ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

TIK Institut für Technische Informatik und Kommunikationsnetze

I-Lin Fang

Digital Signal Processing for Micro-Seismic Event Detection on Embedded Platforms

Master's Thesis

Computer Engineering and Networks Laboratory
Swiss Federal Institute of Technology (ETH) Zurich

Supervision

Matthias Meyer & Jan Beutel
Prof. Dr. Lothar Thiele

May 2018

Abstract

Identification of seismic events from continuously recorded audio data is an extremely difficult task. In spite of the vast amount of research in this field, the signal processing and event detection algorithms have not yet been fully established. The main goal of this project is to develop a signal processing method on a seismic event detection platform. This report presents several approaches for detecting seismic events with a noisy background and evaluates their performance in different scenarios.

Keywords: Audio signal processing, Seismic event, Embedded system.

Contents

Chapter 1

Introduction

Recently, wireless sensor networks (WSN) with acoustic sensors are widely used in monitoring and tracking seismic activities. Since the workplaces of these devices are usually energy limited, their energy efficiency is significantly important. Due to the high sampling rate of the input data, the transmission via WSN and its temporary storage on the node's memory have a big influence on the energy budget. Therefore, to reduce the energy consumption, unnecessary data transmissions and storage should be avoided.

1.1 Motivation

In the PermaSense project a system was installed in the high-altitude Swiss Alps with a WSN for collecting geophysical data [1]. This system aims to provide a long-term, high-quality wireless sensing and data recovery solution with near complete recovery and near real-time delivery of the data [2]. The sensor nodes of this system record and transmit continuous audio stream to a server through WSN for further classification.

When using a sensor network for data gathering in extremely harsh environments, decreasing energy consumption is of paramount importance. To enhance the energy efficiency of this system, we developed a pre-classification method on the nodes to automatically discard the unwanted data.

To be more specifically, an event detection algorithm was implemented on an embedded platform, which would serve as a node in WSN to filter out the uninteresting data before further processing. Once any seismic event is observed, the event data will be further classified on this embedded platform, which is challenging, since the processing capability and the memory size are limited.

Compared with the original design this new platform is not only able to improve the energy efficiency, but also detect seismic events earlier. Instead of warning neighboring areas of urgent seismic activities such as rockfalls after classification on the server, as shown in Figure 1.1, the new platform in Figure 1.2 can detect and alarm on the nodes themselves. Furthermore, the pre-classification on the nodes can reduce the amount of data transmitted to the server.

Figure 1.1: In the current system, the event classification only runs on the server. Since the transmission takes some time, there exists certain delay for mountaineers to receive an alarm from the warning system and this delay may cause some damages.

Figure 1.2: In the new system the local nodes can also classify seismic event, making it possible to warn mountaineers earlier and to prevent some potential damages.

1.2 Goal

The main goal of this project is to develop a signal processing method on a seismic event detection platform. Compared with the original design in Figure 1.3, this platform should be able to record seismic data, process it for transmission and perform pre-classification. Everything should run on an embedded platform, as shown in Figure 1.4. The result would be a system, which is able to perform multiple inter-dependent tasks and to stop the processing, if an intermediate result is sufficient to make a decision.

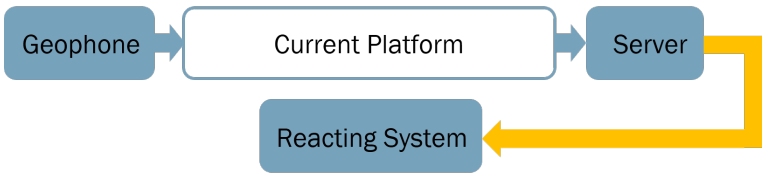

Figure 1.3: When any seismic data is recorded by the Geophone sensors, our current platform transmits the input data to the server directly via WSN. Once the audio data is classified as urgent event, the reacting system will be informed and warn the neighboring area.

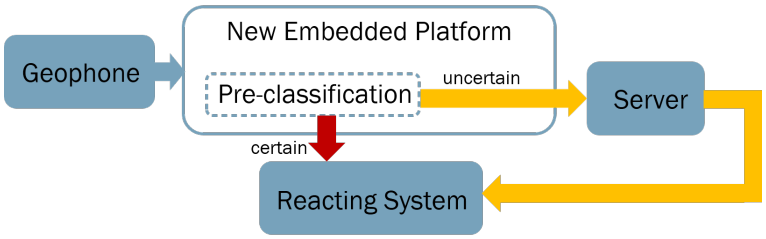

Figure 1.4: The new embedded platform can pre-classify the input data. If the result is certainly urgent, then the reacting system will be triggered directly by the embedded nodes. When the result of the pre-classification is unclear, the data will be transmitted to the server for further classification.

1.3 Outline

In the next Chapter 2, we will first describe the event detection algorithms for seismic research that we are going to implement and evaluate. Then our system model and experimental setup will be presented in Chapter 3. After the introduction of our system setup, the experimental results will be demonstrated in Chapter 4, where we will evaluate the performance of each event detection algorithm. Finally, we will conclude our works and further discuss about the future outlook in Chapter 5.

Chapter 2

Theory

Identification of seismic events from continuously recorded audio data is an extremely difficult task. In spite of the vast amount of research in this field, the signal processing and event detection algorithms have not yet been fully established. Until now, there exist a wide spectrum of triggering algorithms, ranging from a very simple amplitude threshold type to sophisticated ones based on pattern recognition approaches [3]. In this section, several approaches for detecting seismic events with a noisy background are introduced and their performance will be evaluated in the following chapters.

2.1 Event Detection in Time Domain

2.1.1 Amplitude Threshold Triggering (ATT)

ATT detects seismic events by a user-defined threshold value. This algorithm simply compares the signal amplitude with the threshold value. Once the value of an input signal exceeds the threshold, the seismic event will be noticed.

The threshold value of ATT can be set according to the background noise level. Therefore, even when the noise levels fluctuate, the algorithm will still be able to capture seismic events, without triggering falsely due to noise signals.

2.1.2 STA/LTA Triggering

STA/LTA Triggering computes the average energy level within a short-term average (STA) leading time window and the one in a long-term average (LTA) trailing time window, as shown in Figure 2.1. If the average value captured in the STA is larger than the background levels in the LTA, the ratio STA/LTA will be greater than 1. While the ratio between the two averages is higher than a specified threshold value, there will be a seismic event detected.

The advantage of the STA/LTA Triggering method is that high-amplitude signals may not trigger, if the increase of amplitude is not significant compared with the past. For instance, when a noisy background leads to a relatively large value of LTA, only an extraordinary increase of signal amplitude can rise up the value of STA to trigger an alarm.

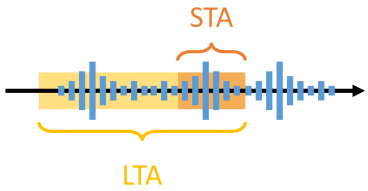

Figure 2.1: This figure illustrates the working principle of STA/LTA Triggering. The orange block reveals the short-term average (STA) leading window, while the yellow one is the long-term average (LTA) trailing window.

