

**Detection**

**Smart Firearm**

**Contents**

[Executive Summary 2](#_Toc84346289)

[Algorithm 3](#_Toc84346290)

[Revision History 7](#_Toc84346291)

# Executive Summary

This document is illustration of the developed shot detection algorithm for smart fire arm project.

# Algorithm

Step 1: Create Data set that contains the norm of linear acceleration vector of shot. (n samples with length m).

Data Description:

- The sensor signal chosen to detect the trigger of the gun is the norm of linear acceleration (in global frame) measurements.

Note: that the norm of the vector is constant regardless of the chosen projection frame.

- Linear acceleration is defined as the measurement of the accelerometer with the gravity vector subtracted.

- The length of each instance in the data set is specified by examining collected data.

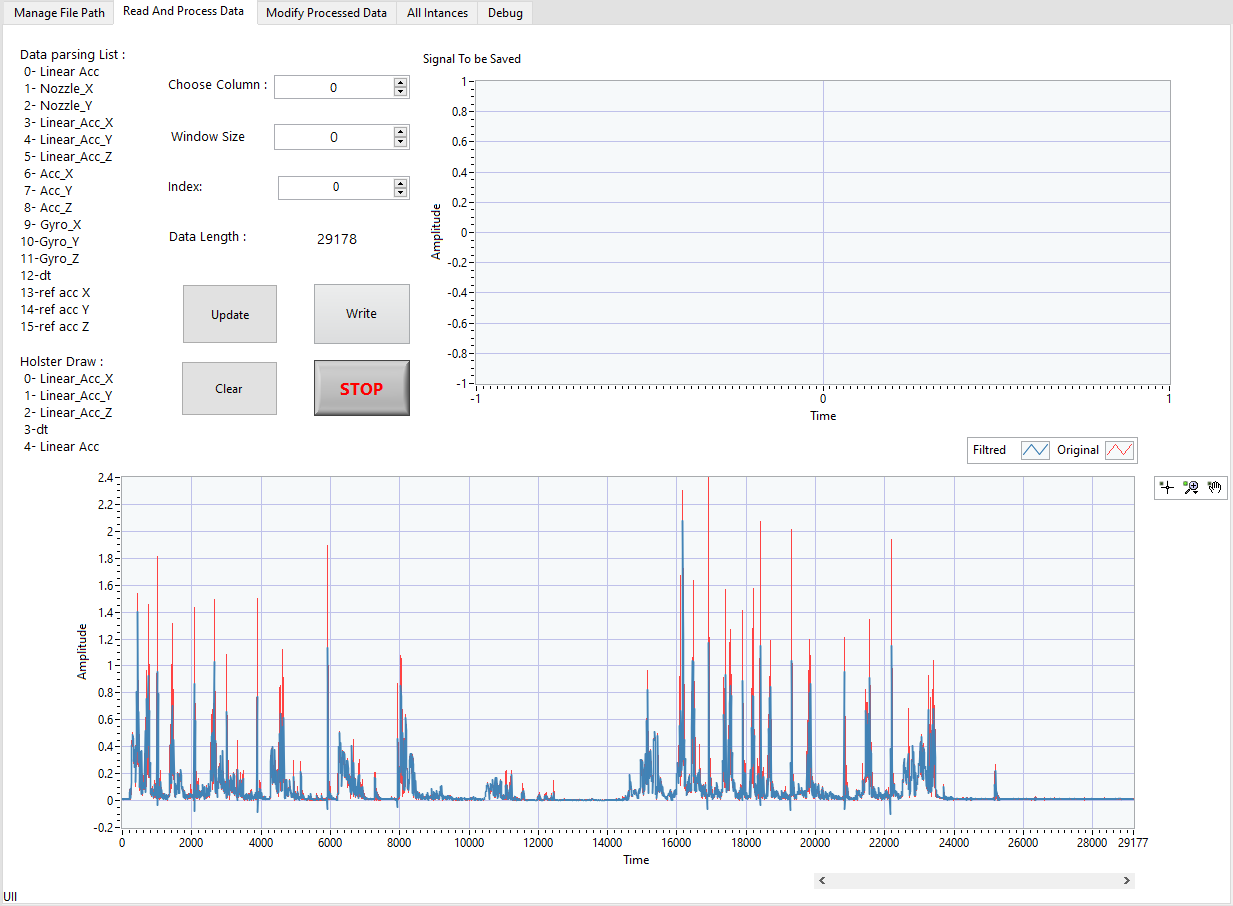
Note: Instance length depends on sensor rate used in acquisition; in order to facilitate the examination, parsing and set creation a VI with the name (manipulate\_data.vi) was developed as shown in figure 1.

Figure 1: Manipulate data VI

- Used filter:

Savitzky-Golay filter which smooths a noisy signal by the piece-by-piece fitting of a polynomial function to the signal.

