Title
Vehicle Detection by Sensor Network Nodes

Authors
Ding, Jiagen
Cheung, Sing-Yiu
Tan, Chin-woo
et al.

Publication Date
2004-10-01

CALIFORNIA PATH PROGRAM
INSTITUTE OF TRANSPORTATION STUDIES
UNIVERSITY OF CALIFORNIA, BERKELEY

Vehicle Detection by Sensor Network Nodes

Jiagen(Jason) Ding, Sing-Yiu Cheung,
Chin-woo Tan and Pravin Varaiya
University of California, Berkeley

California PATH Research Report
UCB-ITS-PRR-2004-39

This work was performed as part of the California PATH Program of the University of California, in cooperation with the State of California Business, Transportation, and Housing Agency, Department of Transportation; and the United States Department of Transportation, Federal Highway Administration.

Report for Task Order 4224

October 2004

ISSN 1055-1425

Vehicle Detection by Sensor Network Nodes

by

Jiagen(Jason) Ding, Sing-Yiu Cheung, Chin-woo Tan and Pravin Varaiya

,

Abstract

Vehicle Detection by Sensor Network Nodes

by

Jiagen(Jason) Ding, Sing-Yiu Cheung, Chin-woo Tan and Pravin Varaiya

,

This report presents the algorithm development and experimental work of the sensor node signal processing for vehicle detection. The signals used for vehicle detection are acoustic and magnetic signals. The acoustic signals are characterized by short time FFT analysis and two acoustic vehicle detection algorithms are proposed: the Adaptive Threshold algorithm (ATA) and the Min-max algorithm (MMA). The ATA detects vehicle by searching for a sequence of 1's after slicing the acoustic energy curve using an adaptive threshold. The MMA detects vehicles by searching the local maximum in the acoustic energy curve. Real time tests and offline simulations demonstrate the effectiveness of the two algorithms. For magnetic signals, a simple threshold slicing algorithm is utilized and real time tests give good performance. Finally, FPGA implementation of ATA is also presented for power efficiency requirement and the implementation justifies the use of dedicated hardware for low power implementation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The idea of deploying sensors to monitor/measure the behaviour of a system is not novel; however, some of the technological and economic issues remain challenging. In particular, many issues need to be considered for the price one is willing to pay for collecting information and making system improvement. For example, can we collect the data we want with only wired sensors? Wireless sensors offer the flexibility advantage, but just like any portable device, the limit of the energy source is always a concern. Can we deploy a network of sensors so that we have a high density and fidelity of instrumentation? A high density of sensors is an obvious benefit, but it also means more cost. In other words, is large-scale deployment economically feasible? All these issues, nonetheless, can be categorised into three inter-related categories: cost, benefit, and technological limitation. These three issues will dictate the choice of the sensing device for applications such as vehicle detection.

A vehicle detection system requires four components: a sensor to sense the signals generated by vehicles, a processor to process the sensed data, a communication unit to transfer the processed data to the base station for further processing, and an energy source.

Current vehicle detection technologies are not suitable for large scale deployment as they are usually destructive, disruptive and have a high cost of installation and maintenance. Thanks to MEMS (micro electro-mechanical systems) technology, all of these components could now be integrated into a tiny single

Figure 1.1: Installation of Inductive Loop Detector

device(a Mote). Thus, future vehicle detection system can be a large scale sensor network formed by interconnecting low cost sensor nodes (tiny motes) via wireless communication, which can be deployed easily. One of such sensor nodes is developed in Smart-Dust project by the Department of EECS at University of California at Berkeley[1, 2].

In the following, current vehicle detection technologies will first be reviewed. Secondly, the sensor node (Mote) technology will be then presented, which may be used for future vehicle detection system. Finally, the sensor network will be briefly introduced with respect to the vehicle detection application.

1.1 A brief overview of current technologies[3]

1.1.1 Intrusive sensors

Intrusive sensors are those that need to be installed under the pavement, in saw-cuts or holes on the roads. Popular intrusive sensors include inductive loops, magnetometers, micro-loop probes, pneumatic road tubes, piezoelectric cables and other weigh-in-motion sensors. The main advantage of these sensors is their high accuracy for vehicle detection while the drawbacks include the disruption of traffic for installation and repair, resulting in a high installation and maintenance cost. Figure 1.1 shows the intallation of inductive loop detectors on a road.

1.1.2 Non-intrusive Sensors

To overcome the disadvantage of intrusive sensors, nonintrusive sensors were developed such as those aboveground vehicle detection sensors. Aboveground sensors can be mounted above the lane of traffic or

Figure 1.2: Video Image Vehicle Monitoring(left) and Passive Infrared Vehicle Monitoring(right)

Figure 1.3: Dust Family

on the side of a roadway where they can view multiple lanes of the traffic at angles perpendicular to or at an oblique angle to the traffic flow direction. Technologies used in aboveground sensors include video image processing (VIP), microwave radar, laser radar, passive infrared, ultrasonic, passive acoustic array, and combinations of these sensor technologies. However, these non-intrusive sensors tend to be large size and power hunger. Figure 1.2 shows that the imaging processing and passive infrared technologies are used in vehicle monitoring and detection.

1.2 Smart-Dust Sensor Node-Hardware Platform

Smart-Dust sensor[2] is one of several potential sensor nodes which could be used for the future vehicle detection system. In a Smart-Dust sensor node, essential components for vehicle detection (processor, memory, sensor and radio) are integrated together as small as a quarter through MEMS technology. Together with its low power design[4, 5], a network of smart-Dust sensor nodes is a feasible candidate for performing the vehicle detection. Figure 1.3 shows the different generations of Smart Dust sensor nodes(Motes). The left photo is a 1st generation Smart Dust sensor node called “Rene Mote”. To its right, from left to right, are the “MICA Mote”(2dn generation), “MICA2 Mote”(3rd generation) and “MICA2-Dot Mote(3rd generation)”.

Smart-Dust sensor nodes are designed by EECS department in UC Berkeley and Intel [1] in a modular component approach and it consists of two major components: mother board and sensor board. Thus, different sensor boards could be attached to the same mother board for different applications. Smart-Dust sensor node could potentially be used in a wide range of applications such as vehicle detection, enemy montioring in the battlefield, temperature measurement in a building, evironmental monitoring and etc. In the following, basic components of Smart Dust will be addressed in more detail.

Basic Components of Smart Dust

The basic components of MICA mote(Fig. 1.4) in the Smart-Dust family are listed in Table 1.1. The mother board consists of an Atmel 90LS8535 processor, 512KB SRAM, 8KB Flash RAM and a RF transceiver for wireless communication. The Senor board consists of a 10-bit analog to digital converter, a Magnetomer(Honeywell HMC1002), a temperature sensor, a photo camera and an accelerometer sensor.

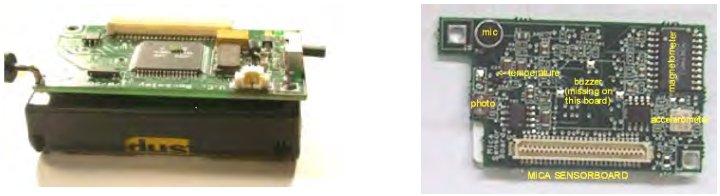

Figure 1.4: MICA Mote

Table 1.1: Components of Smart Dust Mote

Mother Board	Sensor Board
Atmel 90LS8535 processor (clocked at 4 MHz)	10-bit analog to digital converter
RF Monolithics transceiver (916.50 MHz)	Magnetometer(Honeywell HMC1002)
512KB SRAM, 8KB Flash RAM	Microphone (Panasonic WM-62A)
	Temperature Sensor
	Photo Camera
	Accelerometer Sensor

In a vehicle detection system, the sensors we adopted are the magnetometer and acoustic sensors. Next, the basic operating principles of magnetometer and acoustic sensors are reviewed.

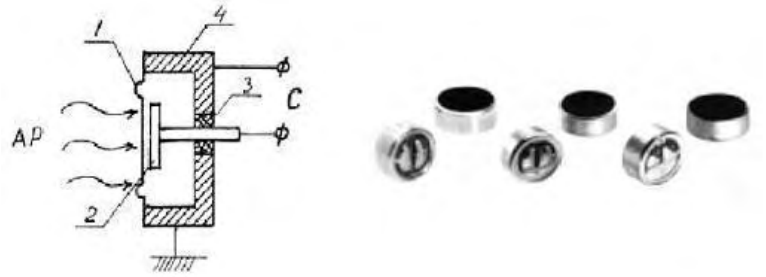

Figure 1.5: Condenser Microphone, AP-the acoustic pressure, C-the variable capacitance, 1-the metal diaphragm, 2-the metal disk, 3-the insulator and 4-the case

1.2.1 Acoustic Sensors

The acoustic sensor in the Smart Dust sensor node is a condenser type microphone. The schematic for an typical condenser acoustic sensor is shown in Fig. 1.5. It includes a stretched metal diaphragm that forms one plate of a capacitor. A metal disk placed close to the diaphragm acts as a backplate. A stable DC voltage is applied to the plates through a high resistance to keep electrical charges on the plates. When a sound field excites the diaphragm, the capacitance between the two plates varies according to the variation in the sound pressure. The change in the capacitance generates an AC output proportional to the sound pressure, which is ultralow-frequency pressure variation. A high-frequency voltage (carrier) is applied across the plates and the acoustic sensor output signal is the modulated carrier.

The photo in the right of Fig 1.5 shows the Panasonic WM-62A condenser microphones used in Smart Dust Motes. Figure 1.6 shows a typical vehicle acoustic signal waveforms. In Fig. 1.6, the sampling frequency is 64 Hz and the waveform amplitude is the raw ADC readouts.

The problem for acoustic sensor vehicle detection is to achieve robust vehicle detection under various acoustic noise corruption. Since the sensor nodes (Smart Dusts) are powered by battery, the solution has to be low power. This report will propose two state machine based vehicle detection algorithms, which achieve relatively reliable detection. Dedicated hardware implementation of the detection algorithm is also proposed for satisfying the power efficiency requirement.

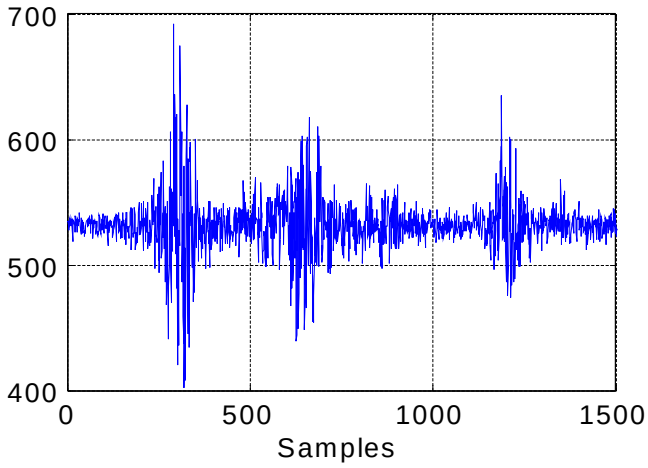

Figure 1.6: Waveforms of Acoustic Signals Emitted from Cars

1.2.2 Magnetic Sensors

The Honeywell HMC1002 magnetometer on the MICA sensor board is a magnetoresistive sensor. The anisotropic magnetoresistive (AMR) sensor is one type that has a wide Earth's field sensing range and can sense both the strength and direction of the Earth field[6].

The AMR sensor is made of a nickel-iron (Permalloy) thin film deposited on a silicon wafer and patterned as a resistive strip. The strip resistance changes about 2-3% when a magnetic field is applied. Typically, four of these resistive strips are connected in a Wheatstone bridge configuration so that both magnitude and direction of a field along a single axis can be measured. The key benefit of AMR sensors is that they can be bulk manufactured on silicon wafers and mounted in commercial integrated circuit packages.

Figure 1.7 shows a typical change of Earth magnetic field along one axis when a vehicle passes over the AMR sesnor. In Fig. 1.7, the sampling frequency is 64Hz and the waveform magnitude is the A/D readout.

The problem associated with magenometer vehicle detection is similiar to the acoustic sensors but the magnetic signals are much cleaner than acoustic signals.

