Title

Development and implementation of an integrated framework for structural health monitoring

Author

Fraser, Michael Swain

Publication Date

2006

UNIVERSITY OF CALIFORNIA, SAN DIEGO

Development and Implementation of an Integrated Framework for Structural Health Monitoring

A dissertation submitted in partial satisfaction of the
requirements for the degree of Doctor of Philosophy

in

Structural Engineering

by

Michael Swain Fraser

Committee in charge:

Professor Ahmed Elgamal, Chair
Professor Joel P. Conte
Professor Francesco Lanza di Scalea
Professor Michael Todd
Professor Mohan Trivedi
Professor Ramesh Rao

2006

The dissertation of Michael Swain Fraser is approved, and it is acceptable in quality and form for publication on microfilm:

Chair

University of California, San Diego

2006

DEDICATION

To my parents for all their support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Support of this research was provided by the U.S. National Science Foundation, under ITR Grant No.0205720. This support is gratefully acknowledged. The author also wishes to thank the California Institute for Telecommunications and Information Technology (UCSD Cal-(IT)2, http://www.calit2.net) and its Director Professor Ramesh Rao for providing fellowship support to this project.

I am indebted to Professor Ahmed Elgamal for his years of guidance in this endeavor. The efforts provided by Professor Joel P. Conte in the development of the integrated framework are gratefully acknowledged. My thanks to Dr. Tony Fountain and Neil Cotofana for their aid in the development of the web portal, database systems, and data loader programs. The advice and assistance of Professor Trivedi, Tarak Gandhi, and Jeff Ploetner on cameras and image acquisition for the Voigt Bridge testbed is appreciated. To Professors Vistasp Karbhari and Frieder Seible, my gratitude for providing access to the composite bridge deck panels.

I wish to acknowledge Mr. Minh Phan for all his help over the years, in particular for his seminal contributions towards development of the web portal and database systems. Also, my thanks go out to Mr. Dung Nguyen for his work with the data loader programs. I would also like to thank James Linjun Yan for all his guidance with the neural networks and for his assistance with OpenSees. To Miss Kendra Oliver a special thanks for all of her work in helping to create the labeled

datasets and more importantly for her support during the last couple of years when things got bumpy. Finally, to Dmitry Yudovsky, Eric Tsang, Albert Sanchez, and Catherine Pagni my thanks for the fun we had while locked down in the lab all these years.

VITA

1999	B.S., University of California, San Diego. Magna Cum Laude
2003	M.S., University of California, San Diego
2003	California Institute for Telecommunications and Information Technology (CAL-IT^2) Graduate Fellow
2004	California Institute for Telecommunications and Information Technology (CAL-IT^2) Graduate Fellow
2006	Ph.D., University of California, San Diego

PUBLICATIONS

Zonta, D., Elgamal, A., Fraser, M., and Priestley, M. J. N. (2005). “Analysis of Change in Dynamic Properties of a Frame-Resistant Test Building.” *Journal of Engineering Structures* (in review).

Elgamal, A., Fraser, M., McMartin, F. (2005). “On-line Educational Shake Table Experiments,” *Journal of Professional Issues in Engineering Education & Practice*, ASCE.

Elgamal, A., Fraser, M., and Zonta, D. (2005). “Webshaker: Live Internet Shake-Table Experiment for Education and Research” *Computer Applications in Engineering Education*.

Yan, L., Fraser, M., Oliver K., Elgamal A., Conte J. P., and Fountain T. (2005). “Traffic Pattern Recognition using an Active Learning Neural Network and Principal Component Analysis.” *Proc. of the 8th International Conference on the Application of Artificial Intelligence to Civil, Structural and Environmental Engineering*, Rome, Italy, August 30 – September 2.

Fraser, M. and Luco, J. E. (2005). “Ambient Vibration Study of Tissue Culture Incubators.” *Report prepared for the Genomics Institute of the Novartis Research Foundation.*

Yan, L., Fraser, M., Elgamal, A., and Conte, J.P. (2004). "Applications of Neural Networks in Structural Health Monitoring." *Proc. of the 3rd China-Japan-US Symposium on Structural Health Monitoring and Control*, Dalian, China, October 13-16.

Elgamal, A., Conte, J.P., Yan, L., Fraser, M., Masri, S.F., El Zarki, M., Fountain, T., and Trivedi, M. (2004). "A Framework for Monitoring Bridges and Civil Infrastructure." (Invited Paper) *Proc. of the 3rd China-Japan-US Symposium on Structural Health Monitoring and Control*, Dalian, China, October 13-16.

Fraser,M., Yan, L., He, X., Elgamal, A., Conte, J.P., and Fountain, T. (2004). "Simple Neural Network Application for Traffic Monitoring," *Proc. of the International Conference on Computaional & Experimental Engineering and Sciences (ICCES'04)*, Madeira, Portugal, July 26-29.

Fraser, M., Elgamal, A,. Oliver, K., and Conte, J.P. (2004). “Data Fusion Application For Health Monitoring,” *Proc. of the 1st International Workshop on Advanced Smart Materials and Smart Structures Technology*, Honolulu, HI, January 12-14.

Elgamal, A., Conte, J.P., Fraser, M., Masri, S., Fountain, T., Gupta, A., Trivedi, M., and El Zarki, M. (2003). “HEALTH MONITORING FOR CIVIL INFRASTRUCTURE,” *Proc. of the 9th Arab Structural Engineering Conference, (9ASEC)*, Abu Dhabi, UAE, Nov. 29-Dec. 1.

Elgamal, A., Conte, J.P., Masri, S., Fraser, M., Fountain, T., Gupta, A., Trivedi, M., and El Zarki, M. (2003). “Health Monitoring Framework for Bridges and Civil Infrastructure,” *Proc. of the 4th International Workshop on Structural Health Monitoring*, Stanford, CA, September 15-17.

Conte, J.P., Elgamal, A., Masri, S., Fraser, M., Fountain, T., Gupta, A., Trivedi, M., and El Zarki, M. (2003). “Health Monitoring Framework and Structural Analyses,” *UCSD and Caltrans Workshop on Structural Health Monitoring and Diagnostics of Bridge Infrastructures*, March 7.

Fraser, M., Elgamal, A., Conte, J.P., Masri, S., Fountain, T., Gupta, A., Trivedi, M., and El Zarki, M. (2003). “Elements of an Integrated Health Monitoring Framework,” *Proc. of the 2003 SPIE Conference on Nondestructive Evaluation for Health Monitoring and Diagnostics*, San Diego, CA, March 2-6.

Fraser, M., Elgamal, A., Conte, J.P., Karbhari, V., Zhao, L., and Seible, F. (2002) “PILOT HEALTH MONITORING FRAMEWORK FOR COMPOSITE BRIDGE-DECK PANELS,” *Proc. of the Third Middle East Symposium on Structural Composites for Infrastructure Applications (MESC-3),* Aswan, Egypt, Dec 17-20.

Zonta, D., Elgamal, A., Fraser, M., and Priestley, M. J. N. (2002). “A VIBRATIONAL BASED METHODOLOGY FOR AFTER-EARTHQUAKE DAMAGE ASSESSMENT OF MULTISTORY FRAME-RESISTANT BUILDINGS,” *Proc. of the 12th European Conference on Earthquake Engineering*, London, UK, September 9-13.

Elgamal, A., Fraser, M., and Pagni, C. (2002). “INTERNET LIVE SHAKE-TABLE TESTING FOR EDUCATION IN EARTHQUAKE ENGINEERING,” *Proc. of the International Conference on Engineering Education (ICEE-2002),* Manchester, UK, August 18-22.

Fraser, M., Elgamal, A., Zonta, D. (2002). “The Webshaker Pilot Project: An Internet Framework For Real-Time Monitoring and Control Systems of Civil Engineering Structures,” *Proc. of the 3rd World Conference on Structural Conference*, Como, IT, April 7-12.

Elgamal, A., Zonta, D., and Fraser, M. (2001). “A Pilot Web-Shared Controlling and Monitoring System for Real-Time Assessment of Civil Engineering Structures,” *Proc. of the 2nd ECCE Symposium on Information and Communication Technology in the Practice of Building and Civil Engineering*, Espoo, Finland.

ABSTRACT OF THE DISSERTATION

Development and Implementation of an Integrated Framework

for Structural Health Monitoring

by

Michael Swain Fraser

Doctor of Philosophy in Structural Engineering

University of California, San Diego, 2006

Professor Ahmed Elgamal, Chair

Information technologies are increasingly facilitating life-long monitoring for civil infrastructure applications. Sensor data, including video signals, can be used for long-term structural condition assessment, traffic-load regulation, and emergency response following earthquakes or other natural / man-made hazards. In such a strategy, data from thousands of sensors may be analyzed with real-time and long-term assessment and decision-making implications.

Addressing the above, a flexible and scalable framework has been developed and implemented. This framework networks and integrates on-line real-time heterogeneous sensor data, computer vision, and archiving systems.

Two integrated systems for structural health monitoring have been established as demonstration testbeds located on the University of California, San Diego (UCSD) campus. These systems handle all tasks of monitoring including, data acquisition, data transmission, data archiving, and database querying. From a bridge-deck testbed, the use of time synchronized sensor plus video data has been pioneered. Using this testbed, over 400,000 sets of synchronized traffic-induced strain time histories with video have been recorded and archived over a three year period. Using extracted features from image processing, unique cleansed datasets of labeled traffic, well suited for use with supervised machine learning algorithms, have been created and are available on-line. Elements of this dataset are employed for estimation of traffic speed and vehicle classification efforts (in collaboration with co-workers).

A computational model of the bridge deck system was created to provide strain time histories similar to those actually recorded. Within the numerical simulation framework, it is shown that neural networks perform satisfactorily in providing accurate estimates of vehicle speeds, wheelbases, and axle weights. A neural network-based damage detection methodology has also been constructed, verified on the numerically simulated data, and tested on the recorded data.

An integrated system for structural health monitoring, ready for deployment at the thousands of heterogeneous sensors level, has been deployed on the Voigt Drive / Interstate-5 overcrossing testbed, also on campus at UCSD. An important aspect of this 4-span, 2-lane bridge testbed lies in its availability for collaborative interdisciplinary research. System details and data from a shakedown test conducted with a pilot sensor array are presented and discussed.

1 Introduction to Structural Health Monitoring

This chapter includes a brief literature review of visual inspection for damage detection, non destructive testing, structural monitoring systems, and an overview for new heterogeneous sensor networks for civil infrastructure health monitoring applications. Finally, an outline of the thesis chapters is provided.

1.1 Current State of Civil Infrastructure

The deterioration of civil infrastructure in North America, Europe and Japan has been well documented and publicized. In the United States, 50 percent of all bridges were built before the 1940's and approximately 42 percent of these structures are structurally deficient [e.g., Stalling et al., 2000; ISIS Canada, 2000; Aktan et al., 2001a]. In seismically active regions such as the West Coast of the United States and Japan, the problem of gradual deterioration of the infrastructure over time is compounded by the sudden damage events (or exacerbation of existing damage) due to the occurrence of earthquakes [e.g., Elgamal et al., 2002 and Conte et al., 2003]. The above statistics underline the importance of developing reliable and cost effective methods for managing the massive rehabilitation investments needed in the years ahead.

1.2 Visual Inspection for Damage Identification

Visual inspection is the traditional method for inspecting structures for damage. However, this method has some inherent drawbacks, the first of which is the damage must have progressed far enough to be visually observable. Second, through visual inspection only the extent of damage is assessed based on subjective criteria. This leads to the associated issue of two inspectors examining the same signs of deterioration and making different judgments as to the extent and significance of damage. Even if the damage is successfully identified, the final problem facing the engineer is accurately assessing its effect on the overall health of the structure [Aktan et al., 2001a]. Visual inspection also requires much time and effort, and may overlook locations of limited and/or no accessibility.

A study conducted by the Federal Highway Administration’s NDE Center [FHWA, 2001] on the accuracy of visual inspection of short-to-medium span bridges concluded that at least 56% of the bridges given an average condition rating were done incorrectly [Aktan et al., 2001a]. Other findings [Turner-Fairbank Highway Research Center, 2005] of interest included:

- Inspectors commonly failed to make notes regarding observed defects – an obvious failure as these observations are critical to understanding the assigned condition assessment.
- A recommendation to change the bridge rating system from simply assigning a rating of 0-9 to a more detailed evaluation which provides greater accuracy and reliability regarding the health of the structure.
- The quality of the inspection is greatly affected by the ease of inspector’s access to bridge elements and inspector’s comfort with height of structure.
- Recommendations for increased training of inspectors to recognize commonly occurring forms of defects.

For extended structures, such as long-span suspension bridges, the difficulty is further compounded. Successful visual inspection of these structures is dependent on inspecting all possible damage scenarios at all critical locations, not an easily accomplished task even for an experienced inspector. Reportedly, the in-depth visual inspection carried out on the Brooklyn bridge in New York consumes three months of time at a cost of over one-million dollars [Yanev, 2000 and Aktan et al., 2001b]. Despite all these limitations, visual inspection remains today the most commonly practiced damage detection method.

1.3 Non Destructive Testing

A more promising means of damage detection involves the analysis of the dynamic signature of the structure for features indicative of damage. This field of study, known as non-destructive evaluation (NDE), attempts to detect damage in the structure based on changes in the vibration characteristics. This method is based around the assumption that damage will alter the physical properties of the structure consequently altering the structural response [Farrar and Doebling, 1997].

Several methods have been proposed to detect any changes in structural response and to link it with the associated damage. Most of these methods can be classified as belonging to one of three techniques:

- Damage detection based on modal parameters. These methods start with extracting modal parameters (such as natural frequencies, mode shapes, and damping ratios) from the recorded data and using these values to calculate other physical properties, e.g. system stiffness matrix. Various system parameters (including changes in natural frequencies, changes in mode shapes, changes in modal curvature, Ritz vector, matrix updating) have been studied and their application as damage indicators explored [Doebling et al., 1996; Farrar and Jauregui, 1996; Farrar and Doebling, 1997; Sohn et al,. 2003].

- Damage detection using statistical pattern recognition approaches. This class of techniques treats damage detection as a pattern recognition problem whereby using a learned mapping, the causes are discerned from the measured system response. Within these techniques, the application of neural networks has become a favored mapping identification tool [Garret et al., 1992; Ghaboussi, 1993 and 1994; Sohn et al., 2001]. Problems associated with these methods involve: How best to perform feature extraction / data reduction on the recorded data to manage the size of the neural networks. When used as a supervised learning tool it is necessary to use data from the undamaged and damaged states of the structure. Often times data from the damaged structure is difficult, if not impossible, to come by; When multiple damage locations are considered (especially with variable damage levels) the amount of training data increases exponentially.
- Damage detection using time series prediction. In order to avoid some of the difficulties associated with the previous methods, researchers are looking at ways to perform damage detection based on time series prediction [Sohn and Farrar, 2001]. These methods are typically based on

using vibration measurements from a healthy structure to train a neural network to predict the system response. When damaged, there will be a change in the measured system response and will be seen in the error between the measured and predicted response [Masri et al., 1996, Nakamura et al., 1998, and Xu et al., 2003].

Comprehensive reviews of these methods are available in Sohn et al., 2003 and Doebling et al., 1996.

1.4 Early Attempts at Structural Monitoring

Early attempts at the application of the aforementioned damage detection methods consisted of special tests involving the installation of sensors on a given structure and recording data over a relatively short period of time. Once recorded, the instrumentation was removed and the data was post-processed and analyzed off-site. For bridges, this testing typically involves closing at least one lane of traffic while the sensors are being installed (Figs. 1.1 and 1.2). Figures 1.1 and 1.2 show the installation by the author of a temporary array of accelerometers using an articulated boom truck, on the Byron Road Bridge in Tracy, CA (pilot test conducted by the author and co-workers). In addition, to minimize uncertainty in the traffic loading, the bridges are shut down to all traffic for short periods of time while ambient vibration data was recorded. This procedure thereby becomes a time consuming procedure, which must be closely coordinated with local traffic management agencies.

As data is only recorded when the sensors are installed on the structure, these methods are only able to capture progressive deterioration when several data sets, recorded over an extended period of time, are examined. For locating suddenly occurring damage, for example if damage is suspected from a visual inspection, the structure must be instrumented and data recorded. This data is later analyzed and compared to either previously recorded data or to a simulated model for locating damage, a procedure which can take several weeks to go from planning, to field testing, to analysis. In this regard, this experimental technique does not lend itself to rapid structural assessment.

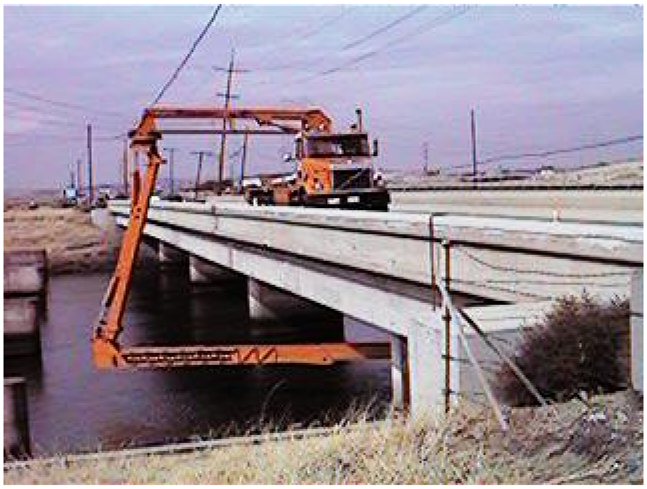


Figure 1.1: Installation of Accelerometers on the Byron Road Bridge in Tracy, CA.

Figure 1.2: Close-up of Installation on the Byron Road Bridge in Tracy, CA.

1.5 Continuous Full-Scale Structural Health Monitoring

One improvement was to permanently install the sensors on the bridge to reduce the amount of time require for testing and minimize the impact on traffic (in particular associated with sensor installation) [Aktan et al., 2000; Gaun et al., 2006; Chen and Feng, 2006]. However, because of the instrumentation costs and hardware limitations (associated with the number of analog-to-digital channels available) only a limited number of sensors were utilized [Talbot and Stayanoff, 2005]. Further, in these tests, only one type of sensor was often used [Livingston, 2004]. As an example, Figure 1.3 shows the sensor layout for the Vincent Thomas Bridge in San Pedro, CA [Wahbeh et al., 2005 and Smyth et al., 2003]. This instrumentation scheme, composed of 26 accelerometers, is common to many of the bridges instrumented by Caltrans. From the data collected from this limited density sensor array, engineers are left with much guess work as to the actual health of the structure. Common analysis techniques using this type of data involved modal analysis (e.g. tracking the first few natural frequencies and damping ratios over time [Lus et al., 1999 and Smyth et al., 2003]). For determining mode shapes and modal curvatures of the bridge, it is necessary to make assumptions regarding the behavior of the structure. For the above example of the Vincent Thomas Bridge, engineers noting the symmetry of the bridge decided to only instrument one-half of the bridge. Thereby, when mode shapes are calculated using the recorded data, it is only possible to calculate mode shapes for half of the bridge and then assume symmetric behavior [Abdel-Ghaffar and Housner, 1997]. For local damage, where the damage is not distributed over the bridge, this assumption of symmetric behavior is no longer valid.

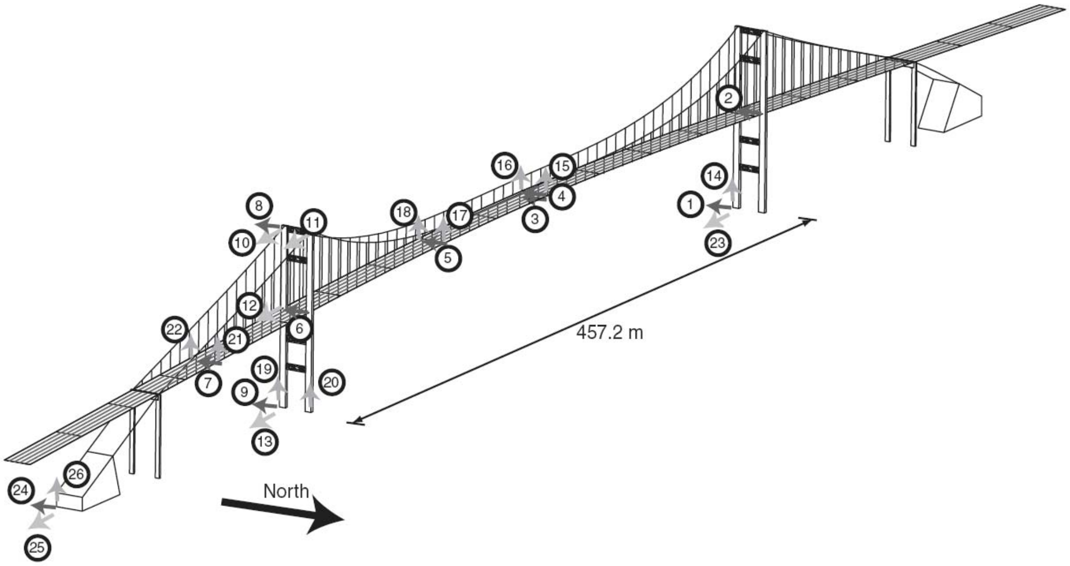


Figure 1.3: Sensor Array for Vincent Thomas Bridge in San Pedro, Ca [Wahbeh et al., 2005].

Another improvement which further improved structural health assessment involved the periodic transmission of data from the structure under ambient vibration. This periodic data helps to form a baseline for which new data can be compared against for locating damage. Originally, this data was transmitted using telephone-based modems which, with their limited bandwidth, restricted the amount and frequency of data that could be transmitted.

With the recent developments in sensors, PC-based data acquisition hardware, wireless technologies [Law et al., 2005; Lynch and Loh, 2005; and Lynch, 2005], and broadband data transmission, there is the potential to acquire data from a great number of sensors on a structure and stream the data in near real-time for rapid damage assessment [Elgamal et al., 2003a, b, 2004; Conte et al., 2003; and Fraser et al., 2003]. It is expected this next generation of structural health monitoring systems [Wang, 2005] will make it possible to detect two types of infrastructure deterioration: (i) progressive deterioration over time due to environmental effects, and (ii) sudden deterioration due to natural hazards such as earthquakes and hurricanes, man-made disasters, and acts of terrorism [Elgamal et al., 2002]. Further, these systems will allow transportation management to understand the true state of health and rate of degradation of each significant bridge of the transportation system, which often cannot be determined from visual inspections only. This critical information provides a rational basis for the optimum allocation of limited financial resources towards the maintenance, rehabilitation and strengthening of the transportation system as a whole.

Since the occurrence of the 1994 Northridge earthquake and the 1995 Kobe earthquake, there has been a quantum jump in the number of civil structures that have been instrumented for monitoring purposes. Furthermore, plans are underway to install a variety of strong-motion vibration sensors in many civil structures (in some cases many hundreds of sensors in a single structure). In addition to the strong motion instrumentation, various other sensor types will also be integrated within these new larger-scale sensor arrays.

These new sensors, which include thermocouples, thermistors, wind monitoring stations, and relative humidity probes, seek to provide insight into environmental effects on the measured structural response. By integrating many different types of vibration sensors, each with their own unique advantages/disadvantages, further insight into the structures vibrational response will be gained. For example, low-cost capacitive accelerometers are very useful for capturing the low frequency motion associated with the first few natural frequencies. However, because of their limited frequency range, these sensors are not useful for capturing high-frequency motion associated with crack initiation/propagation. For this application, piezoelectric seismic accelerometers and acoustic sensors coupled with high speed data acquisition is required.

A final collection of sensors will serve to capture traffic passing over a bridge and monitor its effect on the individual structural elements. This group of instrumentation includes digital video cameras, strain gages (both electrical resistance and fiber optic), and load cells. In addition to providing the needed insight into the loading on the structure for bridge maintenance, these sensors will also provide a means for automatically monitoring traffic to detect overloaded and speeding vehicles for traffic enforcement [Aktan et al., 2001a].

1.6 Hurdles for the Successful Deployment of a Structural Monitoring System

Clearly, the main issue now facing the structural health monitoring community is not the lack of measurements per se, but rather how to measure, acquire, process, and analyze the massive amount of data that is currently coming on-line (not to mention the terabytes of streaming data that will inundate potential users in the near future) in order to extract useful information concerning the condition assessment of the monitored structures [Elgamal et al., 2002]. Such an effort will not only require collaboration among the data analysis, structural engineering, video processing, wireless and sensor network communities, but it will require a comprehensive approach to data management and analysis. The complexity of data sources, (including real-time sensor and video streams, and the output of physics-based and statistical models), and the need to perform advanced real-time and off-line analyses (often requiring the integration of real-time sensor data with simulation model output) necessitates a scaleable high-performance computational infrastructure (Fig 1.4) [Elgamal et al., 2002; Elgamal et al., 2003; Fraser et al., 2003].

Finally, for these new monitoring systems to become widely adopted, they must provide clear benefits to infrastructure owners and engineers rather than inundating them with massive amounts of disjoint data. Further, to absorb the considerable expense of these monitoring systems in new structures, the cost should be included as part of the overall construction cost rather than as part of the yearly maintenance budget.

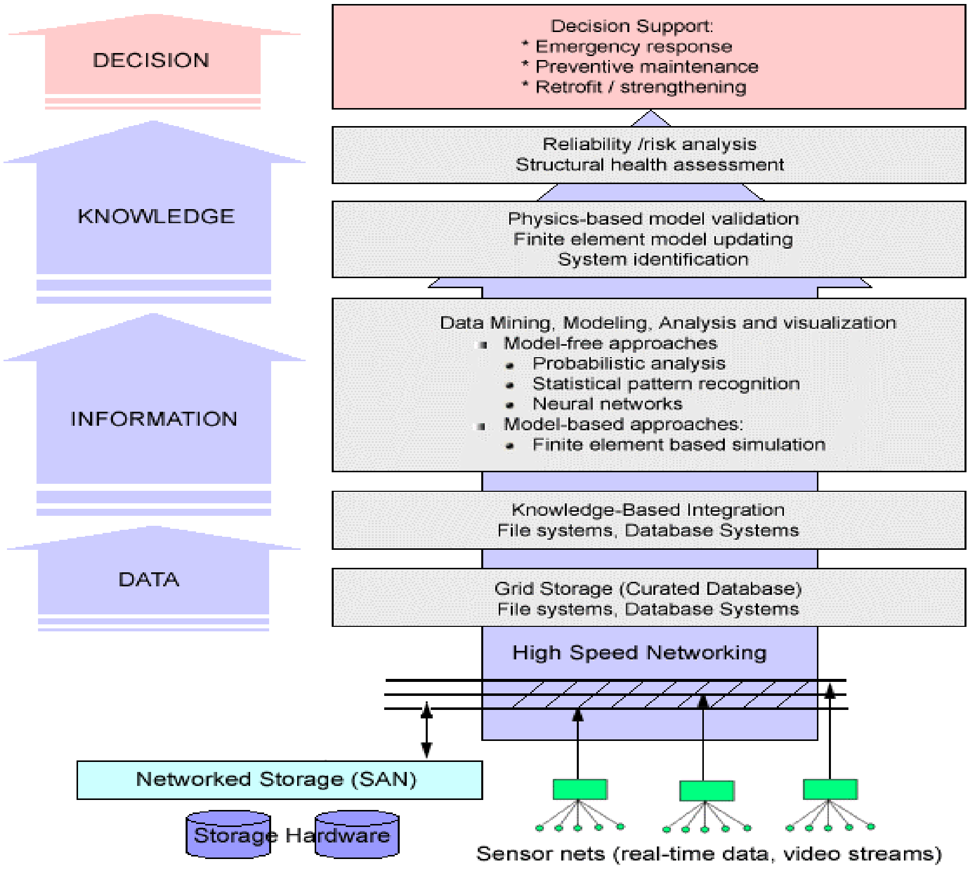

DECISION
KNOWLEDGE
INFORMATION
DATA
Decision Support:
* Emergency response
* Preventive maintenance
* Retrofit / strengthening
Reliability /risk analysis
Structural health assessment
Physics-based model validation
Finite element model updating
System identification
Data Mining, Modeling, Analysis and visualization
Model-free approaches
Probabilistic analysis
Statistical pattern recognition
Neural networks
Model-based approaches:
Finite element based simulation
Knowledge-Based Integration
File systems, Database Systems
Grid Storage (Curated Database)
File systems, Database Systems
High Speed Networking
Networked Storage (SAN)
Storage Hardware
Sensor nets (real-time data, video streams)

Figure 1.4: Knowledge Discovery Hierarchy: Chart Developed by Tony Fountain, UCSD San Diego Supercomputer Center (SDSC), Joel P. Conte, and Ahmed Elgamal [Elgamal et al., 2003b]

1.7 How This Research Contributes to Advancing the Current State of Practice

The research presented within seeks to develop a framework with an open and flexible architecture able to integrate current and future research in the field of structural health monitoring (e.g., local non-destructive evaluation techniques such as acoustic emissions). The overall research framework addresses development of: (1) networked sensor arrays, (2) a high-performance database with data cleansing and error checking, data curation, storage and archival, (3) computer vision applications, (4) tools of data analysis and interpretation in light of physics-based models for real-time data from heterogeneous sensor arrays, (5) visualization allowing flexible and efficient comparison between experimental and numerical simulation data, and (6) data fusion. In order to satisfy these requirements, this research is making use of recent advances in (1) high-performance databases, knowledge-based integration, and advanced query processing, (2) instrumentation and data acquisition hardware, (3) computer vision and related feature extraction algorithms, and (4) data mining, model-free and model-based advanced data analysis, and visualization. As part of a larger NSF funded Information Technology Research program, an integrated system based on the knowledge discovery hierarchy represented in Figure 1.4 is being built with this framework to achieve the above-mentioned objectives. This system integrates all tasks from sensor configuration, data acquisition and control, to decision-making and resources allocation (Figure 1.5). It is the aim of this research to not only lead to a versatile integrated framework for condition assessment and damage detection under normal operating conditions, but also to be beneficial in providing rapid response (in virtually real time) due to sudden dynamic loads or terrorist acts.

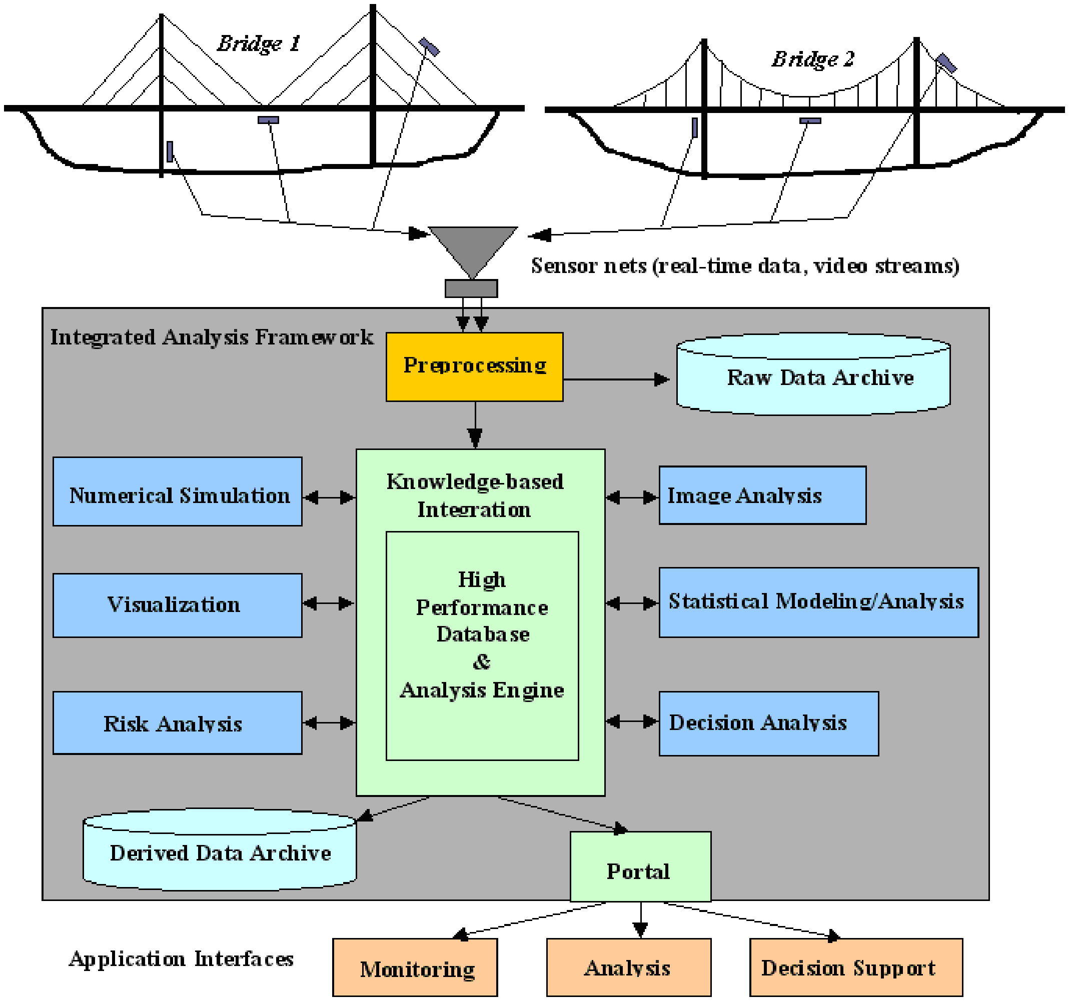


Figure 1.5: Conceptual System Architecture [Conte et al., 2003]

1.8 Thesis Outline

In the following chapters:

- As envisioned in this thesis, the components of a state-of-the-art structural monitoring system are detailed (Chapter 2). This includes information related to motion and environmental sensors, data / image acquisition, data streaming, data archival, and web-based dissemination.
- Next, a demonstration testbed is presented (Chapter 3). Within this chapter, the monitoring system employed on a series of bridge-deck panels subject to traffic loading is discussed.
- Image processing is performed on the video recorded from the aforementioned bridge-deck testbed (Chapter 4). The image processing / feature extraction operation is analyzed and the findings discussed. From this analysis procedure, a new and unique data set composed of traffic induced strain time histories (with time-synchronized video) sorted by vehicle type is created.
- To simulate the response of the bridge decks under vehicular loading, a one-dimensional finite element model is constructed (Chapter 5). This model is created to generate data otherwise unavailable with the real system (high-speed traffic, damaged state…).
- Vehicle property estimation and damage detection / classification is performed on the simulated data using time history analyses (Chapter 6).

- Neural networks are then applied to the data from the finite element model to determine vehicle properties and for damage detection purposes(Chapter 7).
- The strain data recorded on the bridge decks are analyzed to calculate speeds and wheelbases of passing vehicles, to examine distributions of peak vehicular strains, and to check for changes in the system response over time (Chapter 8).
- Details of a second structural monitoring testbed installed on a 300-foot long concrete multi-cell box girder overcrossing are presented (Chapter 9).
- Finally, a summary of the dissertation is provided along with key contributions and findings from this research (Chapter 10).

1.9 Summary

The deterioration of the civil infrastructure in North America, Europe and Japan has been well documented and underlines the importance of developing reliable and cost effective methods for the massive rehabilitation investments needed in the years ahead. In managing the transportation system of the nation or of a state, it is essential to understand the true state of health and rate of degradation of each significant bridge in the transportation system, which often cannot be determined from visual inspections only. With the evolutions in nondestructive testing and the advancements in sensor technology, computational power, and PC-based data acquisition systems, continuous structural health monitoring using dense heterogeneous sensor arrays is now becoming technically and economically feasible.

2 Components of a Structural Monitoring System

Within this chapter, components of relevance to a state-of-the-art global structural monitoring system are detailed. First, an overview of the various sensors that may be employed for dynamic and environmental monitoring is presented. Next, options for data and image acquisition are discussed. Finally, details related to database systems and web portals for data archival and dissemination are provided.

2.1 Heterogeneous Sensor Array

Any monitoring system intended for structural health monitoring may incorporate a heterogeneous array of sensors. Most current monitoring systems only make use of one sensor type and have a single purpose, whether it be monitoring concrete shrinkage/expansion, wind speeds, or vibrations under ambient loading [Livingston, 2004]. To fully understand the true state of health and rate of deterioration of the structure, it will be necessary to combine the data gathered from a multitude of sensor types to garner new information.

2.1.1 Vibration Monitoring

The first class of sensors is intended to measure the dynamic signature of structure looking for changes in the vibration characteristics. As was discussed in Chapter 1, these changes are indicative of a change in the structural response which can result from the onset of damage. The most prevalent sensors for vibration monitoring include strain gages, accelerometers, and displacement transducers. These sensors are briefly reviewed in the Section 2.2

2.1.2 Environmental Monitoring

In addition to measuring the dynamic response of civil engineering structures, it is also necessary to perform environmental monitoring. Reasons include understanding the change in the structural and sensor responses due to changes in temperature [Worden et al., 2005], monitoring wind induced vibrations (a primary source of excitation particularly with long-span suspension bridges) [He et al., 2005], and chemical monitoring which can provide useful information regarding corrosion. Typical sensors for environmental monitoring are discussed in Section 2.2.4.

2.1.3 Spatial Resolution

As important as the types of sensors being utilized in structural monitoring is the number of sensors being employed. As additional sensors are added and the spacing between them decreases, the accuracy of the observed deflection curves improve. Dense instrumentation arrays also make possible the detection of an increased number of modes of vibration. Within these newly observed modes, it is possible to examine local modes or local changes in global modes. The methods discussed in Chapter 1, which rely on the quality of the modal extraction, would benefit leading to increases in our ability to detect the onset of damage in it earliest stages.

2.2 Typical Sensors

2.2.1 Strain Sensors

Strain gages measure the expansion and contraction of material due to mechanical stress or thermal effect. Like all transducers, these sensors rely on indirect measurements for determining strains. Two common sensor types are the electrical resistance strain gage and the fiber optic strain gage.

2.2.1.1 Electrical Resistance Strain Gage

The metallic foil-type strain gage consists of a grid of wire filament (a resistor) of approximately 0.001 in. (0.025 mm) thickness, bonded directly to the strained surface by a thin layer of epoxy resin. When a load is applied to the surface, the resulting change in surface length is communicated to the resistor and the corresponding strain is measured in terms of the electrical resistance of the foil wire, which varies linearly with strain. The foil diaphragm and the adhesive bonding agent must work together in transmitting the strain, while the adhesive must also serve as an electrical insulator between the foil grid and the surface.

When selecting a strain gage, one must consider not only the strain characteristics of the sensor, but also its stability and temperature sensitivity. Unfortunately, the most desirable strain gage materials are also sensitive to temperature variations and tend to change resistance as they age. For tests of short duration, this may not be a serious concern, but for continuous, long-term monitoring applications, one should include temperature and drift compensation.

Bonded resistance strain gages have been proven to be reliable. They are relatively inexpensive, can achieve overall accuracy of better than +/-0.10%, are available in a short gage length, are only moderately affected by temperature changes, have small physical size and low mass, and are highly sensitive [http://www.omega.com/literature/transactions/volume3/strain.html]. Bonded resistance strain gages can be used to measure both static and dynamic strains.

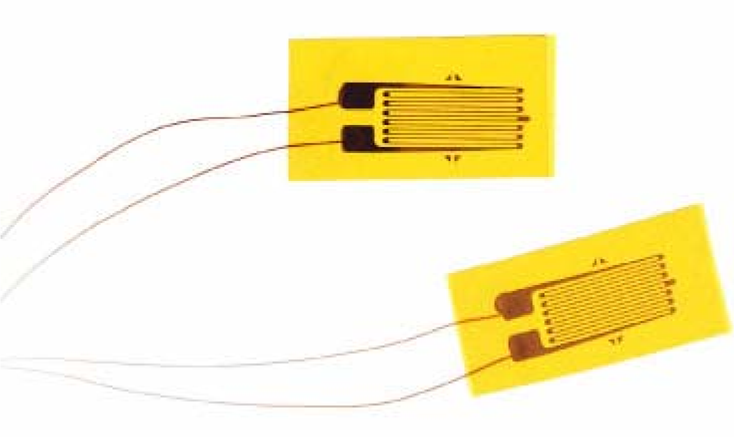


Figure 2.1; Typical Metal-Foil Strain Gages

(http://www.omega.com/literature/transactions/volume3/strain.html)

2.2.1.2 Fiber Optic Strain Gage

Fiber optic strain gages measure the strain by shifting the light frequency of the light reflected down the fiber from the Bragg grating, which is embedded inside the fiber itself. These sensors are immune to EM and RF interference and offer better resolution and resistance to corrosive environments than conventional electrical strain gages. Another important advantage with this sensor is 100 or more gages may be embedded within the same fiber. Using a multiplexing interrogator greatly simplifies the sensor installation and lowers the cost of both the individual sensors and associated cabling [Livingston, 1999 and Todd et al., 1999]. A disadvantage is they are a more delicate sensor, potentially making them unsuitable for certain field applications.

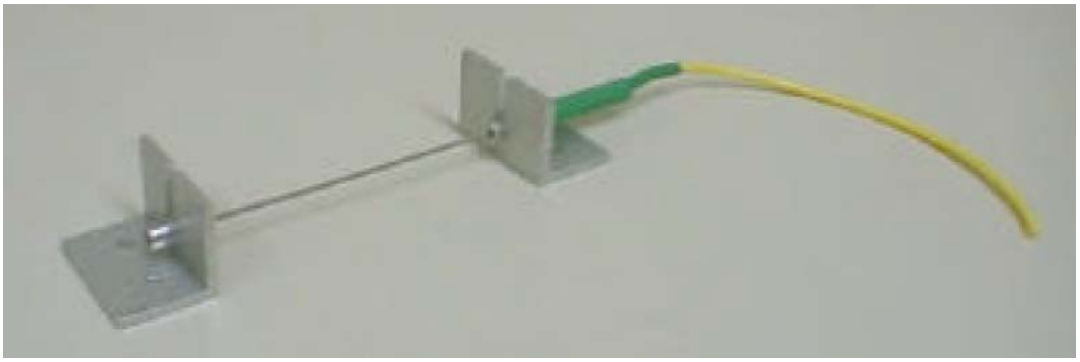


Figure 2.2: Long Gage Fiber Sensor [Schulz et al., 2001]

2.2.2 Displacement Sensors

For monitoring displacements, two general techniques are most commonly used. Direct measurements using either linear variable differential transformers (LVDT's, Fig. 2.3) and non-contacting laser-based sensors or indirect methods involving the use of image processing techniques. Direct measurements are the easiest method, but they are limited to unidirectional measurements. Indirect methods allow the possibility of two-dimensional measurements; however, these measurements are much more computationally expensive, as they require post-processing to yield displacements. This technique is addressed further in the image processing section of this paper (Section 4.7).

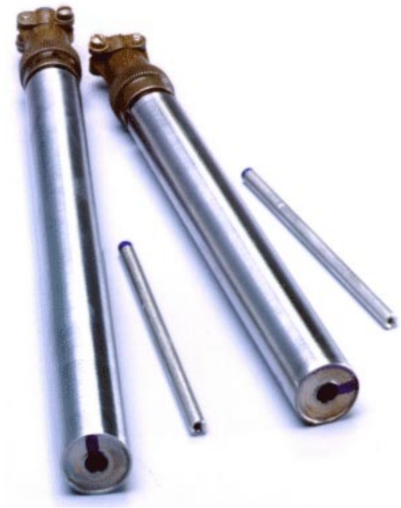


Figure 2.3: Macro Sensors DC 750-3000 LVDT

(http://www.macrosensors.com/)

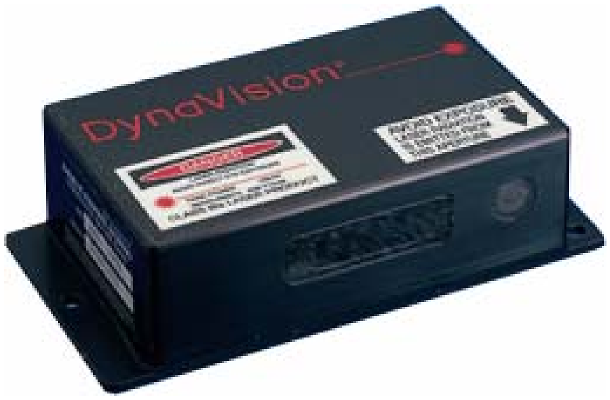


Figure 2.4: Laser Measurement International SPR-04 Single Point Non-contacting Displacement Transducer (http://www.lmint.com/cfm/index.cfm?It=900&Id=23&Se=93&Sv=0)

2.2.2.1 Dynamic GPS

A newly emerging technology for dynamic displacement measurements on civil engineering involves the use of GPS receivers. Previously, GPS has been utilized for synchronizing system clocks in the hardware deployed as part of remote monitoring systems. Through the use of high speed Geodetic receivers operating at speeds greater than 50 Hz (Fig. 2.5) with high rate base line monitoring, relative displacement accuracies on the order of 1 millimeter in the horizontal directions and 2 millimeters in the vertical direction can be achieved [Bock et al., 2000 and Bock et al., 2006]. These techniques require placing a GPS antenna (but not necessarily the GPS interface module) at each location where one wants to monitor displacements on the structure. In addition, to calculate relative displacements, it is necessary to have one or more additional reference stations (with the same hardware and sampling rate) installed near the structure.

Using TCP/IP communication, the appropriate monitoring software, such as RTD Net software (http://www.geodetics.com/WebSite/products/RTD-Family.html), controls the receivers and flow of data from the antennas to an off-site central archiving computer. Once the instantaneous position is archived, it can be analyzed in near real time with a latency of less than one second. The same GPS receivers used for monitoring displacements are also suitable for synchronizing multiple acquisition computers on a distributed sensing array (see Section 2.3.2.2 for further details).

Figure 2.5: Thales Navigation Z-Max and DG14 GPS Receiver and Antenna (http://www.thalesnavigation.com/en/)

2.2.3 Accelerometers

With strain gages, accelerometers are the most commonly used sensor in structural monitoring. A brief summary of the most common accelerometer types is presented and their inherent benefits/drawbacks summarized (Table 2.1).

2.2.3.1 Force Balance Accelerometers

Force balance accelerometers (FBA, Fig. 2.6) are spring mass devices which feature high sensitivity over a relatively low frequency range. The ability to make low frequency measurements with high accuracy makes this class of accelerometers particularly well suited for seismic and general structural monitoring applications (http://kinemetrics.com). Because of their exceptional resolution, this class of accelerometer became the standard for permanent structural monitoring systems. However, due to their high costs and evolving sensor technologies, other types of accelerometers are becoming more popular.

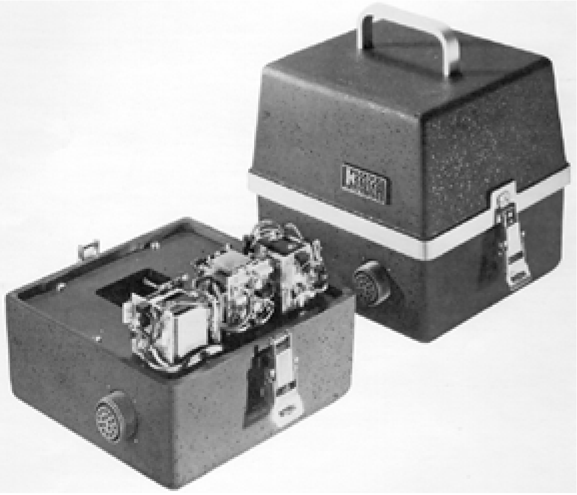


Figure 2.6: Kinemetrics FBA-3 Force Balance Accelerometer

(http://kinemetrics.com/product_Content.asp?newsid=158)

2.2.3.2 Piezoelectric Accelerometers

This class of sensor makes use of the piezoelectric effect discovered in 1880 by Pierre and Jacques Curie, whereby certain crystals exhibit electrical charges under mechanical loading. Piezoelectric accelerometers incorporate a crystal sensing element which has the property of emitting a charge when subjected to a force. As these are active electrical systems, the crystals produce an electrical output only when they experience a change in load, they cannot perform true static measurements. Typical piezoelectric accelerometers (Fig. 2.7) offer higher measurement and frequency ranges than force balance accelerometers at the expense of resolution and inability to measure down to 0 Hz (http://www.pcb.com). Lower costs are another advantage of piezoelectric accelerometers over FBA's.

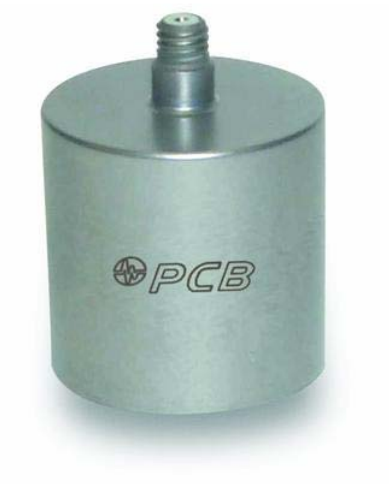


Figure 2.7: PCB Model 393B04 Piezoelectric Accelerometer

(http://www.pcb.com/)

2.2.3.3 Capacitive Accelerometers

Capacitive accelerometers (Fig. 2.8) measure acceleration by monitoring a change in electrical capacitance. Within these sensors, the sensing element consists of two parallel plate capacitors acting in a differential mode. These capacitors operate in a bridge circuit, along with two fixed capacitors, and alter the peak voltage generated by an oscillator when the structure undergoes acceleration [http://www.sensorland.com/HowPage011.html]. Like the force balance accelerometer, capacitive accelerometers typically operate in a low frequency range. Unlike piezoelectric accelerometers, these sensors can measure down to 0 Hz. The principal advantage of capacitive accelerometers is their low cost, making them attractive for dense sensor arrays; however, the resolution of these sensors is typically less than either force balance or piezoelectric accelerometers.

Figure 2.8: Crossbow CXL01LF1 Capacitive Accelerometer

(http://www.xbow.com/Products/productsdetails.aspx?sid=32)

Table 2.1: Comparison of 3 Typical Accelerometers

Accelerometer Type / Model	Measurement Range	Voltage Sensitivity	Frequency Range	Broadband Resolution	Weight
FBA – Kinemetrics FBA-3	±1 g	2500 mV/g	DC - 50 Hz	2.5x10^{-10} g rms	7 kg
Piezoelectric – PCB 393B04	±5 g	1000 mV/g	0.06 Hz to 450 Hz	3x10^{-6} g rms	0.05 kg
Capacitive – Crossbow CXL01LF1	±1.25 g	2000 mV/g	DC - 50 Hz	5 x10^{-4} g rms	0.068 kg

2.2.4 Environmental Sensors

2.2.4.1 Monitoring Surface Temperature of Structural Elements

There are several very important reasons for monitoring the temperature of the instrumented structure. First, and perhaps most importantly, many studies have shown that a structure’s natural frequencies and mode shapes are greatly affected by temperature [Worden et al., 2005]. Other considerations include performing temperature compensation to recorded sensor data (i.e. apparent strains and linear scaling of accelerometers), predictions for ice formation, and the assurances of proper concrete curing during construction. To measure the temperature of the structural elements of the bridge (i.e. truss-stiffener members), thermocouples and thermistors are utilized. A thermocouple consists of two dissimilar metals, joined together at one end, which produce a small unique voltage at a given temperature. The Omega “Cement-On” Model COT (Fig. 2.9) is an economical fast response thermocouple for surface temperature measurements. These sensors are constructed of 30 gage (0.010") diameter Copper and Constantan wire embedded in a paper-thin laminate intended for surface applications by bonding with added adhesive. The nominal operational temperature range is -190 to 205°C (-310 to 401°F). These sensors are employed in the monitoring system deployed on the Voigt Bridge on the University of California, San Diego (UCSD) campus and is discussed in Chapter 9.

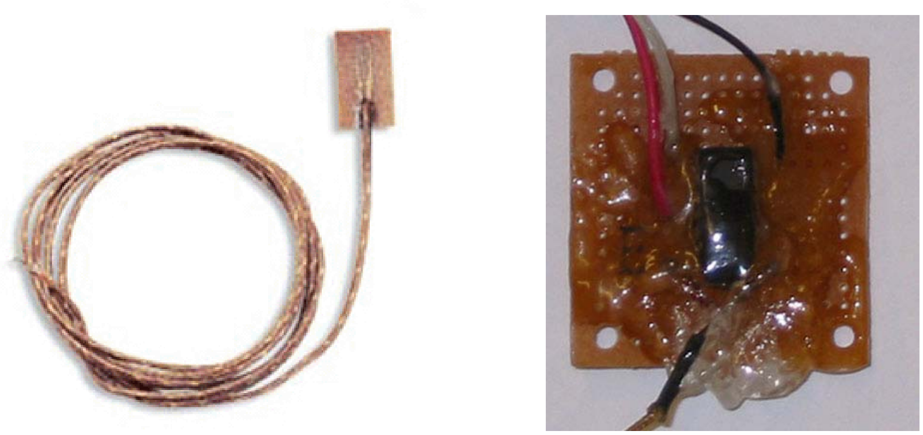


Figure 2.9: Omega "Cement-On" Model COT Thermocouple (Left, http://www.omega.com/ppt/pptsc.asp?ref=CO-K) and Signal Conditioning Chip (Right)

2.2.4.2 Monitoring Ambient Conditions

To measure the relative humidity and air temperature, dual purpose sensors like the Pace Scientific Temperature/RH Sensors (Fig. 2.10) are readily available (http://www.pace-sci.com/humidity.htm). This is a precision relative humidity and temperature probe, which outputs an analog output easily measured with standard data acquisition hardware. Another advantage is it does not require any signal conditioning, just a 5 vdc excitation at 2 ma. The humidity output from the sensor is typically 0.8 vdc at 0% Relative Humidity (RH) and 3.9 vdc at 100% RH. To convert from measured voltage to percent relative humidity, linear scaling is performed with ± 2% RH accuracy from 0 – 95% RH. Temperature compensation for the RH measurements is similarly performed.

Figure 2.10: Pace Scientific Temperature/RH Sensor (http://www.pace-sci.com/humidity.htm)

2.2.4.3 Wind Speed Monitoring

As wind induced vibrations are a major source of vibration, particularly on long-span suspension bridges [He et al., 2005] where they can govern design, wind speeds and directions are important quantities to be measured. An inexpensive option for these measurements is the WMS-22B (Fig. 2.11) series sensor capable of monitoring wind speed and direction at discrete locations on a bridge. Each measured parameter is converted into a 4 to 20mA signal, which through signal conditioning can be acquired through the conventional data acquisition hardware.

These stations are capable of measuring wind speed from 1 to 136 miles-per-hour (mph) with 1 mph resolution. For wind direction, the station offers 0º to 360º capability with 2º resolution.

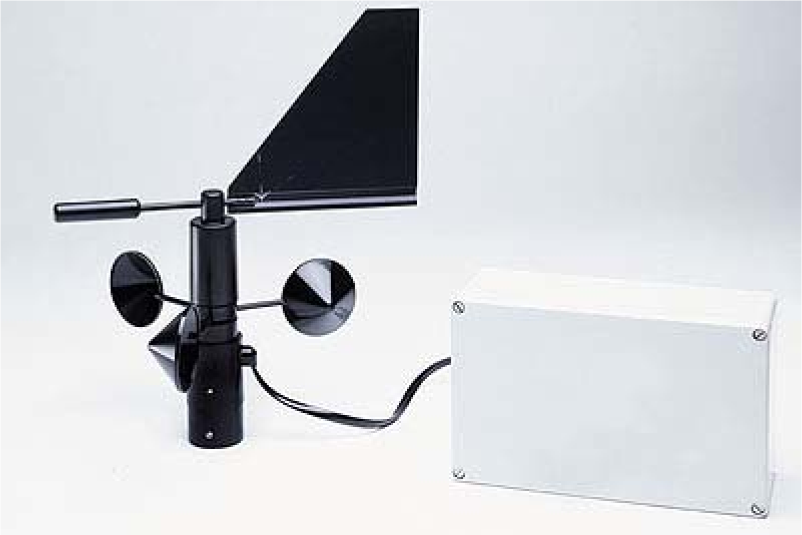


Figure 2.11: WMS-22B Series Wind Monitoring Station

(http://www.omega.com/ppt/pptsc.asp?ref=WMS-20)

2.2.4.4 Chemical Sensing

Chemical sensing, such as the detection of chlorides in reinforced concrete, can be a useful tool for detecting corrosion in these structures [Livingston, 1999]. Other factors to be monitored include the pH during the curing phase of concrete construction and half-cell potential (http://www.concretecorrosion.net/html_en/maitrise/potent.htm) or redox potential for assessing the condition of steel reinforcement in concrete.

Figure 2.12: SCRIBE Half-Cell Potential Meter (http://www.cmtinstruments.com/combined-logging-meter-and-half-cell-potential-meter.htm)

2.2.5 Computer Vision

Within this research, video will serve two important purposes:

1. To provide quantitative information about the pattern of the traveling traffic loads (and indirectly, an idea about magnitude of these loads) by using pattern recognition/video-processing techniques. Integration of acting traffic loads (or load patterns) with the corresponding measured strains will reduce uncertainty during the system-identification analysis phase (by limiting the scope of possible causative load configuration scenarios).

2. To quantitatively measure relative bridge deck motions, component motions, and differential joint-motions. In this regard, no sensors are currently available for accurate measurement of displacements along an extended structure such as a bridge (we usually rely on double integration of acceleration records, but this may introduce significant error). As is demonstrated in the following demonstration application, computer vision makes it possible to generate this data; however, before it can be practically applied for real-time analysis of high-speed video (greater than 10 frames-per second), more work is necessary to develop and optimize robust approaches for capturing, processing, and analyzing video.

2.2.5.1 Image Analysis for Monitoring Displacements on Bridges

The second application, one that has drawn interest recently, is the two and three-dimensional measurement of displacements on bridges [Smyth et al., 2003]. One example is to track displacements of seismic dampers installed on bridges to monitor the amount of energy dissipated. In this application, linear variable differential transformers (LVDT's) are not suitable as they are capable of making single axis measurements only. Consequently, optical techniques have drawn attention. One possible means of applying these techniques involves the use of a high-resolution camera and laser target or a high resolution camera with an infrared illuminator and reflector [Wahbeh et al., 2003]. Using image processing techniques similar to those previously discussed, the displacements on the bridge can be calculated by tracking the motion of the laser or target.

A simple demonstration was conducted wherein a visible laser was shown onto a wall, placing a red dot where the laser contacted the wall (Fig. 2.13). A camera was stationed nearby and recorded the dot on the wall as the laser was moved. While the use of the moving laser on a stationary target for monitoring displacements on a civil engineering structure is not as practical as using a stationary illuminator with a reflector attached to a moving structure, it does produce the same resulting images. Ultimately, both methods produce a set of images in which the target (or laser dot) is the only moving object.

The original image of the laser striking the wall is shown in Fig. 2.13. A suitable background image was formed by recording when the laser was turned off. When the difference between the background and target images is taken the resulting image is shown in Fig. 2.14 (after some additional processing and cleansing). By repeating this process for each image recorded and calculating the horizontal and vertical centroids of the laser, it is possible to chart the path of the laser (Fig. 2.15) and to generate displacement time histories of the motion (Fig. 2.16).

With this method, the resolution of the displacement calculation is a function of the resolution of the camera and the size of the portion of the structure being observed (which is controlled by the zoom capabilities of the camera lens and the distance the camera is mounted from the structure). Further, the frequency range over which this technique is suitable for monitoring displacements is controlled by the sampling (frame) rate of the camera and video acquisition system. For long-span suspension bridges, which are typically governed by their lower modes and have as many as 20 or more resonant frequencies under 1 Hz [He et al., 2005], easily obtainable frame rates of 10 Hz are suitable. With sampling rates of this order, local processing is possible; thereby rendering the monitoring system able to send only the extracted displacements (or velocities) rather than raw images. This can provide for a savings on bandwidth of the order of 10 Bytes versus an excess of 1 Megabyte per sample.

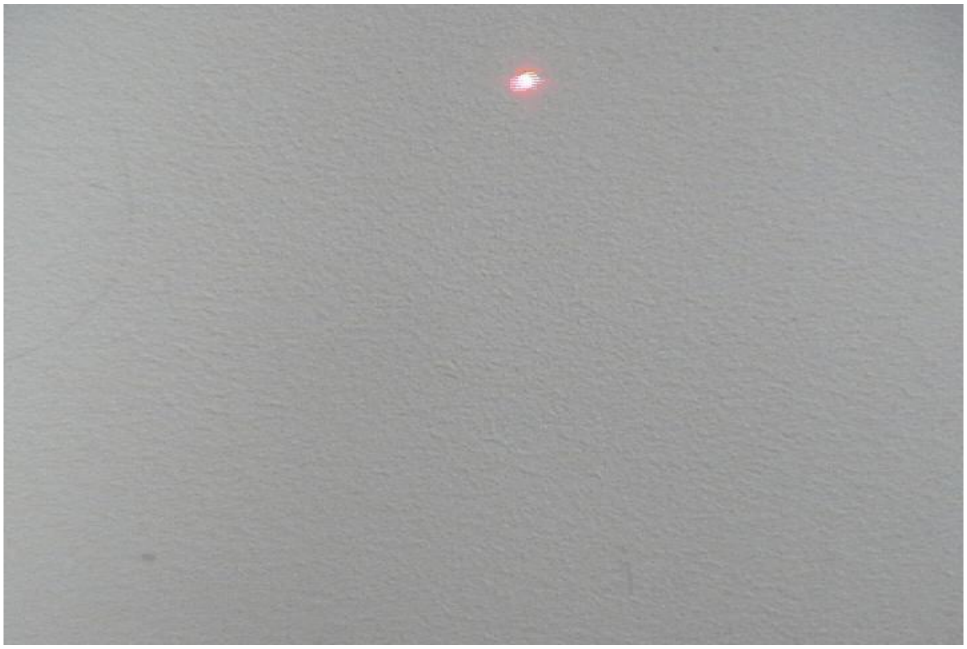


Figure 2.13: Original Image of Laser Target

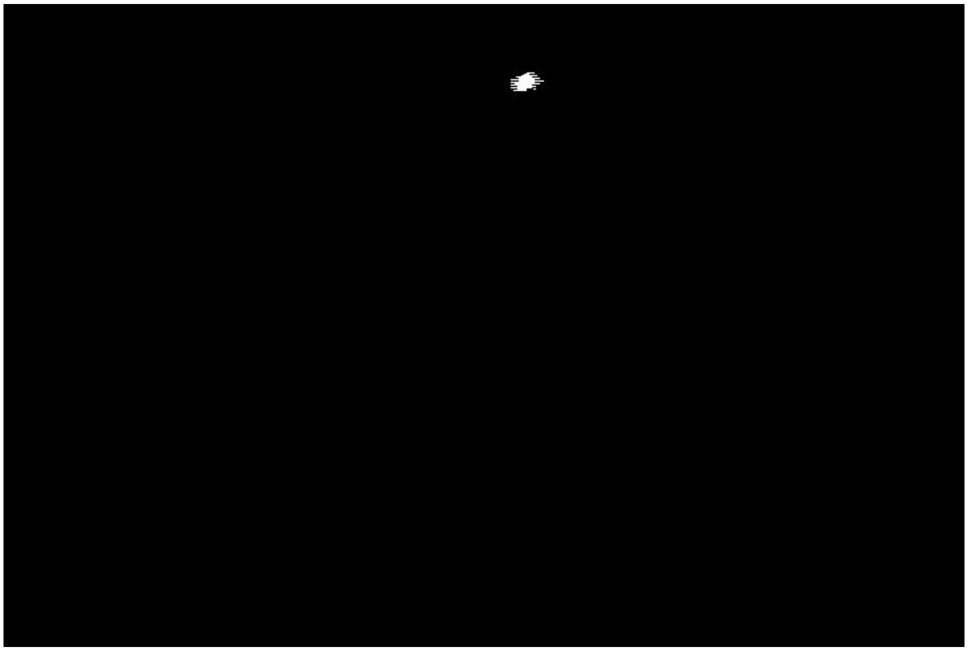


Figure 2.14: Resulting Image from Image Processing

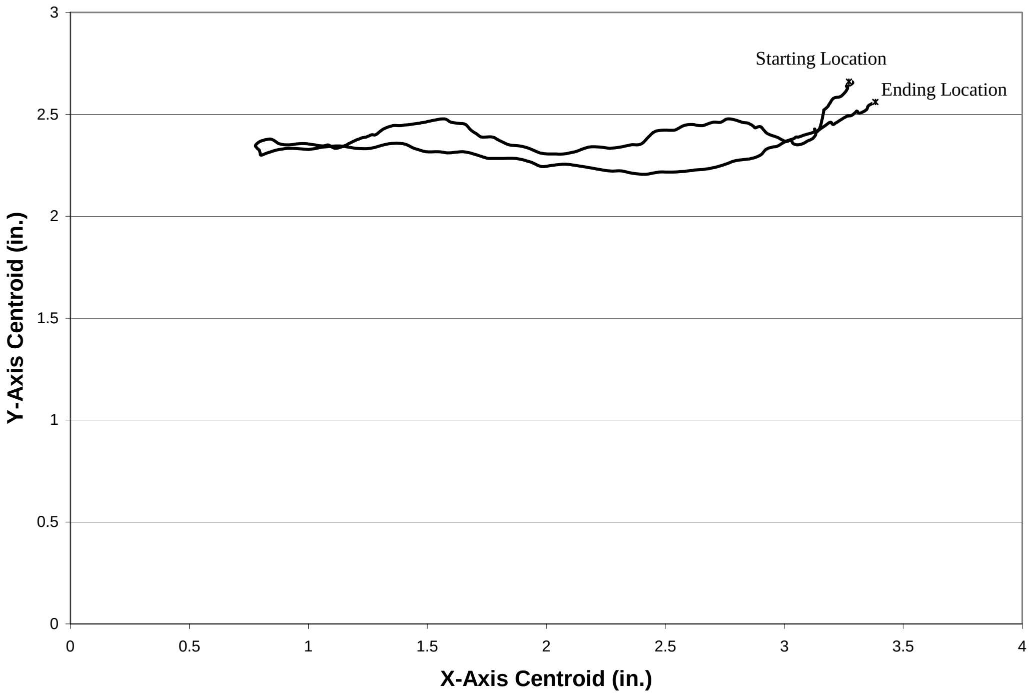


Figure 2.15: Path of Laser Target (From Image Extraction)

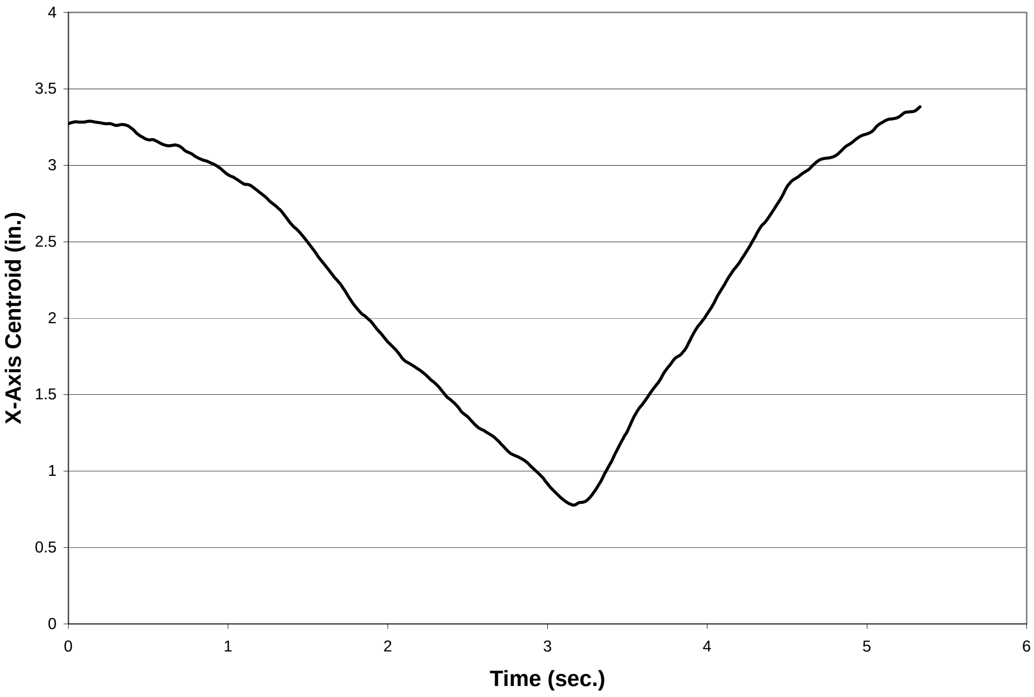


Figure 2.16: Time History of the Horizontal Centroid Location

2.3 Data Acquisition

2.3.1 Data Acquisition versus Data Logging

Two types of data collection are used for acquiring sensor data from bridges. These are datalogging and data acquisition. Data logging techniques are used when acquiring a relatively short duration of data. The datalogging software, controls the A/D conversion and writes the digitized data to memory directly on the datalogger. Once the acquisition has finished, the data must be downloaded to another computer where it can then be processed. A notable disadvantage of this method is that it is not suitable for continuous monitoring. For example, the

Crossbow model AD2012 datalogger (Fig. 2.17) with its eight 12-bit analog inputs and programmable sampling rate from 1 to 1000 Hz is only capable of 540,000 samples before the memory is filled and must be downloaded.

Figure 2.17: Crossbow Model AD2012 12-bit Datalogger (http://www.xbow.com)

2.3.2 Data Acquisition Options

For continuous monitoring applications, data acquisition software and hardware is necessary. Examples of such equipment include National Instruments LabVIEW software and analog-to-digital (A/D) boards as well as Kinemetrics' Quanterra Q330 digital recorder with Antelope software. These systems allow for continuous A/D conversion. The data, once digitized, can be saved to local memory or can be streamed to or from a remote location using a variety of techniques.

In future monitoring efforts on full-size structures, the combined use of a dense array of dynamic sensors and advanced model-free and model-based data analysis and interpretation methods offers a very promising support tool for (1) monitoring the state-of-health of a bridge portfolio, (2) optimum allocation of rehabilitation resources, and (3) evaluation of the effectiveness of the rehabilitation measure on a given bridge [Elgamal et al., 2002]. For these sensor arrays, many more sensors than are currently employed on the composite deck testbed (16 strain gages), discussed in Chapter 3, will be required.

In these applications it is necessary to select scalable hardware capable of reaching channel counts of over 1000. An example of such hardware is the National Instruments PXI-based E series I/O boards, discussed on the second bridge testbed in Chapter 9. Each of these acquisition boards is capable of acquiring signals from up to 64 sensors while maintaining the required sampling rate and necessary time synchronization between channels.

Another important aspect in continuous health monitoring of remote structures is optimizing the operation and reliability of the data acquisition setup. While it may be physically possible for a system to digitize the signal from over a thousand channels at a very high rate, it may not be capable of storing the data to disk, let alone streaming to another location. This issue is explored later in the paper. Ensuring the reliability of the data acquisition software is critical in applications on remote structures. Of concern here is protecting the data acquisition computer from common network problems (such as viruses) as well as preventing system conflicts (for example between the data acquisition and anti-virus software). In addition, the inherent stability of the Windows operating system is a significant concern for continuous long-term monitoring. For these reasons, the use of National Instruments Real-Time operating system and software has been explored (Chapter 9). The inherent advantage with the real time engine software is its ability to act as a standalone operating system, thereby decreasing susceptibility to viruses (which are often written to specifically attack Windows systems) and software conflicts.

2.3.2.1 PC-Based Data Acquisition

In developing pilot structural monitoring systems to test the reliability of the data acquisition, transmission, local/post processing, archival, and dissemination methods, it was decided to focus on standard PC-based data acquisition hardware. These systems (Fig. 2.18) are built around a simple personal computer with a Windows operating system. These PCs each have a single PCI-data acquisition board supporting sixteen to sixty-four analog inputs. The author has successfully employed these systems for remotely monitoring strains on a composite bridge-deck panel system, measuring accelerations and temperatures on a long span suspension bridge (Fig. 2.19), and conducting web-based shake table experiments (Fig. 2.20) [Elgamal et al., 2001 and 2005a, b]. The composite bridge-deck monitoring system is emphasized in this paper and the setup is reported in Chapter 3 while the data recorded and analyzed is examined in Chapters 4 and 8.

It was found that these standard systems are well suited for handling applications involving a limited number (less than 64) of analog sensors of various types. The data acquisition hardware employed with the afore mentioned shake table collected data from a heterogeneous sensor array composed of Macro Sensors DC 750-3000 LVDTs, Endevco model 7705-1000 and PCB model 3701 capacitive accelerometers, and LMI SPR-04 non-contacting displacement transducers. In addition, the same I/O board provided an analog output used to drive the shake table (Fig. 2.21).

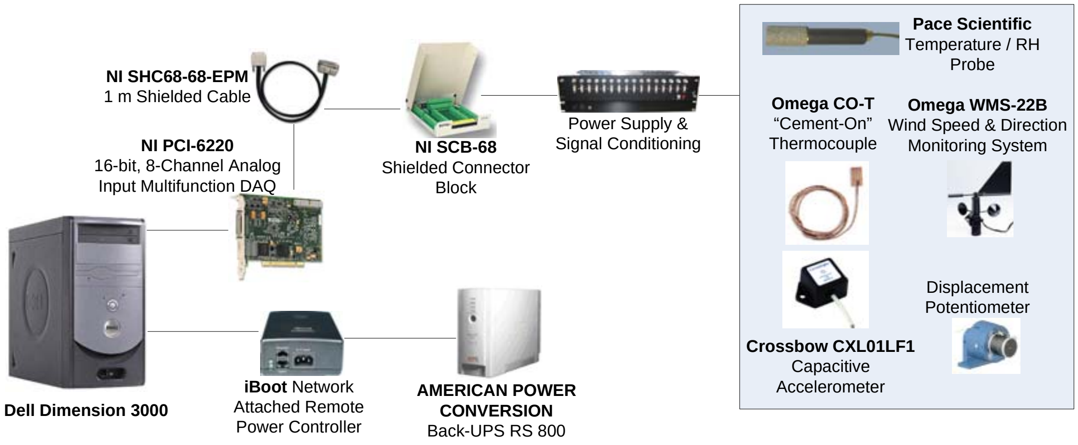
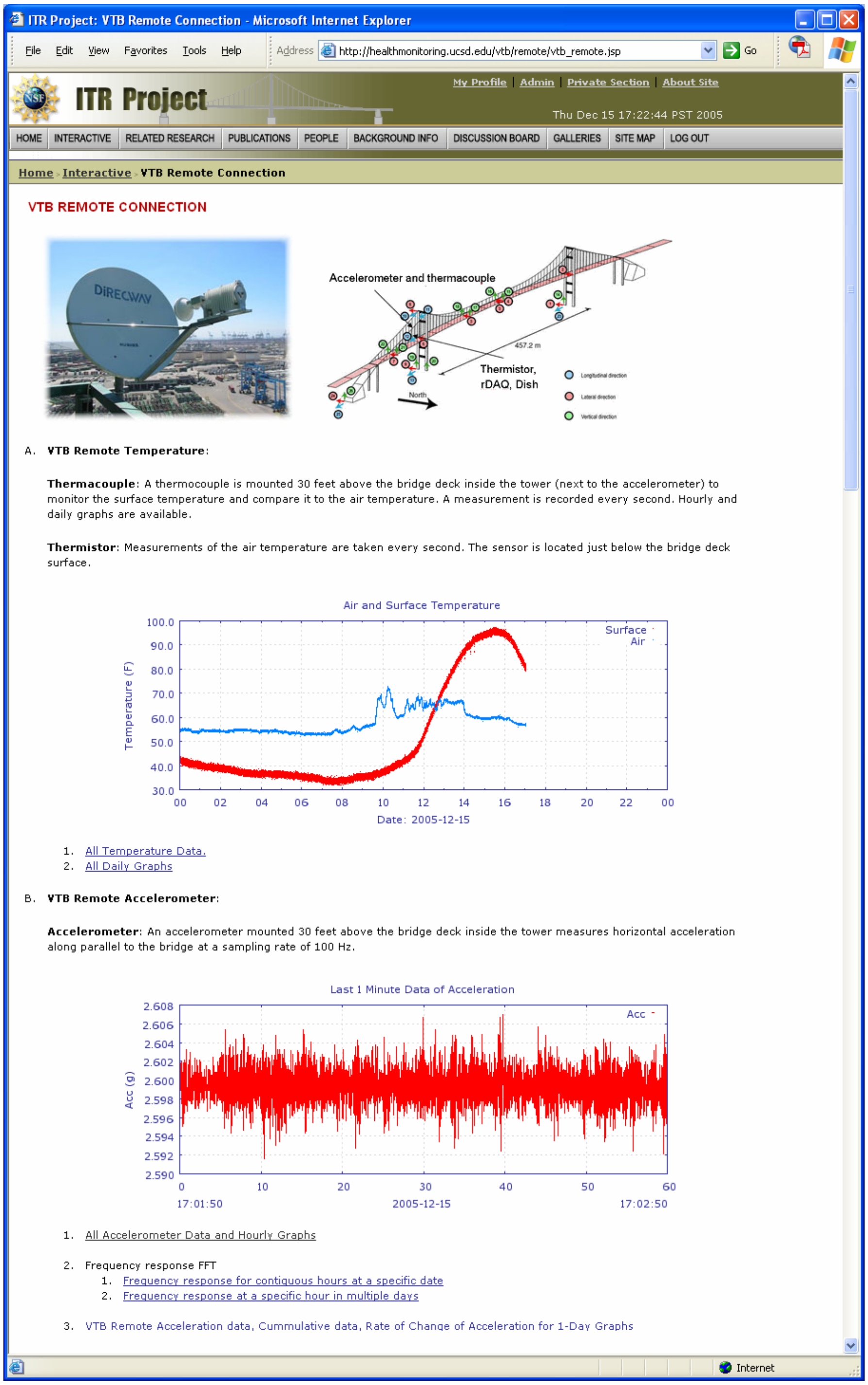


Figure 2.18: PC-Based Monitoring System Coupled with a Sample of Available Motion and Environmental Sensors


ITR Project: VTB Remote Connection - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://healthmonitoring.ucsd.edu/vtb/remote/vtb_remote.jsp
Go
ITR Project
My Profile | Admin | Private Section | About Site
Thu Dec 15 17:22:44 PST 2005
HOME | INTERACTIVE | RELATED RESEARCH | PUBLICATIONS | PEOPLE | BACKGROUND INFO | DISCUSSION BOARD | GALLERIES | SITE MAP | LOG OUT
Home › Interactive › VTB Remote Connection
DIRECWAY
Accelerometer and thermacouple
457.2 m
Thermistor, rDAQ, Dish
North
Longitudinal direction
Lateral direction
Vertical direction

VTB REMOTE CONNECTION

A. **VTB Remote Temperature:**

Thermacouple: A thermocouple is mounted 30 feet above the bridge deck inside the tower (next to the accelerometer) to monitor the surface temperature and compare it to the air temperature. A measurement is recorded every second. Hourly and daily graphs are available.

Thermistor: Measurements of the air temperature are taken every second. The sensor is located just below the bridge deck surface.

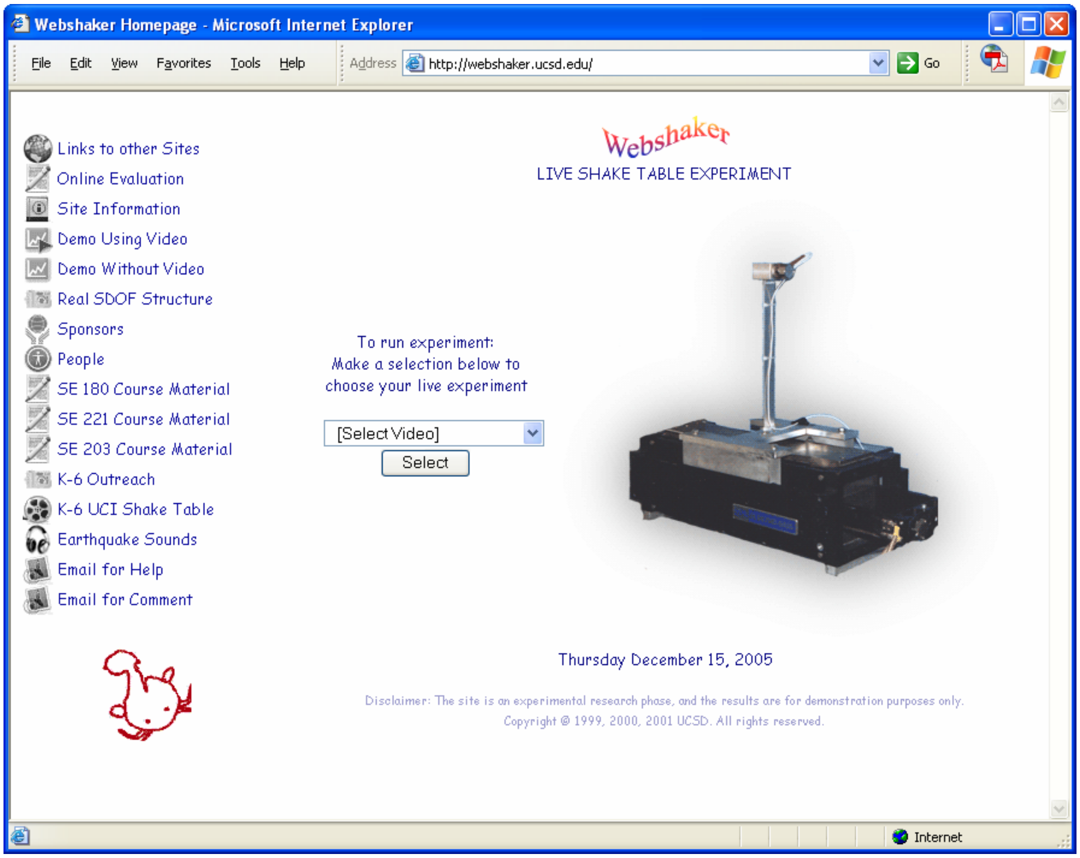
1. All Temperature Data.
2. All Daily Graphs

B. **VTB Remote Accelerometer:**

Accelerometer: An accelerometer mounted 30 feet above the bridge deck inside the tower measures horizontal acceleration along parallel to the bridge at a sampling rate of 100 Hz.

1. All Accelerometer Data and Hourly Graphs
2. Frequency response FFT
   1. Frequency response for contiguous hours at a specific date
   2. Frequency response at a specific hour in multiple days
3. VTB Remote Acceleration data, Cummulative data, Rate of Change of Acceleration for 1-Day Graphs

Figure 2.19: Homepage of the Remote Monitoring System Employed on the Vincent Thomas Suspension Bridge in San Pedro, CA


Webshaker Homepage - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Address
http://webshaker.ucsd.edu/
Go
Internet

Links to other Sites
Online Evaluation
Site Information
Demo Using Video
Demo Without Video
Real SDOF Structure
Sponsors
People
SE 180 Course Material
SE 221 Course Material
SE 203 Course Material
K-6 Outreach
K-6 UCI Shake Table
Earthquake Sounds
Email for Help
Email for Comment

Webshaker
LIVE SHAKE TABLE EXPERIMENT

To run experiment:
Make a selection below to
choose your live experiment

[Select Video]
Select

Thursday December 15, 2005

Disclaimer: The site is an experimental research phase, and the results are for demonstration purposes only.
Copyright © 1999, 2000, 2001 UCSD. All rights reserved.

Figure 2.20: Homepage of Webshaker: a Web-Based Shake Table Facility

(http://webshaker.ucsd.edu)

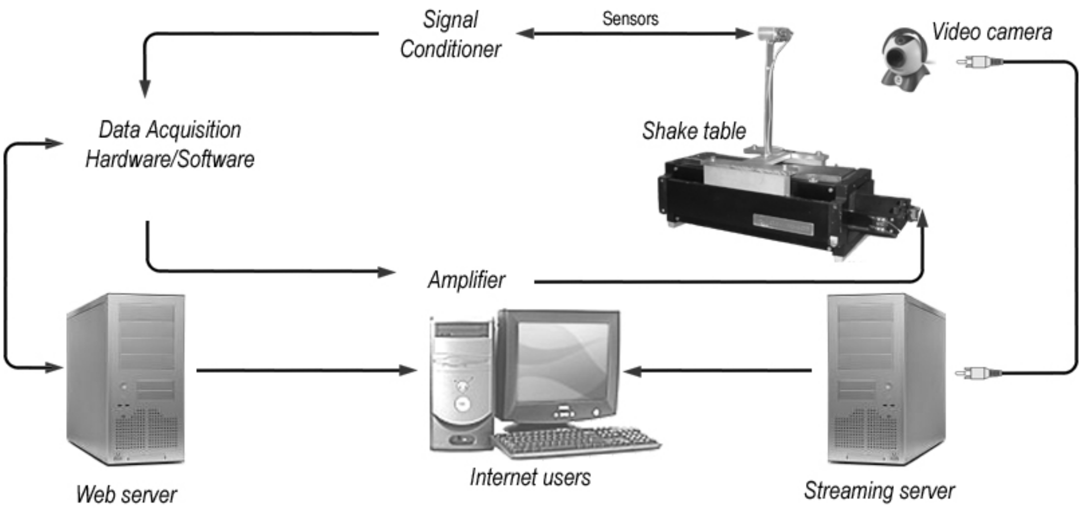


Figure 2.21: Framework of the Webshaker Shake Table Monitoring/Control System

2.3.2.2 Distributed Data Acquisition on Extended Structures

For measuring the low frequency dynamic response of extended structures (like long-span suspension bridges) along with associated environmental properties, it is proposed to use separate generalized distributed data acquisition systems (Fig. 2.22). Unlike the PC-based systems described in the previous section, these are easily expanded to accommodate the large number of sensors required on these massive structures. The distributed data acquisition cluster is capable of acquiring, analyzing, and transmitting over 512 analog input channels. The main advantages of a distributed data acquisition system are: (1) the distance between the sensor and the analog-to-digital converter is minimized, thus substantially reducing interference into the analog signal, and (2) by separating the data acquisition into manageable clusters, local data processing and more importantly data transmission (in light of the limited bandwidth available on most structures) become feasible.

For potential structural monitoring of a long-span suspension bridge [Conte et al., 2006], a sample distributed data acquisition cluster built around a National Instruments PXI-1045 18-slot PXI Chassis with a National Instruments PXI-8186 PXI Embedded Controller (Pentium IV processor, 1 GB of RAM, and 100 GB hard drive) is presented. The National Instruments PXI-1045 18-slot Chassis is then populated with sixteen PXI-6130E Data Acquisition Boards, providing up to 512 channels of 16-bit analog differential input or over 1000 channels of single-ended analog input. To measure the low frequency response of the bridge, it is intended to use a sampling rate of 200 samples per second with this system, much less than the maximum sampling rate of 3.125 kS/s. This combined with the 16-bit analog-to-digital converter makes these boards a reasonable selection. Because of the relatively slow sampling speed, anti-aliasing filters are not included with these systems. In addition, as these distributed data acquisition systems are meant to accommodate a variety of different sensor types, each with their own power requirements, signal conditioning is not be handled through the PXI chassis. The distributed data acquisition system shown in Figure 2.22 is similar to the system employed on the testbed in Chapter 9.

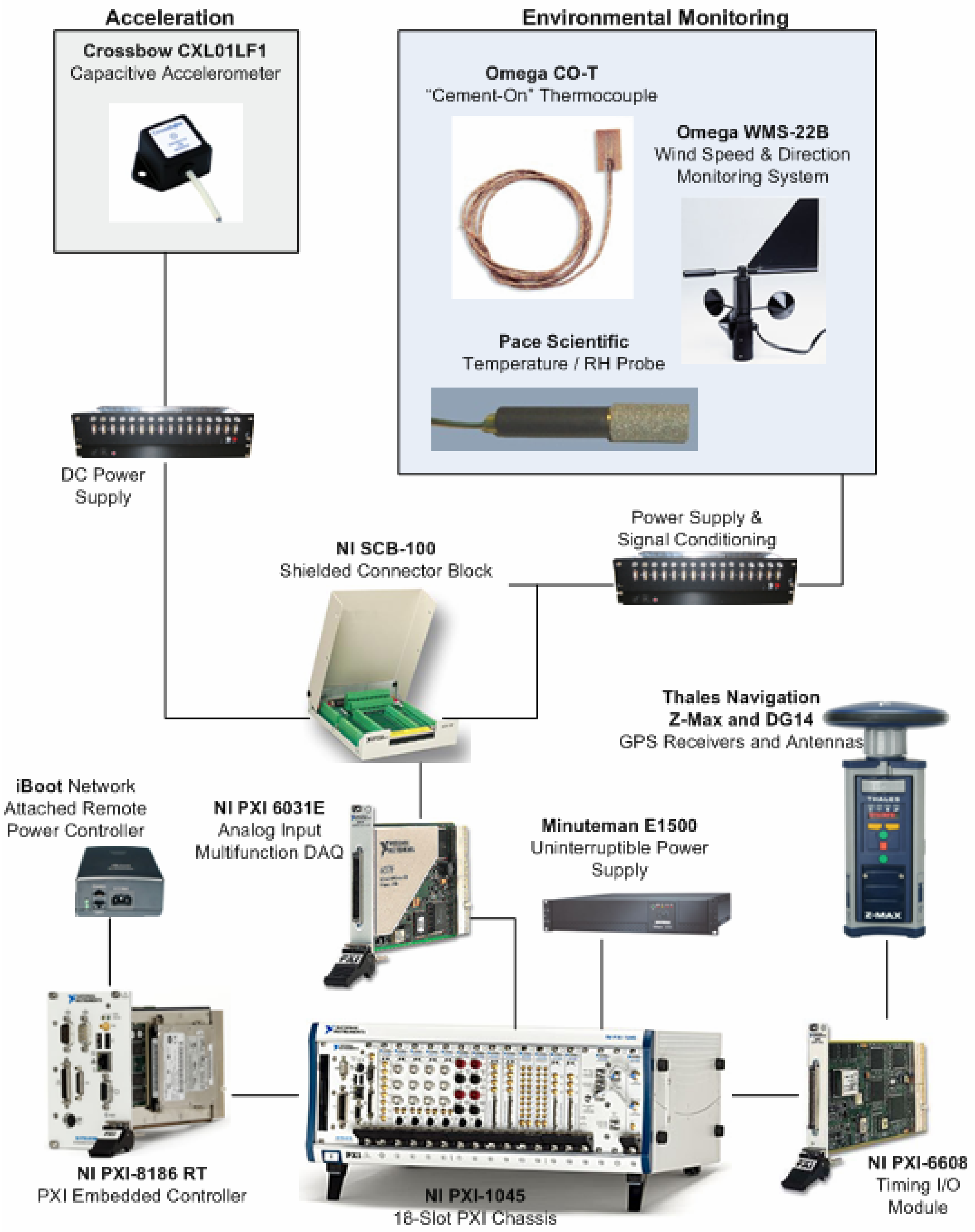


Figure 2.22: Typical Distributed Data Acquisition Cluster

For local damage detection, where sample rates on the order of 1 kHz are necessary, high-speed data acquisition systems are required. As with the conventional data acquisition system previously described, on extended structures multiple high-speed systems will be employed, each collecting data from a cluster of sensors. Each high-speed data acquisition unit (Fig. 2.23) supports 128 channels of simultaneously sampled analog input, with signal conditioning and anti-alias filtering. The core of the high speed acquisition system is a National Instruments PXI-1045 18-slot chassis with a National Instruments PXI-8186 RT Embedded Controller. The PXI-8186 RT utilizes a 2.2 GHz Pentium IV processor with 1.0 GB of RAM and features dual-boot capabilities (Real-Time operating system and Windows XP). For stand-alone bridge monitoring operation, the data acquisition, local data processing, and data transmission codes are embedded on the system so they start automatically when the system boots, thereby requiring no human interaction. Within each PXI-1045 chassis, sixteen PXI-4472 Dynamic Acquisition Signal Modules are situated. The PXI-4472 board has eight analog inputs with 24-bit resolution delta-sigma modulating analog-to-digital converters (that simultaneously digitize the input signals at software programmable rates from DC to 45 kHz). In addition to the acquiring data, the PXI-4472 incorporates for each channel Integrated Electronic Piezoelectric (IEPE) signal conditioning and anti-aliasing filters (both analog and real-time digital filters implemented in the hardware).

Figure 2.24 shows how a typical distributed sensing array would be employed on a suspension bridge. This system is composed of a central distributed data acquisition cluster, 2 high-speed acquisition clusters, and 2 image acquisition/processing clusters (discussed in Section 2.4.4).

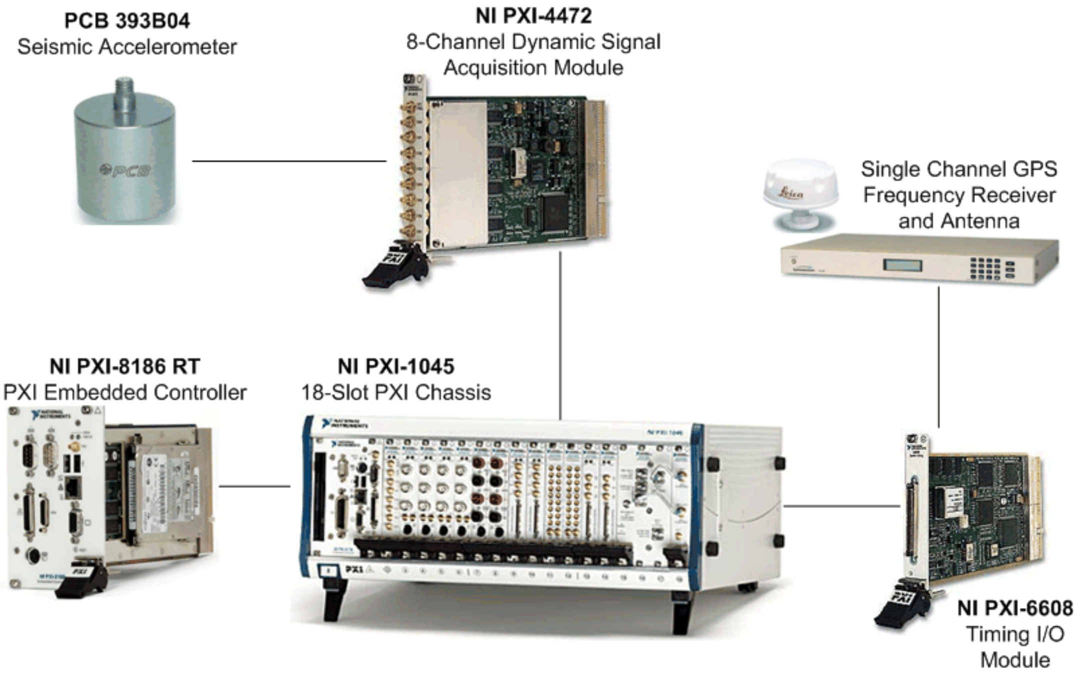


Figure 2.23: High Speed Data Acquisition Cluster

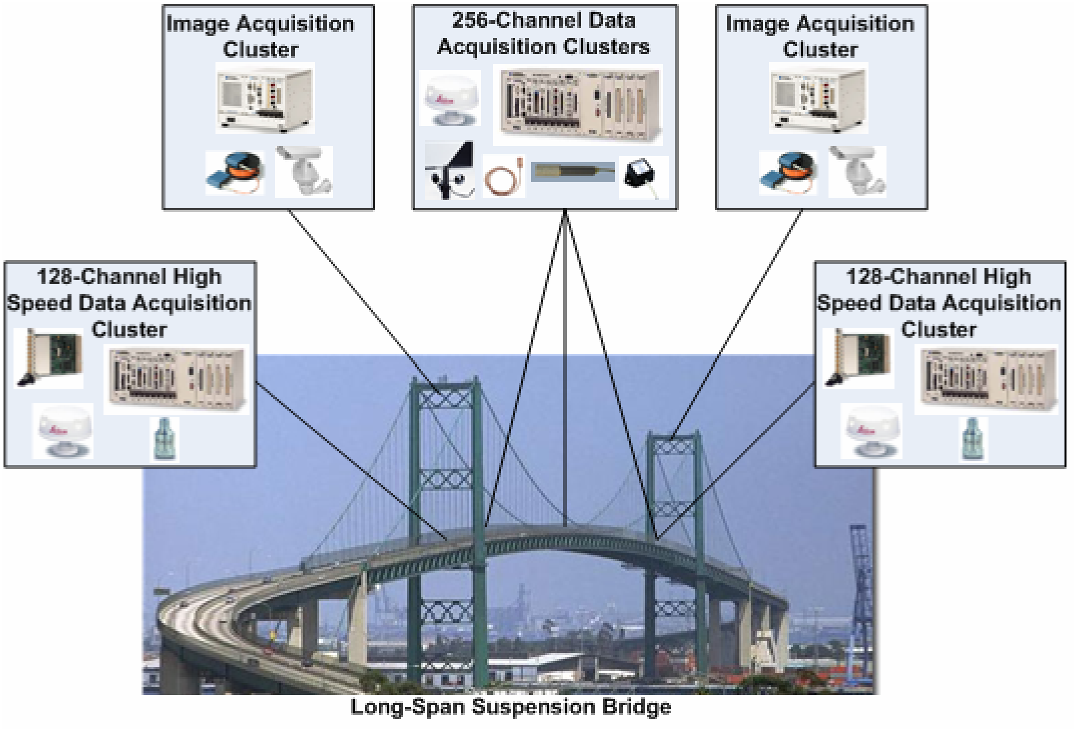


Image Acquisition Cluster
256-Channel Data Acquisition Clusters
Image Acquisition Cluster
128-Channel High Speed Data Acquisition Cluster
128-Channel High Speed Data Acquisition Cluster
Long-Span Suspension Bridge

Figure 2.24: Typical Distributed Sensing Array on Extended Structure

2.4 Image Acquisition

2.4.1 Low-Cost CCD Cameras

The first video acquisition method involves the use of a standard analog camera with a generic video acquisition board. An example is the use of a typical charge-coupled device (CCD) camera (such as the one shown in Figure 2.25) with an Osprey 100 (Fig. 2.26) video capture card (http://www.viewcast.com/products/osprey.html).   Using commercial video-streaming technology (e.g., RealPlayer http://www.realplayer.com or Microsoft

Windows Media Player http://www.microsoft.com/windows/mediaplayer), a live or recorded video signal can be sent to a remote user in near real-time.

While useful (http://webshaker.ucsd.edu [Elgamal et al., 2005a,b]) this method still suffers from problems associated with loss in image quality due to signal compression and difficulty in converting the streamed video into a format suitable for image processing (such as jpg stream) and archiving.

Figure 2.25: Typical Charge-Coupled Device (CCD) Camera

(http://www.finecctv.com/)

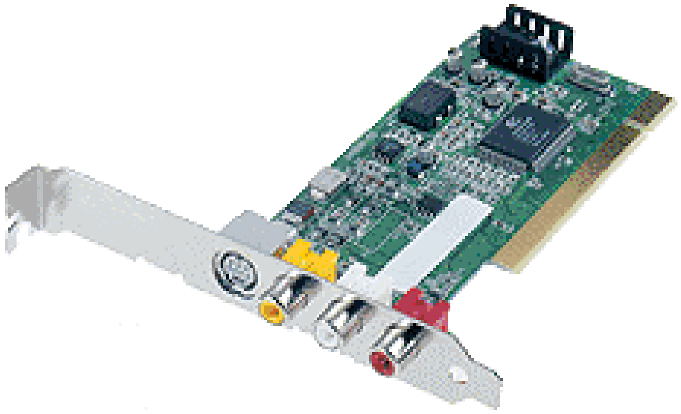


Figure 2.26: Osprey 100 Video Capture Card

(http://www.viewcast.com/products/osprey.html)

The MATLAB Image Acquisition Toolbox (http://www.mathworks.com/products/imaq/) implements an object-oriented approach to image acquisition. Using toolbox functions, an object is created that represents the connection between MATLAB and specific image acquisition devices (such as an Osprey 100 card). Using properties of the object, it is possible to control various aspects of the acquisition process, such as the amount of video data to capture. Once a connection to a device is established, image data is acquired by executing a trigger. In the toolbox, all image acquisition is initiated by a trigger. The toolbox supports several types of triggers that allows for controlling when an acquisition takes place. For example, using hardware triggers one can synchronize an acquisition with an external device. Similar to the National Instruments setup detailed in Section 2.4.3, the advantages are the ability to better control the acquisition rate, improved time-stamping, and formatting output.

2.4.2 Network Cameras

A Network Camera is a camera with a built in Internet web server, which is connected directly to a 10/100 Mbit Ethernet network. Live images, at up to 30 frames per second, are viewed directly from a web browser. An example is the AXIS 2120 (Fig. 2.27) Network Camera (http://www.axis.com/products/cam_2120), currently employed on the UCSD composite decks testbed (details can be found in Chapter 3), which is as a stand-alone camera, with a built in web server. In the basic mode of operation you use the URL of the camera to access the images and the internal web pages of the camera. Operating in this mode, there is no need for an additional PC as the unit acts as its own standalone web server. In addition, the camera can transmit images to a remote computer via an FTP connection.

Figure 2.27: Axis 2120 Network Camera (http://www.axis.com)

2.4.3 Image Acquisition Boards

National Instruments offers several PCI and PXI image acquisition boards (http://sine.ni.com/apps/we/nioc.vp?cid=1292&lang=US). These boards connect to any color or monochrome camera (similar to the Osprey board discussed previously) through either S-video or BNC cable. However, these boards work with National Instruments LabVIEW software or C++ for controlling the acquisition and output formats. An example (Figure 2.28) is the NI 1411 board (http://sine.ni.com/apps/we/nioc.vp?cid=11356&lang=US) which is capable of acquiring color images at up to 30 frames per second. A distinct advantage with this system over the previous methods is the ability to accurately time stamp the acquired images, especially when connected to the data acquisition computer and images are acquired simultaneously with sensor data.

Figure 2.28: National Instruments 1411 Image Acquisition Board

(http://sine.ni.com/nips/cds/view/p/lang/en/nid/11356)

2.4.4 Image Acquisition Cluster for Real-Time Feature Extraction

In full-scale structural monitoring, image acquisition/processing systems will allow for analyzing traffic flow on a bridge as well as monitoring low frequency dynamic response of select bridge subcomponents. In addition to performing image acquisition, these systems will perform local processing on the recorded images for feature extraction. The information derived from the image analysis algorithms along with segments of the compressed video can then be streamed for archival.

A sample image acquisition cluster (Fig. 2.29) may be composed of a National Instruments PXI-1031 4-slot Chassis with a National Instruments PXI-8186 PXI Embedded Controller (Pentium IV processor and 1 GB of RAM) and a National Instruments PXI-8252 IEEE 1394 Interface Board. Each cluster can accommodate up to three fire wire (IEEE 1394b) cameras, such as Sony's DFW-X710 IEEE 1394 color digital cameras. These cameras incorporate a 1/3 type 800,000 pixel CCD, partial scanning with 16x16 zones, selectable output modes, and an external trigger function with an exposure range from 1/100,000 to 17.5 seconds (allowing for the clear capture of fast moving objects or still images in low light environments). Fire wire cameras allow for longer cable runs than are possible with analog or other digital cameras. By employing a Prosilica FB1394B Bridge Set to convert from IEEE 1394b (fire wire) to IEEE 1304a (optical fiber) back to IEEE 1394b, cable runs of over 1 mile are possible with negligible signal degradation. The bulk of the cable run is made using the IEEE 1304a cable. For monitoring the flow of traffic and the low frequency response of a bridge, a medium speed frame rate of 15 pictures per second could be used, thereby allowing for real time processing of the images from cameras attached to each of the acquisition clusters.

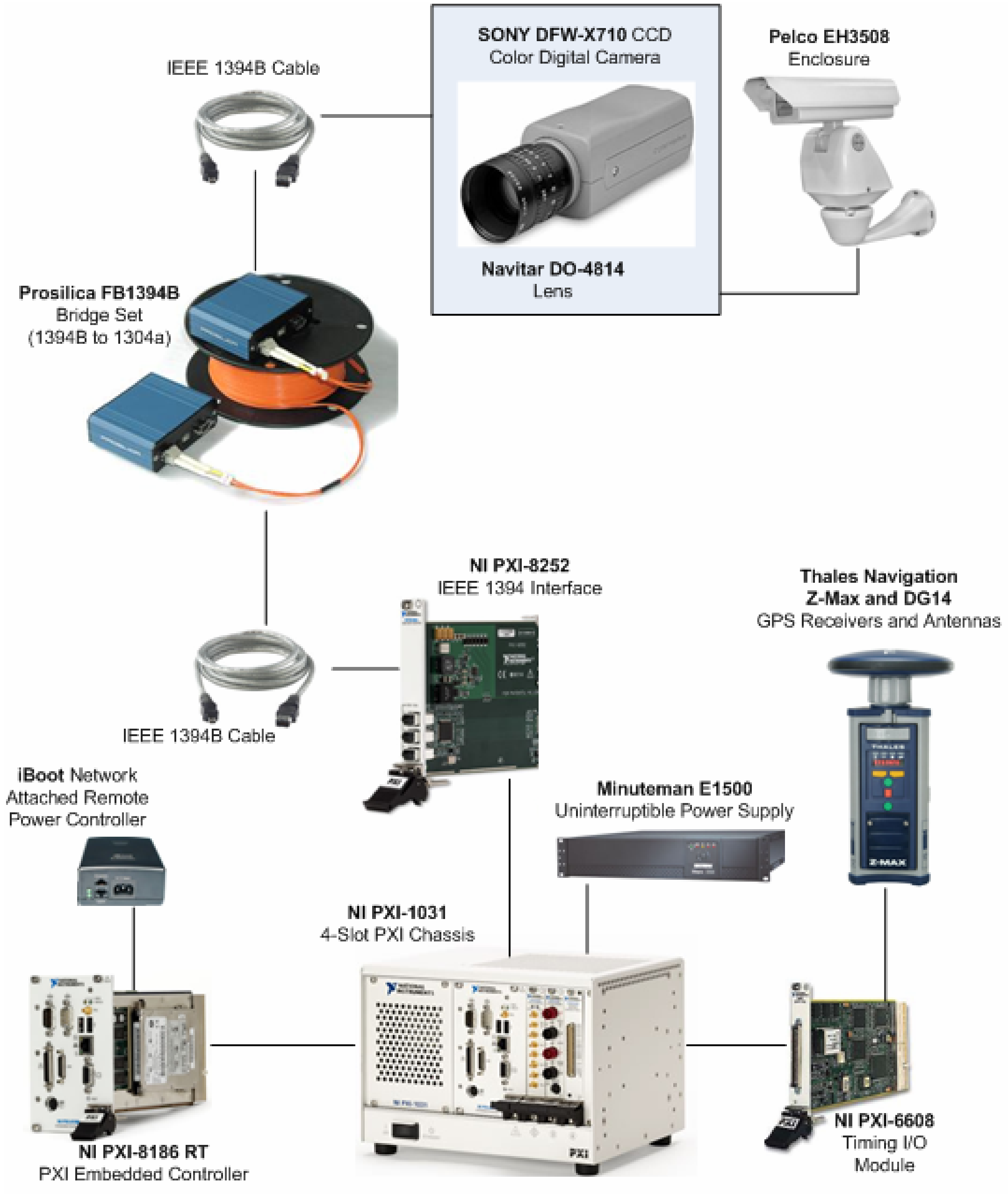


Figure 2.29: Typical Image Acquisition Cluster

2.5 Temporal Synchronization of Sensor Data

2.5.1 Synchronizing Analog Sensors

There exists an obvious need to synchronize all of the data acquired from the collection of sensors dispersed on a structure. This issue becomes a concern when calculating mode shapes based on recorded accelerations or strains (collected by one or more data acquisition clusters) as it is critical the data be synchronized or the calculations of the phase angle, necessary for determining the relative sign of the shape, will be greatly compromised. Another example (as examined in Chapter 4) involves an array of strain gages installed on a series of composite bridge-deck panels situated in a roadway on the UCSD campus and correlating these measurements with video recorded of passing traffic.

For this problem, the simplest cases are synchronizing the data collected from sensors connected to a single I/O device or from multiple I/O devices installed in a single acquisition cluster. For the single device case, the timing is controlled by the scan clock on the acquisition hardware through the data acquisition software. For multiple devices, this can be slightly more complicated. With PXI-based hardware, the timing is handled by the PXI chassis clock. If multiple PCI boards are being used there is no single clock responsible for synchronizing the acquisition from all of the boards. This means any acquisition taken will only be accurate to within one time step of the acquisition rate. However, if only moderate accuracy is required, the data may be oversampled and then resampled to provide sufficient synchronization. However, this only works with hardware installed in a single acquisition cluster. The synchronization of data collected from multiple systems is addressed in the following sections.

2.5.2 Synchronizing Multiple Data Acquisition Computers

Synchronizing the data collected by multiple data acquisition clusters or sensor and video data (either from an image acquisition cluster or network camera) requires additional considerations. Three approaches to this problem are discussed in the following sections.

2.5.2.1 National Institute of Standards and Technology (NIST) Internet Time Service

The NIST Internet Time Service (http://www.boulder.nist.gov/timefreq/service/its.htm) allows users to synchronize computer clocks via the Internet. The service responds to time requests from any Internet client in several formats including the DAYTIME, TIME, and NTP protocols.

The Network Time Protocol (NTP) is the most commonly used Internet time protocol, and the one that provides the best performance. Large computers and workstations often include NTP software with their operating systems. The client software runs continuously as a background task that periodically gets updates from one or more servers. The client software ignores responses from servers that appear to be sending the wrong time, and averages the results from those that appear to be correct.

Many of the available NTP software clients for personal computers do not perform any averaging. Instead, they make a single timing request to a signal server (just like a Daytime or Time client) and then use this information to set their computer's clock. The proper name for this type of client is SNTP (Simple Network Time Protocol).

The NIST servers listen for a NTP request on port 123, and respond by sending a udp/ip data packet in the NTP format. The data packet includes a 64-bit timestamp containing the time in UTC seconds since January 1, 1900 with a resolution of 200 ps [http://www.boulder.nist.gov/timefreq/service/its.htm].

Commercial software is available for making use of NIST Internet Time Service for updating the system clock in a data/video acquisition computer. For example, Absolute Time Server (http://www.adjusttime.com/atcs.php), a full-featured time server, works as a WinNT service and is fully RFC-868 and RFC-2030 compatible provides network level time synchronization on Windows 98, ME, NT4.0, 2000, and XP platforms. This software acts as a background process, has very low system resources consumption, and can be started before user logon. Absolute Time Server works by periodically checking and synchronizing the computer's system clock with NIST time servers.

Problems with this program are associated with possible network latency. Consequently, the system clock can differ from the NIST clock by as much as several seconds. In addition, programs which make use of the computer's system clock, like LabVIEW when time-stamping recorded data, have problems when the system clock updates in the middle of an acquisition. Most times, the LabVIEW software ignores the updated time. For the time change to take affect in the LabVIEW software, the data acquisition program must be manually stopped and restarted.

A more attractive option for time synchronization is to incorporate NIST/NTP synchronization directly into the data acquisition process. In LabVIEW, which has been heavily employed by the author in all of the monitoring systems, it is possible to make use of the included TCP functions to establish a connection with an NIST time server and using a single timing request on the signal server read the necessary information to reset the computer's system clock. A sample of this protocol [Gatling, 2005] is shown in Figure 2.30-2.32. The basic steps involved in this process are:

1. Check the hour on the system clock for the first data point coming out of the current buffered data set.
2. If the hour is the same as from the previous iteration, no action is taken.
3. Else, a connection with a specified NIST Time Server is established and a new time is read. The connection is closed and the new time is used to reset the system clock.

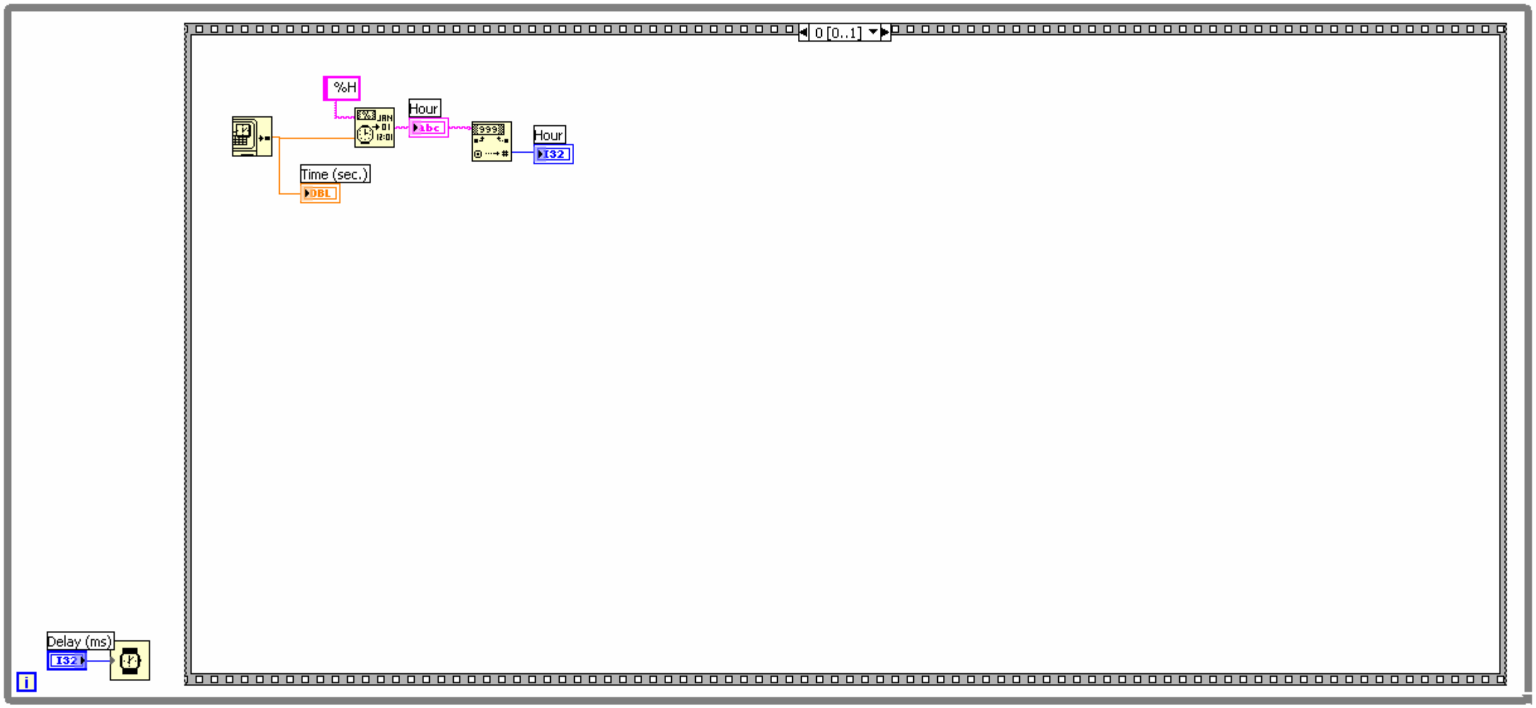


Figure 2.30: LabVIEW Code for Resetting System Clock, Step 1 – Check Current Hour of System Clock

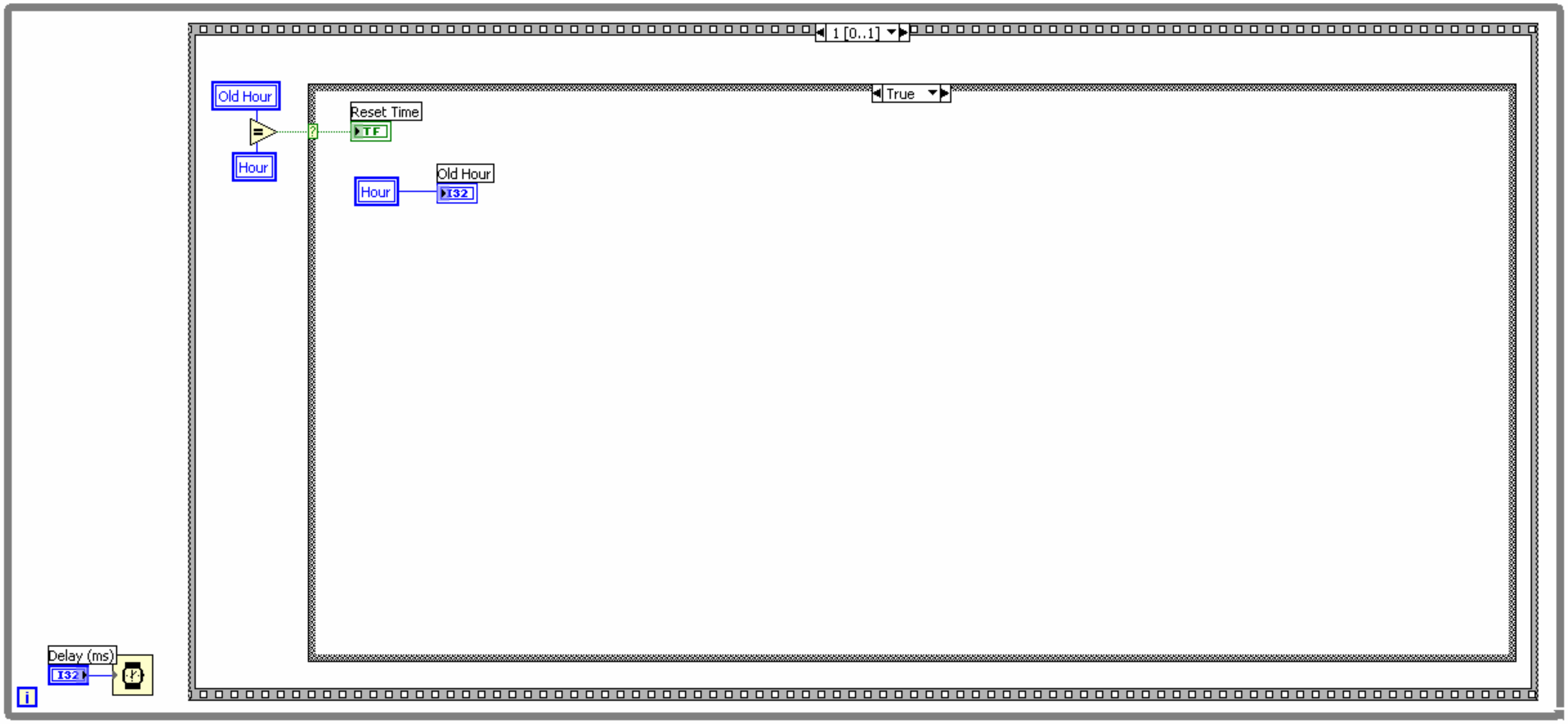


Figure 2.31: LabVIEW Code for Resetting System Clock, Step 2 - If the Current Hour Is the Same As In Previous Iteration, Then No Action is Taken

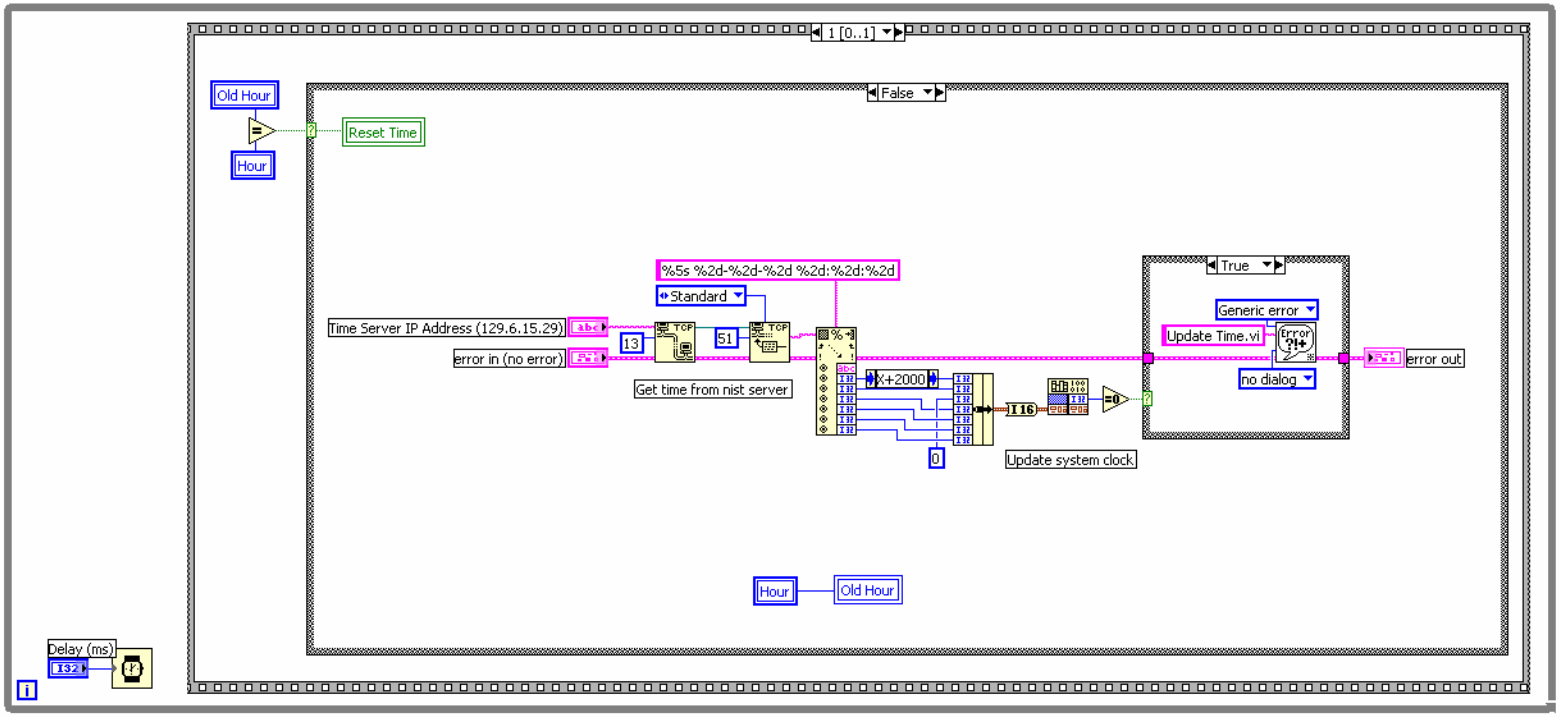


Figure 2.32: LabVIEW Code for Resetting System Clock, Step 3 - If the Hour Has Changed, Then Establish A Connection with Specified NIST Time Server and Use the Read Time to Reset the System Clock

Similarly to the Absolute Time Server software, this method is also susceptible to problems associated with possible network latency. Consequently, the system clock can still differ from the NIST clock by as much as several seconds. However, there is not a problem with the data acquisition software recognizing the updated time.

2.5.2.2 GPS Synchronization

The most reliable way of updating the system clock on a computer is to make use of a GPS clock. The price of GPS equipment has finally dropped to the point where it is affordable. Receiver/decoder modules are available for under $100 and antennas for about $50. Complete hacker-friendly units with antenna, receiver, decoder, and detailed documentation can be had for $400 or less (http://www.gpsclock.com).

With only the most basic GPS serial output, typical accuracy is only approximately one-second. To improve on this accuracy, more advanced time and frequency receivers should be utilized. These improved hardware components can bring the resolution down to the 100 nanosecond level. For future research on large-scale structures, in particular when synchronization is required on multiple remote computers, this option is being explored.

In future systems, Global Positioning System (GPS) will serve a dual role in the envisioned structural health monitoring network. Using a GPS receiver, the coded satellite signals may be utilized for computing time and displacements at the location of the receiver. The output from the GPS receiver along with a National Instruments PXI-6608 Timing I/O Module allows the scan clock on any of the PXI chassis to be synchronized with the GPS signal. When possible, only one acquisition computer in a particular segment of the bridge needs to be connected through a timing I/O module to a GPS receiver. Neighboring acquisition computers, located within approximately 100 meters, can be synchronized to this particular PXI chassis using the PXI trigger bus. With this system, we do not experience either of the problems associated with the NIST time servers – namely network latency and conflicts with the data acquisition are no longer a problem. The biggest problems associated with GPS clocks are the higher costs and complex installations.

2.6 Data Transmission

As mentioned in Chapter 1, early efforts in structural monitoring involved the use of data logging hardware, which required visiting the installation site to manually retrieve the recorded data. This method of data collection does not lend itself to the detection of rapidly occurring damage resulting from events like earthquakes and acts of terrorism. There exists an obvious need to be able to move the acquired data from a remote bridge location to a more convenient location to both verify proper operation of the data acquisition system and to perform post processing operations. The rapid advances and expansion of broadband Internet has made possible the periodic and even continuous transmission of data. Several methods are well suited for moving these large bodies of data, including: TCP/IP, FTP, and mapped network drives. The advantages and disadvantage of each of the methods are summarized in the following sections.

2.6.1 Data Transmission Protocols

2.6.1.1 Mapped Network Drive

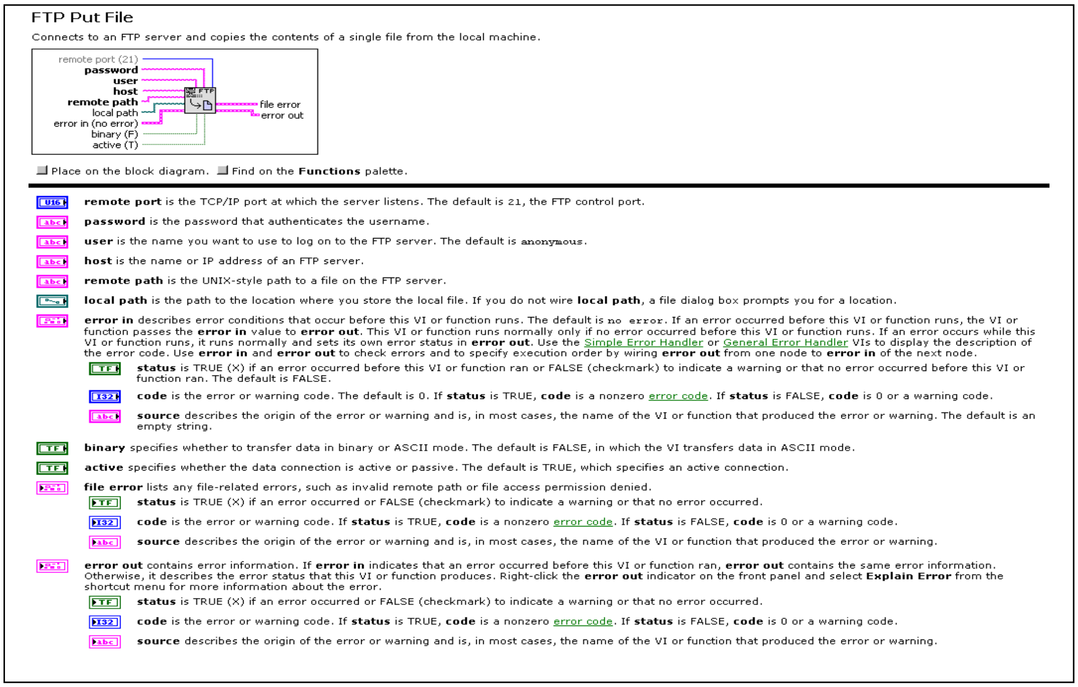
Mapped network drives are the easiest way to save data over a network. In principal, this is very similar to saving a file on a shared folder on another computer (for example, by using Microsoft’s network neighborhood). This method was employed for transmitting data recorded by the 2-channel peak hourly strain system deployed on the UCSD composite bridge-deck panels. The data, once acquired, is placed into five-second packets and saved over the Internet to a mapped network drive. Details of this system are provided in Chapter 3. From this exercise, several limitations with mapped network drives were encountered. First, and most importantly, this transmission method is only suited for local transmission. For remote monitoring, where the acquisition and archiving computers are on separate networks, this transmission method is unavailable. In addition, this method only works on extremely stable networks. If a network outage is experienced, no data can be transmitted which will crash the data transmission program. This can be extremely dangerous if this transmission process is incorporated into the data acquisition program.

2.6.1.2 File Transfer Protocol (FTP)

File Transfer Protocol (FTP) provides a means for transferring individual files, one at a time, from one computer to another over the Internet. There are two

FTP modes: normal (or active) and passive (PASV). In both modes, the user allocates two ports for file transfer, which are the control and data channels. For security reasons, the passive method is preferred as the data channel is established by the external user as where in normal mode it is initiated by the FTP server. In normal mode, the FTP server creates the outbound connection to the external user thereby creating a problem for security and firewall software (http://war.jgaa.com/ftp/?cmd=show_page&ID=ftp_pasv).

To send data using FTP protocol from a remote location, the connection with the FTP server is established, the data is sent, and the connection is then severed (Fig. 2.33). This method is well-suited for transferring large files like compressed periodic sensor data; however, it does not work as well for continuously streaming data. The reason is for each transfer operation, the connection with the remote server must be established before the data packet can be sent. Constantly connecting and disconnecting from the FTP server is a wasteful step. Consequently, a better transfer method using TCP/IP protocol is explored in the following section.



FTP Put File

Connects to an FTP server and copies the contents of a single file from the local machine.

Place on the block diagram. Find on the **Functions** palette.

remote port is the TCP/IP port at which the server listens. The default is 21, the FTP control port.

password is the password that authenticates the username.

user is the name you want to use to log on to the FTP server. The default is anonymous.

host is the name or IP address of an FTP server.

remote path is the UNIX-style path to a file on the FTP server.

local path is the path to the location where you store the local file. If you do not wire **local path**, a file dialog box prompts you for a location.

error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the **error in** value to **error out**. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in **error out**. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring **error out** from one node to **error in** of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

binary specifies whether to transfer data in binary or ASCII mode. The default is FALSE, in which the VI transfers data in ASCII mode.

active specifies whether the data connection is active or passive. The default is TRUE, which specifies an active connection.

file error lists any file-related errors, such as invalid remote path or file access permission denied.

- **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning.

error out contains error information. If **error in** indicates that an error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

- **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning.

Figure 2.33: LabVIEW FTP Put File Command

2.6.1.3 TCP/IP

With Transmission Control Protocol / Internet Protocol (TCP/IP) it is possible to communicate over single networks or interconnected networks (Internet), including those separated by large geographical distances. TCP/IP routes data from one network or Internet computer to another, and because it is available on most computers, it can transfer information between diverse systems.

As with all other communications protocols, TCP/IP is composed of layers (http://www.yale.edu/pclt/COMM/TCPIP.HTM):

- IP - is responsible for moving packet of data from node to node. IP forwards each packet based on a four byte destination address (the IP number).
- TCP - is responsible for verifying the correct delivery of data from client to server. TCP adds support to detect errors or lost data and to trigger retransmission until the data is correctly and completely received.
- Sockets - is a name given to the package of subroutines that provide access to TCP/IP on most systems.

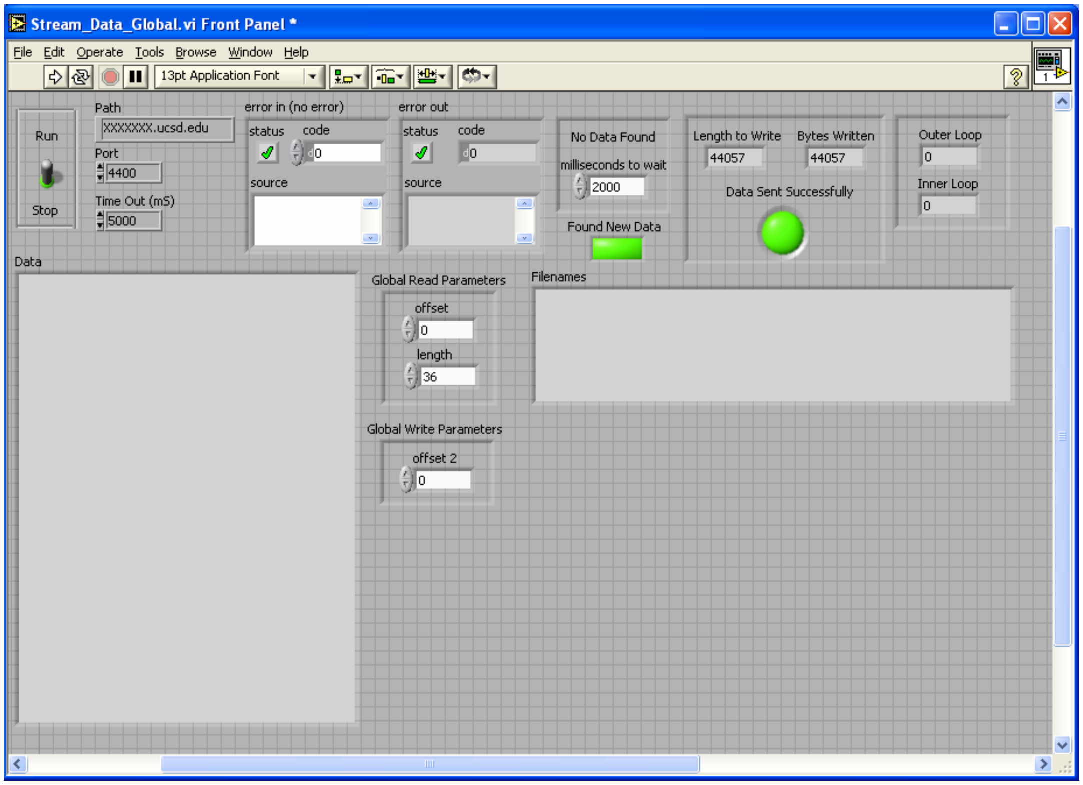
With a TCP/IP connection, it is not necessary to save any of the acquired data to the local disk. Instead, the data can be streamed directly from the system memory. In addition to providing faster data streaming rates, this method also reduces the requirements for the data acquisition computer. The TCP/IP operation can be performed as part of the LabVIEW data acquisition program. In this scheme, the data is acquired from the buffer, placed in packets, and sent via the TCP/IP connection. This method is currently being employed to stream the 16 channels of data from the UCSD composite deck sections (http://healthmonitoring.ucsd.edu/streamdata.jsp) described in detail in Chapter 3.

The basic framework of sending data via a TCP/IP connection is outlined in the following sample code written in National Instruments LabVIEW software (http://www.ni.com/labview/). This program (Fig. 2.34) was written to run in the background and transmit data stored locally by a data acquisition program. Each packet of data acquired by the acquisition program was written to the hard drive and the filename was appended to a LabVIEW global variable. This program relies on two while loops (Fig. 2.35). The first loop, referred to as the outer loop, is responsible for opening and closing the TCP/IP connection. The second (inner) loop controls and monitors the data transmission and deletes the sent data from the local hard drive.

The first step in this program is to establish the TCP/IP connection labeled 1a in Figure 2.35. To accomplish this, a path (IP address), port number of the TCP/IP listener, and a time out are specified (Fig's. 2.34 and 2.35) in the outer loop. Next (step 1b in Fig. 2.35), the program opens the global variable "Data To Be Sent" and checks the file size to determine if there is new data to be transmitted. If no data is found (1c in Fig. 2.35), the program waits the number of milliseconds specified by "Seconds to Wait" before checking again for new data.

If one or more data files are found, the program opens up the first of these files (2a, Fig. 2.36) and determines the size of the data set in bits (2b, Fig. 2.36). Next, the program sends the data using the TCP/IP connection (2c, Fig. 2.36) and checks the amount of data sent (2d, Fig. 2.36). The amount of data to be sent is compared to that which was sent (3a, Fig. 2.37). If these numbers are identical, the data files is deleted (3b) and the filenames is removed from the global variable containing the list of data still to be sent (3c, Fig. 2.38). If a data transmission error is encountered as indicated by the size of the data not equaling what was sent (4a, Fig. 2.39) or a time out during the TCP/IP write, the inner loop is closed (4b, Fig.

2.39) and the TCP/IP connection is closed (4c, Fig. 2.39) and then reset (4d, Fig. 2.39) in the outer loop.