Standard

$$\begin{aligned} STA_i &= \frac{1}{ns} \sum_{j=i-ns}^{i} x_i^2 \\ LTA_i &= \frac{1}{nl} \sum_{j=i-nl}^{i} x_i^2 \\ \frac{STA}{LTA}_i &= \frac{STA_i}{LTA_i} \end{aligned} \tag{2.1}$$

Recursive

$$\begin{aligned} STA_i &= \frac{1}{ns} \times x_i + (1 - \frac{1}{ns}) \times STA_{i-1} \\ LTA_i &= \frac{1}{nl} \times x_i + (1 - \frac{1}{nl}) \times LTA_{i-1} \\ \frac{STA}{LTA}_i &= \frac{STA_i}{LTA_i} \end{aligned} \tag{2.2}$$

2.1.3 Z-detector

Z-detector indicates how many standard deviations σ a data point x_i is away from the sample's mean μ, assuming a Gaussian distribution. After making appropriate transformations to the selected feature space of the dataset, the z-detector of any data point can be calculated.[10] For instance, if the threshold is set to 1, the data point, which is more than 1 standard deviation afar from the mean value, will be considered as the start of a seismic event. As shown in Figure 2.2, if the data point falls into the yellow regions, then a seismic event will be detected.

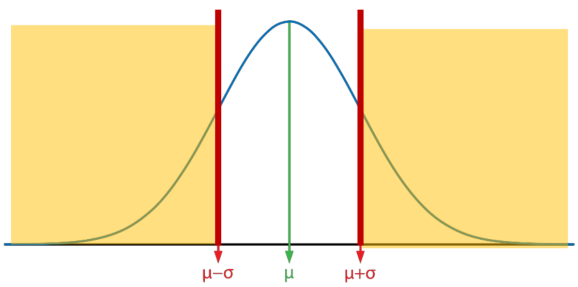

Figure 2.2: This figure illustrates the working principle of Z-detector. The red line is the threshold and the yellow region displays the valid area, where seismic events will be detected.

The advantage of Z-detector is that it is able to automatically adjust to variance of the background noise level. If the background variance is small, a minor change in input can cause a major change in output. If the background variance is large, a major output change requires a significant input change.[4]

$$Z_i = \frac{x_i - \mu}{\sigma} \tag{2.3}$$

2.2 Event Detection in Frequency Domain

2.2.1 Power Spectral Density (PSD)

PSD reveals the amplitude of "power" as a function of frequency, where 'power' is considered to be the average of the square of FFT´s magnitude. In other words, PSD shows the strength distribution of frequencies. Energy within a specific frequency range can be obtained by integrating PSD within that frequency range.

To use PSD for event detection, individual segments of data are compared with an selected detection threshold. The ability to detect individual frequency bands is an important alternative over time-domain based methods, since it eliminates the need for an a priori band-pass filtering. However, it also needs parameter settings, but may be able to detect more small events from a noisy background.[5]

$$Power = \frac{|amplitude\ of\ FFT|^2}{N},\ where\ N:\ the\ normalization\ factor. \tag{2.4}$$

Chapter 3

System Overview

In past, seismologists used to continuously record the seismic signal with analog seismographs on thermal ink paper. However, with the development of new measurement and digital techniques, seismographs have progressed from simple analog recording to digital recording based on microprocessors, micro-controller units (MCU), DSP and other advanced PC processors, making the signal processing and analysis easier and seismic event detection more reliable [3]. In this section, we will describe our experimental system. We developed an embedded platform for audio signal processing, where several algorithms for event detection are implemented and evaluated.

3.1 System Model

Figure 3.1 displays the original system model, which consists of a geophone sensor, a ADC circuit, an embedded platform, a WSN and a server. In the beginning, the analog voltage levels generated by a geophone sensor are sampled as digital signals by the ADC circuit. Afterwards, the amplitude of digital signals will be compared with a low threshold. If the data's value exceeds said low threshold, the digital signal will be fed into the embedded platform, which will then transmit the recorded signal to the server via the WSN directly without any further signal processing.

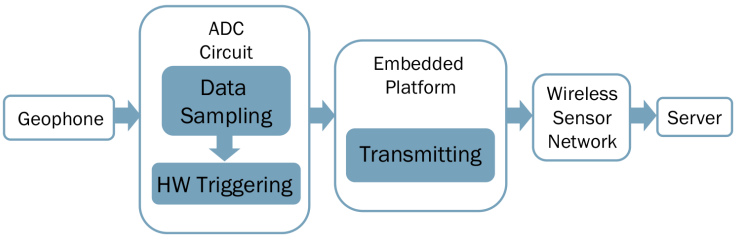

Figure 3.1: The model of the current system, where the embedded platform sends the input signal to the server without any further processing.

The new system model is shown in Figure 3.2, where two modules, namely signal processing and pre-classification, are added. Instead of bypassing the digital signals to the server directly, the input data will be further processed on the embedded platform.

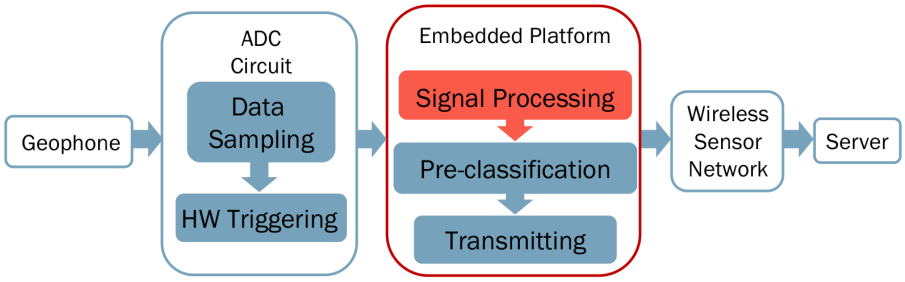

Figure 3.2: The model of the new designed system, where two modules, namely signal processing and pre-classification, are added.

In this project, we focus on signal processing module, where algorithms for event detection, Fast Fourier Transform (FFT) and band energy are implemented, as shown in Figure 3.3. The sampled digital signals will first pass through windowing scheme and then event detection triggering. Only a signal that exceeds a certain threshold will be considered as a valid seismic event. The event data will be on one hand stored in the on-chip memory and on the other hand subject to a FFT followed by energy band to adjust frequency resolutions. The resulting spectrogram is then processed by the pre-classification and the resulting decisions will be sent through the WSN.

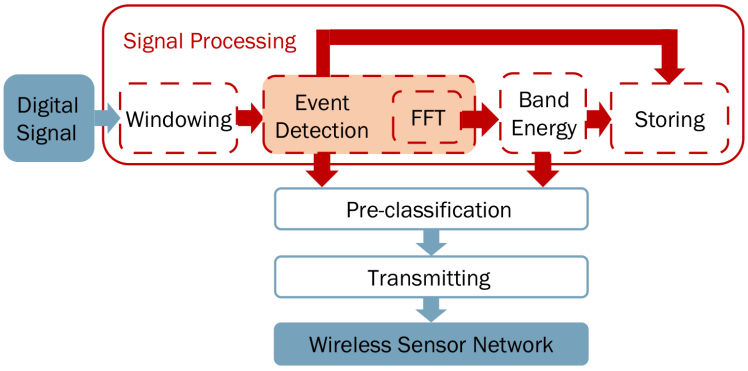

Figure 3.3: The details of the signal processing step in the newly designed system

3.2 Experimental System Setup

To develop and evaluate the performance of the signal processing module, a system for experiments is designed and can then be adopted on the edge nodes of the WSN. As shown in Figure 3.4, this system is composed of a PC, a USB key, a micro-controller unit (MCU) and a 5V power supply.

