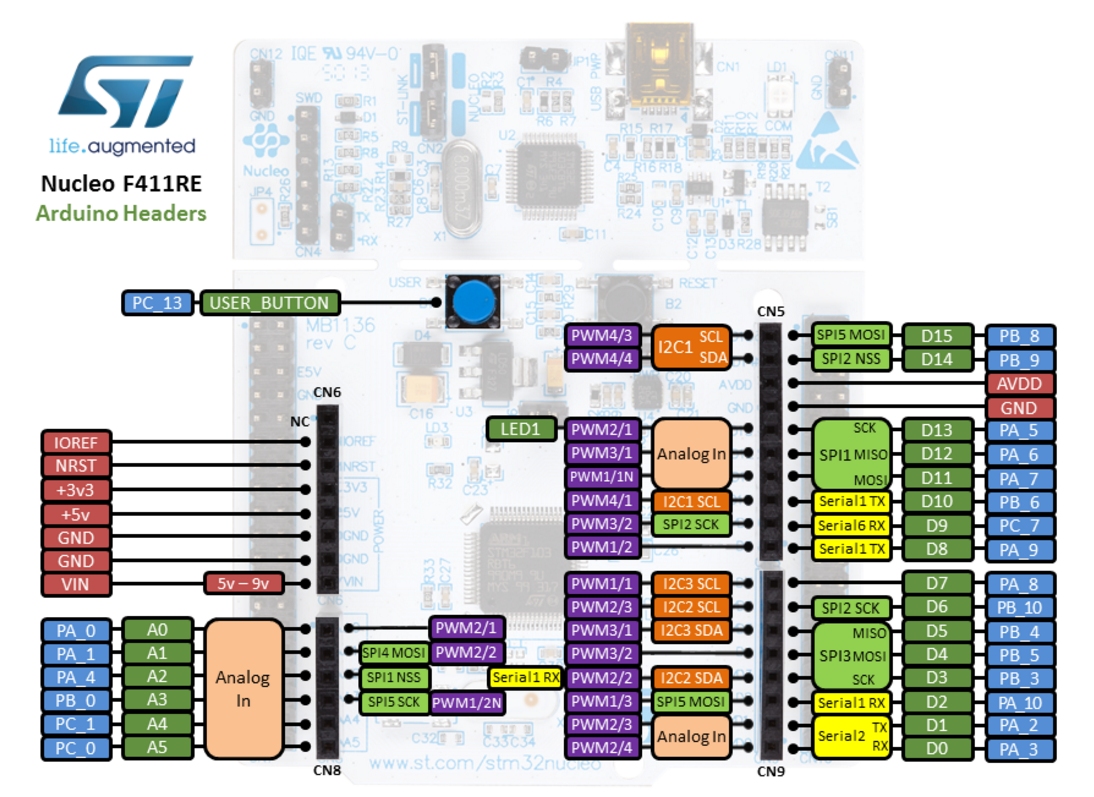
* STM32F411RET6 in LQFP64 package
* ARM®32-bit Cortex®-M4 CPU with FPU
* 100 MHz max CPU frequency
* VDD from 1.7 V to 3.6 V
* 512 KB Flash
* 128 KB SRAM
* GPIO (50) with external interrupt capability
* 12-bit ADC with 16 channels
* RTC
* Timers (8)
* I2C (3)
* USART (3)
* SPI (5)
* USB OTG Full Speed
* SDIO

Nucleo features

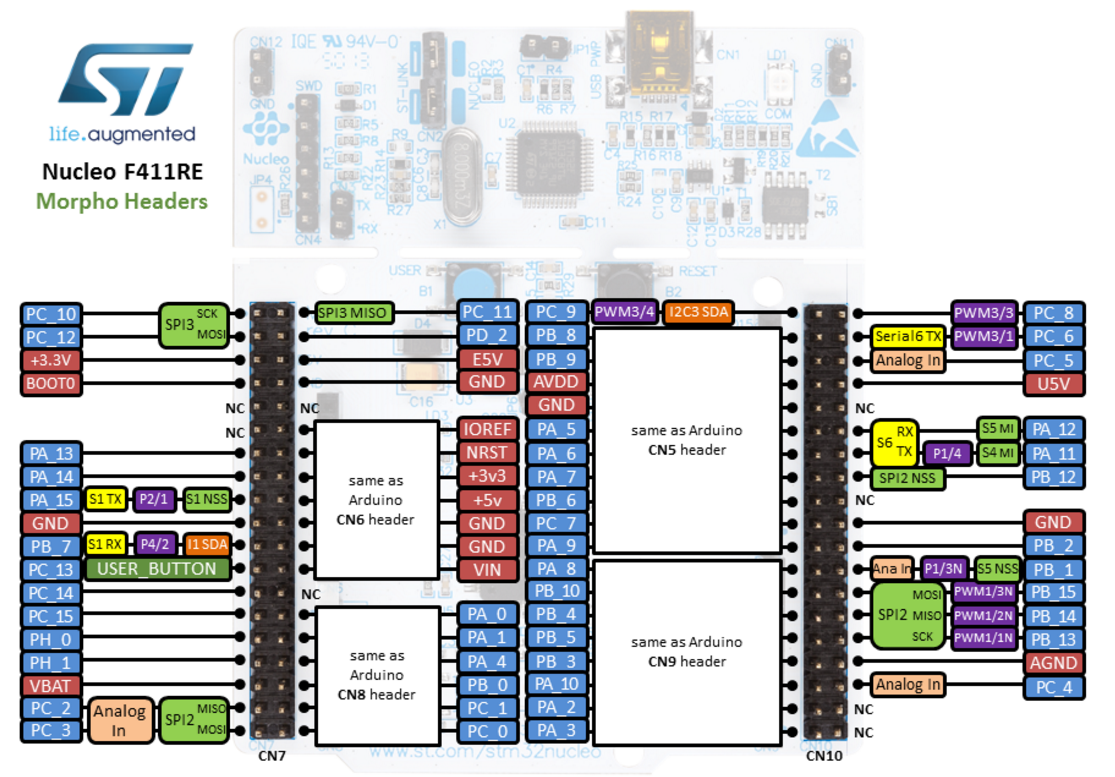
* Two types of extension resources
* Arduino Uno Revision 3 connectivity
* STMicroelectronics Morpho extension pin headers for full access to all STM32 I/Os
* On-board ST-LINK/V2-1 debugger/programmer with SWD connector
* Selection-mode switch to use the kit as a standalone ST-LINK/V2-1
* Flexible board power supply
* USB VBUS or external source (3.3 V, 5 V, 7 - 12 V)
* Power management access point
* User LED (LD2)
* Two push buttons: USER and RESET
* USB re-enumeration capability: three different interfaces supported on USB
* Virtual Com port
* Mass storage (USB Disk drive) for drag'n'drop programming
* Debug port