Path: XXXXXXX.ucsd.edu
Port: 4400
Time Out (mS): 5000
error in (no error): status ✓ code 0
error out: status ✓ code 0
milliseconds to wait: 2000
Length to Write: 44057
Bytes Written: 44057
Outer Loop: 0
Inner Loop: 0
offset: 0
length: 36
offset 2: 0

Figure 2.34: LabVIEW Front Panel of TCP/IP Data Streaming Program

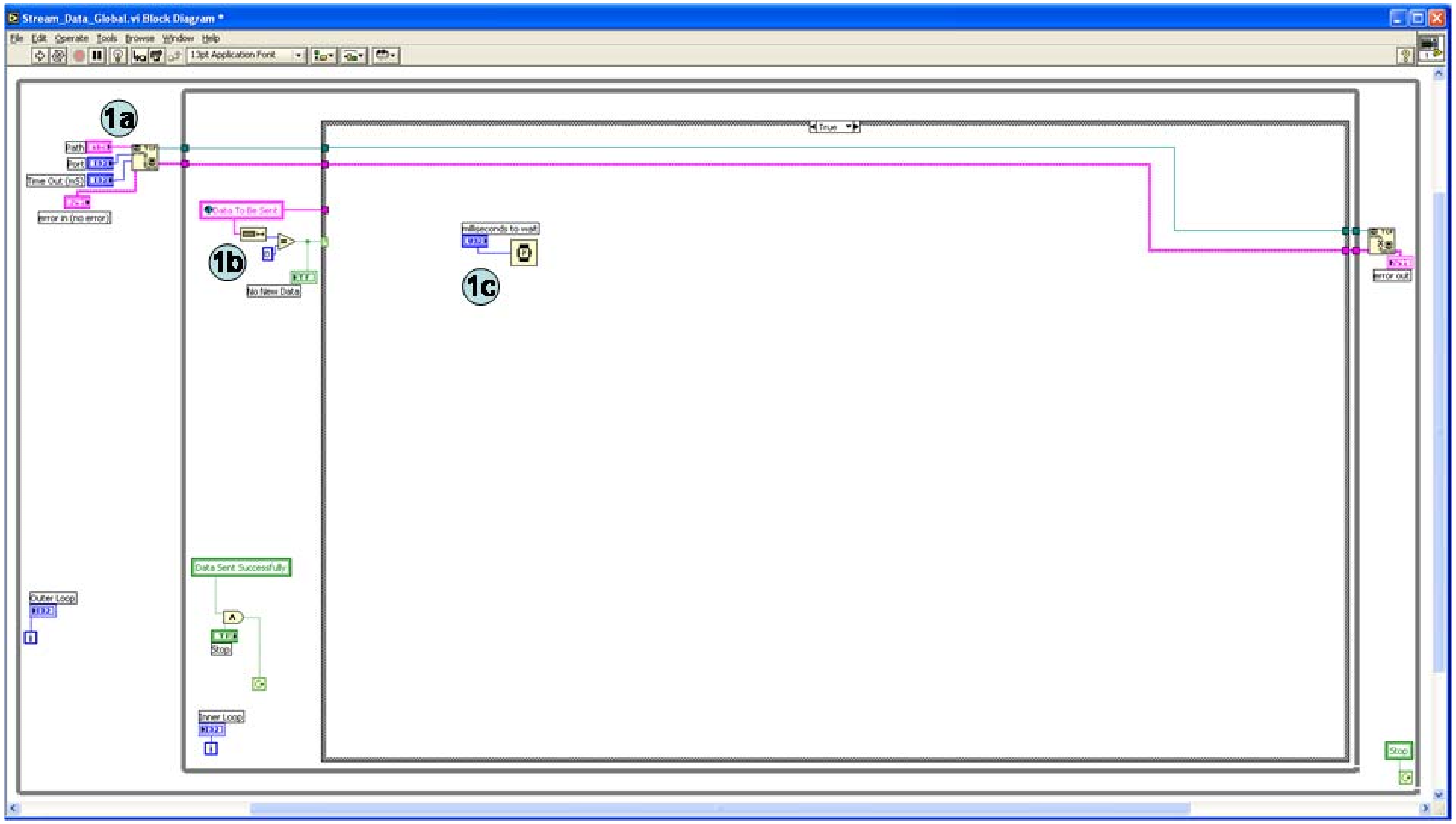


Figure 2.35: Portion of LabVIEW Data Streaming Program Responsible for Forming TCP/IP Connection, Checking for New Data, and Waiting if No New Data is Found

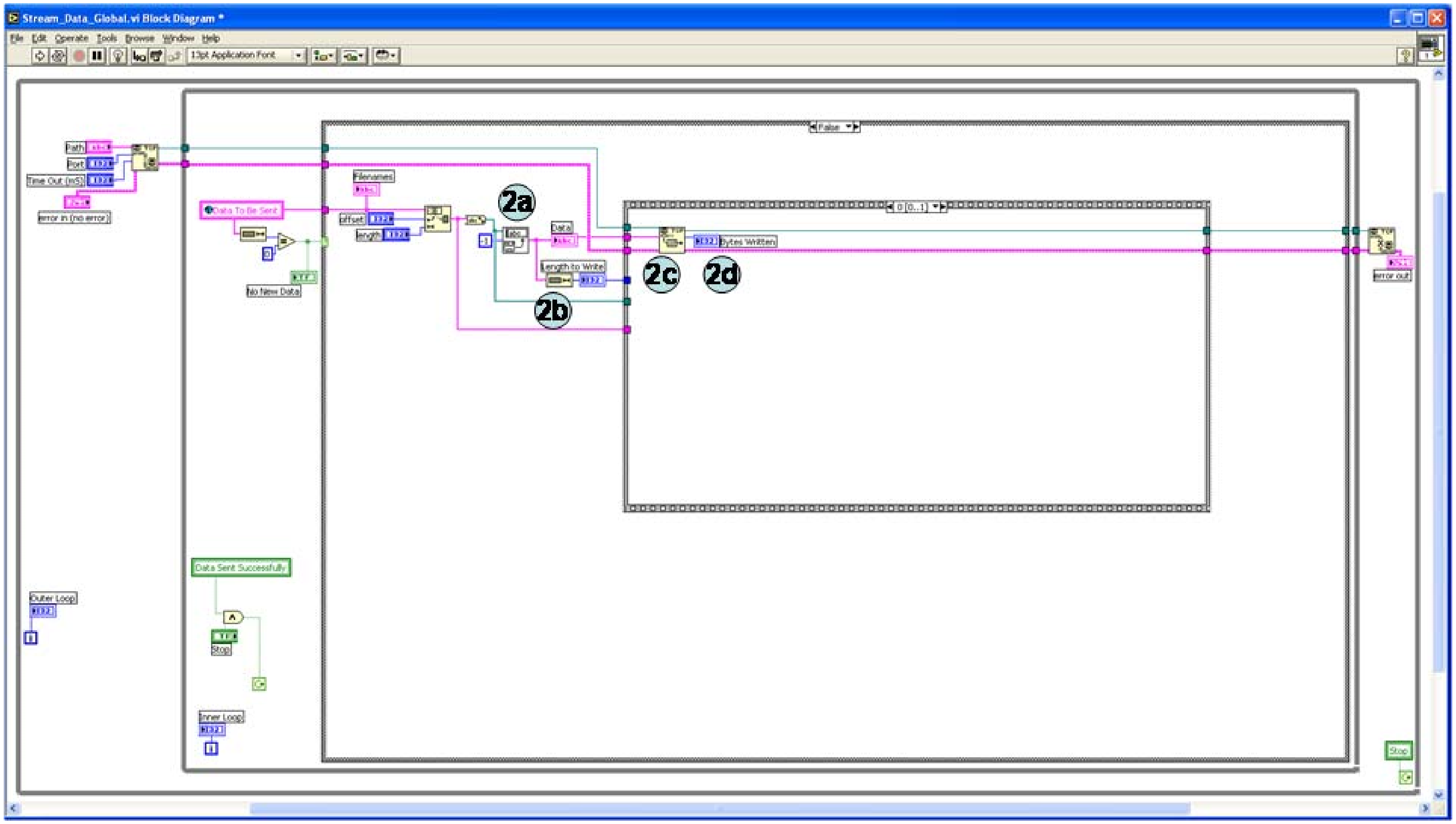


Figure 2.36: Portion of LabVIEW Data Streaming Program Responsible for Open Data File, Checking File Size, Transmitting Data, and Checking Amount of Data Transmitted

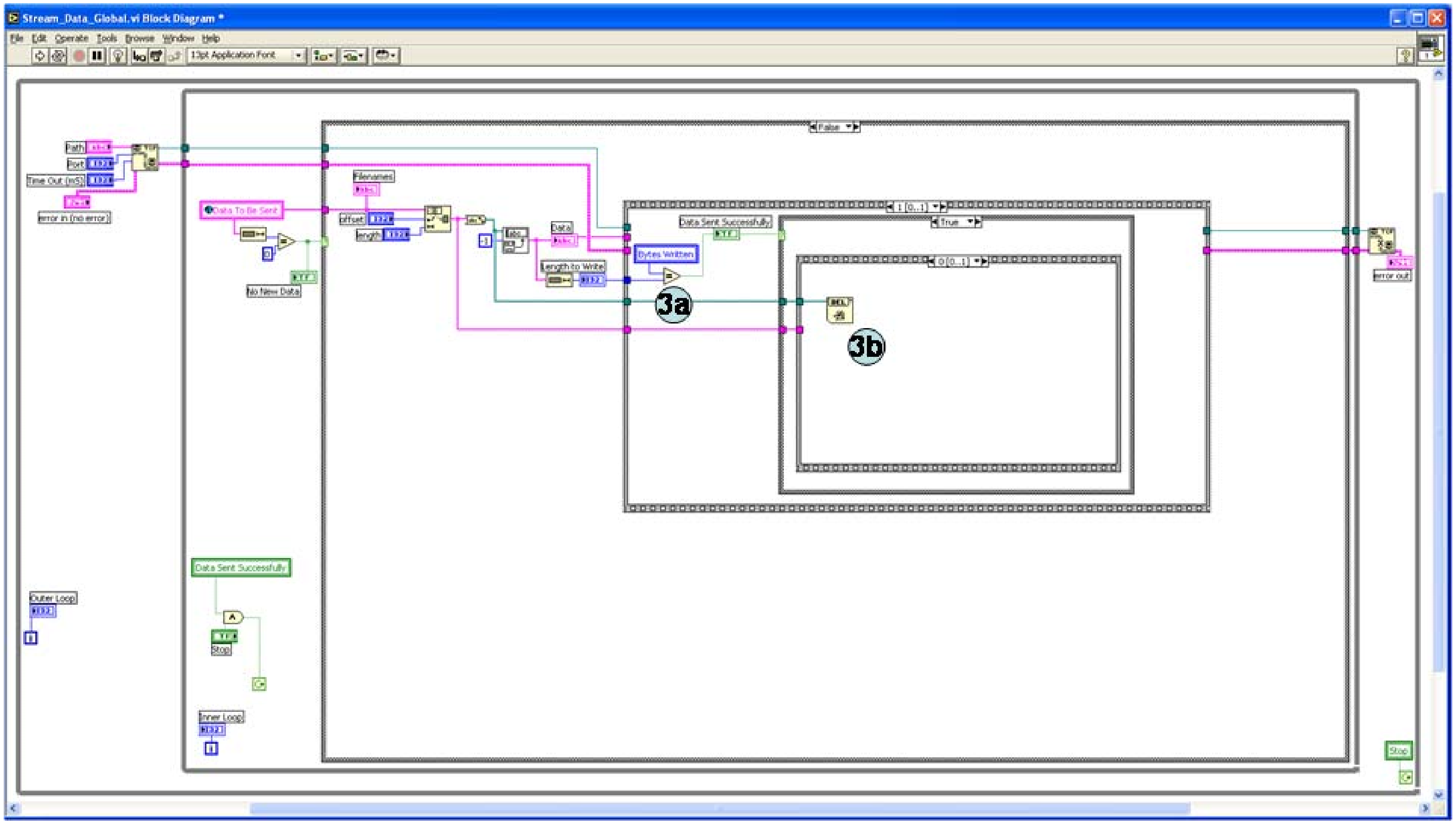


Figure 2.37: Portion of LabVIEW Data Streaming Program Responsible for Comparing Amount of Data Sent versus Size of File and Deleting Data Pending Successful Transmission

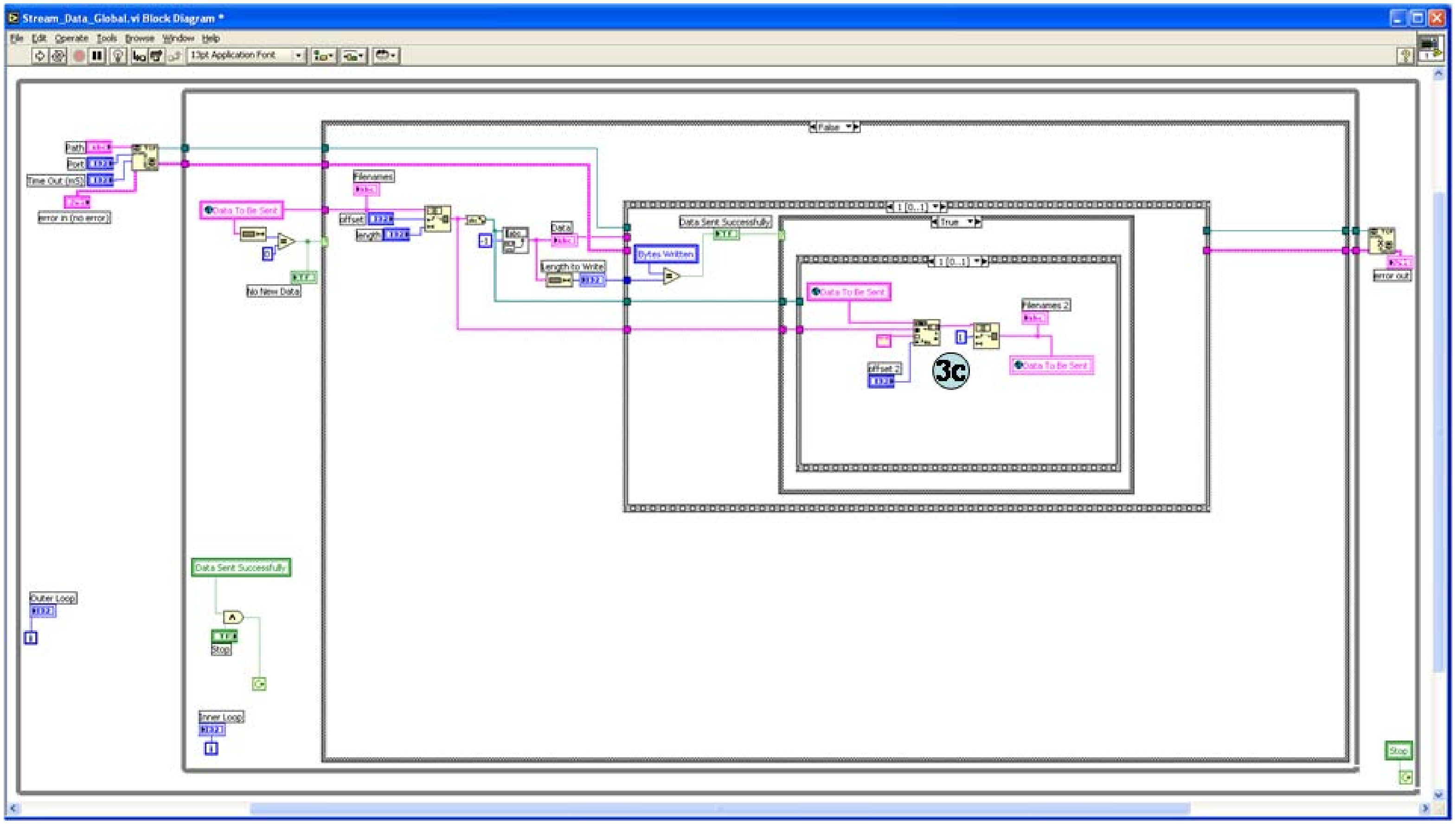


Figure 2.38: Portion of LabVIEW Data Streaming Program Responsible for Updating the List of Data to be Sent Pending Successful Transmission

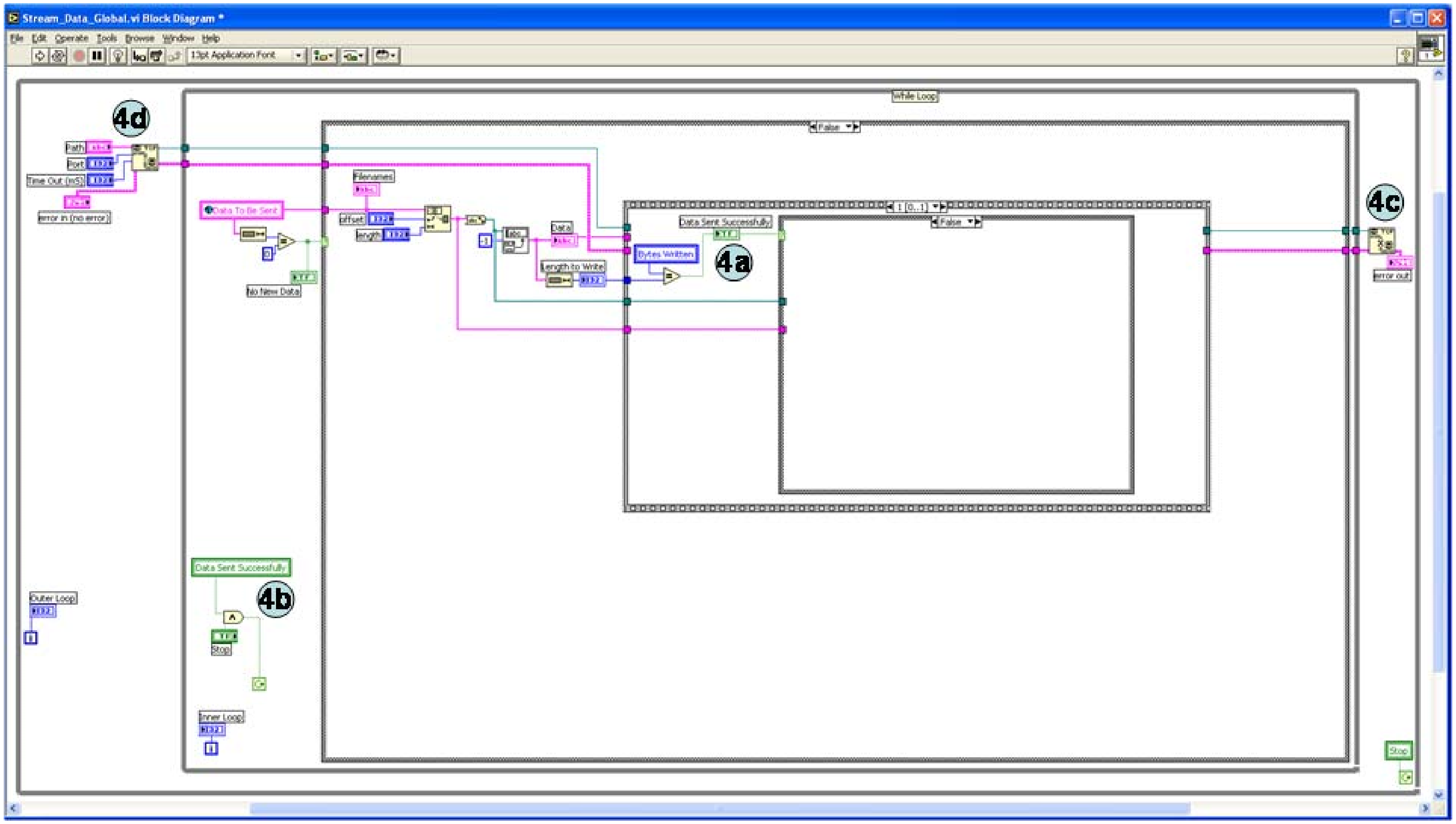


Figure 2.39: Portion of LabVIEW Data Streaming Program Responsible for Closing Inner Loop and Resetting the TCP/IP

Connection if Write Errors are Experienced

A necessary component of a TCP/IP data stream is the development of a program to read the data from the specified TCP/IP data port and to load it into the database. For the testbed detailed in Chapter 3, this is being accomplished using a PERL script loader program. This program reads and stores 10,000 time steps of data from the TCP/IP connection and stores them to a buffer. Once the 10,000th row is read, the program loads all of the data into the DB2 database, clears the buffer, and restarts the process. This insertion process (10,000 rows) takes approximately 20 seconds on a conventional PC desktop.

A potential bottleneck in the program potentially exists when multiple computers are streaming data to a single machine which handles buffering and insertion into the database (Fig. 2.40). The insertion of the data into the database is the most time consuming process and if the data is not inserted quick enough, the insertion computer will eventually run out of memory. Among the possible solutions to this problem are: using a faster computer (which includes using SCSI hard drives in conventional computers or utilizing high-end database machines, such as the IBM Regatta machines) or to employ multiple computers in the data insertion phase, as shown in Figure 2.41.

To test the portability of the developed data buffering/insertion program on future expanded monitoring systems incorporating hundreds of sensors, a test was conducted using simulated data. 100-channels of simulated data (0.005 time step) were generated on three separate computers and streamed using TCP/IP protocols to a single buffer/insertion server. The ability of the server to manage the large data flow was verified in successful uploading of all of the simulated data into a database, over a two week period, without dropping any time steps.

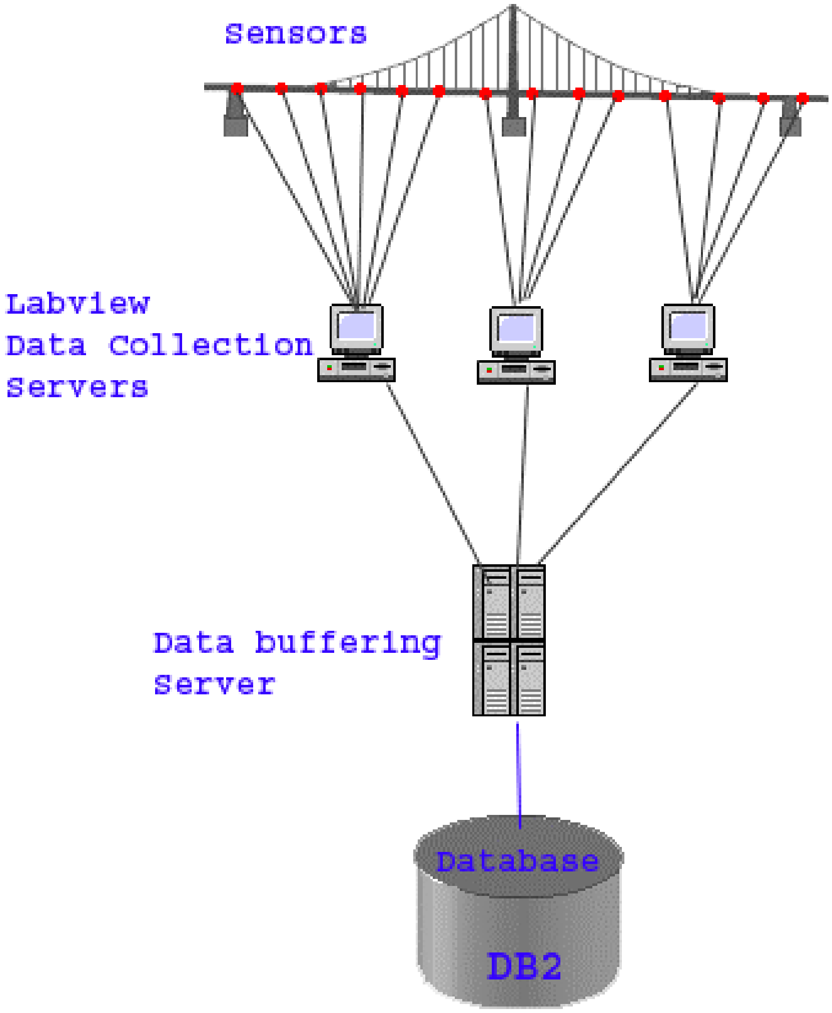


Figure 2.40: Single Insertion Data Loader

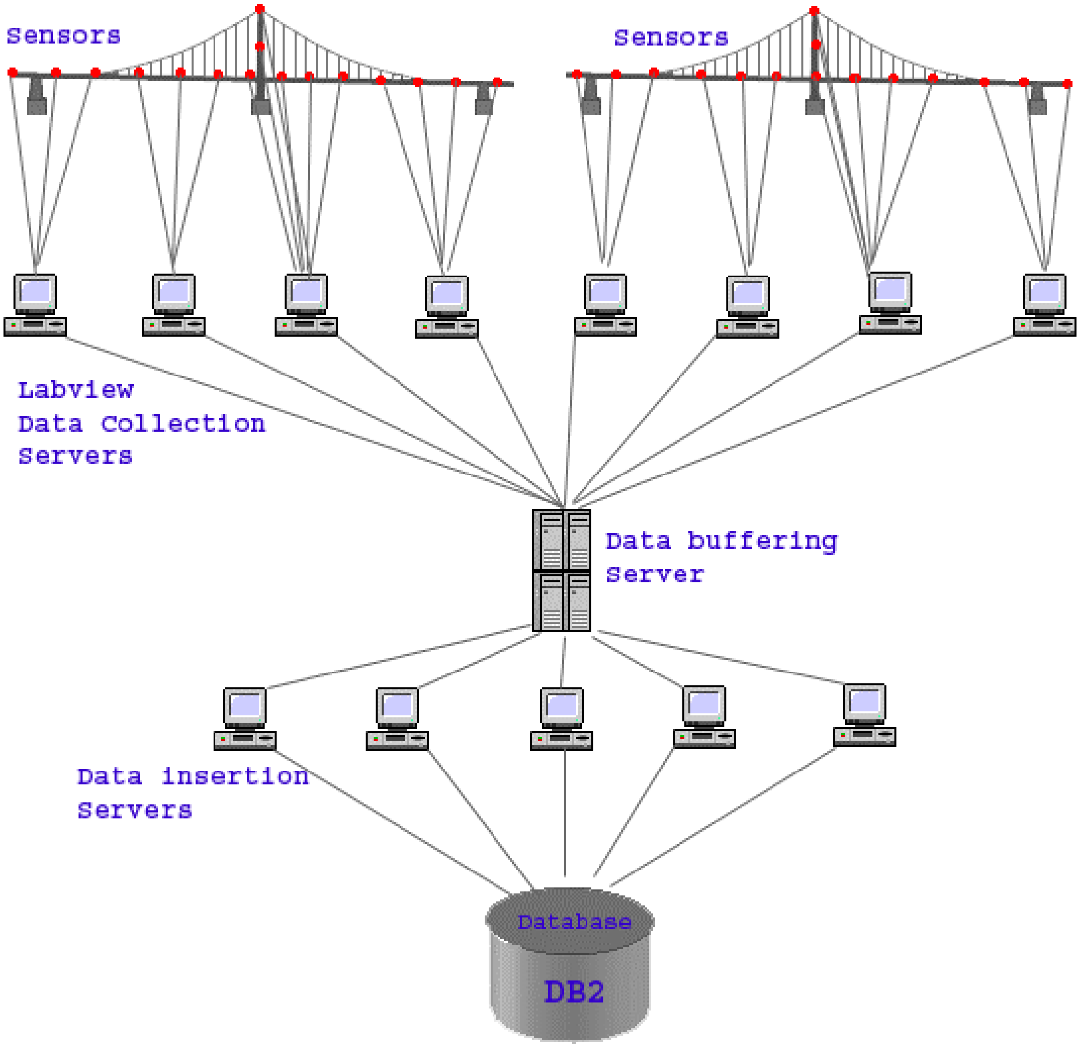


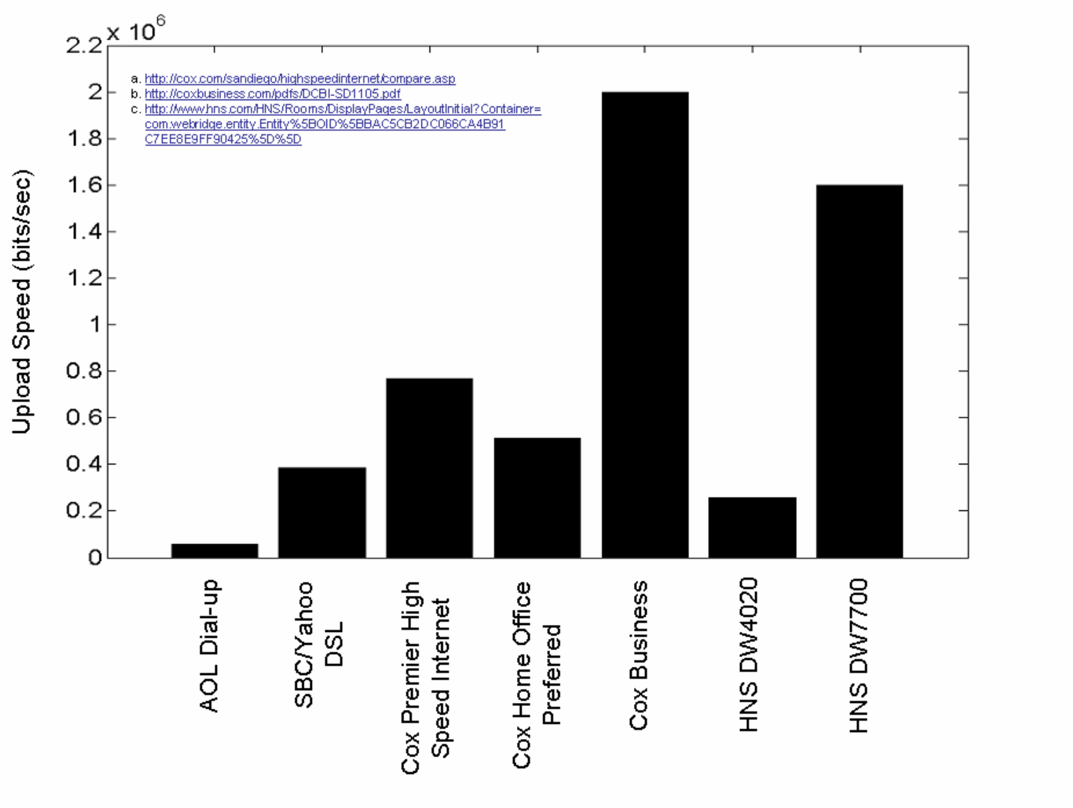
Figure 2.41: Multi Insertion Server Model

2.6.2 Requirements on Bandwidth

The amount of data (including both sensor and video data) capable of being transmitted from a remote monitoring site is governed by the upload speed of the employed Internet Service Provider (ISP) and download speed on the archiving side. As the archiving databases are connected to high speed networks (T1 or better), it is the upload speed available at the field sites which are critical.

2.6.2.1 Broadband Internet Connectivity

A brief comparison of typical Internet upload (uplink) speeds is provided in Figure 2.42. Within this figure, it can be seen that dial-up access can only provide up to 3% of what is available through cable modems. DSL, while faster than dial-up, is still less than 20% of an optimal cable connection. Satellite-based Internet speeds vary greatly from 256 kilobits-per-second (kbps) on the DW4020 terminal to 1.6 Megabits-per-second (Mbps), or 80% of cable speeds. Finally, within cable service, these are widely differing speeds based on which cable package is purchased. Also with cable Internet, the speeds reported in Figure 2.42 are the maximum possible. In practice, cable service provides tend to cap these numbers at values significantly below the optimal ones shown here.



a. http://cox.com/sandiego/highspeedinternet/compare.asp
b. http://coxbusiness.com/pdfs/DCBI-SD1105.pdf
c. http://www.hns.com/HNS/Rooms/DisplayPages/LayoutInitial?Container=com.webridge.entity.Entity%5BOID%5BBAC5CB2DC066CA4B91C7EE8E9FF90425%5D%5D

Figure 2.42: Comparison of Typical Upload Speeds (January 2006)

2.6.2.2 Alternate Internet / Network Solutions

To bring data from a remote location where conventional broadband-Internet connections (such as DSL and cable) are not available, the feasibility of satellite-based Internet is being explored. In future full-scale monitoring applications, transmitting data from the bridge will be accomplished using separate satellite-based data links attached to each of the data/image acquisition clusters (Fig. 2.43). The system investigated is the DIRECWAY system (www.hns.com). Hughes Network Services (HNS) DW6000 Terminal system utilizes two-way satellite service to provide broadband IP anywhere in the world. The DW6000 receives and transmits data through the system's antenna and outdoor electronics, over the satellite, and via the DIRECWAY Network Operations Center. TCP connections can be initiated to or from hosts at the remote locations. An additional advantage is the intranet communications are secure and isolated from other enterprise intranets and from remotes accessing the "public" Internet operating in the same network. With the DW6000, information rates of 48 Mbps for receiving and 256 kbps are possible.

The 4020 terminal (Fig. 2.44) provides high-performance delivery of accelerated IP traffic. It is compatible with all common TCP/IP applications, UDP, or RTP traffic and works with any OS which supports IP over Ethernet. The satellite-dish (Fig. 2.45) supports Ku-band frequency band (14.0 – 14.5 GHz uplink, 10.95 – 12.75 GHz downlink) with outbound carrier data rates of 4.6 Mbps to 48 Mbps and inbound carrier data rates: 64, 128 or 256 Kbps. Newer systems, such as the DW7700 (Fig. 2.46), support speeds scalable up to 90 Mbps and upload speeds of 1.6 Mbps.

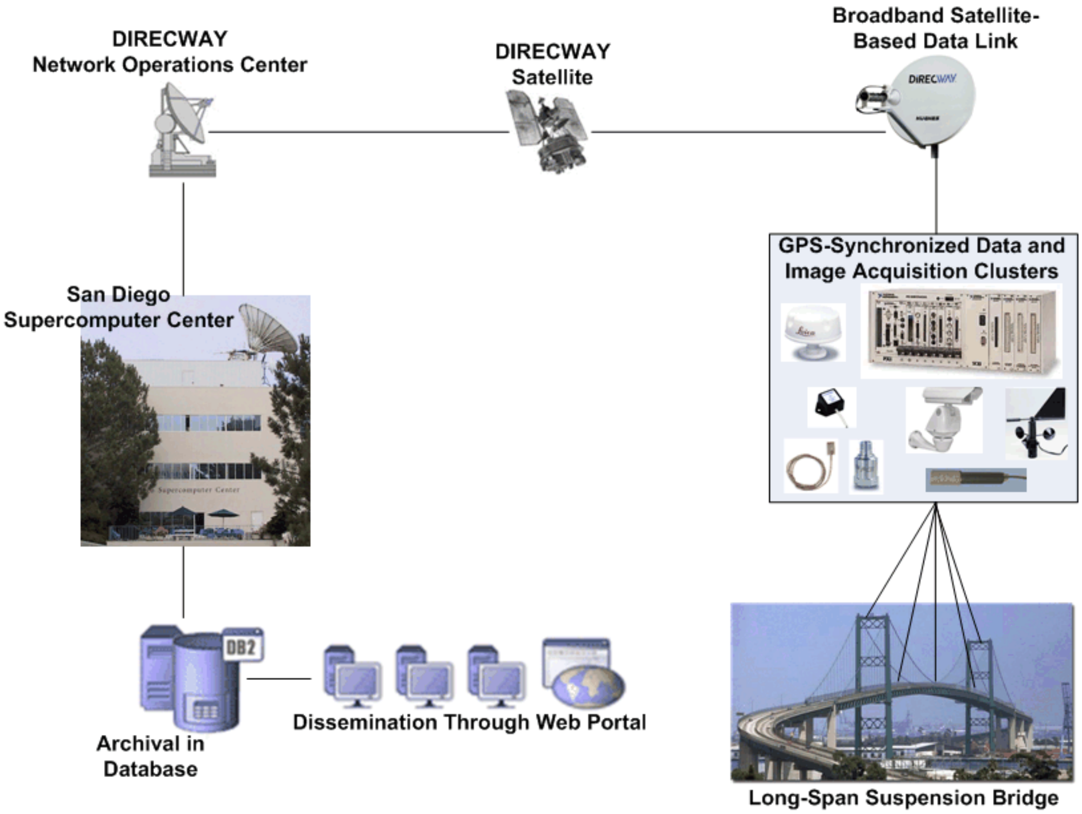

DIRECWAY
Network Operations Center
DIRECWAY
Satellite
Broadband Satellite-
Based Data Link
DIRECWAY
GPS-Synchronized Data and
Image Acquisition Clusters
San Diego
Supercomputer Center
DB2
Archival in
Database
Dissemination Through Web Portal
Long-Span Suspension Bridge

Figure 2.43: Satellite-Based Data Transmission from Bridge to UCSD Campus

(http://healthmonitoring.ucsd.edu)

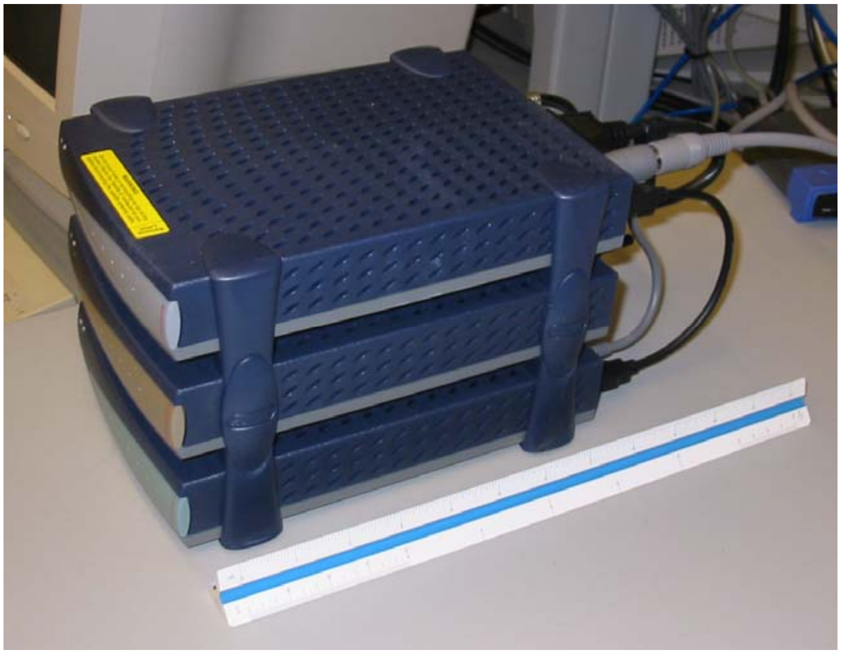


Figure 2.44: DIRECWAY 4020 Two-Way Broadband Terminal (www.hns.com)

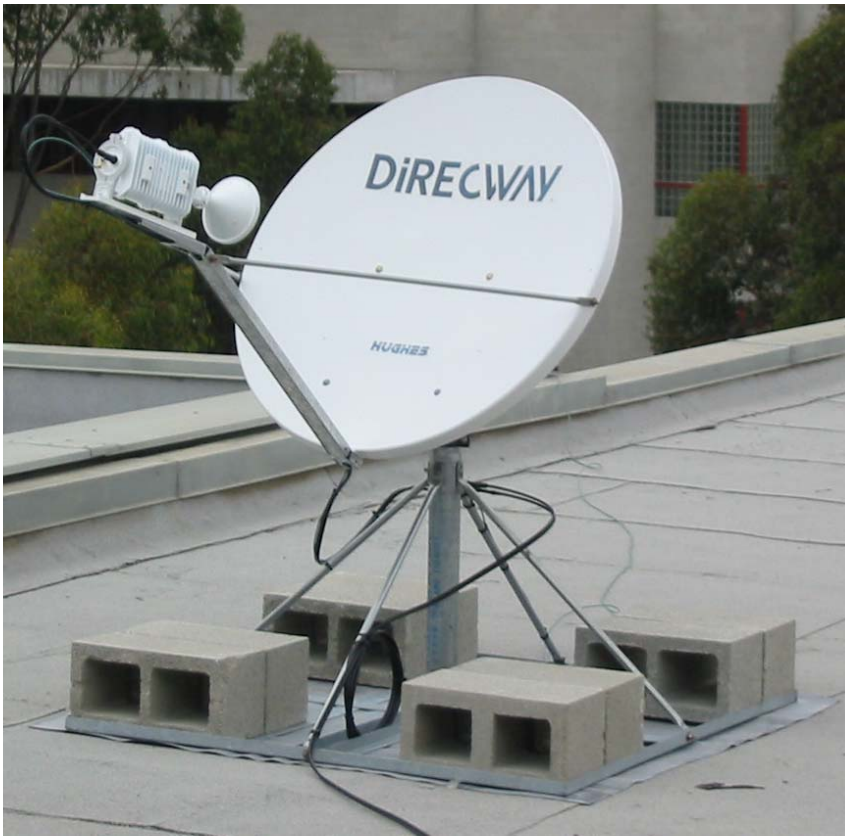


Figure 2.45: DIRECWAY Satellite Dish Installed on the SERF Building at UCSD

Figure 2.46: DIRECWAY DW7700 Broadband Satellite Router (www.hns.com)

2.7 Data Archiving

One of the main issues facing the structural health monitoring community is not the lack of measurements per se, but rather how to measure, acquire, process, and analyze the massive amount of data that is currently coming on-line in order to extract useful information concerning the condition assessment of the monitored structures [Elgamal et al., 2002]. A first step in this direction is the housing of the incoming data. Simply acquiring data and storing it as a series of text (ASCII) files is not a particularly desirable or practical strategy especially when performing post-processing data analysis (although this is still common with many current monitoring systems). Consequently, a series of DB2 databases have been established to house the sensor and video data from the UCSD composite bridge-deck testbed (http://healthmonitoring.ucsd.edu/compositedeck.jsp). Details regarding these particular databases is found in Section 3.4.

Within these databases, a key consideration to the system architecture is the design of the database schema, which will be used to catalogue sensor network information, sensor data and metadata, as well as derived data products for distribution and further analysis. Among the challenges of building a comprehensive schema are ensuring that it will be scalable with the large volumes of data that it will eventually contain, as well as facilitating the querying and analysis of the data from within the database. Monitoring the health of civil infrastructure systems requires a comprehensive approach to data management and analysis. The complexity of data sources (including real-time sensor and video streams, and the output of physics-based and statistical models), and the need to perform advanced near real-time and off-line analyses (often requiring the integration of real-time sensor data with simulation model output) necessitates a scaleable high-performance computational infrastructure.

A central component is a data management system capable of storing raw and preprocessed sensor data from multiple sensor streams, video data from potentially multiple cameras, simulated data produced by load and sensor response models, and derived data produced by a variety of analysis tools. The role of the system is to provide query support to the structural analyst who may wish to retrieve stored or computed information from a single data source, or from a virtual data source constructed by integrating multiple actual data sources. As shown in

Figure 2.47, the data management system must be able to interface with a variety of analysis tools, either to export data to them, or to store back results of computations performed by these engines (as with the image processing discussed in Chapter 4 and neural network strain analysis in Chapter 8 [Yan et al., 2005]), producing derived data that may itself be queried.

2.8 Query Support for Archived Data

To make use of the data after it has been archived, it must be made available for use with the analysis software. This means being able to isolate specific portions of the recorded data from within the entire recorded data set and load them for analysis. For this reason, the data was stored within a series of databases, rather than just storing series of text files and images on a large number of hard drives.

These database systems provide a means for quickly and easily accessing any or all of the recorded data and metadata. Now, for example to look at recorded strain time histories in which peak strains exceed a certain threshold, a quick search of the metadata tables provides this information. In contrast, if this was done by opening each of the corresponding data text files, loading the data from within the files, and then searching for the maximum values, the amount of time required would be orders of magnitude greater. As detailed in Section 3.5.1, the data recorded from the UCSD testbeds may now be queried using either a direct connection to the DB2 databases or on-line using a web browser and the developed web portal.

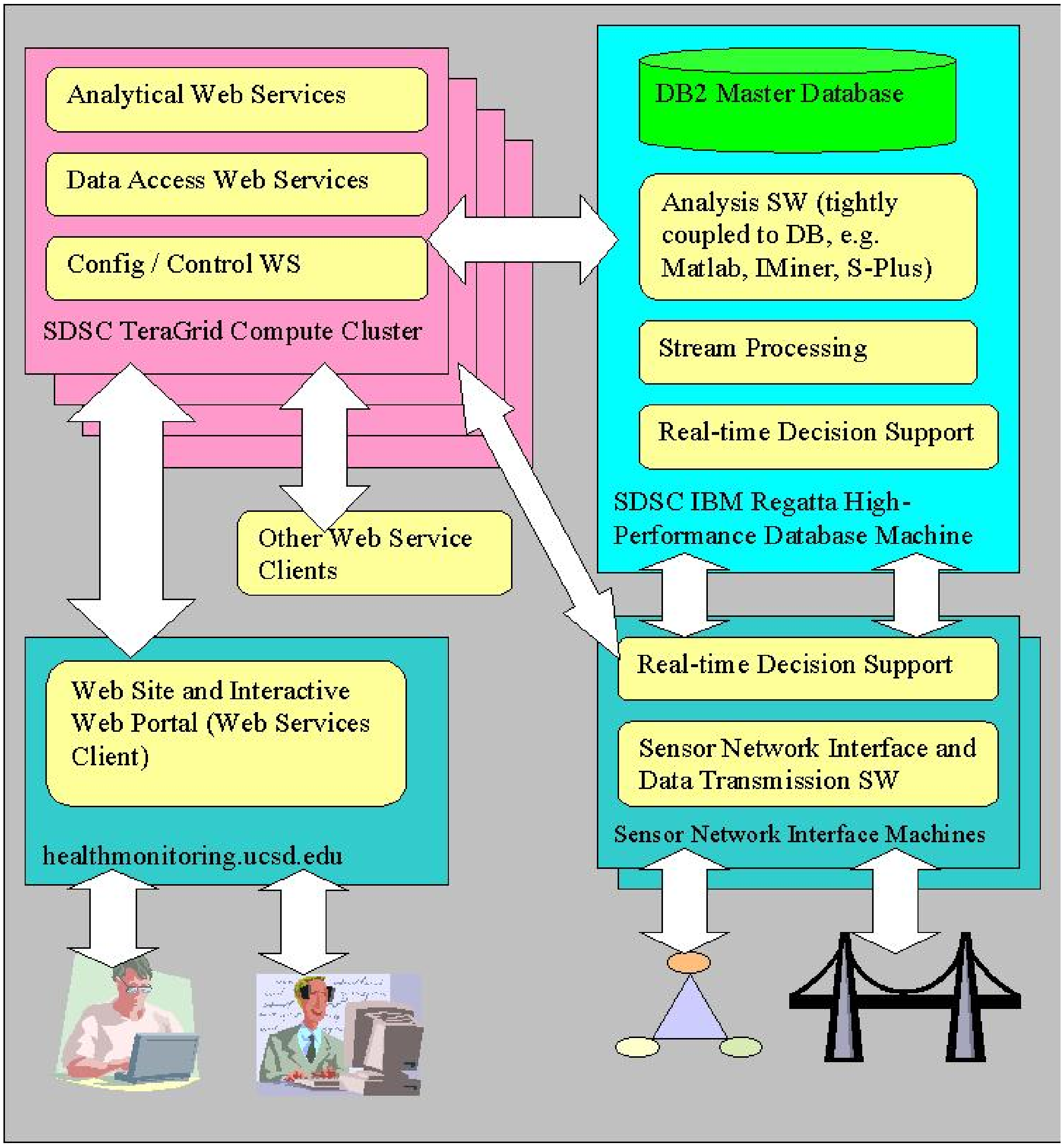

Analytical Web Services
Data Access Web Services
Config / Control WS
SDSC TeraGrid Compute Cluster
DB2 Master Database
Analysis SW (tightly coupled to DB, e.g. Matlab, IMiner, S-Plus)
Stream Processing
Real-time Decision Support
SDSC IBM Regatta High-Performance Database Machine
Other Web Service Clients
Real-time Decision Support
Sensor Network Interface and Data Transmission SW
Sensor Network Interface Machines
Web Site and Interactive Web Portal (Web Services Client)
healthmonitoring.ucsd.edu

Figure 2.47: Envisioned Integration of Sensor Networks with Database Systems and Web Portal: Courtesy of Tony Fountain, SDSC [Elgamal et al., 2003c]

2.9 Web Portal for Data Dissemination

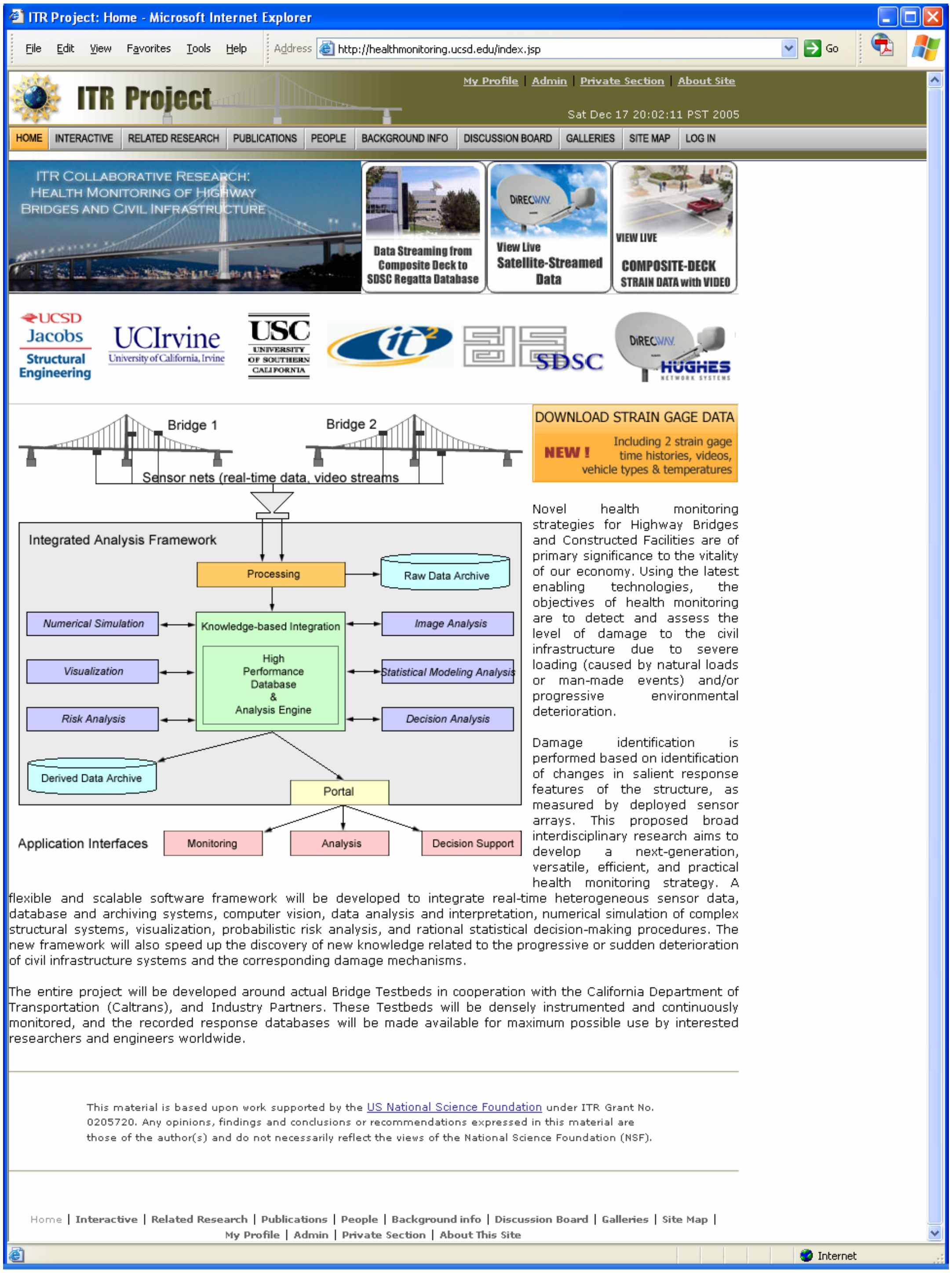
To provide users with limited or no database programming skills access to the recorded data, a web portal was created (Fig. 2.48). This portal (created with the assistance of project webmaster Minh Phan, http://healthmonitoring.ucsd.edu) allows users to utilize and apply pre-written queries to all of the data recorded from the UCSD structural monitoring testbeds. Using a standard web browser, like Internet explorer, world wide access is made available for browsing through the data returned by these queries.

On the composite bridge deck, these functions include:

- The ability to query the raw data for the peak strains registered over a period of time (http://healthmonitoring.ucsd.edu/compositedeck2/search1.jsp and http://healthmonitoring.ucsd.edu/compositedeck16/searchPeak.jsp).
- Use the labeled vehicle type data set to examine strain time histories corresponding to a particular type of vehicle (http://healthmonitoring.ucsd.edu/compositedeck2/searchVehicleType.jsp, http://healthmonitoring.ucsd.edu/compositedeck16_all/vehicle16.jsp, and Section 4.6 of this paper).
- Browse through processed/analyzed data, such as extracted features from recorded video (http://healthmonitoring.ucsd.edu/compositedeck16/search5.jsp and Section 4.5 in this document) and vehicle classifications assigned from an employed neural network

(http://healthmonitoring.ucsd.edu/compositedeck2/predict.jsp, Fig. 2.49 [Yan, 2006]).

Within the UCSD testbed, Web Services are employed to achieve modularity and extensibility between the various components of the system and also between this and other systems. Initially, Web Services seemed most appropriate in the areas of system and sensor network-related configuration and control, as an interface to the data being collected and the analysis routines provided. Also investigated was the use of Web Services in other areas, including: interaction between the analysis and processing components of the system, data and video streaming, as well secure data access and transport. Furthermore, the Web Services approach provides a vendor-neutral, platform-independent, standards-based, and industry supported means of achieving this decoupling. Further details regarding the web portal are provided in Section 3.5.


ITR Project: Home - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://healthmonitoring.ucsd.edu/index.jsp
Go
ITR Project
My Profile | Admin | Private Section | About Site
Sat Dec 17 20:02:11 PST 2005
HOME INTERACTIVE RELATED RESEARCH PUBLICATIONS PEOPLE BACKGROUND INFO DISCUSSION BOARD GALLERIES SITE MAP LOG IN
ITR COLLABORATIVE RESEARCH: HEALTH MONITORING OF HIGHWAY BRIDGES AND CIVIL INFRASTRUCTURE
Data Streaming from Composite Deck to SDSC Regatta Database
View Live Satellite-Streamed Data
VIEW LIVE COMPOSITE-DECK STRAIN DATA with VIDEO
UCSD Jacobs Structural Engineering
UCIrvine University of California, Irvine
USC UNIVERSITY OF SOUTHERN CALIFORNIA
it2
SDSC
DIRECWAY
HUGHES NETWORK SYSTEMS
Bridge 1
Bridge 2
Sensor nets (real-time data, video streams)
Integrated Analysis Framework
Processing
Raw Data Archive
Numerical Simulation
Knowledge-based Integration
Image Analysis
Visualization
High Performance Database & Analysis Engine
Statistical Modeling Analysis
Risk Analysis
Decision Analysis
Derived Data Archive
Portal
Application Interfaces
Monitoring
Analysis
Decision Support
DOWNLOAD STRAIN GAGE DATA
NEW !
Including 2 strain gage time histories, videos, vehicle types & temperatures
Internet

Novel health monitoring strategies for Highway Bridges and Constructed Facilities are of primary significance to the vitality of our economy. Using the latest enabling technologies, the objectives of health monitoring are to detect and assess the level of damage to the civil infrastructure due to severe loading (caused by natural loads or man-made events) and/or progressive environmental deterioration.

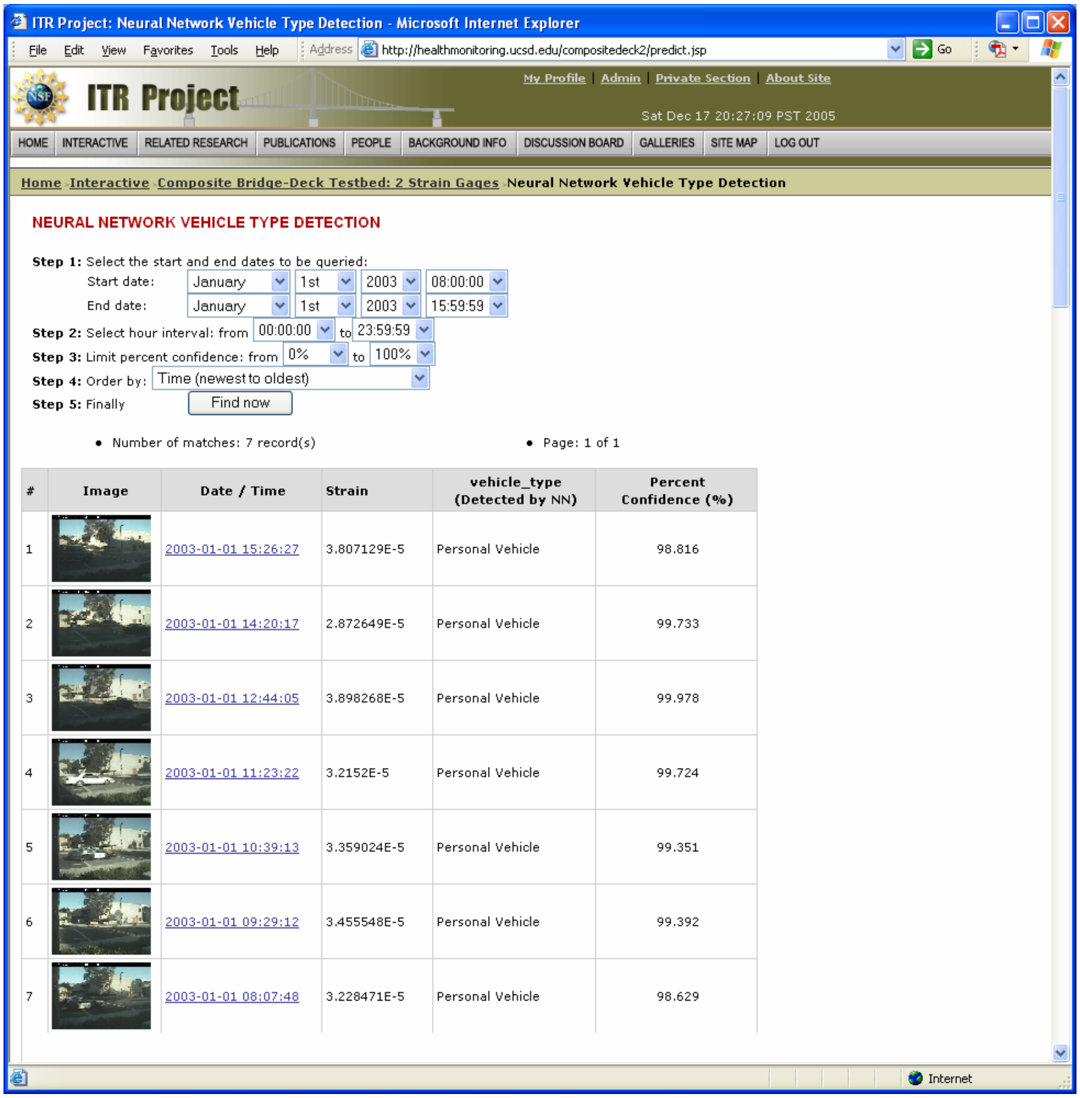
Damage identification is performed based on identification of changes in salient response features of the structure, as measured by deployed sensor arrays. This proposed broad interdisciplinary research aims to develop a next-generation, versatile, efficient, and practical health monitoring strategy. A flexible and scalable software framework will be developed to integrate real-time heterogeneous sensor data, database and archiving systems, computer vision, data analysis and interpretation, numerical simulation of complex structural systems, visualization, probabilistic risk analysis, and rational statistical decision-making procedures. The new framework will also speed up the discovery of new knowledge related to the progressive or sudden deterioration of civil infrastructure systems and the corresponding damage mechanisms.

The entire project will be developed around actual Bridge Testbeds in cooperation with the California Department of Transportation (Caltrans), and Industry Partners. These Testbeds will be densely instrumented and continuously monitored, and the recorded response databases will be made available for maximum possible use by interested researchers and engineers worldwide.

This material is based upon work supported by the US National Science Foundation under ITR Grant No. 0205720. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF).

Home | Interactive | Related Research | Publications | People | Background info | Discussion Board | Galleries | Site Map |
My Profile | Admin | Private Section | About This Site

Figure 2.48: Web Portal for UCSD Structural Monitoring Testbeds


ITR Project: Neural Network Vehicle Type Detection - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Address
http://healthmonitoring.ucsd.edu/compositedeck2/predict.jsp
Go
ITR Project
My Profile
Admin
Private Section
About Site
Sat Dec 17 20:27:09 PST 2005
HOME
INTERACTIVE
RELATED RESEARCH
PUBLICATIONS
PEOPLE
BACKGROUND INFO
DISCUSSION BOARD
GALLERIES
SITE MAP
LOG OUT
Home › Interactive › Composite Bridge-Deck Testbed: 2 Strain Gages › Neural Network Vehicle Type Detection
Find now
Internet

NEURAL NETWORK VEHICLE TYPE DETECTION

Step 1: Select the start and end dates to be queried:
Start date: January 1st 2003 08:00:00
End date: January 1st 2003 15:59:59

Step 2: Select hour interval: from 00:00:00 to 23:59:59

Step 3: Limit percent confidence: from 0% to 100%

Step 4: Order by: Time (newest to oldest)

Step 5: Finally Find now

- Number of matches: 7 record(s)
- Page: 1 of 1

#	Image	Date / Time	Strain	vehicle_type (Detected by NN)	Percent Confidence (%)
1		2003-01-01 15:26:27	3.807129E-5	Personal Vehicle	98.816
2		2003-01-01 14:20:17	2.872649E-5	Personal Vehicle	99.733
3		2003-01-01 12:44:05	3.898268E-5	Personal Vehicle	99.978
4		2003-01-01 11:23:22	3.2152E-5	Personal Vehicle	99.724
5		2003-01-01 10:39:13	3.359024E-5	Personal Vehicle	99.351
6		2003-01-01 09:29:12	3.455548E-5	Personal Vehicle	99.392
7		2003-01-01 08:07:48	3.228471E-5	Personal Vehicle	98.629

Figure 2.49: Web Page for Displaying Assigned Labels Using A Neural Network Applied to Recorded Strains [Yan, 2006].

2.10 Summary

Within this chapter, the elements composing a state of the art structural monitoring system have been detailed. First, an overview of the dynamic and environmental sensors required for a heterogeneous sensor array was presented. For continuous monitoring applications, where data logging is not practical, PC-based data/image acquisition has been addressed, and a key issue of data/image synchronization was noted. Data transmission protocols for streaming data from a remote bridge site to more suitable location were explored. As simply storing acquired data as a series of a text files is not a viable option in post-processing data analysis, issues related to database systems were discussed. A test was conducted in which data from 300 simulated sensors were streamed to a single buffering/insertion server which was able to successfully manage the flow of data and upload the entire data set (generated over a 2 week period) to a database.

3 Application – UCSD Composite Bridge-Deck Panels (Testbed for Framework)

Within this chapter, a demonstration testbed for the application of the developed integrated structural monitoring framework is presented. The monitoring system has been deployed on a series of composite bridge-deck panels located on the University of California, San Diego campus which are subject to traffic loading. A brief background on the bridge decks (including the reason they were selected for this study) is provided followed by a summary of the equipment used for the data and image acquisition. Next, issues related to data/video synchronization, data transmission, data archiving, and web-based data dissemination are discussed. Finally, details of the traffic data that has been recorded are summarized.

3.1 Reason for Using As a Testbed

For testing the algorithms and methodologies developed as part of a NSF sponsored ITR research grant (for which the material presented within this paper directly relates), it was decided to select a small structure on the UCSD campus to instrument and monitor. This testbed for monitoring technologies would serve a crucial role in developing, troubleshooting, and evaluating the reliability of all aspects related to structural monitoring. This structure had to be located nearby and easily accessible to allow for constant modifications required during the initial installation.

The University of California, San Diego (UCSD) composite bridge-deck panels [Zhao, 1999 under the supervision of Professors Vistasp Karbhari and Frieder Seible of UCSD] were selected for their existing array of strain gages and location near the Science and Engineering Research Facility (SERF) Building, centrally located on the UCSD campus. These bridge decks were installed (with instrumentation) in a roadway on campus and are subjected to a nearly continuous stream of traffic (including campus shuttle busses, passenger vehicles, delivery trucks, and construction equipment). As shown in Figure 3.1, the panels are located between the two white lines traversing the road. This structure proved extremely useful and convenient as it was already instrumented with an array of strain gages and existing conduits brought all of the cables from the sensors into one of the labs in the SERF building. By having ready access to an existing sensor array (Fig. 3.2), the expenditure of time required to have the monitoring testbed up and running was greatly reduced. To begin measuring strains on the bridge decks, all that had to be done was connect the strain gage leads to the appropriate signal conditioning modules, which were then run to the data acquisition hardware. In this regard, the time consuming, expensive, and labor intensive operations of selecting a structure, receiving necessary permissions, and installing the sensors were bypassed. Since the signal from all of the strain gages, passed directly into a nearby lab (Fig. 3.3), it was decided to house all of the data acquisition hardware within, which proved to be both time and cost saving as environmental enclosures were not required. Additional advantages included the presence of high speed Internet within the lab which enabled continuous data transmission and the presence of a large window overlooking the composite decks that provided a suitable location for installing a camera to record passing traffic.

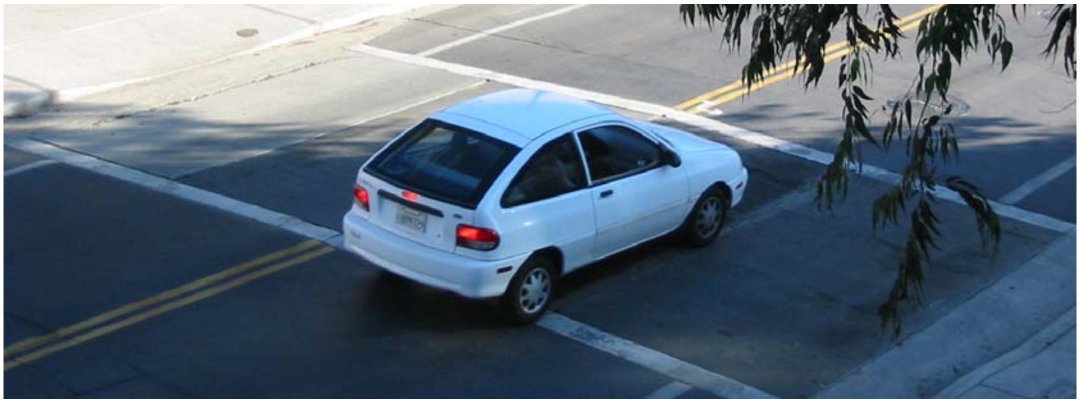


Figure 3.1: UCSD Composite Bridge-Deck Panels

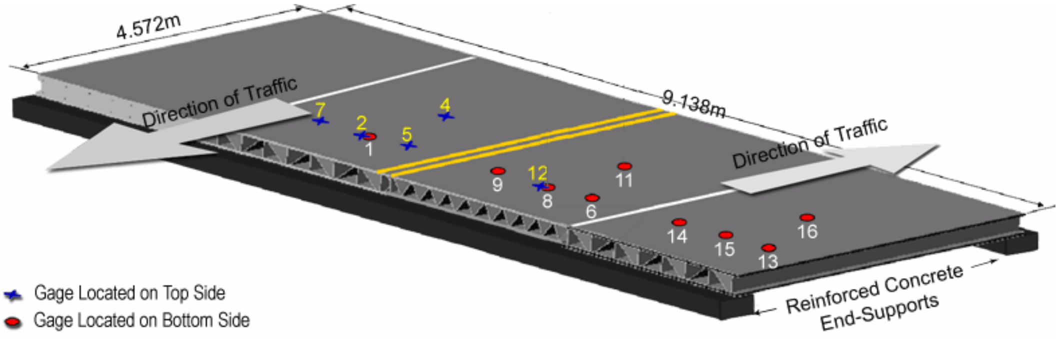


Figure 3.2: Existing Strain Gage Array

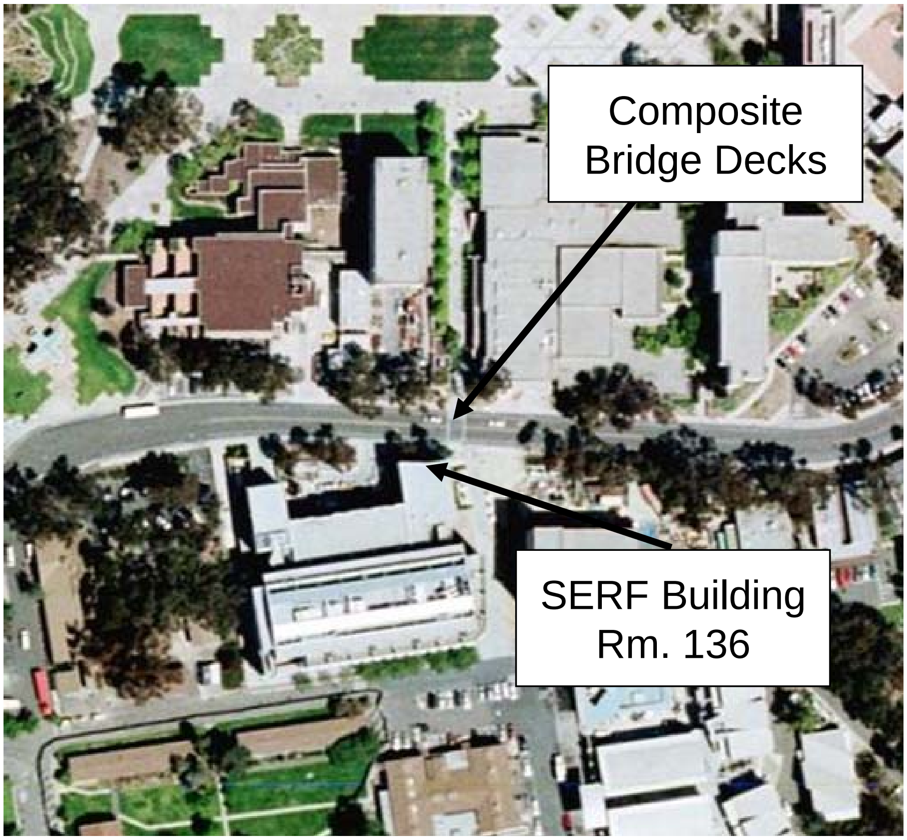


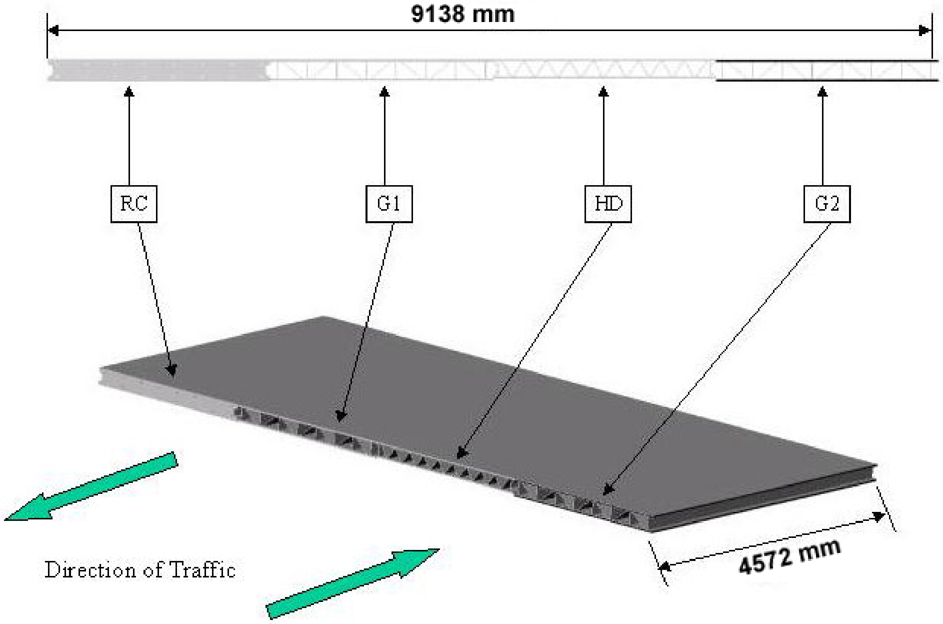
Figure 3.3: Aerial View of Composite Bridge Decks and SERF Building (Photo Courtesy of UCSD FacilitesLink, https://facilities.ucsd.edu)

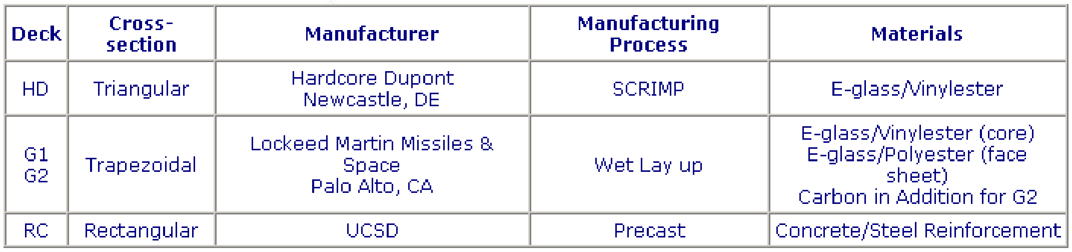
While the composite bridge decks were extremely useful in getting a functional monitoring testbed operational in the shortest time possible, one disadvantage was associated with the layout of the existing sensor array. In adopting this testbed, it was necessary to make due with the existing sensor array as changes to the sensor network are not easily accomplished. To access the sensors, the decks would have to be removed from the roadway and temporary concrete decks (which had to be cast in advance) installed in their place. This work also requires shutting down at least one lane of traffic on a campus road used heavily by shuttle busses and delivery trucks. While useful for monitoring deformation during static load testing, under traffic loading the existing sensor array provided relatively redundant information on the west side of the composite bridge decks and much less information from the east side. Further, while over 30 sensors were originally installed in 1996, many of them (including all of the displacement transducers) were no longer functional. When the project was initiated, 16 of the strain gages functioned (Fig. 3.2); however, upon completion only 14 were still operational. Again, replacing gages or adding other types of sensors to the decks was therefore not a possibility as this necessitated removing the decks.

3.2 Background

The three field-size fiber-reinforce polymer (FRP) deck panels and one reinforced concrete deck panel (Figures 3.4 and 3.5) were installed along a roadway (spanning an underlying open instrumentation conduit) on the campus of UCSD in the summer of 1996 [Zhao, 1999]. Each panel measures 4.58 m (15 ft) in length and 2.29 m (7.5 ft) in width. The instrumentation conduit was excavated in the street and reinforced concrete end-supports were cast with a clear span of 4.0 m (13 ft). The longitudinal direction of the panels is parallel to that of traffic direction on the street. The panels were connected to each other and to the curbs by shear keys filled with concrete (shown in Figure 3.6), and supported at the ends in the longitudinal direction on hydrostone. Polymer-concrete wearing surface was applied individually to the top surface of each panel prior to installation. The installation of one of the panels is shown in Figure 3.7. These installed panels were proof-tested to approximately 36 tons (80 kip) prior to installation.

As reported in Zhao, 1999, the objective of the original test setup was to monitor the change in deflection response of the deck panels over time. This was done by comparing the panel behavior from time to time with the assumption that the shear key connections between neighboring decks are rigid and do not degrade over time. Thus, changes in deflection under the same load over time would reflect a stiffness change of the deck panels only. These efforts included measuring deflection of the deck panels at different time intervals under the same static loading conditions. Two tests - one on February 20, 1997 and the other on April 3, 1999 - were conducted [Zhao, 1999]. Dead load was applied at the center of each panel using two reinforced concrete blocks with a total weight of 108 kN (24.3 kip). Readings from strain gages mounted on the top and bottom surfaces of the deck panels (at mid- and quarter span) were recorded [Zhao, 1999]. No visual damage in either the surface of the composite panels or the shear keys between the panels was observed at the time of the tests. Further details regarding the test setup and experimental results are available in Zhao, 1999.





Deck	Cross-section	Manufacturer	Manufacturing Process	Materials
HD	Triangular	Hardcore Dupont Newcastle, DE	SCRIMP	E-glass/Vinylester
G1 G2	Trapezoidal	Lockeed Martin Missiles & Space Palo Alto, CA	Wet Lay up	E-glass/Vinylester (core) E-glass/Polyester (face sheet) Carbon in Addition for G2
RC	Rectangular	UCSD	Precast	Concrete/Steel Reinforcement

Figure 3.4: Construction Details of Deck Sections.

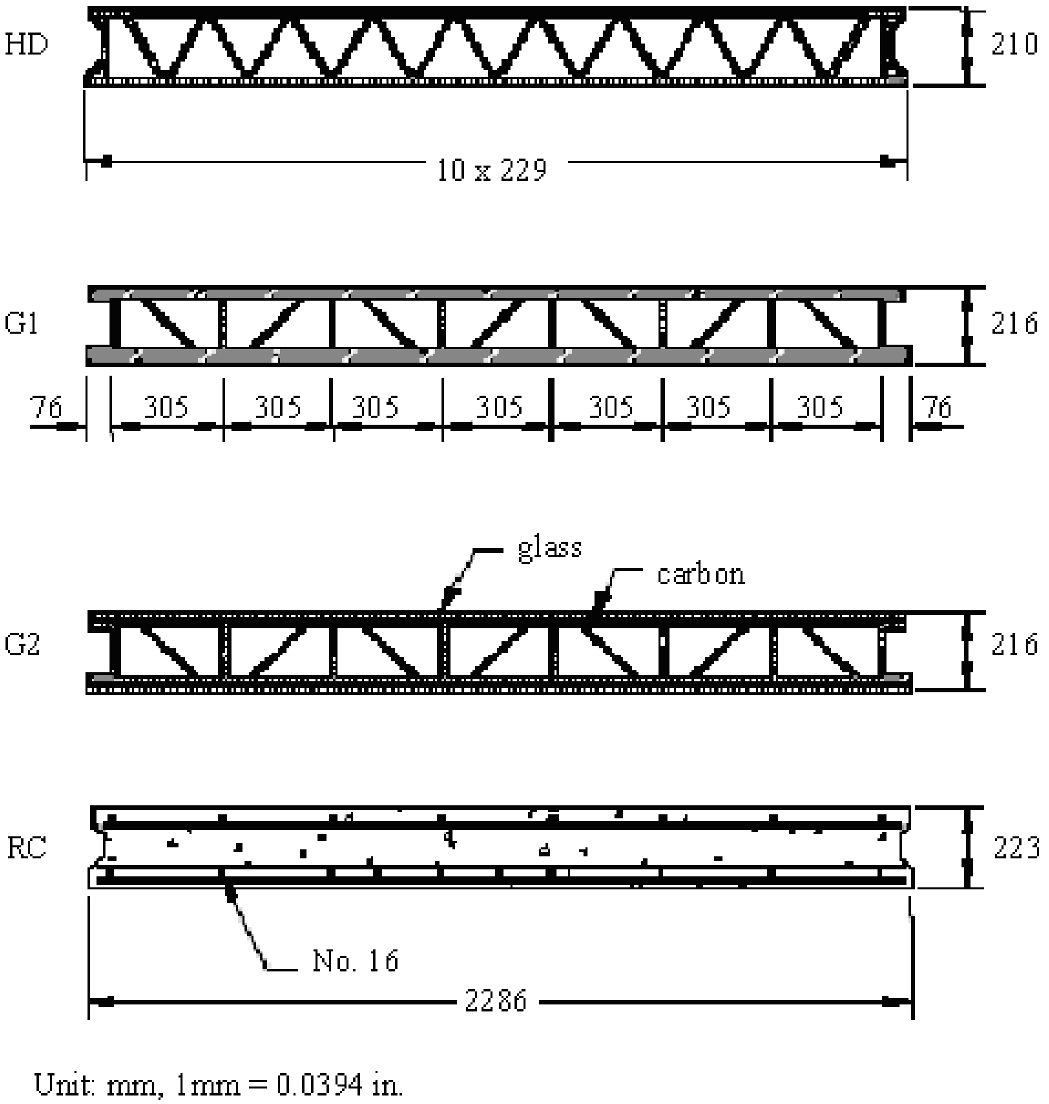


Figure 3.5: Schematic of the Cross-Sections of the Decks (see Fig. 3.3 for HD, G1, G2, and RC Definitions, Adapted from Zhao, 1999).

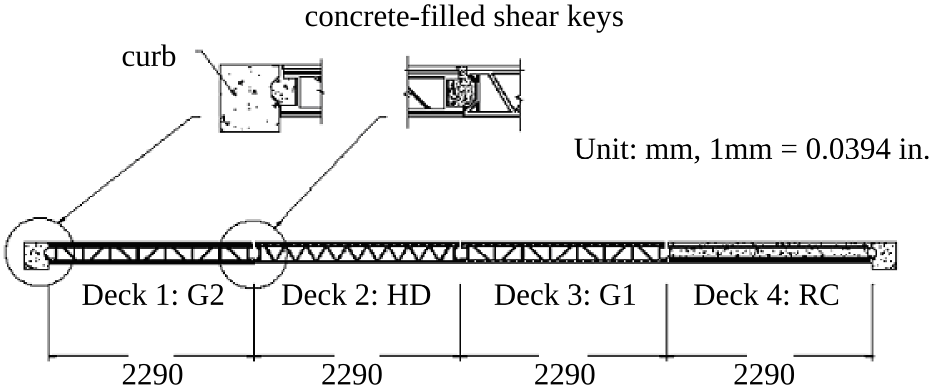


Figure 3.6: Concrete-filled Shear Keys [Zhao, 1999].

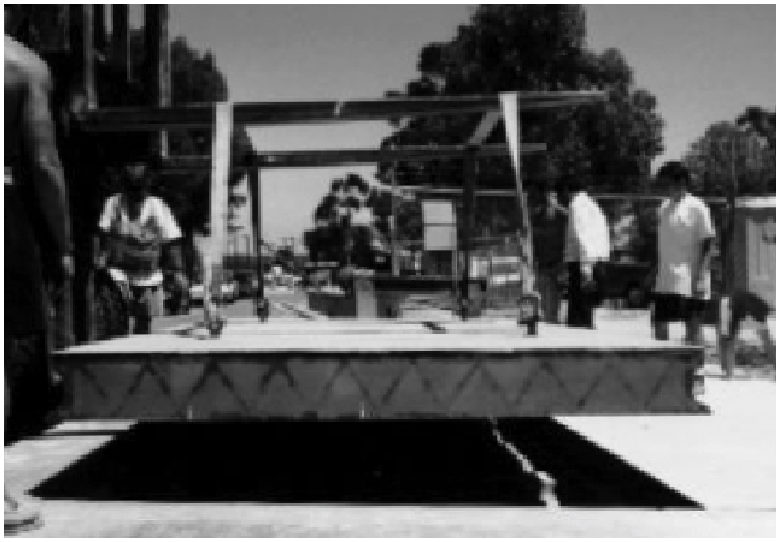


Figure 3.7: Field installation of a FRP Composite Deck Panel [Zhao, 1999].

3.3 Current Monitoring System

On February 19, 2002, live data from selected strain gages along with audio and video feeds (Figures 3.8 and 3.9) became available on-line on a 24/7 basis over a web-site for worldwide access (http://healthmonitoring.ucsd.edu). The recorded audio, video, and strain data are archived within a database to provide a convenient environment for developing intelligent algorithms for data analyses, and for monitoring traffic loads. This monitoring environment is ideal for development and verification of such computer-based automated algorithms.

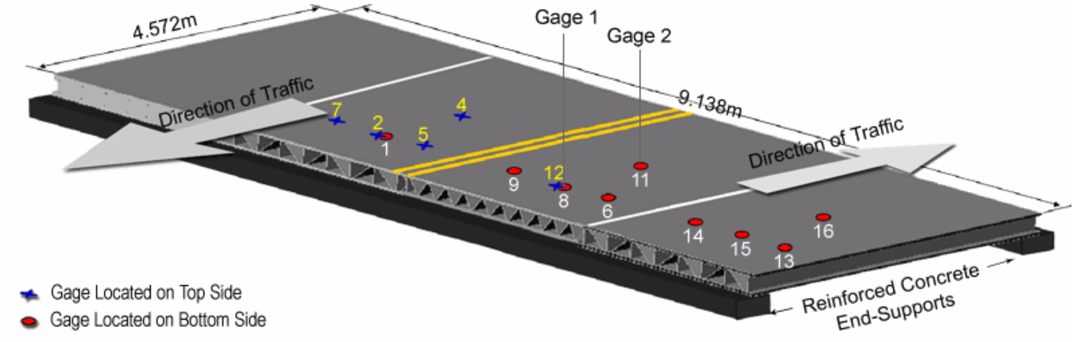
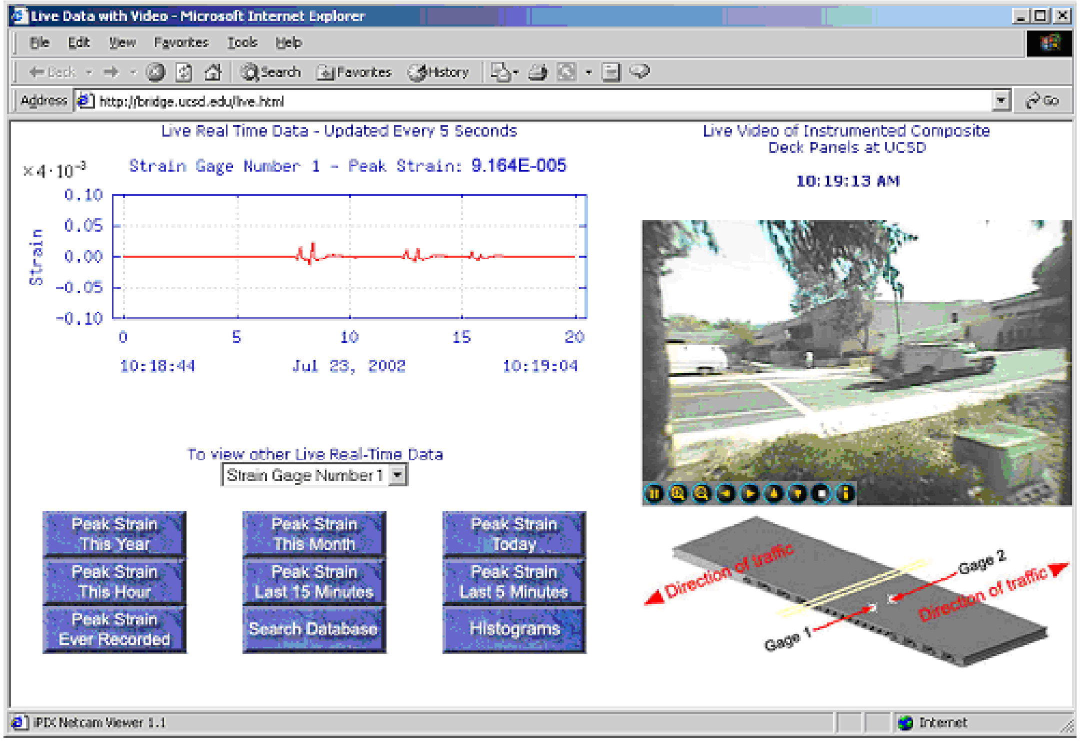


Figure 3.8: Location of Gages 1 and 2 in Existing Sensor Array

3.3.1 Experimental Setup

The current health monitoring website setup utilizes the array of strain gages (Fig. 3.8) mounted on the underside of the composite decks [Zhao, 1999].

These sensors are connected to a Personal Computer, by means of an Analog to Digital (A/D) converter. Through a stand-alone software interface (LabVIEW VI http://ni.com/), the computer controls the acquisition of output response. In the first generation system, this computer also acted as a web-server, allowing the remote Internet user access to the available on-line functions. Later, more sophisticated, versions streamed the acquired data to a remote computer where it was archived within a database and made available on-line through a webportal using improved database querying features. An attached network video camera along with a java refresh applet or ActiveX viewer provided a continuous live image to the user (Fig. 3.9).



Live Real Time Data - Updated Every 5 Seconds

Strain Gage Number 1 - Peak Strain: 9.164E-005

To view other Live Real-Time Data

Live Video of Instrumented Composite Deck Panels at UCSD

10:19:13 AM

Figure 3.9: Pilot Monitoring Website (http://healthmonitoring.ucsd.edu) for Composite Bridge-deck Panels Testbed

3.3.2 Hardware Components and Data Acquisition

The experimental hardware employed on the original pilot system was composed of the following items:

- National Instruments (http://ni.com/) PCI-MIO-16E-1 analog I/O board (Fig. 3.10) with the following characteristics: 1.25 MS/s, 12-bit, eight differential analog inputs, 2 differenetial analog outputs, ± 10 V range.
- Axis 2120 network camera (Fig. 3.11) with iPIX 180° network camera viewer allows for an unlimited number of users, virtual pan/tilt/zoom,

up to 30 frames per second (http://visualsecurity.com/html/ipixcam.html).

- TML general purpose foil strain gages (Fig. 3.12, http://straingage.com) with 10 mm gage length.
- Encore Model 633 strain gage bridge amplifier (Fig. 3.13) (http://www.encore-elec.com/m633.htm).

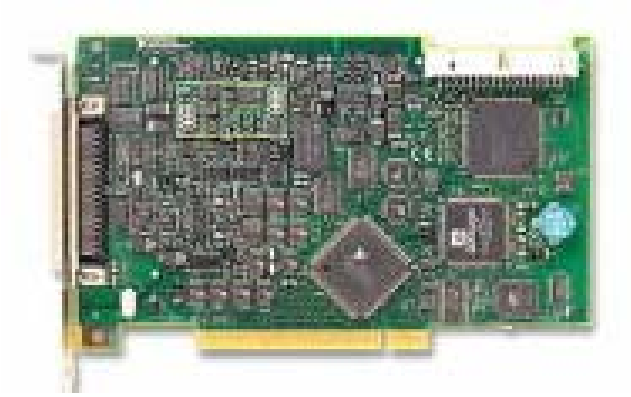


Figure 3.10: National Instruments PCI-MIO-16E-1 Analog I/O Board

Figure 3.11: Axis 2120 Network Camera

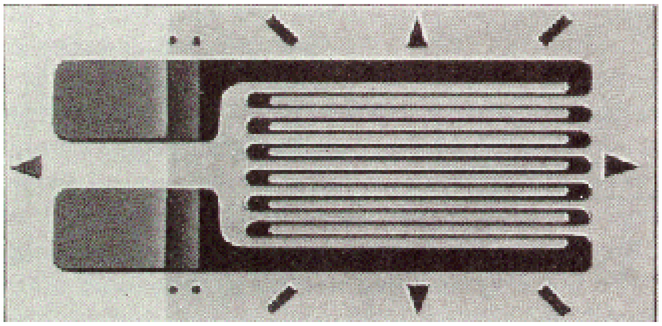


Figure 3.12: Typical Foil Strain Gage

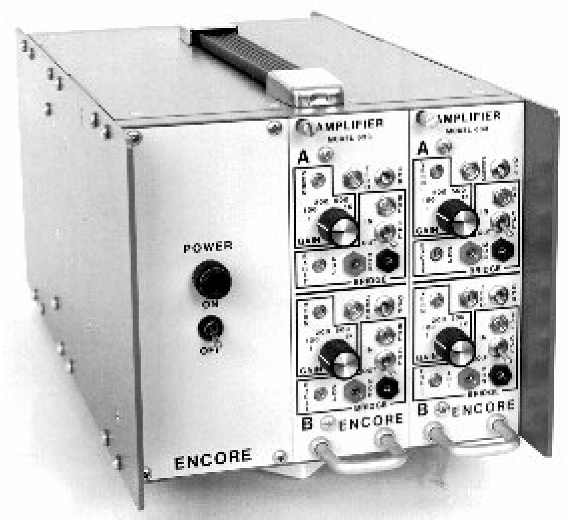


Figure 3.13: Encore Model 633Strain Gage Bridge Amplifier

The employed server computer was a PC running Microsoft Windows NT Server 4.0. A multi-tasking environment allowed the Internet Information Services (IIS) 4.0 Web Server (http://www.microsoft.com/iis), National Instruments LabVIEW, and gnuplot (for generation of the figures) to run at the same time. These applications controlled the Web interface and data acquistion/processing. Coordination and execution of all tasks (Fig. 3.14) were conducted in the Web Server environment, by making use of HTML and Perl [Asbury et al. 1997] scripts. Data acquisition and basic signal conditioning (e.g., filtering) were carried out by LabVIEW. A PERL script was utilized for updating the gif image of the digital strain data on the website (Fig. 3.9). The script also searched through the data files determining the maximum values and stored these within a simple database (Fig. 3.15) as a series of individual tables.

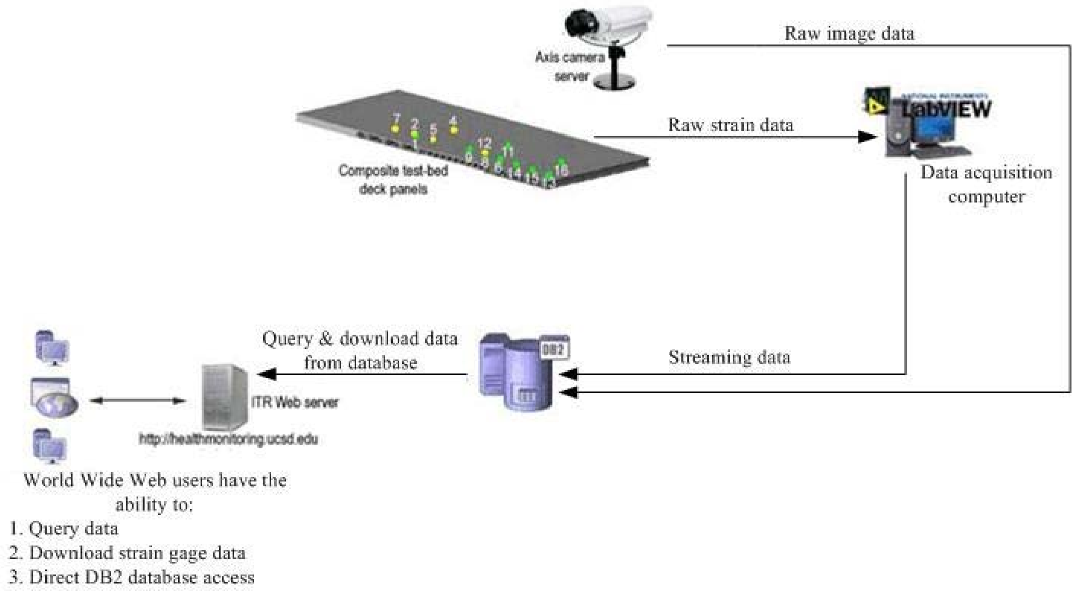
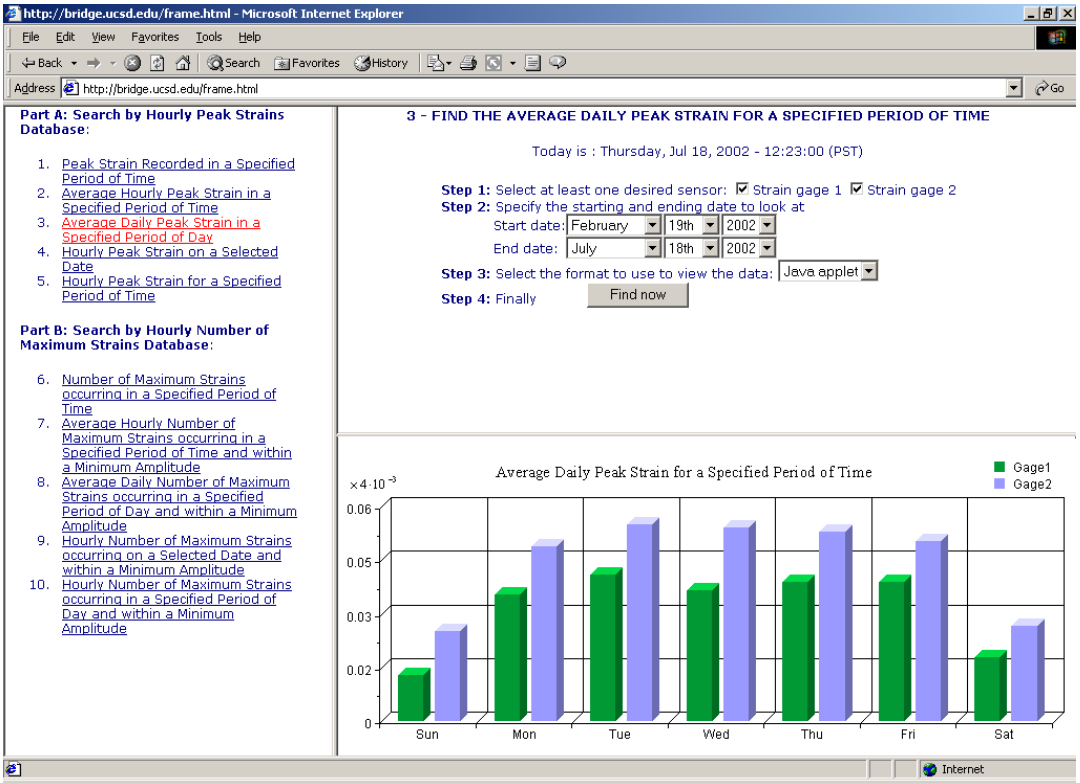

Raw image data
Axis camera server
Raw strain data
LabVIEW
Data acquisition computer
7
2
5
4
1
12
11
9
8
6
14
15
13
16
Composite test-bed deck panels
Streaming data
DB2
Query & download data from database
ITR Web server
http://healthmonitoring.ucsd.edu
World Wide Web users have the ability to:
1. Query data
2. Download strain gage data
3. Direct DB2 database access

Figure 3.14: Framework for Pilot 2-Channel Peak Hourly Strain Monitoring System


http://bridge.ucsd.edu/frame.html - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Go
Internet
Average Daily Peak Strain for a Specified Period of Time
Gage1
Gage2
×4·10 -3
0.06
0.05
0.03
0.02
0
Sun
Mon
Tue
Wed
Thu
Fri
Sat

Address: http://bridge.ucsd.edu/frame.html

Part A: Search by Hourly Peak Strains Database:

1. Peak Strain Recorded in a Specified Period of Time
2. Average Hourly Peak Strain in a Specified Period of Time
3. Average Daily Peak Strain in a Specified Period of Day
4. Hourly Peak Strain on a Selected Date
5. Hourly Peak Strain for a Specified Period of Time

Part B: Search by Hourly Number of Maximum Strains Database:

6. Number of Maximum Strains occurring in a Specified Period of Time
7. Average Hourly Number of Maximum Strains occurring in a Specified Period of Time and within a Minimum Amplitude
8. Average Daily Number of Maximum Strains occurring in a Specified Period of Day and within a Minimum Amplitude
9. Hourly Number of Maximum Strains occurring on a Selected Date and within a Minimum Amplitude
10. Hourly Number of Maximum Strains occurring in a Specified Period of Day and within a Minimum Amplitude

3 - FIND THE AVERAGE DAILY PEAK STRAIN FOR A SPECIFIED PERIOD OF TIME

Today is : Thursday, Jul 18, 2002 - 12:23:00 (PST)

Step 1: Select at least one desired sensor: ☑ Strain gage 1 ☑ Strain gage 2
Step 2: Specify the starting and ending date to look at
Start date: February 19th 2002
End date: July 18th 2002
Step 3: Select the format to use to view the data: Java applet
Step 4: Finally [Find now]

Figure 3.15: Pilot Web-Based Interface for Archived Strain Database.

Subsequent monitoring systems separated the data acquisition from the data streaming and relied on additional computers for data archiving and dissemination. Similar PC-based acquisition hardware was again utilized in these efforts. The analog to digital converter currently in use is a National Instruments (http://ni.com/) PCI-6031E analog I/O board (Fig. 3.16) with the following characteristics: 100 kS/s, 16-bit, thirty-two differential analog inputs, 2 differential analog outputs, ± 10 V output range. This board, which is controlled through the LabVIEW software, digitizes the analog signal (voltage) from the sensors at a programmable rate (in the current setup 200 Hz) and places the data in a circular buffer.

When a specified number of samples are collected in the buffer, the data is loaded into the LabVIEW software and basic signal processing operations are conducted. This includes offset removal (using either the mean or median values), scaling (converting from voltages to units of strain), and filtering. The data once processed is saved to disk as an ASCII file. While these operations are taking place, new data continues to load into the circular buffer. Once the write is completed, the data is loaded into either a second LabVIEW VI or by a PERL script which handles the data streaming/transfer (discussed further in the following section).

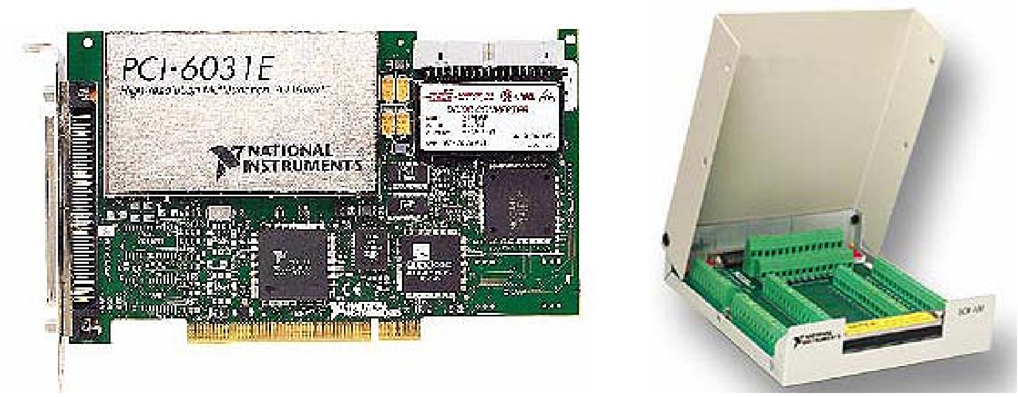


Figure 3.16: National Instruments PCI-6031E Data Acquisition Board and SCB-100 Connector Block

Currently, an Axis 2120 Network camera (Figure 3.12) is employed on the UCSD composite bridge-decks. Image acquisition is controlled directly by the camera’s built in Internet web server, which is connected directly to a 100 megabit Ethernet network.   In the basic mode of operation, the URL of the camera is used to access the images and the internal web pages of the camera. These web pages allow for setting the acquired image size/compression and controlling the lighting/brightness/white balance (Fig 3.17), configuring network settings (Fig. 3.18). Operating in this mode, there is no need for an additional PC, as the unit acts as its own stand alone web server. The acquisition frame rate is set through the web page shown in Fig. 3.19. In theory, this camera should be able to acquire up to 25/30 images/sec, with the time-stamp embedded in the image filename; however, because of the limited processing capabilities of the camera’s internal processor and use of File Transfer Protocol to stream the images, consistent frame rates of only 4 FPS were achieved.

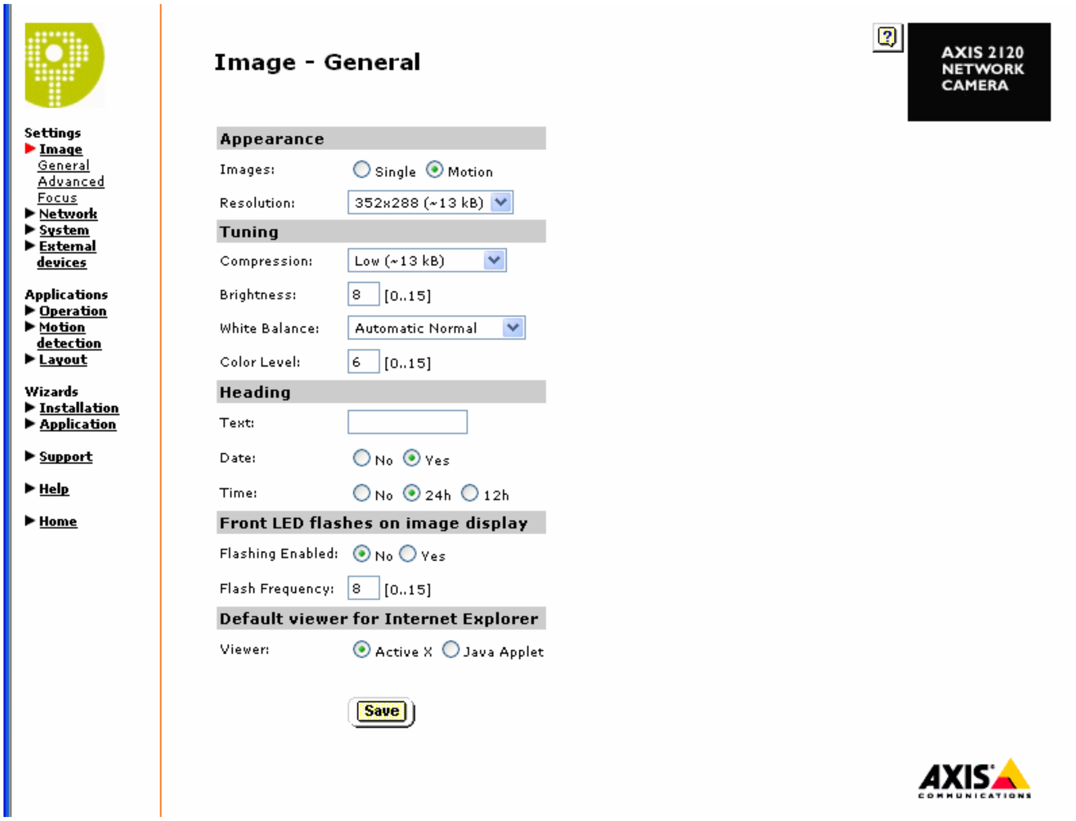
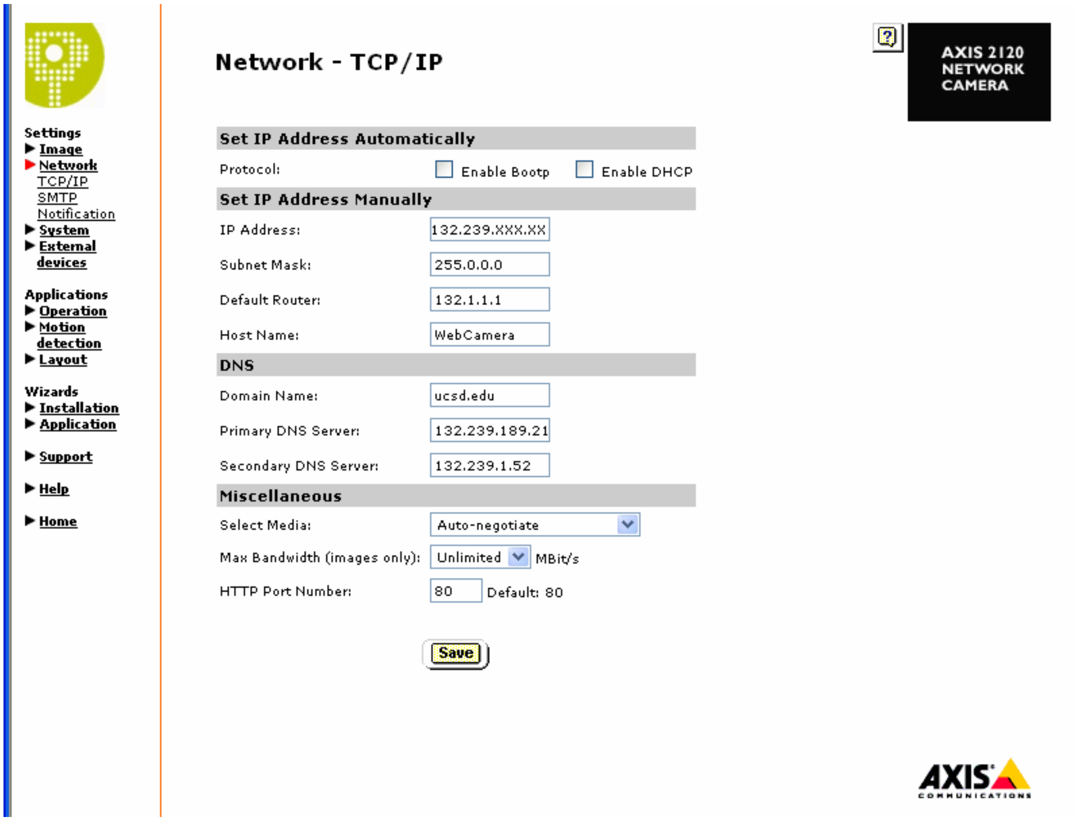


Image - General

Appearance	
Images:	○ Single ● Motion
Resolution:	352x288 (~13 kB)
Tuning	
Compression:	Low (~13 kB)
Brightness:	8 [0..15]
White Balance:	Automatic Normal
Color Level:	6 [0..15]
Heading	
Text:	
Date:	○ No ● Yes
Time:	○ No ● 24h ○ 12h
Front LED flashes on image display	
Flashing Enabled:	● No ○ Yes
Flash Frequency:	8 [0..15]
Default viewer for Internet Explorer	
Viewer:	● Active X ○ Java Applet

Figure 3.17: Internal Web Page on Network Camera for Setting Image Quality.



Network - TCP/IP

Set IP Address Automatically

Protocol: □ Enable Bootp □ Enable DHCP

Set IP Address Manually

IP Address:	132.239.XXX.XX
Subnet Mask:	255.0.0.0
Default Router:	132.1.1.1
Host Name:	WebCamera

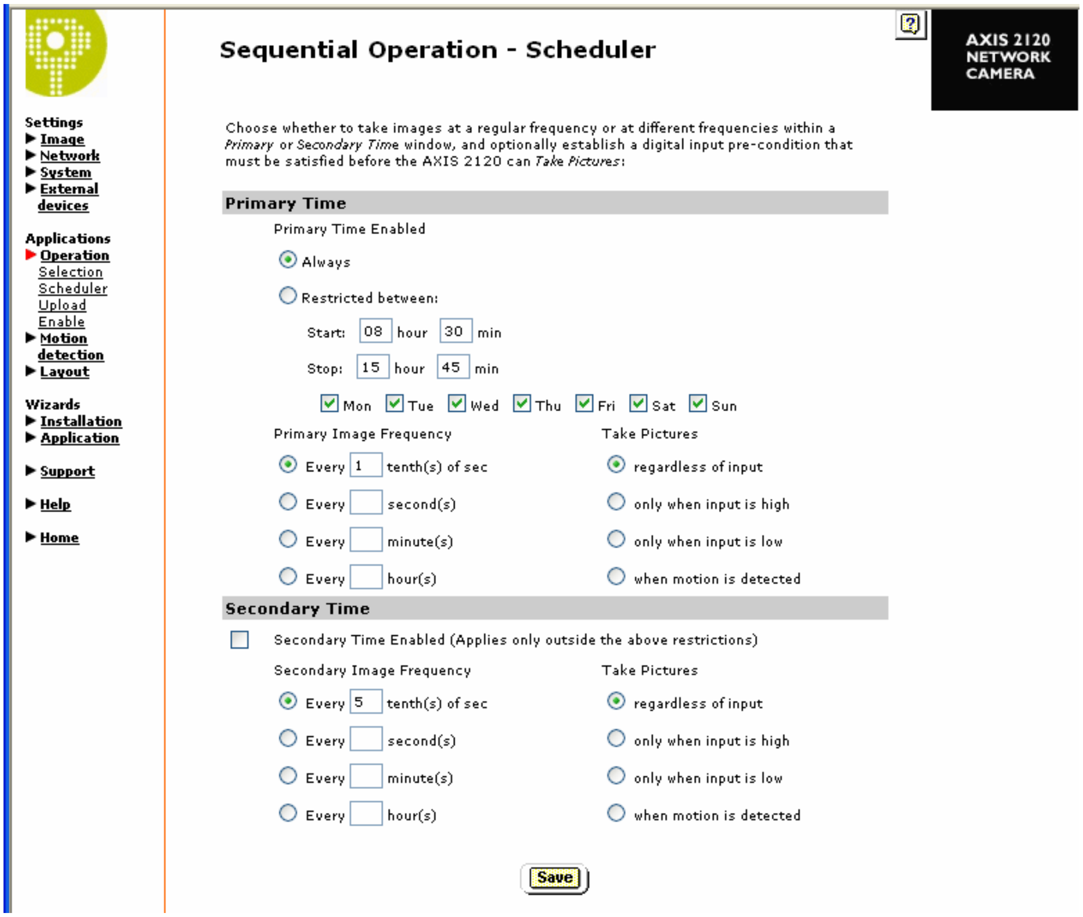
DNS

Domain Name:	ucsd.edu
Primary DNS Server:	132.239.189.21
Secondary DNS Server:	132.239.1.52

Miscellaneous

Select Media:	Auto-negotiate
Max Bandwidth (images only):	Unlimited MBit/s
HTTP Port Number:	80 Default: 80

Figure 3.18: Internal Web Page on Network Camera for Configuring Network Settings



Sequential Operation - Scheduler

Choose whether to take images at a regular frequency or at different frequencies within a *Primary* or *Secondary Time* window, and optionally establish a digital input pre-condition that must be satisfied before the AXIS 2120 can *Take Pictures*:

Primary Time

Primary Time Enabled

● Always

○ Restricted between:

Start: 08 hour 30 min

Stop: 15 hour 45 min

☑ Mon ☑ Tue ☑ Wed ☑ Thu ☑ Fri ☑ Sat ☑ Sun

Primary Image Frequency	Take Pictures
● Every 1 tenth(s) of sec	● regardless of input
○ Every second(s)	○ only when input is high
○ Every minute(s)	○ only when input is low
○ Every hour(s)	○ when motion is detected

Secondary Time

□ Secondary Time Enabled (Applies only outside the above restrictions)

Secondary Image Frequency	Take Pictures
● Every 5 tenth(s) of sec	● regardless of input
○ Every second(s)	○ only when input is high
○ Every minute(s)	○ only when input is low
○ Every hour(s)	○ when motion is detected

Figure 3.19: Network Camera Web Page for Setting the Image Acquisition Rate

Finally, a generic analog microphone was place near the composite decks to record the sounds of passing traffic. This microphone was connected to the “microphone in” port of the data acquisition computer and was recorded using an additional LabVIEW VI. This VI controls the sampling rate of the audio, acquires the audio using a buffered acquisition, and converts each audio time history into a compressed wav file. Converting to a wav file, as opposed to saving the data as an ASCII file, provided significant savings in file size.

A schematic of the experimental setup, showing the interaction of the various components in the acquisition, streaming, and archiving phases is shown in Figure 3.20. Synchronizing the data from each of these sensors is of particular interest, especially when attempting to fuse the data from multiple instruments (i.e., comparing each of the strain time histories with the corresponding video). Within the composite deck monitoring framework, NTP servers were utilized for synchronizing the data acquisition computer and network camera. For the network camera, this was accomplished using the webpage shown in Fig. 3.21, which allows for specifying an NTP server for connecting the camera to.

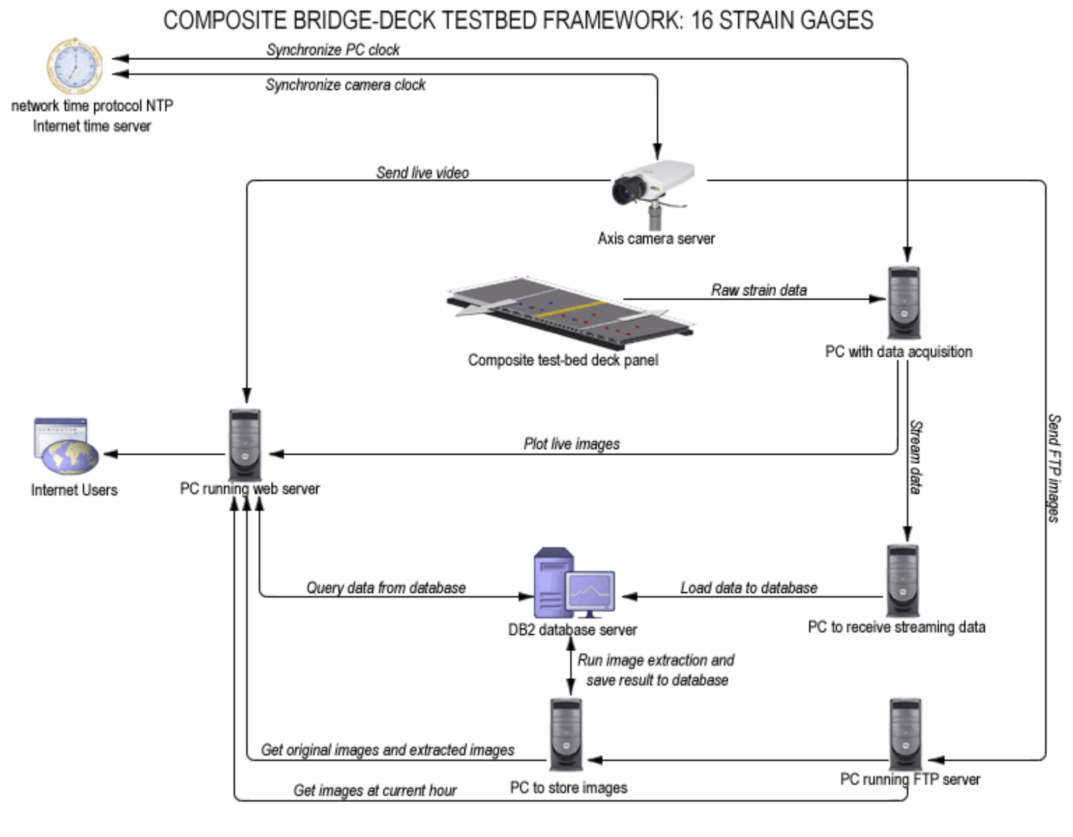
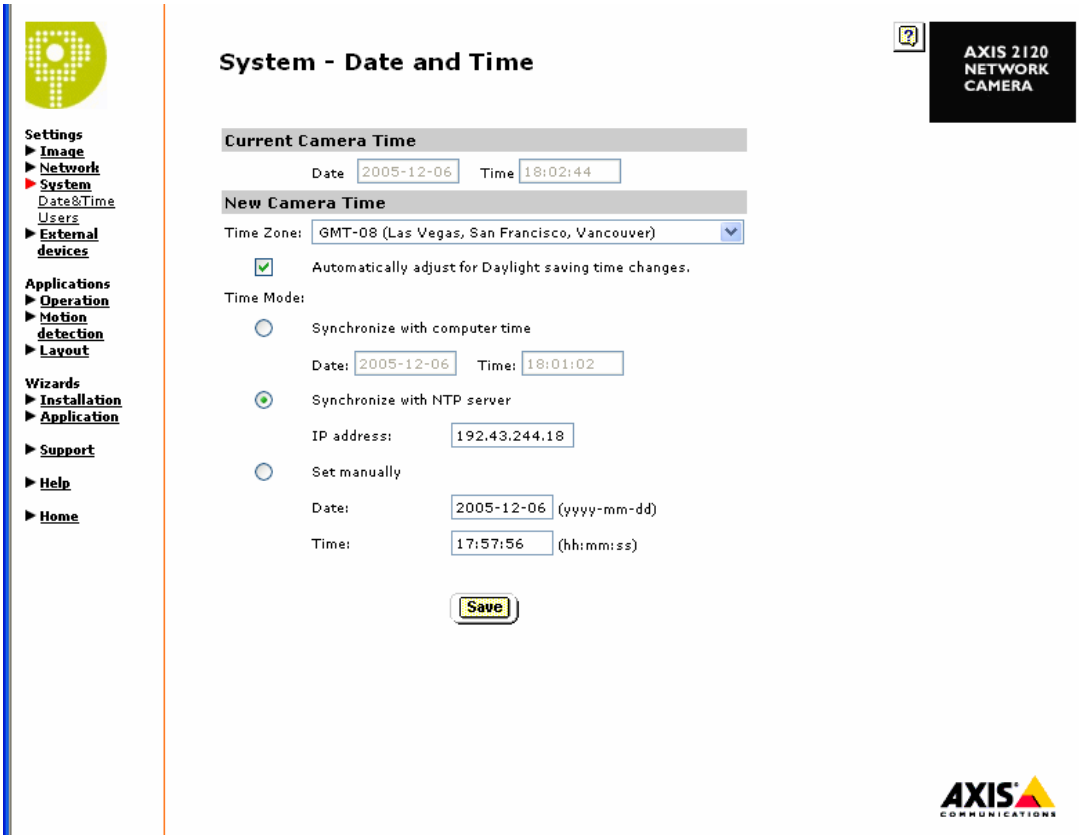


Figure 3.20: Schematic of Experimental Setup for Acquiring, Streaming, and Archiving Data from the Composite Bridge-Deck Testbed



System - Date and Time

Current Camera Time

Date 2005-12-06 Time 18:02:44

New Camera Time

Time Zone: GMT-08 (Las Vegas, San Francisco, Vancouver)

☑ Automatically adjust for Daylight saving time changes.

Time Mode:

○ Synchronize with computer time

Date: 2005-12-06 Time: 18:01:02

● Synchronize with NTP server

IP address: 192.43.244.18

○ Set manually

Date: 2005-12-06 (yyyy-mm-dd)

Time: 17:57:56 (hh:mm:ss)

Figure 3.21: Internal Web Page on Network Camera for Configuring NTP Time Synchronization Settings

For the data acquisition computer, this proved more troubling as early on experiments with commercial software, such as Absolute Time Server (http://www.adjusttime.com/atcs.php), proved unreliable. Problems with this class of software are associated with programs utilizing the computer's system clock, like LabVIEW when time-stamping recorded data and controlling the acquisition timing, which have problems when the system clock updates in the middle of an acquisition. Most times in these instants, the LabVIEW software simply ignored the updated time. It was found for the time change to take affect in the LabVIEW software, the data acquisition program had to be stopped and then restarted. Primarily for this reason, it was decided to incorporate the NTP time synchronization directly into the data acquisition program. LabVIEW can make use of the included TCP functions to establish a connection with an NIST time server and using a single timing request on the signal server read the necessary information to reset the computer's system clock. A sample of this protocol is shown in Figure 2.30 – 2.32. The code checks the current time on the system clock and compares this number against the value for the previous iteration (buffer). If the hour is unchanged, then no action is taken; however, if the hour has changed, then a connection with NIST time server is established and using the Read Time subVI in LabVIEW the clock is reset.

With both systems, network latency led to occasional errors in the time synchronization. These errors are evident when the recorded video is played alongside the strain time histories (Fig. 3.22). The peaks in the strains correspond to the vehicle axles passing over the strain gages; however, when comparing the time history with the video it is not until after a few seconds after the strain gages have recorded the peak corresponding to the second axle crossing the middle of the decks that the bus arrives at this point in the video. It is worth noting that with the PC-based data acquisition hardware, it is impossible to simply set the time on the NTP clock once and only make adjustments for daylight savings changes. On the bridge-deck monitoring project, it was observed that the data acquisition process consumed enough system resources that the Windows system clock slowed down and began to lag the actual time. Over a very short period of time, the error tended to become significant, thereby necessitating frequent updates; however, these errors did not interfere with associating the video clips with the associated strain time histories. The errors associated with time synchronization have been addressed in the developed system employed on the Voigt Bridge testbed and are discussed in Chapter 9.

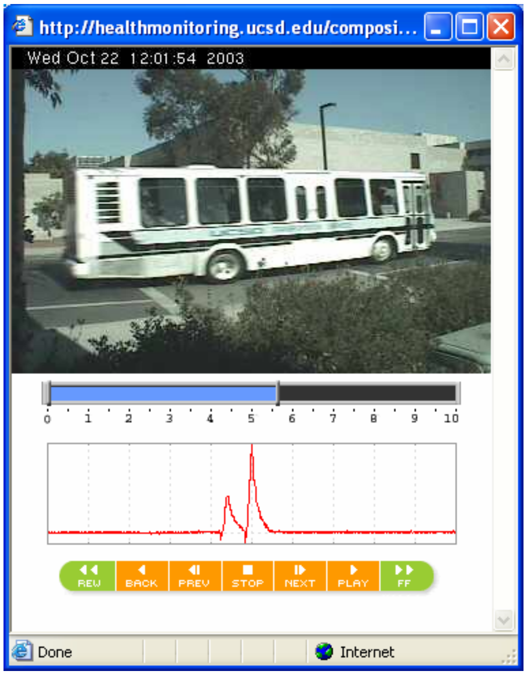


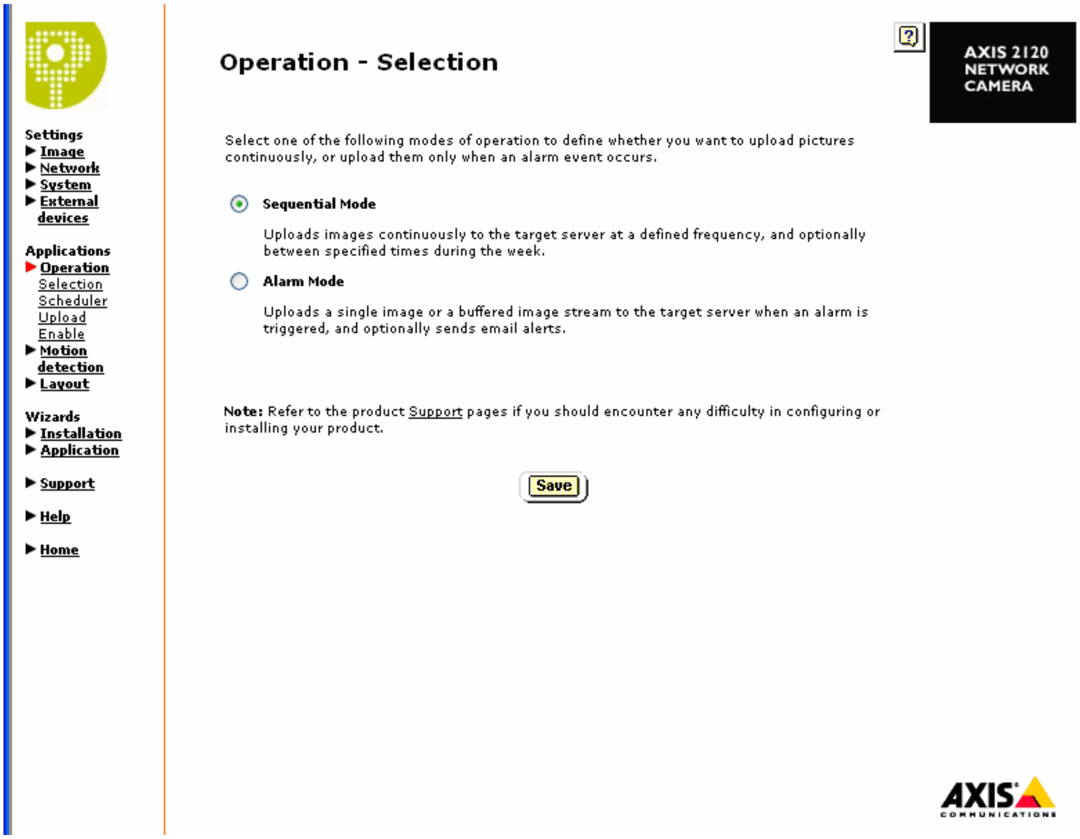
Figure 3.22: Recorded Video and Strain Time History Showing an Error in the Time Synchronization

3.3.3 Data Streaming

Within the framework developed for the composite-bridge deck monitoring, streaming the data recorded by the strain gages, video camera, and microphone is accomplished using three distinctly different methods. The simplest of the data transmission operations is the network camera, followed closely by the audio, and then strain gages, which is by far the most complex.

File Transfer Protocol (FTP) is utilized to transfer the images from the installed network camera to a remote computer where the images are archived. This streaming process is controlled through the internal processor on the network camera. Through a pair of built in web pages, the camera is configured to first operate in sequential mode (Fig. 3.23), whereby images from the camera are uploaded continuously to a target server at a defined frame rate (as opposed to triggered operation). Next, the transfer protocol, remote server, and image path/filename settings are configured (Fig. 3.24). Within this page, the camera allows the time stamp coming from the NTP synchronized clock to be used for creating each image filename.

On the destination side of this operation, a PC-based system with an enabled FTP server is required. The employed server must allow write access to the camera, and for security reasons; access to the FTP site was restricted solely to the network camera’s IP address and required appropriate user name and password (easily configured on the camera). It was found unprotected FTP sites set up on high-speed networks (like the one available at UCSD) tend to draw unwanted off-campus traffic.



Operation - Selection

Select one of the following modes of operation to define whether you want to upload pictures continuously, or upload them only when an alarm event occurs.

● Sequential Mode

Uploads images continuously to the target server at a defined frequency, and optionally between specified times during the week.

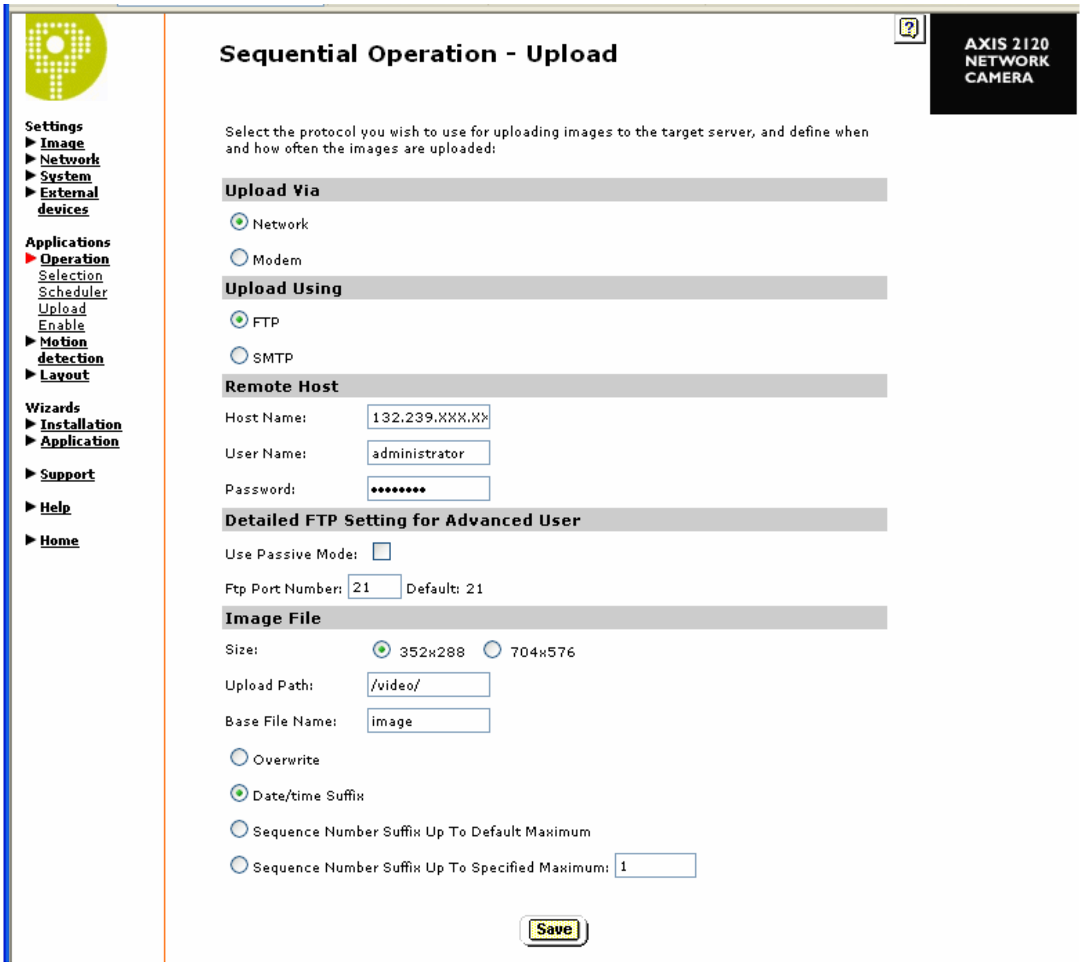
○ Alarm Mode

Uploads a single image or a buffered image stream to the target server when an alarm is triggered, and optionally sends email alerts.

Note: Refer to the product Support pages if you should encounter any difficulty in configuring or installing your product.

Save

Figure 3.23: Network Camera Web Page for Configuring Camera Operation



Sequential Operation - Upload

Select the protocol you wish to use for uploading images to the target server, and define when and how often the images are uploaded:

Upload Via

● Network

○ Modem

Upload Using

● FTP

○ SMTP

Remote Host

Host Name:	132.239.XXX.XX
User Name:	administrator
Password:	••••••••

Detailed FTP Setting for Advanced User

Use Passive Mode: □

Ftp Port Number: 21 Default: 21

Image File

Size:	● 352x288 ○ 704x576
Upload Path:	/video/
Base File Name:	image

○ Overwrite

● Date/time Suffix

○ Sequence Number Suffix Up To Default Maximum

○ Sequence Number Suffix Up To Specified Maximum: 1

Figure 3.24: Network Camera Web Page for Configuring Transfer Protocol, Remote Server, and Image Path/Filename Settings

For transferring the Wav files created from the audio data by the LabVIEW software, a mapped network drive was utilized. Within the LabVIEW software, when the audio buffer is filled and the data is converted to a wav file, a sequential loop first saves the audio file to the local hard drive. Next, using the “Copy File” command, the file is moved over the campus network to a shared folder on the archiving computer. Pending successful transmission of the Wav file, it is deleted from the acquisition computer and the sequence is repeated for the next file. This proved to be the simplest method for transferring data between the two computers located on campus. For remote installations (off campus), the Wav file would instead be saved to the local hard drive off the acquisition computer and then transferred using FTP rather than a mapped network drive.

The last remaining portion of data, the strain gage data, proved to be the most difficult to stream. The first method utilized for transferring the strain data recorded by the 2 Channel setup was done using mapped network drives. Starting at the composite deck site, the analog signals generated by the two strain gages are continuously digitized on the data acquisition computer through utilization of National Instruments LabVIEW software (http://www.ni.com/labview/). Once digitized, the signals are stored in a buffer. When five seconds of data (or 1000 samples taken at a rate of 200 samples per second) collects in the buffer, the data is output and appended to the previous fifteen seconds of data. These two strain time histories (each twenty seconds long) are then saved to disk on the data acquisition computer as an ASCII formatted text file (data.txt). Additionally, a second text file (t0.txt) is created to save the time stamp (start date and time) of the time history. In the data acquisition iterations that follow, the strain and date/time filenames remain unchanged and the program simply overwrites the old data.

Implementing a set of ‘while loop’ commands, a permanently running Perl script loads the file data.txt. To ensure that the entire time history of a vehicle is captured, the Perl script scans only a five second window of strains (between 7 and 12 seconds in the 20 second time history) for the peak absolute strains in both gages (Fig. 3.25).

If the peak absolute strain for either gage exceeds the previously recorded values for the current hour, the old peak strain time histories are overwritten by the new ones in the Peak Hourly Strain (PHS) text files. Note, each of these three files (maxData1hour1.txt, maxData1hour2.txt, maxData1hour2b.txt) contain a single column of data with 4000 rows: (1) the PHS time history for Gage 1, (2) the PHS time history for Gage 2, and (3) the Gage 1 strain time history associated with the peak hourly Gage 2 data, respectively. The two time stamps corresponding to these files are stored separately.

At the end of each hour, the PHS time histories are loaded into the DB2 database on a networked computer. A Perl script, called on by Windows Task Scheduler, accomplishes this by reading the strain data and time stamps (from maxData1hour1.txt, maxData1hour2.txt, maxData1hour2b.txt), and loading the data into the database.

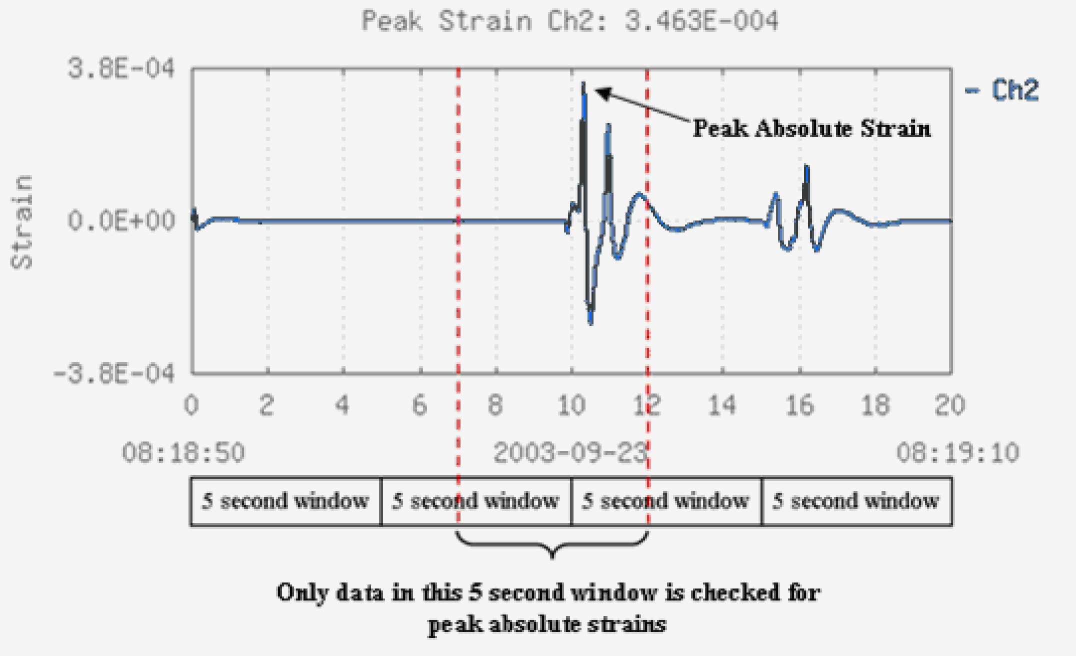


Figure 3.25: Twenty Second Strain time History Composed of 4 Five Second Windows

While this method has the advantage of utilizing local processing to minimize the amount of data that must be transferred, it was deemed cumbersome. Consequently, in later work with acquiring data from all 16 strain gages, an alternate transmission methodology was employed. As with 2 channel setup, the analog signals generated by the sixteen strain gages are continuously digitized on the data acquisition computer through utilization of National Instruments LabVIEW software and stored in a buffer. When five seconds of data (or 1000 samples taken at a rate of 200 samples per second) collects in the buffer, the data is not saved this time. Instead, when the data comes out of the buffer, basic signal processing is performed and the data is then placed into a packet and transferred to a remote computer using Transmission Control Protocol / Internet Protocol (TCP/IP). As discussed in Section 2.6, TCP/IP makes it possible to communicate over single networks or interconnected networks (Internet), even over large geographical distances. Further, because TCP/IP is available on most computers, it can transfer information between diverse systems. With a TCP/IP connection, it is not necessary to save any of the acquired data to the local disk. Instead, the data can be streamed directly from the system memory. In addition to providing faster data streaming rates, this method also reduces the requirements on the data acquisition computer.

A necessary component of the TCP/IP data stream was the development of a program to read the data from the specified TCP/IP data port and to load it into the database. Within the setup currently being presented, this was accomplished using a PERL-based loader program. This program reads and stores 10,000 time steps of data from the TCP/IP connection and stores them to a buffer. Once the 10,000th row is read, the program loads all of the data into the DB2 database, clears the buffer, and restarts the process. This insertion process (10,000 rows) takes approximately 20 seconds on a conventional PC desktop.

An additional refinement was made later to the employed strain data transfer program. For remote monitoring applications, it is expected that network problems will be encountered. These could be in the form of the network speed dropping or becoming unavailable all together for extended periods of time. In these instants, the TCP/IP program built into the data acquisition will be unable to transmit all of the data. Consequently, incoming data will continue to collect in the buffer until it overflows and causes the whole acquisition program to crash. This was addressed by separating the data acquisition and data streaming program into separate files. Now, the data is digitized, processed, and then saved to disk as a LabVIEW global variable. Once the write is completed, the data is loaded into a second VI which packages the data and sends it out using TCP/IP as before. Pending successful transmission, the old data is deleted from the acquisition computer. If there are any problems with the data transfer, the TCP/IP connection is closed, the computer waits 2 seconds, re-opens the TCP/IP connection, and then tries sending the data again. This process is repeated until the data is successfully sent. This method, best utilizes the available bandwidth on networks with variable performance.

3.4 Data Archiving

As has been repeatedly emphasized in the previous chapters, it is becoming the situation where acquiring and processing data is no longer the biggest problem. Difficulties are arising in managing the massive amounts of data coming from these newly emerging monitoring systems. In particular, how to make sense of the data as it arrives and how to get it into the analysis software? To address these issues, a series of databases were constructed to house the flow of sensor and video data from the UCSD composite bridge-deck testbed (http://healthmonitoring.ucsd.edu)

3.4.1 Database Architecture for 2-Channel Peak Hourly Strain Monitoring System

When a peak hourly strain (PHS) is detected by one of the two strain gages that monitor traffic crossing over the composite decks, the corresponding strain time histories and video are saved to disk and uploaded into the DB2 database. (Figure 3.26 maps the strain data archiving procedure from the composite decks to the database, the details of which have been described in the preceding sections.) Within this database, the time-stamped peak hourly strains for Gages 1 and 2 (Fig. 3.9 and 3.25) are recorded. Their corresponding strain time histories are stored in separate files. In addition, the strain data in Gage 1 is recorded whenever Gage 2 data is logged.

The DB2 database contains three metadata tables (tb_peak1, tb_peak2, tb_peak2b), which are updated hourly with the peak absolute strain and the time at which it occurred, as well as separate hourly strain time history tables. For each hour, a new row is inserted into the three metadata tables. Each row consists of two values, the mpeak and mdate (Fig. 3.27). The mpeak is the peak absolute strain, and the mdate is the corresponding date/time information (yyyy-mm-dd-hh.mm.ss) when the peak strain was recorded. Simultaneously, a new table is created each hour to store the twenty second strain time histories (4000 rows). These tables each contain three columns - (1) mvalue1 (the PHS time history for Gage 1), (2) mvalue2 (the PHS time history for Gage 2), and (3) mvalue2b (the Gage 1 strain time history associated with the peak hourly Gage 2 data), respectively. The names of these tables (tb_yyyy_mm_dd_hh) are constructed using the date/time information (i.e. the table for 2 PM on October 4, 2004 is called tb_2004_10_04_14). It is worth noting that both gages do not necessarily experience peak absolute hourly strains for the same vehicle. Therefore, the time stamp in tb_peak_1 corresponds to the data in column 1 of the hourly strain time history table. The tb_peak_2 and tb_peak_2b time stamps are identical and are associated with columns 2 and 3 in the hourly strain time history table.

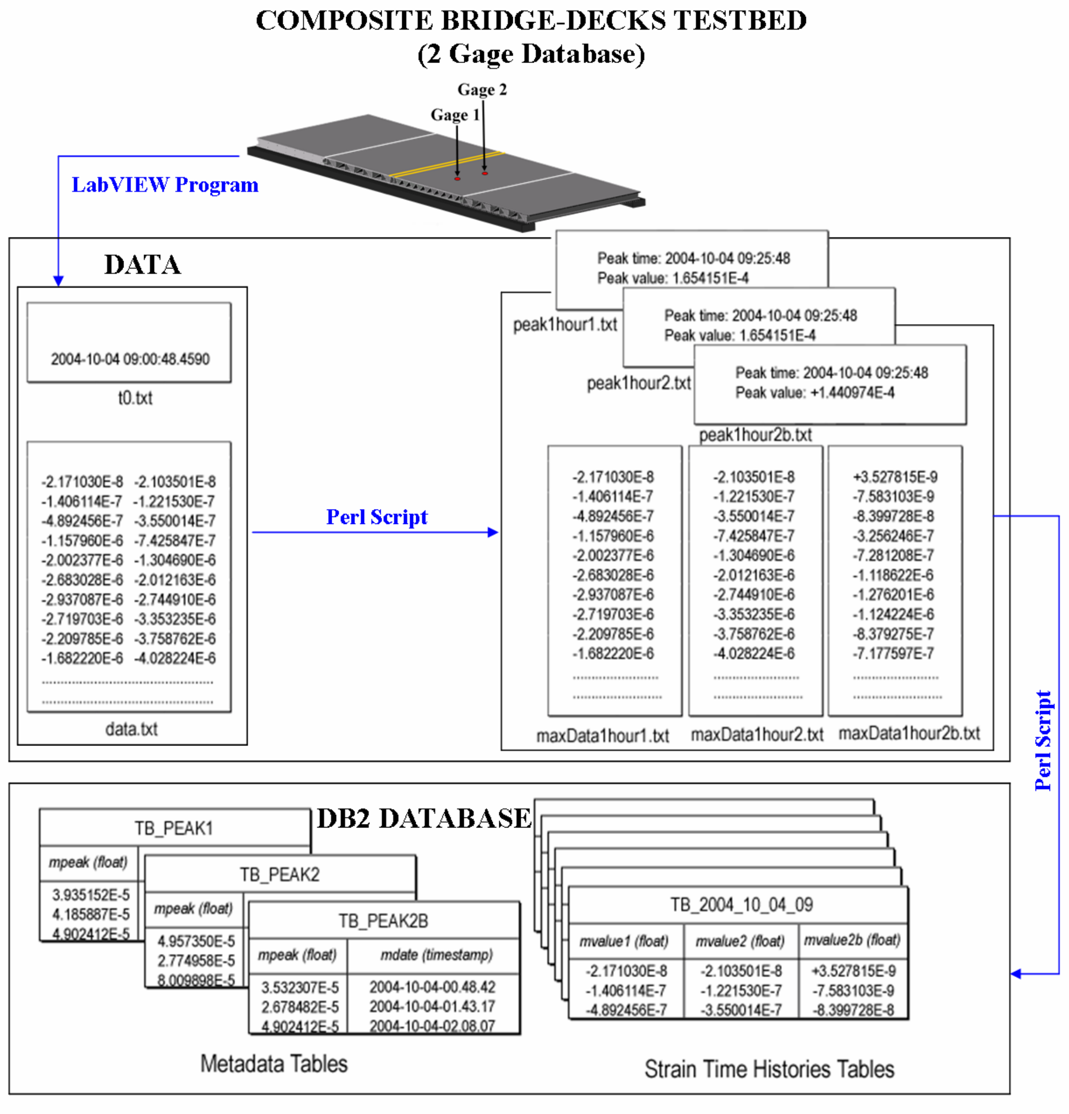
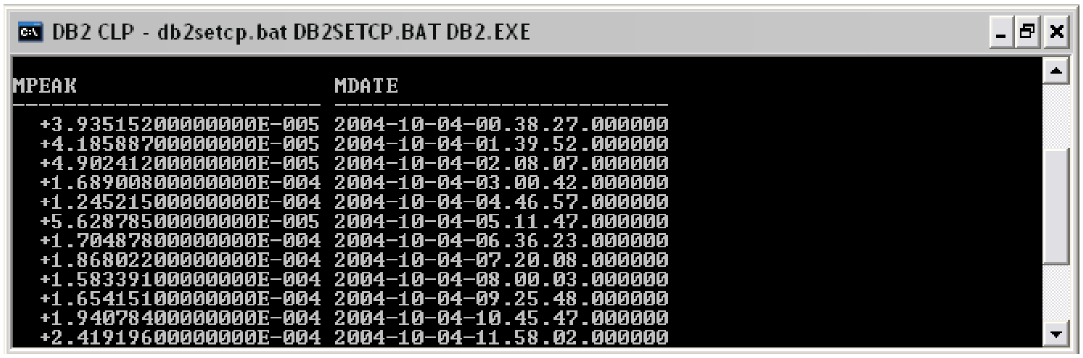


Figure 3.26: Two-Channel Peak Hourly Strain Data Archiving Architecture.

