University of Alberta

An intelligent seat occupancy detection system

By

Guillermo Barreiro ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Science

Department of Electrical and Computer Engineering

Edmonton, Alberta
Spring 2004

Your file Votre référence
ISBN: 0-612-96449-3
Our file Notre référence
ISBN: 0-612-96449-3

ABSTRACT

Today's electronic systems rely on pattern recognition for a variety of tasks. One of the industrial sectors that has broadly explored and used pattern recognition for its continuous improvement is the automotive industry, especially in motor vehicle safety.

Air bags systems are the main component of the motor vehicle safety systems included in any car. Unfortunately, deaths reportedly have been caused by airbags inflating in low severity crashes. To reduce the number of fatalities produced by airbags, car manufacturers are urged to create intelligent airbag deployment systems. The main intention of this work is to design and implement the pattern recognizer for a prototype of a car seat occupancy system using artificial intelligence methods, namely fuzzy clustering, self-organizing maps, and neural networks. Experimentation on the final systems is also described, and accuracy in recognition, robustness to failures and possible improvements are reported.

To the people who bring inspiration to my life:
my daughter Alejandra,
my wife Yency,
my sister Lina and
my parents Guillermo y Alba

ACKNOWLEDGMENT

The author would like to thank:

Dr. Petr Musilek for his continuous support, valuable feedback and patience through the development of this project.

Ecotemp International for providing data and other resources for this project.

My friends Motoko, Yukie, Jorge, Gabriel, Omar, Mariana and the Mexican bunch for making life in Edmonton a little more like home.

Valenverg Futbol Club because to lose is to win a little.

TABLE OF CONTENTS

List of Tables

List of Figures

Chapter 1.

Introduction

1.1 Preface

Humans are very good at recognizing friendly faces, printed characters, musical melodies and most of the environment that surrounds them. The senses process signals such as sound or light waves that have been modulated and then map them into some kind of recognition space. The blueprint of these signals can be considered a *pattern*. When it is decided that an object from a population P belongs to a known subpopulation S, then *pattern recognition* is being done [16].

Today's electronic systems rely on pattern recognition for a huge variety of tasks: speech control on autonomous environments for people with physical limitations [4], star telemetry on space satellite images [3], cost estimation on software development projects [13], etc. The list gets longer as scientists explore the unreachable boundaries of pattern recognition techniques, and the applications given to them are turning more and more interesting every day.

In general, almost any "intelligent" or "smart" device or decision-making system contains some kind of pattern recognition module implemented inside it. Industrial applications follow this trend and are continuously demanding faster, smarter and more robust recognition systems. One of the industrial sectors that has broadly explored and used pattern recognition for its continuous improvement is the automotive industry. From the use of vision guided robots for the assembly of car parts [34] to the traction control in formula 1 race cars [35], pattern recognition is employed to the fullest. Recently, one of the areas in which the automotive industry is focusing the use of pattern recognition is motor vehicle safety.

Motor vehicle safety is one of the main concerns in today's societies. There are about 10'000.000 crashes involving injured people in North America, the European Community and Japan each year [21]. In the United States 85,011 people died and 5'959,000 were injured in traffic accidents in the years 2001 and 2002 [28]. In the province of Alberta 768 people died and 54,047 were injured in the years 2000 and 2001 [29][30].

Although alarming, this statistics could be much higher if airbag protection systems were not used. These systems are one of the main components in the motor vehicle safety features included in any car. Deaths in frontal crashes are reduced about 26% among drivers using safety belts and about 32% among drivers without belts by use of airbags systems [31].

However, there is a down side to the use of airbag protection systems. Since 1990, 231 deaths reportedly have been caused by airbags inflating in low severity crashes [31]. This is considered a high number for a car feature that is supposed to save lives. To reduce the number of fatalities produced by airbags the U.S federal government urged car manufacturers to create intelligent airbag deployment systems. These systems should be capable of distinguishing which type of passenger is sitting in the front seat of a car or small truck and should deploy the air bag with a pressure in accordance to the weight or other characteristics of the subject in the passenger's seat.

The development of such system is a demanding task for the use of pattern recognition techniques. High dimensionality of the inputs, cross over on the training data and slow real time performance could be among the possible bottlenecks in the development of such project. Overcoming them represents an ideal challenge for a master's of science thesis project.

The main intention of this work is to design and implement the pattern recognizer for a prototype for a car seat occupancy system (CSOS) based on the regulations stated by the U.S. federal government through the National Highway Traffic Safety Administration (NHTSA). This work was developed in cooperation with Ecotemp International, a local producer of automotive related products for Canada and USA

1.2 Background on the problem

In the Intermodal Surface Transportation Efficiency Act of 1991 (ISTEA), the U.S. congress directed the NHTSA to amend Federal Motor Vehicle Safety Standard 208 (FMVSS 208) to require all passenger cars and light trucks to provide automatic occupant protection by means of air bags. In addition, all the states of the union approved the mandatory use of child safety seats. The NHTSA recommended newborn children to be secured in a rear facing child restrain system and placed in the back seat of the vehicle. NHTSA testing in 1991 showed that placing a rear facing safety seat in the front seat of the vehicle equipped with a passenger side airbag was very dangerous for the infant. Crash data verified this fact.

Adult occupants, almost all of them drivers, also suffered fatal and nonfatal injuries from airbag deployments. Fatalities resulted from the occupants being too close to the airbag when it deployed. However, airbags provide significant net benefit to adult occupants in preventing fatal and serious nonfatal injuries, and these benefits are greater than the risks of not having an airbag system at all.

In order to address this problem the NHTSA published in the Federal Registar, January 6, 1997, a Notice of Proposed Rulemaking on Airbag Deactivation (Docket 74-14, Notice 107). This notice proposed to allow dealers and repair businesses to deactivate the passenger side airbag, the driver side airbag or both, upon written authorization of the vehicle owner. The notice also stipulated dealers and repair businesses to let the owner of the vehicle know of the risks of turning the air bag off versus leaving it on in those circumstances [19]. This notice was much debated, for leaving the choice of the airbag activation to the vehicle owner could not have been the best solution. Thus, the NHTSA came with a final rule on occupant crash protection described in Docket NHTSA 00-7013. The rule amends occupant crash protection standard to require that future airbags be designed to create less risk of serious airbag induced injuries than current airbags, particularly for small woman and young children [26]. This rule specifically requires the airbags to be able to be partially deployed, or not deployed at all, if in doing so any risk of injury can be prevented to the occupant of the car seat. The rule also requires car manufacturers to have a system with the cited characteristics included in their fleet by 2007.

1.3 Organization of this thesis

The work developed in this thesis was structured on the patter recognition developing cycle described by Richard Duda [8]. According to this developing cycle data must first be collected, both to train and to test the system. Then, the main characteristics of this data must be extracted, as they affect both the choice of appropriate discriminating features and the choice of models for the different categories. The training process follows, using some or all of the data to determine the system parameters. At last, based on the evaluation of the results, a call for repetition of one or several steps in the process must be ordered to obtain satisfactory results. Figure 1.1. depicts this design cycle.

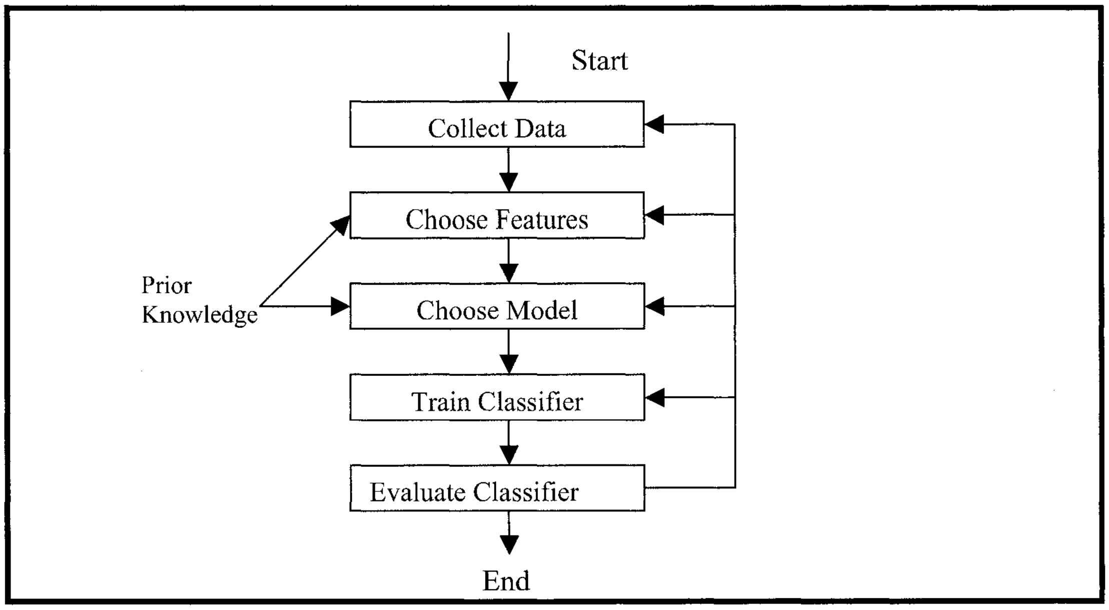


Figure 1.1. Pattern recognizer design cycle

The rest of this document is organized as follows:

Chapter 2 lists and explains the design constrains prior to the beginning of the development of the system. The main issues considered in this chapter are the partition of the observation space into different classes and the description of the transducers used to generate the feature vectors to create the observation space.

Chapter 3 describes the process of collecting the training feature vectors. A quantitative description of them will be included. This chapter also discusses the choice of the features used for the completion of the pattern recognition system.

Chapter 4 reviews the algorithms, methods and techniques initially considered appropriate for the successful development of the project. Preliminary results that led to the choice of a model used for the final development of the system are exposed as well.

Chapter 5 gives details on the final implementation of the pattern recognizer. Verification rates as well as robustness measurements of the system are presented as evaluation criteria to make sure the final pattern recognition system is not only accurate but reliable as well.

Finally, Chapter 6 presents general conclusions and a discussion on the current state of the project and possible directions on future research.

Chapter 2.

Design Considerations

Just as in any other engineering problem this project had some initial design considerations. They were related mainly to the *class system* and the nature of the input feature vectors to the pattern recognizer.

The *class system* refers to the way the occupants in the car seat should be classified according to the number of classes specified in that particular system. The FMVSS 208 specifies three different class systems:

1. A system capable of identifying the occupants of car seats to belong to one of three different classes

2. A system capable of identifying the occupants of car seats to belong to one of six different classes

3. A system capable identifying the occupants of car seats to belong to one of nine different classes

The nature of the input feature vectors of the pattern recognizer deals mainly with the transducers used to generate the feature vectors to create the observation space.

Explanation of these design considerations should help the reader have a clear understanding of the initial state of the project. The required classification groups are presented in the first part of the chapter, followed by an outline of the positions used for training purposes. Finally, the characterization and final layout of the transducers used to generate the input information to the pattern recognizer is presented.

Some technical issues regarding automotive industrial standards are not explained thoroughly here as they are not considered to be within the scope of this document.

2.1 Class system

The NHTSA introduced the FMVSS 208 changes regarding airbag regulation in early 2000 [33]. The main intention of the changes to the standard was to set clear rules for occupant crash protection, specifically in the case of airbag deployment. The first issue specified in this document was the scope of classes or groups in which the occupants of car seats were to be separated. Three different recognition systems were schemed, and their development was planned to be progressive, beginning with a 3-class system and having as final goal a 9-class system.

The 3-class system was an alternative to the on/off airbag switch system proposed in [19]. This system would basically decide weather the airbag should deploy or not in case of a car crash based on the weight of the occupant in the car seat. The 6 and 9 class systems were thought as information sources for a more complex airbag deployment system, which would not only activate or deactivate the airbag, but also, in case of a car crash and if necessary, would deploy the airbag with a safe pressure for the car seat occupant.

Table 2.1. gives detailed descriptions of each class system.

Class system	**Class #**	**Weight Thresholds**	**Description**
3	1	Empty Seat	
	2	Below 103 Lbs	Child or small occupant
	3	Above 103 lbs	Mid-size to large occupant
6	1	Empty Seat	
	2	Any Child Seat Facing rearward	With up to manufacturers weight rating
	3	Any Child Seat facing forward	With up to manufacturers weight rating
	4	12 to 59 lbs	Small to mid-size child
	5	60 to 103 lbs	Large child to small adult
	6	Above 103 lbs	Mid-size to large adult
9	1	Empty Seat	
	2	Any Child Seat Facing rearward	With up to manufacturers weight rating
	3	Any Child Seat facing forward	With up to manufacturers weight rating
	4	12 to 39 lbs	Small child
	5	40 to 59 lbs	Child
	6	60 to 99 lbs	Large Child
	7	100 to 129 lbs	Small adult
	8	130 to 179 lbs	Mid-size adult
	9	Above 180 lbs	Large adult

Table 2.1. Description of classes in the different recognition systems

This project has been focused on developing a pattern recognizer for the 3-class system listed above.

2.2 Testing Positions

The FMVSS 208 introduced not only the groups in which occupants should be classified according to their weight or shape, but it also specified a set of positions that would be used to test the accuracy and effectiveness of a CSOS in the rigid barrier frontal crash test. The sets of positions were specified for: rear facing infant seats, forward facing infant seats, convertible infant seats, car bed infant seats, 3 year old dummy, 6 year old dummy, 5th percentile female dummy (representing a light female occupant) and 50th percentile male dummy (representing a light male occupant). Additionally to this, a list of baby car seats suitable for the rigid barrier frontal crash test was also included [27]. According to the standard any CSOS designed should comply with test regulation on any of these baby seats as well as on the set of positions for dummies mentioned earlier.

The NHTSA also encouraged car manufacturers to have their own set of testing positions additionally to those specified by the FMVSS 208. This set of positions was referred to as the *due care positions provision,* and their intention was to make CSOS more robust. It was believed that if car manufacturers had made a good faith effort in designing their vehicles and had had adequate quality control measures, the vehicles would not be deemed to be in noncompliance with the FMVSS 208 standard.

For the design of the pattern recognizer of this CSOS testing positions from the FMVSS 208 and due care positions from two major car manufacturers were used as templates to generate the input feature vectors. Table 2.2. shows statistics on the number of positions listed for each type of occupant. A complete list of these sets of positions and their description is listed in Appendix A.

Measurement Standard	**3 year old Dummy**	**6 year old Dummy**	**5th Percentile Female Dummy**	**50th Percentile Male Dummy**	**Baby seats**
FMVSS 208	8	4	1	1	364
Car Manufacturer A	8	9	8	8	340
Car Manufacturer B	14	21	24	N/A	N/A

Table 2.2. Number of positions required to be recognized according to different measurement standards

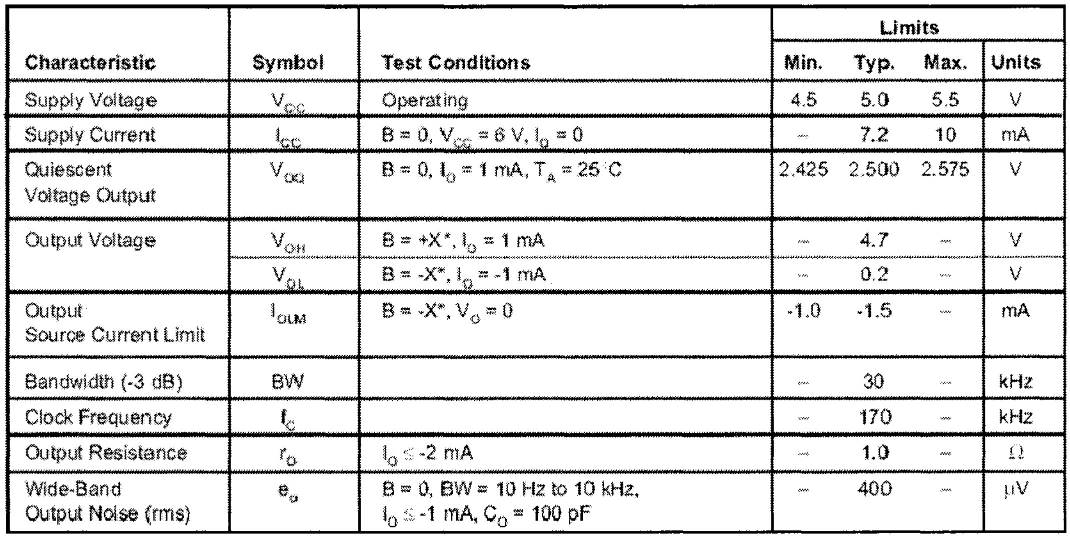
2.3 System Input

In order to design the pattern recognizer an understanding of the signals used as inputs is necessary. For this project a mat of linear Hall-effect sensors distributed along the bottom bun of a standard car seat was used. The choice of these sensors was based on automotive industry requirements, topic which is outside the scope of this document.

2.3.1 Description of Sensors

For this project hall-effect sensors were used. These are sensitive, temperature-stable linear Hall-effect sensors. They provide a voltage output that is proportional to the applied magnetic field. In this project 5V were used as supply voltage, and output values in the range 0.5 – 4.7 V were expected.

These magnetic sensors were considered ideal for use in linear and rotary position sensing systems in harsh environments, especially in automotive and industrial applications over extended temperatures to -40° C and +150° C. A brief description of the electrical characteristics of these sensors is presented in Table 2.3. For more information on the sensors refer to [1].



Characteristic	Symbol	Test Conditions	Limits Min.	Limits Typ.	Limits Max.	Units
Supply Voltage	V_{CC}	Operating	4.5	5.0	5.5	V
Supply Current	I_{CC}	$B = 0, V_{CC} = 6$ V, $I_O = 0$	–	7.2	10	mA
Quiescent Voltage Output	V_{OQ}	$B = 0, I_O = 1$ mA, $T_A = 25$ °C	2.425	2.500	2.575	V
Output Voltage	V_{OH}	$B = +X^*, I_O = 1$ mA	–	4.7	–	V
	V_{OL}	$B = -X^*, I_O = -1$ mA	–	0.2	–	V
Output Source Current Limit	I_{OLM}	$B = -X^*, V_O = 0$	-1.0	-1.5	–	mA
Bandwidth (-3 dB)	BW		–	30	–	kHz
Clock Frequency	f_C		–	170	–	kHz
Output Resistance	r_O	$I_O \leq -2$ mA	–	1.0	–	Ω
Wide-Band Output Noise (rms)	e_o	$B = 0$, BW = 10 Hz to 10 kHz, $I_O \leq -1$ mA, $C_O = 100$ pF	–	400	–	µV

Table 2.3. Electrical Characteristics of Hall-effect sensors

2.3.2 Layout and Distribution of Sensors

In order to gather relevant information from occupants of car seats, an array of the sensors described in 2.1 was placed in the bottom bun of the car seat.

The initial sensor layout was designed based on the principle that the minimal number of sensors to be placed in a structure depends on the number of parameters needed to retrieve its curvature [24]. Thus, simple flexion can be measured with one sensor, and composed flexion with two. In a two-dimensional surface a minimum of 4 sensors would be required. In theory, this would be the number of sensors necessary for this project. However, a 4-sensors arrangement (described as a 2x2 matrix) would not be able to capture all the positions described in section 2.2, as they involve spatial displacement (shift, turn) of the subject. An expansion of this arrangement of sensors in two dimensions in its rows and two dimensions in its columns was necessary. This expansion was based on the assumption that more distributed sensors improve the overall spatial resolution of the sensor assembly [7]. The resulting layout was a 4x4 array of sensors.

The first physical implementation of this sensor layout consisted of two surfaces, A and B. Surface B was fixed to the bottom of the seat bun, and contained the hall-effect sensors. Surface A was fixed to the top of the seat bun and contained activation magnets. This implementation proved to output correct voltage readings, but was too difficult to assemble inside the car seat. A second implementation was designed consisting of only one surface. This surface was fixed to the bottom of the seat bun and contained both the sensors and activation magnets inside a plastic hollow spring. This implementation worked properly and was easy to install in the car seat, reasons why it was chosen. Corresponding sensor layout is depicted on figure 2.1. Each sensor was given a label for easy reference.

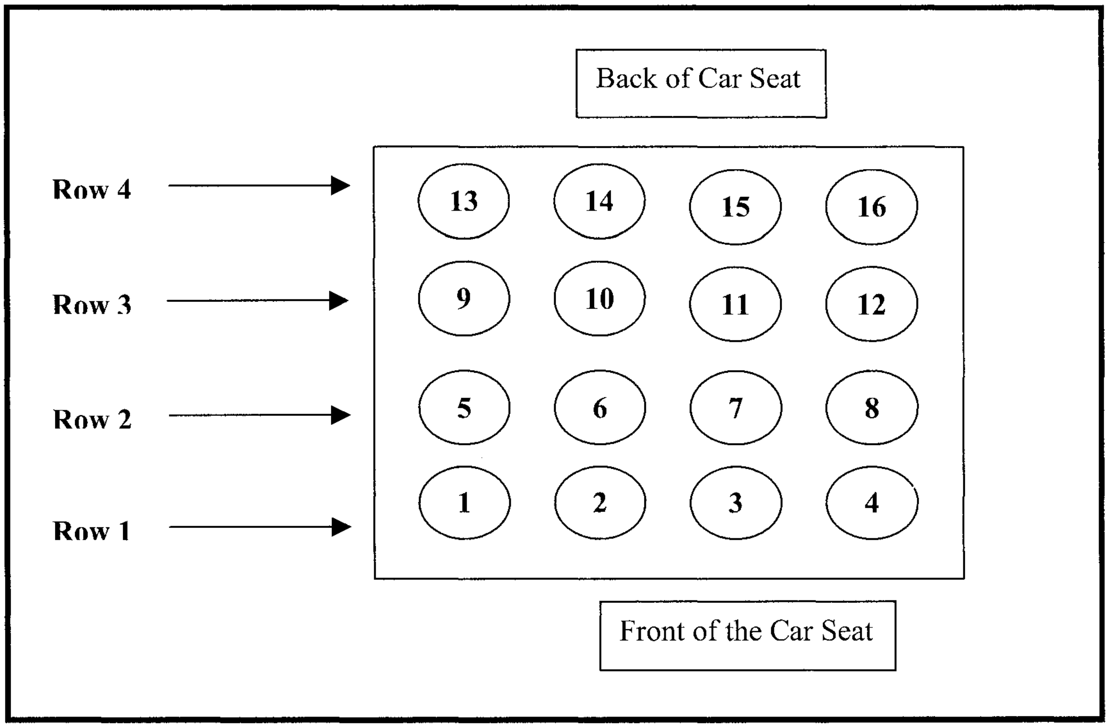


Figure 2.1. Layout of sensors in car seat cushion

2.3.3 Sensor Layout Optimization

The initial implementation of the sensor layout proved to work correctly in representing the patterns produced by human and dummy occupants in the car seat. However, the measurements on some baby seats were too vague as some of the sensor readings did not show any sign of activation at any test. This was due to the shape of the bun of the car seat. The middle of it was flat and the outer side had high bolsters to bring comfort to the car occupant. It allowed the passenger's bottom to slide into an oval shape position. This position exerted pressure to the center of the bun, providing excellent readings from the sensor layout. Unfortunately, the rigid structure of the bottom of the baby seats created a bridge-like effect between the two bolsters of the seats, avoiding sensors to be activated in the middle of the seat.

Industrial pressure measuring software confirmed this assumption. Figure 2.2 shows the pressure exerted on the car seat from a Cosco™ Touriva baby seat with a 3 year old dummy. As seen, the major amount of pressure is exerted at the back and to the sides of the seat, while the center remains untouched.