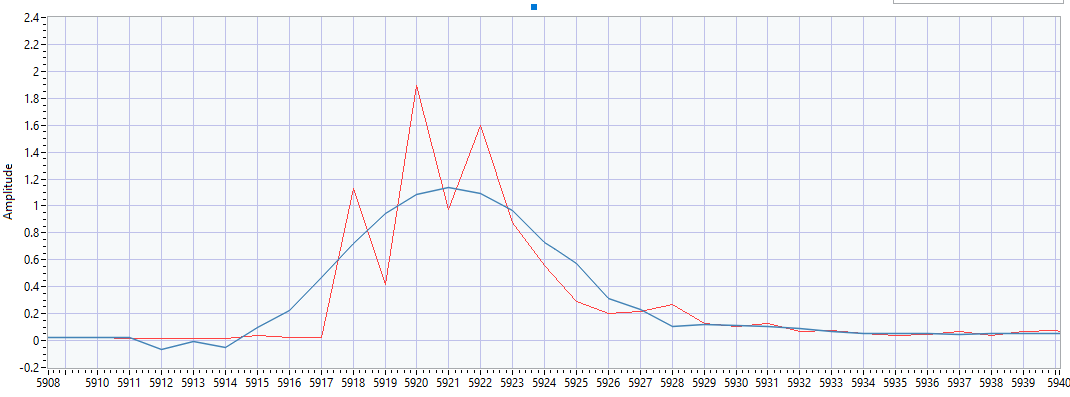
Chosen window size equal to 13 (side:6) and polynomial of the third order.

Figure 2: Filtered and Unfiltered Signals. (Acquisition freq. 400 HZ)

Step 2:

Define examination window, the signal within this window will be compared to the saved data set to classify the signal. The minimum of the window size must be twice the size of the shot.

Example:

For sampling rate equal to 400 Hz and the shot signal length equal to 20 sample ().

The signal within the examination window must be update with every specified t second such that ; the collected sample point within the T period (33 sample) is with length T\*freq is filtered and fed the beginning on the window the previous signal is shifted as illustrated in figure 3.

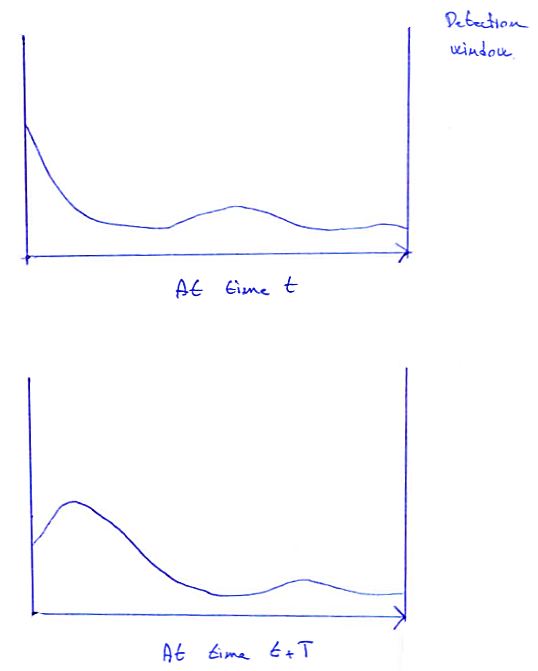
****

Figure 3: Detection window Updating

Step 3: Calculate cross correlation signal within detection window to data set S.

Initialize max\_correlation\_array= zeros (1, n);

Count=0;

For =1:1: size(S,1) %size(S,1) =n

Normalize (Signal within window)

Normalize ( instance)

Compute correlation between the instance and signal within window.

Cross\_corr\_array = . General Continuous Form

max\_correlation\_array (1, i) = max (Cross\_corr\_array);

cond == max (Cross\_corr\_array) > Max\_Resemblance (pre-defined);

if cond==True

Count++;

End

end

**Note:** countis equal to the number of elements within the dataset that has a maximum resemblance higher that prespecified Max\_Resemblance.

cond 1 == avg(max\_correlation\_array) > Det-Min Resemblance Count (pre-defined).

cond 2 == count > Det-Resemblance (pre-defined).

cond 3 == max(max\_correlation\_array) > Max Resemblance (pre-defined).

cond 4 == Det-Acc Lower Thresh (pre-defined) < max (Signal within window).

cond 5 == max (Signal within window) < Det-Acc Upper Thresh (pre-defined).

if all previously specified conditions are satisfied the signal within the detection is considered to be a candidate shot signal.

if in next iteration the signal within the window satisfies the conditions the signal is classified as shot and the algorithm passes the signal analyzes.

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Doc Revision** | **Author** | **Description** | **Date** |
| 1.0 | Mohamad AKOUM | Detection | 05/10/2021 |