On the hour, all video images for the previous hour are loaded from temporary storage into an image archiving computer. A second Perl script is called to isolate video image sequences corresponding to the peak strain time histories for the past hour. Each twenty-second image sequence (composed of eighty jpg images taken approximately 0.25 seconds apart) is saved to disk using a naming scheme, which indicates the year, month, day, hour, minute, and second that the sequence started, as well as the individual frame number in the sequence (for example, image_20041004092528_01.jpg). Matlab (http://mathworks.com/products/matlab) then performs image processing (discussed in detail in Chapter 4) on the two sets of images, and outputs the results directly into two tables in the database (tb_image_extraction1 and tb_image_extraction2). The image processing outputs, per frame, include the image ID, the image area, the vertical and horizontal components of the vehicle centroid, and the width and height dimensions of the bounding box that encompasses the vehicle. Therefore, 160 rows are added to the table tb_image_extraction1 each hour. The table tb_image_extraction2 contains only the image ID and peak vehicle areas detected (i.e. 1 row of metadata is added each hour).



```
MPEAK                    MDATE
------------------------ --------------------------
  +3.93515200000000E-005 2004-10-04-00.38.27.000000
  +4.18588700000000E-005 2004-10-04-01.39.52.000000
  +4.90241200000000E-005 2004-10-04-02.08.07.000000
  +1.68900800000000E-004 2004-10-04-03.00.42.000000
  +1.24521500000000E-004 2004-10-04-04.46.57.000000
  +5.62878500000000E-005 2004-10-04-05.11.47.000000
  +1.70487800000000E-004 2004-10-04-06.36.23.000000
  +1.86802200000000E-004 2004-10-04-07.20.08.000000
  +1.58339100000000E-004 2004-10-04-08.00.03.000000
  +1.65415100000000E-004 2004-10-04-09.25.48.000000
  +1.94078400000000E-004 2004-10-04-10.45.47.000000
  +2.41919600000000E-004 2004-10-04-11.58.02.000000
```

Figure 3.27: Sample DB2 Database Peak Hourly Strain and Time Stamp Metadata Table (tb_peak1)

3.4.2 Database Architecture for 16-Channel Continuous Strain Monitoring System

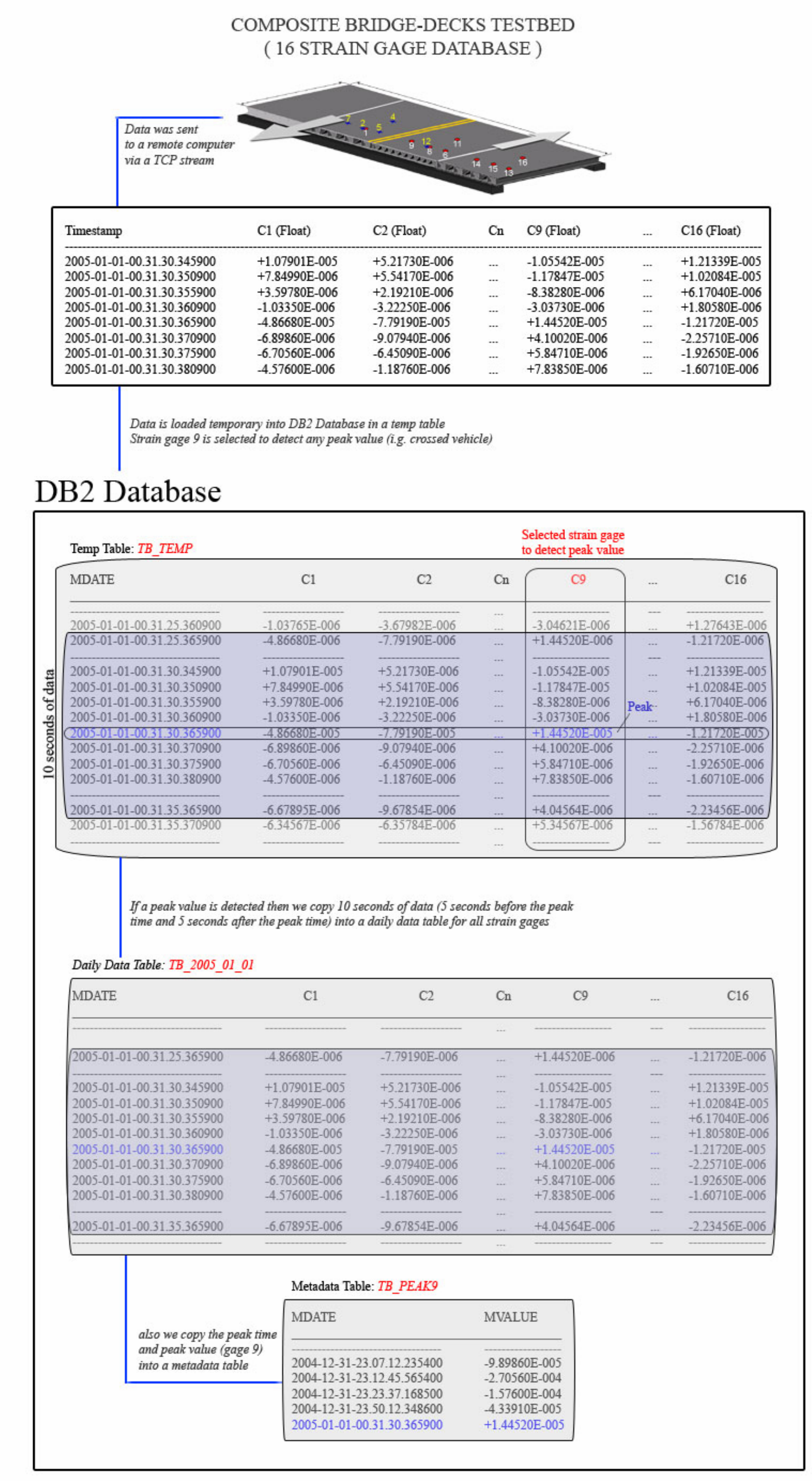
On the hour, A Perl script connects to the database and loads the entire set of strain time histories recorded by the 16-channel monitoring system (sixteen time histories each composed of 720,000 time steps) stored in the temporary buffer table (tb_data_temp). This script uses a peak detection algorithm applied to the data from gage 9 to determine when a vehicle crosses the bridge decks. This detection program relies on finding the peaks corresponding to the vehicle’s axles. To be classified as belonging to a vehicle, the detected peaks must exceed a predefined threshold of $5x10^{-6}$ and to prevent false identification, due to spurious electrical spikes in the data, the strain must exceed the threshold for at least 0.05 seconds. The detection algorithm follows the following operations:

1. Set the threshold limit for Gage 9 ($\varepsilon_{min} = 5 \text{ x } 10^{-6}$) and initial start time (start of hour), T_1 and end time (end of hour), T_2.

2. Within a recursive loop, the time history from T_1 to T_2 is scanned for the peak strain. If the maximum strain is below the threshold, then the program jumps out of the loop. Else, a vehicle is detected and the program finds the time stamp corresponding to the peak strain. The peak strain and time stamp are saved to the metadata table (tb_peak9) and the ten-second strain time history (5-seconds before and after the peak) is saved into a daily table (tb_yyyy_mm_dd).
3. The entire hourly strain time history is then divided into three parts. The first part is composed of the data to the left of the peak strain [T_1 : T_{peak} – 5 seconds]. The second portion of the data is the peak strain which has been loaded into the database [T_{peak} – 5 seconds : T_{peak} + 5 seconds]. The third and final section is the data to the right of the peak strain [T_{peak} – 5 seconds : T_2].
4. Steps 2 and 3 are repeated on the data to the left of the peak until no additional peaks are found.
5. Steps 2 and 3 are repeated on the data to the right of the peak until no additional peaks are found.
6. Finally, all of the data in the temporary table (tb_data_temp) from the start of the hour to the end of the hour is deleted. During this processing, new data continues to be loaded into the table by the data loader Perl script.

Only the data with vehicles crossing are loaded into the database, the remaining data is either deleted or placed into temporary storage and then discarded (after one week).

When the vehicle detection program finishes running, a second Perl script is run to isolate the forty images associated with each vehicle in the metadata table tb_peak9. The Matlab-based image processing program is run and the extracted features (peak area and corresponding vertical and horizontal centroids) are loaded into the metadata table tb_image_extraction16. The framework of this database and the interaction of the various tables is shown in Figure 3.28.



COMPOSITE BRIDGE-DECKS TESTBED
(16 STRAIN GAGE DATABASE)

Data was sent to a remote computer via a TCP stream

Timestamp	C1 (Float)	C2 (Float)	Cn	C9 (Float)	...	C16 (Float)
2005-01-01-00.31.30.345900	+1.07901E-005	+5.21730E-006	...	-1.05542E-005	...	+1.21339E-005
2005-01-01-00.31.30.350900	+7.84990E-006	+5.54170E-006	...	-1.17847E-005	...	+1.02084E-005
2005-01-01-00.31.30.355900	+3.59780E-006	+2.19210E-006	...	-8.38280E-006	...	+6.17040E-006
2005-01-01-00.31.30.360900	-1.03350E-006	-3.22250E-006	...	-3.03730E-006	...	+1.80580E-006
2005-01-01-00.31.30.365900	-4.86680E-005	-7.79190E-005	...	+1.44520E-005	...	-1.21720E-005
2005-01-01-00.31.30.370900	-6.89860E-006	-9.07940E-006	...	+4.10020E-006	...	-2.25710E-006
2005-01-01-00.31.30.375900	-6.70560E-006	-6.45090E-006	...	+5.84710E-006	...	-1.92650E-006
2005-01-01-00.31.30.380900	-4.57600E-006	-1.18760E-006	...	+7.83850E-006	...	-1.60710E-006

Data is loaded temporary into DB2 Database in a temp table
Strain gage 9 is selected to detect any peak value (i.g. crossed vehicle)

DB2 Database

Temp Table: *TB_TEMP*

Selected strain gage
to detect peak value

MDATE	C1	C2	Cn	C9	...	C16
--------	--------	--------	...	--------	---	--------
2005-01-01-00.31.25.360900	-1.03765E-006	-3.67982E-006	...	-3.04621E-006	...	+1.27643E-006
2005-01-01-00.31.25.365900	-4.86680E-006	-7.79190E-006	...	+1.44520E-006	...	-1.21720E-006
--------	--------	--------	...	--------	---	--------
2005-01-01-00.31.30.345900	+1.07901E-005	+5.21730E-006	...	-1.05542E-005	...	+1.21339E-005
2005-01-01-00.31.30.350900	+7.84990E-006	+5.54170E-006	...	-1.17847E-005	...	+1.02084E-005
2005-01-01-00.31.30.355900	+3.59780E-006	+2.19210E-006	...	-8.38280E-006	...	+6.17040E-006
2005-01-01-00.31.30.360900	-1.03350E-006	-3.22250E-006	...	-3.03730E-006	...	+1.80580E-006
2005-01-01-00.31.30.365900	-4.86680E-005	-7.79190E-005	...	+1.44520E-005	...	-1.21720E-005
2005-01-01-00.31.30.370900	-6.89860E-006	-9.07940E-006	...	+4.10020E-006	...	-2.25710E-006
2005-01-01-00.31.30.375900	-6.70560E-006	-6.45090E-006	...	+5.84710E-006	...	-1.92650E-006
2005-01-01-00.31.30.380900	-4.57600E-006	-1.18760E-006	...	+7.83850E-006	...	-1.60710E-006
--------	--------	--------	...	--------	---	--------
2005-01-01-00.31.35.365900	-6.67895E-006	-9.67854E-006	...	+4.04564E-006	...	-2.23456E-006
2005-01-01-00.31.35.370900	-6.34567E-006	-6.35784E-006	...	+5.34567E-006	...	-1.56784E-006
--------	--------	--------	...	--------	---	--------

If a peak value is detected then we copy 10 seconds of data (5 seconds before the peak time and 5 seconds after the peak time) into a daily data table for all strain gages

Daily Data Table: TB_2005_01_01

MDATE	C1	C2	Cn	C9	...	C16
--------	--------	--------	...	--------	---	--------
2005-01-01-00.31.25.365900	-4.86680E-006	-7.79190E-006	...	+1.44520E-006	...	-1.21720E-006
--------	--------	--------	...	--------	---	--------
2005-01-01-00.31.30.345900	+1.07901E-005	+5.21730E-006	...	-1.05542E-005	...	+1.21339E-005
2005-01-01-00.31.30.350900	+7.84990E-006	+5.54170E-006	...	-1.17847E-005	...	+1.02084E-005
2005-01-01-00.31.30.355900	+3.59780E-006	+2.19210E-006	...	-8.38280E-006	...	+6.17040E-006
2005-01-01-00.31.30.360900	-1.03350E-006	-3.22250E-006	...	-3.03730E-006	...	+1.80580E-006
2005-01-01-00.31.30.365900	-4.86680E-005	-7.79190E-005	...	+1.44520E-005	...	-1.21720E-005
2005-01-01-00.31.30.370900	-6.89860E-006	-9.07940E-006	...	+4.10020E-006	...	-2.25710E-006
2005-01-01-00.31.30.375900	-6.70560E-006	-6.45090E-006	...	+5.84710E-006	...	-1.92650E-006
2005-01-01-00.31.30.380900	-4.57600E-006	-1.18760E-006	...	+7.83850E-006	...	-1.60710E-006
--------	--------	--------	...	--------	---	--------
2005-01-01-00.31.35.365900	-6.67895E-006	-9.67854E-006	...	+4.04564E-006	...	-2.23456E-006
--------	--------	--------	...	--------	---	--------

also we copy the peak time and peak value (gage 9) into a metadata table

Metadata Table: *TB_PEAK9*

MDATE	MVALUE
--------	--------
2004-12-31-23.07.12.235400	-9.89860E-005
2004-12-31-23.12.45.565400	-2.70560E-004
2004-12-31-23.23.37.168500	-1.57600E-004
2004-12-31-23.50.12.348600	-4.33910E-005
2005-01-01-00.31.30.365900	+1.44520E-005

Figure 3.28: Sixteen-Channel Strain Data Archiving Architecture

3.4.3 Recorded Data

Captured by the records in both databases are a diverse variety of vehicle types ranging from golf carts to five-axle semi-trucks (Figs. 3.29 and 3.30). For the two-channel peak hourly strain database, there are 25,685 vehicles recorded between 2/19/02 and 4/22/05. In the sixteen-channel continuous monitoring database, there are a total of 439,654 vehicles recorded between 10/9/03 and 3/31/05. The range of strains imposed on the gages varied from approximately 0 to $4.5 \mathrm{x} 10^{-4}$ for the peak hourly strain system and from 0 to $3.0 \mathrm{x} 10^{-4}$ for the 16-channel continuous monitoring system. Each vehicle's strain time histories display a unique response signature where each distinct positive peak indicates a vehicle axle crossing. For example, a comparison of the data recorded by the two-channel and sixteen-channel systems for four typical vehicle types is shown in Figures 3.29 and 3.30. Note, that the sample strain time histories in Figure 3.29 have been filtered using a bandpass filter (3rd order Butterworth lowpass filter with 0.5 Hz low cutoff and 15 Hz high cutoff frequencies) to reduce the appearance of electrical noise present in the original system.

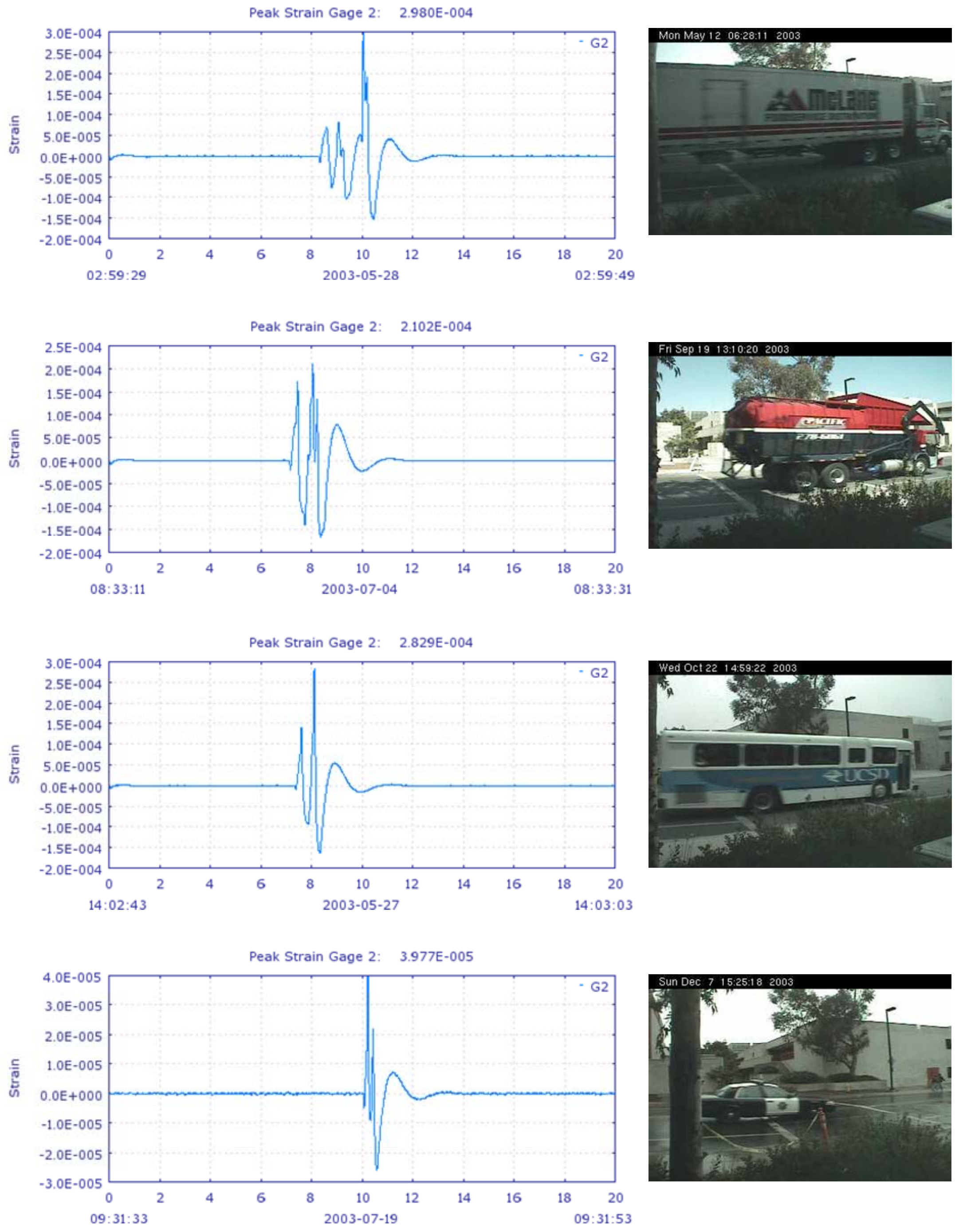


Figure 3.29: Sample of Recorded Data from 2-Channel Peak Hourly Strain Monitoring System

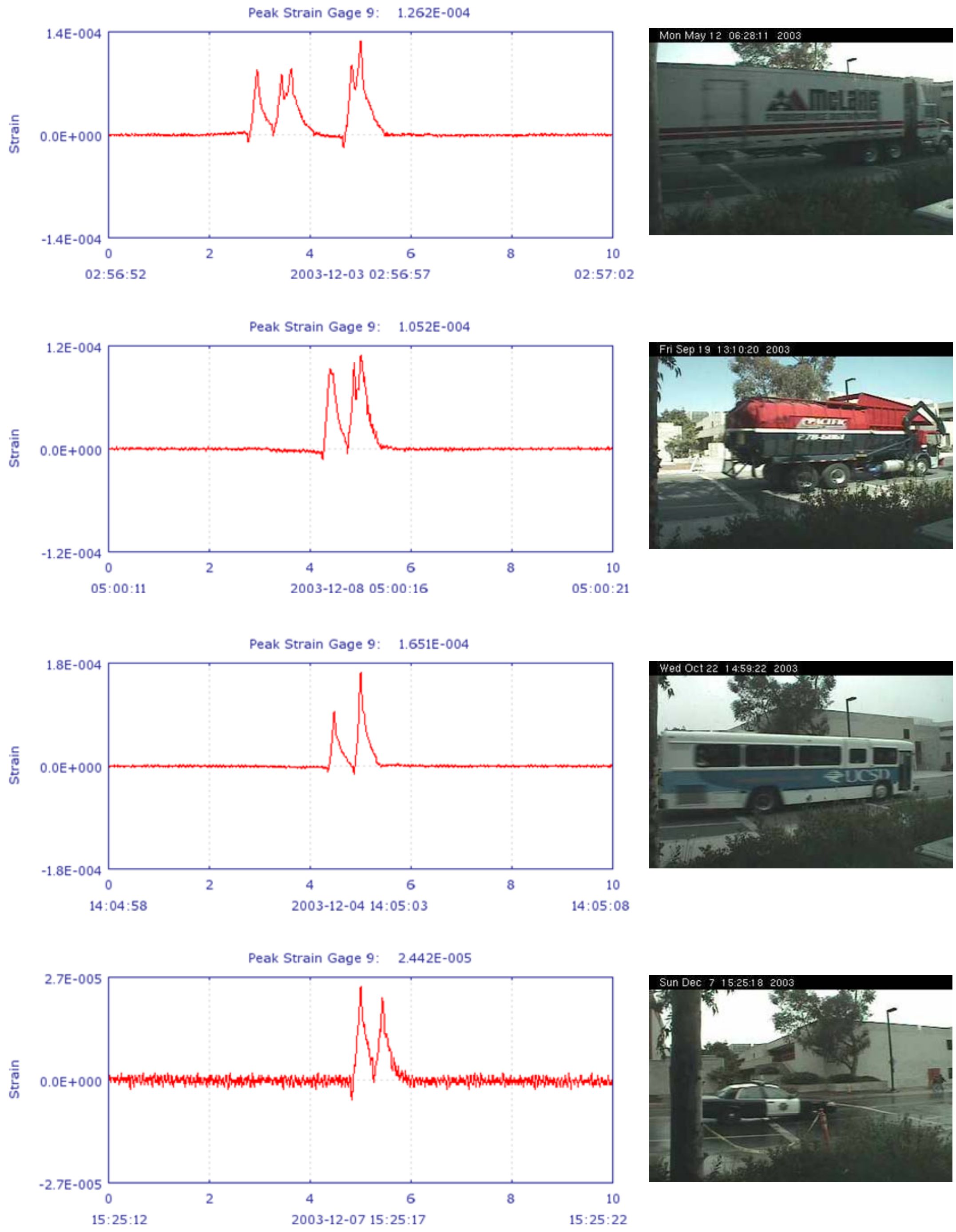
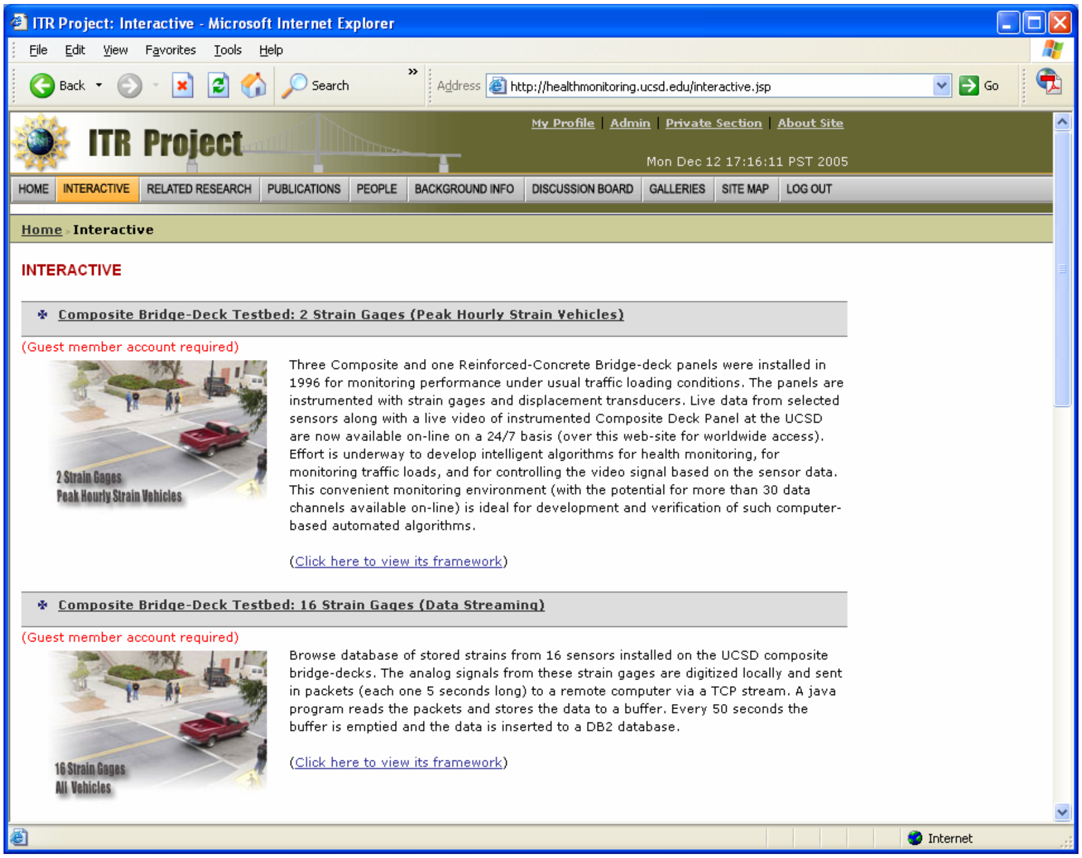


Figure 3.30: Sample of Recorded Data from 16-Channel Continuous Strain Monitoring System

3.5 Web Portal

Work was begun concurrently in several areas in order to build the envisioned high-performance software system infrastructure [Elgamal et al., 2002] required for this project (http://healthmonitoring.ucsd.edu). One of the first steps consisted of using ASIS Java-based Web Service development tools to adapt existing CGI-accessible analysis functions. These functions, which included the ability to query for data containing the peak registered strain over a period of time and extracted features from recorded video (Fig. 3.31). Within these efforts, Web Services is a key technology in helping to build a modular and interoperable analysis, monitoring and response system [Elgamal et al., 2003c].

Within the UCSD testbed, Web Services are being employed to achieve modularity and extensibility between the various components of the system and also between this and other systems. Initially, Web Services seemed most appropriate in the areas of system and sensor network-related configuration and control, as an interface to the data being collected and the analysis routines provided. Also being investigated is the use of Web Services in other areas, including: interaction between the analysis and processing components of the system, data and video streaming, as well secure data access and transport. Furthermore, the Web Services approach provides a vendor-neutral, platform-independent, standards-based and industry supported means of achieving this decoupling.


ITR Project: Interactive - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search
Address http://healthmonitoring.ucsd.edu/interactive.jsp Go
ITR Project
My Profile | Admin | Private Section | About Site
HOME INTERACTIVE RELATED RESEARCH PUBLICATIONS PEOPLE BACKGROUND INFO DISCUSSION BOARD GALLERIES SITE MAP LOG OUT
Home Interactive
2 Strain Gages
Peak Hourly Strain Vehicles
16 Strain Gages
All Vehicles
Internet

Mon Dec 12 17:16:11 PST 2005

INTERACTIVE

Composite Bridge-Deck Testbed: 2 Strain Gages (Peak Hourly Strain Vehicles)

(Guest member account required)

Three Composite and one Reinforced-Concrete Bridge-deck panels were installed in 1996 for monitoring performance under usual traffic loading conditions. The panels are instrumented with strain gages and displacement transducers. Live data from selected sensors along with a live video of instrumented Composite Deck Panel at the UCSD are now available on-line on a 24/7 basis (over this web-site for worldwide access). Effort is underway to develop intelligent algorithms for health monitoring, for monitoring traffic loads, and for controlling the video signal based on the sensor data. This convenient monitoring environment (with the potential for more than 30 data channels available on-line) is ideal for development and verification of such computer-based automated algorithms.

(Click here to view its framework)

Composite Bridge-Deck Testbed: 16 Strain Gages (Data Streaming)

(Guest member account required)

Browse database of stored strains from 16 sensors installed on the UCSD composite bridge-decks. The analog signals from these strain gages are digitized locally and sent in packets (each one 5 seconds long) to a remote computer via a TCP stream. A java program reads the packets and stores the data to a buffer. Every 50 seconds the buffer is emptied and the data is inserted to a DB2 database.

(Click here to view its framework)

Figure 3.31: Web Portal for Accessing Recorded Data from Composite Bridge Decks

3.5.1 Database Querying

Two methods are currently available for providing users access to the data contained within the two composite bridge-deck databases. The first uses a direct connection to the DB2 databases to establish read access. This requires the user to have the DB2 Connect (client) software (http://www.ibm.com/software/data/db2/db2connect/edition-ee.html) installed on their system. This software provides the application programming interface (API) drives and connectivity infrastructure for direct connection from Windows applications to the database servers. Further, database connectivity add-ins for analysis software, like Matlab's Database Toolbox (http://www.mathworks.com/products/database/) and National Instruments LabVIEW Database Connectivity Toolset (http://www.ni.com/toolkits/lv_db_conn.htm), simplify this operation allowing data to be queried from the database and loaded directly into the analysis software for processing. However, there are several disadvantages associated with this manner of accessing the data:

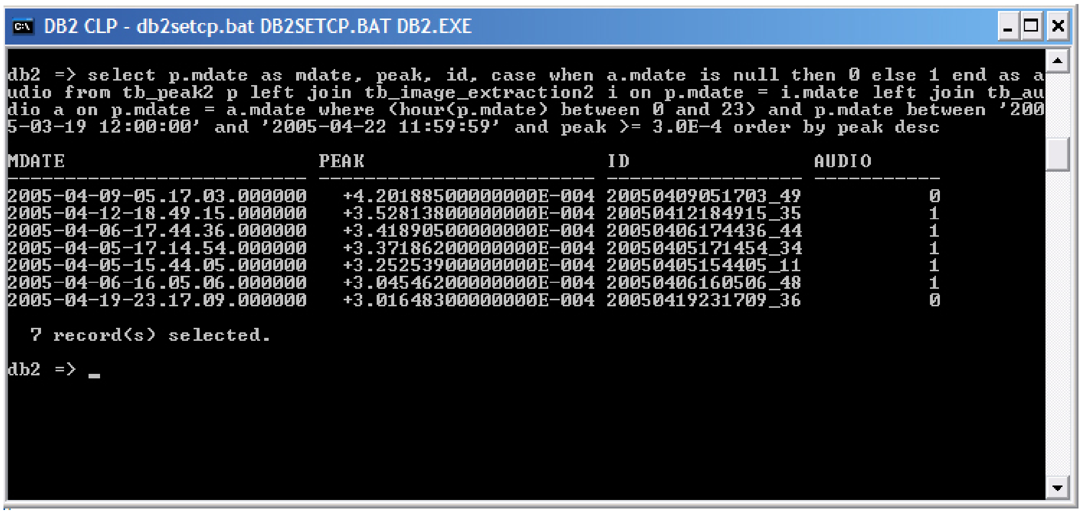
1. Users must be provided access to the databases (usernames, passwords, IP addresses, which are supplied by contacting the ITR project webmaster - mnphan@ucsd.edu - and requesting support).
2. Users must have a thorough understanding of the database architecture and be able to locate the data and metadata in the appropriate tables.
3. To query the data, users must be fluent in the database syntax for writing the queries necessary to retrieve the data.

A series of queries (Table 3.1) illustrating these difficulties is now presented.

Table 3.1: Sample DB2 Queries

Query #	**DB2 Query**
1	select p.mdate as mdate, peak, id, case when a.mdate is null then 0 else 1 end as audio from tb_peak2 p left join tb_image_extraction2 i on p.mdate = i.mdate left join tb_audio a on p.mdate = a.mdate where (hour(p.mdate) between 0 and 23) and p.mdate between '2005-03-19 12:00:00' and '2005-04-22 11:59:59' and peak >= 3.0E-4 order by peak desc
2	select i.mdate, p.peak, i.area from tb_image_extraction2 i, tb_peak2 p where p.mdate = i.mdate and (area between 5000 and 25000) and (i.mdate > '2005-04-21 11:00:00') order by mdate
3	with tb_peak9_temp as (select * from tb_peak9 where date(mdate) = '2005-02-01' and hour(mdate) = 8), tb_image_extraction16_temp as (select * from tb_image_extraction16 where date(mdate) = '2005-02-01' and hour(mdate) = 8) select p.mdate, mvalue, id from tb_peak9_temp p left join tb_image_extraction16_temp i on (p.mdate = i.mdate) order by mdate
4	select mdate, c8, c11 from tb_2005_02_01 where mdate between timestamp('2005-02-01-08.38.01.872900') - 5 second and timestamp('2005-02-01-08.38.01.872900') + 5 second order by mdate fetch first 100 row only
5	with tb_peak9_temp as (select * from tb_peak9 where date(mdate) = '2005-02-01' and hour(mdate) = 8), tb_image_extraction16_temp as (select * from tb_image_extraction16 where date(mdate) = '2005-02-01' and hour(mdate) = 8 and area between 5000 and 25000) select i.mdate, p.mvalue, i.area from tb_image_extraction16_temp i, tb_peak9_temp p where p.mdate = i.mdate order by mdate

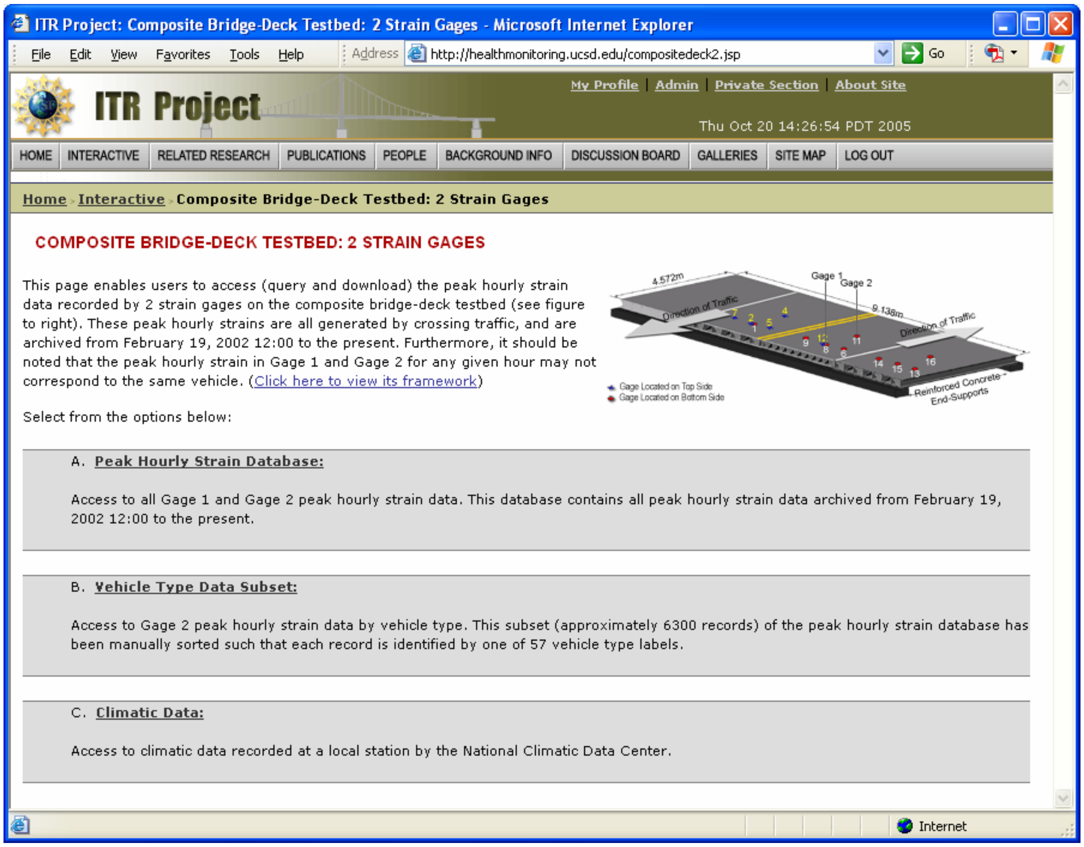
The first query connects to the two-channel peak hourly strain database and returns from the database the time stamp (in date/time format), gage 2 peak strain, image name (corresponding to the image with the largest area determined from image processing), and availability of audio for all of the peak hourly data recorded between noon of March 19, 2005 and 11:59:59 AM on April 22, 2005, where the peak strain was greater than 3 x10^{-4}. The data is sorted in descending order based on the time recorded. A screen capture (taken from DB2 Command Line Processor) of this query and the data returned is provided in Figure 3.32.



```
db2 => select p.mdate as mdate, peak, id, case when a.mdate is null then 0 else 1 end as a
udio from tb_peak2 p left join tb_image_extraction2 i on p.mdate = i.mdate left join tb_au
dio a on p.mdate = a.mdate where (hour(p.mdate) between 0 and 23) and p.mdate between '200
5-03-19 12:00:00' and '2005-04-22 11:59:59' and peak >= 3.0E-4 order by peak desc

MDATE                      PEAK                     ID                 AUDIO
-------------------------- ------------------------ ------------------ -----------
2005-04-09-05.17.03.000000   +4.20188500000000E-004 20050409051703_49            0
2005-04-12-18.49.15.000000   +3.52813800000000E-004 20050412184915_35            1
2005-04-06-17.44.36.000000   +3.41890500000000E-004 20050406174436_44            1
2005-04-05-17.14.54.000000   +3.37186200000000E-004 20050405171454_34            1
2005-04-05-15.44.05.000000   +3.25253900000000E-004 20050405154405_11            1
2005-04-06-16.05.06.000000   +3.04546200000000E-004 20050406160506_48            1
2005-04-19-23.17.09.000000   +3.01648300000000E-004 20050419231709_36            0

  7 record(s) selected.

db2 => _
```

Figure 3.32: Screen Capture Taken from DB2 Command Line Program for the Sample Query.

The second means of accessing the data from the composite bridge-decks databases is through a web-based interface. This user interface was established for convenient online querying and downloading of the recorded data. All data archived from noon on 12/17/2002 through the present is available. For the two-channel peak hourly strain monitoring system, the webpage http://healthmonitoring.ucsd.edu/compositedeck2.jsp (Fig. 3.33) allows guests to access the (A) “Peak Hourly Strain Database,” (B) “Vehicle Type Data Subset,” and (C) “Climatic Data.” Brief descriptions of each link are provided for ease of use.


ITR Project: Composite Bridge-Deck Testbed: 2 Strain Gages - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://healthmonitoring.ucsd.edu/compositedeck2.jsp
Go
ITR Project
My Profile | Admin | Private Section | About Site
Thu Oct 20 14:26:54 PDT 2005
HOME
INTERACTIVE
RELATED RESEARCH
PUBLICATIONS
PEOPLE
BACKGROUND INFO
DISCUSSION BOARD
GALLERIES
SITE MAP
LOG OUT
Home » Interactive » Composite Bridge-Deck Testbed: 2 Strain Gages
4.572m
Gage 1
Gage 2
9.136m
Direction of Traffic
Direction of Traffic
7
2
1
5
4
9
12
8
6
11
14
15
13
16
Reinforced Concrete End-Supports
Gage Located on Top Side
Gage Located on Bottom Side
Internet

COMPOSITE BRIDGE-DECK TESTBED: 2 STRAIN GAGES

This page enables users to access (query and download) the peak hourly strain data recorded by 2 strain gages on the composite bridge-deck testbed (see figure to right). These peak hourly strains are all generated by crossing traffic, and are archived from February 19, 2002 12:00 to the present. Furthermore, it should be noted that the peak hourly strain in Gage 1 and Gage 2 for any given hour may not correspond to the same vehicle. (Click here to view its framework)

Select from the options below:

A. Peak Hourly Strain Database:

Access to all Gage 1 and Gage 2 peak hourly strain data. This database contains all peak hourly strain data archived from February 19, 2002 12:00 to the present.

B. Vehicle Type Data Subset:

Access to Gage 2 peak hourly strain data by vehicle type. This subset (approximately 6300 records) of the peak hourly strain database has been manually sorted such that each record is identified by one of 57 vehicle type labels.

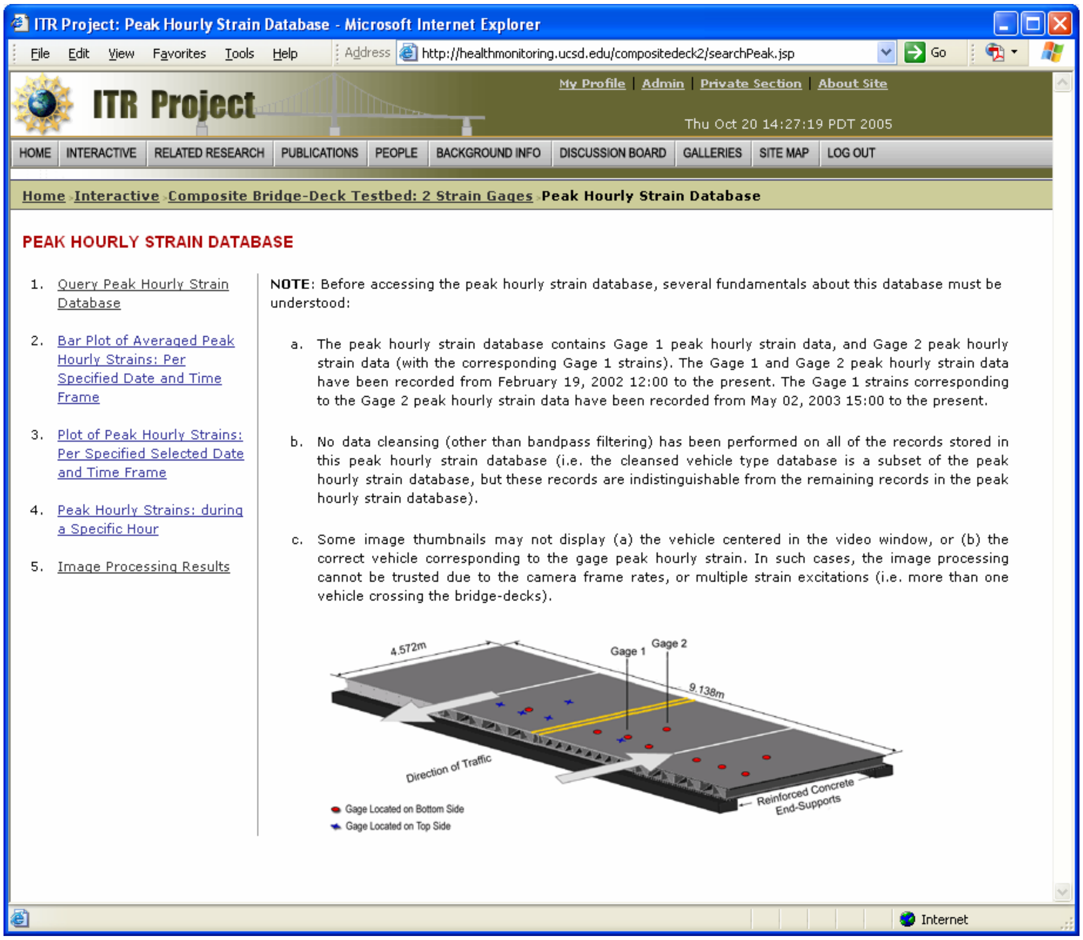
C. Climatic Data:

Access to climatic data recorded at a local station by the National Climatic Data Center.

Figure 3.33: Homepage of the 2-Channel Peak Hourly Strain Database Interface

Selecting the first link takes users to the Peak Hourly Strain Database (Fig. 3.34). This database contains all Gage 1 PHS records and Gage 2 PHS records (with the corresponding Gage 1 strain data). This is the most generic way of accessing the complete set of peak hourly strains and images recorded. By selecting “Query Peak Hourly Strain Database,” users are provided a pre-written query for connecting to the database and retrieving the recorded data. For this query (Fig. 3.35), the data can be sorted based on gage number, start and end dates, hour interval (for example, this option allows users to focus on daytime traffic where the image quality best), and the minimum and maximum peak hourly strains. The same sample query presented above (except sorted in descending order based on peak strain) with samples of the returned data are shown in Figures 3.36 and 3.37. The most obvious advantages to this database interface are the structure of the database is not important and users no longer have to be fluent in database syntax. As with the first method, this interface has restricted access requiring a Guest level account on the web portal; however, anyone may complete the on-line registration (http://healthmonitoring.ucsd.edu/register.jsp) to receive an account.

In the data returned in Figure 3.36, there are a couple of features worth noting. First, the peak strain, along with the entire strain time history, is included on the right hand side. By clicking on the image of the time history, users are taken to an interactive java applet allowing them to zoom in on the data. The picture on the left-hand-side (in this example of the campus shuttle buses) corresponds to the image with the largest area as determined from the image processing / feature extraction algorithm. Clicking on the “View recorded video” button plays the 80-frame traffic video. If audio data is available, it will be indicated by the microphone present on the first record returned (but not on the second one).


ITR Project: Peak Hourly Strain Database - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://healthmonitoring.ucsd.edu/compositedeck2/searchPeak.jsp
Go
ITR Project
My Profile | Admin | Private Section | About Site
Thu Oct 20 14:27:19 PDT 2005
HOME | INTERACTIVE | RELATED RESEARCH | PUBLICATIONS | PEOPLE | BACKGROUND INFO | DISCUSSION BOARD | GALLERIES | SITE MAP | LOG OUT
Home › Interactive › Composite Bridge-Deck Testbed: 2 Strain Gages › Peak Hourly Strain Database
4.572m
Gage 1
Gage 2
9.138m
Direction of Traffic
Reinforced Concrete End-Supports
Gage Located on Bottom Side
Gage Located on Top Side
Internet

PEAK HOURLY STRAIN DATABASE

1. Query Peak Hourly Strain Database
2. Bar Plot of Averaged Peak Hourly Strains: Per Specified Date and Time Frame
3. Plot of Peak Hourly Strains: Per Specified Selected Date and Time Frame
4. Peak Hourly Strains: during a Specific Hour
5. Image Processing Results

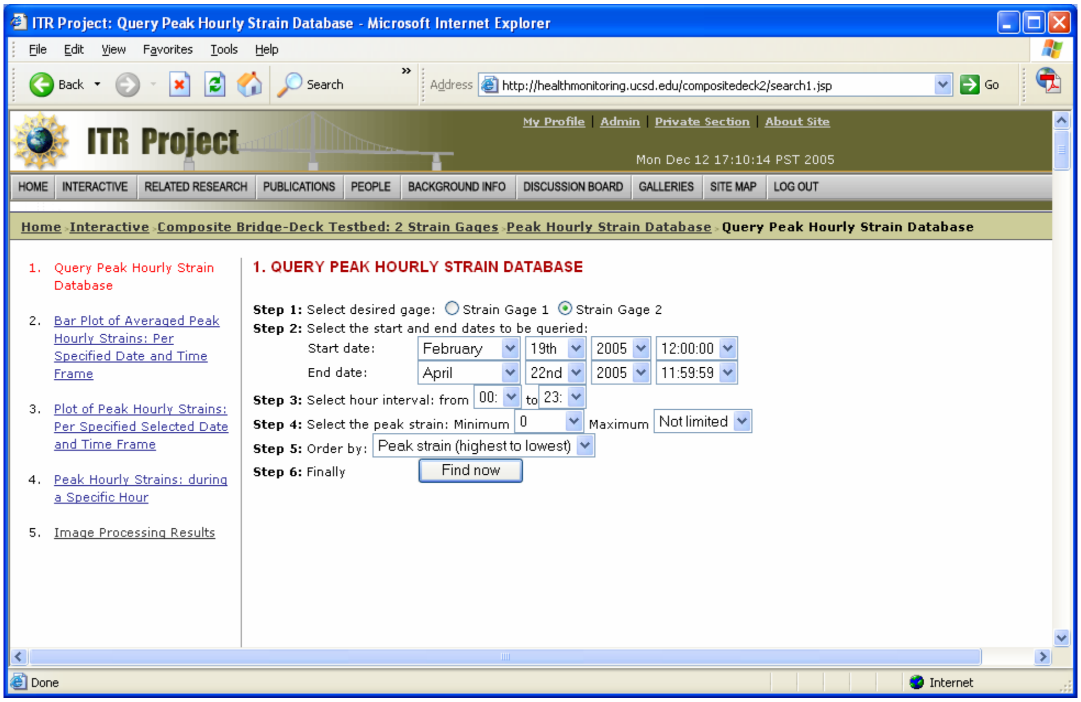
NOTE: Before accessing the peak hourly strain database, several fundamentals about this database must be understood:

a. The peak hourly strain database contains Gage 1 peak hourly strain data, and Gage 2 peak hourly strain data (with the corresponding Gage 1 strains). The Gage 1 and Gage 2 peak hourly strain data have been recorded from February 19, 2002 12:00 to the present. The Gage 1 strains corresponding to the Gage 2 peak hourly strain data have been recorded from May 02, 2003 15:00 to the present.

b. No data cleansing (other than bandpass filtering) has been performed on all of the records stored in this peak hourly strain database (i.e. the cleansed vehicle type database is a subset of the peak hourly strain database, but these records are indistinguishable from the remaining records in the peak hourly strain database).

c. Some image thumbnails may not display (a) the vehicle centered in the video window, or (b) the correct vehicle corresponding to the gage peak hourly strain. In such cases, the image processing cannot be trusted due to the camera frame rates, or multiple strain excitations (i.e. more than one vehicle crossing the bridge-decks).

Figure 3.34: Webpage for the Peak Hourly Strain Database

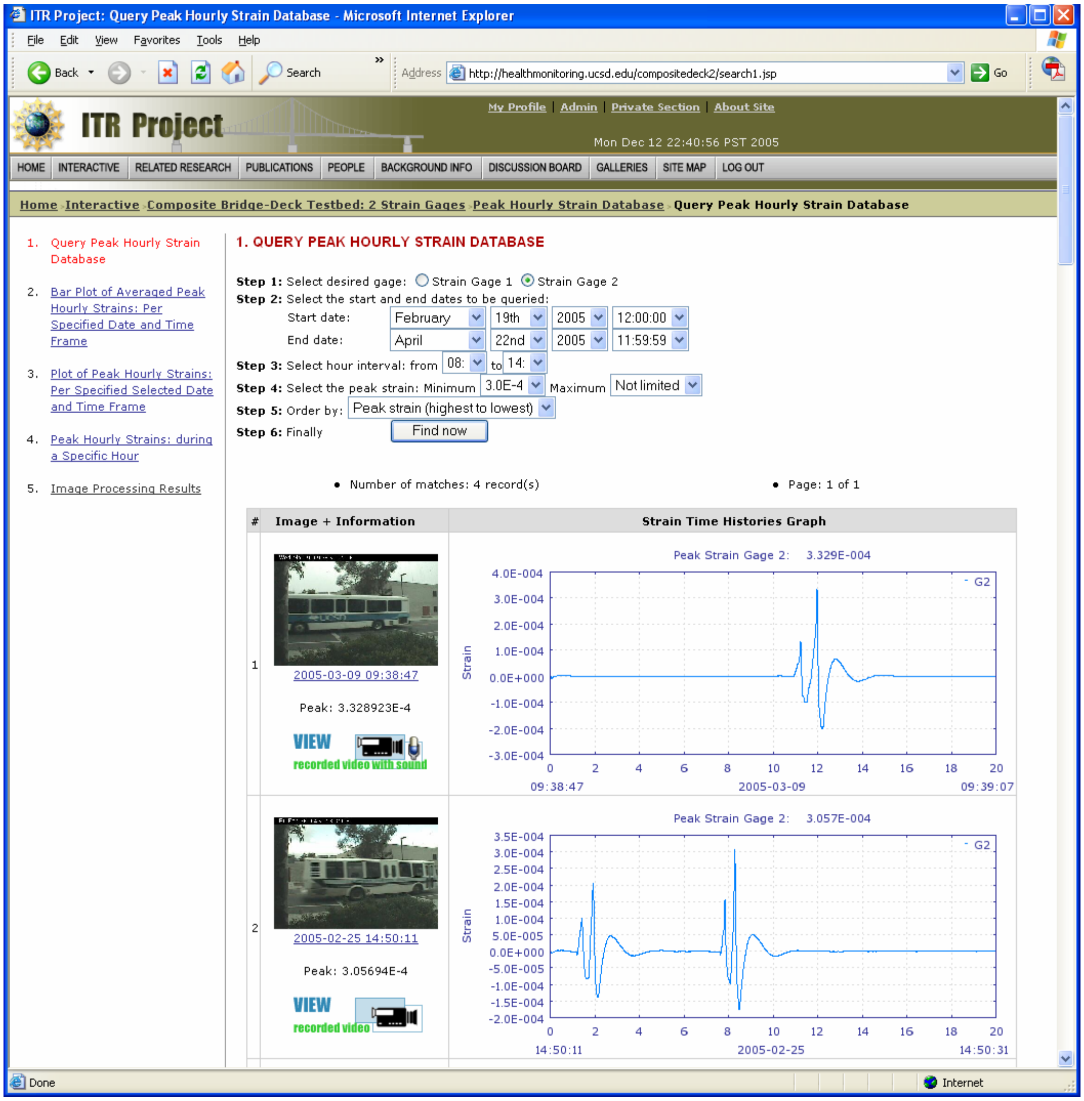

ITR Project: Query Peak Hourly Strain Database - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Address http://healthmonitoring.ucsd.edu/compositedeck2/search1.jsp
Go
ITR Project
My Profile | Admin | Private Section | About Site
Mon Dec 12 17:10:14 PST 2005
HOME
INTERACTIVE
RELATED RESEARCH
PUBLICATIONS
PEOPLE
BACKGROUND INFO
DISCUSSION BOARD
GALLERIES
SITE MAP
LOG OUT
Home › Interactive › Composite Bridge-Deck Testbed: 2 Strain Gages › Peak Hourly Strain Database › Query Peak Hourly Strain Database
Done
Internet

1. Query Peak Hourly Strain Database
2. Bar Plot of Averaged Peak Hourly Strains: Per Specified Date and Time Frame
3. Plot of Peak Hourly Strains: Per Specified Selected Date and Time Frame
4. Peak Hourly Strains: during a Specific Hour
5. Image Processing Results

1. QUERY PEAK HOURLY STRAIN DATABASE

Step 1: Select desired gage: ○ Strain Gage 1 ● Strain Gage 2
Step 2: Select the start and end dates to be queried:
Start date: February 19th 2005 12:00:00
End date: April 22nd 2005 11:59:59
Step 3: Select hour interval: from 00: to 23:
Step 4: Select the peak strain: Minimum 0 Maximum Not limited
Step 5: Order by: Peak strain (highest to lowest)
Step 6: Finally Find now

Figure 3.35: Web for Querying the 2-Channel Strain Data



ITR Project: Query Peak Hourly Strain Database - Microsoft Internet Explorer

Address http://healthmonitoring.ucsd.edu/compositedeck2/search1.jsp

ITR Project

My Profile | Admin | Private Section | About Site

Mon Dec 12 22:40:56 PST 2005

Home › Interactive › Composite Bridge-Deck Testbed: 2 Strain Gages › Peak Hourly Strain Database › **Query Peak Hourly Strain Database**

1. Query Peak Hourly Strain Database
2. Bar Plot of Averaged Peak Hourly Strains: Per Specified Date and Time Frame
3. Plot of Peak Hourly Strains: Per Specified Selected Date and Time Frame
4. Peak Hourly Strains: during a Specific Hour
5. Image Processing Results

1. QUERY PEAK HOURLY STRAIN DATABASE

Step 1: Select desired gage: ○ Strain Gage 1 ● Strain Gage 2

Step 2: Select the start and end dates to be queried:

Start date: February 19th 2005 12:00:00

End date: April 22nd 2005 11:59:59

Step 3: Select hour interval: from 08: to 14:

Step 4: Select the peak strain: Minimum 3.0E-4 Maximum Not limited

Step 5: Order by: Peak strain (highest to lowest)

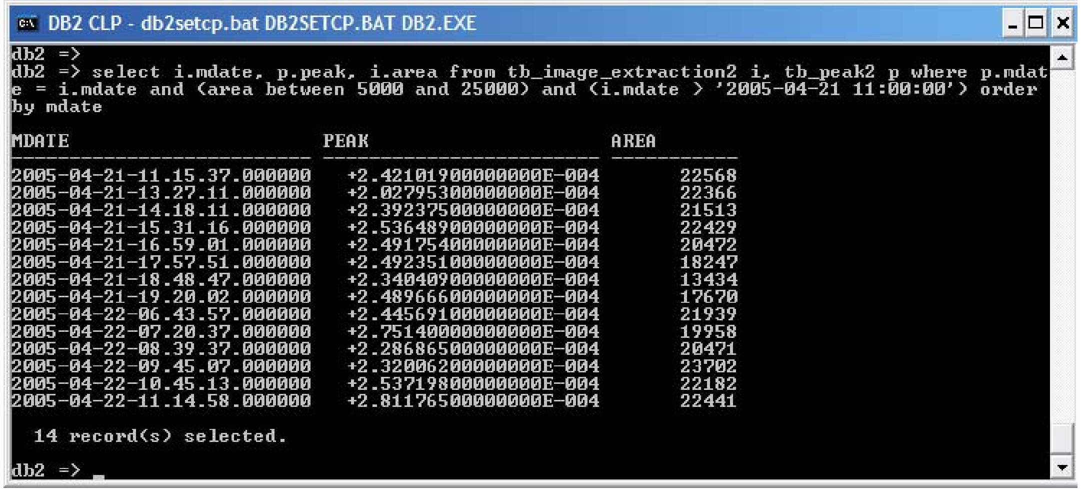
Step 6: Finally

- Number of matches: 4 record(s)
- Page: 1 of 1

#	Image + Information	Strain Time Histories Graph
1	2005-03-09 09:38:47 Peak: 3.328923E-4 VIEW recorded video with sound	Peak Strain Gage 2: 3.329E-004
2	2005-02-25 14:50:11 Peak: 3.05694E-4 VIEW recorded video	Peak Strain Gage 2: 3.057E-004

Figure 3.36: Results for the Sample Query

A disadvantage of the on-line database interface is the number of available queries is limited. Should users wish to add additional features to their queries, they will need to either contact the webmaster for assistance or connect directly to the database. An example (Query 2, Table 3.1) is presented in which the peak area (from feature extraction) is returned with the same data queried in the previous query. In this query, an additional constraint that the peak area must be between 5,000 and 25,000 pixels^2 was also imposed. The returned data is displayed in the screen capture below (Fig. 3.37).



```
db2 =>
db2 => select i.mdate, p.peak, i.area from tb_image_extraction2 i, tb_peak2 p where p.mdat
e = i.mdate and (area between 5000 and 25000) and (i.mdate > '2005-04-21 11:00:00') order
by mdate

MDATE                      PEAK                     AREA
-------------------------- ------------------------ -----------
2005-04-21-11.15.37.000000   +2.42101900000000E-004       22568
2005-04-21-13.27.11.000000   +2.02795300000000E-004       22366
2005-04-21-14.18.11.000000   +2.39237500000000E-004       21513
2005-04-21-15.31.16.000000   +2.53648900000000E-004       22429
2005-04-21-16.59.01.000000   +2.49175400000000E-004       20472
2005-04-21-17.57.51.000000   +2.49235100000000E-004       18247
2005-04-21-18.48.47.000000   +2.34040900000000E-004       13434
2005-04-21-19.20.02.000000   +2.48966600000000E-004       17670
2005-04-22-06.43.57.000000   +2.44569100000000E-004       21939
2005-04-22-07.20.37.000000   +2.75140000000000E-004       19958
2005-04-22-08.39.37.000000   +2.28686500000000E-004       20471
2005-04-22-09.45.07.000000   +2.32006200000000E-004       23702
2005-04-22-10.45.13.000000   +2.53719800000000E-004       22182
2005-04-22-11.14.58.000000   +2.81176500000000E-004       22441

  14 record(s) selected.

db2 => _
```

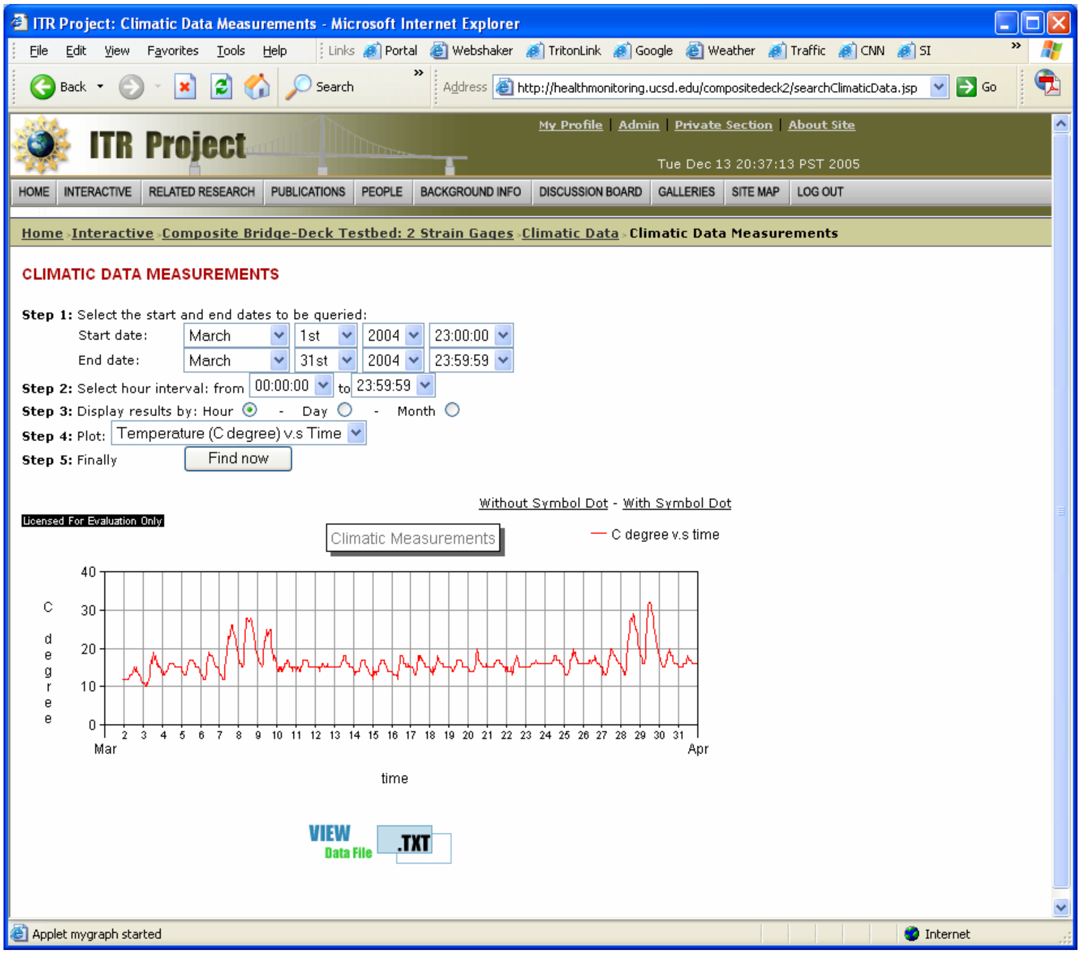
Figure 3.37: Screen Capture Taken of the Second Sample Query.

The second link on the main 2-channel database web page, Vehicle Type Data Subset, allows users sole access to the 6265 records with vehicle type labels (a subset of the Peak Hourly Strain Database). Within this section, users are permitted to select one of the following options: (1) “Query Vehicle Type Data Subset,” (2) “Scatter Plot of Gage 2 Peak Hourly Strain(s) vs. the Corresponding Gage 1 Strain(s),” (3) “Gaussian Fit of the Peak Strains Histogram,” and (4) “Download

Data from the Vehicle Type Data Subset." This data set is discussed in detail in Section 4.6.

The Climatic Data (http://healthmonitoring.ucsd.edu/compositedeck2/searchClimaticData.jsp), the third link on the main page, provides users with helpful tips for obtaining temperature and relative humidity data for the local area that may be desired in a through analysis of strain variations. A plot of the hourly temperatures recorded at a nearby weather station along with it's associated query are shown in Fig. 3.38.

Three additional web pages taken from the 2-channel peak hourly strain database are included in Figures 3.39 – 3.41. The first of these figures shows a bar plot of the average of the hourly peak strains recorded between February 19, 2002 and April 22, 2005. In Figure 3.40, the data is instead averaged by day. It is worth noting the strains in Fig. 3.43 are less in the early morning and in Fig. 3.44 on the weekends when the campus shuttle buses are not running and the peak hourly strains typically are caused by lighter passenger vehicles. Finally, in Fig. 3.45, the peak hourly strains are plotted versus time for the week of April 15-22, 2005. As with the other two figures, the strains are lowest in the late night / early morning and on the weekends.


ITR Project: Climatic Data Measurements - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Links Portal Webshaker TritonLink Google Weather Traffic CNN SI
Back Search
Address http://healthmonitoring.ucsd.edu/compositedeck2/searchClimaticData.jsp Go
ITR Project
My Profile | Admin | Private Section | About Site
Tue Dec 13 20:37:13 PST 2005
HOME INTERACTIVE RELATED RESEARCH PUBLICATIONS PEOPLE BACKGROUND INFO DISCUSSION BOARD GALLERIES SITE MAP LOG OUT
Home Interactive Composite Bridge-Deck Testbed: 2 Strain Gages Climatic Data Climatic Data Measurements

CLIMATIC DATA MEASUREMENTS

Step 1: Select the start and end dates to be queried:
Start date: March 1st 2004 23:00:00
End date: March 31st 2004 23:59:59

Step 2: Select hour interval: from 00:00:00 to 23:59:59

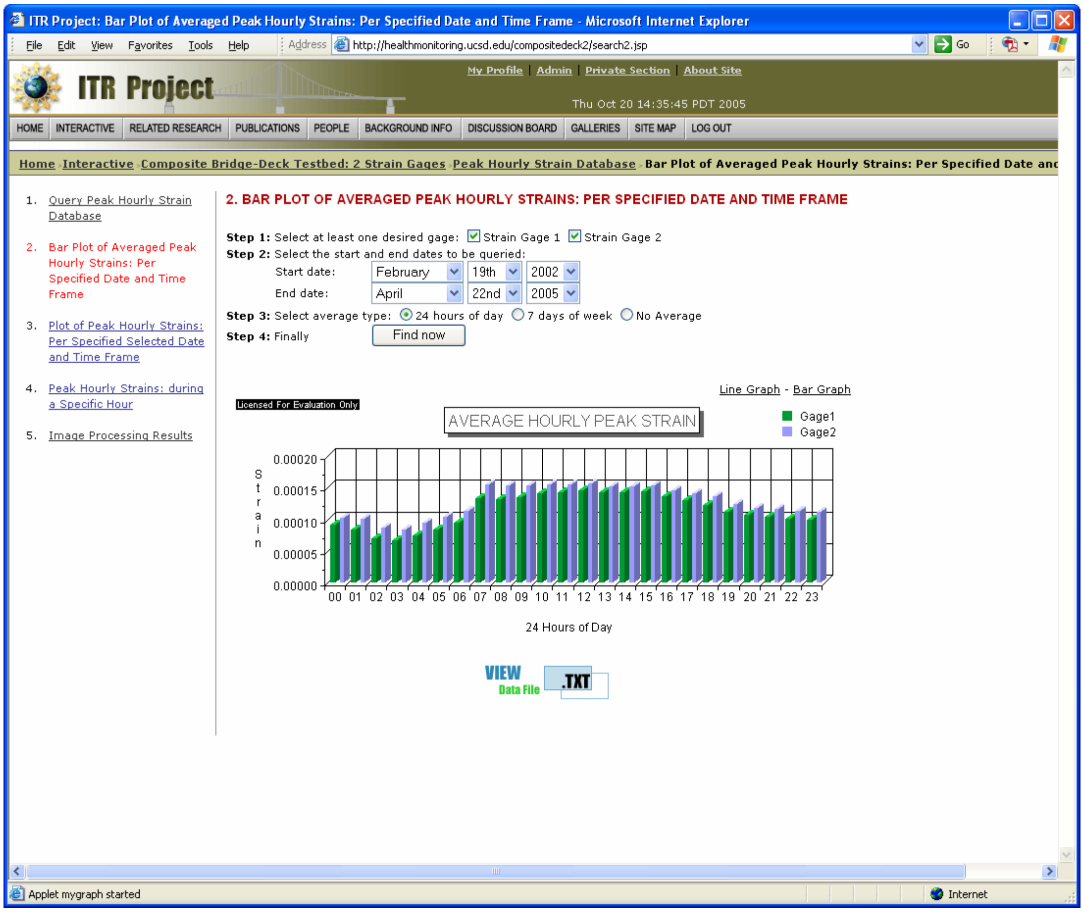
Step 3: Display results by: Hour ● - Day ○ - Month ○

Step 4: Plot: Temperature (C degree) v.s Time

Step 5: Finally Find now

Without Symbol Dot - With Symbol Dot
Licensed For Evaluation Only
Climatic Measurements
C degree v.s time
C degree
40
30
20
10
0
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Mar
Apr
time
VIEW Data File .TXT
Applet mygraph started
Internet

Figure 3.38: Sample of Climatic Data and Associated Query.


ITR Project: Bar Plot of Averaged Peak Hourly Strains: Per Specified Date and Time Frame - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://healthmonitoring.ucsd.edu/compositedeck2/search2.jsp
Go
ITR Project
My Profile | Admin | Private Section | About Site
Thu Oct 20 14:35:45 PDT 2005
HOME
INTERACTIVE
RELATED RESEARCH
PUBLICATIONS
PEOPLE
BACKGROUND INFO
DISCUSSION BOARD
GALLERIES
SITE MAP
LOG OUT
Home › Interactive › Composite Bridge-Deck Testbed: 2 Strain Gages › Peak Hourly Strain Database › Bar Plot of Averaged Peak Hourly Strains: Per Specified Date and
1. Query Peak Hourly Strain Database
2. Bar Plot of Averaged Peak Hourly Strains: Per Specified Date and Time Frame
3. Plot of Peak Hourly Strains: Per Specified Selected Date and Time Frame
4. Peak Hourly Strains: during a Specific Hour
5. Image Processing Results

2. BAR PLOT OF AVERAGED PEAK HOURLY STRAINS: PER SPECIFIED DATE AND TIME FRAME

Step 1: Select at least one desired gage: ☑ Strain Gage 1 ☑ Strain Gage 2

Step 2: Select the start and end dates to be queried:

Start date: February 19th 2002

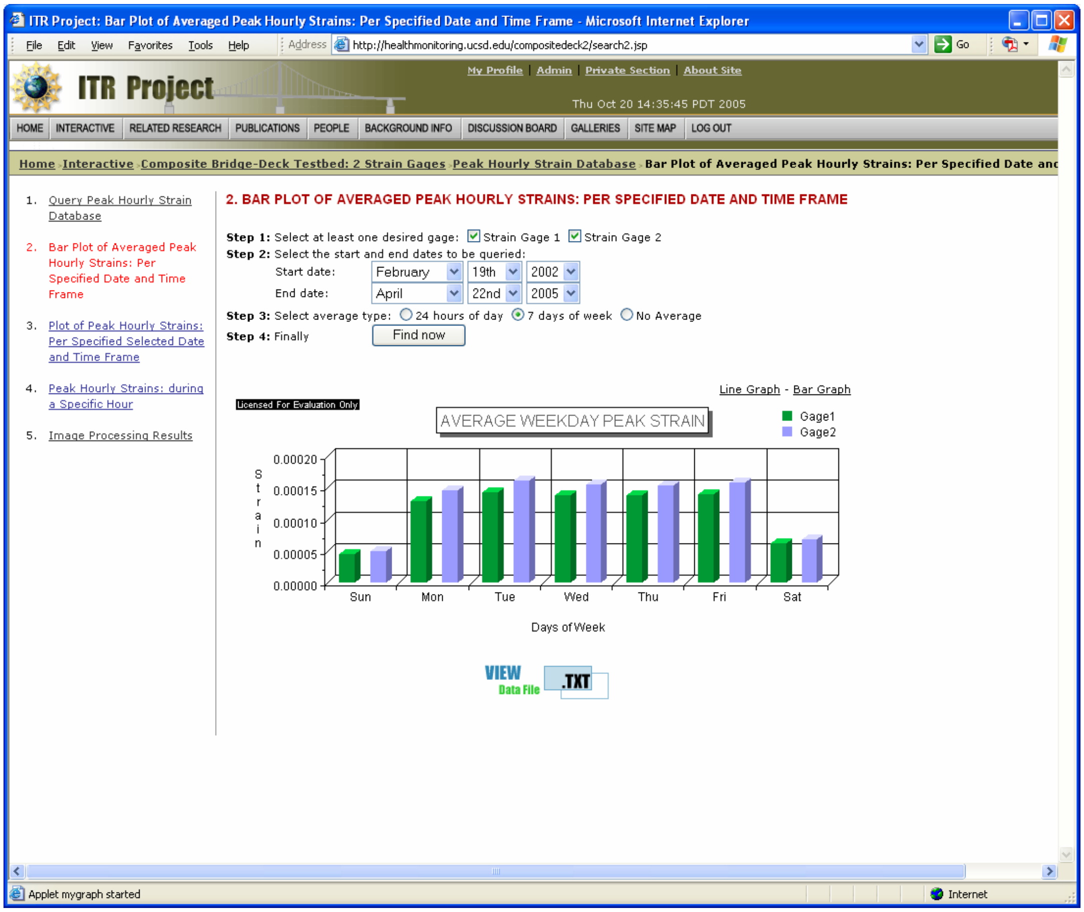
End date: April 22nd 2005

Step 3: Select average type: ● 24 hours of day ○ 7 days of week ○ No Average

Step 4: Finally

Find now
Line Graph - Bar Graph
Licensed For Evaluation Only
AVERAGE HOURLY PEAK STRAIN
Gage1
Gage2
Strain
0.00020
0.00015
0.00010
0.00005
0.00000
00
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24 Hours of Day
VIEW Data File
.TXT
Applet mygraph started
Internet

Figure 3.39: Averaged Hourly Peak Strains



ITR Project: Bar Plot of Averaged Peak Hourly Strains: Per Specified Date and Time Frame - Microsoft Internet Explorer

Address http://healthmonitoring.ucsd.edu/compositedeck2/search2.jsp

My Profile | Admin | Private Section | About Site

Thu Oct 20 14:35:45 PDT 2005

Home › Interactive › Composite Bridge-Deck Testbed: 2 Strain Gages › Peak Hourly Strain Database › Bar Plot of Averaged Peak Hourly Strains: Per Specified Date anc

1. Query Peak Hourly Strain Database
2. Bar Plot of Averaged Peak Hourly Strains: Per Specified Date and Time Frame
3. Plot of Peak Hourly Strains: Per Specified Selected Date and Time Frame
4. Peak Hourly Strains: during a Specific Hour
5. Image Processing Results

2. BAR PLOT OF AVERAGED PEAK HOURLY STRAINS: PER SPECIFIED DATE AND TIME FRAME

Step 1: Select at least one desired gage: ☑ Strain Gage 1 ☑ Strain Gage 2

Step 2: Select the start and end dates to be queried:

Start date: February 19th 2002

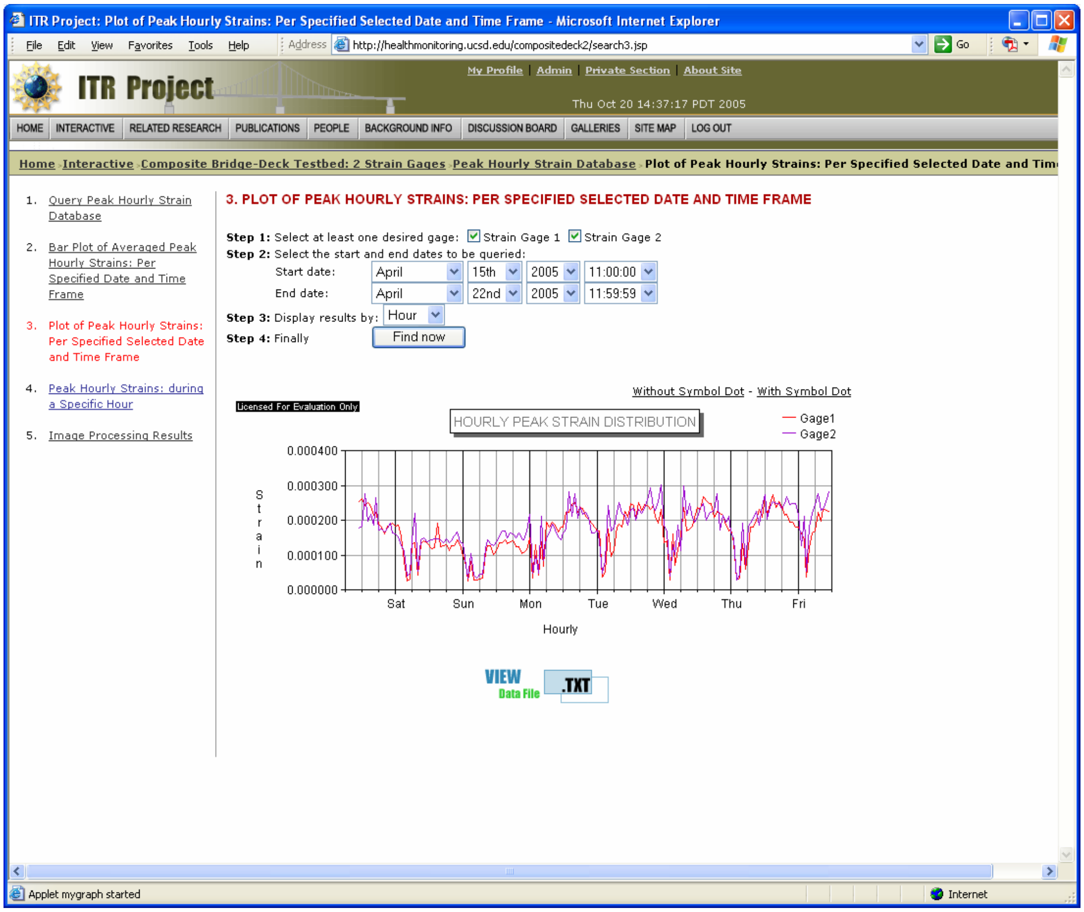
End date: April 22nd 2005

Step 3: Select average type: ○ 24 hours of day ● 7 days of week ○ No Average

Step 4: Finally [Find now]

Applet mygraph started

Figure 3.40: Averaged Daily Peak Strains



ITR Project: Plot of Peak Hourly Strains: Per Specified Selected Date and Time Frame - Microsoft Internet Explorer

Address http://healthmonitoring.ucsd.edu/compositedeck2/search3.jsp

Thu Oct 20 14:37:17 PDT 2005

Home › Interactive › Composite Bridge-Deck Testbed: 2 Strain Gages › Peak Hourly Strain Database › Plot of Peak Hourly Strains: Per Specified Selected Date and Tim

1. Query Peak Hourly Strain Database
2. Bar Plot of Averaged Peak Hourly Strains: Per Specified Date and Time Frame
3. Plot of Peak Hourly Strains: Per Specified Selected Date and Time Frame
4. Peak Hourly Strains: during a Specific Hour
5. Image Processing Results

3. PLOT OF PEAK HOURLY STRAINS: PER SPECIFIED SELECTED DATE AND TIME FRAME

Step 1: Select at least one desired gage: ☑ Strain Gage 1 ☑ Strain Gage 2
Step 2: Select the start and end dates to be queried:
Start date: April | 15th | 2005 | 11:00:00
End date: April | 22nd | 2005 | 11:59:59
Step 3: Display results by: Hour
Step 4: Finally [Find now]

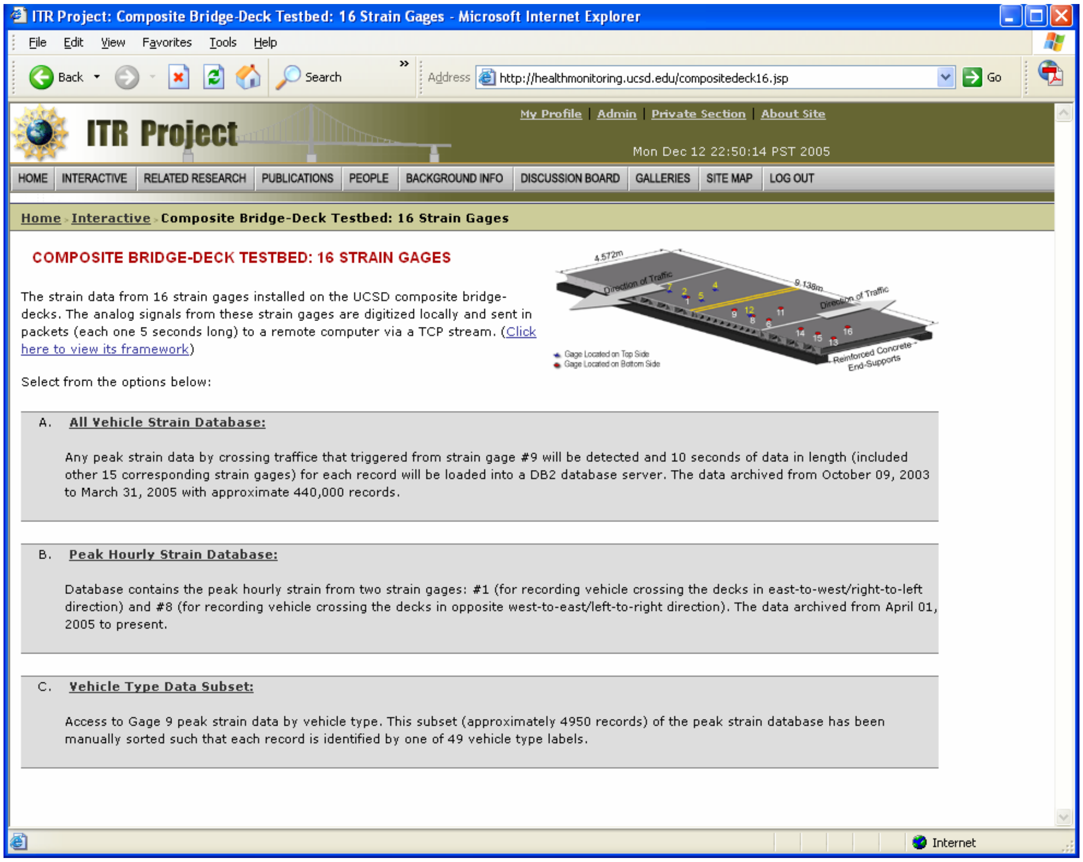
Without Symbol Dot - With Symbol Dot

Applet mygraph started

Internet

Figure 3.41: Plot of Hourly Peak Stains versus Time

A similar web-based interface (Fig. 3.42) was created to provide access to the data recorded by the 16-channel continuous strain monitoring system. The available query for recorded vehicle strains is shown in Figure 3.43. This query allows users to specify the start and end dates, hour interval, and minimum and maximum strains, and the returned data (Fig. 3.44) is sorted by ascending/descending order of peak strain or date recorded.



ITR Project: Composite Bridge-Deck Testbed: 16 Strain Gages - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Address http://healthmonitoring.ucsd.edu/compositedeck16.jsp
Go
ITR Project
My Profile | Admin | Private Section | About Site
HOME
INTERACTIVE
RELATED RESEARCH
PUBLICATIONS
PEOPLE
BACKGROUND INFO
DISCUSSION BOARD
GALLERIES
SITE MAP
LOG OUT
4.572m
9.138m
Direction of Traffic
Direction of Traffic
1
2
4
5
7
9
11
12
13
14
15
16
6
8
Gage Located on Top Side
Gage Located on Bottom Side
Reinforced Concrete End-Supports
Internet

Mon Dec 12 22:50:14 PST 2005

Home » Interactive » Composite Bridge-Deck Testbed: 16 Strain Gages

COMPOSITE BRIDGE-DECK TESTBED: 16 STRAIN GAGES

The strain data from 16 strain gages installed on the UCSD composite bridge-decks. The analog signals from these strain gages are digitized locally and sent in packets (each one 5 seconds long) to a remote computer via a TCP stream. (Click here to view its framework)

Select from the options below:

A. **All Vehicle Strain Database:**

Any peak strain data by crossing traffice that triggered from strain gage #9 will be detected and 10 seconds of data in length (included other 15 corresponding strain gages) for each record will be loaded into a DB2 database server. The data archived from October 09, 2003 to March 31, 2005 with approximate 440,000 records.

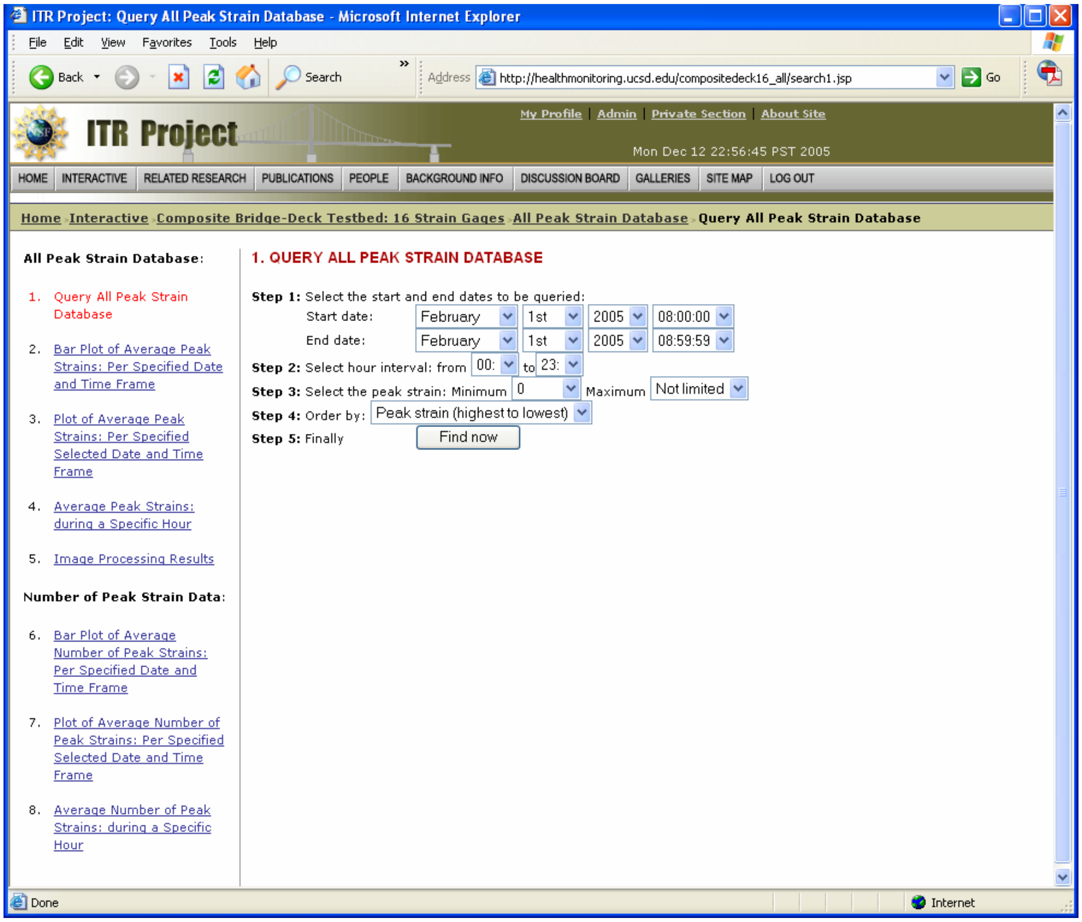
B. **Peak Hourly Strain Database:**

Database contains the peak hourly strain from two strain gages: #1 (for recording vehicle crossing the decks in east-to-west/right-to-left direction) and #8 (for recording vehicle crossing the decks in opposite west-to-east/left-to-right direction). The data archived from April 01, 2005 to present.

C. **Vehicle Type Data Subset:**

Access to Gage 9 peak strain data by vehicle type. This subset (approximately 4950 records) of the peak strain database has been manually sorted such that each record is identified by one of 49 vehicle type labels.

Figure 3.42: Web Page for 16-Channel Database Interface


ITR Project: Query All Peak Strain Database - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Address
Go
ITR Project
My Profile | Admin | Private Section | About Site
HOME INTERACTIVE RELATED RESEARCH PUBLICATIONS PEOPLE BACKGROUND INFO DISCUSSION BOARD GALLERIES SITE MAP LOG OUT
Done
Internet

Address: http://healthmonitoring.ucsd.edu/compositedeck16_all/search1.jsp

Mon Dec 12 22:56:45 PST 2005

Home · Interactive · Composite Bridge-Deck Testbed: 16 Strain Gages · All Peak Strain Database · **Query All Peak Strain Database**

All Peak Strain Database:

1. Query All Peak Strain Database
2. Bar Plot of Average Peak Strains: Per Specified Date and Time Frame
3. Plot of Average Peak Strains: Per Specified Selected Date and Time Frame
4. Average Peak Strains: during a Specific Hour
5. Image Processing Results

Number of Peak Strain Data:

6. Bar Plot of Average Number of Peak Strains: Per Specified Date and Time Frame
7. Plot of Average Number of Peak Strains: Per Specified Selected Date and Time Frame
8. Average Number of Peak Strains: during a Specific Hour

1. QUERY ALL PEAK STRAIN DATABASE

Step 1: Select the start and end dates to be queried:

Start date: February 1st 2005 08:00:00

End date: February 1st 2005 08:59:59

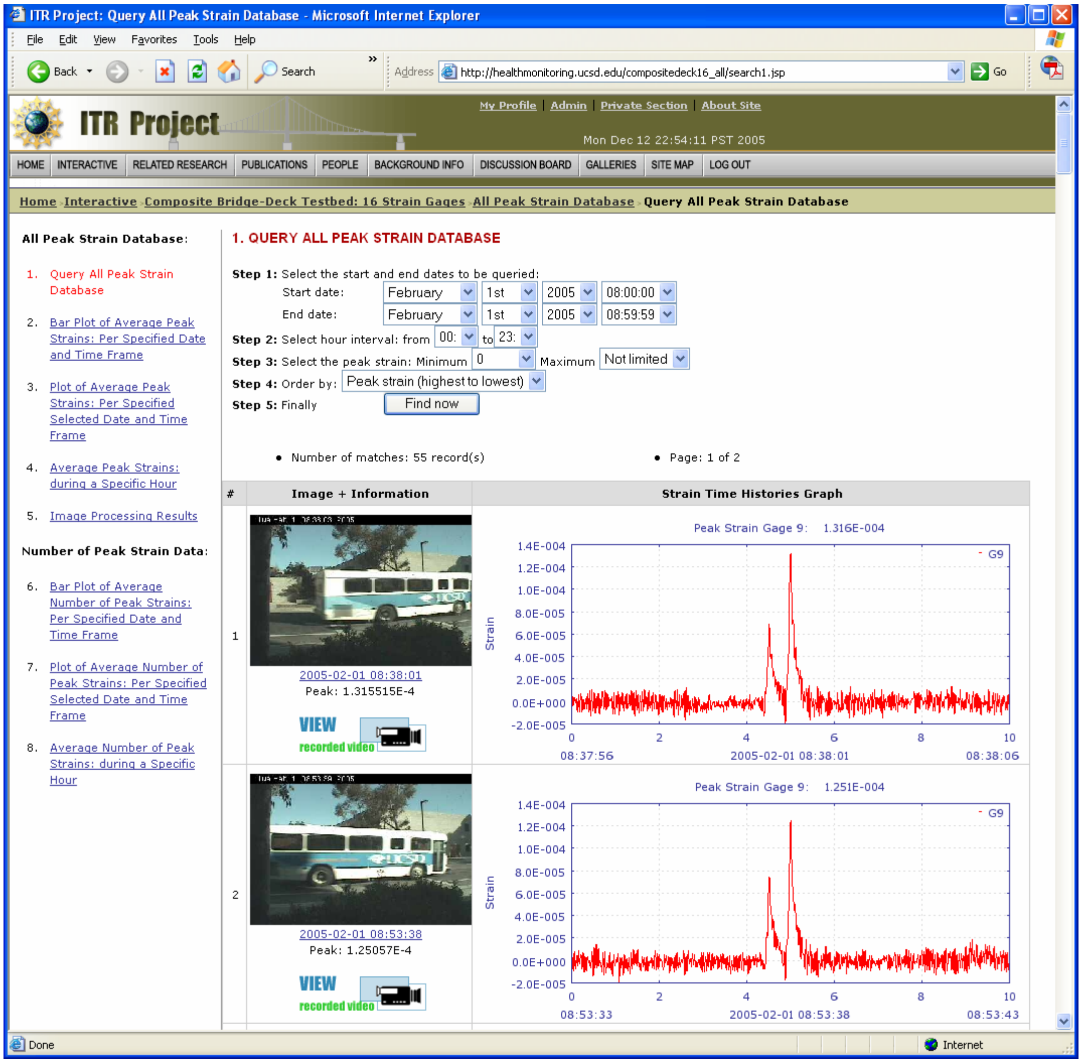
Step 2: Select hour interval: from 00: to 23:

Step 3: Select the peak strain: Minimum 0 Maximum Not limited

Step 4: Order by: Peak strain (highest to lowest)

Step 5: Finally Find now

Figure 3.43: Web-based Query Page for 16-Channel Strain Data



ITR Project: Query All Peak Strain Database - Microsoft Internet Explorer

Address http://healthmonitoring.ucsd.edu/compositedeck16_all/search1.jsp

My Profile | Admin | Private Section | About Site

Mon Dec 12 22:54:11 PST 2005

Home › Interactive › Composite Bridge-Deck Testbed: 16 Strain Gages › All Peak Strain Database › Query All Peak Strain Database

All Peak Strain Database:

1. Query All Peak Strain Database
2. Bar Plot of Average Peak Strains: Per Specified Date and Time Frame
3. Plot of Average Peak Strains: Per Specified Selected Date and Time Frame
4. Average Peak Strains: during a Specific Hour
5. Image Processing Results

Number of Peak Strain Data:

6. Bar Plot of Average Number of Peak Strains: Per Specified Date and Time Frame
7. Plot of Average Number of Peak Strains: Per Specified Selected Date and Time Frame
8. Average Number of Peak Strains: during a Specific Hour

1. QUERY ALL PEAK STRAIN DATABASE

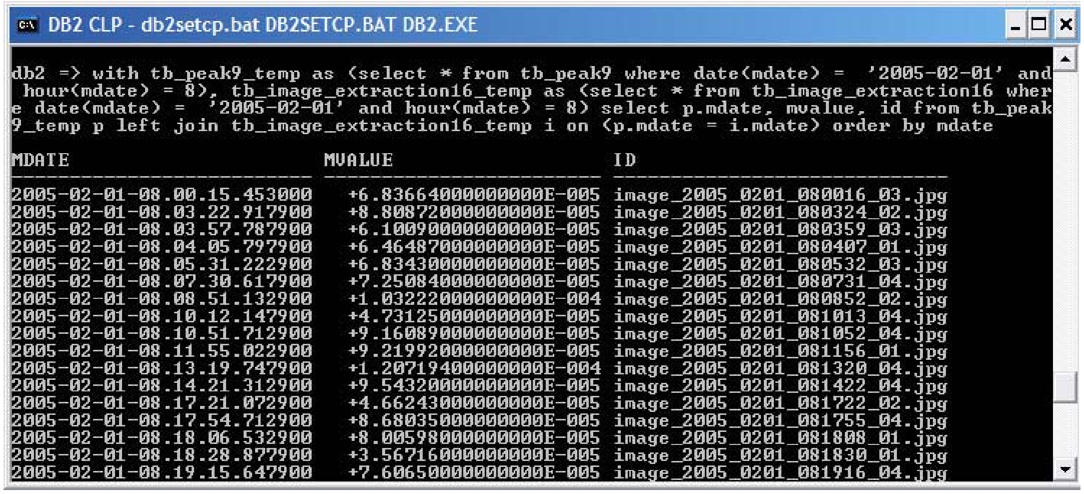
Step 1: Select the start and end dates to be queried:
Start date: February 1st 2005 08:00:00
End date: February 1st 2005 08:59:59
Step 2: Select hour interval: from 00: to 23:
Step 3: Select the peak strain: Minimum 0 Maximum Not limited
Step 4: Order by: Peak strain (highest to lowest)
Step 5: Finally Find now

- Number of matches: 55 record(s)
- Page: 1 of 2

#	Image + Information	Strain Time Histories Graph
1	2005-02-01 08:38:01 Peak: 1.315515E-4	Peak Strain Gage 9: 1.316E-004
2	2005-02-01 08:53:38 Peak: 1.25057E-4	Peak Strain Gage 9: 1.251E-004

Figure 3.44: Results of Query from 16-Channel Strain Database

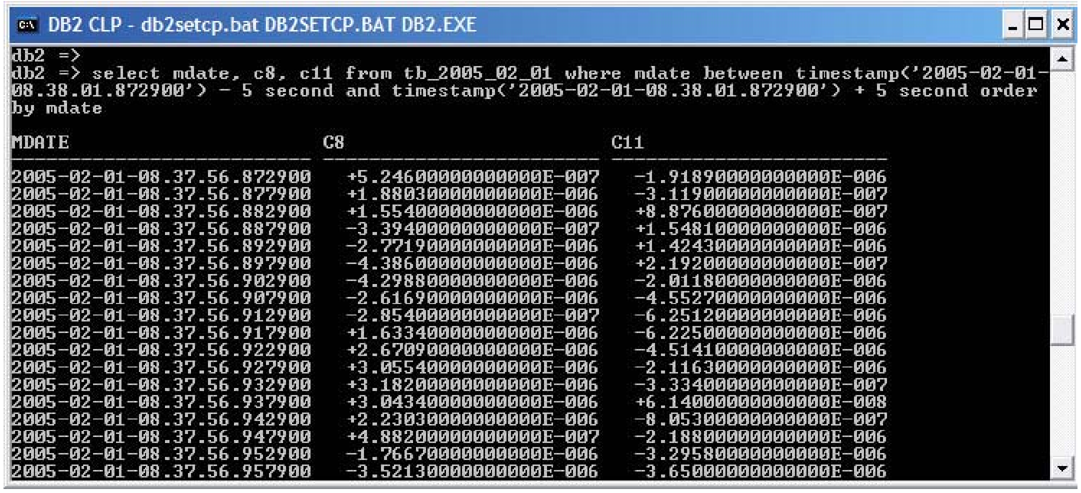
As with the 2-channel database, the 16-channel database can also be queried using a direct connection with DB2. The same query string used in the sample from the web page above is shown in Query 3 (Table 3.1). A screen capture taken from DB2 Command Line Processor of the metadata data returned is shown in Fig. 3.45



```
db2 => with tb_peak9_temp as (select * from tb_peak9 where date(mdate) =  '2005-02-01' and
 hour(mdate) = 8), tb_image_extraction16_temp as (select * from tb_image_extraction16 wher
e date(mdate) =  '2005-02-01' and hour(mdate) = 8) select p.mdate, mvalue, id from tb_peak
9_temp p left join tb_image_extraction16_temp i on (p.mdate = i.mdate) order by mdate

MDATE                      MVALUE                    ID
-------------------------- ------------------------- ------------------------------
2005-02-01-08.00.15.453000   +6.83664000000000E-005 image_2005_0201_080016_03.jpg
2005-02-01-08.03.22.917900   +8.80872000000000E-005 image_2005_0201_080324_02.jpg
2005-02-01-08.03.57.787900   +6.10090000000000E-005 image_2005_0201_080359_03.jpg
2005-02-01-08.04.05.797900   +6.46487000000000E-005 image_2005_0201_080407_01.jpg
2005-02-01-08.05.31.222900   +6.83430000000000E-005 image_2005_0201_080532_03.jpg
2005-02-01-08.07.30.617900   +7.25084000000000E-005 image_2005_0201_080731_04.jpg
2005-02-01-08.08.51.132900   +1.03222000000000E-004 image_2005_0201_080852_02.jpg
2005-02-01-08.10.12.147900   +4.73125000000000E-005 image_2005_0201_081013_04.jpg
2005-02-01-08.10.51.712900   +9.16089000000000E-005 image_2005_0201_081052_04.jpg
2005-02-01-08.11.55.022900   +9.21992000000000E-005 image_2005_0201_081156_01.jpg
2005-02-01-08.13.19.747900   +1.20719400000000E-004 image_2005_0201_081320_04.jpg
2005-02-01-08.14.21.312900   +9.54320000000000E-005 image_2005_0201_081422_04.jpg
2005-02-01-08.17.21.072900   +4.66243000000000E-005 image_2005_0201_081722_02.jpg
2005-02-01-08.17.54.712900   +8.68035000000000E-005 image_2005_0201_081755_04.jpg
2005-02-01-08.18.06.532900   +8.00598000000000E-005 image_2005_0201_081808_01.jpg
2005-02-01-08.18.28.877900   +3.56716000000000E-005 image_2005_0201_081830_01.jpg
2005-02-01-08.19.15.647900   +7.60650000000000E-005 image_2005_0201_081916_04.jpg
```

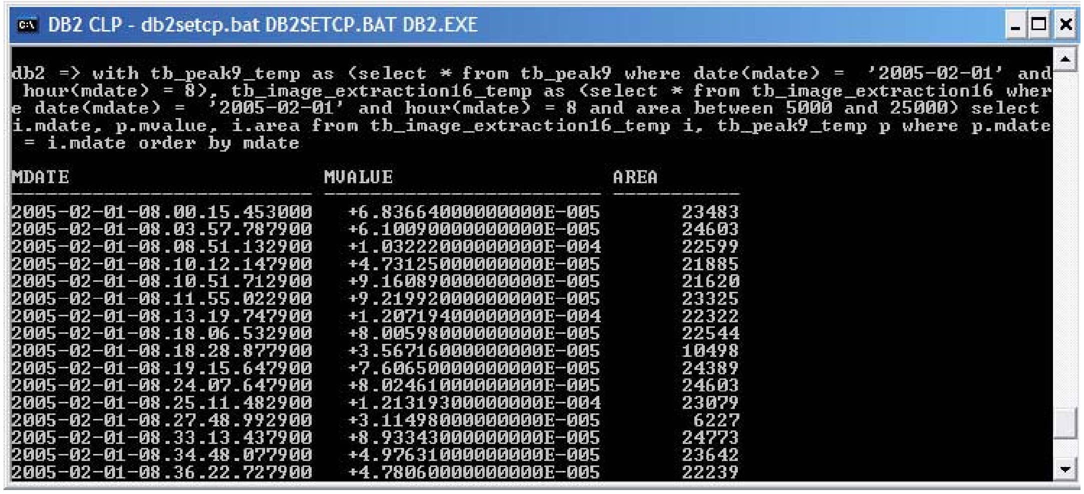
Figure 3.45: Screen Capture of the Returned Metadata from Query 3.

Two additional queries are also presented (Table 3.1). Query 4 returns (Fig. 3.46) the first 100 time steps of the strain data recorded by gages 8 and 11 (along with the corresponding date/time information). For this query, the start and end times correspond to five-seconds before/after a detected peak in the data. This is the type of query used to generate the strain time histories shown in Figure 3.30. The final example (Query 5) returns the date/time, channel 9 peak strain, and corresponding peak area (Fig. 3.47) recorded between 8:00 – 8:59 AM on February 1, 2005 in which the peak area is between 5,000 and 25,000 pixels^2.



```
db2 =>
db2 => select mdate, c8, c11 from tb_2005_02_01 where mdate between timestamp('2005-02-01-
08.38.01.872900') - 5 second and timestamp('2005-02-01-08.38.01.872900') + 5 second order
by mdate

MDATE                      C8                         C11
-------------------------- ------------------------ ------------------------
2005-02-01-08.37.56.872900   +5.24600000000000E-007   -1.91890000000000E-006
2005-02-01-08.37.56.877900   +1.88030000000000E-006   -3.11900000000000E-007
2005-02-01-08.37.56.882900   +1.55400000000000E-006   +8.87600000000000E-007
2005-02-01-08.37.56.887900   -3.39400000000000E-007   +1.54810000000000E-006
2005-02-01-08.37.56.892900   -2.77190000000000E-006   +1.42430000000000E-006
2005-02-01-08.37.56.897900   -4.38600000000000E-006   +2.19200000000000E-007
2005-02-01-08.37.56.902900   -4.29880000000000E-006   -2.01180000000000E-006
2005-02-01-08.37.56.907900   -2.61690000000000E-006   -4.55270000000000E-006
2005-02-01-08.37.56.912900   -2.85400000000000E-007   -6.25120000000000E-006
2005-02-01-08.37.56.917900   +1.63340000000000E-006   -6.22500000000000E-006
2005-02-01-08.37.56.922900   +2.67090000000000E-006   -4.51410000000000E-006
2005-02-01-08.37.56.927900   +3.05540000000000E-006   -2.11630000000000E-006
2005-02-01-08.37.56.932900   +3.18200000000000E-006   -3.33400000000000E-007
2005-02-01-08.37.56.937900   +3.04340000000000E-006   +6.14000000000000E-008
2005-02-01-08.37.56.942900   +2.23030000000000E-006   -8.05300000000000E-007
2005-02-01-08.37.56.947900   +4.88200000000000E-007   -2.18800000000000E-006
2005-02-01-08.37.56.952900   -1.76670000000000E-006   -3.29580000000000E-006
2005-02-01-08.37.56.957900   -3.52130000000000E-006   -3.65000000000000E-006
```

Figure 3.46: Screen Capture of Returned Strains for Query 4.



```
db2 => with tb_peak9_temp as (select * from tb_peak9 where date(mdate) =  '2005-02-01' and
 hour(mdate) = 8), tb_image_extraction16_temp as (select * from tb_image_extraction16 wher
e date(mdate) =  '2005-02-01' and hour(mdate) = 8 and area between 5000 and 25000) select
i.mdate, p.mvalue, i.area from tb_image_extraction16_temp i, tb_peak9_temp p where p.mdate
 = i.mdate order by mdate

MDATE                      MVALUE                     AREA
-------------------------- ------------------------ -----------
2005-02-01-08.00.15.453000   +6.83664000000000E-005       23483
2005-02-01-08.03.57.787900   +6.10090000000000E-005       24603
2005-02-01-08.08.51.132900   +1.03222000000000E-004       22599
2005-02-01-08.10.12.147900   +4.73125000000000E-005       21885
2005-02-01-08.10.51.712900   +9.16089000000000E-005       21620
2005-02-01-08.11.55.022900   +9.21992000000000E-005       23325
2005-02-01-08.13.19.747900   +1.20719400000000E-004       22322
2005-02-01-08.18.06.532900   +8.00598000000000E-005       22544
2005-02-01-08.18.28.877900   +3.56716000000000E-005       10498
2005-02-01-08.19.15.647900   +7.60650000000000E-005       24389
2005-02-01-08.24.07.647900   +8.02461000000000E-005       24603
2005-02-01-08.25.11.482900   +1.21319300000000E-004       23079
2005-02-01-08.27.48.992900   +3.11498000000000E-005        6227
2005-02-01-08.33.13.437900   +8.93343000000000E-005       24773
2005-02-01-08.34.48.077900   +4.97631000000000E-005       23642
2005-02-01-08.36.22.727900   +4.78060000000000E-005       22239
```

Figure 3.47: Screen Capture of Returned Date/Time, Channel 9 Peak Strain, and Peak Area from Query 5.

3.6 Summary

A research testbed for applying the aforementioned structural monitoring framework was established. PC-based data acquisition was employed along with a network camera for continuous monitoring of a series of instrumented composite bridge-decks. Over a three-year period, time synchronized video and strain data (achieved by separately synchronizing the network camera and data acquisition computers with an NTP Internet time server) have been recorded and archived in a series of databases, and made available for on-line querying. The continuously recorded data was post processed and separated into 439,654 discrete events, each composed of a ten-second strain time history and 40 images of a vehicle crossing the bridge decks. Local processing potential for data reduction and event detection was demonstrated with the peak hourly strain database established for the 2-channel monitoring system.

4 Video Analysis

Image processing has been performed on the video recorded from the aforementioned bridge-deck testbed. In this chapter, the image processing / feature extraction operation is analyzed and the findings discussed. From this analysis procedure, a new and unique data set composed of traffic induced strain time histories (with time-synchronized video) sorted by vehicle type has been created. Within this research, recorded video was analyzed for two purposes. The first, and primary, emphasis was to combine video and other sensor data to define loads (and speed of moving traffic), along with response (measured strains / displacements / accelerations…) on the structure. This work opens new and far-reaching avenues, as current research only relies on monitoring and analysis of the structural response..

4.1 Matlab-based Feature Identification

The image sequences captured using video cameras give a large amount of data. However, to extract useful information such as location, velocity, and type of vehicles, the image sequences need to be processed using image analysis and computer vision. In order to determine the properties of the vehicles on the bridge, the vehicles should be separated from the background images. To separate moving objects from stationary background, the "background subtraction method" is typically used [Friedman and Russell, 2001 and Ridder et al., 2000]. This method dynamically forms a model of the scene background, compares each new frame to the model, and separates the pixels that have different properties from the background.

A series of Matlab programs were developed to process the images that have been archived as part of the monitoring activities of the UCSD Powell Laboratory testbed composite deck panels (http://healthmonitoring.ucsd.edu/research.jsp). These programs make use of not only the base Matlab software, but also specific functions found in the Matlab Image Analysis toolbox. As previously discussed, since December 17, 2002, images captured when the hourly peak recorded strains occur have been stored. When an hourly peak strain is detected by one of the two strain gages that have been monitoring traffic crossing over the composite decks, a twenty second long time history, containing the peak signal, is saved to disk and uploaded into a database. When the time history is saved, the corresponding video is also saved. This twenty-second long video is composed of eighty jpg images, each taken approximately 0.25 seconds apart. Each images in the sequence is saved to disk using a naming scheme, which indicates the year, month, day, hour, minute, and second that the sequence started, as well as the individual frame number in the sequence (for example, image_20030510112939_1.jpg).

The basic strategy for the feature extraction algorithm used is to compare changes in the absolute difference of two images. Matab begins with reading the sequence name and the number of frames in the sequence. The first image in the sequence is then read and temporarily stored as the background for the rest of the sequence (Fig. 4.1). To ensure that there are no significant objects in this background image that would cause the lingering appearance of static objects, a comparison is made with the second image in the sequence. A new image is created from the absolute difference between the two color images. For each pixel in the resulting image, any difference between the two images will show up in the value stored at the pixel. For no change, the value will be zero and for pixels where there is a change, the greater the difference the larger the stored value. The resulting image is converted to a binary image (black and white) and the values at each of the pixels are summed and compared to a threshold value. If the summed value exceeds the threshold, it is assumed that there is at least one moving object that is contaminating the feature extraction process. To correct this problem, the program sets the second image as the background and takes the absolute difference with the next picture in the sequence (in this case the third image) and compares it to the threshold value. This process is repeated until a suitable background image is located.

Then, using a “for loop,” each image in the sequence (Fig. 4.2) is compared to this background image (Fig. 4.1). For each iteration in the loop, the area of all differing clusters of pixels and the horizontal and vertical centroids of the largest area of differing pixels are determined. In addition, the dimensions of a “bounding box” which fits around the largest group of pixels is determined. By monitoring the change in location of the horizontal centroid, the direction of the traffic can be determined and from the area, the type of traffic can be classified. To improve on the quality of the resulting image, after the absolute difference is taken between two images (Fig. 4.3) and before it is converted to black and white, the resulting image is brightened (by multiplying every value by a constant) (Fig. 4.5). The image is then converted to black and white (binary) so that the region properties can be determined (area, centroid, bounding box, Fig. 4.6). However, before the region properties are determined the image is cropped to remove sections affected by shadows (Fig. 4.7), the holes in the image are filled (Figure 4.8), and then the whole image is filtered to help remove any scattering of stray pixels (Figure 4.9).

Next, two sets of processed images to be displayed on the website. The first of these is a set of black and white images showing the objects detected. A second set of images is prepared which takes the original color images and superimposes onto them a green dot indicating the location of the centroid of the largest object detected as well as a red bounding box which encompasses this detected object (Figures 4.10 – 4.13). Finally the data from the feature extraction process (Area, Area of largest object detected, horizontal centroid, vertical centroid, length & height of bounding box, direction, and vehicle type) is loaded into the database (Table 4.1 & Figure 4.13).

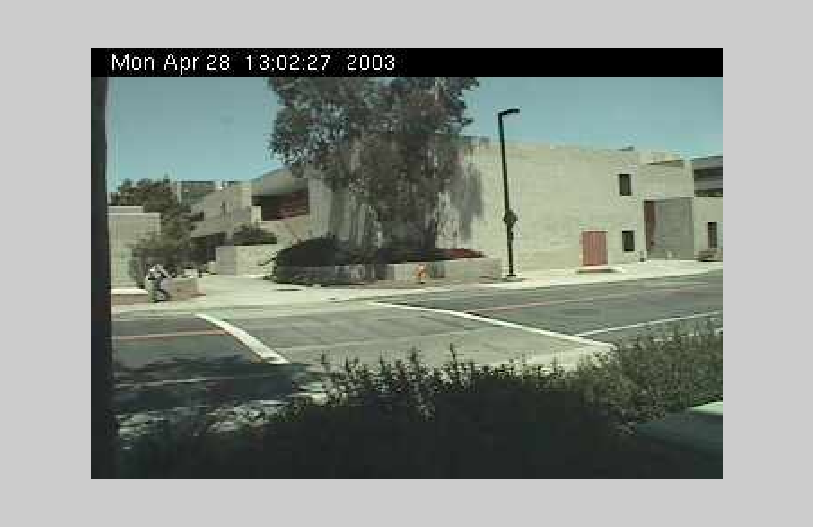


Figure 4.1: Background Image

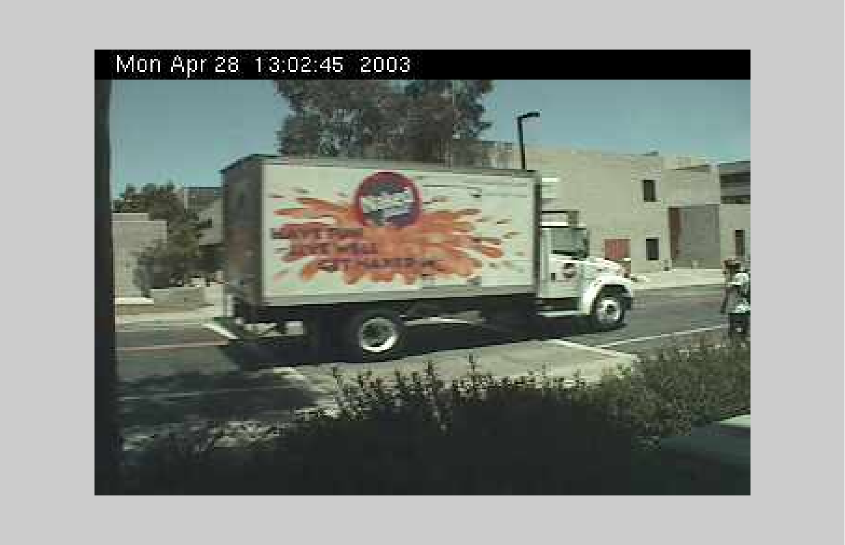


Figure 4.2: Additional Image in Sequence

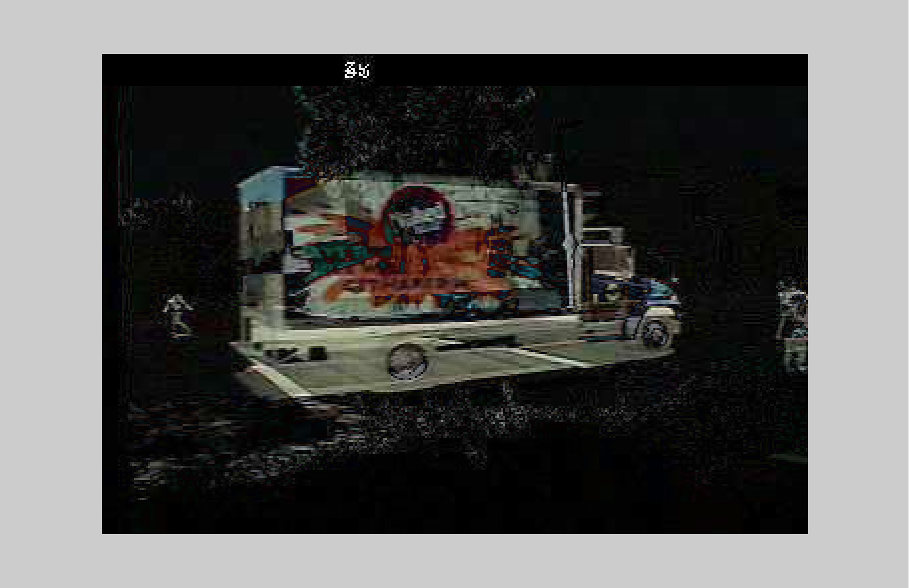


Figure 4.3: Absolute Difference of Background and Image in Sequence

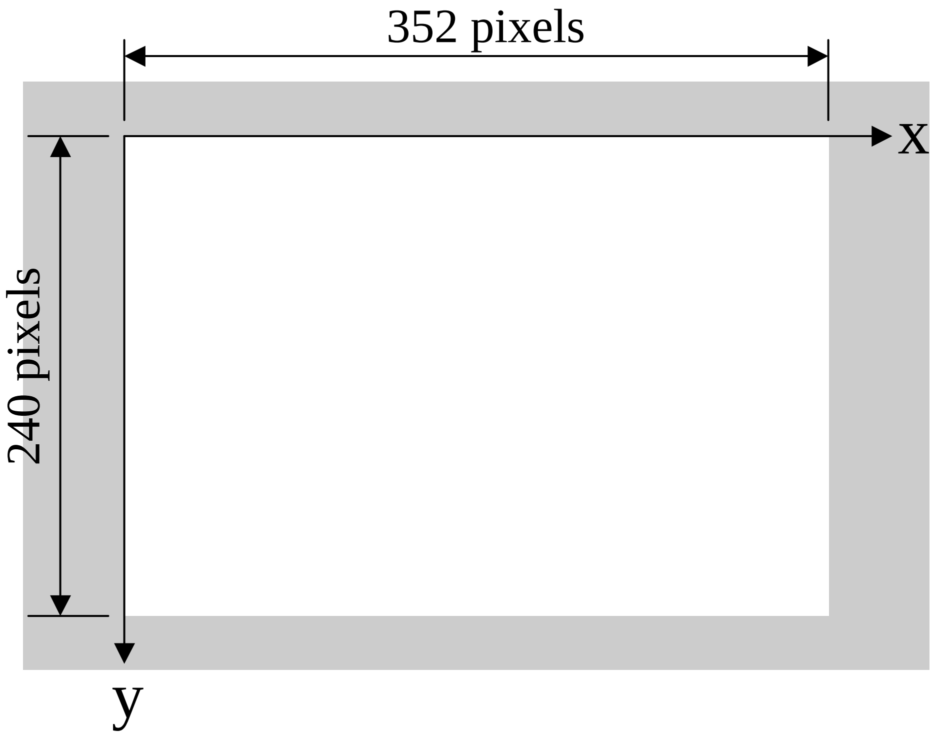


Figure 4.4: Pixel Coordinate System Used in Matlab

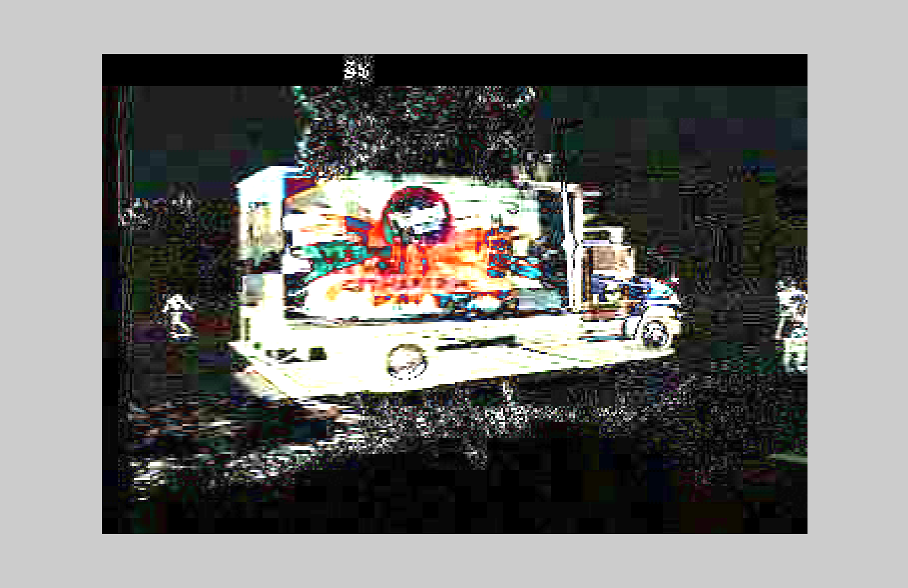


Figure 4.5: Absolute Difference After Brightening

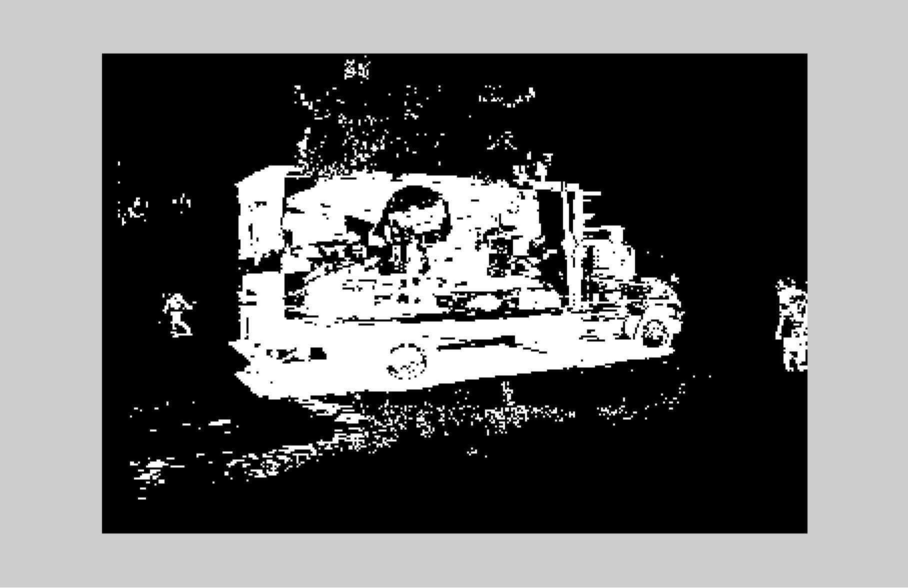


Figure 4.6: Binary (Black and White) Image

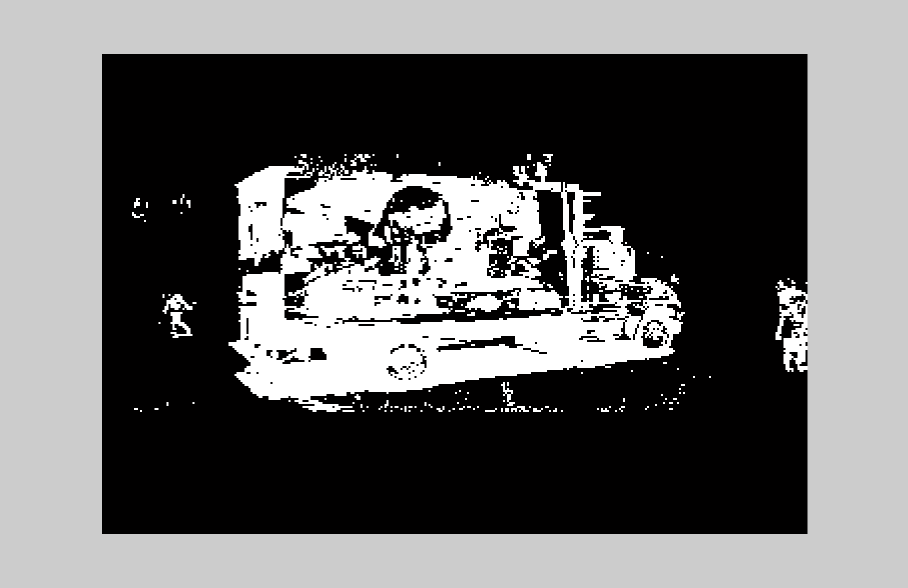


Figure 4.7: Black and White Image after Cropping

Figure 4.8: Binary Image after Holes Have Been Filled

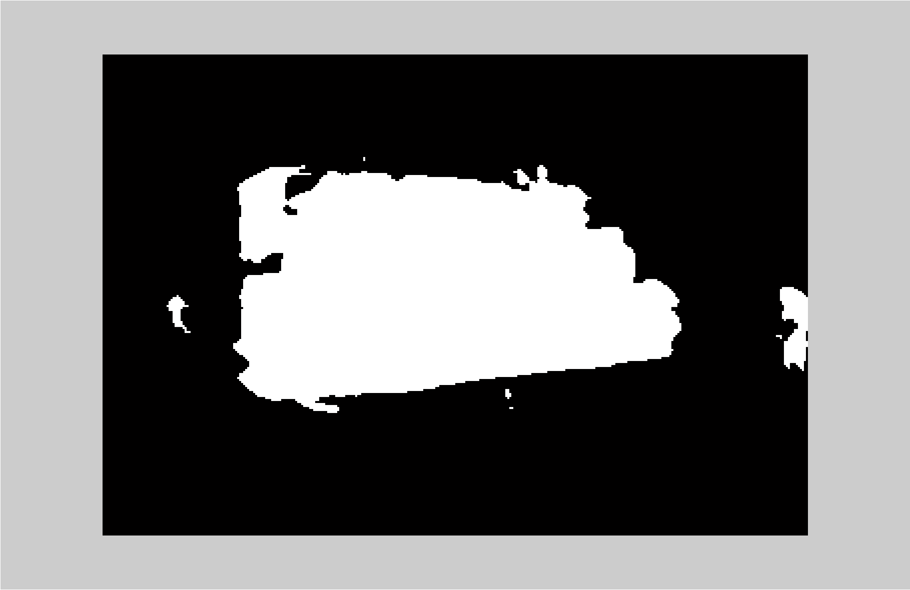


Figure 4.9: Resulting Image after Filtering

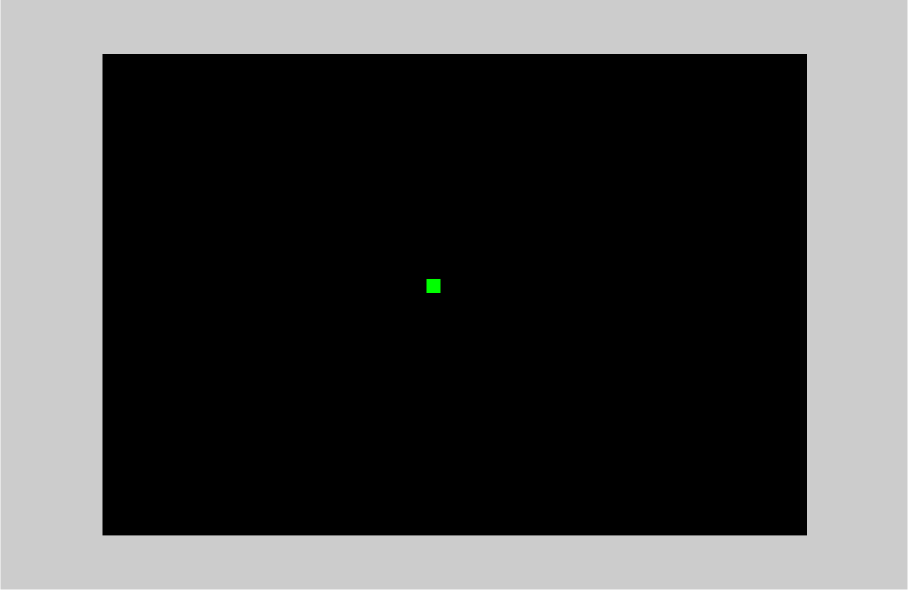


Figure 4.10: Centroid of Largest Object – Note All Other Pixels Are Equal to Zero

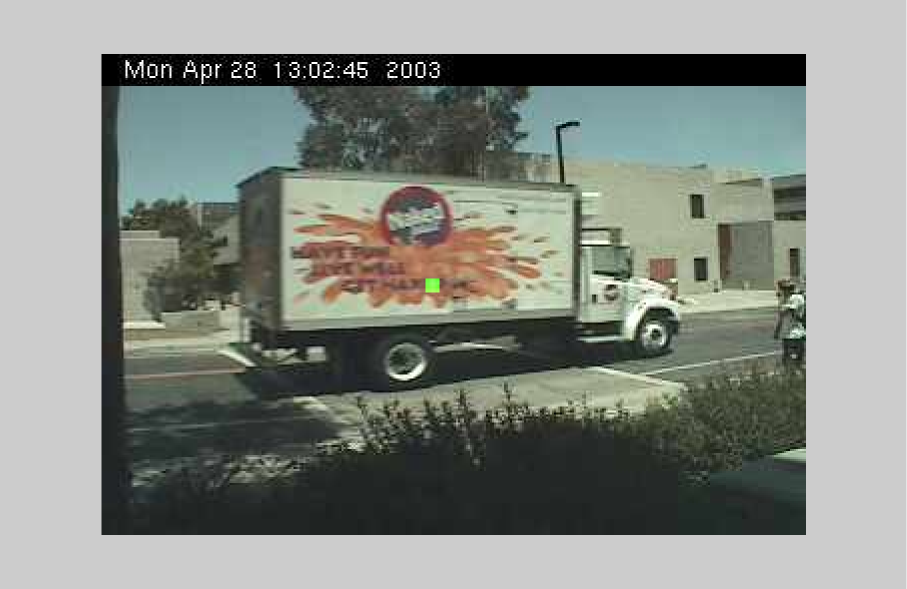


Figure 4.11: Original Figure with Superimposed Location of Centroid (Green Dot).

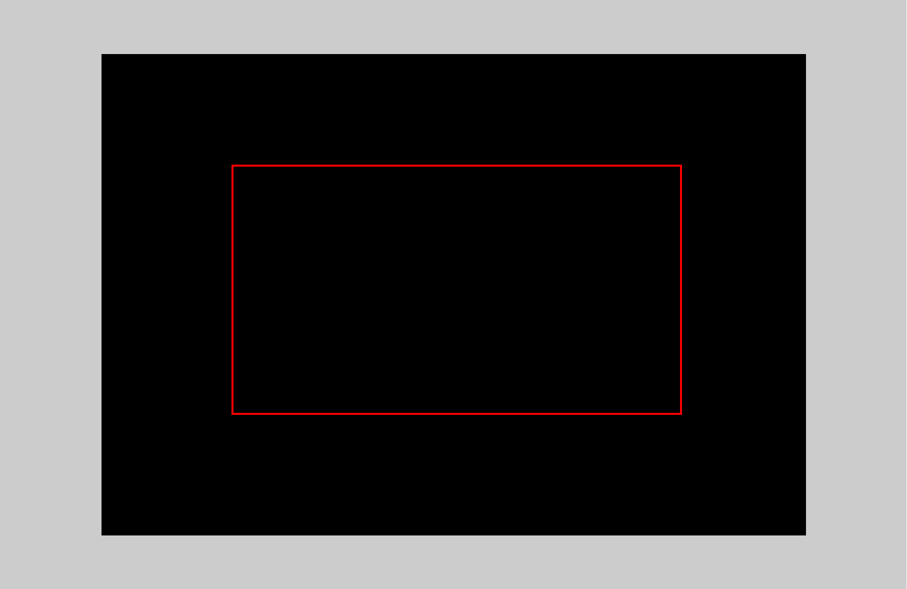


Figure 4.12: Bounding Box – Note Every Pixel Not On the Border of the Bounding Is Equal to Zero.

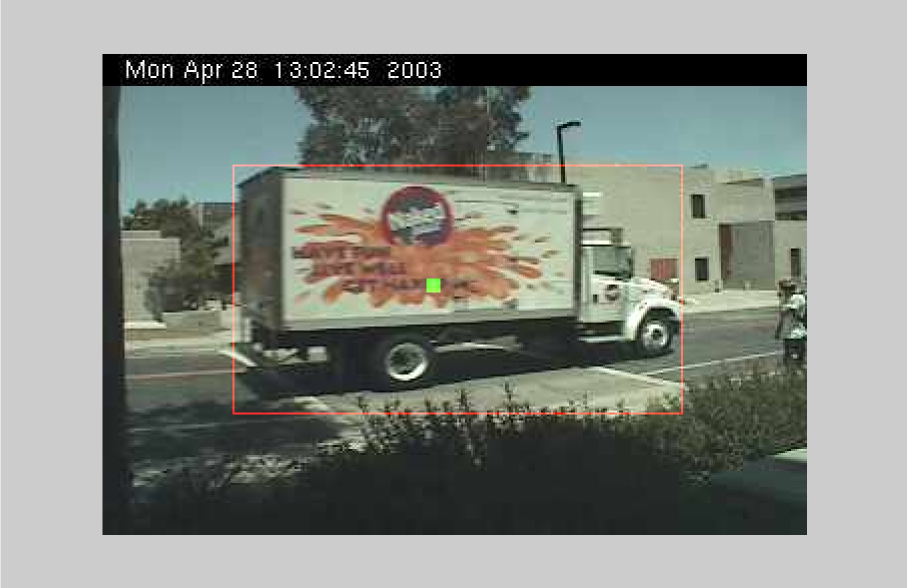


Figure 4.13: Original Figure with Superimposed Bounding Box (Red box) and Centroid (Green Dot).

Table 4.1: Region Properties for Sample Feature Extraction

ΣArea (pixels) =	20805
Area of largest object (pixels) =	20287
Horizontal Centroid ($\bar{x}$) (pixels) =	166.0585
Vertical Centroid ($\bar{y}$) (pixels) =	116.0282
Horizontal Length of Bounding Box (Pixels) =	224
Vertical Height of Bounding Box (Pixels) =	124
Direction =	East
Classification =	4 (Large Truck)

4.2 Classification (Schema) of Extracted Features

A stimulus is a load pattern generated by direct observation (by post-processing traffic information captured by video cameras), from a load generation software used by simulation engines or by a physical load mechanism applied to the structure. A novel aspect of this research is to develop tools to allow for the creation of a load database for video data. For video data, the goal is to be able to record the types (an 18-wheeler vs. a compact sedan) and positions of load objects at specific time instants within the database. This will be stored as a spatially indexed valid-time temporal data coming from the video analysis engine, which will be converted to a load by a video data wrapper process. This is to be accomplished using a separate lookup table or a load generation function for each object recorded at a time instant.

To one day realize these goals, it is necessary to develop a schema for the video data and determine which extracted features from the video to consider for defining the stimulus provided by the passing traffic. With the quality of video available from the composite decks, the following features are utilized: detected area of passing traffic, horizontal and vertical centroids, dimensions and aspect ratio of bounding box. For each recorded image, these features are determined and stored within the database. Based on these features, a very rough estimation of the vehicle type is made; however, a more comprehensive classification method based on the use of neural networks applied to recorded strains and video is detailed in Elgamal et al., 2005.

Once the video limitations, discussed in the following section, have been overcome additional features will become available and greatly aid this load estimation. These features include the exact location of the vehicle at a known time instant, edges of the vehicle, color density, etc. It is expected, these improved features will make possible cotraining of a neural network based on recorded strains and extracted video features, thereby greatly increasing the power of the traffic identification tools and allowing for true load estimations.

4.3 Limitations

There are two main categories of limitations/drawbacks associated with the video and image processing techniques. The first class is associated with the hardware and the second with the image processing algorithms. With the hardware, insufficient resolution, insufficient and inconsistent frame rates, and indexing errors are perhaps the most significant sources of error. If the camera resolution is too low, it becomes difficult to find the edges and shape of an object that one is trying to detect, particularly for methods that rely on edge detection for determining the objects dynamic properties (i.e., speed). Further, low resolution can also lead to errors in attempting to separate the background from the object creating gaps in the object which are mistaken for the background (Fig. 4.14). This problem proved quite troublesome on the composite decks, but has been remedied with higher resolution cameras incorporated the improved system employed on the Voigt Bridge testbed (discussed in detail in Chapter 9).

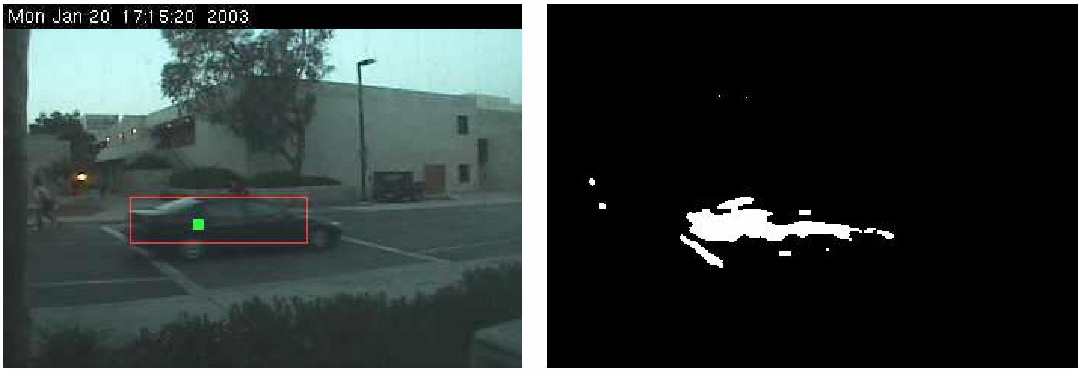


Figure 4.14: Original and Processed Image with Only Partial Detection and Gaps Resulting From Insufficient Resolution

Insufficient frame rates lead to problems with failing to detect portions of high speed vehicles. On the composite deck testbed, the frame rate was controlled by the internal processor of the network camera. The processor transmitted the images over the campus network to a remote computer located in another lab using an FTP connection. While the network connection was capable of very high frame rates, the internal processor was only able to maintain 4 frames-per-second. This low speed frame rate led to missed traffic in the recorded video data as seen in Figure 4.15. This problem became more pronounced at nights and weekends when vehicle speeds tended to increase. In addition to being hampered by slow frame rates, network cameras, like the one used on the composite decks, also suffer from inconsistent frame rates. While the camera employed in this application sampled at a nominal 4 Hz frame rate, the time between frames was not consistently 0.25 sec. This made calculations of speed, based on a constant frame rate, impossible. Again, these are problems that have been solved on the newly employed testbed on Voigt Bridge, where the acquisition rate is controlled by the data acquisition computer.

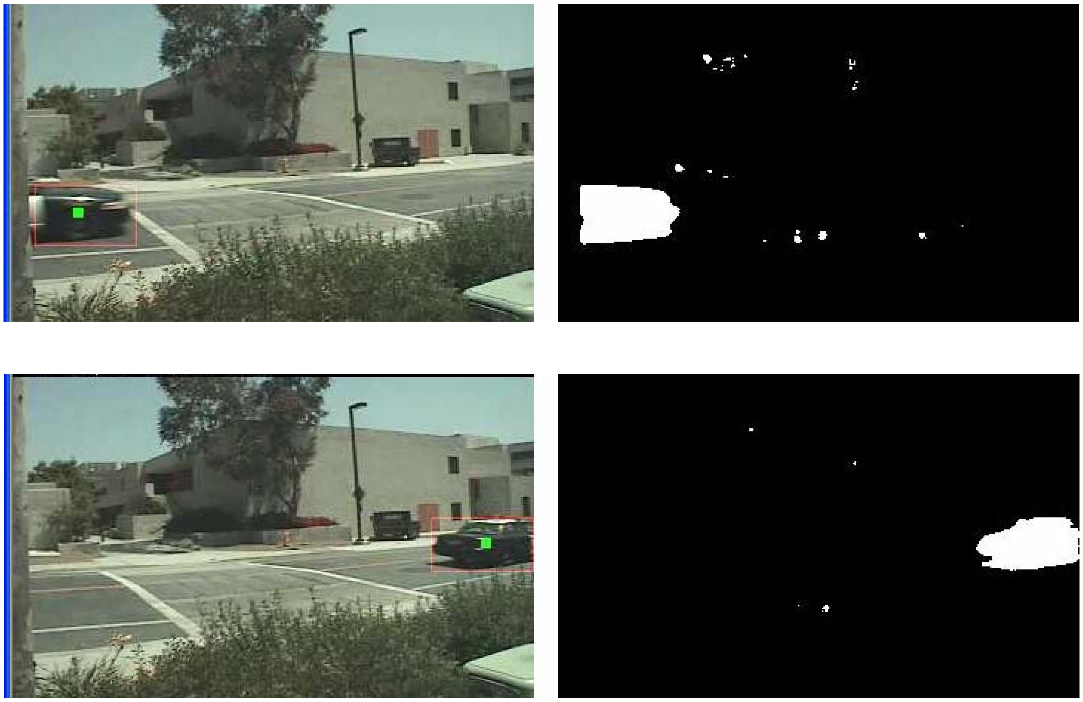


Figure 4.15: Consecutive Frames Showing only Front and Rear of Passing Vehicle.

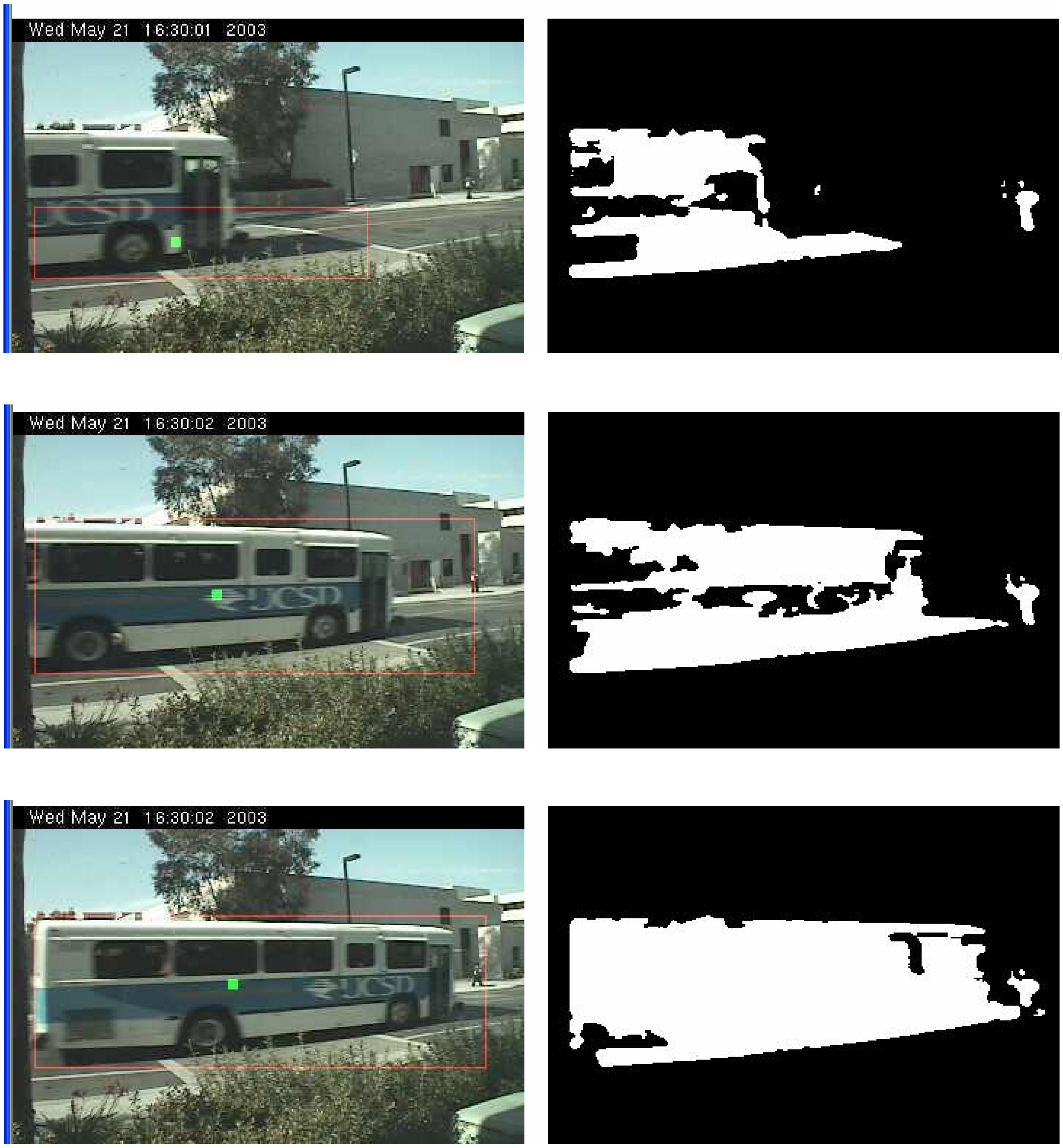


Figure 4.16: Three Consecutive Images (Taken of a UCSD Campus Shuttle Bus) Which Demonstrates the Inconsistent Frame Rate of the Network Camera.

