University of Southern Queensland
Faculty of Engineering & Surveying

Development of a Mobile Vehicle Classification System

A dissertation submitted by

Colin W. Otto

in fulfilment of the requirements of

ENG4112 Research Project

towards the degree of

Bachelor of Engineering Electrical and Electronic

Submitted: November, 2006

Abstract

The requirement of non-invasive counting and classifying devices has grown significantly in the last few years due to contributing factors from Occupational Health and Safety Standards developed by state road authorities. There is significant risk associated with departmental personnel working in proximity to traffic. The risk is elevated further for these personnel working on the road whilst installing the devices in high speed and high volume traffic environments.

The focus of this project was to develop a technology capable of classifying vehicles into the Austroads 94 12 Bin Classification Standard. The restrictions for the system were such that no devices or apparatus are placed on the carriageway, the detection system would not affect motorists in anyway and the system would be portable and suitable for multilane high-speed roads.

The system chosen to develop was based on the use of 2 infrared lasers. The project deals with noise in the laser data via comparison of denoising techniques such as Low Pass Filtering, Frequency Domain Thresholding and Discrete Wavelet Transform. It was found in the project that Wavelet Denoising offers the best noise reduction for the signal and was far superior at retaining the signal properties. The accuracy of the developed system was found to be low compared with that of commercially available systems. However, the discrepancy is attributed to sensor synchronization, a problem able to be solved by the introduction of a designated processor.

University of Southern Queensland
Faculty of Engineering and Surveying

ENG4111/2 *Research Project*

Limitations of Use

Certification of Dissertation

I certify that the ideas, designs and experimental work, results, analyses and conclusions set out in this dissertation are entirely my own effort, except where otherwise indicated and acknowledged.

I further certify that the work is original and has not been previously submitted for assessment in any other course or institution, except where specifically stated.

Colin W. Otto

0011121354

____

Signature

____

Date

Acknowledgments

I would first like to thank Wei Xiang for his guidance throughout the project. Wei was very helpful and often acted as a bouncing board for my ideas. I would also like to thank the Department of Main Roads, and in particular Jeshua Brouwer for the opportunity to conduct such an interesting project. Jeshua was excellent to approach for advice when I felt I had hit a wall.

Finally, I would like to acknowledge the sacrifice made by my family for the duration of this degree. My wife, Kerri-anne has been very supportive and understanding of the effort required to achieve what I have done. To my children Ryan and Logan, I hope I have had some influence upon your future education, and thank you for being part of mine.

COLIN W. OTTO

University of Southern Queensland

November 2006

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Traffic Data Collection

Modern day traffic engineers have long relied upon traffic surveys for collection of data. Most traffic control and design problems demand a fairly detailed knowledge of the operating characteristics of the traffic concerned (Taylor, Bonsall & Young 2000). These characteristics include:-

- Traffic counts and classifications.
- Speed surveys.
- Vehicle mass and dimensions.
- Annual Average Daily Traffic (AADT)
- Design hour volume.

Extensive research in automated traffic surveys, produced by sensors either embedded in or above the road surface, have been conducted over the past 50 years. Counting and classification systems developed from a wide variety of methods include:-

- Pneumatic tubes
- Piezo sensors
- Inductive loops
- Magnometer sensor
- Infrared traffic logger (TIRTL)
- Light detection And ranging (LiDAR)
- Radar
- Ultra sonic
- Video

These counting systems can be further classified into 2 distinct areas:-

- Invasive - A system where the sensors are installed within or upon the carriageway (Martin, Feng & Wang 2006).
- Non-Invasive - A system where the sensors are installed above or adjacent to the carriageway with no disruption to traffic flow (Martin et al. 2006).

The requirement of non-invasive counting and classifying devices has grown significantly in the last few years due to contributing factors from Occupational Health & Safety Standards developed by state road authorities. There is significant risk associated with departmental personnel working in proximity to traffic. The risk is even greater for these personnel working on the road, installing the devices, in high speed and high volume environments. The Department of Main Roads Queensland, has developed a standard that requires personnel working within 1.2 metres of traffic travelling at 70 km/hr or greater to apply control methods (*Working In Proximity to Traffic* 2005).

1.2 Classification

Vehicle classification is the classification of the vehicle into one of a number of distinct groups. It is usually applied in the disaggregation of traffic data, for example in the analysis and reporting of weigh-in motion data or the determination of Annual Average Daily Traffic (AADT). There is a current AUSTROADS Vehicle Classification System, comprising 12 classes, that has served well in its current form for the past 5 years (Aamport 2000).

1.2.1 AUSTROADS CLASSIFICATION STANDARD

The current AUSTROADS Vehicle Classification System was updated in 1994, following AUSTROADS project RUM.3.D.8 (Aamport 2000). The previous system was found to be deficient in certain areas. These deficiencies were addressed, and the current system is summarized in Table 1.1 below, and presented in detail in Appendix B.

Table 1.1: AUSTROADS Vehicle Classification System Summary

Class	Description
1	Short Vehicle
2	Short Vehicle Towing
3	Two Axle Truck or Bus
4	Three Axle Truck or Bus
5	Four Axle Truck
6	Three Axle Articulated
7	Four Axle Articulated
8	Five Axle Articulated
9	Six Axle Articulated
10	B-Double
11	Double Road Train
12	Triple Road Train

The AUSTROADS Vehicle Classification System determines vehicles based on 3 levels. They are:

1. **Length** eg. Short, up to 5.5m
2. **Axles and Axle Groups**, eg. 3,4 or 5 axles in 3 groups
3. **Vehicle Type**, eg. Four axle truck

1.3 Workplace Health and Safety

The primary risk associated with much of the work conducted by departmental personnel is working in proximity to traffic. This is reflected in the number and severity of incidents where people are struck by vehicles while in proximity to traffic.

All personnel who access the road reserve and work in proximity to traffic are at risk, but it is greatest for those working on the road, in high speed and high traffic volume environments, in low visibility conditions or less controlled environments. Where personnel are within 1.2 metres of traffic travelling at 70 km/hr or greater, the risk is rated as extreme and controls are to be applied.

The potential for injury to personnel is significantly decreased when effective traffic control measures are implemented, or alternatively personnel are able to removed from the dangerous situation.

Elimination of Risk

If possible, personnel should eliminate the risk by conducting tasks in a manner that does not require access to the roadway. However, it is recognised that direct physical observation is often the only way of conducting tasks. While the "MUTCD Pt 3 Work on Roads" describes access durations on the roadway of 5 minutes or less, the implementation of this procedure allows for longer durations based on risk assessment at very low volumes. Advance warning signage is not required for the activities conducted under this procedure. The criteria used in this procedure are based on the number of vehicles per hour, as this reflects actual conditions, rather than the AADT (Annual Average Daily Traffic count) which is based on a 24 hour average.

The present system of traffic data collection implemented by the Department of Main Roads requires that personnel enter the carriageway to set up the devices.

1.4 Project Outline

The proposed project is divided into three sections, research, development of a trial system and testing.

Research

The project will require research into a suitable technology to detect and count traffic as per the AustRoad 94 classification system (12-Bin). The technology should be such that no devices or apparatus are placed on the carriageway, the detection system would not affect motorists in anyway and the system would be portable and suitable for multilane high-speed roads.

Development of a Trial System

The project will develop a trial system that uses the chosen technology. The trial system should be able to count vehicles as per the AustRoads 12-Bin classification system and ideally would be able to identify vehicles in individual lanes, individual vehicle speeds and headway between vehicles. Furthermore the system should be mobile and simple to setup in the roadside environment.

Testing

The project will develop a testing process to evaluate the accuracy of the trial system. The data should be compared against approved/calibrated counting devices.

1.5 Axle Sensors

The sensor chosen for the project is the Universal Laser Sensor (ULS), produced by Laser Technology Incorporated. The device has three modes of operation:-

- Averaging Mode
- Binning Mode
- Look Down Mode

The chosen mode of operation was Averaging Mode, where a number of measurements are taken and then averaged, for a more accurate result of distance measurement.

For the purpose of the project, two sensors were placed adjacent to the carriageway, approximately 1 metre apart.

1.6 Dissertation Outline

Chapter 2 Factors affecting successful collection of data
In order for the system to function efficiently, a number of considerations must be made as to circumstances that may inhibit the ability of the system to collect data. This section looks at the likely obstacles and addresses possible solutions.

Chapter 3 Evaluation of existing detector methods
Investigation in to existing systems of Vehicle Counting and Classification. This section also looks into the criteria of system setup, as well as the accuracy of the system, in order to determine a benchmark for accuracy of the project.

Chapter 4 Universal laser sensor

An investigation of how the sensors operate and an analysis of the correct pulse firing rate for the lasers, as well the required number of pulses for each measurement. The mode of operation is explained and the safety precautions of the sensors are stated.

Chapter 5 Processing of signals

This section looks at fundamentals of signal processing and the application of these principles to the signals involved with the project. Issues such as sampling rate and noise removal are addressed. Different noise removal techniques are evaluated from their ability to improve the quality of the signal received from the sensors.

Chapter 6 Wavelet denoising

Wavelet Transforms offer a relatively new concept that has the advantage over Fourier analysis in that it offer both Scale(Time) information and Translation(Frequency) information. The application of wavelet analysis in this section primarily focuses on the ability to significantly denoise a signal whilst retaining the important high and low frequency content of the sensor signal.

Chapter 7 Comparison of denoising techniques

This section uses a Noise Reduction Ratio developed for the project to compare the amount of noise removed by each denoising technique. As well as noise removal, the chapter investigates how each denoising technique affected the most important information within the signal, the rising and falling edges.

Chapter 8 Classification algorithm

This chapter provides a step-through process as to what operation each block of the algorithm is performing. Firstly collecting and validating sensor data, then how axles are detected, speeds of vehicles and axle separations. This data is then used to classify the vehicles into the 12 bin system. Further statistical parameters are also determined. The actual Matlab code is provided in Appendix D.

Chapter 9 Results

Comparison of denoising techniques on a real sensor signal to confirm theory previously presented. A test procedure to determine the accuracy of the system is also presented. Firstly, a vehicle of known speed and wheel base dimensions is used to determine the accuracy of the axle detection and speed determination, followed by a method to determine whether the algorithm was able to correctly classify vehicles into the 12 Austroads 94 Classification Bins. The results from the test procedures are presented and reasons for any deviations are discussed.

Chapter 10 Conclusions

The problems encountered for the duration of the project are discussed in detail. Synchronization of the two data channels and options not provided by the ULS interface are some examples of the issues addressed.

Chapter 2

Factors affecting successful collection of data

Successful collection of data using a system such as the one presented in this thesis is dependent upon the environment that the sensors could be expected to operate within. The environment that the ULS is required to operate within includes factors such the shape of the road surface. Other factors that may inhibit data collection is the position of the vehicles on the carriageway relative to one another and limitations of the ULS itself.

2.1 Carriageway

When considering detecting vehicles within a carriageway, the construction and shape of the carriageway must be considered. Since the ULS is an optical device, line of sight is of the utmost importance. In order to be able to detect only axles, the laser must be low enough to the road surface so as to detect the wheels of the vehicle, but not the underside of the vehicle.

Water upon the surface of the road leads to vehicles aquaplanning, and as a preventative measure, roads are constructed with what is known as crossfall. Crossfall is defined as the slope, normal to the alignment, of the surface of any part of the carriageway (*Road Planning and Design Manual* 2004).

Straight sections of road traditionally have crossfall from the median, sloping downward to the shoulder of the carriageway as seen in Figure 2.1. Alternatively, the carriageway may slope downwards from the shoulder to the median if the conditions so require. In either case, the crossfall of the surface should pose no problems with line of sight as the crossfall is constant.

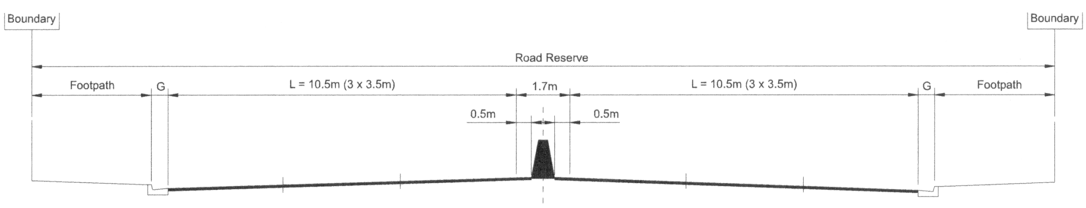

Figure 2.1: Multilane Divided Highway, Road Planning and Design Manual, 2004 P.7-75

The designs of the carriageways seen in Figures 2.2 and 2.3, require consideration of the crossfall for ULS line of sight ability.

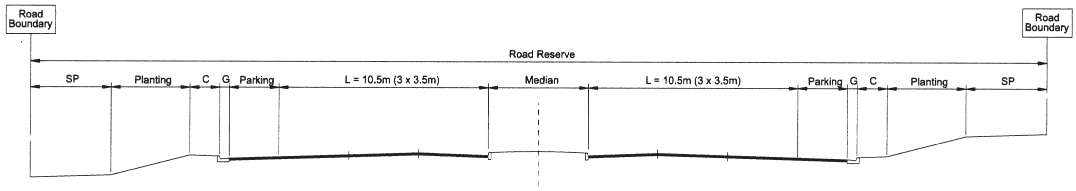

Figure 2.2: 6 Lane Divided Highway, Road Planning and Design Manual, 2004 P.7-76

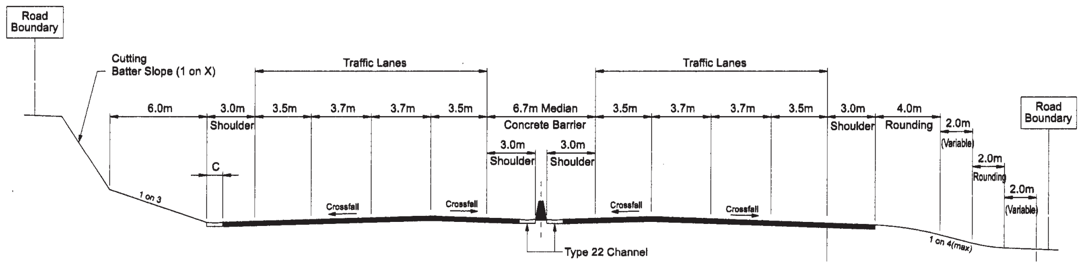

Figure 2.3: 8 Lane Divided Highway, Road Planning and Design Manual, 2004 P.7-78

Here the centre lane has crossfall opposing the crossfall of the other lanes. The height of the infra-red beam at the far side of the carriageway could possibly be higher than the rocker panel of the vehicle being detected.

Table 2.1 shows typical crossfalls, used by Department of Main Roads, dependant on the type of road surface used (*Road Planning and Design Manual* 2004).

Table 2.1: Typical Pavement Crossfalls

Road Surface	Traffic Lane (%)	Shoulder (%)
Cement Concrete	2.0 - 3.0	2.0 - 4.0
Asphaltic Concrete	2.5 - 3.0	2.5 - 4.0
Sprayed Seal	3.0 - 4.0	3.0 - 4.0
Unsealed	3.5 - 4.0	4.0 - 5.0
Within Floodways	1.0 - 2.0	1.0 - 2.0

To determine whether the sensor is capable of measuring distance without interference from the undercarriage of a vehicle in the far lane, the minimum ground clearance of a vehicle needs to be considered.

The *TRANSPORT OPERATIONS, ROAD USE MANAGEMENT–MASS, DIMENSIONS AND LOADING, REGULATION 2005 - SECT 27* states,

1. A person must not drive a vehicle on a road if the vehicle has a ground clearance of less than–
   (a) at a point within 1m of an axle–100mm; and
   (b) at the midpoint between adjacent axles – one-thirtieth of the distance between the centre of each axle; and
   (c) at any other point–the distance that allows the vehicle to pass over a peak in the road if the gradient on either side of the peak is 1:15.
2. In this section–
   (a) ground clearance of a vehicle means the minimum distance between the ground and the vehicle's underside, other than its tyres, wheels, wheel hubs, brake backing plates, flexible mudguards and mudflaps.

Source (*TRANSPORT OPERATIONS (ROAD USE MANAGEMENT–MASS, DIMENSIONS AND LOADING) REGULATION* 2005).

The lowest vehicle that could be expected would be a vehicle from Class 1. Point 1(a) is not applicable since this point would be shielded from the laser by the wheel, that is it is behind the wheel. An average wheel base, or distance between the wheels, of a Class 1 vehicle is 2.8 metres. From point 1(b), the minimum ground clearance of this vehicle is calculated by dividing the wheel base by thirteen.

$$\text{Ground Clearance} = \frac{\text{Wheel Base}}{13}$$

So for a vehicle with a wheel base of 2.8 m, the minimum ground clearance must be 215 mm.

From Figure 2.2, it can be seen that the lane width is 3.5 metres. The height of the laser sensor above the carriageway is less than 30 mm. Taking the maximum crossfall from Table 2.1 as four percent, the height of the infra-red beam at the far side of the downward sloping lane can be determined.

The cross fall can be converted into degrees by taking the arctangent of the crossfall.

$$\text{Crossfall (degrees)} = \tan^{-1}\frac{3}{100}$$

The change in angle is also doubled since the first lane is sloping upward, and the second lane sloping downward. The height of the laser at the far side of the downward sloping lane will be, the height of the ULS sensor above the carriageway plus twice the change in height across one lane.

The crossfall for one lane is 105 mm, so the fall across the far lane is 210 mm. The height of the ULS sensor, 25 mm, also needs to be added to this giving the height of the infra-red beam as 235 mm.

The height of infra-red beam at the far side of the far lane is actually higher than the minimum ground clearance required. This could result in axle detections not being able to be recorded by the ULS. As the ground clearance used is a minimum, the majority of vehicles would be expected to have a ground clearance greater than this which may allow successful axle detection.

2.2 Occlusion

Occlusion refers to the overlapping of vehicles due to the perspective of the sensor. Occlusion will affect the system when two separate vehicles occupy the sensor at the same time. Failing to detect and resolve the presence of occlusion will lead to data errors, such as misclassification of vehicles, incorrect count of the number of vehicles and missing axle counts. Figure 2.4 shows the output of a laser sensor for two vehicles in two separate lanes. In the top of the figure, there is a representation of the formation of the axles as passing by the laser sensor. Here, each axle can be independently detected since only one axle occupies the laser sensor at any one time.

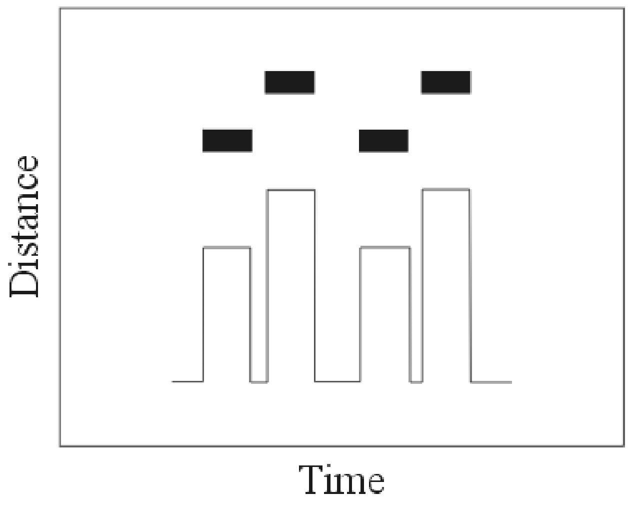

Figure 2.4: Typical Laser Output showing 2 separate vehicles in 2 lanes

In Figure 2.5 the first axle of the vehicle in lane two is partially occluded by the vehicle in lane one. The smaller spike on the first axle strike could be determined to be an axle in a further lane, provided that the spike existed for a consistent period. A noise spike would be expected to only exist for a brief burst of one or two samples, however a presence can be determined by a longer duration such as three or four samples. If the sampling rate of the laser sensor is set high enough, this will be accomplishable.

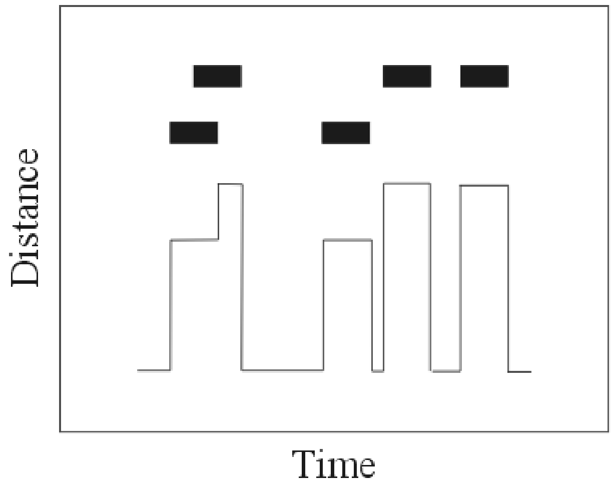

Figure 2.5: Laser Output showing partial occlusion

The third possible scenario for occlusion is the total blocking of an axle in lane two by an axle in lane 1, as in Figure 2.6 on laser sensor one output. The probability of the two vehicles staying in exactly the same positions as they pass both sensors would be low, that is, they would not be expected to be traveling at the exact same speed. It is therefore reasonable to assume that on either laser sensor one or two each axle would be present. This is represented in Figure 2.6.

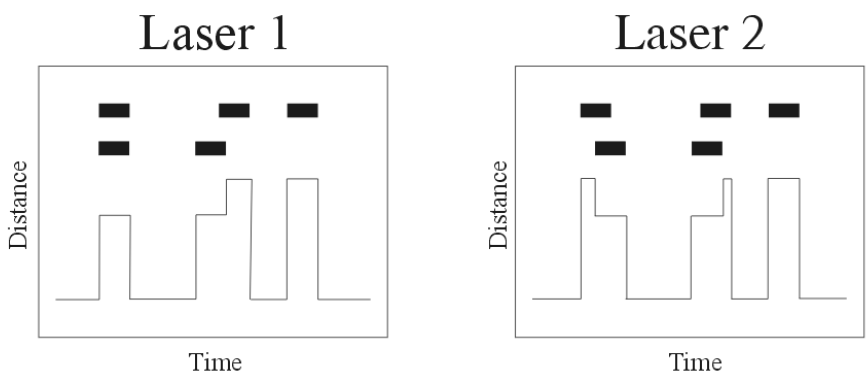

Figure 2.6: Laser Output showing occlusion on both sensors

To determine all axles in each lane, the lane data can first be separated into individual lanes as in Figure 2.7, where the data for lane two is shown.

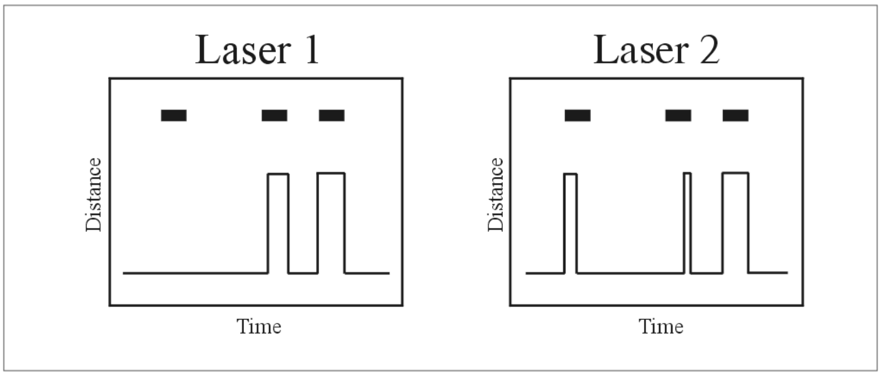

Figure 2.7: Data separated into lanes

The vehicles speed would not be expected to dramatically change over the separation of the two laser sensors, hence the axle spacings should remain constant. The two waveforms can be added and analysed conjointly, removing the earlier situation of occlusion of the first axle, as in Figure 2.8.

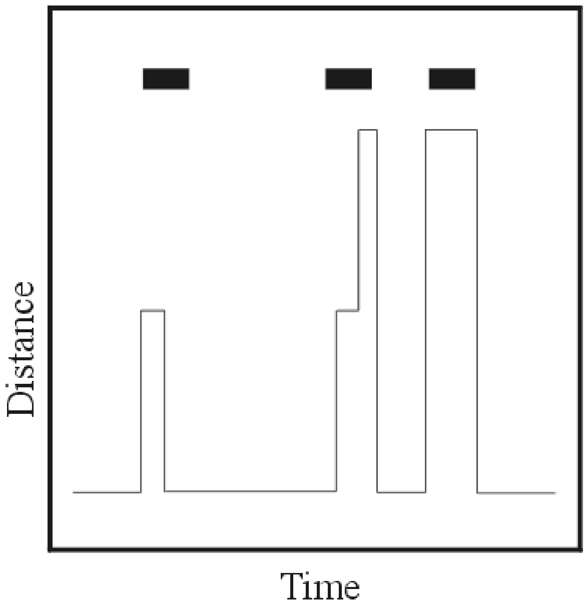

Figure 2.8: Laser sensor output summed.

From here it can be seen that there exist three distinct axles.

2.3 Summary

From the material presented in this chapter, it can be determined that there are considerable challenges for the system to overcome. The various challenges presented here are also faced by existing commercially available systems. Occlusion is a particular problem common to all systems whether they be pneumatic tubes, infra-red beams or video in a multi-lane environment. There is no way of determining the effect of any of the challenges except by use of trials. It would be desirable to be able to conduct trials in a controlled environment, recreating the situations that would be likely to cause problems for the system and thereby determining the effects. The limited resources available for this project will prevent this from taking place.

Chapter 3

Evaluation of existing detector methods

3.1 Pneumatic Road Sensor

The pneumatic road sensor, the first intrusive traffic detector technology, was invented in the 1920's. Due to its simplicity and low cost, the pneumatic road sensor still is widely used today. The road sensors sense vehicle pressure and send a burst of air pressure along a rubber tube when a vehicles tyre passes over them. The pulse of air pressure closes an air switch and sends an electrical signal that marks the passage of a vehicle. Pneumatic road sensors can detect volume, speed, and classification by axle count and spacing. The detectors typically are used for short term traffic counting.

3.1.1 Metrocount 5600 Vehicle Speed Accuracy

The Metrocount 5600 Roadside Unit is a portable device capable of measuring vehicle speeds and distance between axles from the use of two pneumatic sensors that simply log the time of detected air pulses. There are two types of errors encountered by the system introduced from several sources (Metrocount 2001).

- Random Errors, such as the inherent timing uncertainty of the Roadside Unit.
- Systematic Errors, such as the sensor length and spacing.

The random errors are due to the discrete resolution of the measuring device. Systematic errors cause errors that tend to be fixed and thus create an offset in the true measurement.

The Metrocount 5600 Roadside Unit (RSU) has a resolution of sensor hits of 833 microseconds. The uncertainty of the calculated speed measurement to the true vehicle speed is given in Table 3.1.

Table 3.1: Metrocount 5600 Speed Error

True Speed	Speed Error (km/hr)	Speed Error (%)
10	0.0232	0.23193
20	0.0930	0.46493
30	0.2097	0.69902
40	0.3737	0.9342
50	0.5852	1.1705
60	0.8447	1.4079
70	1.1525	1.6464
80	1.5088	1.886
90	1.9141	2.1268
100	2.3687	2.3687

Source (Metrocount 2001).

Figure 3.1 shows the layout for a typical traffic survey using the Metrocount 5600 Roadside Unit. Within the setup, there are four possible sources of data error.