Nucleo pinout

Arduino-compatible headers

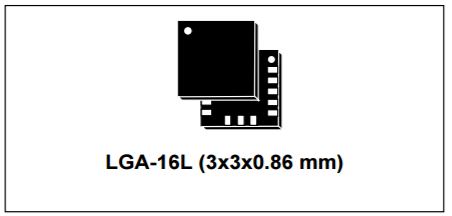


Morpho headers

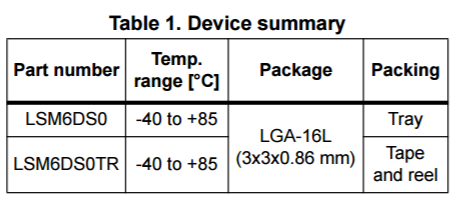
These headers give access to all STM32 pins.

**LSM6DS0 Sensor - 3D accelerometer and 3D gyroscope**

The LSM6DS0 is a system-in-package featuring a 3D digital accelerometer and a 3D digital gyroscope. ST’s family of MEMS sensor modules leverages the robust and mature manufacturing processes already used for the production of micromachined accelerometers and gyroscopes. The various sensing elements are manufactured using specialized micromachining processes, while the IC interfaces are developed using CMOS technology that allows the design of a dedicated circuit which is trimmed to better match the sensing element characteristics. The LSM6DS0 has a full-scale acceleration range of ±2/±4/±8/±16 g and an angular rate range of ±245/±500/±2000 dps. The LSM6DS0 has two operating modes in that the accelerometer and gyroscope sensors can be either activated at the same ODR or the accelerometer can be enabled while the gyroscope is in power-down. The LSM6DS0 is available in a plastic land grid array (LGA) package.

Features

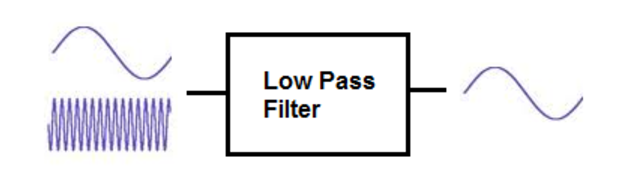
* Analog supply voltage: 1.71 V to 3.6 V
* Independent IOs supply (1.71 V)
* “Always on” eco power mode down to 1.8 mA
* 3 independent acceleration channels and 3 angular rate channels
* ±2/±4/±8/±16 g full scale
* ±245/±500/±2000 dps full scale
* SPI/I2C serial interface
* Embedded temperature sensor
* Embedded FIFO
* ECOPACK®, RoHS and “Green” compliant

Applications

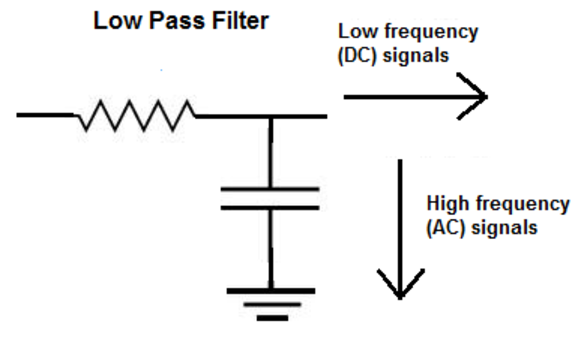
* GPS navigation systems
* Impact recognition and logging
* Gaming and virtual reality input devices
* Motion-activated functions
* Intelligent power saving for handheld devices
* Vibration monitoring and compensation
* Free-fall detection
* 6D orientation detection

**Low Pass Filter**

Theory – how low pass filter works

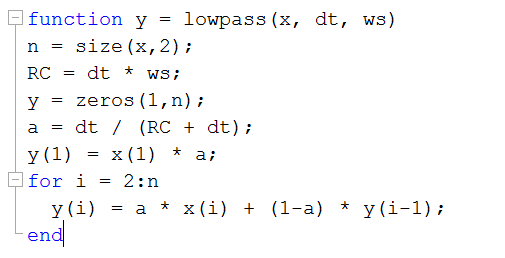


A low pass filter is a filter which passes low-frequency signals and blocks, or impedes, high-frequency signals.

Low-frequency signals go through much easier and with less resistance while high-frequency signals have a much harder getting through, which is why it's a low pass filter. Low pass filters can be constructed using resistors with either capacitors or inductors. A low pass filter composed of a resistor and a capacitor is called a low pass RC filter. A low pass filter composed of a resistor and an inductor is called a low pass RL filter. Both circuits have the effect of passing through low frequency signals while impeding high-frequency ones.

Formula

For vector x of size n, time between samples dt and frequency ws, LPF is computed as the following:

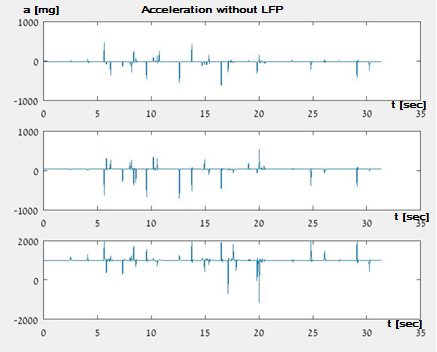


We get the filtered vector y of size n.