The second class of errors is associated with the algorithms employed for feature extraction in the image processing operation. One problem that occurs when a single image is used to form the background in the background subtraction method is the creation of ghosts. If an object is present in the background image, but not in subsequent images it will show up as a stationary detected object (Fig. 4.17). In this figure, there is a pedestrian crossing the street in the first frame, but not in the later ones. When these later frames are processed and the difference between the two images is taken, the pedestrian continues to shows up. This problem can be reduced by forming a background from the combination of multiple images [Achler, O and Trivedi, M., 2004]. Many background subtraction methods are not robust enough to handle the possible change of scene lighting, moving shadows, etc. In recent years, most researchers have noticed these issues and a lot of efforts have been made towards solving them [Mikik et al., 200 and Prati et al., 2003]. An important issue that still needs to be addressed here is moving shadows. Severe shadows can result in large errors in object localization and can cause serious problems for the algorithms that use the moving object detection results for subsequent processes.

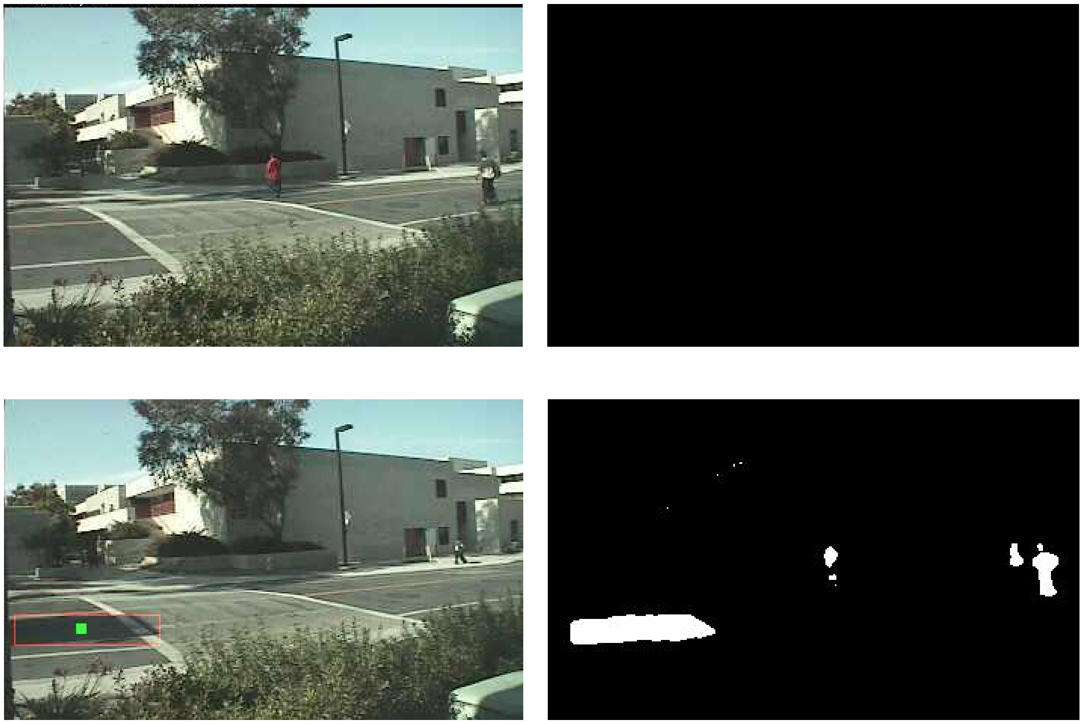


Figure 4.17: Background Image (Top) and Additional Frame (Bottom) with a Ghost of the Pedestrian (Right Hand Side of Image)

4.4 Data Fusion Application

Demonstration applications for the fusion of the raw video and extracted features with the recorded sensor data from the strain gages and microphone have been developed [Fraser et al., 2004]. This fusion of the sensor data is essential for understanding the interaction between the various sensors and extracting as much information as possible from each data source. One such example is detailed in Figure 4.18, which shows the original color image of a passing truck with the vehicle centroid (indicated by the green square) and bounding box (red box) in the left hand picture, the processed binary image (right hand picture), the time-synchronized quarter span (gage 1) strain time histories (below the video), as well as the tabulated data from the image processing algorithm. This derived data includes the direction of the vehicle, area of the detected vehicle (in pixels squared), the dimensions of the bounding box, a vehicle type (based on the aforementioned area).

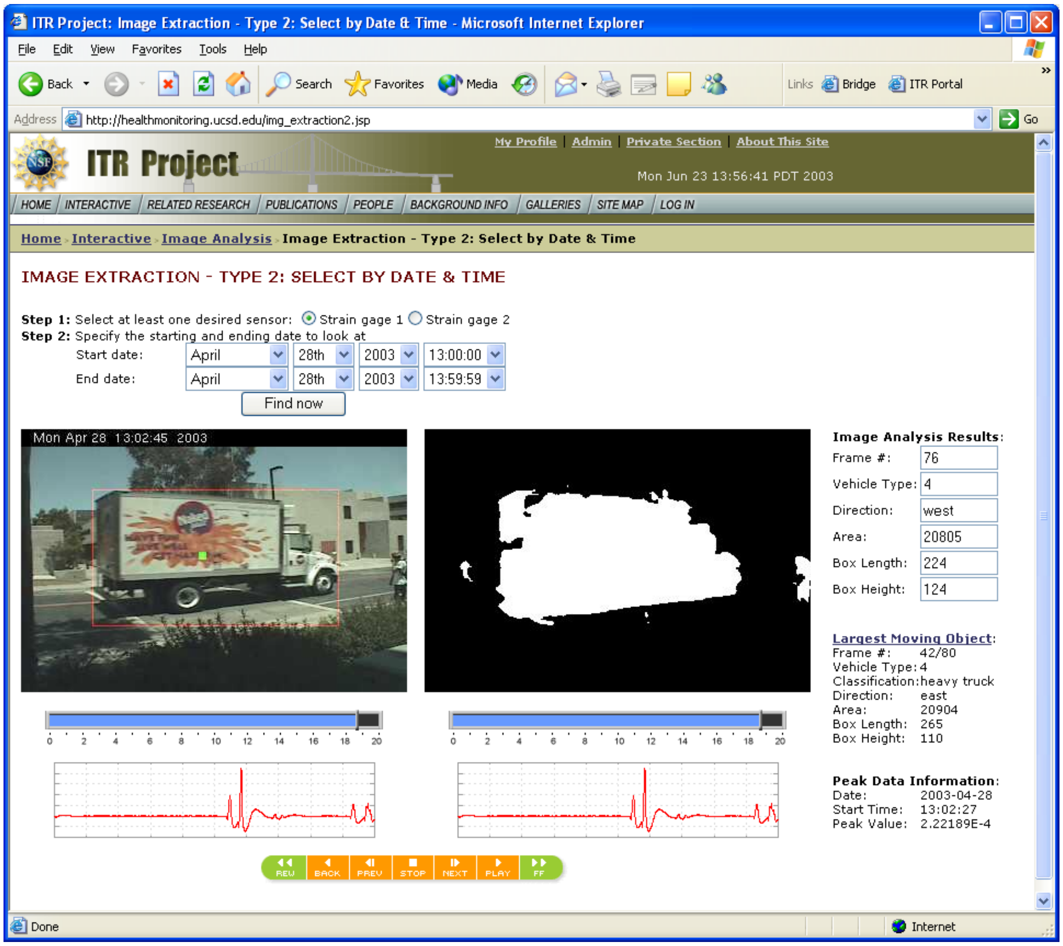

ITR Project: Image Extraction - Type 2: Select by Date & Time - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Links Bridge ITR Portal
Address http://healthmonitoring.ucsd.edu/img_extraction2.jsp
Go
ITR Project
My Profile | Admin | Private Section | About This Site
Mon Jun 23 13:56:41 PDT 2003
HOME | INTERACTIVE | RELATED RESEARCH | PUBLICATIONS | PEOPLE | BACKGROUND INFO | GALLERIES | SITE MAP | LOG IN
Home . Interactive . Image Analysis . Image Extraction - Type 2: Select by Date & Time
Mon Apr 28 13:02:45 2003
0 2 4 6 8 10 12 14 16 18 20
0 2 4 6 8 10 12 14 16 18 20
REV BACK PREV STOP NEXT PLAY FF
Done
Internet

IMAGE EXTRACTION - TYPE 2: SELECT BY DATE & TIME

Step 1: Select at least one desired sensor: ● Strain gage 1 ○ Strain gage 2
Step 2: Specify the starting and ending date to look at

Start date:	April	28th	2003	13:00:00
End date:	April	28th	2003	13:59:59

Find now

Image Analysis Results:

Frame #:	76
Vehicle Type:	4
Direction:	west
Area:	20805
Box Length:	224
Box Height:	124

Largest Moving Object:

Frame #:	42/80
Vehicle Type:	4
Classification:	heavy truck
Direction:	east
Area:	20904
Box Length:	265
Box Height:	110

Peak Data Information:

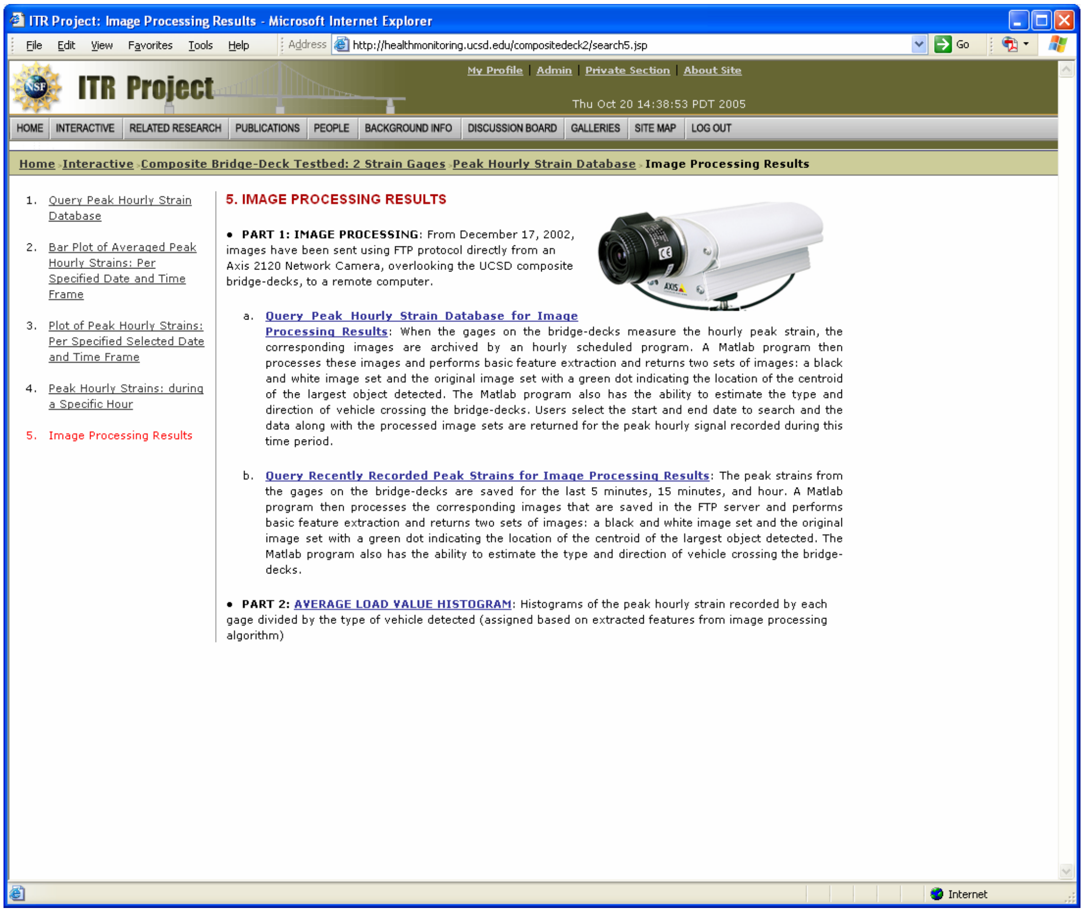
Date:	2003-04-28
Start Time:	13:02:27
Peak Value:	2.22189E-4

Figure 4.18: Example of Data Fusion with Results of Image Extraction and Strain Data in Database

4.5 Website for Displaying Video (with Extracted Features) with Sensor Data

Within the ITR portal, a webpage (http://healthmonitoring.ucsd.edu/imageAnalysis.jsp) was created to query and process image sequences from the database and return extracted features (Fig. 4.19). The first portion of the webpage allows users to query the database for the 2-channel peak hourly strain (Fig. 4.20). For these queries, users may display results based on the time the data was recorded, by the peak strain, or maximum area of the largest detected object. Returned features include the strain time histories, the original image set with a green dot indicating the location of the centroid of the largest object detected, the processed black and white image set, area and centroids of the largest object detected in each frame, and estimates of the type and direction of vehicle crossing the bridge-decks (based on the size of the vehicle detected and the horizontal motion of the centroid). A collection of data for varying vehicle types is included in Figures 4.21 – 4.26.

For this first batch of queries, all of the data has been previously processed and archived within the database. A second webpage allows users to view recently recorded data. Since the image processing algorithms are run on the hour, to view the extracted data corresponding to the peak strain for the last 5 minutes, 15 minutes, or current hour, it is necessary to first call on and run the Matlab image processing program for this data.


ITR Project: Image Processing Results - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Address
http://healthmonitoring.ucsd.edu/compositedeck2/search5.jsp
Go
ITR Project
My Profile | Admin | Private Section | About Site
Thu Oct 20 14:38:53 PDT 2005
HOME
INTERACTIVE
RELATED RESEARCH
PUBLICATIONS
PEOPLE
BACKGROUND INFO
DISCUSSION BOARD
GALLERIES
SITE MAP
LOG OUT
Home › Interactive › Composite Bridge-Deck Testbed: 2 Strain Gages › Peak Hourly Strain Database › Image Processing Results
Internet

1. Query Peak Hourly Strain Database
2. Bar Plot of Averaged Peak Hourly Strains: Per Specified Date and Time Frame
3. Plot of Peak Hourly Strains: Per Specified Selected Date and Time Frame
4. Peak Hourly Strains: during a Specific Hour
5. Image Processing Results

5. IMAGE PROCESSING RESULTS

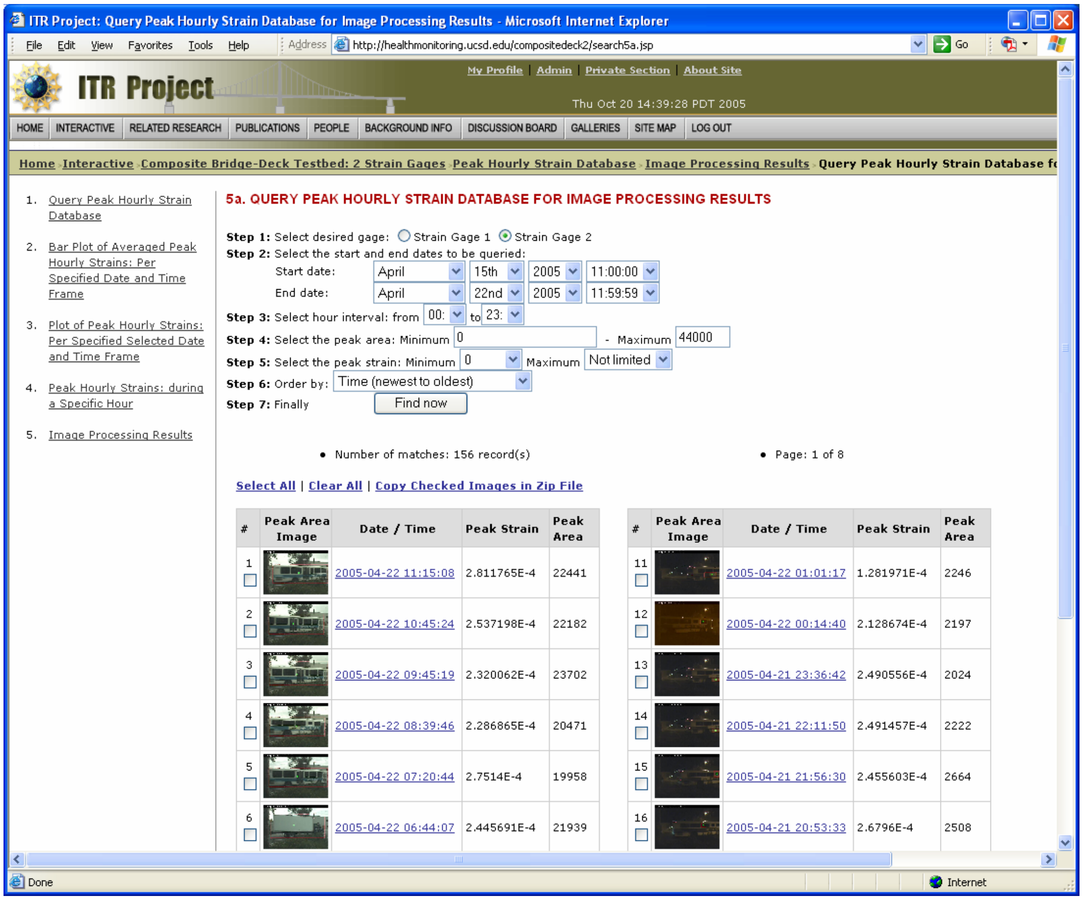
- **PART 1: IMAGE PROCESSING**: From December 17, 2002, images have been sent using FTP protocol directly from an Axis 2120 Network Camera, overlooking the UCSD composite bridge-decks, to a remote computer.

  a. **Query Peak Hourly Strain Database for Image Processing Results**: When the gages on the bridge-decks measure the hourly peak strain, the corresponding images are archived by an hourly scheduled program. A Matlab program then processes these images and performs basic feature extraction and returns two sets of images: a black and white image set and the original image set with a green dot indicating the location of the centroid of the largest object detected. The Matlab program also has the ability to estimate the type and direction of vehicle crossing the bridge-decks. Users select the start and end date to search and the data along with the processed image sets are returned for the peak hourly signal recorded during this time period.

  b. **Query Recently Recorded Peak Strains for Image Processing Results**: The peak strains from the gages on the bridge-decks are saved for the last 5 minutes, 15 minutes, and hour. A Matlab program then processes the corresponding images that are saved in the FTP server and performs basic feature extraction and returns two sets of images: a black and white image set and the original image set with a green dot indicating the location of the centroid of the largest object detected. The Matlab program also has the ability to estimate the type and direction of vehicle crossing the bridge-decks.

- **PART 2: AVERAGE LOAD VALUE HISTOGRAM**: Histograms of the peak hourly strain recorded by each gage divided by the type of vehicle detected (assigned based on extracted features from image processing algorithm)

Figure 4.19: Webpage for image Processing Results from Composite Bridge Deck Testbed


ITR Project: Query Peak Hourly Strain Database for Image Processing Results - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://healthmonitoring.ucsd.edu/compositedeck2/search5a.jsp
Go
ITR Project
My Profile | Admin | Private Section | About Site
Thu Oct 20 14:39:28 PDT 2005
HOME
INTERACTIVE
RELATED RESEARCH
PUBLICATIONS
PEOPLE
BACKGROUND INFO
DISCUSSION BOARD
GALLERIES
SITE MAP
LOG OUT
Home » Interactive » Composite Bridge-Deck Testbed: 2 Strain Gages » Peak Hourly Strain Database » Image Processing Results » Query Peak Hourly Strain Database fo
1. Query Peak Hourly Strain Database
2. Bar Plot of Averaged Peak Hourly Strains: Per Specified Date and Time Frame
3. Plot of Peak Hourly Strains: Per Specified Selected Date and Time Frame
4. Peak Hourly Strains: during a Specific Hour
5. Image Processing Results
Done
Internet

5a. QUERY PEAK HOURLY STRAIN DATABASE FOR IMAGE PROCESSING RESULTS

Step 1: Select desired gage: ○ Strain Gage 1 ● Strain Gage 2
Step 2: Select the start and end dates to be queried:
Start date: April 15th 2005 11:00:00
End date: April 22nd 2005 11:59:59
Step 3: Select hour interval: from 00: to 23:
Step 4: Select the peak area: Minimum 0 - Maximum 44000
Step 5: Select the peak strain: Minimum 0 Maximum Not limited
Step 6: Order by: Time (newest to oldest)
Step 7: Finally Find now

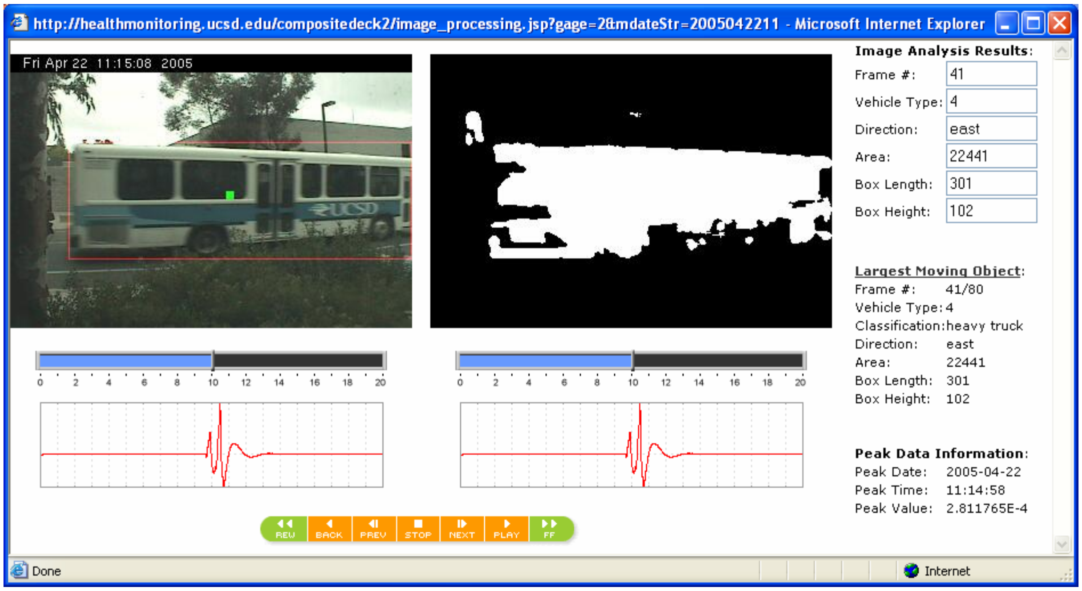
- Number of matches: 156 record(s)
- Page: 1 of 8

Select All | Clear All | Copy Checked Images in Zip File

#	Peak Area Image	Date / Time	Peak Strain	Peak Area
1 □		2005-04-22 11:15:08	2.811765E-4	22441
2 □		2005-04-22 10:45:24	2.537198E-4	22182
3 □		2005-04-22 09:45:19	2.320062E-4	23702
4 □		2005-04-22 08:39:46	2.286865E-4	20471
5 □		2005-04-22 07:20:44	2.7514E-4	19958
6 □		2005-04-22 06:44:07	2.445691E-4	21939

#	Peak Area Image	Date / Time	Peak Strain	Peak Area
11 □		2005-04-22 01:01:17	1.281971E-4	2246
12 □		2005-04-22 00:14:40	2.128674E-4	2197
13 □		2005-04-21 23:36:42	2.490556E-4	2024
14 □		2005-04-21 22:11:50	2.491457E-4	2222
15 □		2005-04-21 21:56:30	2.455603E-4	2664
16 □		2005-04-21 20:53:33	2.6796E-4	2508

Figure 4.20: Results of Querying Archived Strain Data and Image Processing Results – Arranged by Date.



http://healthmonitoring.ucsd.edu/compositedeck2/image_processing.jsp?gage=2&mdateStr=2005042211 - Microsoft Internet Explorer

Fri Apr 22 11:15:08 2005

Image Analysis Results:

Frame #:	41
Vehicle Type:	4
Direction:	east
Area:	22441
Box Length:	301
Box Height:	102

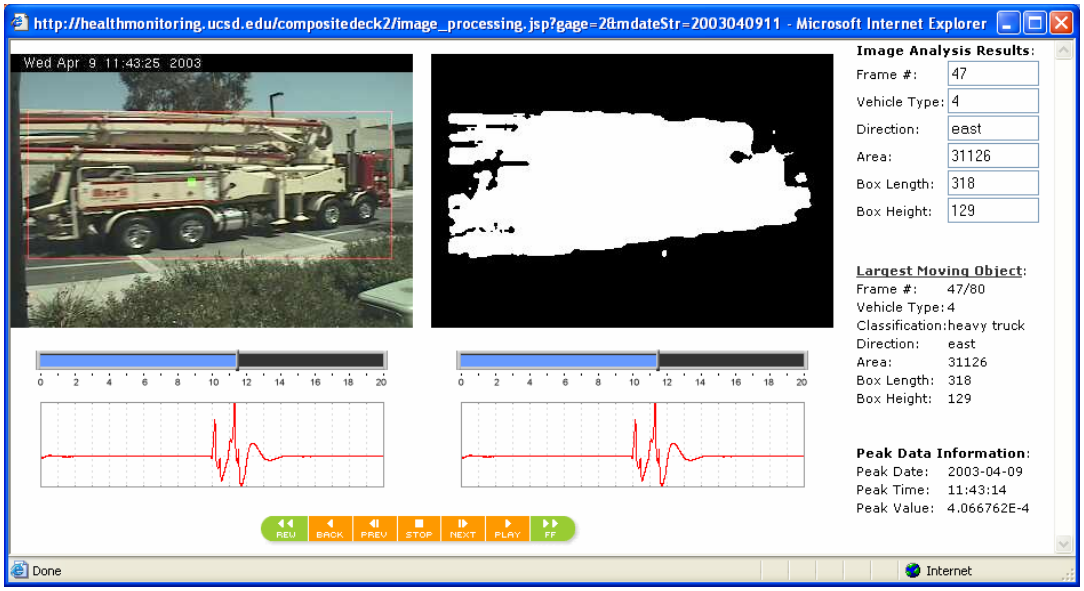
Largest Moving Object:

Frame #:	41/80
Vehicle Type:	4
Classification:	heavy truck
Direction:	east
Area:	22441
Box Length:	301
Box Height:	102

Peak Data Information:

Peak Date:	2005-04-22
Peak Time:	11:14:58
Peak Value:	2.811765E-4

Figure 4.21: Example of Data Fusion of Recorded Strains, Recorded Video, and Extracted Features for a Campus Shuttle Bus



http://healthmonitoring.ucsd.edu/compositedeck2/image_processing.jsp?gage=2&mdateStr=2003040911 - Microsoft Internet Explorer

Wed Apr 9 11:43:25 2003

Image Analysis Results:

Frame #:	47
Vehicle Type:	4
Direction:	east
Area:	31126
Box Length:	318
Box Height:	129

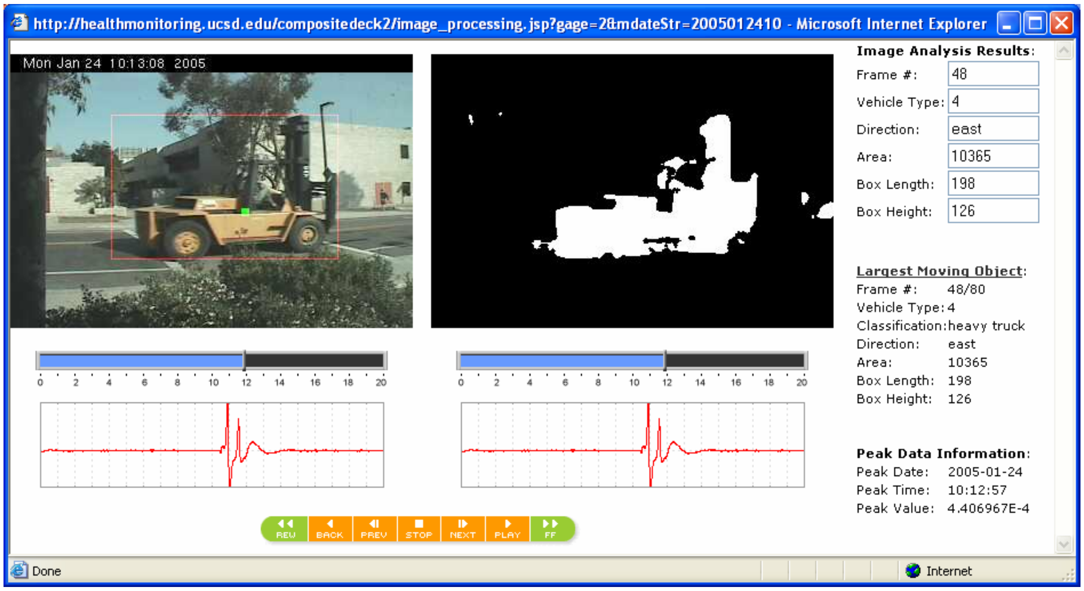
Largest Moving Object:

Frame #:	47/80
Vehicle Type:	4
Classification:	heavy truck
Direction:	east
Area:	31126
Box Length:	318
Box Height:	129

Peak Data Information:

Peak Date:	2003-04-09
Peak Time:	11:43:14
Peak Value:	4.066762E-4

Figure 4.22: Example of Data Fusion of Recorded Strains, Recorded Video, and Extracted Features for a Large Multi-Axle Truck



http://healthmonitoring.ucsd.edu/compositedeck2/image_processing.jsp?gage=2&mdateStr=2005012410 - Microsoft Internet Explorer

Mon Jan 24 10:13:08 2005

Image Analysis Results:

Frame #: 48

Vehicle Type: 4

Direction: east

Area: 10365

Box Length: 198

Box Height: 126

Largest Moving Object:

Frame #: 48/80

Vehicle Type: 4

Classification: heavy truck

Direction: east

Area: 10365

Box Length: 198

Box Height: 126

Peak Data Information:

Peak Date: 2005-01-24

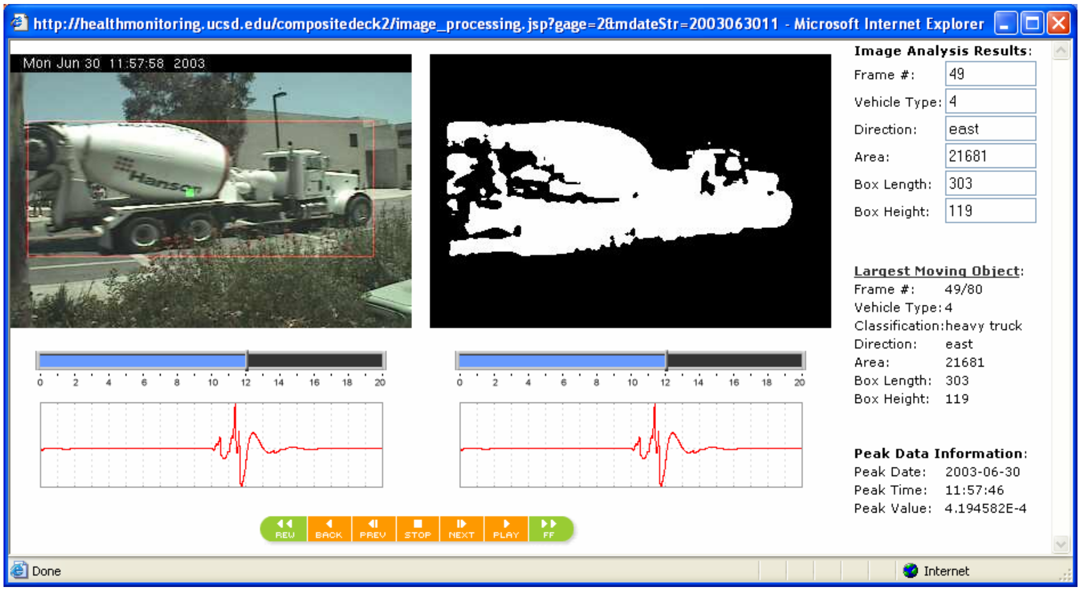
Peak Time: 10:12:57

Peak Value: 4.406967E-4

Done

Internet

Figure 4.23: Example of Data Fusion of Recorded Strains, Recorded Video, and Extracted Features for a Forklift Truck



http://healthmonitoring.ucsd.edu/compositedeck2/image_processing.jsp?gage=2&mdateStr=2003063011 - Microsoft Internet Explorer

Mon Jun 30 11:57:58 2003

Image Analysis Results:

Frame #: 49

Vehicle Type: 4

Direction: east

Area: 21681

Box Length: 303

Box Height: 119

Largest Moving Object:

Frame #: 49/80

Vehicle Type: 4

Classification: heavy truck

Direction: east

Area: 21681

Box Length: 303

Box Height: 119

Peak Data Information:

Peak Date: 2003-06-30

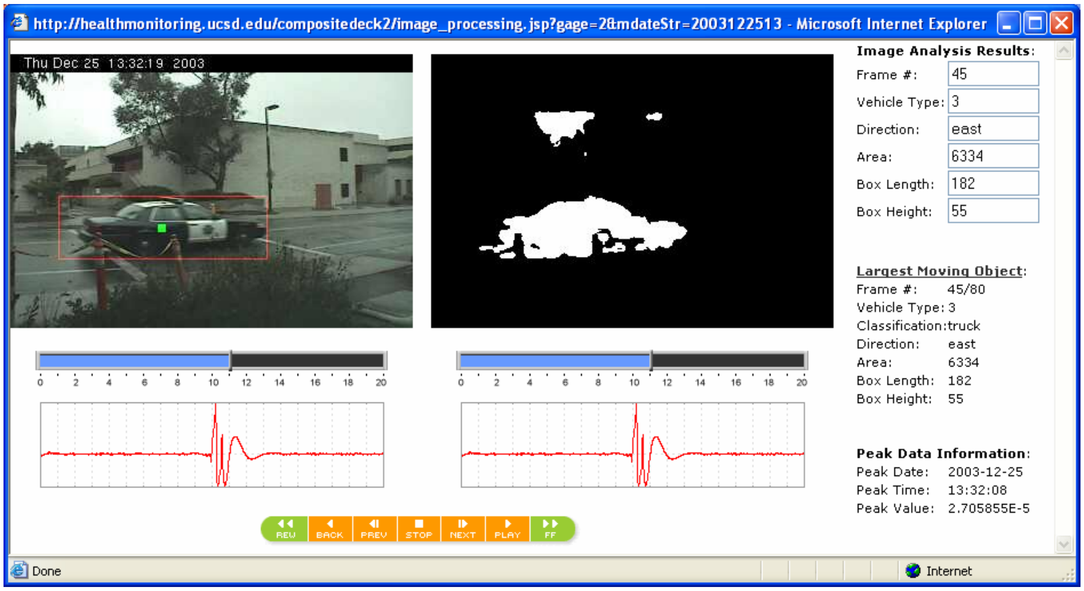
Peak Time: 11:57:46

Peak Value: 4.194582E-4

Done

Internet

Figure 4.24: Example of Data Fusion of Recorded Strains, Recorded Video, and Extracted Features for a Concrete Truck



http://healthmonitoring.ucsd.edu/compositedeck2/image_processing.jsp?gage=2&mdateStr=2003122513 - Microsoft Internet Explorer

Thu Dec 25 13:32:19 2003

Image Analysis Results:

Frame #:	45
Vehicle Type:	3
Direction:	east
Area:	6334
Box Length:	182
Box Height:	55

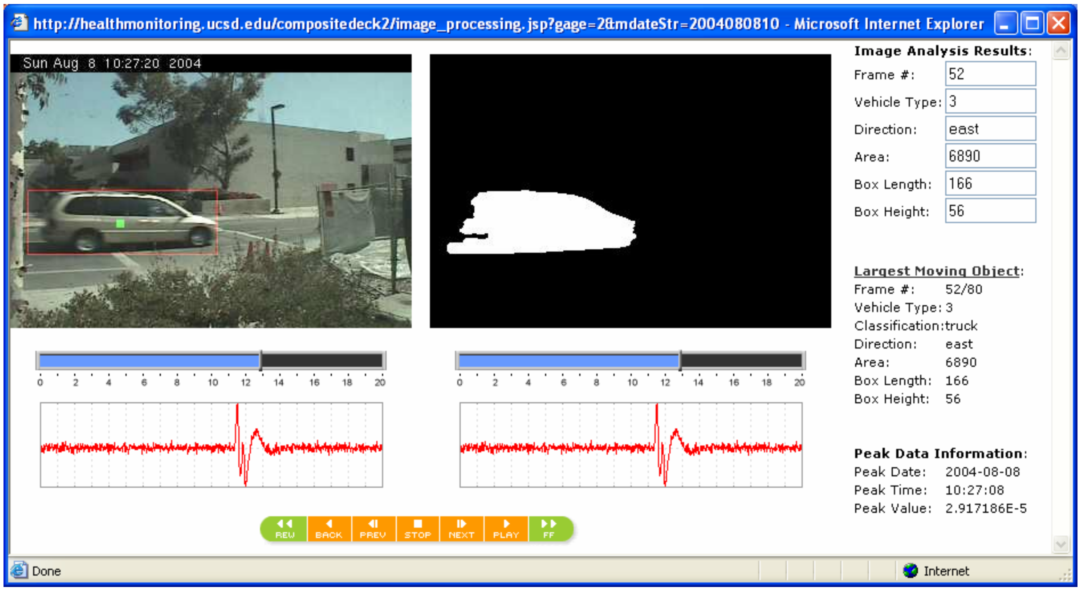
Largest Moving Object:

Frame #:	45/80
Vehicle Type:	3
Classification:	truck
Direction:	east
Area:	6334
Box Length:	182
Box Height:	55

Peak Data Information:

Peak Date:	2003-12-25
Peak Time:	13:32:08
Peak Value:	2.705855E-5

Figure 4.25: Example of Data Fusion of Recorded Strains, Recorded Video, and Extracted Features for a Campus Police Car



http://healthmonitoring.ucsd.edu/compositedeck2/image_processing.jsp?gage=2&mdateStr=2004080810 - Microsoft Internet Explorer

Sun Aug 8 10:27:20 2004

Image Analysis Results:

Frame #:	52
Vehicle Type:	3
Direction:	east
Area:	6890
Box Length:	166
Box Height:	56

Largest Moving Object:

Frame #:	52/80
Vehicle Type:	3
Classification:	truck
Direction:	east
Area:	6890
Box Length:	166
Box Height:	56

Peak Data Information:

Peak Date:	2004-08-08
Peak Time:	10:27:08
Peak Value:	2.917186E-5

Figure 4.26: Example of Data Fusion of Recorded Strains, Recorded Video, and Extracted Features for a Minivan

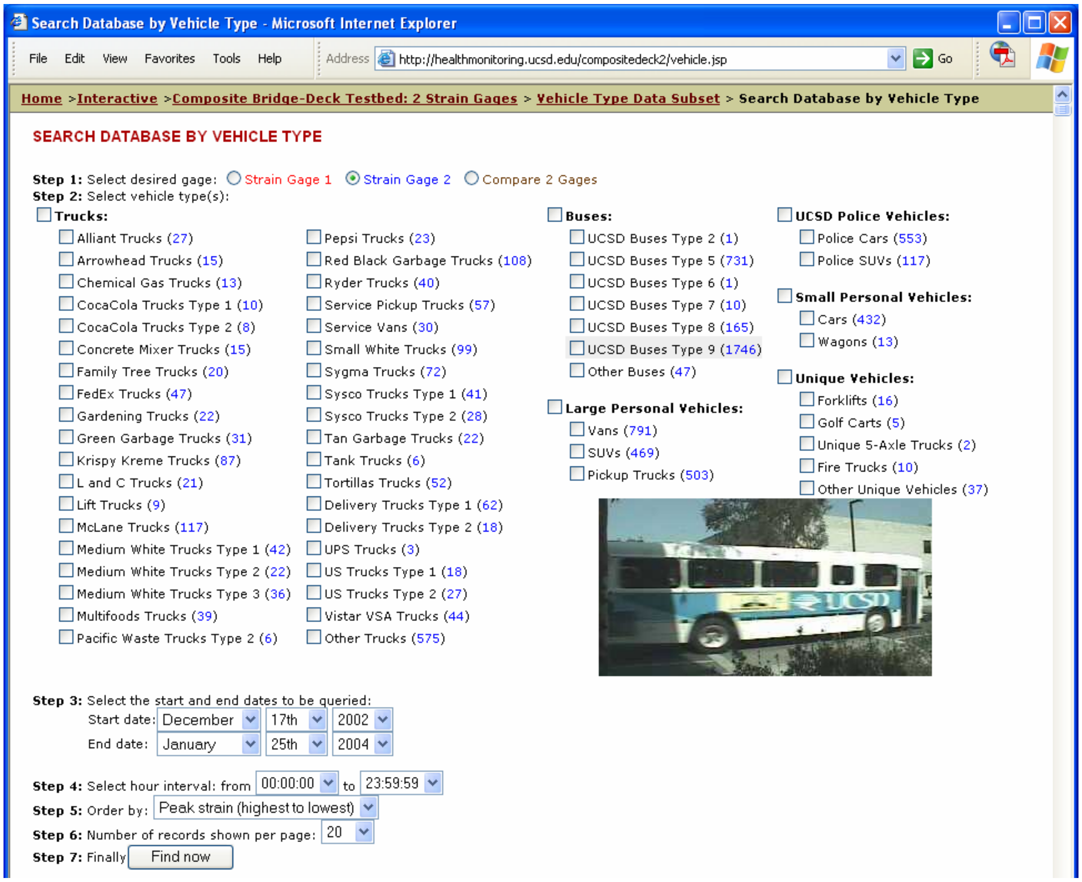
4.6 Creation of Labeled Data Sets for Analysis

An important preliminary step in the data analysis process was classification of the vehicles into distinct bins based on vehicle types. Viewing captured still-frame thumbnails and videos, the vehicles were manually sorted into bins including Trucks, Buses, Large Personal Vehicles, UCSD Police Vehicles, Small Personal Vehicles, and Unique Vehicles. Within each of these broad bins, further sorting was performed to identify types of vehicles with consistent number of axles (i.e. vehicle models), axle spacing, and/or weight distribution. Additionally, this extensive binning process incorporated cleansing of the data, removing all data records where (1) multiple vehicle crossings occur three seconds before or after the absolute peak strain, or (2) time synchronization of the video and strain time histories vary significantly.

The resulting data set (compiled in collaboration with Kendra Oliver) contains 6265 records binned into 57 vehicle types (Fig. 4.27). Profiles of the fifty seven vehicle types with corresponding images are available in Appendix 1. It is the author’s opinion that this data set can prove extremely valuable for testing algorithms for traffic management and vehicle classification using artificial intelligence algorithms. In particular for supervised learning methods, this type of labeled data is required for the training and evaluation phases. For each of these records, the dataset includes two strain gage time histories, videos, still-frame image thumbnails (both the original color thumbnail and the image-processed binary thumbnail, discussed later in further detail), vehicle types, and environmental temperatures. This dataset is publicly available over the worldwide web at http://healthmonitoring.ucsd.edu (Fig. 4.27) for either (1) querying (by selecting “Interactive” and then “Composite Testbed Deck Panels: 2 Channels” in the tool bar, Fig. 4.28), (2) downloading (by clicking on the “Download Strain Gage Data” button, Fig. 4.29), or (3) direct DB2 database access (by requesting further assistance).

In addition, simple analysis applications are available on-line using this labeled data. For examples, users may query the database and examine interactive scatter plots of channel 1 and 2 peak strains for each of the vehicle types (Fig. 4.30). These plots are useful in drawing rough distinctions in the data (for example separating cars from small trucks from large trucks). Another example is shown Figure 4.31, which shows the peak strain distributions and Gaussian fit of the recorded data of one or more selected vehicle types (in this case UCSD Buses Type 9).

A similar data set (compiled in collaboration with Kendra Oliver and Laura Flores), composed of 4,950 records has also been created for the 16-channel continuous monitoring system (http://healthmonitoring.ucsd.edu/compositedeck16_all/vehicle16.jsp).


Search Database by Vehicle Type - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://healthmonitoring.ucsd.edu/compositedeck2/vehicle.jsp
Go

Home >Interactive >Composite Bridge-Deck Testbed: 2 Strain Gages > Vehicle Type Data Subset > Search Database by Vehicle Type

SEARCH DATABASE BY VEHICLE TYPE

Step 1: Select desired gage: ○ Strain Gage 1 ● Strain Gage 2 ○ Compare 2 Gages

Step 2: Select vehicle type(s):

□ **Trucks:**

- □ Alliant Trucks (27)
- □ Arrowhead Trucks (15)
- □ Chemical Gas Trucks (13)
- □ CocaCola Trucks Type 1 (10)
- □ CocaCola Trucks Type 2 (8)
- □ Concrete Mixer Trucks (15)
- □ Family Tree Trucks (20)
- □ FedEx Trucks (47)
- □ Gardening Trucks (22)
- □ Green Garbage Trucks (31)
- □ Krispy Kreme Trucks (87)
- □ L and C Trucks (21)
- □ Lift Trucks (9)
- □ McLane Trucks (117)
- □ Medium White Trucks Type 1 (42)
- □ Medium White Trucks Type 2 (22)
- □ Medium White Trucks Type 3 (36)
- □ Multifoods Trucks (39)
- □ Pacific Waste Trucks Type 2 (6)
- □ Pepsi Trucks (23)
- □ Red Black Garbage Trucks (108)
- □ Ryder Trucks (40)
- □ Service Pickup Trucks (57)
- □ Service Vans (30)
- □ Small White Trucks (99)
- □ Sygma Trucks (72)
- □ Sysco Trucks Type 1 (41)
- □ Sysco Trucks Type 2 (28)
- □ Tan Garbage Trucks (22)
- □ Tank Trucks (6)
- □ Tortillas Trucks (52)
- □ Delivery Trucks Type 1 (62)
- □ Delivery Trucks Type 2 (18)
- □ UPS Trucks (3)
- □ US Trucks Type 1 (18)
- □ US Trucks Type 2 (27)
- □ Vistar VSA Trucks (44)
- □ Other Trucks (575)

□ **Buses:**

- □ UCSD Buses Type 2 (1)
- □ UCSD Buses Type 5 (731)
- □ UCSD Buses Type 6 (1)
- □ UCSD Buses Type 7 (10)
- □ UCSD Buses Type 8 (165)
- □ UCSD Buses Type 9 (1746)
- □ Other Buses (47)

□ **Large Personal Vehicles:**

- □ Vans (791)
- □ SUVs (469)
- □ Pickup Trucks (503)

□ **UCSD Police Vehicles:**

- □ Police Cars (553)
- □ Police SUVs (117)

□ **Small Personal Vehicles:**

- □ Cars (432)
- □ Wagons (13)

□ **Unique Vehicles:**

- □ Forklifts (16)
- □ Golf Carts (5)
- □ Unique 5-Axle Trucks (2)
- □ Fire Trucks (10)
- □ Other Unique Vehicles (37)

Step 3: Select the start and end dates to be queried:

Start date: December 17th 2002

End date: January 25th 2004

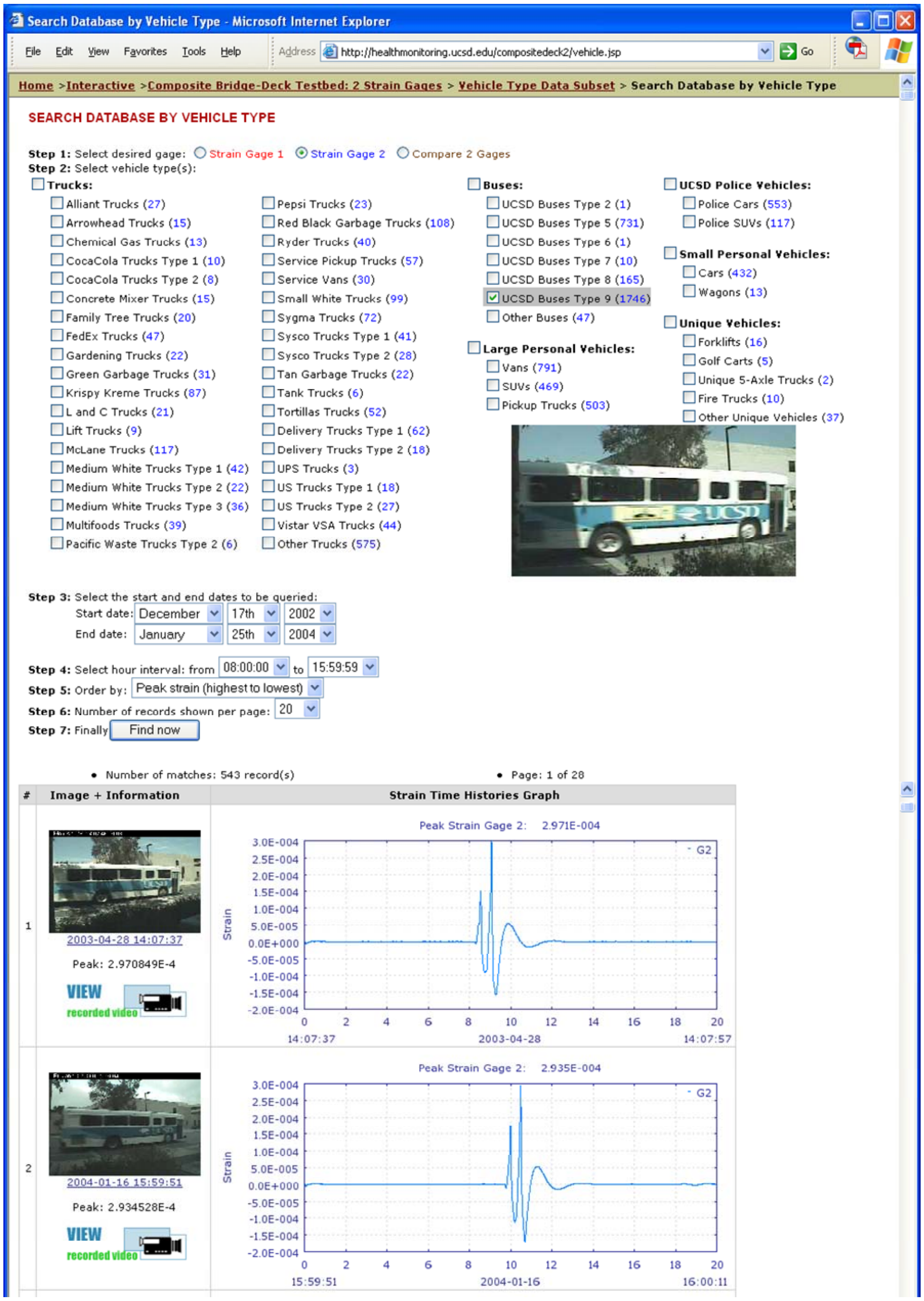
Step 4: Select hour interval: from 00:00:00 to 23:59:59

Step 5: Order by: Peak strain (highest to lowest)

Step 6: Number of records shown per page: 20

Step 7: Finally Find now

Figure 4.27: Website for Accessing the Labeled Data Set Containing 6265 Records Binned into 57 Vehicle Types (Webpage Created with the Assistance of Minh Phan and Kendra Oliver)



Search Database by Vehicle Type - Microsoft Internet Explorer

http://healthmonitoring.ucsd.edu/compositedeck2/vehicle.jsp

Home >Interactive >Composite Bridge-Deck Testbed: 2 Strain Gages > Vehicle Type Data Subset > Search Database by Vehicle Type

SEARCH DATABASE BY VEHICLE TYPE

Step 1: Select desired gage: ○ Strain Gage 1 ● Strain Gage 2 ○ Compare 2 Gages

Step 2: Select vehicle type(s):

□ **Trucks:**

- □ Alliant Trucks (27)
- □ Arrowhead Trucks (15)
- □ Chemical Gas Trucks (13)
- □ CocaCola Trucks Type 1 (10)
- □ CocaCola Trucks Type 2 (8)
- □ Concrete Mixer Trucks (15)
- □ Family Tree Trucks (20)
- □ FedEx Trucks (47)
- □ Gardening Trucks (22)
- □ Green Garbage Trucks (31)
- □ Krispy Kreme Trucks (87)
- □ L and C Trucks (21)
- □ Lift Trucks (9)
- □ McLane Trucks (117)
- □ Medium White Trucks Type 1 (42)
- □ Medium White Trucks Type 2 (22)
- □ Medium White Trucks Type 3 (36)
- □ Multifoods Trucks (39)
- □ Pacific Waste Trucks Type 2 (6)
- □ Pepsi Trucks (23)
- □ Red Black Garbage Trucks (108)
- □ Ryder Trucks (40)
- □ Service Pickup Trucks (57)
- □ Service Vans (30)
- □ Small White Trucks (99)
- □ Sygma Trucks (72)
- □ Sysco Trucks Type 1 (41)
- □ Sysco Trucks Type 2 (28)
- □ Tan Garbage Trucks (22)
- □ Tank Trucks (6)
- □ Tortillas Trucks (52)
- □ Delivery Trucks Type 1 (62)
- □ Delivery Trucks Type 2 (18)
- □ UPS Trucks (3)
- □ US Trucks Type 1 (18)
- □ US Trucks Type 2 (27)
- □ Vistar VSA Trucks (44)
- □ Other Trucks (575)

□ **Buses:**

- □ UCSD Buses Type 2 (1)
- □ UCSD Buses Type 5 (731)
- □ UCSD Buses Type 6 (1)
- □ UCSD Buses Type 7 (10)
- □ UCSD Buses Type 8 (165)
- ☑ UCSD Buses Type 9 (1746)
- □ Other Buses (47)

□ **Large Personal Vehicles:**

- □ Vans (791)
- □ SUVs (469)
- □ Pickup Trucks (503)

□ **UCSD Police Vehicles:**

- □ Police Cars (553)
- □ Police SUVs (117)

□ **Small Personal Vehicles:**

- □ Cars (432)
- □ Wagons (13)

□ **Unique Vehicles:**

- □ Forklifts (16)
- □ Golf Carts (5)
- □ Unique 5-Axle Trucks (2)
- □ Fire Trucks (10)
- □ Other Unique Vehicles (37)

Step 3: Select the start and end dates to be queried:

Start date: December 17th 2002

End date: January 25th 2004

Step 4: Select hour interval: from 08:00:00 to 15:59:59

Step 5: Order by: Peak strain (highest to lowest)

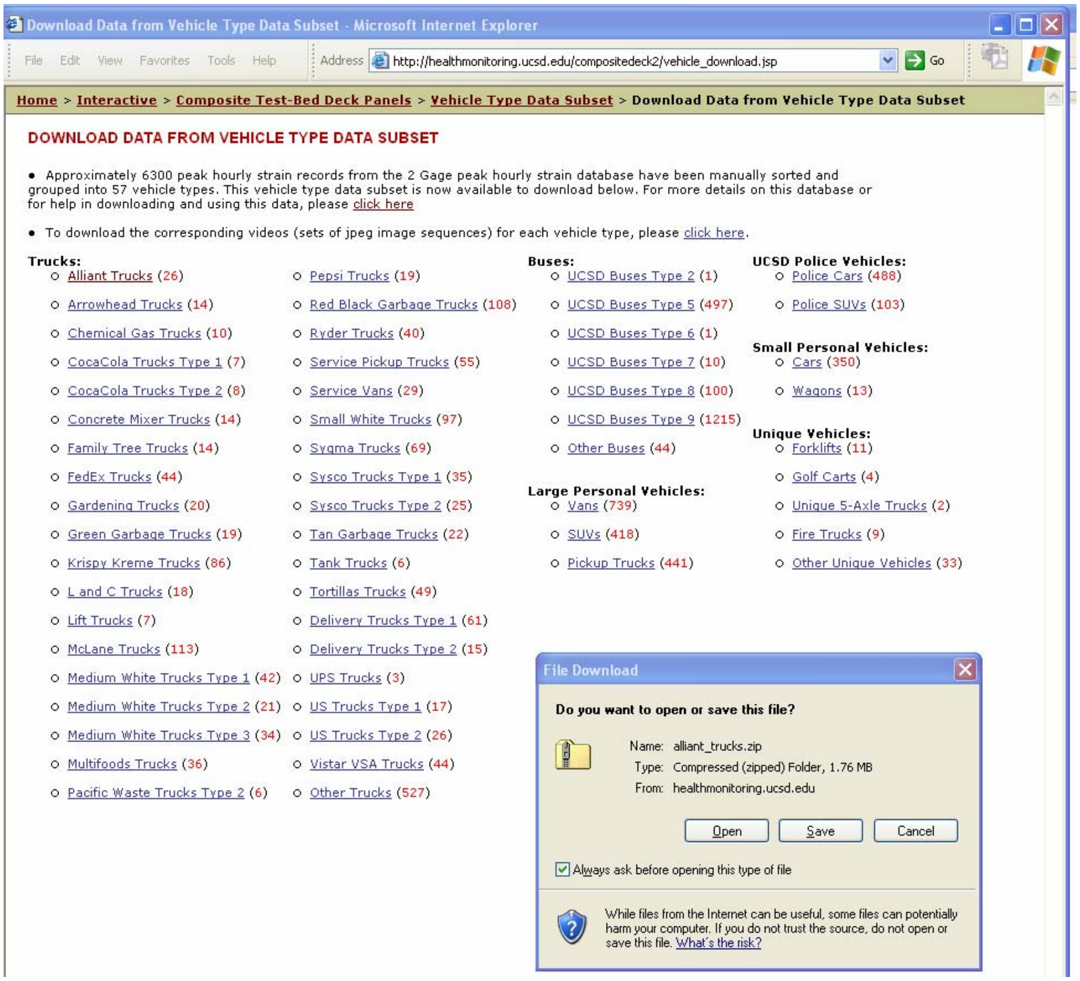
Step 6: Number of records shown per page: 20

Step 7: Finally Find now

- Number of matches: 543 record(s)
- Page: 1 of 28

#	Image + Information	Strain Time Histories Graph
1	2003-04-28 14:07:37 Peak: 2.970849E-4 VIEW recorded video	Peak Strain Gage 2: 2.971E-004 14:07:37 2003-04-28 14:07:57
2	2004-01-16 15:59:51 Peak: 2.934528E-4 VIEW recorded video	Peak Strain Gage 2: 2.935E-004 15:59:51 2004-01-16 16:00:11

Figure 4.28: Results of Querying 2-Channel Labeled Data Set for UCSD Buses Type 9 and Strain Gage 2 (Webpage Created with the Assistance of Minh Phan and Kendra Oliver)


Download Data from Vehicle Type Data Subset - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://healthmonitoring.ucsd.edu/compositedeck2/vehicle_download.jsp
Go

Home > Interactive > Composite Test-Bed Deck Panels > Vehicle Type Data Subset > Download Data from Vehicle Type Data Subset

DOWNLOAD DATA FROM VEHICLE TYPE DATA SUBSET

- Approximately 6300 peak hourly strain records from the 2 Gage peak hourly strain database have been manually sorted and grouped into 57 vehicle types. This vehicle type data subset is now available to download below. For more details on this database or for help in downloading and using this data, please click here
- To download the corresponding videos (sets of jpeg image sequences) for each vehicle type, please click here.

Trucks:
- Alliant Trucks (26)
- Arrowhead Trucks (14)
- Chemical Gas Trucks (10)
- CocaCola Trucks Type 1 (7)
- CocaCola Trucks Type 2 (8)
- Concrete Mixer Trucks (14)
- Family Tree Trucks (14)
- FedEx Trucks (44)
- Gardening Trucks (20)
- Green Garbage Trucks (19)
- Krispy Kreme Trucks (86)
- L and C Trucks (18)
- Lift Trucks (7)
- McLane Trucks (113)
- Medium White Trucks Type 1 (42)
- Medium White Trucks Type 2 (21)
- Medium White Trucks Type 3 (34)
- Multifoods Trucks (36)
- Pacific Waste Trucks Type 2 (6)
- Pepsi Trucks (19)
- Red Black Garbage Trucks (108)
- Ryder Trucks (40)
- Service Pickup Trucks (55)
- Service Vans (29)
- Small White Trucks (97)
- Sygma Trucks (69)
- Sysco Trucks Type 1 (35)
- Sysco Trucks Type 2 (25)
- Tan Garbage Trucks (22)
- Tank Trucks (6)
- Tortillas Trucks (49)
- Delivery Trucks Type 1 (61)
- Delivery Trucks Type 2 (15)
- UPS Trucks (3)
- US Trucks Type 1 (17)
- US Trucks Type 2 (26)
- Vistar VSA Trucks (44)
- Other Trucks (527)

Buses:
- UCSD Buses Type 2 (1)
- UCSD Buses Type 5 (497)
- UCSD Buses Type 6 (1)
- UCSD Buses Type 7 (10)
- UCSD Buses Type 8 (100)
- UCSD Buses Type 9 (1215)
- Other Buses (44)

Large Personal Vehicles:
- Vans (739)
- SUVs (418)
- Pickup Trucks (441)

UCSD Police Vehicles:
- Police Cars (488)
- Police SUVs (103)

Small Personal Vehicles:
- Cars (350)
- Wagons (13)

Unique Vehicles:
- Forklifts (11)
- Golf Carts (4)
- Unique 5-Axle Trucks (2)
- Fire Trucks (9)
- Other Unique Vehicles (33)

File Download

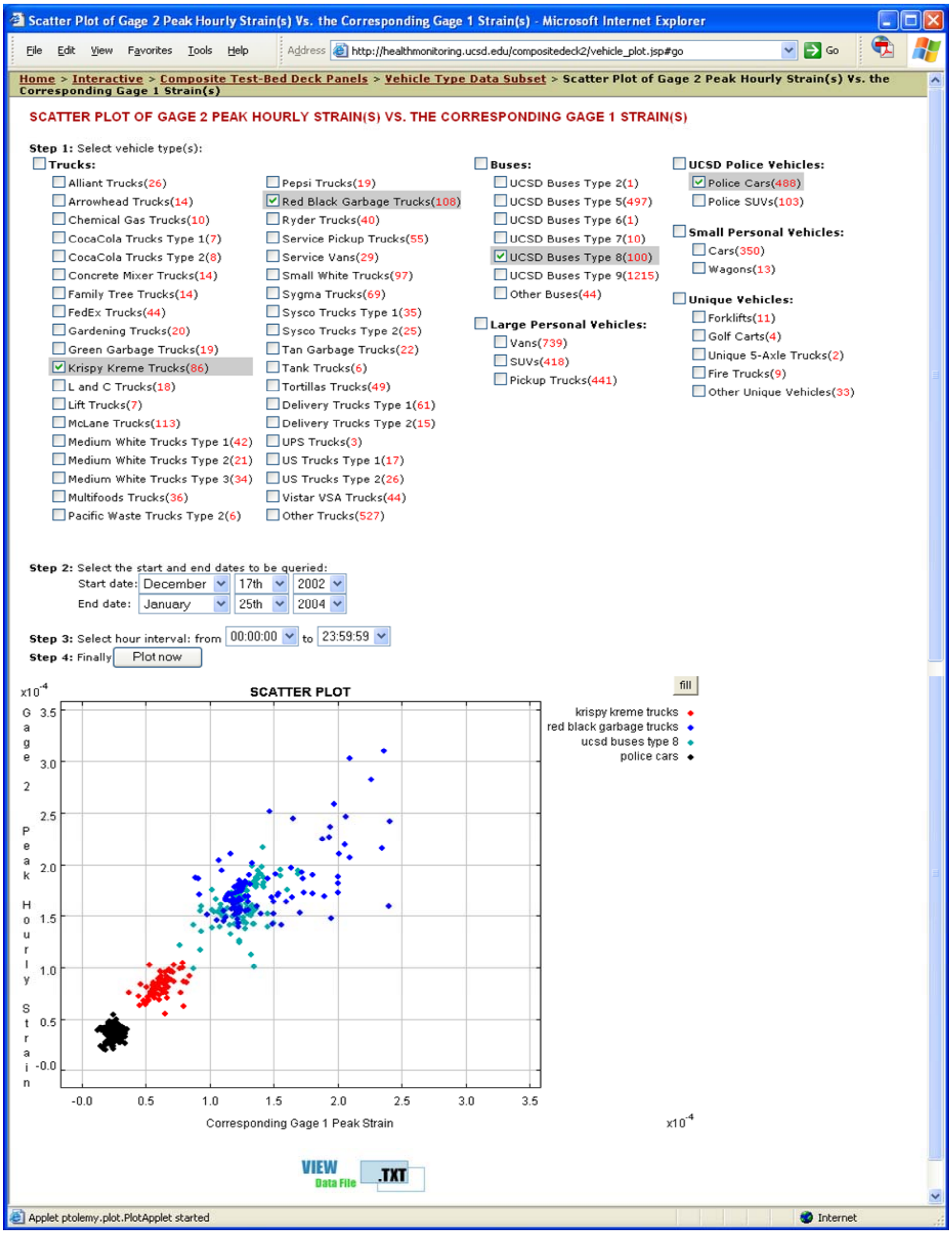
Do you want to open or save this file?

Name: alliant_trucks.zip
Type: Compressed (zipped) Folder, 1.76 MB
From: healthmonitoring.ucsd.edu

☑ Always ask before opening this type of file

While files from the Internet can be useful, some files can potentially harm your computer. If you do not trust the source, do not open or save this file. What's the risk?

Figure 4.29: On-line Data Download from 2-Channel Labeled Data Set for Alliant Trucks (Created in Collaboration with Minh Phan and Kendra Oliver)


Scatter Plot of Gage 2 Peak Hourly Strain(s) Vs. the Corresponding Gage 1 Strain(s) - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Address
http://healthmonitoring.ucsd.edu/compositedeck2/vehicle_plot.jsp#go
Go
Home > Interactive > Composite Test-Bed Deck Panels > Vehicle Type Data Subset > Scatter Plot of Gage 2 Peak Hourly Strain(s) Vs. the Corresponding Gage 1 Strain(s)

SCATTER PLOT OF GAGE 2 PEAK HOURLY STRAIN(S) VS. THE CORRESPONDING GAGE 1 STRAIN(S)

Step 1: Select vehicle type(s):

□ **Trucks:**

□ Alliant Trucks(26)
□ Arrowhead Trucks(14)
□ Chemical Gas Trucks(10)
□ CocaCola Trucks Type 1(7)
□ CocaCola Trucks Type 2(8)
□ Concrete Mixer Trucks(14)
□ Family Tree Trucks(14)
□ FedEx Trucks(44)
□ Gardening Trucks(20)
□ Green Garbage Trucks(19)
☑ Krispy Kreme Trucks(86)
□ L and C Trucks(18)
□ Lift Trucks(7)
□ McLane Trucks(113)
□ Medium White Trucks Type 1(42)
□ Medium White Trucks Type 2(21)
□ Medium White Trucks Type 3(34)
□ Multifoods Trucks(36)
□ Pacific Waste Trucks Type 2(6)
□ Pepsi Trucks(19)
☑ Red Black Garbage Trucks(108)
□ Ryder Trucks(40)
□ Service Pickup Trucks(55)
□ Service Vans(29)
□ Small White Trucks(97)
□ Sygma Trucks(69)
□ Sysco Trucks Type 1(35)
□ Sysco Trucks Type 2(25)
□ Tan Garbage Trucks(22)
□ Tank Trucks(6)
□ Tortillas Trucks(49)
□ Delivery Trucks Type 1(61)
□ Delivery Trucks Type 2(15)
□ UPS Trucks(3)
□ US Trucks Type 1(17)
□ US Trucks Type 2(26)
□ Vistar VSA Trucks(44)
□ Other Trucks(527)

□ **Buses:**

□ UCSD Buses Type 2(1)
□ UCSD Buses Type 5(497)
□ UCSD Buses Type 6(1)
□ UCSD Buses Type 7(10)
☑ UCSD Buses Type 8(100)
□ UCSD Buses Type 9(1215)
□ Other Buses(44)

□ **Large Personal Vehicles:**

□ Vans(739)
□ SUVs(418)
□ Pickup Trucks(441)

□ **UCSD Police Vehicles:**

☑ Police Cars(488)
□ Police SUVs(103)

□ **Small Personal Vehicles:**

□ Cars(350)
□ Wagons(13)

□ **Unique Vehicles:**

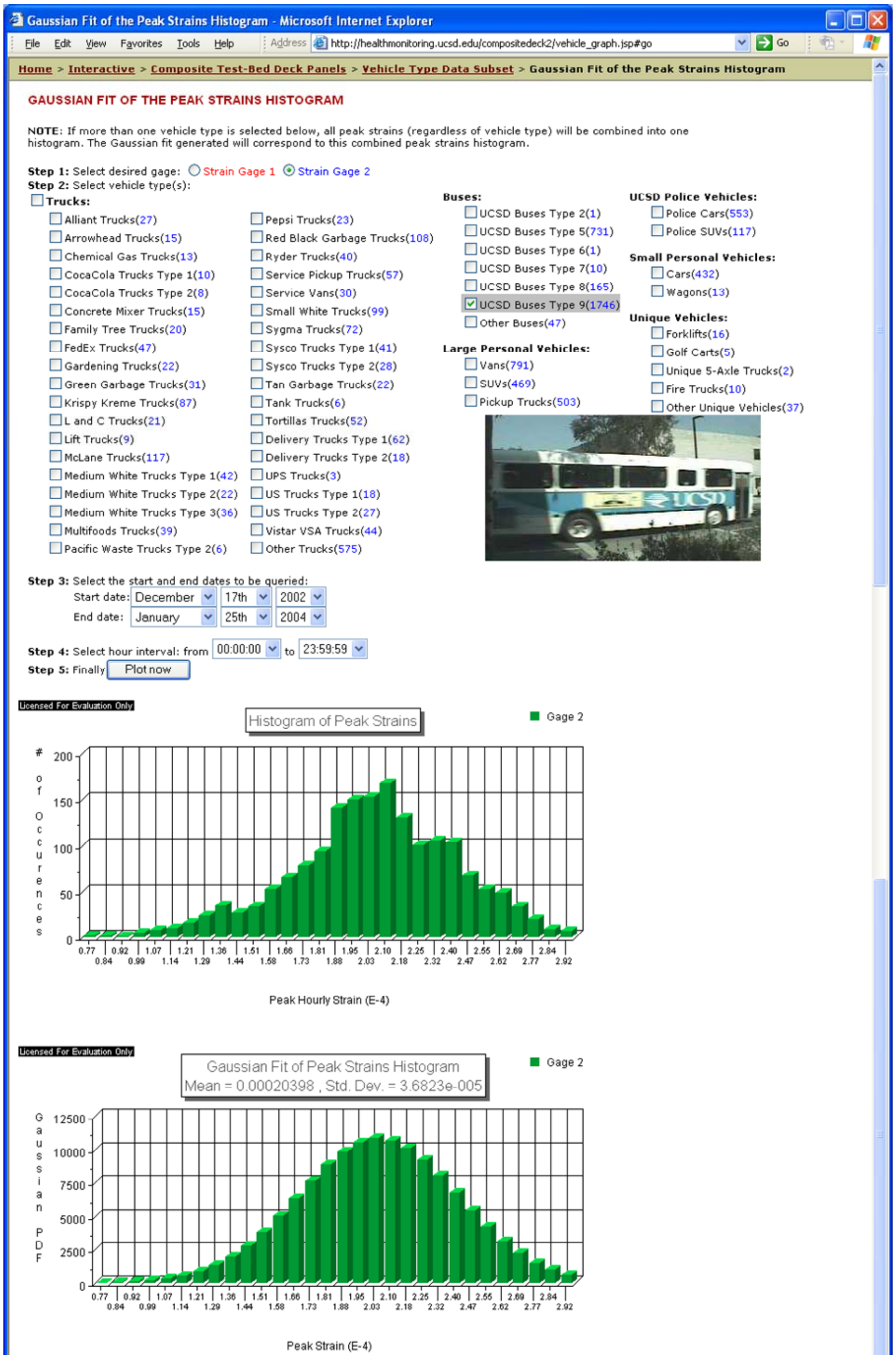
□ Forklifts(11)
□ Golf Carts(4)
□ Unique 5-Axle Trucks(2)
□ Fire Trucks(9)
□ Other Unique Vehicles(33)

Step 2: Select the start and end dates to be queried:
Start date: December 17th 2002
End date: January 25th 2004

Step 3: Select hour interval: from 00:00:00 to 23:59:59

Step 4: Finally Plot now

Figure 4.30: Scatter Plot of Channel 1 and 2 Peak Strains for Krispy Kreme Trucks, Red and Black Garbage Trucks, UCSD Buses Type 9 and Police Cars (Created in Collaboration with Minh Phan and Kendra Oliver)



Gaussian Fit of the Peak Strains Histogram - Microsoft Internet Explorer

Address http://healthmonitoring.ucsd.edu/compositedeck2/vehicle_graph.jsp#go

Home > Interactive > Composite Test-Bed Deck Panels > Vehicle Type Data Subset > Gaussian Fit of the Peak Strains Histogram

GAUSSIAN FIT OF THE PEAK STRAINS HISTOGRAM

NOTE: If more than one vehicle type is selected below, all peak strains (regardless of vehicle type) will be combined into one histogram. The Gaussian fit generated will correspond to this combined peak strains histogram.

Step 1: Select desired gage: ○ Strain Gage 1 ● Strain Gage 2

Step 2: Select vehicle type(s):

□ **Trucks:**

- □ Alliant Trucks(27)
- □ Arrowhead Trucks(15)
- □ Chemical Gas Trucks(13)
- □ CocaCola Trucks Type 1(10)
- □ CocaCola Trucks Type 2(8)
- □ Concrete Mixer Trucks(15)
- □ Family Tree Trucks(20)
- □ FedEx Trucks(47)
- □ Gardening Trucks(22)
- □ Green Garbage Trucks(31)
- □ Krispy Kreme Trucks(87)
- □ L and C Trucks(21)
- □ Lift Trucks(9)
- □ McLane Trucks(117)
- □ Medium White Trucks Type 1(42)
- □ Medium White Trucks Type 2(22)
- □ Medium White Trucks Type 3(36)
- □ Multifoods Trucks(39)
- □ Pacific Waste Trucks Type 2(6)
- □ Pepsi Trucks(23)
- □ Red Black Garbage Trucks(108)
- □ Ryder Trucks(40)
- □ Service Pickup Trucks(57)
- □ Service Vans(30)
- □ Small White Trucks(99)
- □ Sygma Trucks(72)
- □ Sysco Trucks Type 1(41)
- □ Sysco Trucks Type 2(28)
- □ Tan Garbage Trucks(22)
- □ Tank Trucks(6)
- □ Tortillas Trucks(52)
- □ Delivery Trucks Type 1(62)
- □ Delivery Trucks Type 2(18)
- □ UPS Trucks(3)
- □ US Trucks Type 1(18)
- □ US Trucks Type 2(27)
- □ Vistar VSA Trucks(44)
- □ Other Trucks(575)

Buses:

- □ UCSD Buses Type 2(1)
- □ UCSD Buses Type 5(731)
- □ UCSD Buses Type 6(1)
- □ UCSD Buses Type 7(10)
- □ UCSD Buses Type 8(165)
- ☑ UCSD Buses Type 9(1746)
- □ Other Buses(47)

Large Personal Vehicles:

- □ Vans(791)
- □ SUVs(469)
- □ Pickup Trucks(503)

UCSD Police Vehicles:

- □ Police Cars(553)
- □ Police SUVs(117)

Small Personal Vehicles:

- □ Cars(432)
- □ Wagons(13)

Unique Vehicles:

- □ Forklifts(16)
- □ Golf Carts(5)
- □ Unique 5-Axle Trucks(2)
- □ Fire Trucks(10)
- □ Other Unique Vehicles(37)

Step 3: Select the start and end dates to be queried:

Start date: December 17th 2002

End date: January 25th 2004

Step 4: Select hour interval: from 00:00:00 to 23:59:59

Step 5: Finally Plot now

Figure 4.31: Peak Strain Distribution of UCSD Buses Type 9 and Corresponding Gaussian Fit (Created in Collaboration with Minh Phan and Kendra Oliver)

4.7 Summary

The image sequences captured using the employed network camera on the composite bridge-deck monitoring system provide a large amount of data. However, to extract useful information (vehicle types, location...) from the raw video, the image sequences needed to be processed using image analysis. To determine the properties of the vehicles crossing over the bridge-decks, a background subtraction algorithm was used to separate moving objects (vehicles) from the stationary background. For each of the aforementioned strain time histories, image processing extracted the picture most clearly showing the vehicle crossing the bridge decks, thereby making it possible to sort and bin the time histories by vehicle type. This operation allowed for examining the distribution of vehicles crossing the decks, applicable for studying probability distributions of traffic loads and their effects (e.g., strains). Further, by sorting and binning the recorded traffic, a unique data set was established in which the measured bridge response and images corresponding to the loads (passing traffic) are both available. This data is extremely useful for applying artificial intelligence algorithms for vehicle classification and property estimation as well as for monitoring for changes in the bridge system response over time.

5 Numerical Simulation of Bridge Deck System

To simulate the response of the bridge decks under vehicular loading, a one-dimensional finite element model was constructed and is discussed in this chapter. This model was created to generate data otherwise unavailable with the real system (e.g., high-speed traffic and data from damaged states). In this numerical environment, input is obviously fully defined (speed, wheelbase, axle weights…). Consequently, insights can be gained as to the potential of machine learning algorithms in extracting knowledge from the actual recorded data sets.

To simulate the response of the bridge deck system, particularly under traffic loading, a one-dimensional finite element model was analyzed using the computational framework OpenSees (http://opensees.berkeley.edu/). In a future full-scale health monitoring system, it is envisioned that OpenSees [McKenna and Fenves, 2000] will constitute the required computational engine for mechanics-based modeling and analysis of bridge systems. OpenSees (Open System for Earthquake Engineering Simulation) is an open source software framework to simulate the response of structural and geotechnical systems to earthquake and dynamic loads in general. The object-oriented framework of OpenSees allows the structural response simulation to be factorized into independent classes such as model building, finite elements, constitutive material models, boundary conditions and constraints, solution strategies, equation solvers, time integration algorithms, and recorders emulating sensors. In addition, OpenSees supports a wide range of simulation models, solution procedures, and distributed computing models

[McKenna and Fenves, 2000]. It also has very attractive capabilities for automatic parameterization of a structural model, probabilistic modeling, response sensitivity analysis and reliability analysis.

With each finite element model, it was assumed the four bridge decks (three fiber-reinforced polymer and one reinforced concrete) could be modeled using equivalent homogeneous materials. The properties of the equivalent sections (Young's Modulus and mass density) were determined from static (dead load) [Zhao, 1999] and dynamic free vibration testing.

5.1 Analytical Simulation of Bridge Deck System Using One-Dimensional Finite Element Model

The one-dimensional finite element model shown in Figure 5.1 is composed of 61 nodes and 60 elastic beam-column elements, each three-inches long. For the beam-column elements, it is necessary to enter values for cross sectional area, Young's modulus, moment of inertia, and assign mass values to nodal degrees of freedom [Mazzoni et al., 2005]. To simulate the boundary conditions of the bridg*e* decks, which are connected to the sill through shear keys*,* simple support boundary conditions were assumed, with a 6 inch overhang beyond the supports.

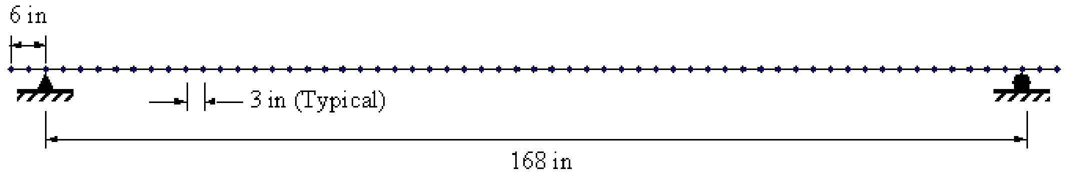


Figure 5.1: Simplified One-Dimensional Beam-Column Finite Element Model of Bridge Decks

5.2 Model Calibration

The finite element model was calibrated from recorded static and dynamic test data. During the static test, dead loads were applied to each of the individual decks and deflections were measured at various locations along the length of the deck. From the load-defection data, the corresponding Young's Modulus of a simplified simply supported beam was calculated. To determine the mass of the bridge decks, an impulse was applied to generate a free vibration response, which was recorded using accelerometers installed on the surface of the decks. From the recorded acceleration time histories and their associated Fast Fourier Transforms, the fundamental frequency of the bridge deck system was determined and applied for calculating the unknown mass.

5.2.1 Static Dead Load Test

During static testing [Zhao, 1999], deck HD was placed in a simply supported beam configuration and loads were applied along the midspan using hydraulic actuators as shown in Fig. 5.2. Under this loading configuration, a 3.30 in deflection was observed at midspan, corresponding to a 221 kip load. These values were used with the simplified simply supported beam system shown in Fig. 5.3 to determine the unknown elastic modulus of deck HD.

.

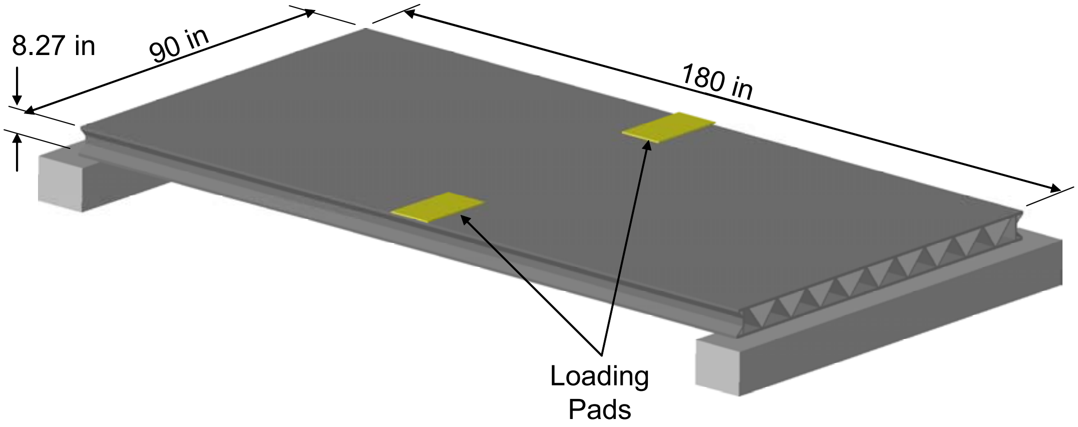


Figure 5.2: Static Loading Configuration for Composite Bridge Deck HD

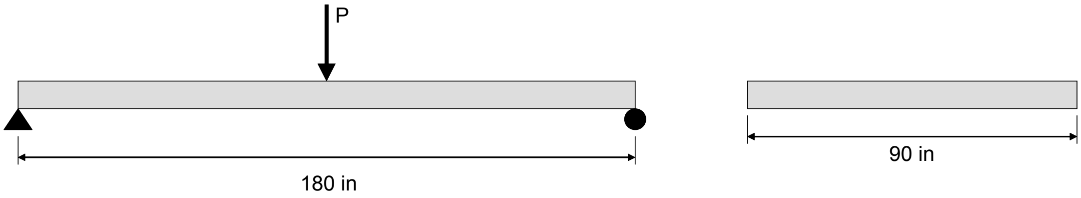


Figure 5.3: Simplified Simply Supported Beam

From simplified geometry:

$$I = \frac{bh^3}{12} = \frac{(90\,in)(8.27\,in)^3}{12} = 4242.07\,in^4 \qquad \textbf{Eq. 5.1}$$

From Static Load Test:

$$\Delta = \frac{PL^3}{48EI} = \frac{221\,kips(180\,in)^3}{48EI} = 3.3\,in. \qquad \textbf{Eq. 5.2}$$

$$(EI)_{HD} = \frac{PL^3}{48\Delta} = \frac{221\,kips(180\,in)^3}{48(3.3\,in)} = 8.1368\times10^6\ kips\cdot in^2 \qquad \textbf{Eq. 5.3}$$

$$E_{HD} = \frac{EI}{I} = \frac{8.1368\times10^6\ kips\cdot in^2}{4242.07\,in^4} = 1918.12\frac{kips}{in^2} \qquad \textbf{Eq. 5.4}$$

To determine the stiffness of the bridge deck system, the above Young's Modulus was used with a uniform rectangular beam measuring 180 inches in length, 360 inches wide, and 8.27 inches high. This yielded an equivalent stiffness (EI) of $3.255\times10^7\,Kips\cdot in^2$.

5.2.2 Dynamic Free Vibration Test

A series of dynamic tests were then carried out; whereby, a 15 pound hammer (with an internal load cell) was used to apply a vertical impact at the center of the bridge decks. A Crossbow CXL01LF1 capacitive accelerometer (±1 g, 0.56 mg resolution, 0-50 Hz frequency range) was installed on deck 2 near the impact location (Fig. 5.4) and measured the vertical vibration during the free vibration response following each impact (Fig. 5.5). The accelerometer was bolted to an aluminum anchor plate and attached to the surface of the roadway using a semi-permanent tar adhesive. As the accelerometer was on top of the bridge decks it could only be installed while no vehicles were present. This meant when a lull in traffic was observed, the accelerometer and anchor plate were placed on the decks and data was acquired. When a vehicle approached, the accelerometer and plate then had to be hurriedly removed. A Crossbow AD2012 data logger was used to power the sensor and to perform the analog-to-digital conversion. Using this datalogger, a series of twenty second long time histories (acquired with a 200 Hz sampling rate) were recorded. A sample of one of these time histories is included in Figure 5.5. Within this time history, there are a series of three small impacts in the first 2 seconds, followed by three larger impacts over the next 2 seconds, and then one last medium impact. For each hammer hit, it can be seen the motion has returned to zero before the next impact was applied. A close-up of the first large impact is shown in Figure 5.6. Fast Fourier Transforms (FFT) of the recorded acceleration time histories were analyzed and the fundamental frequency for the unloaded bridge deck system was determined to be 22.4 Hz (Fig. 5.7).

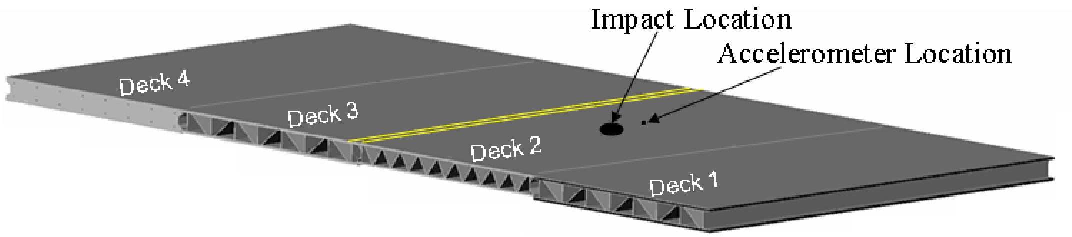


Figure 5.4: Locations of Impact and Accelerometer on Composite Bridge Decks

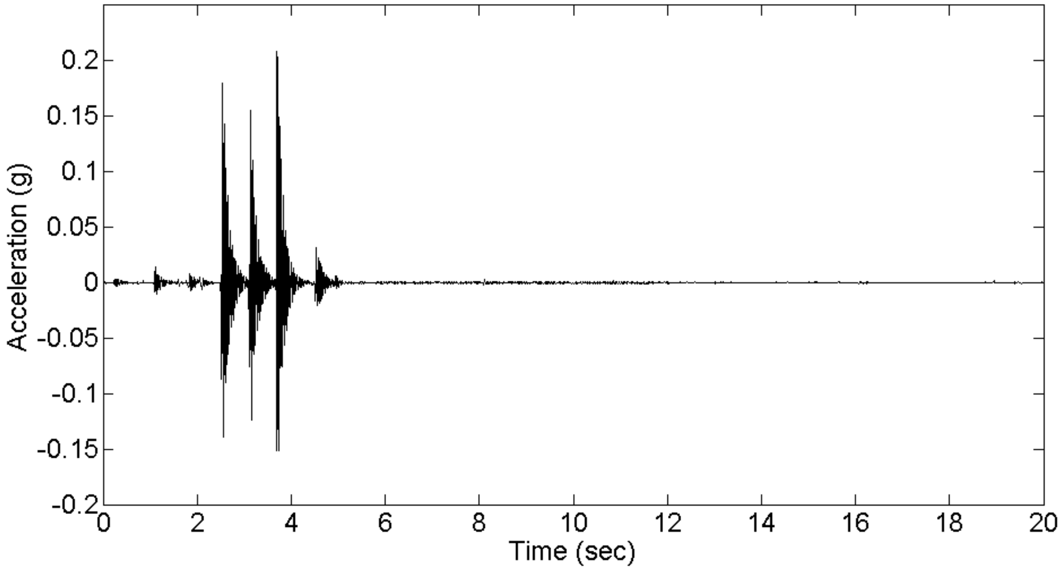


Figure 5.5: Vertical Acceleration Time History for Hammer Impacts Applied at Midspan.

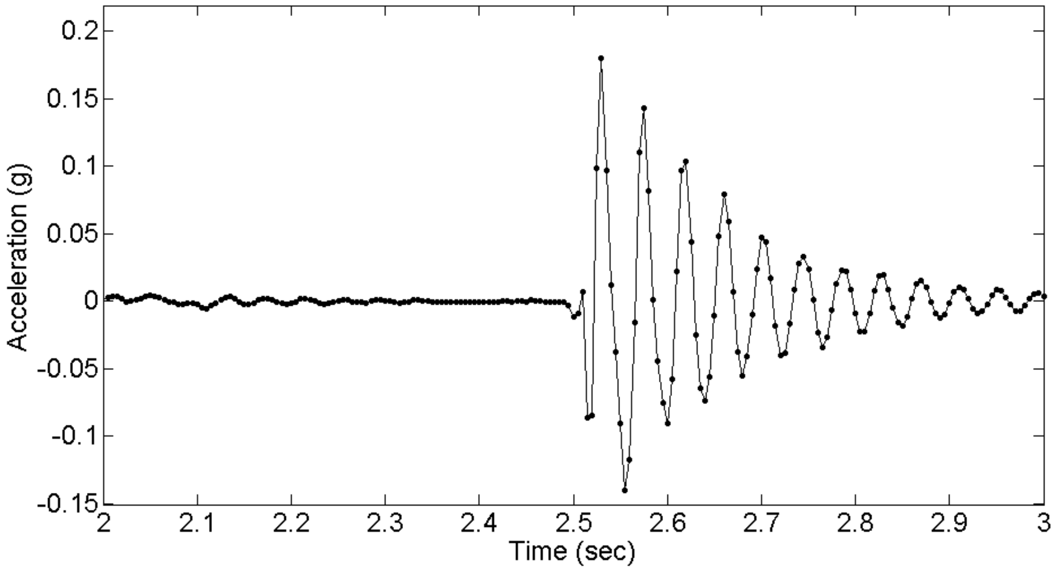


Figure 5.6: Close-up of Acceleration Time History Detailing the Impact and Free Vibration Phases.

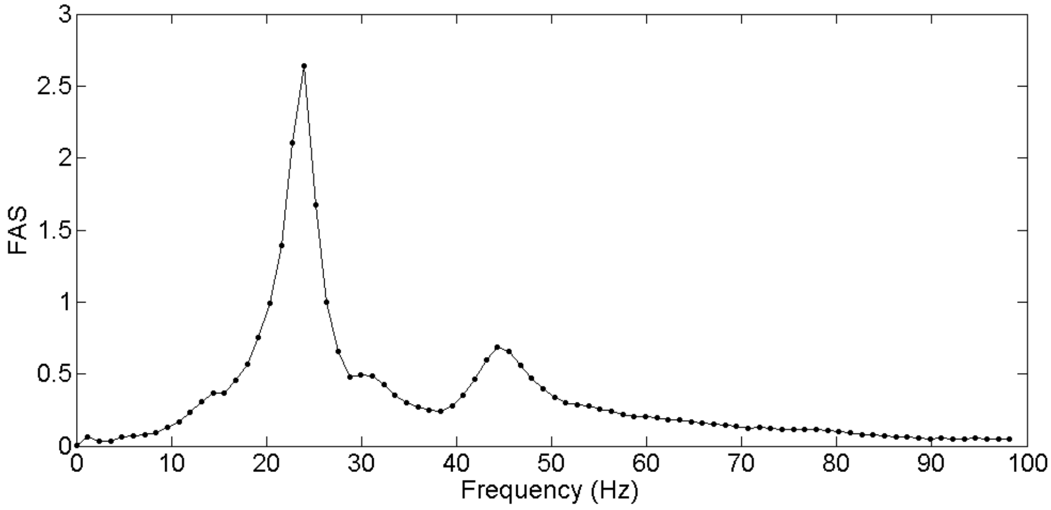


Figure 5.7: Fast Fourier Transform of Recorded Free Vibration Acceleration Time History

By treating the composite decks as a simply supported beam, with the previously determined equivalent stiffness, the mass and density of the equivalent deck system were determined by solving for the mass in equation 5.7 [Chopra, 2000] that yields the same fundamental frequency as was measured (f_1 = 22.4 Hz). For a simply supported beam, each natural frequency may be calculated by,

$$\omega_i = \frac{n^2\pi^2}{L^2}\sqrt{\frac{EI}{M}} \qquad \textbf{Eq. 5.5}$$

or in hertz,

$$f_i = \frac{n^2\pi}{2L^2}\sqrt{\frac{EI}{M}} \qquad \textbf{Eq. 5.6}$$

For the first mode, n = 1:

$$f_1 = \frac{\pi^2}{2L^2}\sqrt{\frac{EI}{M}} \qquad \textbf{Eq. 5.7}$$

Rearranging and solving for the unknown mass per unit length, M, yields:

$$M = \frac{\pi^2 EI}{4 f_1^2 L^4} \qquad \textbf{Eq. 5.8}$$

Substituting in the Young's modulus determined from the static test data with the cross sectional area and moment of inertia determined from the deck geometry, the mass, weight, density, and specific weight were determined. Finally, by calculating the logarithmic decrement during the free vibration following one of the hammer impacts (Fig. 5.8), an estimate for damping was determined (Table 5.1). From Table 5.1, it is evident the damping estimates vary depending on which cycles are examined. While the largest damping ratio is observed under the largest amplitudes which would seem to indicate an amplitude dependent damping, it is felt that the variation in damping ratio should attributed to errors in the calculation based on the sampling rate used for acquiring the acceleration time histories. Had faster sampling been used, the estimated values for this test would have been more consistent. For modeling the system, an average value of 5%, similar to the value found using several seconds of free vibration, was used. It was also assumed this damping ratio was the same for all modes. These assumptions allowed for determining equivalent Rayleigh damping. Equations 5.9 and 5.10 are for the coefficients used to build the Rayleigh damping matrix. The properties of the equivalent simply supported beam are summarized in Table 5.2.

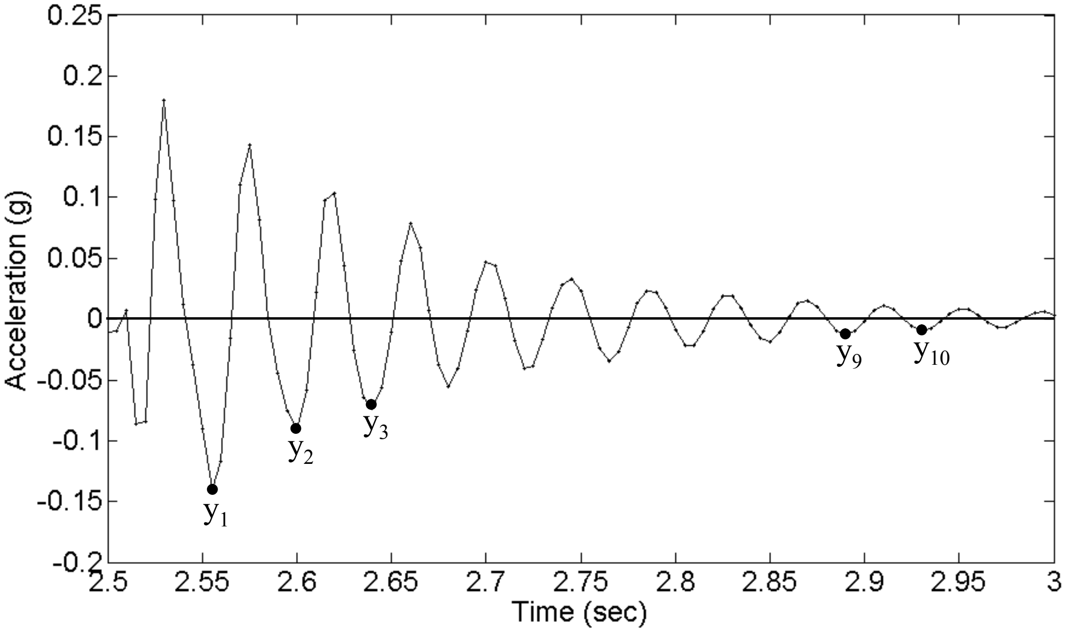


Figure 5.8: Free Vibration Time History Used for Calculating Damping

Table 5.1: Logarithmic Decrements and Damping Ratios Estimated from Free Vibration

Cycle Number	Logarithmic Decrement, δ	Damping Ratio, ζ
y_1 to y_2	0.4361	6.94%
y_2 to y_3	0.2088	3.32%
y_9 to y_{10}	0.2803	4.46%
y_1 to y_{10}	0.3001	4.78%

$$a_0 = \xi \frac{2\omega_i \omega_j}{\omega_i + \omega_j}$$ **Eq. 5.9**

$$a_1 = \xi \frac{2}{\omega_i + \omega_j}$$ **Eq. 5.10**

Table 5.2: Properties of Equivalent Simply Supported Beam

Mass, m	$0.02744 \frac{kips \cdot \sec^2}{in}$
Weight, W	$10.605\, kips$
Density, ρ	$5.1204 \times 10^{-8} \frac{kips}{in^3} \frac{\sec^2}{in}$
Specific Weight, γ	$1.9785 \times 10^{-5} \frac{kips}{in^3}$
Young's Modulus, E	$1918.12 \frac{kips}{in^2}$
Moment of Inertia about Strong Axis, I	$1.697 \text{ x } 10^4 in^4$
Rayleigh Coefficient, a_o	11.3097
Rayleigh Coefficient, a_1	0.000141

These mass properties along with the previously determined elastic modulus were utilized to construct the finite element model shown in Figure 5.1. This model, composed of 61 nodes and 60 elastic beam column elements, was analyzed using the computational framework OpenSees (http://opensees.berkeley.edu/). For each node, a nodal mass corresponding to each degree-of-freedom had to be assigned. For this study, nodal masses were assigned using a tributary area based on neighboring elements. Next, for the elastic beam column elements, the cross sectional area of the element, Young's Modulus, and moment of inertia were assigned based on the previous calculations. By using 60 identical elements, the corresponding 3 inch element length proved suitable for traffic modeling and damage analysis.

An impulse load was applied to the middle node of the finite element model to generate a free vibration response (Fig's. 5.8 and 5.9), which when analyzed in the frequency domain (Fig's. 5.10 and 5.11), confirmed the equivalent finite element beam had the same fundamental natural frequency as the actual bridge deck system.

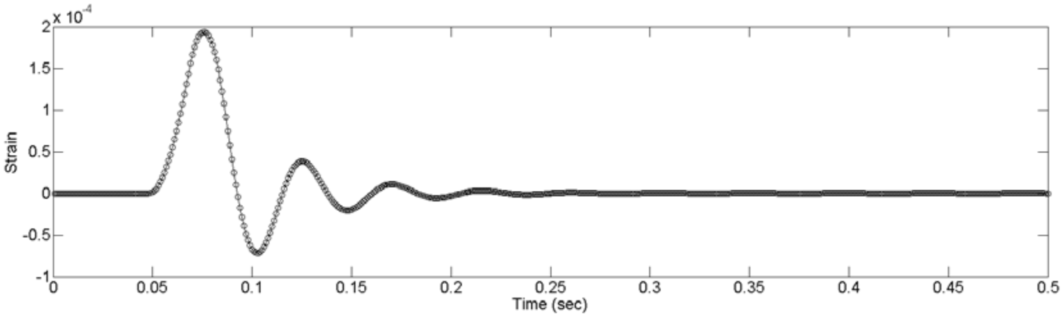


Figure 5.9: Quarter Span Free Vibration Strain Time History.

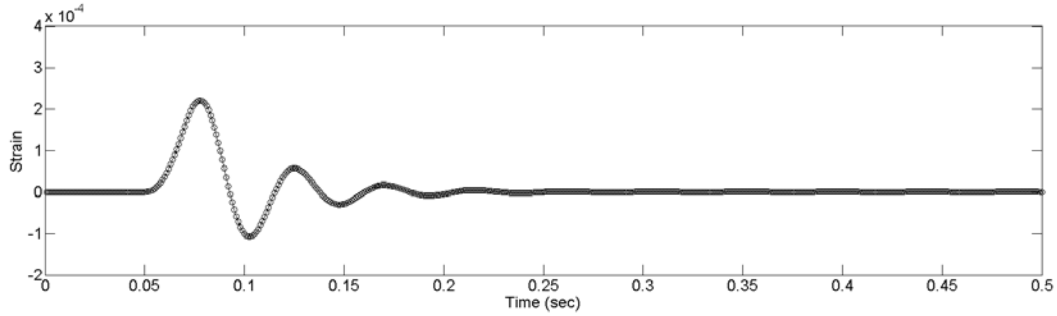


Figure 5.10: Midspan Free Vibration Strain Time History.

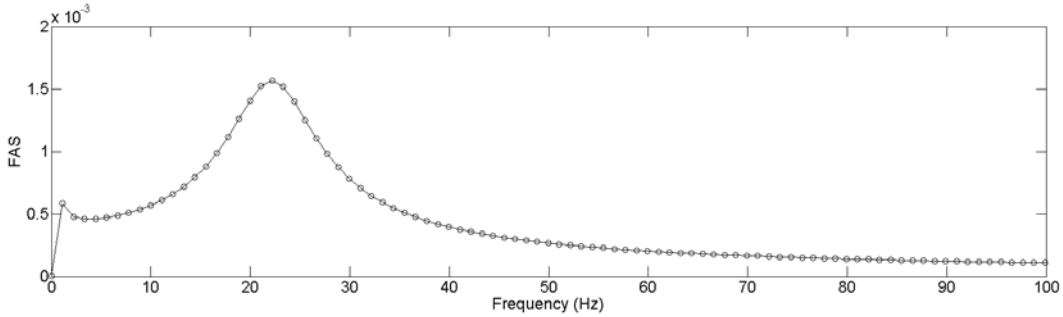


Figure 5.11: FFT of Quarter Span Free Vibration Strain Time History.

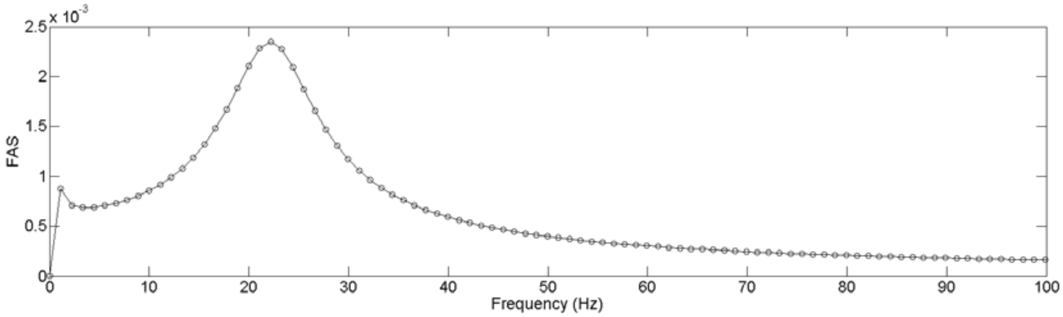
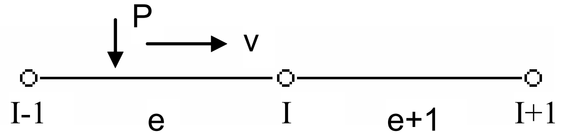


Figure 5.12: FFT of Midspan Free Vibration Strain Time History.

5.3 Traffic Modeling

To simulate the bridge-decks' response to traffic crossing over, the one-dimensional finite element model (Fig. 5.1) was again analyzed using OpenSees. In this study, a single vehicle passing over the bridge-decks is simulated by ignoring the interaction between the beam and vehicle, instead considering only the moving forces [Foda and Abduljabbar, 1998 and Chen and Feng, 2006]. To simplify the simulation, the vehicles are considered to be the moving forces only, in other words, the interaction between the beam and vehicles is ignored. Since the loads can only be prescribed at the nodes in the finite element model, the time history of moving load at each node must be defined according to the vehicle velocity. For a typical finite element mesh (Fig. 5.12), the time history is usually assumed such that the load arrives at node I at time t_I and keeps a constant value until it arrives at the next node. This type of time history is easily implemented in the finite element model, however, the size of each element must be small and the number of elements is controlled by the slowest velocity of the moving load and the period of the beam [Saadeghvaziri, 1993]. To improve the accuracy of the results, the time history expressed in equations 5.11 and 5.12 is used to represent the moving load. The time history includes two parts: the force and the moment, which can be derived by the equivalent nodal load definition using the mechanics of structures. Node I first experiences loading at time t_{i-1}. This load continues to increase, reaching a maximum value at time t_I after which it decreases to zero loading at time t_{I+1} (Fig. 5.13).



I : Node number
e : Element number
P : Moving load
v : Velocity of moving load

Fig. 5.13: Typical Finite Element Model

$$f_{force}{}^{I}(t)=\begin{cases}\dfrac{-2(t-t_{I-1})^{3}+3t_{p}(t-t_{I-1})^{2}}{t_{p}^{3}}P & t_{I-1}\le t<t_{I}\\ \dfrac{2(t-t_{I})^{3}-3t_{p}(t-t_{I})^{2}+t_{p}^{3}}{t_{p}^{3}}P & t_{I}\le t<t_{I+1}\\ 0 & others\end{cases} \qquad \textbf{Eq. 5.11}$$

$$f_{moment}{}^{I}(t)=\begin{cases}\dfrac{-(t-t_{I-1})^{3}+t_{p}(t-t_{I-1})^{2}}{t_{p}^{2}}vP & t_{I-1}\le t<t_{I}\\ \dfrac{-(t-t_{I})^{3}+2t_{p}(t-t_{I})^{2}-t_{p}^{2}(t-t_{I})}{t_{p}^{2}}vP & t_{I}\le t<t_{I+1}\\ 0 & others\end{cases} \qquad \textbf{Eq. 5.12}$$

In equations 5.11 and 5.12, t_I is the time corresponding to the load arriving at node I and t_p is the time required for the load to traverse one element:

$$t_{p}=t_{I}-t_{I-1}=t_{I+1}-t_{I}=\frac{\Delta L}{v} \qquad \textbf{Eq. 5.13}$$

, where ΔL is the element length and v is the vehicle's speed. It should be noted in equation 5.13, all element lengths are assumed to be the same.

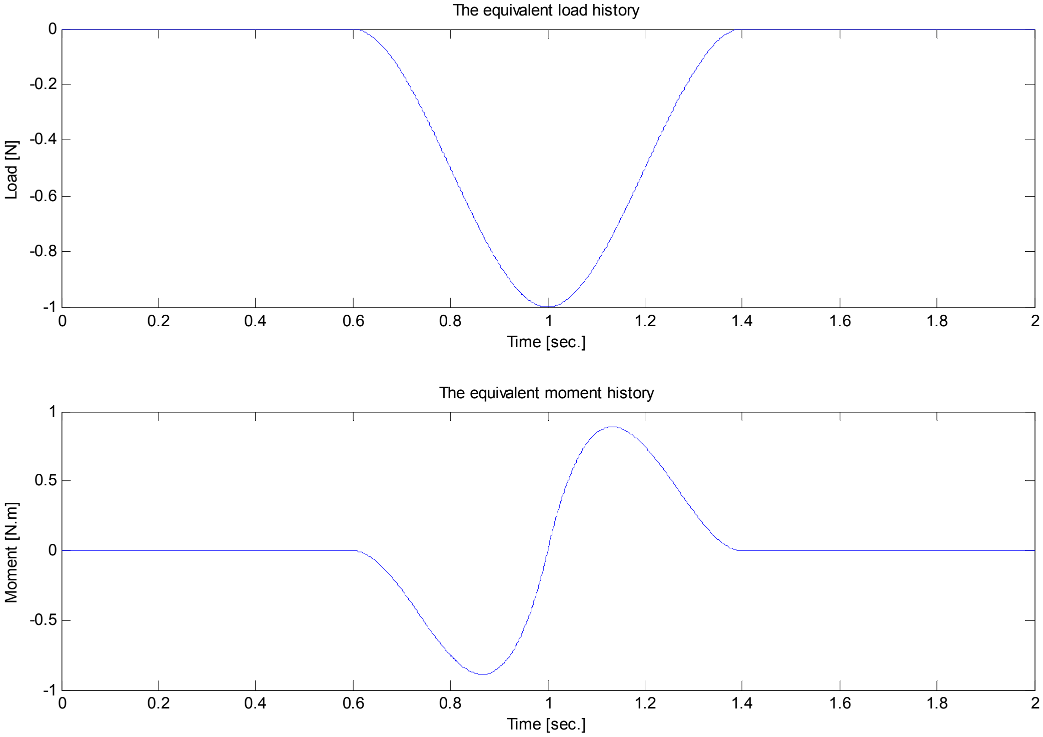


Figure 5.14: Equivalent Load and Moment Time Histories for Single Axle Moving Load, where v=15 m/s (33.5 MPH), t_p=0.4 sec., t_1=1.0 sec, and P=1 N.

Simulating a vehicle passing over the bridge decks thereby involves generating 122 individual time history files, two for each node. Each of these 4-second long input force time history files is 100 kBytes (4000 time steps with a 0.001 second time increment); therefore, each simulation requires generating 12 Mbytes of input files. For running large numbers of traffic and damage scenarios, Matlab programs were written to generate the necessary finite element tcl scripts and input motion files and batch commands were used to run all of the tcl scripts in OpenSees. This process required entering specific values (which later were chosen at random) for the vehicles speed and wheelbase. For all cases, a unit force was assigned to all the axle weights. Within the model's tcl script, scaling factors were then used to produce the desired axle weights.

The linear analysis was conducted using Newmark's method for the average acceleration case with a modified Newton algorithm. The Modified Newton Algorithm uses the modified Newton-Raphson method to advance to the next time step. The difference between this method and the Newton-Raphson method is that the tangent stiffness is not updated at each step, thus avoiding expensive calculations needed in multi-DOF systems; however, for a linear analysis this is not important. In all of the finite element modeling, a time step of 0.001 seconds was used.

Rather than recording the output displacements and forces at all of the nodes and elements in the mesh, it was decided to focus on specific elements at key locations. For initial studies, the elements at one-quarter and middle span were used (Fig. 7.11) and later sixteen elements evenly spaced along the length of the deck were used (Fig. 7.12). For each of these recording elements, the bending moments (M) were output as individual text files. As a post processing operation, these files were loaded into Matlab and strains (ε) were determined in accordance with Hooke's Law, equation 5.14 [Popov, 1976].

$$\varepsilon = \frac{\sigma}{E} = \frac{Mc}{EI} = \frac{M}{EI} \cdot \frac{h}{2}$$ **Eq. 5.14**

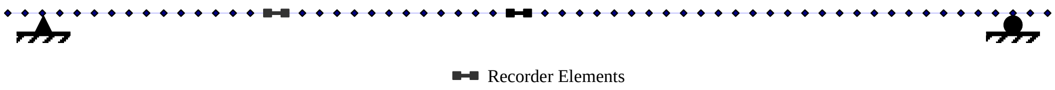


Figure 5.15: One-Dimensional Finite Element Model with 2 Recording Elements

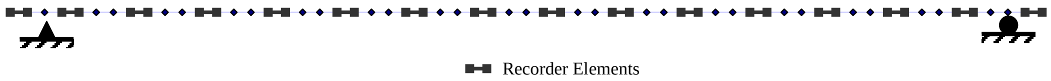
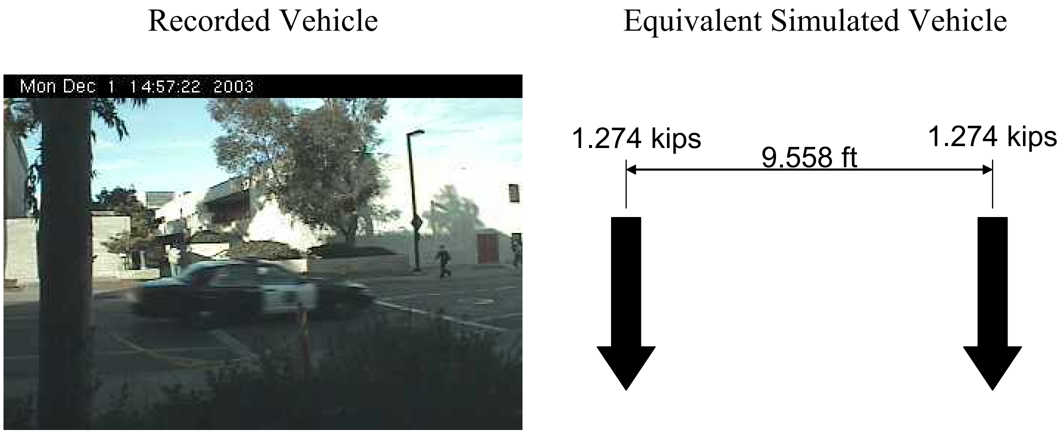


Figure 5.16: One-Dimensional Finite Element Model with 16 Recording Elements

To test the accuracy of the constructed FE model, a comparison was performed between a recorded campus police car and an equivalent simulated vehicle (Fig. 5.16). A campus police car was selected as the wheelbase and approximate axle weights of the modified Ford Crown Victoria were known. In this simulation, a speed of 26 MPH was used along with axle weights of 1.274 kips front and back. The recorded and simulated strains are shown in Figure 5.17. From this figure, it is seen the model does a good job predicting the peak strains caused by each axle. Also, the duration of loading, from when the front axle first touches the decks to when the rear axle passes, is also consistent. Problems are associated with unloading between the two axles. In the recorded data, the strains reach a much lower level in between the two peaks than in the simulated data. Another difference is in the width of each peak as the simulated peaks are wider than the recorded ones. In general, the results were deemed suitable for an initial study. While the FE model does not provide an exact representation of the actual composite bridge-deck system, it is close enough to provide insight into the response and behavior of the actual structure.


Recorded Vehicle

Equivalent Simulated Vehicle

Figure 5.17: Recorded Police Car and Equivalent Simulated Vehicle

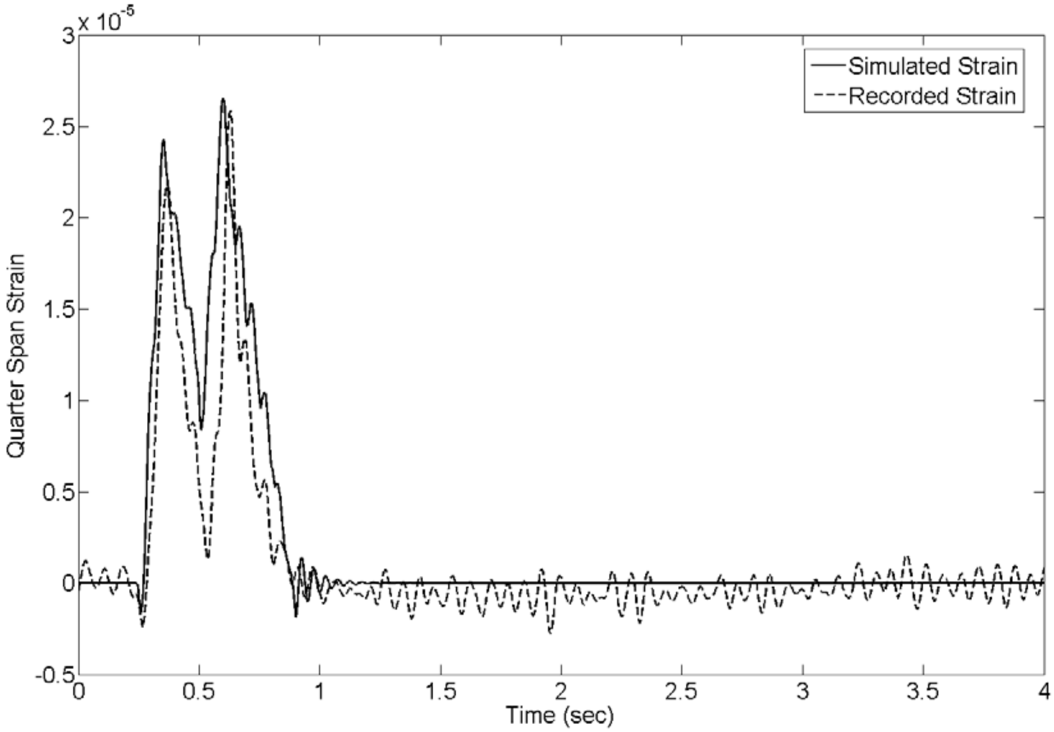


Figure 5.18: Recorded and Simulated Strain Time Histories for Campus Police Car

To illustrate the change in strain response for different vehicle types, three sets of midpsan strain time histories are included in Figures 5.18-5.20. For this example, three distinctly different vehicle types were considered, and the properties of these vehicles (Car, 2-Axle Truck, and 2-Axle Bus) are summarized in Table 5.3. Each figure contains the response for a different speed (10, 20, and 30 MPH respectively). Within the time histories, there are several features worth noting:

- Each distinct peak corresponds to an axle passing over the recording element.

- As the speed of the vehicle increases, the width of the response decreases and the distance between the peaks, corresponding to each of the wheels passing over the recording element, also decreases.
- In this example, as the speed increases the peak strains remain fairly constant.
- For vehicles with wheelbases less than 14 feet, the first axle does not move off the decks before the second moves on, and consequently the strains do not return to zero in between the two peak axle strains.

While capable of generating strain time histories similar to the ones measured on the bridge decks, the finite element model does not, nor was it ever intended to, provide an exact replication of the measured data. Instead, it is only meant to produce comparable data which may be used to test the various analysis algorithms.

Table 5.3 Properties for 3 Typical Simulated Vehicles

Vehicle Type	**Wheelbase (ft)**	**Front Axle Weight (Kips)**	**Rear Axle Weight (Kips)**	**Similar Observed Traffic**
Car	9	2	2	
Garbage Truck	13	9	19.5	
Bus	18	11	25	

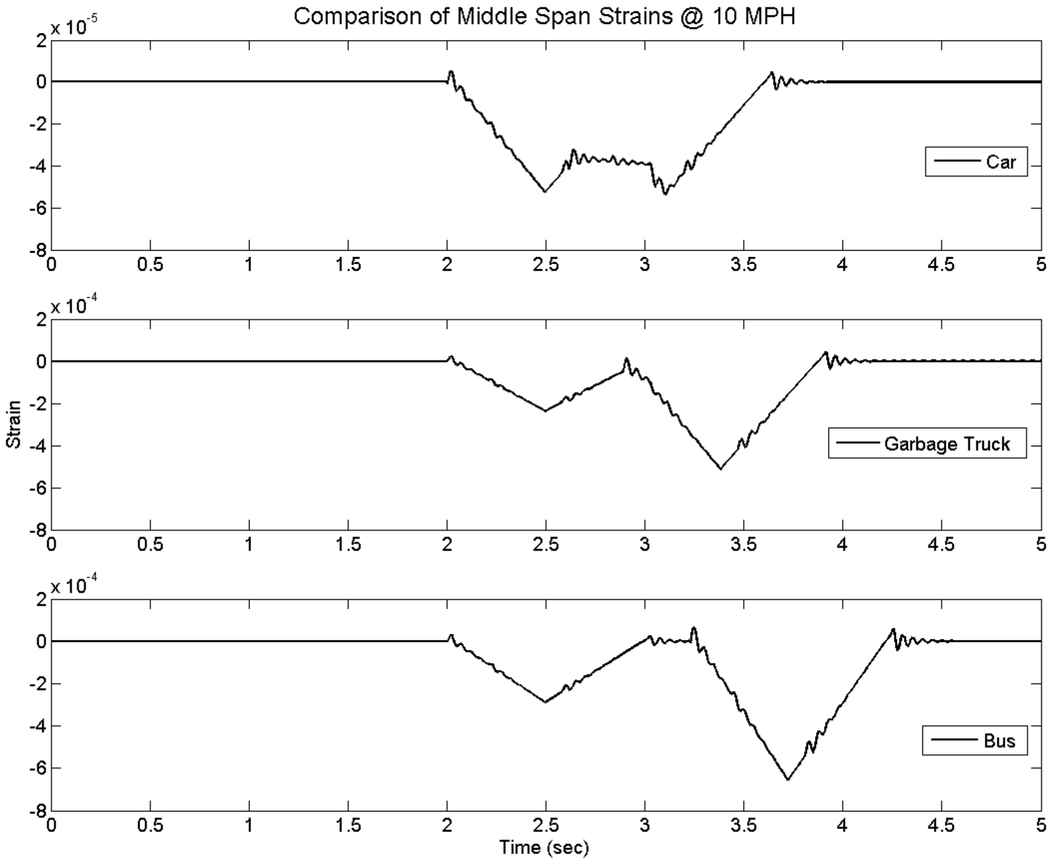


Figure 5.19: Comparison of Simulated Midspan Strains (Element 30) for Car, 2-Axle Truck, and 2-Axle Bus Crossing at 10 MPH

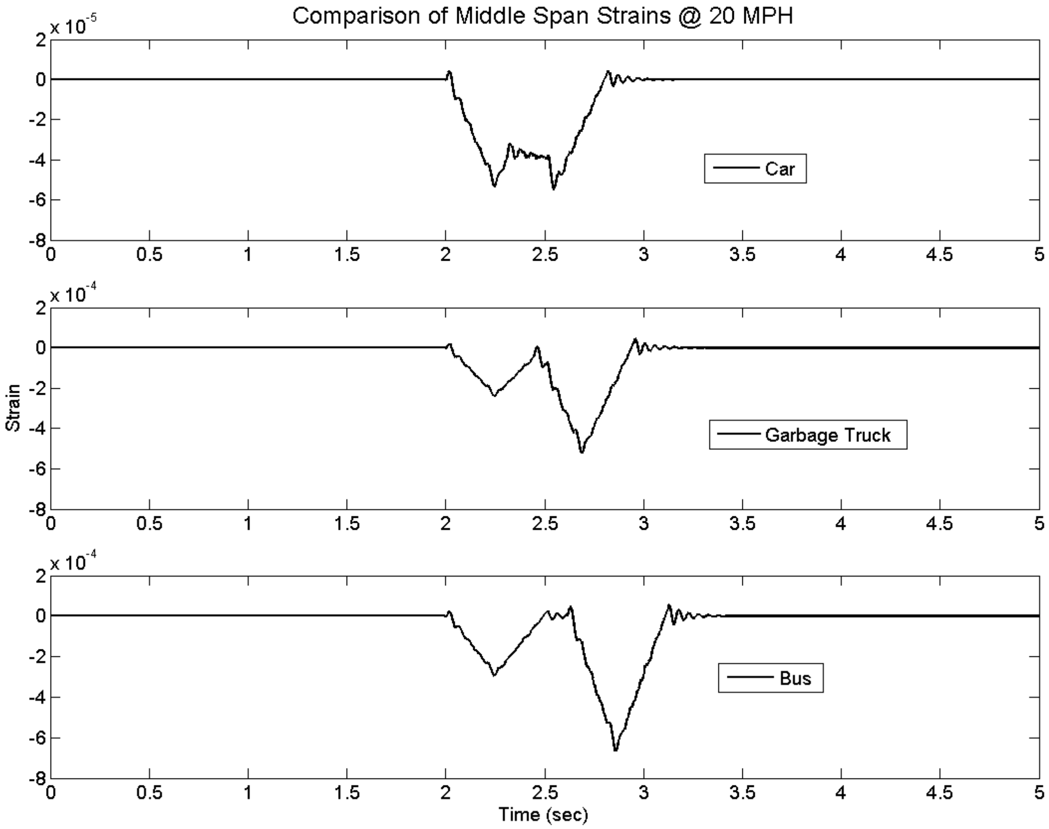


Figure 5.20: Comparison of Simulated Midspan Strains (Element 30) for Car, 2-Axle Truck, and 2-Axle Bus Crossing at 20 MPH

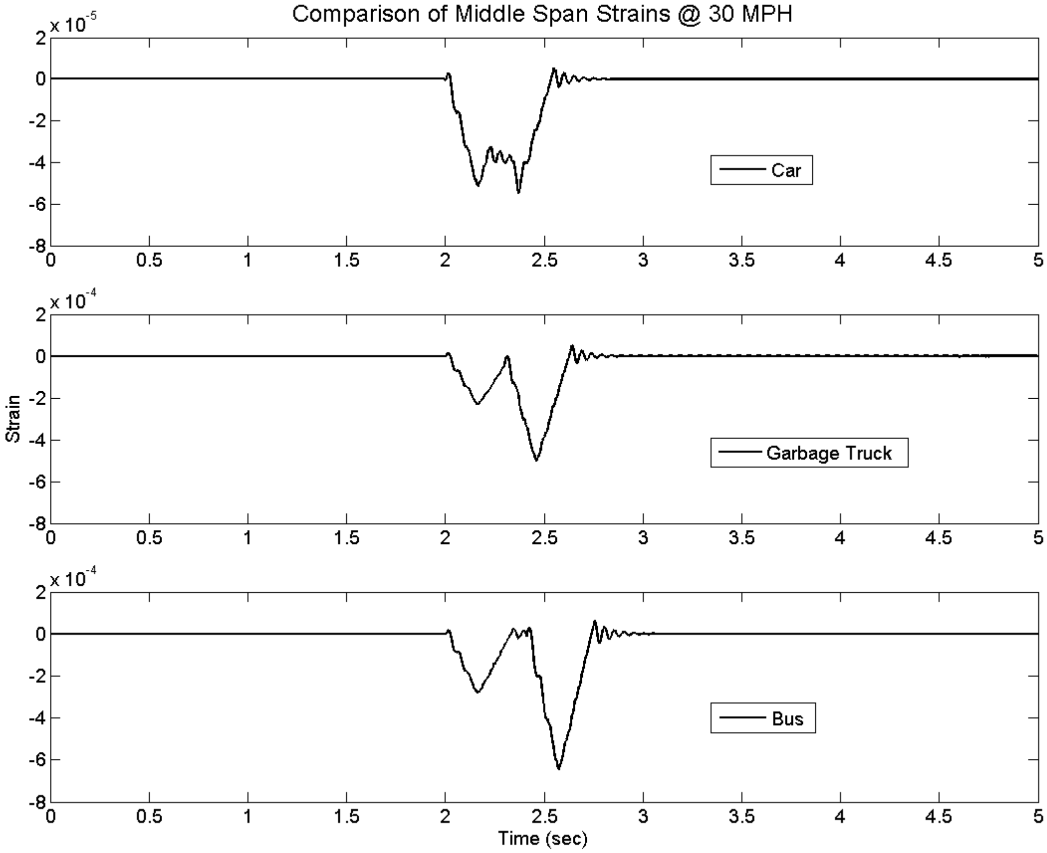


Figure 5.21: Comparison of Simulated Midspan Strains (Element 30) for Car, 2-Axle Truck, and 2-Axle Bus Crossing at 30 MPH

5.4 Summary

The application of supervised learning techniques for vehicle property estimation requires information related to vehicle speed, wheelbase, and axle weights. As these pieces of information are not available in the strain data recorded on the composite bridge decks, training neural networks based solely on the recorded strain data was impossible. Similar problems exist in training neural networks to detect and classify damage in the bridge decks where vehicle induced strains taken from the damaged state of the bridge decks is unavailable, and damaging the decks was not a possibility. Consequently, a computational model of the bridge decks had to be created to provide strain time histories similar to those recorded on the actual system. Good results were obtained using a one-dimensional finite element model in which the interaction between the structure and vehicle were ignored (instead traffic was modeled as a series of moving loads). The generation of the finite element input and load files was implemented into Matlab which allowed for the efficient generation and execution of the large numbers of runs (thousands) necessary for the conducted parametric studies and neural network training.

6 Vehicle Property Estimation and Damage Detection

Within this chapter, strain time histories generated by the one-dimensional finite element model subject to simulated traffic loads were analyzed to garner information regarding the vehicles properties and for damage detection. For estimating the speeds and wheelbases of the simulated traffic (useful properties for classifying traffic by vehicle type) strain time histories were analyzed for a data set consisting of a single vehicle type (constant wheelbase and axle weights) and for random traffic. Next, the same finite element model was used to model traffic crossing the bridge-deck system in the original (undamaged) and damaged states. In this exercise, damage was simulated by reducing Young's Modulus (stiffness) for one or more elements. By examining changes in the computed peak strains, it was possible to locate damage and to make approximations as to the level of the reduction in stiffness.

6.1 Vehicle Identification and Property Estimation

6.1.1 Traffic Identification for Typical Bus

The first data set used for traffic identification relates to a single vehicle type passing over the bridge decks at various speeds. For these scenarios, a wheelbase of 168 in, front axle weight of 11 kips, and rear axle weight of 25 kips, similar to a fully-loaded 2-axle bus, were used. 631 individual scenarios were numerically simulated (wheelbase and axle weights were held constant and the speed varied from 2 miles-per-hour to 65 miles-per-hour in 0.1 MPH increments).

The first vehicle parameter to be estimated was speed. This required outputting the quarter span and midspan bending moments and calculating the associated strains. To calculate the speed, it was necessary to determine the instant in time when the front axle passes over each of the recording elements. These values correspond to the two predominant positive peaks in the strain time histories. To locate these local maxima, a Matlab program which searches for and returns all local maxima and minima in a vector was implemented [Koptenko, 2003]. Maxima with amplitudes lower than the set threshold of 5x10^{-5} were ignored, as these discarded maxima are predominantly found in the free vibration phase.

The results of this program run on the original strain time histories from the 2-axle bus moving at 16 MPH are shown in Figure 6.1. In this figure, the solid line is the quarter span strain with a solid circle denoting the first maxima and squares subsequent maxima. Similarly, the dashed line is the midspan strain with an “x” for the first maxima and asterisk for remaining maxima. Ideally, only one maxima per axle will be identified; however, other maxima associated with the bridge deck system vibrating about its natural frequency during the loading/unloading phase are also present. To remove these unwanted peaks, a 5th order lowpass Butterworth digital filter was applied to the strain data prior to locating maxima (Fig. 6.2). Using a lowpass filter with a cutoff frequency less than the bridge decks natural frequency sufficiently smoothed the data thereby greatly improved the results. As shown in Figure 6.2 for the previous time histories filtered using a 15 Hz cutoff frequency, only two peaks (corresponding to the two axles) were identified for each time history.

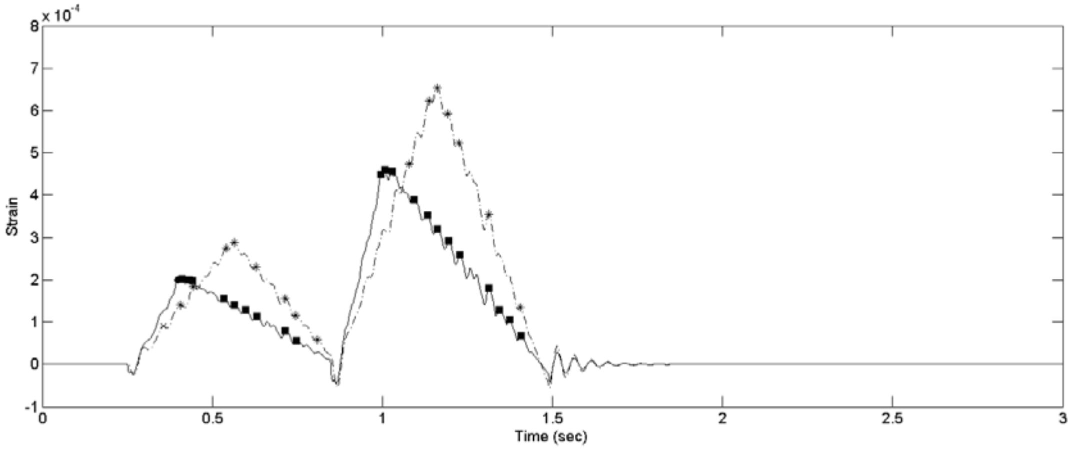


Figure 6.1: Unfiltered Midspan and Quarter Span Strain Time Histories for Simulated Bus Crossing at 16 MPH.

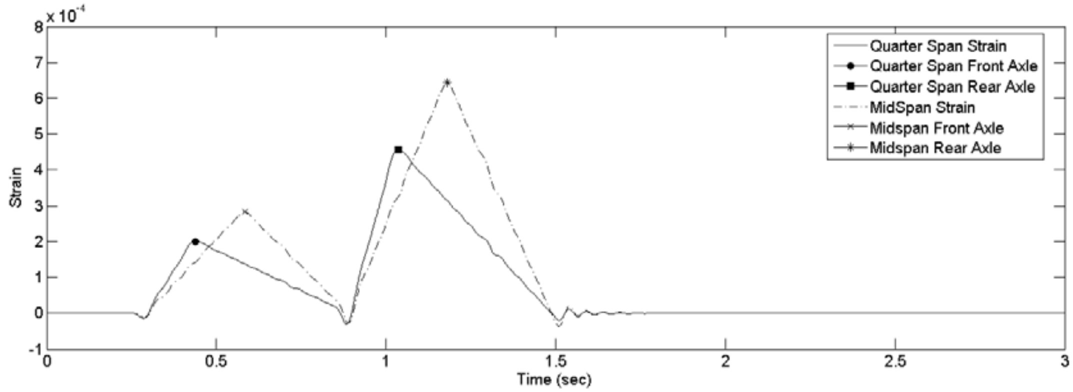


Figure 6.2: Filtered Midspan and Quarter Span Strain Time Histories for Simulated Bus Crossing at 16 MPH.

Once the maxima corresponding to the front axle crossing the recording elements have been determined, the speed can be calculated by dividing the distance in between the recording elements (3.5 ft) by the change in time between the axle first arriving at quarter and then midspan. For the example shown in Figure 6.2, the calculated speed was 16.45 MPH while the actual speed used in the numerical simulation was 16 MPH. This method was applied to the 631 bus patterns and the results of the calculations are shown in Figure 6.3, for the unfiltered, lowpass filter with 15 Hz cutoff frequency, and lowpass filter with 5 Hz cutoff frequency. The results for the unfiltered data are very poor, particularly at low speeds. This is attributed to the algorithm incorrectly identifying peaks not associated with the axles (as shown in Fig. 6.1). The best results were obtained by applying a filter with a 15 Hz cutoff frequency, for which case the average error was 6.34 MPH. From Fig. 6.3, there are two important observations:

In the data that was filtered using the 15 Hz cutoff frequency, at low speeds (less than 8 miles-per-hour) the algorithm occasionally overestimated the speed. In some cases, the error exceeded 100%. The explanation for these extremely high errors is shown in Fig. 6.4. Even after applying the lowpass filter, additional peaks associated with the bridge deck system vibrating at its first natural frequency were found. Figure 6.4 shows the strain time histories for a simulated bus traveling at 5.6 MPH. In this figure, the solid line is the quarter span strain time history and the dashed line the middle span. For the quarter span strain, the 1st identified peak (indicated by the ●) correctly corresponds to the front axle passing over the quarter point of the bridge decks. However, the first peak in the middle span strain time history (indicated with an “x”) does not correspond to the front axle crossing the middle of the bridge decks. Since the time difference between the first peaks is approximately 15% of the expected value, the calculated speed is six and a half times too large. This also explains the error at low speeds in the unfiltered data.

In the filtered data, as the speed increases the estimated speed diverges from the actual speed. From Figure 6.3, as the cutoff frequency decreases, the speed at which the estimated speed begins to diverge from the actual value used in the finite element simulation also decreases and the errors at the higher speeds worsens. For the data filtered using a 15 Hz cutoff frequency, at speeds beyond 40 MPH, the scatter in the estimation increases and begins to diverge from the actual value. For the data filtered with a 5 Hz cutoff frequency, the results are very good for speeds below 12 MPH. Beyond this speed, the estimated speeds diverge very quickly and the average error for the 631 cases was 28.5 MPH. For both of the filtered cases, the estimated speeds can be represented using a 2nd order polynomial.

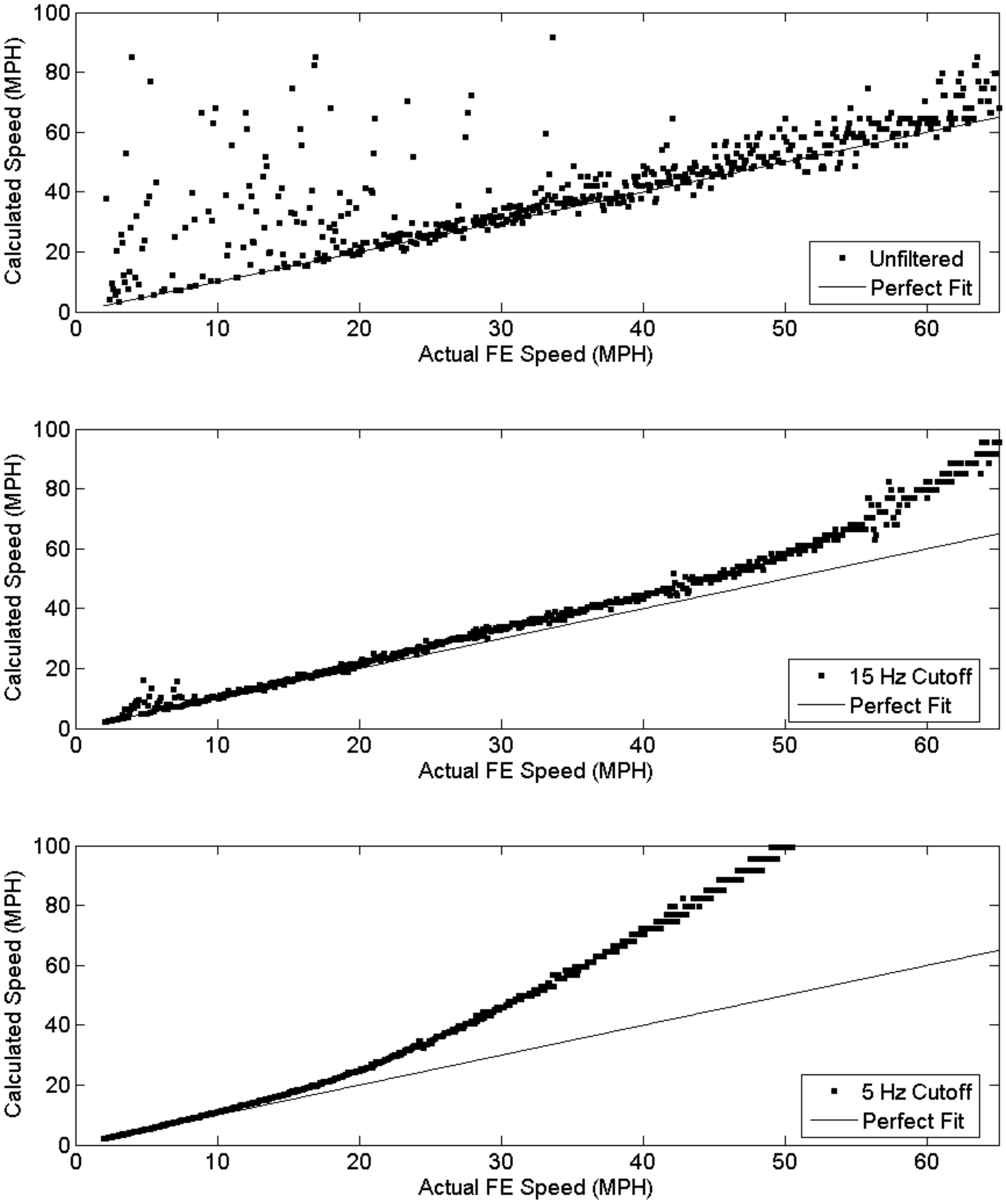


Figure 6.3: Comparison of Actual Speed Used in Finite Element Simulation Versus Calculated Value for Unfiltered, 15 Hz Cutoff Frequency, and 5 Hz Cutoff Frequency Cases.

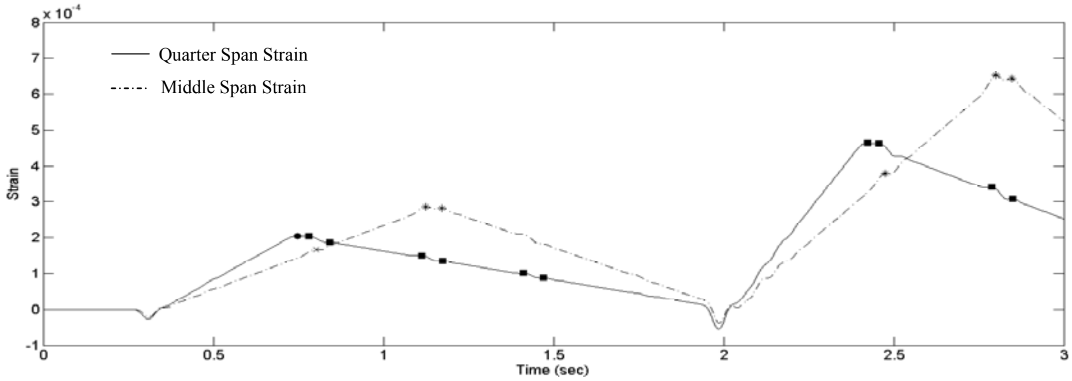


Figure 6.4: Strain Time Histories (Filtered with a 15 Hz Lowpass Filter) for a Simulated Bus Traveling at 5.7 MPH

The explanation for the divergence is related to the phase shift of the filter as is illustrated in Figures 6.5 and 6.6. As the cutoff frequency decreases, and the effect of the filter becomes more pronounced, the time history shifts to the right. This shift becomes more pronounced for higher speeds leading to the increase in the divergence.