- Sensor Length
- Sensor Spacing
- Relative Sensor Angle
- Angle of Vehicle Incidence

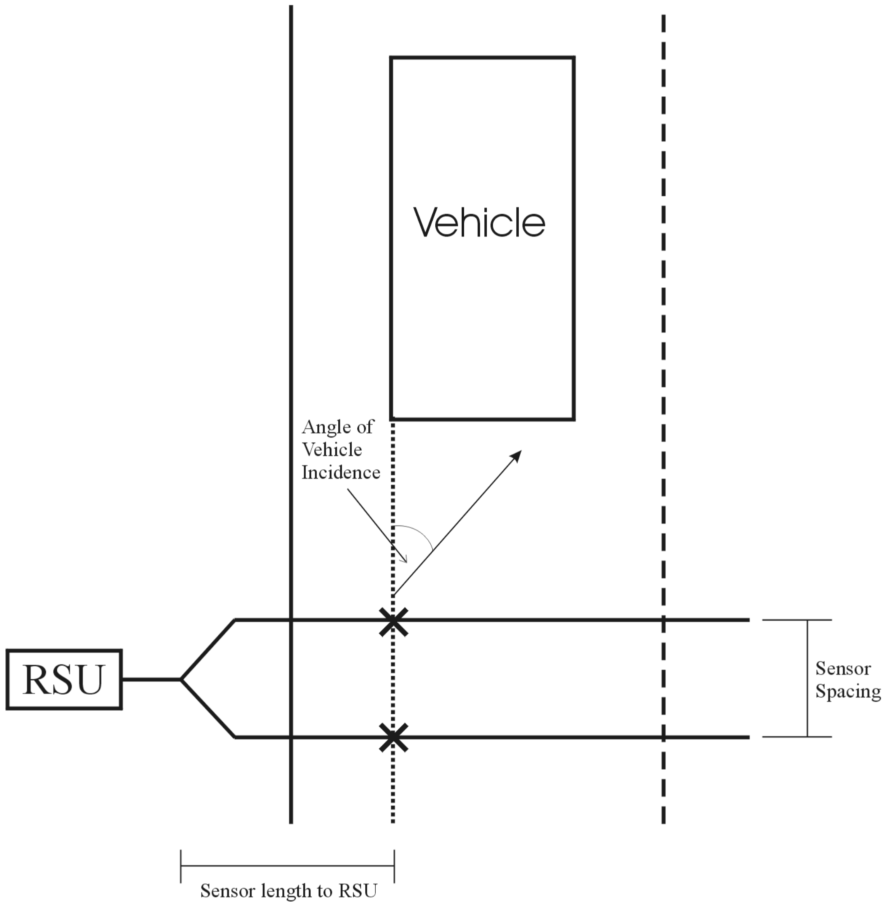

Figure 3.1: Setup Diagram for the MC5600

Sensor Length

Air pulses travel down the tubes at the speed of sound. A difference in length of the tube from the position where the wheel struck the tube to the RSU introduces a delay into one channel. The result is that vehicles travelling in one direction will be recorded as going too fast, and vehicles in the other direction will be recorded as going too slow(Metrocount 2001).

Sensor Spacing

Any variation in the spacing of the pneumatic tubes will result in errors in speed measurement. The deviation from the required spacing is inversely proportional to the error in the speed. If the spacing is 10% larger than required, the resulting speed calculation will be 10% slower than the actual speed of the vehicle(Metrocount 2001).

Relative Sensor Angle

The tubes must be installed parallel to each other, or the speed will be dependant on the position across the lane of the tube strike(Metrocount 2001).

Angle of Vehicle Incidence

Vehicles crossing the tubes should be doing so perpendicularly. Any deviation from the normal will result in increased distance to traverse the pair of tubes. An angle of incidence less than 10° results in a maximum error of 1.5% (Metrocount 2001).

3.1.2 System Accuracy

An independent analysis (Bastian 1994) of the Metrocount 5600 pneumatic classifier, conducted by Microcom Pty Ltd for Department of Main Roads Western Australia found that of the 1637 vehicles classified by the device, an accuracy of 99.7% was achieved.

A separate independent survey (Pell 2002) conducted by the Department of Infrastructure, Energy and Resources found that the Metrocount 5600 had almost 100% accuracy over the 93 heavy vehicles that it classified.

It is apparent that the pneumatic sensor classification method is a very accurate system. There was however, no test data available as to the accuracy of the sensors when determining wheel base and speed. Since the accuracy of the overall system is high, it would be expected that the accuracy of the individual components would also be high, as the correct classification of vehicles is dependant upon the correct determination of the speed and wheel base dimensions.

3.2 The Infrared Traffic Logger (TIRTL or RIRTM)

The TIRTL system is a portable, light based vehicle counter, classifier and speed measurement product that measures vehicle axle breaks. The system can be employed either as a temporary field installation or as a permanent installation operating on carriageways of up to eight lanes or distances up to 200 metres.

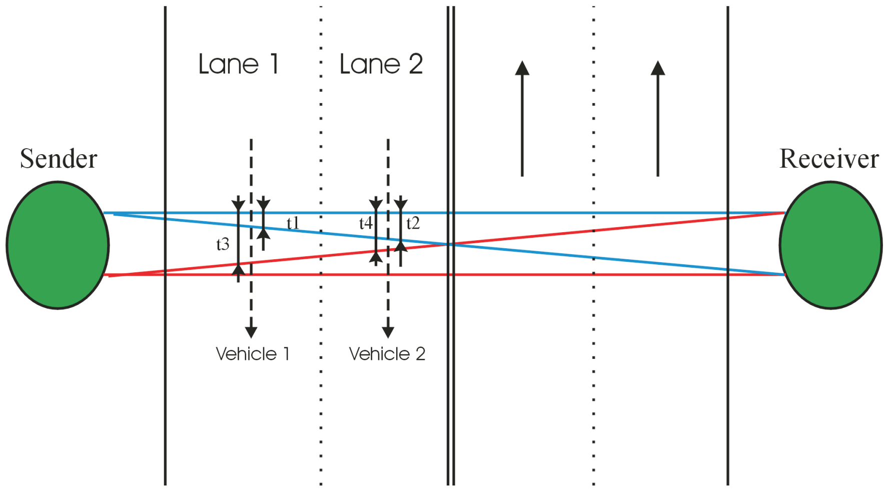

Figure 3.2: Setup Diagram for The Infra-Red Traffic Logger

The system consists of an Infra-Red (IR) beam transmitter and receiver located on opposite sides of the carriageway, as seen in Figure 3.2. Classification of the data is undertaken in real time onboard the receiver unit. Vehicle classification may be undertaken to either of the pre-programmed standards, Austroads 94 or US Federal Highway Administration Scheme F. Alternative classification may be undertaken based on features such as vehicle or on a user defined classification scheme.

The system uses the order of timing of IR beams to determine the location and speed of passing vehicles. Vehicle speed is calculated from the time between the outer beam breaks, whereas location can be determined from the crossed beam breaks.

In the RIRTM Product Outline (*TIRTL Product Overview* n.d.) it is claimed that: "For vehicles travelling at an average speed of 100km/hr, and with an average separation of 3 seconds from each other, the vehicle classification accuracy of the TIRTL system is better than 99.6% - 2 Lanes, 99% - 3 Lanes, 98.4% 4 Lanes, 97.8% - 5 Lanes, and 97.2 % - 6 Lanes. For less dense traffic, the accuracy improves dramatically."

An independent survey provided by Department of Main Roads Queensland conducted a number of field trials with the device. The device was tested over 4 different sites and found to have an error of up to 2% (Traffic & Road Use Management Division ITS Infrastructure 2002).

A number of trends became obvious during the data analysis. It was noticed that certain events confused the system, after which it would output a string of unclassified vehicles, or make incorrect classifications. Example events that confused the system include pedestrians walking through the beams, cars coming to a halt blocking the beams, vehicles pulling out of a driveway or pulling over to park and bicycles. Occasionally two vehicles crossing very close together would confuse the system leading to erroneous classification. Cars with low suspension were also a common cause of a string of unclassified vehicles.

The error in the overall traffic counts for the sites ranged between 0.5 % and 2.0% of the real vehicle numbers as determined by the video footage. The overall commercial vehicle predictions ranged from 0.1% and 0.4% of the real number of vehicles and compares favourably to pneumatic tube based classifying equipment (Traffic & Road Use Management Division ITS Infrastructure 2002).

Correct site setup appeared to have a major influence on the number of unclassified vehicles reported at the site. Correct alignment of the transmitter and receiver was a major contributing factor, and the camber of the road also contributed to the units ability to sense only the axles of the vehicles (Traffic & Road Use Management Division ITS Infrastructure 2002).

3.3 Summary

In this chapter, two systems have been analyzed. One system is the most common system used in Australia for counting and classifying vehicles, and the other system is similar in operation to the one proposed by the project in that it uses infra-red beams to detect axles. It should be noted that neither of the systems investigated in this chapter are able to satisfy the project objectives.

The pneumatic system requires that workers enter the carriageway to secure the sensors to the carriageway, exposing the workers to inherent dangers in high volume or high speed traffic environments. The TIRTL also requires a receiver unit on the opposite side of the carriageway, so the installer may be exposed to the same dangers. The sensors chosen for the project require only to be placed adjacent to the carriageway, and are therefore non-invasive.

For the sensors to be deemed acceptable, the speed accuracy should be greater than 97% in error up to 100km/hr, and the classification accuracy should also be greater than 97%. If these accuracies can be achieved, then further research should be conducted into turning the system presented in the project into a commercially viable system.

Chapter 4

Universal Laser Sensor

4.1 ULS Signal

The sensors transmits energy in short bursts with comparatively long spaces between pulses. In Figure 4.1, it can be seen that the system operates very similar to that of a pulsed radar system.

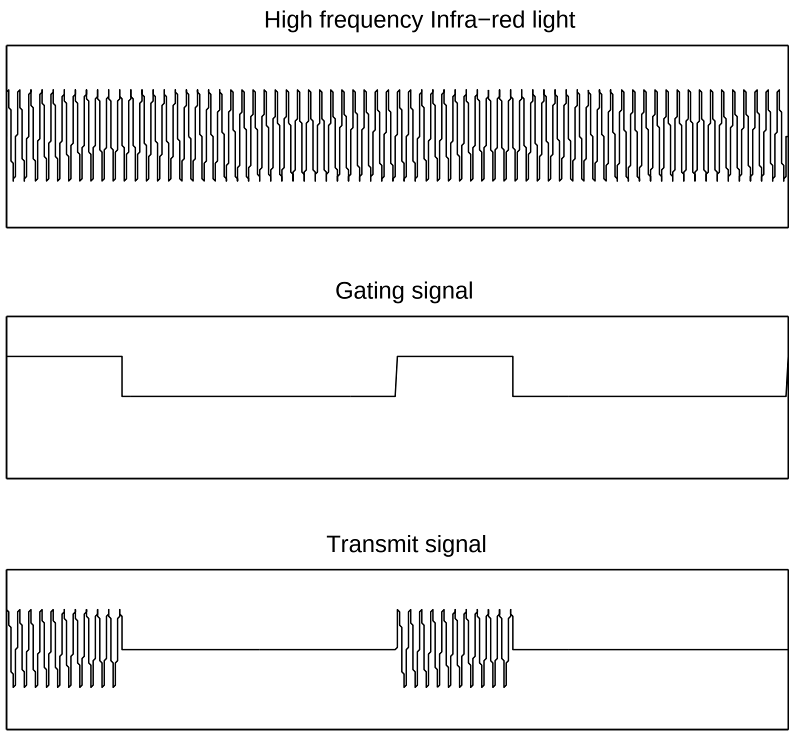

Figure 4.1: Transmitted signal of ULS

Bursts of the infra-red signal are gated via the use of the pulse repetition frequency, which is labeled Pulse Firing Rate Frequency (PRF) by the manufacturer of the sensors. Each burst of signal travels out to a target, and is reflected from the target. This reflection is then received by the sensor. The time of flight of the burst is recorded in order to determine the distance to the target. The speed of light is relatively constant across a large array of atmospheric conditions, so the distance may be calculated reliably, no matter the atmospheric condition.

Ideally, the burst of infra-red light must return to the sensor before the next burst is transmitted. This leads to a maximum range of distance that the sensor will be able to determine. Conversely, the burst of infra-red light must not return to the sensor before the completion of the current transmission. This is the limitation of the minimum distance that the sensor may be able to accurately determine.

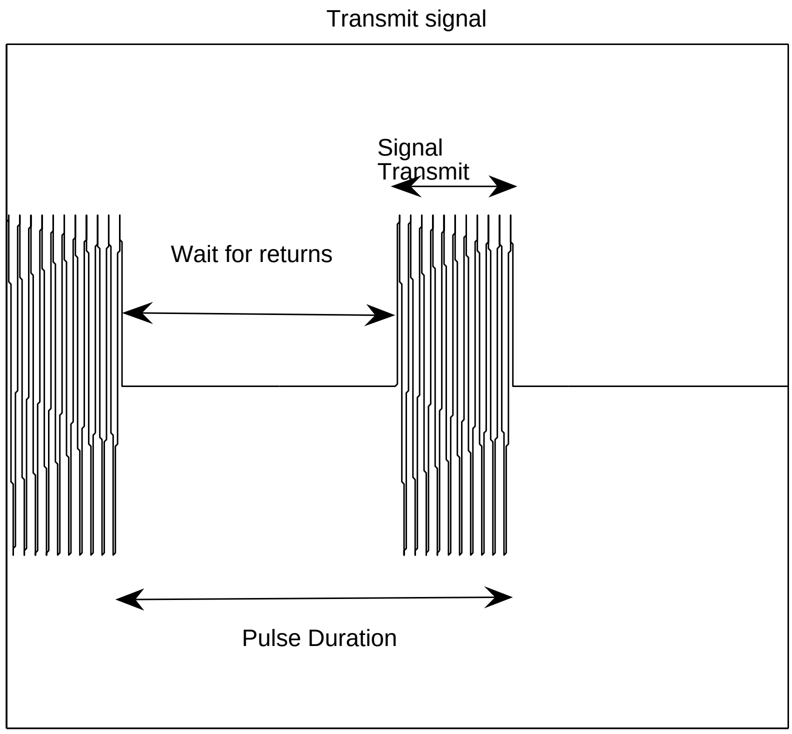

Figure 4.2: Transmission of burst of infra-red signal

In Figure 4.2, this can be seen more closely. The duty cycle of the gating signal effectively determines the minimum and maximum distance achievable by the sensor. The duty cycle is the ratio of high and low states in the signal. The pulse repetition frequency is directly related to the number of actual measurements taken by the sensors. In Figure 4.2 there would have been two measurements recorded by the sensor. In the next section, it is discussed how the sensors are able to improve the reliability of the measurements through the use of an averaging parameter.

4.2 Device Settings

The Universal Laser Sensor (ULS) is a user-configurable device, which allows adjustment of settings to optimize measurement performance depending on the application. The device allows setting of the laser pulse firing rate frequency (PRF), from 10 to 5000Hz. To determine a distance to a target, the device averages the distance over a user defined number of pulses. This is termed pulses per measurement (PPM).

The pulses per measurement divided by the pulse firing rate frequency, establishes the measurement output or update data rate.

$$\text{Output Data Rate} = \frac{PPM}{PRF}$$

The settings chosen for the project are,

- PRF = 4000
- PPM = 16

The resulting output data rate is a single measurement every 0.004 seconds, or a measurement frequency of 250 Hz.

Unit Analysis

$$\frac{pulses/measurement}{pulses/second} = \frac{seconds}{measurement}$$

For a high output data rate, either the PPM must be low, or the PRF must be high. Considering the case of a semi-trailer, where the grouped axles are considerably closer than in a passenger vehicle, there must exist sufficient laser pulses between axle detections in order to be able to distinguish between axles.

Figure 4.3 shows the situation where the axles of a semi-trailer must be detected.

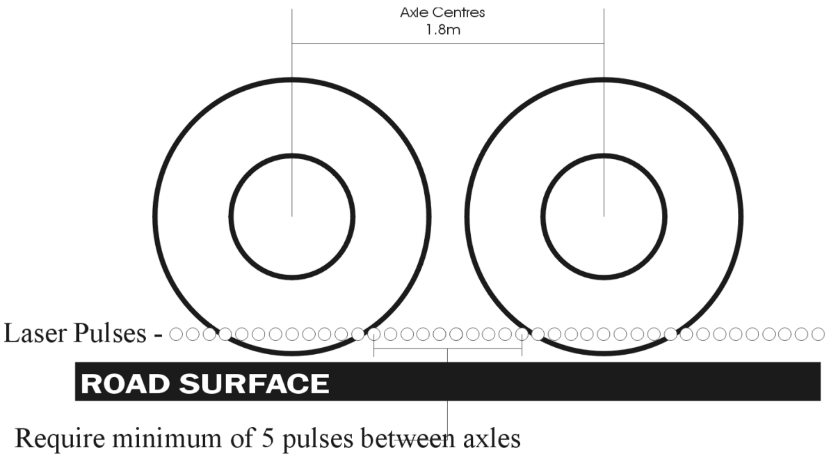

Figure 4.3: ULS Sampling of a Typical Axle Group

The vehicle can be assumed to be travelling at 100 km/hr. This equates to a speed, V of 27.78 m/s. The distance between the tyres at the height of the ULS, d, is 635mm. The resolution of the ULS Interface Capture Software is chosen to be 0.004 seconds per measurement.

Let the number of pulses be n.
Let the velocity of the vehicle be V.
Let the distance between trailing and leading edges of tyres be d.
Let the minimum time between measurements be t.
The following equation can be derived to determine the number of measurements that will be recorded between the first and second tyre.

The time between each measurement given by:-

$$t = \frac{\frac{d}{n}}{V}$$

Rearranging to find n;

$$n = \frac{d}{tV}$$

This gives a value for n as 5.71, therefore, it would be expected that there be 5 measurements recorded between the trailing and leading edges of the tyres.

The values of PPM and PRF must also be defined.

The limiting factor again is the chosen resolution of the ULS Interface Capture Software. The minimum time, t is 0.004 seconds.

$$t = \frac{PPM}{PRF}$$

Choosing the *Pulses per Measurement* as 16, gives the required *Pulse Rate Frequency* as 4000.

Minimum good pulses is the number of good returns required to average as an output result. As the minimum good pulses increases, the averaging increases, which improves the accuracy of the measurement.

Minimum Pulse Width is used to reject receive pulses less than the value specified, in nano-seconds. This allows rejection of weak measurements.

These values become the settings required for Averaging Mode.

Figure 4.4: Graphical User Interface for ULS

4.3 Averaging Mode

The ULS pules rangefinder sends a single pulse of light, typically 8 ns in duration, to a target. It measures the time it takes for the pulse to return to the rangefinder. Since the speed of light is relatively constant in all barometric atmospheres, the distance to the target can be calculated. One individual pulse would not provide a very accurate measurement, since we are dealing with picoseconds (i.e. 10^{-12} seconds). The rangefinder uses a series of pulses and averages them for a more accurate measurement result. The trade off is having a fast measurement rate with lower accuracy as compared to a slower measurement rate for higher accuracy (*ULS Hardware/Software Interface Specifications* 2005). For this reason, the measurement rate should be as low as acceptably possible.

The Averaging Mode also provides a feature called dithering, which can dramatically increase the accuracy of the ULS measurements, without requiring more pulses per measurement. However, the dithering process requires multiples of 32 good receive pulses for correct operation. For the application of the ULS in this thesis, the number of pulses per measurement is less than 32, and hence dithering cannot be used.

4.4 Safety Precautions

Internal Laser Pointer

- The laser pointer's visible laser is not considered FDA(CFR21) Class I eye safe. It is Class IIm. Care should be taken when using any laser pointing device.
- Do not stare directly into the visible laser beam.

Pulsed Laser

- Avoid staring directly at the laser beam for prolonged periods. The ULS is designed to meet FDA eye safety requirements and is classified as eye-safe to FDA (FCR21) Class I 7mm limits, which means that virtually no hazard is associated with directly viewing the laser output under normal conditions. As with any laser device, however, reasonable precautions should be taken in its operation.
- It is recommended that you avoid staring into the transmit aperture while firing the laser. The use of optical instruments with the ULS may increase eye hazard.

Source: (*ULS Hardware/Software Interface Specifications* 2005)

4.5 Summary

The final configuration for the ULS was determined as PRF set to 4000 Hz, and the PPM set to 16. This results in a measurement frequency of 250 Hz. From the Nyquist Theorem, the system should be reliably able to detect frequencies up to and including 125 Hz. This will become important in Chapter 5, that deals with filtering of signals.

Another important factor covered in this chapter is the one of eye safety. One of the project objectives is that the system must not affect motorists in any way. As the lasers are considered FDA (FCR21) Class I eye safe, motorists will face no danger if they were to stare into the transmitter momentarily. There is virtually no risk of this as the motorist would have to be at ground level to do so.

Chapter 5

Processing of Signals

5.1 Fundamentals

Sampling Rate

The first step toward processing a signal is digital sampling of the signal. This is where a continuous time signal, x(t) is sampled at discrete intervals. The output of the sampling process is a discrete-time signal, x(nT).

There are two important parameters to consider when sampling a signal,

- The appropriate value for the sampling interval, T, or the sampling frequency, Fs, which is the equivalent to $\frac{1}{T}$.
- The ability to recover the signal x(t) exactly from the sampled values of x(nT)

The first point is solved by application of the Nyquist Theorem. The Nyquist Sampling Theorem states,

" *x(t) is a bandlimited signal with the maximum signal frequency* Ω_m*, then x(t) is uniquely determined from its samples x(n): n=0,1,2,3,...N-1, if the sampling interval* $T\ \frac{\pi}{\Omega_m}$ *seconds, or, alternatively, if the sampling frequency* $f_S\ \frac{\Omega_m}{\pi}$ *Hz."*(Kumar 2005)

The second point is addressed with the interpolation formula given below in equation (5.1)(Kumar 2005). If the sampling satisfies the Nyquist sampling theorem, then the signal values between samples is given by (Kumar 2005),

$$x_r(t) = \sum_{n=0}^{N-1} x(n) \frac{\sin[\pi(t - nT)/T]}{\pi(t - nT)/T} \tag{5.1}$$

5.2 Fourier Analysis of Signals

Any periodic function, which might represent an arbitrary waveform, can be decomposed into a series of sine and cosine term, called a Fourier series. The representation is specified by determining the numerical coefficients of each sine and cosine term in the Fourier series expansion. These Fourier coefficients measure the intensity of each harmonic component in the original waveform.

The waveform, f(x), may be represented as follows (Kreyszig 1999),

$$f(x) = a_0 + \sum_{n=1}^{\infty}(a_n \cos(\frac{n\pi x}{L}) + b_n \sin(\frac{n\pi x}{L}) \tag{5.2}$$

Where;

- L is half the period of oscillation
- a_0 a_n and b_n are the Fourier coefficients determined by (Kreyszig 1999),

$$a_0 = \frac{1}{2L}\int_{-L}^{L} f(x)dx \tag{5.3}$$

$$a_n = \frac{1}{L}\int_{-L}^{L} f(x)\cos(\frac{n\pi x}{L})dx \tag{5.4}$$

$$b_n = \frac{1}{L}\int_{-L}^{L} f(x)\sin(\frac{n\pi x}{L})dx \tag{5.5}$$

- n is the n^{th} Fourier co-efficient

The application of Fourier series holds the requirement for a periodic signal. To determine the frequency content of any signal, another technique may be applied, Fourier transform.

The continuous Fourier Transform is mathematically defined as (Kreyszig 1999),

$$X(\Omega) = \int_{-\infty}^{\infty} x(t)e^{-j\Omega t} \tag{5.6}$$

A discrete version of the continuous Fourier transform can be implemented in Matlab, the Fast Fourier Transform (FFT). The Fast Fourier Transform has the requirement that the number of samples of the signal be a power of two. Using this Matlab function, it is possible to determine the spectral content of signals.

5.3 Fourier Analysis of a Square Wave

Consider the following squarewave representation.
For a period of 2 π,

$$f(x) = |^{0:0<x<\pi}_{1:\pi<x<2\pi}$$

The DC component,a_0 can be found from

$$a_0 = \frac{1}{2L}\int_{-L}^{L} f(x)dx$$

$$a_0 = \frac{1}{2\pi}\int_{0}^{2\pi} f(x)dx$$

$$a_0 = \frac{1}{2\pi}[\int_{0}^{\pi} 0dx + \int_{\pi}^{2\pi} 1dx]$$

$$a_0 = \frac{1}{2}$$

The Cosine co-efficient $\mathrm{a_n}$, can be found from,

$$a_n = \frac{1}{L}\int_{-L}^{L} f(x)\cos(\frac{n\pi x}{L})dx$$

$$a_n = \frac{1}{\pi}\int_{0}^{2\pi} f(x)\cos(nx)dx$$

$$a_n = \frac{1}{\pi}\left[\int_{0}^{\pi} 0\cos(nx)dx + \int_{\pi}^{2\pi} 1\cos(nx)dx\right]$$

$$a_n = \frac{1}{\pi}\int_{\pi}^{2\pi} \cos(nx)dx$$

$$a_n = \left[\frac{1}{n\pi}\sin(nx)\right]_{\pi}^{2\pi}$$

$$a_n = 0$$

The Sine co-efficient b_n, can be found from,

$$b_n = \frac{1}{L}\int_{-L}^{L} f(x)\sin(\frac{n\pi x}{L})dx$$

$$b_n = \frac{1}{\pi}\int_{0}^{2\pi} f(x)\sin(nx)dx$$

$$b_n = \frac{1}{\pi}\left[\int_{0}^{\pi} 0\sin(nx)dx + \int_{\pi}^{2\pi} 1\sin(nx)dx\right]$$

$$b_n = \frac{1}{\pi}\int_{\pi}^{2\pi} \sin(nx)dx$$

$$b_n = \left[\frac{-1}{n\pi}\cos(nx)\right]_{\pi}^{2\pi}$$

$$b_n = \frac{-1}{n\pi}\cos(2n\pi) + \frac{1}{n\pi}\cos(n\pi)$$

If n is odd,

$$b_n = \frac{-2}{n\pi}$$

If n is even,

$$b_n = 0$$

So the waveform approximation is,

$$f(x) = \frac{1}{2} + \sum_{n=1}^{\infty} \frac{-2}{n\pi}\sin(nx), \text{ where n is 1, 3, 5, ...}$$

This implies that only the odd harmonics of the fundamental frequency are present.

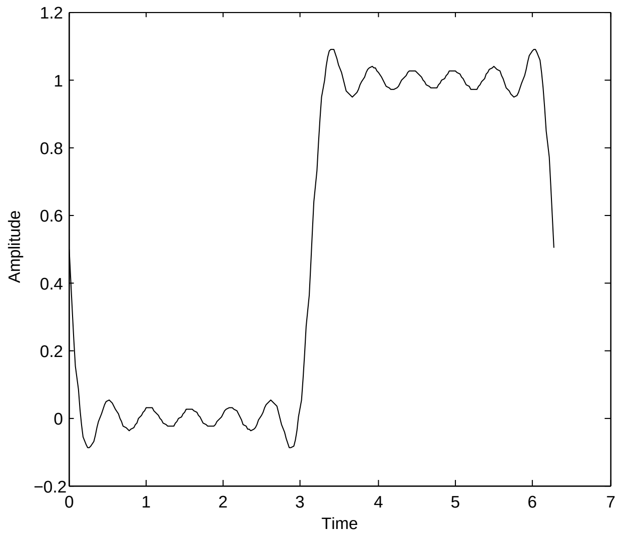

Figure 5.1: 11 Component Squarewave constructed using Matlab

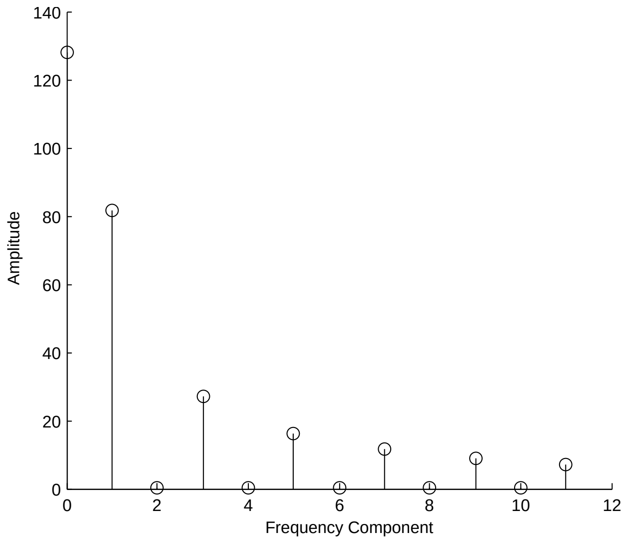

Figure 5.2: Fourier Transform of 11 Component Squarewave

Approximation of a square-wave with a period of 2π from only eleven harmonics is shown in Figure 5.1. These harmonics can be seen in Figure 5.2, which is the Fourier transform of the approximated square-wave.