At first, the seismic audio data is put into the USB key via the PC. During the experiment, the MCU reads out the seismic audio data from the USB key and applies signal processing algorithms on it. For evaluation purposes, the MCU was designed to write out the results in text files to the USB key. Afterwards, we further analyze the results on PC with some Python programs.

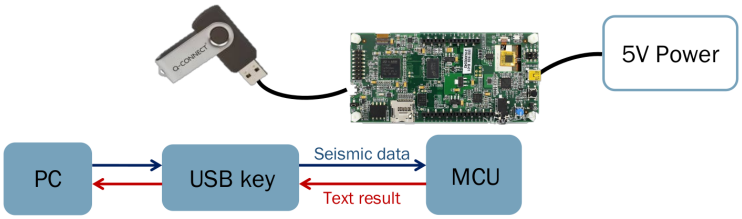

Figure 3.4: The system setup for experiments

3.2.1 Seismic Dataset

The seismic data for experiments was collected in Dirruhorn mountain by 6 geophone sensors, as shown in Figure 3.5. The dataset was collected from 10th July 2017 to 6th September 2017, consisting of 73251 recorded seismic events. The threshold value for the amplitude triggering implemented on the geophone sensors was about 20. Once the input data exceeded this threshold, the recording started. Nevertheless, when this seismic dataset was collected, the system was designed to trigger all sensors at the same time. Therefore, the lengths of data from these 6 channels are the same. In other words, once a sensor was triggered, the other 5 sensors started to record, too.

Besides the audio data from geophone sensors, the rain information was also provided. The time unit of recording is 2 minutes. This rain dataset contains information about the rain intensity, the rain accumulation and the rain duration. In our experiment, we use the rain intensity as the standard to label rainy or non-rainy events.

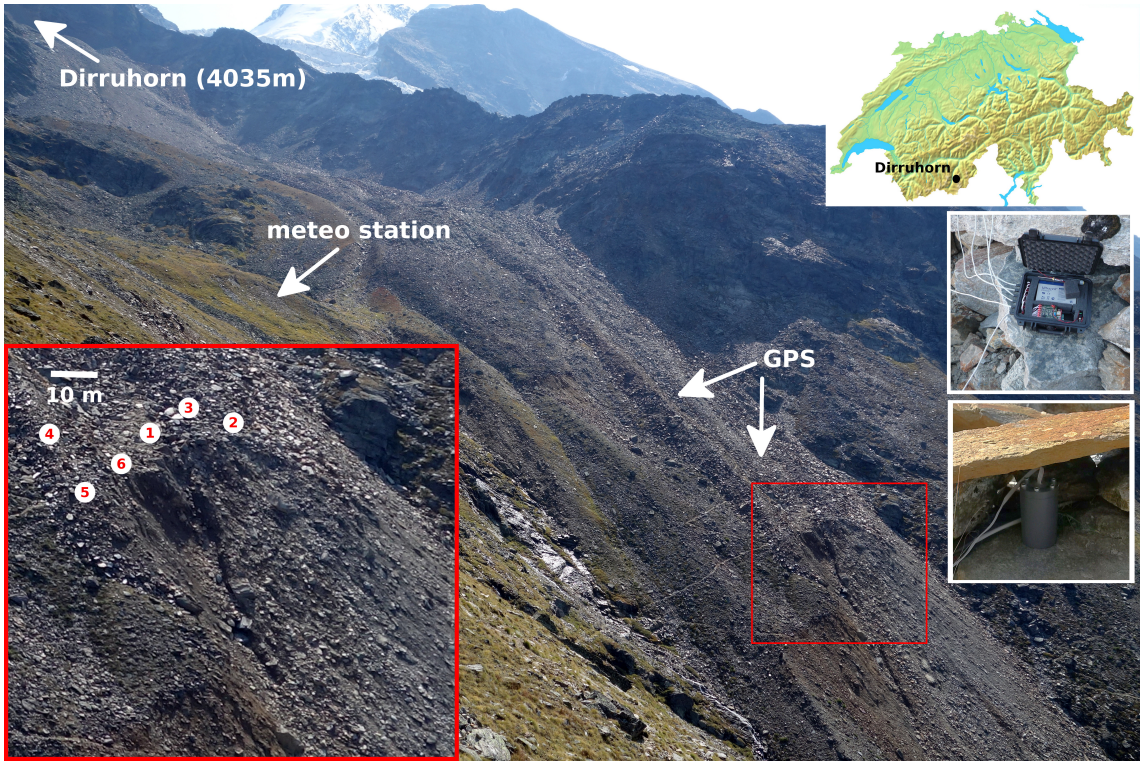

Figure 3.5: The picture of 6 geophone sensors distributed in Dirruhorn mountain

3.2.2 Embedded Platform

To build the embedded platform, we chose an ARM Cortex-M4 core-based STM32F469NIH6 MCU, which is integrated on the STM32F469I discovery board from STMicroelectronics, for the experiments. This MCU contains a 32-bit Cortex-M4 CPU with 12 general purpose registers, a floating point unit (FPU) with additional 32 single precision registers and a full set of DSP instructions [6].

Since the ARM DSP provides a solid library, the CMSIS DSP Software Library [7], it makes the implementation of FFT much easier. Furthermore, the library includes Low Layer (LL) and Hardware Abstraction Layer (HAL) APIs that cover the microcontroller hardware, together with an extensive set of examples running on STMicroelectronics boards, as shown in Figure 3.6. Our experimental platform was adapted from one of the official example, namely "audio playback and record" [8].