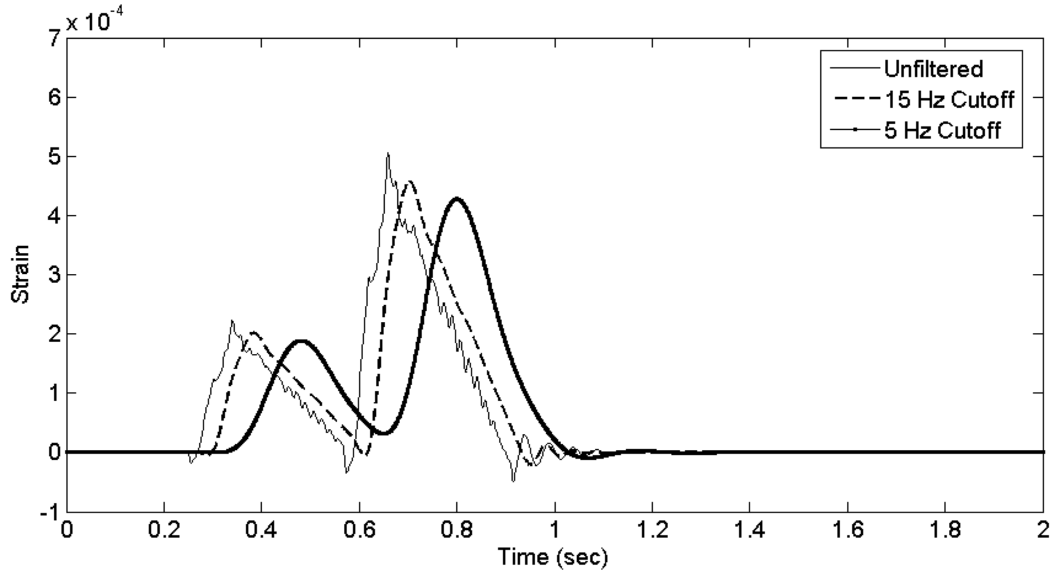


Figure 6.5: Comparison of Quarter Span Strain Time Histories for Unfiltered and Filtered (15 Hz Cutoff Frequency and 5 Hz Cutoff Frequency) Cases

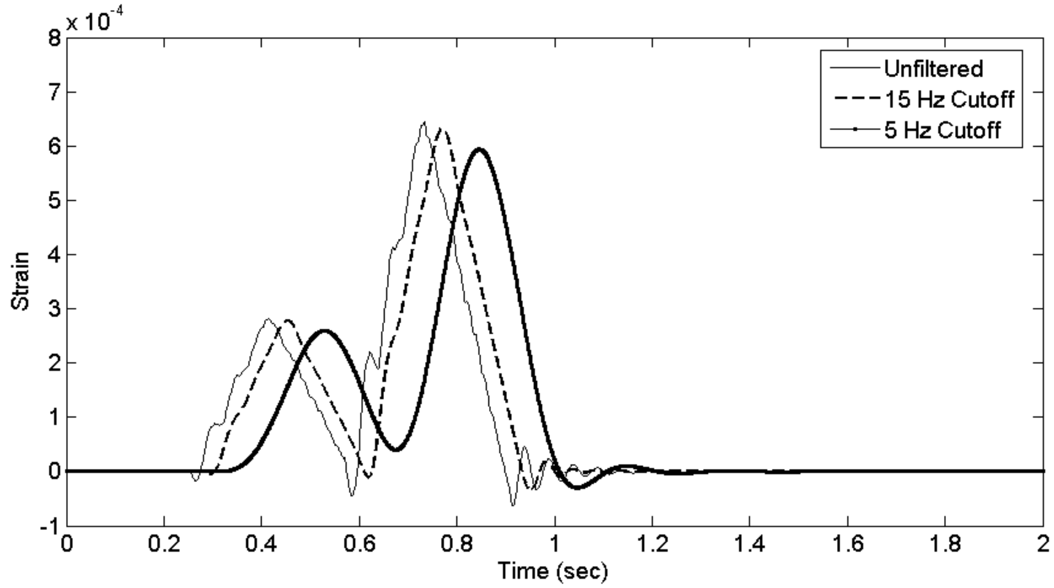


Figure 6.6: Comparison of Midspan Strain Time Histories for Unfiltered and Filtered (15 Hz Cutoff Frequency and 5 Hz Cutoff Frequency) Cases

To adjust for the affect of the filtering, polynomials were fit to the 15 Hz and 5 Hz cutoff frequency calculations and are represented respectively by equations 6.1 and 6.2.

$$Speed_{Calculated} = 0.0115 \cdot \left(Speed_{Actual}\right)^2 + 0.5461 \cdot Speed_{Actual} + 4.785 \qquad \textbf{Eq. 6.1}$$

$$Speed_{Calculated} = 0.0247 \cdot \left(Speed_{Actual}\right)^2 + 0.7639 \cdot Speed_{Actual} + 0.5461 \qquad \textbf{Eq. 6.2}$$

An adjusted speed was calculated for each case by plugging the calculated speeds into equations 6.1 or 6.2 and solving for the actual speed. The resulting adjusted speeds are shown in Figures 6.7 and 6.8. The average error has decreased to 1.50 MPH for the filtered data using a 15 Hz cutoff frequency and 0.38 MPH for the 5 Hz cutoff.

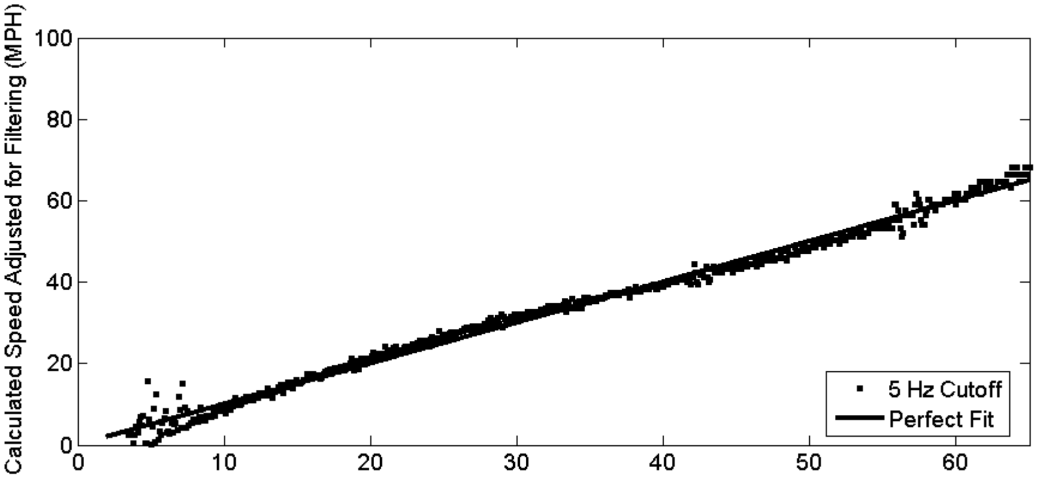


Figure 6.7: Comparison of Actual Speed Used in Finite Element Simulation Versus Calculated Speed Adjusted for Lowpass Filter (15 Hz Cutoff Frequency).

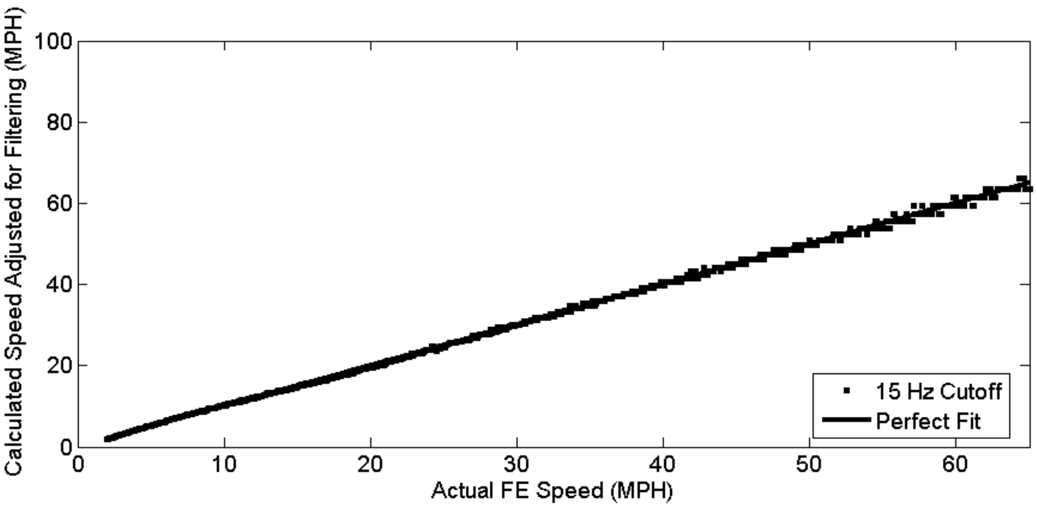


Figure 6.8: Comparison of Actual Speed Used in Finite Element Simulation Versus Calculated Speed Adjusted for Lowpass Filter (5 Hz Cutoff Frequency).

In addition to calculating the speed of a passing vehicle, it is also possible to estimate the wheelbase. This calculation is made by multiplying the calculated speed by the time delay between the front and rear axles crossing the quarter span recording element. The resulting wheelbase calculations for the 631 bus cases are shown in Figure 6.9 along with the adjusted values in 6.10, where the average error is 5.87 in.

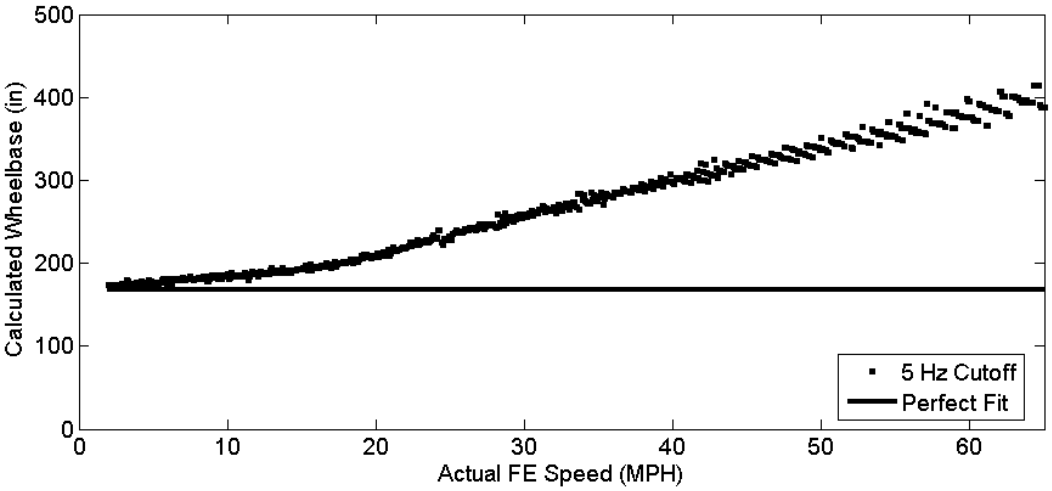


Figure 6.9: Comparison of Actual Speed from Finite Element Simulation Versus Calculated Wheelbase Using Filtered Data (5 Hz Cutoff Frequency).

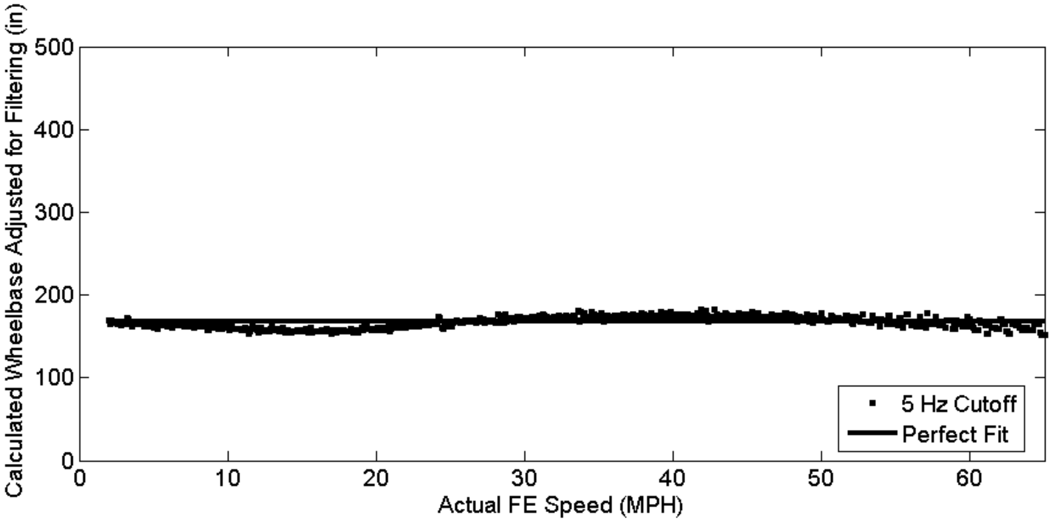


Figure 6.10: Comparison of Actual Speed from Finite Element Simulation Versus Adjusted Wheelbase Using Filtered Data (5 Hz Cutoff Frequency).

6.1.2 Traffic Identification for Random Vehicle Type

The second data set analyzed was composed of 300 load cases in which random values were assigned to the speeds, wheelbases, and axle weights. In this study, the speed ranged from 5 to 40 MPH, wheelbase from 120 to 240 in, front axle weight from 2 to 12 kips and rear axle weight from front axle weight plus 0 to 10 kips. Histograms of the vehicle speeds, wheelbases, front axle weight, and rear axle weight are shown in Figures 6.11-6.14.

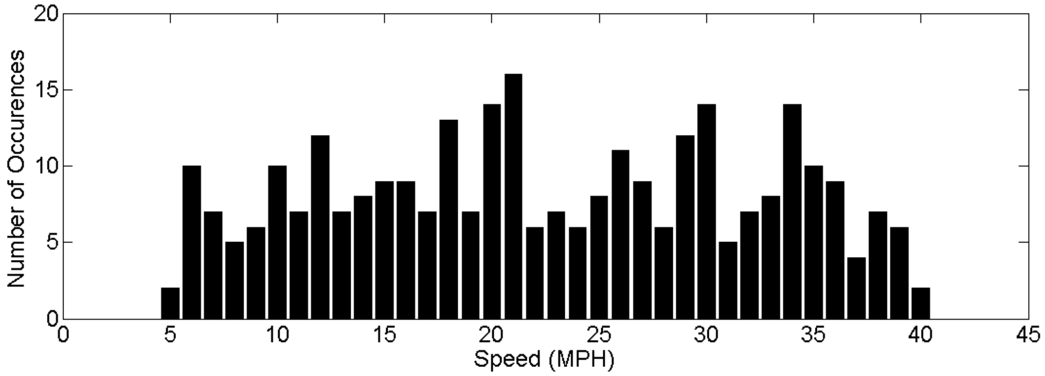


Figure 6.11: Histogram of Simulated Vehicle Speeds

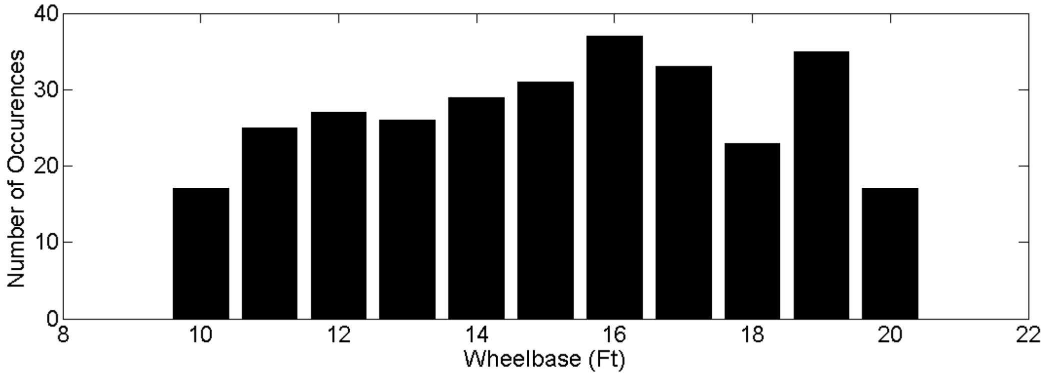


Figure 6.12: Histogram of Simulated Vehicle Wheelbases

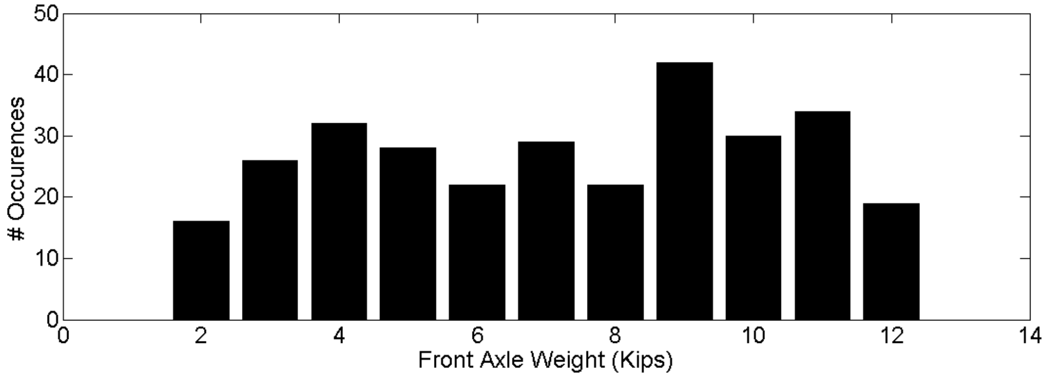


Figure 6.13: Histogram of Simulated Vehicle Front Axle Weights

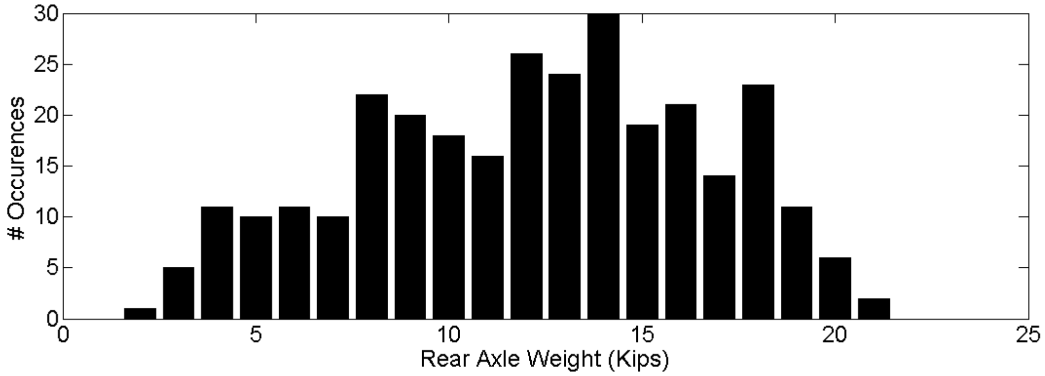


Figure 6.14: Histogram of Simulated Vehicle Rear Axle Weights

A 5th order lowpass filter with a 5 Hz cutoff frequency was applied to each of the strain time histories and as with the first data set, the speed and wheelbase of each vehicle were calculated. The average error for speed calculation is 14.20 MPH and after adjusting for filter affects, using the relationships determined for the previous data set, the error decreases to 1.87 MPH (Fig. 6.15). For the wheelbase, the average error is 79.57 in and 32.69 in respectively (Fig. 6.16).

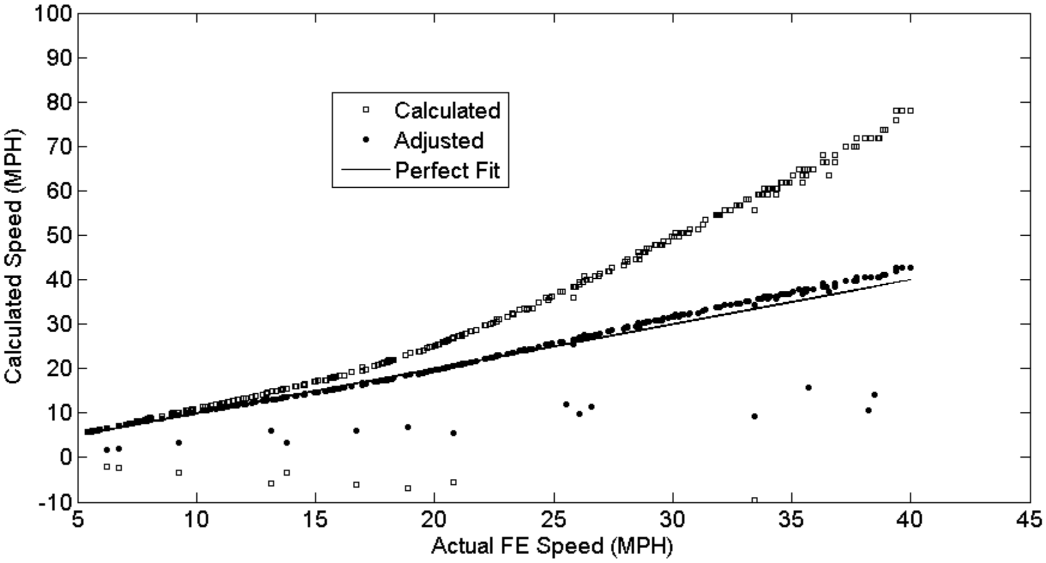


Figure 6.15: Comparison of Actual Speed from Finite Element Simulation versus

Calculated Speeds Using Filtered Data (5 Hz Cutoff Frequency).

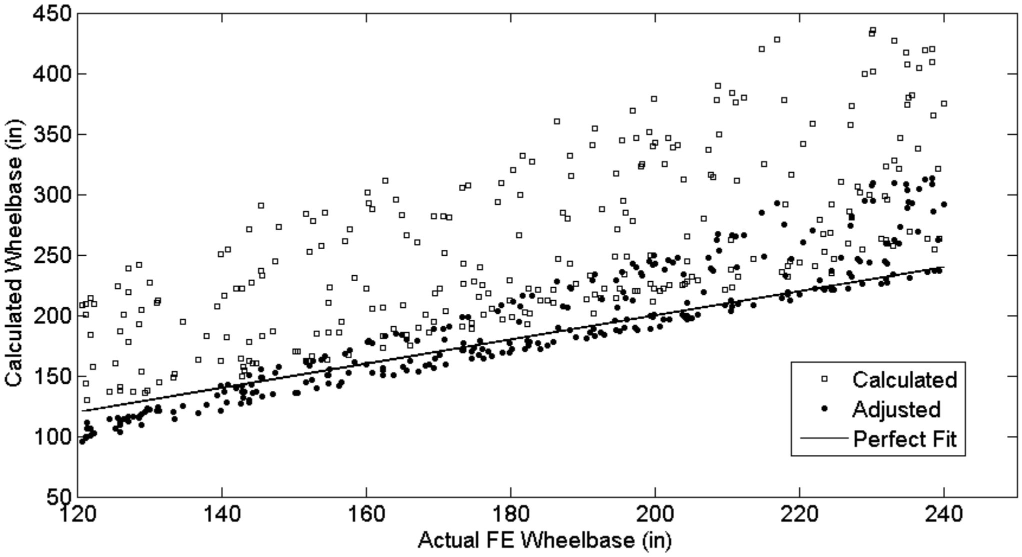


Figure 6.16: Comparison of Actual Wheelbase from Finite Element Simulation versus Calculated Wheelbase Using Filtered Data (5 Hz Cutoff Frequency).

6.2 Damage Identification Using the One-Dimensional FE Model

The previously discussed one-dimensional finite element model with moving loads was used to simulate traffic passing over the undamaged and damaged bridge decks. To simulate damage, Young's modulus of one or more elements is reduced. By examining the response of the bridge decks and looking for a change in strain between the undamaged and damage scenarios it is possible to detect damage.

6.2.1 Damage Detection Using a Single Vehicle Type

For this initial study, damage patterns were restricted to either uniform damage or damage occurring in one of the three local damage zones (Fig. 6.17) located at quarter, middle, and three-quarter span. The first step involved verifying a change in the bridge deck response could be observed in the damaged configuration. For this, the simplest case of a single vehicle was considered. To start, the value of Young's Modulus was reduced by 25% for all of the elements and then for only those elements that form local damage zone 1. The time histories for element 16 in Fig. 6.18 show no change in the bending moment between the undamaged and damaged configurations. However, when strains are computed in accordance with equation 5.14, it is clear that there is a very noticeable change in strain (Fig's. 6.19 and 6.20). This is a very useful observation as this is the property that is easily measured on the real bridge deck system (using either electrical resistance or fiber optic strain gages).

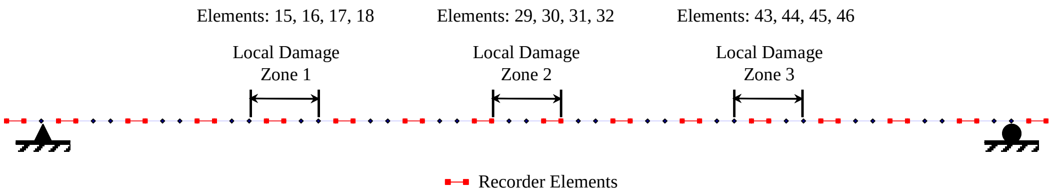


Figure 6.17: One-Dimensional Finite Element Model Detailing Location of Recorder Elements and Local Damage Zones.

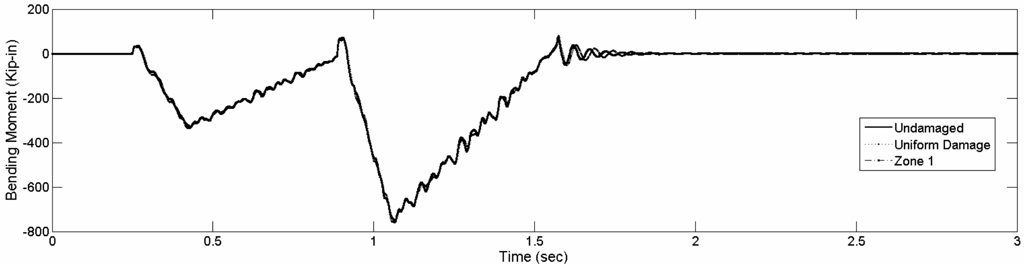


Figure 6.18: Element 16 Bending Moment Time History for Undamaged, 25% Uniform Reduction in Stiffness, and 25% Reduction in Zone 1.

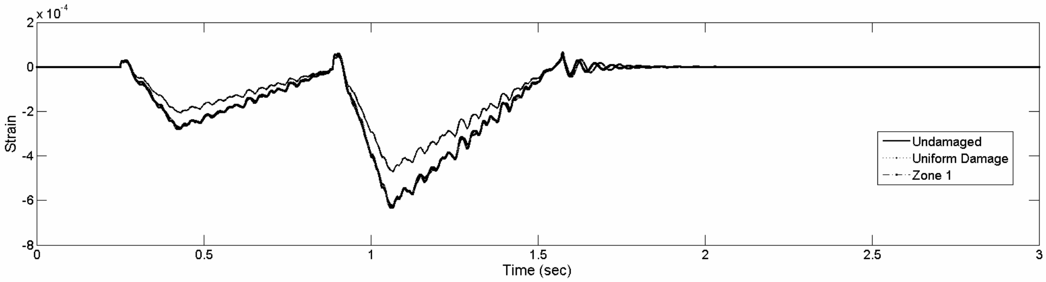


Figure 6.19: Element 16 Strain Time History for Undamaged, 25% Uniform Reduction in Stiffness, and 25% Reduction in Zone 1.

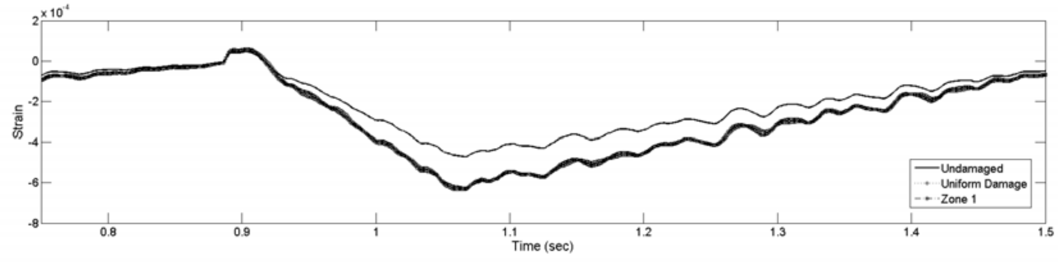


Figure 6.20: Close-up of Element 16 Strain Time History for Undamaged, 25% Uniform Reduction in Stiffness, and 25% Reduction in Zone 1.

Next, the strains in the undamaged elements were examined for a 25% reduction in stiffness for Zone 2 elements. The strain time histories for elements 16, 28, and 32 are shown in Figure 6.21-6.23. From these figures, there is no change in strain for any of the undamaged elements, even those (element 28) situated right next to the damaged elements.

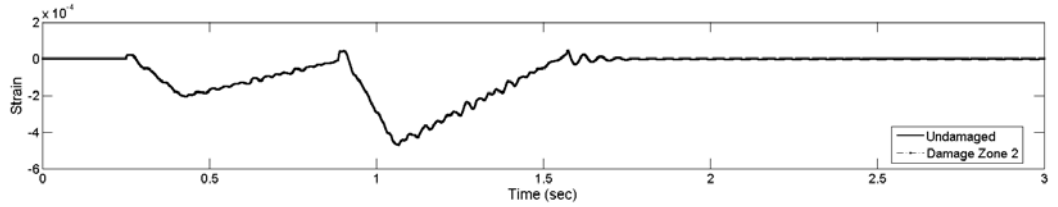


Figure 6.21: Element 16 Strain Time History for 25% Reduction in Zone 2 Stiffness.

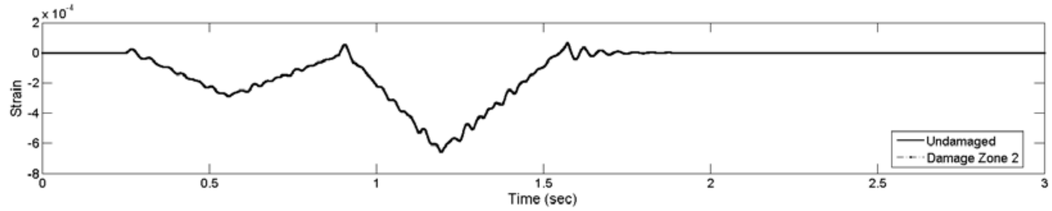


Figure 6.22: Element 28 Strain Time History for 25% Reduction in Zone 2 Stiffness.

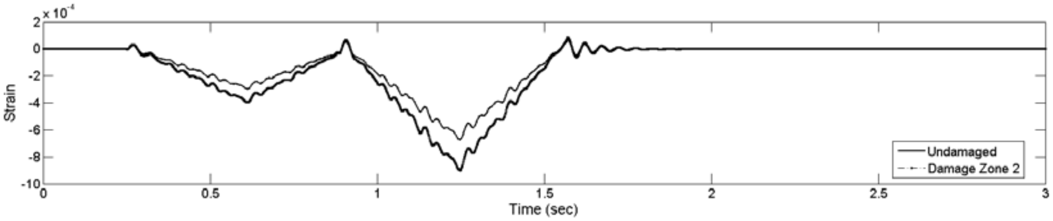


Figure 6.23: Element 32 Strain Time History for 25% Reduction in Zone 2 Stiffness.

As only those elements which are damaged present a change in strain, it was decided to focus on the peak strain from each of the recording elements. In the undamaged state, the strain profile generated by plotting only the peak strains at the recording elements is a very smooth function (Fig. 6.24). When damage occurs in one of the local damage zones, there is a very pronounced change in the strain profile at the location of the damage (Fig. 6.24).

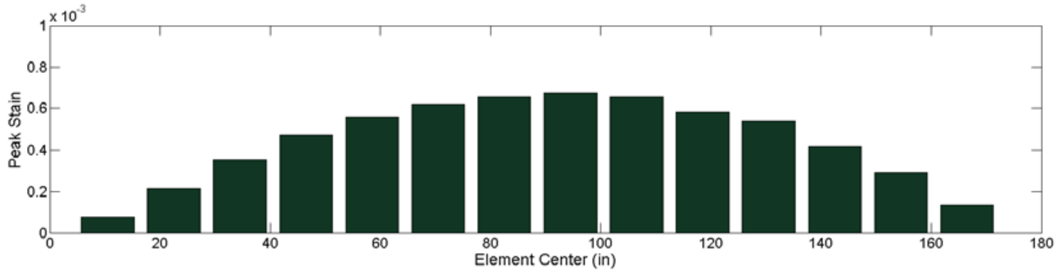


Figure 6.24: Peak Strain Profile for Undamaged State.

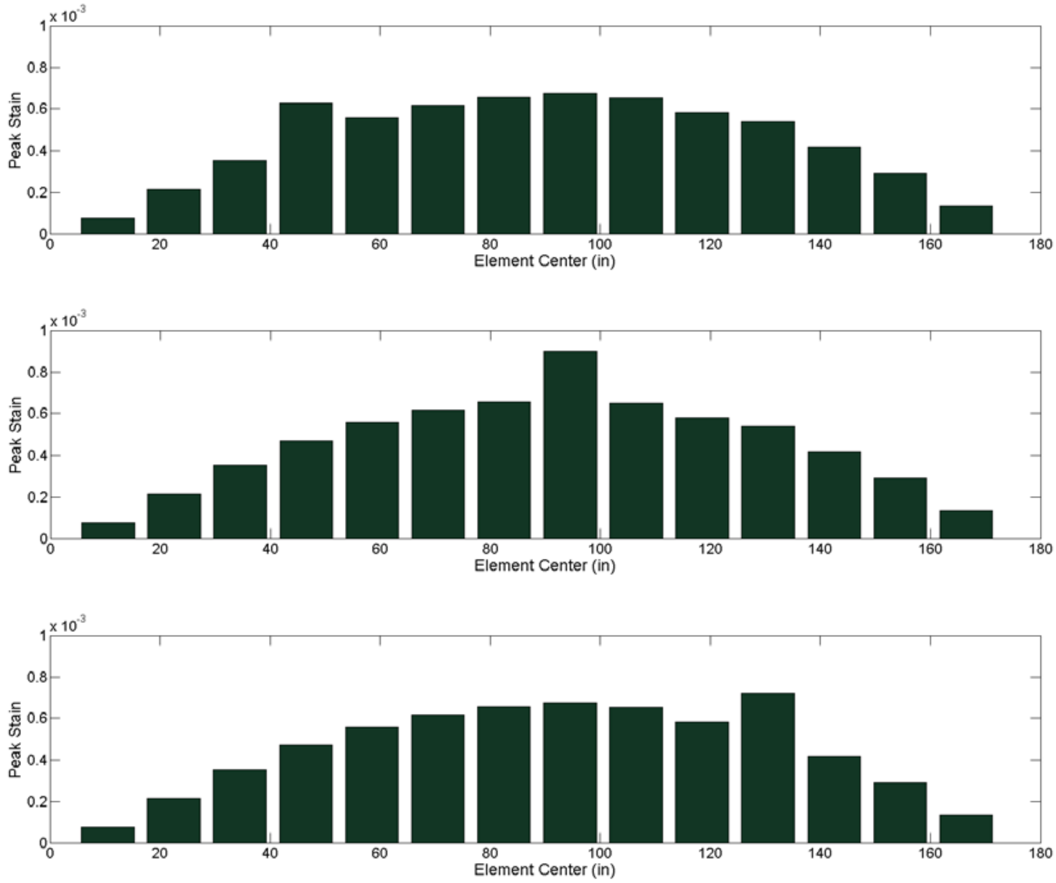


Figure 6.25: Peak Strain Profile for Local Damage in Zones 1, 2, and 3 (Top, Middle, Bottom).

When the strain profiles for the local damage scenarios are subtracted from the profile for the undamaged state, the change is even clearer (Fig. 6.26). As expected, the increase in the peak strain is inversely proportional to the decrease in Young's Modulus. This observation makes it possible to quantify the level of damage; however, it is only possible when data from the undamaged state for the same vehicle traveling at exactly the same speed is available. Consequently, this method of detecting and quantifying the level of damage is not practical for random traffic loading.

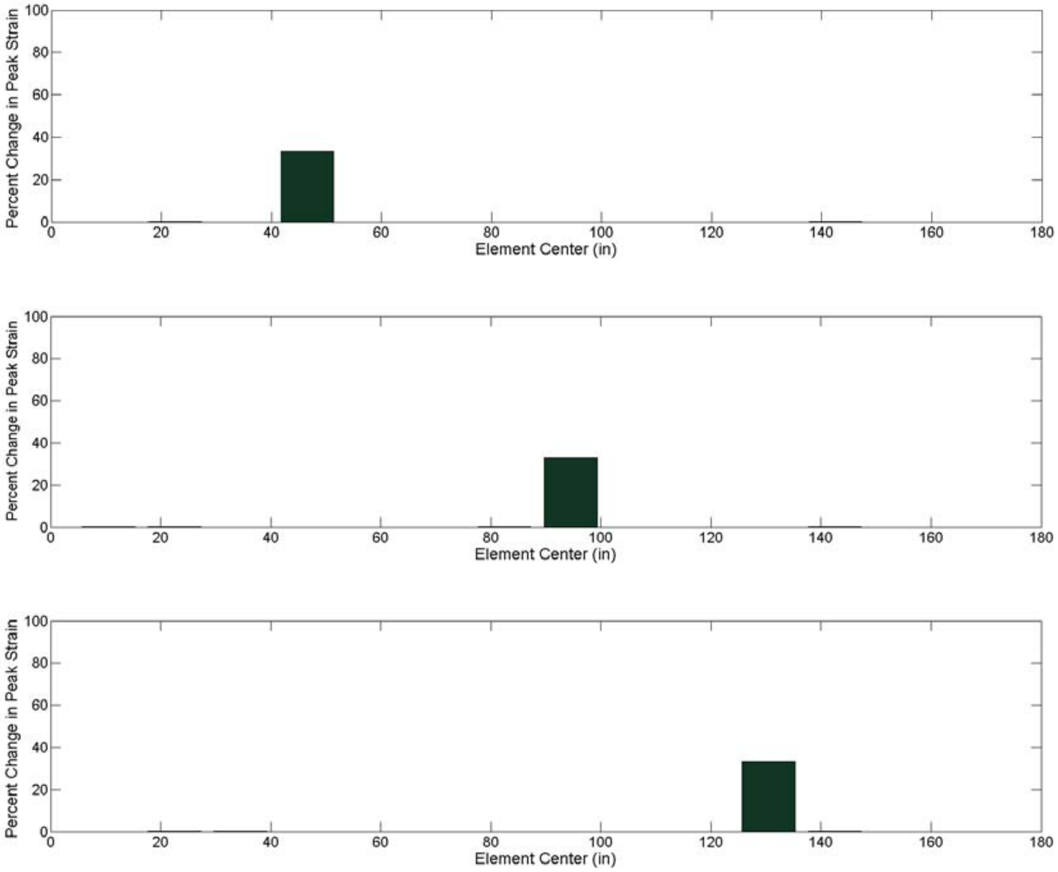


Figure 6.26: Change in Strain Profile Corresponding to 25% Reduction in Stiffness for Zone 1, 2, and 3 Elements.

6.2.2 Damage Detection Using Random Traffic

As peak strains at the recorder elements appear to be a good indicator of damage, a study was conducted using the aforementioned traffic set from section 6.1.2, consisting of 300 random vehicle parameters. The 300 input patterns were separated into four cases: undamaged, local damage in Zone 1, local damage in Zone 2, and local damage in Zone 3. For the damaged scenarios the value of the modulus of the affected elements was randomly reduced up to 50%. Damage was restricted to one zone at a time and the values of Young's modulus for the four elements that constitute each damage zone were uniformly reduced.

A plot of the percent reduction in Young's modulus for the 300 cases is shown in Figure 6.27. For each simulation, the resultant forces and moments were output and recorded for the 16 elements shown in Figure 5.15. From the moments, the strains were calculated and the peak strain profiles along the length of the beam were saved. For each of the four damage scenarios, 50 of the 75 strain profiles were selected at random and first normalized (such that the sum of the sixteen normalized strains equals 1.0) and then averaged to form a typical profile representative of each damage case (Fig. 6.28). It should be noted that this form of normalizing the strain profiles is only one of a vast number of possible ways.

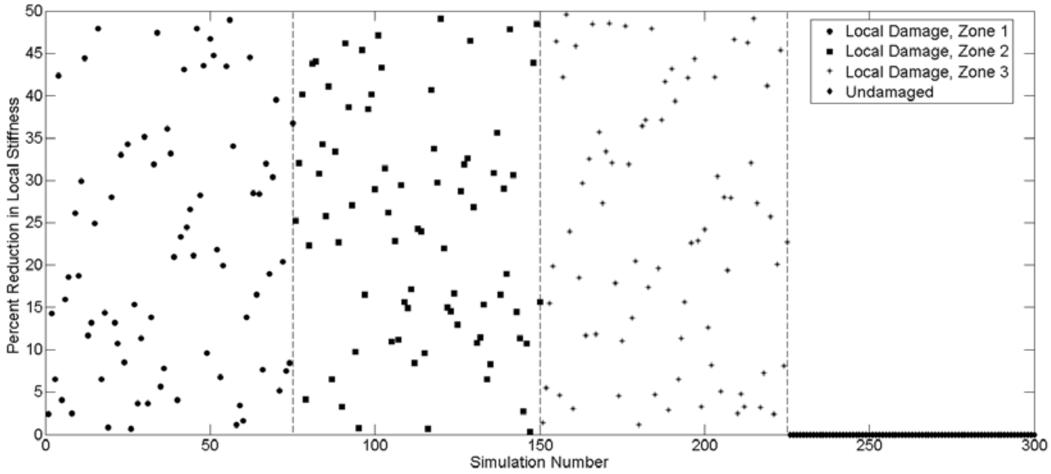


Figure 6.27: Plot of Percent Reduction in Local Stiffness versus Simulation Number.

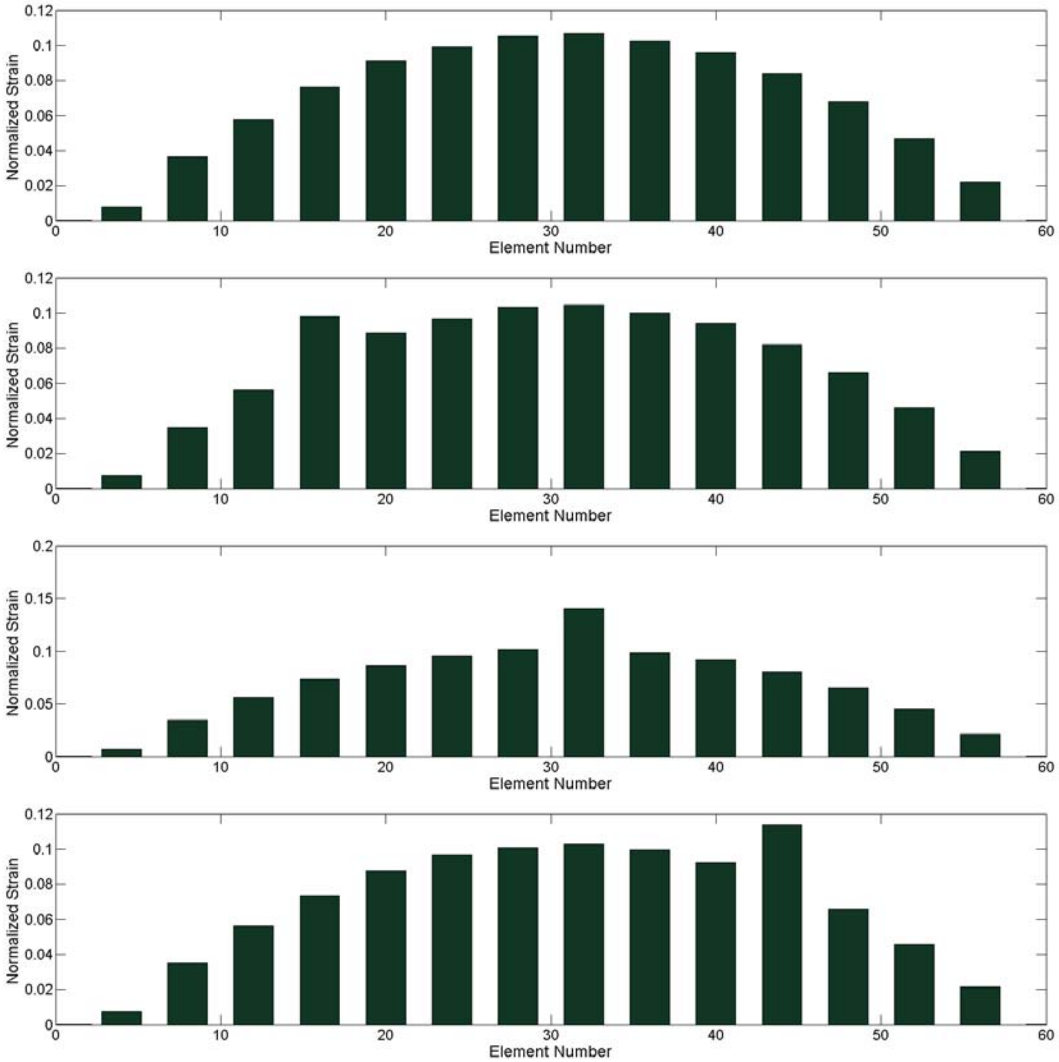


Figure 6.28: Typical Strain Profiles (50 Averages) for Undamaged, Zone 1 Local Damage, Zone 2 Local Damage, and Zone 3 Local Damage Cases.

The 300 strain profiles were compared to each of the four typical profiles and the root-mean-squares (RMS) of the absolute differences between the average and individual profiles was calculated (Fig. 6.29). For each load case, a damage assessment was performed by deciding which of the four typical profiles the individual strain profile most closely fits. This was done by finding the minimum RMS value for each case and assigning a damage type (undamaged, Zone 1, Zone 2, or Zone 3) based on this value.

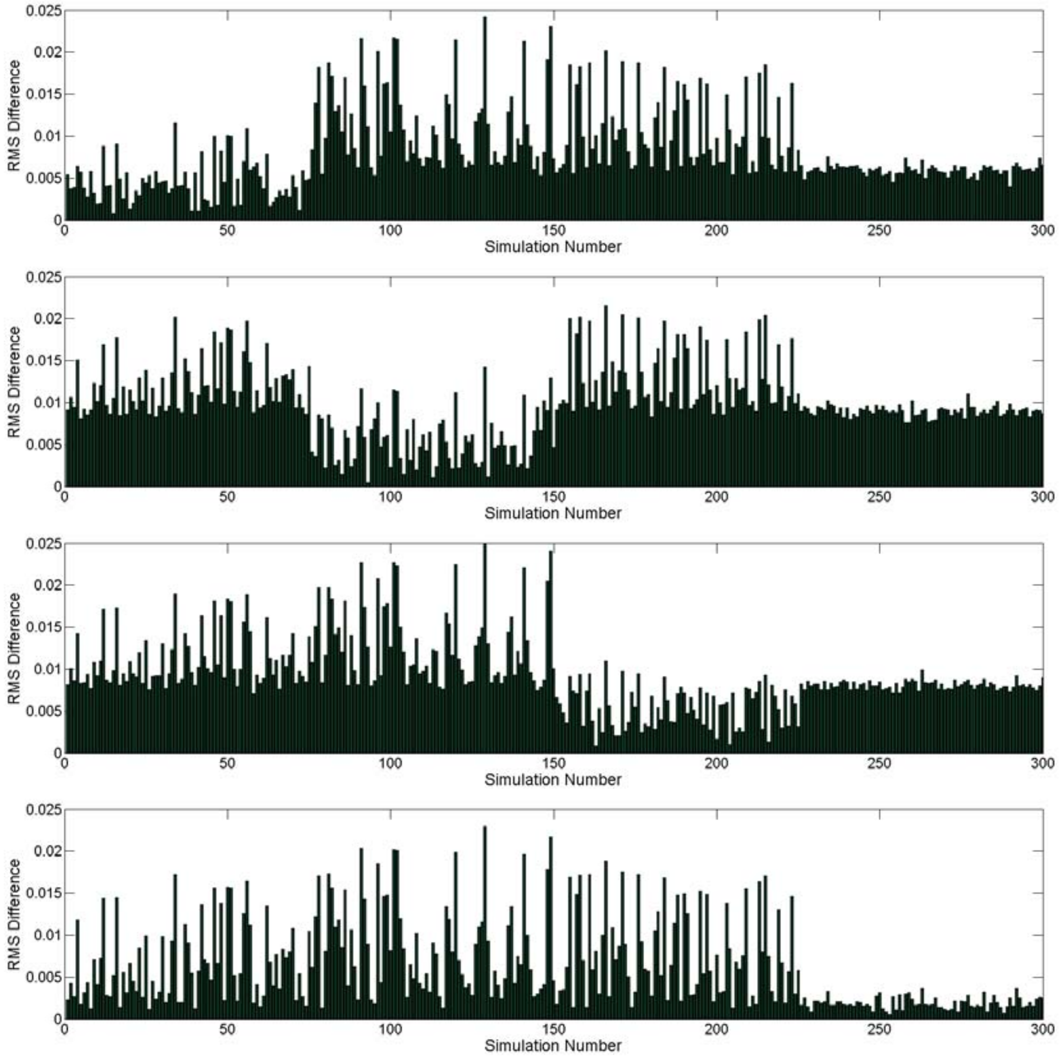


Figure 6.29: RMS of Difference Between Strain Profile and Typical Zone 1, Zone 2, Zone 3, and Undamaged Profiles.

Based on this method, 73.67% of the 300 records shown in Figure 6.30 were correctly labeled. Further conclusions which may be drawn from this figure include:

- All cases in which the damage was represented by a reduction in Young's Modulus greater than 17% were correctly labeled.
- All but one of the damage scenarios in which the reduction was less than 10% were incorrectly labeled as undamaged.
- All of the undamaged cases were properly labeled.
- For scenarios in which the damage was represented by a 5 to 17% reduction in Young's Modulus, over 80% of the 59 cases were incorrectly labeled as undamaged.

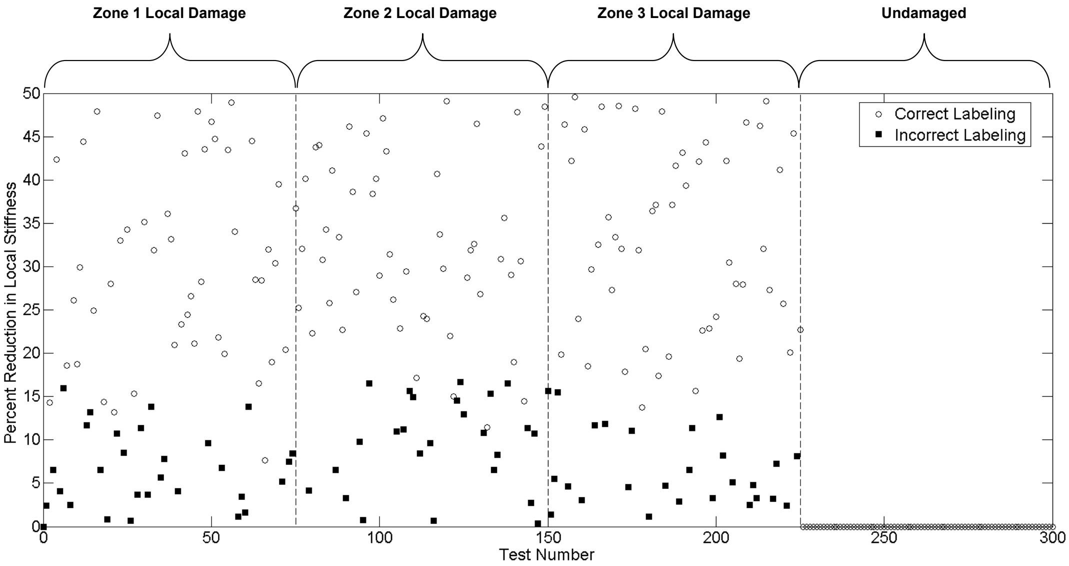


Figure 6.30: Scatter Plot of Damage Labeling versus Percent Reduction in Young's Modulus.

Overall, this method proved only moderately effective in identifying damage and assigning the proper labeling. From Figure 6.31, in all cases in which the method correctly identified the presence of damage it also assigned the proper location. For damage levels less than approximately 15%, the method was unlikely to recognize the damage, thereby making it unsuitable for detecting early on the onset of progressive damage. Further, this method is nonconservative as there were no false positive damage indications. In summary, this method only proved moderately effective at detecting damage. In particular, for low level damage (less than 15%), it tended to assign an undamaged label (only 10% of these damage patterns were properly identified). Also, this method provide little information regarding the extent of the damage. For these reasons, an improved damage detection/classification method using Neural Networks was developed and is presented in the following chapter.

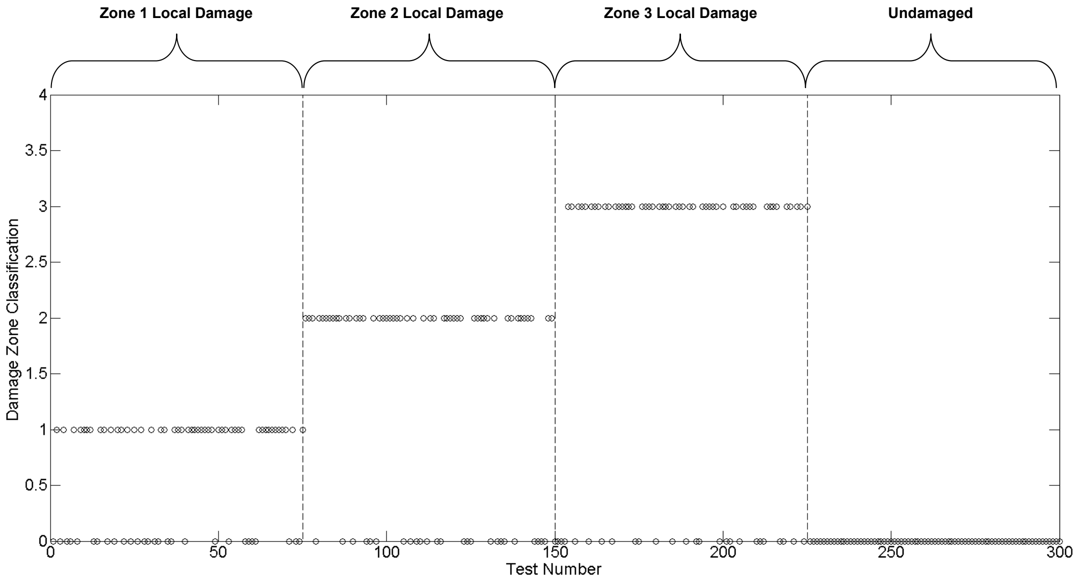


Figure 6.31: Damage Zone Classification versus Actual Damage Zone

6.3 Summary

The strain time histories generated from the finite element model were used for vehicle property estimation and damage detection. These operations were first done using the peak strains and then later with neural networks. Peak middle and quarter span axle strains (along with the time at which they occur) from the strain data were used for calculating speeds and wheelbases for each simulated vehicle. For the random traffic, the average error in the speed calculation was 1.87 MPH and 32.69 inches for wheelbase. For damage identification, the peak strains at sixteen locations distributed along the length of the FE model were utilized. For the case of a single vehicle type with constant speed, wheelbase, and axle weights, it was possible to accurately determine the extent and location of damage. Under the more difficult case of random traffic loading (in which vehicle properties are randomly generated), it was only possible to detect local reductions in stiffness over 15%.

7 Application of Neural Networks for Vehicle Property Estimation and Damage Identification/Classification Using Simulated Data

In this chapter, neural networks are applied to the data from the finite element model to determine vehicle properties and for damage detection purposes. First, neural networks are applied for estimating vehicle properties (speeds, wheelbases, and axle weights) from traffic data in which one property at a time is changed and thereafter from random traffic data. In this study, multiple forms of feature extraction were explored. Next, neural networks were used with peak strains to perform damage detection and classification. Finally, an alternate method is detailed in which strain time histories from one location in the finite element model are used as inputs to a neural network which then predicts strains at a neighboring location, and comparisons between the predicted and actual strains are used to detect and assess the level of damage.

7.1 Utilizing Neural Networks within an Integrated Health Monitoring Framework

Using the latest enabling technologies, the objectives of health monitoring are to detect, locate, and assess the level of structural damage to the civil infrastructure. As changes in a systems dynamic response can result from either the presence of damage or from environmental/operational factors (loading conditions, thermal loading, …), another important factor for bridge monitoring is determining traffic loading parameters, namely vehicle speed, wheelbase, and axle weights.

Within this chapter neural network applications for traffic identification and damage detection within a general framework of structural health monitoring are discussed. As previously presented, an integrated health monitoring framework is being developed to incorporate all tasks from sensor configuration, data acquisition and control, to decision-making and resources allocation. Within this framework, neural networks, which do not require information concerning the phenomenological nature of the system being investigated, are employed to detect changes in model-unknown structures.

Neural networks, which were originally developed to form a mathematical model of the human brain [Bishop, 1998], derive their name from their correlation to a collection of interconnected neurons (the cells that perform information processing in the brain) [Russell and Norvig, 1995]. They have been shown to work well in widely varying fields, providing satisfactory performance in tasks that otherwise prove difficult to solve explicitly using other numerical techniques. To date, these networks have been successfully applied for:

- Pronunciation of written English text by a computer [Sejnowski and Rosenberg, 1987], which involved learning the mapping from text to phonemes.
- Reading zip codes on hand-addressed envelopes, a character recognition problem [Le Cun et al. 1989].
- Steering a vehicle on a single highway lane by observing the performance of a human driver [Pomerleau, 1993].

- Within civil engineering, Masri et al. (1996) and Nakamura et al. (1998) have applied neural networks to recorded vibration data for damage detection.

While often treated as a black box, neural networks require careful attention in their construction. In developing the network, the user must consider the number and types of units and activation functions to use and how these units are to be connected. For training, it is necessary to select which features to use as inputs for the network, as well as what are the outputs (for the example of damage detection on a civil structure, this could be: occurrence of damage, location of damage, level of damage, type of damage, or some combined form).

An important limitation of neural networks is that like any supervised learning method, they require data from the undamaged and damaged states of the structure and generally data from the damaged state is difficult to come by. A common alternative is to generate data from a calibrated finite element model simulating each of the potential damage states. However, the success of this method is closely tied to how accurately the model simulates the actual structure. Another limitation is damage often occurs at more than one location at a time. For simulating training data where all possible damage scenarios must be simulated, the number of damage locations and levels can lead to an exponential increase in the number of patterns which must be considered. Finally, unlike decision trees which follow a logical derivation for new data, neural networks provide no explanation for why a given output value is reasonable, even when they predict well [Russell and Norvig, 1998].

7.2 Vehicle Property Estimation Using Data from One-Dimensional Finite Element Model with Vehicles with Only One Variable Characteristic

The aforementioned computational model of the bridge-deck system (discussed in Chapters 5 and 6) was employed for generating simulated data to explore the potential of neural networks as a traffic identification tool based on changes in strain time histories [Fraser et al,. 2004]. To simulate traffic crossing over the bridge-decks, the one-dimensional finite element model, composed of sixty beam-column elements, (Fig. 7.1) was analyzed using the computational framework OpenSees (http://opensees.berkeley.edu/). The goal of this research is to apply the neural network to identify/predict traffic speeds and wheelbases based on strain time histories. If successful, these techniques may be applied in future research to actual measured data with the ultimate goal of reducing uncertainty during the system-identification analysis phase (by limiting the scope of possible causative load configuration scenarios).

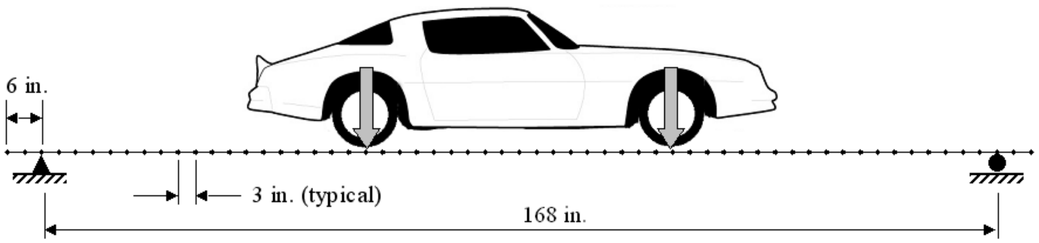


Figure 7.1: Finite Element Model of Composite Bridge-Decks

Considering the range of vehicles likely to be encountered on the bridge-deck system, two classification problems were chosen for this initial study. The first data set consisted of the situation where the vehicle type (axle weight and wheelbase) was kept constant and the speed varied in quarter mile-per-hour (MPH) increments from 5 to 65 MPH. For each of these scenarios, strain time histories were generated at middle and quarter span. In this study, a single vehicle passing over the bridge-decks is simulated by ignoring the interaction between the beam and vehicle, instead considering only the moving forces [Foda and Abdujabbar, 1998 and Chen and Feng, 2006]. This simulation replicates the situation where a campus shuttle bus crosses the decks at a speed somewhere between 5 and 65 MPH (Fig. 8.2). Of the 241 scenarios generated, 201 were selected and served as the training set for the neural network. The remaining 40 cases were reserved as a test set to evaluate the network performance.

Figure 7.2: Typical UCSD Shuttle Buses Crossing Over Composite Bridge Decks

In addition, a second class of scenarios was considered in which the vehicle’s speed and axle weights were held constant and the wheelbase varied from 5 to 30 feet (in 0.25 foot increments). For this class, 81 of the scenarios were used for training and the remaining 20 for testing.

If all the time steps in the strain time histories (for examplea a ten-second long time history with a 0.001 second time increment is composed of 10,000 points) were to be considered features and implemented as neural network inputs, the network structure would grow far too large and the training would become time consuming and effort intensive. Furthermore, because strain time histories are continuous curves, the information in the data is actually highly redundant. Therefore, data compression and feature extraction were performed.

Principal Components Analysis (PCA) was herein employed for feature extraction [Yan et al., 2004 a, b, c]. PCA reduces the dimension of a data set by transforming the original (possibly correlated) features into a new set of uncorrelated features (i.e., Principal Components or PCs [Jolliffe, 1986]. These PCs are ordered so that the first few retain most of the variance displayed by the original features. For this data set, the information of significance was mainly contained in the first 10 to 20 principal components, which were used as the network inputs. In this example, for each pair of time histories corresponding to one particular event, the number of features was reduced from 20,000 (the strain at every one of the time steps in both of the ten-second long strain time histories is considered to be a feature) to the first ten principal components. These ten PCA features were used as the input in the neural network.

Following Yan et al. (2004a), a typical two-layer perceptron neural network (Fig. 8.2) was used as the statistical pattern classifier. This was done in accordance with Bishop (1995) which showed a two-layer neural network using a sigmoidal activation function is capable of approximating any continuous functional mapping. Performance of the network is highly dependent on the activation functions being chosen. According to LeCun (1998), using a symmetric sigmoidal activation function such as a hyperbolic *tanh* often leads to faster convergence during training. Thus, the activation function of the hidden layer employed hyperbolic *tanh* function as recommended by LeCun (1998):

$$g(h) = 1.7159\tanh(2h/3) \qquad \textbf{Eq. 7.1}$$

where *h* is the value of the corresponding hidden unit. The "softmax" activation function (Bishop 1995),

$$y_i = e^{a_i} / \sum_j e^{a_j} \qquad \textbf{Eq. 7.2}$$

was used as the activation function of the output layer so as to obtain a vector of positive values whose sum is 1.0, where y_i is the ith component of network output vector [Y], and a_j is the value of the jth output unit (Fig. 7.3).

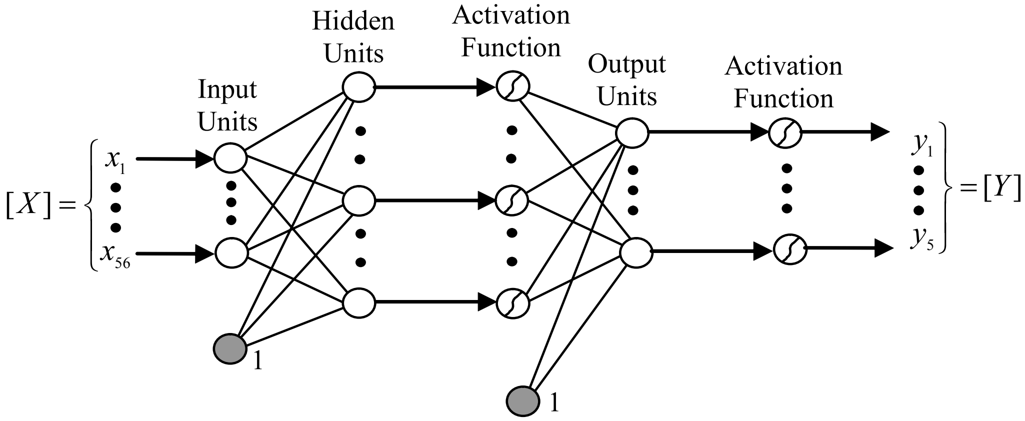


Figure 7.3: Neural Network Structure [Yan et al., 2005]

The neural network was then defined with the 10 input units, 15 hidden units (preliminary studies showed 15 to be sufficient), and a single output unit (corresponding to normalized speed or wheelbase). During training, the example inputs are fed into the network and if the network computes outputs that match the target values (actual values used in the finite element simulation) then nothing is done to the network. However, if there is an error whereby the outputs do not match the targets, then the weights in the network must be adjusted. However, with the two layer neural network used in this problem, there are many weights connecting each input, hidden, and output unit the difficulty becomes in assessing how the error should be divided amongst the contributing weights [Russell and Norvig, 1994]. This problem is compounded in the damage detection/classification application where multiple outputs are used and each of the weights contributes to more than one output. The back-propagation learning algorithm [Bishop, 1995] solves the problem of dividing the contribution of the weights and was therefore used to train the network, and adjust the weights by minimizing the error between network outputs and targets (corresponding desired values for the outputs), where the error was defined by the cross entropy error function [Bishop 1995]:

$$Error = -\sum_i [t_i \log(y_i / t_i)] = -\log \sum_i (y_i t_i) \qquad \textbf{Eq. 7.3}$$

where, t_i is the ith component of target vector [T].

The traffic patterns in the training set were visited sequentially, and the network configuration was updated by the back-propagation learning algorithm after each visit. Since the training set was composed of only 201 traffic patterns, the data was repeatedly used during this phase. Each loop over all patterns visited during training is called an epoch.

This optimized network configuration was then subjected to the test set with 40 traffic patterns. The same PCA transformation used for the training set was applied to the test set so as to extract compatible Principal Component features.

Comparing the network outputs with the targets for all patterns in the test set, the average error in the speed estimation was 0.28 MPH (Fig. 7.4). For wheelbase estimation, the average error was 0.30 feet (Fig. 7.5).

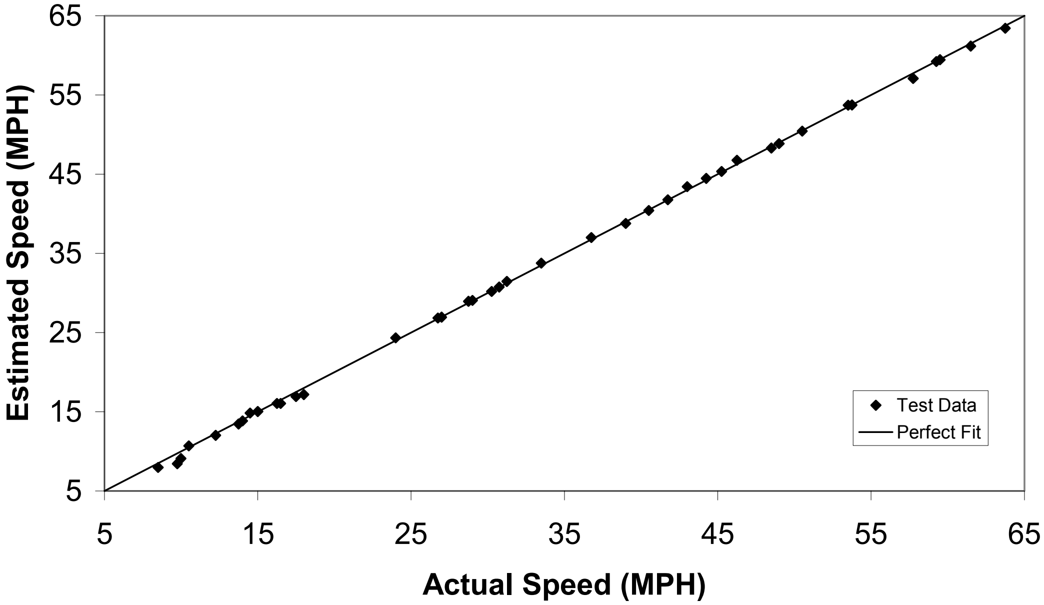


Figure 7.4: Comparison of Estimated and Actual Speeds

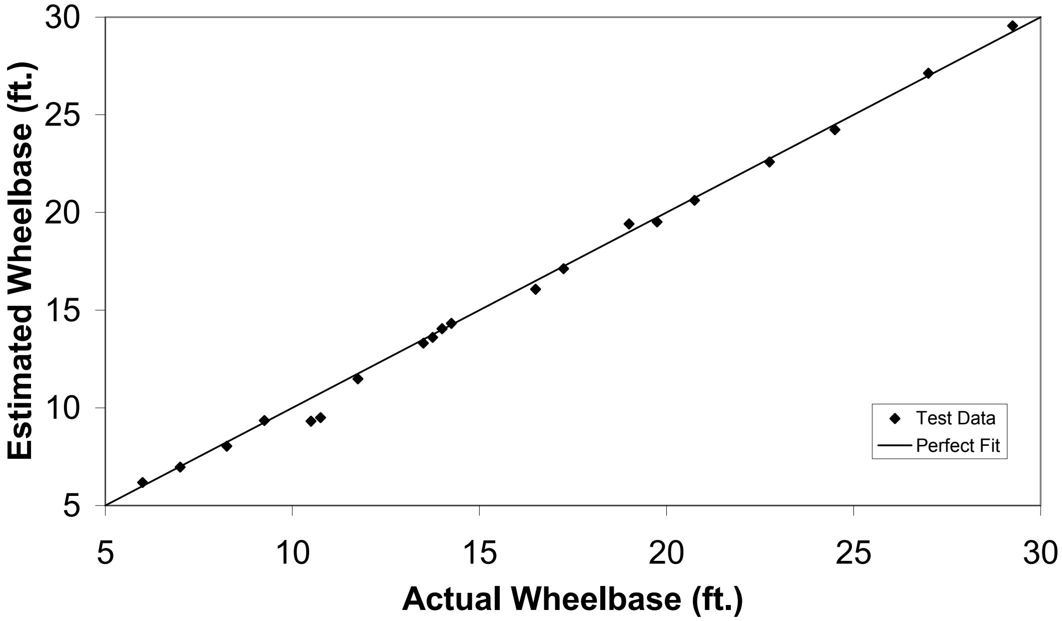


Figure 7.5: Comparison of Estimated and Actual Wheelbases

7.3 Vehicle Property Estimation Using Data from One-Dimensional Finite Element Model with Random Traffic Loading

Having demonstrated the viability of neural networks as a tool for identifying vehicle properties when only one vehicle property (either speed or wheelbase) is varied, the effectiveness under the more realistic scenario of random traffic loading was explored. As training, validating, and testing a neural network requires a large data set, it was impractical to weigh and measure numerous vehicles and them drive each of them over the actual bridge decks. Consequently, the three-hundred sets of strain time histories corresponding to the random traffic generated with the one-dimensional finite element were employed. The properties of this data set were discussed in detail in section 6.1.2. For this data, the speed, wheelbase, front axle weight, and rear axle weight for each vehicle was determined using four separate neural networks.

7.3.1 PCA-Based Feature Extraction

Rather than using the data from all 16 of the recording elements, only two were selected – elements 12 and 28. These elements were chosen as they correspond roughly to the quarter and middle span locations of the bridge-decks and are outside of the local damage zones. This is an important consideration since damage was modeled as a reduction in stiffness. Therefore, the presence of damage (with its corresponding reduced stiffness) causes higher strains which lead to errors in the determination of axle weights. To further reduce the size of the data put into the neural network, Principal Component Analysis was again used thereby reducing the number of features from 8000 to just 20 (corresponding to the first 20 principal components). 200 of the traffic patterns were randomly selected and used for training while the remaining 100 were reserved for testing.

The neural network was then defined with the 10 input units, 15 hidden units (preliminary studies showed 15 to be sufficient), and a single output unit. For optimizing the network configuration, the backpropgation learning algorithm was applied for 200 loops over all of the data (epoch = 200) as was done with single vehicle type.

This optimized network configuration was then subjected to the test set with 100 traffic patterns, and the results of this test are detailed in Figures 7.6-7.9.

Comparisons with the results from this test, the results from the neural network with a single vehicle type, and the data from Chapter 6 are presented in Table 7.1.

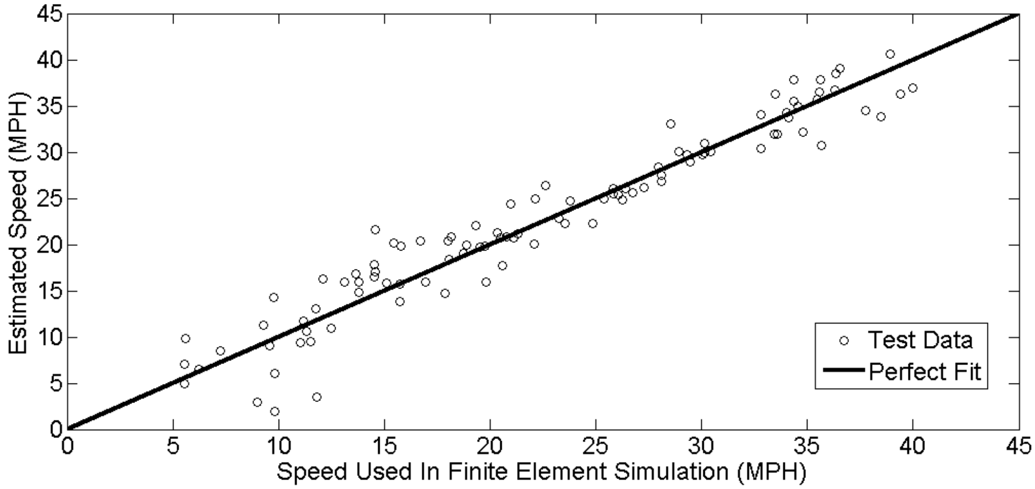


Figure 7.6: Comparison of Simulated Speed versus Value Estimated by Neural Network

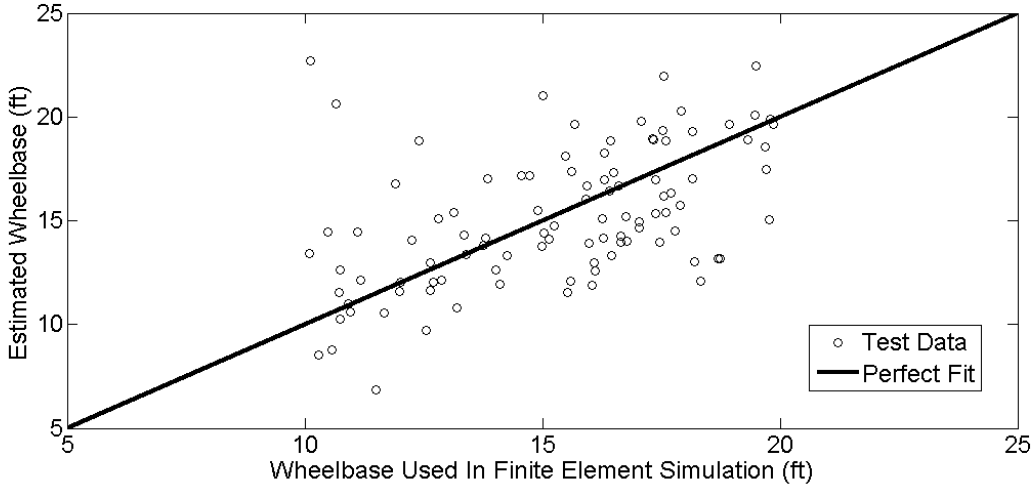


Figure 7.7: Comparison of Simulated Wheelbase versus Value Estimated by Neural Network

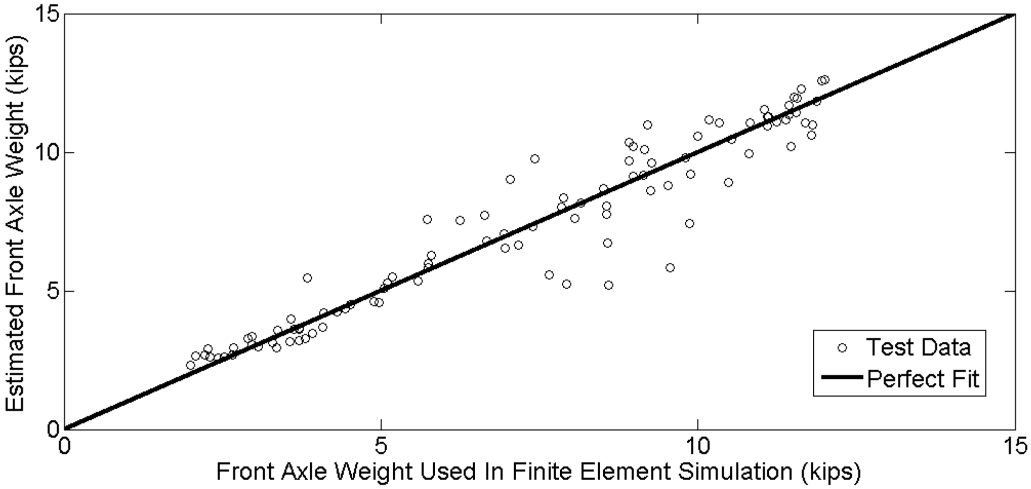


Figure 7.8: Comparison of Simulated Front Axle Weight versus Value Estimated by Neural Network

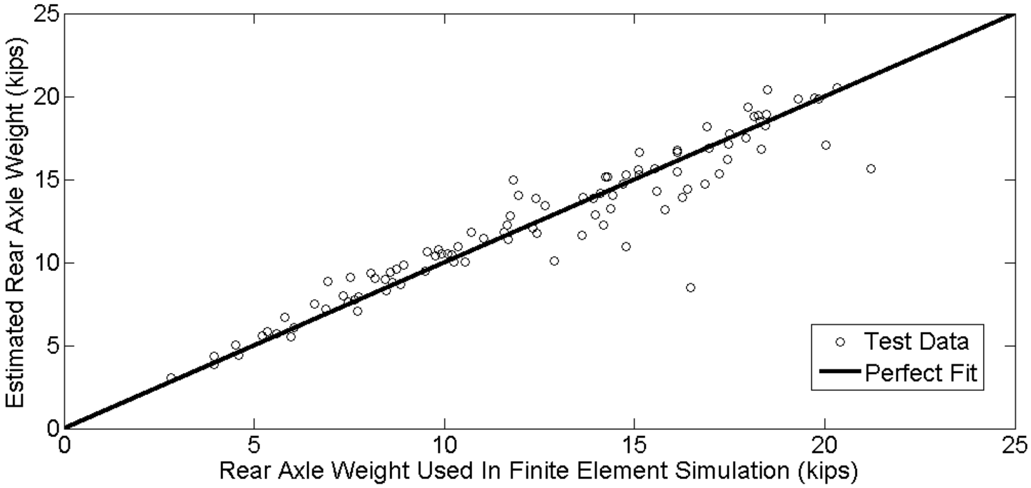


Figure 7.9: Comparison of Simulated Rear Axle Weight versus Value Estimated by Neural Network

Table 7.1: Comparison of Identified Vehicle Properties for Neural Network with Single Vehicle Type, Neural Network with Random Vehicle Type, Random Vehicle Type from Chapter 6.

Vehicle Property	Average Error for Single Vehicle Type with Neural Network	Average Error for Random Vehicle Type with Neural Network	Average Error for Random Vehicle Type from Chapter 6
Speed	0.28 MPH	1.94 MPH	1.87 MPH
Wheelbase	0.30 ft	2.23 ft	2.72 ft
Front Axle Weight	-	0.63 kips	-
Rear Axle Weight	-	0.94 kips	-

7.3.2 Peak Detection for Neural Network Inputs

The peak detection algorithm used in Chapter 6 was applied to the simulated strain time histories from recording elements 12 and 28. Using this algorithm, the arrival time and corresponding peak strain for each of the vehicle’s axles were determined. Figure 7.10 shows the two strain time histories for a typical vehicle with the extracted values summarized in Table 7.2.

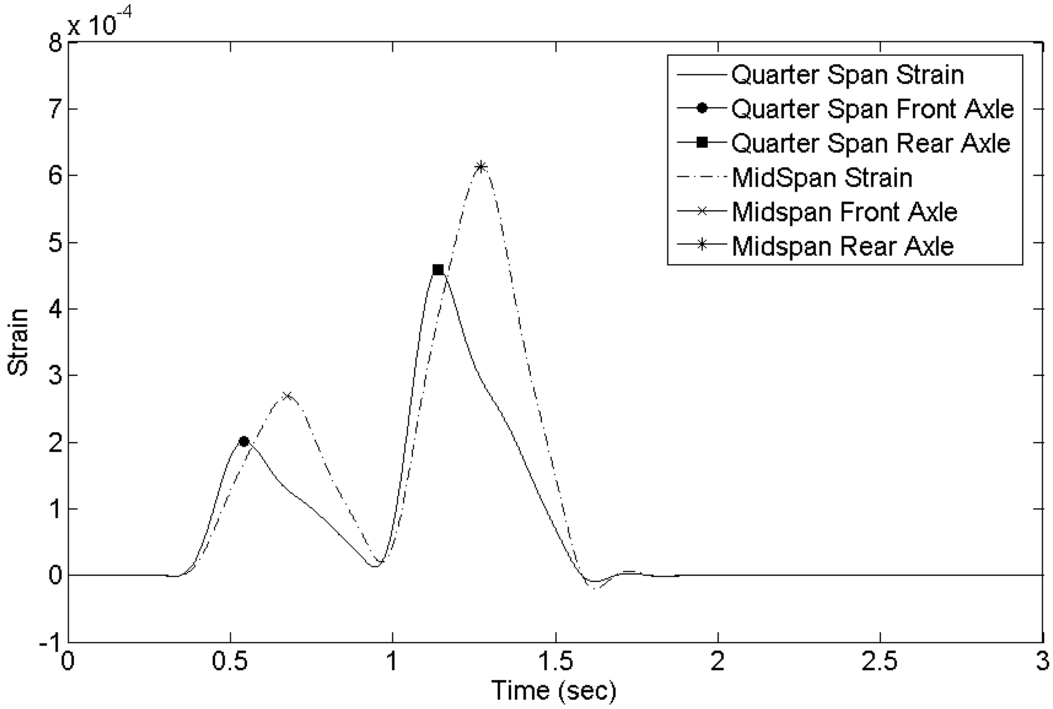


Figure 7.10: Strain Time Histories for Typical 2-axle Vehicle Showing Axle Arrivals Determined from Peak Detection Algorithm

Table 7.2: Extracted Features for Typical 2-Axle Vehicle

1st Axle Arrival at Element 12	0.544 sec
Element 12 Peak Strain from 1st Axle	2.0143 x 10^{-4}
2nd Axle Arrival at Element 12	1.141 sec
Element 12 Peak Strain from 2nd Axle	4.5903 x 10^{-4}

1st Axle Arrival at Element 28	0.674 sec
Element 28 Peak Strain from 1st Axle	2.6971 x 10^{-4}
2nd Axle Arrival at Element 28	1.272 sec
Element 28 Peak Strain from 2nd Axle	6.1243 x 10^{-4}

Using these eight extracted features (which were normalized such that the minimum and maximum ranged from -1 to 1) as inputs, a new set of neural networks were trained and used for estimating the speed, wheelbase, front axle weight, and rear axle weights for a test data set. The results are included in Figures 7.11-7.14, and summarized in Table 7.3. Overall, the results have considerably improved over all the previous methods.

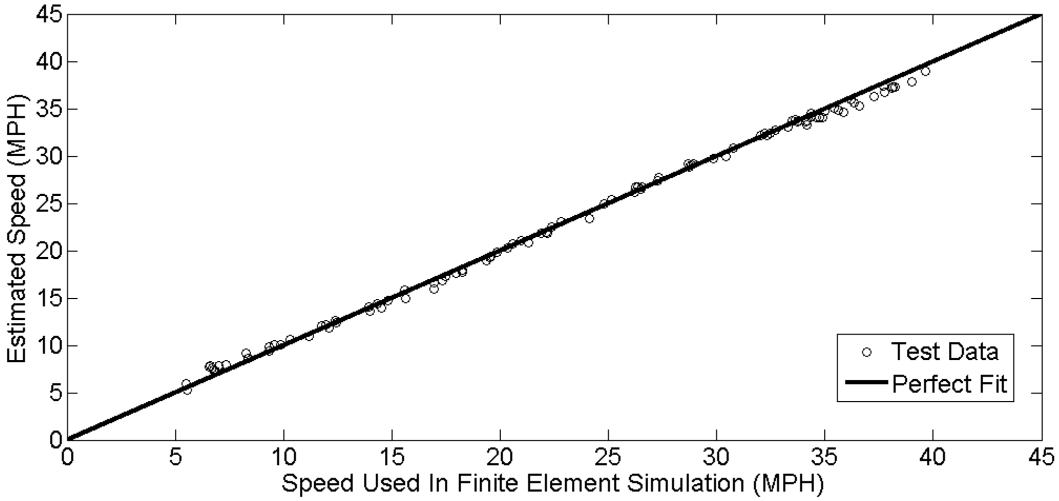


Figure 7.11: Comparison of Simulated Speed versus Value Estimated by Neural Network using Extracted Features from Peak Detection Algorithm

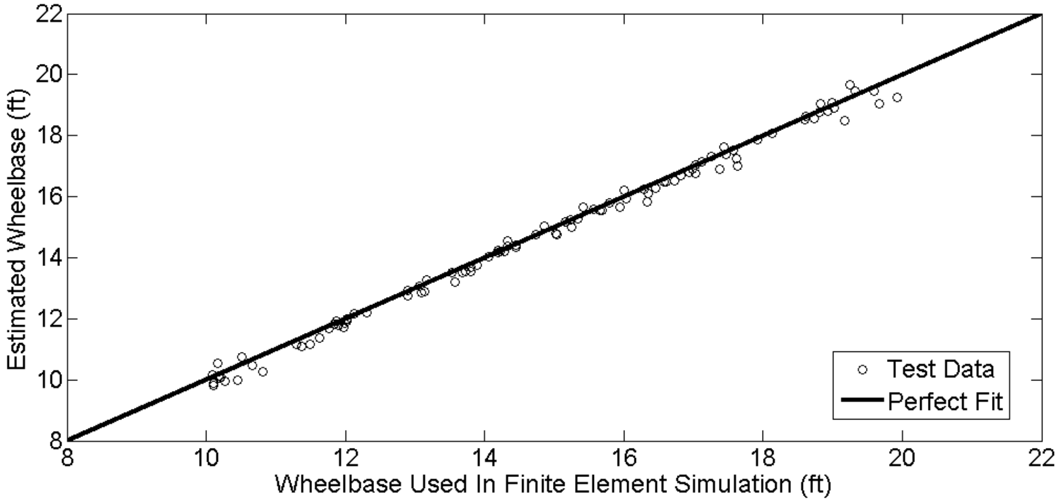


Figure 7.12: Comparison of Simulated Wheelbase versus Value Estimated by Neural Network using Extracted Features from Peak Detection Algorithm

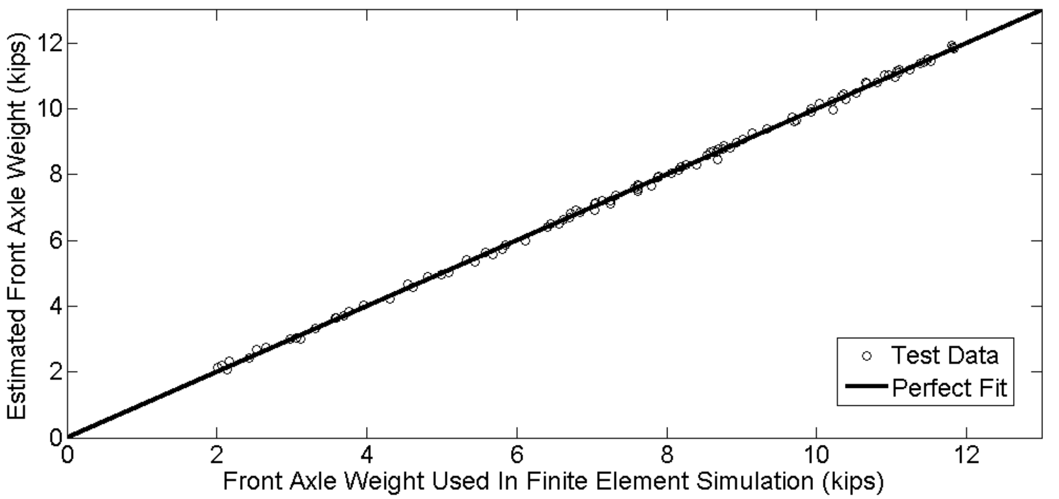


Figure 7.13: Comparison of Simulated Front Axle Weight versus Value Estimated by Neural Network using Extracted Features from Peak Detection Algorithm

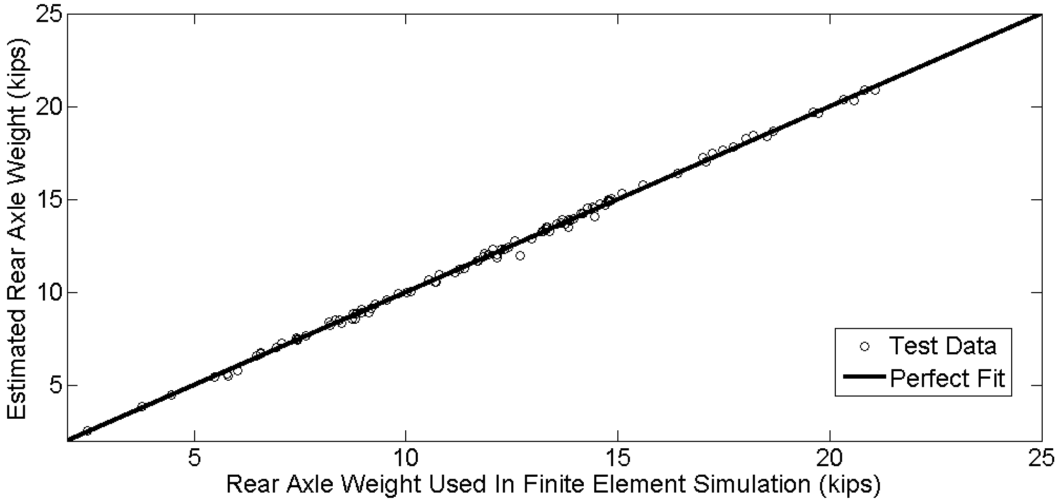


Figure 7.14: Comparison of Simulated Rear Axle Weight versus Value Estimated by Neural Network using Extracted Features from Peak Detection Algorithm

Table 7.3: Average Error for Vehicle Properties Estimated from Neural Networks using PCA and Peak Detection Based Feature Extraction

Vehicle Property	Average Error for Random Vehicle Type with Neural Network using PCA	Average Error for Random Vehicle Type with Neural Network using Peak Detection
Speed	1.94 MPH	0.42 MPH
Wheelbase	2.23 ft	0.17 ft
Front Axle Weight	0.63 kips	0.07 kips
Rear Axle Weight	0.94 kips	0.128 kips

7.4 Damage Detection Using Neural Networks and One-Dimensional Finite Element Model with Random Traffic

The pattern recognition abilities of neural networks make them an attractive tool for damage detection. Like any supervised learning method, neural networks require data from the undamaged and damaged states of the structure for training. While typically scarce for actual civil engineering structures, these can be rather easily (yet tediously) simulated using the one-dimensional finite element model.

Having generated the 300 traffic patterns previously discussed, the next decision is which damage sensitive features should be used as input for the neural network. Previously, Principal Components Analysis was used for selecting features for determining vehicle properties with the neural networks. For damage detection, rather than applying Principal Components Analysis for feature reduction, it was decided to instead use only the peak strains [Liu and Sun, 1997] from the 16 recording elements shown in Figure 5.15. This method affectively reduces the number of features from 64,000 (4,000 time steps for each of the 16 recording elements) to just 16. While all the information in the original time histories is not preserved, this method makes use of strains inherent ability to detect local changes in the system and provides local data from along the full length of the bridge decks.

Further data cleansing was performed and found to significantly improve the results. This cleansing consisted of applying a 5th order lowpass Butterworth filter with a 5 Hz cutoff frequency prior to determining the peak strains. This filter served to smooth the data and remove any free vibration about the approximate 20 Hz fundamental frequency. In addition, the data was normalized by the peak strain recorded at element 24. This served to remove variations in the peak strain due to the random axle weights.

Once the patterns were defined and the damage sensitive features selected, a multi-class neural network was constructed and trained for locating the presence of damage on the simulated bridge deck system. This network was defined with the 16 input units (corresponding to the peak filtered strains normalized by element 24), 32 hidden units (chosen such that the number of hidden units was twice the number of input features), and 4 output units. The 4 output units (summarized in Table 7.4) correspond to the presence of damage at one of the three local damage zones or to the case of no damage present. Of the 300 traffic patterns, 200 were used for training, 40 for validation (used as an objective standard to quantify the error in the neural network output), and 60 for testing.

As when used for vehicle property estimation, the backpropagation learning algorithm was again used to train and optimize the network. The optimized network was then applied to the 60 test records (Fig. 8.9) and the results are shown in Figures 7.15 – 7.18 and Table 7.5.

Table 7.4: Summary of Output Classes

Output Class	Damage Description
1	Local Damage in Zone 1
2	Local Damage in Zone2
3	Local Damage in Zone 3
4	Undamaged

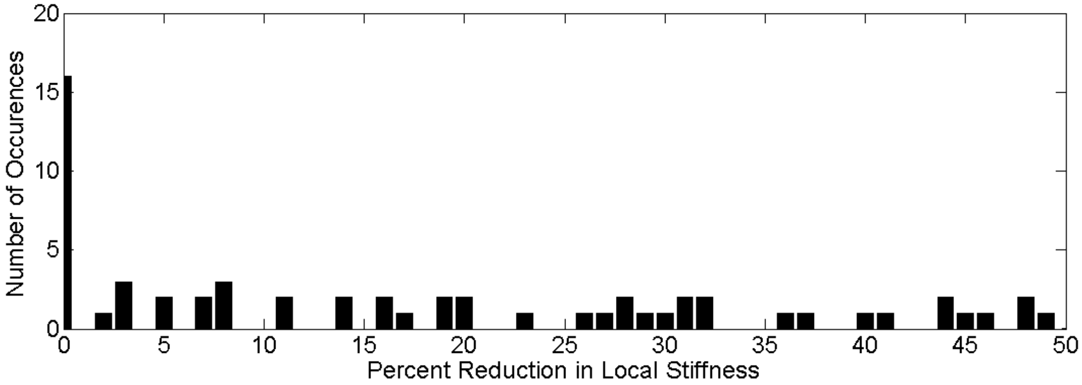


Figure 7.15: Histogram of Percent Reduction in Local Stiffness for Test Data

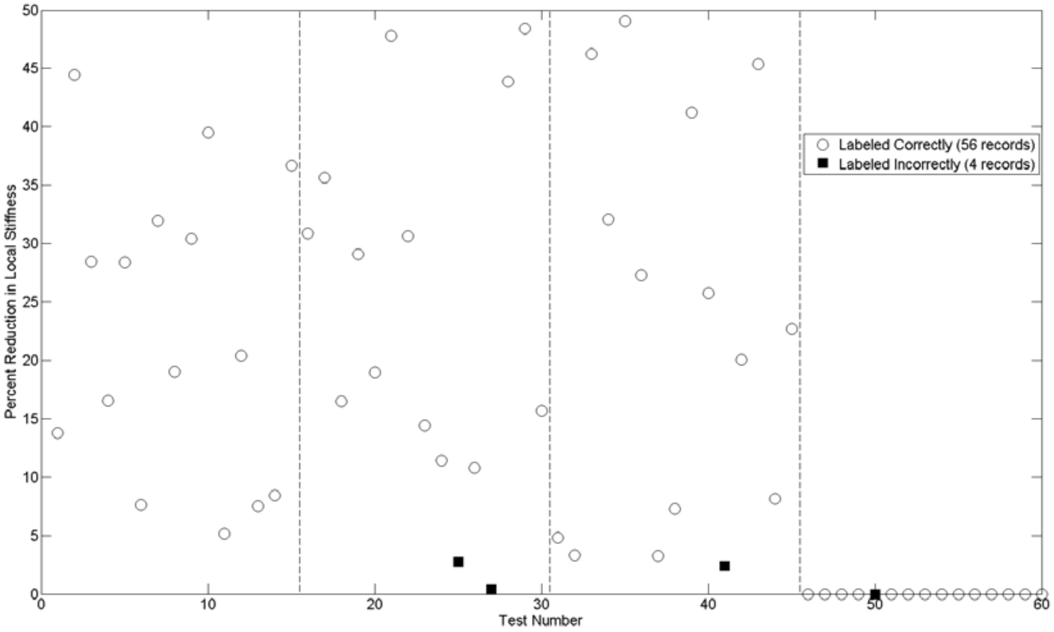


Figure 7.16: Test Number versus Percent Reduction in Local Stiffness (Using Maximum Strain Distributions Filtered and Normalized by Element 24).

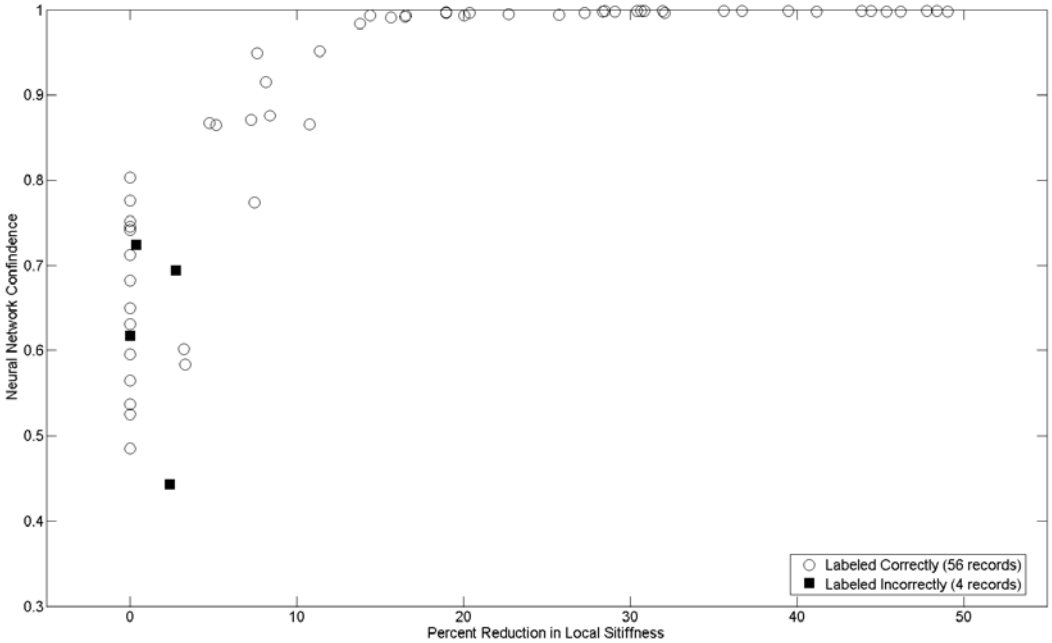


Figure 7.17: Percent Reduction in Local Stiffness versus Neural Network Confidence

Table 7.5: Neural Network Success Rate for Identifying Damage by Damage Class

	Damage Classification				
	1	2	3	4	Total
Number of Successes	15	13	14	14	56
Number of Failures	0	2	1	1	4

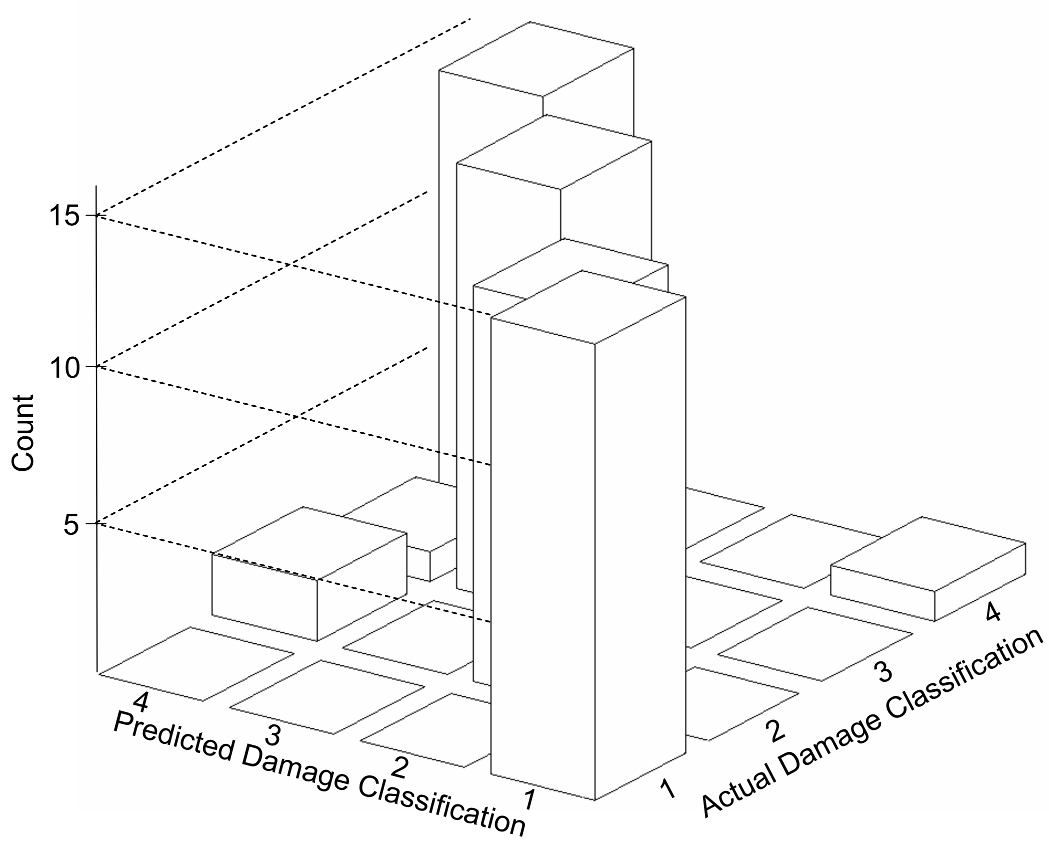


Figure 7.18: 3-D Histogram of Actual Classification versus Classification Assigned By Neural Network.

Overall, this optimized neural network was able to successfully classify 93.3% of the test scenarios, identifying all damage scenarios where the local stiffness reduction was greater than 3%. By filtering the data prior to normalizing the strain distributions, the classification accuracy increased by 5% and the damage threshold for successful identification improved from 8% with the unfiltered strains (Fig. 7.19) to 3% for filtered (Fig. 7.16). In comparison to the results from Chapter

6, the detection threshold has dropped from 15% local damage to 3%, and the overall accuracy has increased from 73.6% to 93.3%.

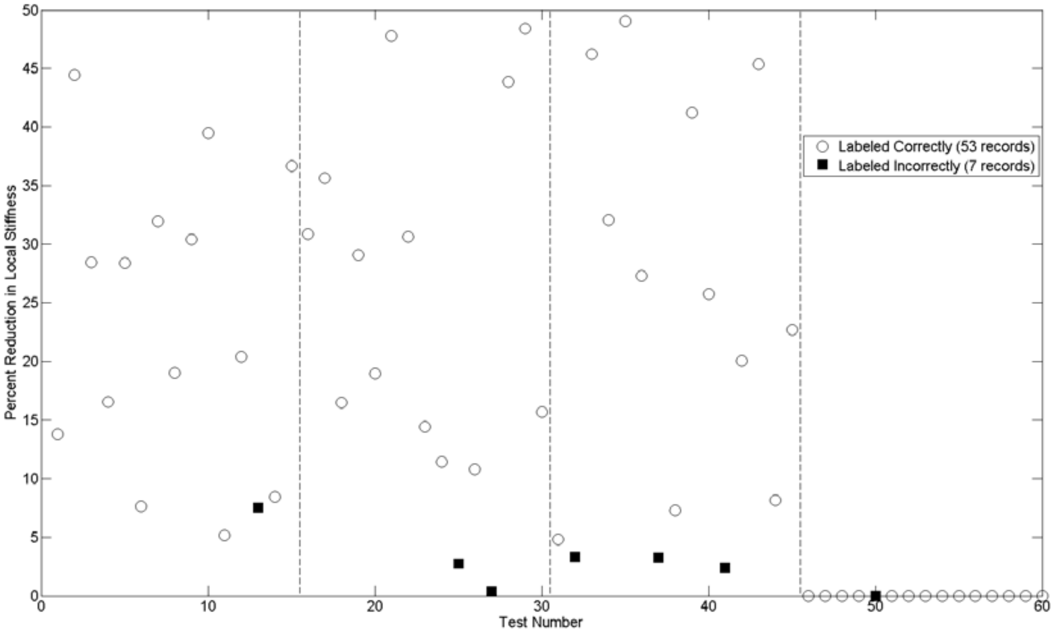


Figure 7.19: Test Number versus Percent Reduction in Local Stiffness for Unfiltered Maximum Strain Distributions (Normalized by Element 24).

7.5 Damage Classification Using Neural Networks and One-Dimensional Finite Element Model with Random Traffic

Having demonstrated the neural networks ability to locate the presence of damage from the simulated data, the next logical step was to apply neural networks for classifying the level of the damage. The most straightforward approach involves training a multiclass neural network where each class represents a range of damage levels for a particular local damage zone. In this example, three damage ranges (1 –

15%, 15 - 30%, and 30 – 50% reduction in Young's Modulus) were considered for each of the three local damage zones. Consequently, ten classes were used (9 damaged and 1 undamaged).

The previously used data set composed of 300 strain patterns provides approximately 25 patterns for each of the damaged classes and 75 undamaged patterns. These sets of 25 patterns were too few for training, validation, and testing so an additional 300 patterns were generated. The properties of this new combined data set (composed of 600 random vehicles) are detailed in figures 7.20 – 7.24.

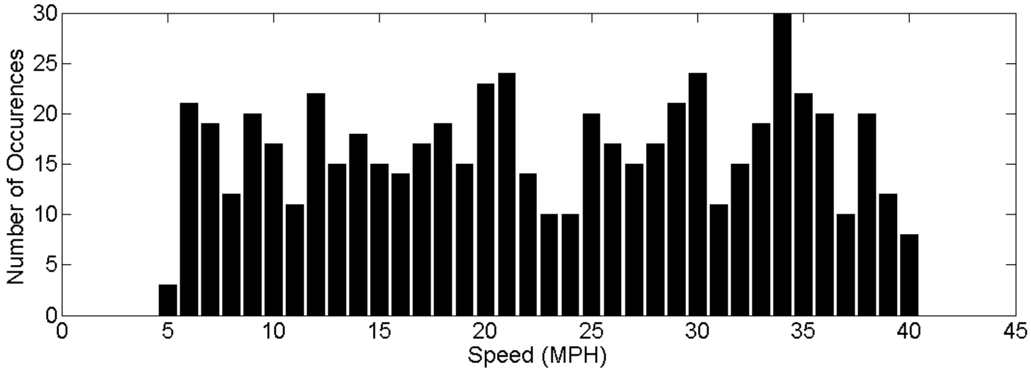


Figure 7.20: Histogram of Simulated Vehicle Speeds

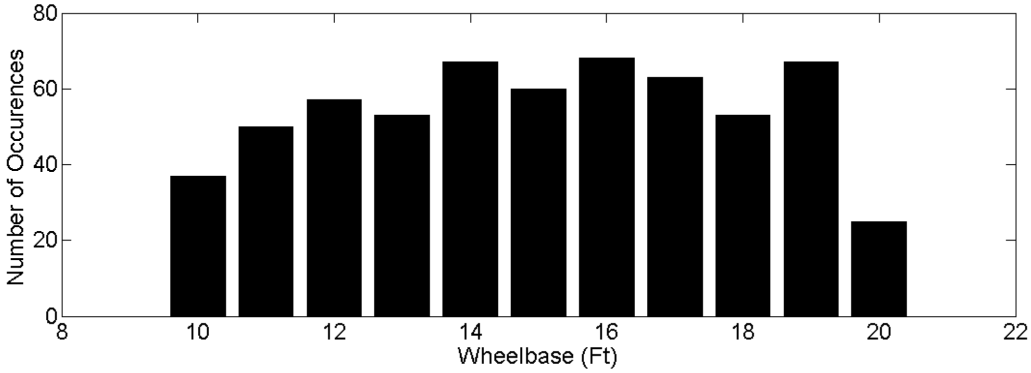


Figure 7.21: Histogram of Simulated Vehicle Wheelbases

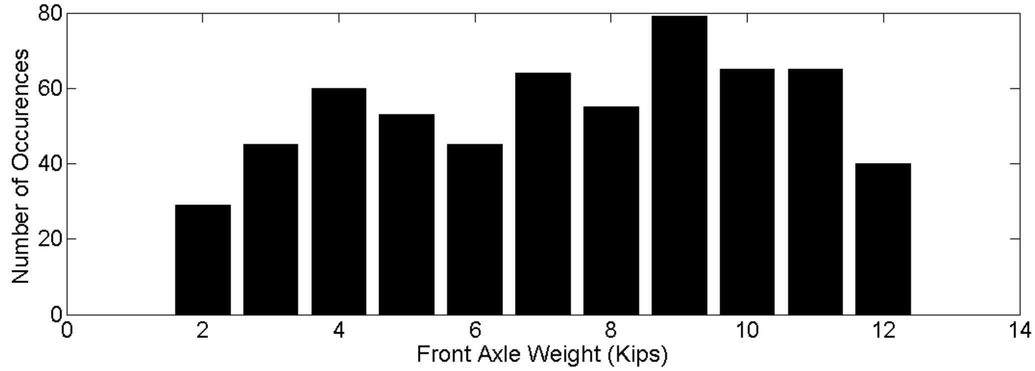


Figure 7.22: Histogram of Simulated Vehicle Front Axle Weights

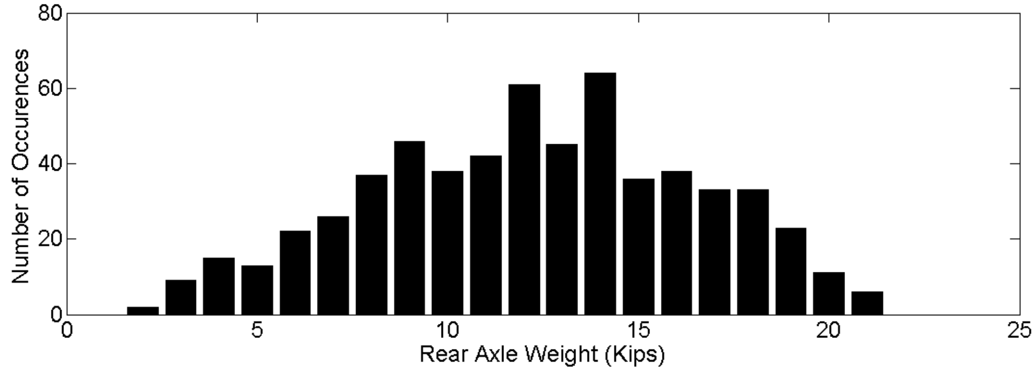


Figure 7.23: Histogram of Simulated Vehicle Rear Axle Weights

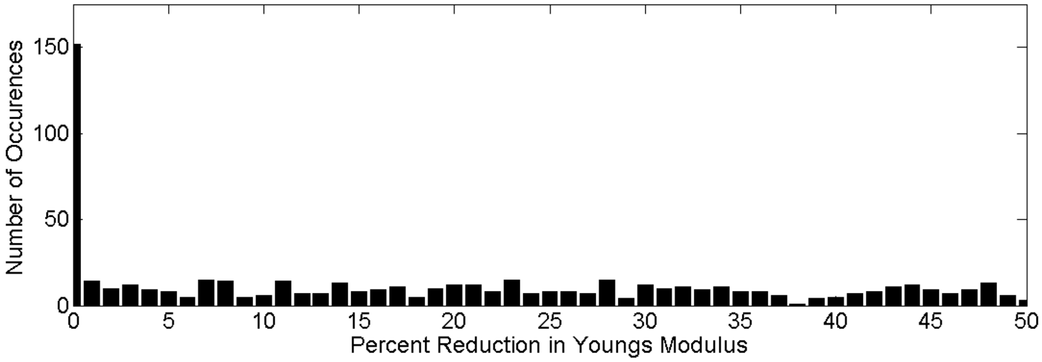


Figure 7.24: Histogram of Percent Reduction in Young's Modulus

350 of the patterns were selected for training and 100 patterns each for validation and testing. To prevent biasing of the training set resulting from adding proportionally more undamaged scenarios, fifty of the undamaged patterns were discarded. The histogram of the damage classes for the training data is shown in Fig 7.25.

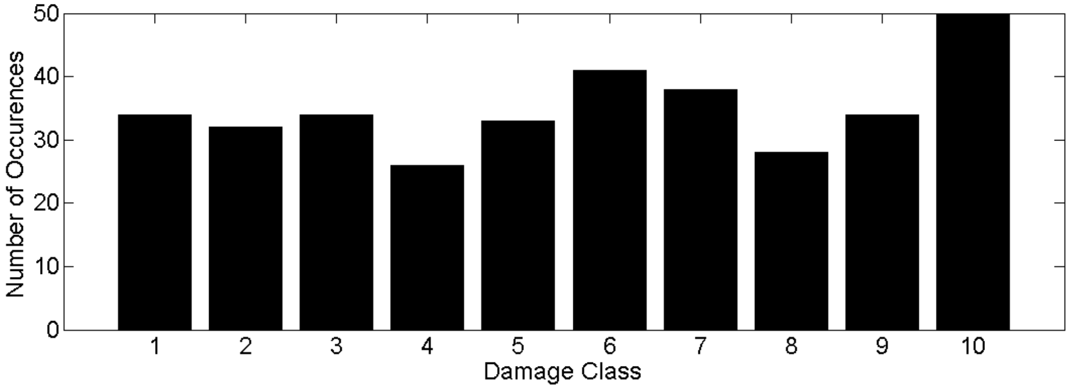


Figure 7.25: Histogram of Training Data Damage Classes

The neural network was defined with the 16 input units (corresponding to the peak filtered strains normalized by element 24), 32 hidden units, and 10 output units (each output corresponding to one damage class). The 10 output units are summarized in Table 7.6. Once the network architecture was determined, the backpropagation learning algorithm was again used to train and optimize the network. The optimized network was then applied to the 100 test records (Fig. 7.26) and the results are shown in Figures 7.27 – 7.29 and Table 7.7.

Table 7.6: Summary of 10 Output Units Utilized in Damage Classification

Output Unit Number	Description of Damage
1	1-15% Reduction in Stiffness for all Elements in Zone 1
2	15-30% Reduction in Stiffness for all Elements in Zone 1
3	30-50% Reduction in Stiffness for all Elements in Zone 1
4	1-15% Reduction in Stiffness for all Elements in Zone 2
5	15-30% Reduction in Stiffness for all Elements in Zone 2
6	30-50% Reduction in Stiffness for all Elements in Zone 2
7	1-15% Reduction in Stiffness for all Elements in Zone 3
8	15-30% Reduction in Stiffness for all Elements in Zone 3
9	30-50% Reduction in Stiffness for all Elements in Zone 3
10	Undamaged

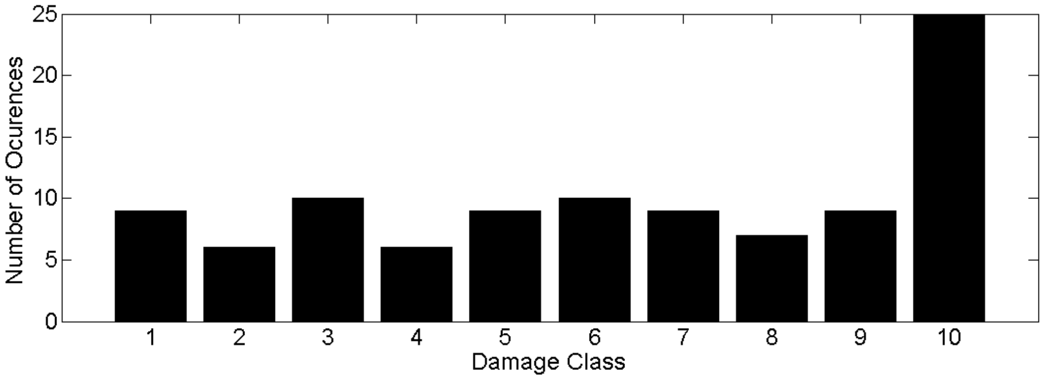


Figure 7.26: Histogram of Test Data Damage Classes

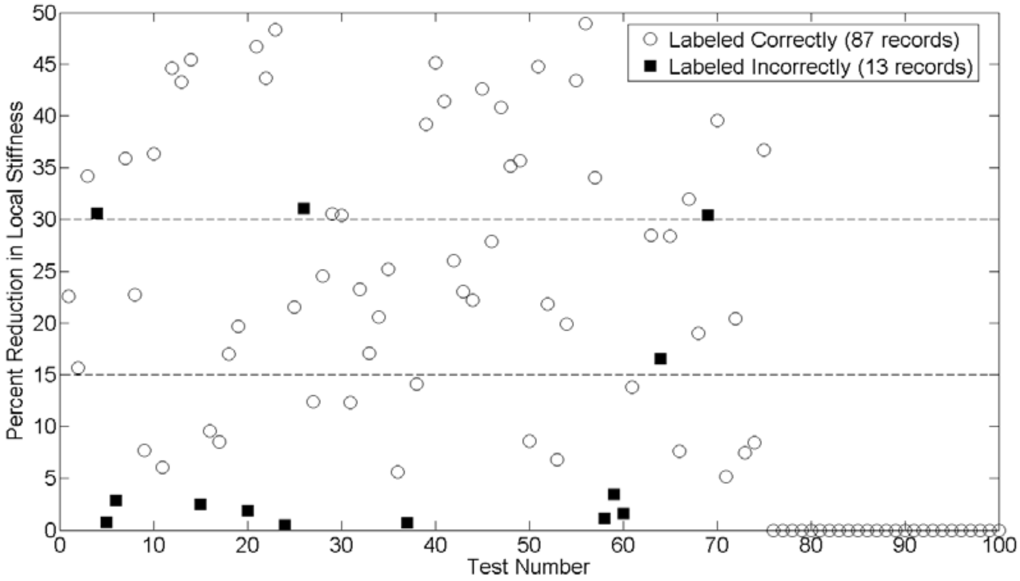


Figure 7.27: Percent Reduction in Local Stiffness versus Neural Networks Ability to Successfully Classify Data

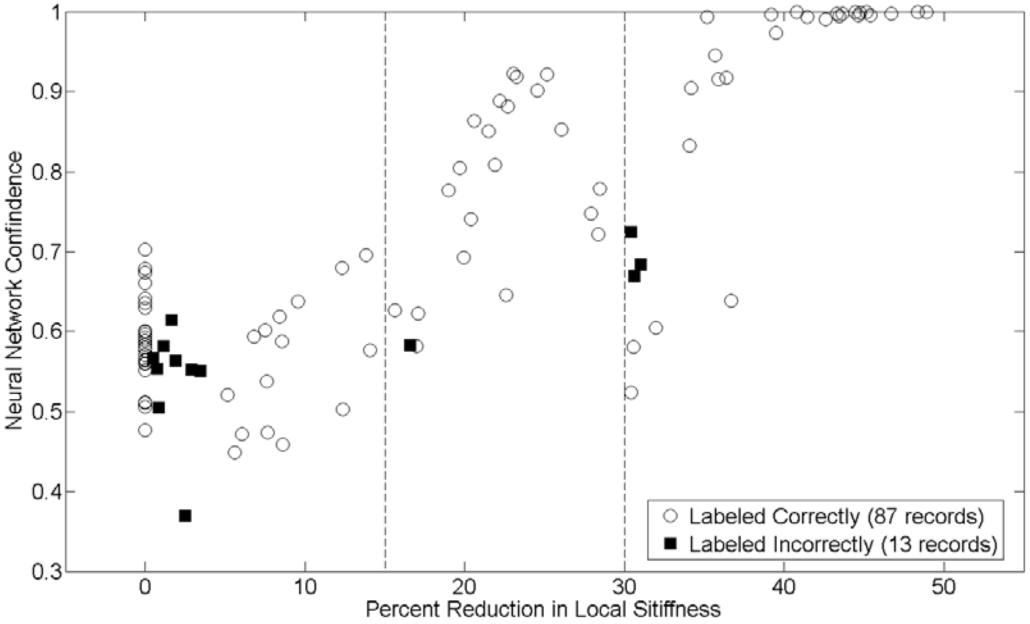


Figure 7.28: Percent Reduction in Local Stiffness versus Neural Network Confidence

Table 7.7: Neural Network Success Rate for Classifying Damage by Output Unit

	Output Unit										
	1	2	3	4	5	6	7	8	9	10	Total
Number of Successes	4	6	9	5	9	9	6	6	8	25	87
Number of Failures	5	0	1	1	0	1	3	1	1	0	13

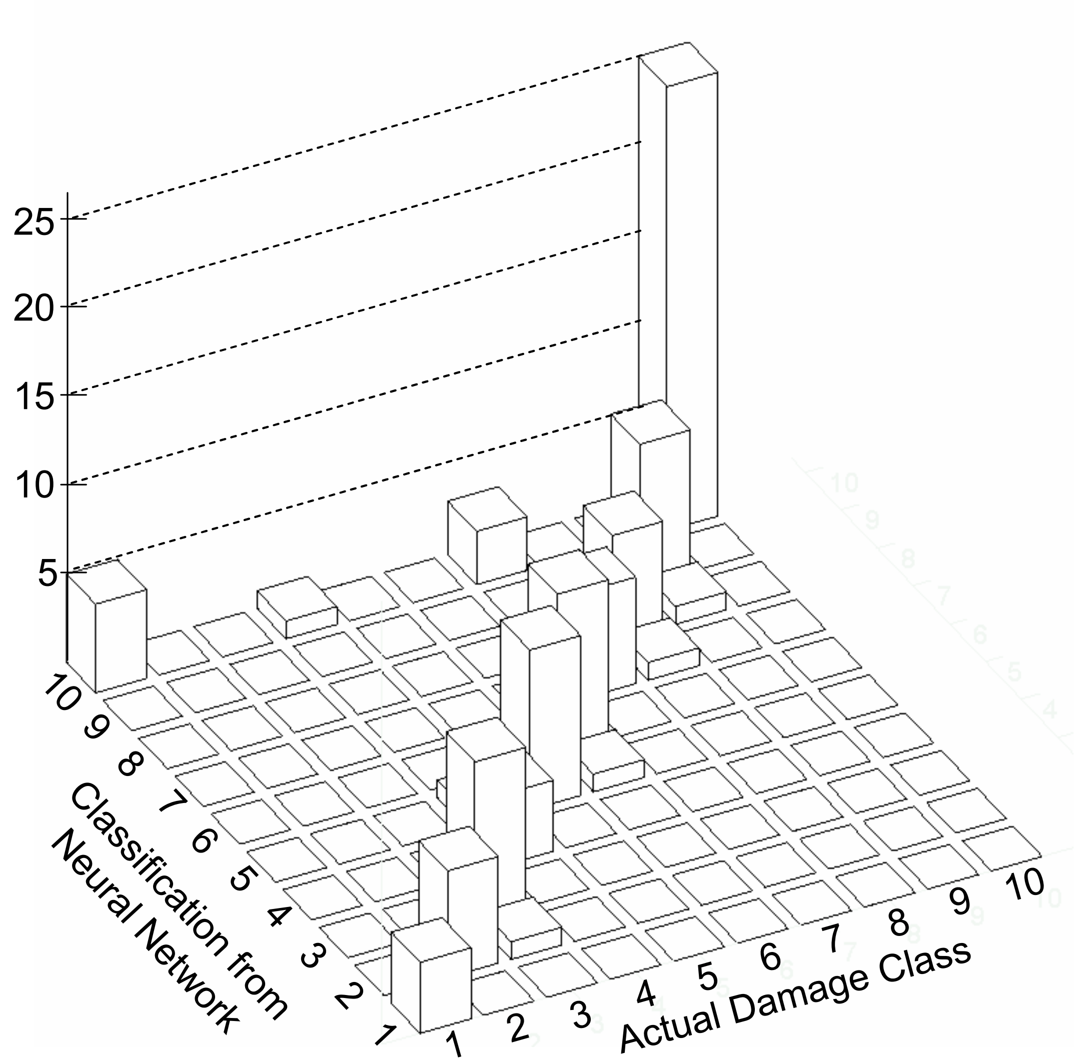


Figure 7.29: 3-D Histogram of Actual Classification versus Classification Assigned By Neural Network.

From Figures 7.29 and 7.27, there are a few general results which may be drawn:

1. For the lightly damaged classes (classes 1, 4, and 7), the neural network had a tendency to assign, incorrectly, the undamaged class (class 10).

Only for the lightly damaged cases, did the neural network incorrectly assign the undamaged label.

2. For the incorrect labeling (with the exception of when the neural network assigns the undamaged label), the neural network always assigns a neighboring label. This is indicative of having trouble near the boundaries between the labels and is confirmed in Fig 7.27.
3. There were no false indications of damage whereby a damaged label was assigned to the undamaged system.
4. The majority of the incorrect labels were associated with the lightly damaged scenarios.

7.6 Damage Detection Using Neural Networks and One-Dimensional Finite Element Model with Random Traffic and Simultaneous Damage Locations.

The next application involves the use of the neural network for damage detection when damage is allowed to occur simultaneously at multiple locations. The various damage scenarios are summarized in Table 7.8. As with the previous example, additional traffic patterns were necessary for training, validation, and testing. It was decided to generate 200 patterns for each damage scenario and to assign 175 patterns from each scenario for training, 25 for validation and 25 for testing. Ultimately, 1600 random traffic patterns were utilized herein, the properties of which are summarized in Figures 7.30 – 7.32.

Three multi-class neural networks were constructed and trained for locating the presence of damage in one of the local damage zones on the simulated bridge deck system. Again, these networks were defined with the 16 input units (corresponding to the peak filtered strains normalized by element 24) and 32 hidden units (chosen such that the number of hidden units was twice the number of input features), and 2 outputs (either damaged or undamaged). Each testing pattern (composed of a set of normalized peaks strains from the 16 recording elements) was applied individually to the three neural networks and the results are shown in Figure 7.33 and summarized in Table 7.9. 98.0% of the testing patterns were correctly identified for Zone 1, 98.5% for Zone 2, and 96.5% for Zone 3. Overall, the neural network correctly identified 93.5% of the combined damage patterns. All but two of the errors are associated with very low damage levels (less than 2%) being misclassified as undamaged.

Table 7.8: Summary of Damage Scenarios

Damage Scenario	**Number of Patterns**	**Simulation Numbers**
Undamaged	200	1-200
Zone 1	200	201-400
Zone 2	200	401-600
Zone 3	200	601-800
Zones 1 & 2	200	801-1000
Zones 2 & 3	200	1001-1200
Zones 1 & 3	200	1201-1400
Zones 1, 2, & 3	200	1401-1600

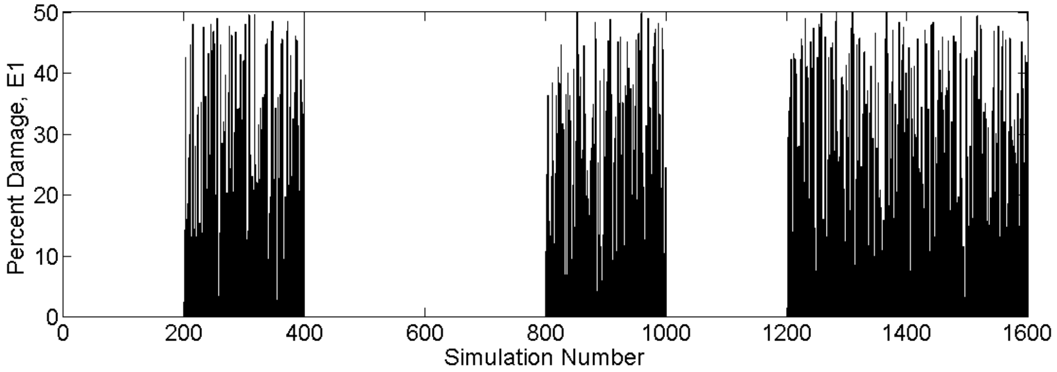


Figure 7.30: Simulation Number versus Percent Reduction in Stiffness for All Elements in Zone 1

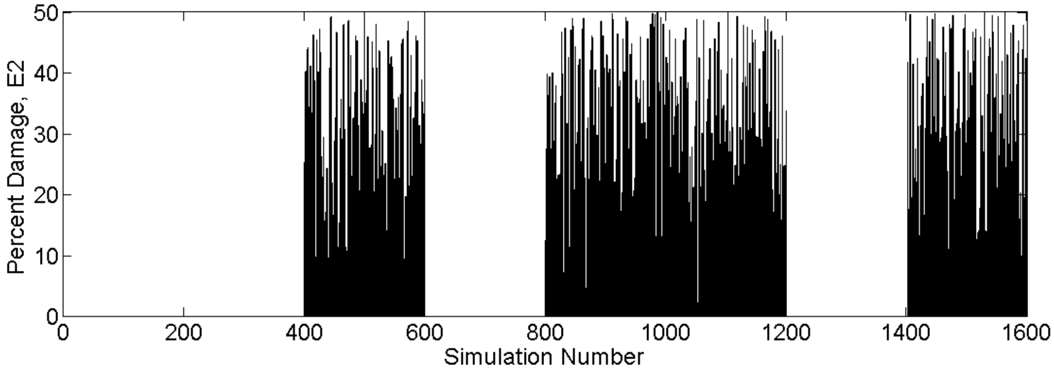


Figure 7.31: Simulation Number versus Percent Reduction in Stiffness for All Elements in Zone 2

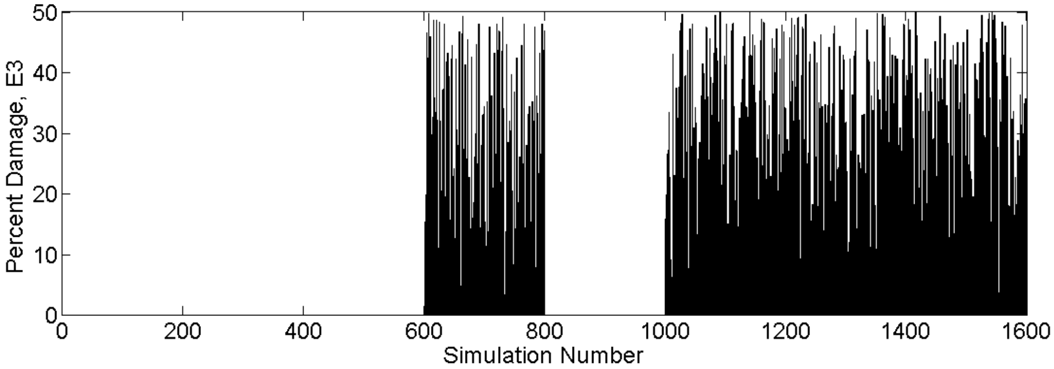


Figure 7.32: Simulation Number versus Percent Reduction in Stiffness for All Elements in Zone 3

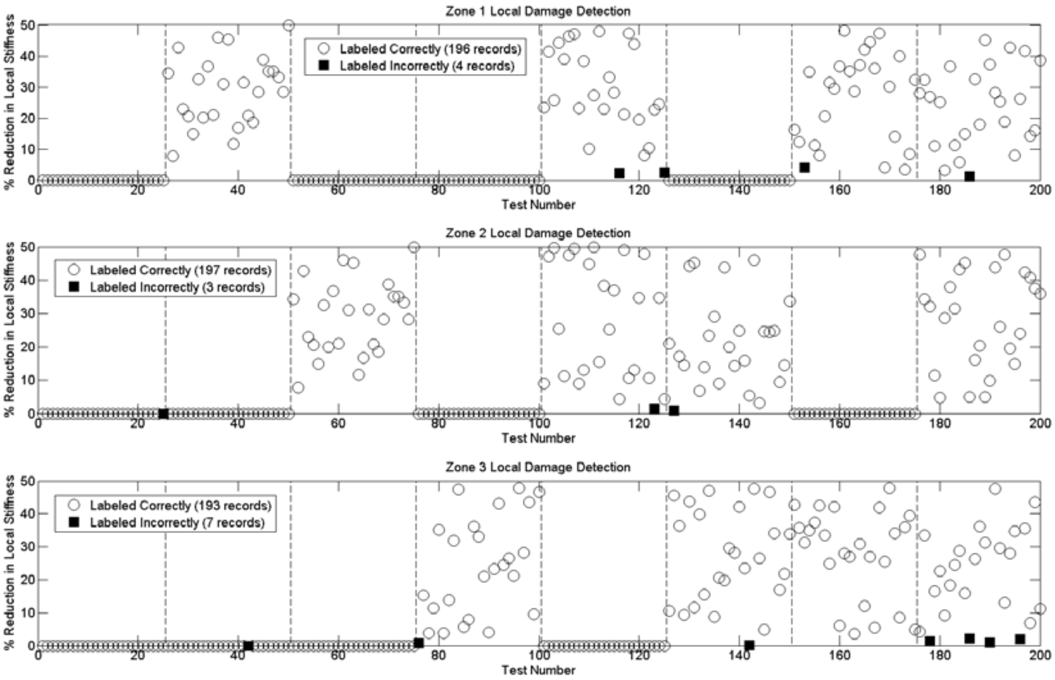


Figure 7.33: Results for Damage Detection

Table 7.9: Results from Damage Detection for Simultaneous Damage Locations

	Accuracy
Zone 1 Results	98.0%
Zone 2 Results	98.5%
Zone 3 Results	96.5%
Overall Results	93.5%

7.7 Damage Classification Using Neural Networks and One-Dimensional Finite Element Model with Random Traffic and Simultaneous Damage Locations.

The final application presented in this chapter involves the same aforementioned data set and three neural networks for classifying the level of damage in each of the local damage zones. Again, three neural networks were utilized, one responsible for each damage zone; however, this time each network had 4 output units (undamaged, 1-15% reduction in stiffness, 15-30% reduction in stiffness, and 30-50% reduction in stiffness).

Table 7.10: Summary of the 4 Output Units Used for Each Neural Network

Output Unit	Damage Description
1	Undamaged
2	1-15% Reduction in Stiffness for all Elements in Damage Zone
3	15-30% Reduction in Stiffness for all Elements in Damage Zone
4	30-50% Reduction in Stiffness for all Elements in Damage Zone

The results for each of the neural networks are presented in Fig's 7.34 – 7.39 and summarized in Table 7.11. From Figures 7.34 - 7.36, most of the incorrect labeling occurs near the boundaries of each class. From Fig's 7.37 -7.39, it can be seen that incorrect labels are assigned to their neighboring classes. Overall, the neural networks were able to accurately classify approximately 97% of the scenarios from each local damage zone and 90.5% of combined simultaneously occurring damage scenarios.

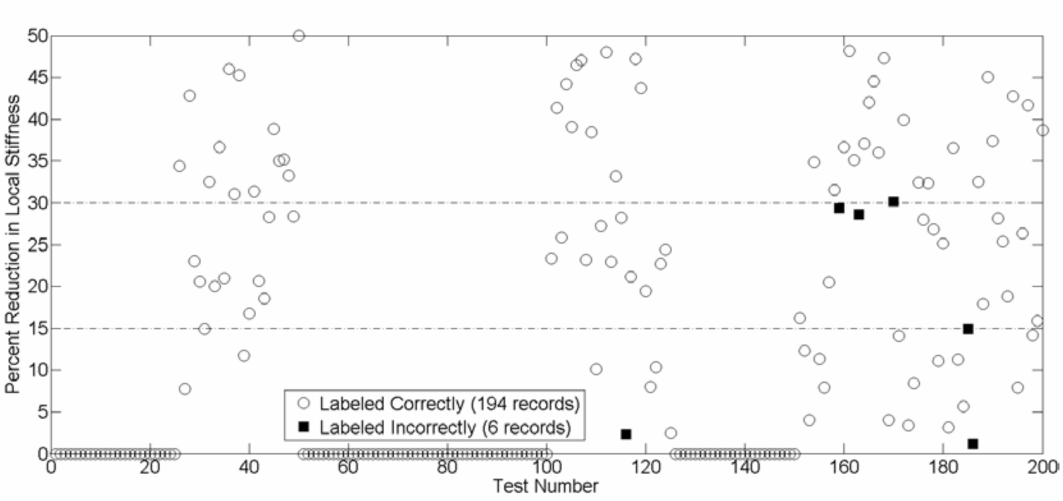


Figure 7.34: Results of Damage Classification for Local Damage Zone 1

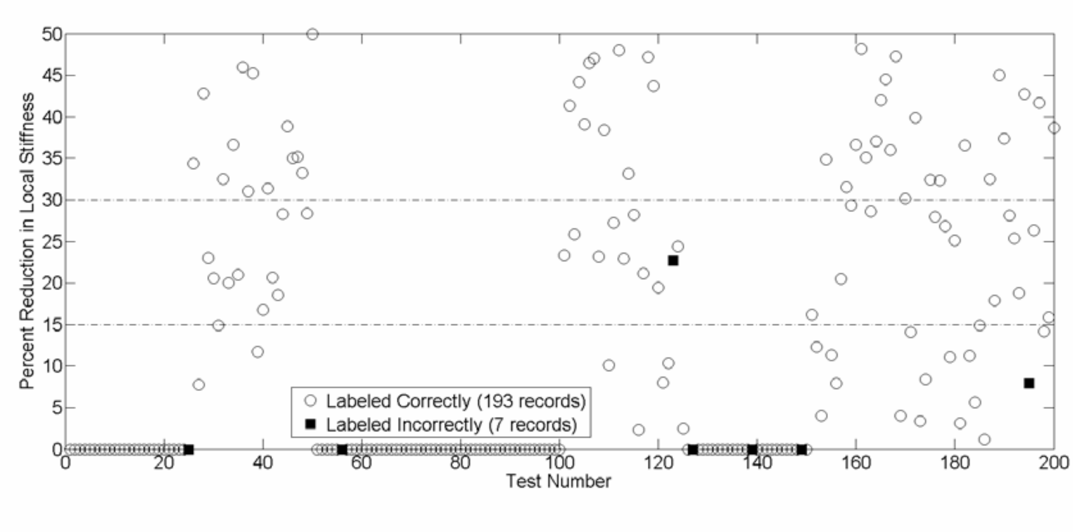


Figure 7.35: Results of Damage Classification for Local Damage Zone 2

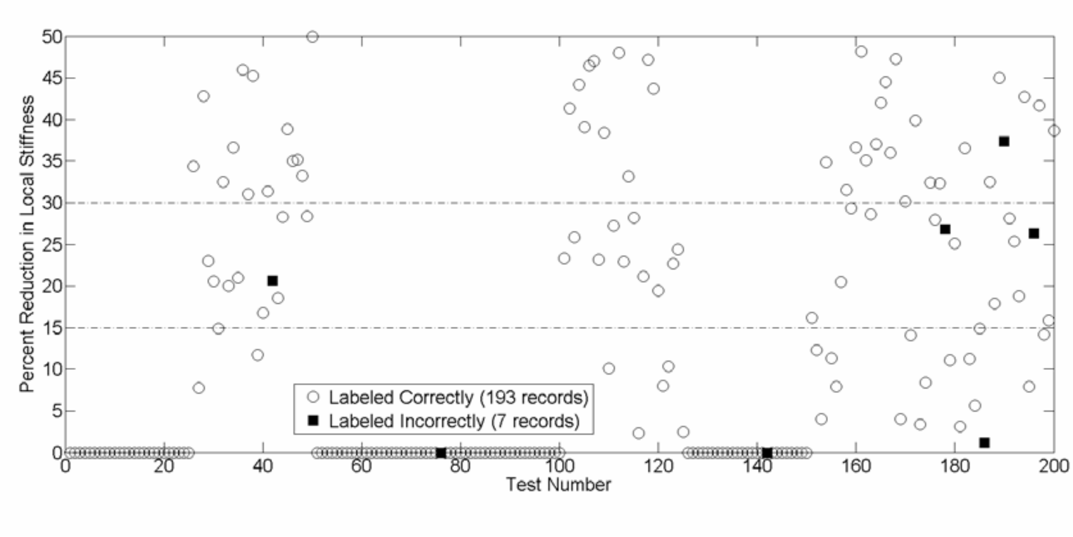


Figure 7.36: Results of Damage Classification for Local Damage Zone 3

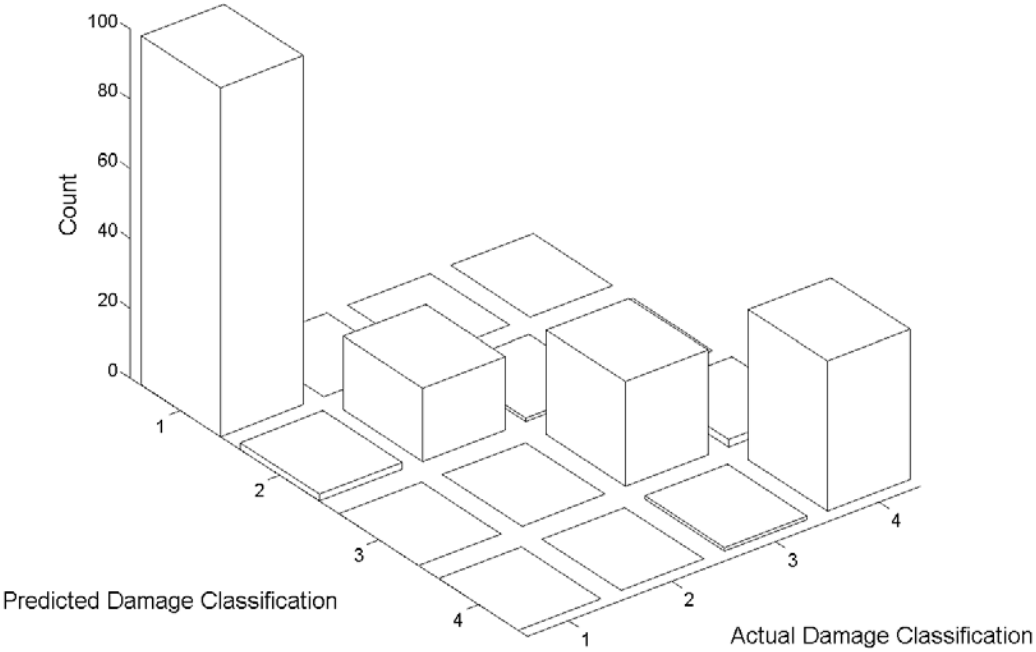


Figure 7.37: Neural Network Classifications for Local Damage Zone 1

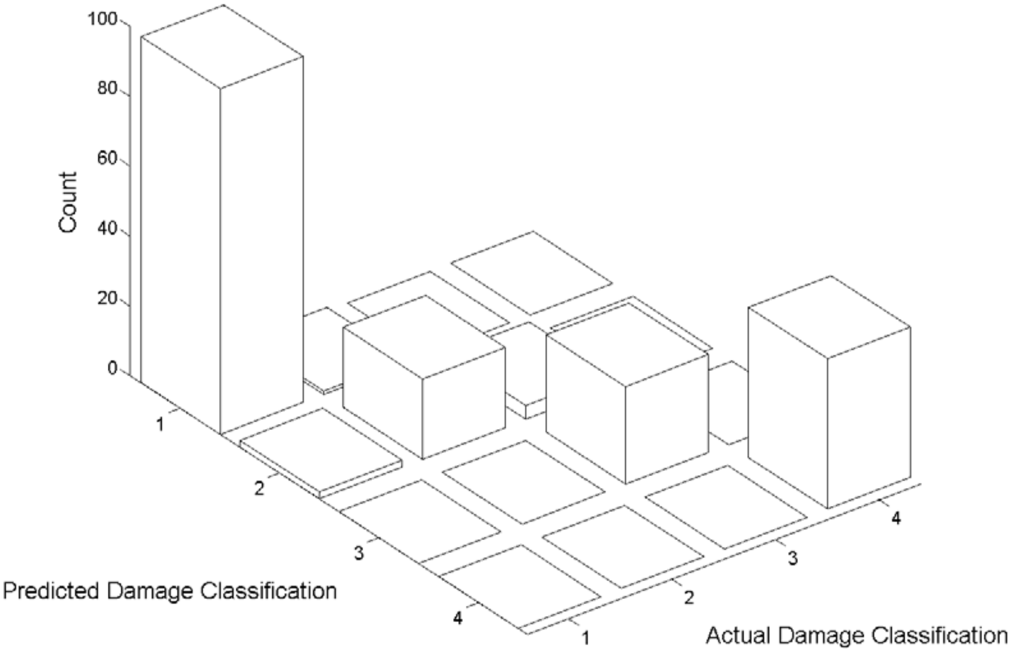


Figure 7.38: Neural Network Classifications for Local Damage Zone 2

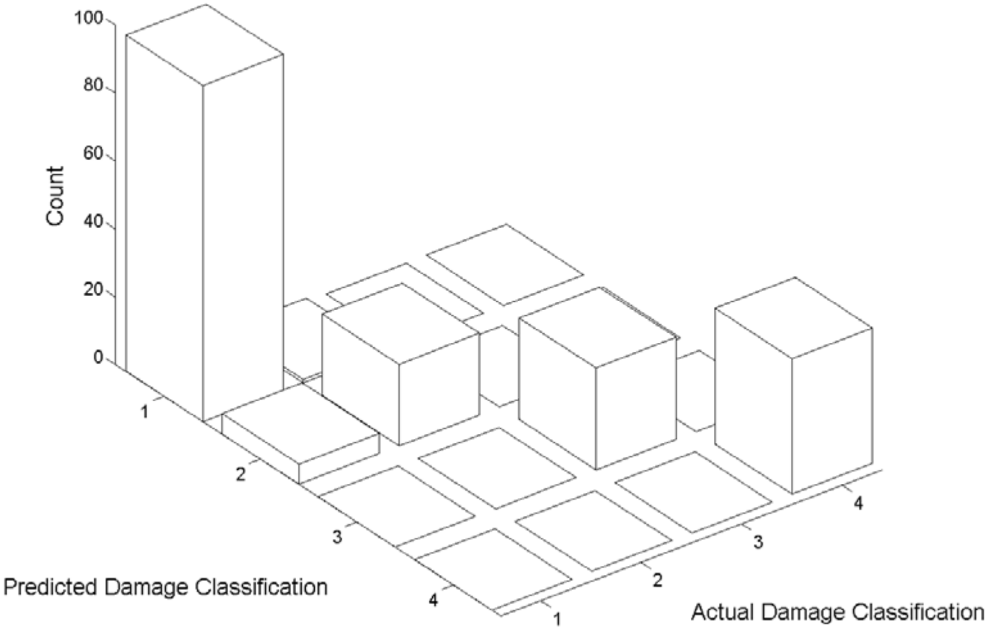


Figure 7.39: Neural Network Classifications for Local Damage Zone 3

Table 7.11: Summary of Damage Classification for Simultaneously Occurring Damage Scenarios

	Accuracy
Zone 1 Results	97.0%
Zone 2 Results	96.5%
Zone 3 Results	96.5%
Overall Results	90.5%

7.8 Utilization of a Predictive Neural Network for Damage Detection

While the aforementioned damage detection method was able to accurately detect and classify over 90% of the data used in the previous example, composed of 200 test scenarios. This method is limited in that it only provided a very rough judgment regarding the level of damage. The output could only place the damage in one of four classes (undamaged, 1-15% reduction in stiffness, 15-30% reduction in stiffness, or 30-50% reduction in stiffness). No information regarding where in these ranges the individual cases fell is available. In addition, this method suffers difficulties near the boundaries of the classes. As has been previously discussed, increasing the number of classes in order to decrease the uncertainty of the exact value of damage requires generating an immense amount of data, particularly when simultaneous damage locations are being considered. In fact, in these cases, the number of scenarios required grows exponentially with the number of output classes.