5.4 Fourier Analysis of a Noise Corrupted Signal

Consider the noise corrupted signal in the following figure,

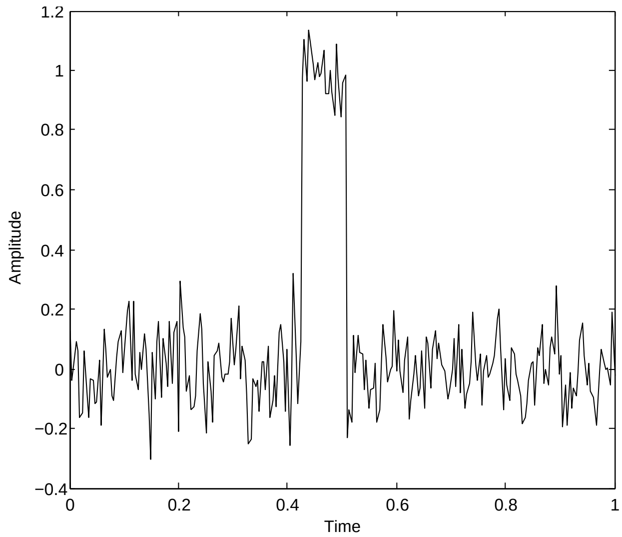

Figure 5.3: Noise Corrupted Distance Measurement

The desired signal would be,

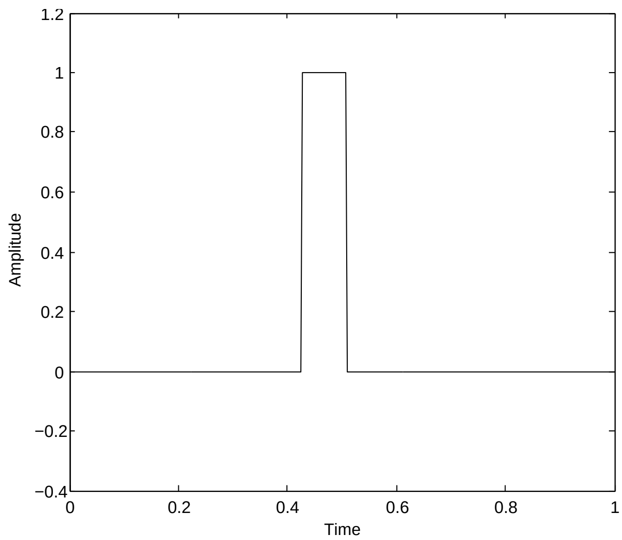

Figure 5.4: Desired Signal

To attempt to eliminate the noise from the signal, the type of noise must first be determined.

5.5 Additive White Gaussian Noise

Additive White Gaussian Noise (AWGN) is defined as any random digital signal sequence with zero mean, in which adjacent sample values are completely uncorrelated (Lynn & Fuerst 1999). Autocorrelation of an AWGN signal is zero at all lags except for a lag of zero (Leis 2002). However, for a finite sample, the non-zero lags will not be zero, but rather a value close to zero due to the effect of working with a finite length sequence (Lynn & Fuerst 1999).

A pure noise sample from the output of the lasers, shown in Figure 5.5, can be used to determine if the noise in the signal is in fact Additive White Gaussian Noise.

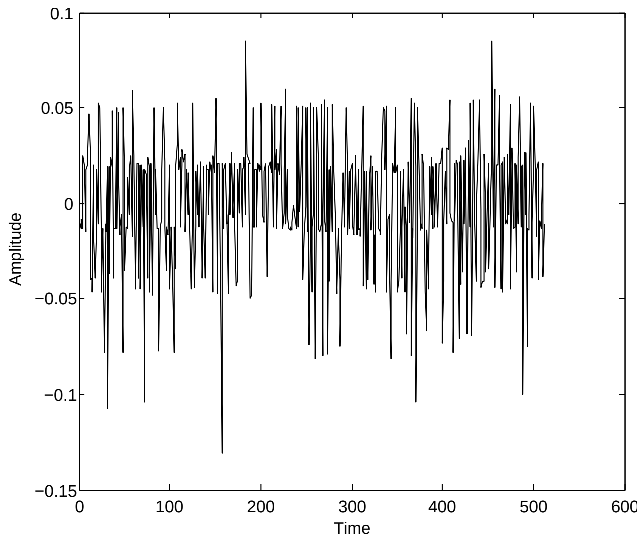

Figure 5.5: Pure Noise Sample

The autocorrelation of the noise sample is shown in Figure 5.6. Here is can be seen that the autocorrelation is approximately zero at all lags except for a lag of zero. The reason for the autocorrelation not being exactly zero is due to the finite size of the sample.

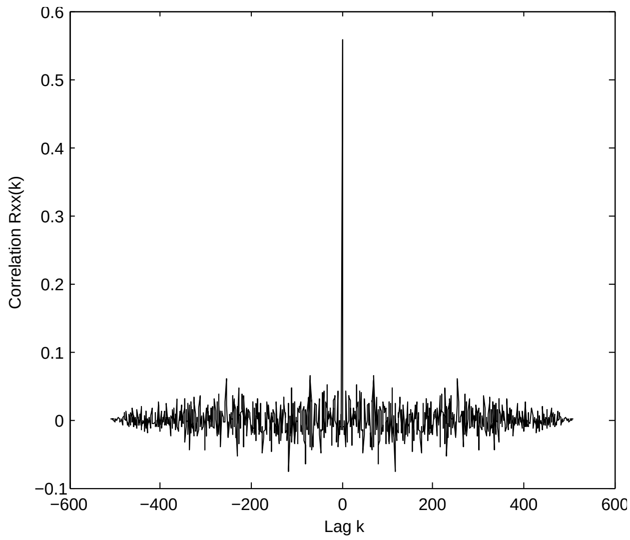

Figure 5.6: Auto Correlation of Pure Noise Sample

The second requirement in determination of the type of the noise is that the noise is ‘white’. The term ‘white’ implies that the noise signal has equal power across the total bandwidth of the spectrum. A Fourier transform of the noise shows the frequency spectrum of the signal in Figure 5.7.

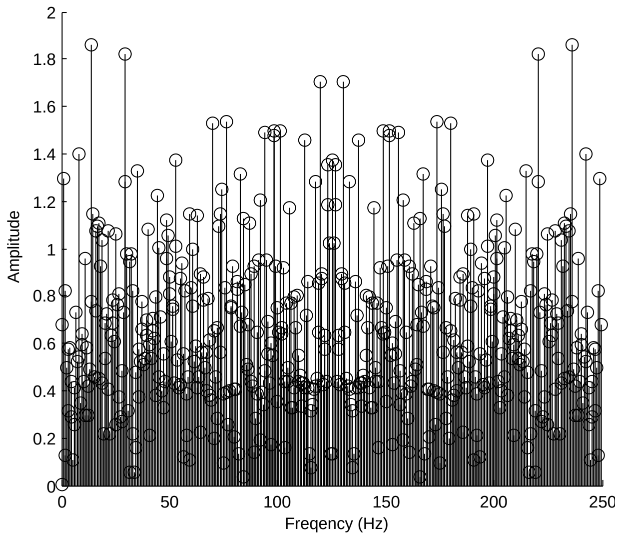

Figure 5.7: Fourier Transfrom of Noisy Signal

Figure 5.7 shows that there are frequencies present across the entire frequency bandwidth of the signal. It should be noted that in this figure, only the lower 125 Hz is reliably represented. The upper 125 Hz is simply a mirror image on the lower half. This is due to the symmetry provided by the Fourier transform.

From this analysis of the noisy sample, it can be assumed that the noise in the output of the signals is Additive White Gaussian Noise.

5.6 Elimination of Noise

Using the previous idea of Fourier analysis of a signal, it can be determined that the desired signal plus noise is made up of many sine and cosine components. It is possible to determine a frequency above which the contributions of the higher frequency components is minimal, then the desired signal can be approximated from the lower frequencies only. There are a number of techniques used in signal processing to reduce the noise present in a signal. The three techniques presented in this thesis are Low Pass Filters, frequency domain thresholding, using Fast Fourier Transform and Inverse Fast Fourier Transform and finally, Wavelet Denoising.

5.7 Low Pass Filter

Low pass filters are used to extract signal information from noisy signals such as the example already presented. A low-pass filter is a filter that passes low frequencies well, but attenuates frequencies higher than the cut-off frequency. First a desired filter response is determined in the frequency domain, and then a filter is designed to approximate the ideal filter.

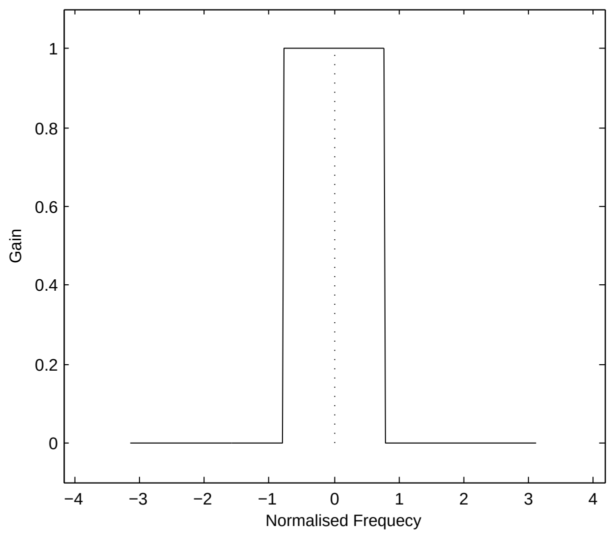

Figure 5.8: Theoretical Low Pass Filter

For the filter in Figure 5.8, the signal bandwidth has been normalised to $\frac{\pi}{2}$ radians. The signal bandwidth is determined by the Nyquist frequency. The Nyquist frequency is defined as the highest frequency component that may be accurately recovered from a digitally sampled signal (Lynn & Fuerst 1999). The bandwidth of the signal is half of the sampling frequency. The corner frequency is $\frac{Fs}{4}$ which equates to 62.5 Hz.
The actual filter response can be calculated from the following equation, (Leis 2002),

$$h_d(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(\omega) e^{j\omega n} d\omega$$

This impulse response calculated from this filter would be infinite. In practise, the filter order must be limited. Assuming the order of the filter to be odd, the magnitude response can be calculated using the following limitation (Leis 2002).

$$\frac{-(N-1)}{2} \leq n \leq \frac{(N-1)}{2}$$

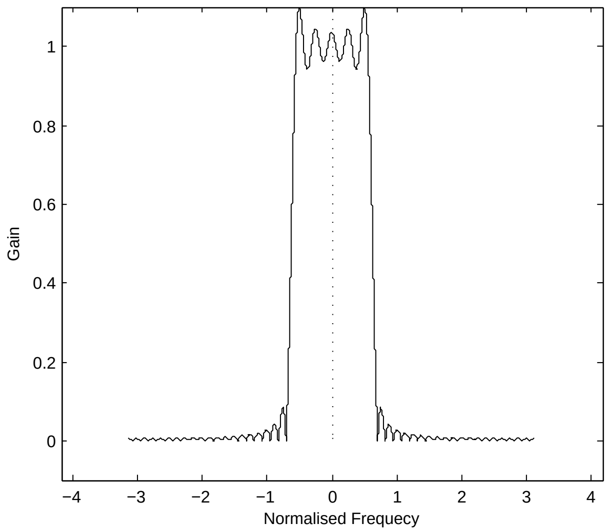

Figure 5.9: 51 Order Filter Response

Using a filter order of 51, the a magnitude response is displayed in Figure 5.9. Here there exist significant ringing in the pass band and stop bands.

The response of the filter may be further improved by the application of a hamming window. Using a Hamming window greatly reduces the ringing. The mathematical definition of a hamming window is (Leis 2002),

$$\omega_n = \left| {}^{0.54-\cos(\frac{2n\pi}{M}):0\leq n\leq M}_{0:Otherwise} \right.$$

The magnitude response, displayed in Figure 5.10, now becomes.

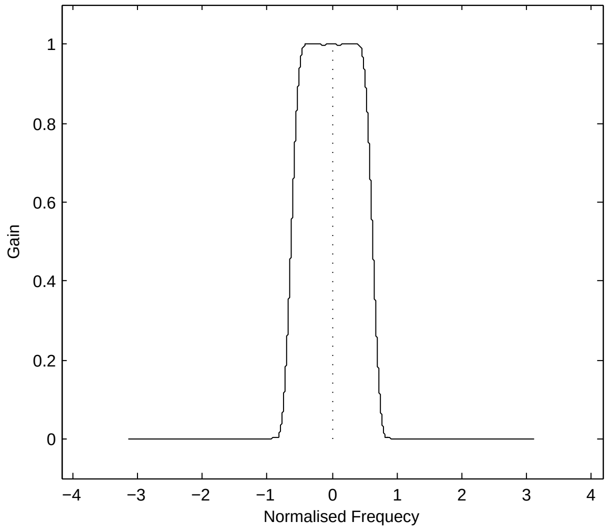

Figure 5.10: Actual Filter Response

By taking the Fourier Transform of the signal shown in Figure 5.3, determining the frequency below which the signal is buried in noise, and applying a low pass filter at this cut off frequency, a smoother approximation to the desired signal can be recovered.

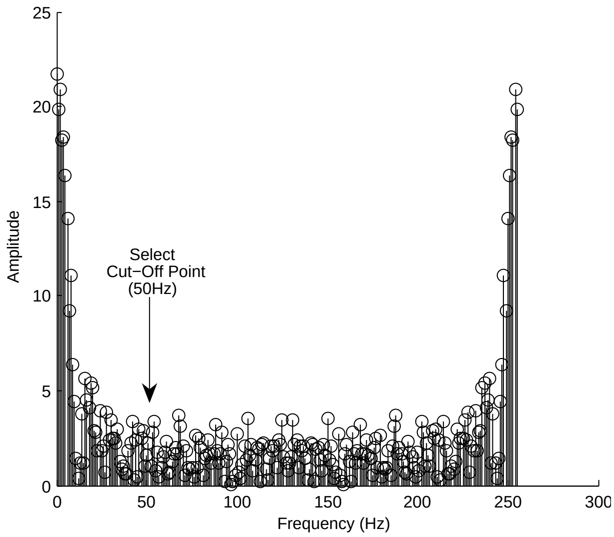

Figure 5.11: Fourier Transform of the Noisy Signal

After application of the low pass filter, it is possible to see that the frequencies higher than the cut off frequency have been removed from the signal, as seen in Figure 5.12.

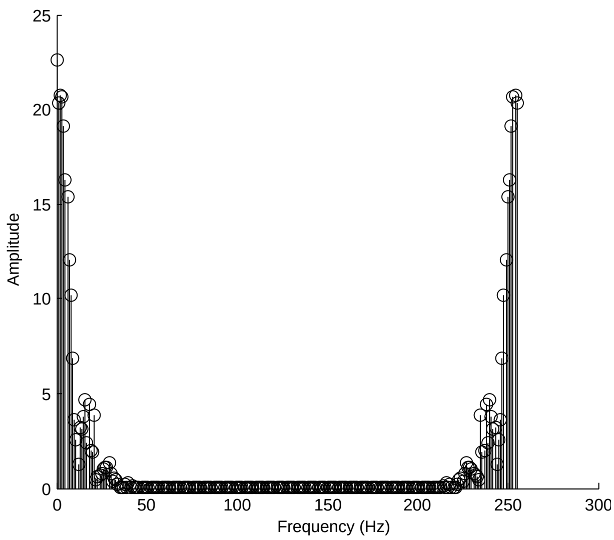

Figure 5.12: Fourier Transform of the Filtered Signal

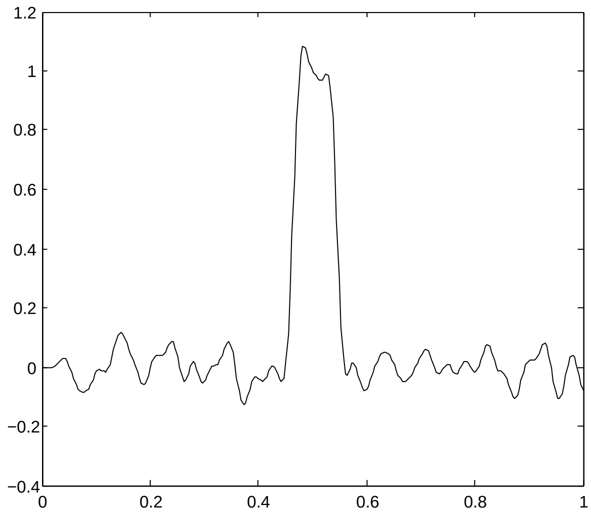

Figure 5.13: Filtered Signal

The result of filtering the signal is shown in Figure 5.13.

5.8 Frequency Domain Thresholding

Another method of filtering specific frequencies from the original noisy signal is known as Frequency Domain Thresholding. In order to remove specific frequencies, the signal is first transformed into the frequency domain using the Discrete Fourier Transform, which is the discretized version of the mathematical equation set out in Equation 5.6. The equation for the discrete Fourier transform is (Leis 2002),

$$X(k) = \sum_{n=0} {}^{N} - 1x(n)e^{-jn\omega_k}$$

where,

$$\omega_k = \frac{2\pi k}{N}$$

Which is the frequency of the k^{th} sinusoid, and N is the number of samples being examined in the window.

Consider Figure 5.14, a 2 hertz sine wave with added white gaussian noise.

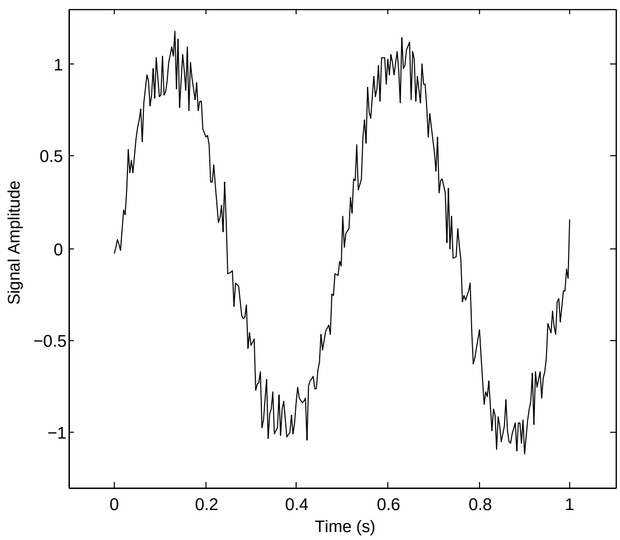

Figure 5.14: 2 Hz sine wave with AWGN

Taking the Fourier Transform of the signal,

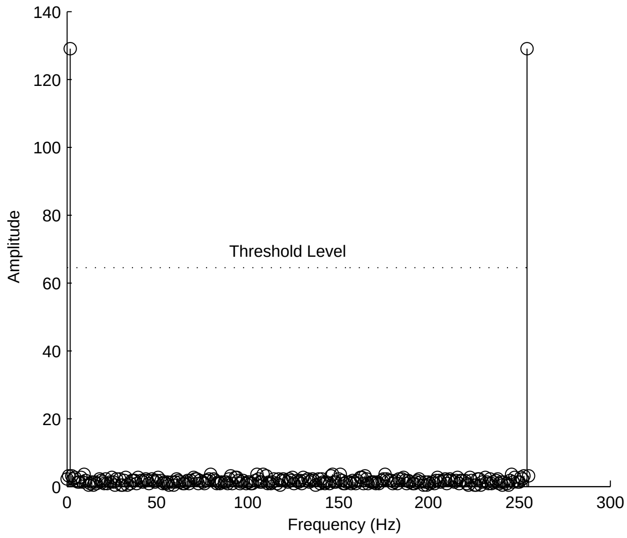

Figure 5.15: Fourier Transform of single frequency signal plus noise

A threshold level is set at 50 percent of the strongest frequency component present in the signal. All frequency components, whose strength is lower than this value, are set to zero.

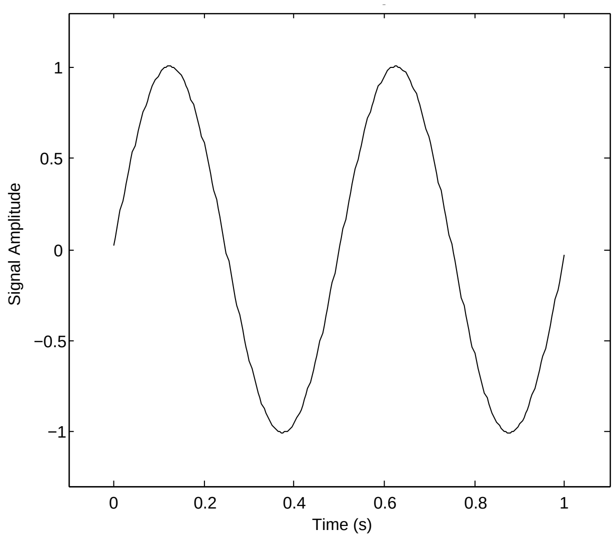

Figure 5.16: Recovered Signal using 50% threshold.

Then taking the inverse Fourier transform will remove the higher frequency components of noise. The original signal may be recovered, as in Figure 5.16.

If the same theory is applied to a signal containing not just a single frequency, but many Fourier components of the signal plus a substantial quantity of additive white gaussian noise, as would be expected from the output of the laser sensors, it is possible to recover a smoothed signal. The signal plus noise is represented in Figure 5.17.

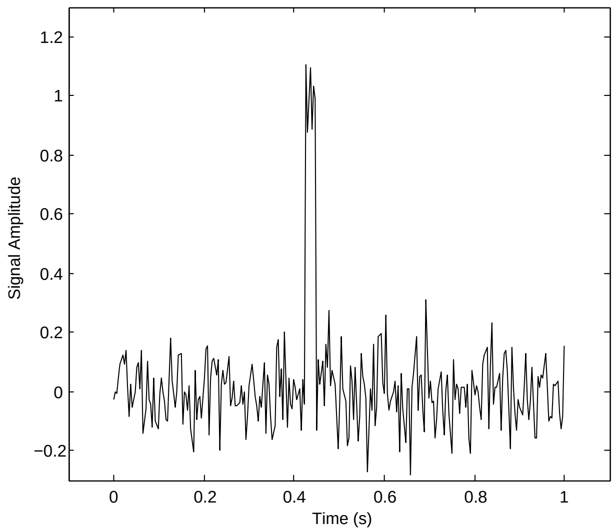

Figure 5.17: Sensor Signal

The threshold level in the frequency domain is set at 50% of the amplitude of the strongest component and is shown in Figure 5.18

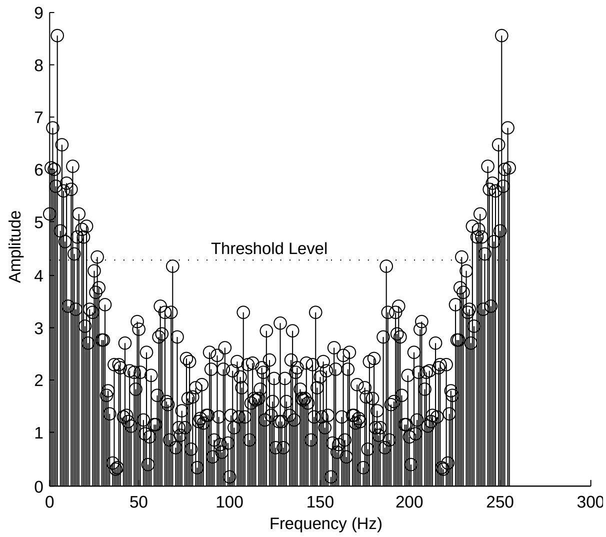

Figure 5.18: Fourier Transform of Noisy Signal Showing Threshold Level

The recovered signal is shown in Figure 5.19,

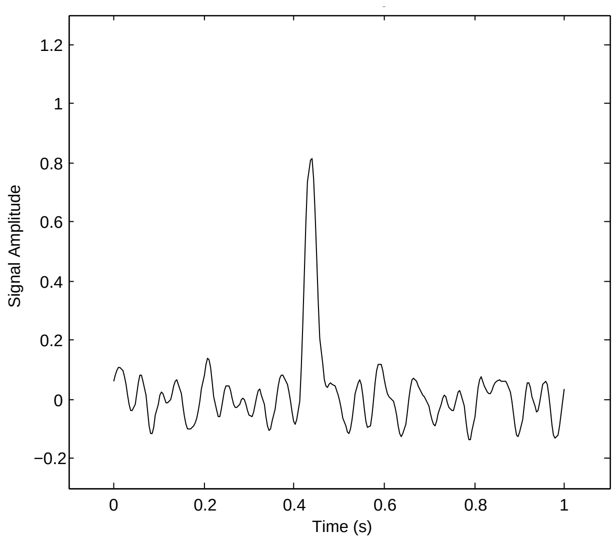

Figure 5.19: Recovered Signal using IFFT

5.9 Summary

From the material presented in this chapter, it can be seen that application of low pass filtering and frequency domain thresholding are both effective methods to reduce noise present in the signal. In Chapter 7, these two methods will be compared against the method presented in Chapter 6, Wavelet Denoising, to determine which method is the most effective technique to reduce the noise present in the signal.

Chapter 6

Wavelets and Wavelet Denoising

6.1 Background

One of the most useful applications of wavelet analysis is to remove unwanted noise from a dataset. While the previous chapter looked at removing noise with the use of a filter, there is a problem associated with this particular method. The shape and width of the filter needs to be chosen carefully to avoid removing too much of the desired signal and also to decrease ringing in the peaks. Another problem is that the noise is Additive White Gaussian Noise, which is distributed across all frequencies randomly.

Fourier transform converts a signal from the time domain into the frequency domain. Application of Fourier transform implies that the signal is stationary, or not changing in time. A signal is stationary if its properties are statistically invariant over time (Meyers 1993). However the signal studied in the project is absolutely changing in time and is therefore non-stationary. The other disadvantage of Fourier analysis is that while the Fourier transform provides information about what frequencies are present, it does not provide information about when the frequencies were present. It will only suffice to say the frequency appeared somewhere within the window of time that was examined.

Wavelet analysis offers a scale-independent and robust method to filter out noise from a non-stationary signal (*Wavelet Basics* 2006). The information provided by a Wavelet Transform retains both the time information, as well as the frequency information. Also, as different families of wavelets have different properties, before any denoising can take place the choice of mother wavelet must be considered.

6.2 WaveletTransform

6.2.1 Continuous Wavelet Transform

A wavelet transform maps a time function into a two-dimensional function of a and t (Chui 1995). The two parameters, a and t, represent scale and translation respectively. Scaling a function is equivalent to compressing or stretching the function, and translation of a function is equivalent to shifting the function in time, or relocating it somewhere on the time axis. Scale may be related to frequency, and translation may be related to time.