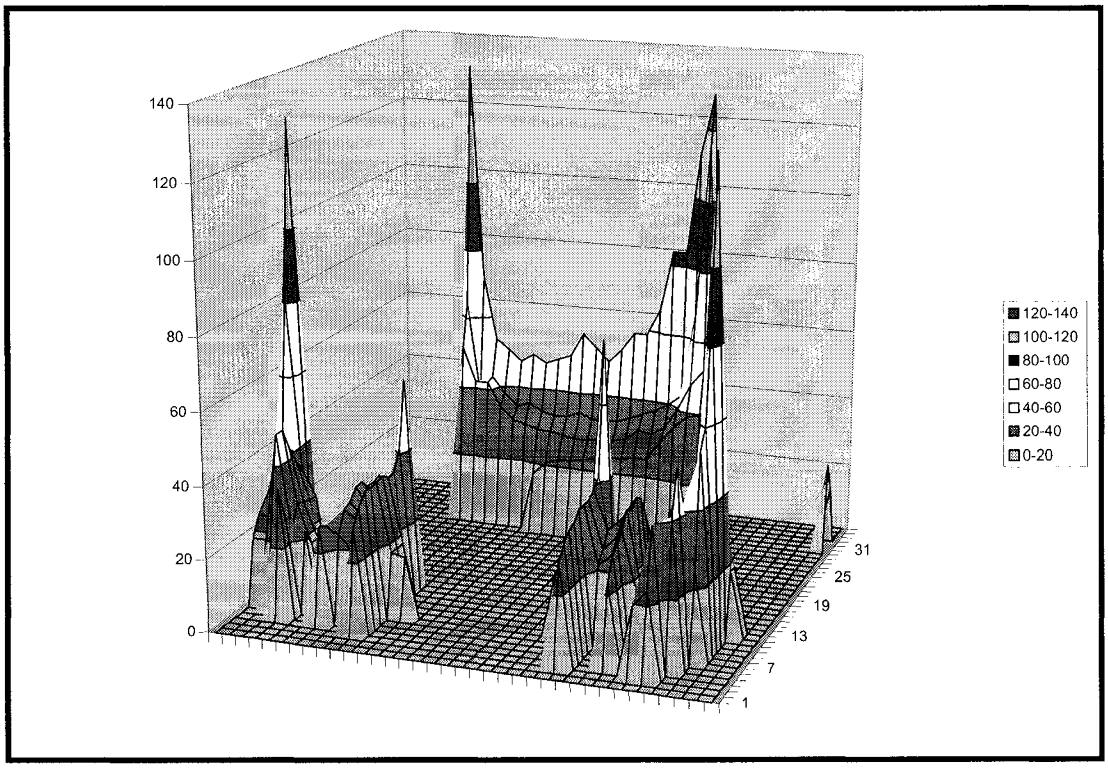


Figure 2.2. Pressure exerted on car seat from a baby seat

The shape of the sensor layout was redesigned to take the facts mentioned above into consideration. Sensors 13 and 16, placed in row 4, and sensors 6 and 7, placed in row 2 were relocated to the outer bolsters. The new sensor configuration was successful in providing more meaningful sensor readings when gathering feature vectors from baby seats. Therefore, this layout, shown in Figure 2.3., was used through the rest of this project.

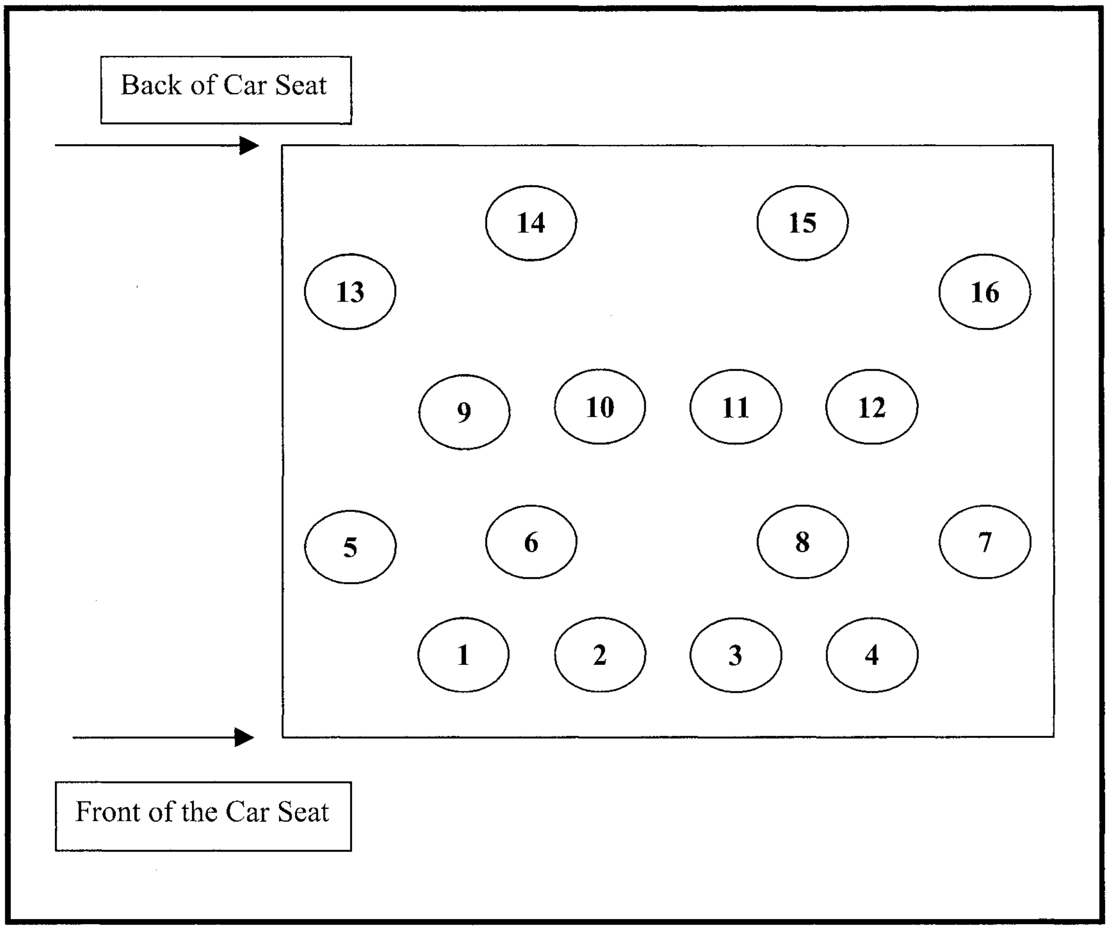


Figure 2.3. Final Sensor Layout

Tightening the seat belt over the baby seat created a small amount of pressure over a side of the sensor layout reflected on the activation of sensor 16.

Chapter 3.

Data Description

As depicted in figure 1.1., the first step involved in developing a pattern recognition system is the collection of data in order to create an observation space. Data collection accounts for a large part of the cost of developing a pattern recognition system [8]. In our case, 60% of the developing effort was spent on collecting and preprocessing data.

3.1 Data values and additional information to be gathered

The recognition of an individual object within a population is called *identification*. The process of grouping objects together into classes (subpopulations) according to their perceived likeness or similarities is called *classification* [16]. The idea of having a classification process without having a group of objects to distinguish from is simply unrealistic. The area of pattern recognition includes both classification and recognition. Thus, a population of objects representing the landscape of the observations space must be created first in order to design a pattern recognizer. Acquisition software was developed for this purpose. A brief description of it is presented in Appendix B. This section focuses on describing the data values and additional information gathered for each object of the observation space.

The objects in the observations are 16-dimensional vectors:

$$S = (S_1, \ldots, S_{16}), \qquad -0.1 \leq S_i \leq 2.5$$

Each dimension of these objects represents the output voltages from a sensor in the layout described in section 2.3.3. The objects represent the pressure landscape generated in the seat by an occupant in a certain position. These positions were described in section 2.2 and are listed in Appendix A.

Because the values of the 16 dimensions of the objects in the observation space are raw data it is necessary to collect additional information for the purposes of design of the recognition system. Thus, besides the 16 sensor values, each object in the observation space has labels describing the type of occupant that generated that pattern, its weight, height and torso ratio, a description of the position itself and the date the information was gathered.

These labels determine the class target of each object in the observation space for different recognition systems when using supervised training algorithms. They also provide a way to trace objects during training, testing and debugging processes and allow easy modifications to the boundaries of classes in the observation space if needed.

3.2 Number of Input Feature Vectors

The first question that arises when gathering data to create an observation space is how many objects are necessary to clearly discriminate among the classes? In our case the question could be refrained to, how many examples of each position are necessary to complete the database? Unfortunately, there is neither concrete nor correct answer to these questions. However, there are good estimates on how to deal with this problem.

3.2.1 Weight Distribution

A consideration when creating the training database for a pattern recognition system would be to have examples of all the different objects included in the observation space. In this case, it would be required to have feature vectors of the different positions of the all car seat occupants.

The FMVSS 208 regulation is targeted towards protecting most vulnerable people when air bags are deployed. Consequently, most of the testing positions are specified for subjects that lie within the range 0 to 100 Kg (0 to 220 lbs). This range can be divided to obtain representative subsets of objects with similar weights, as people with comparable weight have comparable physical characteristics, and to make the data gathering process faster. A 5 Kg interval division of this weight range has been selected to represent the observation space by 20 distinct weight categories.

Several feature vectors of each of the testing position described in Section 2.3 and listed in Appendix A should be gathered in order to assure its representation in the observation space. Three repetitions of each position should ensure enough information for the recognition process.

In conclusion, the necessary number of feature vectors to complete the database would be:

Examples in Database = # of Intervals in weight range * # of Positions * # of Repetitions of each position

$$= 20 * 378 * 3 = 22.680 \qquad \textbf{(1)}$$

3.2.2 Computational Intelligence Methods

A second approximation to the number of feature vectors necessary to complete the recognition system is to assess what amount of information the computational intelligence techniques need in order to succeed. These techniques can be classified into supervised and unsupervised learning techniques. In supervised learning techniques, a category label is provided for each pattern in a training set, and the learning seeks to reduce the sum of the costs of these patterns [7]. In unsupervised learning or clustering there are no explicit labels, and the system forms clusters or "natural grouping" of the input patterns. "Natural" is always defined explicitly or implicitly in the clustering system itself [8]. Thus, there is no minimum number of objects in the observation space for unsupervised learning techniques.

The only supervised learning technique used in this project was the Neural Networks, so the proper choice of the number of examples required in the observation space was bounded to them.

The Vapkin-Chervonenkis Dimension (VCDim)

The VCDim gives a good estimation of the number of input feature vectors needed to train a neural network and allows consideration of the items listed above.

For any assignment of values to its internal parameters θ (weights, thresholds, etc.) a neural network N with binary outputs computes a function $x \rightarrow N(\theta, x)$ from D into {0;

1}, where *D* is the domain of the network inputs *x*. The VCDim of *N* is a number which may be viewed as a measure of the richness (or diversity) of the collection of all functions $x \rightarrow N(\theta, x)$ that can be computed by *N* for different values of its internal parameters θ. That is why the VCDim of a neural network is related to the number of training examples that are needed in order to train *N* to compute (or approximate) a specific target function h: D → {0, 1} [5].

Typically the VCDim grows polynomially (in many cases, between linearly and quadratically) with the number of adjustable parameters of the neural network. In particular, if the number of training examples is large compared to the VCDim, the network's performance on training data is a reliable indication of its future performance on subsequent data. The notion of the VCDim, which was introduced in [22], is not specific to neural networks. It applies to any parameterized class *F* of functions $x \rightarrow f(\theta, \mathrm{x})$ from some domain *D* into {0, 1}, where θ ranges over some given parameter space [5].

In the general case one defines the VCDim of *F* (VCdim(*F*)) as the size of the largest subset *D'* of its domain *D* so that *D'* is shattered by *F* (or more precisely: by the restrictions of the function $x \rightarrow f(\theta, \mathrm{x})$ in *F* to inputs x Є D'). In other words: the VCDim of *F* is the size of the largest subset *D'* of its domain *D* for which every dichotomy *h* over *D*' can be computed by some function in *F* [5].

For neural networks, the VCDim lies in the following range:

$$2 * \left[\frac{N_1}{2} \right] * n \leq VCDim \leq 2 * N_w * \log(e^{N_N}) \qquad \textbf{(2)}$$

Where N_1 is the number of hidden nodes, n is the dimension of the input, N_w is the total number of weights in the network, N_N is the total number of nodes and e is the base of natural logarithm.

The biggest neural network estimated for this project had a16 inputs, 30 hidden neurons, 3 outputs architecture, resulting in a VCDim of:

$$VCDim = 2 * 570 * \log(e^{49}) = 24259$$

In conclusion, according to the VCDim the observation space should contain about 24300 input feature vectors.

3.3 General Statistics

The database was gathered after taking into consideration the estimations described in section 3.2. It ensured every position of each car seat occupant to have at least one input feature vector, and each class to include approximately the same number of input feature vectors. It contained 23625 feature vectors. Figure 3.1. shows its distribution.

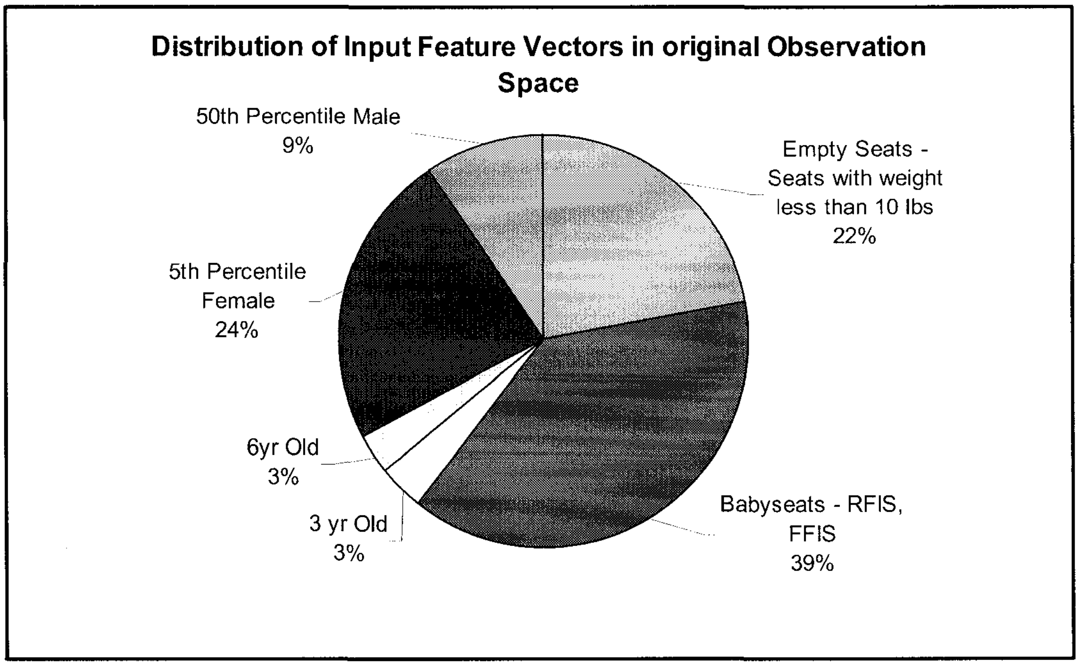


Figure 3.1. Distribution of original Observation space

Unfortunately, not all the feature vectors were valid. Technical and human errors lead to the collection of some input feature vectors that are mislabelled or biased. The main reasons for the collection of erroneous data are:

1. Electromechanical problems in sensory system, e.g. the wiring between the sensor inside the seat bun and the data acquisition hardware. This type of problem is typically demonstrated by voltage readings below the voltage range threshold.
2. Feature vectors unlabelled or mislabelled due to human error. Such input feature vectors could not be used for training purposes as there was no way to identify to which type of occupant they belong to.

These types of problems are only relevant for the data acquisition part of the project. This data will be used to train and evaluate the model that performs the pattern recognition. The final parameters of this model will be included in the production model. Therefore, correcting this errors at an early stage will lead to the proper development of the pattern recognizer, but will not play a roll when producing the final product.

Input feature vectors with these types of errors accounted for 6.23% of the initial observation space. After the deleting those erroneous feature vectors, 22153 remained. Their distribution is showed in figure 3.2.

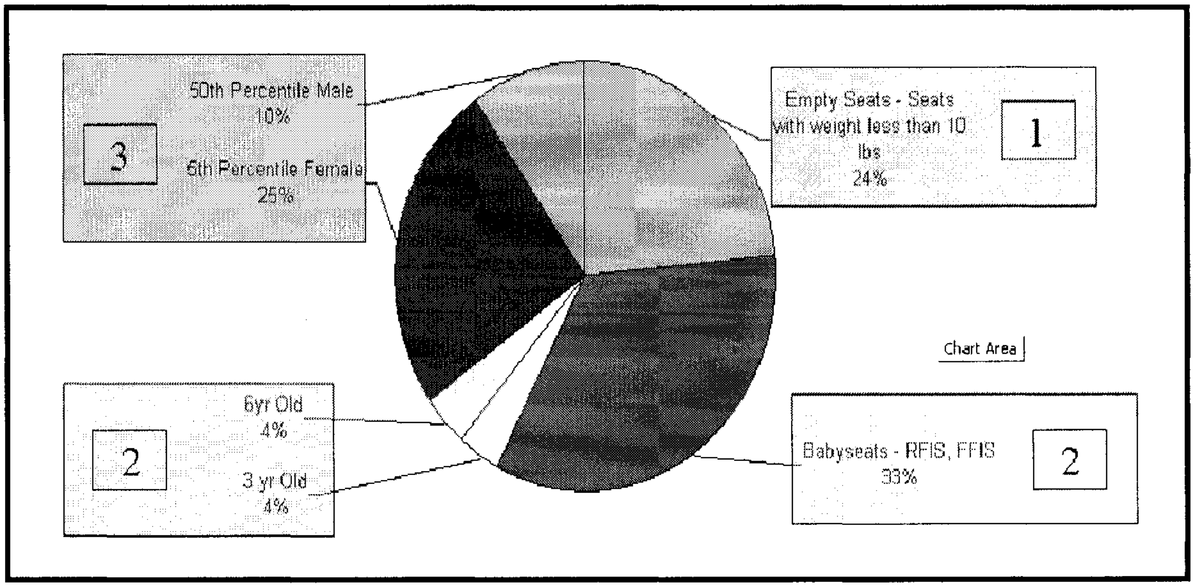


Figure 3.2. Distribution of final Observation Space

3.4 Feature Selection

The next step in the design cycle of a pattern recognition system, after the collection of the data, is to choose the features that best represent the observation space.

Features are attributes that objects in a population possess and that can be sensed or measured to distinguish between one or more different subpopulations [16]. The main goals for mapping the observation space into a reduced feature space are to:

1. Retain as much of the original information as possible

2. Remove as much as possible of the redundant and irrelevant information that could cause extraneous noise and degrade the classification performance

3. Render the measurement data to variables that are more suitable for decision making.

Although useful, feature selection is not always necessary. An ideal feature selector would yield a representation that makes the job of the classifier trivial; however, an omnipotent classifier would not need the help of a sophisticated feature selector [8]. Besides, the feature selection process should start with large sets of attributes and should count with some human experience on the perception of the object [16]. For this project, the number of input sensors was relatively low and no experience on this type of systems was at hand, at least from direct sources.

In this pattern recognition system the sensors output voltages were the only measurable attributes in the pattern vectors. The number of sensors was optimized for the size and shape of the car seat (Chapter 2), and a reduction in their number could lower the accuracy in the measurement of the distribution of the pressure generated by the occupants across the car seat.

Therefore, no optimization was performed on the features of the input vectors in the observation space at this phase of the project. A relevance analysis of these features was performed, though, after the pattern recognition system was finalized. Results of this analysis are presented in section 5.4.2.

Chapter 4.

Data Analysis

The next step in the design of a pattern recognition system, once the data for the creation of the observation space has been collected, is the choice of a proper model or method for the recognition itself. Two main questions arise when choosing a model to do pattern recognition:

1. How fit is a model for the specified purpose?
2. How does the hypothesized model differ from the true nature of the underlying patterns?

The first of these questions is answered throughout this chapter. The second question is answered in chapter 5.

For this project a systematic approach to model selection was applied. Different supervised and unsupervised algorithms were tested to compare their effectiveness in creating the class systems mentioned in Chapter 2. The choice of these models was based on their effectiveness according to the literature, previous experience in their development and use, and real time performance.

The first type of models used is the cluster-based unsupervised learning algorithms. The main goal of a clustering algorithm is to find groups or structures with similar elements in a data set; the clustering algorithm receives input data elements and returns clusters or classes of data according to their similarity. The clustering algorithm partitions the space according to certain prototypes and their distances to the rest of the elements in the data set. They are useful when the distribution of the observation space is unknown and an initial description is required. In our case, the desired division of the observation space is known. However, the clustering methods can be used for verifying whether these required classes correspond to the natural clusters, if any, present in the data.

The second type of models used is the supervised learning algorithms. In supervised learning, a category label is provided for each pattern in a training set, and the learning seeks to reduce the sum of the cost for these patterns. These methods are more robust to noise, take less training time and, in general, perform better than the unsupervised learning algorithms. However, they behave like black boxes, and do not allow the user to have a visualization of the distribution of the observation space.

4.1 Fuzzy Clustering

The fist clustering method used to study the nature of the data and its possible grouping into classes is fuzzy *c*-means. This method is based on the minimization of a performance index defined as the sum of all vectors in a cluster domain to the cluster center [6].

In simple terms, the fuzzy *c*-means algorithm calculates the membership value of a given example with respect to the defined number of clusters. Thus, a particular example can have a relative high belongingness to more than one cluster. This property allows creating an initial representation on the consistency and overlapping of the training data set used in any project, it also provides for an easy visualization on the distribution of the observation space.

4.1.1 Algorithm

The general fuzzy *c*-means algorithm is compound of the following steps [6]:

Given a training data set containing M examples and *c* classes, user defined number of intervals n_{Fi} for feature F_i

1. For class c_j do, j = 1, 2, …, *c*

2. Choose K = n_{Fi} as the initial number of cluster centers.

3. Distribute the values of the feature among the K cluster centers "prototypes", based on the minimal distance criterion. As the result, feature values will cluster around the updated K cluster centers.

4. Compute K new cluster centers such that for each cluster the sum of the squared distance from all points in the same cluster to the new cluster center is minimized.

5. Check if the updated K cluster centers are the same as the previous ones, if yes go to step 1, otherwise go to step 3.

As a result, the final boundaries for the feature will be the minimum value the feature takes on, mid points between any two nearby cluster prototypes found for all clusters, and the maximum value the feature takes on.

It is necessary to know the membership of each element to each of the cluster in order to know to which group the element belongs. The partition matrix *u* specifies these values for each element in the data set. In this particular case, it is assumed that the sum of the membership of a specific feature is equal to 1, therefore

$$u = \{u_{ik} \in [0,1] | \sum_{i=1}^{c} u_{ik} = 1\} \quad \textbf{(3)}$$

Where *c* denotes the clusters to find in the data set, and *k* is the *k*th element in the data set.

The partition matrix is obtained using the following equation,

$$u_{ik} = \frac{1}{\sum_{i=1}^{c} \left(\frac{\|x_k - v_i\|}{\|x_k - v_j\|} \right)^2}; \ \forall k \in [1,2,\cdots,n], \ \ \forall i \in [1,2,\cdots,c], \quad \textbf{(4)}$$

Where $\|.\|$ denotes distance, v_i is the *i*th cluster center, x_k is the *k*th element in the data set, *n* is the number of elements in the data set, and *c* is the number of clusters to be found.

The function to optimize the prototypes is,

$$v_i = \frac{\sum_{k=1}^{n} u_{ik} x_k}{\sum_{k=1}^{n} u_{ik}} \quad \textbf{(5)}$$

The fuzzy clustering aims to achieve a classification that is closer to the real world, because the object itself is usually of ambiguous or fuzzy nature [25]. Under this framework the following tests were carried out.

4.1.2 Experimentation and Results

Three different experiments were conducted to measure the effectiveness of fuzzy clustering. This method was also used to assess the consistency in the observation space. Figures 4.1., 4.2. and 4.3. show the results to these experiments.