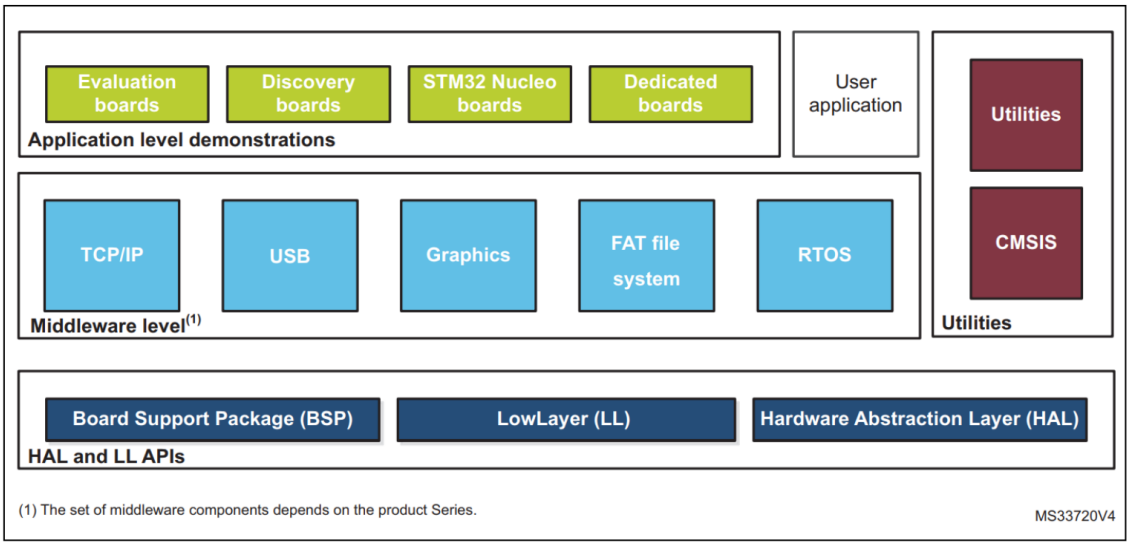

Figure 3.6: STM32CubeF4 firmware components

Chapter 4

Experimental Results

In the last chapter, the system setup for experiments is demonstrated. By analyzing offline data collected on geophone sensors in a real-world environment, some interesting patterns can be seen. This chapter reveals the results of experiments. Since every algorithm has their own features, they may fit different criteria in certain scenarios individually. By observing their behaviors in various scenes, their performances can then be evaluated.

4.1 Initialization

4.1.1 Selection of Algorithms' Parameters

Since some event detection algorithm contain one or two tunable parameters, which influence the results significantly, these parameters should be set before the experiment. According to [9], a typical value of STA duration is between 1 and 2 second for regional events and a LTA duration of 60 seconds is a common initial value. Therefore, we chose to set the length of STA as 1 second and length of LTA as 60 seconds. Since to some extent the STA works like a signal filter, it makes sense to select the windowing length of Z-detector and PSD the same as the length of STA. In the end, the parameters were selected as shown in the Table 4.1.

Table 4.1: The tunable parameters of event detection algorithms

ATT	-	
STA/LTA	length of STA = 1 (sec)	length of LTA = 60 (sec)
Z-detector	length of windowing = 1 (sec)	
PSD	length of windowing = 1 (sec)	

4.1.2 Selection of Experimental Thresholds

After fixing the parameters of algorithms, the thresholds to decide the triggering condition need to be chosen, too. Figure 4.1 reveals the result of applying the event detection algorithms on a known rockfall event. The output values from these algorithms are in very different ranges, which makes them difficult to be compared under the same threshold. Therefore, we need to find several sets of threshold with individual threshold values for each algorithm.

Due to the range of input data is from -2047 to 2047, we first directly picked 5 thresholds for ATT as 250, 500, 1000, 1500 and 2000. Next, we tuned through all numbers to find the values, that

make STA/LTA, Z-detector and PSD to generate the same number of total triggered events. Table 4.2 lists these 5 sets of the event detection algorithms' thresholds.

Table 4.2: The tuned values of triggering thresholds

	ATT	STA/LTA	Z-detector	PSD
1st	250	1.1	1	700
2nd	500	2.7	1.7	2700
3rd	1000	6.5	3	13000
4th	1500	10	3.5	25000
5th	2000	13	4	40000

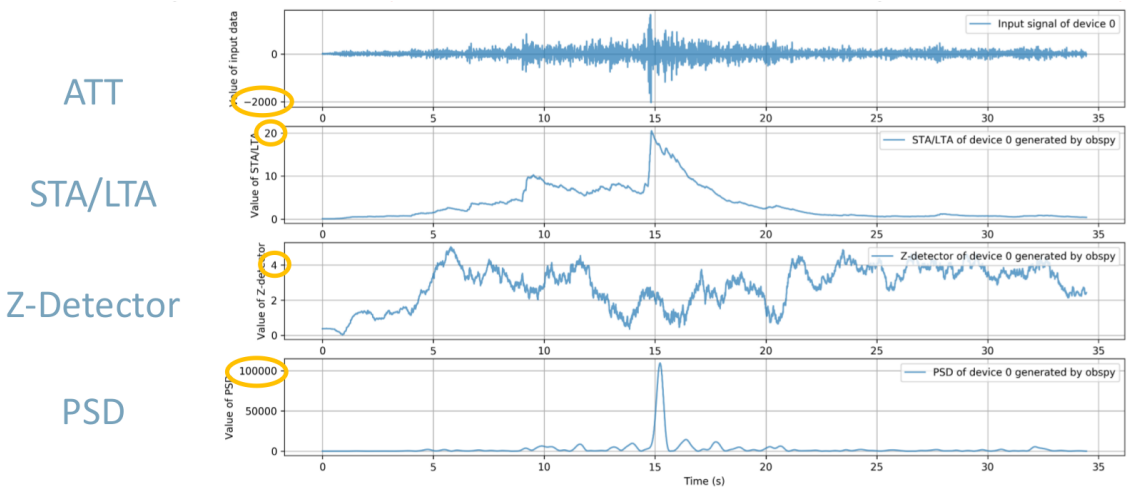

Figure 4.1: The result of applying event detection algorithms on a known rockfall event

4.2 Event Detection

4.2.1 The Analysis of Individual Detected Events

After setting the algorithms' parameters and the experimental thresholds, we test the event detection functions with the offline seismic data in all conditions of the 5 threshold sets. Figure 4.2, for example, displays the result of a known rockfall event in the 2nd set of threshold. At first, the input data and output values of the triggering functions are quite small. As the amplitude of input signals increases, the values of the triggering functions also increase. The accumulating speed of Z-detector is obviously the fastest and it detects this seismic event earliest within all triggering algorithms at 5 seconds after the recording. STA/LTA is the next one to detect an event, namely at 8 seconds. The detecting moments of ATT and PSD are close, i.e. at 15 seconds. In addition, the shape of PSD values looks quite similar to the one of the input signal's amplitude, since the working principles of ATT and PSD are both based on the input signal's energy.