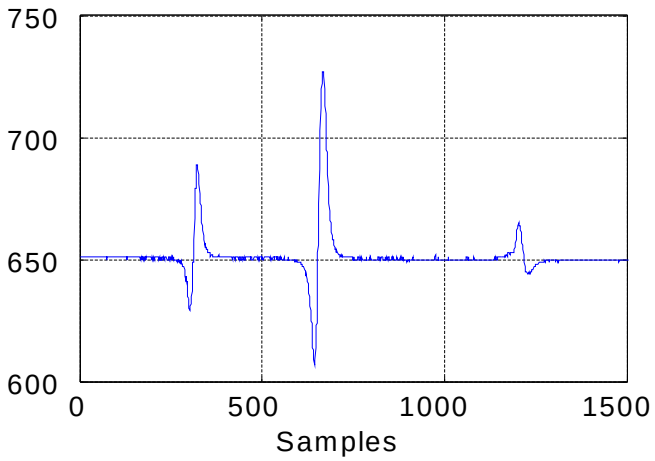

Figure 1.7: Waveforms of Magnetic Signals

1.3 TinyOS [1]-Software Platform

TinyOS [1] is an open-source operating system developed by the EECS department at UC Berkeley, which was designed for the Smart-Dust [2] hardware platform. It has a component-based runtime environment designed to provide support for deeply embedded systems that require concurrency intensive operations while constrained by minimal hardware resources.

The software architecture [7] is divided into a collection of software components. A complete system configuration consists of a tiny scheduler and a graph of these components. A component has four interrelated parts: a set of command handlers, a set of event handlers, an encapsulated fixed-size frame and a bundle of simple tasks.

Tasks, commands, and handlers execute in the context of the frame and operate on its state. To facilitate modularity, each component also declares the commands it uses and the events it signals. These declarations are used to compose the modular components in a per-application configuration. The composition process creates layers of components where higher level components issue commands to lower level components and lower level components signal events to the higher level components. Physical hardware represents the lowest level of components.

With this modular components architecture, the operating system is allowed to efficiently share a single

execution context across multiple components. For example, the same top level program codes could be applied to different sensors by linking them to lower level codes of the corresponding sensors. This allows a high level of flexibility for development on both hardware and software.

1.4 Sensor Network

The vehicle sensor network [8] is formed by wireless by interconnecting the smart-dust sensor nodes. The sensor network can gather signals from multiple points from the same lane or multiple lanes. Thus, traffic speeds , travel times and other traffic parameters can be estimated from the signals coming from the sensor network. The sensor network also povides redundancy for the reliability of the whole vehicle detection system.

1.5 Outline of this report

Next chapter will discuss the sensor operating principle in more details. The characteristics of the sensor signals is presented in chapter 3. Chapter 4 proposes the vehicle detection algorithms with the real time test and simulation results. Chapter 5 discusses the FPGA implementation of the vehicle detection algorithms for power efficiency. Conclusions are given in Chapter 6.

Chapter 2

Sensors for Vehicle Detection

This chapter will address the acoustic and magnetic sensors for vehicle detection in more details. Next, we discuss the pickup patterns and frequency responses of the acoustic sensors and their effects on vehicle detection followed by the basic principle of magnometers for the vehicle detection.

2.1 Acoustic Sensor

The acoustic sensor picks up the acoustic information by sensing the sound pressure change which is discussed in Chapter 1. Acoustic sensors are divided into two types based on their pickup patterns: unidirectional microphones and omnidirectional microphones.

2.1.1 Pickup Pattern of Acoustic Sensors

An acoustic sensor's pickup pattern is three dimensional in character and shows how the microphone responds, in frequency and level, to sound from different directions. Omnidirectional microphones pick up sound from all directions. Unidirectional microphones reject or reduce sound from their sides and rear. The left part of Fig. 2.1 shows the pickup pattern for an omnidirectional microphone. Notice that the loss in output (in dB) experienced as a constant when the sound source moves 360 degrees around a fixed microphone at a fixed distance. The right part of Fig. 2.1 shows the pickup pattern for a unidirectional

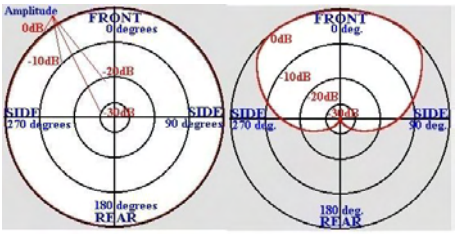

Figure 2.1: Microphones

microphone. The most common unidirectional is called a cardioid. Cardioid is a mathematically descriptive term that denotes the geometric form of the pickup pattern. In the cardioid pattern, side pickup is moderately reduced in a cardioid microphone and rear pickup is dramatically reduced. For the vehicle detection application, the unidirectional microphones may be better than omnidirectional ones since the unidirectional microphones have better rejection of sound from nearby vehicles.

2.1.2 Frequency Response of Panasonic Acoustic Sensors

The Panasonic WM-62A omnimicrophones(see Fig. 1.5) used in the Mote sensor board(Fig. 1.3) have features such as small size, high resistance to vibration, and pins for flexible PCB. Figure 2.2 shows the typical frequency response of the panasonic WM-62A acoustic sensors. It is noted that the frequency response is flat over large frequency range. For the vehicle detection application, filtering may be necessary to reject the uninterested frequency components.

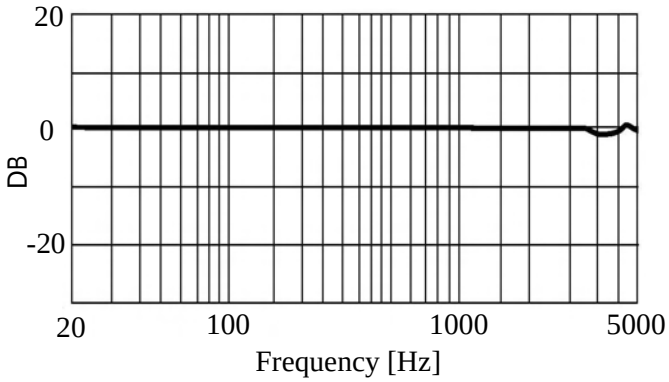

Figure 2.2: Microphones

2.2 Magnetometers

2.2.1 Basic Principle

The magnetization vector (M) in the Permalloy thin film resistors(Fig. 2.3) is set parallel to the length of the resistors. Assume that there is a current in the film flowing at a 45 degree angle to the length of the film. If an external magnetic field is applied normal to the side of the film , the Magnetization vector will rotate and change the angle θ. This causes the resistance value to vary and produce a voltage output change in the Wheatstone bridge which is formed by configuring four thin film resistors. This magnetoresistive effect is used to sense the earth magnetic field.

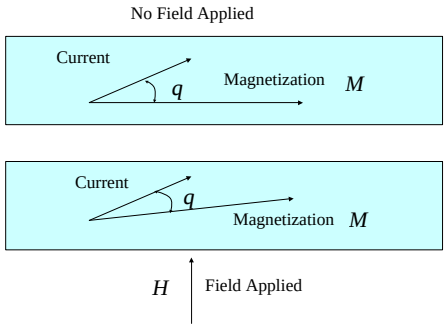

Figure 2.3: Magenetic Vector

2.2.2 Vehicle Detection

The magnetometers available today can sense magnetic fields within the earth's field-below 1 gauss. They can be used for detecting the vehicles, which are ferrous objects that disturb the earth's field. The earth's field provides a uniform magnetic field over wide area in the scale of kilometers and a car, a ferrous object, can creates a local disturbance in this field. This local field disturbance can be sensed by the magnetometers for vehicle detection. Figure 2.4 shows that the Earth's magnetic field is disturbed by a car.

After presenting the basic principles of sensors, the characteristics of the measured acoustic and magnetic signals will be studied and algorithms are proposed for reliable low cost vehicle detection.

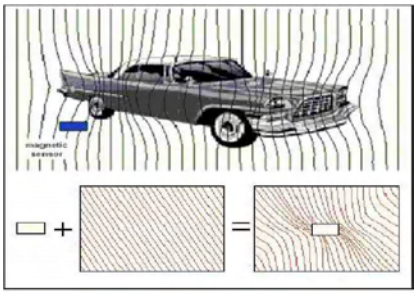

Figure 2.4: Earth Field with a Car

Chapter 3

Characterization of Sensor Signals

This chapter will address the characterization of sensor signals. First, the acoustic signals emitted from vehicles will be studied by analyzing the spectrum using Short Time Fast Fouier Transform(SFFT). We study the acoustic signals from background noise, stationary vehicles and moving vehicles at various vehicle speeds. Next, the magnetic signals from magnetometers will also be investigated. We study magnetic signals with different sensor orientation and sensor locations.

3.1 Acoustic Signals

Short-Time Fourier Transform is a way to have an estimate of the signal spectrum in a short interval. SFFT is utilized here since the measured vehicle acoustic signals are non-stationary. Figure 3.1 shows the moving window idea used in SFFT. In the discrete domain, the N-point short time FFT is defined as follows:

$$X(n,k) = \sum_{m} x(n)w(n-m)e^{-j2\pi mk/N} \tag{3.1}$$

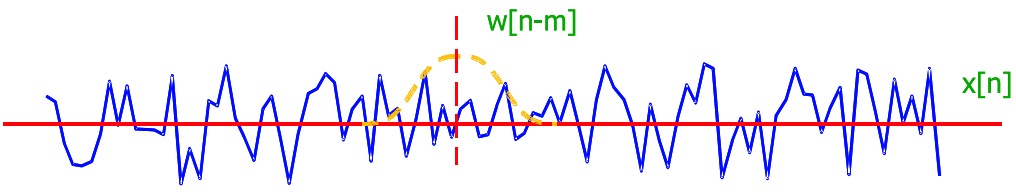

Figure 3.1: Short time fft schematic

where $x(n)$ is the signal for analysis and $w(n)$ is the window function. Popular window functions include rectangular, hamming window and etc. SFFT can be interepted as a sequence of discrete time Fourier Transforms(DFT) as the window $w[n-m]$ slides along the signals [9]. In the following, the Hamming window is chosen for the acoustic signal SFFT analysis.

3.1.1 Field Test Setup

Figure 3.2 shows the test setup for the vehicle acoustic signal measurement. In Fig. 3.2, the acoustic sensor is an omni-microphone(RadioShack $33-3025A$) and the microphone output is connected to the MIC input of a laptop computer. The measured acoustic signal is digitalized with sampling rate $11kHz$ and 8 bit resolution.

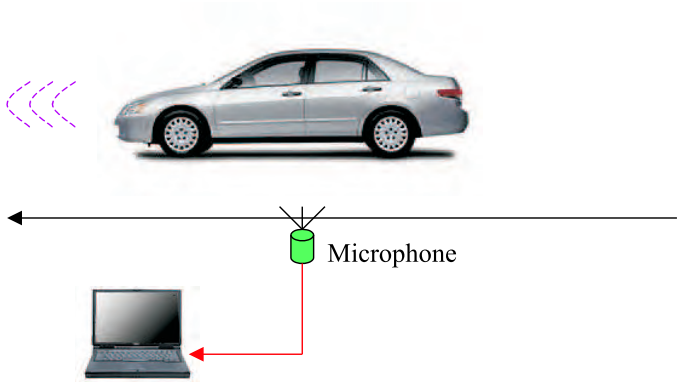

Figure 3.2: setup

3.1.2 Field Test Results

Acoustic Signals of Vehicle Engine

In this section, the field test results are presented in both time domain waveforms and the corresponding SFFT's. The time domain waveforms are the normalized sound pressure. The field testing was done in Richmond Field Station and the testing vehicles are Ford Van and Mazada LX. First, Fig. 3.3 shows the background acoustic signals. Notice that the acoustic energy in the background is highly dependent on the environmental windage and the windage energy mainly concentrates between DC and $500Hz$.