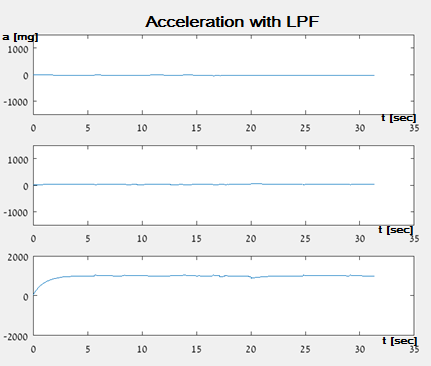
Low Pass Filter experiment:  
parameters: dt = 1/frequency (1/200) , ws = frequency (200)

In order to record data for this experiment we tapped the sensor to create high frequency peaks. We analyzed the data from the accelerometer for each of the 3 axes. The expected results are that the data without the lowpass filter will contain AC and DC parts and will be unstable, while the filtered data will contain only the DC part and will be stable.

Unfiltered Data:



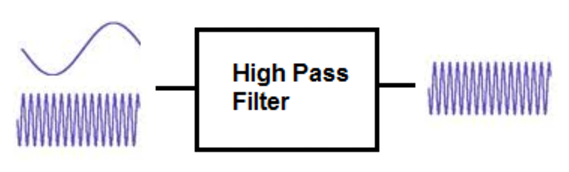
Filtered data:



As we can see from the experiment, the filtered data contains only the signal’s DC while the raw data contains both the AC and DC parts.

**High Pass Filter**

Theory – how high pass filter works



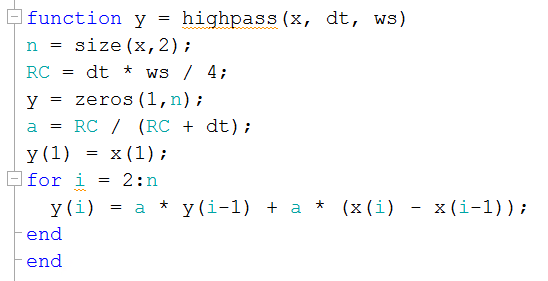
A high pass filter is a filter which passes high-frequency signals and blocks, or impedes, low-frequency signals.

In other words, high-frequency signals go through much easier and low-frequency signals have a much harder getting through, which is why it's a high pass filter.

High pass filters can be constructed using resistors with either capacitors or inductors. A high pass filter composed of a resistor and a capacitor is called a high pass RC filter. And a high pass filter with a resistor and an inductor is called a high pass RL filter.

Formula:

For vector x of size n, time between samples dt and frequency ws, HPF is computed as the following:

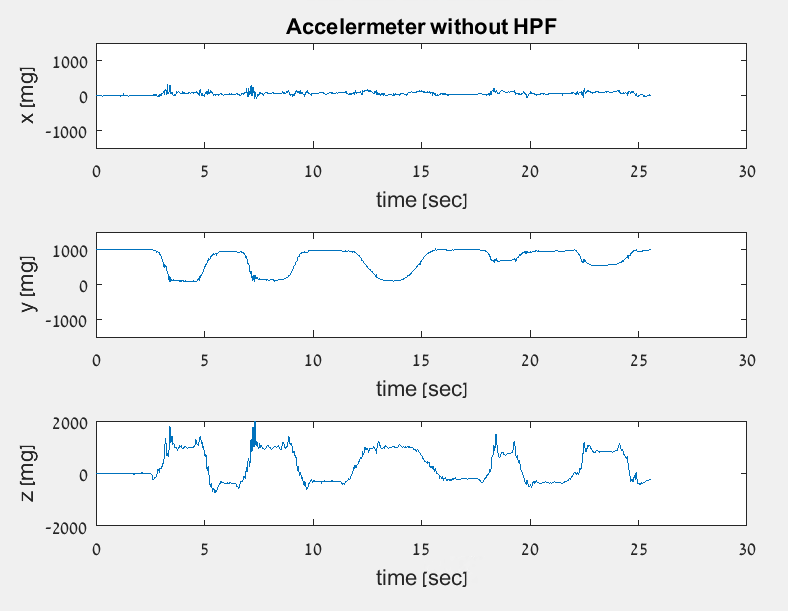


We get the filtered vector y of size n.

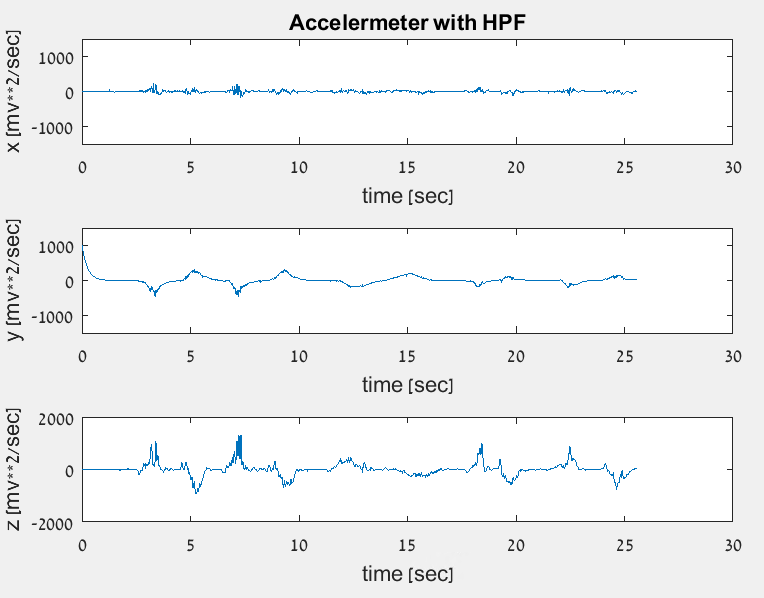
High Pass Experiment:  
parameters: dt = 1/frequency (1/200) , ws = frequency (200)

In order to record data for this experiment we tapped the sensor to create high frequency peaks and moved the sensor gently so the DC part won’t be 0. We analyzed the data from the accelerometer for each of the 3 axes. The expected results are that the data without the high-pass filter will contain AC and DC parts, while the filtered data will contain only the AC part and the DC will be around 0.