For these reasons, a new method of using neural networks for damage detection is presented. Ideally, this new method would:

1. Only use information (traffic patterns) from the healthy (undamaged) state of the structure for network training.
2. Be capable of detecting damage occurring simultaneously at multiple locations.
3. Provide an output that can be directly correlated with the damage level.

Only requiring data from the undamaged structure is an important advantage as the amount of data required during network training is much less than the in the previous method. Further, for deployment on actual structures, a calibrated FE model is no longer required. Instead, the neural network is simply trained on the data from the healthy structure.

Within this section, a method in which a predictive neural network is used for damage detection is presented. The premise is strain time histories taken at one location on a structure are fed into a series of neural networks which predict the strains at other locations along the structure. The error between these predicted strains and the actual values measured at these locations are then monitored. With this method, all of the training is done using data from the undamaged structure. When damage occurs (in one or more locations), the mappings employed by the neural networks responsible for the damage and undamaged locations no longer are valid. Consequently, the error between the predicted and measured strains increases and the magnitude of this error is utilized for assessing the extent of the damage.

An example is now presented using data generated from the one-dimensional finite element model of the composite bridge-deck panel system. In this example, the strain time histories from quarter span are used as inputs for a neural network. The neural network uses these inputs to output a prediction of the middle span strain time histories, which can then be compared to the actual values generated by the finite element model.

7.8.1 Principle Component Analysis of Strain Time Histories for Feature Reduction

Prior to applying the quarter span and midspan strain time histories generated from the finite element model to the neural network, Principle Component Analysis (PCA) was applied to reduce the number of features. In this application it was found through visual inspection of the reconstructed time histories that using only the PC's that preserved 99.0% percent of the variance in the data provided good results. For the 4-second long (4001 time steps) quarter span strain time histories, 22 PC's were required. Similarly, 21 PC's were used for midspan strains. Figures 7.40 a-d and 7.41 a-d, show both the quarter and middle span strain time histories for a typical vehicle moving at 5, 10, 15, 20, 25, 30, 35, and 40 miles-per-hour. The projections of the principle components for the same time histories are included in Figures 7.42 a-d and 7.434 a-d, respectively. All of the employed time histories in this section start 0.4 seconds before the front axle first makes contact with the bridge decks.

Figures 7.44 a-d and 7.45 a-d show the original quarter span strain time histories as well as the time histories reconstructed from the normalized principle components, using the inverse PCA transformation. Figures 7.76 a-d and 7.77 a-d detail the original and corresponding reconstructed midspan strain time histories.

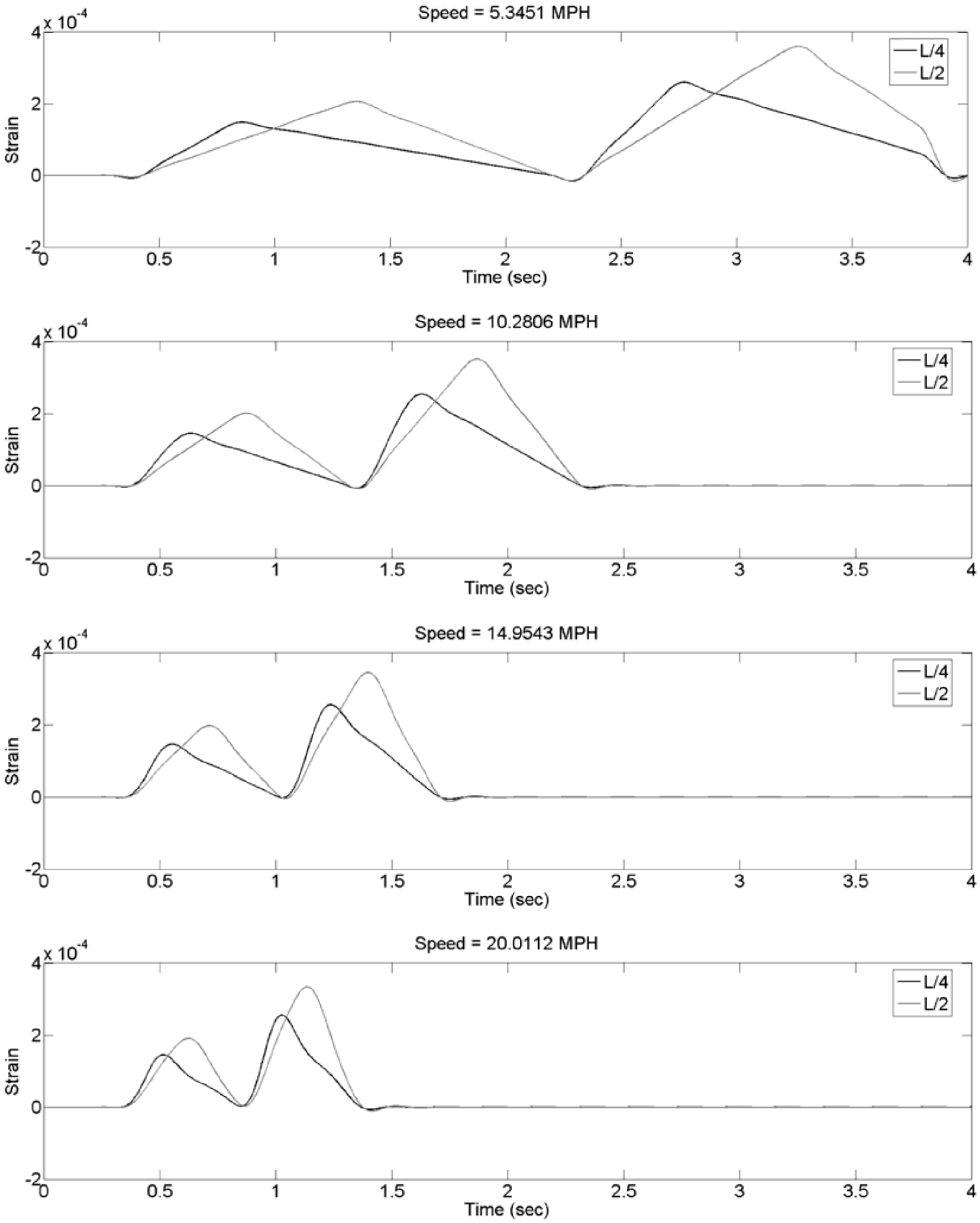


Figure 7.40 a-d: Quarter Span and Midspan Strain Time Histories for a Typical Simulated Vehicle Traveling at approximately 5, 10, 15, and 20 MPH (from top to bottom).

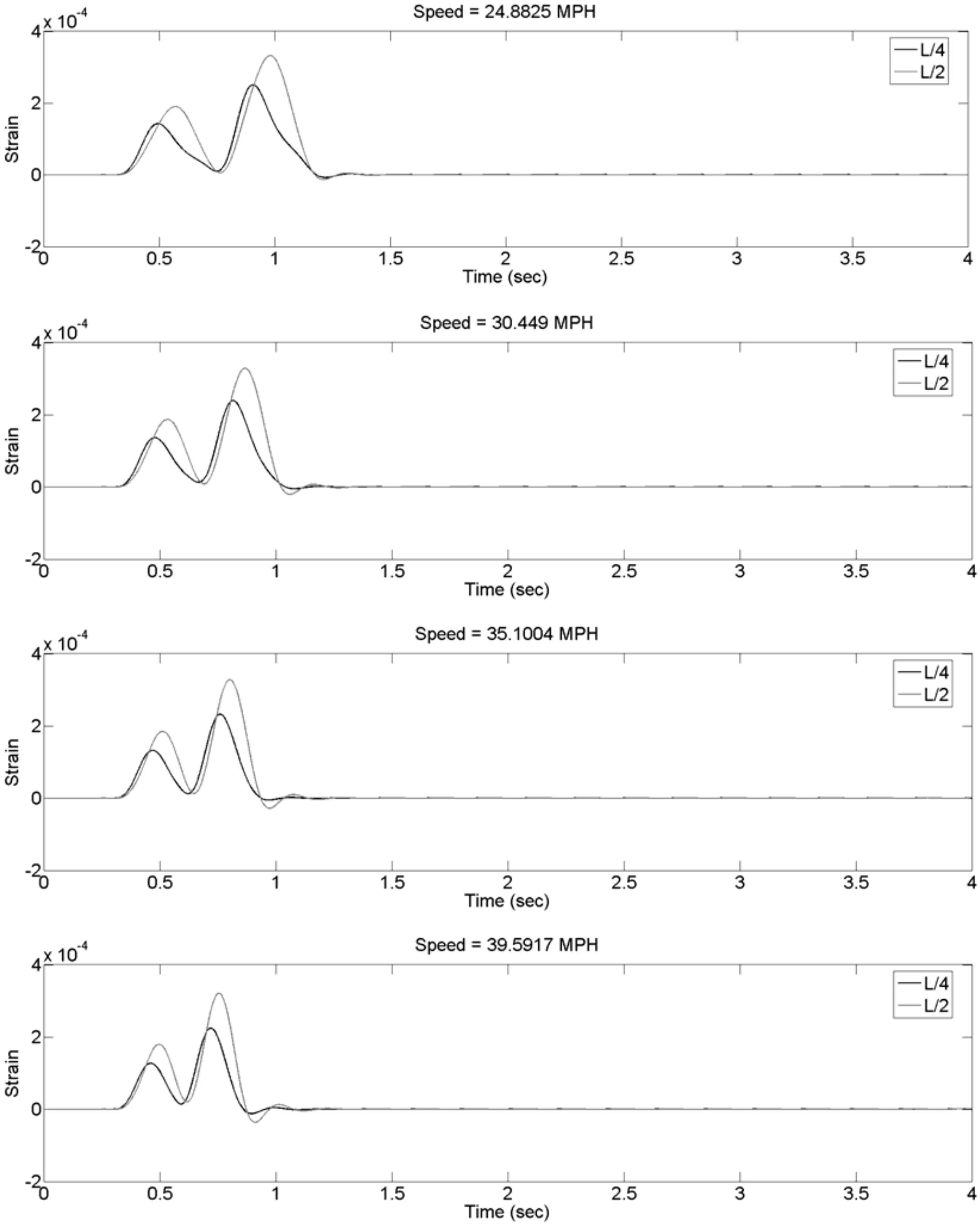


Figure 7.41 a-d: Quarter Span and Midspan Strain Time Histories for a Typical Simulated Vehicle Traveling at approximately 25, 30, 35, and 40 MPH (from top to bottom).

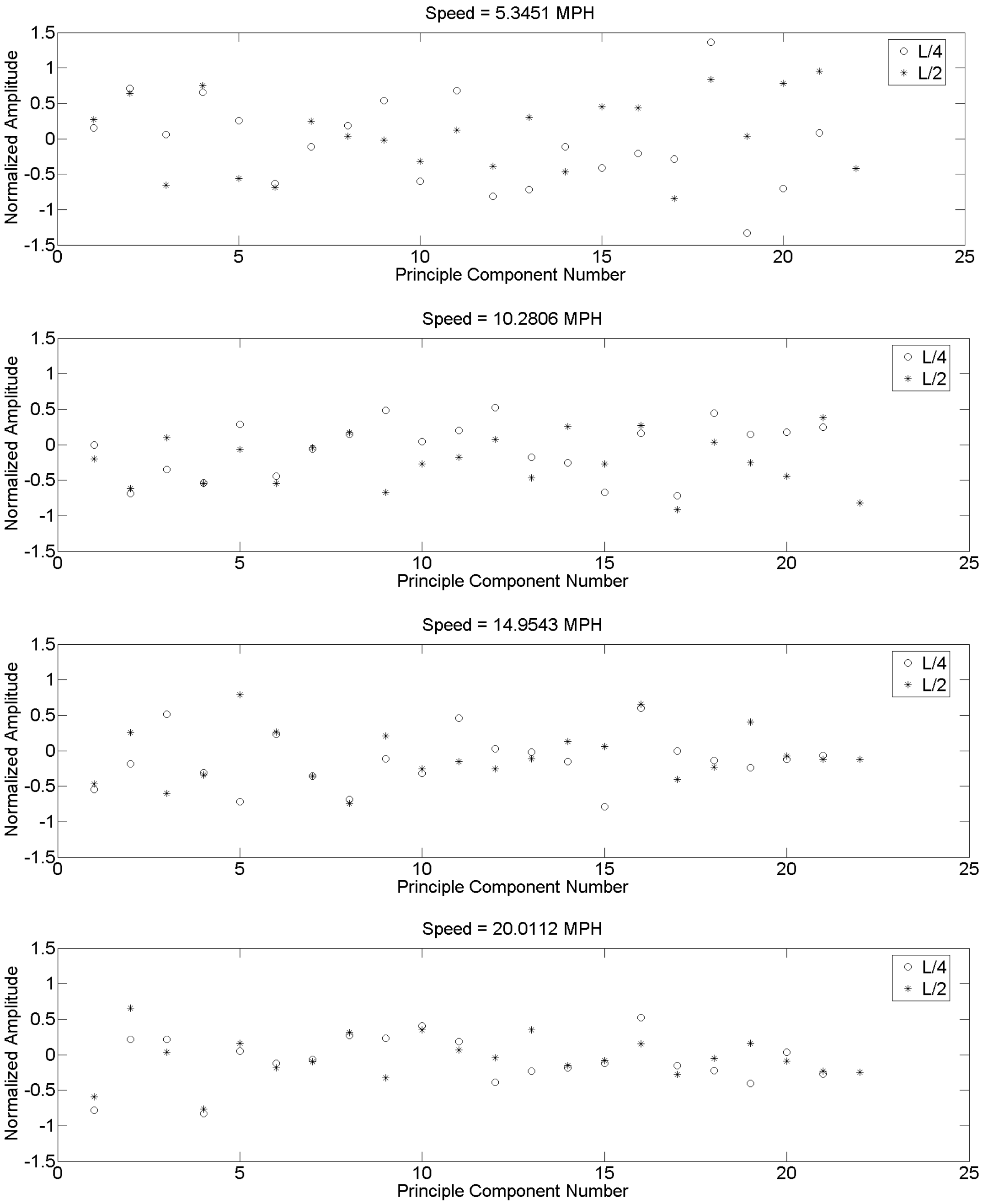


Figure 7.42 a-d: Quarter Span and Midspan Principle Component Projections for a Typical Simulated Vehicle Traveling at approximately 5, 10, 15, and 20 MPH (from top to bottom).

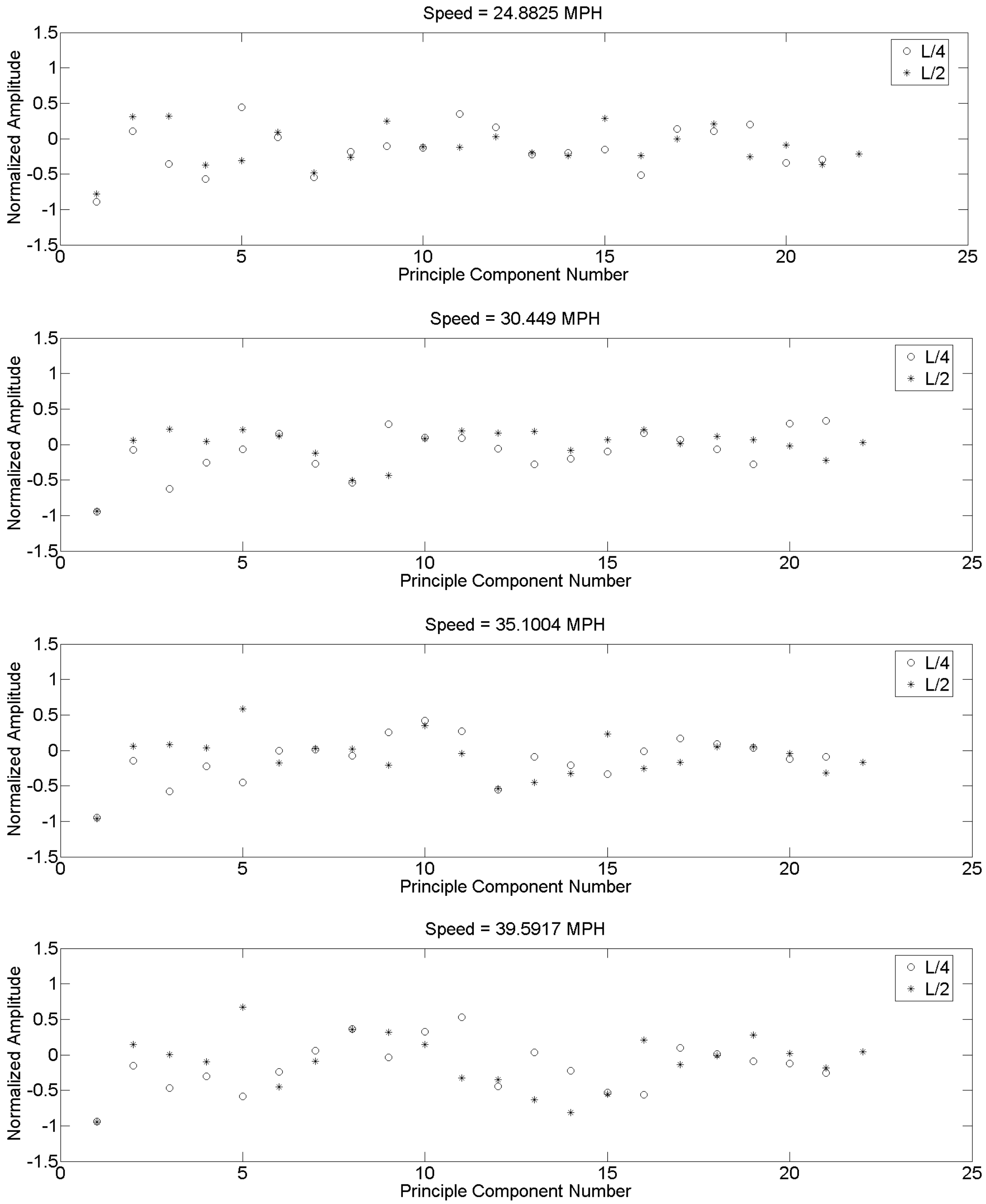
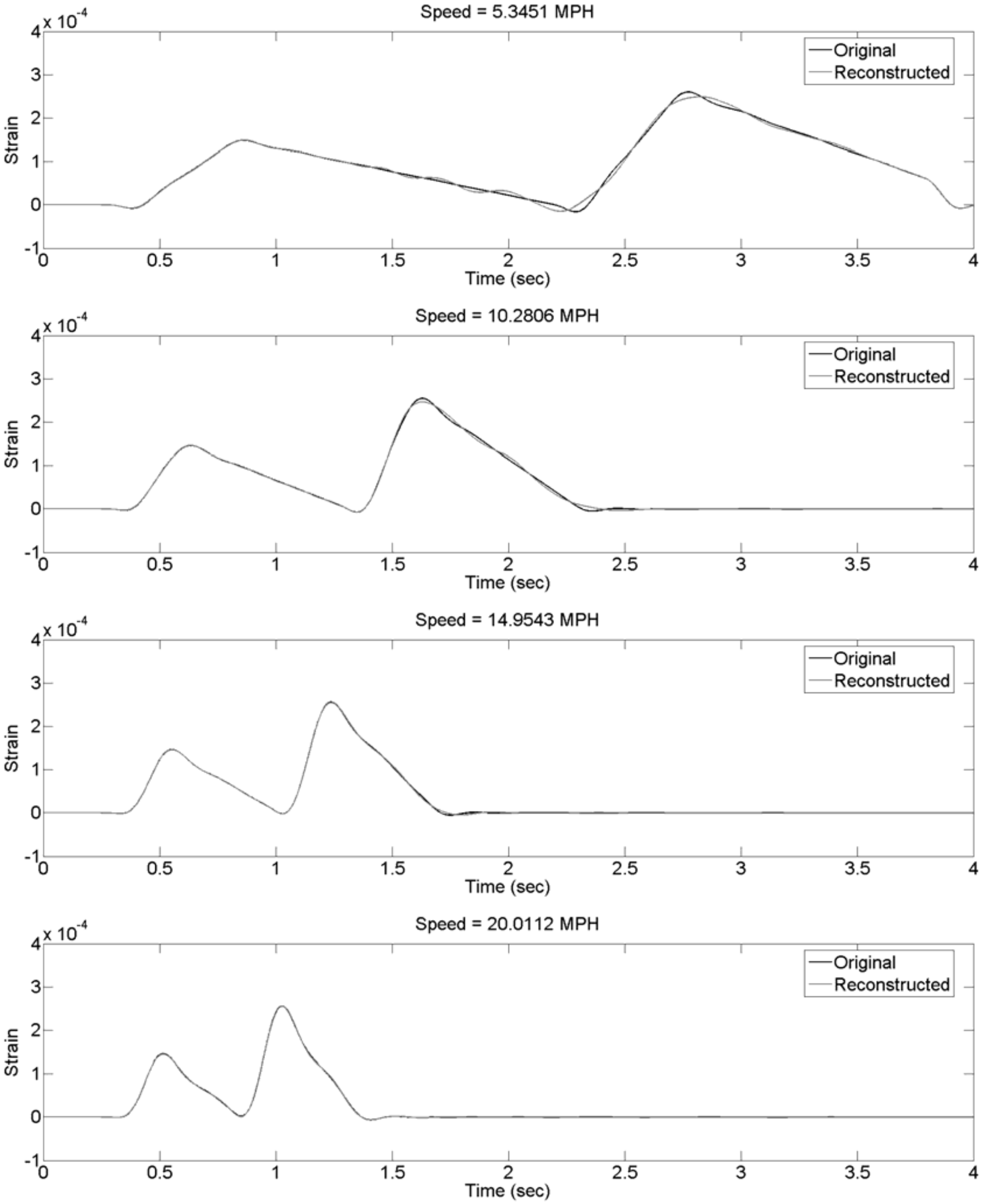


Figure 7.43 a-d: Quarter Span and Midspan Principle Component Projections for a Typical Simulated Vehicle Traveling at approximately 25, 30, 35, and 40 MPH (from top to bottom).



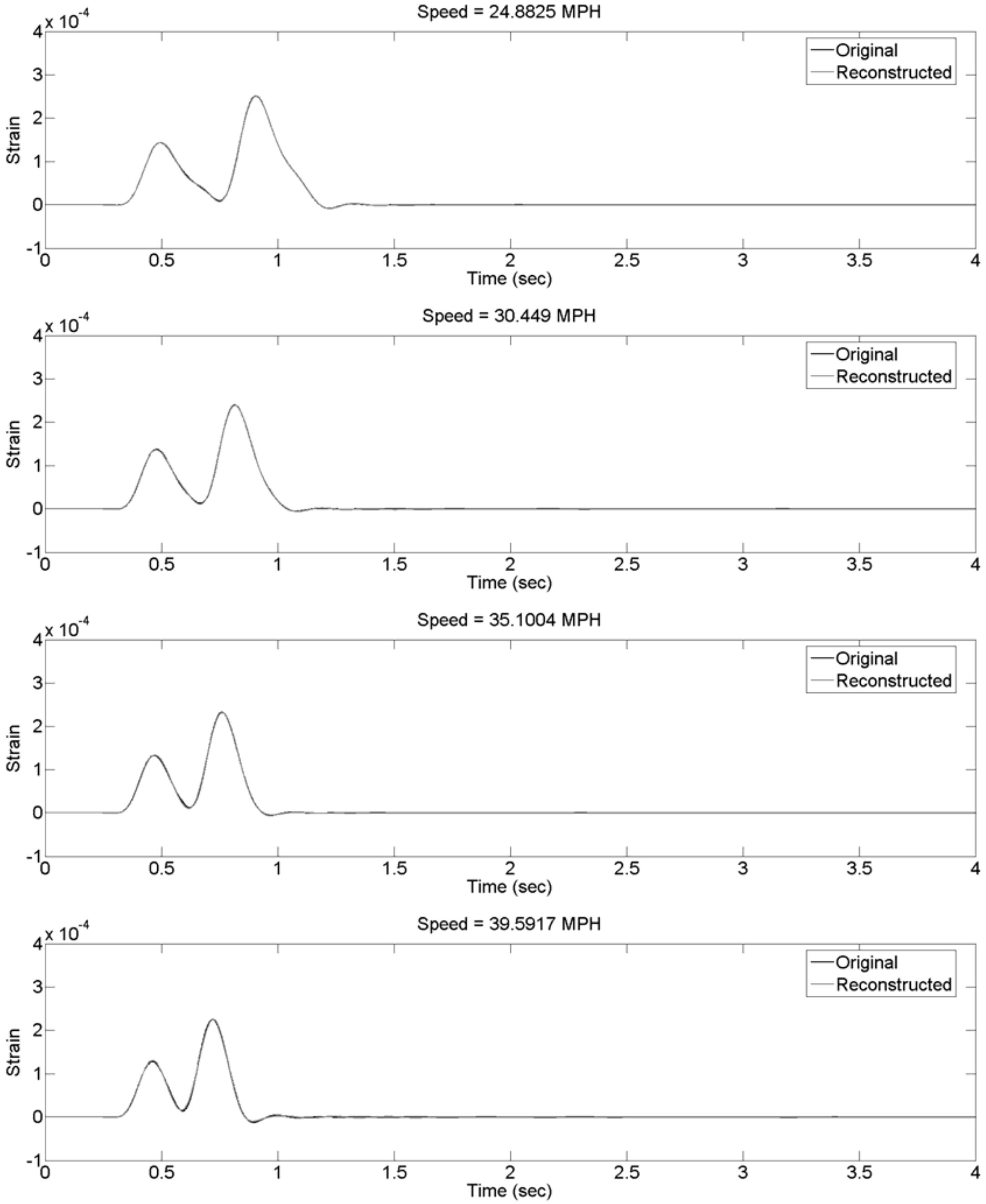
Speed = 5.3451 MPH

Speed = 10.2806 MPH

Speed = 14.9543 MPH

Speed = 20.0112 MPH

Figure 7.44 a-d: Quarter Span Original and Reconstructed Strain Time Histories for Speeds of 5, 10, 15, and 20 MPH.



Speed = 24.8825 MPH

Speed = 30.449 MPH

Speed = 35.1004 MPH

Speed = 39.5917 MPH

Figure 7.45 a-d: Quarter Span Original and Reconstructed Strain Time Histories for Speeds of 25, 30, 35, and 40 MPH.

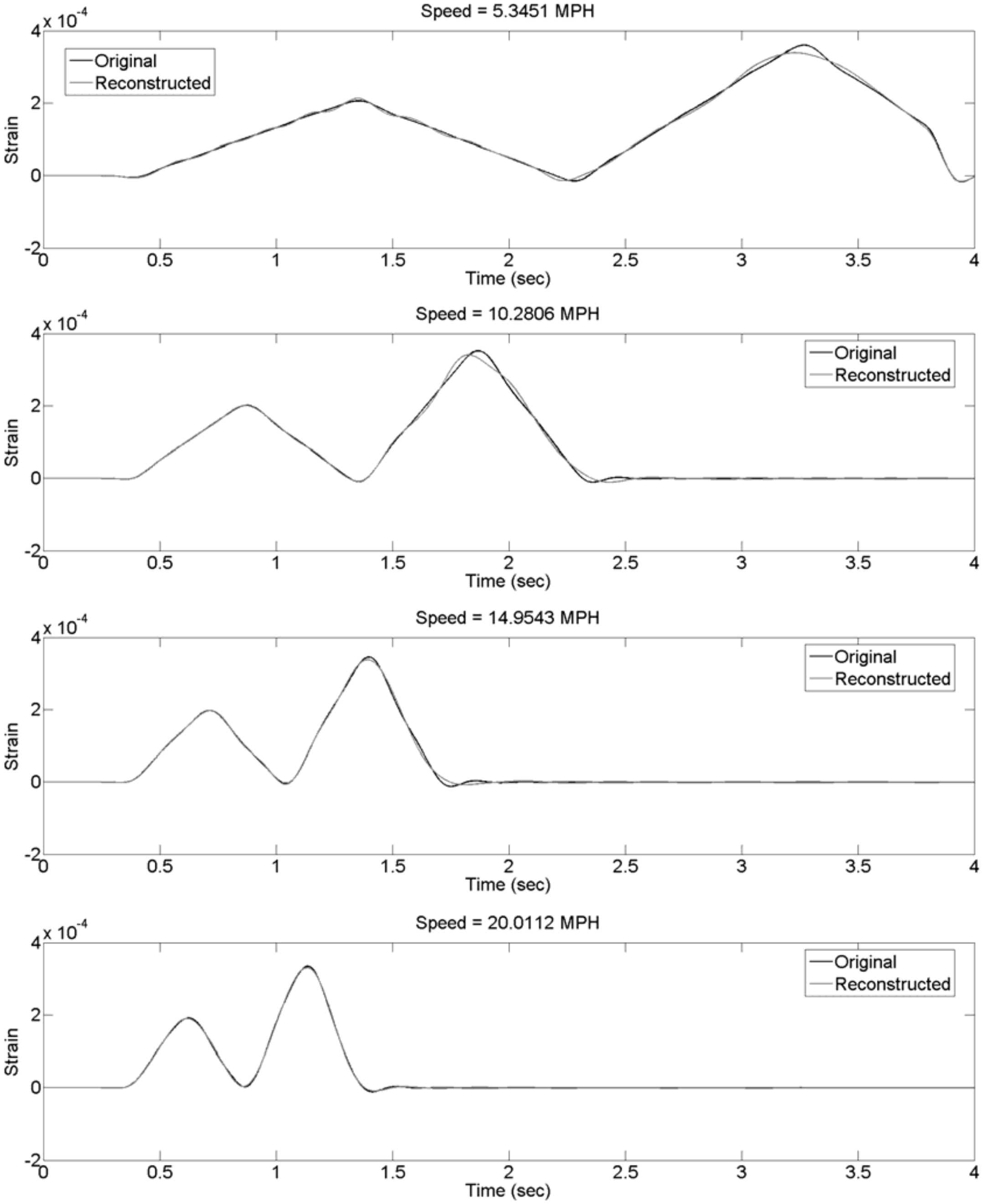


Figure 7.46 a-d: Midspan Original and Reconstructed Strain Time Histories for Speeds of 5, 10, 15, and 20 MPH.

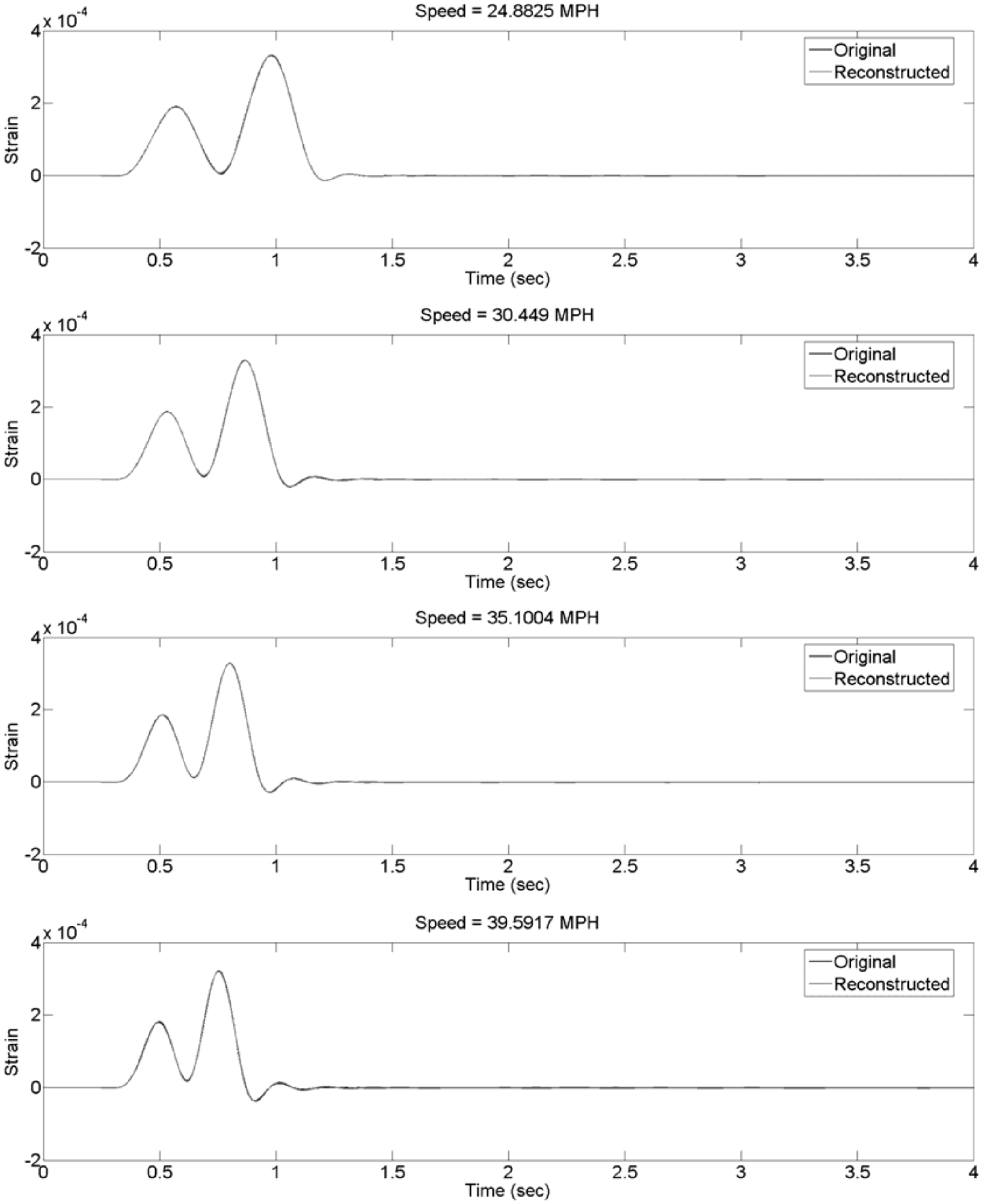


Figure 7.47 a-d: Midspan Original and Reconstructed Strain Time Histories for Speeds of 25, 30, 35, and 40 MPH.

7.8.2 Neural Network Training

The original neural network was trained using 200 simulated traffic patterns, in which the vehicles' speeds, wheelbases, and axle weights were random variables (Figure 7.48). From this figure, it can be concluded too few records were used for training (based on the normalized error for validation data set increasing after a few epoch resulting from over training). When an additional 100 traffic scenarios were added to the training data and, the training curve became acceptable (Fig. 7.49). Therefore, a training set composed of 300 traffic scenarios (with random speeds, wheelbases, and axle weights) was employed.

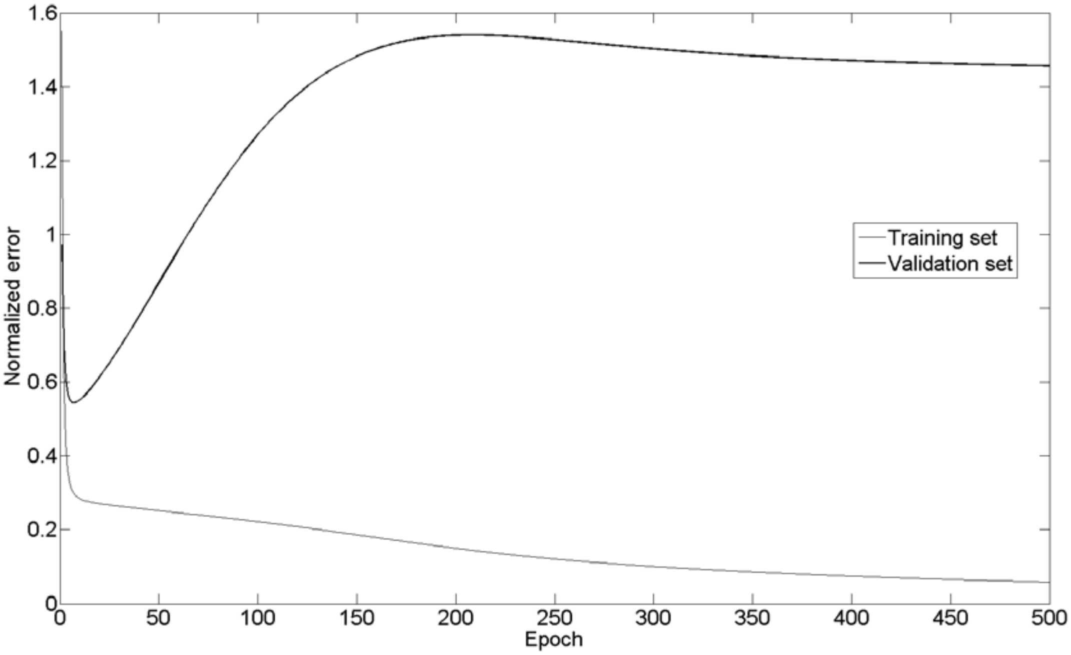


Figure 7.48: Original Network Training Curve for 200 Traffic Patterns

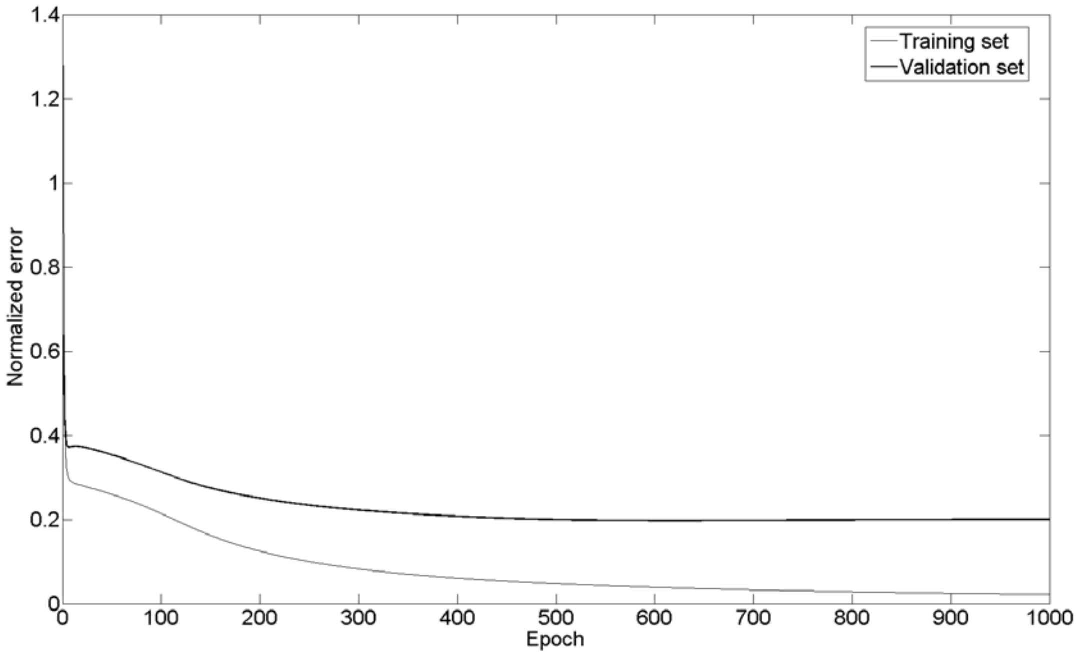


Figure 7.49: Network Training Curve for 300 Traffic Patterns

7.8.3 Local Damage Detection Using Strains from a Single Vehicle Moving at a Fixed Speed

The first application for the trained neural network was to detect damage when a single vehicle (with fixed wheelbase and axle weights) moving at a constant speed is used. In this example, data was simulated for the undamaged as well as 5, 10, 15, 20, 25, 30, 35, 40, and 45% reduction in stiffness of the elements in local damage zone 1. The original strain time histories (output from the FE model), reconstructed strains using the inverse PCA transformation, and the predicted middle span strains are shown in Figures 7.50-7.59. From these figures, it can be seen that the original and reconstructed strains are nearly identical, as indicated by the nearly perfect overlap. As the damage level increases, the strains at the quarter span increase and the value of the middle span strains from the FE model remain constant. However, since the predicted midspan strains are based on the quarter span values, they increase as the damage increases. Consequently, the error between the predicted and actual midspan strains increases as the level of damage increases. This exercise was repeated for a second random vehicle and the error between the actual and predicted midspan strains are shown in Figure 7.60. A normalized error has been adopted which was found to be sensitive to damage even under random loading. This normalized error is calculated by first taking the difference between the predicted and actual finite element strains. Next, the root mean square (RMS) value of the absolute difference vector is computed. Finally, this value is divided by the square root of the are under the finite element strain time history.

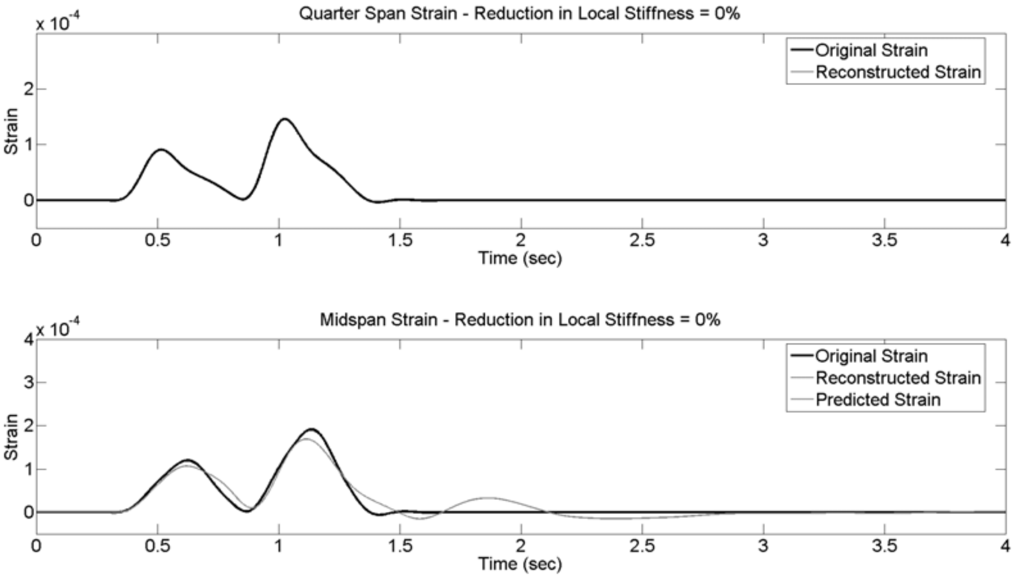


Figure 7.50: Quarter and Middle Span Strains for Undamaged Configuration

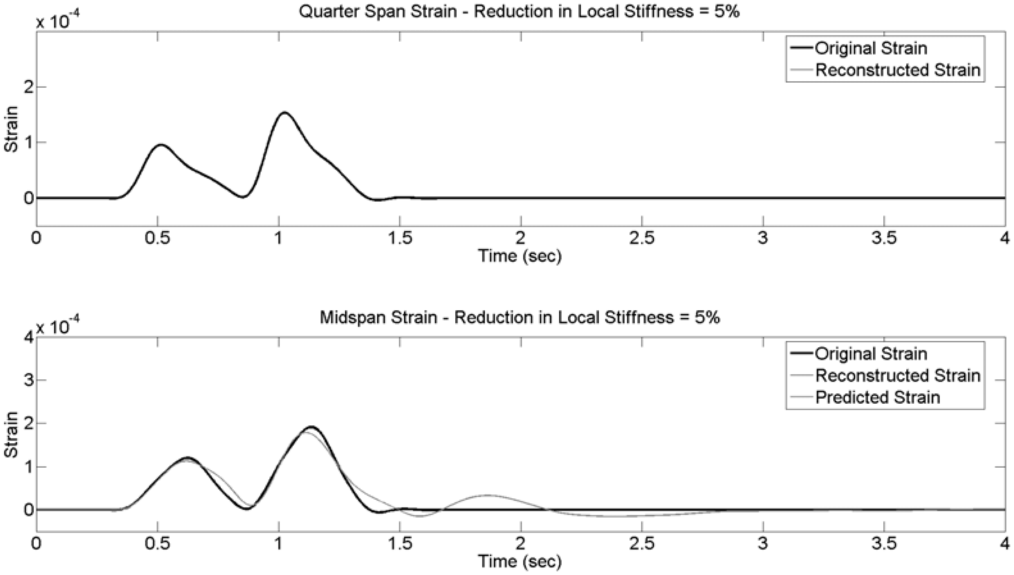


Figure 7.51: Quarter and Middle Span Strains for 5% Reduction in Stiffness

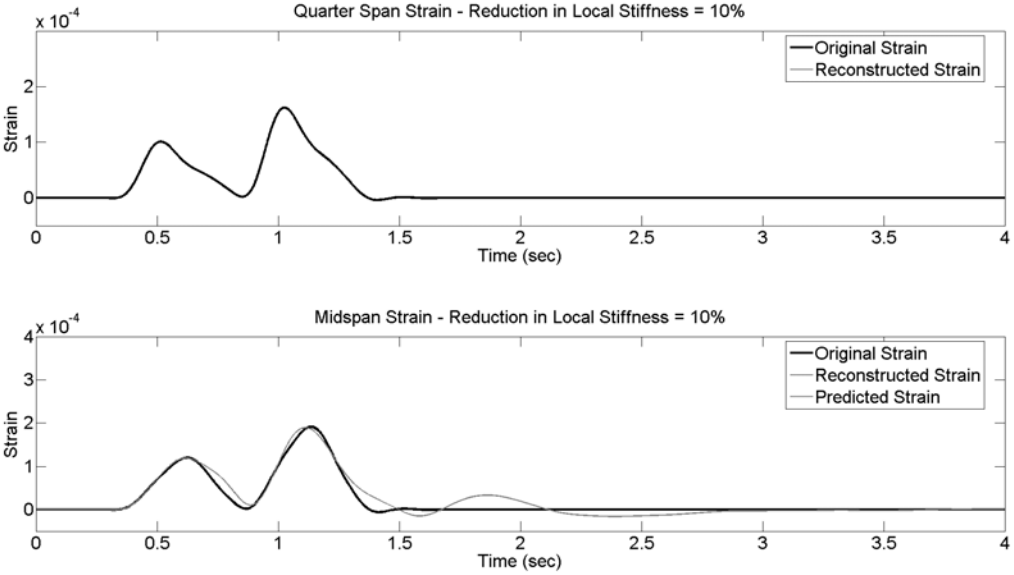


Figure 7.52: Quarter and Middle Span Strains for 10% Reduction in Stiffness

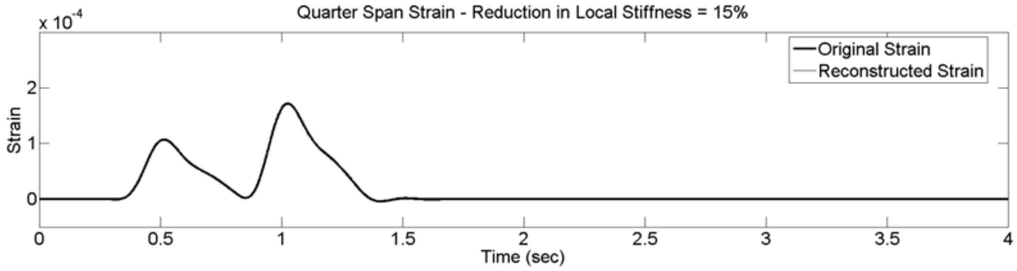


Figure 7.53: Quarter and Middle Span Strains for 15% Reduction in Stiffness

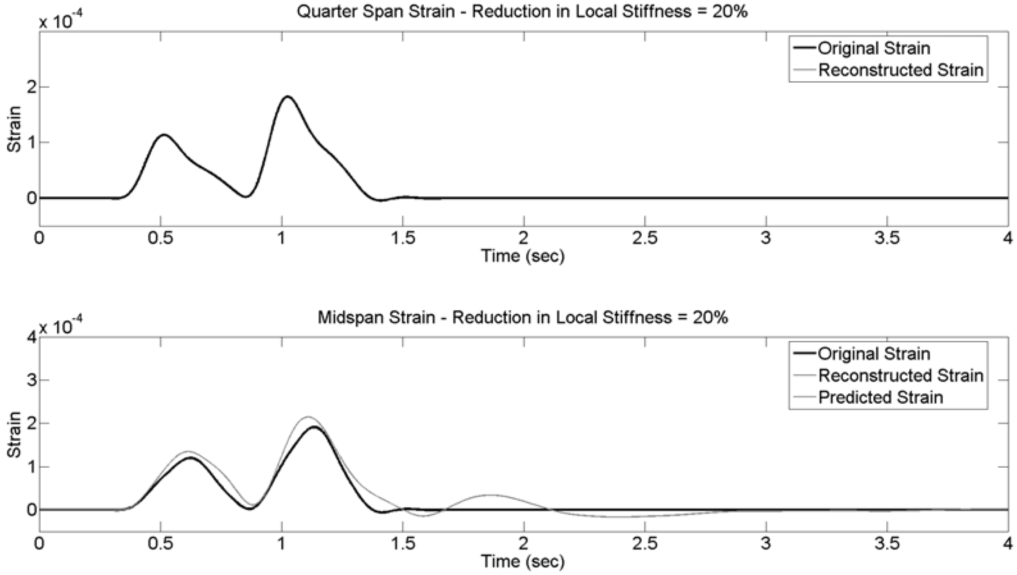


Figure 7.54: Quarter and Middle Span Strains for 20% Reduction in Stiffness

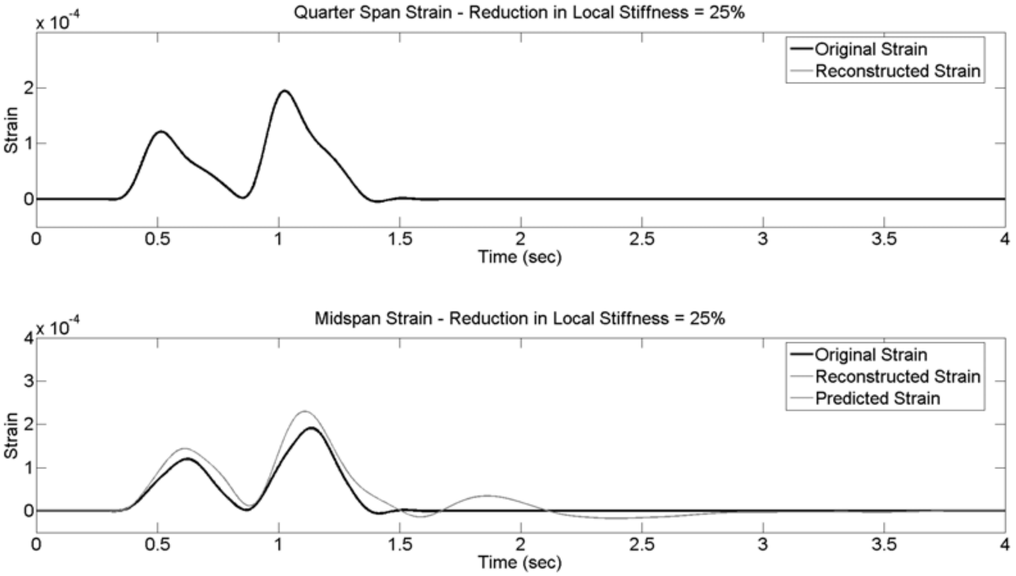


Figure 7.55: Quarter and Middle Span Strains for 25% Reduction in Stiffness

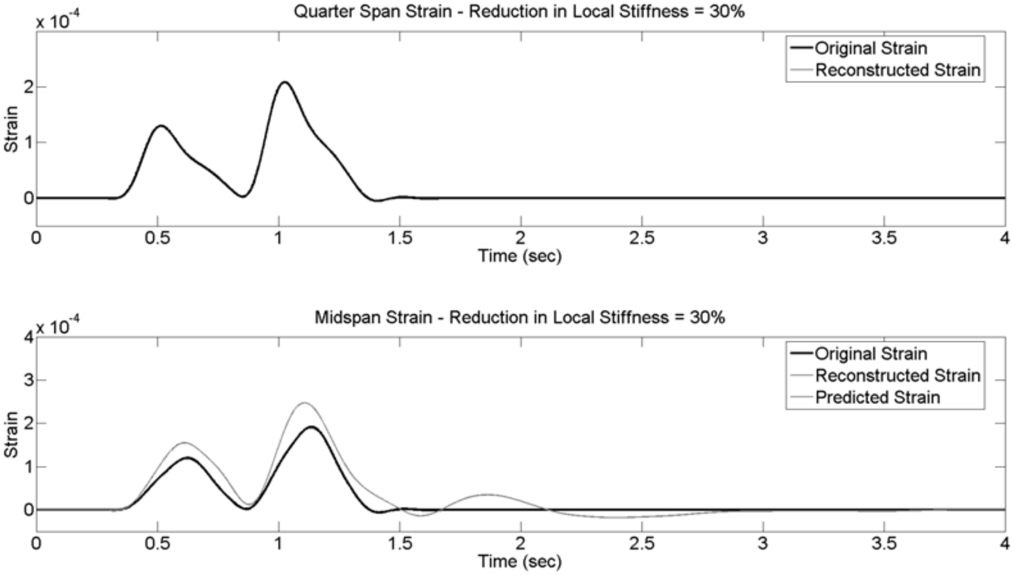


Figure 7.56: Quarter and Middle Span Strains for 30% Reduction in Stiffness

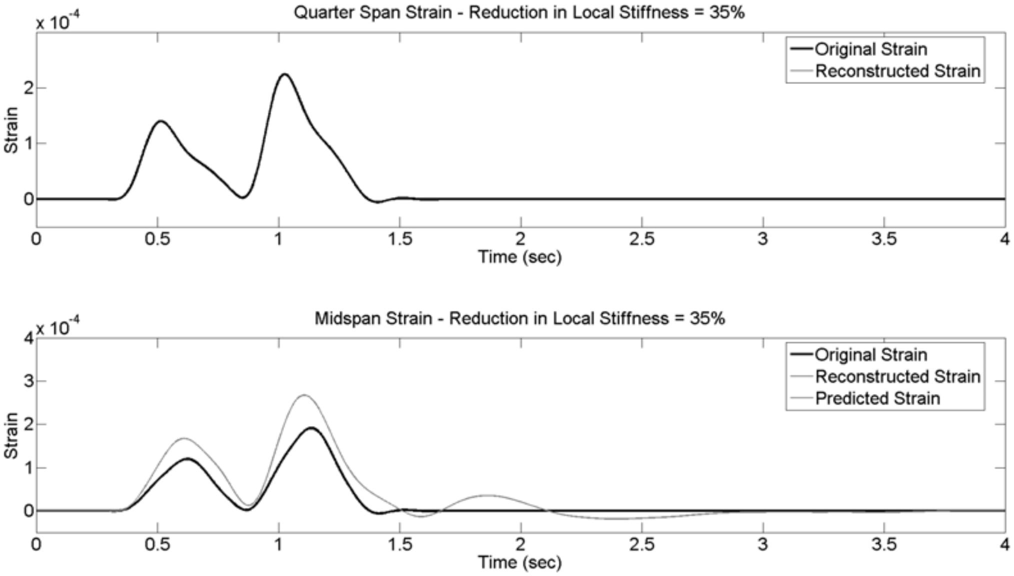


Figure 7.57: Quarter and Middle Span Strains for 35% Reduction in Stiffness

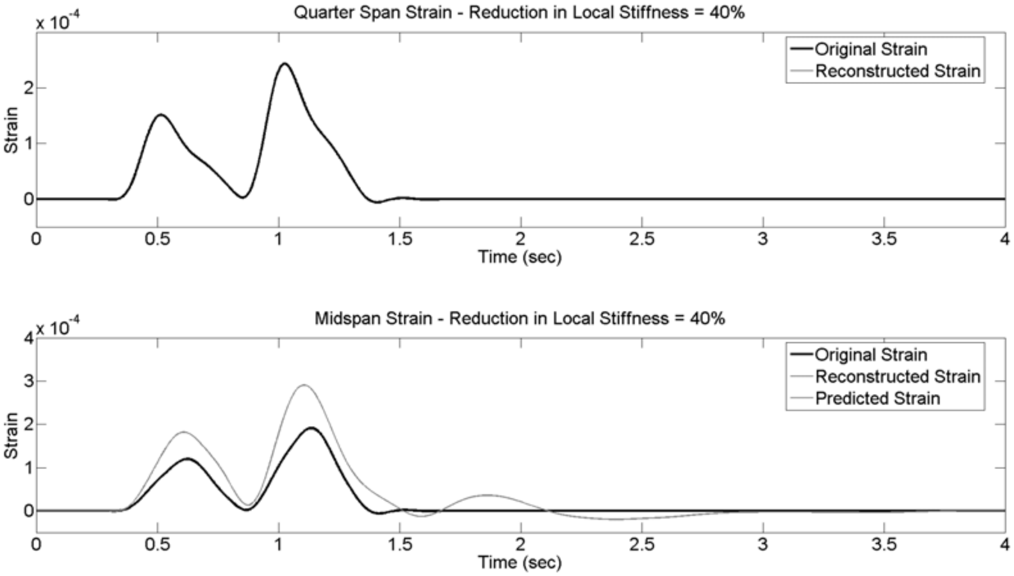


Figure 7.58: Quarter and Middle Span Strains for 40% Reduction in Stiffness

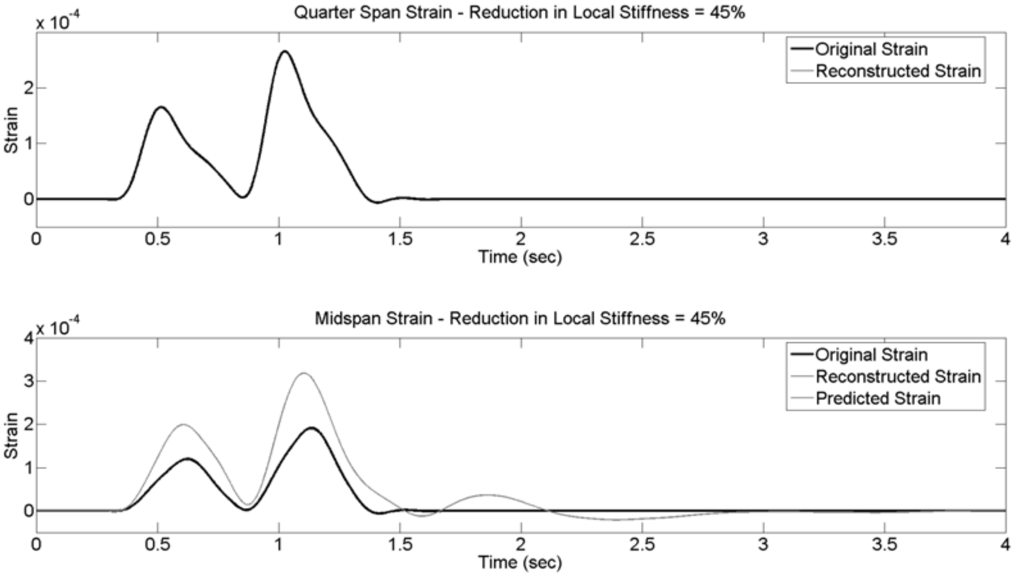


Figure 7.59: Quarter and Middle Span Strains for 45% Reduction in Stiffness

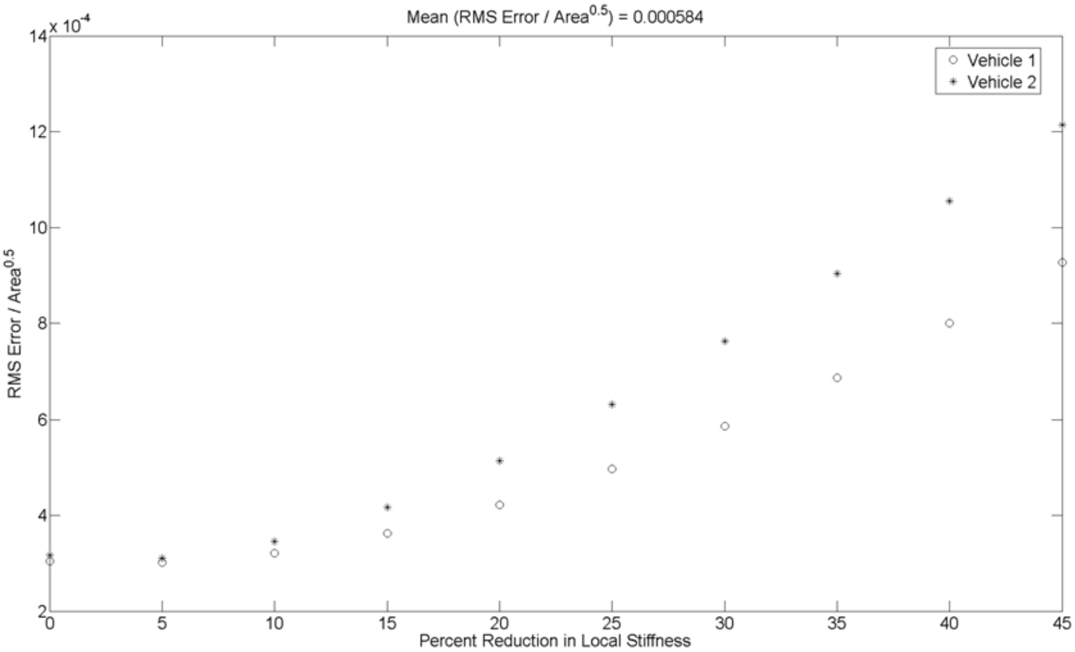


Figure 7.60: Percent Reduction in Stiffness versus Normalized Error

7.8.4 Local Damage Detection Using Strains from a Single Vehicle Moving at Random Speeds

Next, the difficulty of the problem was increased by taking the same singular vehicle and this time running the simulation at random speeds. This was done to determine if the error (and therefore network performance) is dependent on the speeds used in the simulation. This was done by using the same neural network that had been trained on the 300 random vehicles and testing with a data set composed of 300 traffic patterns in which the vehicle wheelbase and axle weights were constant but the speed was a random value (Fig. 7.61). For 100 of these patterns, no damage was present, while in the other 200 patterns local damage existed in zone 1.

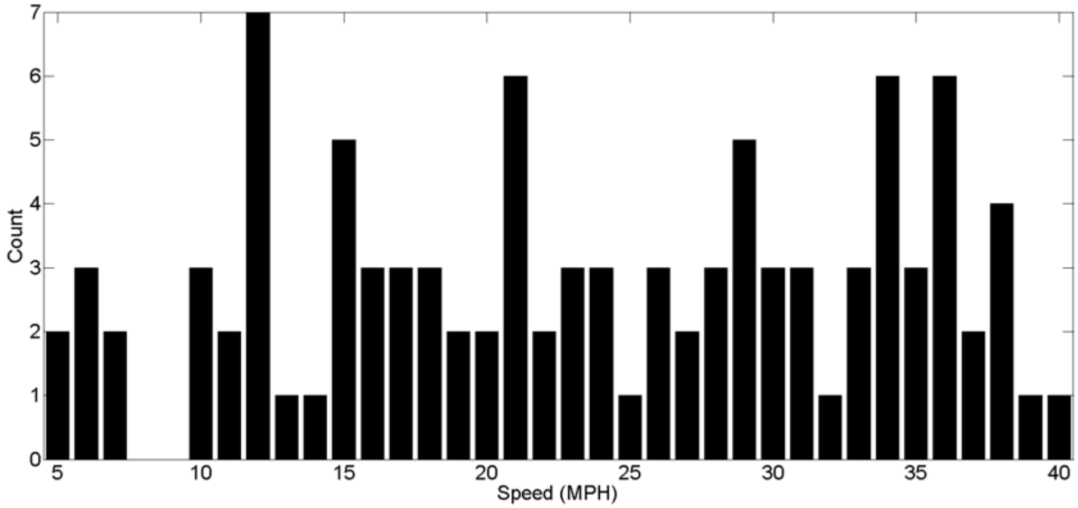


Figure 7.61: Histogram of Simulated Speeds

As with the previous example, the normalized errors between the predicted and actual middle span strains were calculated and are plotted in Fig. 7.62. In general, as the level of damage increases, the normalized error also increases. In Figure 7.63, the level of damage is plotted against the normalized error. From this figure, it appears that below 10-15% of damage, the normalized error between the undamaged and damaged cases is indistinguishable. Finally, the normalized errors are plotted against the speeds used in the finite element simulation (Fig. 7.64). While, the true dependence on speed is unclear, it does appear from the undamaged records (as indicated by the * in Fig. 7.64) the normalized error is affected by speed. This issue is addressed later in section 7.8.7 of this paper.

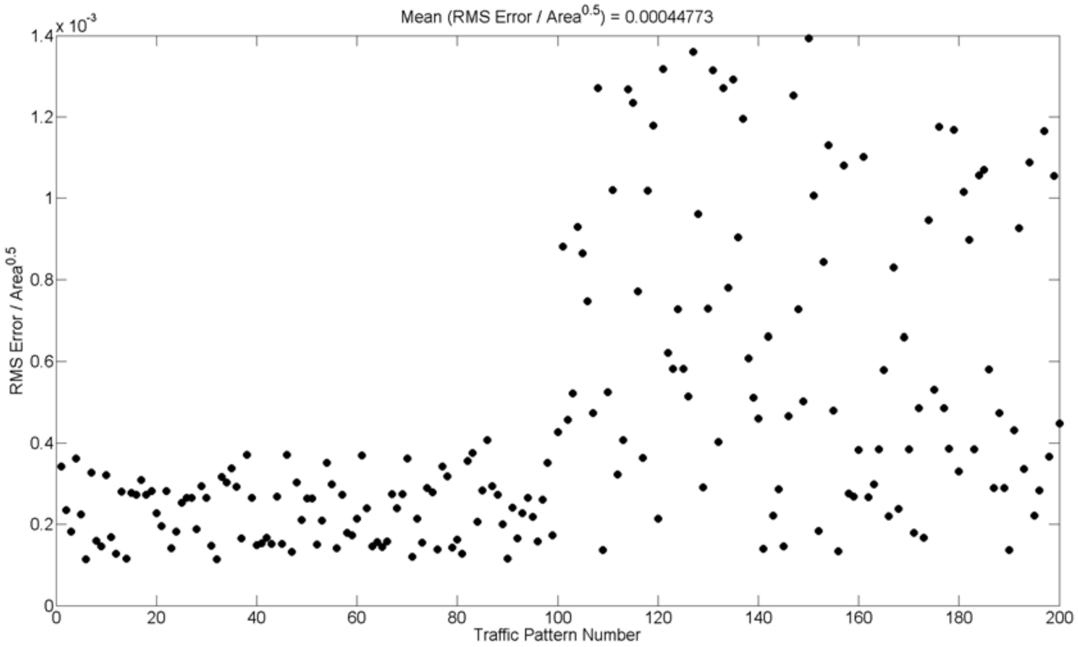


Figure 7.62: Traffic Patten Number versus Normalized RMS Error

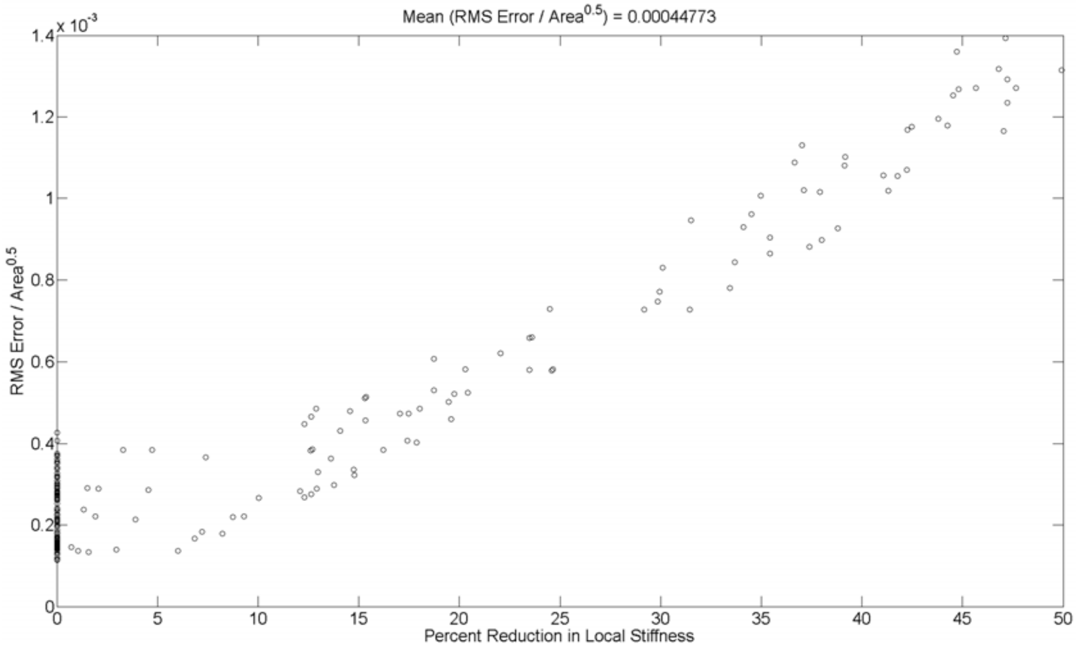


Figure 7.63: Percent Reduction in Local Stiffness versus Normalized Error

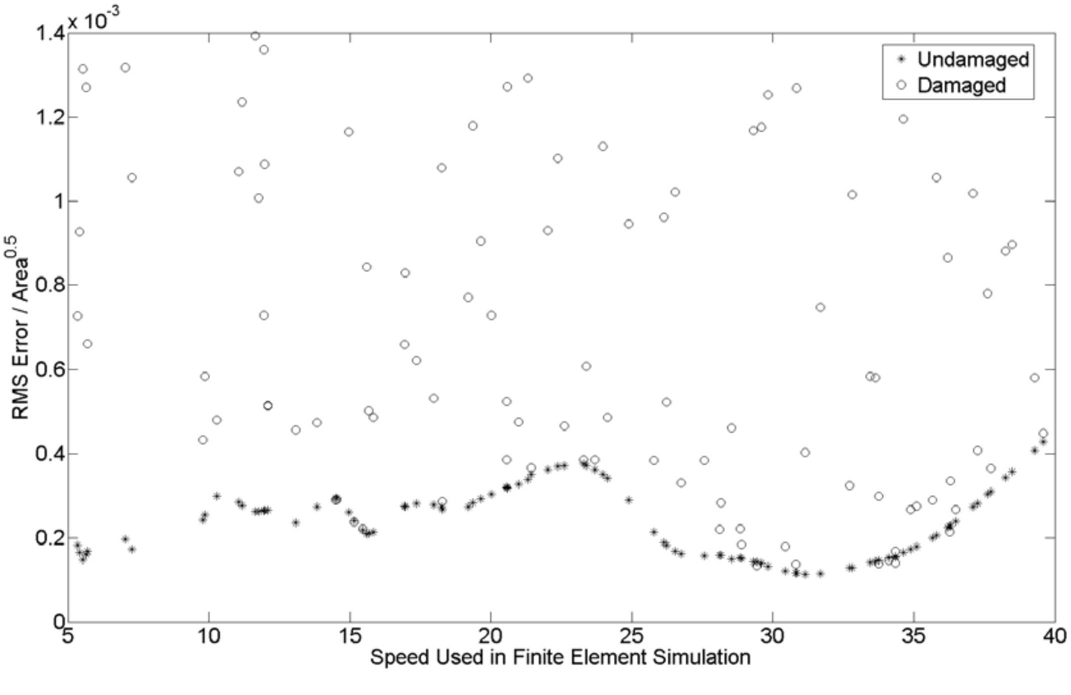


Figure 7.64: Speed Used in FE Simulation versus Normalized Error

7.8.5 Characterizing Performance of the Neural Network for Variable Axle Weight in Undamaged Configuration

Having shown the normalized error between the actual strains and the values predicted by the neural network are dependent on the speed used in the finite element simulation, a similar check was made of the axle weights used. This was done by generating fifty traffic patterns using the one-dimensional finite element model in which the vehicle speed and wheelbase were held constant at 25 miles-per-hour and 15 feet, respectively. For these patterns, the axle weights varied according to:

Front Axle Weight (kips) = 2 + 10*Rand[0 1] **Eq. 7.4**

Rear Axle Weight (kips) = Front Axle Weight + 10*Rand[0 1] **Eq. 7.5**

Histograms of the front and rear axle weights are shown in Figures 7.65 and 7.66. Figure 7.67 is a plot of front versus rear axle weights.

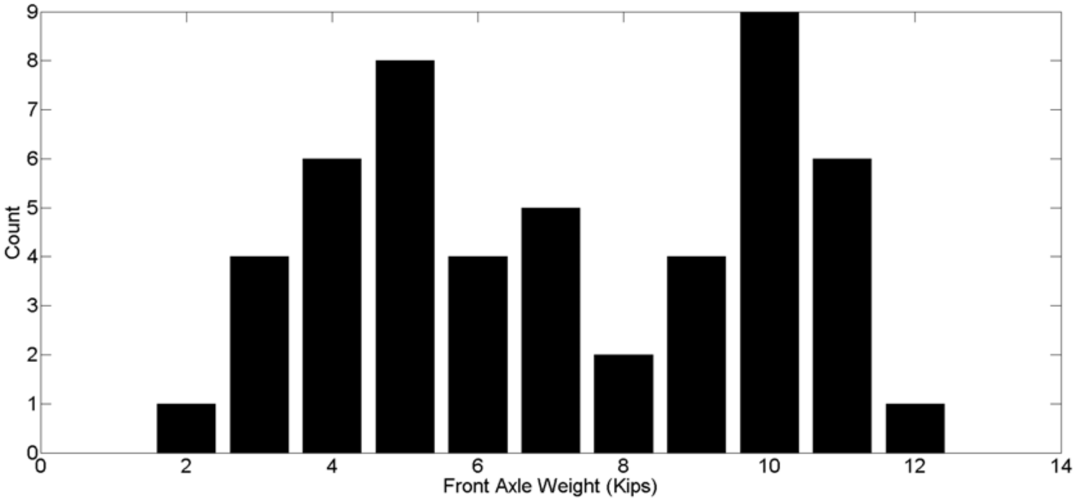


Figure 7.65: Histogram of Front Axle Weights

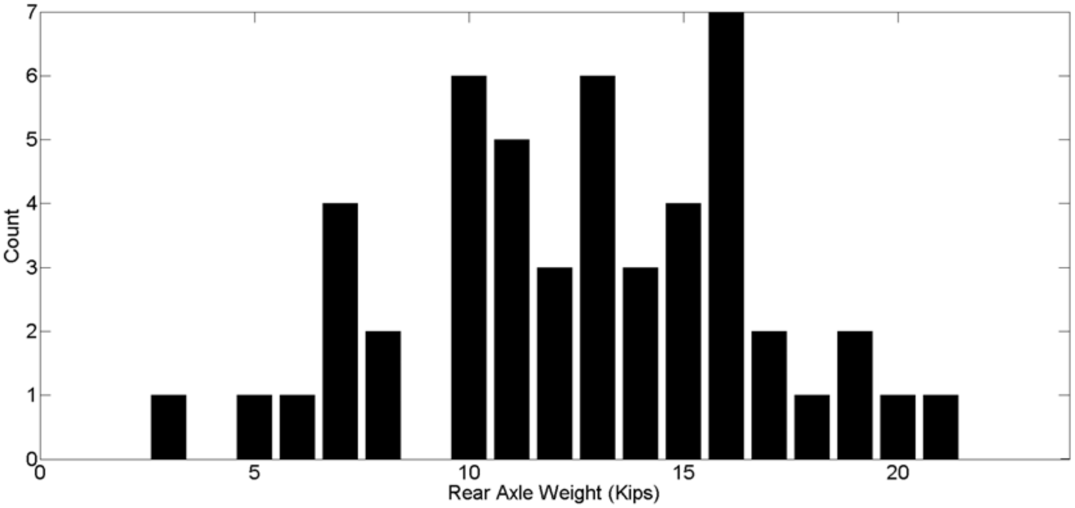


Figure 7.66: Histogram of Rear Axle Weights

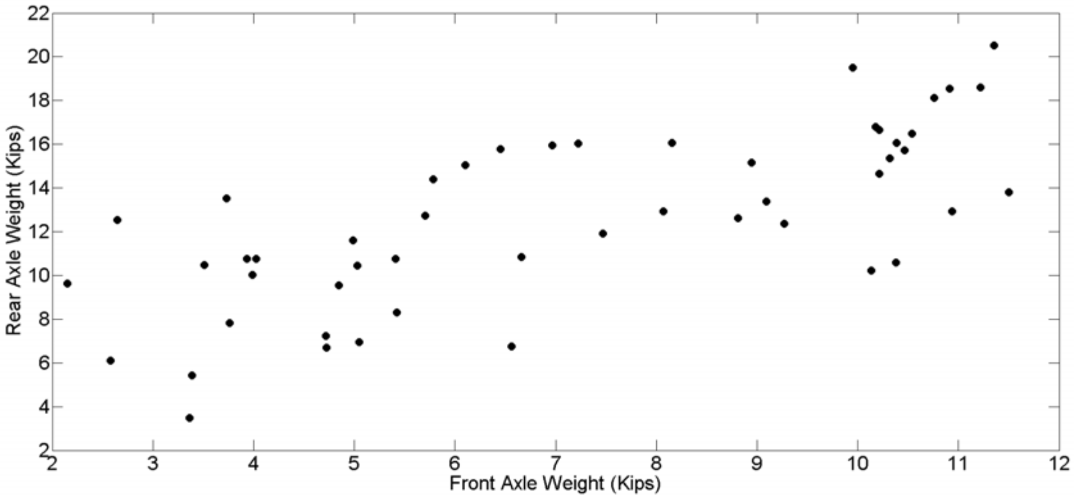


Figure 7.67: Front Axle Weight versus Rear Axle Weight

The fifty sets of strain time histories were reduced using the previously discussed PCA transformation and fed into the trained neural network. The predicted principle components were then converted back to strain time histories using the inverse PCA transformation, and the normalized RMS error was calculated. Plots of the normalized error versus front axle weight, rear axle weight, and total vehicle weight are shown in Fig's 7.68-7.70. From these figures, there does not appear to be a correlation between axle weight and normalized error.

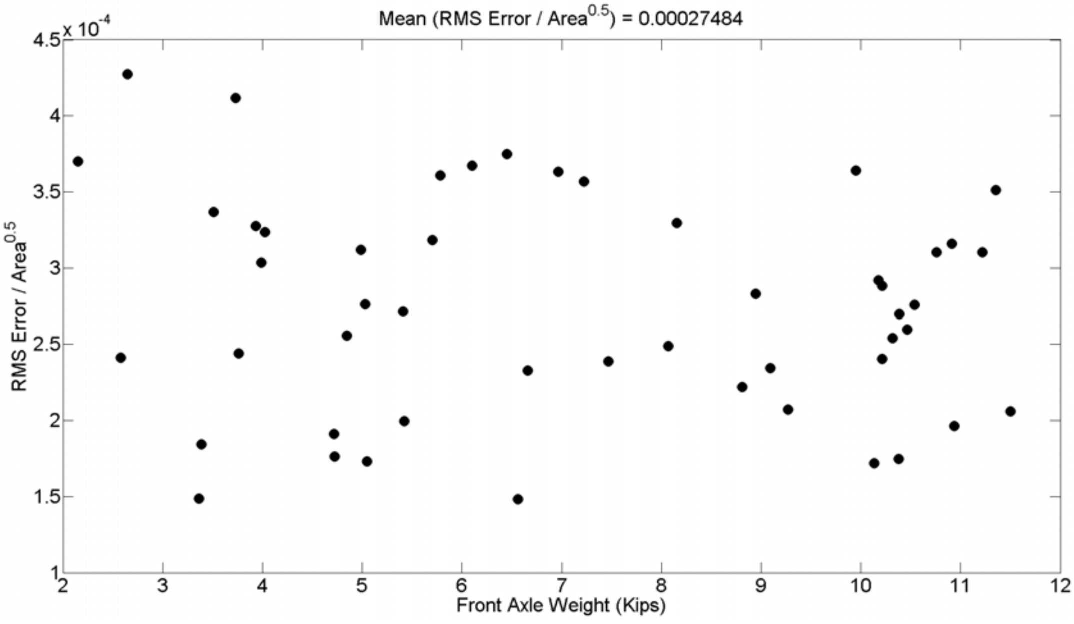


Figure 7.68: Front Axle Weight versus Normalized RMS Error

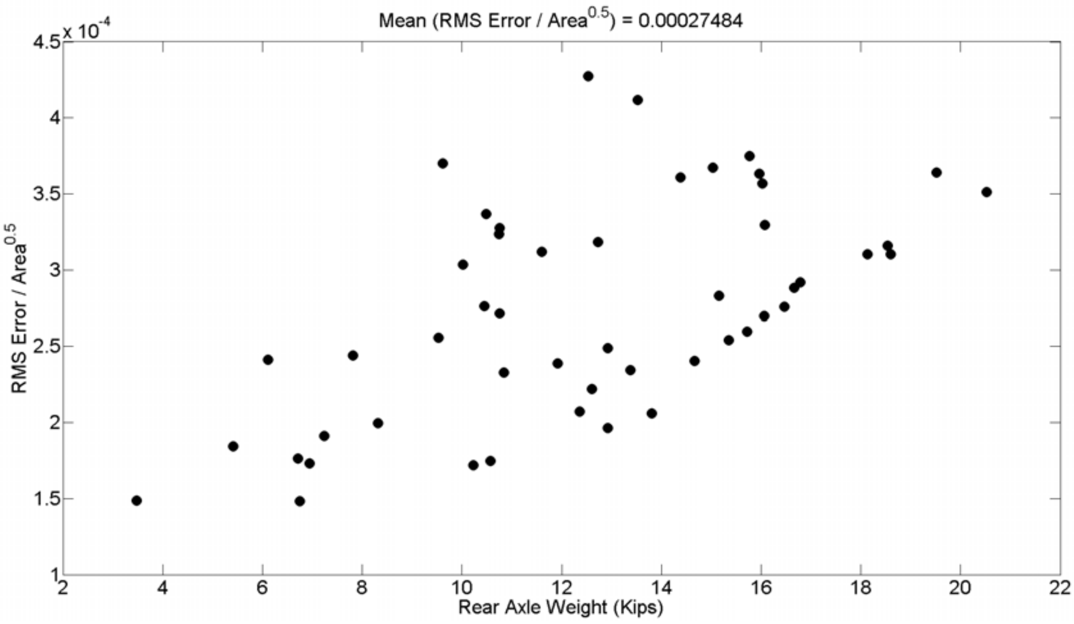


Figure 7.69: Rear Axle Weight versus Normalized RMS Error

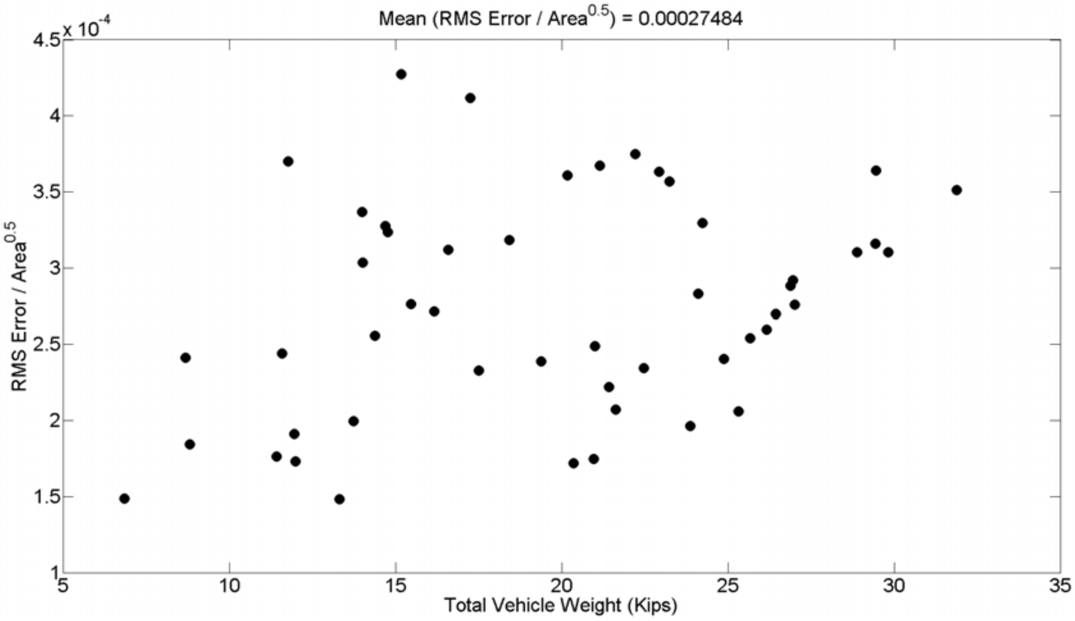


Figure 7.70: Total Vehicle Weight versus Normalized RMS Error

7.8.6 Local Damage Detection Using Strains Time Histories Generated Under Random Traffic Loading

A new data set was constructed in which the speeds, wheelbase, and axle weight for 500 patterns were randomly selected. 100 of these patterns were for the undamaged state. Of the remaining 400 patterns, 200 had damage in local zone 1 and 200 in local zone 2. The damage was again modeled by reducing the stiffness of all elements in the local damage zone by the same random value. The strain time histories were then reduced to PC's and fed into the neural network. The normalized errors between the actual and predicted midspan strain time histories were calculated and are plotted in Fig. 7.71. The normalized errors are also plotted versus the percent reduction in stiffness in Fig. 7.72. From this figure, it appears the normalized error can be used to recognize damage greater than approximately 10%.

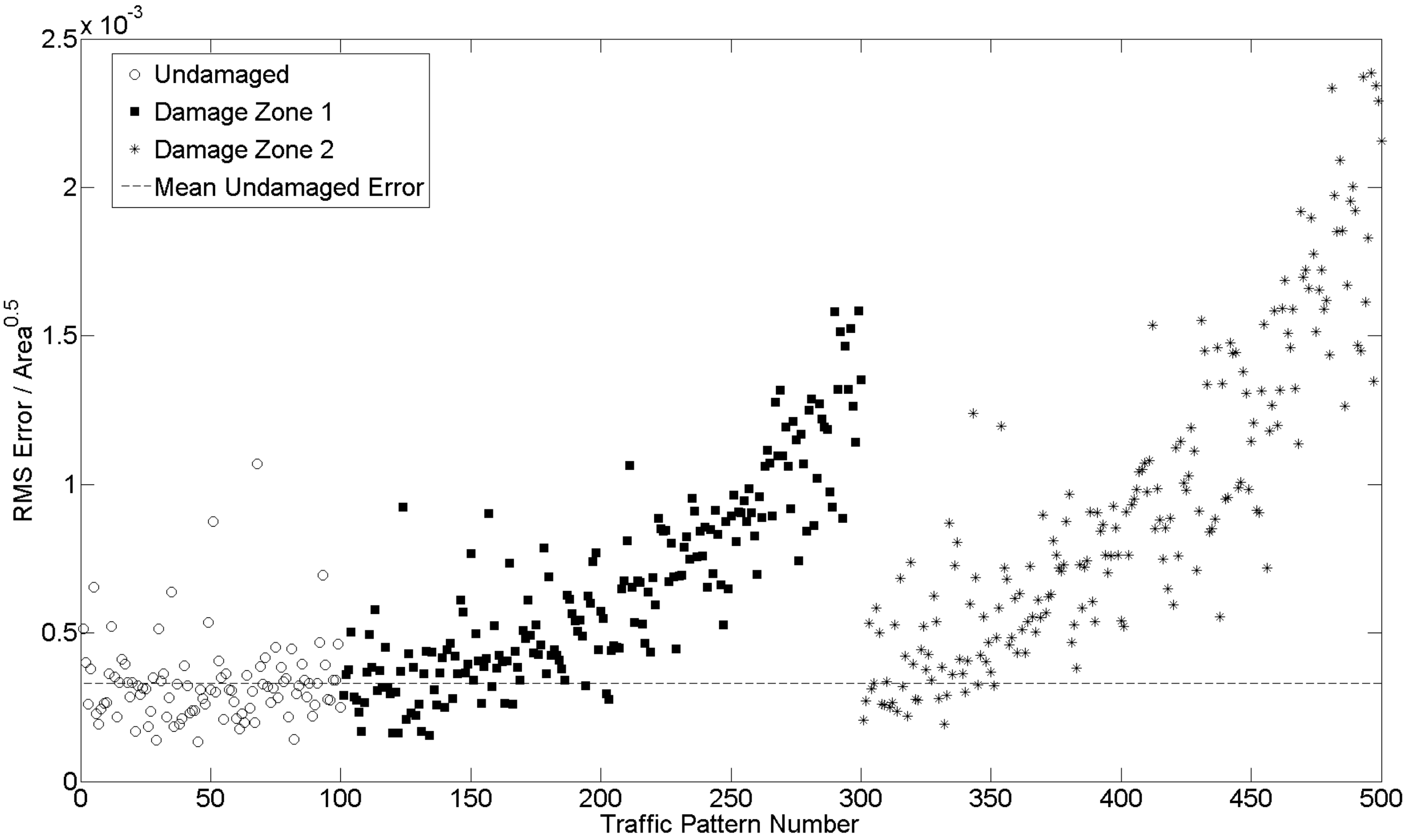


Figure 7.71: Simulation Number (Sorted by Percent Damage in Each Zone) versus Norm. Error

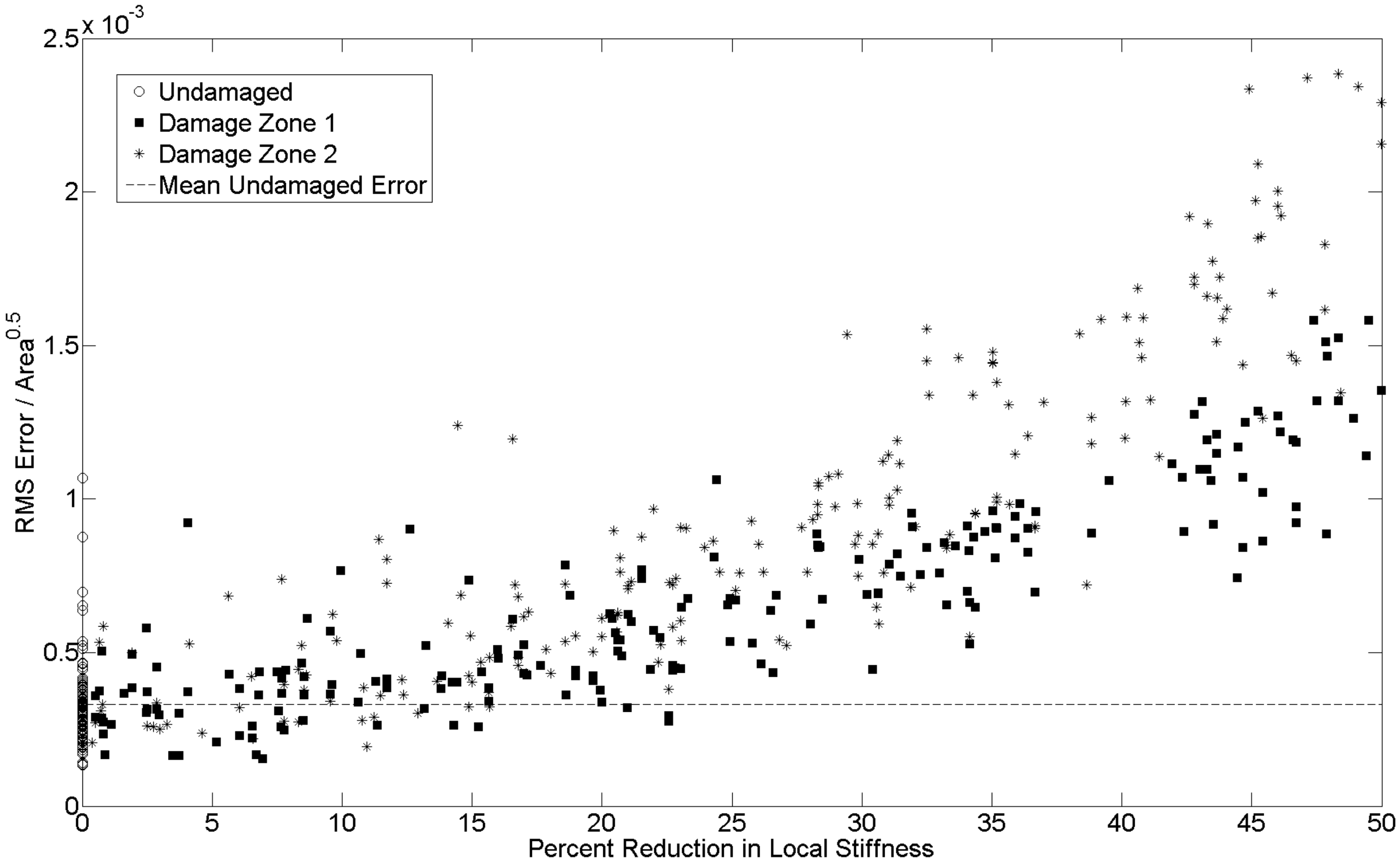


Figure 7.72: Percent Damage versus Normalized Error

7.8.7 Alternate Procedure for Normalizing the Strain Time Histories

To address the dependence of the normalized error on the speeds used in the finite element simulations and to reduce the threshold beyond which damage may be detected and alternate method for normalizing the strain time histories is presented. Rather than performing Principal Component Analysis on a four-second long time history, an alternate strategy is employed in which only the portion of the strain time histories corresponding to when the vehicle is actually crossing the bridge decks is used.

This procedure consists of first finding the two peak strains corresponding to the front and rear axles passing over the recording element (or strain gage). The peaks are indicated by the filled circle in Fig's. 7.73 and 7.74. Next, the zero crossing to the left of the first peak and to the right of the second peak strain are found. This portion of the data is isolated as it is during this time that the vehicle is in contact with the bridge decks. As the PCA program requires all of the input and target strain time histories contain the same number of time steps, it is necessary to resample the data to provide a constant number of points (in this example 1000 points). Depending on the speed used in the simulation, it may be necessary to decimate the data (as with the simulated bus traveling 10 MPH in Fig. 7.73b) or to interpolate (simulated bus traveling 40 MPH in Fig. 7.74b). The resampled data used for inputs into the PCA program are shown in Figures 7.73c and 7.74c.

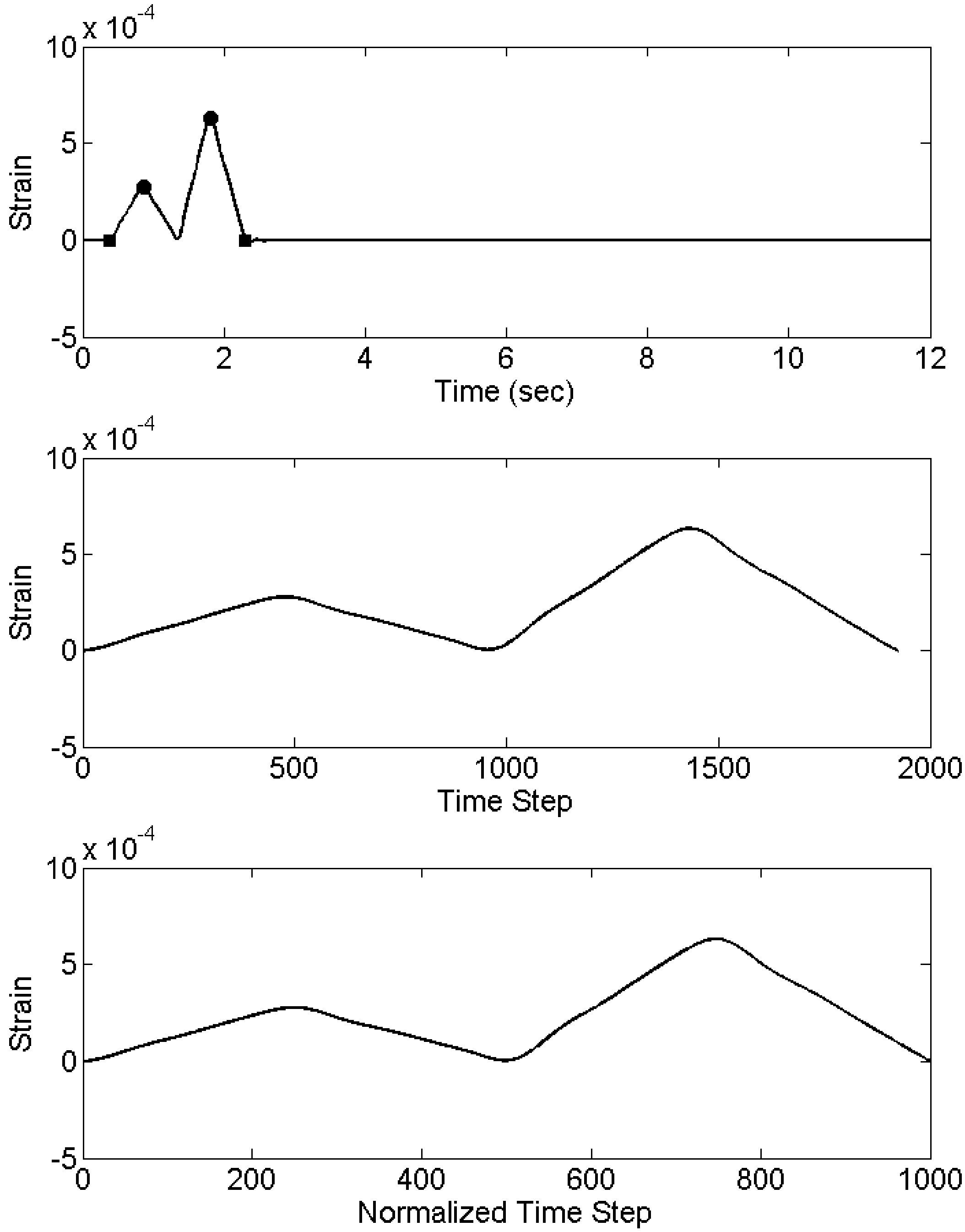


Figure 7.73: Time Scale Normalization of a Simulated Bus Traveling at 10 MPH: Original Time History (top), Section of Interest (Middle) and Resampled (Bottom)

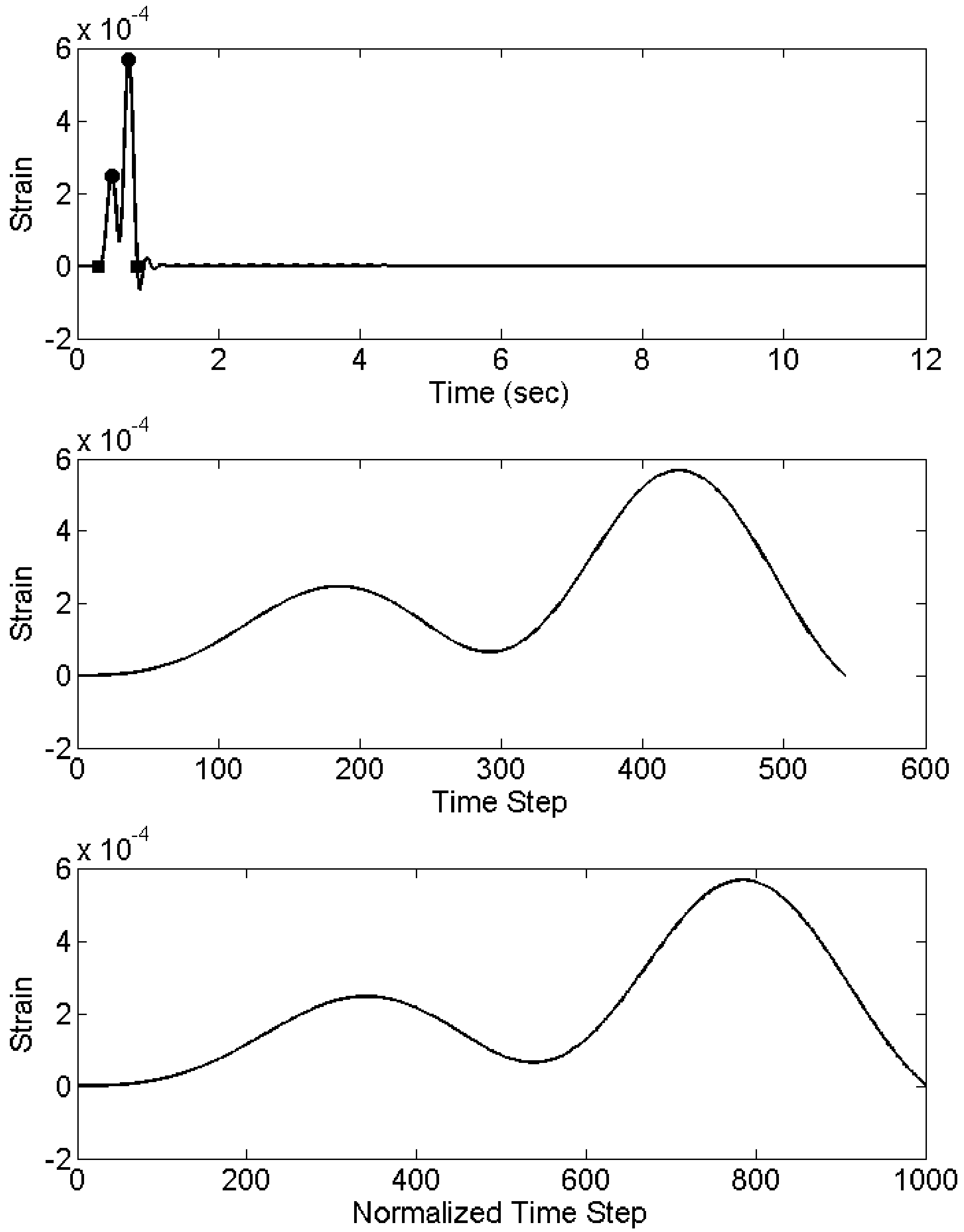


Figure 7.74: Time Scale Normalization of a Simulated Bus Traveling at 40 MPH

Original Time History (top), Section of Interest (Middle) and Resampled (Bottom).

This normalization method was applied to the 500 traffic patterns described in the previous section. From Figure 7.75, for the 100 undamaged scenarios the normalized error is independent of the speed, wheelbase, or axle weight. This is a marked improvement over the previous damage detection method. When applied to the damaged data (Fig. 7.76 and 7.77), the threshold for detecting the presence of damage decreases to 2% (down from approximately 10% in the previous example).

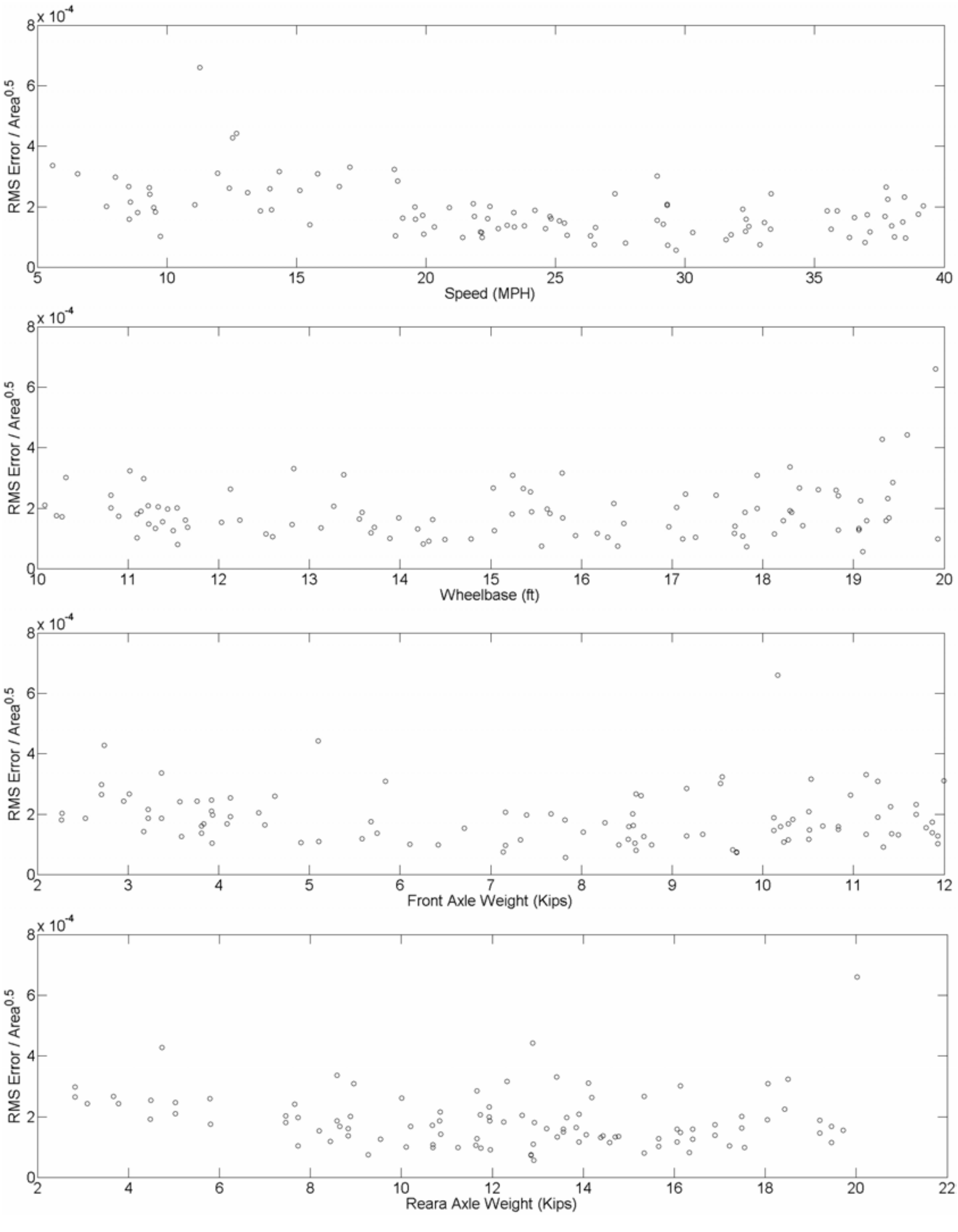


Figure 7.75: Normalized Error versus Speed, Wheelbase, Front Axle Weight, and Rear Axle Weight (Top to Bottom).

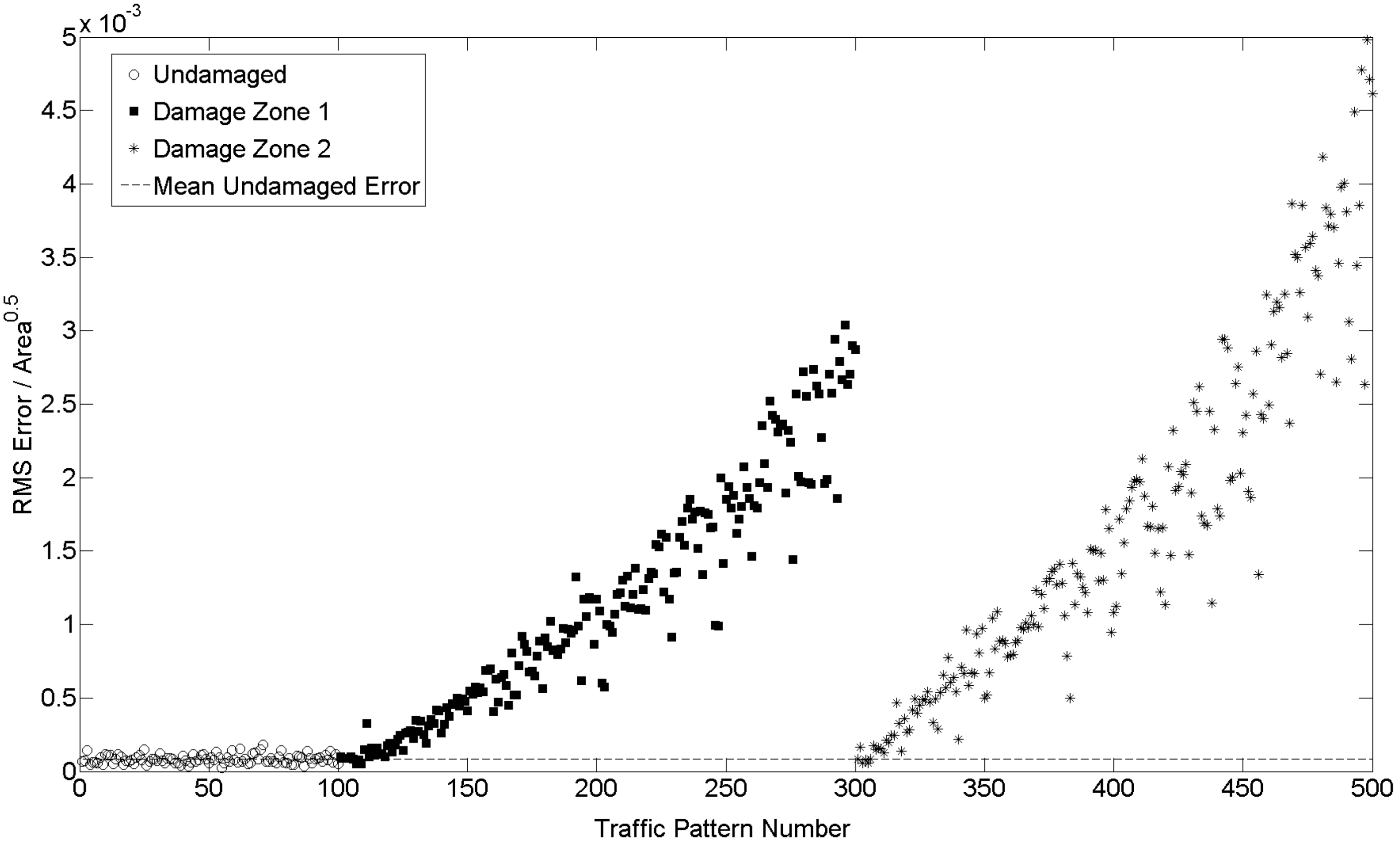


Figure 7.76: Sorted Traffic Scenario Number versus Normalized Error

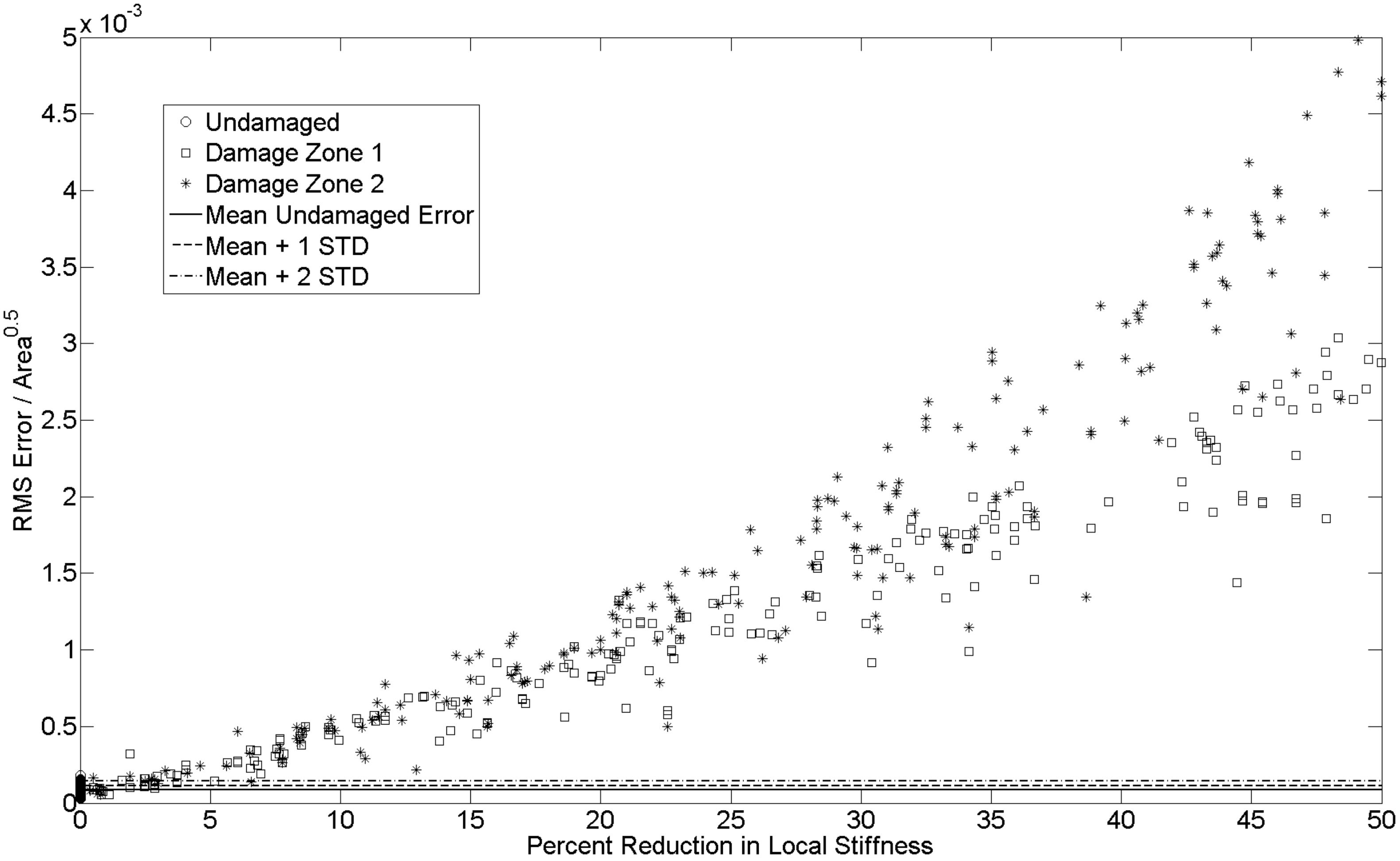


Figure 7.77: Percent Reduction in Stiffness versus Normalized Error

7.9 Summary

For vehicle property estimation using neural networks it was decided to begin with the most straightforward case: in the finite element model one vehicle property (speed or wheelbase) was changed at a time and the neural networks abilities to identify the changes were examined. In this study, strain time histories from middle and quarter span were used along with Principal Component Analysis for feature reduction. Under these conditions, network performance was very good. Next, each vehicle property (speed, wheelbase, front axle weight, and rear axle weight) was treated as a random property and a new set of strain time histories were analyzed. In this exercise, different methods of feature reduction were explored, including peak axle strains and principal component analysis. It was found using peak axle strains (along with the corresponding time at which these strains occur) as inputs gave the best results, and accuracies under random traffic loading of 0.42 MPH for speeds, 0.17 ft for wheelbases, 0.07 kips for front axle weights, and 0.13 kips for rear axle weights were achieved. One limitation with applying this method to recorded strain time histories is the amount of data required for network training.

For damage detection under random traffic loading, neural networks had to be employed. Neural networks were able to detect and classify the level of damage, with 90% accuracy, even when damage was allowed to occur in simultaneous locations. By training separate networks to monitor each damage zone and combining the output from each, the amount of training data required is vastly less than using a single network (which requires many more damage scenarios). However, as with all supervised learning techniques, the more refined the classification levels, the more data required for network training. Therefore, an alternate method of damage detection was explored which did not require any training data from the damaged state of the structure (a big advantage when applying the detection methods to real data). This new method used strain time histories from one location on the FE model of the bridge decks to train a neural network to predict an associated time history at another location. Principal Component Analysis was employed to reduce the number of features in the each of the input and target strain time histories from 4000 to less than 10. As the neural network is trained using data exclusively from the undamaged state when damage occurs, the error between predicted and measured (generated) strains within these damaged regions increases. A normalized error was utilized providing correlation between the percent reduction in stiffness (level of damage) and the difference in the predicted and observed strains, even under random traffic loading. Ultimately, changes in stiffness as low as 2% could be consistently detected, providing a significant improvement over the previously employed methods. Again, as this method does not require data from the damaged structure and can be trained using random traffic patterns, it is easily implemented on actual data from the composite bridge decks.

8 Analysis of Strain Data Recorded on UCSD Bridge Decks

Using the analysis techniques detailed in the preceding chapters, the data recorded on the UCSD bridge-deck panels have been analyzed. Presented within this chapter are the results from the data in the continuous monitoring database (16 strain gages and time synchronized video, see Section 3.3.1 for further details). The first set of calculations involved determining the speeds and wheelbases of passing traffic. Using these calculated speeds, the dynamic amplification of the peak strains resulting from changes in speed was analyzed. Finally, predictive neural networks, similar to those employed in Section 7.8, were used to look for changes in the response of the bridge deck system over time.

8.1 Traffic Analysis Using Strain Time Histories

For calculating speeds and wheelbases of traffic, it was decided to focus on the data from a few select vehicle types from the set of labeled data (Section 4.6). In these exercises, the strain time histories from 5 vehicle types (Passenger Cars, Vans, UCSD Bus Type 5, UCSD Bus Type 8, UCSD Bus Type 9) were used. The number of available records, minimum and maximum channel 9 peak strains, and typical image are summarized in Table 8.1 By performing these calculations on one vehicle type at a time, it is possible to examine the error in the calculations based on the histograms of calculated wheelbase. Since only one vehicle type is used at a time, all of the resulting wheelbases should have the same value. The scatter in this data directly correlates with the level of error in both the speed and wheelbase calculations. For monitoring the bridge deck response over time with the series of neural networks, the data set contained a more general collection of vehicles. This strain data was generated by typical two-axle vehicles and is discussed later in this chapter. A flowchart outlining the sensor data analysis is provided in Figure 8.1.

Table 8.1: Vehicle Types Used in Speed and Wheelbase Calculations

Vehicle Type	Sample Image	Number of Recorded Crossings	Minimum Channel 9 Peak Strain	Maximum Channel 9 Peak Strain
Passenger Cars		1147	5.232 $x10^{-6}$	6.686 $x10^{-5}$
Vans		220	5.814 $x10^{-6}$	9.302 $x10^{-5}$
UCSD, Bus Type 5		253	7.558 $x10^{-6}$	1.517 $x10^{-4}$
UCSD Bus Type 8		919	6.976 $x10^{-6}$	1.238 $x10^{-4}$
UCSD Bus Type 9		1143	9.302 $x10^{-5}$	1.651 $x10^{-4}$

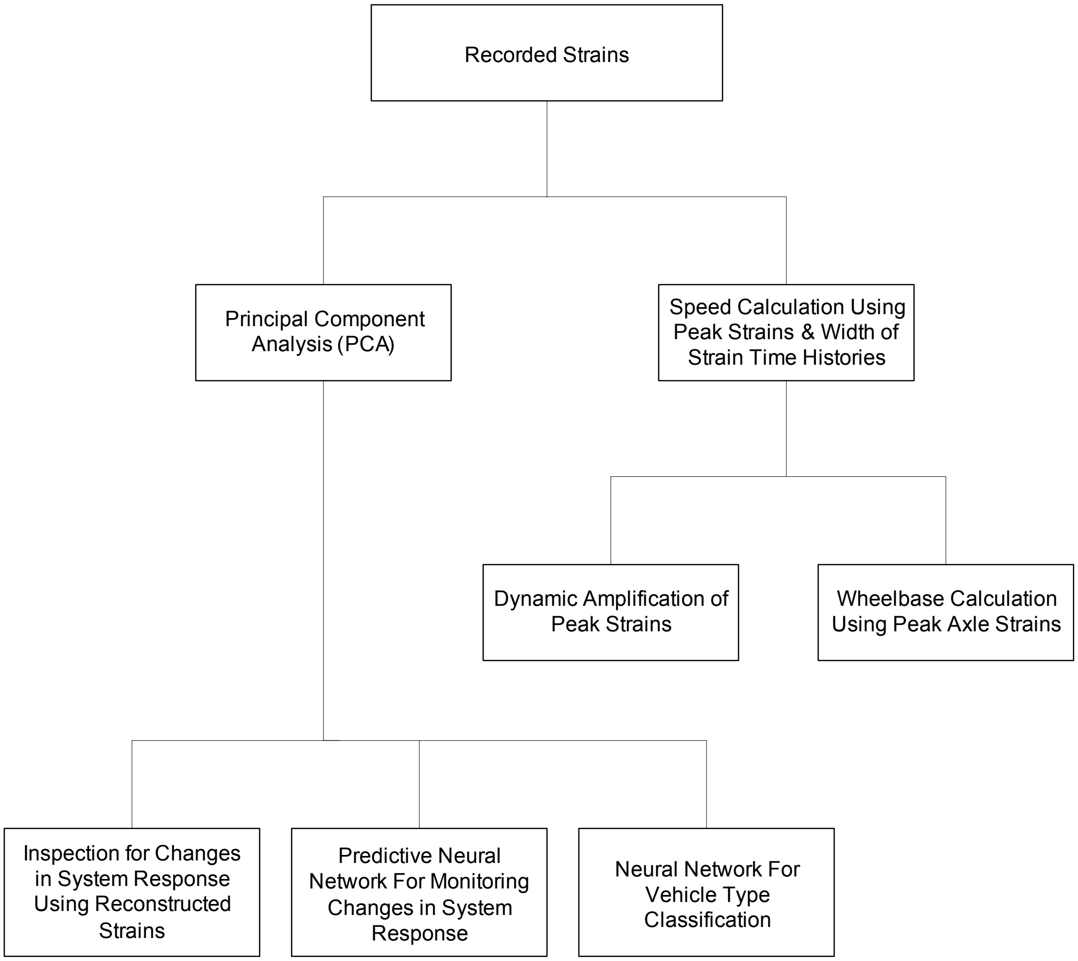

Recorded Strains
Principal Component Analysis (PCA)
Speed Calculation Using Peak Strains & Width of Strain Time Histories
Dynamic Amplification of Peak Strains
Wheelbase Calculation Using Peak Axle Strains
Inspection for Changes in System Response Using Reconstructed Strains
Predictive Neural Network For Monitoring Changes in System Response
Neural Network For Vehicle Type Classification

Figure 8.1: Flowchart for Sensor Data Analysis

8.1.1 Calculating Vehicle Speeds from Labeled Data

The strain time histories from the aforementioned 5 vehicle types were used to determine the speeds of each of these vehicles. These calculations use the records from the quarter span and middle span sensors installed on Deck 2 (gages 8 and 11, Fig. 8.2). Typical time histories from these two strain gages are shown in Fig. 8.3. Prior to analyzing the data, it is important to understand the shapes of these time histories and how they are formed. Figure 8.4 indicates 6 key locations that control the shape of the time histories. These points numbered from 1 to 6 (left to right) are the :

1. Outer left edge of the decks,
2. Beginning of the free span,
3. Quarter span,
4. Middle span,
5. End of the free span,
6. Outer right edge of the decks.

From inspection of the time histories (Fig. 8.5), it can be seen that when an axle is located between points 1 and 2, strain gages mounted on the underside of the bridge decks (like gages 8 and 11) will experience negative strains, resulting from reverse bending. Once the axle passes point 2, the gages will begin to measure positive strains. These strains increase until reaching their peak value when the axle is situated directly over the gage. For strain gage 8, this is when the axle is over the quarter span location and over the middle span location for gage 11. After crossing the strain gage, the strains decrease reaching zero when the axle passes over point 5. Between points 5 and 6, the gages again record negative strains which eventually tend to zero when the axle moves past point 6 and off of the bridge decks.

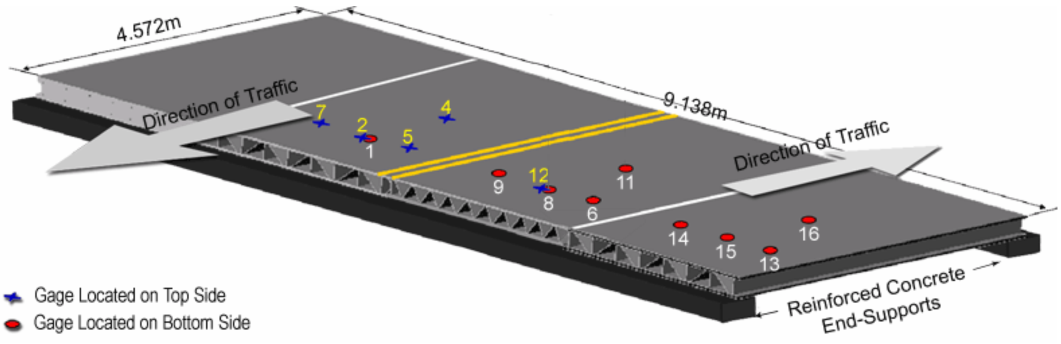


Figure 8.2: 16 Channel Strain Gage Array

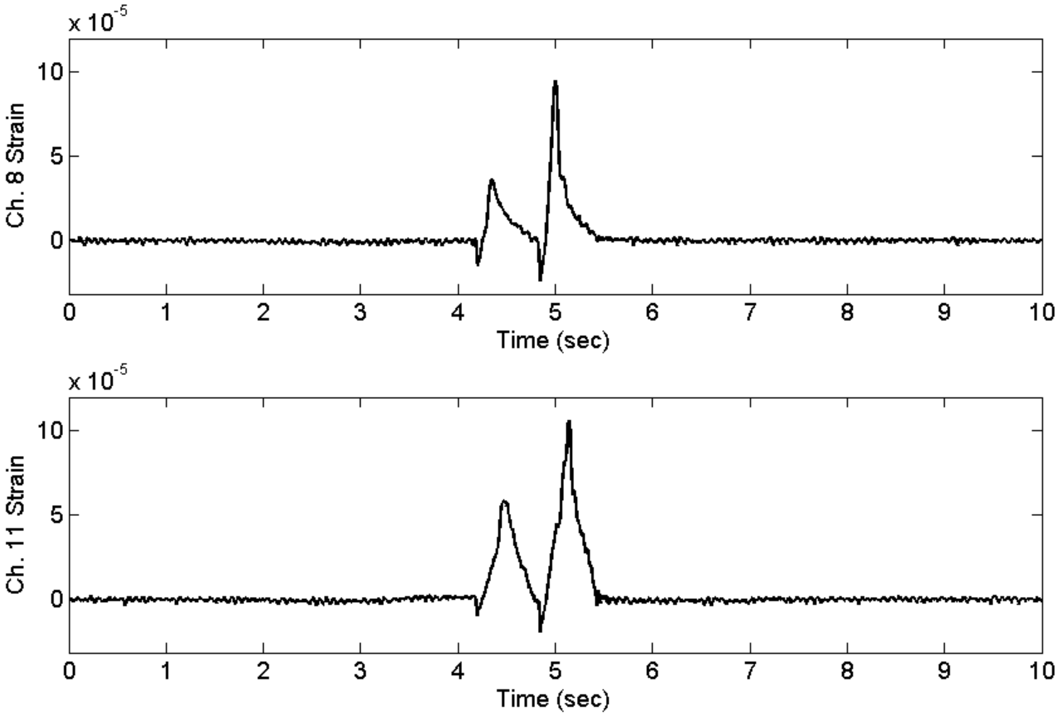


Figure 8.3: Typical Quarter Span (Channel 8, Top) and Middle Span (Channel 11, Bottom) Strain Time Histories

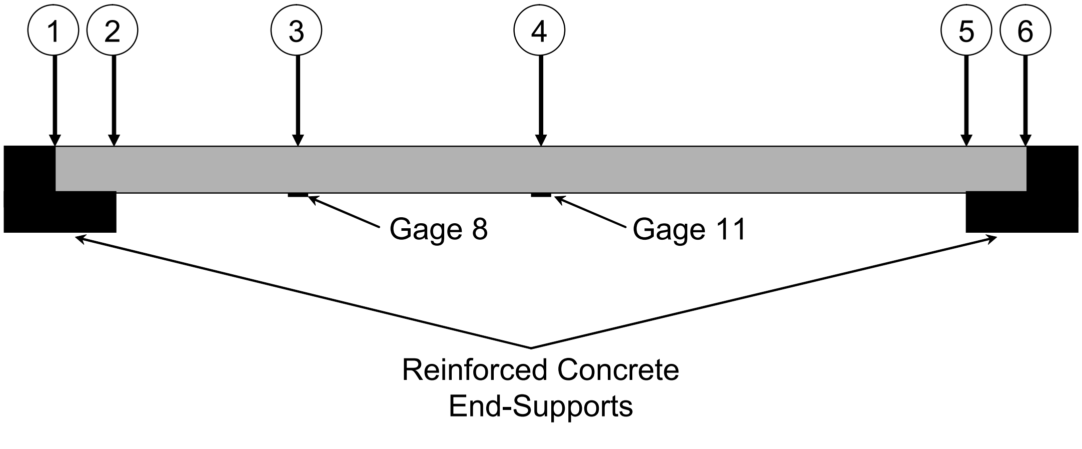


Figure 8.4: Side View of Composite Bridge Decks

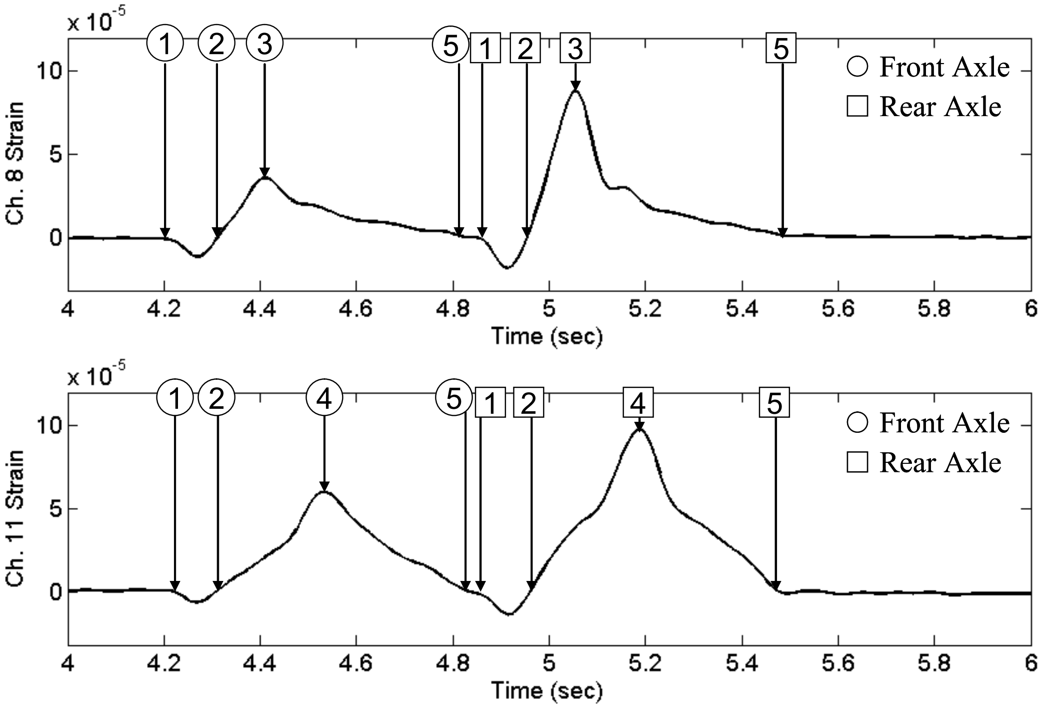


Figure 8.5: Close-up of Strain Time Histories with Indicators for Crossing Significant Locations

As peak strains are registered when an axle passes over the sensor, this provides the first means of calculating speeds. By measuring the difference in time between when the first axle passes over the two strain gages it is possible to determine the vehicle's speed. This is done with the knowledge that the two gages are spaced 3.75 feet apart. Therefore, by dividing this distance between the two sensors by the difference in arrival time, the speed is found. As mentioned previously, the accuracy of this method depends heavily on how accurately these peaks can be found. Because of the modest sampling rate of 200 Hz used, for a vehicle traveling at the speed limit of 25 MPH being off by just one time step in detecting the peaks causes an error of approximately 1.14 MPH or 4.5% in the speed calculation. This high level of sensitivity is an unfortunate disadvantage of the close sensor spacing.

A second method for calculating speeds, which is not as sensitive to the exact time used in the calculation involves measuring the amount of time it takes an axle to cross over the unsupported portion of the bridge decks, between points 2 and 5 in Figure 8.3. Since this method relies on strains recorded over a longer distance, it is less effected by small errors in determining exact arrival times. For the same example of a vehicle traveling at 25 MPH, a one time step error only produces a 0.28 MPH (or 1.1%) error.

The steps involved in both of these calculations are now presented in the quarter span strain time histories shown in Fig. 8.6 and middle span time histories in Fig. 8.7. Both methods first require the peak strains corresponding to each axle be found. Because of the high frequency vibration associated with the structure vibrating at its natural frequency of 22.4 Hz, a digital low pass Butterworth filter (5th order roll off and 5 Hz cutoff frequency) was applied to the data prior to determining the peak strains. Using the same Matlab-based program described in Section 6.1, peak axle strains were first found in this filtered data (top figures in Fig's 8.6 and 8.7). These peak values are indicated by the filled circles in these figures. To avoid the nonlinear effects of the phase angle shift in the time histories (resulting from filtering) on the speed calculations, the corresponding maximum strains in the original time histories were found (2nd plot from the top in Figures 8.6

and 8.7). This was done by simply finding the maximum strain (as indicated by the filled circles) in the neighborhood of the peak values in the filtered data. These values from the original data were then used to calculate the set of speeds based on peak axle strains.

To determine the speeds based on the amount of time required to cross the unsupported portion of the bridge decks, the peak strain corresponding to the vehicle front axle was used to locate the instants in time when the axle reaches points 2 and 5 in Fig. 8.4. This was done by finding the zero crossings around the first peak, a simple procedure easily done in Matlab. Once these two times, indicated by the squares in the 2nd from the bottom plots in Figures 8.6 and 8.7, are determined, the speed is calculated by dividing the clear span of the bridge decks (14 feet) by the time taken to cross this distance.