Next, the engine acoustic signals were measured by turning on the engine but keeping vehicles at station-

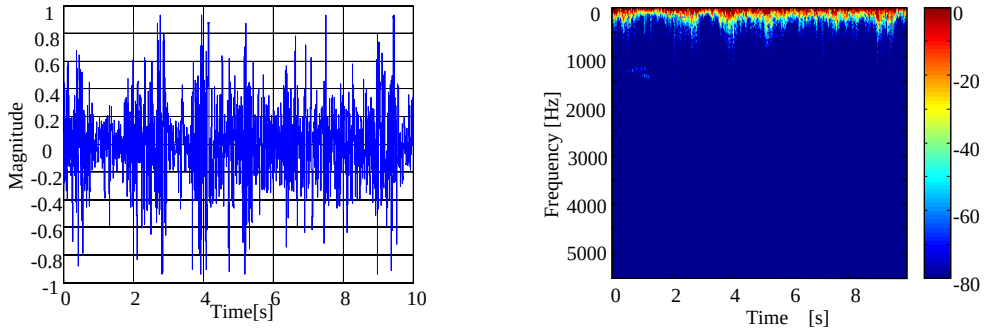

Figure 3.3: Background Acoustic Signals in Time and Frequency Domain

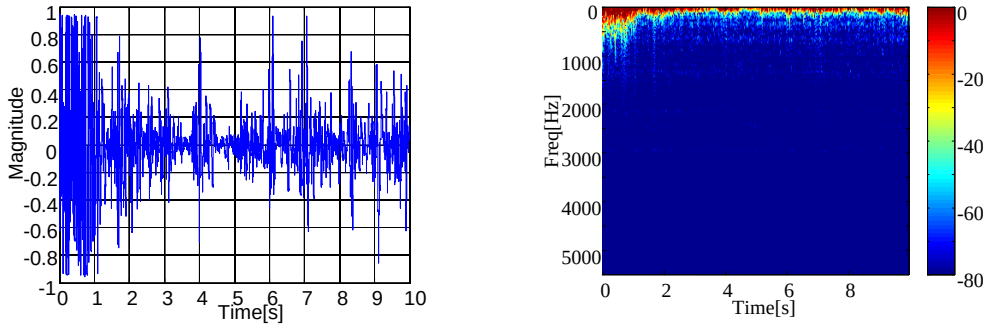

Figure 3.4: Acoustic Signals of Mazda 626 Engine in Time and Frequency Domain

ary. Figures 3.4 and 3.5 show the engine acoustic signals emitted from Mazda 626 and Ford WindStrar respectively by placing the microphone under the front bumper. Compared to the background acoustic spectrum, the engine acoustic singals have harmonics above 500Hz, which may be coming from engine cranking.

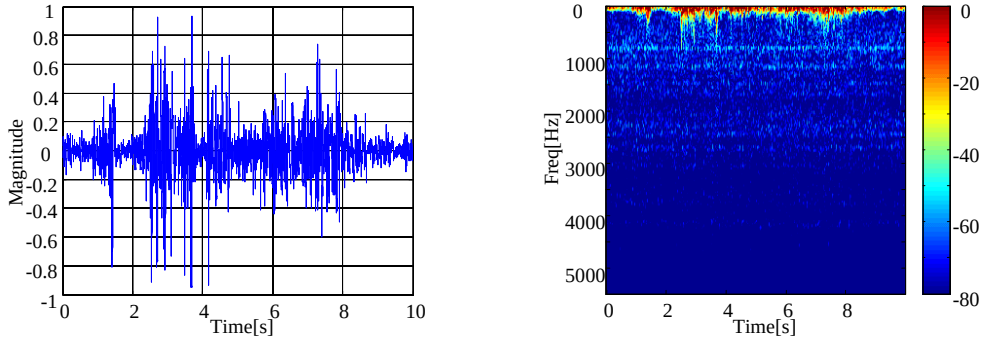

Figure 3.5: Acoustic Signals of Ford WindStar Engine in Time and Frequency Domain

Figure 3.6 shows the engine acoustic signals emitted by Mazda 626 in time and frequency domain by placing the acoustic sensor close to the engine exhuast.

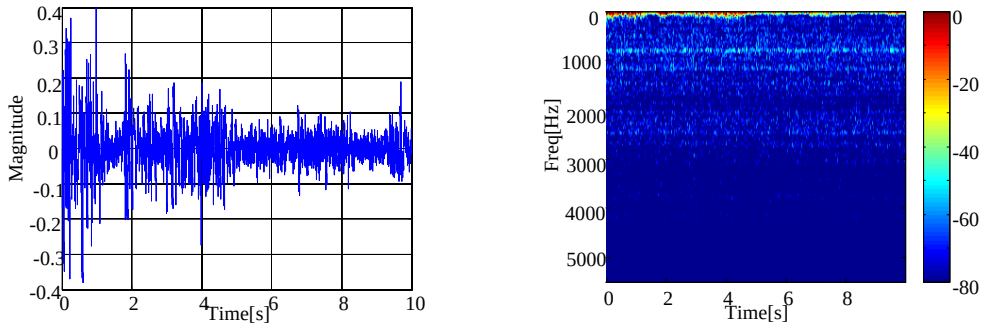

Figure 3.6: Acoustic Signals of Mazda 626 Engine in Time and Frequency Domain

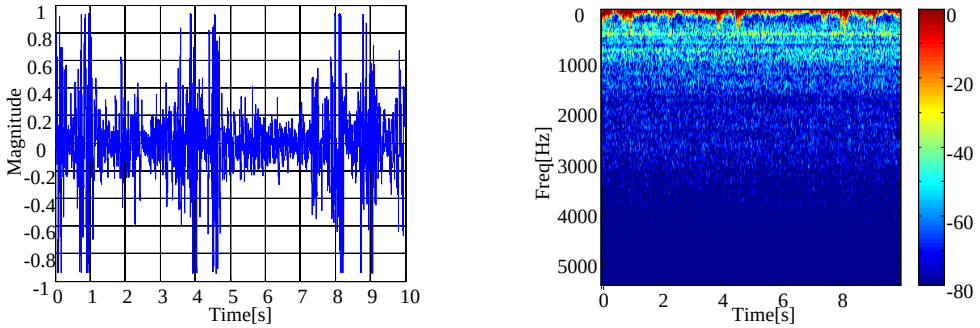

Figure 3.7: Acoustic Signals of Mazda Engine with Fan on in Time and Frequency Domain

Figure 3.7 further shows the engine acoustic signals for Mazda 626 when the engine fan was on. It is noted that the fan increases the acoustic energy between $500Hz$ and $1000Hz$.

Acoustic Signals for Moving Vechiles

Finally, the acoustic signals were measured for slow and fast moving vehicles. Figure 3.8 shows the acoustic signals from Mazda 626 when it was running at about $5mph$. Notice that the time domain waveform is serverly smeared by strong wind disturbance. Figure 3.9 shows the acoustic signal measured from Mazda 626 when it was running at about $15mph$ and Fig .3.10 shows the acoustic signal from Mazda 626 running at about 25mph. Noticably, the higher the speed is , the more temporally concentrated the acoustic energy is.

Figure 3.11 shows the acoustic signals emitted from multiple cars. The acoustic signals were recorded at the cross between Euclid street and Hearst street at Berkeley. It is noted that there is temporal acoustic energy concentration corresponding to each vehicle passing by the microphone.

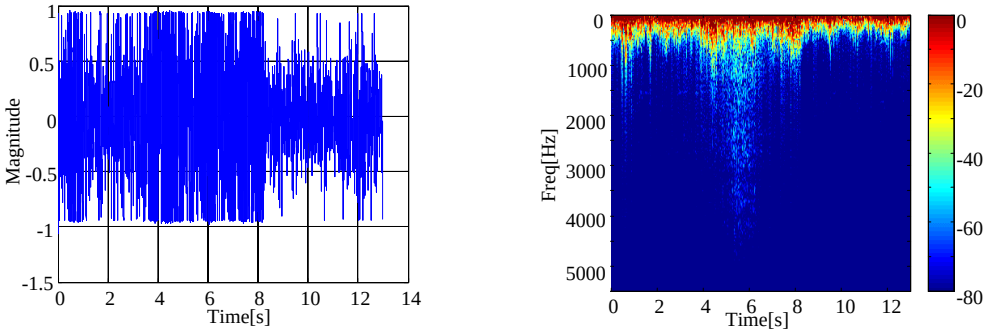

Figure 3.8: Acoustic Signals of Slow Moving Mazda 626 in Time and Frequency Domain

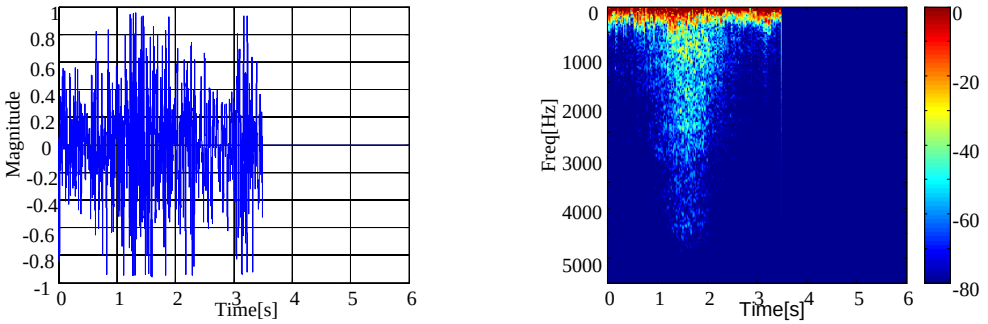

Figure 3.9: Acoustic Signals of Fast Moving Mazda 626 in Time and Frequency Domain

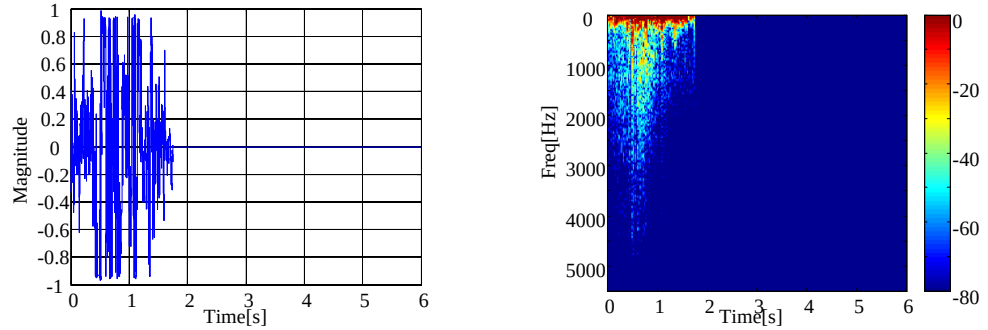

Figure 3.10: Acoustic Signals of Fast Moving Mazda 626 in Time and Frequency Domain

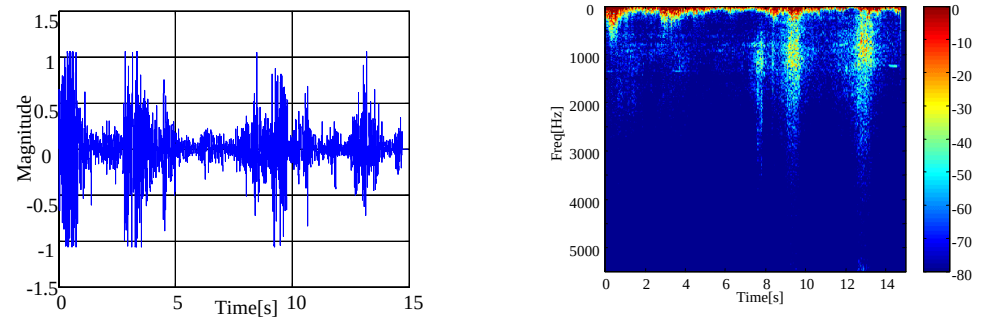

Figure 3.11: Acoustic Signals of Multiple Cars in Time and Frequency Domain

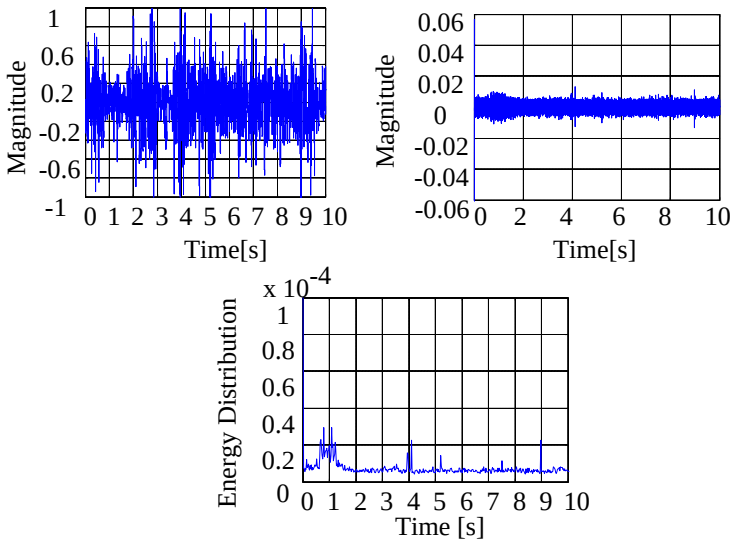

Figure 3.12: Background Acoustic Signals with Bandpass Filtering

3.2 Bandpass of Acoustic Signals

According to the SFFT analysis in last section, we found that the background acoustic and noise are mostly found in the frequency domain below 500Hz. In this section, we would present the analysis of the vehicle acoustic after band-pass filtering with passband from 450Hz to 5000Hz.