In each figure the vertical axis demarks the cluster to which each particular input feature vector was assigned. The horizontal axis demarks the position of each input feature vector in the observation space. The boxes in the grid represent the type of occupant to which that part of the observation space belongs to. The numbers in the boxes represent the following type of occupant:

1. Empty seat, Seats with weight of less than 10 lbs, empty baby seats
2. Rear facing and forward facing baby seats with all type of occupants
3. 3 year old dummy
4. 6 year old dummy
5. 5th female dummy
6. 50th male dummy

In an ideal case all the feature vectors belonging to the same type of occupant should be grouped in the same cluster. Inconsistencies or overlaps in the observation space can be found if feature vectors generated by the same type of occupant are grouped into different clusters.

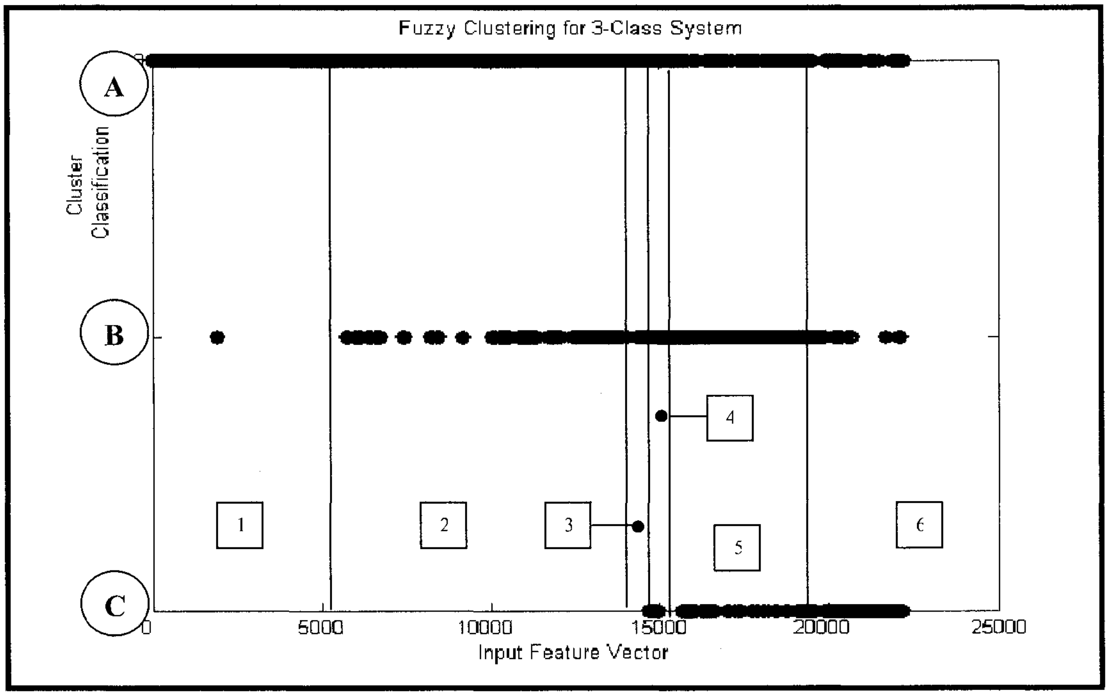


Figure 4.1. Clusters generated by the Fuzzy c-means algorithm for the 3-class system

Figure 4.1. shows the results for the 3-class system clustering. For this class system the empty seat feature vectors should belong to one cluster; all baby seats, 3 year old dummy and 6 year old dummy feature vectors should belong to another cluster and the rest of the observation space should belong to a third cluster. The results show how such partitioning of the observation space is not really achieved.

Almost the entire empty seat feature vectors are grouped into cluster A. This shows that the data collected for this type of occupant is very consistent. However, feature vectors from other types of occupants are included in this cluster as well. This shows that feature vectors in the observation space are similar, regardless of the type of occupant that generates them.

Cluster B grouped most of the 5th percentile female dummy feature vectors, as well as some baby seats and some 50th percentile male dummy features vectors. This is not a good sign. Feature vectors belonging to baby seats should be the major component of this cluster, as they actually are the most frequent feature vectors in this group.

Cluster C grouped most of the 50th percentile male dummies feature vectors, as well as some 5th percentile dummies and some baby seats features vectors. This cluster shows how occupants exerting high pressure on the car seat tend to form a class of their own. Their distance to feature vectors from dummy occupants with lower weight is relatively large.

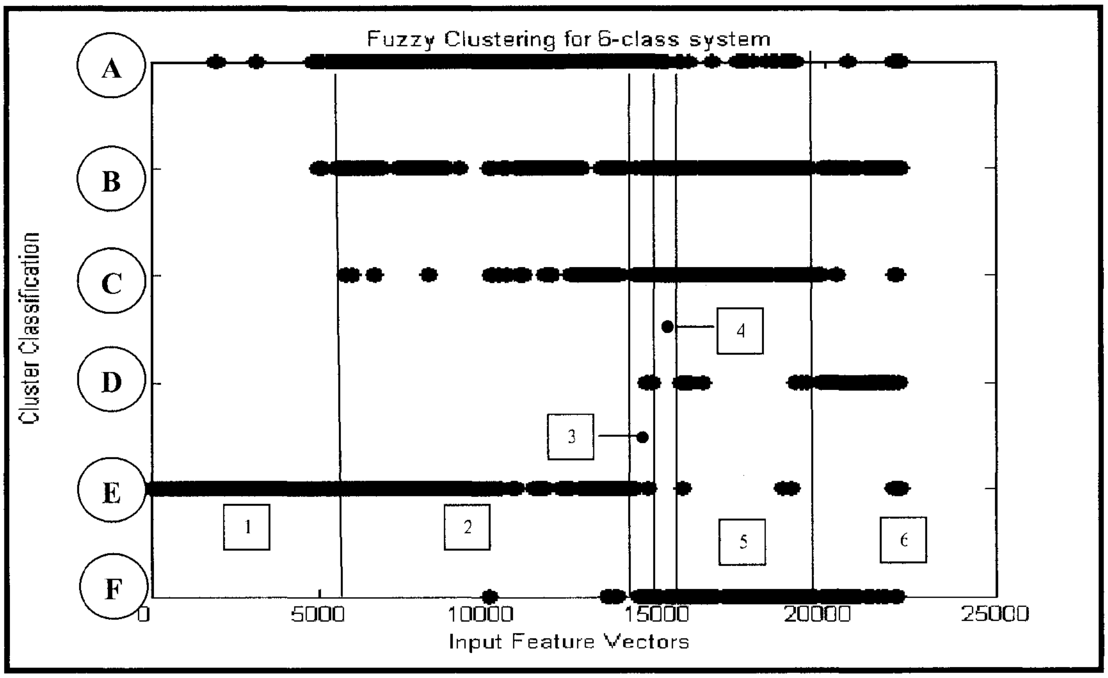


Figure 4.2. Clusters generated by the Fuzzy c-means algorithm for the 6-class system

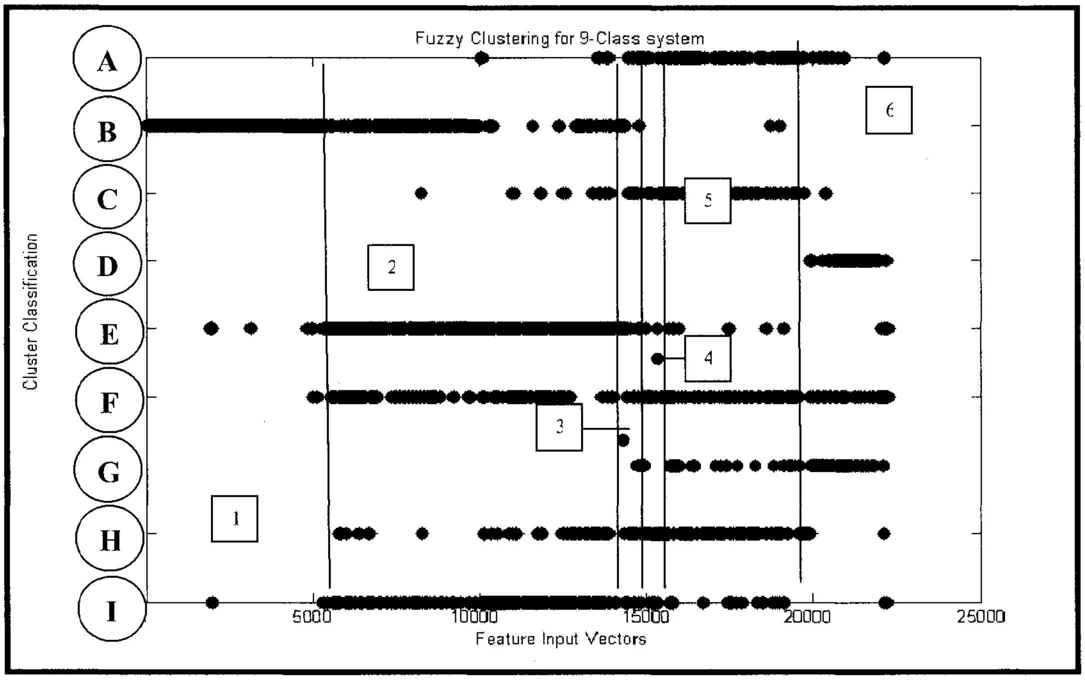


Figure 4.3. Clusters generated by the Fuzzy c-means algorithm for the 9-class system

Figures 4.2. and 4.3. show the results for the 6-class and 9-class systems clustering respectively. These experiments did not lead to any new discoveries from the ones listed for the 3-class system clustering. They just reinforced the idea of high similarities in the observation space among feature vectors of all kind of occupants.

4.1.3 Conclusions

The fuzzy c-means clustering experiments demonstrated that there is high inconsistency in the pressure landscape generated by positions of the same occupant in the car seat.

The only occupant features vectors that proved to be easily classifiable and that showed high consistency are the empty seat feature vectors. It probably happens because the feature vectors represent no pressure state in the car seat. Thus, to recognize such a simple state (a completely flat pressure landscape in our case) is an easy task for any classifier.

Feature vectors from heavy occupants tend to group together better than feature vectors from other type of occupants. The cause can be related to the conclusion above. Heavy occupants generate a highly activated landscape, so they would be exactly at the opposite of the observation space from the empty seat feature vectors. They could be considered as the inverse of the empty seat occupants, leading to an easier detection of them from the pattern recognizer.

4.2 Self-Organizing Maps

Although the fuzzy c-means provided useful information about the consistency and similarities of the input feature vectors, a second clustering algorithm was used in order to have a clearer visualization on the feature vectors that were more likely to be grouped together. The Kohonen Networks, or Self Organizing Maps (SOM), were chosen for this task as they provide representation of the feature space while preferring spatial relations.

The SOM are one of the most common Neural Networks used in the scientific community. They can be visualized as a sheet-like neural-network array, Figure 4.4., where the cells (or nodes) of which become specifically tuned to various input signal patterns or classes of patterns in an orderly fashion [14]. The learning process is competitive and unsupervised, just like in the fuzzy *c*-means algorithm. In the basic version, only one map node (winner) at a time is activated corresponding to each input.

The locations of the responses in the array tend to become ordered in the learning process as if some meaningful nonlinear coordinate system for the different input features were being created over the network.

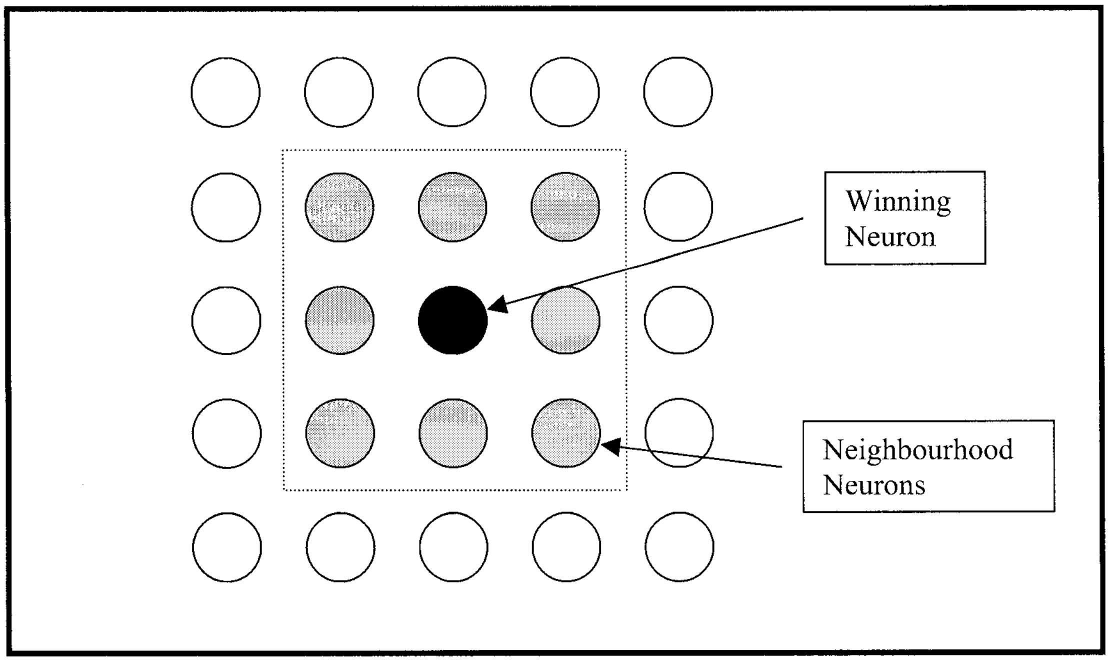


Figure 4.4. Basic Architecture of SOM

Although the SOM can be viewed as model of specific aspects of biological neural nets, it can also be used as a powerful tool for statistical analysis and visualization.

The SOM is both a projection method which maps high-dimensional data space into low-dimensional space, and a clustering method so that similar data samples tend to be mapped to nearby neurons. From the methodological and computational point of view the mathematical and statistical properties of the algorithm can be considered (for instance, the time and space complexity, the convergence properties), as well as the nature of the input data (signals, continuous statistical indicators, symbol strings) and their preprocessing [11]. There exist a number of variants of the SOM in which various distance measures can be used, and the structure of the map interconnections is variable [14].

4.2.1 Algorithm

The SOM contain M neurons in the neural layer, each with a parametric weight vector $\mathbf{v}^{(m)}$ of dimension N, which is the same as the dimension of the input feature vector

$\mathbf{x}^{(m)} = (\mathrm{x}_1^{(q)}, \ldots, \mathrm{x}_N^{(q)})$, where q =1, …, Q (Maximum number of feature vectors).

The basic SOM algorithm is compound to the following steps [16]:

1. Randomize the order of $\{ \mathbf{x}^{(q)} \}$; Initialize $q \leftarrow 1$;

   Initialize parametric weight vectors $\mathbf{v}^{(m)}$ with values between 0 and 1

2. Draw exemplar $\mathbf{x}^{(q)}$ from the exemplar set.

3. Compute distance of vector exemplar $\mathbf{x}^{(q)}$ to all the parametric weight vectors $\mathbf{v}^{(m)}$ via:

$$D_{qm} = \sum_{n=1}^{N} \left| x_n^{(q)} - v_n^{(q)} \right| \qquad \textbf{(6)}$$

   find $\mathbf{v}^{(m^*)}$ with minimum distance D_{qm^*}

4. Update neuron m* via:

$$\mathbf{v}^{(m^*)} = \mathbf{v}^{(m^*)} + \eta 1\ (\mathbf{x}^{(q)} - \mathbf{v}^{(m^*)}) \qquad \textbf{(7)}$$

and update all neurons m ≠ m* via

$$\mathbf{v}^{(m)} = \mathbf{v}^{(m)} + \eta 2\ (\mathbf{x}^{(q)} - \mathbf{v}^{(m^*)}) \qquad \textbf{(8)}$$

5. If stop criterion satisfied then stop. Else $q \leftarrow q + 1$; If $q > Q$ then $q = q - Q$; Go to step 2.

4.2.2 SOM results interpretation

There are different ways to visualize the results of the SOM. They can be divided into three categories based on the goal of the visualization [23]:

1. The first is the task of getting an idea of the overall shape and possible structure of the data

2. The second task comprises the wide field of analyzing the weight vectors $\mathbf{v}^{(m)}$, for example to characterize the clusters or the correlation between weight vector components.

3. The third task is the examination of new data samples for classification and novelty detection purposes.

The creation of *weight maps* is necessary to realize the first task mentioned above. Because the weight vectors $\mathbf{v}^{(m)}$ connect all the neurons of the SOM together and have the same dimensions as the feature input vectors, it can be thought of them as a cubic weight matrix. Thus, the weight maps can be visualized by displaying the weight vectors one layer at the time. Figure 4.6. shows an example for a 3x3 SOM with 4 inputs. The weight matrices have 4 layers corresponding to the 4 dimensions of the input feature vectors. There are 9 weight vectors in the weight matrix, and only 2 connection weight vectors are shown in the figure. The most obvious information these weight maps provide is the correlation among the input features. If two weight maps are similar then the two input features they represent are highly correlated. Otherwise, the two input features are not correlated. Correlation of part of the observation space can be known as well. For example, two weight maps can be very similar in the upper-right corner but are very different in other areas. This means that only the data to the upper-right corner is highly correlated.

A *clustering map* can be created in order to analyze the similarity of the weight vectors $\mathbf{v}^{(m)}$. This map represents the distance between neighboring weight vectors. Any 3D measure can be used to do that. In this experiment the Euclidean distance of two vectors was used. For two N dimensional vectors $\mathbf{x} = (x_1, \ldots, x_N)$, and $\mathbf{y} = (y_1, \ldots, y_N)$ the Euclidean distance D is defined as:

$$D = \sqrt{\sum_{n=1}^{N} (x_n - y_n^n)^2} \qquad \textbf{(9)}$$

In this project similar weight vectors (the ones with small distance values) are represented with bright colors in the clustering map and vice versa.

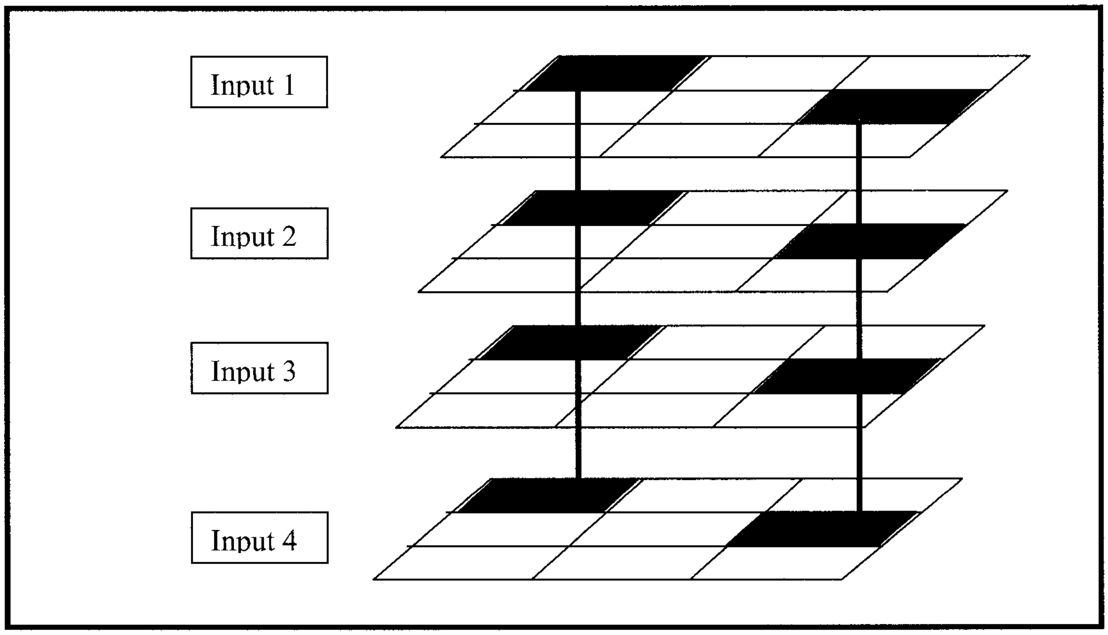


Figure 4.5. A 3x3 SOM weight vector map

A *data distribution map* can be created to achieve the last task mentioned above. After the SOM is trained, the input feature vectors can be mapped back to it. A record of how many input feature vectors were assigned to each neuron can be achieved.

4.2.3 Experimentation and Results

Several SOM were trained using the SOMtool package [2]. The two with the most valuable information are presented for discussion in this document. For all the experiments the following parameters were used:

Number of Iterations:	200
Step gain $\eta 1$:	0.2
Step decline $\eta 2$:	0.1

Figure 4.6. shows the results for a 6x6 SOM. Figure 4.6a. shows the weight maps for this SOM. These maps have no similarity at all for the different input features. Actually, there are almost no regions among the different weight maps that show similarities. According to the theory, the input features of this experiment are not correlated. This assumption could not be true. The pressure distribution across the seat should be similar among close sensors.

Figure 4.6b shows the clustering map for the 6x6 SOM. This map represents how similar the weight vectors are among the different weight maps. In an ideal case, all or most of the neurons should end up with a bright color. This would mean that the observation space is very consistent and the input feature vectors of similar classes are grouped together. Unfortunately that is not the case in this project. Dark colors in the clustering map means inconsistency in the observation space. This result agrees with the conclusions drew from the Fuzzy c-means clustering experiment.

Figure 4.6c shows the distribution map for the 6x6 SOM. Brighter color for each neuron illustrates low densities of examples assigned to that particular node. Dark colors illustrate the contrary. As seen in the map almost 40% of the input feature vectors were assigned to one single neuron. Furthermore, only 8 out of 36 neurons were assigned any feature vectors at all. This trend just confirms the closeness of the input feature vectors in the observation space and the high probability of overlapping among some positions generated by different type of occupants.

Figure 4.7. shows the results for a 9x9 SOM. The results of the data distribution map and the clustering map concur with the results obtained in the 6x6 SOM. However, the results of the weight maps are different. They show certain correlation among different input sensors. Sensors 9, 10, 11 and 12 have very similar weight maps, as well as sensors 16 and 7 and sensors 13 and 6. This means they are contributing with similar information to the SOM. This result follows common sense. Close sensors should contribute with similar sensor readings. This correlation is not necessarily bad. The sensor layout was optimized to measure the places in the car seat with higher pressure. Thus, placing the sensors elsewhere could reduce the correlation of the input features, but could end up not providing useful information for classification purposes because most of the sensor readings could end up being zero. Besides, having sensors nearby helps improve the reliability of the system in case of a sensor failure. Readings from the places of the car seat exposed to relatively high pressure would still be measured.