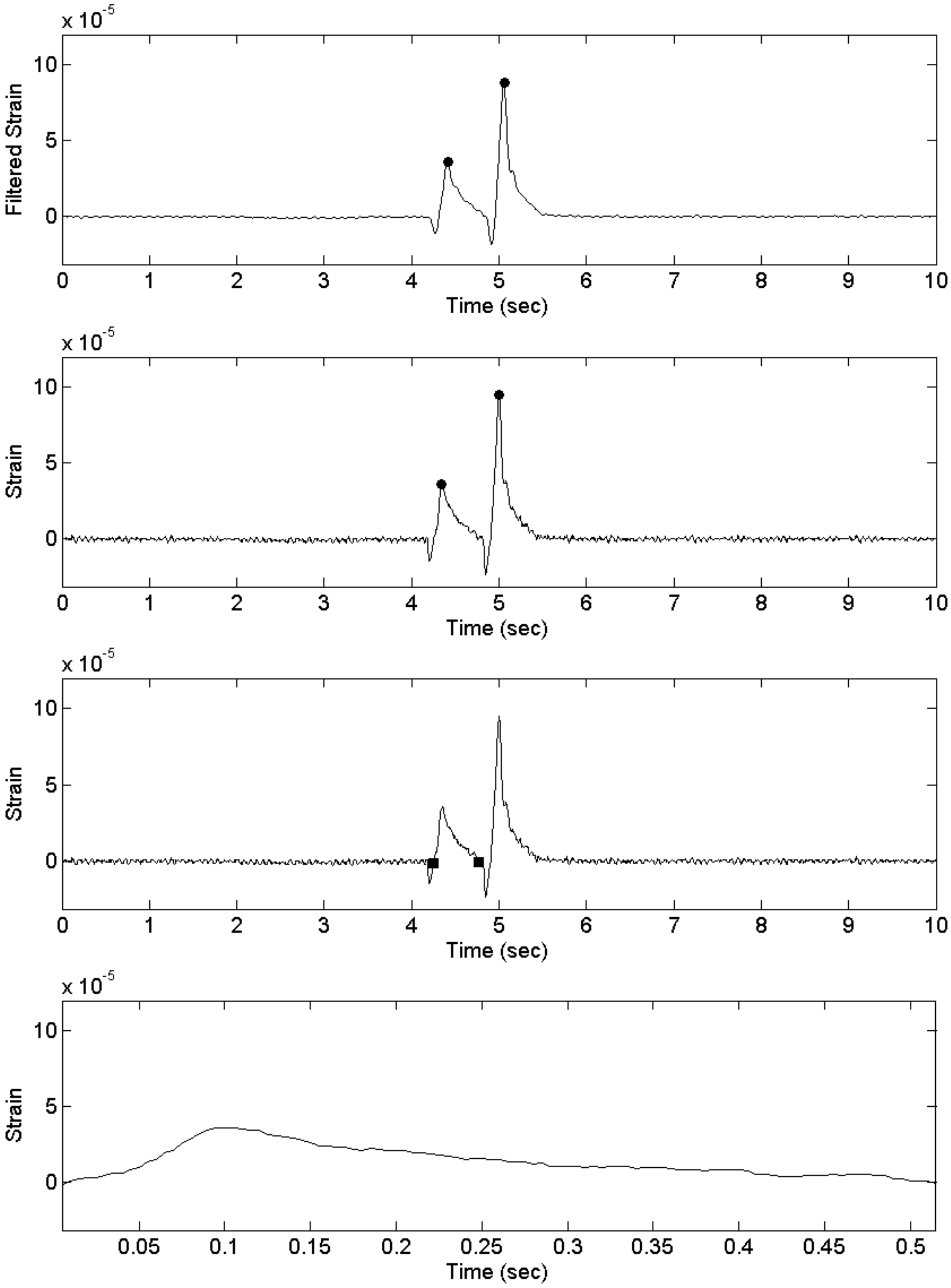


Figure 8.6: Sample Quarter Span (Channel 8) Strain Time History with Indicators for Detected Peak Strains and Times when Crossing Over Supports

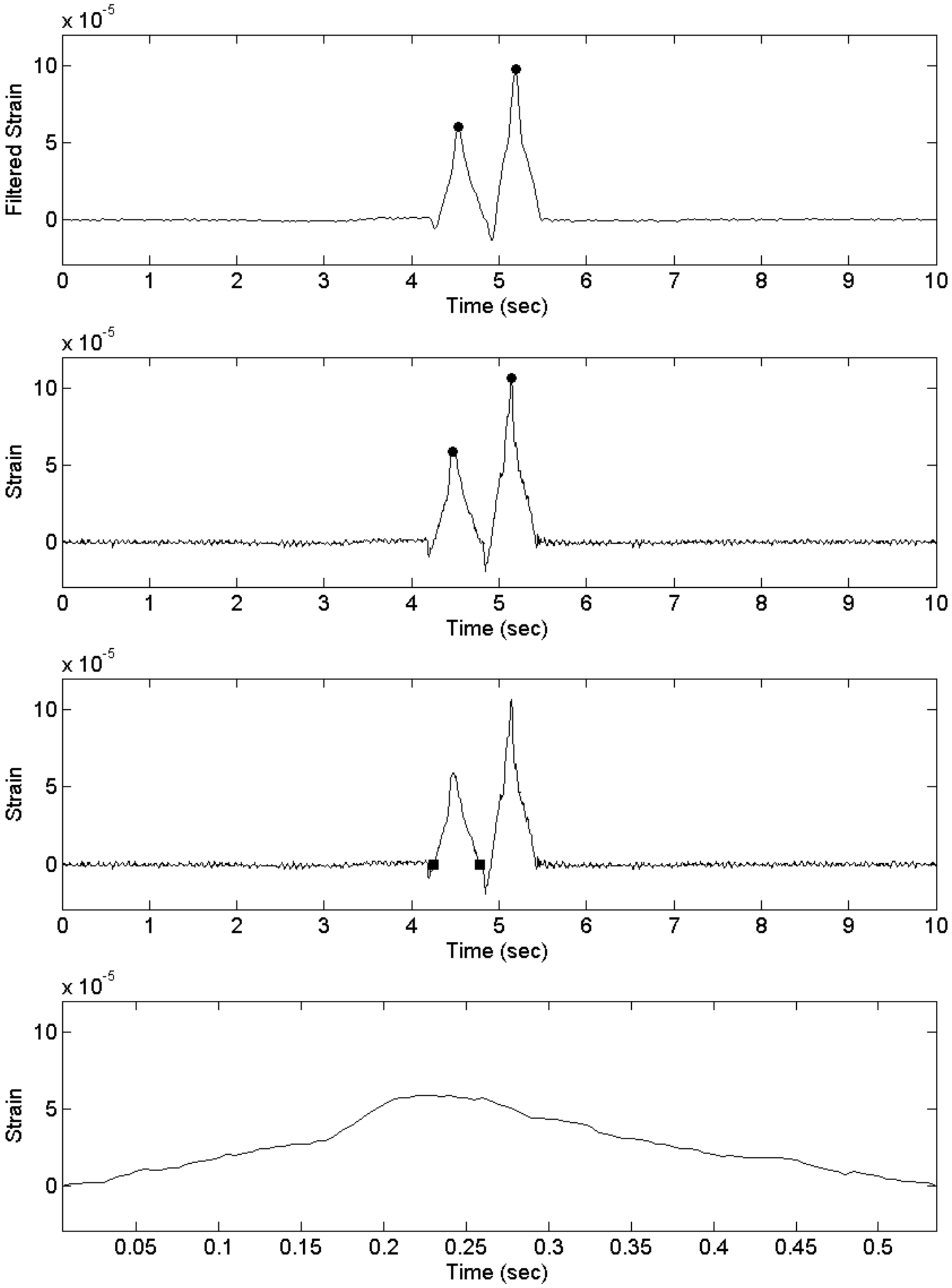


Figure 8.7: Sample Middle Span (Channel 11) Strain Time History with Indicators for Detected Peak Strains and Times when Crossing Over Supports

These two methods were applied to the 3,682 records contained in the 5 vehicle types represented in Table 8.1. For each vehicle type, the calculated speeds from both methods are presented in the Fig's. 8.8 – 8.12.

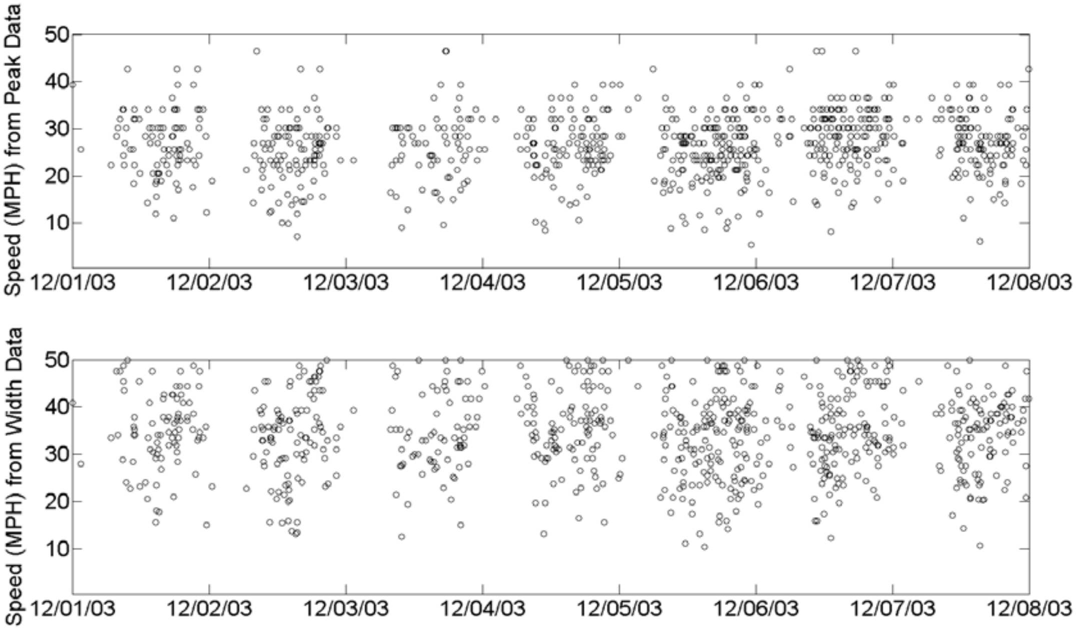


Figure 8.8: Calculated Speeds versus Date for Passenger Cars

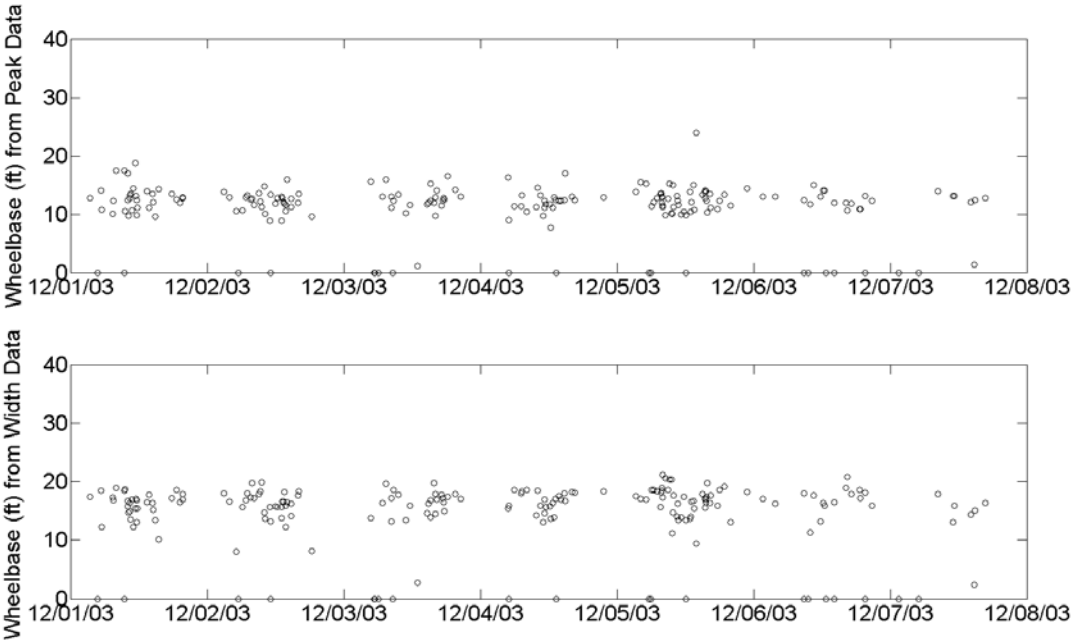


Figure 8.9: Calculated Speeds versus Date for Vans

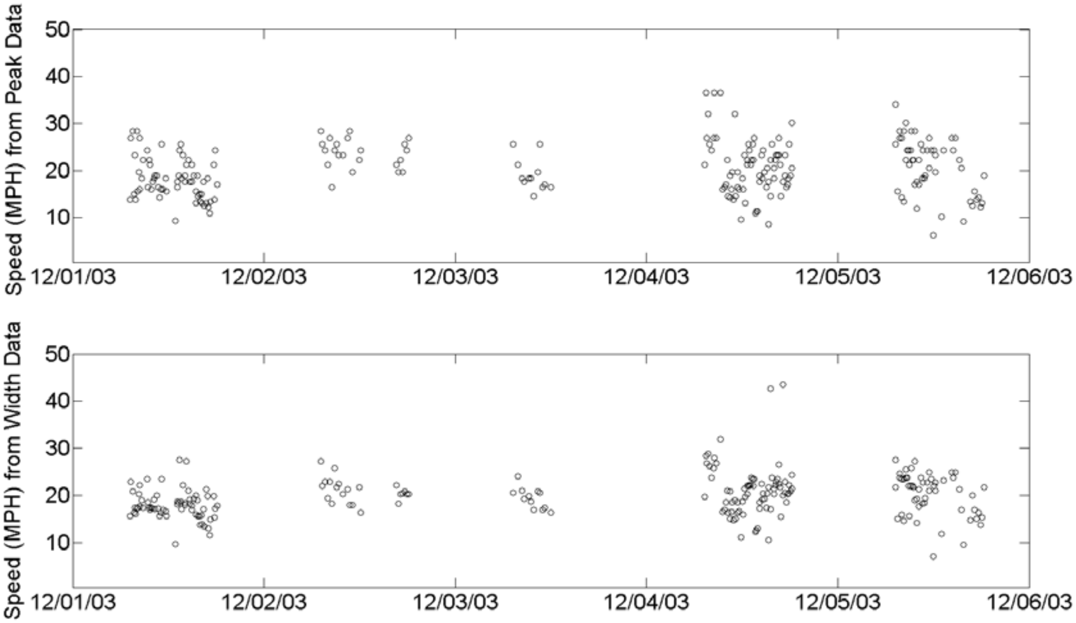


Figure 8.10: Calculated Speeds versus Date for UCSD Type 5 Buses

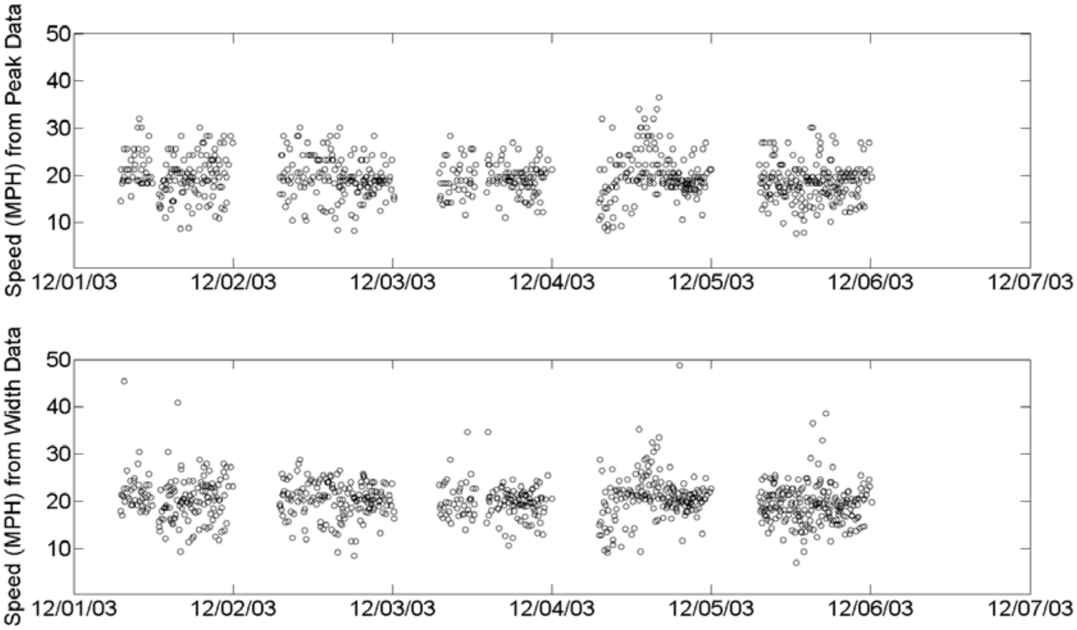


Figure 8.11: Calculated Speeds versus Date for UCSD Type 8 Buses

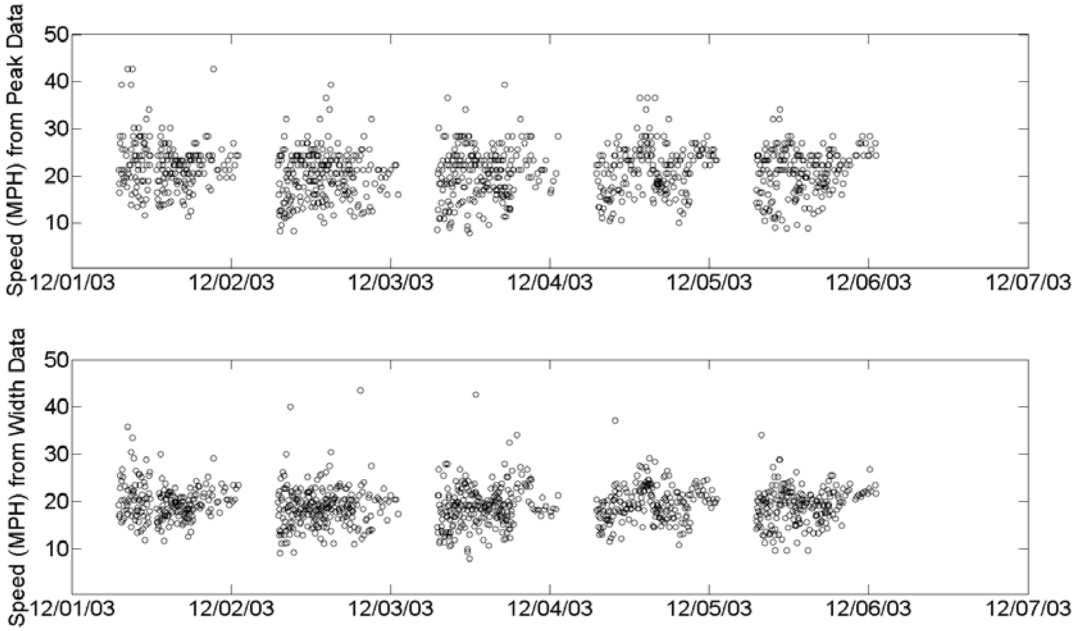


Figure 8.12: Calculated Speeds versus Date for UCSD Type 9 Buses

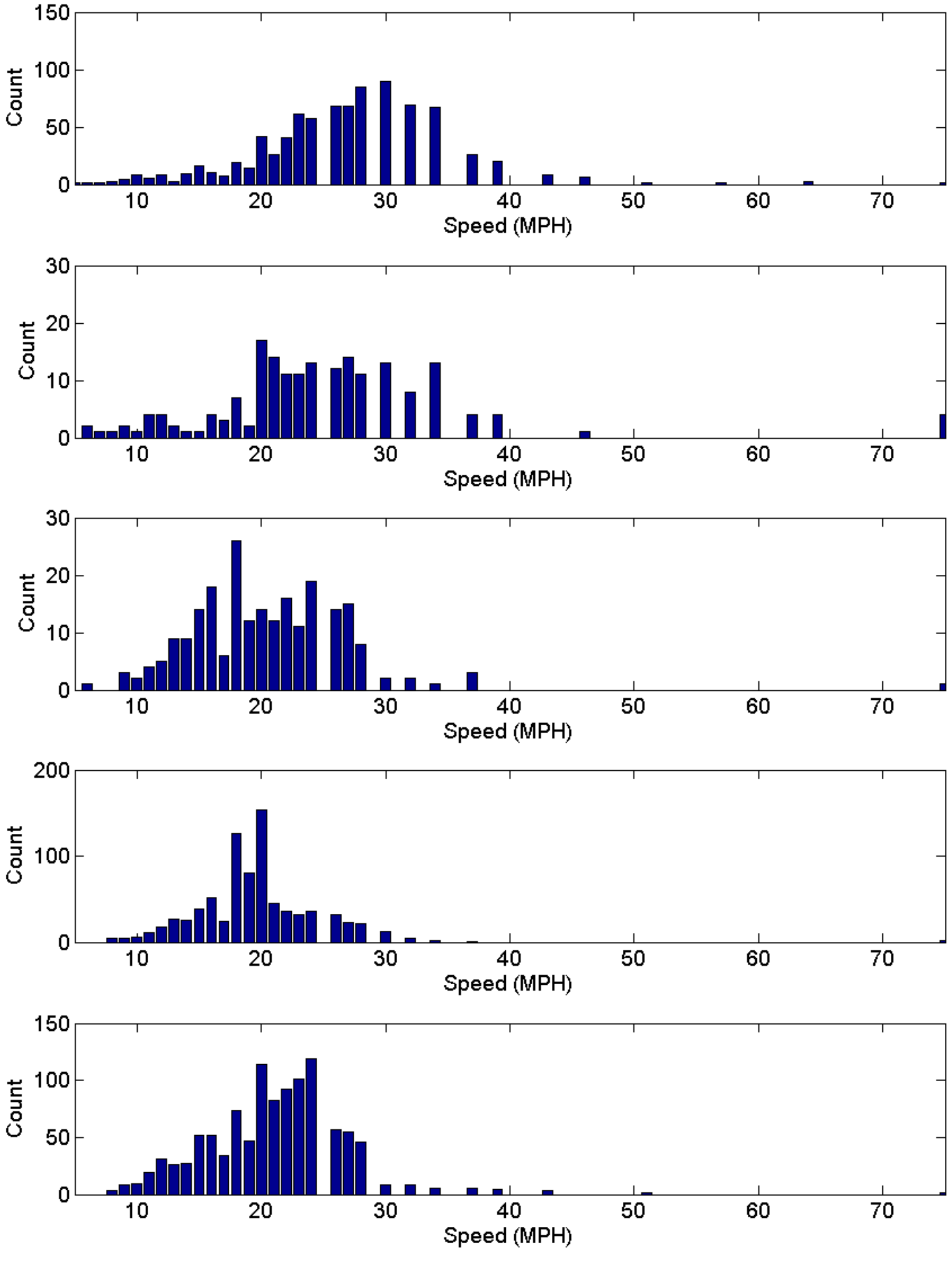


Figure 8.13: Histogram of Calculated Speeds Using Peak Strains for Passenger Cars, Vans, Ucsd Type 5, 8, an9 Buses (Top to Bottom)

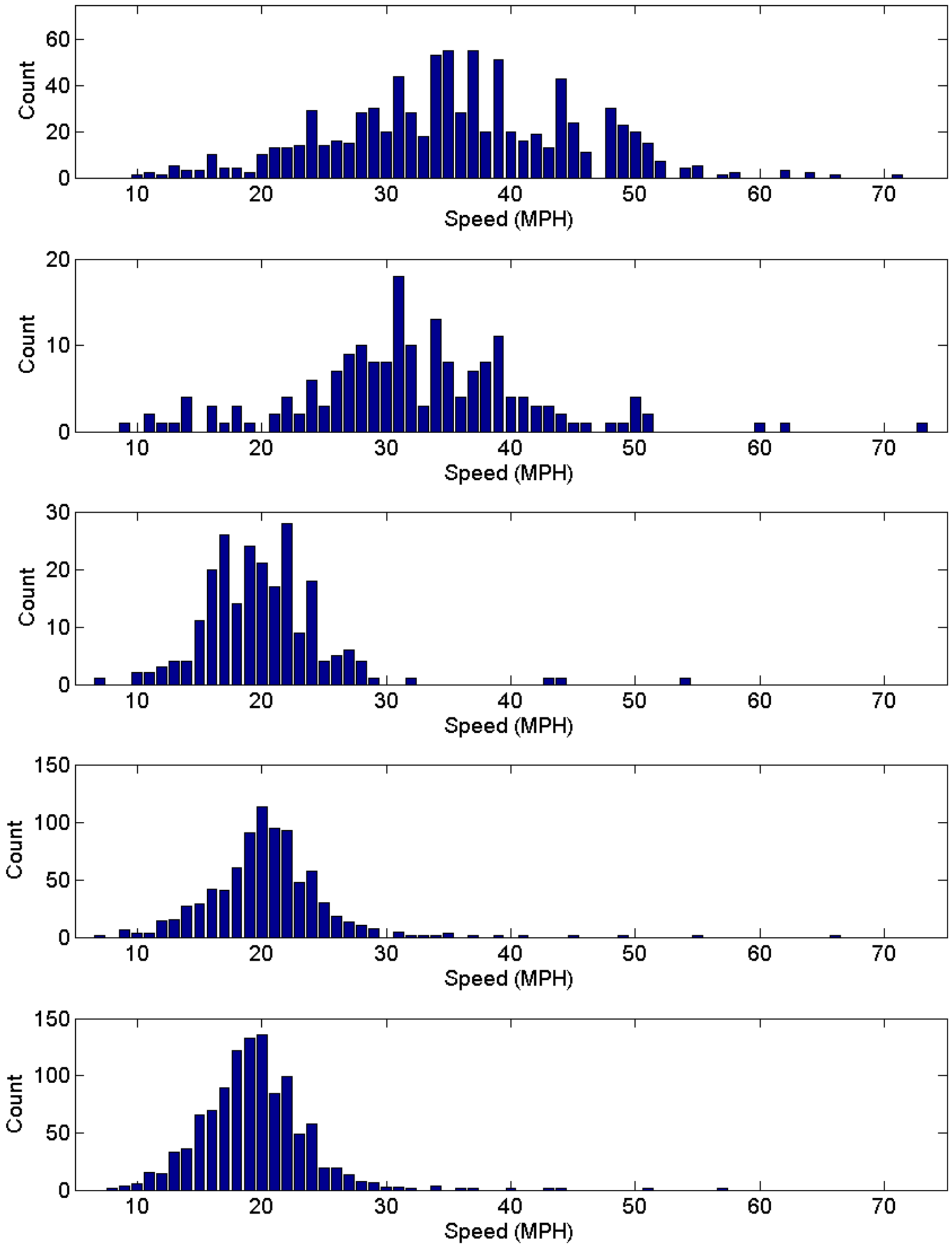


Figure 8.14: Histogram of Calculated Speeds Using Time for First Axle to Cross Bridge Decks for Passenger Cars, Vans, Ucsd Type 5, 8, an9 Buses (Top to Bottom)

8.1.2 Calculating Vehicle Wheelbases from Labeled Data

Having finished calculating speeds in the previous section, these values may now be used to find the wheelbase of each vehicle. Determining the vehicle wheelbase is worthwhile in that it is a useful quantity for:

1. Identifying mislabeled strain time histories in the labeled vehicle type data sets. This is done based on finding outliers in the calculated wheelbases for each class. Some of these outliers are caused by strain time histories which have been incorrectly placed into a vehicle type set (usually attributed due to errors in the time synchronization between the images from the network camera and the recorded strains).
2. Assessing the level of errors in the speed and wheelbase calculations. As many of the labeled vehicle classes are composed of a single vehicle type, the calculated wheelbases for these classes should be constant. The scatter in the calculated wheelbases is directly correlated to the error in the wheelbase and speed calculations.
3. Comparing with the vehicle type classes which have been automatically assigned by a series of neural networks [Yan et al., 2005].

The wheelbase calculations rely on the values of previously determined speeds. Using the data from the middle span strain gage (channel 11) wheelbases are calculated by multiplying the vehicle speed by the difference in time it takes for the front and rear axles to pass over the strain gage. Again as this method relies on peak strains, it is again very sensitive to the accuracy in which the peak values are determined. Plots of the calculated speeds versus wheelbases are shown in Fig's. 8.14 – 8.18 along with the corresponding histograms in Fig's. 8.19 – 8.23.

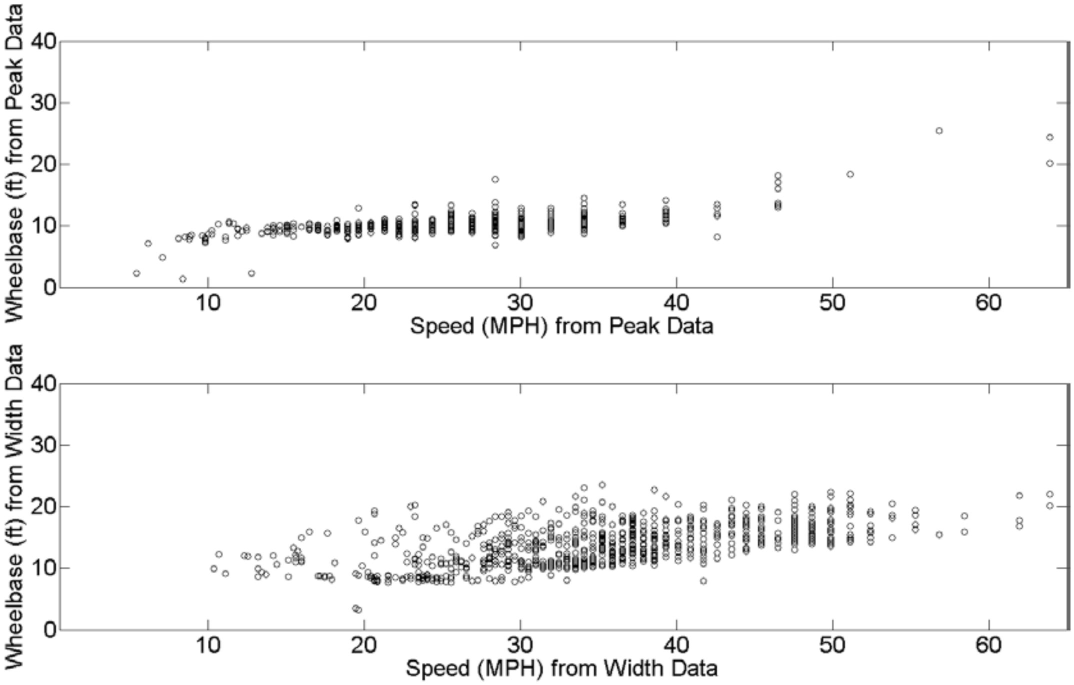


Figure 8.15: Calculated Speed versus Wheelbase for Passenger Cars

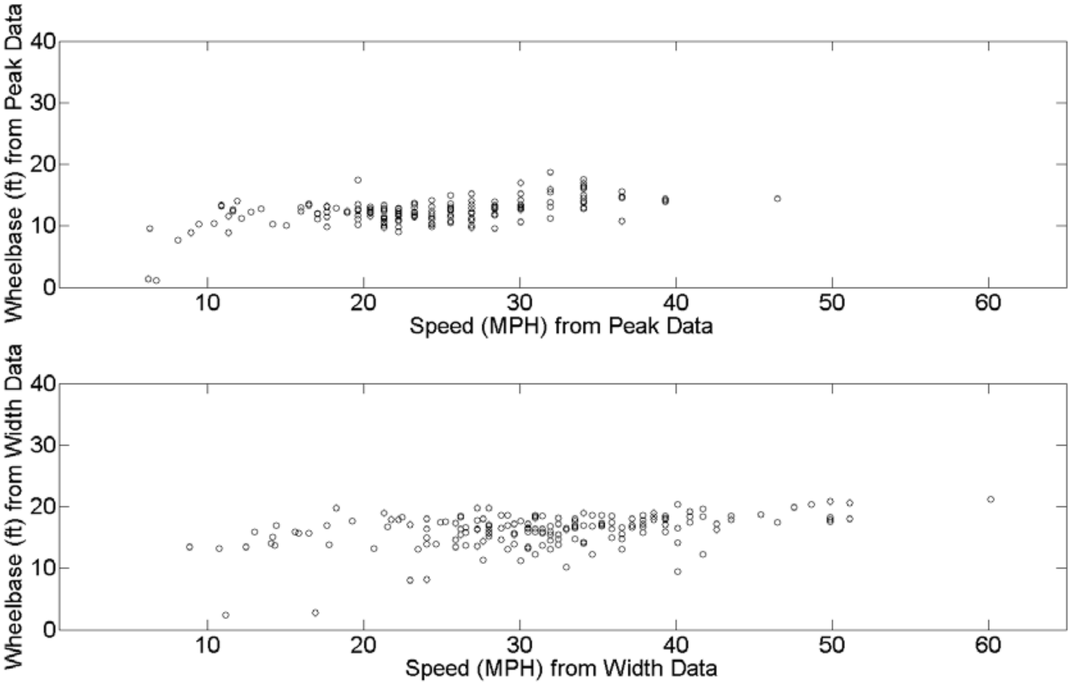


Figure 8.16: Calculated Speed versus Wheelbase for Vans

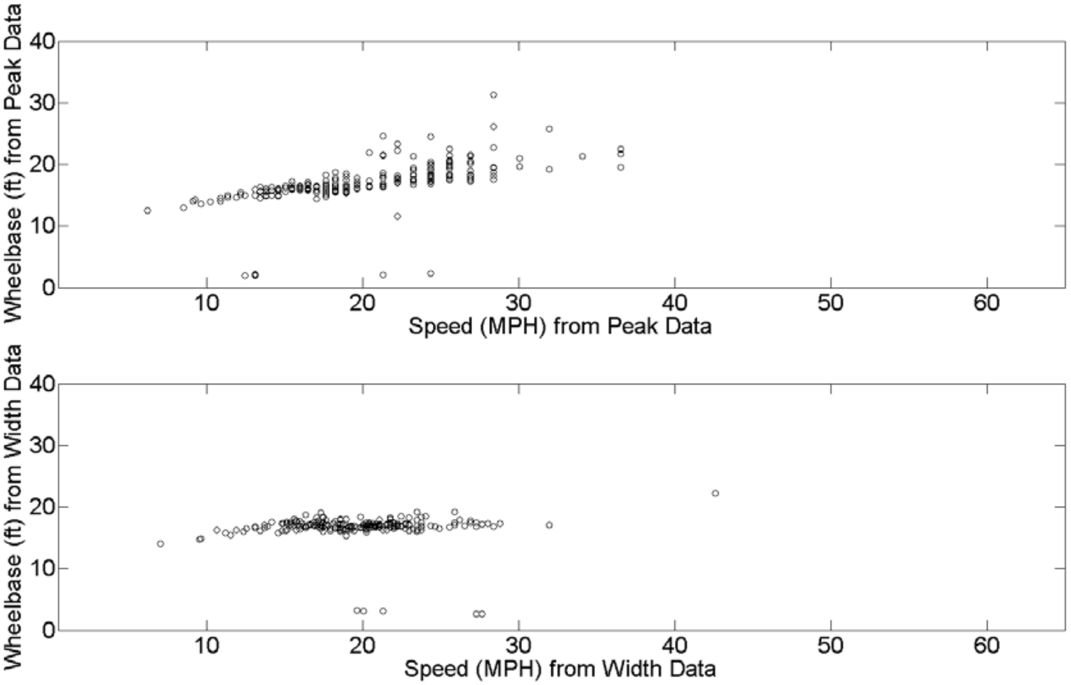


Figure 8.17: Calculated Speed versus Wheelbase for UCSD Type 5 Buses

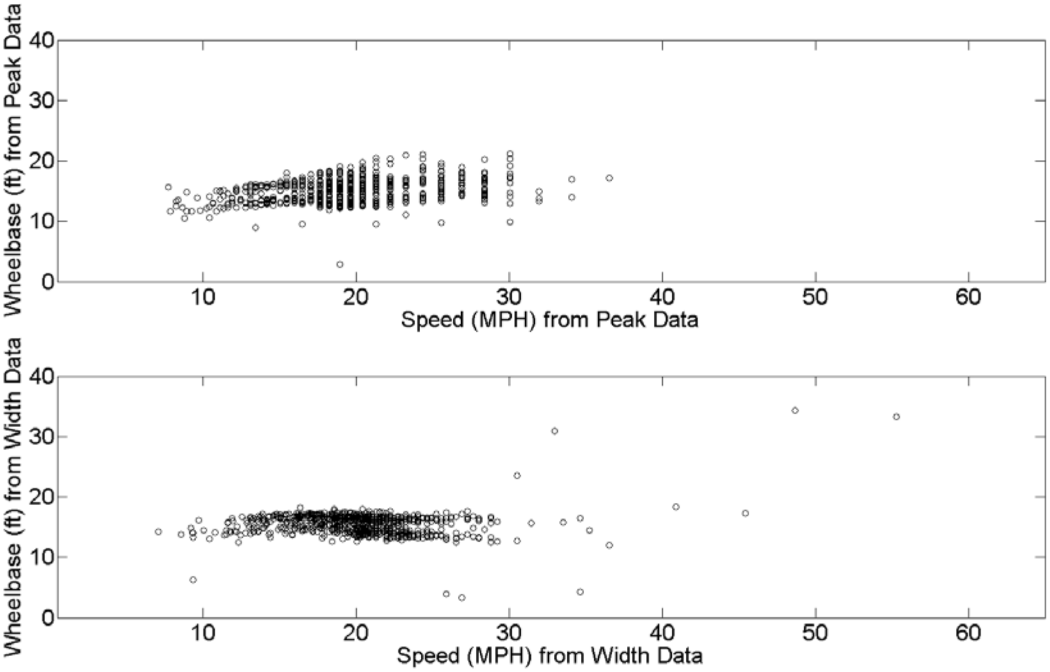


Figure 8.18: Calculated Speed versus Wheelbase for UCSD Type 8 Buses

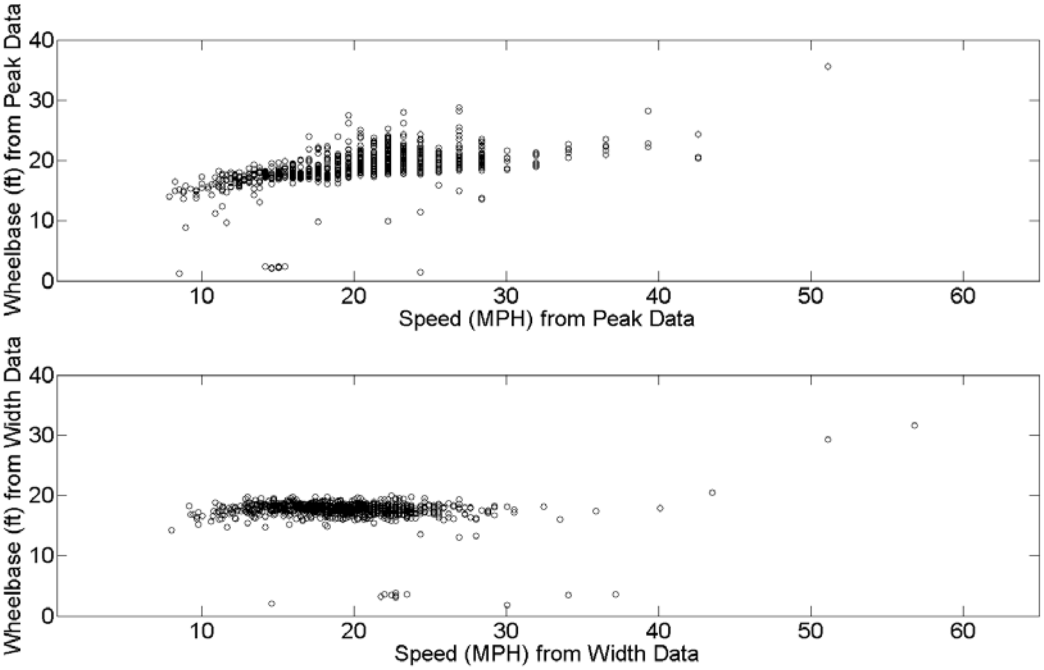


Figure 8.19: Calculated Speed versus Wheelbase for UCSD Type 9 Buses

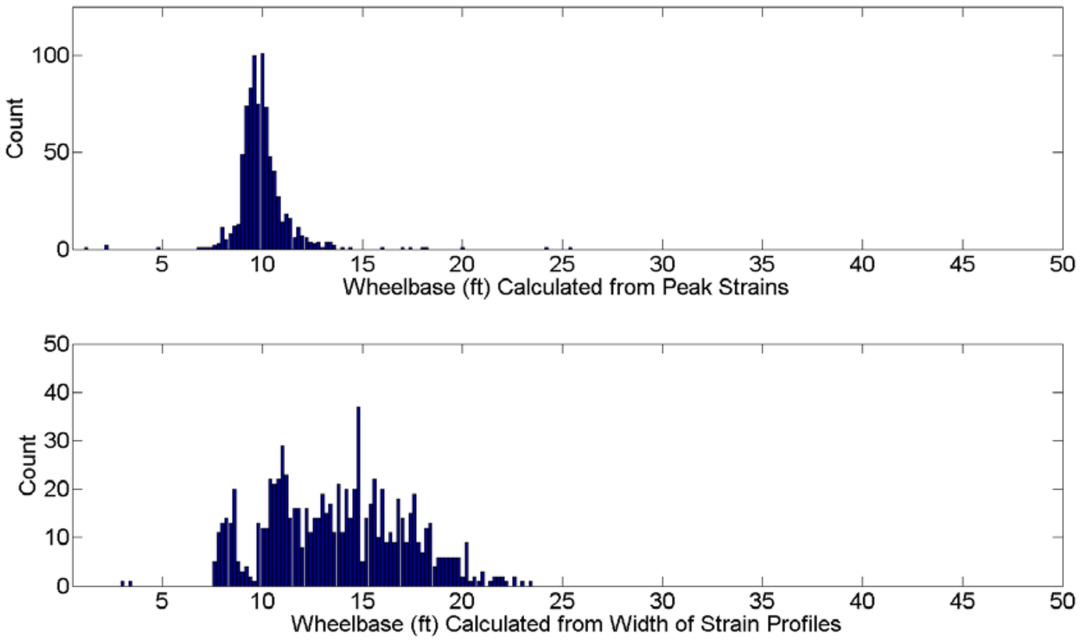


Figure 8.20: Histograms of Calculated Wheelbases for Passenger Cars

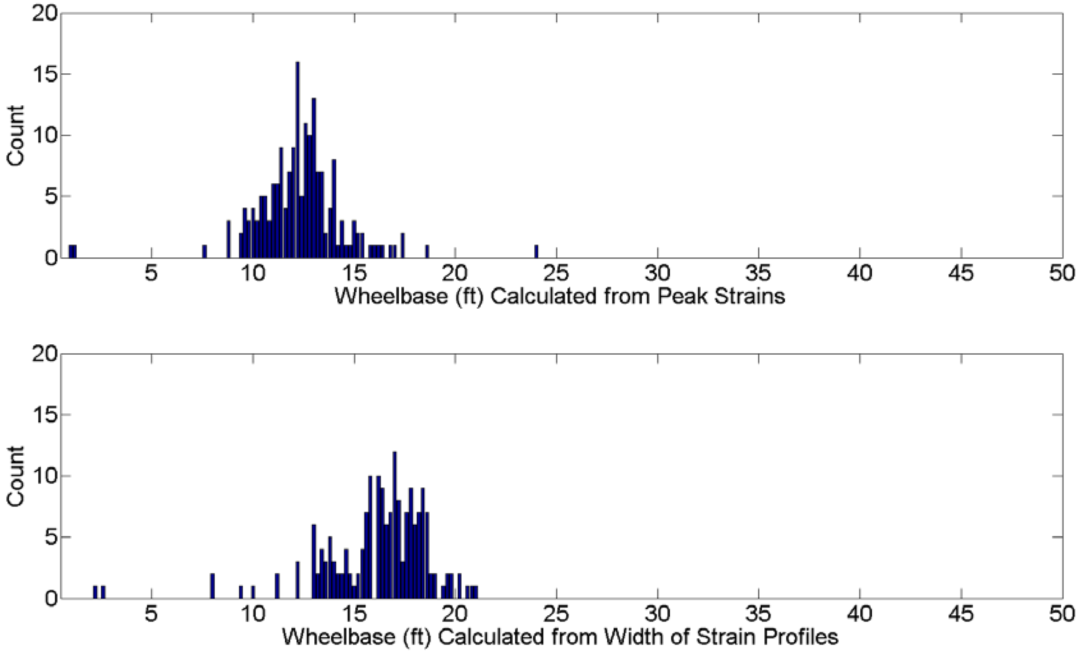


Figure 8.21: Histograms of Calculated Wheelbases for Passenger Vans

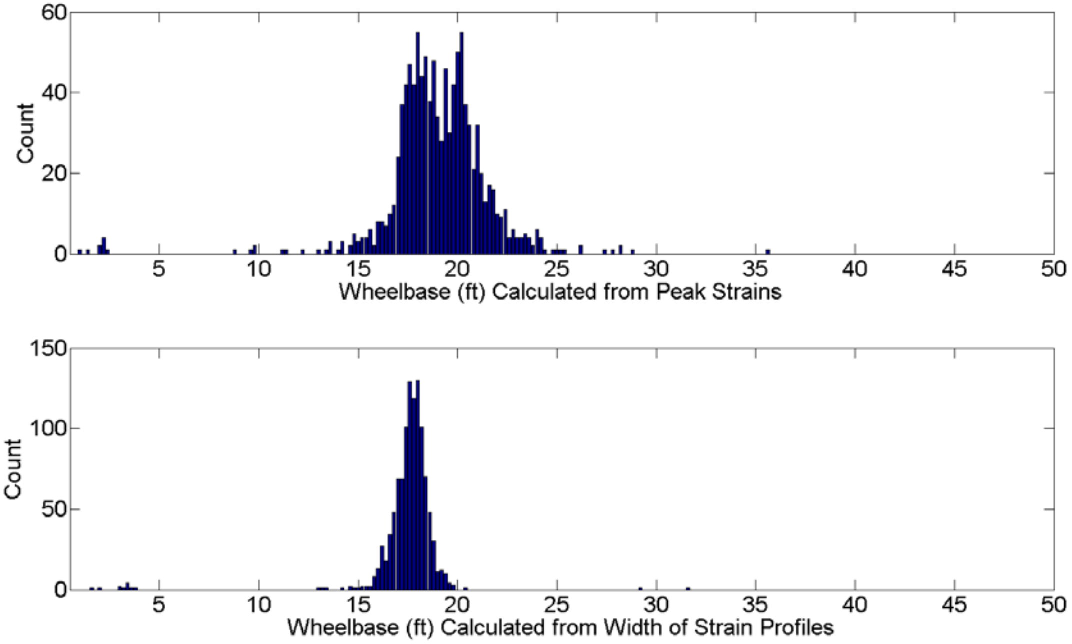


Figure 8.22: Histograms of Calculated Wheelbases for UCSD Type 5 Buses

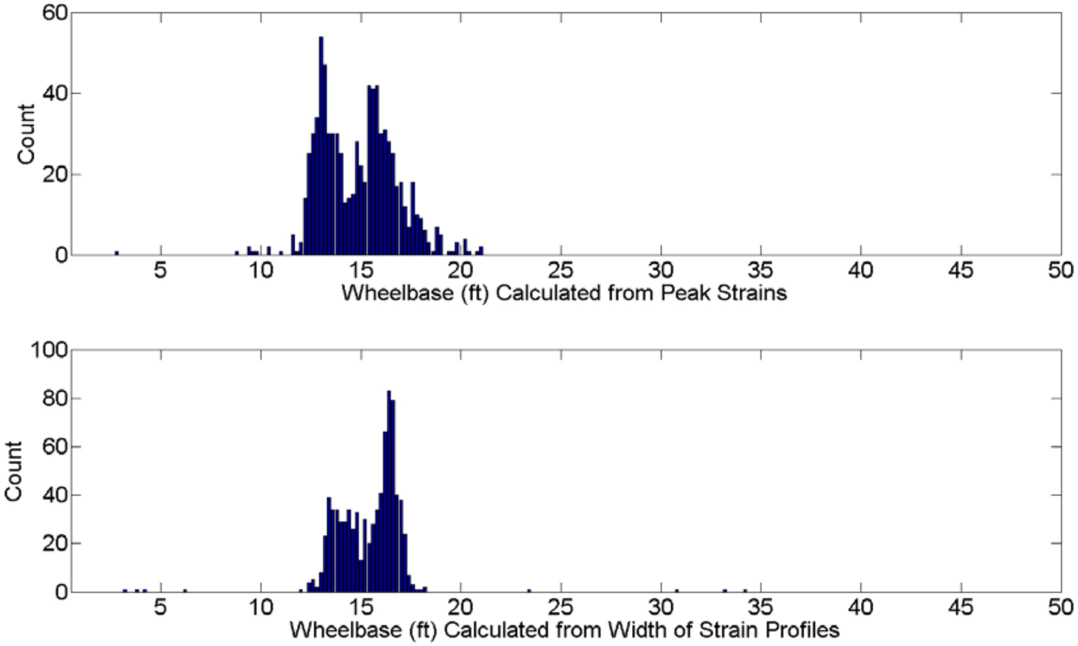


Figure 8.23: Histograms of Calculated Wheelbases for UCSD Type 8 Buses

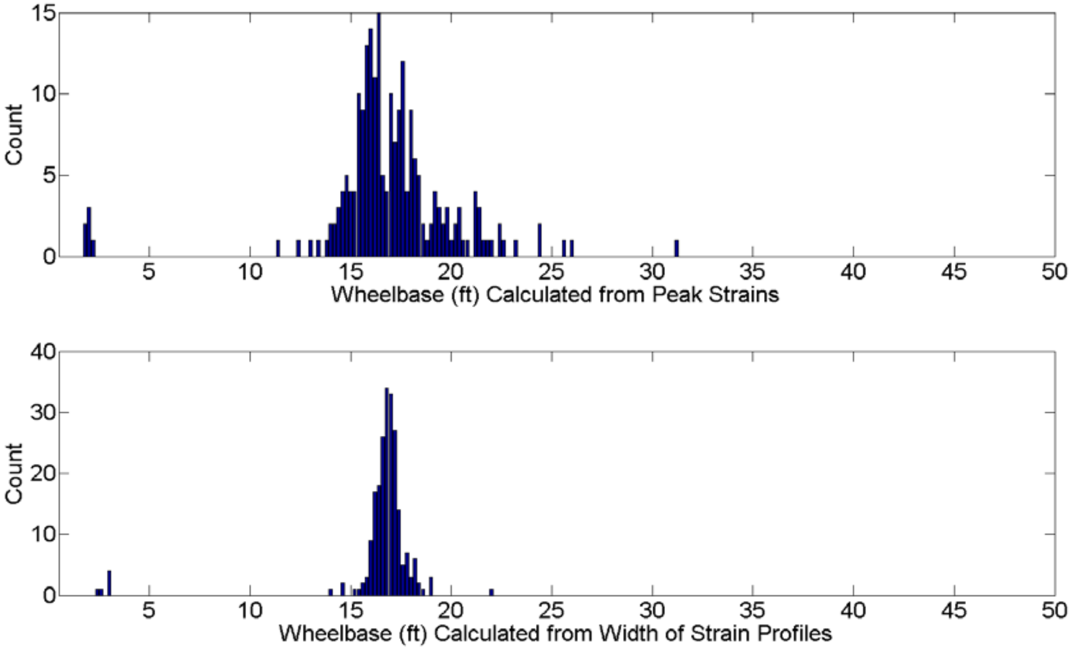


Figure 8.24: Histograms of Calculated Wheelbases for UCSD Type 9 Buses

From these figures, there are a few important observations which may be made. First, calculating speeds and wheelbases based on the time required for each axle to pass over the bridge decks works well for long wheelbase vehicles, but nor for short ones. For vehicles with wheelbases less than 14 feet, the front axle is still on the bridge decks when the rear axle begins crossing. Therefore, as the strains decrease following the peak corresponding to the front axle, the strain is unable to reach zero before the 2nd axle arrives onto the end-support and forces negative strains. Consequently the calculated speeds and wheelbases for these vehicles are too large. This can be clearly seen in 8.24 and 8.25 where the mean wheelbase and the scatter in the histograms are much larger for the calculation based on the width of the strain time histories (rather than peak strains). For long-wheelbase vehicles, the calculations based on the width of the strains is much more accurate. This can be attributed to the methods lower level of sensitivity to small errors in determining the exact instant when the vehicle crosses the strain gages. This is clear in the histograms for the UCSD Type 9 buses. Both methods produce similar mean wheelbases, but the scatter is much less in the second method. From the scatter plots of calculated speeds versus wheelbases (Fig. 8.22) and histograms of calculated wheelbases (Fig. 8.27) for the UCSD Type 8 buses, there appears to be some inconsistency in the sorting of the buses in this class. From these figures, there appears to be buses with two different wheelbases placed together in the same data set. This is something that will need to be investigated later in more detail.

8.2 Peak Strain Analysis

Several factors contribute to determining the remaining service life of bridge decks. Included in these are structural factors like quality of the construction, material factors such as quality of concrete used, and environmental factors. Within the environmental factors, average daily truck traffic is an important consideration. The effects of heavily loaded trucks are critical on the evaluation of deck slabs [Broquet et al., 2004 and Broquet et al, 1999]. Monitoring traffic (particularly for heavy vehicles) is an important aspect of road maintenance and highway safety. Analyzing the effects of truck traffic is essential for modeling pavement wear, tracking fatigue loading, preventing sudden punch failure resulting from overloaded vehicles (Fig. 8.29), and estimating the remaining life expectancy

[Raz et al., 2004 and Oh et al., 2001]. Having information regarding the number and size of vehicles helps with highway safety in that the severity of traffic accidents is related to the size of the vehicles involved [Oh et al., 2001].

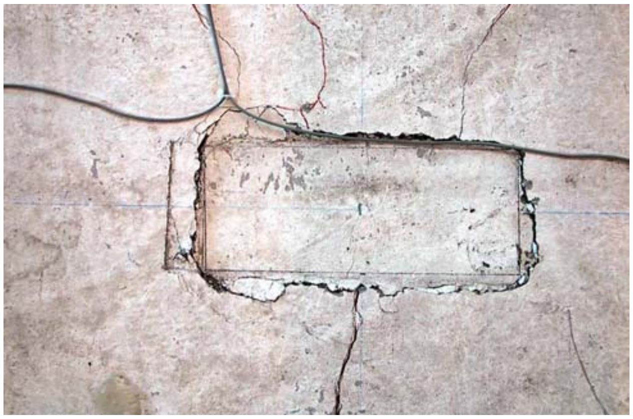


Figure 8.25: Example of Punch Failure on a Concrete Bridge Deck

Inductive loop detectors are commonly used to generate a magnetic signature (maximum magnitude, magnetic length, etc) of passing vehicles. From these signatures, a vehicle type may be assigned using neural networks or a similar artificial intelligence algorithm. From these assigned vehicle labels and the maximum magnetic magnitude recorded, estimates of the axle loads are made. These loads are then applied to a structural model to calculate resulting stresses and strain within the bridge elements. This represents a complex process with many potential sources of error. A more direct method for assessing a vehicle's impact on a structure involves the use of dynamic strain measurements (like those taken on the UCSD composite bridge deck panels). Now, rather than measuring a vehicle property to make estimates of loads and then the bridge response, the strains in the bridge are directly measured, thereby greatly reducing the uncertainty in the bridge response.

An example of using recorded strains from the composite bridge decks is presented in Figures 8.30 and 8.31. These figures are plots of the daily number of vehicles recorded in which the peak strains exceed 5 x10^{-6}, 1 x10^{-5}, 5 x10^{-5}, 1 x10^{-4}, 1.2 x10^{-4}, and 1.4 x10^{-4}. From these figures, it is possible to determine the expected daily number of vehicles causing strains within a certain range. For example, approximately 1000 personal vehicles and 600 buses / delivery trucks cross over the composite decks each work day when school is in session. The large gap in the data is the summer break when campus shuttle buses are not running. Similar smaller gaps are present for the weekends as well as Easter and Christmas holidays. These bands of days in which the traffic volume significantly decreases are important in monitoring traffic volume, particularly for structures like this one in which traffic density follows seasonal trends. A common assumption in structural monitoring is traffic volume remains fairly constant for many bridges [Melhem et al., 2003]. In the case of the composite bridge decks, if a series of traffic measurements using inductive loop detectors were made during one of these periods, the traffic density would be greatly underestimated.

A second set of traffic density plots are included in the six plots of Figure 8.32. These are constructed from the average hourly number of vehicles in which the peak strains exceed the same limits of 5 x10^{-6}, 1 x10^{-5}, 5 x10^{-5}, 1 x10^{-4}, 1.2 x10^{-}

^{4} and 1.4 x10^{-4}. Within these figures, it can be seen that the number of medium and large peak strains significantly increases starting at 7 am when the campus shuttle buses begin running and decreases into the late night as the number of buses operating also decreases.

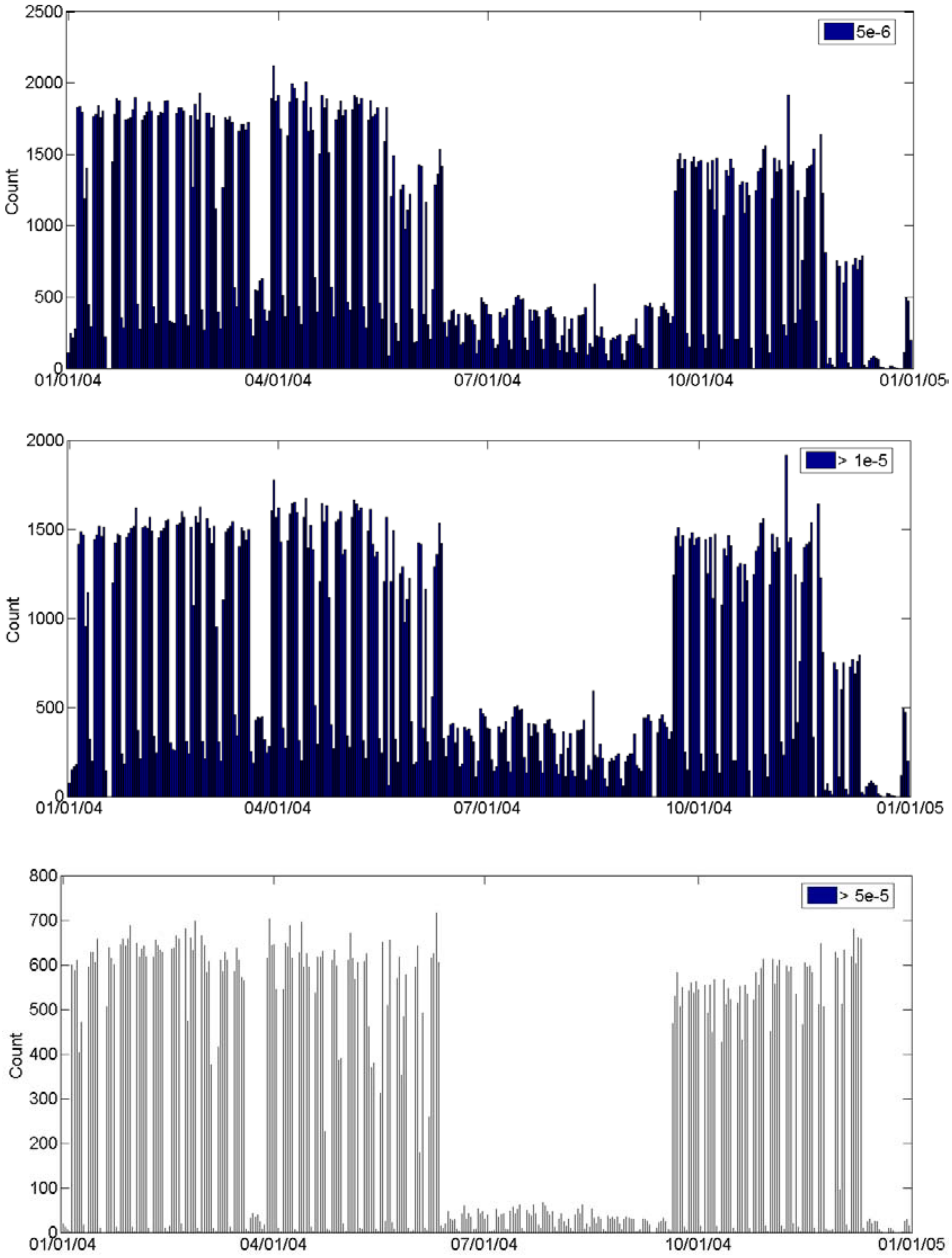


Figure 8.26: Number of Daily Records in which Peak Strain Exceeds 5 x10^{-6} (top), 1 x10^{-5} (Middle), and 5 x10^{-5} (Bottom)

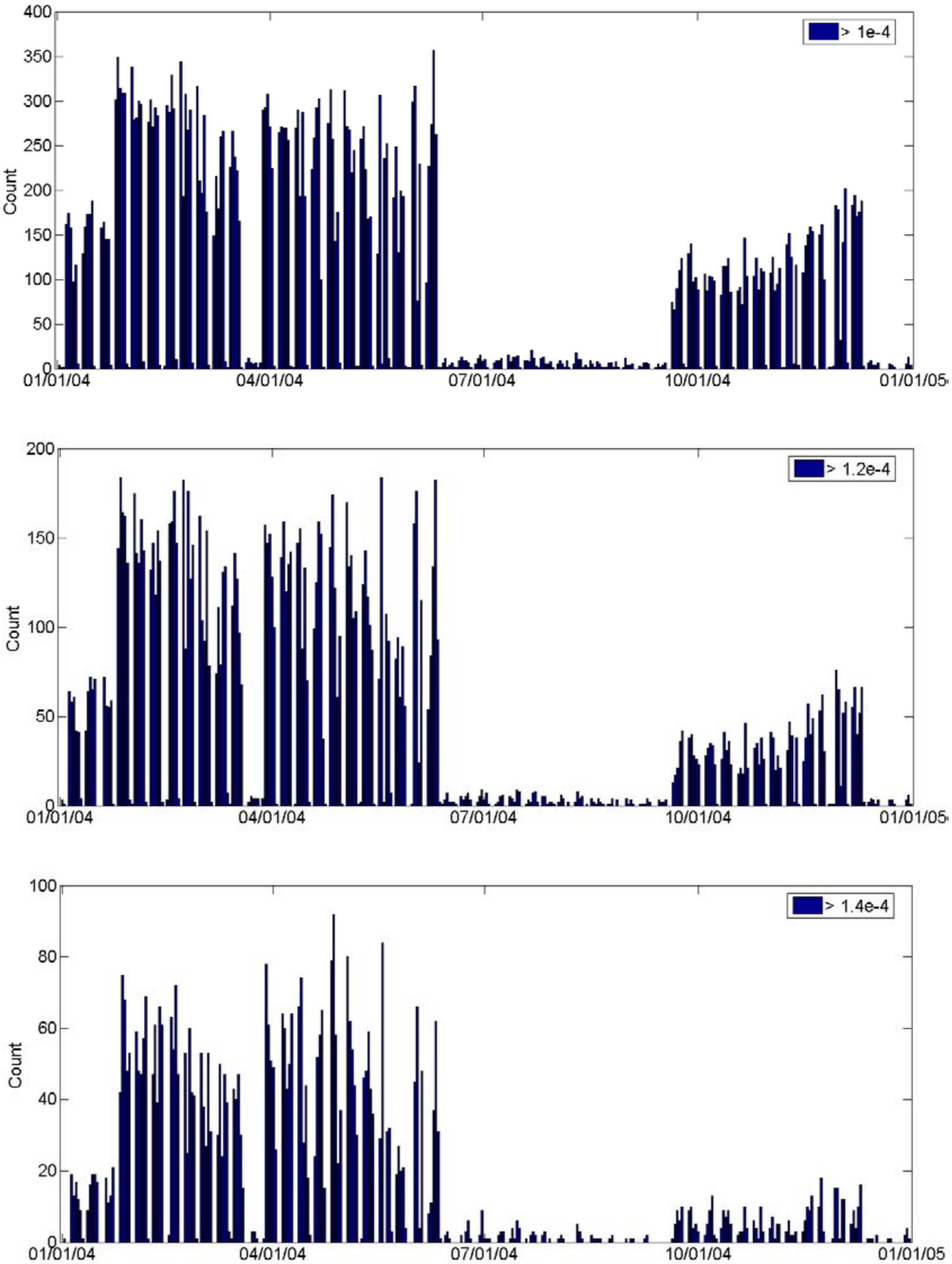


Figure 8.27: Number of Daily Records in which Peak Strain Exceeds 1 x10^{-4} (top), 1.2 x10^{-4} (Middle), and 1.4 x10^{-4} (Bottom)

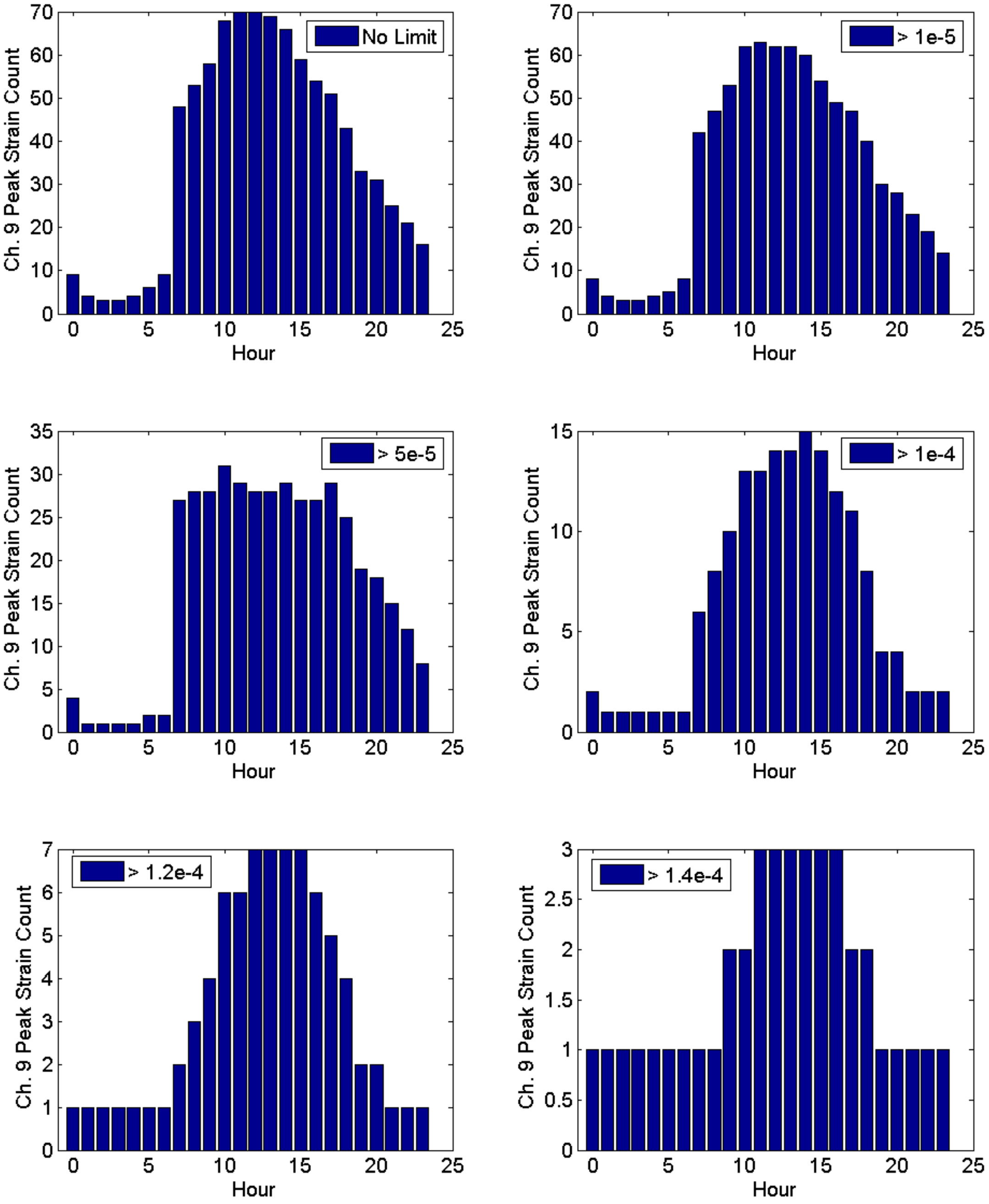


Figure 8.28: Average Hourly Count of vehicles in Which Peak Strains Exceed 5 x10^{-6} (Top-Left), 1 x10^{-5} (Top-Right), 5 x10^{-5} (Middle-Left), 1 x10^{-4} (Middle-Right), 1.2 x10^{-4} (Bottom Left), and 1.4 x10^{-4} (Bottom-Right)

8.3 Dynamic Amplification of Peak Strains

As discussed in Chapter 1, intervention on an existing bridge is typically done based on a series of safety guidelines and criteria. In assessing the dynamic response of the bridge under vehicular loading, dynamic amplification factors (DAF) are usually applied to account for structure/vehicle interaction effects. In this assessment, equivalent dynamic loads are found by multiplying the static vehicle load by dynamic amplification factors. These DAF's are typically generated from measured or simulated data from the global response of the bridge. As noted in Broquet et al., 2004, little research has been done on dynamic amplification factors for bridge slabs. In Broquet et al., 2004, the authors performed a parametric study using numerical simulations to better understand the dynamic behavior of concrete bridge deck slabs and to determine more appropriate DAF's for these systems. The authors argued that heavily loaded trucks govern the safety assessment of bridge deck slabs and the application of inappropriate DAF's may have significant financial implications.

A study was made to apply to data recorded from the UCSD composite bridge-deck panels for determining the dynamic amplification factors resulting from changes in a vehicle's speed and position of the vehicle on the bridge decks. However, because of limitations associated with perspective problems with the installed camera, the exact position of the vehicle on the bridge decks could not be properly ascertained. Consequently, only the peak recorded strains versus speed could be investigated. The labeled vehicle type data set utilized in Section 8.1 of this paper was a natural choice for this study. As each labeled data set is composed of only one vehicle type, the variation in the individual vehicle properties (wheelbase and axle weights) is minimized. The peak strains recorded at quarter and middle span on Deck 2 (respectively by strain gages 8 and 11 in Figure 8.2) were plotted against the speeds calculated in Section 8.1.1. The results for the five vehicle types (passenger cars, vans, UCSD Type 5 Buses, UCSD Type 8 Buses, and UCSD Type 9 Buses, summarized in Table 8.1) are shown in Figure 8.31 – 8.35.

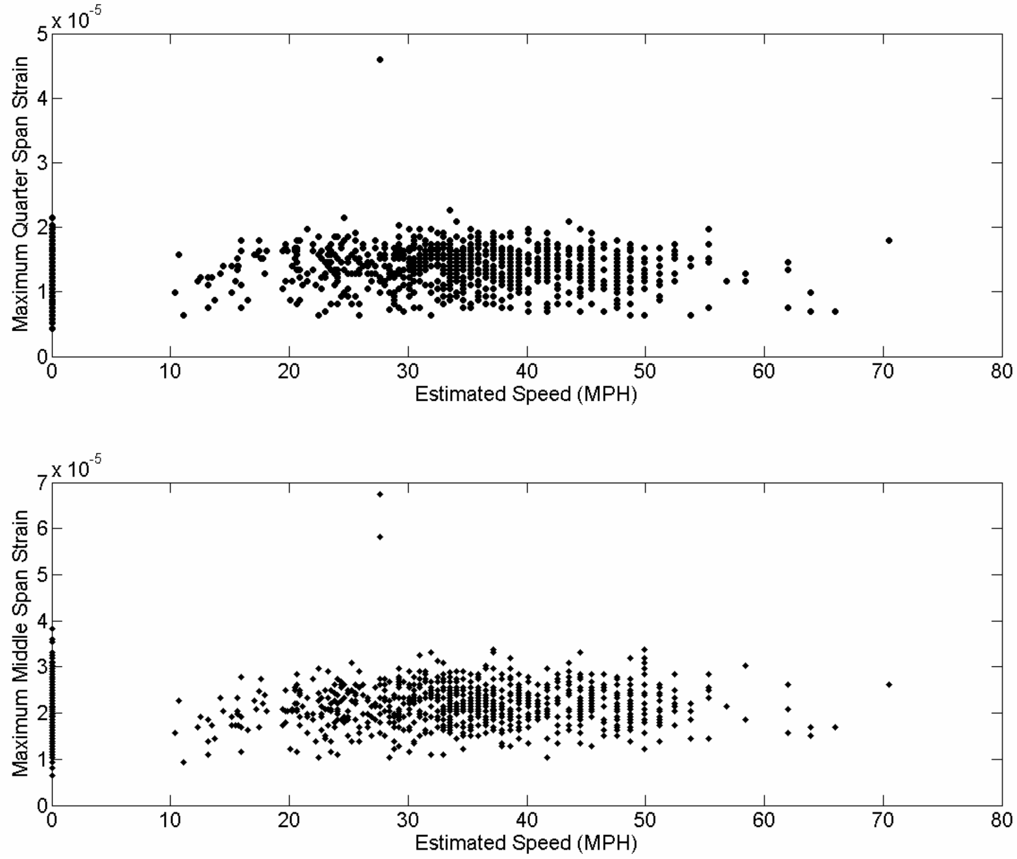


Figure 8.29: Peak Quarter (Top) and Middle (Bottom) Span Strains versus Calculated Speed for Passenger Car Data

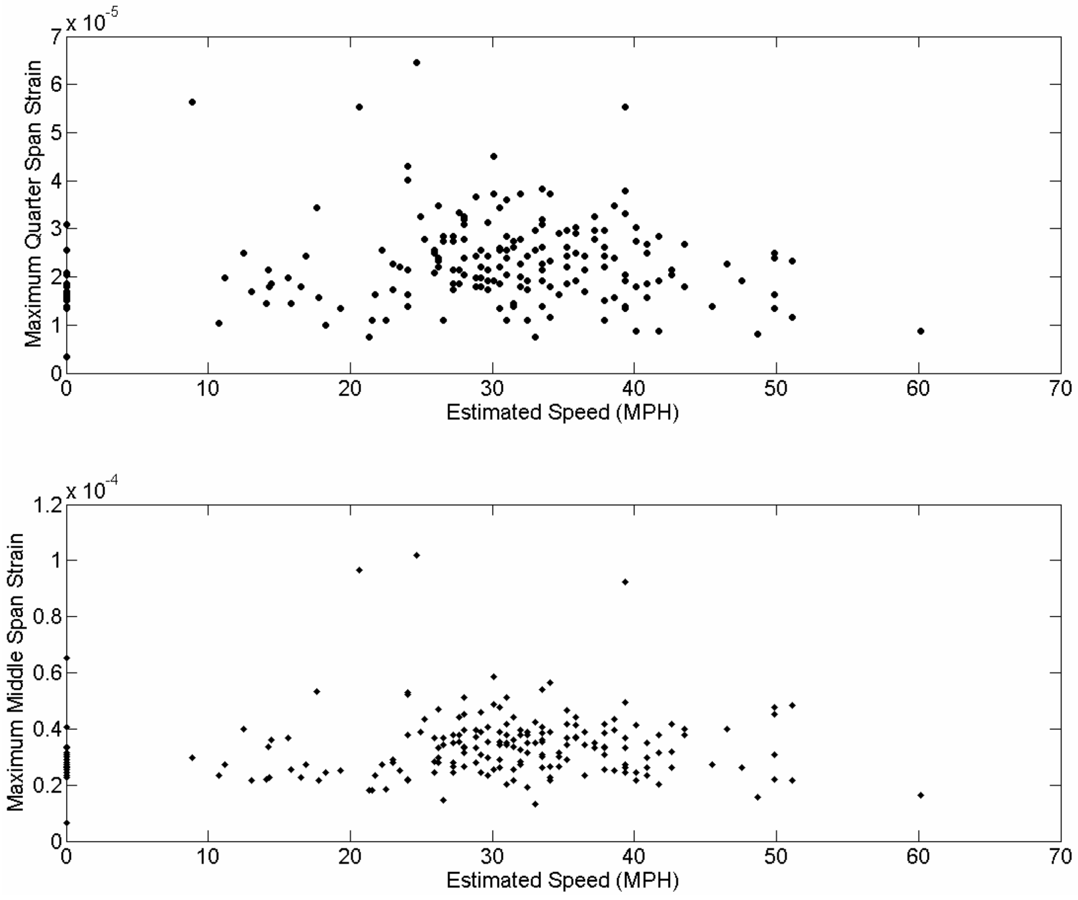


Figure 8.30: Peak Quarter (Top) and Middle (Bottom) Span Strains versus Calculated Speed for Van Data

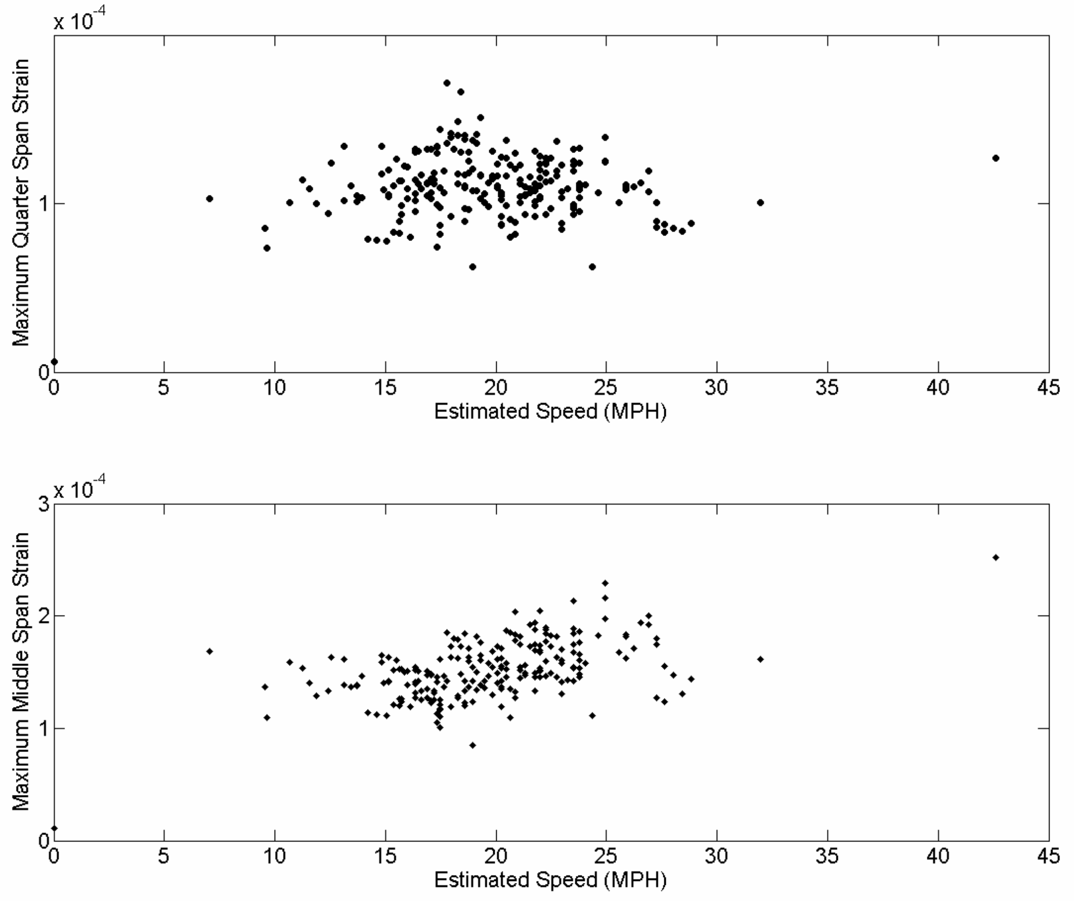


Figure 8.31: Peak Quarter (Top) and Middle (Bottom) Span Strains versus Calculated Speed for UCSD Type 5 Bus Data

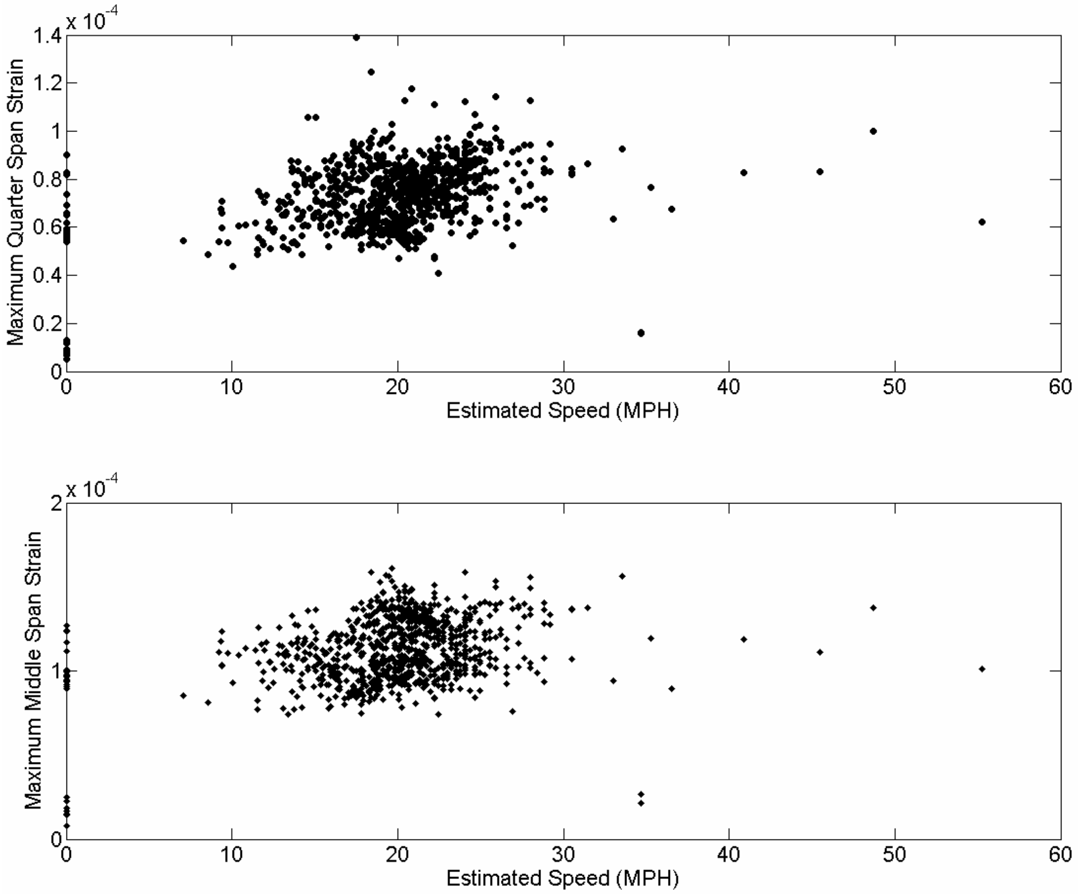


Figure 8.32: Peak Quarter (Top) and Middle (Bottom) Span Strains versus Calculated Speed for UCSD Type 8 Bus Data

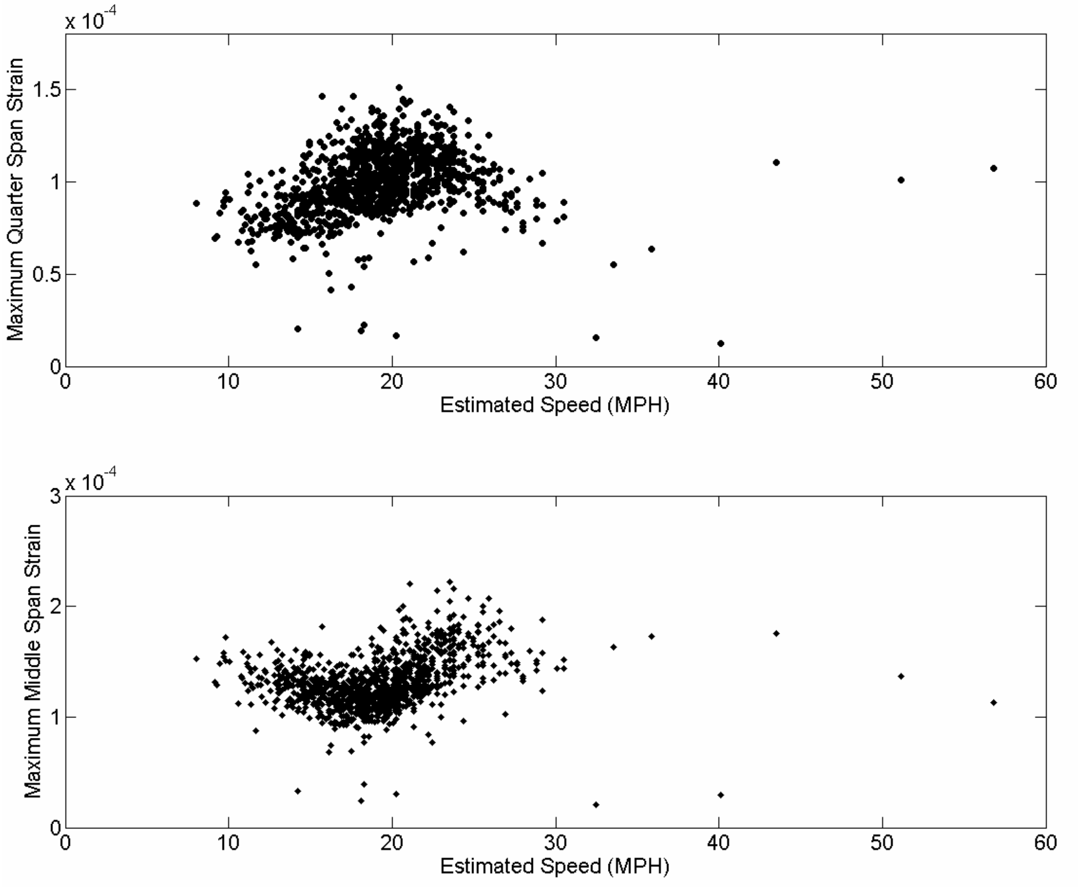


Figure 8.33: Peak Quarter (Top) and Middle (Bottom) Span Strains versus Calculated Speed for UCSD Type 9 Bus Data

A large two-axle vehicle was simulated using the one-dimensional numerical model described in Chapter 5. The peak strains at quarter and middle span versus the speeds used in the simulations are shown in Figure 8.36. In addition, the static loads for the simulated vehicle are indicated at both locations by the dashed lines in these figures. In comparing the results from the recorded and simulated data, it can be seen there is a variation in the peak response versus speed that is present in the simulation data but does not appear in the measured data. In the simulation data, if the DAF is defined as the peak dynamic strain divided by the static value, dynamic amplification factors ranging from 0.955 to 1.18 for middle span strains and 0.935 to 1.20 for quarter span are calculated (ranges which are consistent with a moving load applied to a simply supported beam [Chopra, 2000]). There are two primary explanations for the difficulties in the measured data:

1. The assumption of nearly constant axle weights is clearly not valid, particularly if only a maximum 20% change in strain (as indicated by the finite element data) is expected in the dynamic response. This needs to be addressed in future work through a series of calibration tests in which vehicles of known weight are run across the bridge decks at various speeds while the strains are measured. This data should then be used to form a training set for a neural network to estimate the axle weights of passing vehicles (as was done on the simulation data in Section 7.3.2).
2. While in the numerical model simulating vehicles traveling at high speeds was easily accomplished, this is not the case with the measured data, particularly as the bridge decks are installed a few hundred feet from the campus police station. Consequently most of the measured data is from vehicles moving near or below the 25 MPH speed limit.

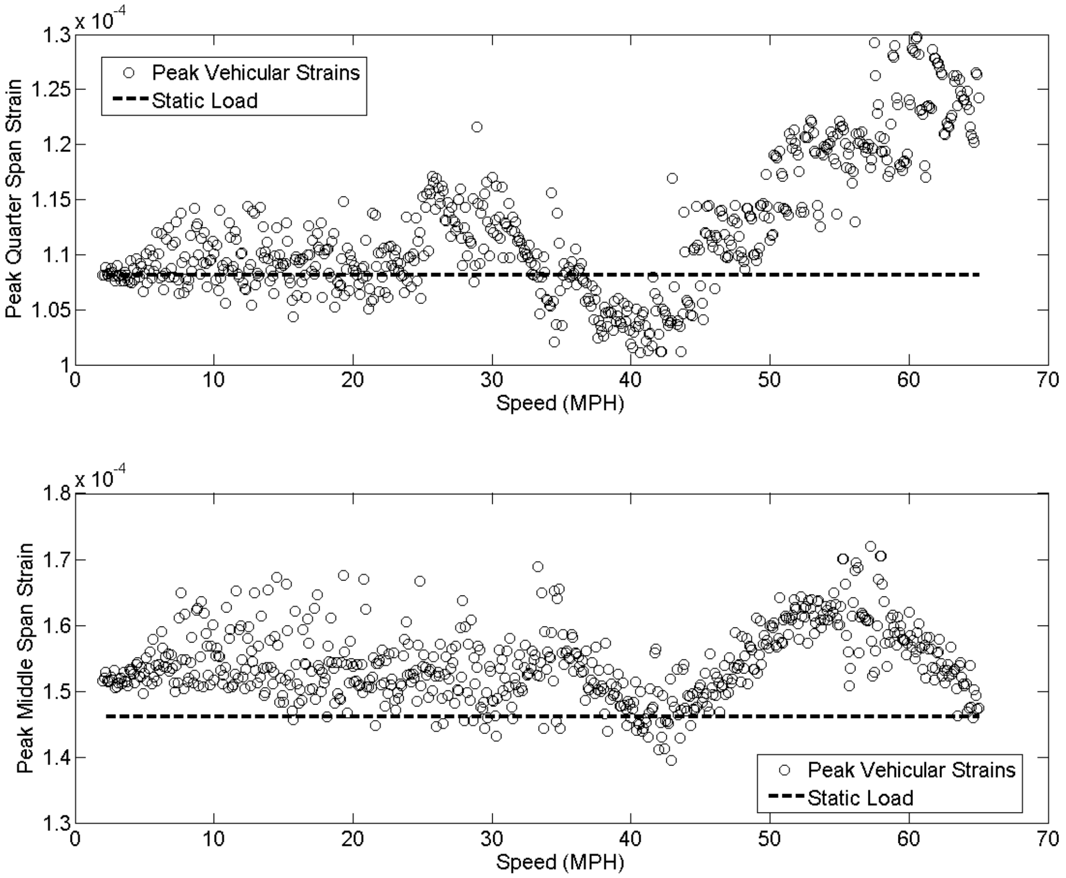


Figure 8.34: Scatter Plot of Speed versus Peak Middle and Quarter Span Strains from Finite Element Model

8.4 Utilizing a Predictive Neural Network to Monitor for Changes in System Response

In Section 7.8 of this paper the feasibility of a predictive neural network was demonstrated for identifying changes in the bridge deck system. This was done by treating the quarter span strain time histories as inputs to a neural network which outputted predicted Principal Components (PC's) for the midspan strains. By taking the inverse PCA transformation of these PC's, the errors between the predicted strain time histories and those output from the finite element model were computed. It was found these errors could be used to detect reductions in stiffness (simulated damage) as low as 2%.

A similar strategy is now employed on a set of strain time histories recorded on the composite bridge decks corresponding to typical 2-axle traffic. The strains recorded by gage 8 were used as inputs for a series of neural networks to predict strains from gages 6, 9, and 11. Training was done on data recorded during December 2003. These neural networks were then tested on the data recorded during 2004. In this exercise (Fig. 8.37), two sets of time histories were compared to the recorded data. The first is the time history reconstructed from the PC's using the inverse PCA transformation learned on the training data. Errors at this level are indicative of new features now found in the test data that were not present during training. The second comparison is between the recorded data and the data reconstructed from the Principal Components output by each of the neural networks using the same sets of inverse PCA transformations. Assuming the first set of errors remains small, then errors here result from changes in the mapping learned by the neural networks during training.

For the example of the simulation data analyzed in Section 7.8, when damage was simulated by reducing the value of Young's Modulus for select elements in the finite element mesh, the amplitude of the strains increased, but the overall shape of the time histories remained unchanged. Consequently, the error in the PCA reconstruction data remained constant while the error from the reconstructed PC’s output from the neural network increased. The same logic is now applied to the measured data.

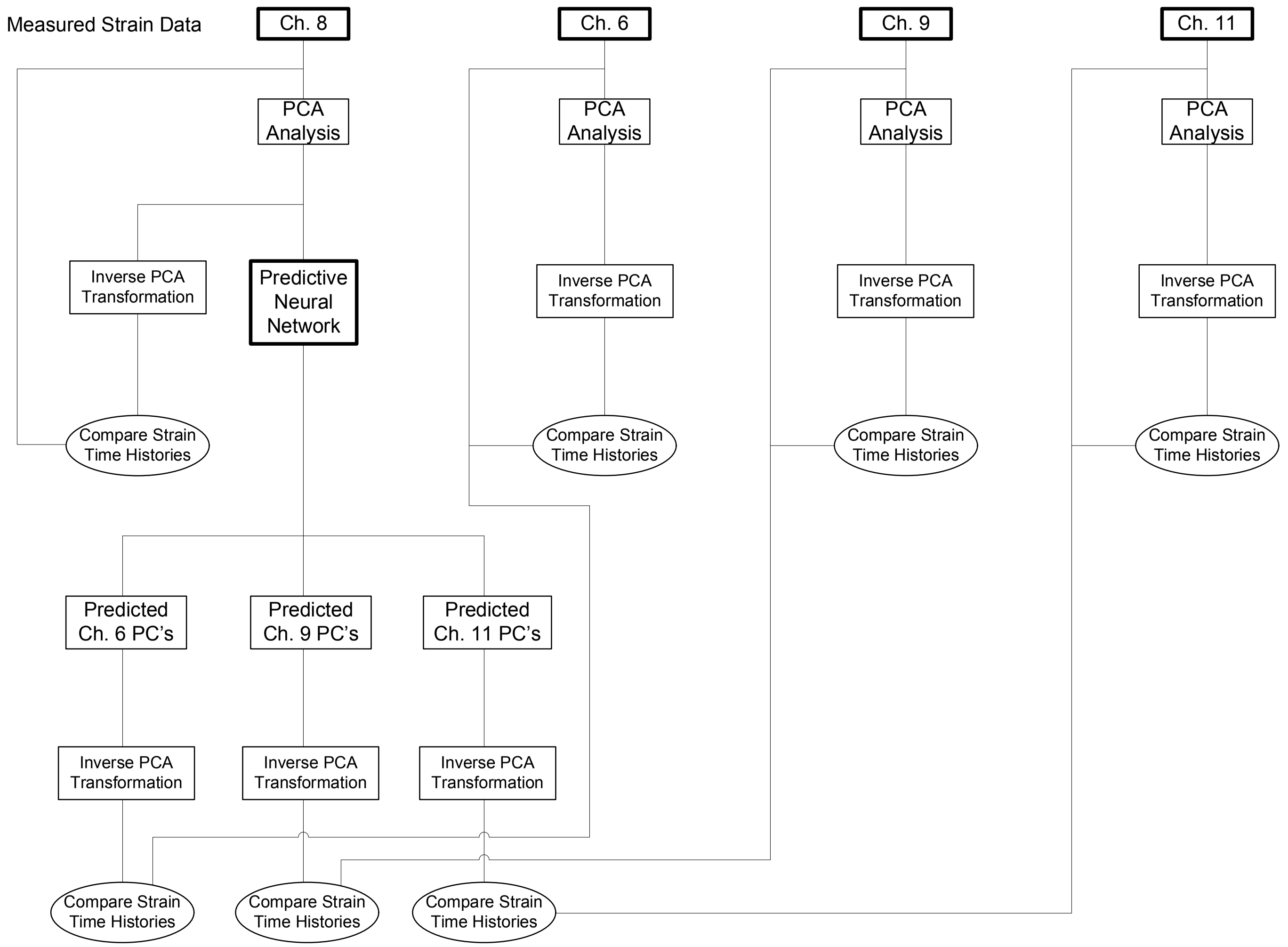

Measured Strain Data
Ch. 8
Ch. 6
Ch. 9
Ch. 11
PCA Analysis
PCA Analysis
PCA Analysis
PCA Analysis
Inverse PCA Transformation
Predictive Neural Network
Inverse PCA Transformation
Inverse PCA Transformation
Inverse PCA Transformation
Compare Strain Time Histories
Compare Strain Time Histories
Compare Strain Time Histories
Compare Strain Time Histories
Predicted Ch. 6 PC's
Predicted Ch. 9 PC's
Predicted Ch. 11 PC's
Inverse PCA Transformation
Inverse PCA Transformation
Inverse PCA Transformation
Compare Strain Time Histories
Compare Strain Time Histories
Compare Strain Time Histories

Figure 8.35: Flowchart for Utilizing Principal Component Analysis and Neural Networks To Monitor for Changes in the Bridge Deck Response

8.4.1 Data Cleansing

Between December 1 and December 31, 2003, a total of 3,621 vehicles were stored in which the peak strains recorded by strain gage #11 were between $8x10^{-5}$ and $1.3x10^{-4}$. In an effort to optimize the network performance, a cleansed data subset was created by examining each of the strain time histories looking for multiple vehicles crossing within the same 10-second time history (Fig's. 8.38 and 8.39), vehicles moving from East to West (Fig's. 8.40 and 8.41), and for 3 or more axle vehicles (Fig's. 8.42 and 8.43). If such an event was noted, the record was discarded and not included into the sorted data. Of the 3,621 total traffic patterns recorded, 2,629 were deemed useable for network training and validation. 1000 of these patterns were randomly selected and used for training and validation (700 and 300, respectively) while the remaining records (1,629) served as a test set to assess the trained network's performance.

This process was done using a peak detection program written in Matlab (see section 6.1 for more details). As only time histories corresponding to a single 2-axle vehicle were used if two or more peaks (axles) were found in a time history, the record was discarded for reasons one and three above. To detect vehicles crossing the bridge decks in the wrong direction, the instants in time when the front axle crosses gages 8 (quarter span) and 11 (middle span) were found. For vehicle moving in the desired direction (West to East), the arrival time at gages 8 will be less than gage 11. Records for which this was not case were discarded. Once the data was sorted in Matlab, it was hand sorted one time to look for unusable records which were missed. This last manual sorting step was done to ensure the training data was as clean as possible.

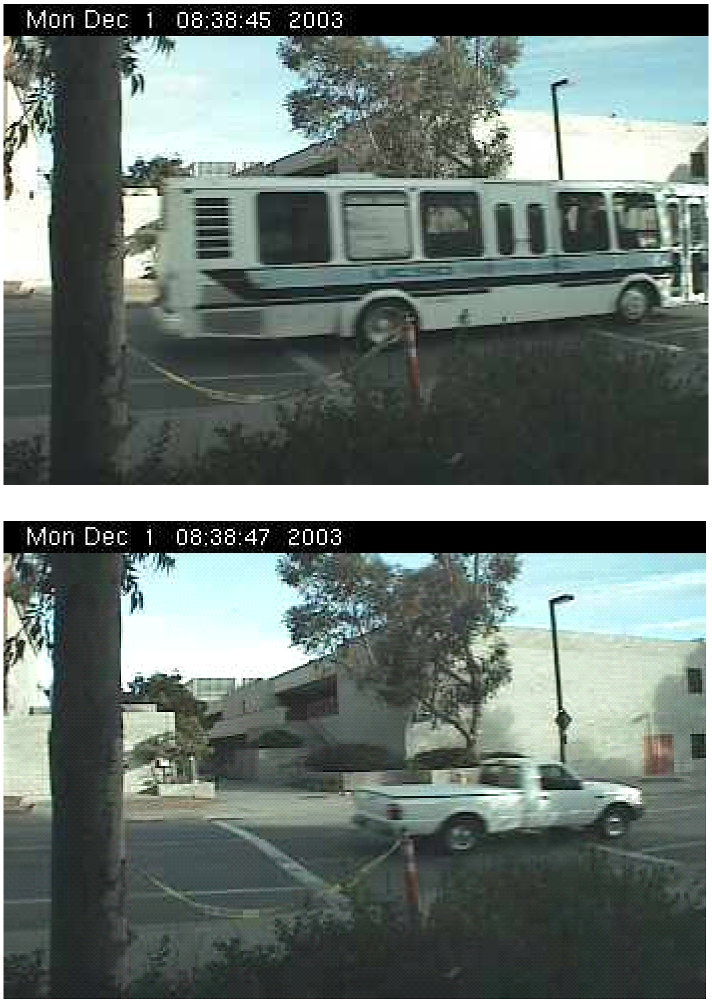


Figure 8.36: UCSD Campus Bus and Pick-up Truck Recorded Crossing Bridge-Decks

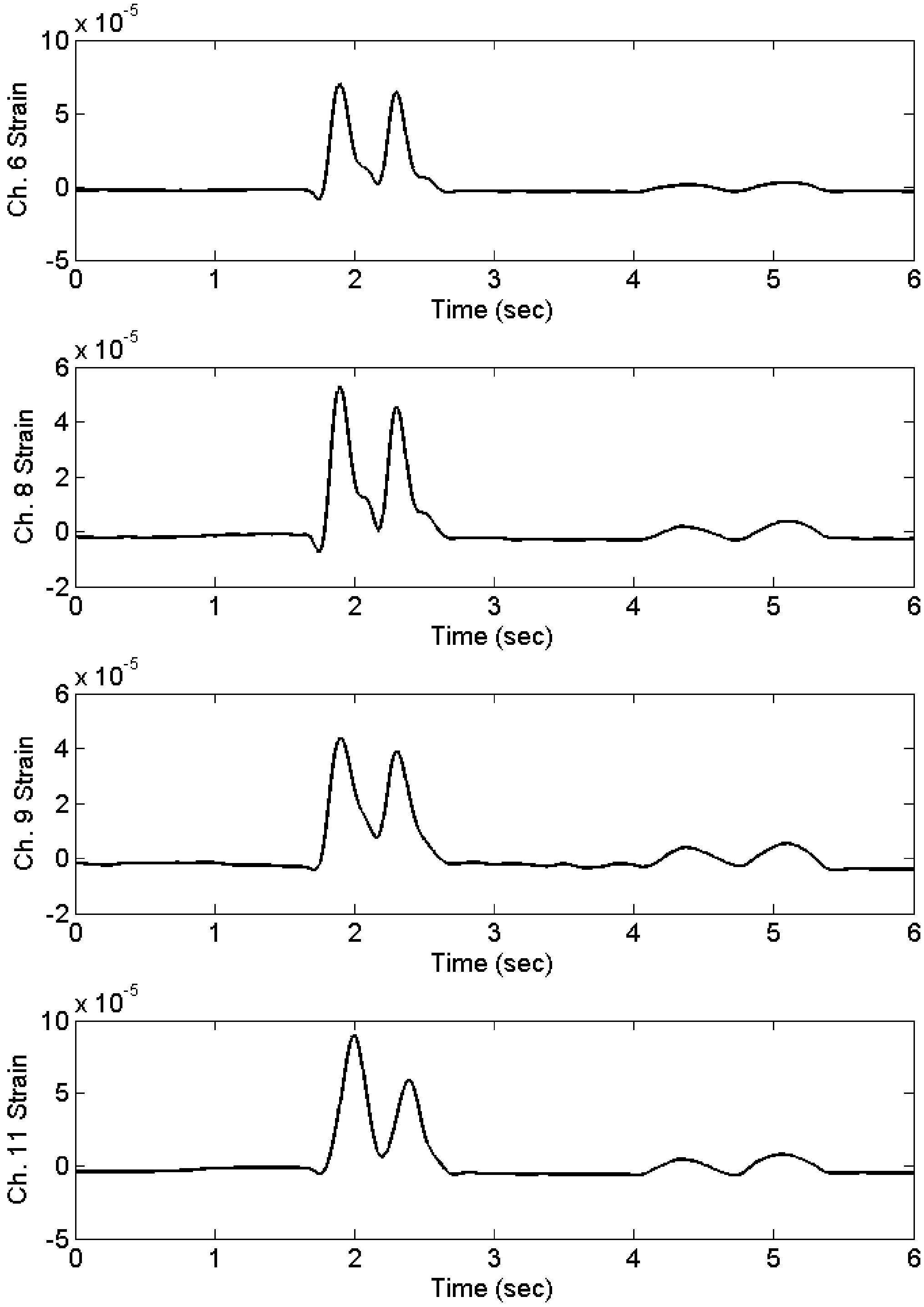


Figure 8.37: Channel 6, 8, 9 , and 11 (top to bottom) Strain Time Histories for UCSD Campus Bus and Pick-up Truck Recorded Crossing Bridge-Decks.

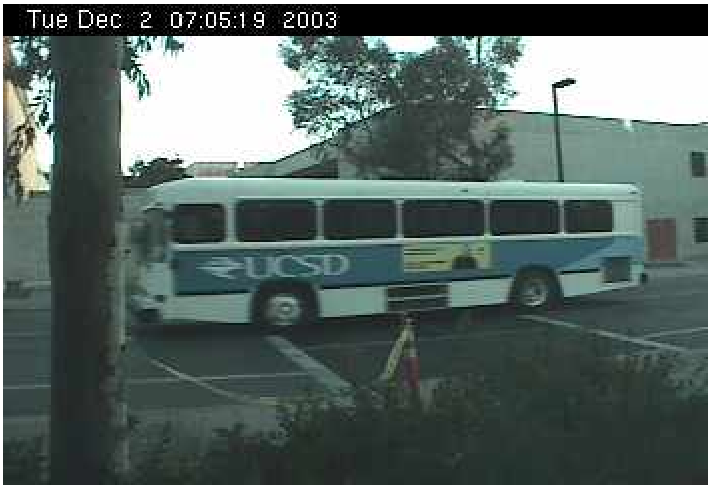


Figure 8.38: UCSD Campus Bus Crossing Bridge Decks from East to West.

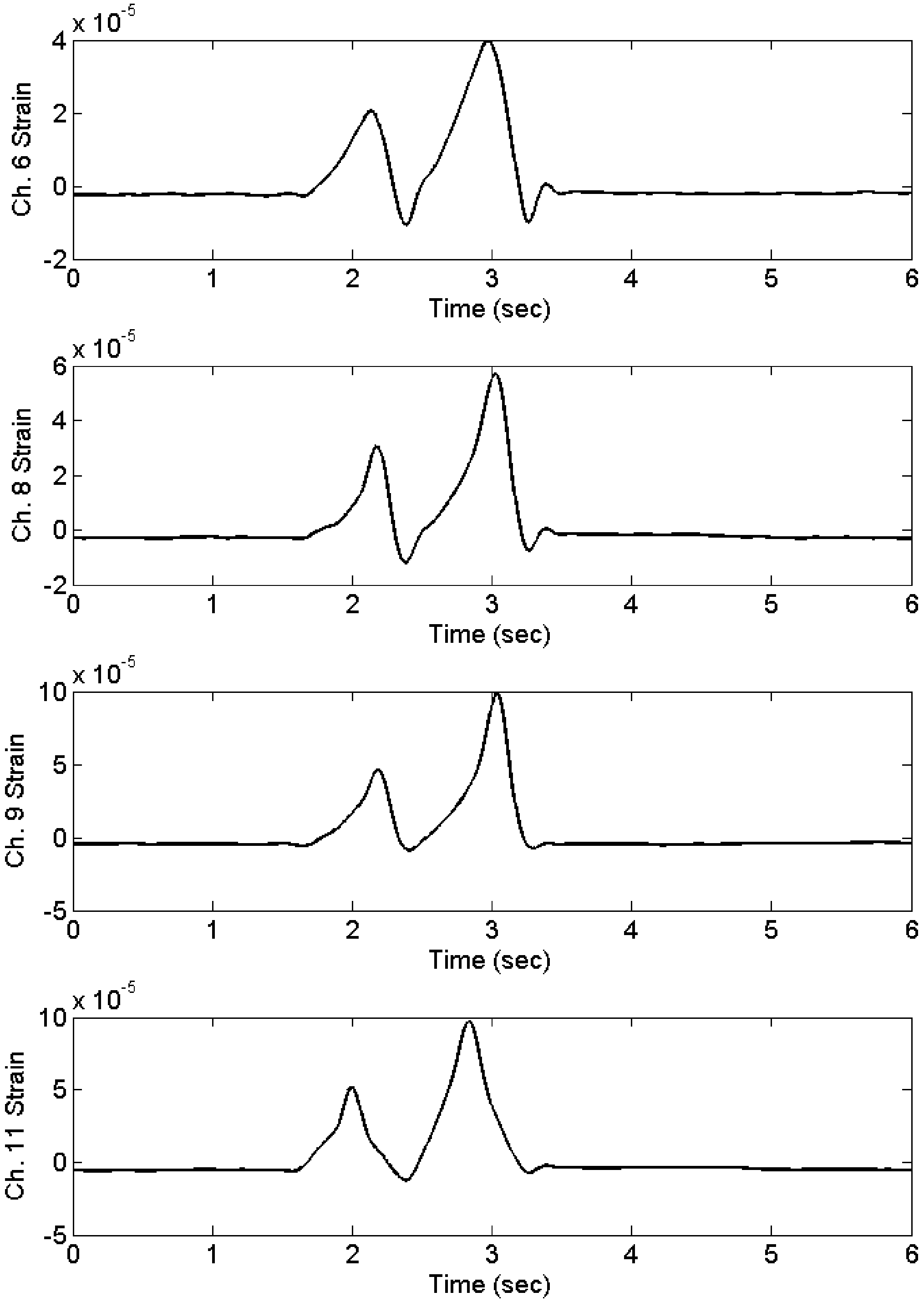


Figure 8.39: Channel 6, 8, 9 , and 11 (top to bottom) Strain Time Histories for Vehicle Crossing Bridge Decks from East to West

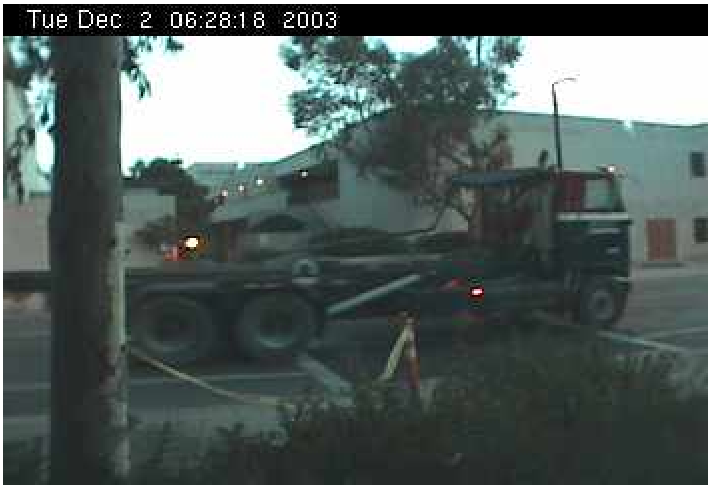


Figure 8.40: 3-Axle Truck Crossing Bridge Decks

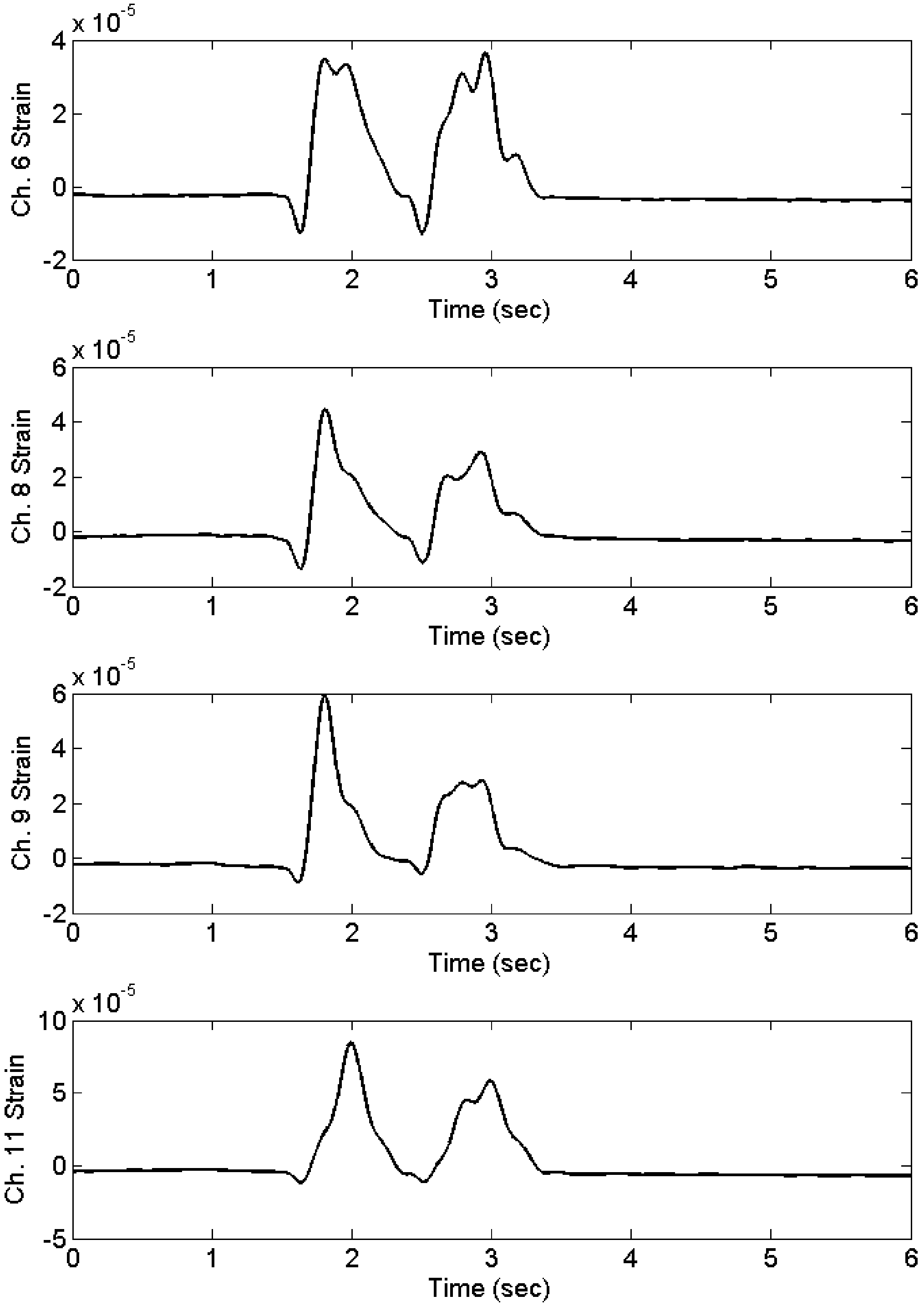


Figure 8.41: Channel 6, 8, 9 , and 11 (top to bottom) Strain Time Histories for 3-Axle Vehicle

8.4.2 Feature Reduction using Principal Component Analysis

Prior to applying the recorded strain time histories to the neural networks, Principal Component Analysis was employed for feature reduction. This was done following the procedures detailed in Sections 7.8 and 8.1.1. For each time history, the instants in time when the front axle moves onto and the rear axle moves off of the unsupported portion of the bridge decks (Fig's. 8.4 and 8.5) were found. In the typical strain time histories shown in Figures 8.44 – 8.47, these are indicated with the filled squares in the upper plots. These portions of the strain time histories were cut out and resampled such that each was composed of exactly 200 points (middle two plots in Fig's. 8.44 – 8.47). Finally, PCA was applied to these resampled time histories. Only those PC's which preserved 99.0% of the original variance in the data were used. For gages 6 and 9, eight Principal Components were used while for gages 8 and 11 nine were required. The normalized PC's for the sample records are shown in the bottom plots in Fig's. 8.44 – 8.47. Figure 8.48 is a plot of Eigenvalue numbers versus the eignevalues for the training data for the four gages. In Figure 8.49, the percent variance is plotted against the number of employed PC's. It was from these plots the number of Principal Components used for training (and consequently testing) were selected. Finally, the first five Principal Components are plotted for each of the four strain gages (Fig's. 8.50 – 8.53).

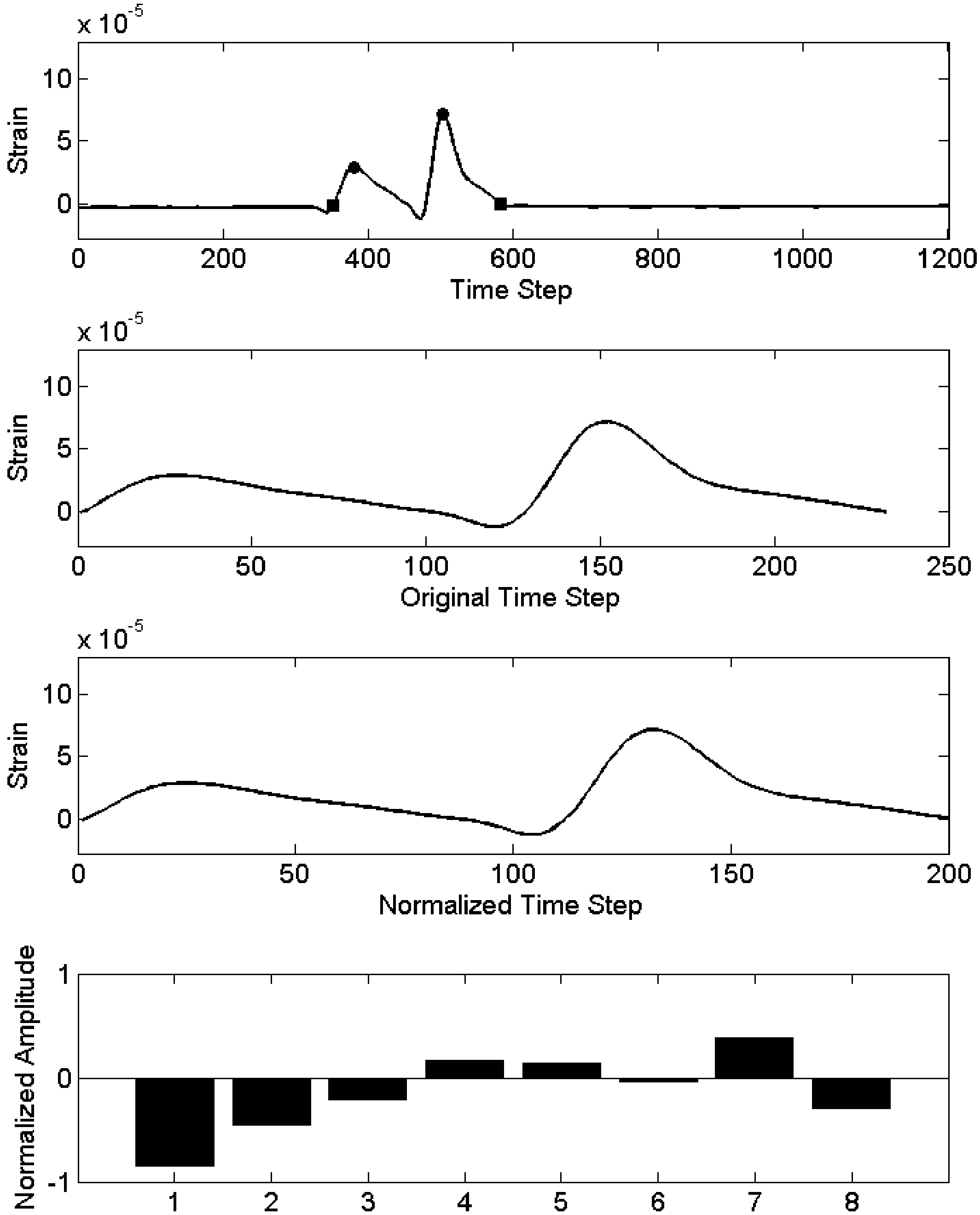


Figure 8.42: Channel 6 Strain Time Histories and Principal Components for Typical Training Pattern

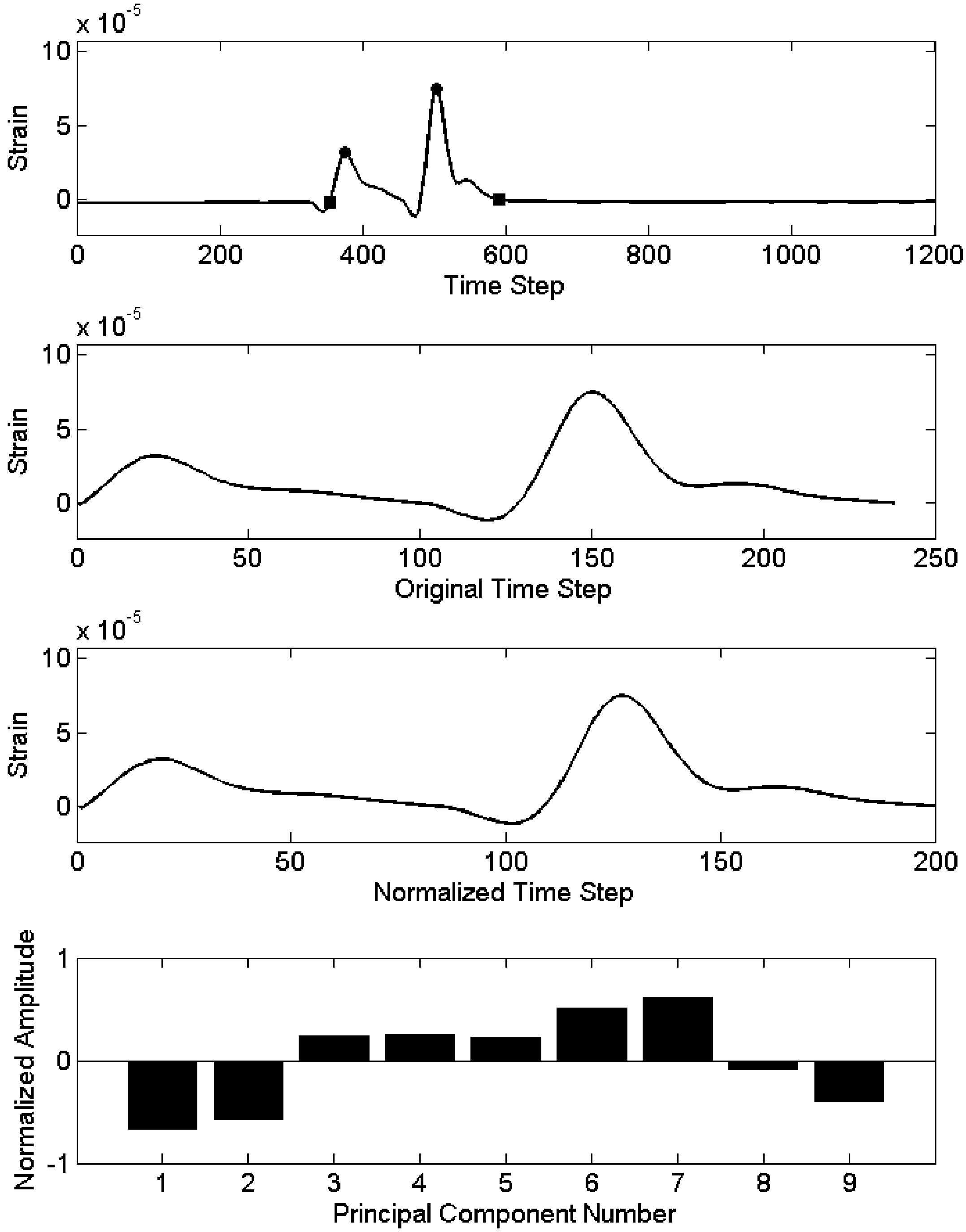


Figure 8.43: Channel 8 Strain Time Histories and Principal Components for

Typical Training Pattern

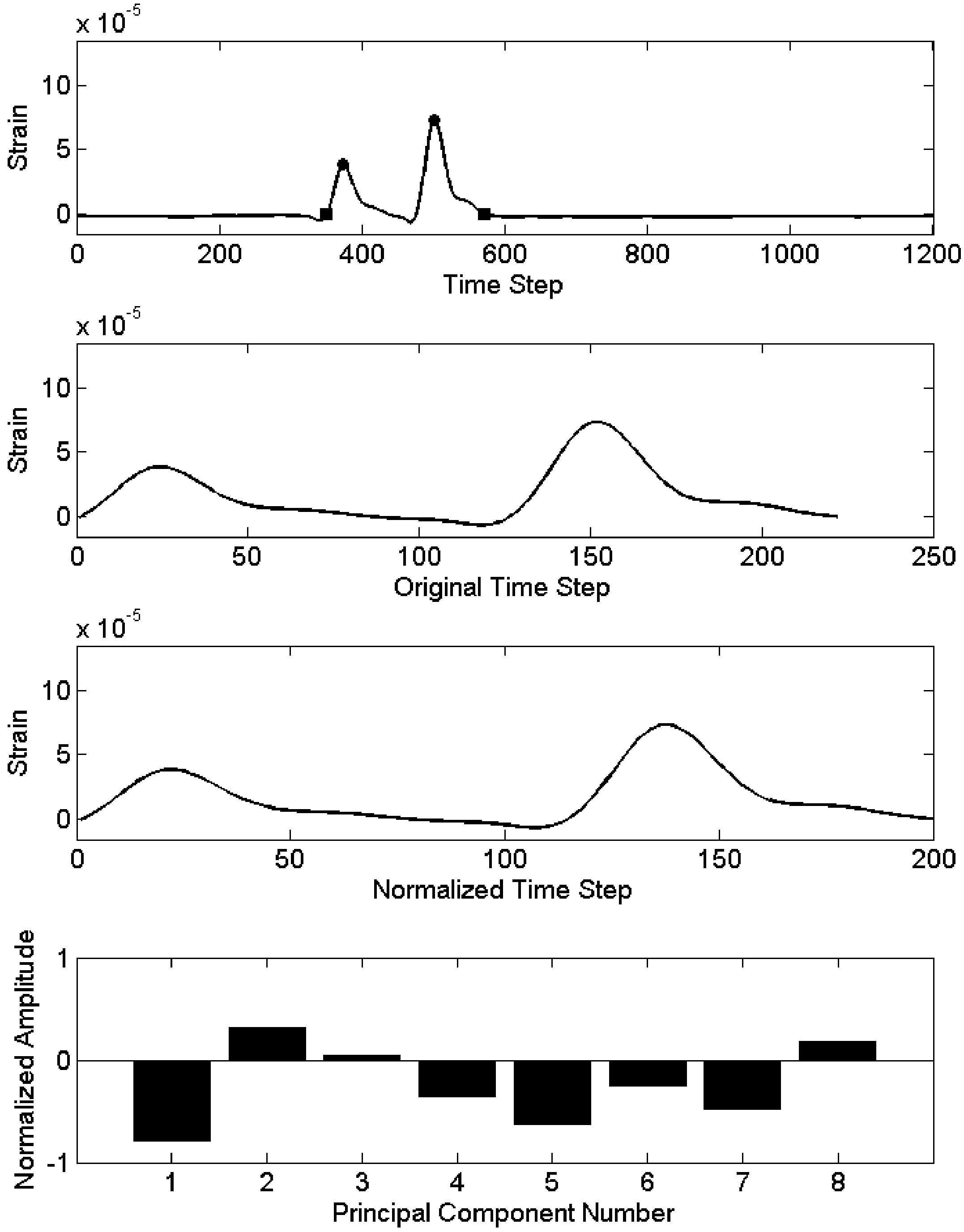


Figure 8.44: Channel 9 Strain Time Histories and Principal Components for Typical Training Pattern

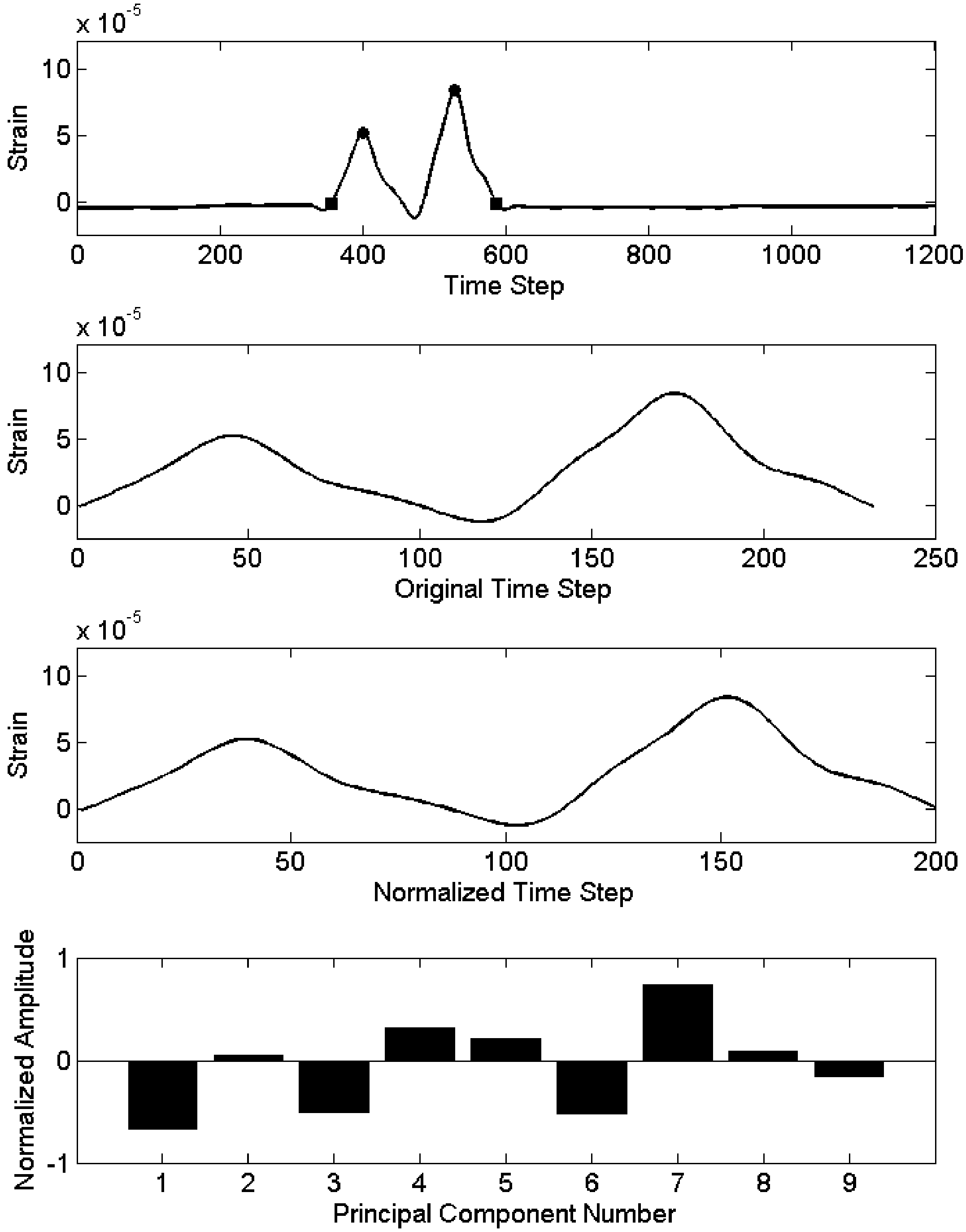


Figure 8.45: Channel 11 Strain Time Histories and Principal Components for Typical Training Pattern

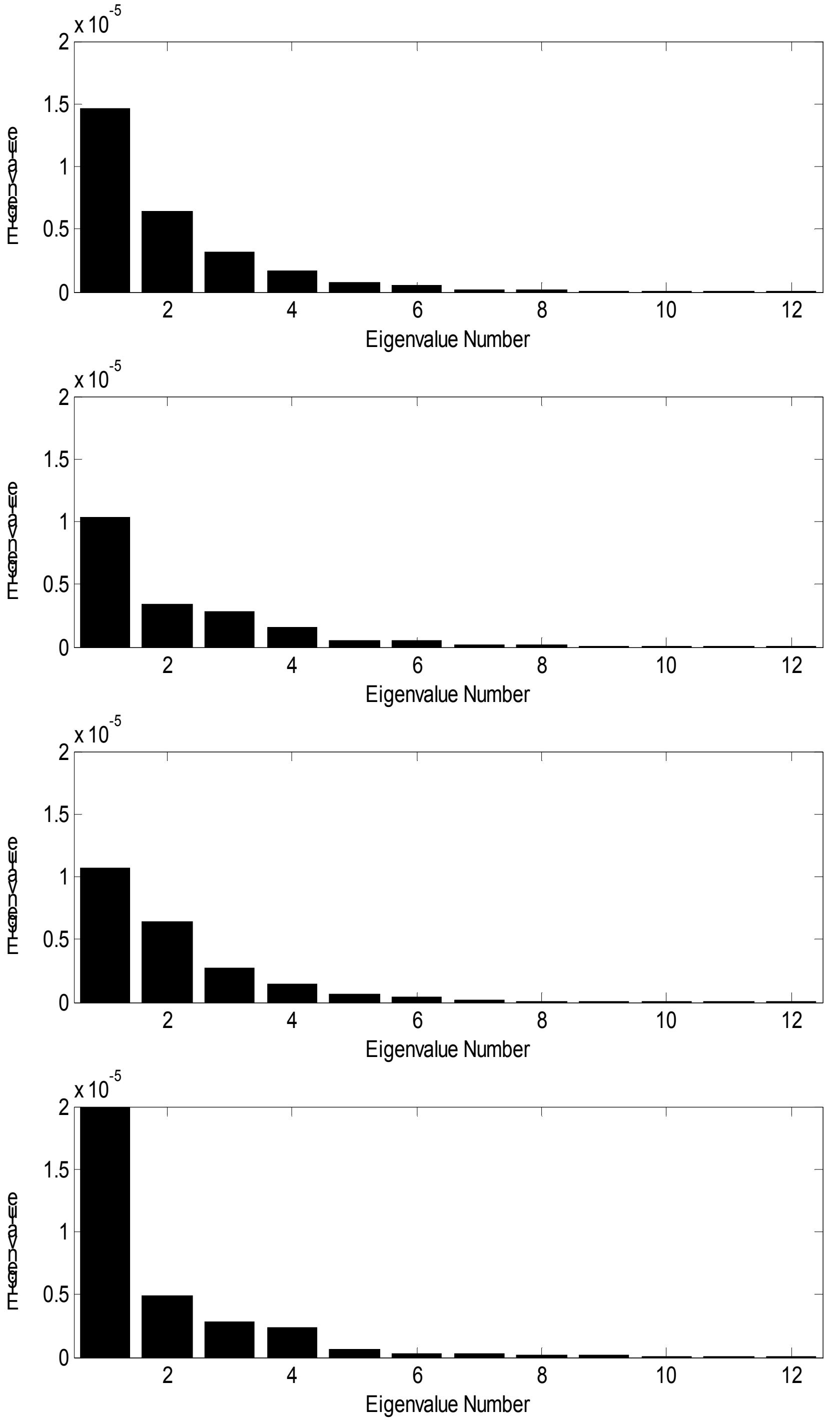


Figure 8.46: Resulting Eigenvalues from PCA of Training Data for Strain Gage 6, 8, 9, and 11 (from Top to Bottom)

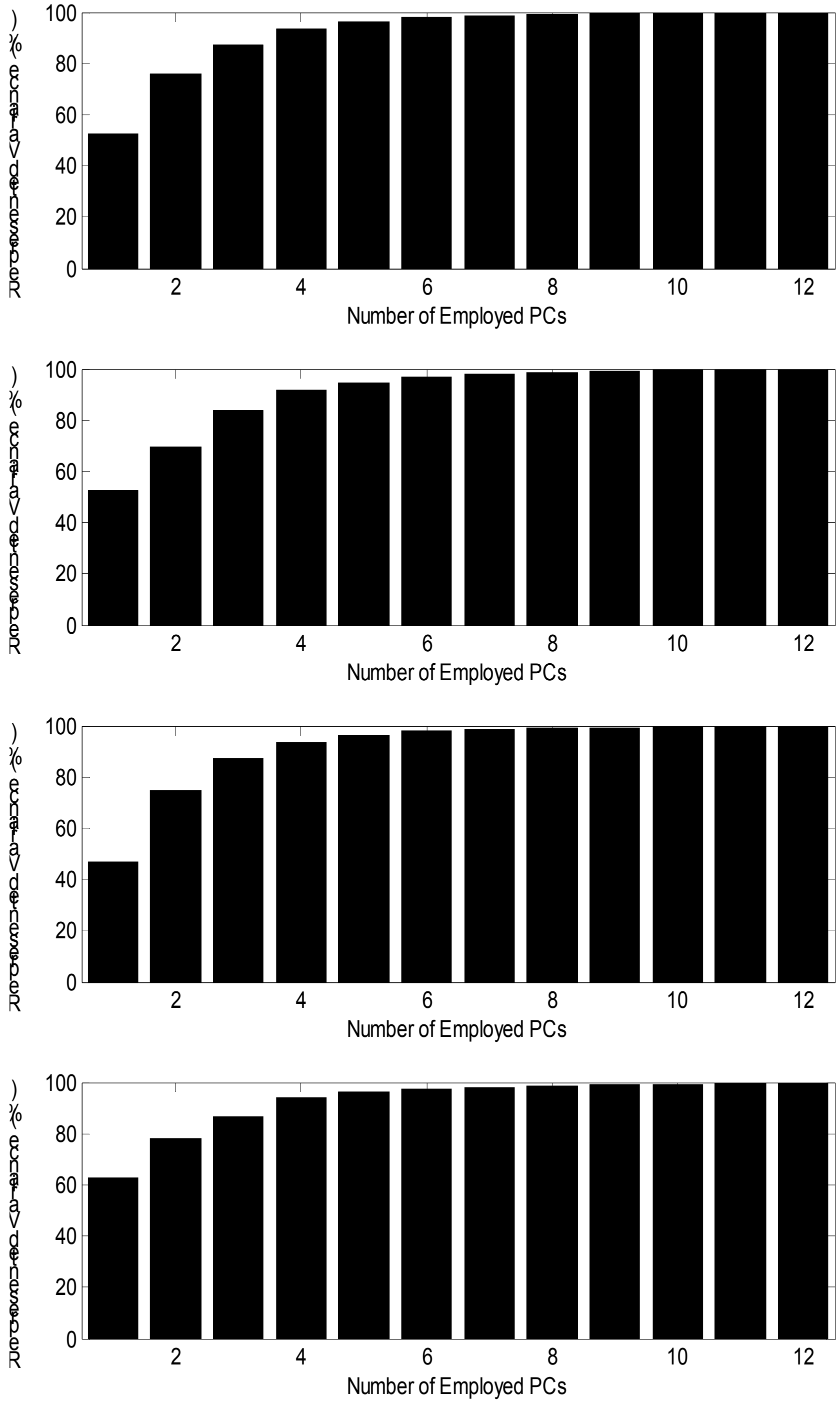


Figure 8.47: Percent Variance versus Number of Employed PC's for Strain Gage 6, 8, 9, and 11 (from Top to Bottom) Training Data

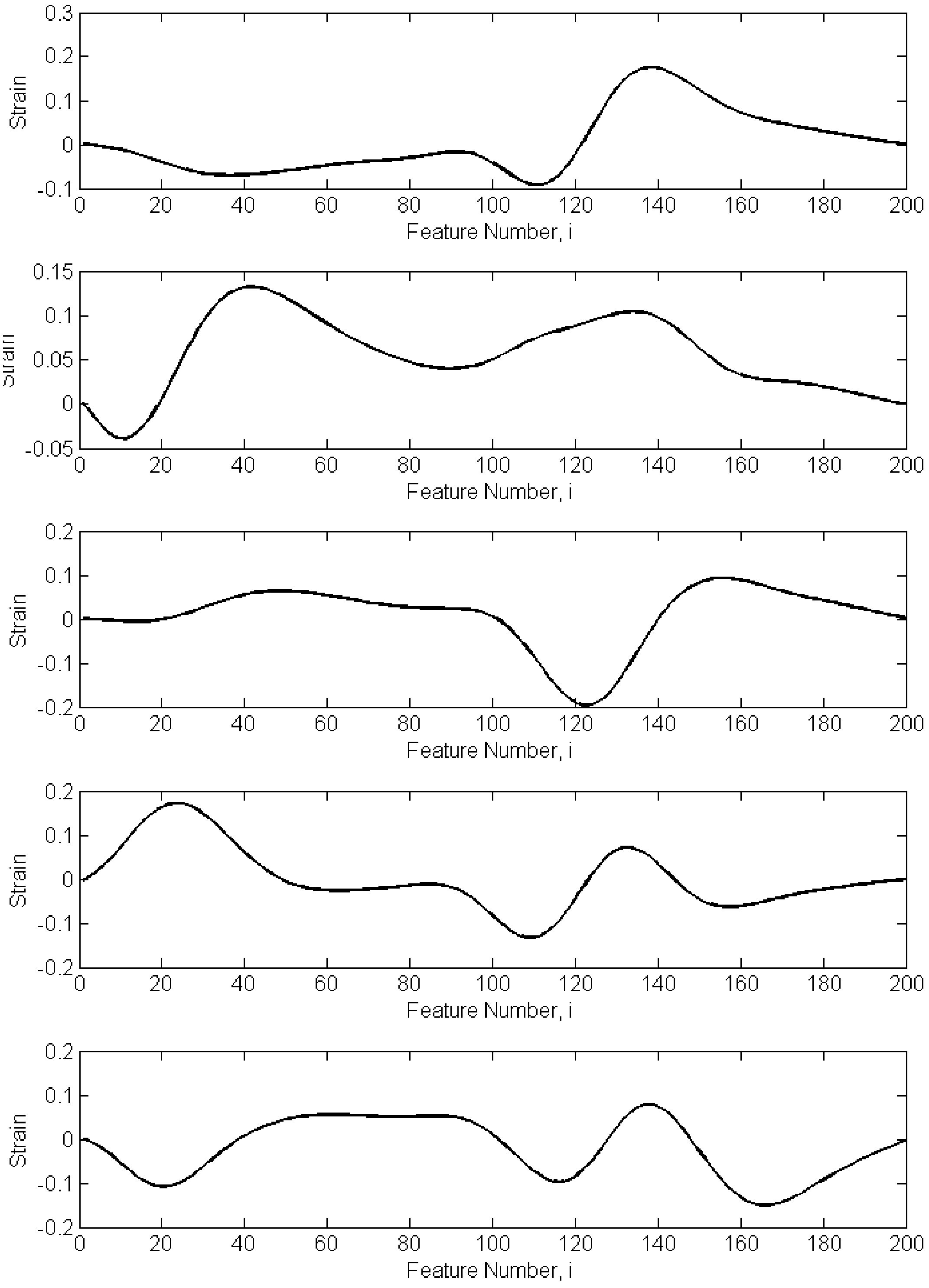


Figure 8.48: The first 5 Principal Components for Strain Gage 6

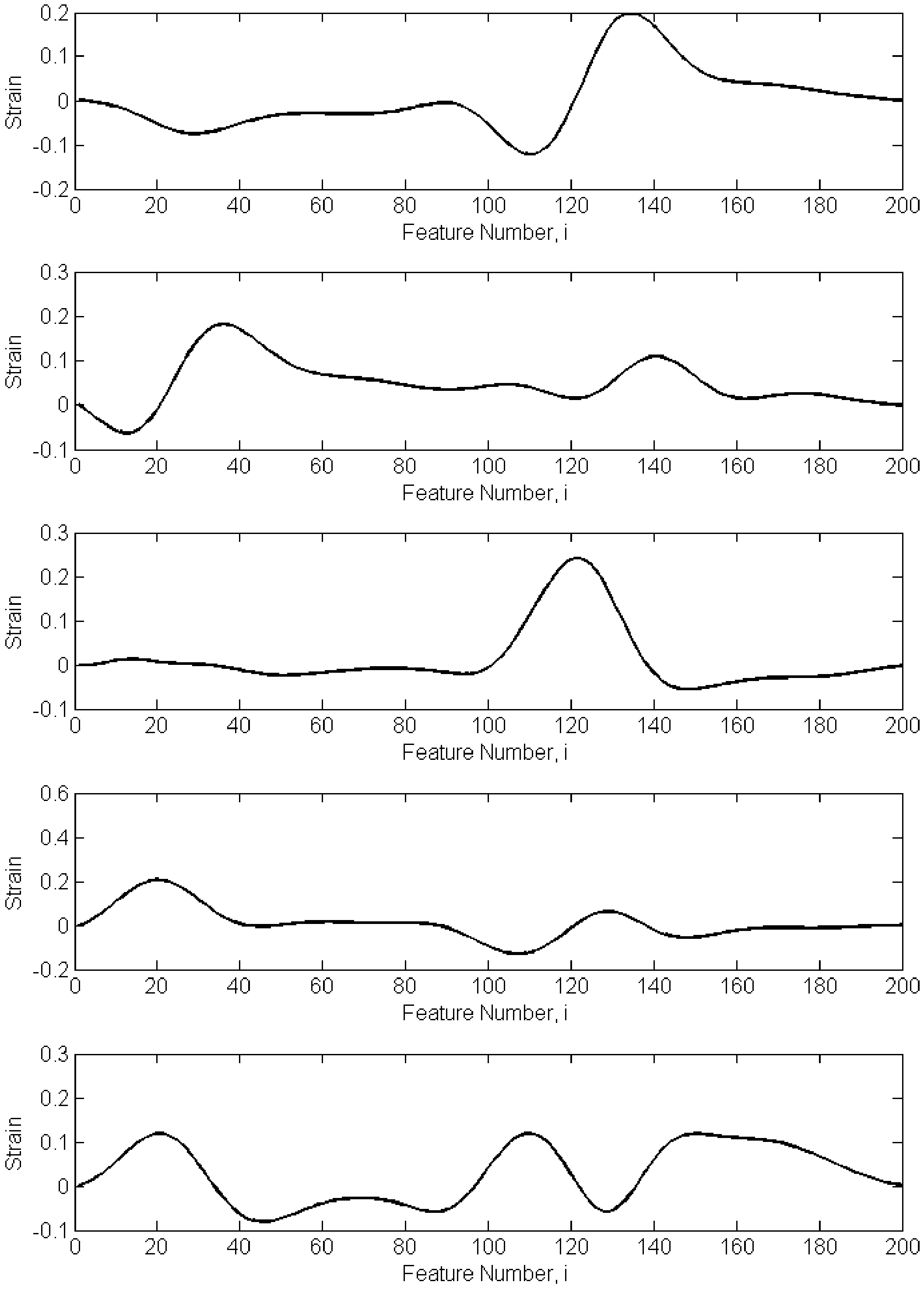


Figure 8.49: The first 5 Principal Components for Strain Gage 8

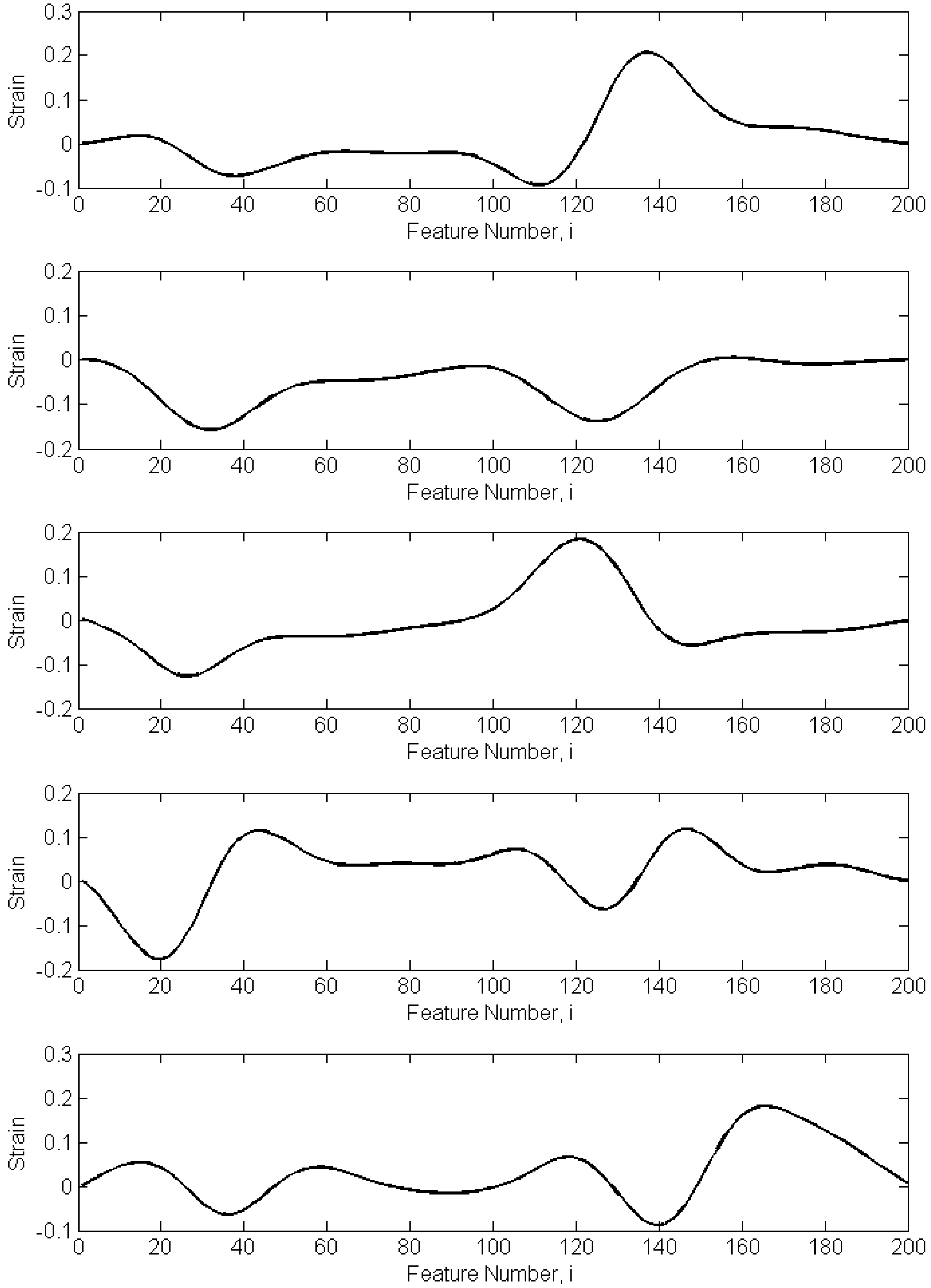


Figure 8.50: The first 5 Principal Components for Strain Gage 9

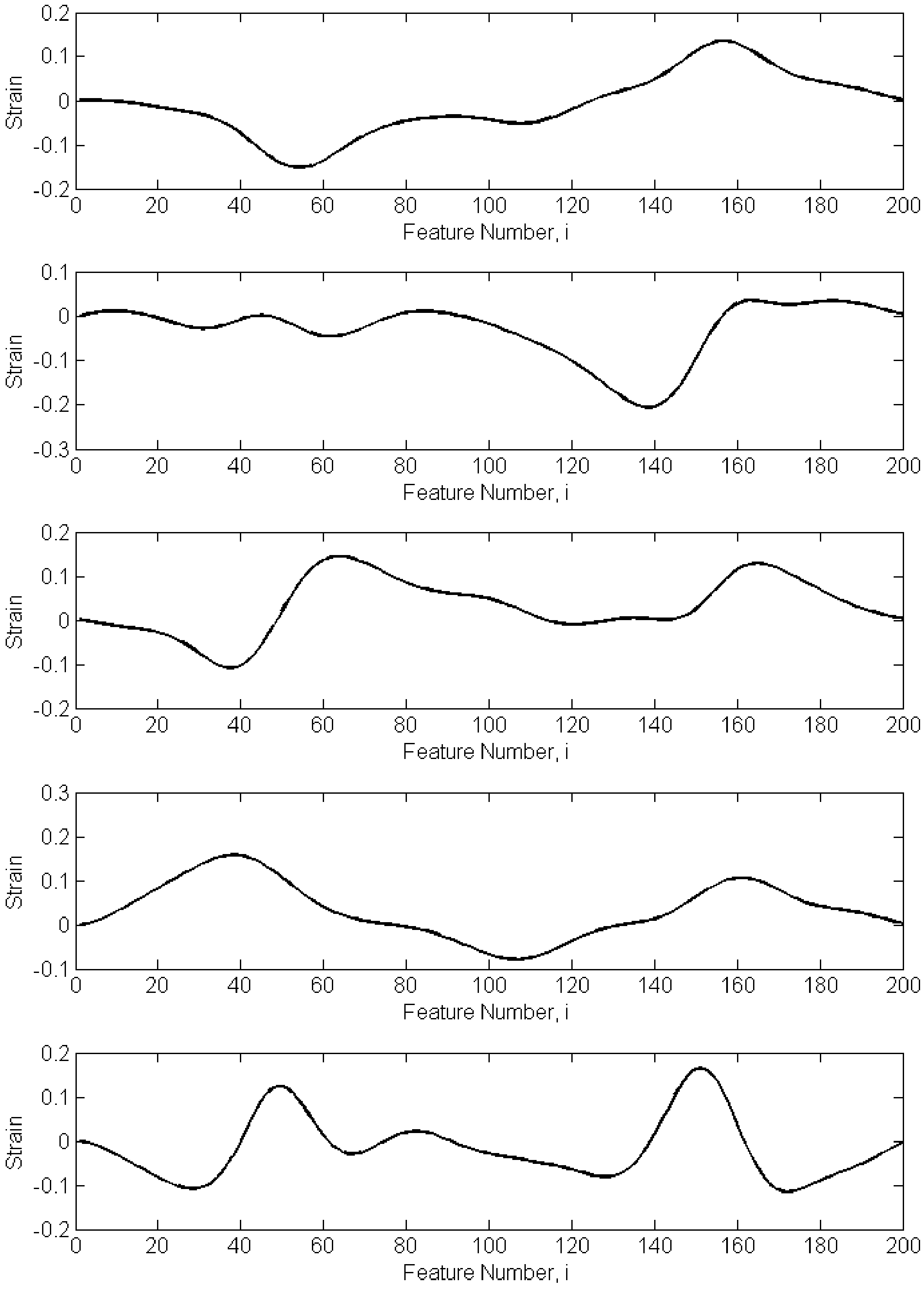


Figure 8.51: The first 5 Principal Components for Strain Gage 11

8.4.3 Strain Reconstruction from Principal Components

Following the flowchart in Fig. 8.37, the first comparison with the recorded strains is made with the strain time histories reconstructed using the inverse PCA transformation. This was done on the training data as well as the testing data from 2003 and 2004. The normalized errors for the training data is plotted in Fig. 8.54 along with the histograms of the errors in Fig. 8.55. Next, the errors and histograms for the test data recorded in 2003 were calculated and are plotted in Fig's. 8.56 and 8.57. These two sets of data form the baseline for which errors from future data is compared to check for variations in the ability of the PCA transformation to represent the data. This was done with the test data from 2004 and is plotted in Fig's. 8.58 - 8.60. The mean errors, standard deviations, and maximum errors are summarized in Tables 8.2 – 8.4.