Figure 3.12 shows the background acoustic signal after band-pass filtering. Notice that its energy distribution remains at a low level after the band pass filtering.

Figure 3.13 shows the acoustic signal of the engine after band pass filtering. Notice the energy distribution (the square of the bandpass filtered acoustic signal) is pretty flat but the magnitude is larger than that of the background acoustic signal after band pass filtering.

Figure 3.14 shows the acoustic signal of a single slow moving vehicle which passed the microphone at about 4s. The magnitude of the wind disturbance is much larger than that of the vehicle acoustic signal in the raw signal. However, the wind disturbance is attenuated significantly by the band-pass filter. And the energy concentration for the vehicle is visible in the energy distribution curve plot.

Figure 3.15 show the acoustic signal of a single fast moving vehicle after band pass filtering. Notice that

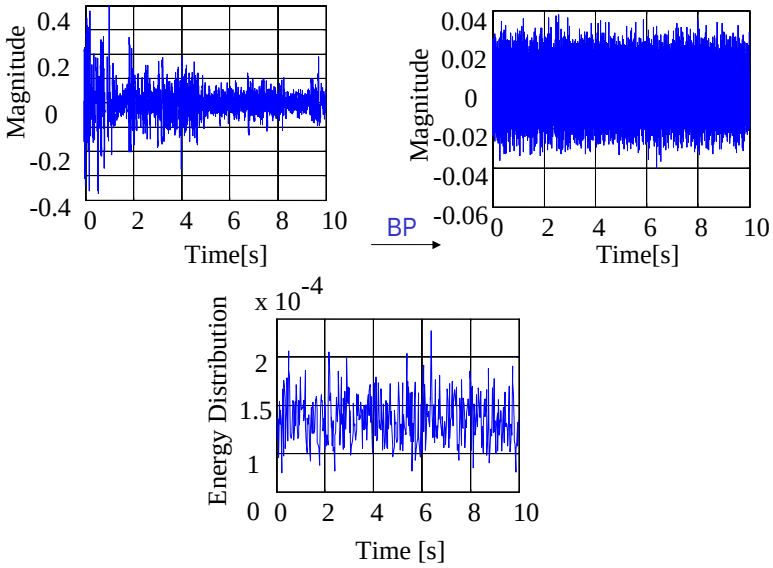

Figure 3.13: Engine Acoustic Signals with Bandpass Filtering (stationary vehicle)

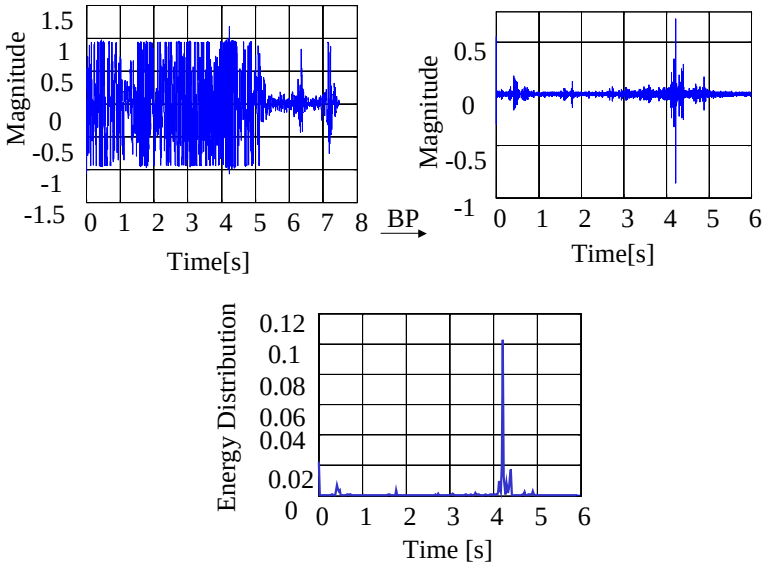

Figure 3.14: Acoustic Signals of Moving Mazda with Bandpass Filtering

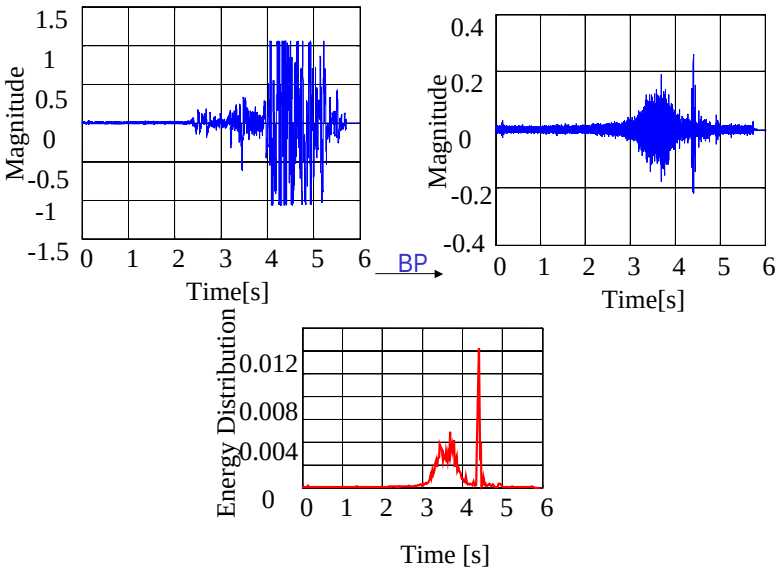

Figure 3.15: Acoustic Signals of Moving Mazda with Bandpass Filtering

the wind disturbance is almost completely rejected by the band pass filter.

3.3 Magnetic Signals

In this section, the magnetic measurements are presented from the magnetometer HMC1002 on the MICA mote. Figure 3.16 shows the experimental setup for magnetic sensor test.

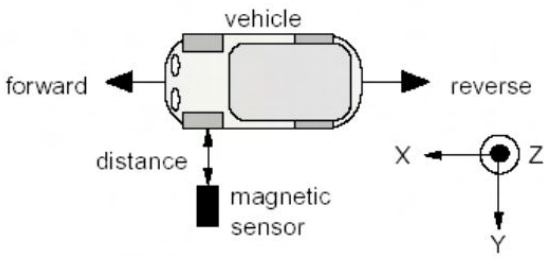

Figure 3.16: Schematic for the Magnetic Sensor Setup

Refer to the Fig. 3.16, the measurements of Z-axis and Y-axis (if the sensor node is placed on the side of road) would be a better choice as they simply give single hill patterns when vehicle pass by. Thus, we would

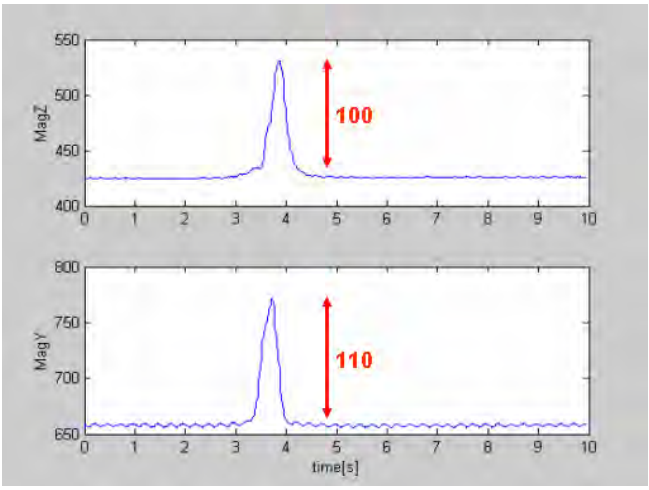

Figure 3.17: A single vehicle moving from left to right (x- to x+) on the near lane

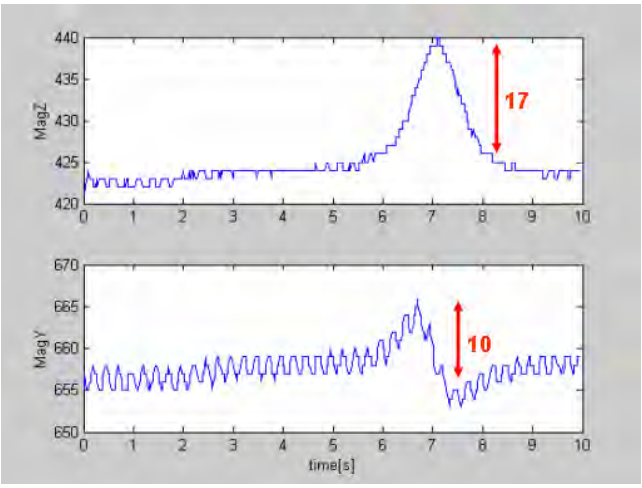

Figure 3.18: A single vehicle moving from left to right (x- to x+) on the far lane

focus our analysis on Z-axis and Y-axis measurements in the following section. All the measurements in this section were taken at 64 Hz for each axis, from the magnetometer HMC1002 on MICA mote, placed on the SIDE of a two-way traffic road.

Figure 3.17 shows the magnetic signal measurement when a single vehicle moving from left to right (x- to x+) on the a lane close to the magetometer. Notice that there is a sharp pulse in the measured signal.

Figure 3.18 shows the magnetic signal measurement when a single vehicle moving from left to right (x- to x+) on the opposite lane . Notice that there is a sharp pulse in the measured signal. Compared to Fig. 3.17, we could find a significant difference between the amplitude of the hill patterns for vehicle moving on different lanes. Applying a simple threshold cut, we could detect vehicles moving on the near lane while dropping vehicles on the far lanes.

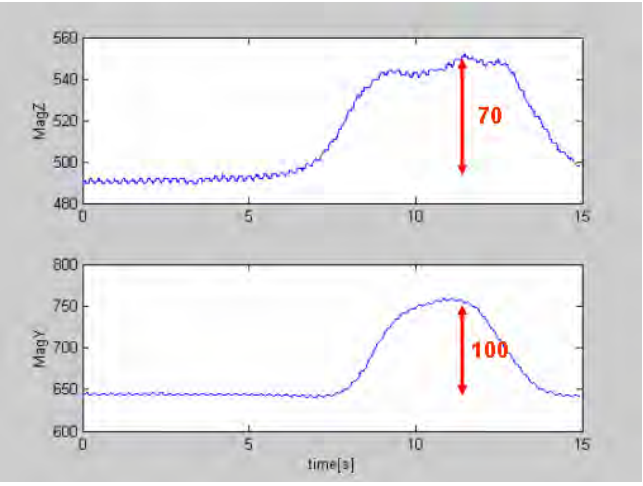

Figure 3.19: A single vehicle that stop-and-go in front of a stop sign, moving from left to right on the near lane

Figure 3.19 shows a single vehicle that stop-and-go in front of a stop sign, moving from left to right on the near lane. In this case, the single hill pattern has a flatten top. And a simple threshold cut would still be working well.