Using this, the continuous wavelet transform (CWT) may be mathematically defined as (Chui 1995),

$$CWT(a,t) = \frac{1}{\sqrt{a}} \int s(t)\Psi\left(\frac{t-\tau}{a}\right) dt$$

where Ψ (t) is the mother wavelet.

From the application of the CWT, there is a need to be able to reverse the transform to reconstruct the original signal. If the mother wavelet, Ψ(t) is invertible, then the function may be recovered from (Chui 1995),

$$s(t) = \frac{1}{c_\Psi} \int_{-\infty}^{\infty} CWT(a,t) \frac{1}{\sqrt{a}} \Psi\left(\frac{t-\tau}{a}\right) \frac{1}{a^2} da\ dt$$

where c_Ψ is a constant that depends only on $\Psi(t)$ and a is positive.

The constant has value (Chui 1995),

$$c_\Psi = \int_0^\infty \frac{|\Psi(\omega)|^2}{\omega} d\omega < \infty$$

which in turn places an admissibility condition on Ψ(t). For $c_\Psi < \infty$, $\Psi(t)$ must be such that,

$$|\Psi(\omega)| < \infty, \text{ for any } \omega,$$

and Ψ(0)=0, implying that,

$$\int \Psi(t)\ dt = 0$$

6.2.2 Wavelet decomposition using Mallet algorithm

It is impossible to compute the continuous wavelet transform of a signal, due to the fact that the calculation of this information would also require an infinite amount of data. Digital signal processing also implies that the signal is discrete, there is no data available between sampling points. To calculate the wavelet transform of a discrete signal requires discrete scales and translations based on power of two, dyadic scale and translation.

In 1988, an efficient way to implement this scheme using filters was developed by Mallat (Mallet 1989). This application of this filtering algorithm effectively produces a fast wavelet transform.

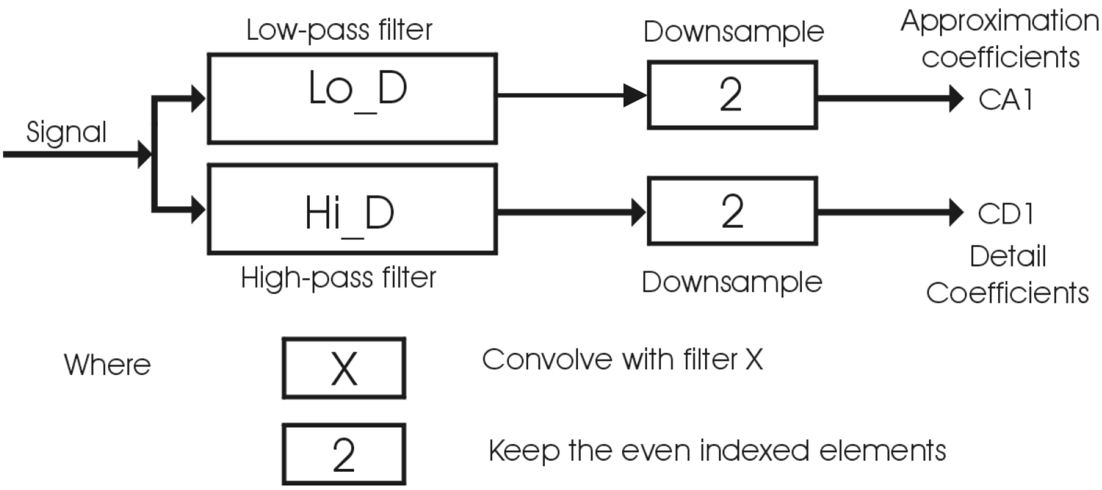

Figure 6.1: Matlab Discrete Wavelet Transform Algorithm

Figure 6.1 shows how Matlab accomplishes a discrete wavelet transform at one level of decomposition. The signal is convolved with both a high pass and a low pass filter. This convolution of the signal with the corresponding filter produces an array of coefficients twice the size of the original signal. Downsampling is performed upon each array of coefficients, keeping only the even numbered coefficients. Following downsampling, the number of coefficients in the combined arrays is equal to the number of samples within the original signal. The signal is now separated into 2 discrete frequency bands.

CA represents the Approximation Coefficients, the lower frequency coefficients, and CD represents the Detail Coefficients, the higher frequency coefficients. This is termed, "One Level of Decomposition", as the signal has been decomposed once. Further decomposition is possible by dividing the lower half frequency window into two more windows as in Figure 6.2.

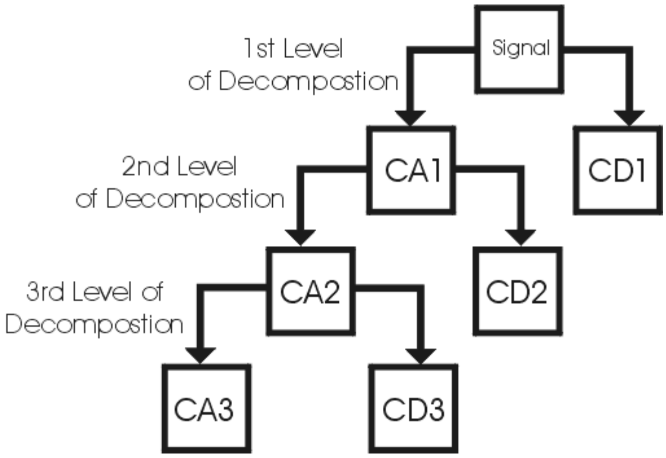

Figure 6.2: 3 Levels of Discrete Wavelet Transform Decomposition

It is possible to continue dividing each lower window until only a single coefficient remains. The required number of levels of decomposition can be reduced by careful choice of mother wavelet, and will be discussed further in the next section.

6.3 Wavelet Families

There are a number of wavelets that are useful in processing signals. The particular properties of a wavelet can improve the ability to determine specific features of the signal being processed. The strength of Wavelet Transform representations is that functions or signals that have similar features to the wavelet function at any scale may be well represented by only a few of the wavelet basis functions (*Wavelet Basics* 2006). Based on this, finding a wavelet with similar properties to that which is expected in the signal should yield the best possible noise reduction via Wavelet Denoising. For comparison, two wavelets are presented.

6.3.1 Haar Wavelet

The first and simplest of all wavelets is the Haar Wavelet. The Haar wavelet is discontinuous, and resembles a step function. It represents the same wavelet as Daubechies1.

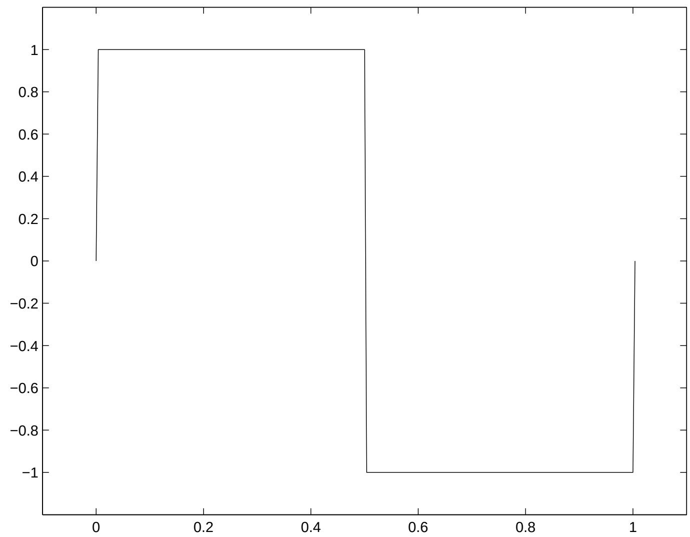

Figure 6.3: Haar 1 Mother Wavelet

The Haar wavelet depicted in Figure 6.3 can be mathematically defined as:-

$$\Psi(x) = 1;\ 0 \leq x \leq 0.5$$

$$\Psi(x) = -1;\ 0.5 \leq x \leq 1$$

$$\Psi(x) = 0;\ \text{Otherwise}$$

This will be the mother wavelet applied to the noisy signal for the project. The ideal, noise free signal has very square edges, where the signal goes from no axle present to an axle present at a distance from the laser.

6.3.2 Daubechies

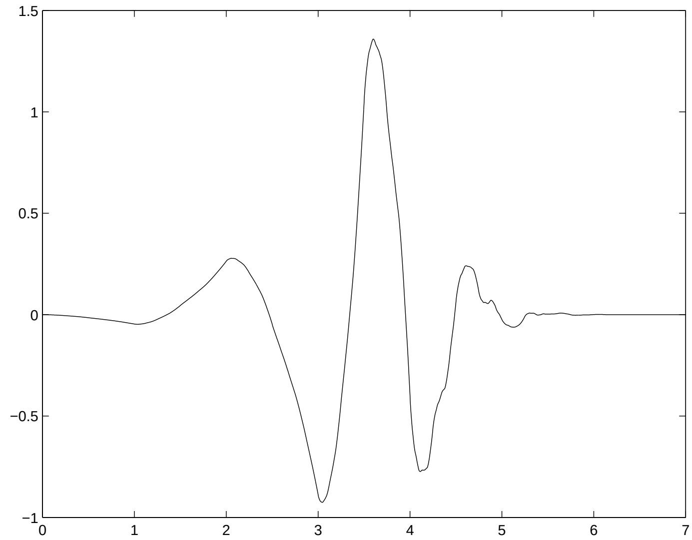

Figure 6.4: Daubechies 4 Mother Wavelet

For the Daubechies family of wavelets, the wavelet function (Mother Wavelet), is orthogonal to all functions which are obtained by shifting the mother right or left by an integer amount. Also, the mother wavelet is orthogonal to all functions which are obtained by dilating (stretching) the mother by a factor of 2^j and shifting by multiples of 2^j units (*Wavelet Basics* 2006). Figure 6.4 represents the Daubechies 4 Wavelet.

6.4 Wavelet Denoising

The basic technique of denoising a signal involves computing the discrete wavelet transform of the signal and then decreasing or discarding the smallest wavelet coefficients. The inverse transform of these coefficients will then be a filtered version of the signal.

Denoising a signal can be ordered into a specific sequence of three events (*Matlab Helpfiles* 2006).

1. Decompose:- Choose a wavelet, and choose a level N. Compute the wavelet decomposition of the signal s at level N.

2. Detail coefficients thresholding:- For each level from 1 to N, select a threshold and apply thresholding to the detail coefficients.

3. Reconstruction:- Compute wavelet reconstruction based on the original approximation coefficients of level N and the modified detail coefficients of levels from 1 to N.

6.4.1 Decomposition

Decomposition of the signal at successive levels involves computing the approximation coefficients and detail coefficients using the selected mother wavelet. This is achieved using the Mallat alogorithm implemented in Matlab. The Mallet algorithm is a two-channel subband coder (*Matlab Helpfiles* 2006).

As seen in Figure 6.5, the signal originally consists of 1000 samples. The signal is then passed through the high pass and low pass filters corresponding to the mother wavelet. Each filtered signal now contains 1000 samples giving a total of 2000 samples. If the signal is then downsampled, or taking every second sample and discarding intermediate samples, combining the two channels, the signal now consists of 1000 samples.

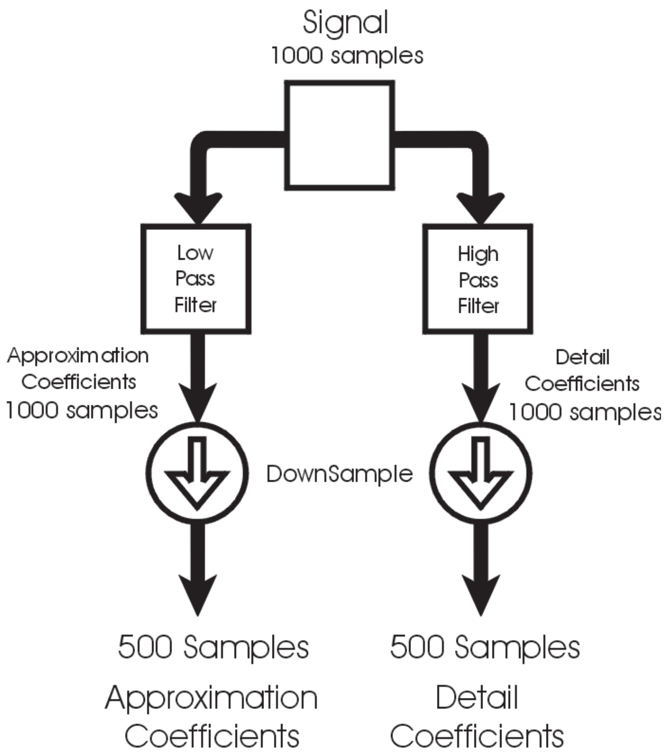

Figure 6.5: Mallat Algorithm

After the required operations have been performed upon the coefficients, the reverse process is applied. Here, the coefficients are upsampled, zeros inserted between samples and applied to the corresponding filters to give reconstruction.

6.4.2 Detail Coefficient Thresholding

As stated earlier, there are two types of thresholding that can be applied, hard and soft. Hard thresholding implies that any detail coefficient below a certain level be set to zero. Soft thresholding has the same criterion, but also reduces the remaining non-zero coefficients by an amount equal to the largest discarded coefficient (*Matlab Helpfiles* 2006).

If W_i is the wavelet coefficient, and W_0 is the chosen threshold,

For Hard Thresholding,

$$W_i = \begin{vmatrix} W_i: & |W_i|>W_0 \\ 0 \; : & |W_i|\leq W_0 \end{vmatrix}$$

For Soft Thresholding,

$$W_i = \begin{vmatrix} sign(W_i)(|W_i|-W_0) : & |W_i|>W_0 \\ 0 \qquad\qquad : & |W_i|\leq W_0 \end{vmatrix}$$

Where sign(W_i) is the sign of the wavelet coefficient.

6.5 Application of Denoising using Discrete Wavelet Transform

Considering the theoretical signal depicted in Figure 6.6.

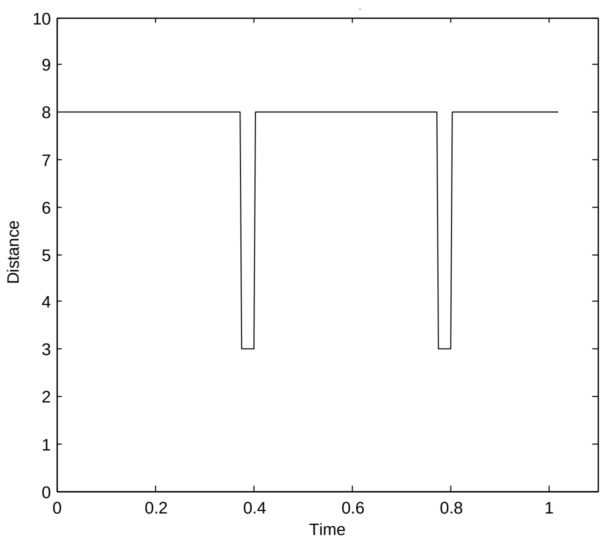

Figure 6.6: Desired Signal

This is the signal that would be ideally required. However, suppose the signal is corrupted by noise as is the signal in Figure 6.7.

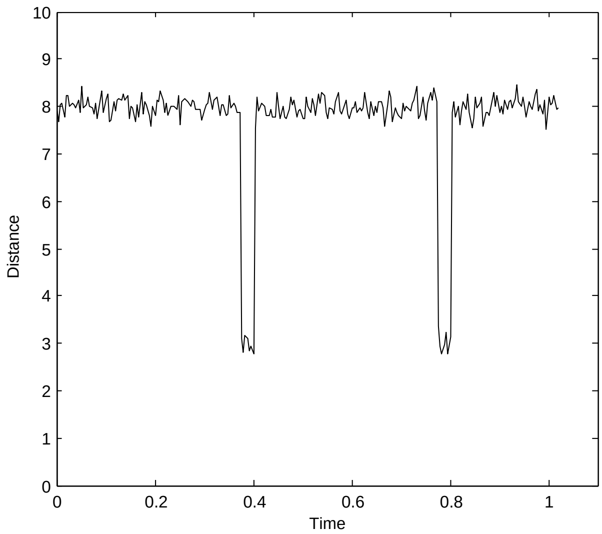

Figure 6.7: Noise Corrupted Signal

Applying the theory discussed in the previous section regarding denoising, where the signal is decomposed using a suitable wavelet, such as the Haar mother wavelet, apply hard thresholding to the detail coefficients, and then reconstruct the signal using the inverse discrete wavelet transform. The effect of multiple levels of decomposition, thresholding and finally reconstruction can be observed through Figures 6.8 to 6.13

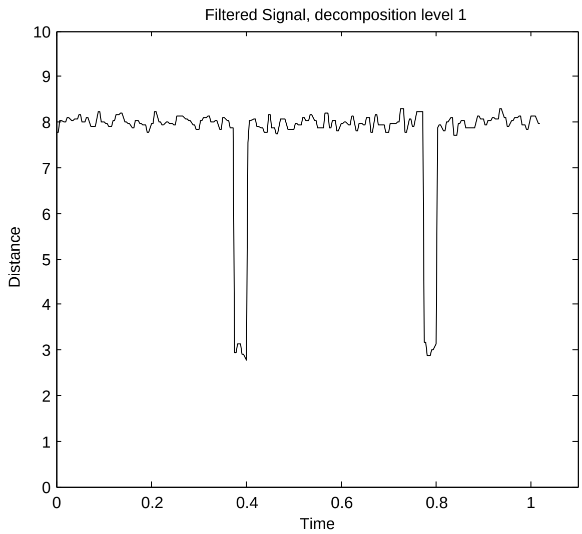

Figure 6.8: Haar Wavelet Decomposition 1 Level

In Figure 6.8, the signal has been decomposed only one level, the detail coefficients thresholded with the hard method, and then been reconstructed. There is significant reduction in the noise present in the signal. A further level of decomposition yields more noise removal, as seen in Figure 6.9.

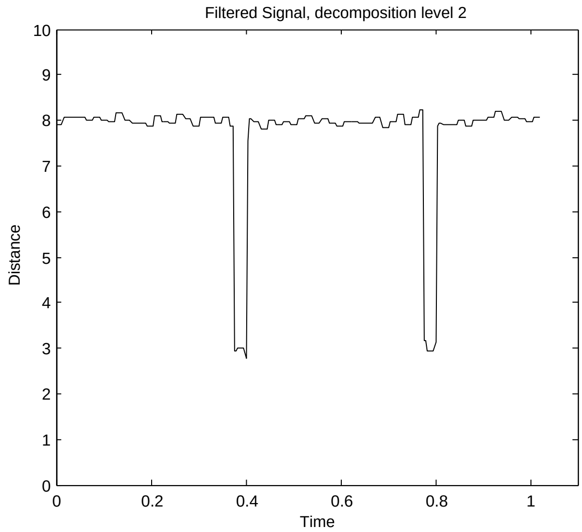

Figure 6.9: Haar Wavelet Decomposition 2 Levels

Figure 6.10 shows reconstructed signal after three levels of decomposition with hard thresholding.

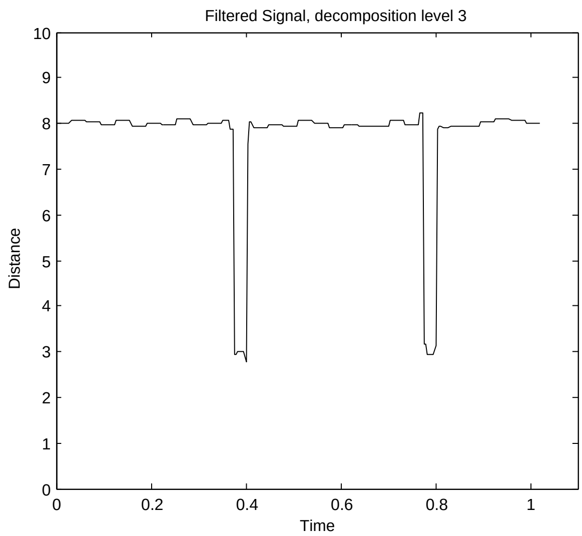

Figure 6.10: Haar Wavelet Decomposition 3 Levels

Figures 6.11, 6.12 and 6.13 show the reconstructed signal after being denoised with four, five and six levels of decomposition respectively.

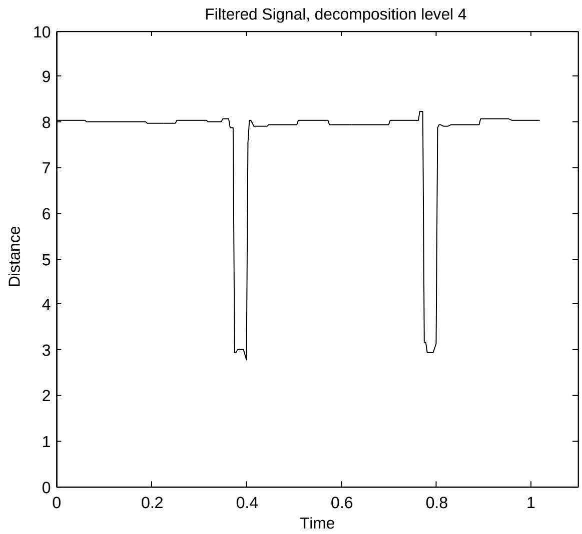

Figure 6.11: Haar Wavelet Decomposition 4 Levels

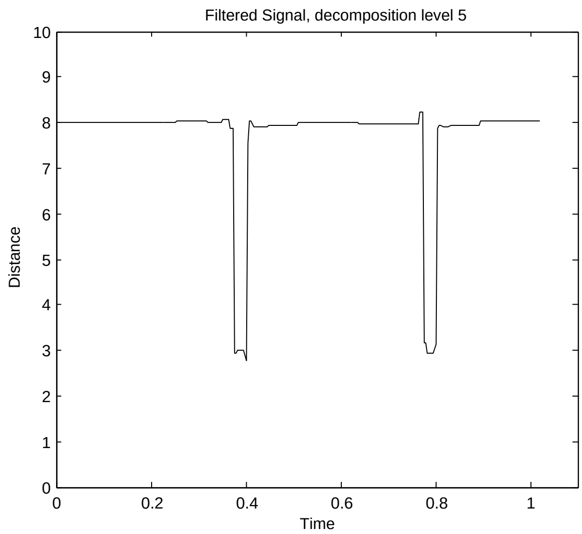

Figure 6.12: Haar Wavelet Decomposition 5 Levels

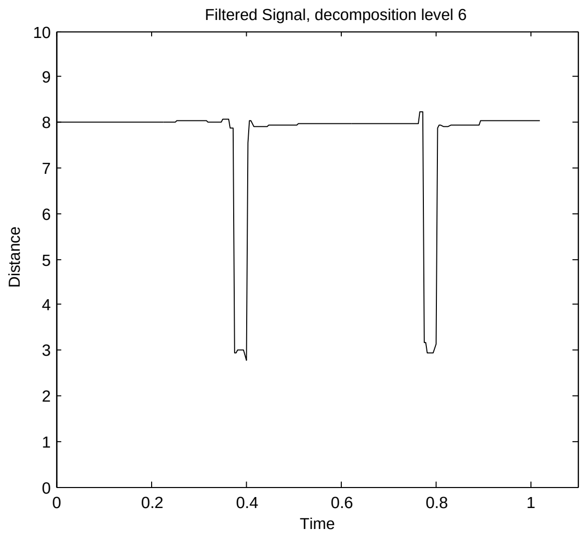

Figure 6.13: Haar Wavelet Decomposition 6 Levels

6.6 Summary

From the work presented in this chapter, it can be seen that Wavelet denoising provides a very effective method to remove noise from a non-stationary signal. In Chapter 7, the Wavelet denoising method will be compared to the methods examined in Chapter 5, Low Pass Filtering and Frequency Domain Thresholding.

Chapter 7

Comparison of Denoising Techniques

7.1 Noise Reduction Ratio

To devise a method by which the various noise removal techniques can be compared, some assumptions must be made. If the waveform is considered as a voltage waveform and an imaginary load introduced, the concept of Root Mean Squared (RMS) can be considered. The model is shown in Figure 7.1.

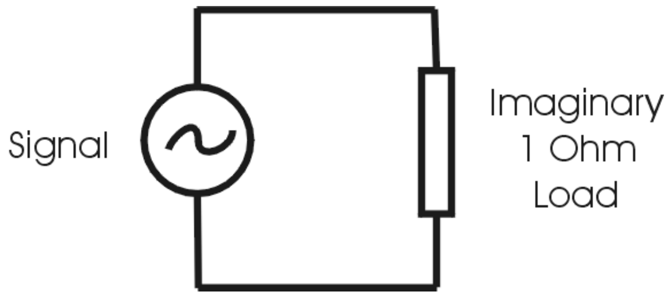

Figure 7.1: Circuit Model of Signal

The steps for finding RMS are:-

- Square the waveform
- Calculate the mean of the waveform
- Take the square root of the result

Since the imaginary load is considered as one ohm, the square of the RMS of the voltage waveform is equivalent to the RMS of the power.

$$\text{Power} = \frac{\text{Voltage}^2}{\text{Resistance}}$$

The signal is one that has been constructed in Matlab, and the noise has been generated and added to the signal, the original noise content of the signal is known. From this, the RMS of the noise power can be calculated.

Also, as the original signal is known, it can be subtracted from the denoised signal.

Any quantity of signal remaining can be assumed to be unwanted noise in the signal. The RMS of the noise power remaining in the signal can then be calculated.

A decibel ratio can then be determined to give an indication of the level of noise reduction provide by each individual technique. The ratio is calculated from,

$$\text{Noise Ratio} = 10\ \log\left(\frac{\text{Noise Power in Denoised Signal}}{\text{Original Noise Power}}\right)$$

7.2 Comparison

Figure 7.2 represents the desired signal, with additive white gaussian noise. This is the signal to which the three techniques of denoising are applied.

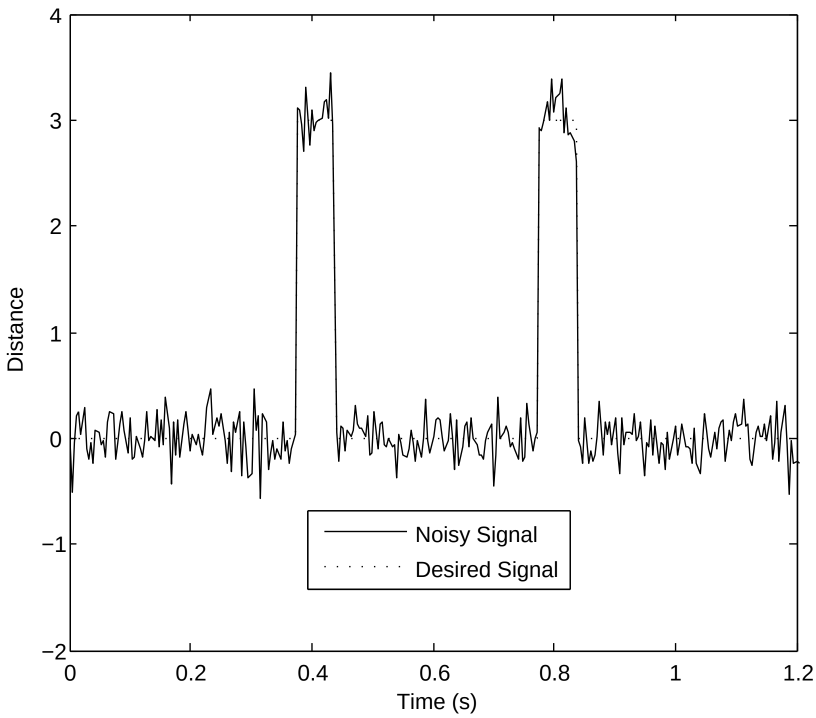

Figure 7.2: Theoretical Signal plus Noise and Desired Signal

The RMS of the noise power contained in the noisy signal is 0.032893 Watts.

Figure 7.3 shows the signal after being denoised by a Low Pass Filter, order of 51, and corner frequency of 50 Hz. The RMS of the noise power is 0.020677. The Noise Reduction Ratio is calculated as -4.0325 dB.