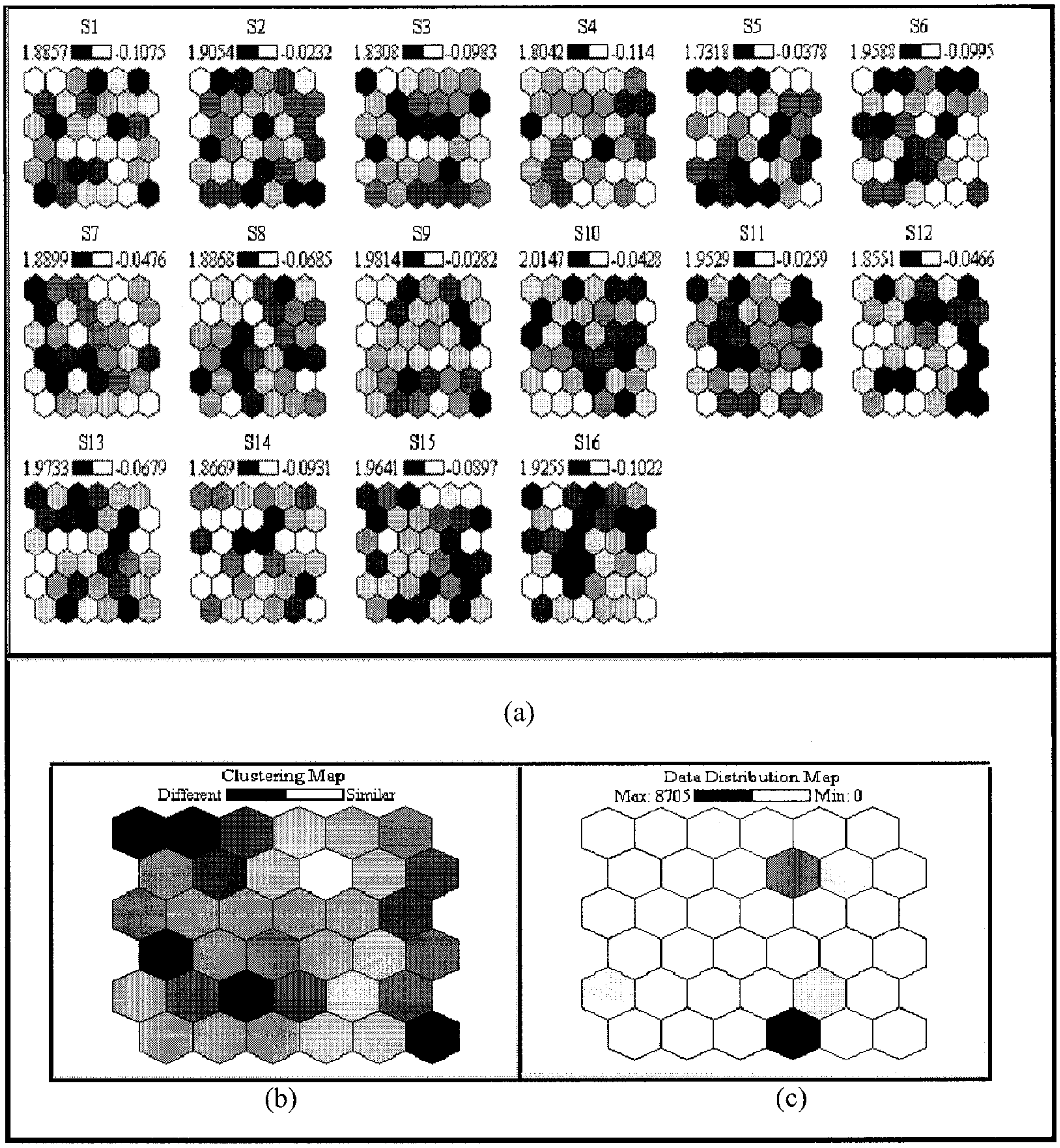


Figure 4.6. 6x6 SOM. (a) Weight Maps. (b) Clustering Map. (c) Distribution Map

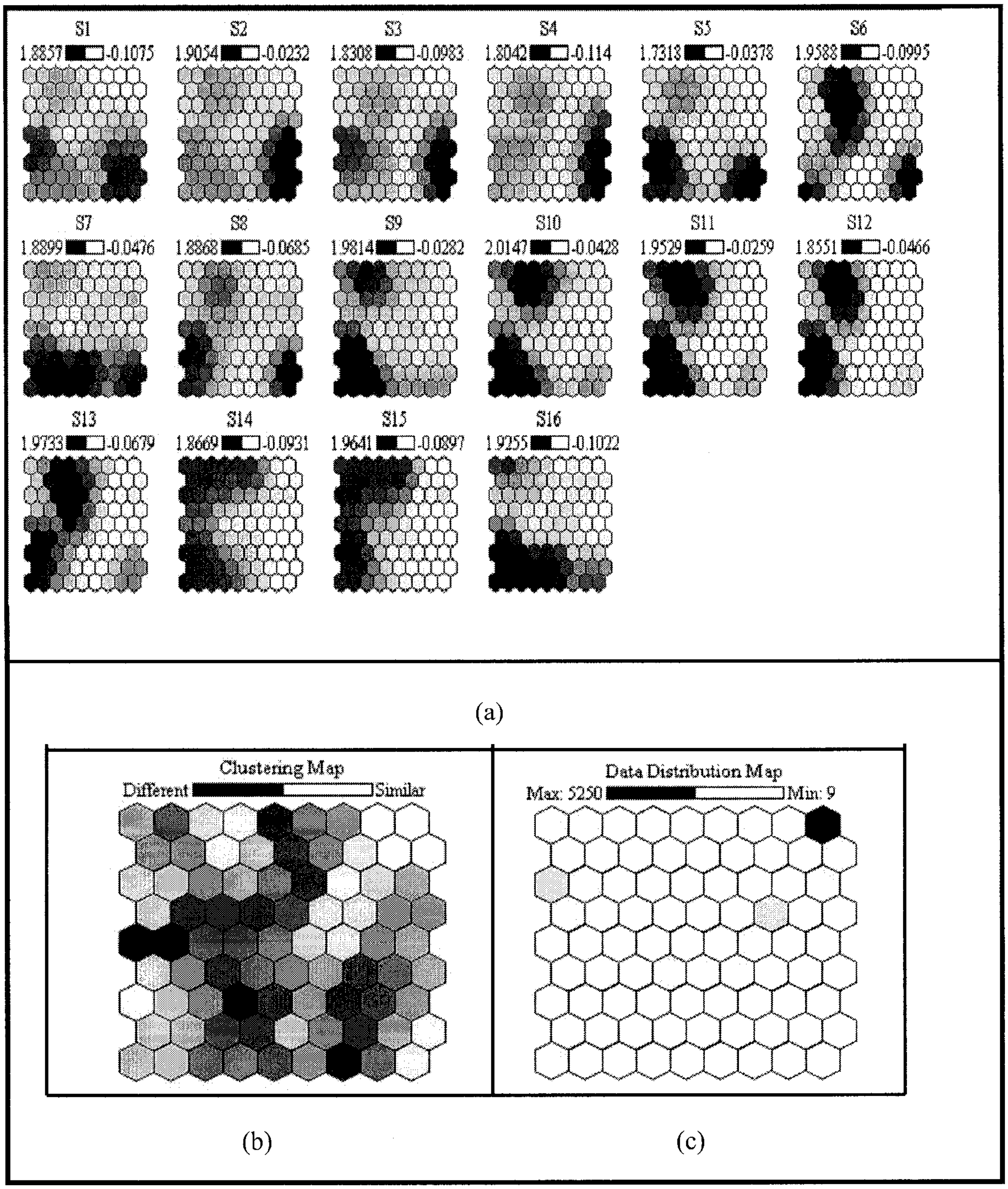


Figure 4.7. 9x9 SOM. (a) Weight Maps. (b) Clustering Map. (c) Distribution Map

This result also proved that the SOM can be successfully used for visualization of high dimensional observation spaces, as long as the structure of the SOM is sufficient (contains enough neurons) to cover the observation space.

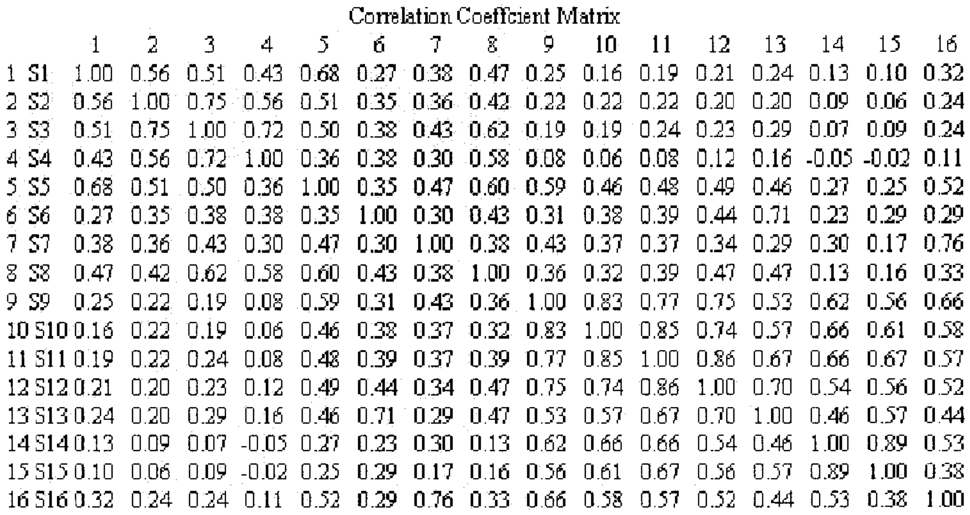
The independence in the information provided by the sensors was also confirmed by calculating the static correlation of the input features over all the observation space:

$$Correlation = \frac{Cov(X, y)}{\sigma X \cdot \sigma y}, \qquad \textbf{(10)}$$

where $Cov(X, y)$ is the covariance obtained using

$$Cov(X, y) = \frac{1}{n}\sum_{i=1}^{n}(X_i - \mu_X)\cdot(y_i - \mu_y), \qquad \textbf{(11)}$$

n is the number of observations in the data set, i is the ith element of the data set, X denotes the input data set, and y is the output data set.



Correlation Coeffcient Matrix

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	S1	1.00	0.56	0.51	0.43	0.68	0.27	0.38	0.47	0.25	0.16	0.19	0.21	0.24	0.13	0.10	0.32
2	S2	0.56	1.00	0.75	0.56	0.51	0.35	0.36	0.42	0.22	0.22	0.22	0.20	0.20	0.09	0.06	0.24
3	S3	0.51	0.75	1.00	0.72	0.50	0.38	0.43	0.62	0.19	0.19	0.24	0.23	0.29	0.07	0.09	0.24
4	S4	0.43	0.56	0.72	1.00	0.36	0.38	0.30	0.58	0.08	0.06	0.08	0.12	0.16	-0.05	-0.02	0.11
5	S5	0.68	0.51	0.50	0.36	1.00	0.35	0.47	0.60	0.59	0.46	0.48	0.49	0.46	0.27	0.25	0.52
6	S6	0.27	0.35	0.38	0.38	0.35	1.00	0.30	0.43	0.31	0.38	0.39	0.44	0.71	0.23	0.29	0.29
7	S7	0.38	0.36	0.43	0.30	0.47	0.30	1.00	0.38	0.43	0.37	0.37	0.34	0.29	0.30	0.17	0.76
8	S8	0.47	0.42	0.62	0.58	0.60	0.43	0.38	1.00	0.36	0.32	0.39	0.47	0.47	0.13	0.16	0.33
9	S9	0.25	0.22	0.19	0.08	0.59	0.31	0.43	0.36	1.00	0.83	0.77	0.75	0.53	0.62	0.56	0.66
10	S10	0.16	0.22	0.19	0.06	0.46	0.38	0.37	0.32	0.83	1.00	0.85	0.74	0.57	0.66	0.61	0.58
11	S11	0.19	0.22	0.24	0.08	0.48	0.39	0.37	0.39	0.77	0.85	1.00	0.86	0.67	0.66	0.67	0.57
12	S12	0.21	0.20	0.23	0.12	0.49	0.44	0.34	0.47	0.75	0.74	0.86	1.00	0.70	0.54	0.56	0.52
13	S13	0.24	0.20	0.29	0.16	0.46	0.71	0.29	0.47	0.53	0.57	0.67	0.70	1.00	0.46	0.57	0.44
14	S14	0.13	0.09	0.07	-0.05	0.27	0.23	0.30	0.13	0.62	0.66	0.66	0.54	0.46	1.00	0.89	0.53
15	S15	0.10	0.06	0.09	-0.02	0.25	0.29	0.17	0.16	0.56	0.61	0.67	0.56	0.57	0.89	1.00	0.38
16	S16	0.32	0.24	0.24	0.11	0.52	0.29	0.76	0.33	0.66	0.58	0.57	0.52	0.44	0.53	0.38	1.00

Table 4.1. Correlation among the input features

In the correlation matrix sensors close to each other show relatively considerable mutual correlations, e.g. sensors 9, 10, 11 and 12. This is normal, as they are located in the middle of the seat and are activated most of the time. Sensors 16 and 7, and 6 and 13, have relatively high correlation as well. The first set of sensors (16-7) is located in the right bolster and the second set (13-6) in the left bolster. It is just natural for them to be depressed for the same positions. There are no cases where sensors located far away from one another have high correlation.

4.2.4 Conclusions

The SOM experiments showed how the sensor layout described in chapter 2 proved to be useful in acquiring meaningful feature vectors. Sensors close to one another generate similar activation values. Therefore, their correlation is considerable. There are no cases were distant sensors generate similar information or, in general, cases of any sensor having identical activation values to another sensor.

The results obtained by the fuzzy *c*-means clustering algorithm in terms of inconsistency and overlapping of the feature vectors in the observation space were confirmed. Clustering algorithms are useful in providing a description of the observation space, but do not perform well in some recognition tasks. Both the fuzzy c-means and the SOM proved futile in partitioning the observation space in all of the considered class systems.

4.3 Neural Networks

Another supervised learning system considered for the development of the pattern recognition system is neural network (NN). NNs are parallel systems inspired by the architecture of biological neural networks, comprising simple interconnected units (artificial neurons or perceptrons) to process information [21]. These units have a transformation function that performs on the weighted sum of inputs to produce an output. All models of artificial neurons resemble their biological counterparts as much as possible [6]. A biological neuron receives an electrical impulse through its dendrites, sums them up, and if the sum exceeds the neuron's body threshold it triggers an electrical signal along its axon. The strength of the incoming signals is determined by the synapse. This phenomenon is simulated in the artificial neural networks as a coefficient, or *weight,* applied to the input signal coming from a specific dendrite.

The ability of NNs to generalize, modify their behavior, and tolerance to noise in the input domain, account for much of the interest in using them as a modeling tool in this project. NN models have some important intrinsic properties, which are advantageous in the modeling context [12]. The *distribution free property*, which allows construction of models without assumption about the underlying distributions of processes of interest; the *learning capability*, which allows the models to be constructed or adjusted solely based on the available data without the intervention of a programmer; and *the parallel processing capabilities*, which permit these models to be transferred to parallel hardware. Many different NN architectures exist, each having their strength and limitations. A complete description of various NN models is provided, e.g. in [16], [15], [18].

4.3.1 Single Layer Perceptron

A single layer perceptron can be considered a hyperplane separator based on its activation function and the change in its inputs, or synaptic weights [16]. Figure 4.8. shows the adjustable synaptic weights on the input lines that can excite or inhibit incoming signals of a perceptron. An input vector **x** = (x_1 ,..., x_M), considered to be a column vector, is linearly combined with the weights **w** = (w_1 ,..., x_M) via the dot product to form the sum

$$s = \sum_{m=1}^{M} w_m x_m = \mathbf{w}^t \mathbf{x} . \qquad \textbf{(12)}$$

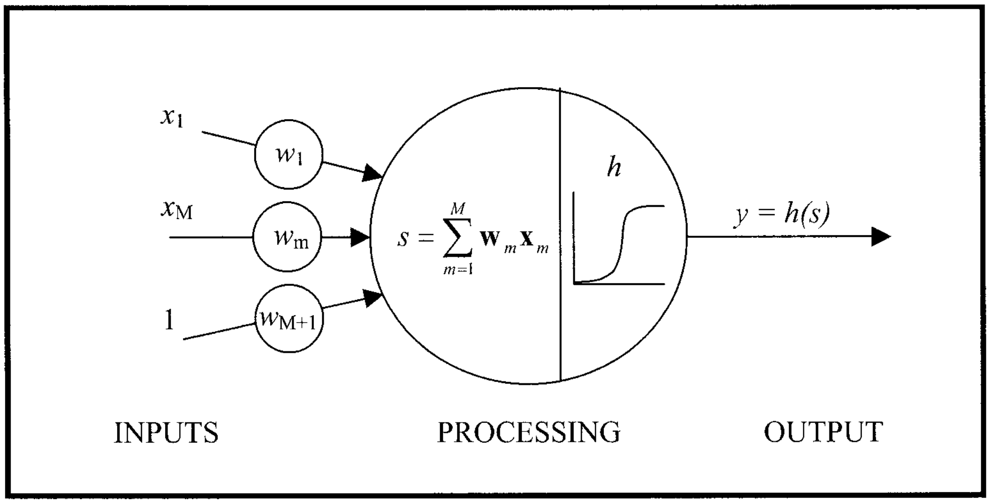


Figure 4.8. Single layer perceptron with unipolar sigmoid function

In the previous figure, the activation function $y = h(s)$ maps the sum s into the proper range of output values. This function is typically chosen as continuous differentiable function so that gradient descent methods can be used to solve for weights that map an input feature vector $\mathbf{x} = (x_1 ,\ldots, x_M)$ into the desired identifier (target) vector $\boldsymbol{y}' = (y'_1 ,\ldots, y'_P)$ that represents a class.

Figure 4.9. presents the shape of the unipolar sigmoid activation function *h(s)*. This function is continuously differentiable and has the form:

$$h(s) = \frac{1}{1 + e^{(-\alpha(s-b))}} \qquad \textbf{(13)}$$

where α is the steepness of the sigmoid function , b is the *bias* that shifts the function center to where e^0 occurs (at $s = b$), where the output is the intermediate value $y = 0.5$. Thus, b is the s-axis center of asymmetry of $h(s)$.

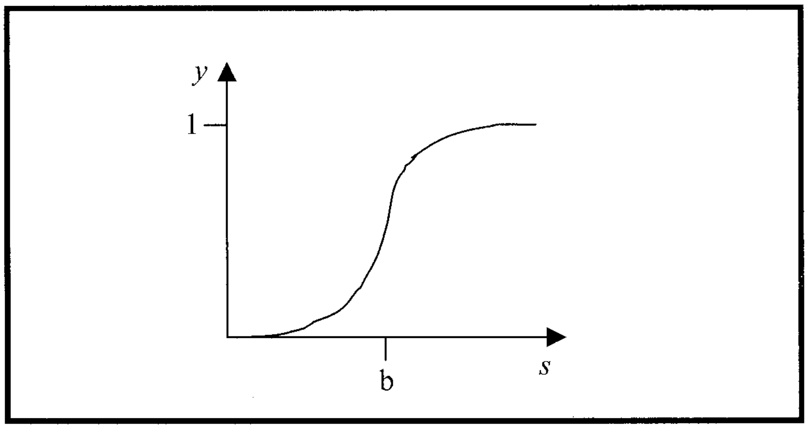


Figure 4.9. Sigmoid Activation Function

4.3.2 Multiple Layer Perceptrons

Multiple layer perceptrons (MLP), or NN, are feed-forward networks with one or more layers of neurons, or perceptrons, between the input and the output. Their capabilities stem from the nonlinearities used within neurons. If neurons were linear elements, then a single-layer network with appropriately chosen weights could exactly duplicate those calculations performed by any MLP. Instead, MLP can form any unbounded convex region in the space spanned by the inputs [15]. Thus, complex recognition systems can be developed using MLP. Figure 4.10. shows a MLP with a single hidden layer of neurons.

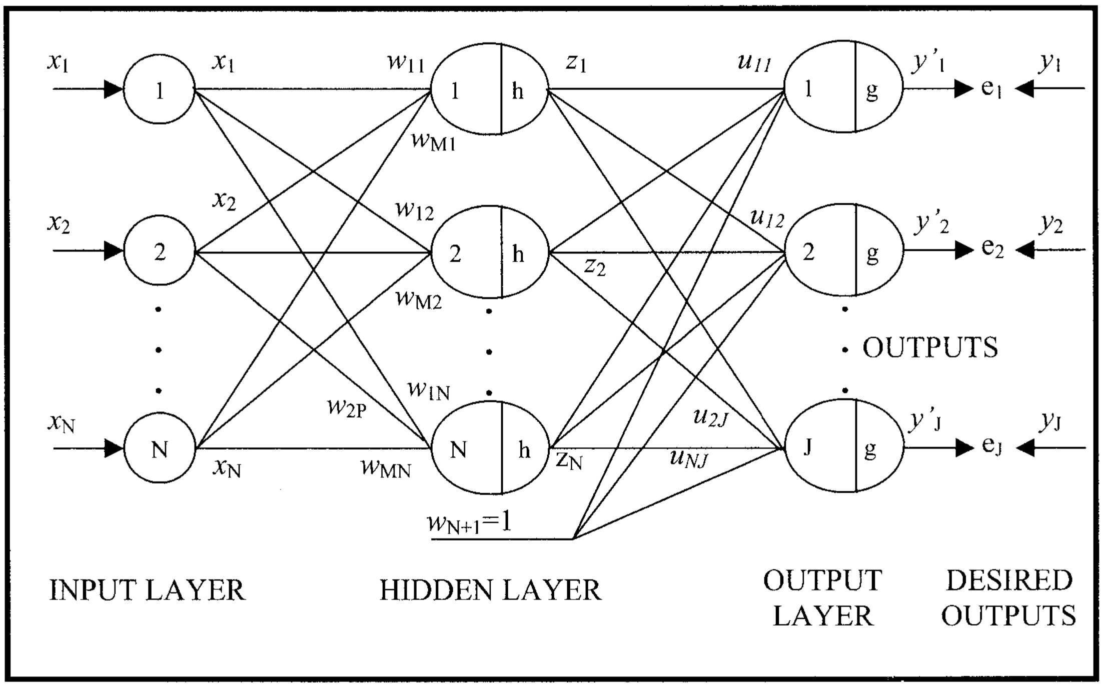


Figure 4.10. Feedforward MLP with one hidden layer

This network has N inputs branching neurons, M hidden neurons, and J output neurons. The weights of the input lines of the middle and output neurons are designated by w_{nm} and u_{mj} respectively.

In figure 4.10., both of the activation functions at the hidden and output layer are unipolar sigmoid functions. The diagram does not show the offset for neurons in the hidden layer to prevent the figure to be unclear, but the neurons of the hidden layer have an offset as one of their inputs, $w_{M+1} = 1$.

There are several algorithms that can be used to adjust, or train, the synaptic weights w_{nm} and u_{mj} in such networks. The most common learning algorithm is called back propagation. It is used in this work

4.3.3 Backpropagation Algorithm

The backpropagation algorithmis an iterative gradient algorithm designed to minimize the mean square error between the actual output of a feed-forward MLP and the desired output.

The basic backpropagation algorithm consists of the following steps [16]:

1. Define the number of input (N), Hidden (M) and output (J) neurons, the number of examples in the training database (Q), the step gain η, and the steepness and bias of the sigmoid function α and b. The value of N depends on the number of features in the input vector $\vec{x}$. The value of J depends on the number of patterns to be recognized

2. Define the maximum number of iterations I, the maximum error E and the minimum error ε

3. Initialize the values of the weights in both the hidden and output layers (w_{nm} and u_{mj}) randomly between -1.0 and 1.0

4. Calculate the output values for the perceptrons of the hidden and output layer (y_m and z_j)

5. Calculate the sum square error (SSE) of the network by the equation 4.4. If the error is less than ε stop iterating. Else go to step 6.

$$E^{(q)} = \sum_{j=1}^{J} \left| t_j^{(q)} - z_j^{(q)} \right|^2 \qquad \textbf{(14)}$$

6. Calculate the derivative error with respect to w_{nm} and u_{mj} :

$$\frac{\partial E}{\partial u_{mj}} = \eta(t_j - z_j) z_j (1 - z_j) y_m \qquad \textbf{(15)}$$

$$\frac{\partial E}{\partial w_{nm}} = \eta \left[(1 - y_m) x_n^{(q)} \sum_{j=1}^{J} \frac{\partial E}{\partial u_{mj}} u_{mj} \right] \qquad \textbf{(16)}$$

7. Update the weights:

$$u_{mj} = u_{mj} - \eta \frac{\partial E}{\partial u_{mj}} \qquad \textbf{(17)}$$

$$w_{nm} = w_{nm} - \eta \frac{\partial E}{\partial w_{nm}} \qquad \textbf{(18)}$$

8. If the number of iterations is equal or greater than I then stop. Else go to step 5.

In this project, the *total mean square error* (TMSE) was used to estimate the inaccuracy of the NN instead of the SSE. The TMSE is a normalized version of the SSE. It allows presenting the error in terms of percentages. The TMSE is defined as:

$$TMSE = SSE / J / Q \qquad \textbf{(19)}$$

Where J is the number of outputs of the NN and Q is the number of examples in the training database. According to Poh [20] the TMSE can be used to determine if the training of the NN was successful. If the verification TMSE is less than twice the training TMSE on the last training iteration, then the training is acceptable. This criterion was used in the training process of the pattern recognition system.