Unfiltered Data:



Filtered Data:

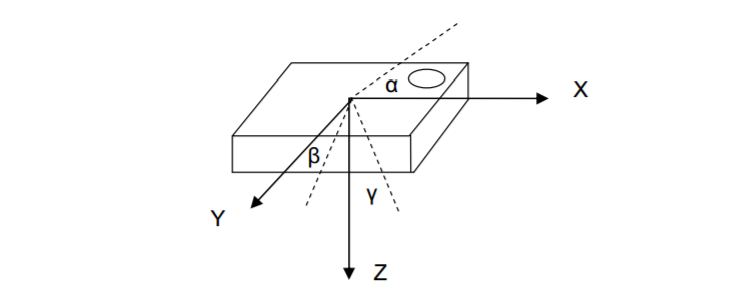


As we can see from the experiment, the filtered data contains only the signal’s AC while the raw data contains both the AC and DC parts.

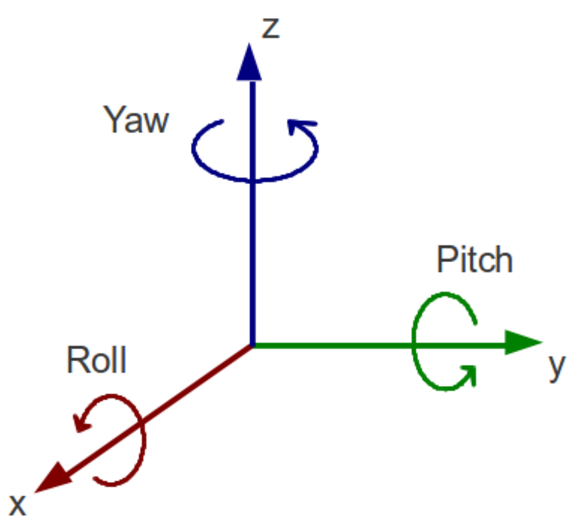
**Calculating Pitch & Roll angles**

Tri-axis tilt sensing:

With a 3-axis accelerometer, the user can use the Z-axis to combine with the X and Y axes  
for tilt sensing in order to improve tilt sensitivity and accuracy as shown in the next figure.   
We use trigonometric equations to calculate pitch and roll tilt angle.  
More information about the equation development can be found in “Tilt calculation” in bibliography.



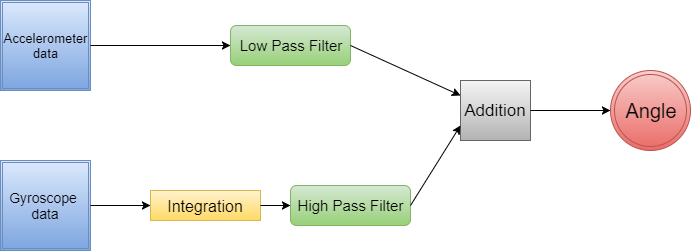
When pitch, roll and yaw angles are as following:



**Complementary Filter**

The gyroscope has a drift, as a result, after few samples the values returned are completely wrong. The accelerometer, on the other hand, returns a true value when the acceleration is progressive but it suffers much from vibrations, returning wrong angle values.

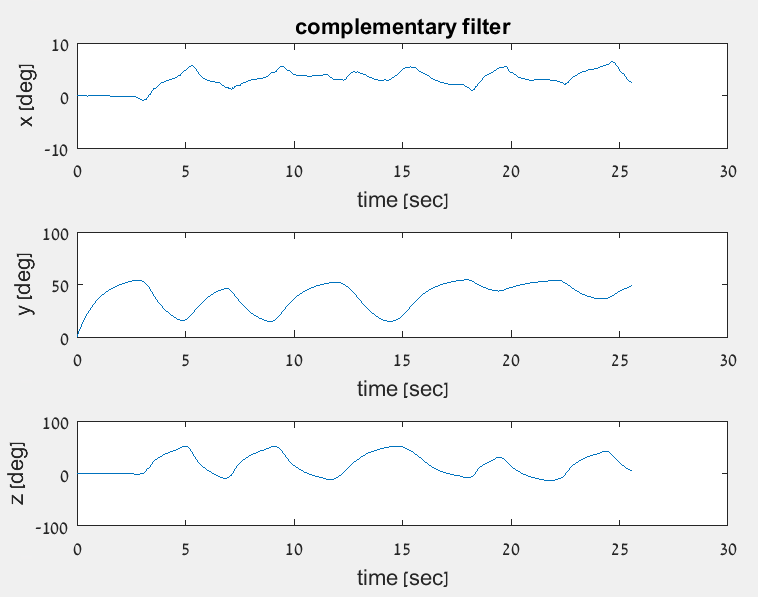
In order to solve this problem and reduce the error, we will use a filter called **complementary filter**. The complementary filter is a combination of LPF, HPF and an integrator. The LPF filters high frequency signals (like the accelerometer’s vibrations) and the HPF filters the low frequency signals (like the gyroscope’s drift). These 2 filters, combined with the integrator (as shown below), produce a signal that is not suffering from the errors of the accelerometer and the gyroscope.