3.4 Summary

This chapter first presented the characteristics of the acoustic signals emitted from vehicles. It was shown that that the background noise is mainly concentrated between DC and 500Hz by the short time Fourier analysis of the acoustic signals. This chapter then presented the characteristics of magnetic signals for vehicle detection. Compared to the acoustic signals, the magnetic signals are much cleaner and a simple threshold may give fairly good detection.

Chapter 4

Algorithm

This chapter presents the acoustic and magnetic vehicle detection algorithms. Two acoustic algorithms are proposed: Adaptive Threshold algorithm (ATA) and Min-max algorithm (MMA). These detection algorithms are both based on the acoustic engergy temporal concentration in the measured acoustic signals. The ATA detects vehicles by searching sequences of 1's after adaptively thresholding the energy distribution curve while the MMA detects vehicles by searching the local maximum points of the acoustic energy distribution curve. The magnetic vehicle detection algorithm is just a simple threshold slicing algorithm[6].

4.1 Adaptive Thresholding Detection

This section presents the adaptive threshold acoustic detection algorithm. The adaptive acoustic detection algorithm consists of energy distribution curve computation, energy signal filtering, state machine detector and threshold adaptation. The block diagram of the adaptive threshold detection algorithm is shown in Fig. 4.1. In Fig. 4.1, Square&Decimator, FIR Filtering, Adaptive Threshold and Decision correspond to energy distribution curve computation, energy signal filtering, threshold adaptation and state machine detector, respectively.

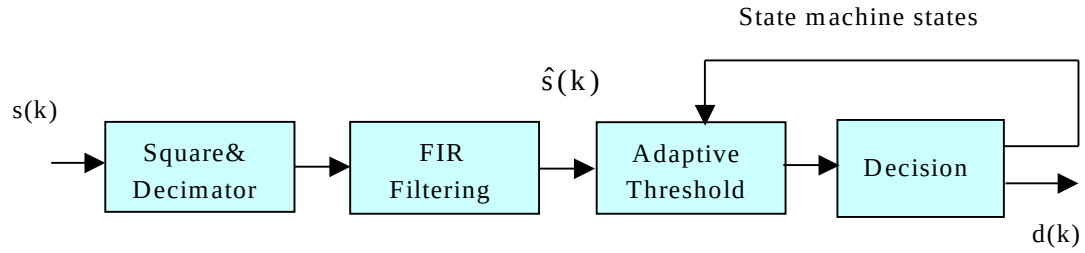

Figure 4.1: Block Diagram of Algorithm

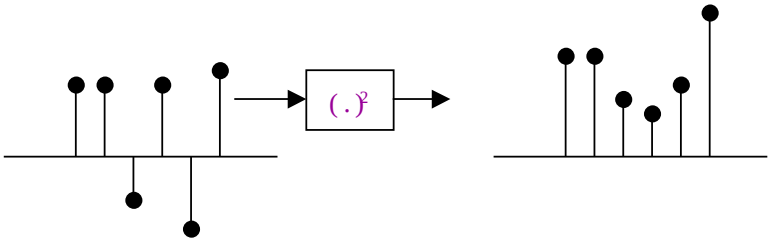

Figure 4.2: Signal Squaring

4.1.1 Energy Distribution Computation

The original measured acoustic signal is first filtered through a band pass filter and the filtering output s(k) is squared (Fig. 4.2) at each sampling point, which is energy distribution signal. In order to reducing the smooth filtering computation, this energy distribution signal may be decimated before passing to smoothing filter (the FIR Filtering block in Fig. 4.1).

4.1.2 Smoothing of Acoustic Engergy Signal

The acoustic energy signal is very jerky and a low pass FIR filter is used to smooth it for later detection. Figure 4.3 shows a typical equal-ripple low pass FIR frequency response. The key parameters for a low pass FIR filter are the -3dB cut off frequency (ω_p), the stop band frequency (ω_s) and the stop band attenuation gain. The FIR filter has the advantage of linear phase and inherent stability. The filtered acoustic energy signal ($\hat{s}(k)$) can be passed to the Adaptive Threshold in Fig. 4.1 for hard decision.

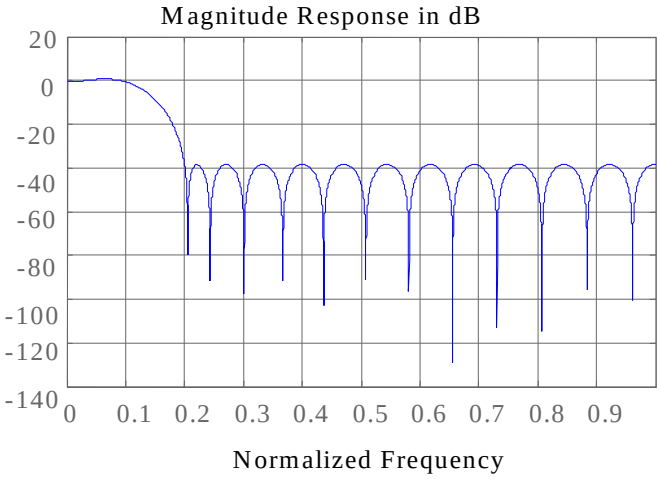

Figure 4.3: FIR Filtering

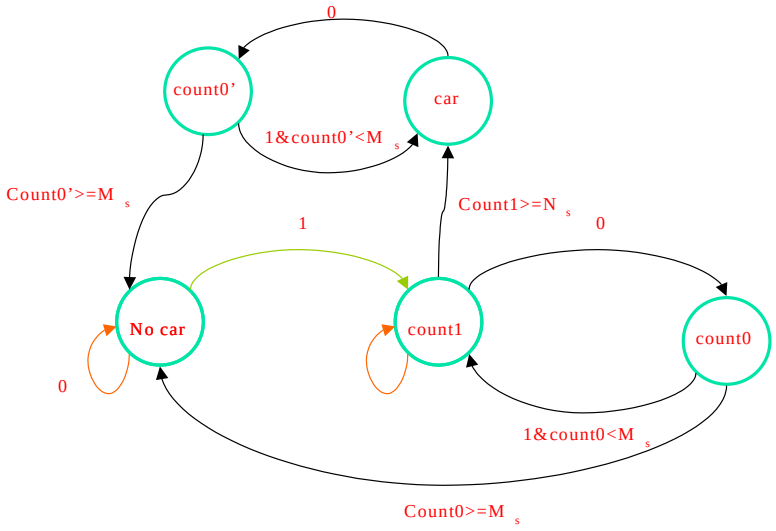

Figure 4.4: Detection State Machine for ATA

4.2 Adaptive Thresholding Decision

Hard decision produces an output (u(k)) 1 if the input sample $\hat{s}(k)$ is larger than detection threshold $T(k)$. Otherwise, hard decision will produce 0. The threshold $T(k)$ is adaptively updated, which will be addressed next.

$$MA(k) = \frac{\hat{s}(k) + \hat{s}(k-1) + ... + \hat{s}(k-M+1)}{M} \tag{4.1}$$

where MA denotes the moving average of the acoustic energy and M is the number of moving average. Then the adaptive threshold $T(k)$ is updated as follows:

if current decision is 1

$$T(k) = \alpha MA(k - M_d) + T_{offset}$$

else

$$T(k) = \beta MA(k - M_d) + T_{offset}$$

where α and β are two parameters for adjusting the moving average(MA), M_d is an integer for delaying the moving average and T_{offset} is a constant which sets the minimum threshold.

4.2.1 State Machine Detection

Figure 4.4 shows the block diagram of state machine for vehicle detection. The state machine consists of :

$$state(x) : \{nocar, car, count1, count0, count0'\} \tag{4.2}$$

$$input(u) : \{1, 0\} \tag{4.3}$$

$$output(d) : \{car, nocar\} \tag{4.4}$$

The input in the state machine is defined as:

$$\begin{aligned} u(k) &= 1 \text{ if } \hat{s}(k) \geq T(k) \\ &= 0 \quad \text{otherwise} \end{aligned} \tag{4.5}$$

There is a counter for each state in $\{count1, count0, count0'\}$ and the counter at each state resets whenever the state machine jumps back from other states to itself.

The state machine starts at state *no car* and stays at this state if the input u(k) is 0. The state machine jumps from state *no car* to state $count1$ if the input is $1(u(k) = 1)$. When the state machine enters state $count1$, the counter counts up and the state machine stays at this state if the input $u(k)$ is 1 and the previous counter value is less than N_s. The state machine jumps from $count1$ to $count0$ if the input is 0 and to *car* if the input is 1 and the previous counter value is not less than N_s. When the state machine enters state $count0$, the counter at this state counts up and the state machine stays at this state if the input u(k) is 0 and the previous counter value is less than M_s. The state machine jumps from $count0$ to $count1$ if the input u(k) is 1 and to state *no car* if the input is zero and the previous counter value is not less than M_s. When the state machine enters state *car*, it will stays at this state if the input is 1 and jumps to $count0'$ if the input is 0. When the state machine enters state $count0'$, the counter at this state counts up and the state machine stays at this state if the input is 0 and the previous counter value is less than M_s. The state machine jumps from $count0'$ to count *car* if the input is 1 and to *no car* if the input is zero and the previous counter value is not less than M_s. One vehicle is detected when the state machine jumps from state $count1$ to state *car*.

It is noted that the counter at the states $\{count1, count0, count0'\}$ and parameters M_s and N_s introduce hysteresis in the detection, which will make the algorithm more robust to the short burst errors in the hard decision.

4.3 Min-Max Detection

Figure 4.5 shows the block diagram for Min-Max algorithm. The Squaure&Decimator and FIR Filtering blocks are the same as in adaptive threshold detection algorithm. Figure 4.6 shows the state machine used in Min-Max Detection algorithm. The state machine consists of

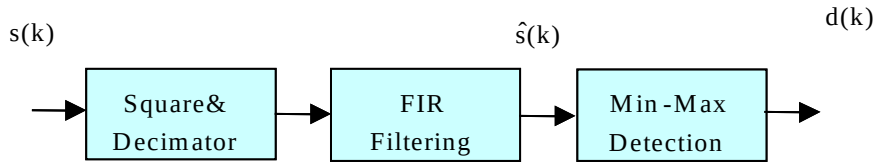

Figure 4.5: Block Diagram of Algorithm

$$state(x) : \{flat, flat\_count\_up, flat\_count\_down, hill\_count\_up, hill\_count\_down\} \tag{4.6}$$

$$input(u) : \{1, 0\}$$

$$output(d) : \{car, nocar\}$$

The input for the decision state machine is the sign of the slope of $\hat{s}$, which is defined as:

$$u(k) = sign(\hat{s}(k) - \hat{s}(k-1)) \text{ if } |(\hat{s}(k) - \hat{s}(k-1)| > min\_delta\_U \tag{4.7}$$

$$= 0 \; otherwise \tag{4.8}$$

where min_delta_U is a pre-defined positive constant.

There is a counter associated with each state in { $Flat\_count\_up$, $Flat\_count\_down$, $Hill\_count\_up$, $Hill\_count\_down$}. When the machine jumps from one state to a new state, the counter associated with the new state resets and only counts up when the state loops back to itself. There are also two variables associated with the state machine: $local\_min$ and $local\_max$. The $local\_min$ is updated as following:

$$local\_min(k) = \tag{4.9}$$

$$min\{\hat{s}(k), local\_min(k-1)\}, \text{ if } x \in \{Flat, Flat\_count\_down\} \tag{4.10}$$

$$\hat{s}(k), \text{ if } x = Hill\_count\_down \text{ and } local\_max(k) - local\_min(k) > Threshold$$

$$unchanged, \; otherwise$$

and the $local\_max$ is updated as following:

$$local\_max(k) = \tag{4.11}$$

$$max\{\hat{s}(k), local\_max(k-1)\},\ \text{if } x = Hill\_count\_up\} \tag{4.12}$$

$$unchanged,\ otherwise$$

The local minimum ($local\_min$) tracks the local minimums in real time while the local maximum($local\_max$) tracks the local maximums. One vehicle is detected if the difference between the local maximum and the local minimum is greater than the threshold ($Threshold$) when the state machine jumps from $Hill\_count\_down$ to $Flat$.