4.3.4 Experimentation and Results

Training, validation and verification databases are built in order to design the NN. The training database contains 65% randomly selected input feature vectors from the observation space. Its main purpose is to provide the teaching inputs for the optimization of the weights in the NN.

The validation database contains 15% randomly selected input feature vectors from the observation space. Its main purpose is to supervise that the NN is not being over trained. This validation set is evaluated by the NN after each training iteration. The validation TMSE should keep decreasing as the algorithm iterates. Increase of the validation TSME indicates specialization or memorization of the training database. This supervision process is referred as *cross validation* in the literature.

The verification database contains the remaining input feature vectors of the observation space (20%). Its main purpose is to verify if the training process has been successful.

The number of inputs of the NN is defined by the number of features in the input vectors. Thus, N = 16.

Since it is not possible to know how many neurons in the hidden layer will provide the best neural network beforehand an iterative process of experimentation is performed. There is a rule of thumb for estimating the size of the hidden layer: the initial number of hidden neurons should be twice the number of classes to be recognized (M = 2K, where K is the number of classes to recognize) [16]. The training process begins with this number of hidden neurons and is increased by one until a NN with 30 hidden neurons is built and tested

The number of outputs varies for the different class systems. Because the purpose of this phase of the project is to determine which model would be more suitable for the recognition process, only NN classifying between a 3-class and a 5-class systems are built and tested. The different class systems are listed in table 4.2.

Number of Outputs	Class #	Weight Threshold
3	1	Empty Seat
	2	Below 103 Lbs
	3	Above 103 lbs
5	1	Empty Seat
	2	Any Child Seat
	3	12 to 59 lbs
	4	60 to 103 lbs
	5	> 103 lbs

Table 4.2. Class systems

All NN are trained for 1000 iterations, with a step gain $\eta = 0.45$, using unipolar sigmoid activation functions with steepness $\alpha = 1$ and bias $b = 0$ in the neurons of the hidden and output layers.

3-Class system

Experiments are performed with NN containing 6 to 30 neurons in the hidden layer. The error graphs for NN trained with 20 up to 25 hidden neurons are shown in figure 4.12. The lines with square points represent the training error. The lines with triangular points represent the validation error.

The validation errors always decrease, meaning that there is no overtraining in any of these NNs. In general, none of the NNs trained showed any sign of overtraining. This figure also shows how the training errors are very low. This means that the NN is able to distinguish clearly among the different classes in this class system in particular.

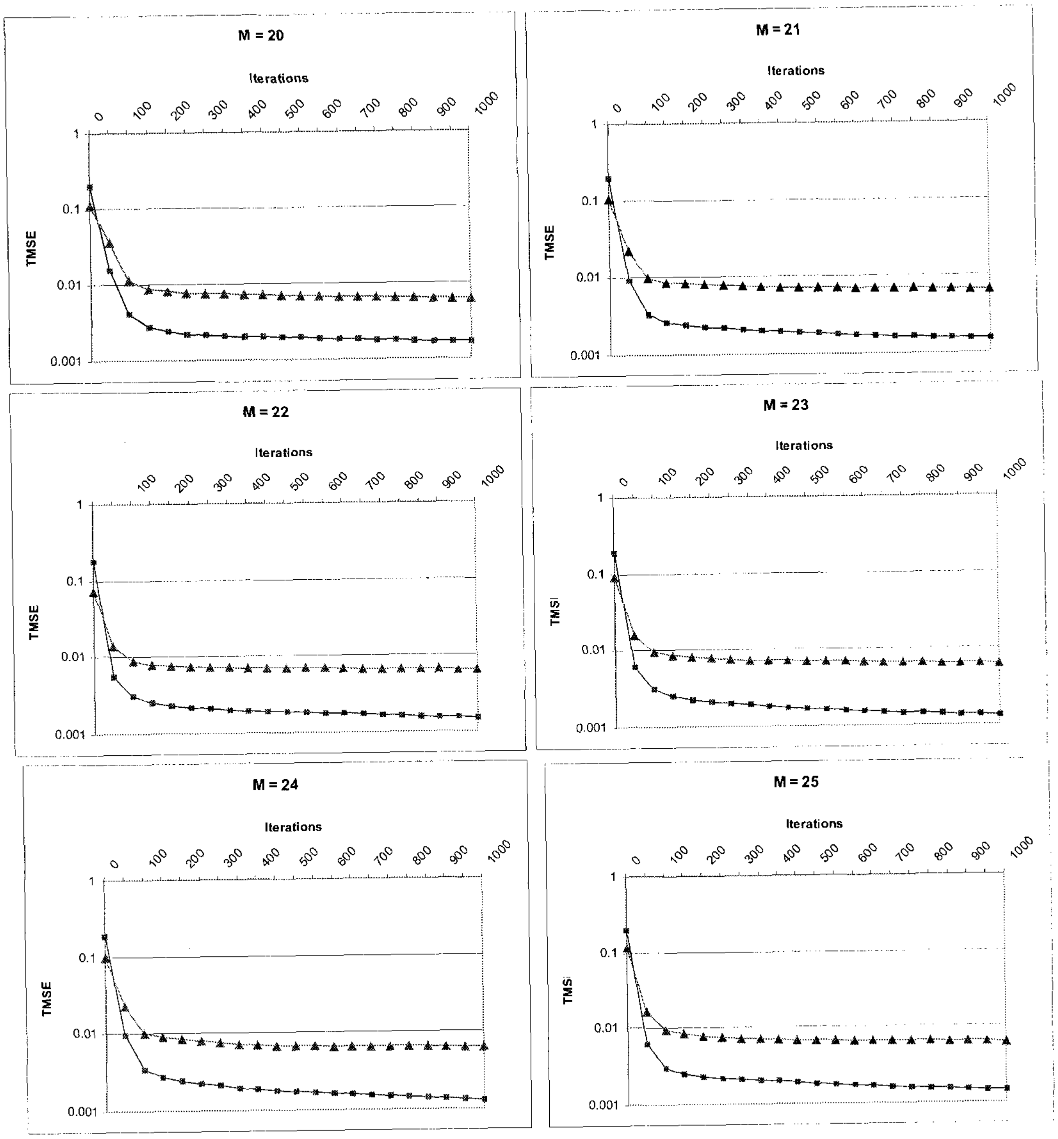


Figure 4.11. Training and Validation error for NN with 20 to 25 hidden neurons for the 3-class system

No. of neurons in hidden layer	Final Training Error (%)	Final Validation Error (%)	Verification Success Rate (%)
6	0.4370	1.3184	97.22
7	4.0518	8.9842	98.28
8	8.2166	18.2086	71.16
9	5.7342	12.716	98.14
10	1.0427	2.6087	98.03
11	0.2213	0.8712	98.10
12	0.6191	1.6769	98.32
13	0.2006	0.7719	98.60
14	0.1908	0.7371	98.46
15	0.2322	0.7724	98.44
16	0.1599	0.6894	98.84
17	0.1710	0.6802	98.66
18	0.1916	0.7125	98.73
19	0.1921	0.7586	98.17
20	0.1658	0.6118	98.96
21	0.1545	0.6570	98.73
22	0.1495	0.6384	98.69
23	0.1336	0.6258	98.73
24	0.1311	0.6342	98.87
25	0.1453	0.6262	98.71
26	0.1498	0.6063	98.82
27	0.1418	0.6486	98.71
28	0.1469	0.5670	98.75
29	0.1374	0.6236	98.71
30	0.1733	0.6444	98.44

Table 4.3. Final error values for NN trained for 3-class system

Table 4.3. shows the summary of the final training and validation error values, as well as the verification success rate, for all the NNs trained for this class system. This table shows that the range of most accurate NN, in terms of training error and verification success rate, lies between 20 and 26 hidden neurons. The NN trained with 24 hidden neurons provided the lowest training error, but the NN trained with 20 hidden neurons have better success rate in verification.

This experiment confirmed that the NN can overcome the problems of noise and overlapping in the observation space. NN provides an excellent accuracy of almost 99% for the 3-class system specified in chapter 2

5-Class System

Experiments are performed with NN containing 10 to 30 neurons in the hidden layer. The error graphs for NN trained with 25 up to 30 hidden neurons are show in figure 4.12. In these graphs the lines with square points represent the training error and the lines with triangular points represent the validation error just as in the previous experiment. For this experiment, the cross validation showed no sign of overtraining or memorization in the training of the NNs either.

The training error is very low for this set of experiments, similar to the NN trained for the 3-class system. These results confirm once more the robustness of the NN to noise, and its capability to distinguish among overlapping classes.

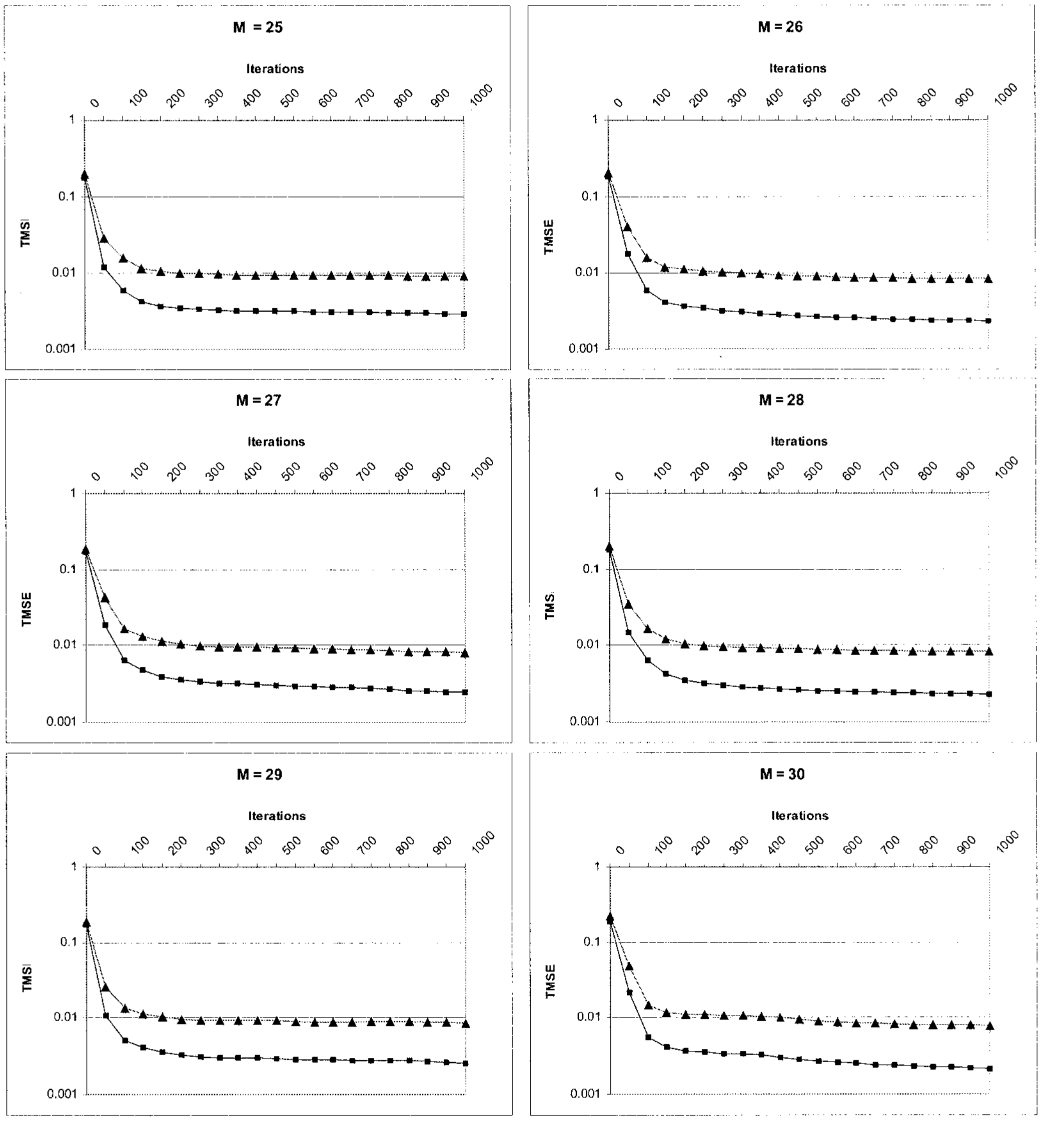


Figure 4.12. Training and Validation error for NN with 25 to 30 hidden neurons for the 5-class system

Table 4.4. shows the summary of the final training and validation error values, as well as the verification success rate, for all the NN trained for this class system. The NN trained with 30 hidden neurons provided the lowest training error, however the NN trained with 28 hidden neurons had more success over verification.

No. of neurons in hidden layer	Final Training Error (%)	Final Validation Error (%)	Verification Success Rate (%)
10	0.4589	1.2394	94.85
11	0.6375	1.6673	95.25
12	0.4061	1.984	96.11
13	0.3639	1.1206	96.13
14	0.4798	1.3508	96.11
15	0.4317	1.2446	96.00
16	0.3416	1.0755	96.23
17	0.3378	1.0560	96.25
18	0.3230	1.0073	96.84
19	0.3913	1.1555	96.68
20	0.2917	0.9235	96.88
21	0.3348	1.002	97.11
22	0.2866	0.9514	97.20
23	0.2427	0.8078	97.20
24	0.2661	0.9207	96.86
25	0.2903	0.9198	96.97
26	0.2324	0.8397	97.22
27	0.2430	0.8111	97.44
28	0.2258	0.8183	97.62
29	0.2523	0.8584	97.15
30	0.2149	0.7839	97.38

Table 4.4. Final error values for NN trained for 5-class system

Although satisfactory, the results for the 5-class system are not as good as the results for the 3-class system. The lowest training error for the 5-class system was 0.08% higher than its equivalent in the 3-class system. The best verification success rate is 1.4% lower in the 5-class system compared to the 3-class system. These variations might have been caused by the inequality in the number of training examples of each class for the 5-class system. While the 3-class system observation space contains about 7000 feature vectors for each class, the 5-class system has about 6000 feature vectors in three of its classes and about 800 in the remaining two classes. This unbalance in the observation space could have caused the NN to focus on the classes that are represented in major proportion in the observation space. However, the NNs deal very well with this unbalance, and are able to distinguish among the classes in an acceptable way.

4.3.5 Conclusions

The set of experiments performed confirmed the ability of neural networks to generalize, modify their behavior, and tolerate noise in the observation space. The overlap of the input feature vectors in the observation space is overcome successfully by the NN alone. This is ideal for the final development of this project.

The implementation of the 3-class system, as specified in Chapter 2, has a very good outcome. The results are very promising, even though a 100% verification success rate was not achieved. A better adjustment of the parameters of the NN and the use of a filtering algorithm to take out the overlapping feature vectors in the observation space would likely lead to a more accurate classification.

It is unlikely that the 6-class and 9-class classifiers could be implemented using NN. The experiment described in this chapter illustrates the importance of balanced training data in terms of proportion of training examples belonging to each class.

In general, out of the classification methods described in this chapter, NN provide the best and practically acceptable results.

Chapter 5.

Final Development of Pattern Recognizer

Training and evaluation are the final steps of the development of a pattern classification system according to the scheme presented in Chapter 1.

Part of the training stage for this project was implemented, tested, and its results presented, in the previous chapter. This chapter focuses on the optimization of the training stage in order to achieve the best possible recognition rates.

Several evaluation methods to test the final pattern recognition system's accuracy and robustness are also presented.

5.1 Filtering of Database

The results presented in Chapter 4 show how the NN is the best model to create the 3-class pattern recognition system. A Training error of 0.13% and a verification success rate of almost 99% show their efficiency discriminating the observation space into three different classes. However, the goal for this project is to achieve 100% recognition rate.

The first step in order to achieve this goal is to overcome any critical overlapping of the input feature vectors for different classes in the observation space. A filter is designed to erase the input features that are similar, but labelled to belong to different classes. This filter calculates the Euclidean distance between two different feature vectors in the observation space. If this distance is below a specified threshold and the two feature vectors are labelled with different targets, both feature vectors are erased from the observation space. Otherwise the feature vectors remain untouched.

The algorithm for this filter is compound to the following steps:

1. Normalize each dimension $S_m^{(q)}$ of the input feature vectors in the observation space using

$$Snorm_m^{(q)} = \frac{S_m^{(q)} - S\min_m}{S\max_m - S\min_m} \qquad \textbf{(20)}$$

2. Define the minimum distance threshold ε between two feature vectors belonging to different classes.

3. Draw feature vector $\mathbf{Snorm}^{(q)}$ from the observation space.

4. Compute distance of feature vector $\mathbf{Snorm}^{(q)}$ to all other feature vectors in the observation space $\mathbf{Snorm}^{(q^*)}$ via:

$$D_{qq^*} = \sum_{m=1}^{16} \left| Snorm_m^{(q)} - Snorm_m^{(q^*)} \right| \qquad \textbf{(21)}$$

5. if $D_{qq^*} < \varepsilon$ and class label is different erase both feature vectors from observation space. Else keep comparing.

6. If q = Q (number of examples in observation space) stop. Else go to step 3.

The main issue when using this algorithm is to determine the value of the minimum distance threshold ε for two feature vectors with different labels. At first, ε was thought to be 5% of the maximum distance in observation space. Because the feature vectors have 16 dimensions and the filter algorithm requires their values to be normalized between 1 and 0, the maximum Euclidean distance in a hypercube with this number of dimensions between the origin and the farthest point would have a value of 4. Therefore, ε = 0.2. However, the farthest distance between two feature vectors in the observation space for this experiment is 3.01 (25% less than the theoretical maximum). In order to keep some proportion in the way the feature vectors are distributed, the value of ε is redefined as 2.0% of the maximum distance. Therefore, ε. = 0.06.

The filter erased 0.6% of the observation space. The final distribution of the feature vectors in the observation space is presented in figure 5.1.

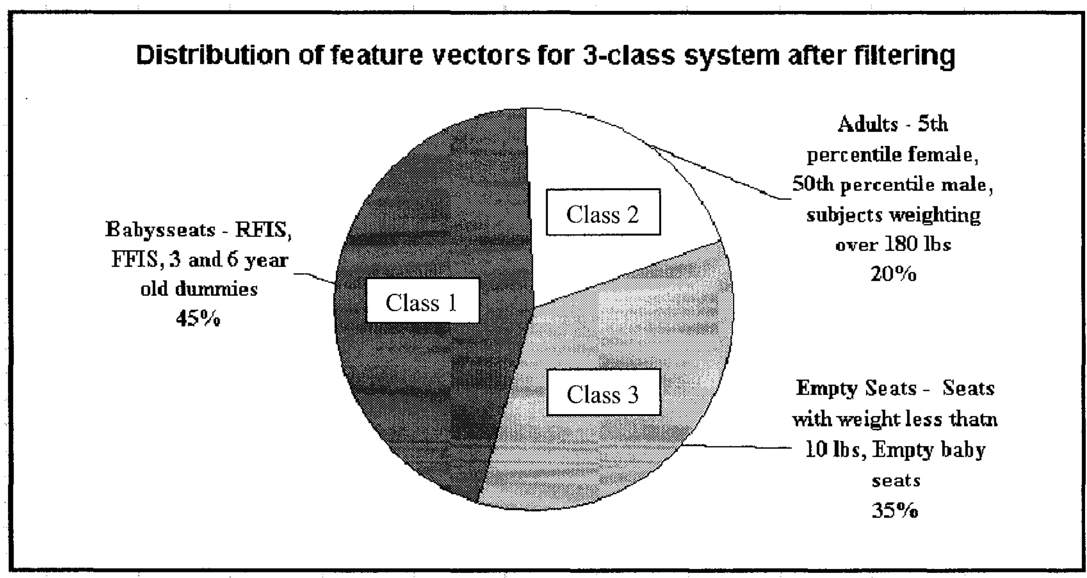


Figure 5.1. Distribution of feature vectors in observation space after filtering process

5.2 Neural Network Design

The final implementation of the recognition system is based on the experiments presented in section 4.3.

The results illustrated in Figure 4.12 shows that 1000 iterations are enough to train the NN properly without making it memorize the training input feature vectors. Therefore, the NN is trained for 1000 iterations for this final implementation, without cross validation. The filtered observation space is divided in two: a training database containing 66% of the feature vectors, and a verification database containing the remaining 33% of the feature vectors.

The rest of the parameters are kept the same: step gain η = 0.45, unipolar sigmoid activation functions with steepness $\alpha = 1$ and bias $b = 0$ in the neurons of the hidden and output layers.

Table 4.3 shows that a NN with 24 hidden neurons had a better training rate, but a NN with 20 hidden neurons has a better verification success rate. The 24-hidden-neuron architecture is chosen for this final implementation because it has a greater capacity to more finely partition the observation space due to its major number of synaptic weights [16].

After several runs, the best NN trained has 0.10% training error and 99.67% verification success rate. In raw numbers, the trained NN misclassified 24 input feature vectors out of 7310 in the verification stage.

5.3 Methods used to further improve the neural network performance

Even though the results presented in the previous section show how accurate the trained recognition system is, 100% verification success rate remains the desired goal. This section explores additional methods to be used in order to achieve this performance.

5.3.1 Normalization of the Observation Space

The observation space of this project can be considered voluminous in feature vectors. These types of search spaces usually contain large amount of noise and missing data. The results from the fuzzy c-means and the SOM experiments show that at least the first of these problems is present in this project.

One way to overcome the problems stated previously could be to provide feature values to the pattern recognizer in a limited range. *A normalization* of the observation space would reduce the amount of noise and could bring values of the feature vectors closer, making up for missing data.

For this experiment the observation space is normalized using formula **(20)**. NNs containing 20 to 24 hidden neurons are trained with the same parameters described in section 5.1.

NN Architecture	Final Training Error (%)	Final Validation Error (%)	Verification Success Rate (%)
16-20-3	0.2329	0.7672	98.33
16-21-3	0.2087	0.7970	98.28
16-22-3	0.2134	0.7552	98.48
16-23-3	0.2016	0.7410	98.48
16-24-3	0.2060	0.7552	98.51

Table 5.1. Final error values for NN trained for 3-class system with normalized observation space

Table 5.1. summarizes the results of this experiment. The NN with 23 hidden neurons shows the lowest training error. However, this training error is 0.1% higher than the one for the non-normalized observation space. The NN with 24 hidden neurons has the best verification success rate. However it is 1.06% lower than the one for the non-normalized observation space.

This experiment leads to the conclusion that besides adding computation time, normalizing the observation space does not improve the recognition rate of the pattern recognition system.

5.3.2 Addition of hidden layers

There is no need for multiple hidden layers in a NN structure, as a single layer is sufficient to form an arbitrarily close approximation to any nonlinear decision boundary provided that there are enough neurons in the hidden layer [18]. However, extra multiple hidden layers can be more efficient in training and modeling highly nonlinear mapping with a large number of nonlinearly separable classes [16], as in this case.