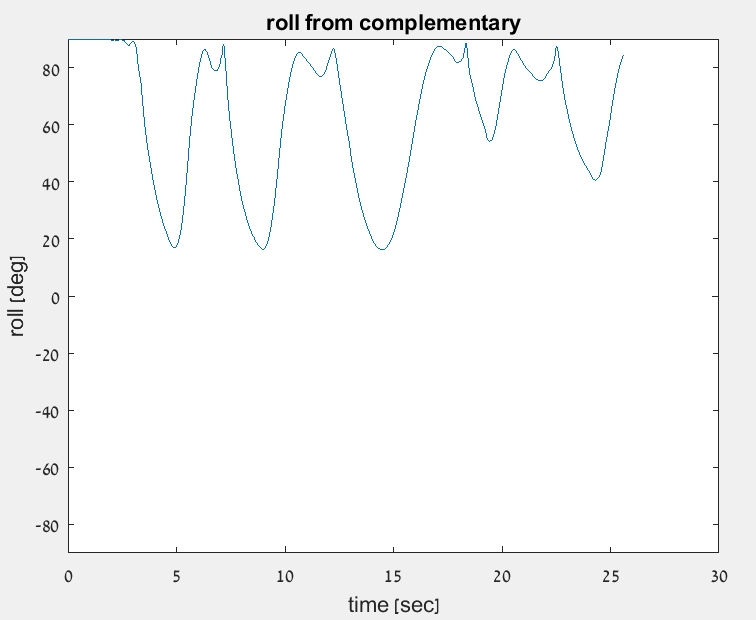
Complementary Filter Experiment:  
filter parameters: dt = 1/frequency (1/200) , ws = frequency (200)

In order to record data for this experiment we changed the angle of the sensors. We analyzed the data from the accelerometer and the gyroscope for each of the 3 axes.

Complementary filter:



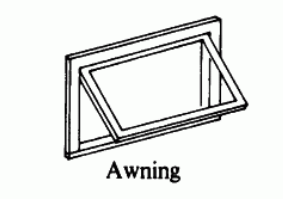
Roll calculation from complementary filter:



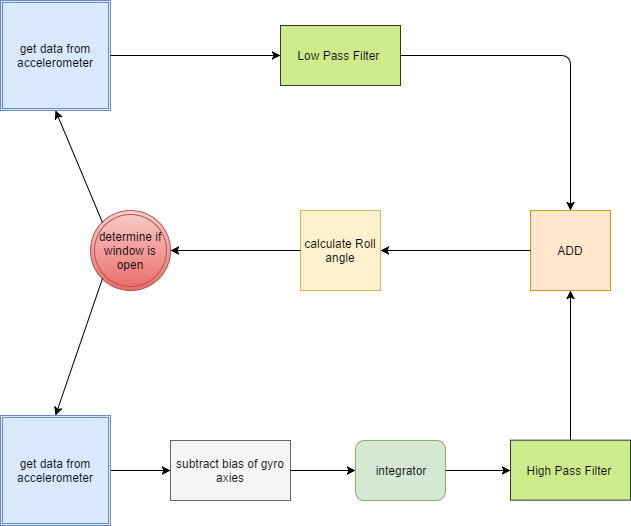
**Opening Hopper and Awning windows**

Theory

This window’s opening can be detected by calculating the change in the angle.  
The system is attached to the window \ door. The accelerometer records about 200 samples per second. The data is passed through LPF. Then, the angle is calculated from the new data.



General Diagram:



Hopper and Awning windows opening detection uses only the Roll angle. We assume the system starts when the window is closed.

Threshold:

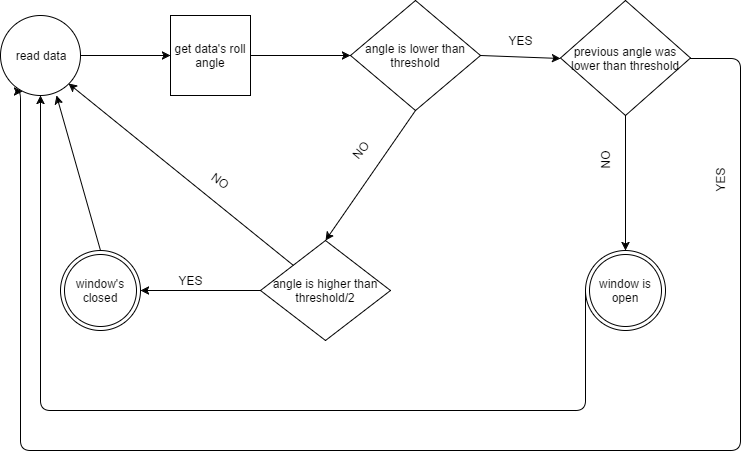
Opening is detected as following:

Closing is detected as following:

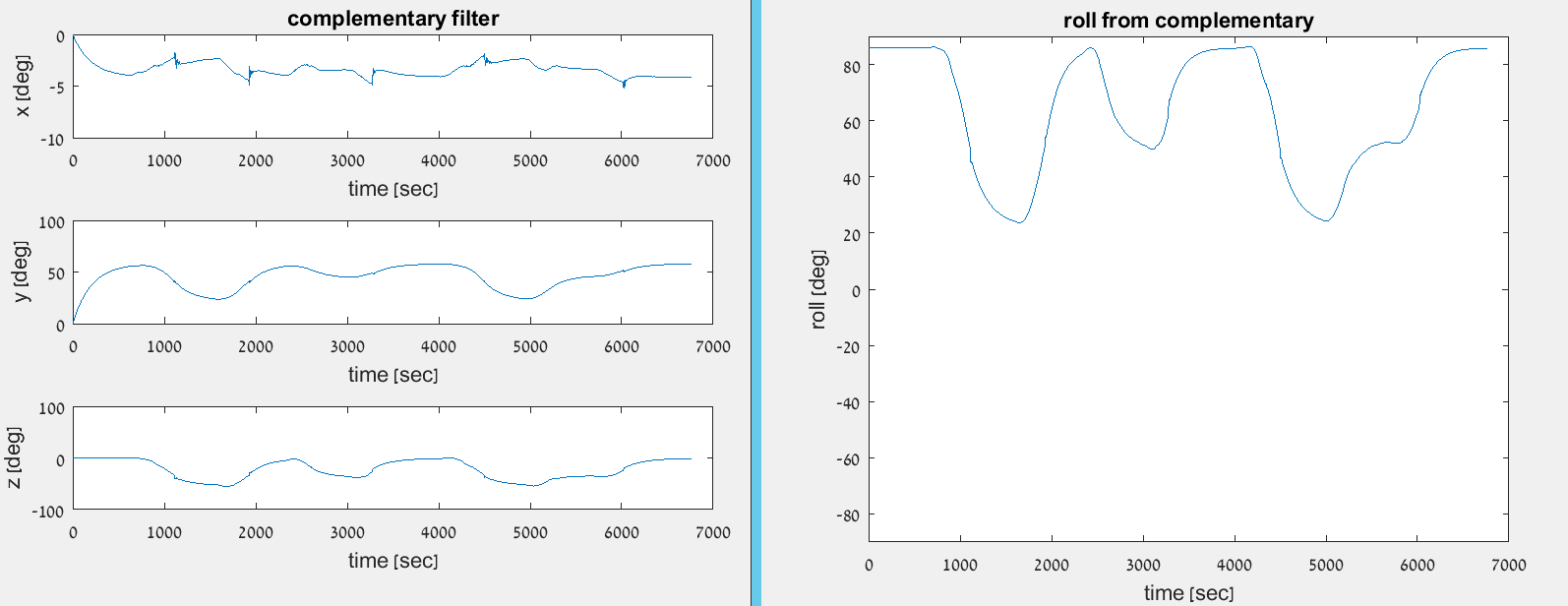
when

The threshold was found after an initial experiment. We found that the window is completely closed at angle of 90.5 degrees (aprox.) in our setup. When angle reaches 20 degrees below the starting angle it is absolutely opened, considering building angle and window straightness.

The following diagram expands and explains the “determine if window’s open” circle:

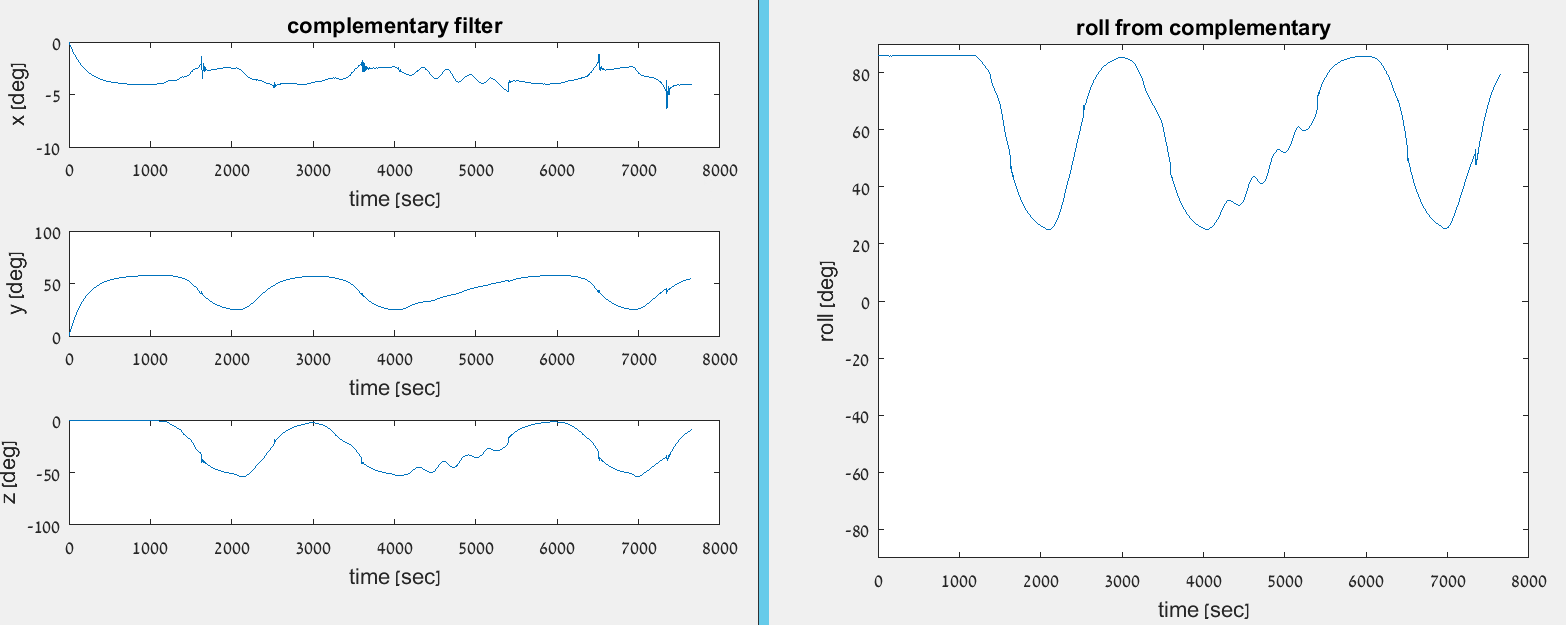


The following graphs are produced from the hopper experiment with filter parameters:   
dt =1/frequency (1/200) , ws = frequency (200)



The left 3 graphs represent the angles from complementary filter.   
The right graph represents the Roll angle. We can see that the hopper window was opened and closed completely 3 times and one mock closing. The program detected 3 openings and ignored the mock as expected.

The following graphs are produced from the awning experiment with filter parameters:  
 dt = 1/frequency (1/200) , ws = frequency (200):



The left 3 graphs represent the angles from complementary filter.   
The right graph represents the Roll angle. We can see that the awning window was opened and closed completely 3 times and one mock closing (closing a little bit each time). The program detected 3 openings and ignored the mock as expected.