The detection state machine starts at $Flat$ state and stays at this state if the slope is not positive ($u(k) \leq 1$). The state machine can jumps from $Flat$ state to $Flat\_count\_up$ state when the current slope is positive ($u(k) = 1$). When the state machine enters $Flat\_count\_up$ state, the counter at this state resets. The state machine stays at $Flat\_count\_up$ state and the counter counts up if the slope is positive ($u(k) = 1$) and the counter has value less than N. The state machine jumps from $Flat\_count\_up$ state to $Flat\_count\_down$ if the the current slope is not positive ($u(k) \leq 1$. The state machine can jump from $Flat\_count\_up$ state to $Hill\_count\_up$ when the slope is positive ($u(k)$ =1) and the counter at $Flat\_count\_up$ has a value not less than N. When the state machine enters $Flat\_count\_down$, the counter at this state resets. The state machine stays at $Flat\_count\_down$ and ther counter counts up if the slope is not positive ($u(k) \leq 1$) and the counter has value less than M. The state machine will jumps from $Flat\_count\_down$ to $Flat\_count\_up$ if the slope is positive ($u(k) = 1$ and jumps back to $Flat$ if the slope is not positive and the counter has a value not less than M. When the state machine enters $Hill\_count\_up$, the state machine stays at this state if the slope is not negative and jumps to $Hill\_count\_down$ if the slope is negative. When the state machine enters $Hill\_count\_down$, the counter at this state resets. The state machine stays at this state and the counter counts up if the slope is negative and the counter has value less than M. The state machine will jump back to $Hill\_count\_up$ if the slope is not negative. The state machine will jump back to $Flat$ state if the slope is negative and the counter has a value not less than M. At state $Hill\_count\_down$, the

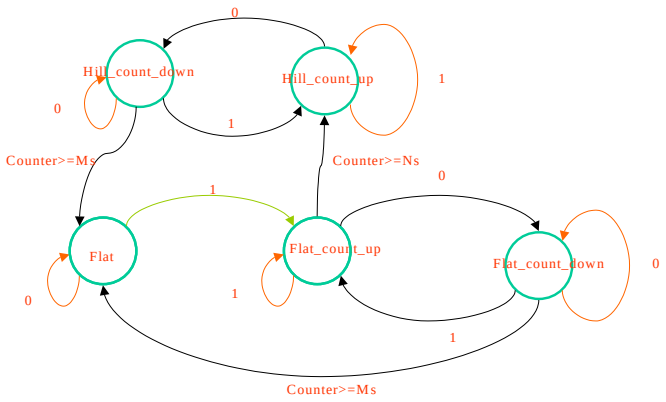

Figure 4.6: Detection State Machine for MMA

difference between *local_max* and *local_min* is checked to determine if one vehicle is detected.

4.4 Algorithm Test

This section will demonstrate the two algorithms presented in previous sections. First, the algorithms are prototyped in a laptop based system as shown in Fig. 3.2. Real time tests and offline simulation results are both presented for this prototype system. Second, the offline algorithm simulation is discussed for the acoustic signals measured by the Mote system for the limited computing resource in the Mote system makes the real time test difficult.

4.4.1 Adaptive Threshold Algorithm

Figure 4.7 shows the use of adaptive threshold algorithm in real time vehicle detection. The decision with 1's at around 1.8, 4, 6.8, 8.2, 10, 12, and 14 second represent vehicle existence at those instants. The states 0,1,2,3 and 4 in the state transition traces are corresponding to states *no car*, *car*, *count*0, *count*1 and *count*0′ respectively. It is noted that the Adaptive Threshold Algorithm gives the correct real time detection. Figure 4.8 shows a long time ATA simulation results. In Fig. 4.8, the blue line corresponds to the energy distribution curve and the red line corresponds the threshold traces. Figure 4.9 shows the zoom

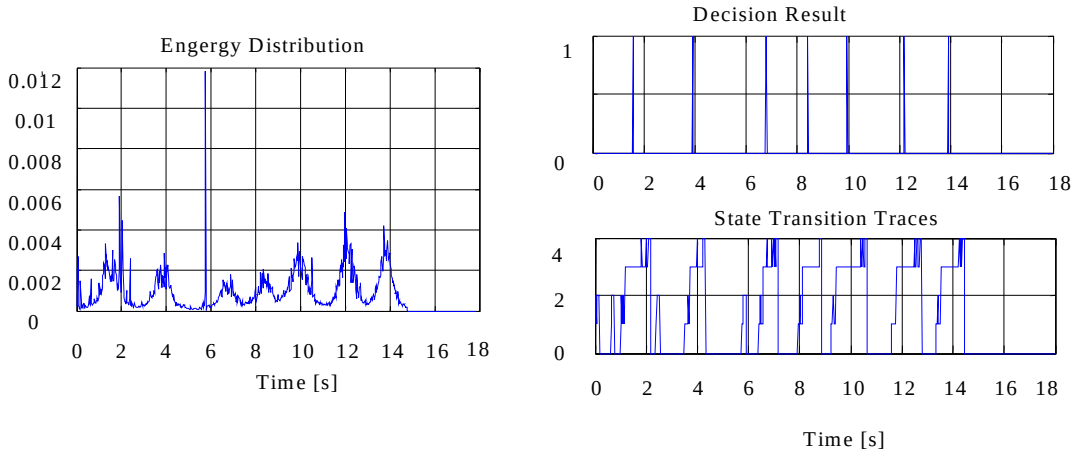

Figure 4.7: Adaptive Threshold Algorithm

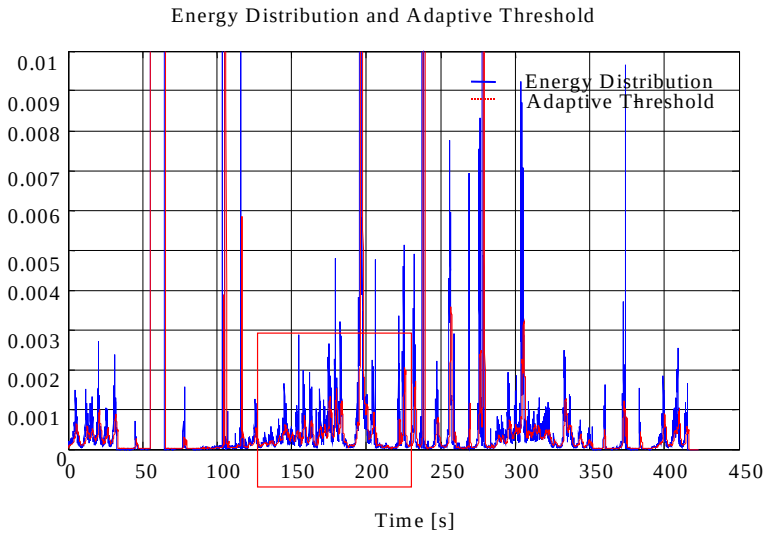

Figure 4.8: Long Time ATA Simulation

in around 160 second in Fig. 4.8.

Next we will study the effect of parameter choices on the performance of the algorithm.

Table 4.1 summarizes the effect of parameter M_s and N_s on the ATA algorithm performance with α=1.5, β=0.7 and threshold offset T_{offset}=2e-5. The smoothing filter is implemented as 40 (M in Eq. 4.1) point moving average and the delay M_d is chosen to be 20. It is noted that large M_s and N_s leads to better robustness but may miss detecting high speed vehicles. M_s and N_s can be chosen by trading off between the algorithm robustness and the speed range of detectable vehicles.

Table 4.2 summarizes the effect of α and β on the ATA performance with (N_s,M_s) = (10,10) and threshold

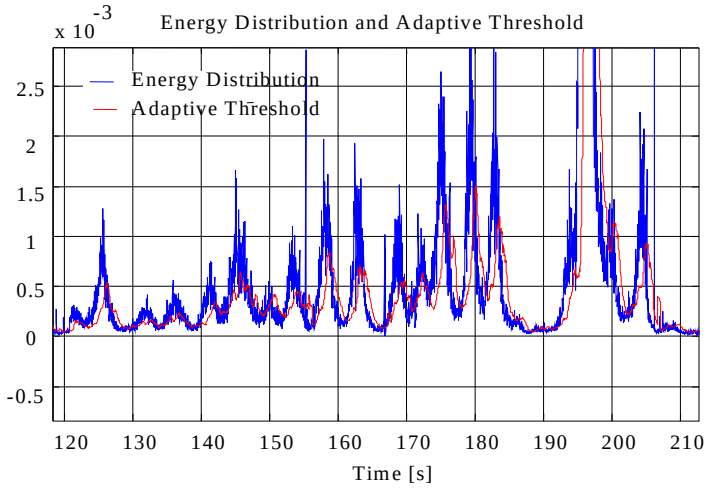

Figure 4.9: Zoom-in for the Long Time ATA Simulation

Table 4.1: M_s and N_s

(N_s,M_S)	Ground truth (# of vehicles)	Detection result (# of vehicles)
(10,6)	63	64
(10,10)	63	62
(10,15)	63	64
(10,20)	63	62
(20,10)	63	60
(15,10)	63	62
(10,10)	63	63
(6,10)	63	70

offset T_{offset} = 2e-5. The smoothing filter is implemented as the 40 (M in Eq. 4.1) point moving average and the delay M_d is chosen to be 20. It is noted that the performance is quite robust to the choices of α and β.

Table 4.3 summarizes the effect of T_{offset} on the ATA performance with (N_s,M_s) = (10,10) and (α,β) = (1.3, 07). The smoothing filter is implemented as the 40 (M in Eq. 4.1) point moving average and the delay M_d is chosen to be 20. It is noted that the T_{offset} is mainly determined by the noise level in the acoustic energy distribution. Zero offset will result in a lot of over-count and too large offset may lead to miss detecting quiet vehicles.

Table 4.4 shows the effect of smoothing filter on the ATA algorithm and Min-max algorithm performance. In Table 4.4, ω_p is the low pass FIR -3dB cutoff frequency and ω_s is the corresponding stop band frequency.

Table 4.2: α and β

(α,β)	Ground truth (# of vehicles)	Detection result (# of vehicles)
(1.8,0.7)	63	58
(1.5,0.7)	63	62
(1.3,0.7)	63	66
(1.5,0.8)	63	63
(1.5,1.0)	63	62

Table 4.3: T_{offset}

T_{offset}	Ground truth (# of vehicles)	Detection result (# of vehicles)
0	63	81
1e-5	63	66
2e-5	63	63
3e-5	63	60
4e-5	63	59
5e-5	63	57

In the Adaptive Threshold algorithm, $\alpha = 1.3$,β =0.7, $N_s = M_s = 10$ and threshold offset T_{offset}=3e-5. In the Min-max algorithm, M_s and N_s are the same as in the adaptive algorithm with minimum height (*Threshold*) 2e-6. It is noted that the MMA algorithm requires more smoothing than the ATA algorithm to avoid too much over-count since MMA algorithm is based on the energy curve slope.

Table 4.4: Filtering Smoothing on Detection Result

(ω_p, ω_s) (Normalized Freq.)	# of coeffs.	Ground truth (# of vehicles)	Adaptive (#of vehicles)	Min-max (# of vehicles)
40 point MA	40	$\tilde{6}3$	65	58
0.01-0.05Hz	67	$\tilde{6}3$	62	68
0.005-0.05Hz	56	$\tilde{6}3$	63	66
0.01-0.1Hz	27	$\tilde{6}3$	70	89

4.4.2 Min-max Algorithm

Figure 4.10 shows the real time detection using Min-max algorithm. The Hill Energy is the just the energy distribution curve used in ATA algorithm. The solid and dash line in the right of Fig.4.10 correspond to the trace of *local_min* and *local_max* respectively.