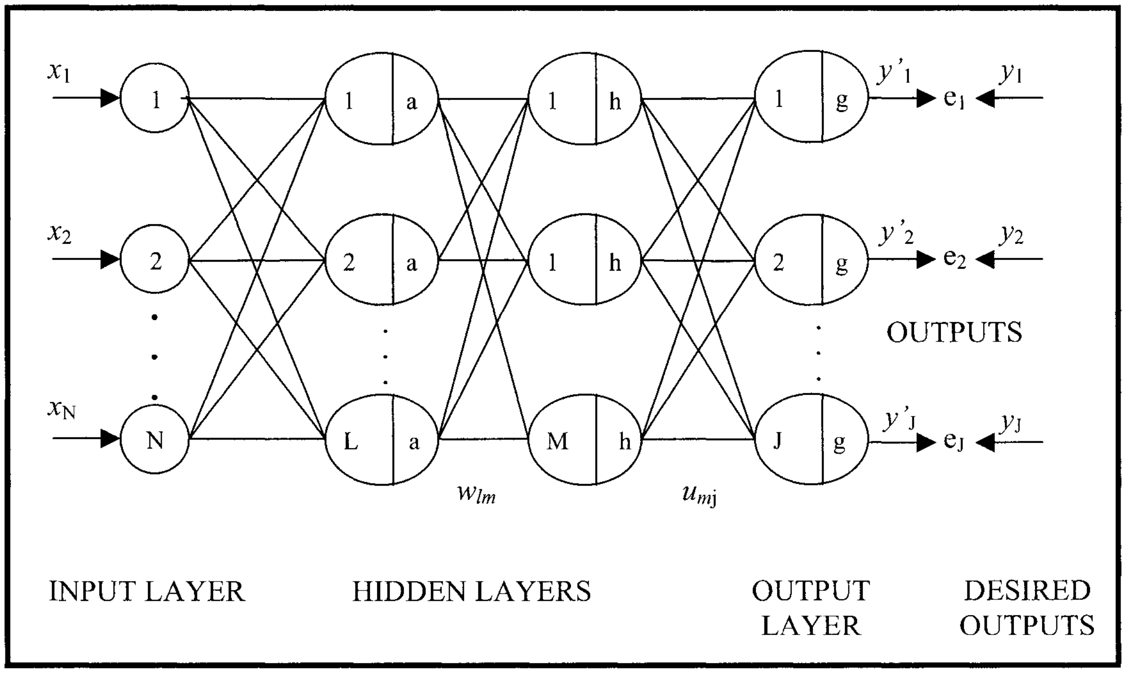


Figure 5.2. Feedforward neural network with multiple hidden layers

A MLP with two hidden layer is presented in Figure 5.2. It has N inputs branching nodes, L neurons in the first hidden layer, M neurons in the second hidden layer, and J output neurons. The weights of the input lines of the middle and output neurons are designated by v_{nl}, w_{lm} and u_{mj} respectively

The backpropagation algorithm is also used to train this type of networks. It is identical to the one described in section 4.3.3, only the updating of the weights in step 6, in the first hidden layer is different:

$$v_{nl} = v_{\ln} - \eta_2 \frac{\partial E}{\partial u_{mj}} [\sum_{m=1}^{M} u_{mj} y_m (1 - y_m) w_{lm}] a_l (1 - a_l) x_n \qquad \textbf{(22)}$$

Just as in the case of the single hidden layer NN, the number of nodes in the multiple hidden layers NN can not be calculated in advance. The best architecture can be reached by experimentation only. In our experiment, the number of neurons in the first hidden layer is around 24, and the number of neurons in the second hidden layer is always less than the number of neurons in the first hidden layer.

These NN are trained for 1000 iterations using 66% of the filtered observation space as training data and 33% as verification data. The step gain remained $\eta = 0.45$, and unipolar sigmoid activation functions with steepness $\alpha = 1$ and bias $b = 0$ are used in the neurons of both hidden layers and the output layer.

Table 5.2. summarizes the results of the architectures tested in this experiment.

NN Architecture	Final Training Error (%)	Verification Success Rate (%)
16-18-8-3	0.485	97.14
16-18-12-3	0.476	97.14
16-20-12-3	0.478	97.14
16-22-14-3	0.568	97.46
16-24-16-3	0.465	97.65
16-25-16-3	0.432	97.40
16-26-18-3	0.455	97.63
16-28-18-3	0.402	98.07
16-30-20-3	0.476	97.75
16-35-25-3	0.413	98.43

Table 5.2. Final values for multiple hidden layer NN trained for 3-class system

Adding more hidden layers to the NN neither improves the training rates nor improves the verification success percentage with respect to the single hidden NN. It seems that attempting to solve a relatively simple problem with a more complicated solution does not bring any benefits, at least in this case.

5.4 Robustness of the Neural Network

The last set of experiments measures how robust the NN is to decrements in the resolution of the values generated by sensor layout and to failures in any of the sensors. Testing of how much the recognition rate decreased in these cases is also conducted.

5.4.1 Rounding of values of input feature vectors

The dimensions of the input feature vectors are floating point numbers with four decimal places. This is a large number of decimal places if processed by hardware. In order to test how robust the NN is, a rounding of the values of the feature vectors from four to two-and-a-half decimal places is conducted. Two-and-a-half decimal places rounding connotes assigning a value of either five or zero to the third decimal place of each dimension of the feature vectors.

For a number with four decimal places denoted as $D.d_1d_2d_3d_4$ (i.e. 1.2345) the rounding criterion was:

1. If $d_3d_4 < 30$, then $d_3 = 0$
2. Else If $30 <= d_3d_4 < 80$, then $d_3 = 5$
3. Else if $d_3d_4 > 80$, then d_2 was incremented one unit and $d_3 = 0$.

After rounding the values of the feature vectors of the observation space, a NN is trained using the 16–24–3 architecture. The success rate over verification is 99.7% on a verification database of 7335 examples. That is an increase of 0.03% over the non rounded observation space. The small improvement of the verification success rate is likely due to observation error. However, it demonstrates that using lower precision representation of sensor values would not affect the overall performance of the system. This is important for eventual hardware implementation of the system using an 8-bit microprocessor architecture.

Real life verification has been conducted using two baby seats, two male adults weighting over 180 Lbs and the empty seat. The system has proven to work correctly for all the testing positions of all the testing subjects. Further testing on all types of occupants is required in order to reach a final and definitive conclusion. However, for this first set of experiments the performance of the system is acceptable.

The rounding of the examples in the training database from four to two and a half decimal places does not affect the performance of the NN for the 3-class system. Further analysis will have to be made for NN involving the recognition of more than 3 classes.

5.4.2 Sensor relevance

The objectives of this experiment is to determine which sensor, if any, can be discarded from the sensor layout described in Chapter 2, and to establish which sensors provide most valuable information to the NN.

Different NNs with the 16-24-3 architecture are trained using the same training database and parameters described in section 5.2. The number of NNs trained is determined by:

$$\text{NNs trained} = 2^{*}\ \text{Number of Inputs} \qquad \textbf{(22)}$$

Thus, the relevance of each sensor is measured several times on NNs that were trained with the same information, but that started their training in different places of the observation space due to the use of random initial weights. This measure allows an overall visualization of the relevance of each sensor.

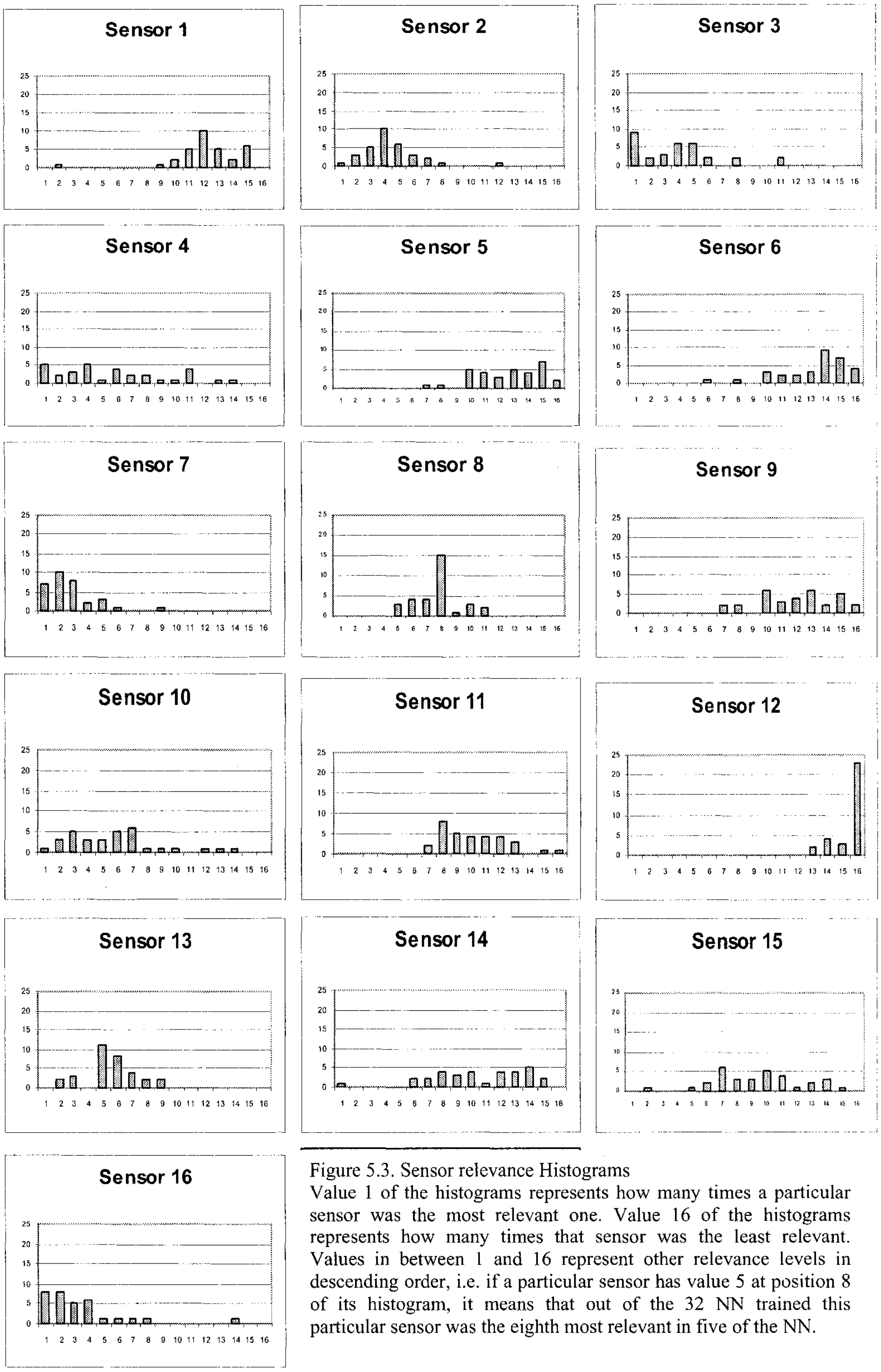


Figure 5.3. Sensor relevance Histograms
Value 1 of the histograms represents how many times a particular sensor was the most relevant one. Value 16 of the histograms represents how many times that sensor was the least relevant. Values in between 1 and 16 represent other relevance levels in descending order, i.e. if a particular sensor has value 5 at position 8 of its histogram, it means that out of the 32 NN trained this particular sensor was the eighth most relevant in five of the NN.

The sum square of the weights (SSW) from the input to the hidden layer is chosen as criterion of relevance. The most relevant features should have high SSW, because these inputs should stimulate the NN in greater proportion by presenting information of higher relevance.

The SSW for each input is defined as:

$$SSW_n = \sum_{m=1}^{M} w_{nm}^2 \qquad \textbf{(23)}$$

Where M is the number of hidden neurons and w_{nm} is the matrix containing the weights from the input layer to the hidden layer of the NN.

Figure 5.3. presents the histograms with the relevance tendency for each sensor. Sensors 2, 3, 7 and 16 appear among the 4 most relevant sensors in more occasions. Their relevancy can be considered high. Sensors 1, 5, 6, and particularly, 12 appear to be least relevant to the NN in most of the cases.

Figure 5.4. shows the averaged SSW of each sensor over all the NN trained. The figure shows that sensors 2, 3, 7 and 16 tend to be the most relevant and sensors 12, 6, 5 and 1 tend to be the most irrelevant, just as in the results discussed above.

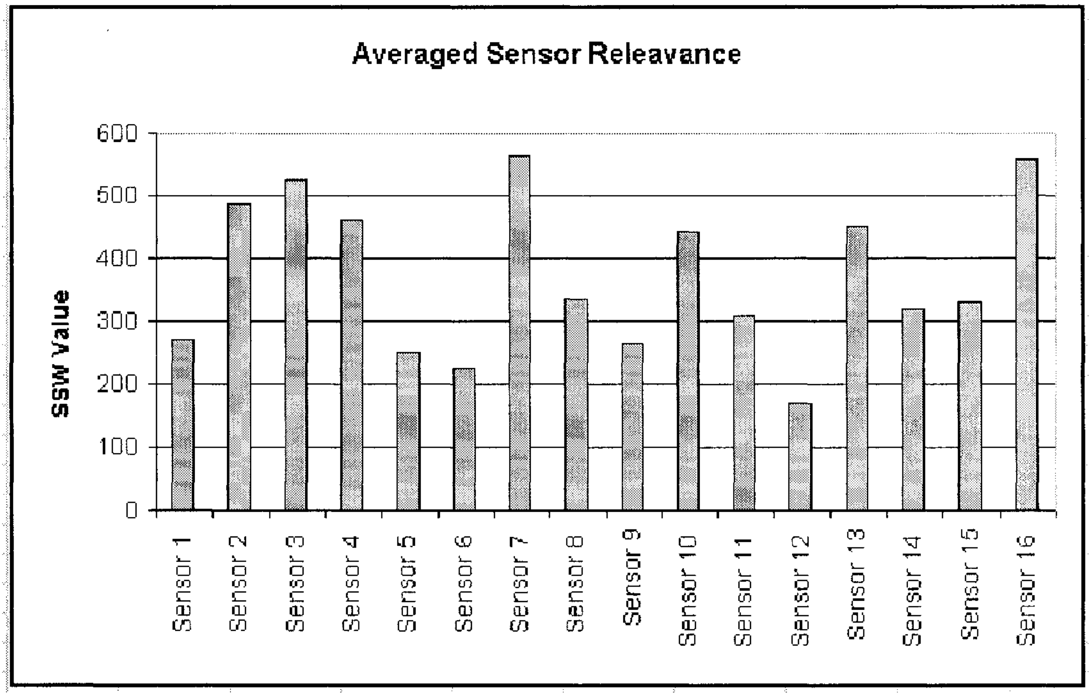


Figure 5.4. Averaged relevance of each input feature

According to the results of this experiment, sensor 12 seems to have little relevance for the system. It could be discarded without compromising the performance of the NN. However, it could not be wise to do so, as the analysis shows how sensors 7 and 16 (right bolster of the car seat) are very relevant. This might be due to the pressure exercised on the seatbelt with the device used to secure the baby seats to the car seat. This form of securing the baby seats might be causing them to lift from the left side of the car seat. Because of this lifting sensor 12 is not being activated fully, making it irrelevant to the NN. This type of securing the baby seats should be reconsidered, as it is causing the relevance of the sensors to be very asymmetrical from the center of the car seat, subsequently it is making the NN to rely on sensors that might not be activated fully in other circumstances. Sensor 13 has high relevance. This is possibly due to the pressure the buckle of the seatbelt exerts on this particular sensor.

5.4.2 Sensitivity to sensor failure

The objective of this experiment is to determine the worst-case success percentages if any one, two or three sensors in the car seat fail at the same time.

Voltage of 0V is outputted by a sensor when it fails. This experiment runs all the input feature vectors through the NN for evaluation, making one sensor fail at the time. Then, the impact of a combination of two and three sensor failing is studied.

The current observation space is used as input data. This database has a considerable number of positions the sensor layout in the car seat could generate, so it covers a substantial amount of the input patterns the NN could receive at any time while a sensor or a number of sensors fail. The success percentages shown here are calculated over all 22118 feature vectors of the observation space.

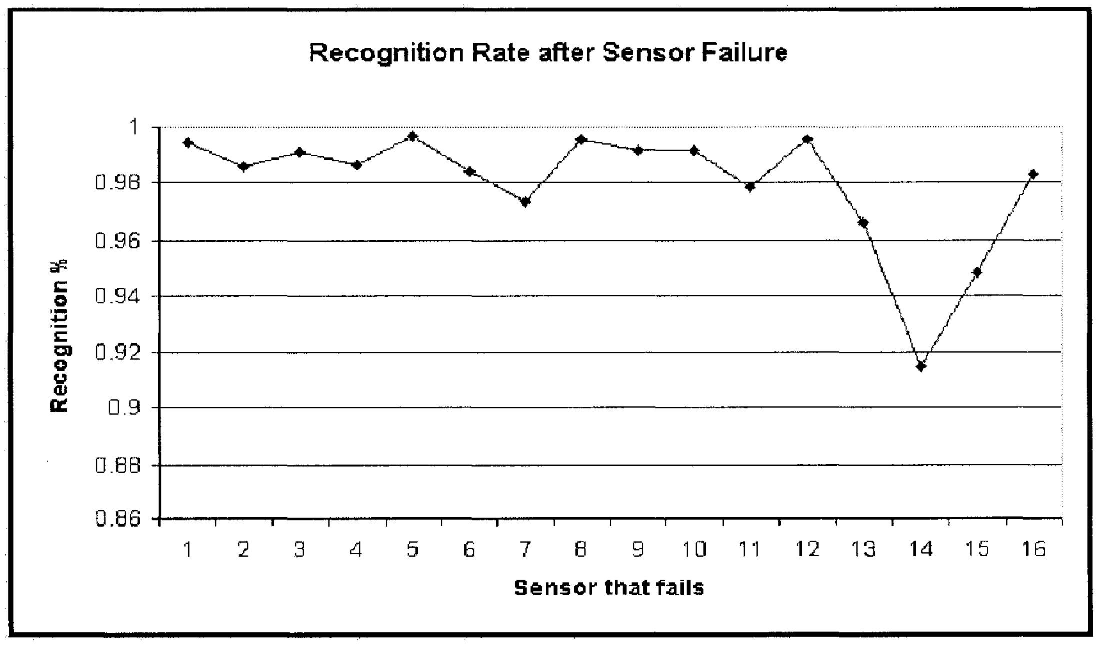


Figure 5.5. Recognition rates after one sensor failure

The results of making one sensor fail at the time shows the NN to be very robust. As long as the sensor that failed is not very relevant to the NN, the recognition rates remain high (Figure 5.5). In some cases, if the least relevant sensor fails (sensor 12) the success rate would still be very high (over 99%). If a relevant sensor failed the success percentage can drop dramatically. The system proved especially sensible to failure of sensor 14. This sensor is highly activated by all occupants in the car seat, bringing very relevant information to the system. After three experiments, given the worst-case scenario, the success rates for the NN range from 90.04% to 91.5%. Considering the lowest possible rates it can be concluded that if any given sensor of the NN fails the success percentage rate would drop to 90%.

Experiments having two and three sensor fail at the same time show the same tendency. Because of large number of cases in these experiments, the results are presented in Appendix C. The worst-case scenarios success rates when two sensors fail range from 81.22% to 83.66%. It can be concluded then that if any two sensors fail at the same time, given the worst case scenario, the success rate would drop to 80%. The worst-case scenarios success rates when any three sensors fail at the same time range from 71.95% to 78.12%. It can be concluded then that if any three sensors fail at the same time, given the worst case scenario, the success rate will drop to 70%.

The NN looses too many features if more than three sensors fail at the same time. It is no longer able to perform proper recognition of the different classes.

Chapter 6

General Conclusions and Discussion

The objective of this thesis is design a pattern recognizer for a car seat occupancy system based on rigorous study of data and using methods of computational intelligence. This work focuses on finding a model or a set of models able to properly distinguish among different classes of occupants in car seats specified by the FMVSS 208 standard. The project's design cycle consists of three main phases: collection of data, choosing of recognition model, and training and evaluation of classifier.

One of the main problems in the initial phase of the project is to quantify the number of objects necessary in the observation space in order to have enough information to create the pattern recognizer. This is a critical issue, as the rest of the system is based on the completion of this observation space. In addition, the process of data gathering is time consuming and relies on help of volunteers, making the preparation and scheduling of the data acquisition sessions essential to meet the development time constrains. According to estimations between 22000 and 24000 examples are needed to successfully train and evaluate a recognition system for this purpose. The final number of objects in the observation space lies within that range and leads to the design of a system that works correctly both in simulation and real life, showing the effectiveness of the estimation methods.

Several recognition methods have been tested once the observation space was completely populated. Some of these algorithms provide a way to visualize the data. They are useful when the distribution of the observation space is unknown and an initial description is required. In our case, the desired division of the observation space is known. However, the clustering methods can be used for verifying whether these required classes correspond to the natural clusters, if any, present in the data. Fuzzy c-means and SOM are used for this purpose. Both methods show that feature vectors representing different positions belonging to different occupants are similar to each other, creating big overlaps among the classes. The only feature vectors that proved to be consistent belong to the empty seat class. This problem can be partly overcome by creating more defined boundaries among the classes by removing those feature vectors that cause the major overlaps. The generalization property of the NN helps to resolve this problem completely.

The results obtained with clustering algorithms also show high correlation in the outputs from sensors that are placed close in the mat. That means that several sensors could be providing the same information to the recognition system and some of them could be eliminated. This could lead to a sensor layout that gives the same information to the NN with fewer sensors, thus decrementing hardware, processing time and NN complexity. The results on NN robustness and sensor failure confirm this assumption. However, they also show how this redundancy in the input features helps the robustness of the NN in case of sensor failure. The main issue of designing this system is to increase vehicle protection in case of an accident. Therefore, it would be safer to leave the configuration the way it is and assume the extra processing cost. This is probably one possible topic to research in future projects.

After evaluating the results of clustering methods, single hidden layer NN are used as the main model to perform recognition in the system. They show to be very effective, especially because they overcome the problem of overlapping of the feature vectors in the observation space and generalize the information given to create the desired classes in the system. Not only were they very accurate for recognition, but also very robust to failures in the system. They proved to work correctly with less accuracy in the input values and even with failure in some of the sensors. Multiple hidden layer NN did not improve recognition rates with respect to single layer hidden NN. Neither did normalizing the input feature vectors to try to overcome the overlapping of the input feature vectors. Single hidden layer NN's recognition efficiency remained high in both simulation and real situations, driving to the conclusion that they were the proper choice for the development of the system.

The objectives of the project are met. The pattern recognition system has been successfully developed. The recognition rate on simulation is 99.6%, and on real life testing is 100%. This difference in recognition rates is due to the addition of positions not specified in the FMVSS 208 standard used in the training and verification processes to add reliability and robustness to strange inputs to the system.

This project focused on the 3-class system described in section 2.1. For the development of the 6 and 9-class systems a combination of recognition techniques might be necessary. The experimentation with fuzzy c-means shows how the overlapping among classes increases with increasing number of classes to recognize. The experimentation with NN in order to classify 5 classes gives satisfactory, although not accurate enough results. The verification success rates on simulation should remain above 99% and the training error below 0.1%. Recognition at two levels may be necessary to develop an accurate pattern recognizer: one model to distinguish the general features at a high level, possibly based on the 3-class system already built, and a model to recognize the finite differences among classes at a low level, one that specializes and distinguishes among classes that are very similar. The goal in mind for future projects will be to achieve 100% recognition rates on real life testing for positions specified by the FMVSS 208 standard and Car Manufacturers due care positions provisions.

Bibliography

[1] Allegro Corporation, Radiometric, Linear Hall-Effect sensors for High Temperature Operation A3515, Data sheet, 2001

[2] Bai, X., Analysis of Software Measures by Self Organizing Maps, MSc Thesis, University of Alberta, Edmonton, Canada, September 2001

[3] Bange, M., Jordan S., Biermann, M., Kaempke, T., Scholz R.D., Fast Object Detection For Use Onboard Satellites, Kluwer Academic Publishers; Experimental Astronomy 13: pp 101-117, 2002

[4] Barreiro, G., Segura, F.E., Autonomous Environment Support for a Person with Physical Limitations, In the preceedings of the Congreso Iberoamericano Iberdiscap2000, Madrid, Spain, October 2000

[5] Bartlett, P.L., Maass, W., Vapnik-Chervonenkis dimension of neural nets, In Michael A. Arbib (editor), The Handbook of Brain Theory and Neural Networks, 2nd Edition, MIT Press, Cambridge, MA.. 2002.