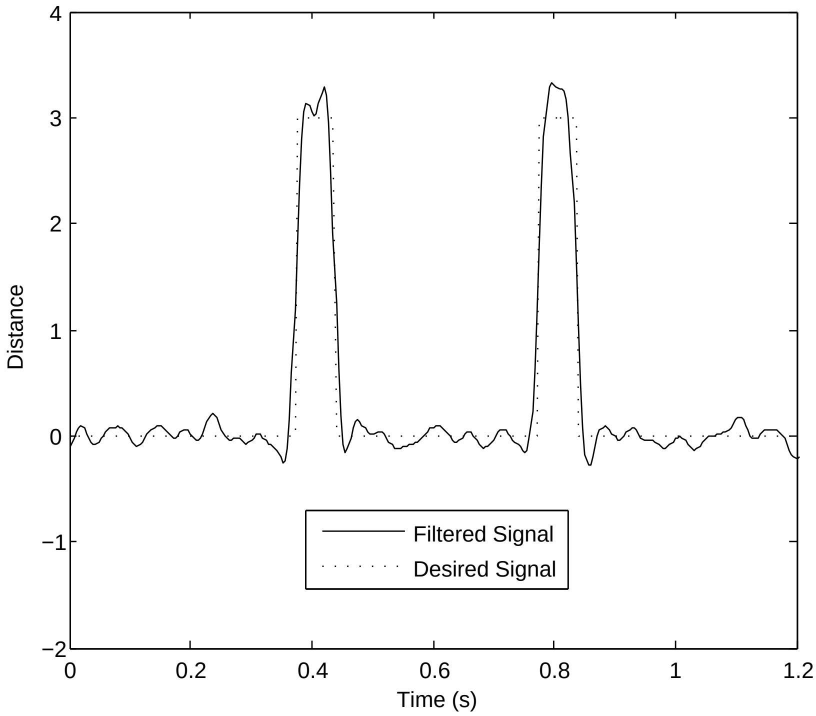

Figure 7.3: Low Pass Filtered Signal and Desired Signal

Figure 7.4 shows the signal after being denoised with a noise floor threshold level of 20%. The RMS of the noise power is 0.031533. The Noise Reduction Ratio is calculated as -0.3668 dB.

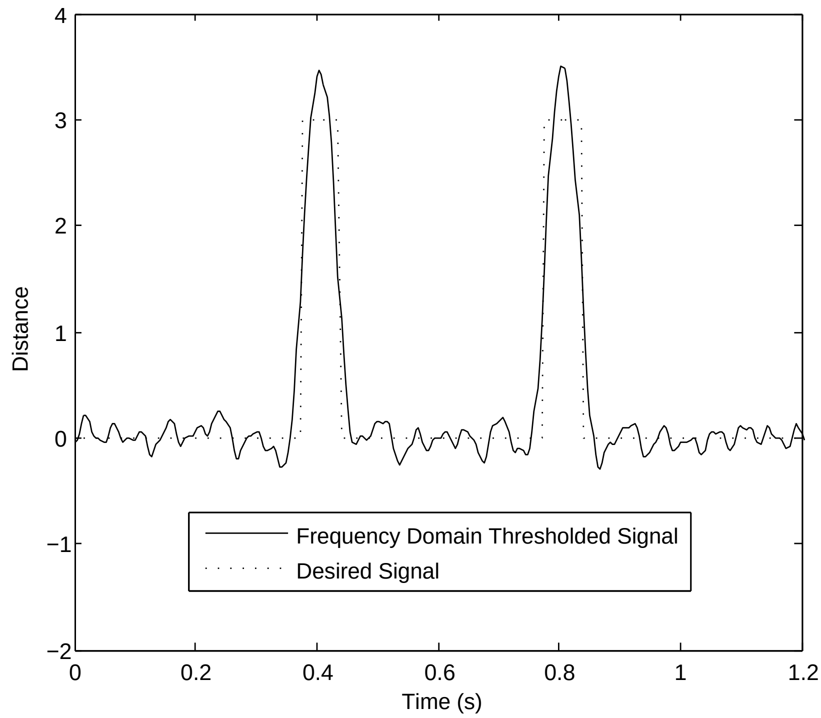

Figure 7.4: Frequency Domain Thresholded Signal and Desired Signal

Figure 7.5 shows the signal after being denoised using a Haar mother wavelet, with four levels of decomposition and hard thresholding. The RMS of the noise power is 0.0086246. The Noise Reduction Ratio is calculated as -11.6273 dB.

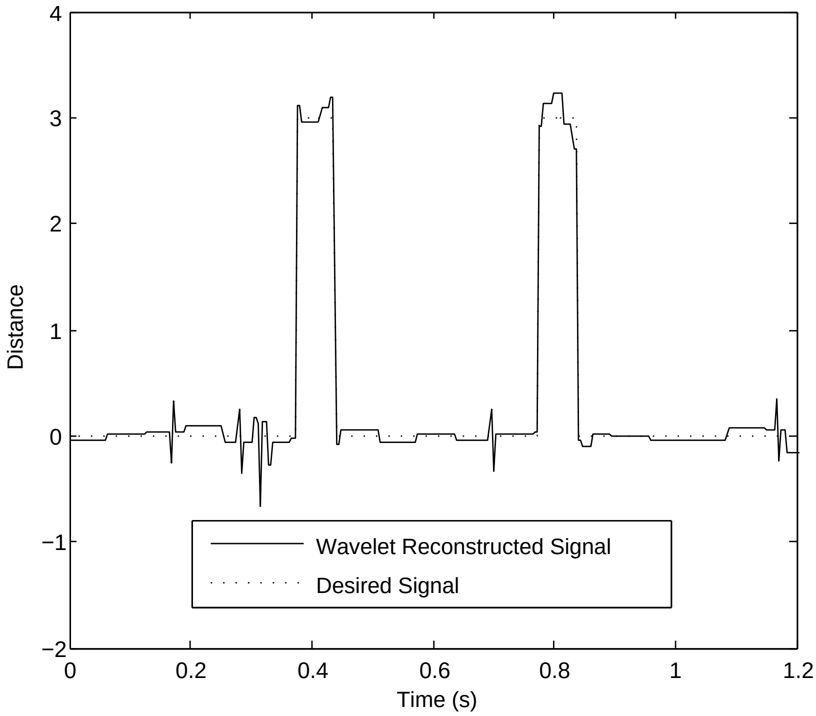

Figure 7.5: Haar Wavelet Denoised Signal and Desired Signal

By comparison of the Noise Reduction Ratios, it can be concluded that the Haar Wavelet denoising technique offers the best noise reduction for this particular application. However, there is another advantage that the Haar wavelet method has over the other two techniques.

The correct speed and wheel base of vehicles is paramount in the design of the required system. The axle detection algorithm is based upon finding the leading edges of the wheels. Any denoising technique must retain the edges of the wheels as close as possible to the original signal.

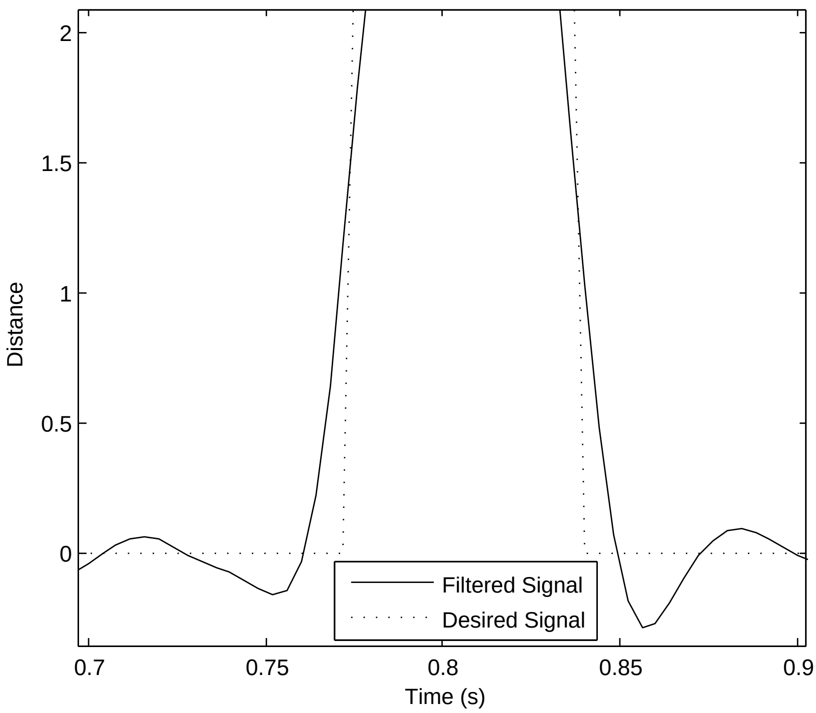

Figure 7.6: Zoomed window of Low Pass Filtered Signal

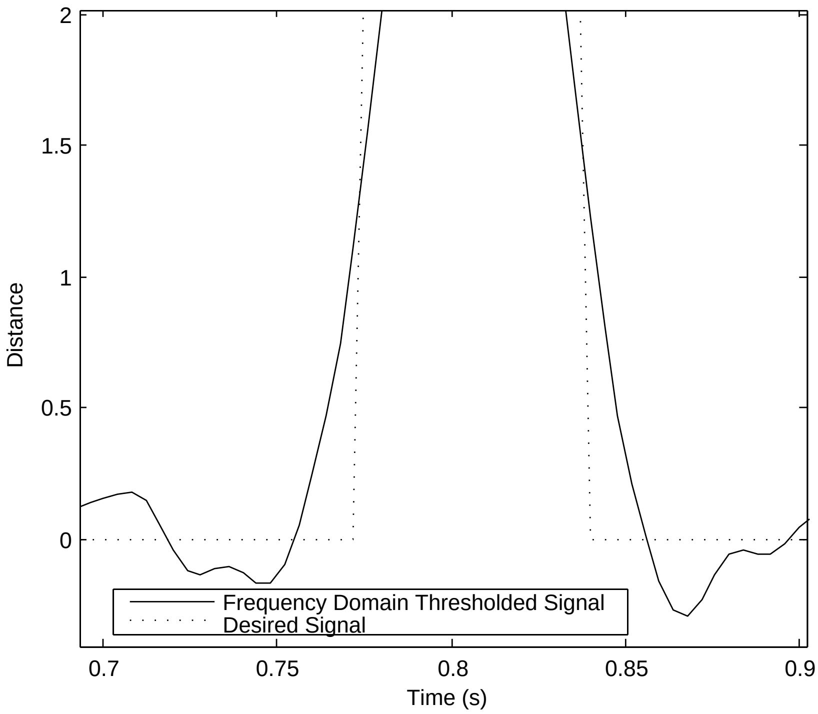

Figure 7.7: Zoomed window of Frequency Domain Thresholded signal

In Figure 7.6 and Figure 7.7, there exists considerable drift from the original axle event time. Where too much drift from the event occurs, errors in speed calculation may occur which are further compounded in determination of the wheel base of the vehicle. This may ultimately result in either misclassification of a vehicle, or in the classification algorithm not being able to classify the vehicle at all.

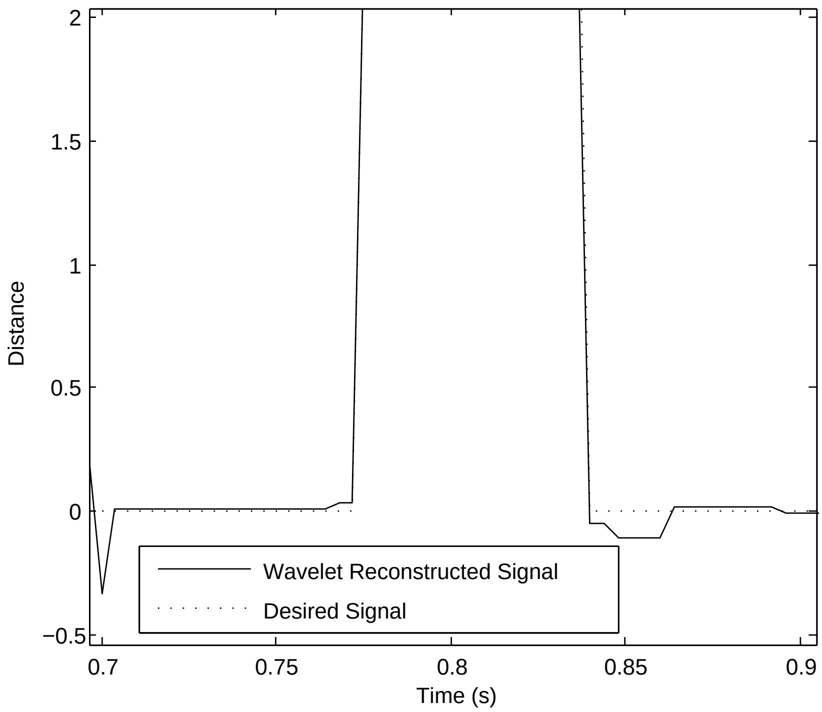

Figure 7.8: Zoomed window of Haar Wavelet Denoised Signal

Figure 7.8 shows how the superior ability of the Haar wavelet to approximate features of the signal in the Discrete Wavelet Transform lead to almost no deviation from the desired signal at the critical event time. It would be expected that from this denoised signal, there would be no error in the calculation of speed or wheel base from the data.

7.3 Summary

From the material presented in this chapter, it can be concluded that the preferred method for the denoising the theoretical measurement data is via the use of the Discrete Wavelet Transform, using a Haar wavelet. The level of signal decomposition should be chosen as six. The method for thresholding the detail coefficients is chosen as hard.

The Matlab command to apply the Wavelet method is as follows.

```
[distance1,CXD,LXD] = wden(distance1,‘rigrsure’,‘h’,‘one’,6,‘haar’);
```

Where,
‘distance1’ is the vector to be denoised,
‘rigrsure’ threshold selection rules within Matlab, are based on the underlying model

$$y = f(t) + n,$$

where n is a white noise.
‘h’ is for Hard Thresholding,
‘one’ is no multiplicative threshold rescaling, since the noise is white,
‘6’ is for six levels of decomposition, and
‘haar’ is the wavelet type.

In Chapter 9, the theoretical results will be applied to a real sensor signal prove the validity of the wavelet denoising method for the application.

Chapter 8

Classification Algorithm

The classification algorithm for the system follows a distinct routine. In order to classify vehicles, a number of parameters are required. These parameters are then used to determine the vehicle and their appropriate classes. This classified vehicle data should be reported, however there are a number of other useful traffic information that can be derived from the data. This section is devoted to the process involved.

8.1 Classification Algorithm

There is a specific sequence of events that the classification algorithm uses to successfully count and classify vehicles detected by the system. The events can be summarized into a number of processes.

- Load the text file containing the measured data.
- Denoise data from both lasers using wavelets.
- Calculate the position of each lane relative to the sensors.
- Separate vehicles into particular lane for analysis.
- Select first lane to classify.
- Detect axles in current lane.
- Determine speed of first axle in data.
- Calculate axle spacings at axle speed.
- Select axles, until axle spacing exceeds 10m as part of current vehicle.
- Determine number of axles and axle groups in current vehicle.
- Determine separation of first and second axles.
- Determine separation of second and third axles if it exists.
- Store vehicle data in array for later classification.
- Continue determining individual vehicle data for particular lane.

- Classify Vehicles in particular lane, and output to text file.
- Increment to next lane.
- Repeat classification process.

The Matlab code for the classification algorithm can be found in Appendix D.

8.2 Importing and Validating Data

The data is recorded from the sensors via a USB to Serial converter interface. In order for the data from the two ULS sensors to be analyzed, the two sensors must be synchronized. The graphical user interface provided by LTI for the ULS provides data capture to an ASCII file in .log format. Typical string measurements are formatted in the text file as seen in Table 8.1.

Measurement No.	Laser 1	Laser 2
1	13:32:09:34, 9.110,m	13:32:09:33, 8.954,m
2	13:32:09:39, 9.084,m	13:32:09:33, 8.985,m
3	13:32:09:39, 9.110,m	13:32:09:33, 8.981,m
4	13:32:09:39, 9.113,m	13:32:09:33, 8.987,m
5	13:32:09:39, 9.045,m	13:32:09:38, 8.926,m
6	13:32:09:39, 9.083,m	13:32:09:38, 8.958,m
7	13:32:09:39, 9.014,m	13:32:09:38, 9.015,m
8	13:32:09:39, 9.086,m	13:32:09:38, 9.012,m
9	13:32:09:39, 9.110,m	13:32:09:38, 8.945,m
10	13:32:09:39, 9.081,m	13:32:09:38, 8.954,m
12	13:32:09:39, 9.015,m	13:32:09:38, 8.987,m
13	13:32:09:39, 9.045,m	13:32:09:38, 8.887,m
14	13:32:09:44, 9.078,m	13:32:09:63, 8.954,m
15	13:32:09:44, 9.105,m	13:32:09:68, 8.982,m

Table 8.1: Typical Data Output from two .log files

The first part of any single line is the time, in 24hr, that the data was read from the serial port. Unfortunately, due to the USB to serial converter cable being used for the project, the time stamp is the same for many measurements. I believe this to be due to the buffered output from the USB to serial cable. The times also do not exactly match. This makes synchronization of the two sensors unreliable. Due to this, using the ULS Interface Capture software was not a possibility.

The alternative to using the ULS Interface Software was to use Matlab to capture the serial data. A simple Matlab script was written to first flush both serial buffers, and then read the serial input data buffer of the first sensor, then read the serial input data buffer of the second sensor. This would continue for a predetermined time as set within the script, ensuring that both streams of sensor data are as close to synchronized as possible.

The serial port is allocated a file identifier. This identifier can then be used to connect to the particular serial port.

```
Las1 = instrfind('Type', 'serial', 'Port', 'COM9', 'Tag', '');
```

The baud rate must also be set.

```
% Set-up com5
set(obj1,'BaudRate',115200,'Terminator','cr');
```

The default 'data bits', 'stop bits' and 'parity' in Matlab serial communication are the required settings for communication to the sensors.

The fscanf function in Matlab reads all the data from the file specified as 'Las1', or com9. As the program starts, the lasers are already running, and as such the input serial buffers will be full. Following the initial flush of the serial input buffer, subsequent readings are recorded from both sensors.

```
% clear the input buffer by flushing each buffer
flushinput(obj1)
flushinput(obj2)

for i=1:2500 % Make 2500 measurements
    data1{i,1} = fscanf(Las1, '%s\n'); % Data from Sensor 1
    data2{i,1} = fscanf(Las2, '%s\n'); % Data from Sensor 2
end
```

The data is now formatted in data1 and data2 as,

```
$BM,1.514
```

where $BM stands for basic measurement, and the floating point number is the distance to the target. The delimiter is a comma.

If there is an error in the measurement,

```
$ER,5
```

$ER indicates an error has occured, and 5 indicates the type of error, which is average not filled in this case.

When interpreting the data for processing, there can only be two options, either the data is valid or is in error. By equating the first three characters of each string to '$BM', the data can be validated. If the data is in error, the algorithm makes the current measurement equal to the previous measurement. Typically, these types of errors only occurred spuriously, and as such were able to be removed via a median filter. The size of the median filter was three. If the error where to occur on a rising or falling edge in the signal, the error is limited to one sampling interval and has minimal effect upon the accuracy of the data. If the error occurs anywhere else within the data, it will have no effect upon the accuracy.

8.3 Lane Separation

The sensors spend most of the time measuring distance to the far side of the road. In order to establish lane division, the average distance to the far side of the carriageway is calculated. From this, the lane width, as determined from the Main Roads Road Planning and Design Manual, is used to set discrete distances at which vehicles will be determined to be in a particular lane. From this, the number of lanes is determined, and is used as the counter for a counter controlled loop for classifying vehicles in particular lanes. Using this, the vehicle data is then separated into lanes, ready for analysis. In Chapter 9, Figure 9.7 shows the result of separating vehicles into particular lanes.

8.4 Axle Detection

Edges characterize the boundaries of leading and trailing edges of wheels of the vehicle and are therefore a problem of fundamental importance in processing the signal. Edge detecting significantly reduces the amount of data and filters out useless information, while preserving the important structural properties of the signal.

There are a number of algorithms used to detect edges. They may be divided into two major categories, Gradient and Laplacian filters. The Gradient filter use the first derivative to find changes in amplitude of the signal. Peaks in the Gradient filtered signal indicate edges in the original signal. The Laplacian filter uses the second derivative to find changes in the first derivative. The Laplacian method can be used to find edges in the original signal by finding the zeros crossings in the Laplacian filtered signal.

Gradient filtering is achieve by convolution of the signal with the filter.

The filter is,

$$\begin{bmatrix} \frac{1}{dt} & \frac{-1}{dt} \end{bmatrix}$$

In Figure 8.1(a), the signal is a pure sine wave. A Gradient filter has been applied to the original signal in Figure 8.1(b). Considering that the derivative of a sine wave is a cosine wave, it can be seen that the original signal has been differentiated. The signal is again differentiated in Figure 8.1(c) to give a negative sine wave.

Gradient filtering a smooth wave such as the one presented in Figure 8.1 does not present much valuable information, as there are no edges to detect within the signal.

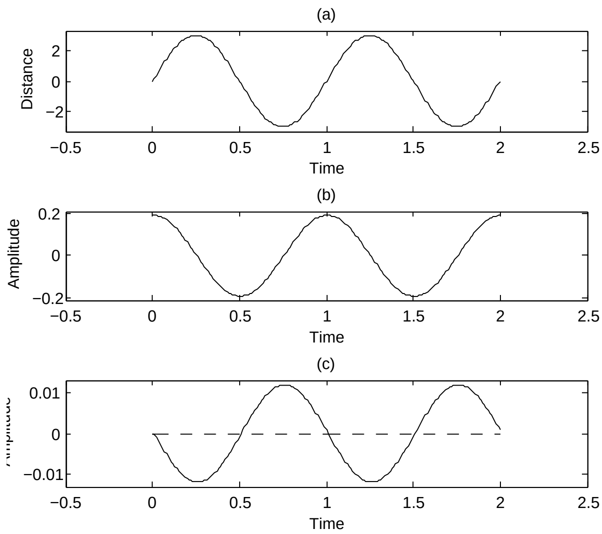

Figure 8.1: Gradient Filtering of Sine Wave

However, Gradient filtering a squarewave such as in Figure 8.2 yields the rising and falling edges of the signal. Figure 8.2 (b) shows the edges of the squarewave in Figure 8.2(a). The Laplacian filter output is shown in Figure 8.2(c), but for this application to this system, yields no more information than provide by the first derivative filter. The positive peaks do indicate the width of the tyre, which could be used to predict the type of vehicle currently being detected if the system were to operate in real time. However, the system uses post processing, and therefore determining the width of the tyre is of no importance.

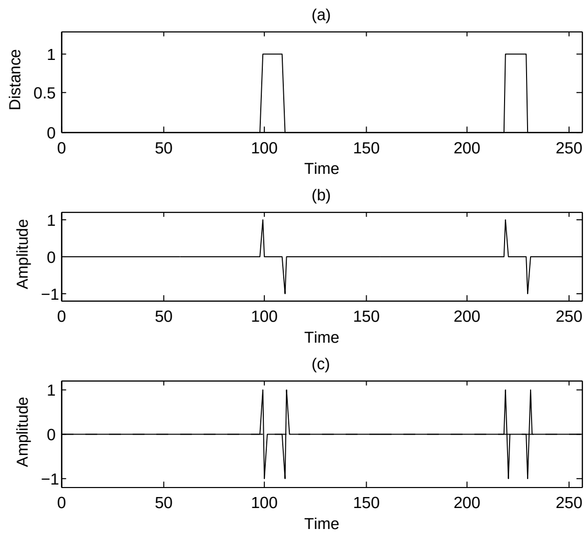

Figure 8.2: Gradient Filtering of Signal plus Noise

A positive peak indicates the leading edge of the wheel, and a negative peak indicates the trailing edge. Typically, only the leading edges of the wheels will be required, from which the distance between axles can be calculated.

8.5 Axle Grouping

As shown in Appendix B, an axle is defined as part of a group if the distance to an adjacent axle is less than 2.1 metres.

The distance between axles is found easily from the detection of the front edge of the wheel as discussed in the previous section. The time between the first positive peak produced from the gradient filter on laser 1 and the first positive peak on laser 2 is taken as the time the vehicle took to travel from laser 1 to laser 2. Since the distance between the lasers is known to be 1 metre, the speed of the vehicle can be found from,

$$\text{Speed (m/s)} = \frac{distance}{time}$$

Once the speed of the vehicle is known, this distance between consecutive axles can be found from edge detections on one set of laser data. The spacing between consecutive axles is calculated by multiplying the time between edges by the speed of the first axle.

$$\text{Distance (m)} = \text{speed} \times \text{time}$$

When the distance between axles exceeds the Austroads '94 Standard of 10m for maximum axle spacing, then the remaining axles in the data are not considered to be part of the current vehicle.

The flow chart for the algorithm implemented in MatLab is set out in Figure 8.3.

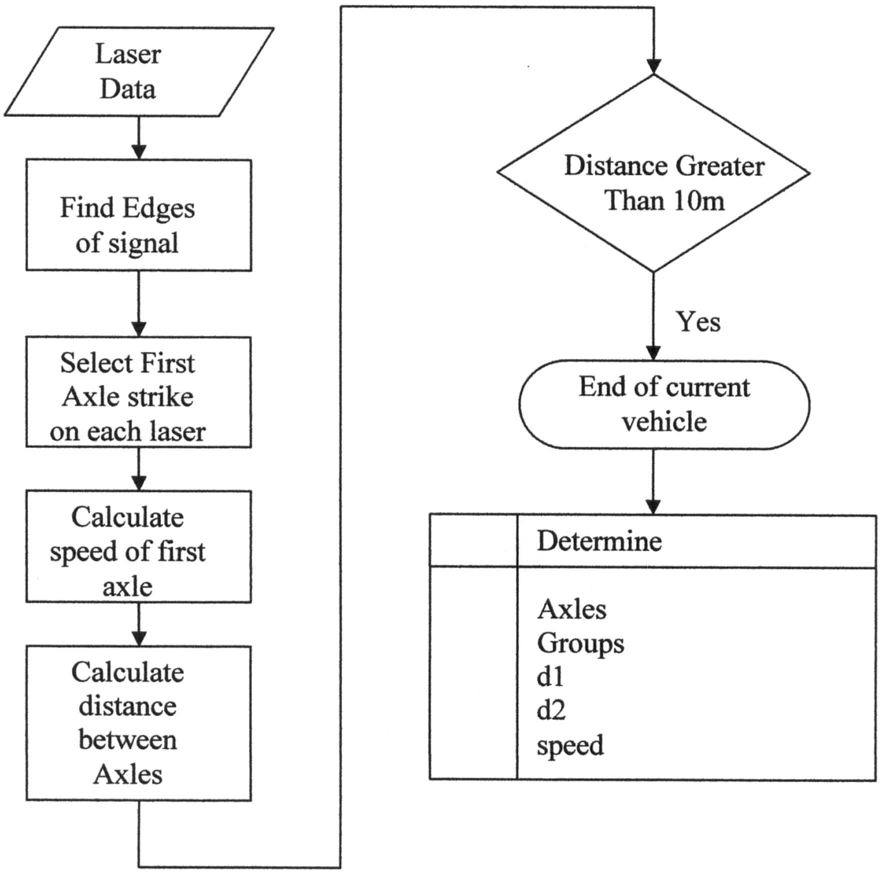

Figure 8.3: Flowchart for collation of vehicle data

The data is then stored in an array as set out below,

$$\begin{bmatrix} \text{Axles} & \text{Groups} & \text{d1} & \text{d2} & \text{speed} \end{bmatrix}$$

Where,

- Axles is the number of axles in vehicle
- Groups is the number of axle groups
- d1 is the axle spacing between first and second axle
- d2 is the axle spacing between second and third axle if it exists
- speed is the speed of the vehicle

8.6 Error Analysis

Any error in speed measurement of vehicles affects the successful determination of axle separations, and hence the classification of the vehicles. It is of utmost importance to determine the maximum error due to the timing of measurements.