**Opening Slide Window**

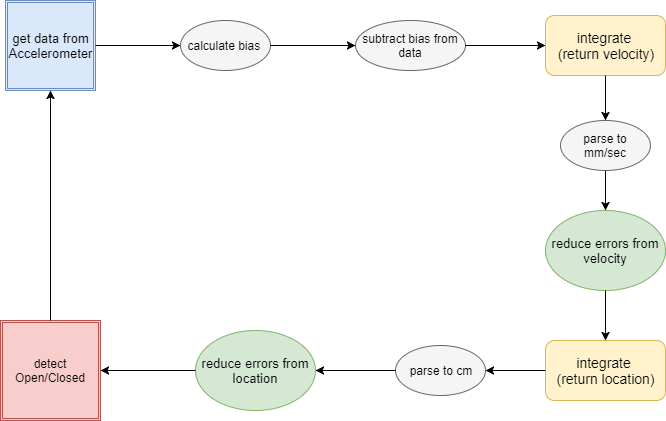
Theory

This window’s opening can be detected by calculating the change in location.

The system is attached to the window \ door. The accelerometer records about 200 samples per second. The data is passed through double integration. Then, the location of the window is calculated from the new data.

The first integration on the acceleration data creates the velocity data. The second integration in integration on velocity, which creates the location data. the starting location is considered as 0. We assume the system starts when the window is closed.

General diagram:



Integration tends to gain errors. Without normalizing the errors on both integrations, the location will be incorrect and move farther each time.

Therefore, after first integration, each value of velocity with is lower than 15 cm/s and all surrounding values are also lower is written as 0. The same is done after the second integration on the values of locations. Each location value lower than 5 and surrounded by lowers values, is written as 0. This filtering corrects the integrations errors. Each integration adds values, but if the windows is not really moving, the only thing that is integrated is noise.

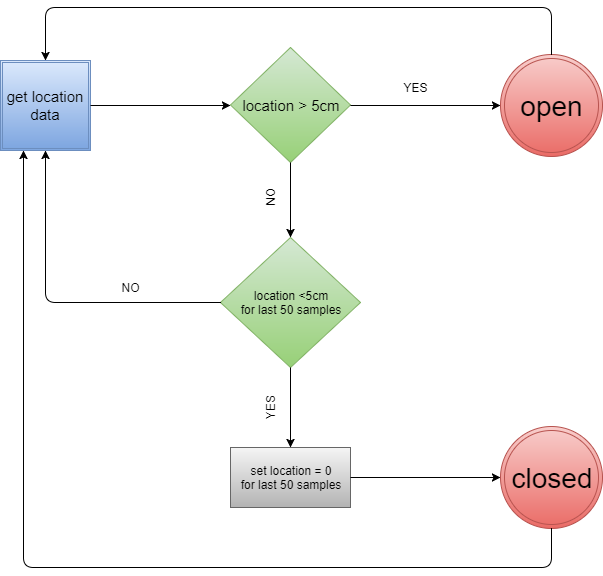
Detection of opening:

Detection of closing:

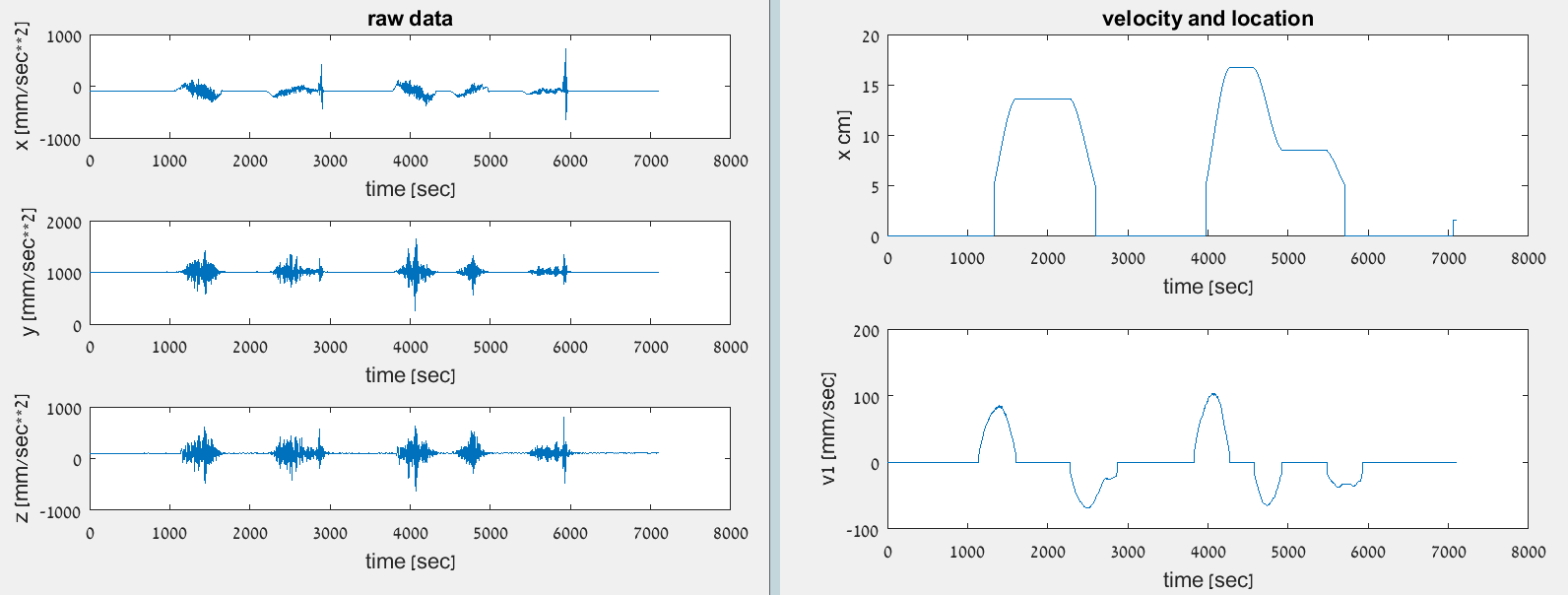
Check 50 samples, if all samples under 5cm – window closed.

Those thresholds were found with few experiments. After 5cm opening, we can be sure that the window is open and it’s not false positive error.