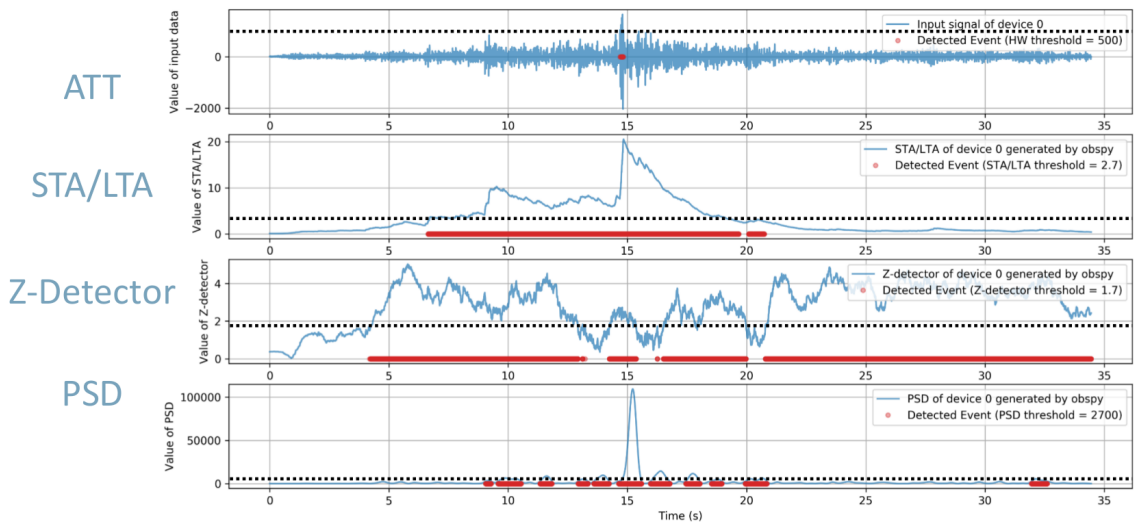

Figure 4.2: The analysis result of a known rockfall event under 2nd set of threshold. The red dots label the moments where the signal exceeds the individual threshold value for every algorithm.

The Timing of Detection

In our system the recording of input data only starts after detecting a seismic event. As one of our goals is to perceive an urgent seismic event as early as possible, the timing of detection is of great importance. To evaluate the detecting speed, the differences of the point in time to detect seismic events between ATT and other 3 algorithms are computed as shown in Figure 4.3 and averaged over the whole dataset. The static results are illustrated in the Figure 4.4.

Compared with ATT, STA/LTA, Z-detector and PSD all more or less have a delay for the event detection. In the beginning, the delays are all negligibly small. However, as the thresholds rise up, they all increase significantly, especially for Z-detector. The reason for these 3 algorithms to cause delays is understandable, since they all function like filters, making them good for de-noising but bad for fast reacting. In conclusion, ATT is the fastest to detect seismic events, while Z-detector is the slowest.

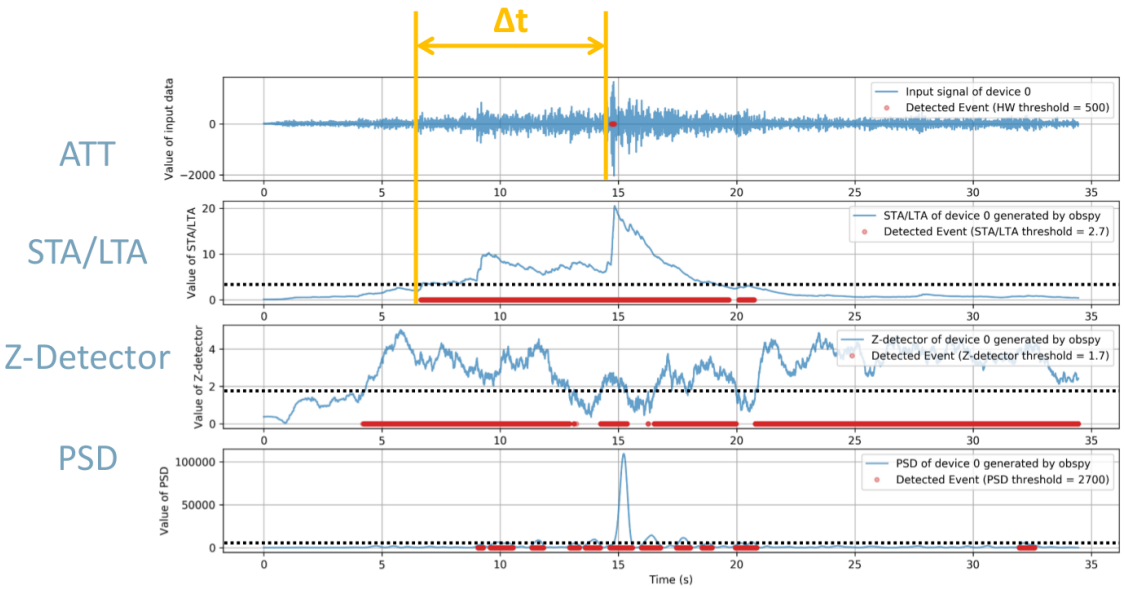

Figure 4.3: This figure shows the way to compute the time differences between ATT and other 3 algorithms for the evaluation on their detecting speed.

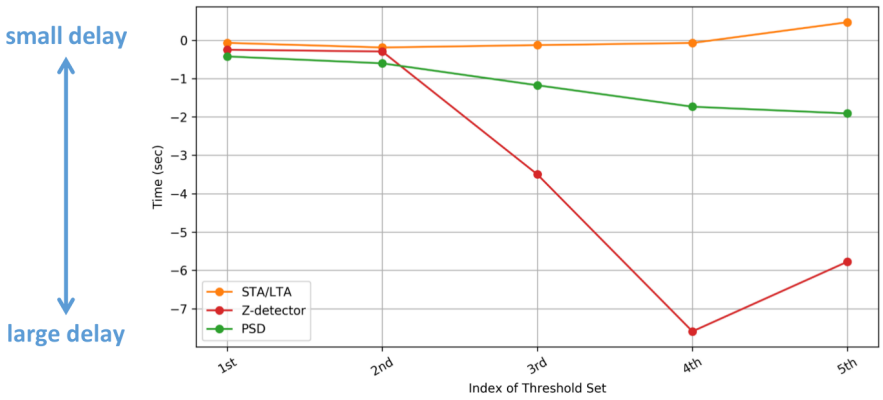

Figure 4.4: The average time delay for STA/LTA, Z-detector and PSD to detect seismic events with respect to ATT

4.2.2 The Distribution of Detected Events

Figure 4.5 to Figure 4.9 demonstrate the result on 2017/7/18 under increasing thresholds from the first set [250, 1.1, 1, 700] to the last set [2000, 13, 4, 40000] for ATT, STA/LTA, Z-detector and PSD. As the thresholds rise up, the numbers of detected events of every algorithms decrease and the distribution concentrates on the rainy period. It means most of seismic events are stimulated by rain, which either directly hits on the sensor or indirectly causes rockfalls. While there are still some events scattered in non-rainy period, they may have been generated by strong wind, helicopters, mountaineers or rockfalls. Because at that time the data was collected, the recording techniques were limited, we can not be sure about the exact reason.