[6] Cios, K., Pedrycz, W., and Swiniarski, R., Data Mining Methods for Knowledge Discovery, Kluwer Academic Publisher, 1998

[7] Danisch, L., Englehart, K., and Trivett, A., Spatially continuous six degrees of freedom position and orientation sensor, Sensor Review, Vol 19, No. 2, 1999, pp 106-112

[8] Duda, O., Hart, P.E., Stork, D.G., Pattern Classification , 2nd edition, Wiley-Interscience, , New York, 2001

[9] Hash, D.R., Horne, B.G., Progress in Supervised Neural Networks: What's New Since Lipmann, IEEE Signal Processing Magazine, January 1993, pp 8 -39

[10] Hathaway, R.J., and Bezdek, J.C., Switching Regression Models and Fuzzy Clustering, IEE transanctions on Fuzzy Systems, Vol. 1, No. 3, August 1993, pp 195-204

[11] Honkela, Y., Self-Organizing Maps in Natural Language Processing, Phd Thesis, University of Helsinki, Finland, December 1997.

[12] Hsu, W., and Tenorio, M. F., Software Engineering Effort Models using Neural Networks, IEEE International Conference on Neural Networks, Piscataway NJ, 1991, pp 1190-1195

[13] Jarillo, G., Analysis of Software Engineering Data Using Computational Intelligence Techniques, MSc. Thesis, University of Alberta, Edmonton, Canada, May 2003

[14] Kohonen, T., Self-Organizing and Associative Memory, 3rd edition, Springer-Verlag, Berlin, 1989

[15] Lippmann, R., Introduction to Computing with Neural Nets, IEEE Acoustics, Speech and Signal Processing Magazine, April, 1987, pp 4-22

[16] Looney, C. G., Pattern Recognition Using Neural Networks, Theory and Algorithms for Engineers and Scientists, Oxford University Press, 1997

[17] Mair, C., Lefley, G. M., et all, An investigation of machine learning based predictive systems, The journal of systems and software, 53, 2000, pp 23-29

[18] Makhoul, A., El-Jaroudi, A., and Schwartz, R., Formation of disconnected decision regions with a single hidden layer, In Proceedings International Conference on Neural Networks, Washington DC, 1989, Vol. I, pp 455-460

[19] National Highway Traffic Safety Administration, Federal Motor Vehicle Safety Standard No. 208 Air Bag On-off switches: Final Economic Assessment, Washington, - U.S. Department of Transportation, November 1997

[20] Poh, H.L., A Neural Network Approach for Marketing Strategies Research and Decision Support, PhD Thesis, Stanford University, 1991

[21] Tessmer J., Comparing International Crash Statistics, Journal of Transportation and Statistics, December 1999, pp 169-166

[22] Vapnik, V. N., Chervonenkis, A. Y., On the uniform convergence of relative frequencies of events to their probabilities, Theory of Probability and its Applications, Vol 16, pp 264-280, 1971

[23] Vesanto, J., SOM-based data visualization methods, Intelligent Data Analysis, Vol 3, pp 111-126, 1999

[24] Vurpillot, S., Inaudi, D., Scano, A., Mathematical model for the determination of the vertical dispkacement from internal horizontal measurements of a bridge, SPIE Smart Structures and Materials, Vol 2719, 1996, pp 46-53

[25] Zeng, T.Q., Cowell, P., Zhou, Q., The User's Approach (UshA) for the Realization of Fuzzy Clustering Theory for Zonal Analysis in Raster-Based GIS, Second Annual Conference of GeoComputation, University of Otago, New Zeland, 1997

WWW Links

[26] Docket No. NHTSA 00-7013 – Notice 1, National Highway Traffic Safety Administration - U.S. Department of Transportation
http://www.nhtsa.dot.gov/airbag/AAPFR/reg/, December 21, 2003

[27] Docket No. NHTSA 99-6407; Notice 1, National Highway Traffic Safety Administration - U.S. Department of Transportation
http://www.nhtsa.dot.gov/cars/rules/rulings/aairbagsnprm/, December 21, 2003

[28] 2002 Annual Assessment - Motor Vehicle Traffic Crash Fatality and Injury Estimates for 2002, National Highway Traffic Safety Administration - U.S. Department of Transportation,
http://www-nrd.nhtsa.dot.gov/pdf/nrd-30/NCSA/Rpts/2003/Assess02.pdf,
December 21, 2003

[29] Government of Alberta, Alberta Traffic Collision Statistics 2000,
http://www.trans.gov.ab.ca/Content/doctype47/production/collisionstats.htm,
December 21, 2003

[30] Government of Alberta, Alberta Traffic Collision Statistics 2001,
http://www.trans.gov.ab.ca/Content/doctype47/production/collisionstats.htm
December 21, 2003

[31] Highway Safety, Airbag Statistics, July 2003,
http://www.highwaysafety.org/safety_facts/airbags/stats.htm, December 21, 2003

[32] http://www.bts.gov/publications/national_transportation_statistics/2002/html/
table_01_15.html, December 21, 2003

[33] Federal Motor Vehicle Standard 208, National Highway Traffic Safety Administration - U.S. Department of Transportation
http://www.nhtsa.dot.gov/cars/rules/import/FMVSS/#SN208, December 21, 2003

[34] Cognex Industries
http://www.cognex.com/pdf/industry/wheels.pdf, December 21, 2003

[35] www.formula1.com, December 21, 2003

APPENDIX A

Appendix A lists all the positions used for training in the development of this project.

A.1 FMVSS 208 Training Positions

3 year old dummy:

1-) Normal back against seat
2-) Back against reclined seat
3-) Back not against seat back
4-) Sitting on seat edge
5-) Standing on seat facing forward
6-) Kneeling on seat facing forward
7-) Kneeling on seat facing rearward
8-) Lying on seat.

6 year old dummy

1-) Normal back against seat
2-) Back against reclined seat
3-) Sitting on seat edge
4-) Leaning on door

5th Percentile Female Dummy

1-) Normal, Back against seat

50th Percentile Male Dummy

1-) Normal, Back against seat

Infant Seats

Britax ™ Handle With Care, Century ™ Assura, Cosco ™ Arriva, Evenflo ™ First Choice:

1-) With Crabi, No Base, No Blanket, Facing Forward, Handle Up
2-) With Crabi, No Base, No Blanket, Facing Forward, Handle Down
3-) With Crabi, No Base, No Blanket, Facing Rearward, Handle Up
4-) With Crabi, No Base, No Blanket, Facing Rearward, Handle Down
5-) With Crabi, No Base, No Blanket, Facing Rearward, Handle Up with 30 lbs seat belt Tension
6-)With Crabi, No Base, No Blanket, Facing Rearward, Handle Down with 30 lbs seat belt tension

7-)With Crabi, No Base, With Blanket, Facing Forward, Handle Up
8-)With Crabi, No Base, With Blanket, Facing Forward, Handle Down
9-)With Crabi, No Base, With Blanket, Facing Rearward, Handle Up
10-)With Crabi, No Base, With Blanket, Facing Rearward, Handle Down
11-)With Crabi, No Base, With Blanket, Facing Rearward, Handle Up with 30 lbs seat belt tension
12-)With Crabi, No Base, With Blanket, Facing Rearward, Handle Down with 30 lbs seat belt tension

Britax ™ Roadstar, Evenflo™ Right Fit:

1-) With 3yr old, Facing Forward
2-) With 3yr old, Facing Forward, with Belt Over Lap (4 lbs tension)
3-) With 6yr old, Facing Forward
4-) With 6yr old, Facing Forward, with Belt Over Lap (4 lbs tension)

Britax™ Round About, Century™ Encore, Century ™ Ste 1000, Cosco™ Olympian, Cosco™ Touriva, Evenflo™ Horizon, Evenflo™ Medallion, Fisher Price™ Safe Embrace:

1-) With Crabi, No Blanket, Facing Forward
2-) With Crabi, No Blanket, Facing Forward with 30 lbs seat belt tension
3-) With 3yr Old, No Blanket, Facing Rearward
4-) With 3yr Old, No Blanket, Facing Rearward with 30 lbs seat belt tension
5-) With Crabi, With Blanket, Facing Forward
6-) With Crabi, With Blanket, Facing Forward with 30 lbs seat belt tension
7-) With Crabi, With Blanket, Facing Rearward
8-) With Crabi, With Blanket, Facing Rearward with 30 lbs seat belt tension
9-) With Crabi, No Blanket, Facing Rearward
10-) With Crabi, No Blanket, Facing Rearward with 30 lbs seat belt tension
11-) With 3yr Old, No Blanket, Facing Forward
12-) With 3yr Old, No Blanket, Facing Forward with 30 lbs seat belt tension
13-) With 3yr Old, With Blanket, Facing Forward
14-) With 3yr Old, With Blanket, Facing Forward with 30 lbs seat belt tension
15-) With 3yr Old, With Blanket, Facing Rearward
16-) With 3yr Old, With Blanket, Facing Rearward with 30 lbs seat belt tension

Graco™ Snug Ride, Century™ Avanta, Cosco Opus, Century™ Smart Fit, Cosco™ Dream Ride, Evenflo™ Discovery, Evenflo™ On my Way:

1-) With Crabi, With Base, No Blanket, Facing Forward, Handle Up
2-) With Crabi, With Base, No blanket, Facing Forward, Handle Down
3-) With Crabi, With Base, No Blanket, Facing Rearward, Handle Up
4-) With Crabi, With Base, No blanket, Facing Rearward, Handle Down
5-) With Crabi, With Base, No Blanket, Facing Rearward, Handle Up with 30 lbs seat belt tension

6-) With Crabi, With Base, No blanket, Facing Rearward, Handle Down with 30 lbs seat belt tension
7-) With Crabi, With Base, With Blanket, Facing Forward, Handle Up
8-) With Crabi, With Base, With Blanket, Facing Forward, Handle Down
9-) With Crabi, With Base, With Blanket, Facing Rearward, Handle Up
10-) With Crabi, With Base, With Blanket, Facing Rearward, Handle Down
11-) With Crabi, With Base, With Blanket, Facing Rearward, Handle Up with 30 lbs seat belt tension
12-) With Crabi, With Base, With Blanket, Facing Rearward, Handle Down with 30 lbs seat belt tension
13-) With Crabi, No Base, With Blanket, Facing Rearward, Handle Up
14-) With Crabi, No Base, With Blanket, Facing Rearward, Handle Down
15-) With Crabi, No Base, With Blanket, Facing Rearward, Handle Up with 30 lbs seat belt tension
16-) With Crabi, No Base, With Blanket, Facing Rearward, Handle Down with 30 lbs seat belt tension
17-) With Crabi, No Base, With Blanket, Facing Forward, Handle Up
18-) With Crabi, No Base, With Blanket, Facing Forward, Handle Down
19-) With Crabi, No Base, No Blanket, Facing Rearward, Handle Up
20-) With Crabi, No Base, No Blanket, Facing Rearward, Handle Down
21-) With Crabi, No Base, No Blanket, Facing Rearward, Handle Up with 30 lbs seat belt tension
22-) With Crabi, No Base, No Blanket, Facing Rearward, Handle Down with 30 lbs seat belt tension
23-) With Crabi, No Base, No Blanket, Facing Forward, Handle Up
24-) With Crabi, No Base, No Blanket, Facing Forward, Handle Down

Century™ Next Step, Cosco™ High Back:

1-) With 3yr old, Facing Forward
2-) With 3yr old, Facing Forward with Belt Over Lap (4 lbs)
3-) With 3yr old, Facing Forward with 30 lbs seat belt tension
4-) With 6yr old, Facing Forward
5-) With 6yr old, Facing Forward with Belt Over Lap (4 lbs)
6-) With 6yr old, Facing Forward with 30 lbs seat belt tension

A.2 Car Manufacturer A due Care Training Positions

3 year old Dummy

1-) Normal, Back Against Seat
2-) Back Against Reclined Seat
3-) Back Not Against Seat Back
4-) Sitting on Seat Edge
5-) Lying across on Seat
6-) Kneeling on Seat Facing Forward

7-) Kneeling on Seat Facing Rearward
8-) Standing on Seat Facing Forward

6 year old Dummy

1-) Normal, Back Against Seat with 4 lbs seat belt tension
2-) Back Against Reclined Seat, 25 degrees
3-) Sitting on Seat Edge
4-) Rotate 30 degrees Clockwise with 4 lbs seat belt tension
5-) Rotate 30 degrees Counter Clockwise with 4 lbs seat belt tension
6-) Facing Inboard with 4 lbs seat belt tension
7-) Facing Outboard with 4 lbs seat belt tension
8-) Leaning on Door with 4 lbs seat belt tension
9-) Sitting on 1" Blanket with 4 lbs seat belt tension

5th Percentile Female Dummy

1-) Normal with Back Against Seat
2-) Normal with Back Against Seat with 4 lbs seat belt tension
3-) Rotate 30 degrees Clockwise with 4 lbs seat belt tension
4-) Rotate 30 degrees Counter Clockwise with 4 lbs seat belt tension
5-) Facing Inboard with 4 lbs seat belt tension
6-) Facing Outboard with 4 lbs seat belt tension
7-) Normal with Legs In with 4 lbs seat belt tension
8-) Sitting on 1" Blanket with 4 lbs seat belt tension

50th Percentile Male Dummy

1-) Normal with Back against Seat
2-) Normal with Back against Seat with 4 lbs seat belt tension
3-) Rotate 30 deg Clockwise with 4 lbs seat belt tension
4-) Rotate 30 deg Counter Clockwise with 4 lbs seat belt tension
5-) Facing Inboard with 4 lbs seat belt tension
6-) Facing Outboard with 4 lbs seat belt tension
7-) Normal with Legs In with 4 lbs seat belt tension
8-) Sitting on 1" Blanket with 4 lbs seat belt tension

Infant Seats

Britax™ Handle With Care, Century™ Assura, Cosco™ Arriva, Evenflo™ First Choice:

1-) With Crabi, No base, No Blanket, With 4" Foam Roll, Facing Rearward with 10 lbs seat belt tension
2-) With Crabi, No base, No Blanket, With 4" Foam Roll, Facing Rearward with 20 lbs seat belt tension

3-) With Crabi, No base, No Blanket, With 4" Foam Roll, Facing Rearward with 32 lbs seat belt tension
4-) With Crabi, With base, No Blanket, With 4" Foam Roll, Facing Rearward with 10 lbs seat belt tension
5-) With Crabi, With base, No Blanket, With 4" Foam Roll, Facing Rearward with 20 lbs seat belt tension
6-) With Crabi, With base, No Blanket, With 4" Foam Roll, Facing Rearward with 32 lbs seat belt tension
7-) With Crabi, With base, With Blanket, Without Foam Roll, Facing Rearward with 10 lbs seat belt tension
8-) With Crabi, With base, With Blanket, Without Foam Roll, Facing Rearward with 20 lbs seat belt tension
9-) With Crabi, With base, With Blanket, Without Foam Roll, Facing Rearward with 32 lbs seat belt tension
10-) With Crabi, No base, With Blanket, Without Foam Roll, Facing Rearward with 10 lbs seat belt tension
11-) With Crabi, No base, With Blanket, Without Foam Roll, Facing Rearward with 20 lbs seat belt tension
12-) With Crabi, No base, With Blanket, Without Foam Roll, Facing Rearward with 32 lbs seat belt tension

Britax™ Roadstar, Evenflo™ Right Fit:

1-) With 3yr Old, With Blanket and Lap Belt (4lbs)
2-) With 6yr Old, With Blanket, Facing Forward with Lap Belt (4lbs)

Britax™ Round About, Century™ Encore, Century™ Ste 1000, Cosco™ Olympian, Cosco™ Touriva, Evenflo™ Horizon, Evenflo™ Medallion, Fisher Price™ Safe Embrace:

1-) With Crabi, with Blanket, Facing Rearward with 10 lbs seat belt tension
2-) With Crabi, with Blanket, Facing Rearward with 20 lbs seat belt tension
3-) With Crabi, with Blanket, Facing Rearward with 32 lbs seat belt tension
4-) With 3yr old, with Blanket, Facing Forward with 10 lbs seat belt tension
5-) With 3yr old, with Blanket, Facing Forward with 20 lbs seat belt tension
6-) With 3yr old, with Blanket, Facing Forward with 32 lbs seat belt tension

Graco Snug Ride, Century Avanta, Cosco Opus, Century Smart Fit, Cosco Dream Ride Evenflo Discovery, Evenflo On my Way:

1-) With Crabi, No base, No Blanket, With 4" Foam Roll, Facing Rearward with 10 lbs seat belt tension
2-) With Crabi, No base, No Blanket, With 4" Foam Roll, Facing Rearward with 20 lbs seat belt tension
3-) With Crabi, No base, No Blanket, With 4" Foam Roll, Facing Rearward with 32 lbs seat belt tension

4-) With Crabi, No base, With Blanket, Without Foam Roll, Facing Rearward with 10 lbs seat belt tension
5-) With Crabi, No base, With Blanket, Without Foam Roll, Facing Rearward with 20 lbs seat belt tension
6-) With Crabi, No base, With Blanket, Without Foam Roll, Facing Rearward with 32 lbs seat belt tension

Century™ Next Step, Cosco™ High Back:

1-) With Crabi, with Blanket, Facing Rearward with 10 lbs seat belt tension
2-) With Crabi, with Blanket, Facing Rearward with 20 lbs seat belt tension
3-) With Crabi, with Blanket, Facing Rearward with 32 lbs seat belt tension
4-) With 3yr old, with Blanket, Facing Forward with 10 lbs seat belt tension
5-) With 3yr old, with Blanket, Facing Forward with 20 lbs seat belt tension
6-) With 3yr old, with Blanket, Facing Forward with 32 lbs seat belt tension

A.3 Car Manufacturer 2 due Care Testing Positions

3 year old Dummy

1-) Normal, Back Against Seat
2-) Sitting on Seat Edge
3-) Leaning on Door
4-) 25 degrees reclined back seat
5-) Sitting back not against seat back
6-) Kneeling on Seat Facing Forward
7-) Kneeling on Seat Facing Rearward
8-) Standing Facing Forward
9-) Lying on Seat Facing Forward
10-) Kneeling Outboard facing Rearward
11-) Standing Facing Rearward
12-) Standing Outboard Facing Rearward
13-) Lying on seat Face up and Head out
14-) Lying on seat Head on Door Armrest

6 year old Dummy

1-) Normally seated
2-) Sitting on Seat Edge
3-) Leaning on Door
4-) 25 degrees reclined Back Seat
5-) Ankles Crossed and Knees Apart
6-) Left Foot Under with Shoes on
7-) Left Foot Under with Shoes off
8-) Pretzel with Shoes Off
9-) Holding Knees with Shoes Off

10-) Relaxed (slouch) with Head on Seat Back and Shoes off
11-) Relaxed (slouch) with Head on Seat Back and Shoes on
12-) Leaning on Center Console with Left ankle under thigh, chin on Head and shoes on
13-) Leaning on Center Console with Left ankle under thigh, chin on Head and shoes off
14-) Leaning back towards Window with left ankle under thigh, hands clasped and shoes off
15-) Leaning back towards Window with left ankle under thigh, hands clasped and shoes on
16-) Kneeling Outboard facing Rearward
17-) Standing Facing Rearward
18-) Standing outboard Facing Rearward
19-) Lying on Seat with Face Up and Head out
20-) Lying on seat with Head on Armrest
21-) Seating with Bookbag on lap

5th Percentile Female Dummy

1-) Normal Back against seat
2-) Legs Rearward
3-) Ankles Crossed and Knees Apart
4-) Left Foot Under with Shoes on
5-) Left Foot Under with Shoes off
6-) Pretzel with Shoes Off
7-) Holding Knees with Shoes Off
8-) Relaxed (slouch) with Head on Seat Back and Shoes off
9-) Relaxed (slouch) with Head on Seat Back and Shoes on
10-) Leaning on Center Console with Right leg over and Hands clasped
11-) Leaning on Center Console with Left leg over,Chin on Hand and Shoes on
12-) Leaning on Center Console with Left leg over,Chin on Hand and Shoes off
13-) Leaning on Center Console with Left ankle under thigh, chin on Head and shoes off
14-) Leaning on Center Console with Left ankle under thigh, chin on Head and shoes on
15-) Leaning on Center Console with Left ankle over thigh,Hands clasped and shoes off
16-) Leaning on Center Console with Left ankle over thigh, Hands clasped and shoes on
17-) Leaning back towards Window with left ankle under thigh, hands clasped and shoes off
18-) Leaning back towards Window with left ankle under thigh, hands clasped and shoes on
19-) Leaning back towards window with left ankle under thigh, hands clasped and shoes on
20-) Leaning back towards window with right leg over and hands clasped
21-) Touching IP
22-) Adjusting Radio
23-) Tie Shoes
24-) Holding Grab Handle

Appendix B

Appendix B gives a brief description of the development and use of the data acquisition software

Platform

The two most frequently developing platforms used are MS Windows and Unix. MS Windows has friendlier graphical user interface and is more popular for end users. Unix is more robust for large business servers and complex processes but has a less friendly graphical user interface. Because this tool was to be designed for ordinary users, the MS Windows 2000 platform was chosen.

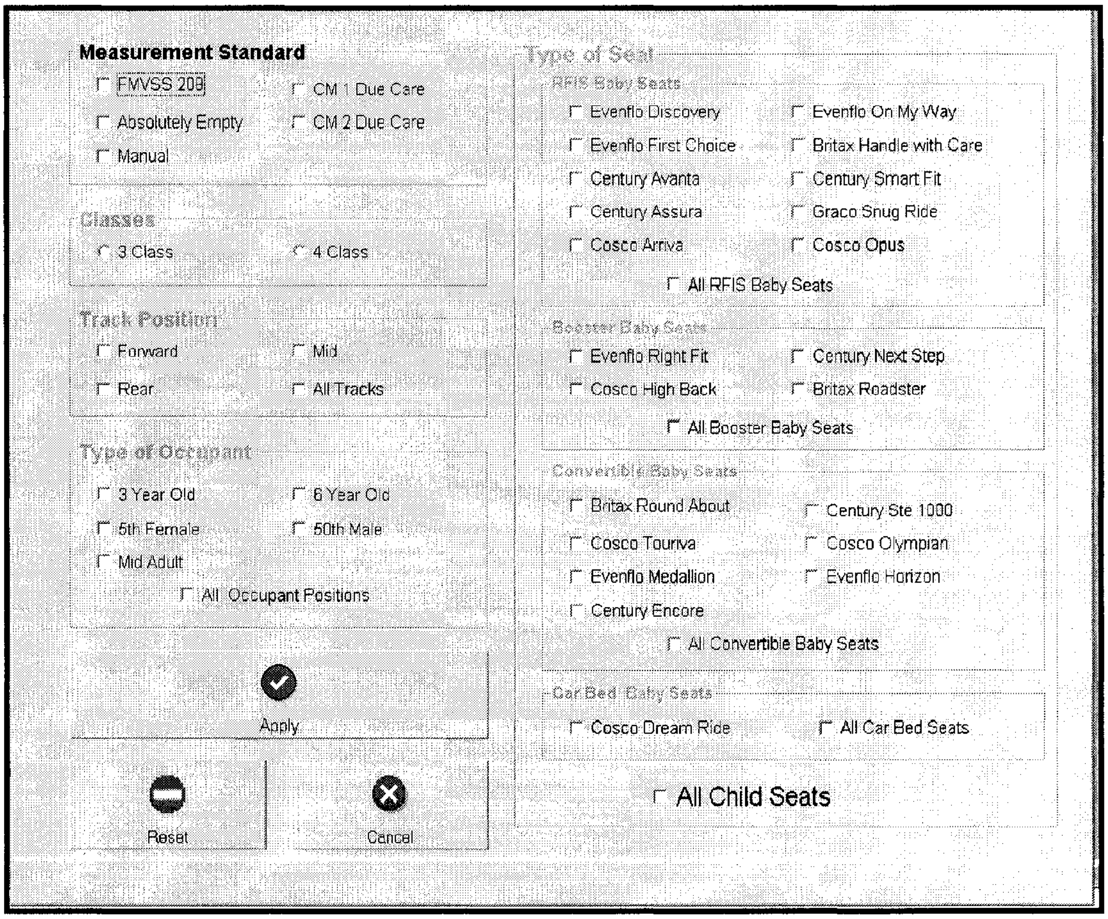
Programming Language

For the development of friendly graphical interfaces several programming languages could be used. The most popular ones are Visual Basic, C++ and Java. They all support GUI programming in a MS Windows platform. Of these, Visual Basic presents the fastest development time. It is like component glue. It can integrate the forms, menus dialog boxes together, making the design easier. The downside of this high level design language is the overhead produced in computations. However, the acquisition software did not require complex algorithms to run, so the performance time of Visual Basic was fine for this piece of software

The actual implementation of the acquisition software was divided in two: a verification parameters module and the data acquisition module.