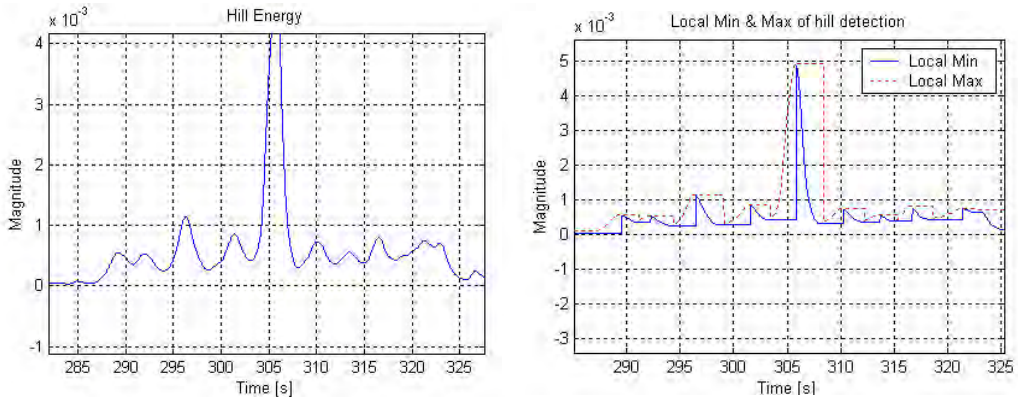

Figure 4.10: Min-max Algorithm

Next, the effect of parameters in the MMA algorithm will be studied.

Table 4.5: Filter Smoothing on the MMA Algorithm

Smoothing filter (cutoff freq, Hz)	Smoothing filter (Stop freq,Hz)	# of filter taps	Ground truth (# of vehicles)	Detection result (# of vehicles)
0.005	0.05	56	63	66
0.005	0.04	72	63	64
0.005	0.03	100	63	61

Table 4.5 shows the MMA detection results with different smoothing filters with Minimum height (*Threshold*) =2e-6 and (M_s, N_s) =(10,10). It is noted that smaller stop frequency ω_s results better smoothing which may cause miss detection while larger stop frequency ω_s may lead to over-count.

Table 4.6: Minimum Height

Minimum Height (Threshold)	Smoothing filter (Stop freq,Hz)	# of filter taps	Ground truth (of vehicles)	Detection result (of vehicles)
1e-7	0.005	56	63	65
5e-7	0.005	56	63	64
1e-6	0.005	56	63	64
2e-6	0.005	56	63	63
3e-6	0.005	56	63	60
5e-6	0.005	56	63	53

Table 4.6 shows the MMA detection results with different Minimum heights (*Threshold*) and (cut off freq., stop freq.) = (0.005,0.04) (M_s, N_s) =(10,10). It is noted that too large Minimum height may lead to miss

detection.

Table 4.7: M_s and N_s

Count_up /down (M_s,N_s)	Smoothing filter (Stop freq,Hz)	# of filter taps	Ground truth # of vehicles	Detection result # of vehicles
(1,1)	0.05	56	63	66
(5,5)	0.05	56	63	64
(10,10)	0.05	56	63	64
(15,15)	0.05	56	63	64

Table 4.7 shows the MMA detection results with different (M_s,N_s) with Minimum height ($Threshold$) =2e-6 and (cut off freq., stop freq.) =(0.005,0.04). It is noted that the MMA algorithm is pretty robust to (M_s, N_s).

4.5 Mote Acoustic Detection

The sampling frequency for the acoustic sensor in Mote system is only $256Hz$. The algorithm block diagram is the same as the laptop prototype system but the original acoustic signal is not band pass filtered. Since the Mote system has limited computing resources, the algorithm is not implemented in the Mote but is simulated offline with the measured acoustic signal by the Mote system. Figure 4.11 shows the Adaptive Threshold Algorithm and the Min-max Algorithm simulation results for the Mote system. In Fig. 4.11, the left top is the original acoustic waveform and the left bottom is energy distribution curve. It is noted that both ATA and MMA end up with correct detection.

More algorithm simulation for Mote measured acoustic signals is summarized in Table 4.8.

Table 4.8: M_s and N_s

Count_up /down (M_s,N_s)	Smoothing filter (Stop freq,Hz)	# of filter taps	Ground truth # of vehicles	Detection result # of vehicles
(1,1)	0.05	56	63	66
(5,5)	0.05	56	63	64
(10,10)	0.05	56	63	64
(15,15)	0.05	56	63	64

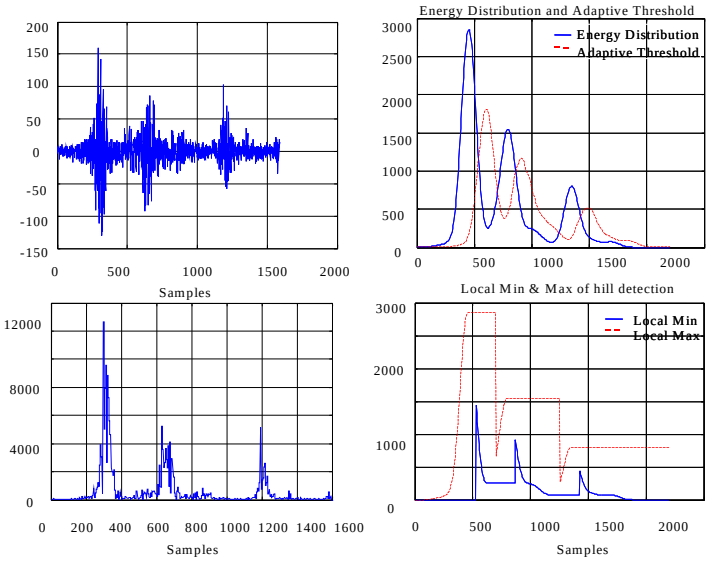

Figure 4.11: Mote Acoustic Detection

Table 4.9: Summary of Mote Detection

Data set	Ground truth # of vehicles	Adaptive Threshold # of vehicles	Min-max Threshold # of vehicles
#1	1	1	1
#2	3	3	4
#3	3	3	3
#4	4	4	4
#5	2	1	1
#6	1	2	2
#7	4	4	4

4.6 Magnetic Detection Algorithm

The threshold slicing algorithm had been implemented on the Smart-Dust [1] [2] MICA sensor mote. The real time testes had been done on Hearst outside Etcheverry with the sensor on the side of road. In this section, we would present the results of these vehicle detection testes.

Figures 4.12 and 4.13 show the results of 2 sample detection testes for 20s. The plots on the top are the raw signal of Z-axis and Y-axis sampling at 64Hz each. The plot on the bottom left shows the detection flag after a threshold cut for the Z and Y axis. And the one on the bottom right shows the final detection flag which is a logic operation of "Z" AND "Y" detection flags. The test in Fig. 4.12 gave a detection of 2 out of 2(ground truth) SUV (vehicles), while that in Fig. 4.13 gave another detection of 7 out of 7(ground

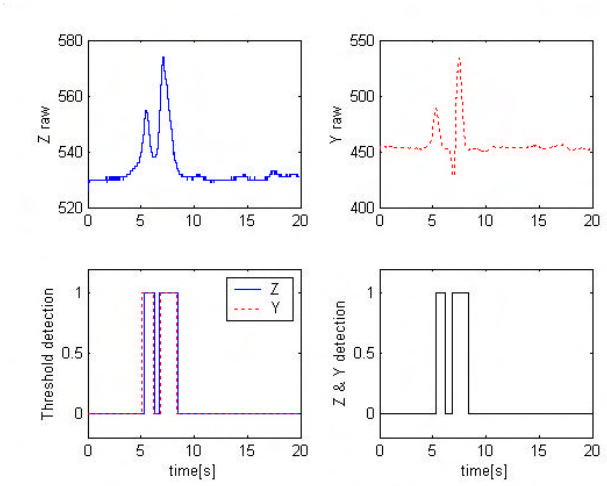

Figure 4.12: Magnetic Threshold Slicing Detection

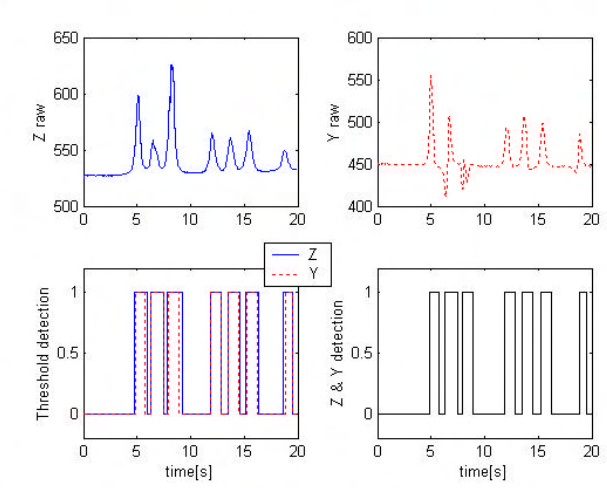

Figure 4.13: Magnetic Threshold Slicing Detection

truth) vehicles.

4.7 Summary

This Chapter discussed two acoustic vehicle detection algorithms: the Adaptive Threshold Algorithm and the Min-max Algorithm. Real time test and offline simulation demonstrated the effectiveness of these two algorithms prototyped in a laptop computer system. The effect of parameters on the two algorithms was also presented. The application of the two algorithms on the Mote measured acoustic signals was shown to be effective. Finally, the threshold slicing algorithm for magnetic vehicle detection was presented.

Chapter 5

FPGA Implementaton

5.1 Introduction

The current Tiny Mote has limited computing resources and the filtering based detection is very computation intensive. ASIC implementation is one possible solution. The feasibility of ASIC implementation can be verified by the Field Programmable Gate Array (FPGA) implementation of the algorithm. FPGA implementation presents the upbound for ASIC implementation in area and power consumption.

The rest of this chapter is organized as follows. Section 2 reviews the FPGA and System Generator for MATLAB. Section 3 presents the implementation of the components in the algorithm. Section 4 shows the algorithm simulation of the FPGA implementation. Conclusion is given section 5.

5.2 FPGA Introduction

FPGA is a 2-D array of logic blocks and flip-flops, where users can configure the interconnection between blocks and the function of each block. The following reviews a typical FPGA based design flow:

5.2.1 Design Entry

The first step in FPGA design flow is to conceptualize the design. In circuit logic design, Hardware Description Language (HDL) is widely used to represent the design. In DSP algorithm design, Simulink is one of very popular simulation tool. For fast FPGA based DSP design, Xilinx provides a dedicated Blocksets for Simulink, which can be used to simulate DSP algorithm and are mapped to HDL automatically.

5.2.2 Synthesis

Once the algorithm design is represented by Xilinx Blocksets, the next step is synthesis. The synthesis translates the HDLs generated from Xilinx Blocksets into an equivalent circuit comprising a set of primitive circuit components that can be directly implemented on an FPGA like a compiler. The synthesis tool produces a final netlist file, which contains a list of all the instances of primitive components in the translated circuit and a description of how they are interconnected.

5.2.3 Placement and Routing

The next step in the implementation flow turns the netlist into bits, which are used to configure the interconnections in FPGA. First, the primitive circuit components in the netlist are assigned specific places on the FPGA. Next, the proper connections must be routed according to the netlist description.

5.2.4 Program Hardware

The last step in the implementation flow is the simple act of transporting the configuration bits to the FPGA.

5.3 FPGA Implementation

Figure 5.1 shows the overall block diagram for the diction algorithm. Detailed description of the detection algorithm is given in chapter 4.