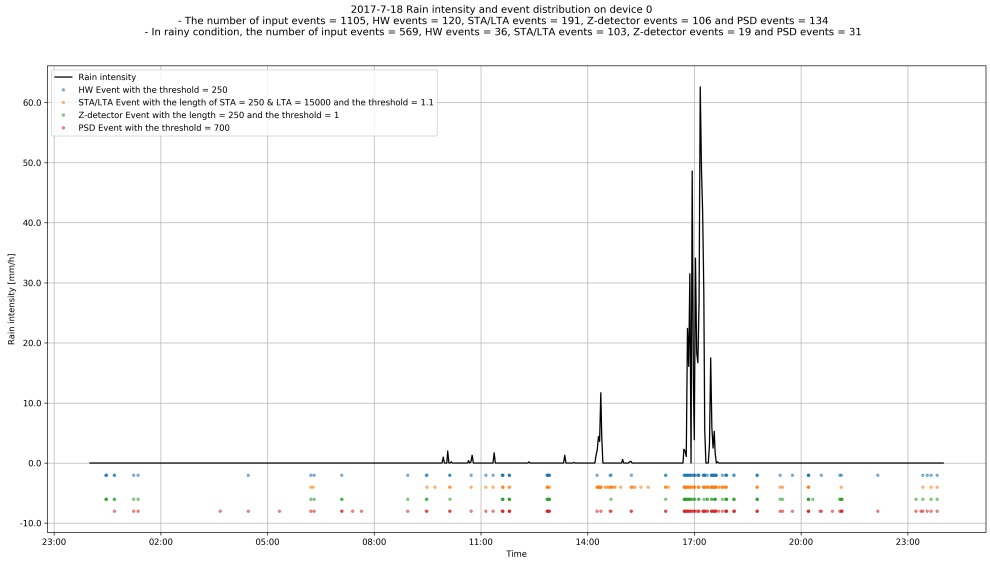

Figure 4.5: The distribution of the detected events by ATT, STA/LTA, Z-detector and PSD on 2017/7/18 in the first threshold set

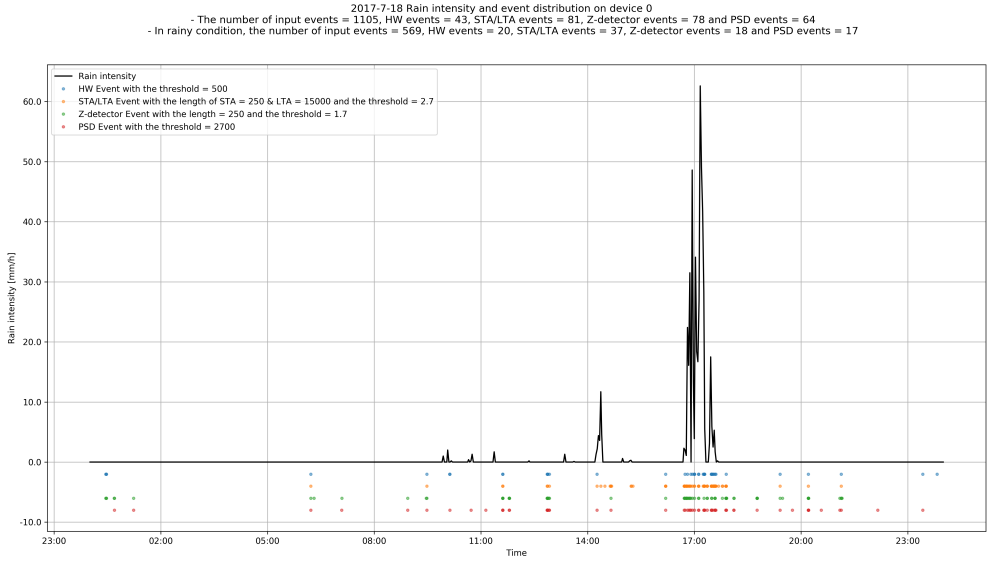

Figure 4.6: The distribution of the detected events by ATT, STA/LTA, Z-detector and PSD on 2017/7/18 in the second threshold set

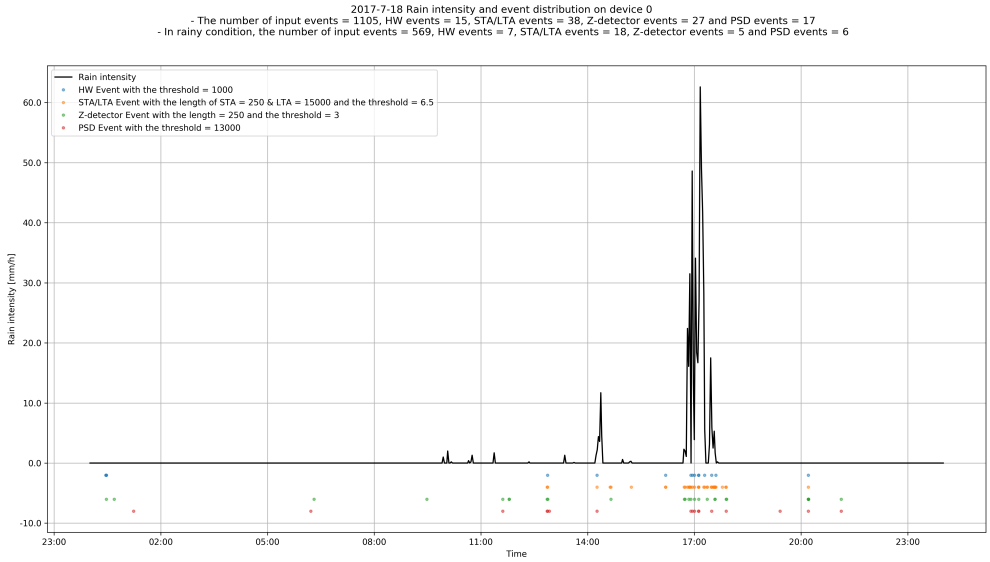

Figure 4.7: The distribution of the detected events by ATT, STA/LTA, Z-detector and PSD on 2017/7/18 in the third threshold set

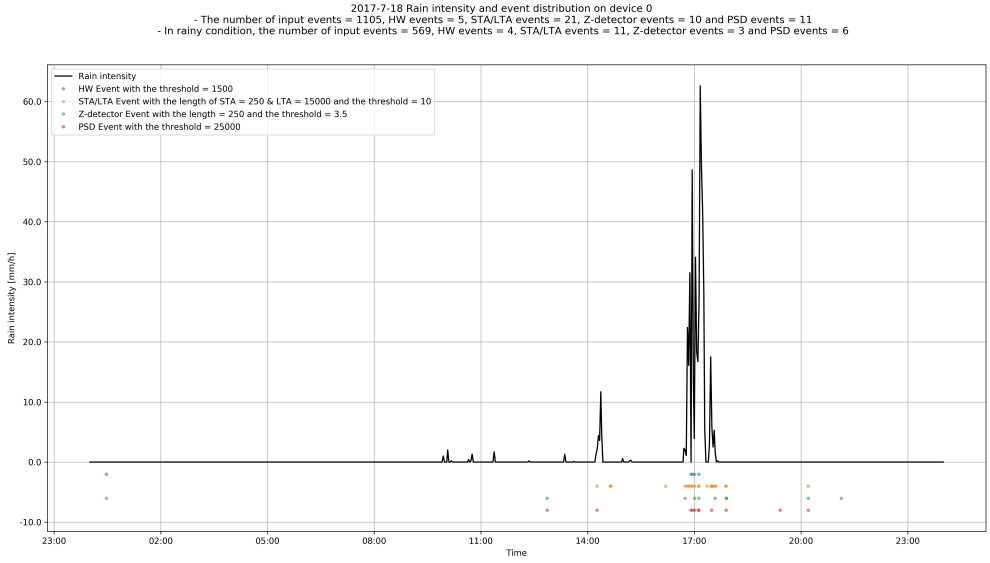

Figure 4.8: The distribution of the detected events by ATT, STA/LTA, Z-detector and PSD on 2017/7/18 in the forth threshold set