The measurement frequency of the system is 250Hz. A measurement is output by the sensor every 0.004 seconds. Using the edge detection method set out in the previous section, the ability of the algorithm to find the edge of the wheel accurately is determined by the speed of the vehicle.

The edge of the wheel passes through the ULS beam somewhere within the single measurement period, consisting of a number of measurements set by the PPM. Since the data is discrete, there is no further information available to determine when the axle passed the sensor.

From this, it can be concluded that the data can have a maximum error of plus and minus one data measurement. Since the number of measurements taken during each axle event is proportional to the speed of the vehicle, the error is directly proportional to the speed of the vehicle. The proportion of the error will be less for vehicles at low speeds and greater for vehicles at higher speeds. In Figure 8.4 the error in kilometres per hour is presented against the relative speeds of the vehicles. In Figure 8.5, the error is presented as a percentage of the relative vehicle speed.

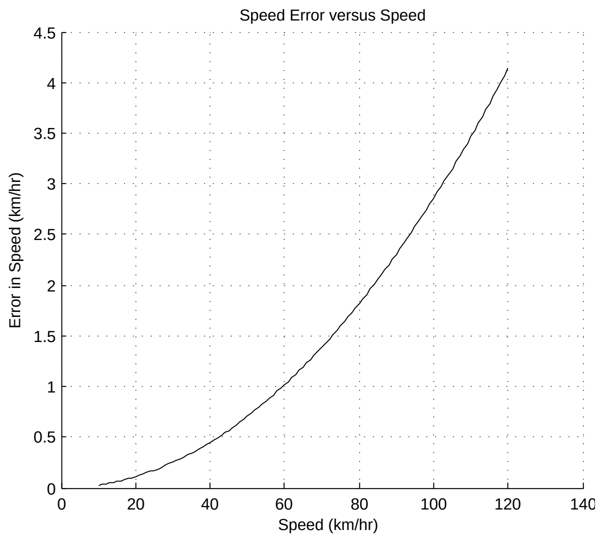

Figure 8.4: Maximum speed error at relative vehicle speeds

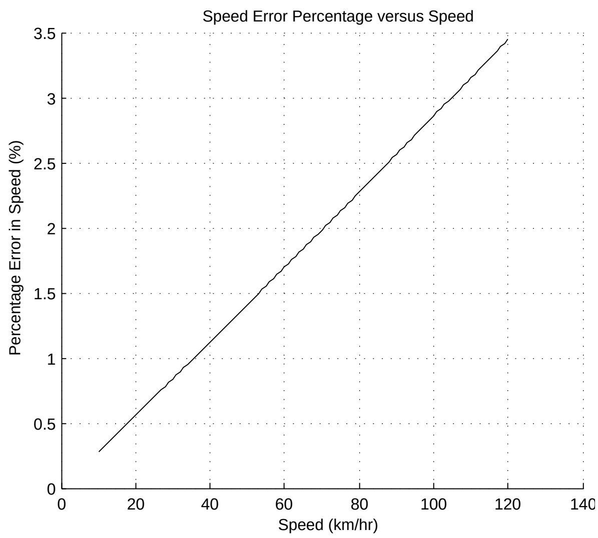

Figure 8.5: Percentage maximum speed error at relative vehicle speeds

A number of techniques could be used in order to reduce the error. The measurement frequency could be increased giving more measurements in the same time period, which will decrease the ratio of an error to total measurements. The limitation for this will be the data transmission rate, that is the time taken by the processor to collect the text string containing the distance measurement from the ULS.

Alternatively, the distance between the sensors could be increased. The axle would have to traverse a greater distance leading to a greater number of measurements be taken.

A comparison of error analysis is presented in Figure 8.6 and 8.7 showing the reduction in error by increasing the distance between the sensors.

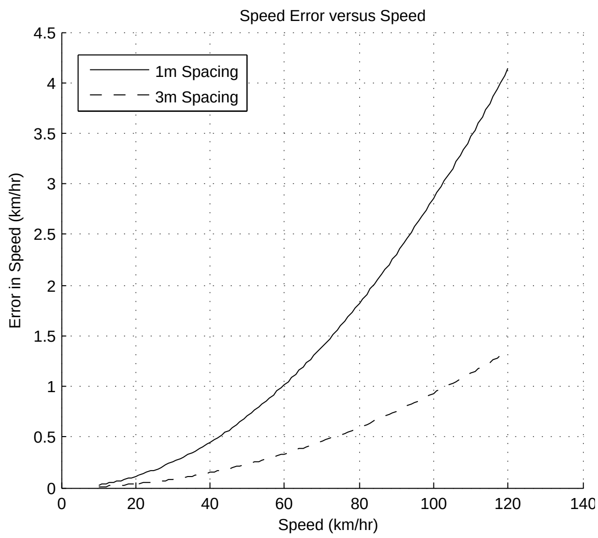

Figure 8.6: Comparison of error for sensor separation.

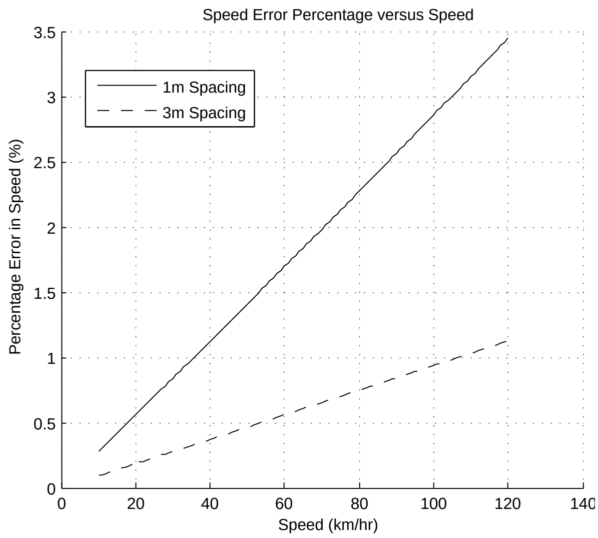

Figure 8.7: Comparison of error percentage for sensor separation.

This gives an option to improve the speed accuracy should it present as a problem, without placing further requirements on the signal processor. However, large spacing between the sensors will result in a physically large system, so consideration must be given to both options.

8.7 Determination of Classifying Parameters

The Austroads 94 12 Bin classifying system bases classification on combinations of four parameters.

1. Number of axles,
2. Number of axles groups,
3. Axle spacing of first and second axles, and
4. Axle spacing of second and third axles if it exists.

Detailed classification information can be obtained from Appendix B.

At this point in the algorithm, the data being processed contains only the vehicles in the lane under consideration. Their relevant positions within the lane are no longer important and as such, all distances to targets is set to one metre using a thresholding algorithm, regardless of the the lane they are in.

Prior to classification, the data is presented in the following format,

$$\begin{bmatrix} \text{Axles} & \text{Groups} & \text{d1} & \text{d2} & \text{speed} \end{bmatrix}$$

For classification, this data is passed into individual functions to determine the class of vehicle. If a matching class is found in the function, a sentinel is set indicate the match. This continues until all classes have been tested. If no match has been produced, the data is appended to an 'unclassifiable' array for analysis at a later time. A typical class function is presented below. This is for determination of a vehicle in class 9. Class 9 is a six axle articulated vehicle, or a rigid vehicle towing a trailer.

```
function [class] = class9(axleData)
%---------------------------------------------------------------
% Input: axle data
% Output: Vehicle Class 9; if 0, no match found
%---------------------------------------------------------------
% Paramaeters
% ( Axles, Groups, D1, D2, Speed, Lane )
% Class 9
% Class 9 is a 6 axle articulated or rigid vehicle and trailer
% Parameters are    (axles = 6 and groups >2) or
%                   (axles > 6 and groups =3)
if ((axleData(1,1) == 6) & (axleData(1,2) > 2))...
        | ( (axleData(1,1) > 6) & (axleData(1,2) == 3))
    class = 9;
else
    class = 0;
end
%---------------------------------------------------------------
%EOF
```

The process loops until there exist no further vehicles in the lane to be classified. All vehicles that were successfully classified are then appended to a text file, formatted as set out below.

```
Vehicle Class 1, Lane 1, Speed 64.3 km/hr
Vehicle Class 1, Lane 1, Speed 69.2 km/hr
Vehicle Class 1, Lane 1, Speed 75.0 km/hr
Vehicle Class 1, Lane 1, Speed 69.2 km/hr
Vehicle Class 1, Lane 1, Speed 64.3 km/hr
Vehicle Class 1, Lane 2, Speed 75.0 km/hr
Vehicle Class 1, Lane 2, Speed 90.0 km/hr
```

The next lane is then loaded and classified. This continues until all lanes have been classified. The data containing any vehicles that did not match a vehicle classification is appended to an array called 'Unclassifiable', for later examination.

8.8 Statistical Data Output

The term traffic study involves both the collection and analysis of data relating to traffic and its characteristics.

Traffic studies are performed to (*AP-G11.3-04 Traffic Studies* 2004):

- provide a basis for planning and designing traffic facilities, including the selection of geometric standards, economic analysis, impact assessment, and the determination of priorities;
- assist traffic operation by indicating the needs for traffic control devices such as signs, traffic signals, pavement markings, and school and pedestrian crossings;
- evaluate the effects of road safety measures and other changes made for traffic by conducting before and after studies;
- determine the basic characteristics and the general laws of traffic behaviour, and

- provide heavy vehicle and freight data to improve pavement analysis and design capability, bridge management capability, and the monitoring of road network performance.

There is obviously a large amount of statistical parameters that can be determined from the data collected by the system. The objectives of the project are to only determine a few of the parameters such as 85th percentile speed, 15k pace and headway.

The 85th percentile speed is the speed below which 85% of vehicles are travelling in free flowing traffic conditions. Road speed limits are typically determined by the 85th percentile speed.

Determination of the 85th percentile speed is through the construction of a probability density function. This can be seen in Figure 8.8.

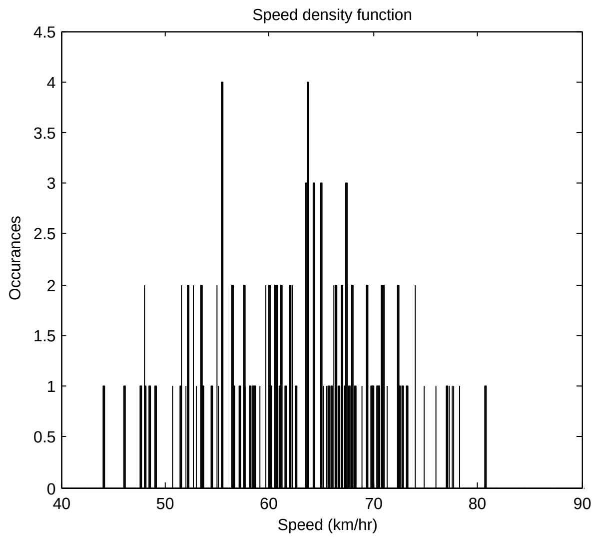

Figure 8.8: Distribution of vehicle speeds throughout survey.

Here the data containing vehicle speeds is sorted into an array where a count of vehicles travelling at particular speeds is recorded. A cumulative density function may then be developed. This is a statistical representation of the total number of vehicles travelling at or below the speed of interest. The highest speed encountered in the survey represents 100% of all vehicles, as all vehicles are travelling at speeds lower than this speed. A cumulative density function can be seen in Figure 8.9.

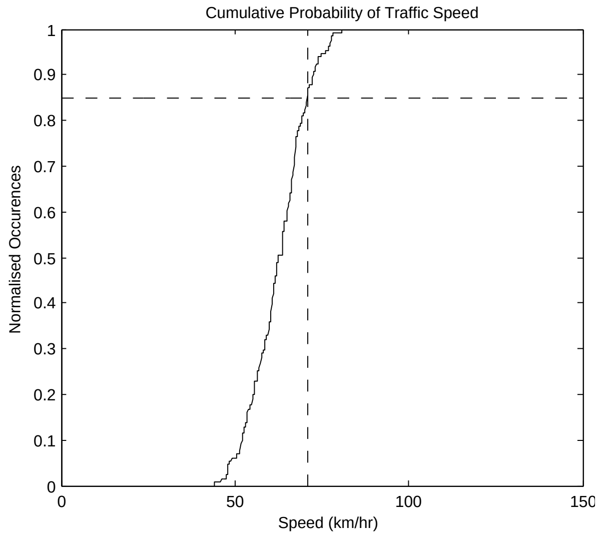

Figure 8.9: Cumulative density for determination of 85th percentile .

The 85th percentile speed is then simply read from the graph on the y axis, at 0.85. The speed translated to the x axis, as in Figure 8.9 , is the 85th percentile speed, in this case, approximately 70 km/hr.

Similar to the 85th percentile speed is the 15k pace. The 15k pace is also determined in the same manner using a probability density function. The difference is rather than finding an individual speed below which vehicles are traveling, a spread of 15km/hr central to a particular speed is used. In other words, the idea is to find a speed plus and minus 7.5km/hr, which has the highest count of vehicles. In the previous example example, as in Figure 8.9 the 15k pace was 66.7 km/hr.

Headway, sometimes referred to as gap, is defined as the time interval between consecutive vehicles in a single lane. Headway can be an indication of traffic flow. In congested traffic situations, the speed of vehicles may be lower, and the headway will typically be less, that is the vehicles tend to follow closer than in free flowing traffic situations. In congested traffic situations, the 85th percentile speed loses meaning, as the vehicle in front is limiting the speed of the vehicle following. By monitoring headway, traffic congestion can be identified, and exclusion of the congested data when determining

85th percentile speed is possible. This will lead to more reliable traffic data. Typical headway employed by Department of Main Roads ARMIS division is set at four seconds.

For the project, the system test data was typically only a small number of vehicles tested and classified contiguously. For this reason, no real results will be presented in Chapter 9 pertaining to the output of statistical data. The traffic data presented here in this section has been generated using random number generators within Matlab in the same format as would be expected from the output of the classification algorithm. The analysis involved the order of hundreds of vehicles to obtain meaningful results, where as a determination of statistical data from a small sample would be meaningless. The objective was to prove that the statistical data could be derived from traffic data supplied by the classification algorithm. The code for the determination of statistical data is presented in Appendix D.

Chapter 9

Results

9.1 Signal Denoising

Due to the high PRF and low PPM parameters of the ULS, there is considerable noise in the measurement data. The vehicles are to be separated into individual lanes and to confidently apply a time domain thresholding algorithm to lane data, the noise in the signal should be minimized.

9.1.1 Low Pass Filter

As outlined in Chapter 5.7, a low pass filter was able to reduce the noise present in the fictitious signal. Although the low pass filter had a relatively small NRR, the major contribution to the noise present in the signal was around the discontinuities of the signal. This is evident in Figure 9.1, where the noise is considered as the deviation from the desired measurement.

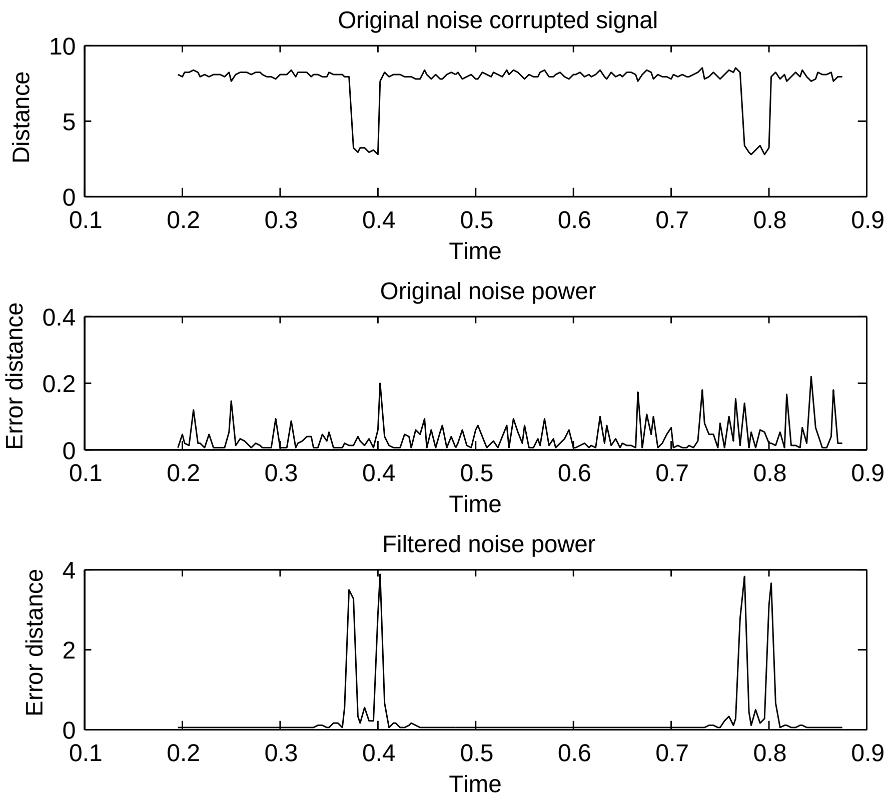

Figure 9.1: Noise removal comparison via low pass filter.

In the top figure, a portion of the original noise corrupted signal is presented. The centre figure shows how originally the noise is distributed throughout the signal. In the lower figure, after filtering, much of the baseline noise is removed. However, around the discontinuities, or where an axle presents on the sensor, the volume of noise increases. This is due to the removal of the higher frequencies allowing the signal to drift from the true edges of the signal.

Application of a gradient filter to this filtered signal will result in incorrect detection of the axles. The axles will be detected at an earlier time than they were truely present, resulting in speed errors. For this reason, a low pass filter does not seem suitable for the system.

The application of a low pass filter to a true sensor output is shown in Figure 9.2. Here we can see that while the noise is attenuated well between axle events, ringing is produced pre and post any axle event, further complicating axle detection.

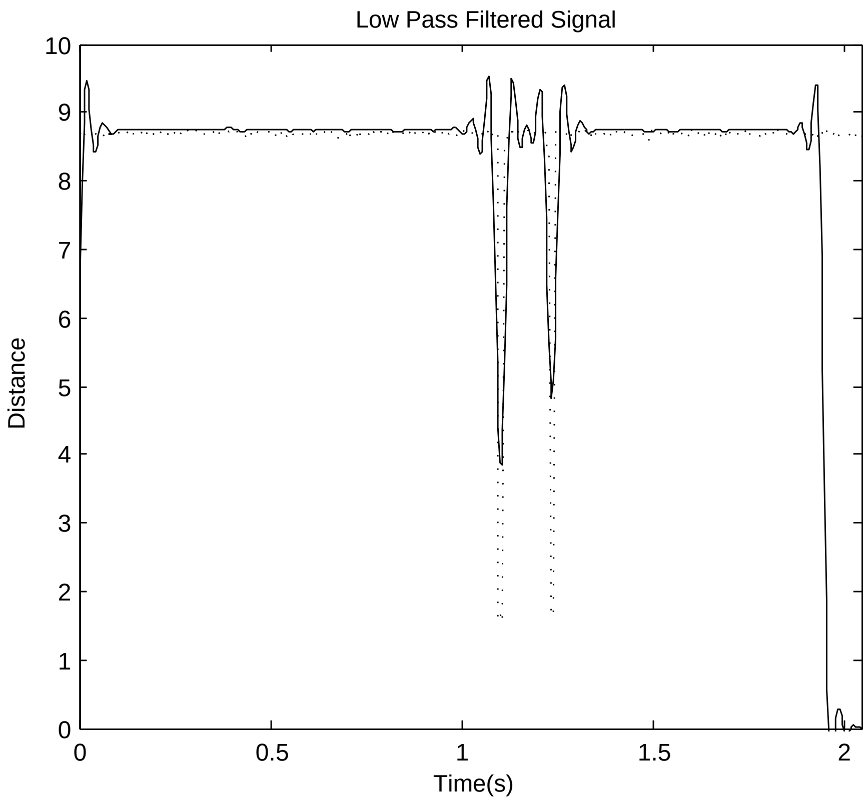

Figure 9.2: Denoising of real sensor output using a low pass filter.

Also in Figure 9.2 the original noisy signal is shown as the dotted line. The filtered signal peaks have been attenuated substantially. While the original signal shows the vehicle at approximately 1.5 metres from the sensor, after filtering the signal, the vehicle ranges from 4 to 5 metres from the sensor. This makes the lane identification unreliable, and further discredits the use of a low pass filter in conjunction with the system.

9.1.2 Frequency Domain Thresholding

In Figure 9.4, the same fictitious signal as presented in the above section, is filtered using frequency domain thresholding, with a -13dB noise floor.

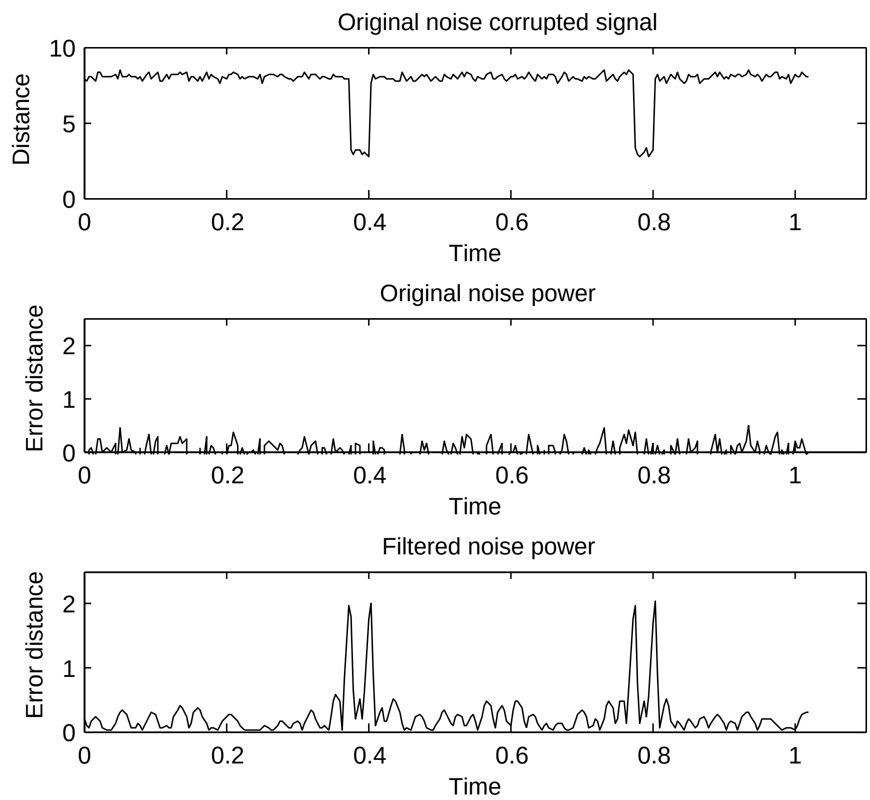

Figure 9.3: Noise removal comparison via frequency domain thresholding

In this application, there is not a great amount of reduction in the amplitude of the noise, as well as the same problem encountered in the low pass filter method where the true edges of the signal are lost.

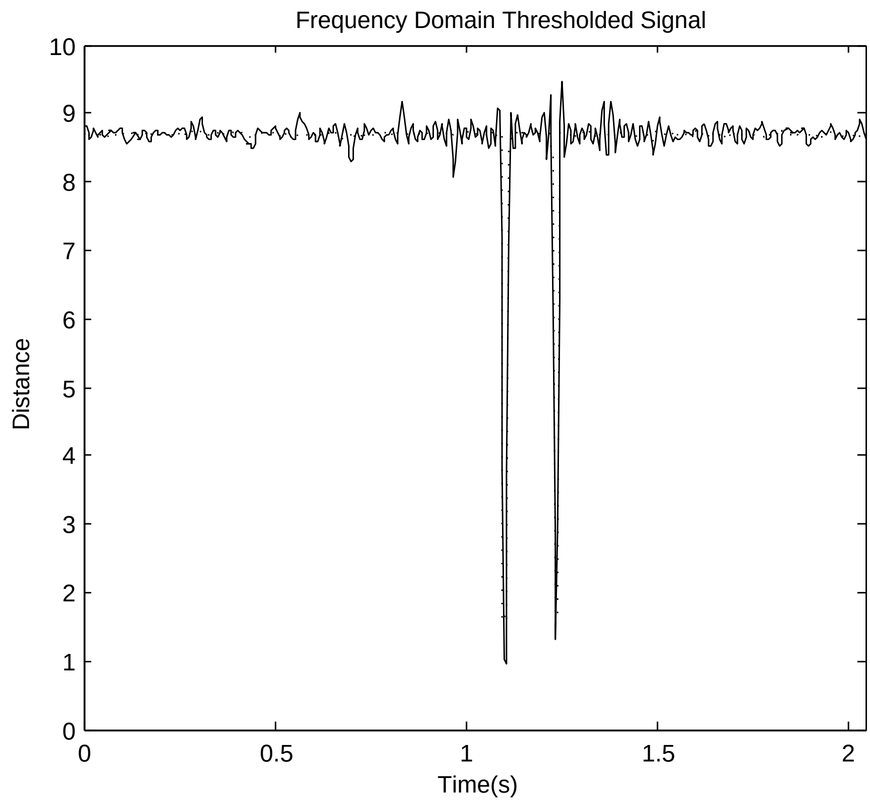

Figure 9.4: Denoising of real sensor output using frequency domain thresholding

In the application of frequency domain method to a real sensor data output, there is also no reduction in the noise content, in fact the noise appears to be increased. This method does not appear favourable for denoising the sensor data output.

9.1.3 Wavelet Denoising

In Figure 9.5, the same fictitious signal is filtered with a Haar mother wavelet, using 6 levels of decomposition.

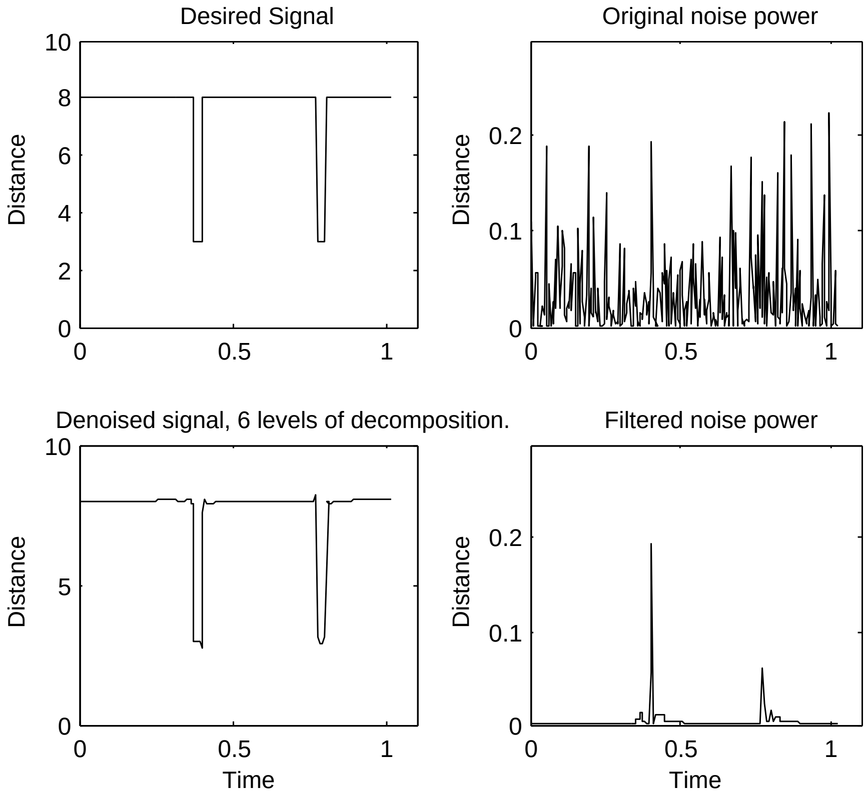

Figure 9.5: Six levels of decomposition with Haar wavelet.