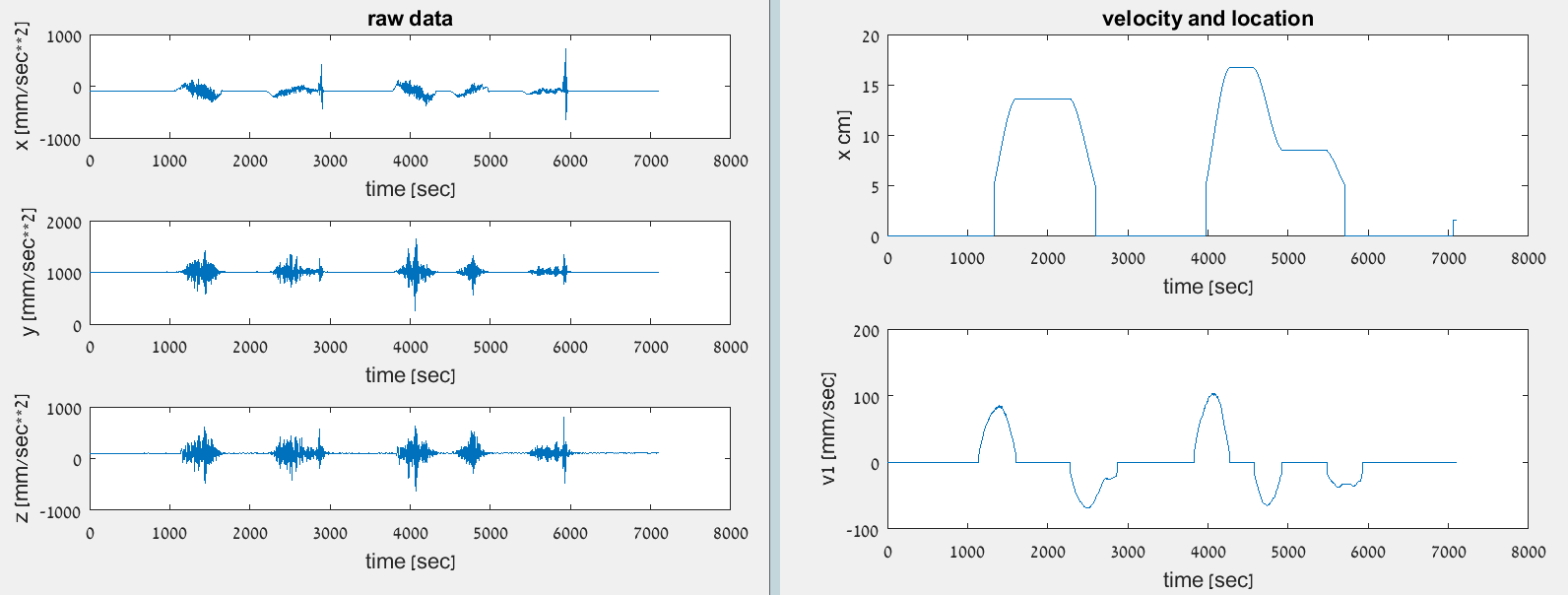
The following diagram expands and explains the “detect window Open/Closed” square and “reduce errors from location” circle:



The following graphs are produced from the sliding experiment with filter parameters:   
dt = 1/frequency (1/200) , ws = frequency (200):



Raw data - each graph represents the raw data from the accelerometer in each axis.



Velocity and location graphs – first graph is the location graph and the second is the velocity graph.  
The velocity graph shows that the velocity grows as the window opens, and backs to zero when opening ends. The negative velocity is the closing of the window, which comes back to zero as window stops closing.

The location graph shows the windows position. We can spot the first opening and closing as position is back to zero. After the second opening the window starts to close but does not close to the end, as we can see. Few seconds later it continues to close and location returns to zero.

The algorithm detects each opening and closing correctly.

**Conclusion**

Results overview

Our results from the matlab simulations match our expectations. All the window types we examined, detected opening and closing of a window with a minimal error. Using different filters such as complimentary, high pass and low pass reduce the error. The set ups of those systems have few requirements, for example – the window has to be closed and the microcontroller’s USB needs to face down.

Transferring the system to a real time program was challenging. In real time we cannot work with static vectors. Therefore, we used dynamic cyclic arrays to hold past data. the errors on the micro-controller are higher as a result of frequency differences. The Awning\Hopper window works better and with lower error than the Sliding window.

Improvements

Although the system has a small error, there are few suggestions to improve the detecting process.

* The threshold values we used can be more accurate with more experiments and analyze. It can reduce the error even more.
* Adding rotation matrix calculation can reduce the error of sliding window drastically and make it much more robust.
* Using smaller hardware will reduce the visibility of the device while installed on a window.
* Using external batteries will remove the power supply wire.
* Add low power mode when the device is idle.

Future development options

There are some ideas for additional development.

* Implement the device on additional window types.
* More efficient algorithm to reduce calculation time and increase frequency.
* Implement a single algorithm that can identify the type of window the device is attached to and act accordingly.
* Implement a warning system to alert the user that the window is open via application installed on the user’s phone.

**Bibliography**

* Magnetic sensors explained:

http://www.protectamerica.com/home-security-blog/tech-tips/door-window-sensors-work\_8461

* ST hardware - Microcontroller:

http://www.st.com/en/evaluation-tools/nucleo-f411re.html

* ST hardware - Environmental sensor:

http://www.st.com/en/ecosystems/x-nucleo-iks01a1.html

* ST hardware - Accelerometer and Gyroscope:

http://www.st.com/en/mems-and-sensors/lsm6ds0.html

* LPF explained:

http://www.learningaboutelectronics.com/Articles/Low-pass-filter.php

* HPF explained:

http://www.learningaboutelectronics.com/Articles/High-pass-filter.php

* Complementary filter explained:

http://d1.amobbs.com/bbs\_upload782111/files\_44/ourdev\_665531S2JZG6.pdf

* Tilt calculations:

http://www.st.com/content/ccc/resource/technical/document/application\_note/d2/d6/22/4f/b9/8f/45/59/DM00119046.pdf/files/DM00119046.pdf/jcr:content/translations/en.DM00119046.pdf