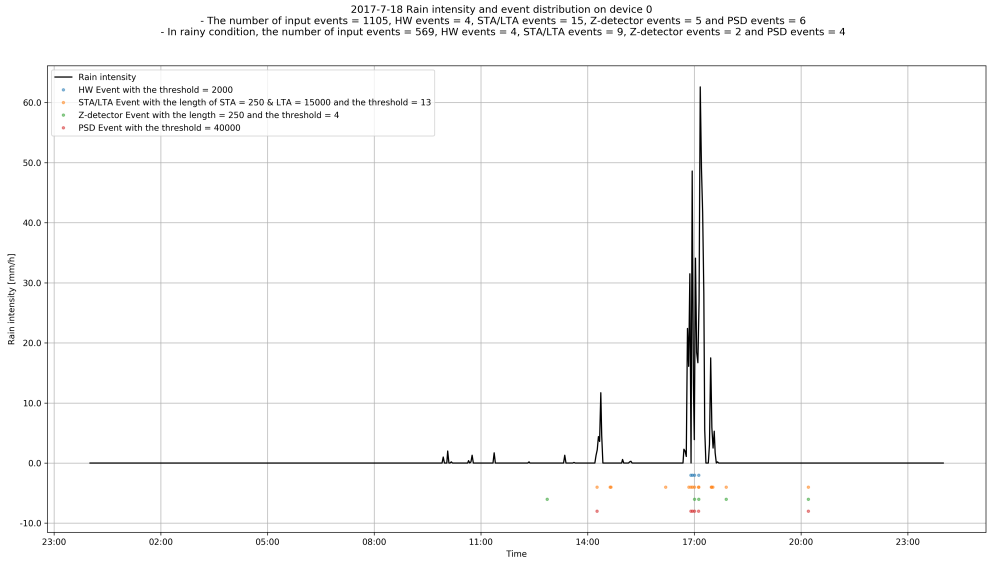

Figure 4.9: The distribution of the detected events by ATT, STA/LTA, Z-detector and PSD on 2017/7/18 in the fifth threshold set

The Ability to De-noise

Inspired by the uneven distribution of the detected events, we further analyze the algorithms' ability to de-noise. In this test, rain is regarded as the source of noise. Therefore, the methods that generate the fewest events at rainy period are considered to be the one performing best against noise. Figure 4.10 plots the proportion of the detected rainy events in all the detected events for different algorithms. We can find the percentage value of ATT is the highest, which means ATT's ability to de-noise is the worst, while the Z-detector with lowest percentage value performs best.

$$The\ ratio\ of\ detected\ rainy\ events\ on\ rainy\ days = \frac{The\ number\ of\ detected\ rainy\ events}{The\ number\ of\ total\ detected\ events} \tag{4.1}$$

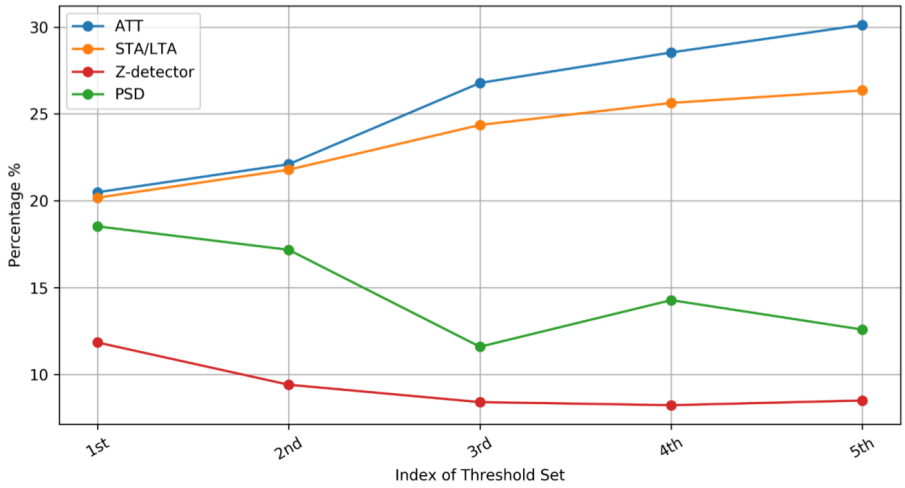

Figure 4.10: The ratio of detected rainy events and all the detected events on rainy days with different thresholds

The Similarity of Algorithms

From the distribution of detected events, we can notice that there are some events only detected by some algorithms instead of all. Figure 4.11, for example, demonstrates an event that STA/LTA, Z-detector and PSD can detect, but ATT misses in the 2nd set of thresholds. Since the thresholds of all algorithms are chosen under the same criterion, their results should be comparable.

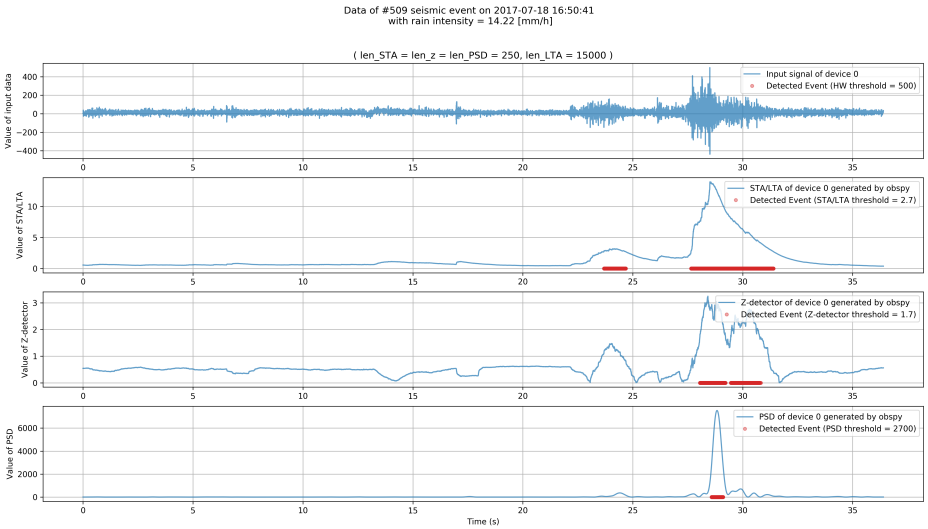

Figure 4.11: Analysis result on an event with relatively small amplitude

Inspired by the fact that different events can only be detected by different algorithms, we would like to further compute the similarity of these methods. First, we compute the common detected rainy events between ATT and other algorithms as shown in Figure 4.12 and then divide it by the total number of detected events. In the Figure 4.13, the overlapping ratios of detected events with respect to the result of ATT under different thresholds is illustrated. The overlapping signals' percentage between ATT and PSD is relatively high due to their same working principle, based on signal's energy. However, the similarity of Z-detector is significantly lower, which means many

events detected by ATT will not be triggered by Z-detector. Therefore, if we would like to increase the coverage of detection, i.e. reduce the number of seismic events that our current platform misses, Z-detector may be a good candidate to be implemented on the embedded platform, since we already have ATT on our current ADC circuit.