Verification Parameters Module

This is the first window that pops up when the acquisition software starts. The main objective of it is to let the user choose the parameters for the data acquisition session. The user has to specify the set of positions of a measurement standard (either FMVSS or one of the car manufacturers due care), the track position of the car seat during the session, and the type of occupant in the seat. It is possible to choose more than one measurement standard, track position or occupant for the same session. A screenshot of this first module is shown in figure B.1



Measurement Standard

□ FMVSS 208 □ CM 1 Due Care
□ Absolutely Empty □ CM 2 Due Care
□ Manual

Classes

○ 3 Class ○ 4 Class

Track Position

□ Forward □ Mid
□ Rear □ All Tracks

Type of Occupant

□ 3 Year Old □ 6 Year Old
□ 5th Female □ 50th Male
□ Mid Adult
□ All Occupant Positions

Type of Seat

RFIS Baby Seats

□ Evenflo Discovery □ Evenflo On My Way
□ Evenflo First Choice □ Britax Handle with Care
□ Century Avanta □ Century Smart Fit
□ Century Assura □ Graco Snug Ride
□ Cosco Arriva □ Cosco Opus
□ All RFIS Baby Seats

Booster Baby Seats

□ Evenflo Right Fit □ Century Next Step
□ Cosco High Back □ Britax Roadster
□ All Booster Baby Seats

Convertible Baby Seats

□ Britax Round About □ Century Ste 1000
□ Cosco Touriva □ Cosco Olympian
□ Evenflo Medallion □ Evenflo Horizon
□ Century Encore
□ All Convertible Baby Seats

Car Bed Baby Seats

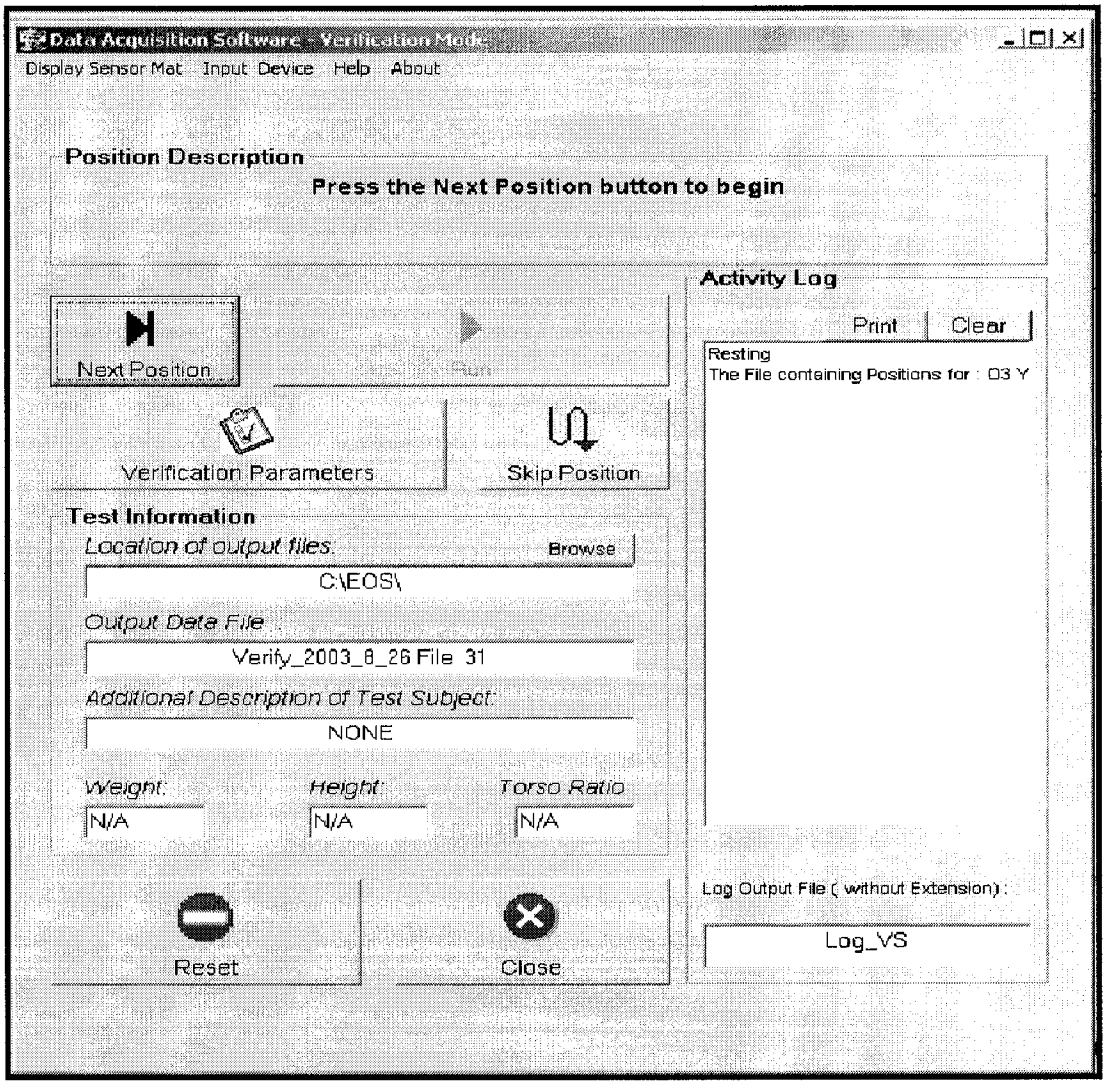
□ Cosco Dream Ride □ All Car Bed Seats

□ All Child Seats

Figure B.1. Verification Parameters Screenshot

Data Acquisition Module

Once the user has chosen the parameters for the data acquisition session the data acquisition window pops up. This module reads the data from the sensors and stores it to file. In this module the user can specify the name and location of the output files, an additional description of the occupant of the seat and weight, height and torso ratio measurements if available. The description of the position for which data is going to be gathered is displayed in the position description frame. Once the Play button is press the data from that position is stored to file and the next position is displayed. The user can end the session at any minute by pressing the reset button. The close button will close the whole application. A screenshot of this module is shown in figure B.2.



Position Description

Press the Next Position button to begin

Activity Log

Resting
The File containing Positions for : 03 Y

Test Information

Location of output files: C:\EOS\

Output Data File : Verify_2003_8_26 File 31

Additional Description of Test Subject: NONE

Weight: N/A

Height: N/A

Torso Ratio: N/A

Log Output File (without Extension) : Log_VS

Figure B.2. Data Acquisition Software

Labelling of the data is done automatically by the acquisition software. The output voltages and a brief description of the occupant and position are stored to file. Based on this information the data will later be assigned a group class label in order to use the computational intelligence techniques to develop the pattern recognizer.

APPENDIX C

Appendix C lists the results of multiple sensor failure

C.1 Two Sensor Failure

The table below shows which pair of sensors was unavailable simultaneously and the success percentage achieved by the NN with those sensors off.

Sensor1	Sensor2	Success Percentage (%)
14	15	83.67
13	14	83.69
11	14	84.28
15	16	85.67
10	15	86.26
14	16	86.46
7	15	86.47
13	15	87.18
7	14	87.37
12	15	87.67
9	15	87.75
9	14	87.96
13	16	88.01
3	15	88.16
2	15	88.26
6	14	88.34
9	13	88.52
2	14	88.6
10	14	88.88
4	15	88.98
3	14	90.09
6	15	90.18
8	15	90.19
4	14	90.4
5	15	90.56
1	15	90.56
5	14	90.89
8	14	91.08
7	13	91.37
2	13	91.52
1	14	91.75
11	15	91.93
12	14	92.02
9	10	92.42
10	11	92.64
9	16	92.71
3	13	92.73
4	13	93.01
7	16	93.47
11	13	93.58
12	13	93.6
5	13	93.62
6	13	93.86

Sensor1	Sensor2	Success Percentage (%)
2	16	94.03
8	13	94.15
10	13	94.16
1	13	94.49
2	7	94.54
4	16	94.82
9	11	94.89
7	9	95
3	16	95.35
6	7	95.37
11	16	95.42
6	16	95.45
4	7	95.48
2	3	95.64
2	9	95.8
10	16	95.83
7	11	95.99
3	7	96.02
8	16	96.11
7	10	96.15
5	16	96.27
2	5	96.31
4	9	96.31
7	8	96.35
5	7	96.56
2	6	96.58
4	11	96.58
2	11	96.58
6	11	96.59
2	4	96.75
4	10	96.84
1	7	96.89
2	8	96.9
7	12	96.93
1	16	97
2	10	97.05
3	11	97.06
9	12	97.12
12	16	97.12
4	6	97.14
11	12	97.26
3	6	97.31
6	10	97.45
1	2	97.55
5	11	97.57
3	4	97.6
3	9	97.61
6	9	97.64
8	11	97.66
3	10	97.67
4	8	97.74
5	9	97.77
1	11	97.9
3	8	97.92
3	5	97.97

Sensor1	Sensor2	Success Percentage (%)
8	9	98.03
5	6	98.04
6	8	98.06
1	6	98.07
4	5	98.12
1	9	98.19
2	12	98.2
10	12	98.24
1	4	98.26
4	12	98.26
5	8	98.31
1	3	98.37
5	10	98.53
6	12	98.54
8	10	98.55
3	12	98.61
1	10	98.71
1	5	99.04
1	8	99.09
5	12	99.25
8	12	99.37
1	12	99.51

Table C.1. Success rate after two sensors fail simultaneously

C.2 Three Sensor Failure

The table below shows which triplet of sensor was unavailable simultaneously, and the success percentage achieved by the NN with those sensors off.

Sensor1	Sensor2	Sensor3	Success Percentage (%)
13	14	15	75.86
9	10	11	75.94
13	14	16	76.23
10	11	14	77.13
14	15	16	77.97
11	13	14	78.18
13	15	16	78.33
9	13	16	78.6
7	14	15	78.75
9	13	14	79.49
2	14	15	79.78
2	13	14	79.85
11	14	15	79.94
9	11	14	80.07
11	14	16	80.17
9	10	15	80.22
10	11	15	80.58
3	13	14	80.6
10	12	15	80.66
10	15	16	80.66
2	14	16	80.75
7	11	14	80.83

3	14	15	80.83
7	13	14	80.88
10	14	15	80.88
7	15	16	80.91
2	15	16	81.02
2	11	14	81.18
2	13	16	81.28
3	15	16	81.32
7	10	15	81.46
4	13	14	81.65
6	14	15	81.68
4	14	15	81.8
6	11	14	81.81
7	14	16	82.03
6	13	14	82.04
9	14	15	82.2
5	13	14	82.34
5	14	15	82.39
3	11	14	82.44
7	13	16	82.55
9	15	16	82.56
8	14	15	82.61
6	14	16	82.63
3	14	16	82.65
4	15	16	82.65
2	7	15	82.69
2	9	13	82.7
8	13	14	82.78
4	14	16	82.78
7	9	13	82.93
9	14	16	82.97
7	13	15	83
12	14	15	83.05
10	13	14	83.1
4	11	14	83.11
12	15	16	83.16
2	7	14	83.29
9	10	13	83.31
1	14	15	83.33
10	14	16	83.38
9	13	15	83.46
5	11	14	83.64
12	13	14	83.66
1	13	14	83.68
8	11	14	83.71
7	9	15	83.72
7	12	15	83.72
3	7	15	83.81
4	7	15	83.81
2	13	15	83.84
6	15	16	83.84
10	13	15	83.86
8	15	16	83.87
11	12	14	83.92
3	13	15	83.92
4	13	16	83.95

3	10	15	83.98
4	10	15	83.99
9	10	14	84.05
2	3	15	84.15
3	13	16	84.15
2	10	15	84.28
8	14	16	84.32
6	7	15	84.33
6	10	14	84.36
2	6	14	84.37
1	11	14	84.39
5	14	16	84.46
5	15	16	84.51
2	9	14	84.54
9	12	15	84.54
6	7	14	84.62
3	9	15	84.62
2	3	14	84.66
6	9	14	84.82
12	13	15	84.84
3	7	14	84.85
2	9	15	84.85
7	9	14	84.86
3	12	15	84.86
6	10	15	84.94
4	9	13	85.06
4	9	15	85.22
7	8	15	85.27
7	10	14	85.3
4	7	14	85.32
2	5	14	85.33
1	15	16	85.35
5	13	16	85.4
8	13	16	85.4
2	12	15	85.51
3	9	13	85.77
5	7	14	85.79
3	9	14	85.82
2	10	14	85.82
8	10	15	85.83
9	12	13	85.86
2	7	13	85.93
4	12	15	85.93
5	7	15	85.95
6	13	16	85.95
1	14	16	85.95
8	13	15	86.05
9	11	13	86.06
7	8	14	86.09
4	13	15	86.09
1	7	15	86.1
2	8	14	86.16
4	6	14	86.18
5	13	15	86.2
12	13	16	86.22
3	6	14	86.24

11	15	16	86.24
2	8	15	86.25
10	13	16	86.27
2	4	15	86.33
1	10	15	86.4
2	4	14	86.45
5	10	15	86.52
4	9	14	86.56
3	4	15	86.61
2	5	15	86.63
6	9	15	86.76
6	13	15	86.87
1	7	14	86.94
9	10	16	86.95
5	9	13	87
3	8	15	87
1	13	15	87.01
2	3	13	87.03
8	9	15	87.09
12	14	16	87.09
2	6	15	87.15
5	9	15	87.17
8	12	15	87.18
7	11	15	87.25
11	13	16	87.25
5	9	14	87.26
1	13	16	87.29
1	9	15	87.46
3	6	15	87.47
7	12	14	87.49
1	12	15	87.5
6	8	14	87.52
8	9	14	87.52
3	10	14	87.52
5	6	14	87.53
1	2	14	87.54
1	2	15	87.54
3	5	15	87.57
8	9	13	87.62
6	12	15	87.64
4	10	14	87.69
9	11	16	87.7
1	3	15	87.71
1	9	13	87.77
5	12	15	87.77
6	9	13	87.91
4	8	15	87.95
1	9	14	88.08
5	10	14	88.09
2	5	13	88.14
2	9	16	88.14
3	4	14	88.27
1	6	14	88.37
8	10	14	88.4
4	6	15	88.47
4	9	10	88.48

4	7	13	88.7
10	11	12	88.72
3	5	14	88.72
3	8	14	88.72
6	12	14	88.72
1	4	15	88.8
9	12	14	88.81
4	9	16	88.82
2	12	14	88.83
3	11	15	88.88
2	8	13	88.93
2	4	13	88.96
10	12	14	88.97
2	7	16	89
11	12	15	89.01
3	7	13	89.02
7	9	16	89.02
6	8	15	89.06
4	8	14	89.07
5	8	15	89.1
2	3	16	89.11
10	11	16	89.11
5	8	14	89.17
4	10	11	89.18
1	10	14	89.18
4	5	15	89.18
6	7	13	89.4
7	12	13	89.4
7	8	13	89.59
4	5	14	89.62
1	6	15	89.62
5	7	13	89.63
1	8	15	89.74
2	11	15	89.77
7	10	13	89.78
2	6	13	89.8
1	3	14	89.9
5	6	15	89.97
1	5	15	90.05
9	11	15	90.16
2	12	13	90.2
1	2	13	90.21
2	4	16	90.24
7	10	11	90.27
2	11	13	90.29
7	11	13	90.3
3	12	14	90.34
4	7	16	90.37
9	10	12	90.39
1	4	14	90.4
2	5	16	90.42
4	11	15	90.51
2	10	13	90.54
9	11	12	90.56
2	9	10	90.57
1	5	14	90.58

1	7	13	90.6
6	7	16	90.65
11	13	15	90.79
2	7	9	90.86
1	8	14	90.98
3	5	13	91
2	10	11	91.05
4	12	14	91.08
3	6	13	91.11
3	8	13	91.11
2	8	16	91.15
5	9	16	91.15
5	12	14	91.17
3	9	16	91.17
6	11	15	91.23
7	9	10	91.25
2	3	7	91.28
3	4	13	91.31
8	9	16	91.32
7	8	16	91.36
8	11	15	91.38
2	6	16	91.41
5	9	10	91.44
4	8	13	91.45
8	12	14	91.47
3	12	13	91.5
5	7	16	91.53
5	8	13	91.57
4	12	13	91.61
3	7	16	91.61
2	9	11	91.64
4	6	13	91.64
10	11	13	91.65
4	5	13	91.67
9	12	16	91.69
2	5	7	91.83
4	10	13	91.83
4	11	13	91.85
1	11	15	91.91
3	11	13	91.94
7	11	16	91.94
7	9	11	91.99
6	10	11	92.01
4	8	16	92.02
1	3	13	92.03
1	12	14	92.06
2	11	16	92.07
8	9	10	92.09
1	9	16	92.13
3	10	11	92.14
5	11	13	92.15
2	7	8	92.18
5	11	15	92.18
4	9	11	92.22
6	9	16	92.22
11	12	13	92.24

2	6	7	92.26
6	11	13	92.34
4	7	9	92.35
5	12	13	92.37
6	9	10	92.38
5	6	13	92.39
7	10	16	92.4
3	4	16	92.47
2	4	7	92.48
6	8	13	92.49
4	10	16	92.49
4	11	16	92.5
10	12	13	92.55
1	4	13	92.58
8	10	11	92.6
2	10	16	92.6
4	6	16	92.62
1	9	10	92.73
1	2	16	92.75
8	12	13	92.77
1	5	13	92.81
3	10	13	92.83
1	7	16	92.84
3	6	16	92.86
8	11	13	92.87
2	3	5	92.88
6	12	13	92.89
2	4	9	92.9
5	10	13	92.96
6	10	13	92.97
5	10	11	92.99
3	5	16	93.03
1	12	13	93.06
1	10	11	93.1
4	5	16	93.12
7	12	16	93.15
1	11	13	93.22
2	7	11	93.24
1	6	13	93.25
1	2	7	93.33
3	9	10	93.34
1	8	13	93.34
2	3	6	93.36
2	5	9	93.38
4	6	7	93.39
8	10	13	93.39
2	5	8	93.41
2	3	8	93.42
3	8	16	93.45
6	11	16	93.45
2	3	4	93.55
6	7	9	93.56
2	7	10	93.58
5	8	16	93.58
3	11	16	93.64
2	12	16	93.73

1	10	13	93.74
3	6	7	93.76
7	9	12	93.76
6	10	16	93.77
3	7	9	93.82
4	7	10	93.82
6	7	11	93.85
6	8	16	93.85
4	7	11	93.89
5	9	11	93.89
2	3	9	93.9
7	8	9	93.93
4	7	8	93.98
5	7	9	94.01
2	4	5	94.04
5	6	16	94.04
6	7	8	94.12
6	7	10	94.15
3	10	16	94.15
2	4	8	94.21
2	8	9	94.22
2	7	12	94.23
2	5	6	94.24
3	4	7	94.25
8	11	16	94.27
2	9	12	94.35
1	4	16	94.35
4	12	16	94.35
11	12	16	94.37
2	3	11	94.4
1	7	9	94.43
5	11	16	94.47
8	9	11	94.48
2	6	11	94.5
2	5	11	94.51
2	4	6	94.52
5	6	7	94.52
1	2	5	94.53
3	9	11	94.54
3	7	8	94.61
4	9	12	94.61
6	9	11	94.62
2	4	11	94.7
5	7	8	94.73
8	10	16	94.73
2	6	8	94.75
1	2	3	94.77
1	2	9	94.77
1	3	16	94.78
5	10	16	94.8
3	5	7	94.83
2	3	10	94.89
1	6	7	94.9
1	9	11	94.9
2	4	10	94.91
4	5	7	94.93

1	6	16	94.93
3	7	10	94.96
4	6	11	94.96
3	7	11	94.99
2	6	9	95.02
2	6	10	95.06
3	12	16	95.07
4	7	12	95.13
10	12	16	95.13
7	8	11	95.18
2	5	10	95.19
4	6	10	95.22
6	7	12	95.23
7	11	12	95.23
1	4	7	95.25
7	8	10	95.27
2	8	11	95.28
1	5	16	95.3
5	7	11	95.33
4	8	9	95.36
1	8	16	95.36
1	11	16	95.39
3	6	11	95.41
1	3	7	95.45
4	5	9	95.47
3	4	11	95.5
5	7	10	95.59
6	12	16	95.59
2	3	12	95.6
3	7	12	95.61
4	11	12	95.63
1	2	6	95.65
7	10	12	95.65
1	7	8	95.66
4	6	9	95.68
2	8	10	95.71
2	11	12	95.71
1	10	16	95.73
1	5	7	95.75
1	2	8	95.79
8	12	16	95.79
4	8	11	95.8
1	7	11	95.84
3	4	9	95.88
3	4	10	95.9
4	10	12	95.93
1	7	10	95.94
1	4	9	95.95
5	12	16	95.95
3	4	6	95.97
3	4	8	95.98
1	2	4	95.99
4	6	8	96
4	5	8	96.01
1	2	11	96.04
4	8	10	96.06

6	8	11	96.08
7	8	12	96.14
3	5	8	96.15
3	11	12	96.18
6	11	12	96.21
5	6	11	96.22
2	5	12	96.22
3	9	12	96.28
5	9	12	96.28
4	5	11	96.3
3	6	8	96.32
1	6	11	96.35
3	8	11	96.38
2	4	12	96.38
5	7	12	96.4
3	5	6	96.44
4	5	10	96.46
1	4	11	96.47
2	10	12	96.49
1	2	10	96.52
3	6	10	96.52
5	8	11	96.56
5	6	8	96.58
2	6	12	96.58
1	7	12	96.59
3	5	11	96.65
1	4	10	96.66
3	4	5	96.68
5	8	9	96.68
8	9	12	96.69
4	5	6	96.7
1	4	6	96.72
1	12	16	96.73
1	3	6	96.75
3	6	9	96.77
1	9	12	96.77
2	8	12	96.79
3	8	9	96.8
3	5	9	96.82
5	11	12	96.93
6	9	12	96.97
4	6	12	97.01
6	8	10	97.05
3	8	10	97.07
1	3	11	97.08
1	6	10	97.09
3	5	10	97.11
5	6	9	97.12
8	11	12	97.12
5	6	10	97.14
6	8	9	97.17
6	10	12	97.19
1	3	9	97.21
3	4	12	97.25
1	6	9	97.26
1	5	9	97.28

1	4	8	97.29
1	3	8	97.31
1	6	8	97.33
1	5	6	97.34
1	5	11	97.34
3	6	12	97.34
1	3	4	97.4
3	10	12	97.41
1	11	12	97.42
1	3	5	97.44
1	5	8	97.44
1	4	5	97.45
1	2	12	97.47
1	8	9	97.55
5	8	10	97.55
1	8	11	97.55
4	8	12	97.55
1	3	10	97.58
3	5	12	97.72
3	8	12	97.83
4	5	12	97.85
1	4	12	98.03
5	10	12	98.03
8	10	12	98.06
1	5	10	98.09
6	8	12	98.12
1	6	12	98.14
5	6	12	98.19
1	8	10	98.25
1	3	12	98.26
1	10	12	98.26
5	8	12	98.27
1	5	12	98.96
1	8	12	99.01

Table C.2. Success rate after three sensors fail simultaneously