Figure 9.6 shows a true sensor output data, as well as the wavelet denoised data. The data was denoised using the theory presented earlier with six levels of decomposition and hard thresholding.

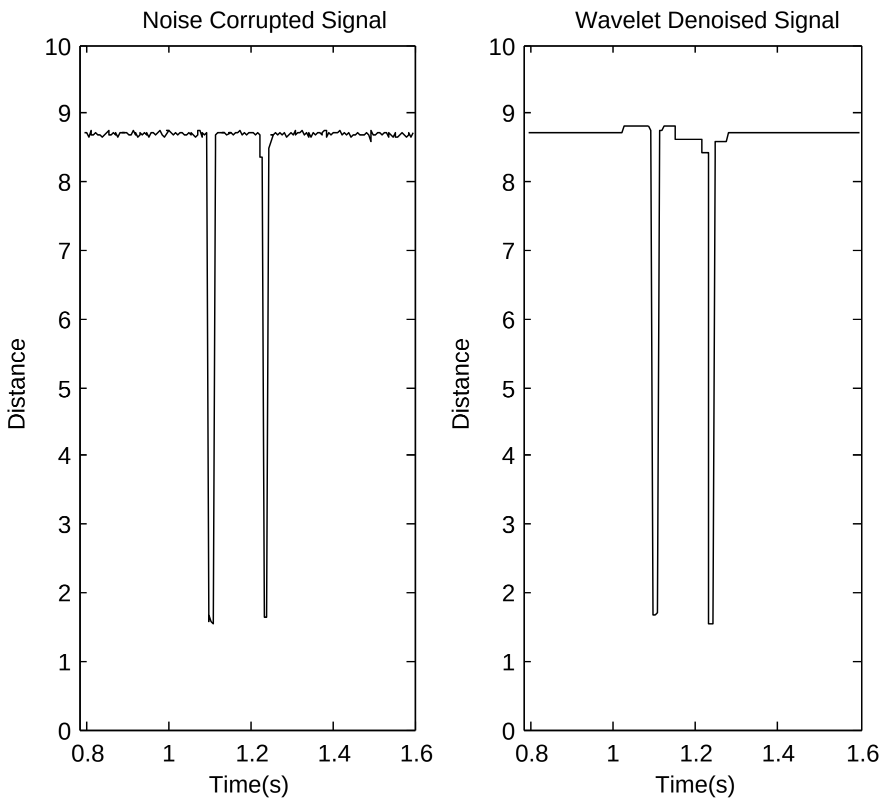

Figure 9.6: Denoising of real sensor output using wavelets

As discussed in Chapter 7, using the theoretical signal, the best denoising technique was found to be through using the discrete wavelet transform. After application of the different methods to a real sensor output data, the wavelet denoising method has proven its ability to denoise the signal, and is the chosen method for denoising the data in the classification algorithm.

9.2 Lane Detail

As a requirement of the project, the system must be able to detect and classify vehicles in multiple lanes. The algorithm sorts through the detected axles to find the closest of all possible axles to the sensors. With this measurement distance, the algorithm assumes that this vehicle is travelling 0.60 metres from the lane edge. The algorithm then calculates the position of subsequent lane edges based on this starting point. The lane width allocation is 3.5 metres as per set out in Chapter 2.1. A trial of the system showing thirteen vehicles in two lanes is shown in Figure 9.7. The figure also illustrates how much variance exists with vehicles distributed across a single lane.

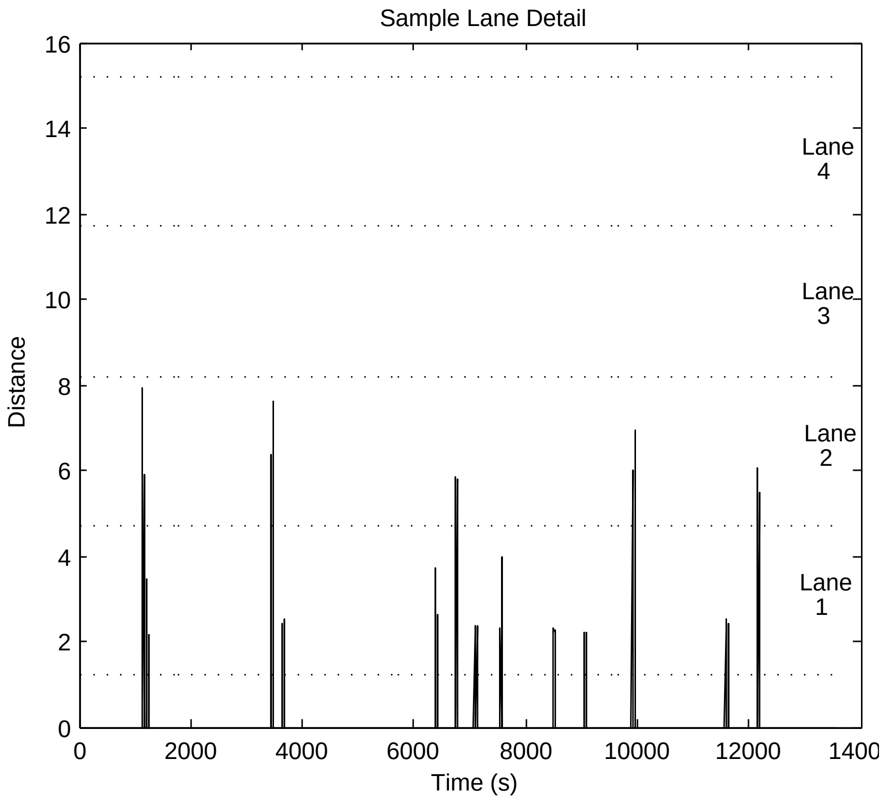

Figure 9.7: 13 Vehicles in Various Positions in Lanes

The algorithm then determines the occupied lanes, seperates vehicles within each lane and will only count and classify vehicles in those lanes.

9.3 Testing Procedure

In Chapter 3, the accuracy of existing counting and classifying systems was investigated in order to determine a benchmark that the project system could be compared against. In this section, these results will be compared against the ones determined from the existing systems.

9.3.1 Speed and Wheelbase Determination

The objective for the test procedure is to determine the accuracy at which the developed system can detect axles and calculate the axle separation of known vehicles at known speeds. The accuracy of the system will be calculated as a percentage deviation from the known value using the following equation,

$$\text{Error} = \left| \frac{\text{Known Value - Calculated Value}}{\text{Known Value}} \right| \times 100\%$$

The test will be first conducted upon vehicles in a single lane. Figure 9.8 shows the layout of the test procedure.

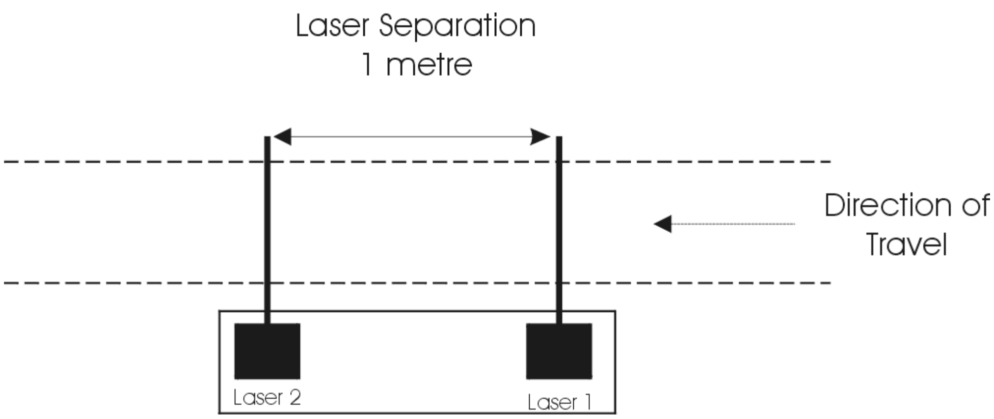

Figure 9.8: Single Lane Test Procedure Layout

In this configuration, the vehicle is first detected on Laser 1, and then on Laser 2. The speed and wheel base dimensions then can be calculated as set out in Chapter 8.2, Axle Grouping.

9.3.2 Speed and Wheelbase Accuracy Results

For determination of accuracy of the system in this section, a Toyota Corolla was used as the pilot vehicle. The wheel base dimensions were known to be 2.48 metres. As the synchronization of the sensors was an issue, the data was able to be manipulated to ensure that the calculated wheelbase was exactly matching the true dimensions. By doing this, the data is guaranteed to be synchronized, and the speed calculated must therefore be reliable.

Table 9.1: System Determined Vehicle Speed Results

True Speed (km/hr)	Calculated Speed (km/hr)	Speed Error (km/hr)	Percent (Error) (%)
10	18.8	8.8	88.0
20	28.6	8.6	43.0
30	36.1	6.1	20.3
40	46.4	6.4	16.0
50	53.9	3.9	7.8
60	61.1	1.1	1.8

While the results presented in Table 9.1 show considerable error in the determination of speeds, the reliability of the True Speed is of question. A speed radar gun was unavailable to accurately determine the speed of the vehicle, so the speed was determined by visual verification of the speed according to the speedometer in the vehicle. The reliability of the speedometer, especially at low speeds is contributory to the magnitude of the error encountered in the data. Future work will see the system speed accuracy verified with the use of a speed radar gun.

9.3.3 Vehicle Class Determination

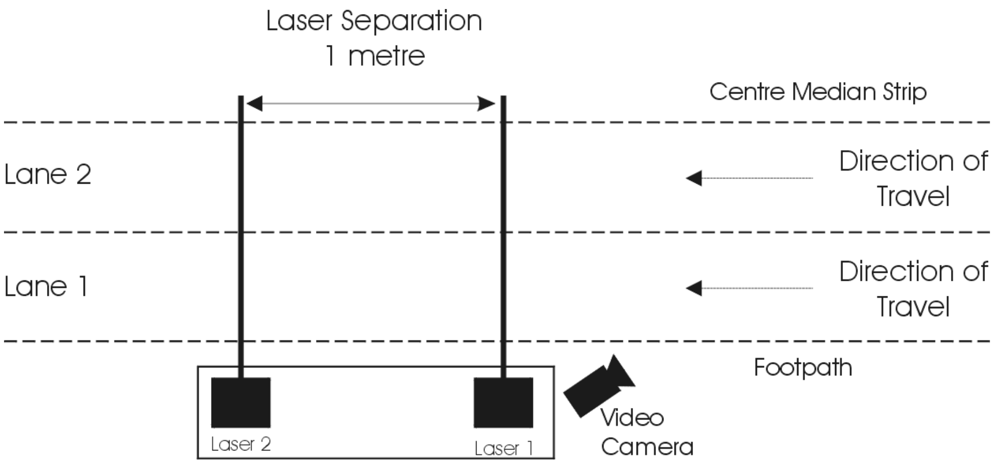

Figure 9.9: Multi-Lane Test Procedure Layout

A test procedure was developed to determine the systems ability to detect vehicles in multiple lanes, as per Figure 9.9.

For this configuration, the system was set up in a single direction, double lane uncontrolled environment, where the existing speed limit was 60km/hr. Video footage of the passing vehicles was recorded so as to later compare the vehicle types determined by the system with vehicle types visually observed.

The results of the testing is presented in the next section.

9.3.4 Vehicle Class Accuracy Results

Table 9.2 show the first 20 classified vehicles of a 57 vehicle classification sample.

Table 9.2: Vehicle Classification Results

Vehicle Number	System Determined Class	Visual Classification	Speed (km/hr)
1	Class 1	Class 1	69.2
2	Class 1	Class 1	75.0
3	Class 1	Class 1	90.0
4	Class 1	Class 1	75.0
5	Class 3	Class 1	75.0
6	Class 1	Class 1	75.0
7	Class 1	Class 1	64.3
8	Class 3	Class 1	100.0
9	Class 3	Class 1	100.0
10	Class 1	Class 1	75.0
11	Class 3	Class 1	90.0
12	Class 3	Class 3	75.0
13	Class 1	Class 1	81.8
14	Class 3	Class 1	69.2
15	Class 1	Class 1	69.2
16	Class 1	Class 1	75.0
17	Class 3	Class 1	75.0
18	Class 3	Class 1	90.0
19	Class 3	Class 1	90.0
20	Class 9	Class 9	69.2

From the classification sample, 39 of the 57 vehicles classified successfully. The speed limit for the carriageway where the data was collected, was 60 km/hr. Of the 39 vehicles that classified successfully, the speed of each vehicle appears to be marginally high. There was no speed radar gun available to verify any of the vehicles speeds. It could be expected that some of the vehicles would be traveling at less than 60 km/hr but the data does not show this.

For the misclassified vehicles, the visual inspection indicated that nearly all misclassified vehicles were the larger wheel base type vehicles such as four wheel drive utilities. These vehicles tend to have wheel bases closer to the the limit of 3.2 metres between class 1 and class 3 classification. The incorrect determination of speed would cause the vehicle to be determined to have a longer wheel base if the speed was recorded too high.

9.4 Summary

The 68% classification accuracy leaves a lot to be desired from the system. Almost certainly the incorrect determination of speed which led to the errors in the classification data was a direct result of the timing and synchronization of the two sensors. Matlab has many serial commands that allow access to serial port properties such as bytes read and bytes available. During testing, with adjacent commands to flush one buffer then immediately flush the other buffer, the bytes available property for the first serial port could be as high as 90 bytes, which is equivalent to 9 measurements. This is attributed to the multitasking of the windows platform, where even though the commands are executed one after the other, the time-slicing nature of windows means that the operating platform could be performing several tasks in between flushing the first buffer and the second. This allows data to build up in the buffer and therefore offsets the sensor data.

For the data presented in Table 9.2, if the data is manipulated by recording the bytes available parameter for the first serial port prior to flushing the second, and then adjusting the measurements recorded from sensor 1 by this amount, the vehicles are classified successfully, and the speeds become realistic. This confirms the problem faced by the system is sensor synchronization. It appears that the only solution to the problem will be via the use of a designated processor for reading out data from the sensors.

Chapter 10

Conclusions

At the outset of the project, the objectives for the project were set as,

- Develop a system to classify vehicles into the Austroads 94 12 Bin classification standard.
- The system should be non-invasive, that is, require that no apparatus be placed on or above the carriageway, and no person should be required to enter the carriageway to set up the system.
- The system should be capable of classifying vehicles in multi-lane, high speed environments
- Expand the algorithm to output data from the developed system to include information such as 85th percentile speed, 15k pace, allow for time exclusions, platooned vehicles as well as other various speed data.

All of the project requirements were able to be met, however there were some difficulties encountered on the way. In this section, I will attempt to outline the problems encountered. Some of the problems were solved temporarily, to a level sufficient that I could continue to press ahead with the project. Some of the problems will require further work, and while others are beyond the scope of the project.

10.1 Data Logger

One of the largest challenges for the project was to synchronize the two laser sensor outputs. As mentioned in earlier chapters, the ability to synchronize the data is of paramount importance. This was not achievable through use of the manufacturers hardware interface software due to the buffering of data, and limited success was achieved through using Matlab running on the Windows platform.

Although some success was achieved using Matlab to a point where the sensors were proved that they could work if synchronization were achieved, for any further progression of the system, a hardware data logger needs to be developed. I have contacted many manufacturers of data loggers and have been advised that there is nothing commercially available to suit the requirements of the system.

The data logger would have the following requirements,

- Two RS232 serial interfaces
- Timestamping of data with resolution of 0.001 seconds
- Sufficient data storage for 7 days of capture
- Be capable of reading and storing data from serial port in less than 0.5ms

One of the limiting factors in development of a data logger will be the data storage. The data capture for the duration of the project was typically of short duration. Most trials of the system were typically of less than thirty seconds. A thirty second data capture has a typical file size in the order of seventy-nine kilobytes. This equates to 2500 bytes for every second of capture.

A system could expect to be operated for a period of one week. This equates to a file size of 1.4 gigabytes. I see this as achievable by today's standard, with the availability of very small flash memory and removable hard drives.

The alternative is to accumulate the data in real time, on the street, as vehicles pass. By doing this, only vehicle information such as axles, groups, axle separations and speed, as presented in Chapter 8, need be recorded. By doing this, the data storage capacity will be reduced greatly.

10.2 ULS Interface

There may also be an opportunity to improve the ULS Hardware Interface Software. The software has an excellent interface for setup and configuration of the sensors, however the capture of output data could be improved.

A number of things could be done to improve the software. The ability to operate multiple sensors from one application would enable synchronization of senors. I could see this may be useful for many other applications apart from the one presented in the project. I was also made aware by a ULS company technician that when operating at five kilohertz PRF, the ULS would drop one shot as bad on every output string. This has a marginal effect on the data but becomes critical as the 'required good pulses' approaches the PPM parameter. Another situation where the data became unreliable was from the ULS not logging errors. For example, if the 'number of good pulses' parameter was not reached, an error is output from the ULS as 'Average not filled, "$ER,5". Due to the ULS not logging or recording the error and simply dropping it, a time skip results. This means that there is no data available for the time interval, but there is also no way of identifying the error due to no logging. If the error was logged, there is an opportunity to take action to correct the data via some sort of median filter.

These were minor problems encountered that were able to be overcome, and are offered here as suggestions for improvement to the already excellent product produced by Laser Technologies Incorporated.

10.3 Crossfall

This was an issue identified at the outset of the project. There is no solution to the problem of crossfall due to the fact that the carriageways must be constructed this way for drainage. The research conducted regarding crossfall was all from Department of Main Roads standard drawings.

Two lanes of traffic presented no problems as the fall across two lanes is uniform in one direction. Multilane carriageways, typically three or four lanes, are only encountered upon highspeed carriageways such as the M1 motorway between Brisbane and the Gold Coast. Due to safety considerations, I did not venture onto such a carriageway to confirm the research. In regards to this, I can only propose that if the carriageways are constructed as they are in drawings, allowing for some tolerances, the system will have limited capability of detecting vehicles in the far lane, if it were to have opposing crossfall. Only testing on such a carriageway could confirm the systems ability, and would be beyond my capabilities for the project.

10.4 Occlusion

Occlusion is the situation discussed in Chapter 2.2 where an axle in a far lane is obscured from the sensors by an axle in the near lane. I attempted to deal with the problem by collating the data from the two senors with limited success. It was found that some unclassifiable errors were introduced when applying this method that were otherwise fine without the technique. Occlusion of axles is a low probability event, and as such, in all of the testing done on real traffic, the situation did not occur.

It is a problem that is associated with all count and classification systems. In a pneumatic tube system, two vehicles striking the tubes simultaneously are not able to be distinguished as well as with a video system where a vehicle in a far lane may be shadowed by a larger vehicle, a semi-trailer, in the near lane.

The most important factor associated with occlusion is the ability to recover quickly from this event. One of the problems found in the research into the TIRTL, was that in the event of occlusion, the system took a long time to recover and produced a string of misclassified vehicles every time after this event. I believe that the method I have employed, placing these unclassifiable axle events into a cumulative array for later analysis allows the system to continue classifying vehicles unperturbed. The work on analyzing the unclassifiable data is for future work, but would be based on attempting to recover vehicles from the data where an axle may be missing. The vehicles recovered from the unclassifiable data could be flagged and the user offered the opportunity to include or exclude them from the classification report.

10.5 Weather

The final foreseeable problem associated with the system is the operation of the system in adverse weather conditions. Due to the prototype construction of the system being non-weatherproof, the system performance could not be evaluated in a situation where the road surface was covered in water. The height of the sensors above the carriageway may affect the sensors when spray from vehicles tyres is being produced. Experiments involving the sensors their ability to operate in the presence of water were not conducted due the high replacement cost of the sensors.

10.6 Summary

Overall, whilst there have been challenges associated with the project, the concept of using infra-red sensors as a replacement for pneumatic tubes has been proved. The idea of recording distance to a vehicle across a carriageway in order to determine it's position across the carriageway is one that can work. Ignoring the issue of synchronization of the sensors, it is possible to determine the speed of vehicles, and using this speed, calculate the axle spacing, determine the number of axles and determine grouping of axles. Once this data is successfully acquired, it is relatively simple to apply to the Austroads 94 12 bin classification system. This was the main objective of the project.

The system is truly non-invasive in the fact that it requires only that the sensors be setup on one side of the carriageway. There is no requirement for workers to enter the carriageway and the system does not affect motorists. This aspect of the system will have a positive impact on injuries sustained by workers setting up traffic count sites.

Appendix A

Project Specification

University of Southern Queensland

FACULTY OF ENGINEERING AND SURVEYING

ENG4111/4112 Research Project

PROJECT SPECIFICATION

FOR: **COLIN OTTO**

TOPIC: DEVELOPMENT OF A MOBILE VEHICLE CLASSIFICATION SYSTEM

SUPERVISORS: Mr Wei Xiang, University of Southern Queensland
Mr Jeshua Brouwer, Department of Main Roads, Qld

PROJECT AIM: The project aims to develop a portable vehicle counting system, capable of separating counted vehicles into the AustRoads 12-Bin classification system. The chosen system should be such that no devices or apparatus are placed on the carriageway and the detection system would not affect motorists in any way.

PROGRAMME: **Issue A, 26 March 2006.**

1. Investigate existing mobile traffic counting and classification methods and determine the accuracy of the systems.

2. Investigate emerging technologies in the field of counting and classification.

3. Develop a trial system that uses the technology chosen in the proposed research. The trial system should be able to count vehicles as per the AustRoads 12-Bin classification system and ideally would be able to identify vehicles in individual lanes, individual vehicle speeds and direction. The classification algorithm developed will be implemented through Matlab using field test data.

4. Develop a testing process to evaluate the accuracy of the trial system. The data should be compared against the earlier research into the accuracy of other existing counting devices and if possible, in various road and environmental conditions.

5. Expand the algorithm to output data from the developed system to include information such as 85th percentile speed, 15k pace as well as other various speed data.

As time permits:-

1. Investigate possible designs for roadside hardware, providing both security of the system and functionality.

2. Investigate converting the completed Matlab algorithm in a C language executable program, capable of running as a stand alone system.

Appendix B

Austroads '94 Vehicle Classification System

Level 1	Level 2		Level 3	AUSTROADS Classification		
Length (indicative)	Axles and Axle Groups		Vehicle Type			
Type	Axles	Groups	Description	Class	Parameters	Dominant Vehicle
Short Up to 5.5m	LIGHT VEHICLES					
		1 or 2	**Short** Sedan, Wagon, 4WD, Utility, Light Van, Bicycle, Motorcycle, etc.	1	d(1) <= 3.2m and axles = 2	
Medium 5.5 to 14.5 m	3, 4 or 5	3	**Short – Towing** Trailer, Caravan, Boat, etc.	2	groups = 3, d(1)>= 2.1m, d(1) <= 3.2 m, d(2) >= 2.1m and axles = 3, 4 or 5	
	HEAVY VEHICLES					
	2	2	**Two Axle Truck or Bus**	3	d(1) > 3.2m and axles = 2	
	3	2	**Three Axle Truck or Bus**	4	axles = 3 and groups = 2	
	>3	2	**Four Axle Truck**	5	axles > 3 and groups = 2	
Long 11.5 to 19.0 m	3	3	**Three Axle Articulated** Three axle articulated vehicle, or Rigid vehicle and trailer	6	d(1) > 3.2m, axles = 3 and groups = 3	
	4	>2	**Four Axle Articulated** Four axle articulated vehicle, or Rigid vehicle and trailer	7	d(2) < 2.1m or d(1) < 2.1m or d(1) > 3.2 m, axles = 4 and groups > 2	
	5	>2	**Five Axle Articulated** Five axle articulated vehicle, or Rigid vehicle and trailer	8	d(2) < 2.1m or d(1) < 2.1m or d(1) > 3.2 m, axles = 5 and groups > 2	
	>=6	>2	**Six Axle Articulated** Six (or more) axle articulated vehicle, or Rigid vehicle and trailer	9	axles = 6 and groups > 2 or axles > 6 and groups = 3	
Medium Combination 17.5 to 36.5 m	>6	4	**B-Double** B-Double, or Heavy truck and trailer	10	groups = 4 and axles > 6	
	>6	5 or 6	**Double Road Train** Double Road Train, or Heavy truck and two trailers	11	groups = 5 or 6 and axles > 6	
Long Combination Over 33.0 m			**Triple Road Train** Triple Road Train, or Heavy truck and three trailers	12	groups > 6 and axles > 6	

Figure B.1: Austroads 94 Classification Standard

Appendix C

Department of Main Roads, Occupational Health and Safety Standard

This document contains commercial in confidence information. It is expected that the information contained in the document is understood to be used as directives for Department of Main Roads personnel only.

C.1 Intent

The intent of this standard is to detail the minimum requirements for all departmental personnel who work in proximity to traffic. Tasks that may be covered under this standard include traffic engineering investigations, planning, developing, viewing, measuring and monitoring of sections of the roadway or road features.

This standard supports and in some cases exceeds the "Manual of Uniform Control Traffic Devices Part 3 Working of Roads" . Where this standard is higher, then the controls listed are to be applied by all departmental personnel.

C.2Risk Summary

C.2.1 Benefits of Implementation

The potential for injury to personnel is significantly decreased when effective traffic control measures are implemented.

C.3 Legislative References

- Workplace Health and Safety Act 1995
- Traffic Operations (Road Use Management) Act 1995

C.3.1 Definitions

MUTCD - Manual of Uniform Traffic Control Devices

Road or Railway - has the same meaning as that in the MUTCD and is, "That portion of the road particularly devoted to the use of vehicles, inclusive of shoulders and auxiliary lanes".

C.4 Requirements

C.4.1 Elimination of Risk

If possible, personnel should eliminate the risk by conducting tasks in manner that does not require access to the roadway, such as by using aerial photographs. However, it is recognized that direct physical observation is often the only way of conducting tasks.

C.4.2 Traffic Management Plan

Where risk of elimination is not possible, control can be achieved through the implementation of an approved traffic management plan in accordance with the "MUTCD Pt 3 Work On Roads". A traffic management plan is to be implemented for all work that impacts on the roadway including lane closures and any activity that requires changes to regulatory signage.

C.4.3 Working in Proximity to Traffic

The level of risk when working in proximity to traffic is dependent on the traffic speed, traffic volume and the length of time (duration) that personnel will be exposed. The document "Working in Proximity to Traffic" details a procedure to determine the length of time personnel may access the roadway dependent on these factors. Where a traffic management plan is not implemented and personnel are required to conduct any task within 1.2 metres of traffic in a 70 km/hr or greater zone, then they are to apply the procedure listed in "Working in Proximity to Traffic", a printout of which is to be carried.

While the "MUTCD Part 3 Work on Roads" describes access durations on the roadway of 5 minutes or less, the implementation of this procedure allows for longer durations based on risk assessment at very low volumes. Advance warning signage is not required for the activities conducted under this procedure. The criteria used in this procedure is based on the number of vehicles per hour, as this reflects actual conditions, rather that the AADT (Annual Average Daily Traffic count) which is based on a 24 hour average.