$$The\ overlapping\ ratio\ of\ X\ algorithm = \frac{The\ number\ of\ detected\ events\ by\ ATT\ and\ X\ both}{The\ number\ of\ detected\ events\ by\ ATT\ only} \tag{4.2}$$

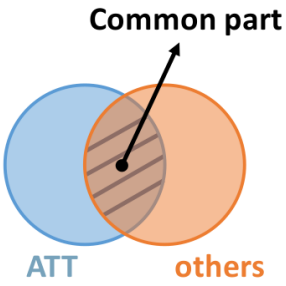

Figure 4.12: The common part between ATT and other algorithms

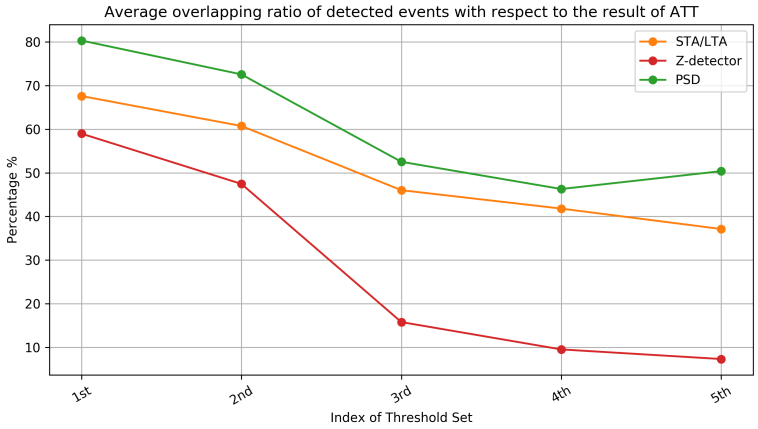

Figure 4.13: The overlapping ratios of detected events with respect to the result of ATT with different thresholds

4.2.3 The Computational Time

In this part, we measured the computational time of each algorithms on our embedded platform. The following Table 4.2 lists the result. Since in practice, the ATT is most likely to be implemented on hardware, we only measured the time consumption for STA/LTA, Z-detector and PSD. Unsurprisingly, PSD needs the longest time to process, since one of its steps is FFT, whose computational cost is very high. The average computational time of Z-detector is the shortest due to having a simple program structure.

Table 4.3: The average computational time of triggering algorithms on the embedded platform

	ATT	STA/LTA	Z-detector	PSD
Time(μs)	0	3.8	1.46	598.8

Chapter 5

Conclusion and Outlook

5.1 Conclusion

In this project, we implemented and evaluated several event detection algorithms for seismic research on an embedded platform. Table 5.1 lists the evaluation results from every tests.

At first, we have shown the simplest algorithm, ATT, is the fastest to detect seismic events within the 4 algorithms, while there exist a relatively large delay for Z-detector. However, when it comes to the ability to de-noise, Z-detector performs best and ATT is the worst one. In addition, the detection result of Z-detector is the most different one from ATT, which means Z-detector is most able to detect the events that ATT misses. In the computational speed test, PSD is the worst due to the need of FFT and the Z-detector is the second best, a little slower than ATT, which will probably be implemented on the ADC circuit.

Although there is no perfect algorithm and every algorithm has their advantages and disadvantages, we can trade some function off for the more important one. For example, if detecting seismic events earlier or saving energy consumption is the most important factor, then we may need to select ATT. However, if eliminating the influence by noise is the first priority, Z-detector may be the best choice. While in the cases that these 2 factors are the same important, the balanced solution, STA/LTA Triggering, will be a good candidate.

	ATT	STA/LTA	Z-detector	PSD	Result
Simlilarity to ATT			○	✕	**Detect missing events** -> **Accuracy↑**
Timing of Detection	○	○	✕		**Detect events earlier** -> **Reaction speed↑**
Ability to De-noise	✕		○	○	**Redundant detection↓** -> **Energy cost↓**
Computational Speed	○		○	✕	**Energy consumption↓** -> **Energy cost↓**

Figure 5.1: The table lists all the results from the experiments

5.2 Outlook

Based on the contribution of this project, in order to detect seismic events more accurately and efficiently, there are several directions able to be further explored in the future.

- Improve the recording coverage of event detection:

  Since currently the data recording only replies on the hardware amplitude triggering on ADC circuit, there may be some events missed. Using different algorithms may increase the accuracy of event detection.

- Eliminate the influence of noise:

  Due to ATT's high sensitivity to noise, there are plenty of events with a short recored period. Since every detected event costs energy to process and even transmit, implementing other event detection methods, which can adjust to the noise level in the background, to de-noise is of paramount importance.

Reference

[1] Igor Talzi, Andreas Hasler, Stephan Gruber, and Christian Tschudin. *"PermaSense: Investigating Permafrost with a WSN in the Swiss Alps."*, In: Proceedings of the 4th Workshop on Embedded Networked Sensors

[2] Jan Beutel et al, *"PermaDAQ: A Scientific Instrument for Precision Sensing and Data Recovery in Environmental Extremes"*, In: Proceedings of the 2009 International Conference on Information Processing in Sensor Networks

[3] B. K. Sharma, Amod Kumar and V. M. Murthy, *"Evaluation of Seismic Events Detection Algorithms"*, In: Journal Geological Society Of India Vol.75, March 2010, pp.533-538

[4] Mitchell Withers*, Richard Aster and Christopher Young et al, *"A Comparison of Select Trigger Algorithms for Automated Global Seismic Phase and Event Detection"*, In: Bulletin of the Seismological Society of America, Vol. 88, No. 1, pp. 95-106, February 1998

[5] Yoones Vaezi and Mirko van der Baan, *"Analysis of instrument self-noise and microseismic event detection using power spectral density estimates"*, In: SEG Houston 2013 Annual Meeting

[6] *"STM32F469 data sheet"*, In: http://www.st.com/content/st_com/ en/support/resources/resource-selector.html?querycriteria= productId=LN1876$$resourceCategory=technical_literature$ $resourceType=datasheet.

[7] *"CMSIS DSP Software Library"*, In: http://www.keil.com/pack/doc/CMSIS/DSP/html/index.html

[8] *"Audio playback and recording using the STM32F4DISCOVERY"*, In: http://www.st.com/resource/en/application_

[9] Amadej Trnkoczy (formerly Kinemetrics SA), *"Understanding and parameter setting of STA/LTA trigger algorithm"*, In: ftp://hazards.cr.usgs.gov/Eq_Effects/GeekPack/Procedures-Configs-Info/1_Dataloggers/K2-Altus/Sta-Lta.PDF

[10] Sergio Santoyo, *"A Brief Overview of Outlier Detection Techniques"*, In: https://towardsdatascience.com/a-brief-overview-of-outlier-detection-techniques-1e0b2c19e561