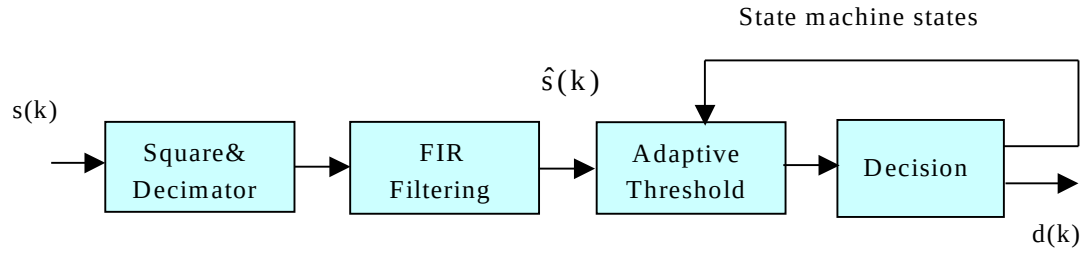

Figure 5.1: Block Diagram of Algorithm

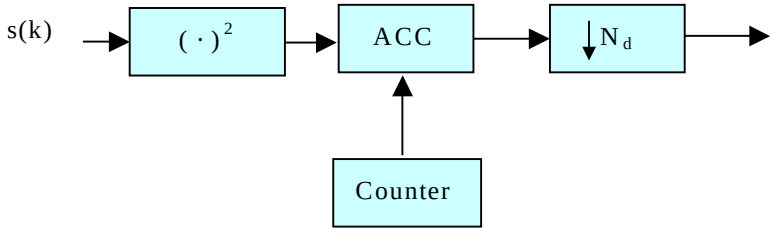

Figure 5.2: Squarer and Decimator

5.3.1 Square Decimator

The Square Decimator block consists of squaring, short length moving average and decimating elements. The detail structure of this block is given in Figure 5.2. In this Figure, ACC is the accumulator block and is reset by counter every N_d samples and the ACC output is also downsampled by N_d.

5.3.2 FIR Filter Implementation

Figure 5.3 shows the Direction form-II implementation of a FIR filter F(z)(Eq. 5.1).

$$F(z) = h_0 + h_1 z^{-1} + ... + h_{N-1} z^{-(N-1)} \tag{5.1}$$

where h_0, h_1, h_2,...,h_{N-1} are the filter taps. Noticing that the FIR filters have symmetric taps, i.e. h_0=h_{N-1}, h_1=h_{N-2}, and etc. Thus the F(z) can be rewritten as:

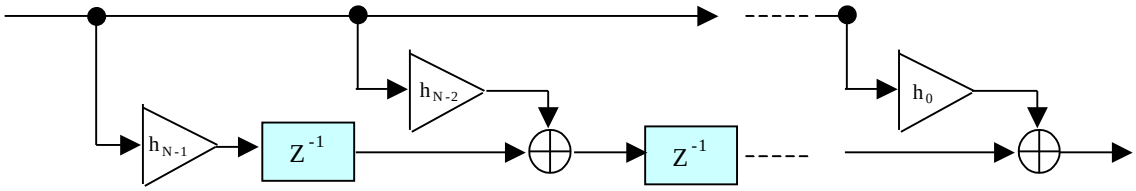

Figure 5.3: Fir filter implementation

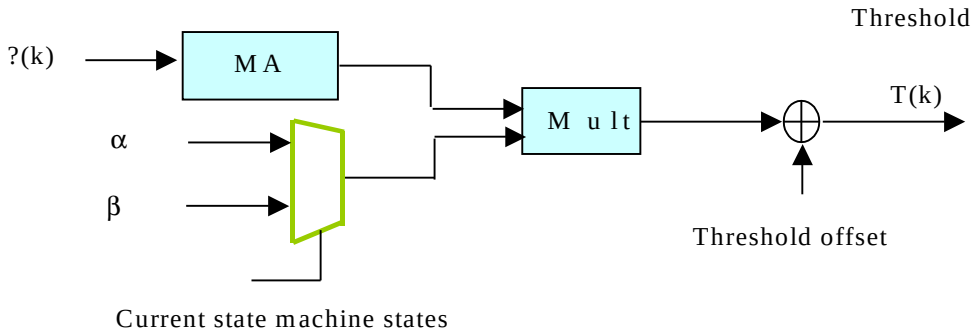

Figure 5.4: Adaptive Threshold Block

$$F(z) = h_0(1 + z^{-(N-1)}) + h_1(z^{-1} + z^{-(N-2)}) + ... \tag{5.2}$$

and the number of multiplication can be further reduced into half.

The advantage of this implementation is that critical path delay is determined by one multiplication and addition and is independent of the number of filter taps.

5.3.3 Adaptive Threshold Block

The detail structure of the Adpative Threshold block is shown in Figure 5.4.

In this figure, MA is long time moving average (¿20 samples average) and a, b and Threshold offset are programmed constants, which can be adjusted for best detection. In order to avoid a lot of additions, MA is actually implemented in an iterative way (Fig. 5.5). Moving Average Implementation The moving

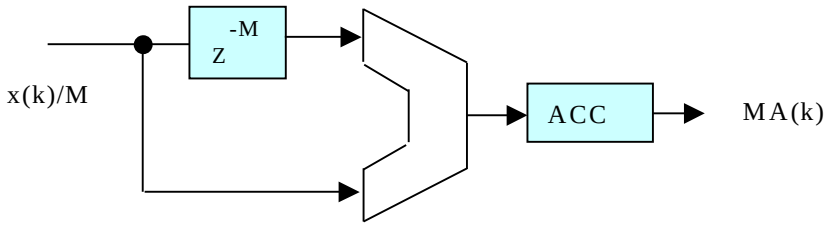

Figure 5.5: Iterative Moving Average

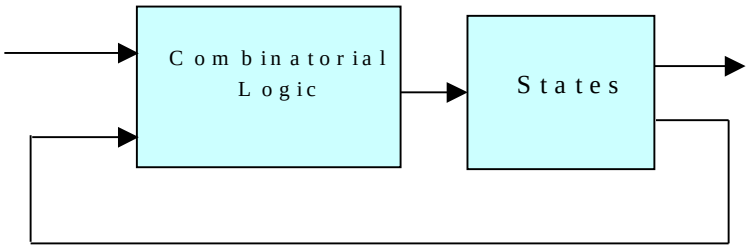

Figure 5.6: Finite State Machine

average can be expressed in an iterative difference equation:

$$MA(k) = MA(k-1) + \frac{[s(k) - s(k-M)]}{M} \tag{5.3}$$

Where MA(k) denotes the moving average output and M is the number of average.

5.3.4 Decision Block Implementation

The decision block is a modified finite state machine [reference here about state machine]. Finite state machine is a powerful tool for describing the discrete event system. A general Moore state machine diagram is shown in Fig.5.6.

The state machine consists of states, inputs and outputs. In a conventional synchronous state machine, states make transition only when the clock ticks. The new states are the combinatorial logic function of the inputs and the current states and the new output is function of current states. One natural implementation of state machine in FPGA is that the states are mapped to flip-flops and combinatorial logic functions are

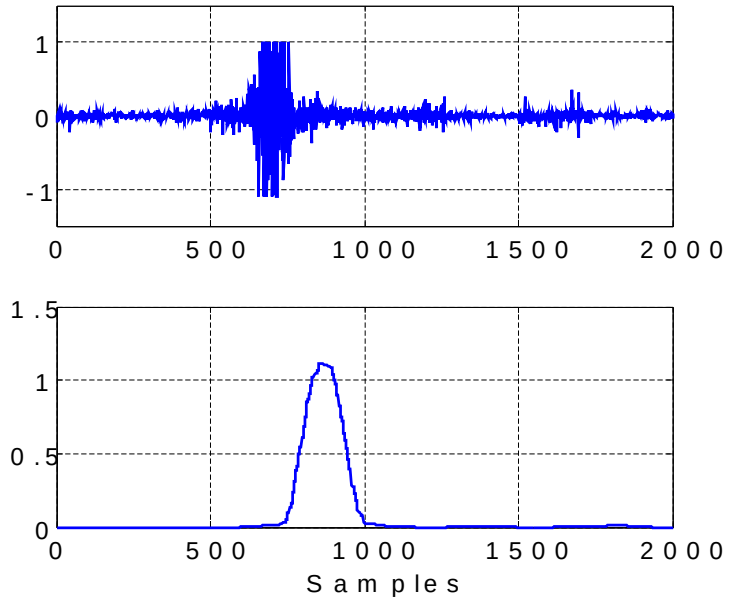

Figure 5.7: Filtering Simulation

Table 5.1: Design Result on Xilinx Virtex-E 2000 FPGA

items	Word length	Supply voltage	Area (%of slices)	Max Freq.	Power at 256 Hz
	8.5	1.8	$\approx 6\%$	$\approx$ 80MHz	0.025mW

mapped to sum of product (first "AND" and then "OR"). In our diction algorithm, the state transition and output also depends on the counter value. Thus a counter is also included in the state machine implementation.

5.4 Design Results

In FPGA implementation, all the data are represented in fixed-point format. The number of bits is chosen to as small as possible for low cost and power consumption.

Figure 5.7 shows the filtering simulation. The top plot is the original acoustic signal waveform s(k) and the bottom plot is the filtered signal s(k). Figure 5.8 shows the detection result and the square pulse shows one vehicle is correctly detected.

All the design results are summarized in Table 5.1

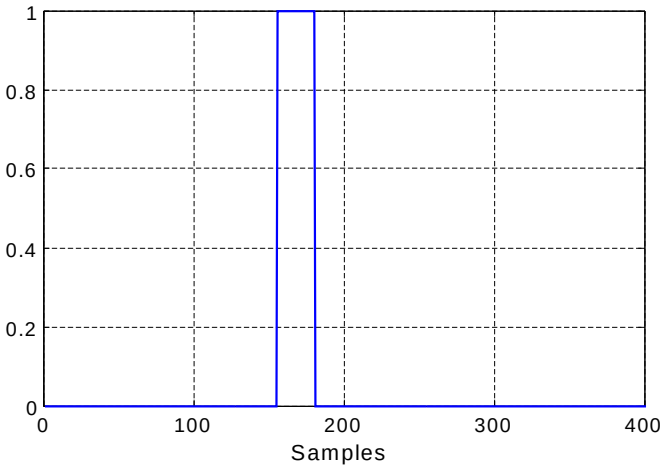

Figure 5.8: Detection Result

5.5 Summary

The chapter evaluates the possible ASIC implementation of the detection algorithm by FPGA implementation. The FPGA implementation demonstrates good power efficiency. The ASIC implementation of the algorithm is justified since the Mote has very limited computation resources and its good power efficiency.

Chapter 6

Conclusion

This research report studied the vehicle detection by the sensor network node (the Mote). The sensors in the Mote system used for vehicle detection are acoustic and magnetic sensors (Magnetometer). The characteristics of acoustic signals emitted from vehicle was analyzed by short time Fourier Transform. Two acoustic vehicle detection algorithms were proposed: the Adaptive Threshold Algorithm and the Min-max Algorithm. Both acoustic algorithms are based on the detection of temporal acoustic energy concentration. Real time tests and offline simulations demonstrated the effectiveness of the two algorithms. A simple threshold slicing algorithm was also used for magnetic vehicle detection and the effectiveness was verified by real time tests. Finally, FPGA implementation of the acoustic algorithm was presented for power efficiency.

Bibliography

[1] TinyOS, A component-based OS for the networked sensor regime, http://webs.cs.berkeley.edu/tos/

[2] Crossbow, Manufacturer of Smart-Dust, http://www.xbow.com/

[3] A Summary of Vehicle Detection and Surveillance Technologies used in Intelligent Transportation Systems, The Vehicle Detector Clearinghouse, Southwest Technology Development Institute (SWTDI) at New Mexico State University (NMSU), Fall 2000

[4] E. Jason Riedy and Robert Szewczyk, Power and Control in Networked Sensors, http://webs.cs.berkeley.edu/tos/

[5] S. Coleri, M.Ergen, and T.J. Koo, Lifetime analysis of a sensor network with hybrid automata modelling, Processings of ACM International Workshop on Wireless Sensor Networks and Applications (Atlanta, GA), Sept. 2002.

[6] Michael J. Caruso, Lucky S. Withanawasam, Vehicle Detection and Compass Applications using AMR Magnetic Sensors, Honeywell, SSEC, 12001 State Highway 55, Plymouth, MN USA 55441 http://www.ssec.honeywell.com

[7] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler, Kristofer Pister, System architecture directions for network sensors, ASPLOS 2000, Cambridge, November 2000

[8] Sinem Coleri, PEDAMACS: Power efficient and delay aware medium access protocol for sensor net-

works, master thesis, Electrical Engineering and Computer Science at University of California, Berkeley, Fall 2002.

[9] MATLAB documentation on SFFT, http://www.mathworks.com/ access/helpdesk/help/toolbox/dspblks/shorttimefft.shtml