C.4.3.1 Exceptions

Personnel may operate without restriction in the following environments when,

- Utilizing existing footpaths
- Whilst remaining in their vehicles
- Operating more than 3 metres beyond the edge of the roadway
- Operating behind fixed guardrail or temporary concrete barriers

C.4.4 Crossing Multi-lane Roads

Crossing of multi-lane roads such as freeways and motorways as a pedestrian can pose a significant risk. The crossing of multi-lane roads as a pedestrian is permitted where all of the following conditions can be met,

1. The procedure "Working in Proximity to Traffic" is fully implemented, and
2. Three lanes or less are being crossed, and
3. Sight distance to approaching vehicles is 250 metres or greater, and
4. A set point marker at 250 metres or greater is identified prior to accessing the roadway.

C.4.4.1 Vehicle Speed and Sight Distance

The "MUTCD Part 3 Work on Roads" and the procedure detailed in this standard proscribe a sight distance of 250 metres. A vehicle traveling at 60 km/hr will take 15 seconds to cover 250 metres; at 90 km/hr it will take approximately 11 seconds to travel 250 metres and at 120 km/hr it will take 7.5 seconds to travel 250 metres. Therefore the 250 metres sight distance proscribed in this procedure is to be strictly adhered to, as it provides the minimum distance that will allow personnel to remove themselves from danger.

C.4.5 Vehicular and Personal Protective Equipment

All vehicles stopping on or beside the roadway are to be fitted with a vehicle mounted warning device in accordance with the "MUTCD Part 3 Work on Roads" except where the use of such device would increase the risk or impede the activity. Such exceptions are a lessening of the minimum standards listed in MUTCD Part 3 and therefore are to be authorized and documented by the business unit manager. This requirement does not apply where the vehicle is parked in a legal parking position, in a marked enforcement bay or more than 3 metres from the edgeline.

As a minimum, all personnel conducting inspection must wear a high visibility safety vest, a broad brimmed hat (unless designated hard-hat area) and enclosed footwear (preferably safety shoes/boots).

Appendix D

Matlab Code

Serial Data Collection Code

Uses 2 subfunctions:- serialdata and dataCollect

```
% CollectData.m
% Written By Colin Otto 0011121354
%------------------------------------
% Clear workspace and command window
clear; clc
%------------------------------------
%------------------------------------
% User specified length of data recording
TimeLength = input('Time to collect data? (Seconds)\n');
% Function to read serial ports
[data1,data2]=serialdata(TimeLength);
% Extract text information and convert
% convert to floating point measurements
[dist1,dist2]=dataCollect(data1,data2);
%------------------------------------
%------------------------------------
% Clean up
clear data1 data2 dt time
%------------------------------------
%------------------------------------
% User to specify filename
fName = input('Enter filename to save traffic data into?\n','s');
% Save data in file specified by user
save(fName)
%------------------------------------
```

Code to access serial ports on com5 and com7

```
function [data1,data2] = serialData(TimeLength)
%SERIALDATA M-Code for communicating with the instruments.
% These steps are:
%       1. Create an instrument object
%       2. Connect to the instrument
%       3. Configure properties
%       4. Read data
%       5. Disconnect from the instrument
%------------------------------------
% Number of measurements to record
amount = TimeLength/0.004;
%------------------------------------
% Com 5 Set-up ( Laser sensor 1 )
% Create a serial port object.
obj1 = instrfind('Type', 'serial', 'Port', 'COM5', 'Tag', '');
% Create the serial port object if it does not exist
% otherwise use the object that was found.
if isempty(obj1)
    % Laser 2
    obj1 = serial('COM5');
else
    fclose(obj1);
    obj1 = obj1(1)
end
%------------------------------------
% Com 7 Set-up ( Laser sensor 2 )
% Create a serial port object.
obj2 = instrfind('Type', 'serial', 'Port', 'COM7', 'Tag', '');
% Create the serial port object if it does not exist
% otherwise use the object that was found.
if isempty(obj2)
    % Laser 1
    obj2 = serial('COM7');
else
    fclose(obj2);
    obj2 = obj2(1)
end
%------------------------------------
```

```
% Connect to instrument object, obj1.
fopen(obj1); fopen(obj2);
%-----------------------------------
% Set-up com5 and com7
set(obj1,'BaudRate',115200,'Terminator','cr');
set(obj2,'BaudRate',115200,'Terminator','cr');
%-----------------------------------
% Flush both serial input buffers
flushinput(obj1); flushinput(obj2)
% Read required amount of measurements
for i=1:amount(n)
    data1{i,1} = fscanf(obj1, '%s\n');
    data2{i,1} = fscanf(obj2, '%s\n');
end
% Disconnect all objects.
fclose(obj1); fclose(obj2);
% Clean up all objects.
delete(obj1); delete(obj2);
%-----------------------------------
```

Code to convert sensor output to required format

```
function [dist1,dist2] = dataCollect(data1,data2)
n=length(data1);
% Convert text string output to required format
for i = 20:n
    [bm,dist2(i)] = strread(data1{i},'%s%f','delimiter',',');
    [bm,dist1(i)] = strread(data2{i},'%s%f','delimiter',',');
end
dist1 = dist1(20:n);
dist2 = dist2(20:n);
```

Classification algorithm

```
% ULSClassifier.m
% Written By Colin Otto 0011121354
% Clear workspace and command window
clear; clc
%-------------------------------------------------------------
% Distance between Laser 1 and Laser 2
LaserSeperation=1; % 1 metre
laneWidth = 3.5; % 3.5 metres
%-------------------------------------------------------------
% Parameters of the ULS
PRF = 4000; PPM = 16;
% Measurement Period
dt = PPM/PRF;
% Measurement Frequency
Fs = 1/dt;
%-------------------------------------------------------------
% Maximum axle spacing in Austroads 94 Classification Standard
maxAxleSpace = 10; % 10 metres
%-------------------------------------------------------------
% file containing traffic data
fname = input('File containing traffic data?\n','s');
load fname;
distance1=dist2; distance2 = dist1;
% Clean up variables
clear dist1 dist2
% Data length power of 2 for Mallet Algorithm in Matlab
a = floor(log2(length(dist1)));
distance1 = distance1(1:2^a);
distance2 = distance2(1:2^a);
time1 = 0:dt:dt*(length(distance1)-1); % Laser 1 time data
time2 = 0:dt:dt*(length(distance2)-1); % Laser 2 time data
%-------------------------------------------------------------
%-------------------------------------------------------------
% Wavelet Denoising
% Minimax Thresholding
% Hard Thresholding
% No rescaling of noise
% Level 6 Decomosition
% Haar or DB1 Mother Wavelet
 distance1 = medianFilter(distance1);
 distance2 = medianFilter(distance2);
 [distance1,CXD,LXD] = wden(distance1,'rigrsure','h','one',6,'haar');
 [distance2,CXD,LXD] = wden(distance2,'rigrsure','h','one',6,'haar');
%-------------------------------------------------------------
```

```
%--------------------------------------------------------------
% Remove any Deep Fade in Measurement Data
 distance1 = medianFilter(distance1);
 distance2 = medianFilter(distance2);
%--------------------------------------------------------------
%--------------------------------------------------------------
% Lane Seperation
 laneStart(1) = min(distance1)-0.6;
 laneStart(2) = laneStart(1) + laneWidth;
 laneStart(3) = laneStart(2) + laneWidth;
 laneStart(4) = laneStart(3) + laneWidth;
 laneStart(5) = laneStart(4) + laneWidth;
%--------------------------------------------------------------
%--------------------------------------------------------------
% Find other side of road
OSOR = (max(distance1) + max(distance2))/2;
distance1(find(distance1>OSOR-1))=0;
distance2(find(distance2>OSOR-1))=0;
%--------------------------------------------------------------
%--------------------------------------------------------------
% Vehicles in Lanes
figure
 plot(distance1)
 hold on
 for i = 1:5
    Threshold = laneStart(i)*ones(size(distance1));
    plot(Threshold,'r:')
 end
 plot(distance2,'k')
%--------------------------------------------------------------
% Work out how many lanes
maxDist1 = max(distance1); maxDist2 = max(distance2);
 if maxDist1 > maxDist2
    maxDist = maxDist1;
 else
    maxDist = maxDist2;
 end
 if maxDist < laneStart(2) + 1.5
    lanes = 1;
 elseif maxDist < laneStart(3) + 1.5
    lanes = 2;
 elseif maxDist < laneStart(4) + 1.5
    lanes = 3;
 else
    lanes = 4;
 end
%--------------------------------------------------------------
```

```
%------------------------------------------------------------
% Seperate into Lanes
 [a,b]=size(distance1);
 Distance1 = zeros(a,b,lanes);
 Dist1 = zeros(a,b);
 [a,b]=size(distance2);
 Distance2 = zeros(a,b,lanes);
 Dist2 = zeros(a,b);
 for i=1:lanes
    % Laser 1
    Dist1(:,:)=0;
    Dist1(find(distance1>=laneStart(i) & distance1<laneStart(i+1)))...
        = distance1(find(distance1>=laneStart(i) & ...
        distance1<laneStart(i+1)));
    Dist1(find(Dist1>0))=1;
    Distance1(:,:,i)=Dist1;

    % Laser 2
    Dist2(:,:)=0;
    Dist2(find(distance2>=laneStart(i) & distance2<laneStart(i+1)))...
        = distance2(find(distance2>=laneStart(i) & ...
        distance2<laneStart(i+1)));
    Dist2(find(Dist2>0))=1;
    Distance2(:,:,i)=Dist2;
 end
%------------------------------------------------------------
%------------------------------------------------------------
count = 1; % Vehicle Count of total vehicles
for lane = 1:lanes
    % Operate on only one lane at a time
    distance1 = Distance1(:,:,lane);
    distance2 = Distance2(:,:,lane);
    % Make length of 2 arrays equal
    if length(distance1) > length(distance2)
        tempLen = length(distance1);
    else
        tempLen = length(distance2);
    end
    distance1 = [distance1,[zeros(1,tempLen-length(distance1))]];
    distance2 = [distance2,[zeros(1,tempLen-length(distance2))]];

    %--------------------------------------------------------
    % Find the edges using gradient filter
    ddistdt1=gradientFilt(distance1,dt);
    ddistdt2=gradientFilt(distance2,dt);
    %--------------------------------------------------------
```

```
%---------------------------------------------------------
exitFlag = 0;
while max(distance1) ~=0 & max(distance2)~=0
    start1 = min(find(ddistdt1>0))-1; % 1st axle crosses laser 1
    start2 = min(find(ddistdt2>0))-1; % 1st axle crosses laser 2
    speed = LaserSeperation / ((start2 - start1) * dt); % m/s

    %Leading and Trailing edges of tyres on both lasers
    % LE is Leading Edge
    % TE is Trailing Edge
    L1LE = (find(ddistdt1>0))-1;
    L1TE = (find(ddistdt1<0))-1;
    L2LE = (find(ddistdt2>0))-1;
    L2TE = (find(ddistdt2<0))-1;

    % Distance between axles at speed of first axle
    AxleSepL1 = (diff(L1LE)*dt + diff(L1TE)*dt)* speed/2;
    AxleSepL2 = (diff(L2LE)*dt + diff(L2TE)*dt)* speed/2;

    VehicleEnd = L2TE(min(find(AxleSepL2 > 10)))+ 3;
    if length(VehicleEnd) == 0
        VehicleEnd = L2TE(find(L2TE==max(L2TE))) + 5;
    end

    % Extract Current Vehicle
    CurrentVehicleL1 = distance1(1:VehicleEnd+3);
    CurrentVehicleL2 = distance2(1:VehicleEnd+3);

    dCVL1dt = gradientFilt(CurrentVehicleL1,dt);
    axleStrikes = find(dCVL1dt>0);
    axles = length(axleStrikes);

    % Find Seperations of axles at speed of first axle
    for i = 2:length(axleStrikes)
        axleSep(i-1) = (axleStrikes(i) - axleStrikes(i-1))* dt * speed;
    end

    % Axle group is where adjacent axle spacing is less than 2.1 m
    groups = axles - length(find(axleSep>0 & axleSep<=2.1));
    D1 = axleSep(1); % Distance between first 2 axles
    % Distance between 2nd and 3rd axles
    if length(axleSep)>1
        D2 = axleSep(2);
    else
        % No third axle
        D2 = 0;
    end
```

```
        % Array for classification data
        Vehicle(count,:) = [axles, groups, D1, D2 , speed,lane];
        count = count + 1; % Increment vehicle counter

        % Update Distance Data
        EndData = length(distance1);
        distance1 = distance1(VehicleEnd+1:EndData);
        distance2 = distance2(VehicleEnd+1:EndData);
        ddistdt1 = ddistdt1(VehicleEnd+1:EndData);
        ddistdt2 = ddistdt2(VehicleEnd+1:EndData);
        axleSep=[];
    end
end
% Classify vehicles
 unclassifiable = [];
 for i = 1:size(Vehicle,1)
    class = classify(0,Vehicle(i,:));
    if class == 0
        unclassifiable = [unclassifiable;Vehicle(i,:)];
    else
        % Output to text file
        fid = fopen('Output Data.txt', 'a');
        fprintf(fid,'Vehicle Class %i, Lane %i, Speed %.1f km/hr\n'...
            ,class,Vehicle(i,6), Vehicle(i,5)*3.6);
        fclose(fid);
    end
 end
```

Median filter code

```
function y1 = medianFilter(y)
% Apply Median Filter to remove any deep fade
% Size of median filter equals 3
 y1=y;
 for i = 2:length(y)-1
    y1(i)=median(y(i-1:i+1));
 end
```

Gradient filter code

```
function dvectordx = gradientFilt(vector,dx)
% Gradient Filter
for i = 2: length(vector)
    dvectordx(i)= [vector(i-1:i)]*([-1,1]').dx;
end
```

Classification driver

Uses 11 individual class determining functions.

```
function class = classify(class,axleData)
% Test vehicle against each vehicle class
% When matching class found remaining tests fail
    if class == 0;
        class = class1or2(axleData);
    end
    if class == 0;
        class = class3(axleData);
    end
    if class == 0;
        class = class4(axleData);
    end
    if class == 0;
        class = class5(axleData);
    end
    if class == 0;
        class = class6(axleData);
    end
    if class == 0;
        class = class7(axleData);
    end
    if class == 0;
        class = class8(axleData);
    end
    if class == 0;
        class = class9(axleData);
    end
    if class == 0;
        class = class10(axleData);
    end
    if class == 0;
        class = class11(axleData);
    end
    if class == 0;
        class = class12(axleData);
    end
```

Individual class determining functions

Each of the following functions is called from within the Classify function. The output of each class function is either the class if a match is found, or if no match is found, zero is returned. Within Classify function, a zero indicates that searching should continue within classify.

Class 1 and 2

```
function [class] = class1or2(axleData)
%---------------------------------------------------------------
% Input: axle data for 3 axle groups
% Output: Vehicle Class 1 or 2; if 0, no match found
%---------------------------------------------------------------
% Paramaeters
% ( Axles, Groups, D1, D2, Speed, Lane )

% Class 1 is a car
% Parameters are d(1) < 3.2m and axles = 2
if (axleData(1,3) < 3.2)... % d(1)
        & (axleData(1,4) == 0)... % Make sure it is not towing
        & (axleData(1,1)==2) %Number of axles = 2
    % Return the class
    class = 1;
%---------------------------------------------------------------
% Class 2 is a car towing a trailer
% Parameters are:-
% d(1) < 3.2 and d(2) > 2.1 and axles <= 5 and groups = 3
elseif (axleData(1,3) < 3.2)... % d(1) < 3.2
        & (axleData(1,3) > 2.1)... % First axles not a group
        & (axleData(1,4) < 10)... % Make sure it is towing
        & ((axleData(1,2)>=3) & (axleData(1,2)<=5)) % Axles 3,4 or 5
    % Return the class
    class = 2;
end
%---------------------------------------------------------------
%EOF
```

Class 3

```
function [class] = class3(axleData)
%--------------------------------------------------------------
% Input: axle data for 2 axle groups
% Output: Vehicle Class 3; if 0, no match found
%--------------------------------------------------------------
% Paramaeters
% ( Axles, Groups, D1, D2, Speed, Lane )
% Class 3
% Class 3 is a 2 axle truck or bus
% Parameters are d(1) > 3.2m and axles = 2
if (axleData(1,3) > 3.2)... % d(1) > 3.2m
        & (axleData(1,1)==2) %Number of axles = 2
    class = 3;
else
    class = 0;
end
%--------------------------------------------------------------
%EOF
```

Class 4

```
function [class] = class4(axleData)
%--------------------------------------------------------------
% Input: axle data for 2 axle groups
% Output: Vehicle Class 3; if 0, no match found
%--------------------------------------------------------------
% Paramaeters
% ( Axles, Groups, D1, D2, Speed, Lane )
% Class 4
% Class 4 is a 4 axle truck or bus
% Parameters are axles = 3 and groups = 2
if ((axleData(1,1)==3)...% Number of axles = 3
        & (axleData(1,2) == 2)) % Groups = 2
    class = 4;
else
    class = 0;
end
%--------------------------------------------------------------
%EOF
```

Class 5

```
function [class] = class5(axleData)
%---------------------------------------------------------------
% Input: axle data for 2 axle groups
% Output: Vehicle Class 5; if 0, no match found
%---------------------------------------------------------------
% Paramaeters
% ( Axles, Groups, D1, D2, Speed, Lane )
% Class 4
% Class 4 is a 4 axle truck or bus
% Parameters are axles > 3 and groups = 2
if (axleData(1,1) >3)...%Number of axles > 3
        & (axleData(1,2) == 2) % Groups = 2
    class = 5;
else
    class = 0;
end
%---------------------------------------------------------------
%EOF
```

Class 6

```
function [class] = class6(axleData)
%---------------------------------------------------------------
% Input: axle data for 3 axle groups
% Output: Vehicle Class 6; if 0, no match found
%---------------------------------------------------------------
% Paramaeters
% ( Axles, Groups, D1, D2, Speed, Lane )
% Class 6
% Class 6 is a 3 axle articulated or rigid vehicle and trailer
% Parameters are d(1) > 3.2, axles = 3 and groups = 3
if axleData(1,3) > 3.2... % D1 > 3.2
    & (axleData(1,1) == 3)...% Number of axles = 3
        & (axleData(1,2) == 3) % Groups = 3
    class = 6;
else
    class = 0;
end
%---------------------------------------------------------------
%EOF
```

Class 7

```
function [class] = class7(axleData)
%--------------------------------------------------------------
% Input: axle data for 4 axle groups
% Output: Vehicle Class 7; if 0, no match found
%--------------------------------------------------------------
% Paramaeters
% ( Axles, Groups, D1, D2, Speed, Lane )
% Class 7
% Class 7 is a 4 axle articulated or rigid vehicle and trailer
% Parameters are d(2) <2.1 or d(1) <2.1 or d(1) >3.2
%   axles = 4 and groups >2
if (axleData(1,4) < 2.1 | axleData(1,3) < 2.1 | axleData(1,3) >
3.2)...
        & (axleData(1,1) == 4) & (axleData(1,2) > 2)
    class = 7;
else
    class = 0;
end
%--------------------------------------------------------------
%EOF
```

Class 8

```
function [class] = class8(axleData)
%--------------------------------------------------------------
% Input: axle data
% Output: Vehicle Class 8; if 0, no match found
%--------------------------------------------------------------
% Paramaeters
% ( Axles, Groups, D1, D2, Speed, Lane )
% Class 8
% Class 8 is a 5 axle articulated or rigid vehicle and trailer
% Parameters are d(2) <2.1 or d(1) <2.1 or d(1) >3.2
%   axles = 5 and groups >2
if (axleData(1,4) < 2.1 | axleData(1,3) < 2.1 | axleData(1,3) > 3.2)
   ...     & (axleData(1,1) == 5) & (axleData(1,2) > 2)
    class = 8;
else
    class = 0;
end
%--------------------------------------------------------------
%EOF
```

Class 9

```
function [class] = class9(axleData)
%--------------------------------------------------------------
% Input: axle data
% Output: Vehicle Class 9; if 0, no match found
%--------------------------------------------------------------
% Paramaeters
% ( Axles, Groups, D1, D2, Speed, Lane )
% Class 9
% Class 9 is a 6 axle articulated or rigid vehicle and trailer
% Parameters are    (axles = 6 and groups >2) or
%                   (axles > 6 and groups =3)
if ((axleData(1,1) == 6) & (axleData(1,2) > 2))...
        | ( (axleData(1,1) > 6) & (axleData(1,2) == 3))
    class = 9;
else
    class = 0;
end
%--------------------------------------------------------------
%EOF
```

Class 10

```
function class = class10(axleData)
%--------------------------------------------------------------
% Input: axle data
% Output: Vehicle Class 10; if 0, no match found
%--------------------------------------------------------------
% Paramaeters
% ( Axles, Groups, D1, D2, Speed, Lane )
% Class 10
% Class 10 is a B Double or a Heavy Truck and trailer
% Parameters are    groups = 4  and axles > 6
if  axleData(1,1) > 6 & axleData(1,2) == 4
    class = 10;
else
    class = 0;
end
%--------------------------------------------------------------
%EOF
```

Class 11

```
function [class] = class11(axleData)
%--------------------------------------------------------------
% Input: axle data
% Output: Vehicle Class 11; if 0, no match found
%--------------------------------------------------------------
% Class 11
% Class 11 is a Double Road Train or a Heavy Truck and  2 Trailers
% Parameters are    groups = 5 or 6  and axles > 6
if  axleData(1,1) > 6 & ( axleData(1,2) == 5 | axleData(1,2) == 6 )
    class = 11;
else
    class = 0;
end
%--------------------------------------------------------------
%EOF
```

Class 12

```
function [class] = class12(axleData)
%--------------------------------------------------------------
% Input: axle data
% Output: Vehicle Class 12; if 0, no match found
%--------------------------------------------------------------
% Class 12
% Class 12 is a Triple Road Train or a Heavy Truck and 3 Trailers
% Parameters are    groups > 6  and axles > 6
if  (axleData(1,1) > 6) & (axleData(1,2) > 6)
    class = 12;
else
    class = 0;
end
%--------------------------------------------------------------
%EOF
```

Code for Statistical Analysis of Traffic Data

When calling this function, the format of the output of the Classification algorithm is,

$$[\text{class, lane, speed, time}]$$

```
function statisticalAnalysis
%---------------------------------
% Load traffic data file
fid = 'OutputData.txt';
%---------------------------------
% Extract data & convert string to double
[class, lane, speed, time] = textread...
    (fid,'%s %s %s %s',-1,'delimiter',',');
for i = 1:size(speed,1)
    speeds(i,:)=str2num(speed{i}(1,:));
    times(i,:) = str2num(time{i}(1,:));
end
 speed = speeds';
 times = times';
%---------------------------------
%---------------------------------
% Sort of speed to 0.1 km/hr precision
speeds = [0:.1:150];
% Array to count occurences
histo = zeros(size(speeds));
%---------------------------------
% Include Headway filtering or not
HW = menu('Do you wish to use headway filtering?','Yes','No');
 if HW == 1
    HWspeed = [];
    HWtime = menu('Headway time?','1 Sec','2 Sec','3 Sec','4 Sec')
        for i = 2:length(speed)
            if ((times(i) - times(i-1)) < HWtime)
            % Do nothing
            else
                HWspeed = [HWspeed,speed(i)];
            end
        end
    speed = HWspeed;
end
%---------------------------------
```

```
%---------------------------------
% 85th Percentile speed
for i = 1 : length(speed)
    current = speed(i);
    index = find(current == speeds);
    histo(index) = histo(index) + 1;
end
 cumulative = cumsum(histo);
 cumulative1 = cumulative/max(cumulative);
%---------------------------------
% 15k Pace
for i = 7:length(cumulative)-8
    k15p(i) = sum(histo(i-6:i+8));
end
 index = find(k15p == max(k15p));
 K15Pace = speeds(index);
%---------------------------------
```

Appendix E

Glossary

Annual Average Daily Traffic AADT is the total volume of traffic passing a roadside observation point over the period of a calendar year, divided by the number of days in that year (365 or 366 days).

Axle counts This is the number of actuations on an axle sensor such as a pneumatic tube as the wheels of vehicles cross over the sensor.

Design Hour Volume DHV is traffic flow rate chosen as the design traffic load for a facility over its design life. Common practice is to choose an nthHHV as the design volume, with the 30th highest hourly volume often used in a rural environment and the 80th HHV in an urban area. The 100 HV is used on National Highways.

Averaging Mode This conventional mode averages multiple pulses or shots. In this mode, the ULS measures quicker and is more accurate. This is a result of LTI's proprietary ASIC and use of high-speed CPU processing on-board.

Target: The surface that a laser spot hits, from which light is reflected to the detector in an optical sensor. This may refer to any type of surface or material at which a sensor is pointed or to a specific object or material designed to reflect light. Target reflectance is the most important factor in determining the maximum range of a sensor.

Cooperative Target: A target that is designed to reflect light to the detector of a sensor. Cooperative targets include glass corner cube retroreflectors and retroreflective tape made by several manufacturers. In some applications, mirrors may also be used as cooperative targets.

Uncooperative Target: Uncooperative targets include any surfaces or material being measured that is not specifically designed to reflect light to the sensor. This includes shiny metal or painted surfaces, liquids, and loose or granular solid substances.

References

Aamport, L. A. (2000), 'Development of an Austroads Heavy Vehicle Nomenclature System', *Austroads Publication* **AP-R174**.

AP-G11.3-04 Traffic Studies (2004), Austroads Incorporated. Guide to Traffic Engineering Practice, Part 3.

Bastian, V. (1994), 'Midland Road Train Trial', *http://www.metrocount.com* .

Chui, C. K. (1995), *Wavelet Basics*, Kulwer Academic Publishers.

Kreyszig, E. (1999), *Advanced Engineering Mathematics*, John Wiley and Sons.

Kumar, B. P. (2005), *Digital Signal Processing Laboratory*, Taylor and Francis.

Leis, J. W. (2002), *Digital Signal Processing: A MATLAB Based Tutorial Approach*, Research Studies Press Ltd.

Lynn, P. A. & Fuerst, W. (1999), *Introductory Digital Signal Processing with Computer Applications*, John Wiley and Sons.

Mallet, S. (1989), 'A theory for multiresolution signal decomposition: the wavelet representation', *"IEEE Pattern Analysis and Machine Intelligence"* **11**.

Martin, P. T., Feng, Y. & Wang, X. (2006), Detector technology evaluation, Technical report, University of Utah, Traffic Lab. http://www.mathworks.com/access/helpdesk/help/techdoc/matlab.shtml.

Matlab Helpfiles (2006), Technical report, The Mathworks. http://www.mathworks.com/access/helpdesk/help/techdoc/matlab.shtml.

Metrocount (2001), *Discussion of MetroCount 5600 Series Roadside Unit speed accuracy.*, Metrocount.
http://www.metrocount.com/downloads/files/flyers/SpeedAccuracy.pdf
current August 2006.

Meyers, Y. (1993), *Wavelets, Algorithms and Applications*, Society for Industrial and Applied Mathematics.

Pell, B. (2002), 'T47-100 and V2.25 Test in Tasmanian Conditions', *http://www.metrocount.com* .

Road Planning and Design Manual (2004).

Taylor, M., Bonsall, P. & Young, W. (2000), *Understanding Traffic Systems: Data, Analysis and Presentation (2nd Ed)*, Ashgate, Aldershot, UK.

TIRTL Product Overview (n.d.), Technical report.

Traffic & Road Use Management Division ITS Infrastructure (2002), The Infra-Red Traffic Logger (TIRTL) Evaluation, Technical report.

TRANSPORT OPERATIONS (ROAD USE MANAGEMENT–MASS, DIMENSIONS AND LOADING) REGULATION (2005), *"Queensland Consolidated Regulations"* **27**.

ULS Hardware/Software Interface Specifications (2005), Technical report, Laser Technologies Incorporated.

Wavelet Basics (2006), Technical report, Stanford Starlab Wavelet Research Group.
http://www-star.stanford.edu/projects/sswrg/basics.html.

Working In Proximity to Traffic (2005), Technical report, Queensland Department of Main Roads.