From inspection of the 2004 errors, there appears to be a deviation in the mean error in each of the 4 channels during the summer months (end of June through August). The errors from the 2004 were averaged using 100 points and are plotted in Fig. 8.61. Within these plots, the increased error during the summer months becomes more pronounced. However, it is unclear whether this is the result of a change in the bridge response or is due to the scarcity of data available during the summer months. As previously mentioned, the campus shuttle buses (which form the bulk of the recorded data) do not operate during this period. From Figure 8.62, it can be seen that during the months of July and August only 4% of the records recorded in the other months were observed.

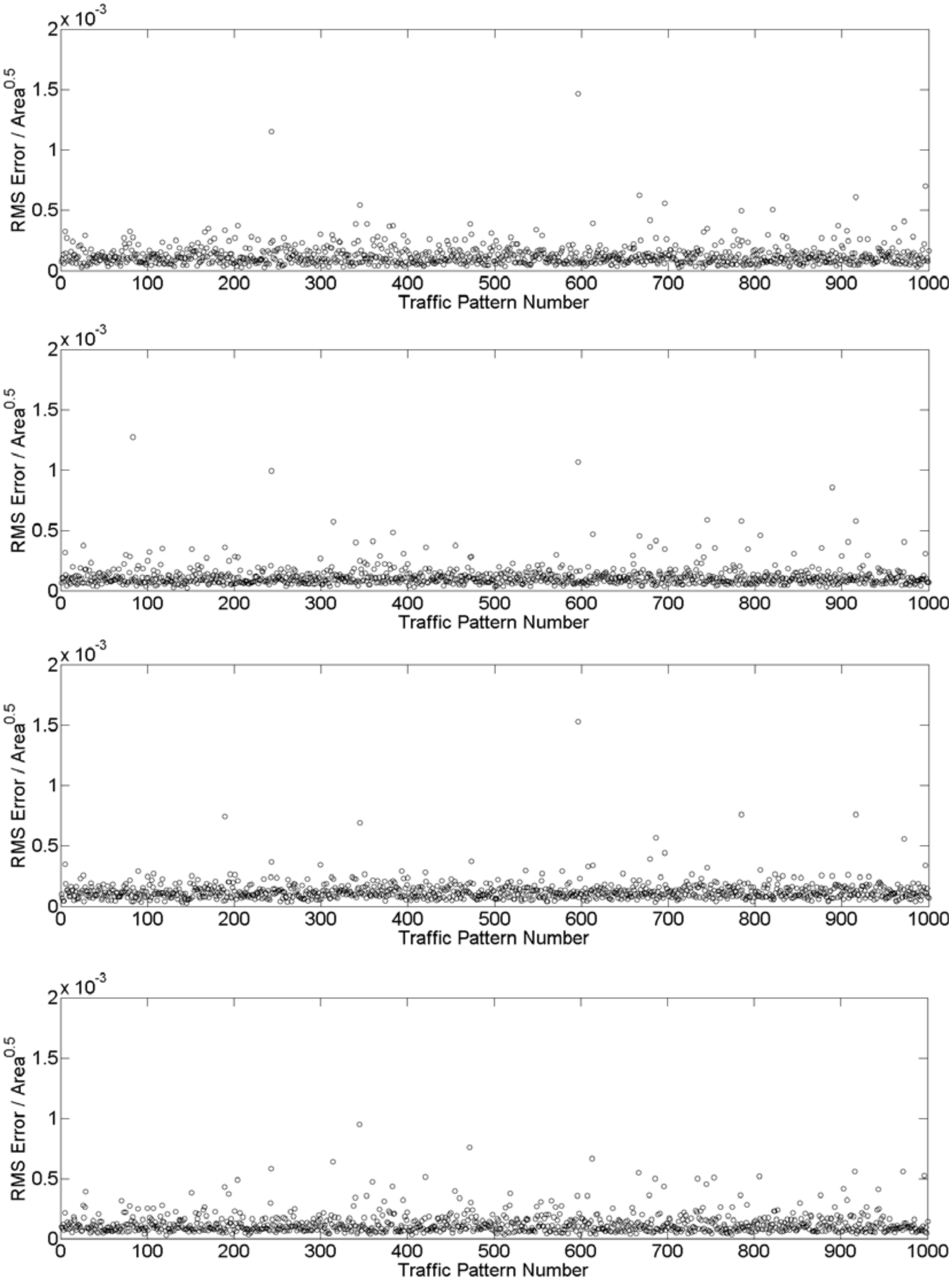


Figure 8.52: Scatter Plot of Normalized Errors for Reconstructed Channel 6, 8, 9, and 11 Training Strains (Top to Bottom)

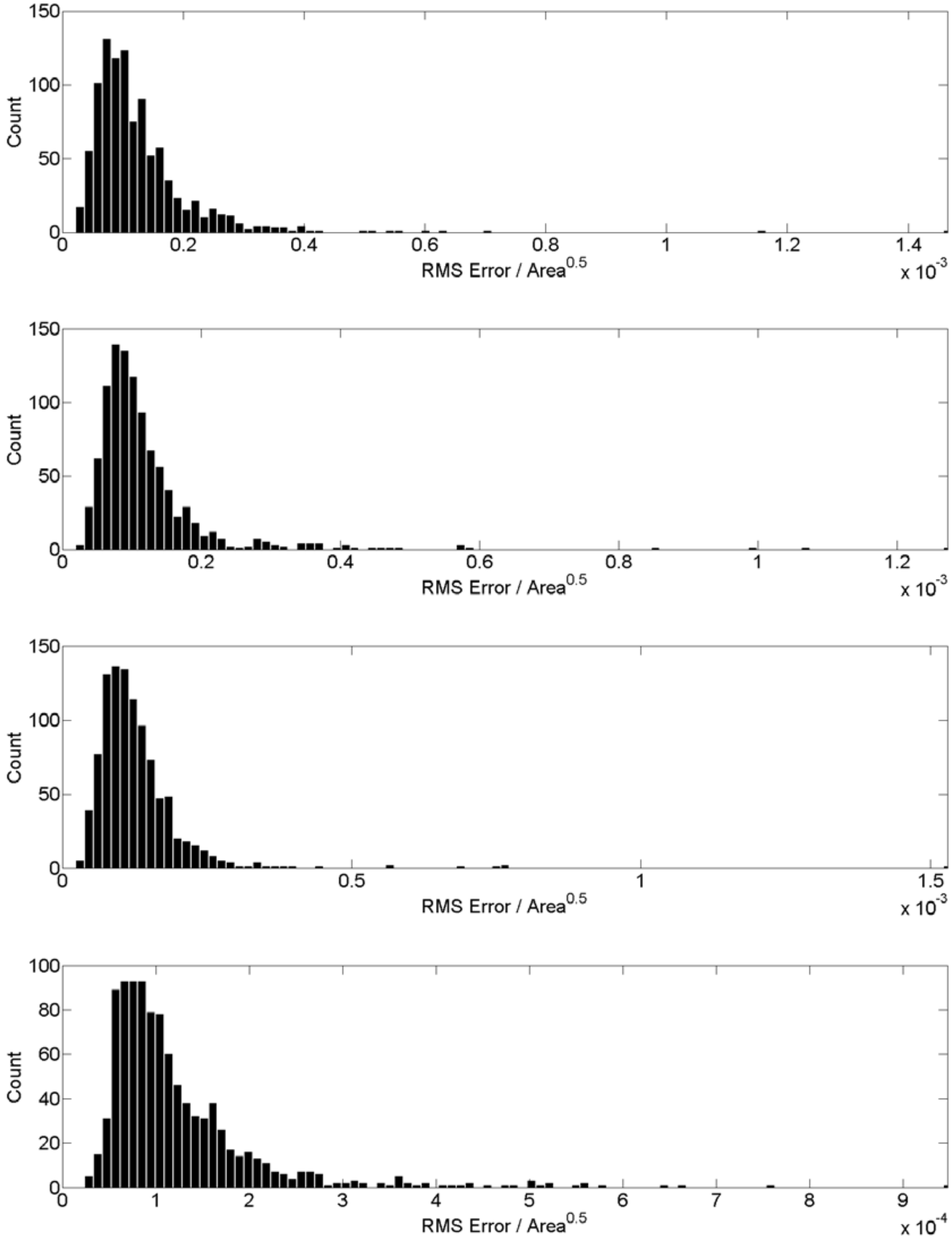


Figure 8.53: Normalized Error Distributions for Reconstructed Training Strain Time Histories from Gages 6, 8, 9, 11 (Top to Bottom)

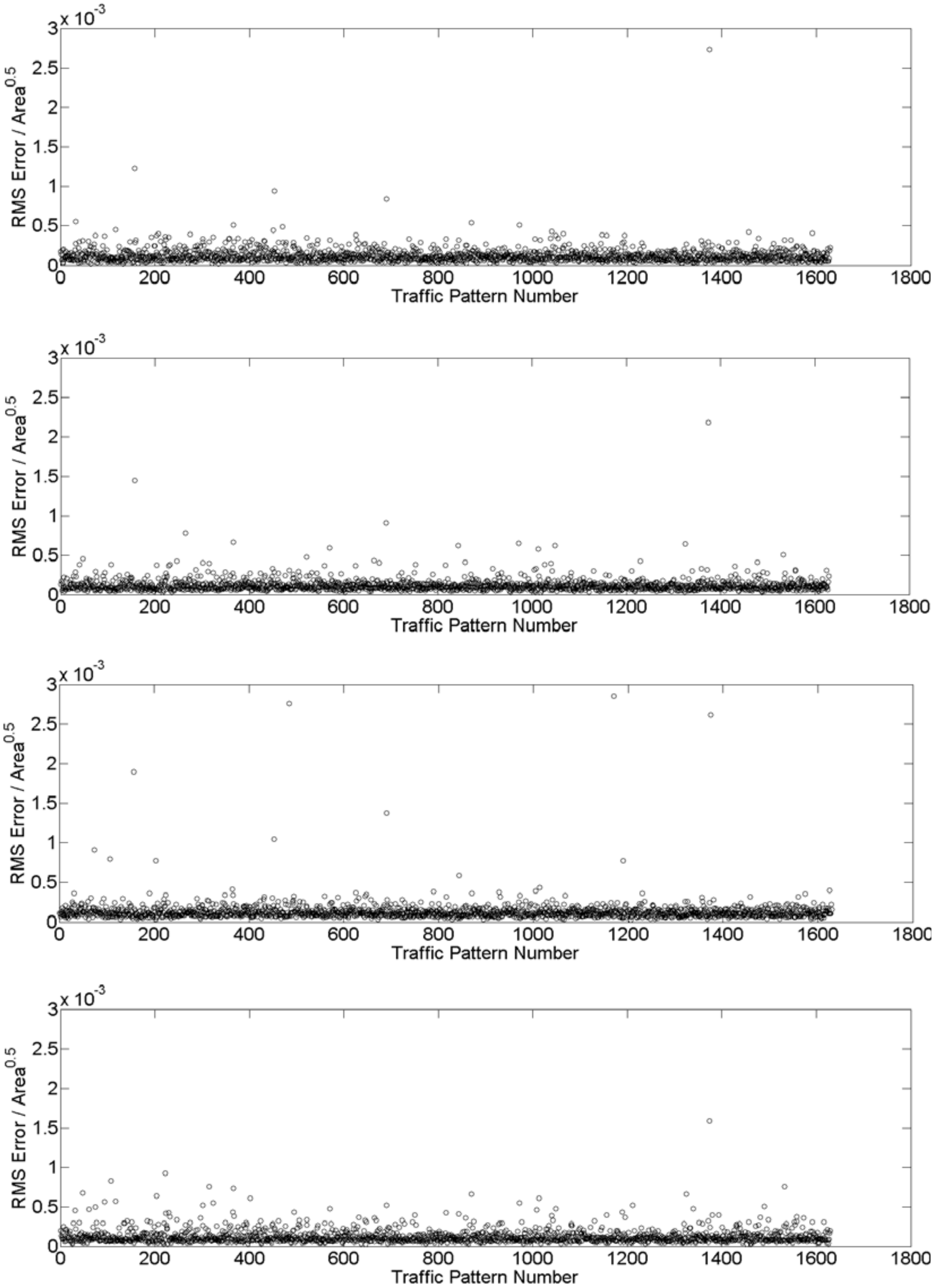


Figure 8.54: Scatter Plots of Normalized Errors for Reconstructed 2003 Test Data - Strain Gages 6, 8, 9, 11 (Top to Bottom)

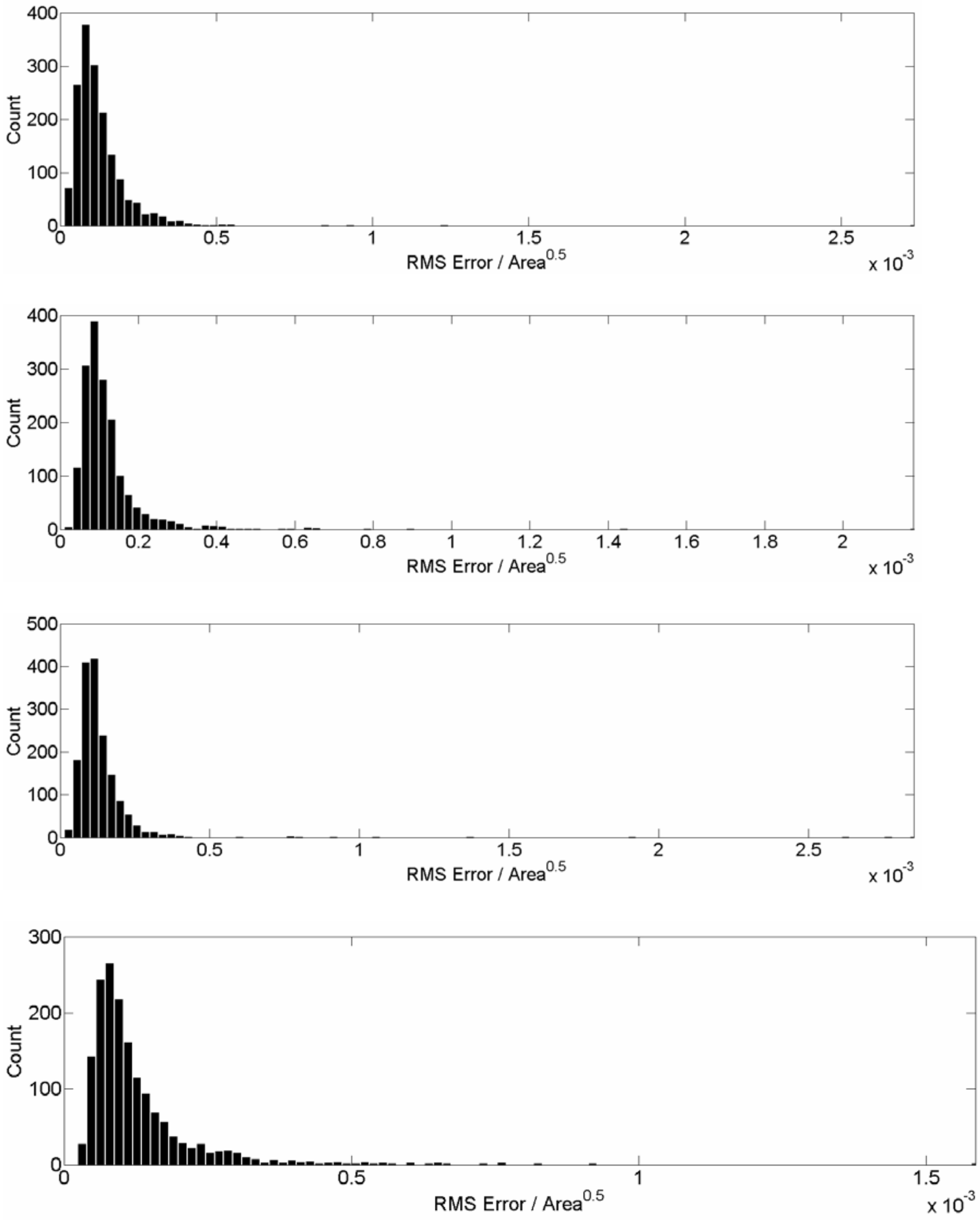


Figure 8.55: Distributions of Normalized Errors for Reconstructed 2003 Test Data - Strain Gages 6, 8, 9, 11 (Top to Bottom)

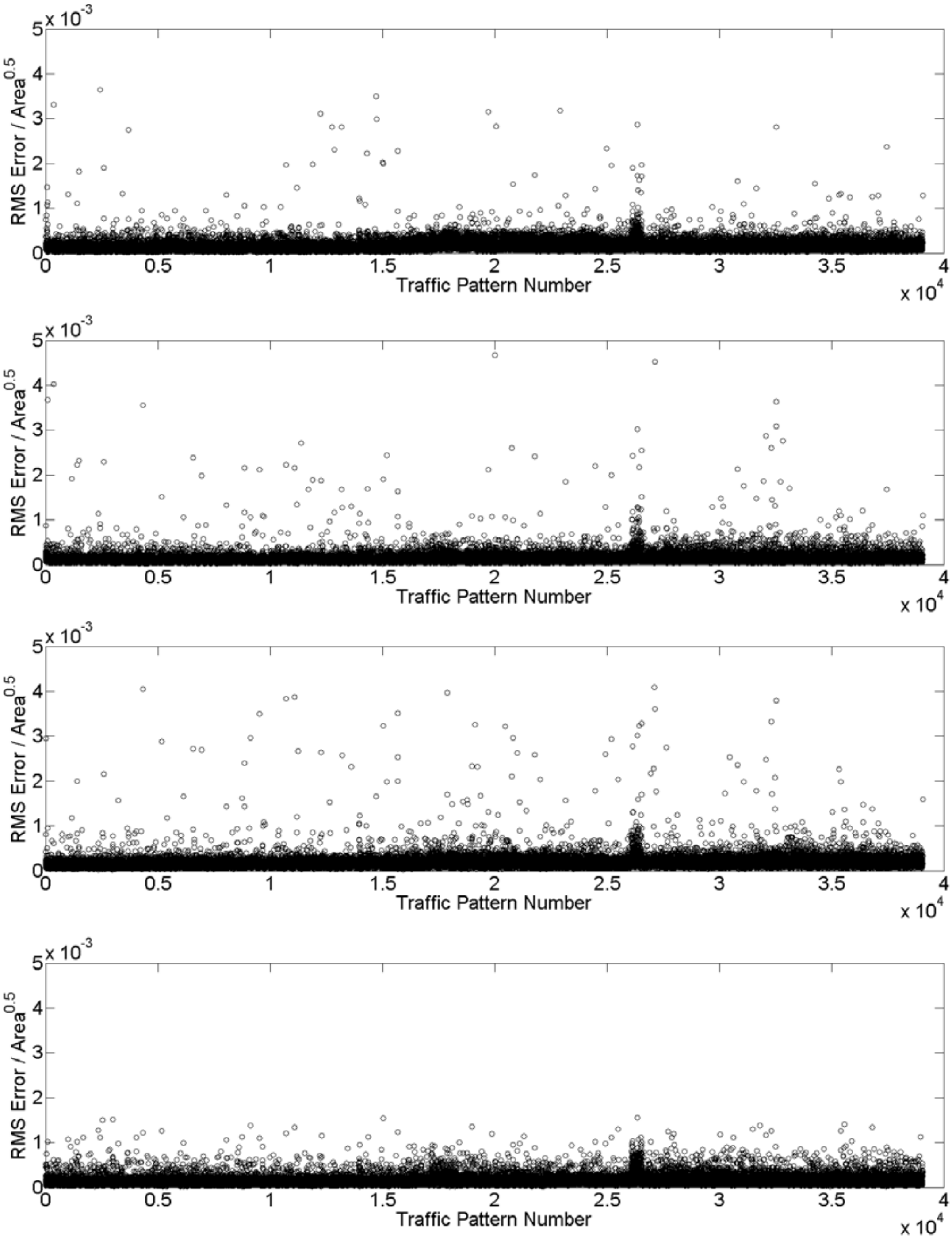


Figure 8.56: Scatter Plots of Normalized Errors for Reconstructed 2004 Test Data - Strain Gages 6, 8, 9, 11 (Top to Bottom)

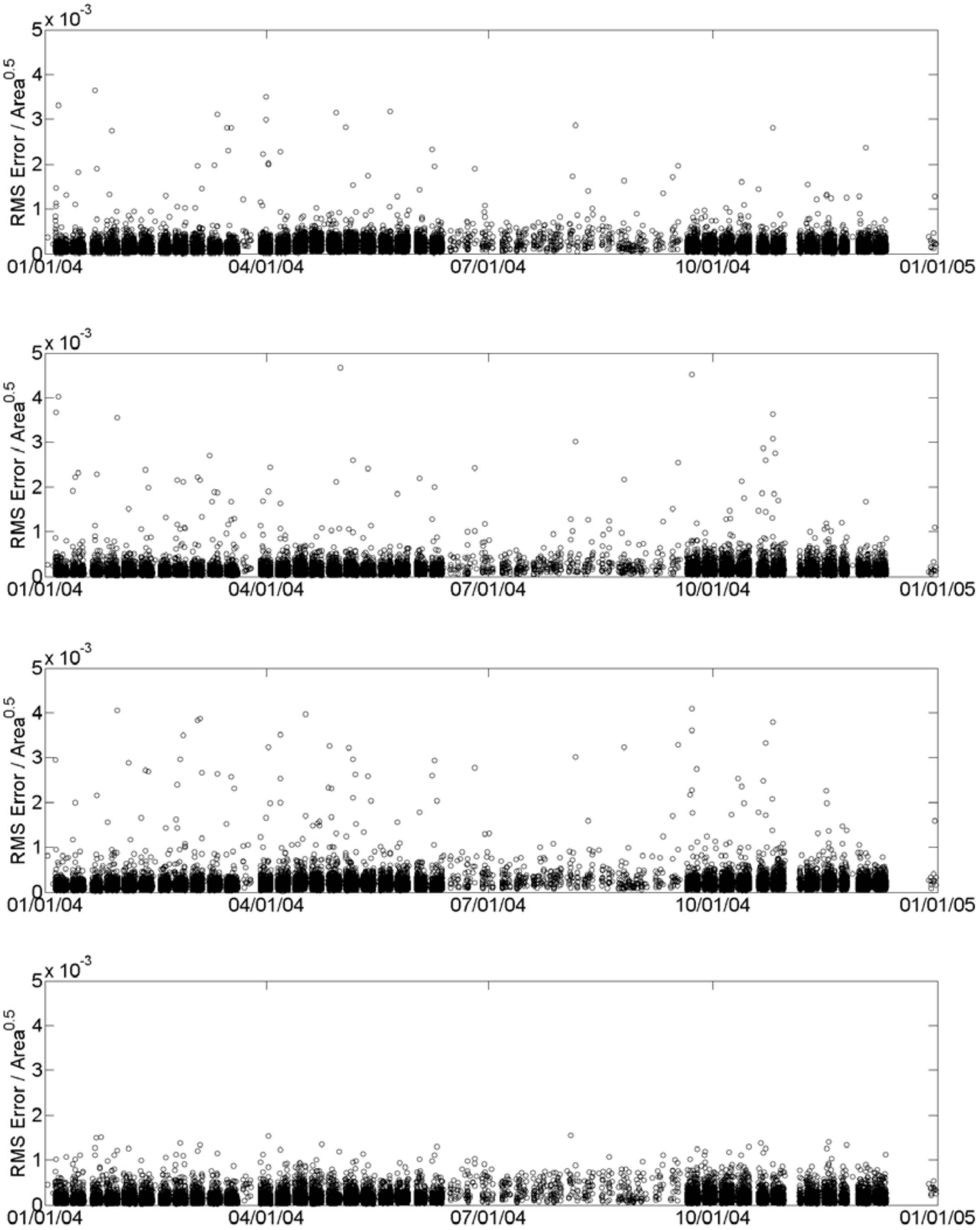


Figure 8.57: Normalized Errors for Reconstructed 2004 Test Data - Strain Gages 6, 8, 9, 11 (Top to Bottom) versus Date Recorded

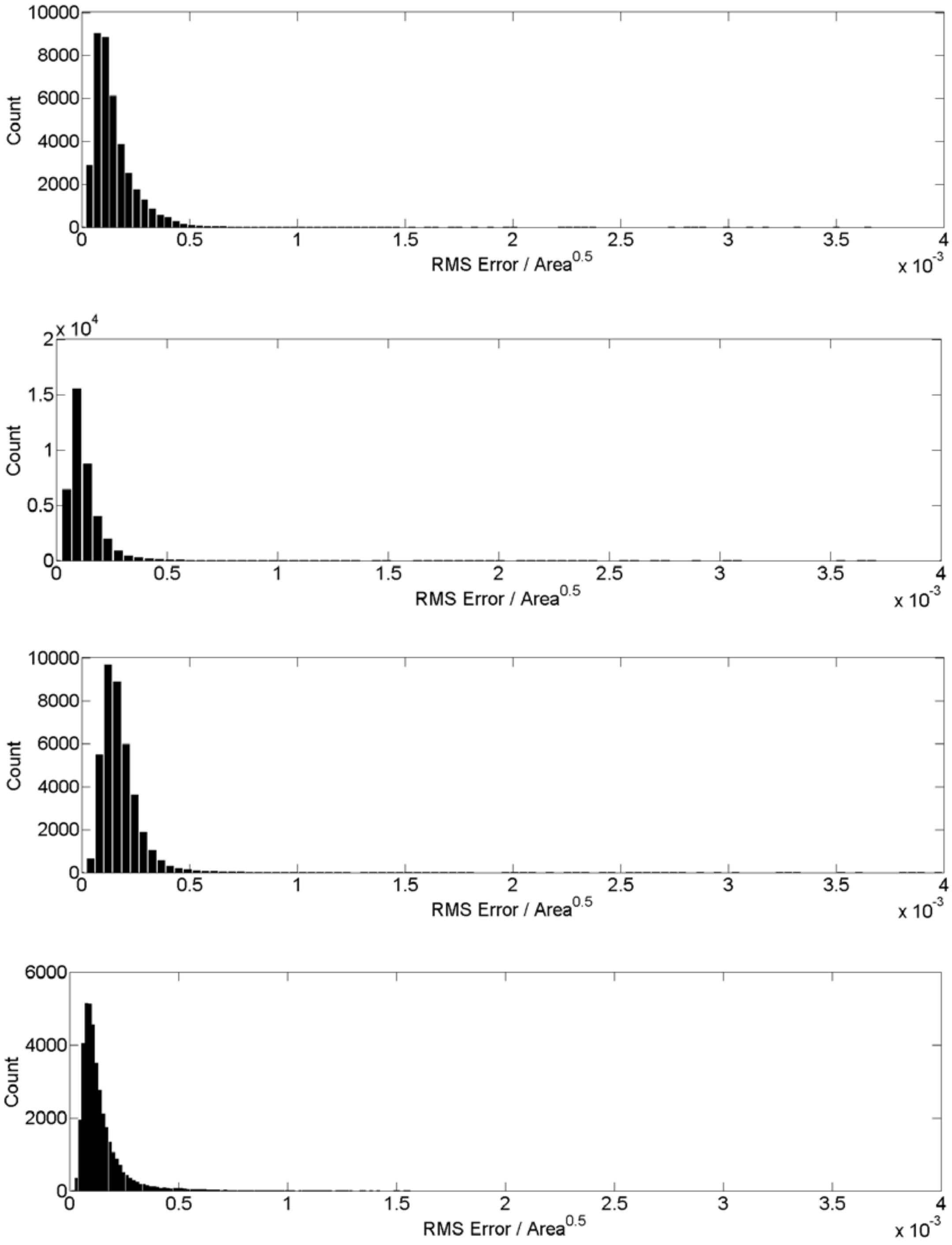


Figure 8.58: Distributions of Normalized Errors for Reconstructed 2004 Test Data - Strain Gages 6, 8, 9, 11 (Top to Bottom)

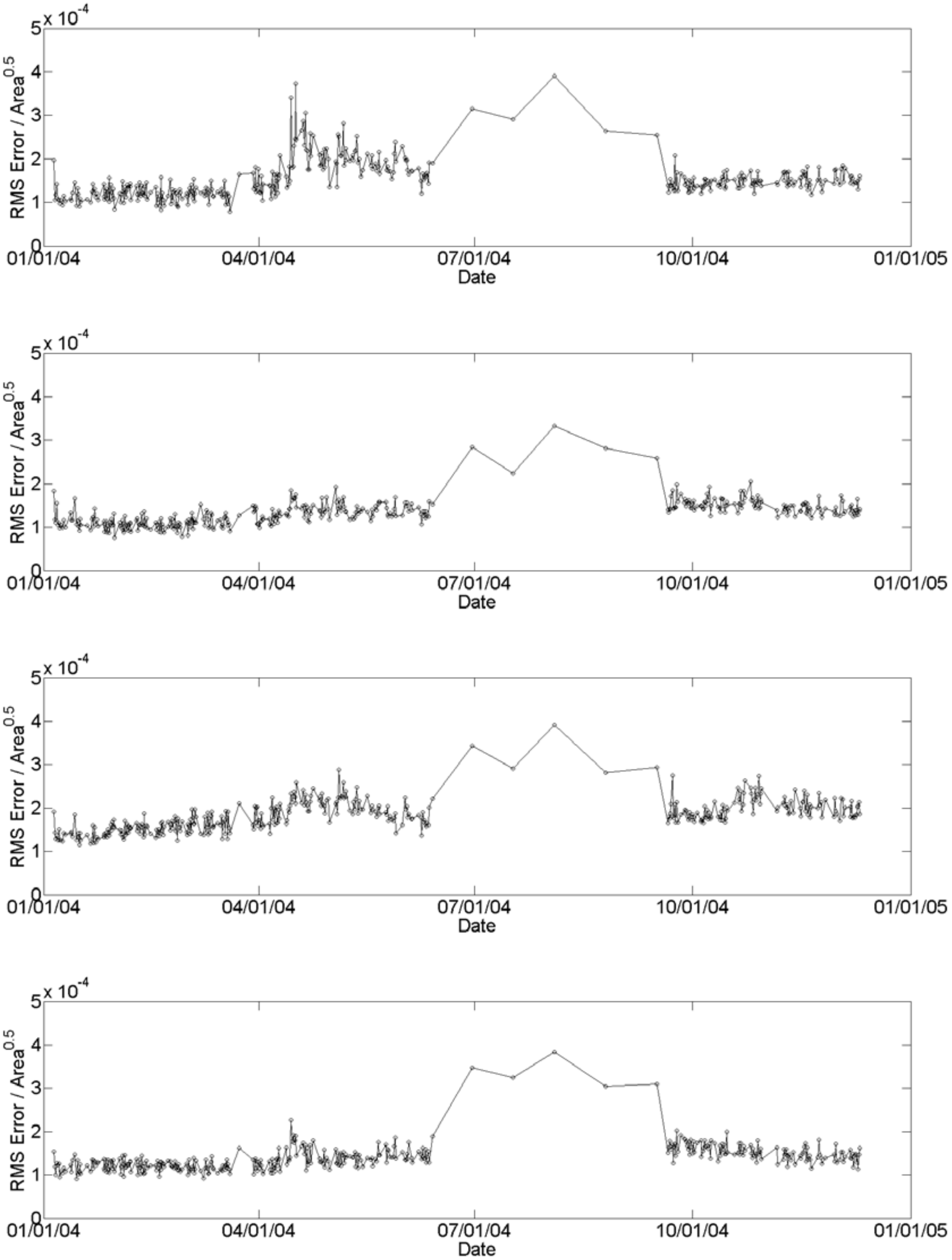


Figure 8.59: Averaged Normalized Errors for Reconstructed 2004 Test Data - Strain Gages 6, 8, 9, 11 (Top to Bottom) versus Date Recorded

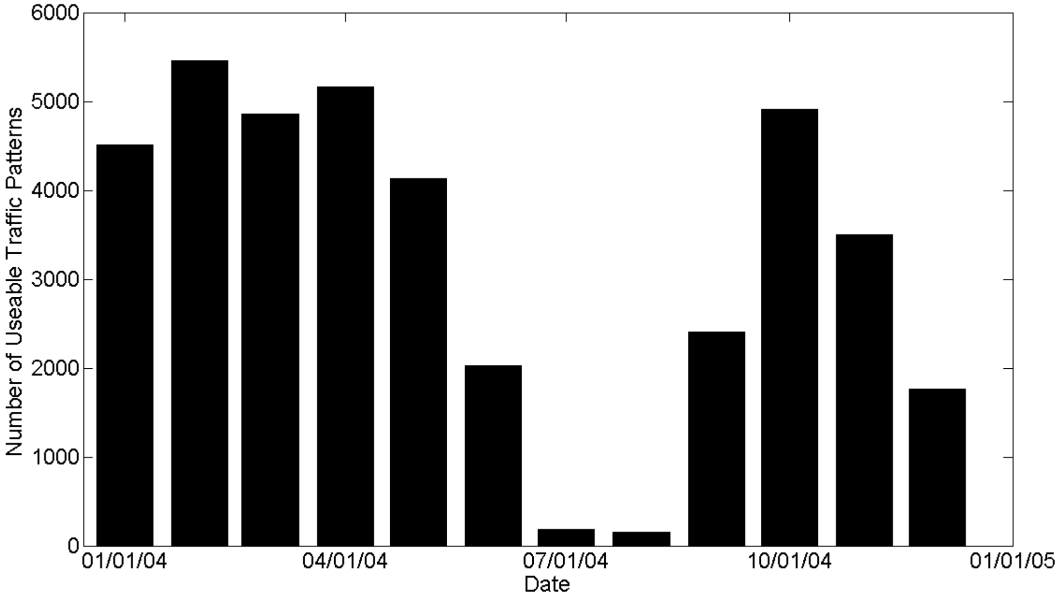


Figure 8.60: Number of Available 2004 Testing Records

Table 8.2: Mean, Standard Deviation, and Maximum Normalized Error for Reconstructed Strain Data

	Mean	Standard Deviation	Maximum
Strain Gage 6	1.262 x10^{-4}	9.324 x10^{-4}	1.465 x10^{-3}
Strain Gage 8	1.182 x10^{-4}	9.168 x10^{-4}	1.272 x10^{-3}
Strain Gage 9	1.268 x10^{-4}	8.282 x10^{-4}	1.528 x10^{-3}
Strain Gage 11	1.242 x10^{-4}	8.854 x10^{-4}	9.472 x10^{-4}

Table 8.3: Mean, Standard Deviation, and Maximum Normalized Error for Reconstructed 2003 Testing Data

	Mean	Standard Deviation	Maximum
Strain Gage 6	$1.025 \text{ x}10^{-4}$	$1.043 \text{ x}10^{-4}$	$2.733 \text{ x}10^{-3}$
Strain Gage 8	$1.190 \text{ x}10^{-4}$	$9.620 \text{ x}10^{-5}$	$2.181 \text{ x}10^{-3}$
Strain Gage 9	$1.346 \text{ x}10^{-4}$	$1.434 \text{ x}10^{-4}$	$2.852 \text{ x}10^{-3}$
Strain Gage 11	$1.249 \text{ x}10^{-4}$	$9.850 \text{ x}10^{-5}$	$1.586 \text{ x}10^{-3}$

Table 8.4: Mean, Standard Deviation, and Maximum Normalized Error for Reconstructed 2004 Testing Data

	Mean	Standard Deviation	Maximum
Strain Gage 6	$1.253 \text{ x}10^{-4}$	$1.225 \text{ x}10^{-4}$	$3.647 \text{ x}10^{-3}$
Strain Gage 8	$1.332 \text{ x}10^{-4}$	$1.215 \text{ x}10^{-4}$	$4.667 \text{ x}10^{-3}$
Strain Gage 9	$1.830 \text{ x}10^{-4}$	$1.056 \text{ x}10^{-4}$	$4.098 \text{ x}10^{-3}$
Strain Gage 11	$1.398 \text{ x}10^{-4}$	$1.212 \text{ x}10^{-4}$	$1.586 \text{ x}10^{-3}$

8.4.4 Neural Network Training

Using the 1000 traffic scenarios described in Section 8.4.1, three neural networks were trained to use the PC's extracted from channel 8 strain time histories to predict strains for channels 6, 9, and 11 (each neural network is responsible for one target gage). The training curves for these three neural networks are shown in Fig's. 8.63 – 8.65. Optimal training cycles for each of the networks are reported in Table 8.5.

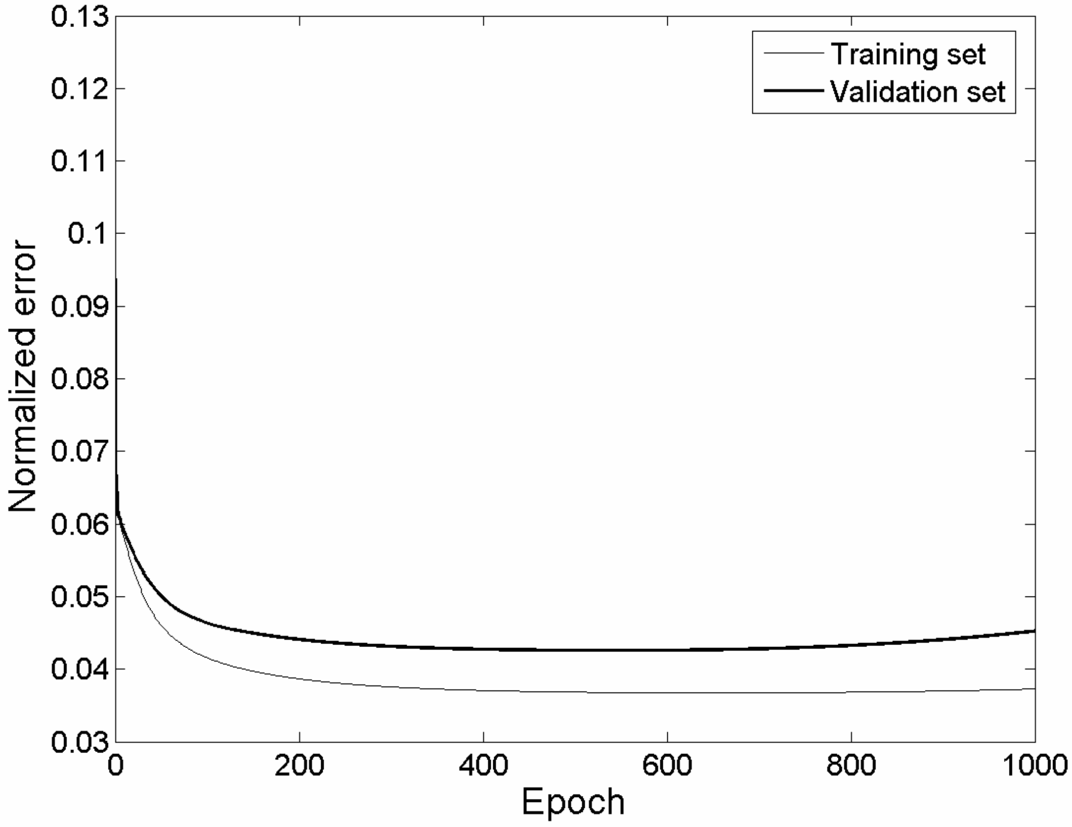


Figure 8.61: Neural Network Training Curve for Predicting Channel 6 Strains

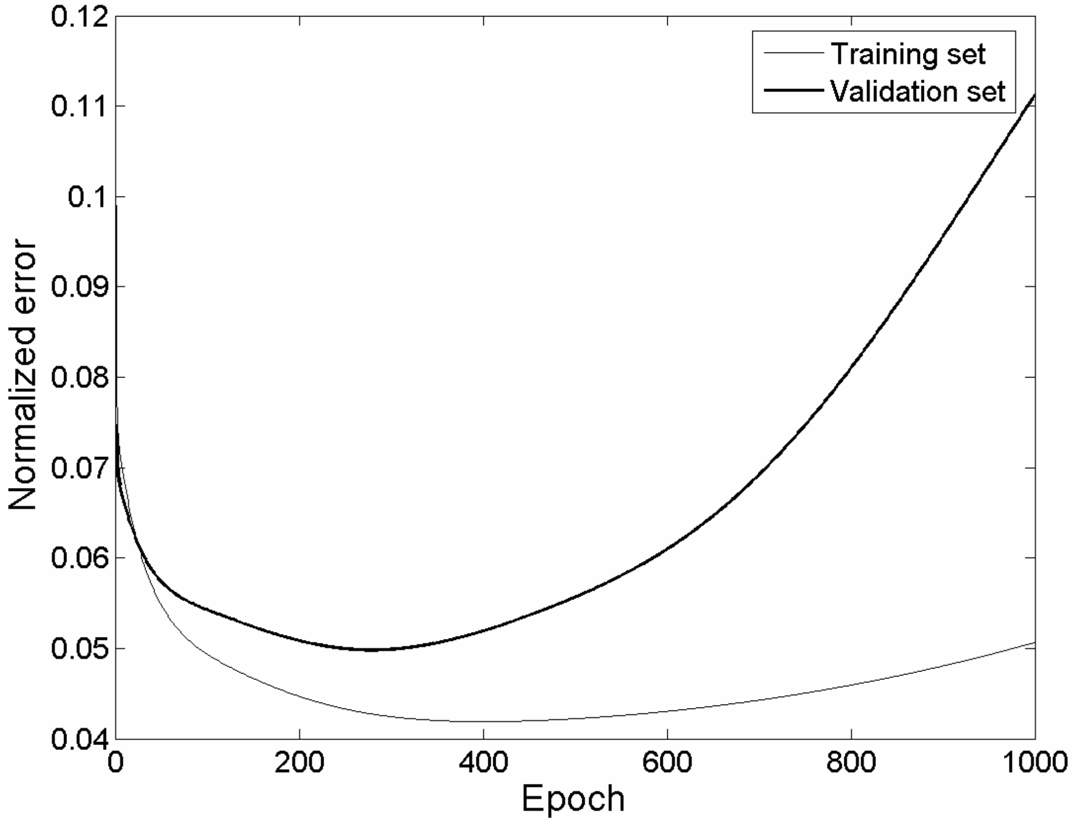


Figure 8.62: Neural Network Training Curve for Predicting Channel 9 Strains

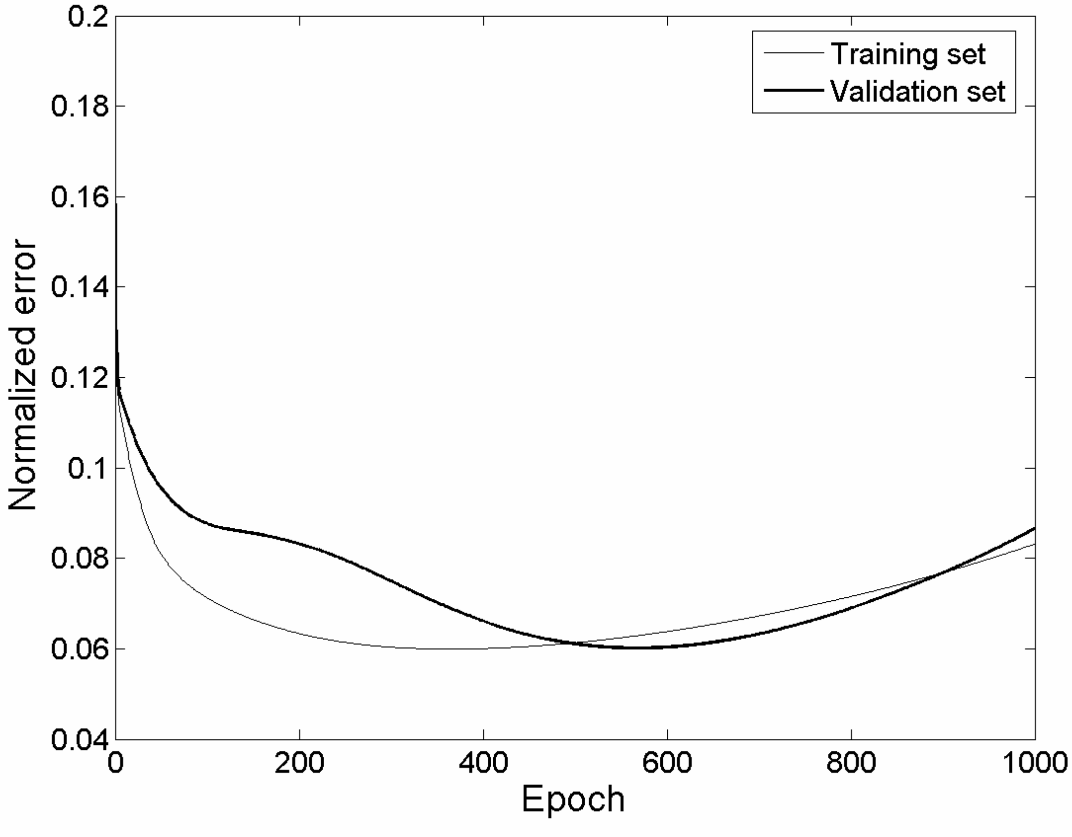


Figure 8.63: Neural Network Training Curve for Predicting Channel 11 Strains

Table 8.5: Optimal Training Cycle for Each Neural Network

Target	Optimal Training Cycle
Strain Gage 6	543
Strain Gage 9	279
Strain Gage 11	567

8.4.5 Application of Neural Network to Typical 2-Axle Traffic Test Data

8.4.5.1 Normalized Errors between Predicted and Measured Testbed Strains for 2003 Test Data

The series of neural networks were applied to the set of Principal Components derived from the 1,629 records in the 2003 test data. The normalized errors between the measured and predicted strains were calculated (Fig. 8.66). Histograms of these errors are plotted in Fig. 8.67 and the mean error, standard deviation, and maximum error are included in Table 8.6.

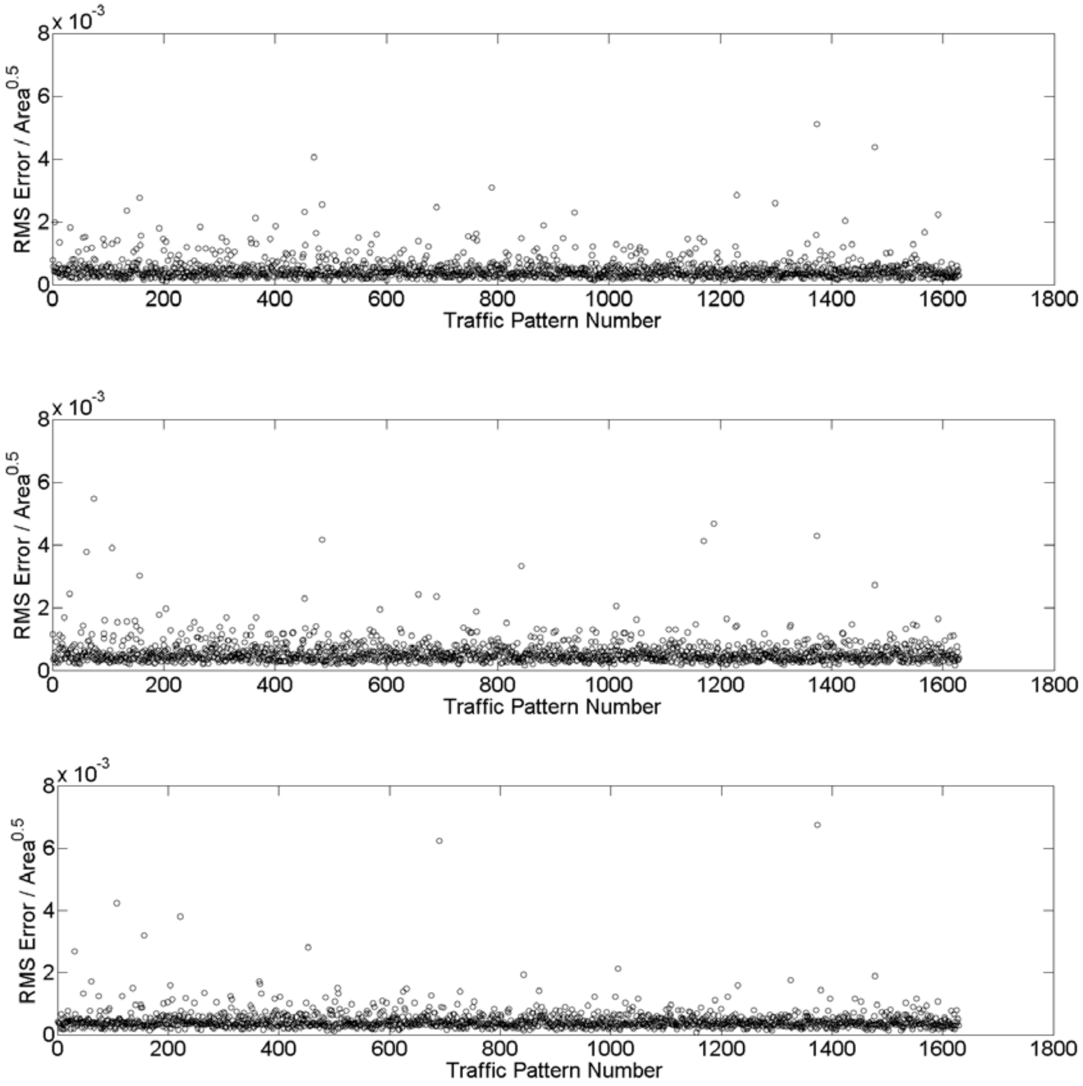


Figure 8.64: Normalized Error for Predicted Channel 6, 9 , 11 (Top to Bottom)

Strains Recorded During December 2003

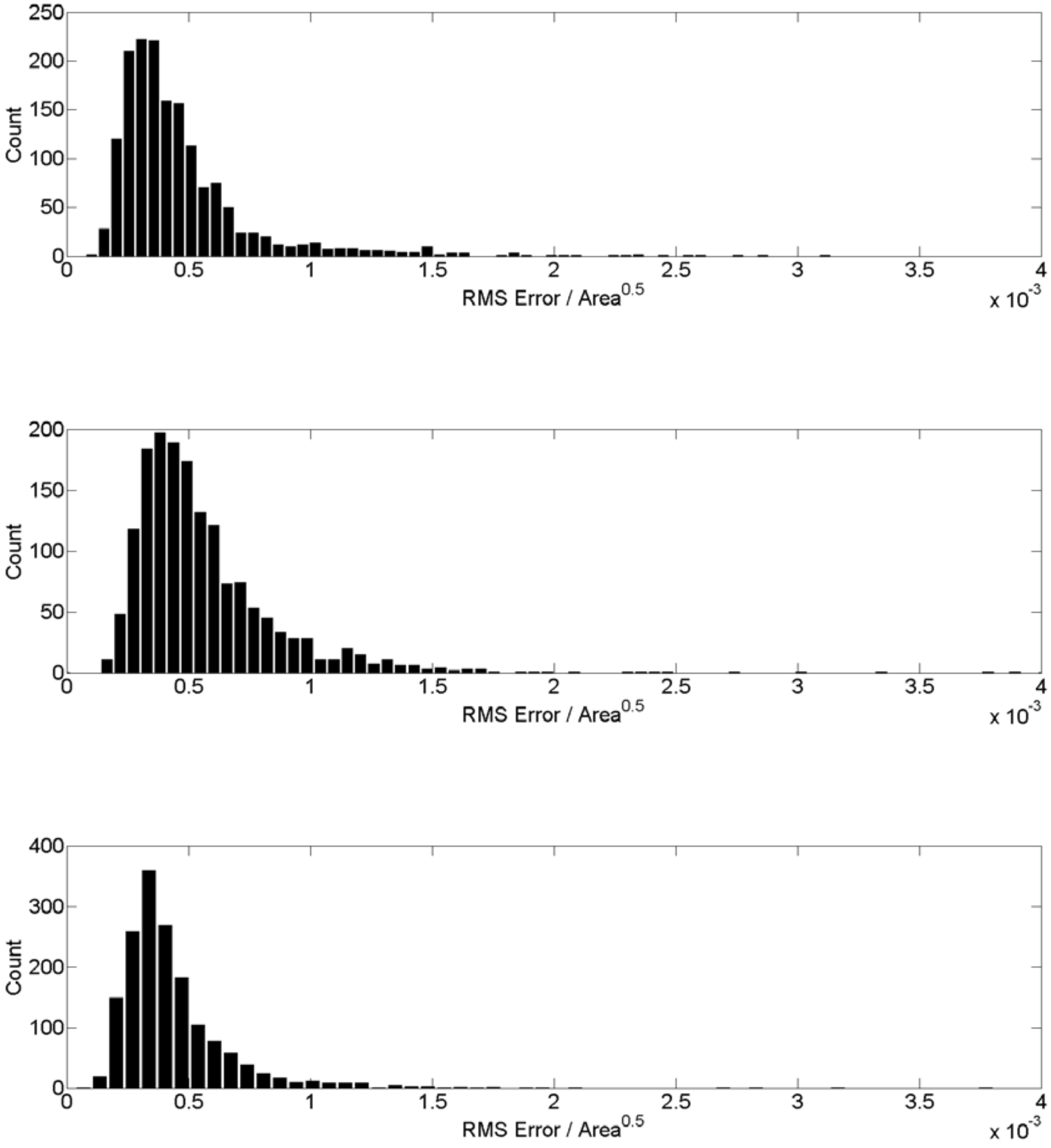


Figure 8.65: Normalized Error Distributions for Predicted Channel 6, 9 , 11 (Top to Bottom) Strains Recorded During December 2003

8.4.5.2 Normalized Errors between Predicted and Measured Testbed Strains for 2004 Test Data

As with the aforementioned 2003 testing data, the 39,062 scenarios in the 2004 testing data set were analyzed using the trained neural networks. Again, the normalized errors were calculated for each target sensor and plotted in Fig's. 8.68 – 8.70 along with histograms of the errors in Fig. 8.71. A 100-point average was formed for each of the strain gage errors and are plotted in Fig's. 8.72 – 8.74 along with the histograms in Fig. 8.75. As with the errors formed from the previous reconstruction data the largest errors are during the summer months. The mean error, standard deviation, and maximum error are included in Table 8.7. Through comparison with the original reconstruction data the mean errors are 4.4 to 5.2 times larger.

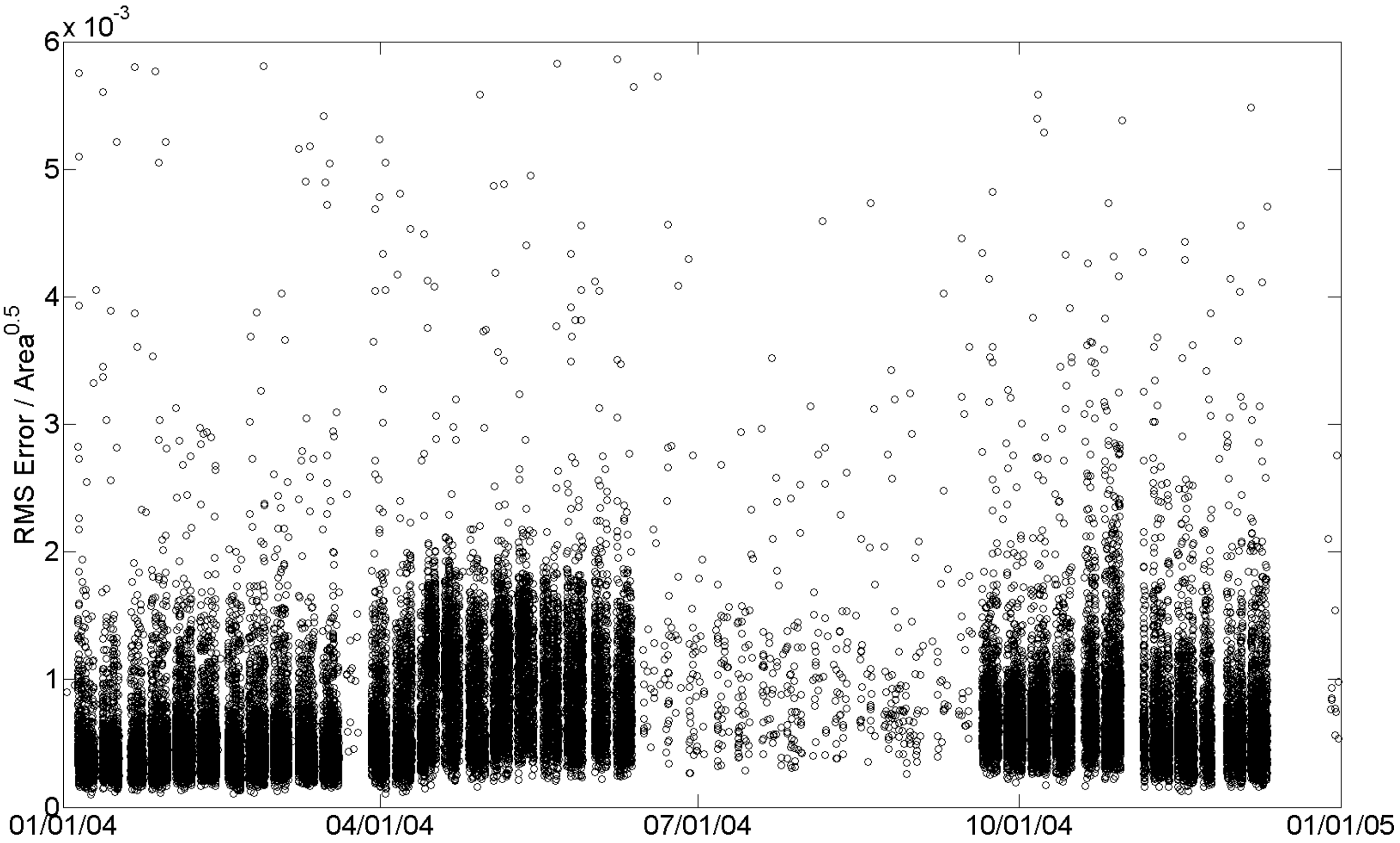


Figure 8.66: Normalized RMS Error for Channel 6 Predicted Strains

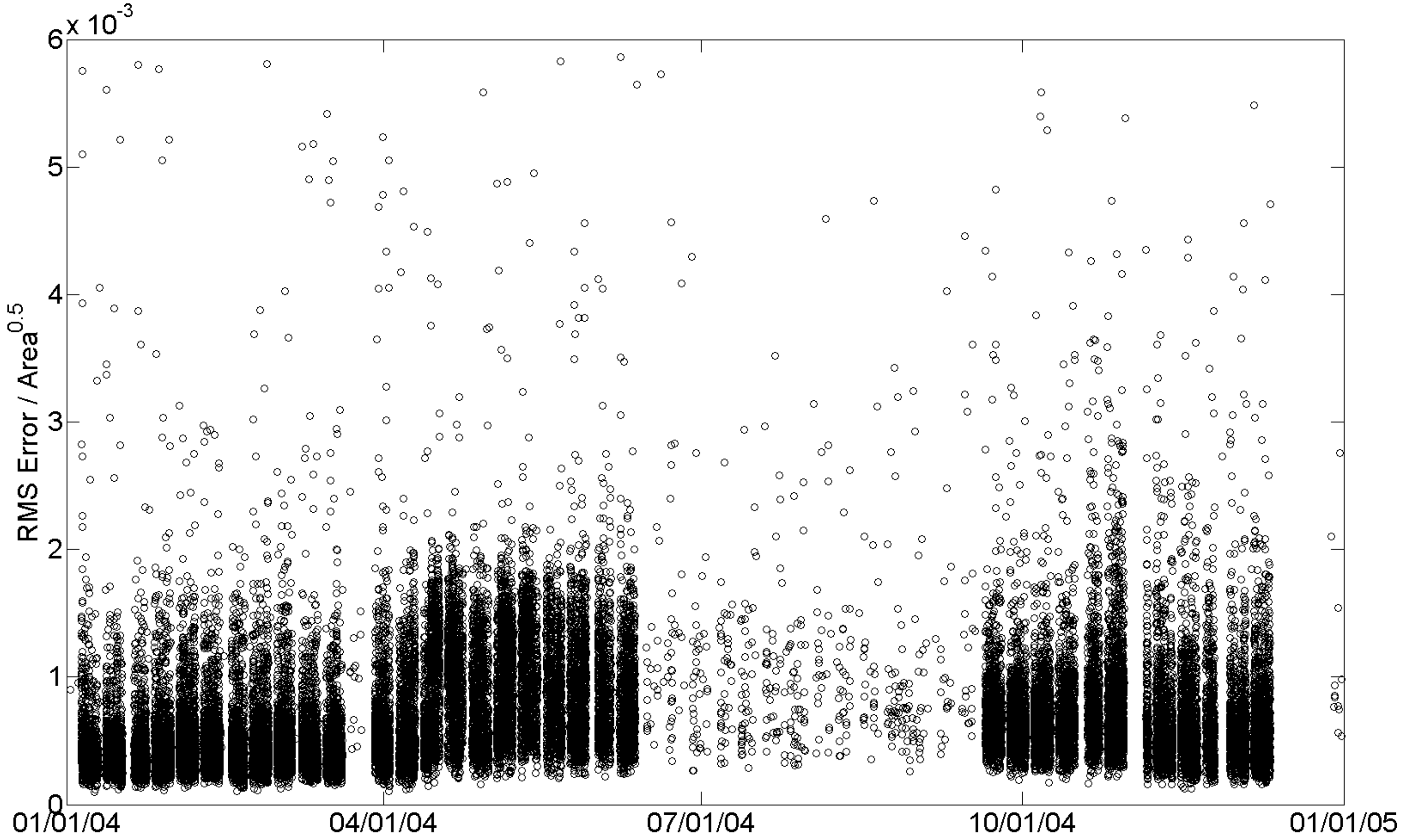


Figure 8.67: Normalized RMS Error for Channel 9 Predicted Strains

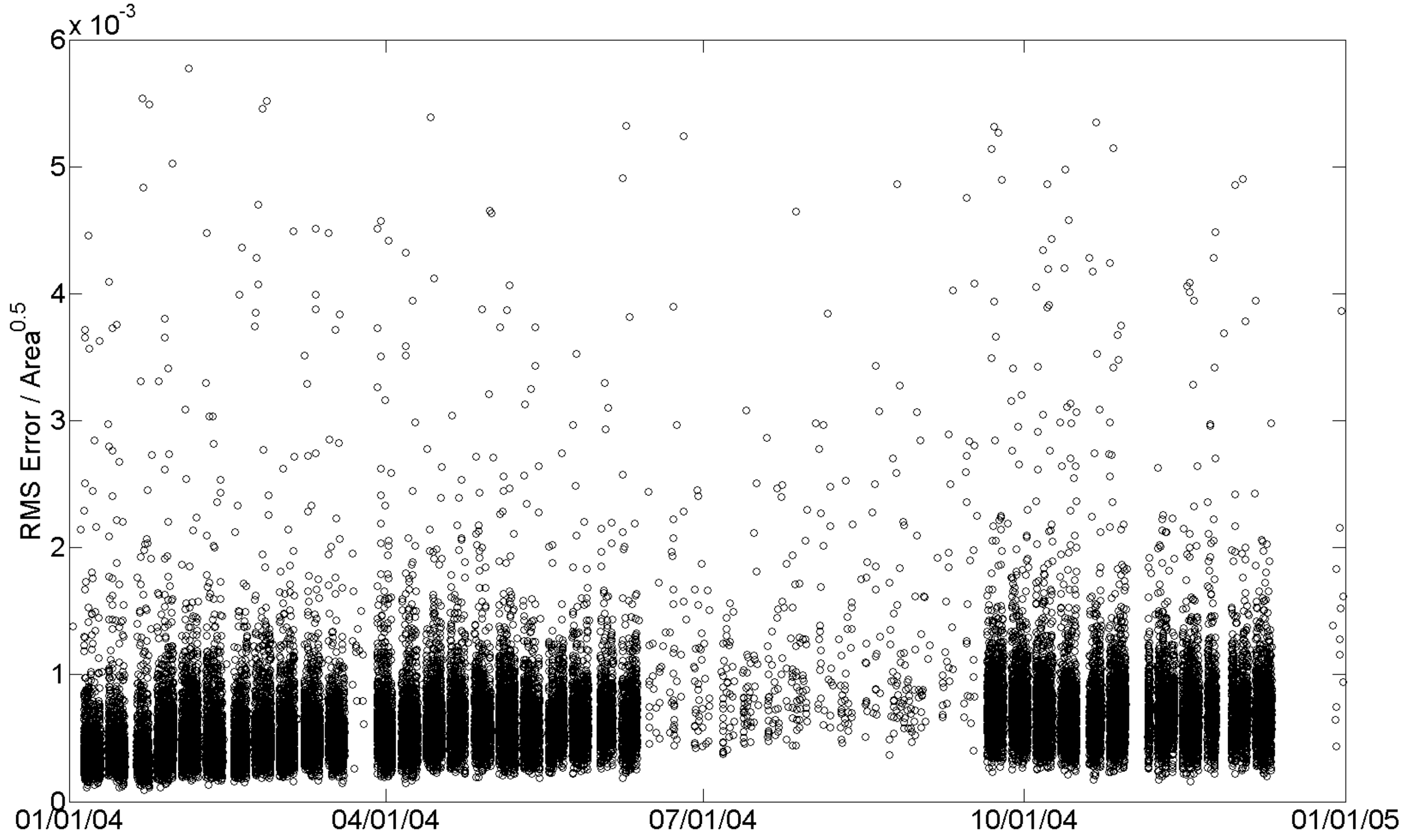


Figure 8.68: Normalized RMS Error for Channel 11 Predicted Strains

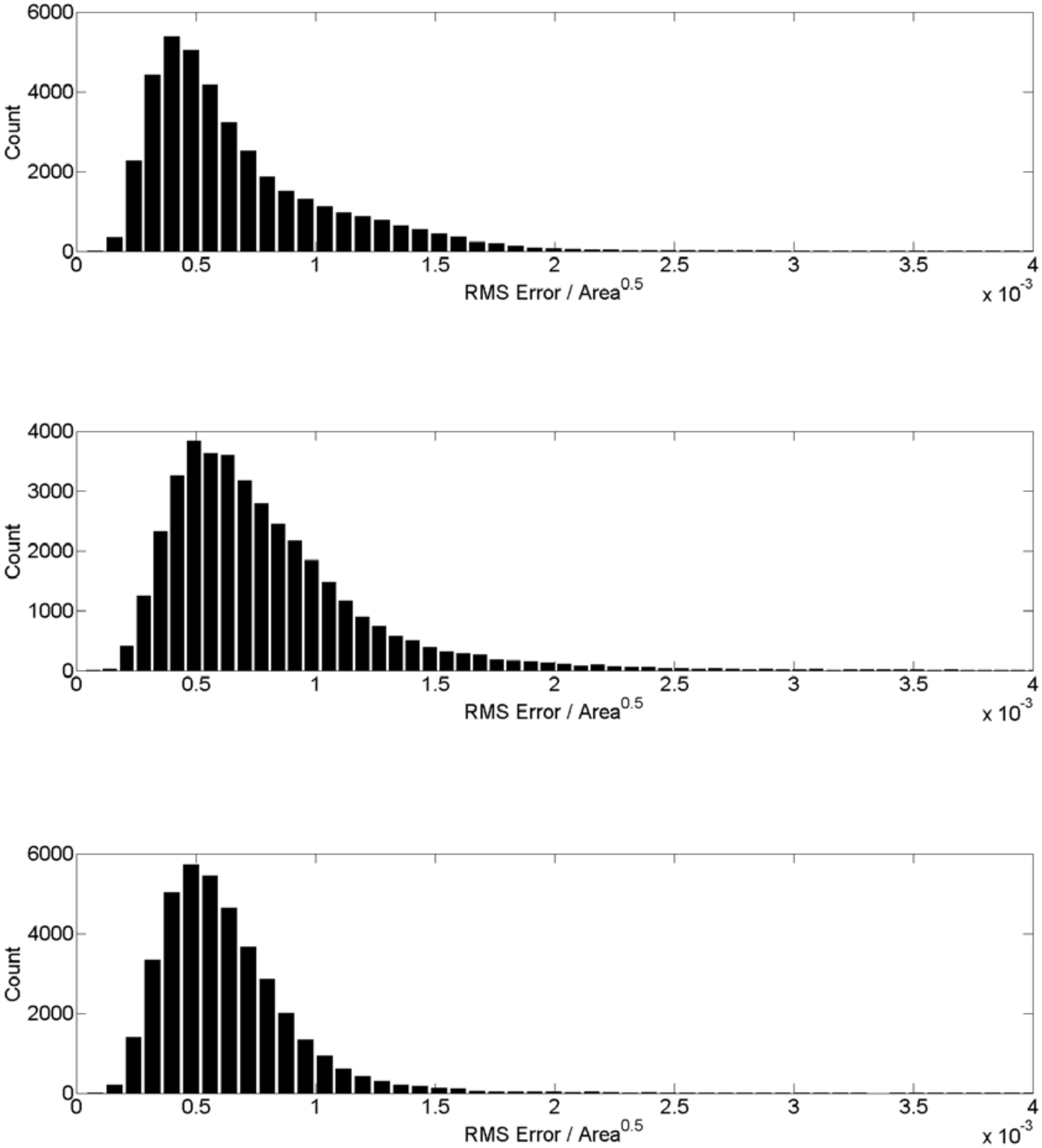


Figure 8.69: Histograms of Normalized RMS Errors for Channels 6 (top), 9 (middle), and 11 (bottom)

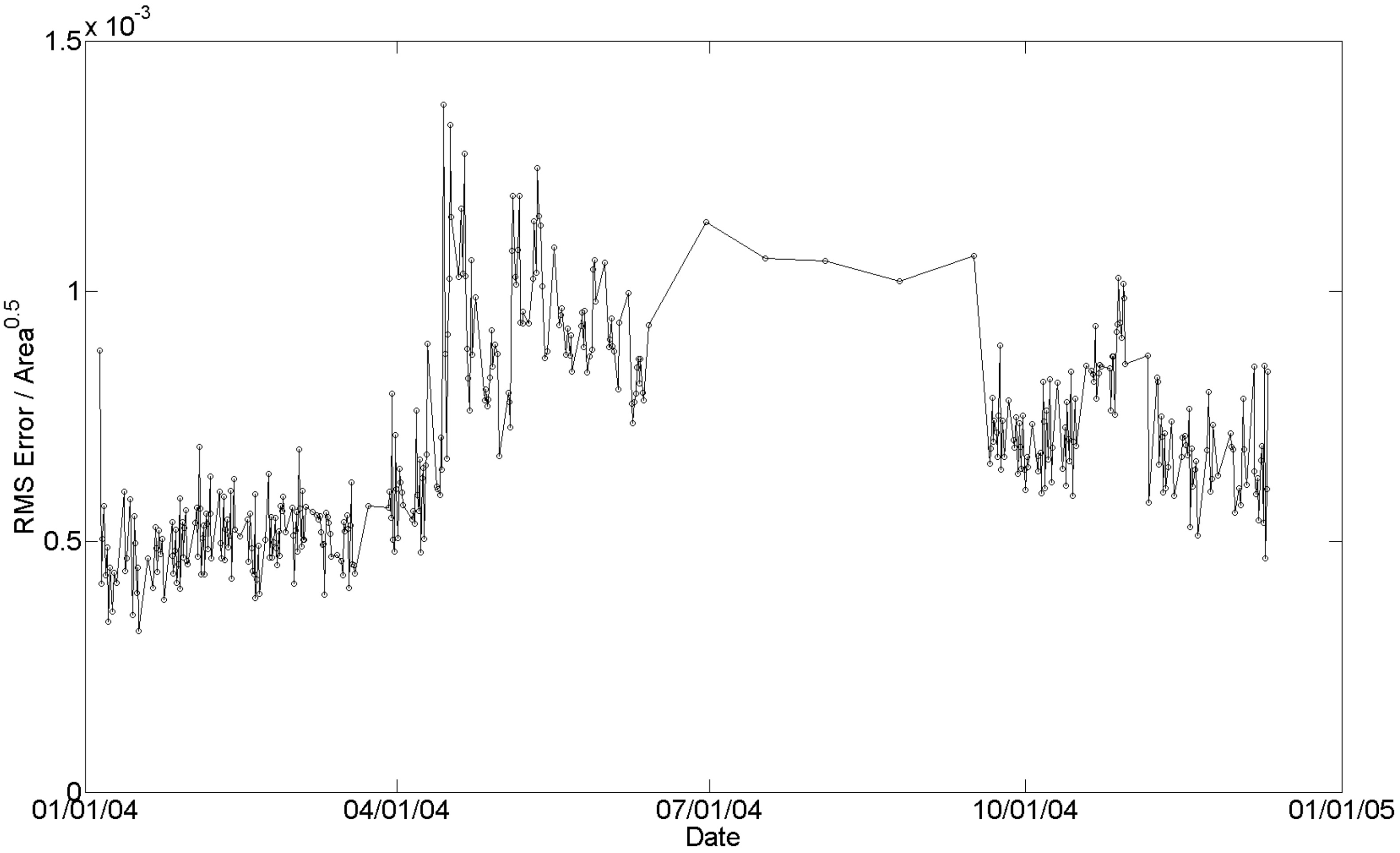


Figure 8.70: 100-point Mean Normalized RMS Errors for Channel 6

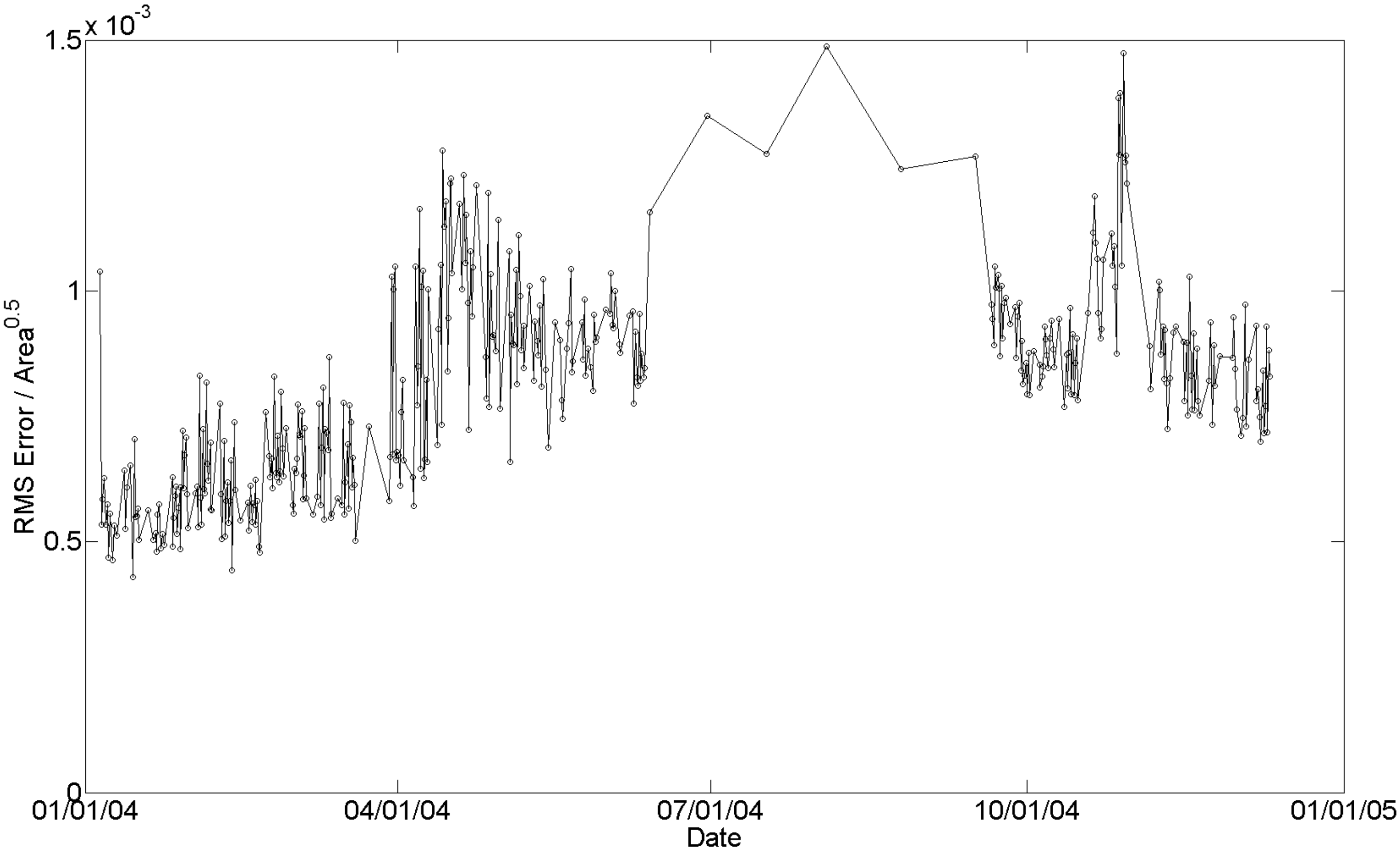


Figure 8.71: 100-point Mean Normalized RMS Errors for Channel 9

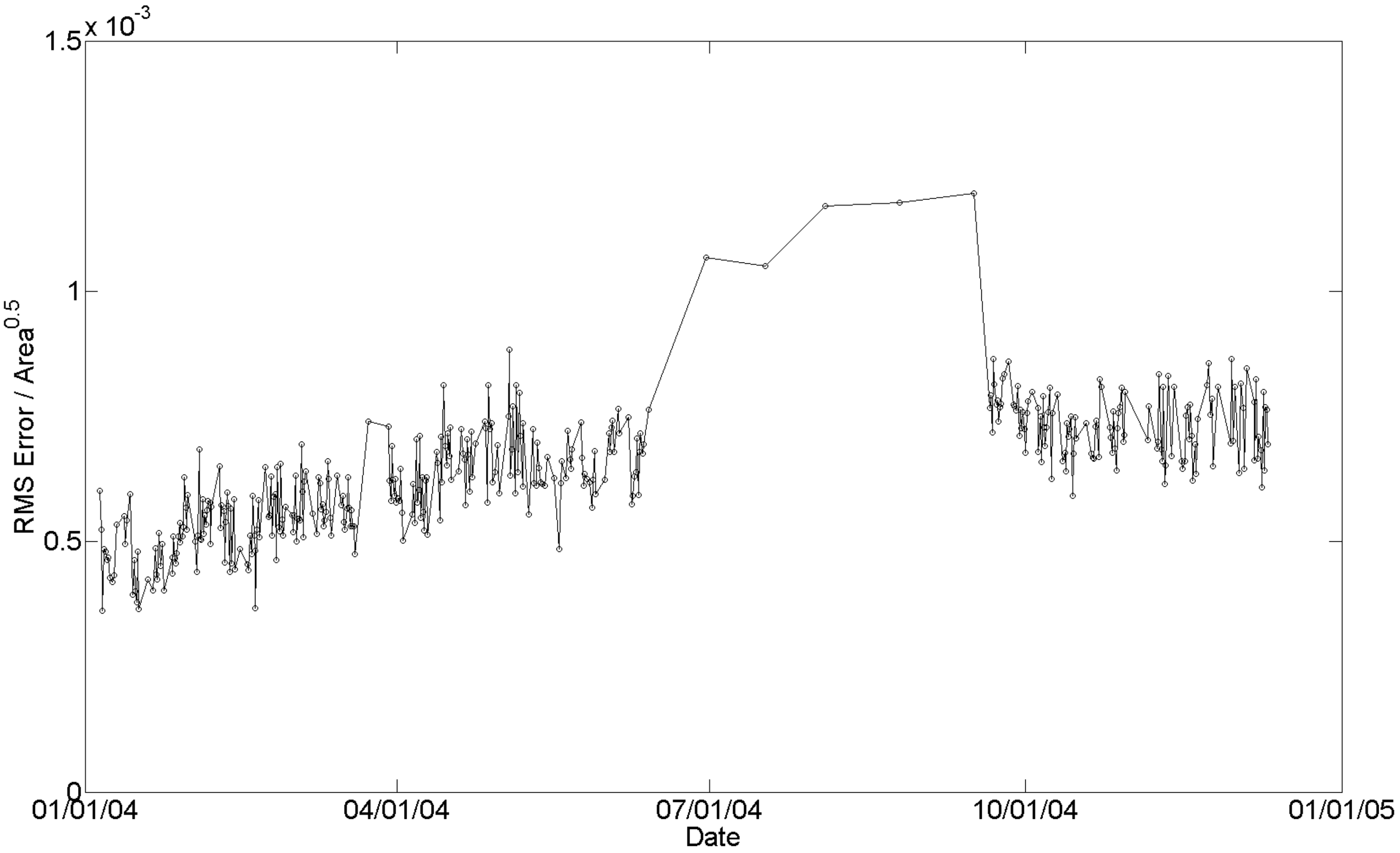


Figure 8.72: 100-point Mean Normalized RMS Errors for Channel 11

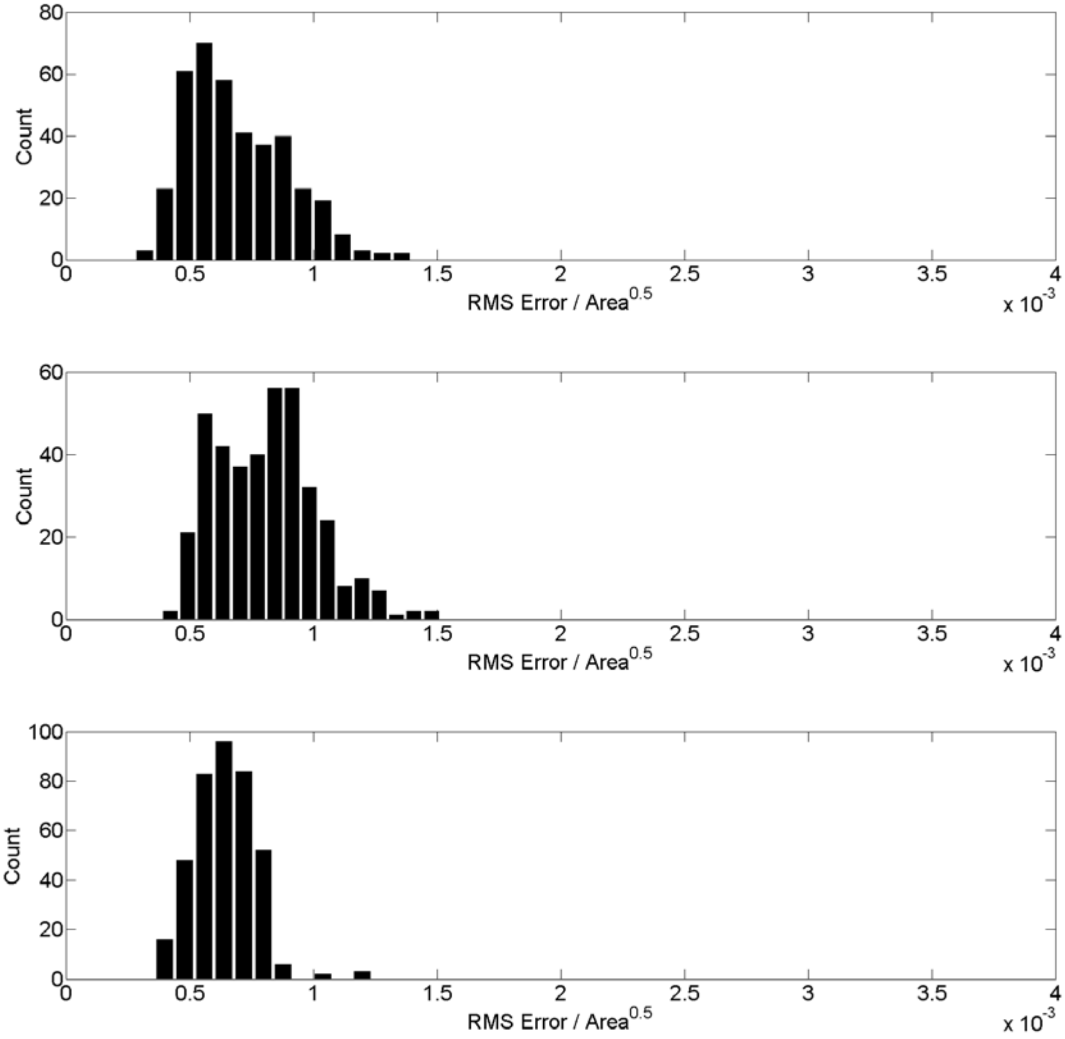


Figure 8.73: Histograms of 100-Point Mean Normalized RMS Errors for Channels 6 (top), 9 (middle), and 11 (bottom)

Table 8.6: Mean, Standard Deviation, and Maximum Normalized Error for Predicted Strains from 2003 Testing Data

	Mean	Standard Deviation	Maximum
Strain Gage 6	4.790×10^{-4}	3.589×10^{-4}	5.102×10^{-3}
Strain Gage 9	5.787×10^{-4}	3.937×10^{-4}	5.482×10^{-3}
Strain Gage 11	4.496×10^{-4}	3.502×10^{-5}	6.735×10^{-3}

Table 8.7: Mean, Standard Deviation, and Maximum Normalized Error for Predicted Strains from 2004 Testing Data

	Mean	Standard Deviation	Maximum
Strain Gage 6	6.887×10^{-4}	4.547×10^{-4}	7.986×10^{-3}
Strain Gage 9	8.088×10^{-4}	5.038×10^{-4}	7.035×10^{-3}
Strain Gage 11	6.418×10^{-4}	3.710×10^{-4}	7.985×10^{-3}

The normalized errors for the predicted and reconstructed strains were averaged by month and are plotted in 8.76 and 8.77. In comparing these two sets of figures, it can be seen that all of the data follows the same trend. In addition, the averaged monthly errors in the predicted data is three to four times higher than those from the original reconstructed data. Again, the increase in error during the summer months is very pronounced. The average monthly errors for the predicted and reconstructed errors are summarized in Table 8.7. Histograms of the monthly errors are plotted in Figure 8.78 for the predicted data and in Fig. 8.79 for the reconstructed data.

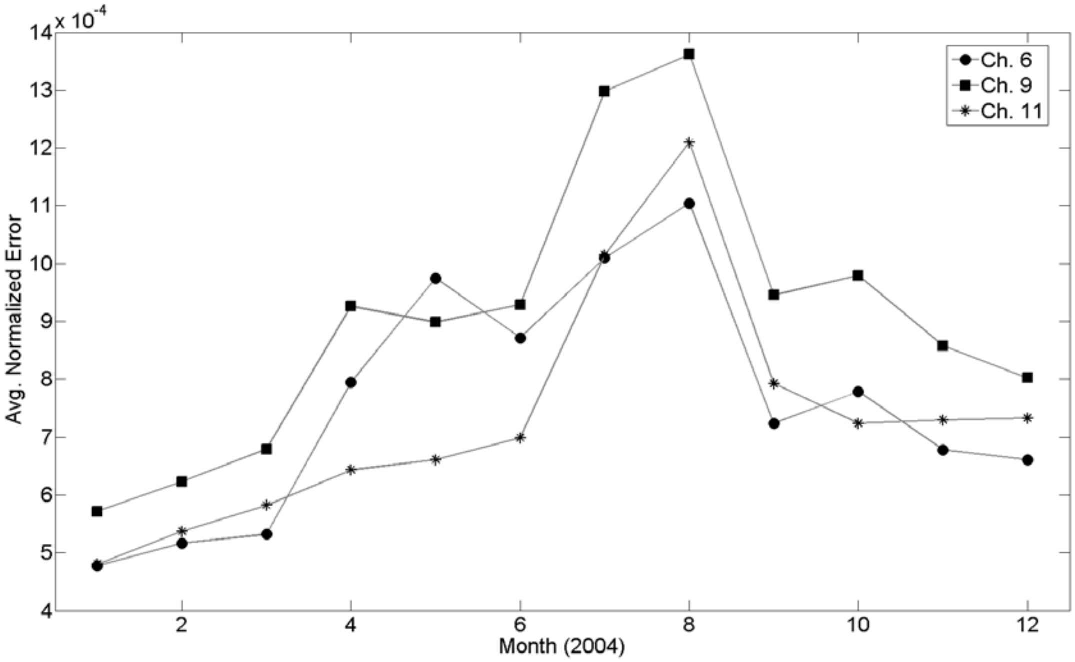


Figure 8.74: Average Monthly Normalized Error for Predicted Strains

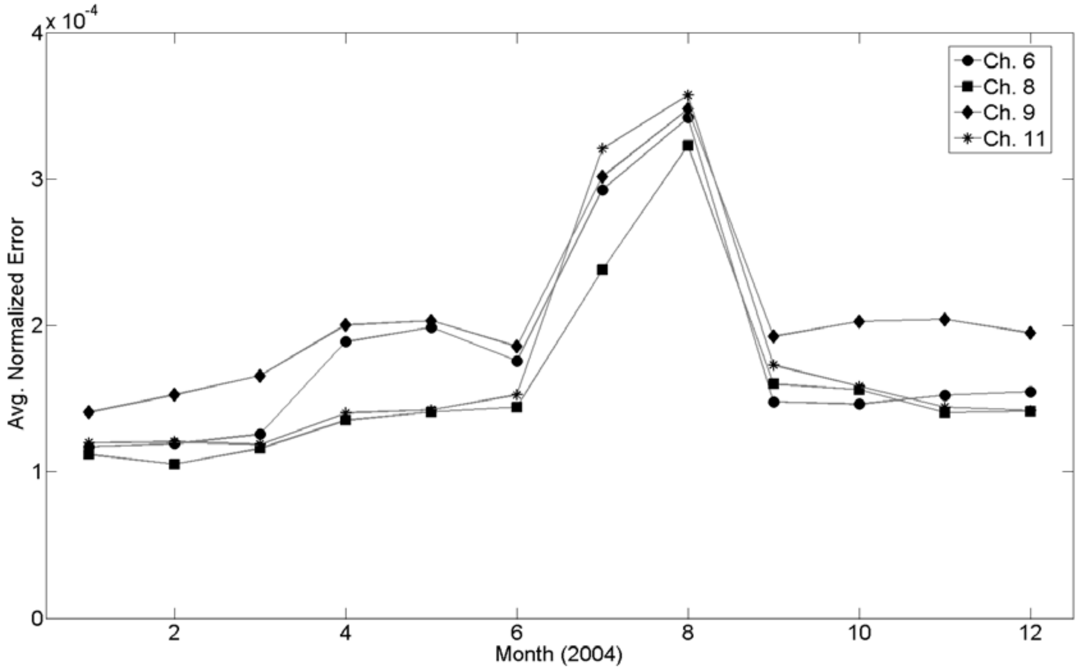


Figure 8.75: Average Monthly Normalized Error for Reconstructed Strains

Table 8.8: Average Monthly Errors for Predicted and Reconstructed Data

Month (2004)	Average Monthly Normalized Error						
	Predicted Strains From Neural Network (x10^{-4})			Reconstructed Strains from Principal Components (x10^{-4})			
	Ch. 6	Ch. 9	Ch. 11	Ch. 6	Ch. 8	Ch. 9	Ch. 11
1	4.78	5.72	4.81	1.17	1.12	1.41	1.20
2	5.16	6.22	5.37	1.19	1.05	1.53	1.21
3	5.32	6.79	5.82	1.26	1.16	1.66	1.19
4	7.95	9.26	6.43	1.89	1.35	2.00	1.40
5	9.74	8.99	6.61	1.99	1.41	2.03	1.42
6	8.72	9.29	6.99	1.76	1.44	1.86	1.53
7	10.10	12.98	10.15	2.93	2.38	3.02	3.21
8	11.04	13.62	12.09	3.42	3.23	3.48	3.57
9	7.24	9.46	7.93	1.48	1.60	1.93	1.73
10	7.79	9.79	7.24	1.46	1.56	2.03	1.59
11	6.78	8.58	7.30	1.53	1.41	2.04	1.44
12	6.61	8.02	7.33	1.55	1.41	1.95	1.42

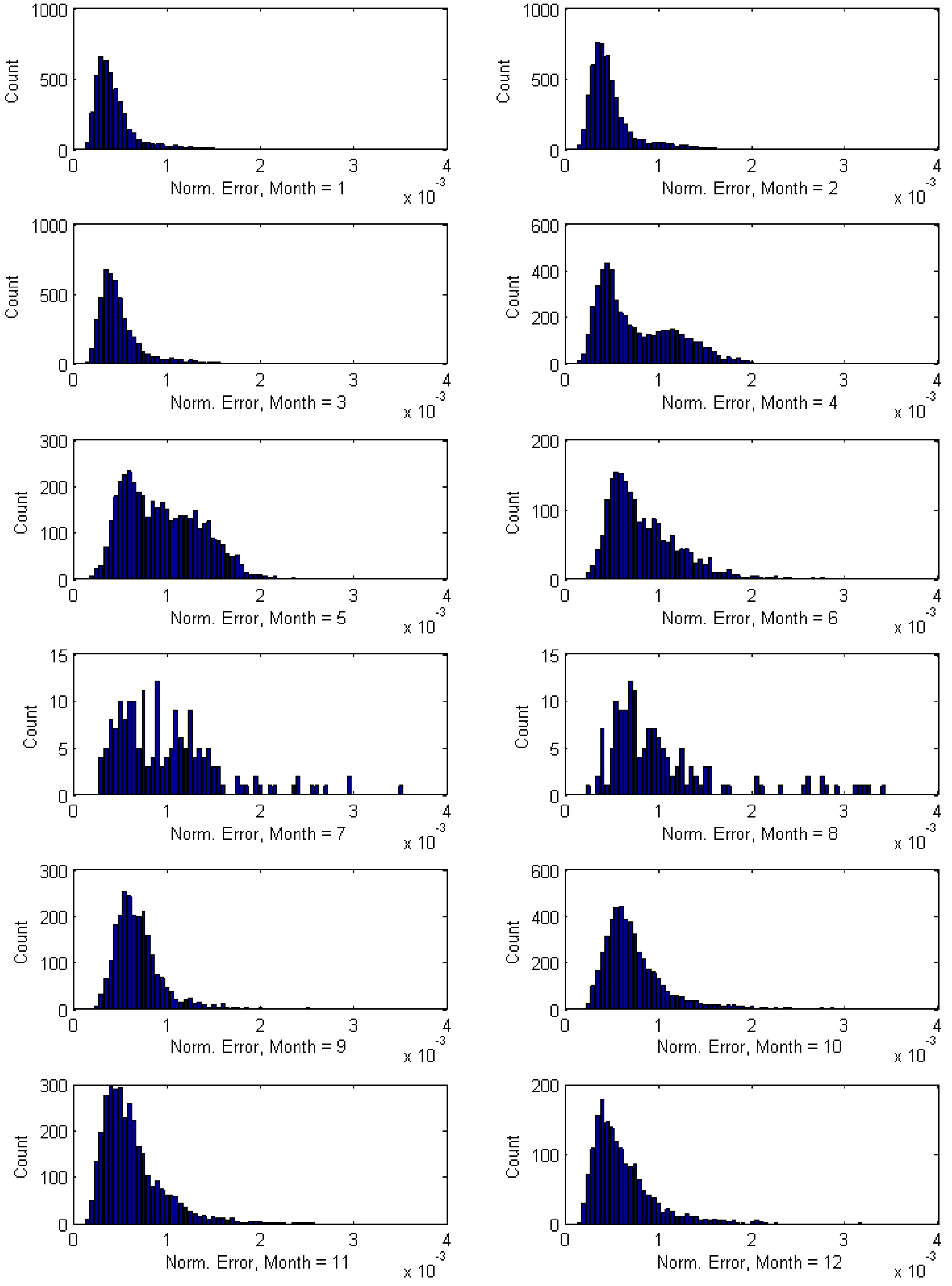


Figure 8.76: Normalized Errors Distributions of Gage 6 Predicted Strains by Month.

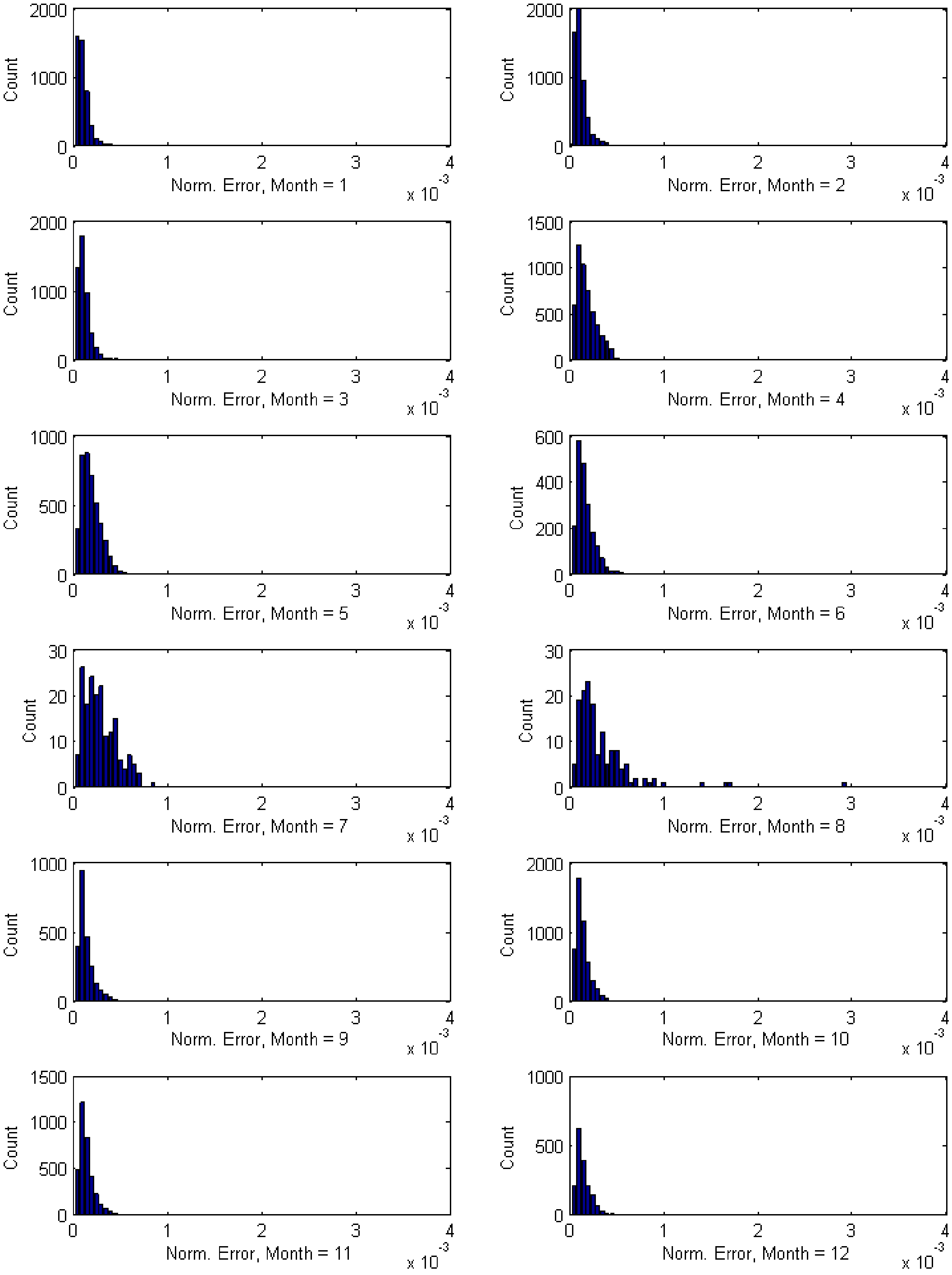


Figure 8.77: Normalized Errors Distributions of Gage 6 Reconstructed Strains by Month.

8.4.5.3 Reconstructed Strain Time Histories for Typical Median Normalized Errors

To visualize the quality of the match between the recorded strain time histories and those reconstructed from the neural network outputs, the time histories corresponding to the median monthly error in channel 6 were plotted for January, March, May, August, and November (Fig's. 8.80 – 8.84). In these figures for each of the 4 strain gages, the original recorded time history is plotted (solid gray line) along with the time history reconstructed using the inverse PCA transformation on the extracted PC's (dashed black line). For gages 6, 9, and 11, the strain time histories reconstructed from the neural network outputs are shown with the solid black line.

Within these figures, the correspondence between the recorded strains and those constructed from the inverse PCA transformation are excellent. However, in all of the channel 6, 9, and 11 data there are small errors between the recorded data and the strains reconstructed from the neural network output. This error takes on two forms. The first is associated with differences in the peak strains. For example, in the data from January (Fig. 8.80), the peak strain for gage 11 is closer to the measured value than the corresponding values from gages 6 and 9. However, in the data from March (Fig. 8.81), the peak strain for gage 6 is closest and the values for gages 9 and 11 are off. In this data, the difference for gage 11 is approximately one-half of that from gage 9. The explanation lies in the data used as input for the neural network. By only using the data from one strain gage, it is not possible to determine where in the North-South direction the vehicle crosses the bridge decks. Therefore, a vehicle crossing close to the curb on the South side of decks will generate larger strains in gage 6 and smaller strains in gage 9 than the same vehicle traveling more towards the center of the roadway. This could be addressed in the neural network with the addition of a second camera capable of capturing the exact position of the vehicles on the bridge decks. Another possibility is to examine the distribution of peak strains recorded by the 16 sensors. For now, this is probably the best method as it would allow for making more refined analyses on the existing data.

The second source of error is associated with the manner in which the orginal time histories have been normalized (resampled to 200 points) prior to the application of Principal Component Analysis for feature reduction. As explained in Section 8.4.2, the resampled time history starts when the front axle first begins to cross the bridge decks and ends when the rear axles pass off of the decks. Small errors in locating these points in the strain data causes shifts between the recorded and predicted strains (channel 9 strain in Fig. 8.84).

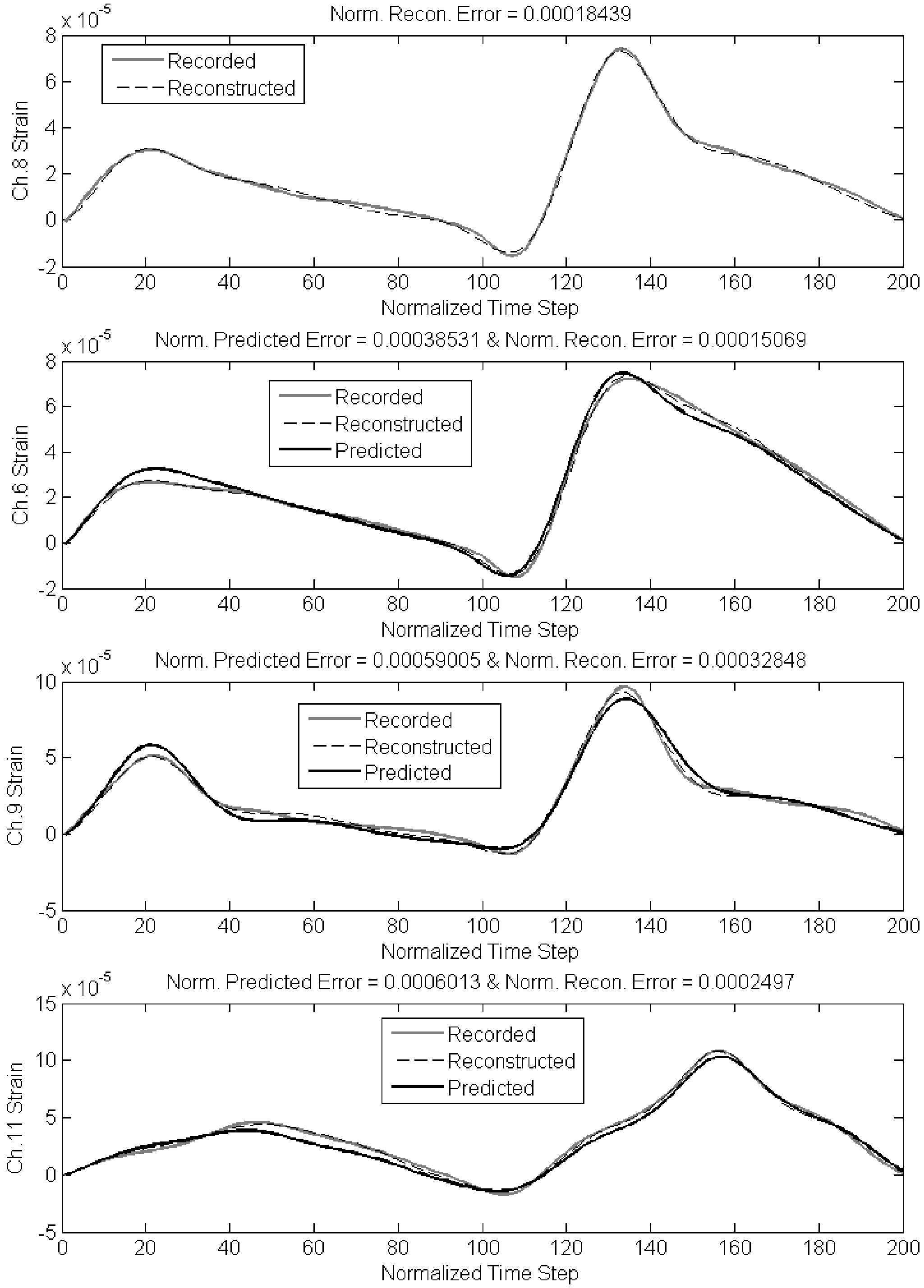
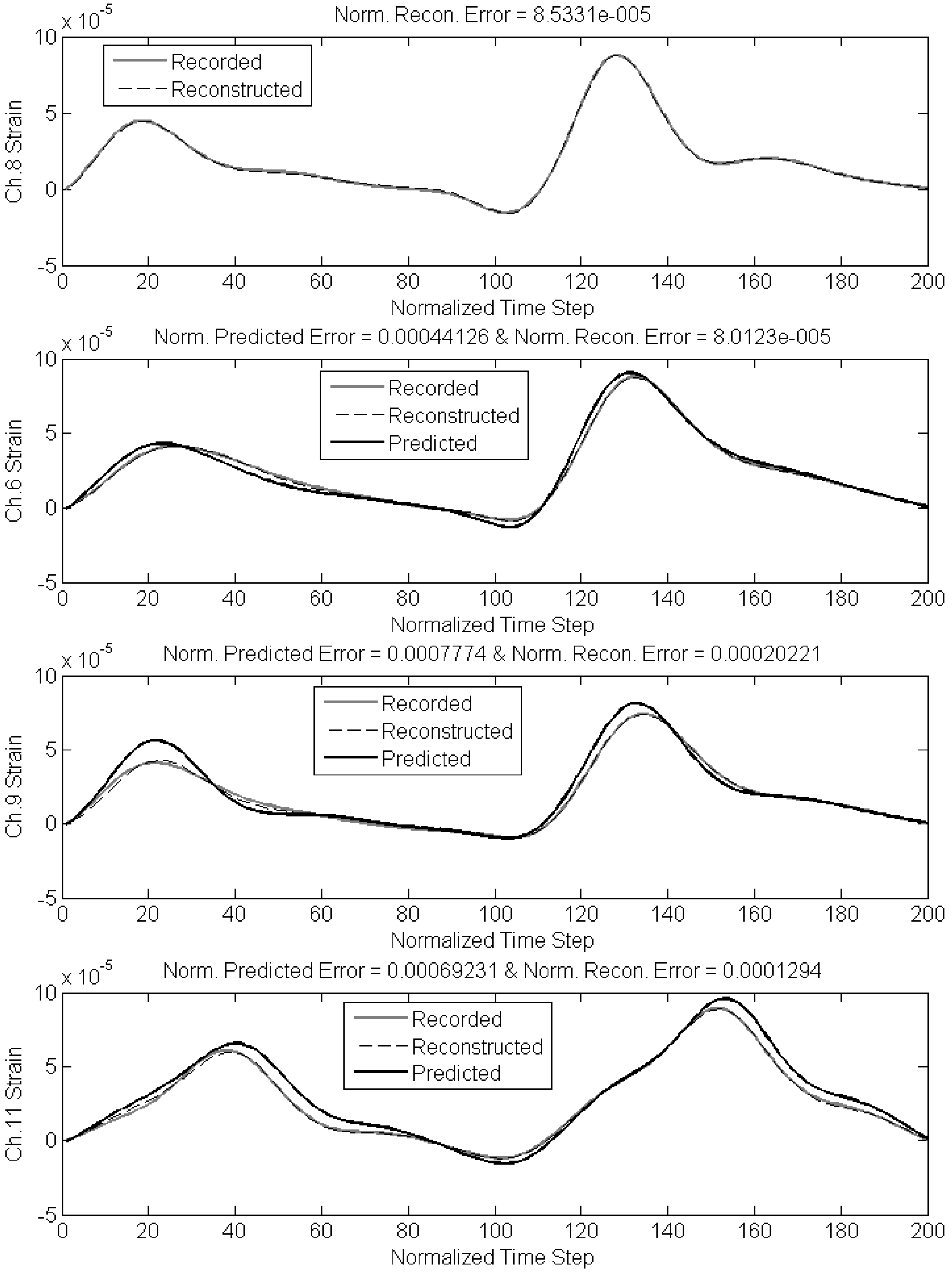


Figure 8.78: Strain Time Histories Associated with the Median Normalized Error for the Sorted January Data (sorted using Normalized Errors for Channel 6 data)


Norm. Recon. Error = 8.5331e-005

Norm. Predicted Error = 0.00044126 & Norm. Recon. Error = 8.0123e-005

Norm. Predicted Error = 0.0007774 & Norm. Recon. Error = 0.00020221

Norm. Predicted Error = 0.00069231 & Norm. Recon. Error = 0.0001294

Figure 8.79: Strain Time Histories Associated with the Median Normalized Error for the Sorted March Data (sorted using Normalized Errors for Channel 6 data)

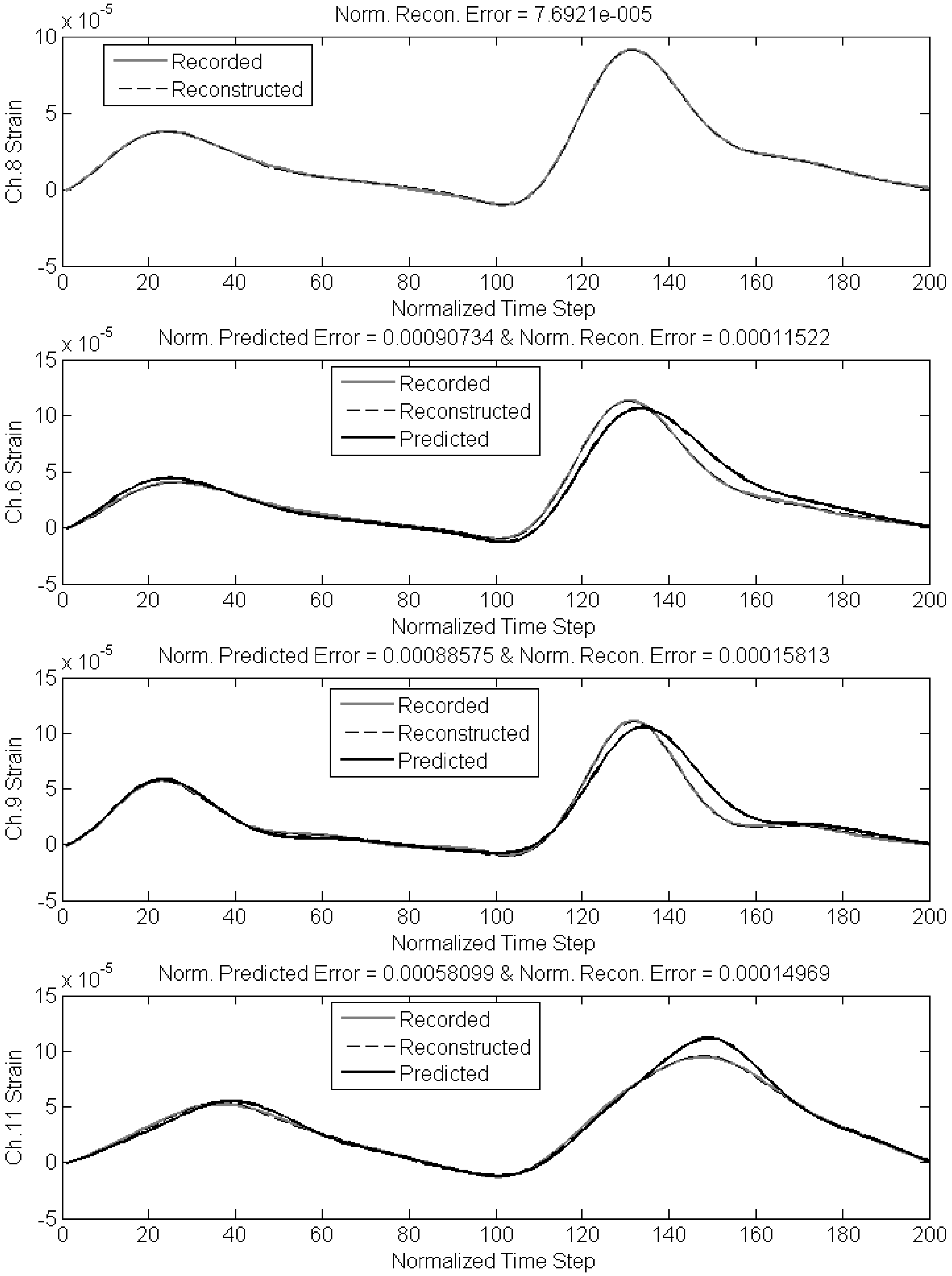


Figure 8.80: Strain Time Histories Associated with the Median Normalized Error for the Sorted May Data (sorted using Normalized Errors for Channel 6 data)

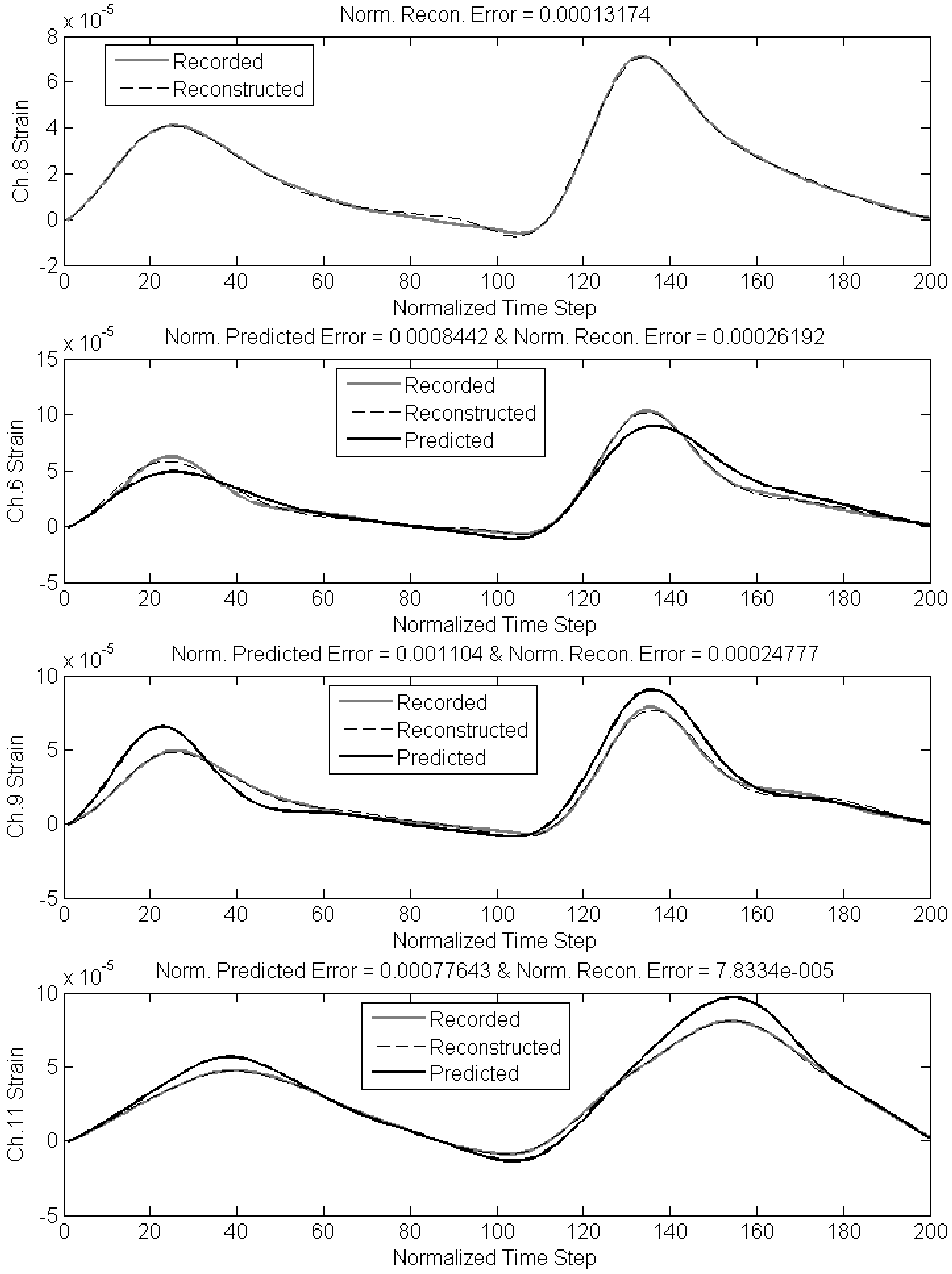


Figure 8.81: Strain Time Histories Associated with the Median Normalized Error for the Sorted August Data (sorted using Normalized Errors for Channel 6 data)

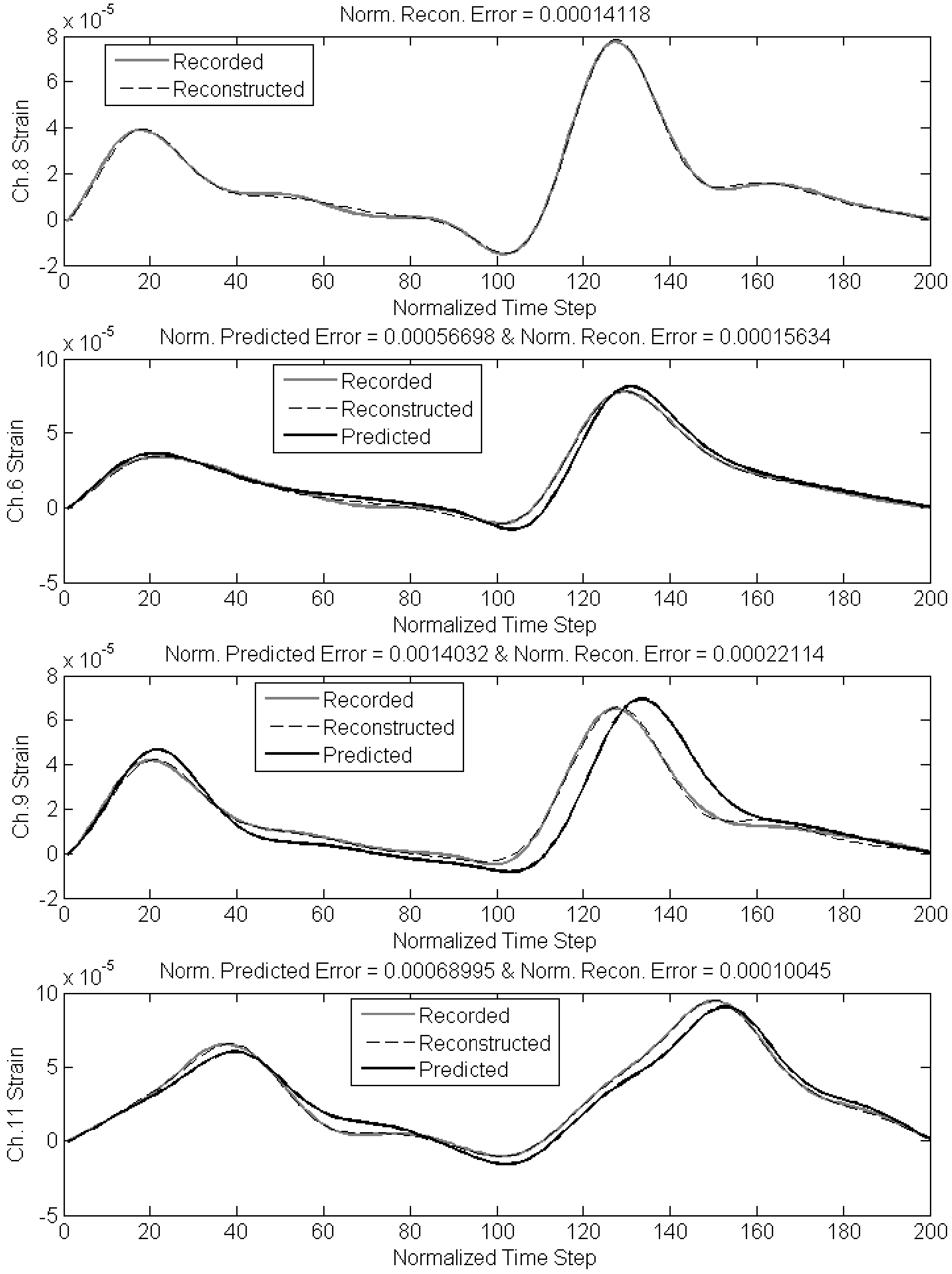


Figure 8.82: Strain Time Histories Associated with the Median Normalized Error for the Sorted November Data (sorted using Normalized Errors for Channel 6 data)

8.4.5.4 Reconstructed Strain Time Histories for Outliers in the Normalized Errors

To visualize the differences between the recorded strains and those reconstructed from the original PCA transformation and from the neural network, four outliers were selected from the channel 6 normalized errors (Fig. 8.85). The corresponding errors from gages 9 and 11 are also indicated in this figure. For each of these outliers (Fig's. 8.86 – 8.89), the original recorded data from the four gages is shown (solid gray line) along with time histories reconstructed from the PC's extracted in the original PCA transformation (black dashed line). For gages 6, 9, and 11, the reconstructed time histories from the neural network output are shown with the solid black line.

From visual inspection of the data in Figure 8.86, the errors indicate problems with the original PCA transformation, which is due to new features not present in the training data. In this case, the new features are due to the added tail at the end of the time history. This tail is caused by the misidentification of the time instant in which the rear axle moves off of the bridge decks. Comparison of the original and reconstructed data shows similar problems with the data recorded by sensors 6 and 9. Because of the problems with feature extraction on the neural network input, all of the outputs are incorrect. Similar problems are visible in the data in Figure 8.87. This time, the input data is correct, but the normalization process for the target data failed hence the large errors between the measured and predicted strains. For the data in Figures 8.88 and 8.89, there is a fair level of error in the reconstructed data, but primarily the problems are with the histories output from the neural networks. It appears the data normalization process was successfully applied to all of the data; therefore, the error should be attributed to a change in the system response.

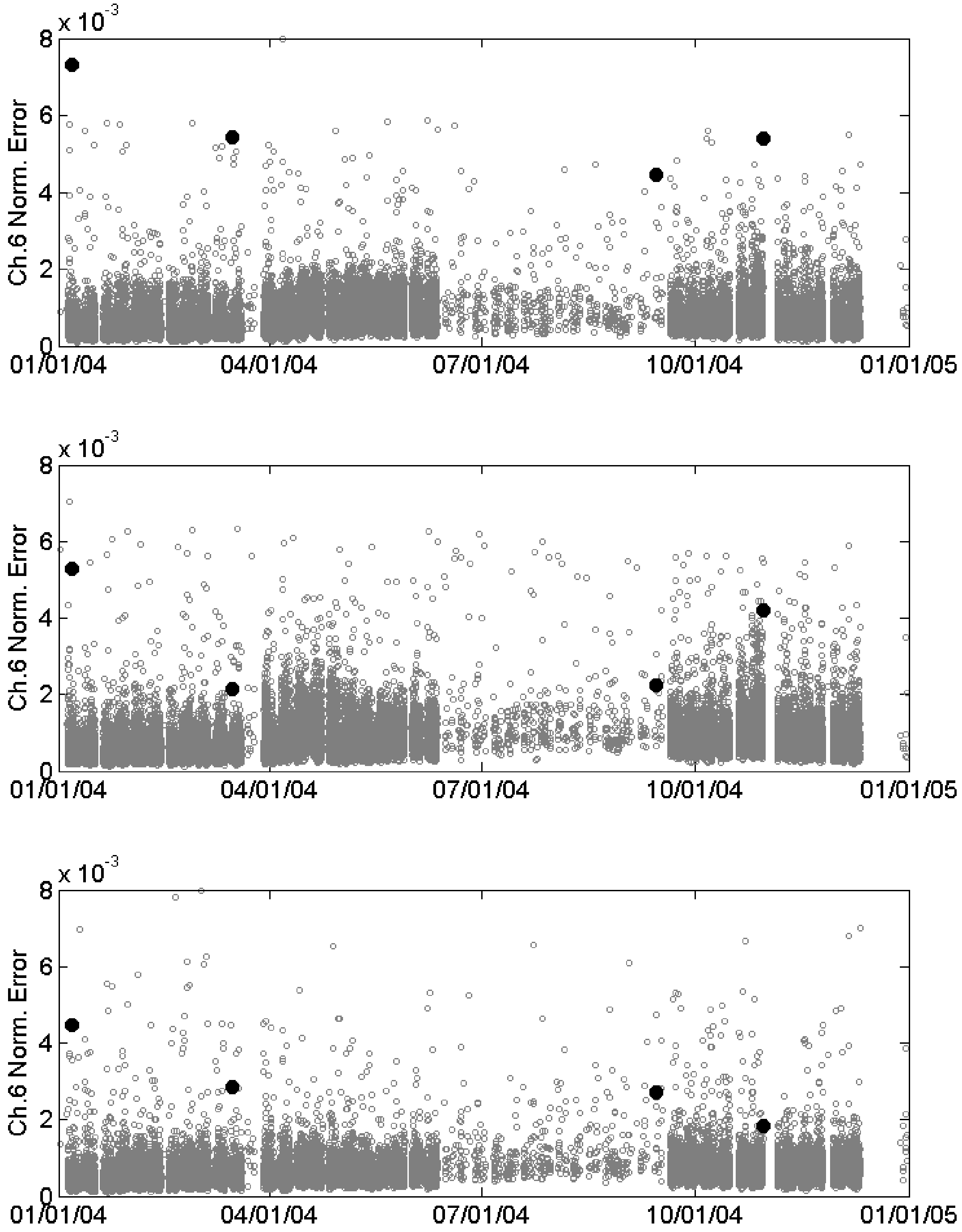


Figure 8.83: Randomly Selected Outliers from Strain Gage 6 Data with Corresponding Normalized Errors from Strain Gages 9 and 11.

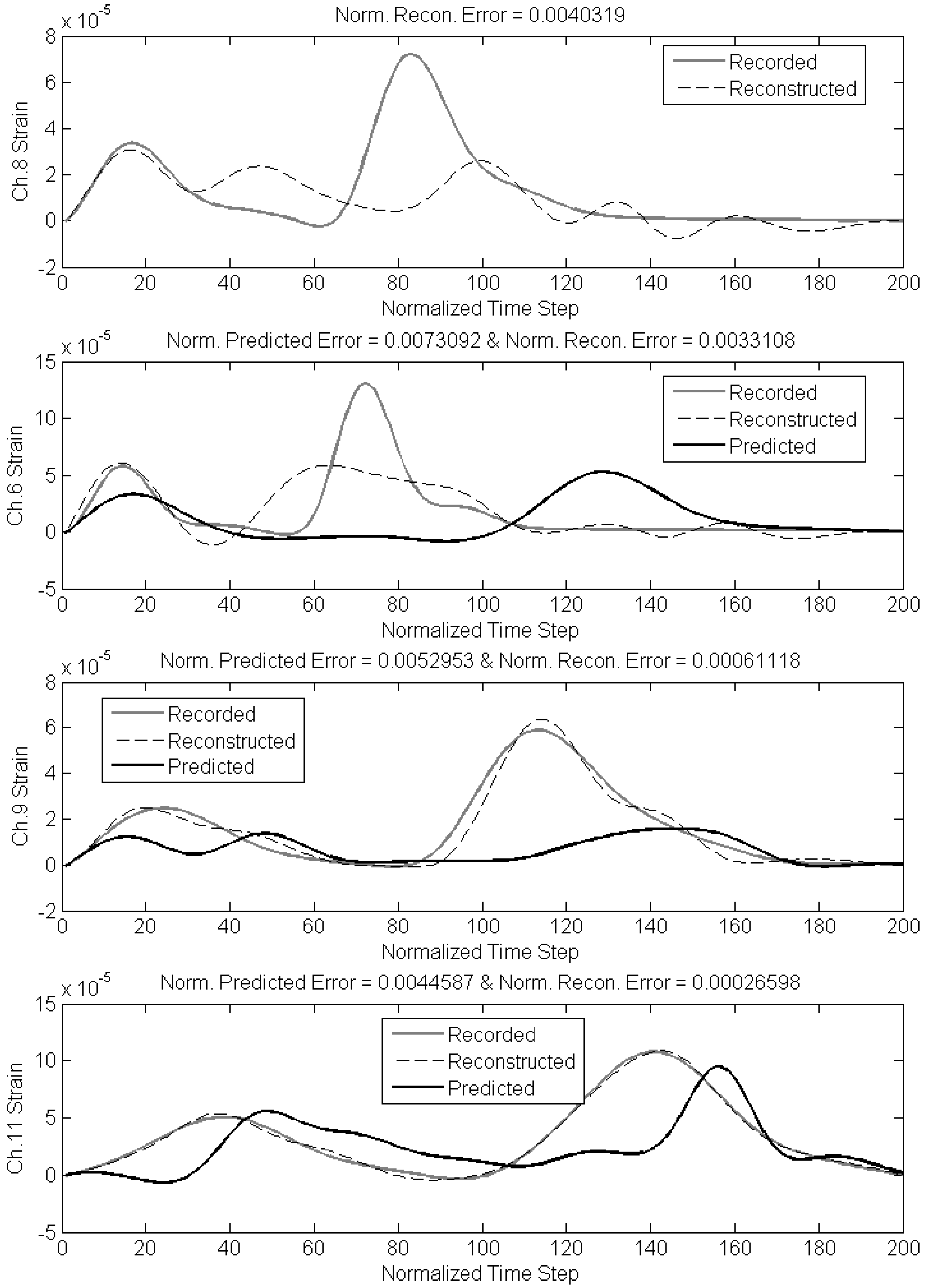
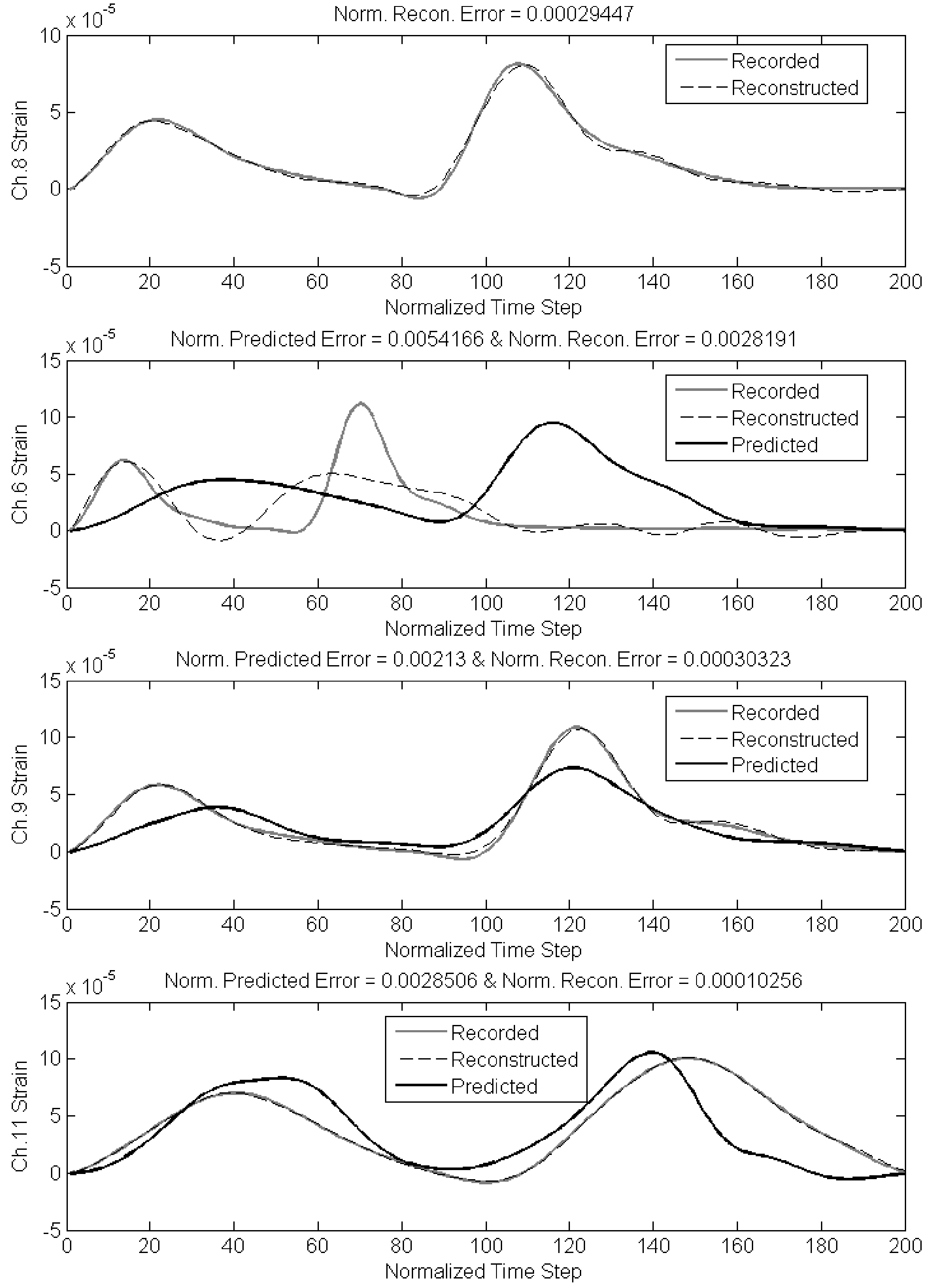


Figure 8.84: Strain Time Histories Associated with the First Outlier

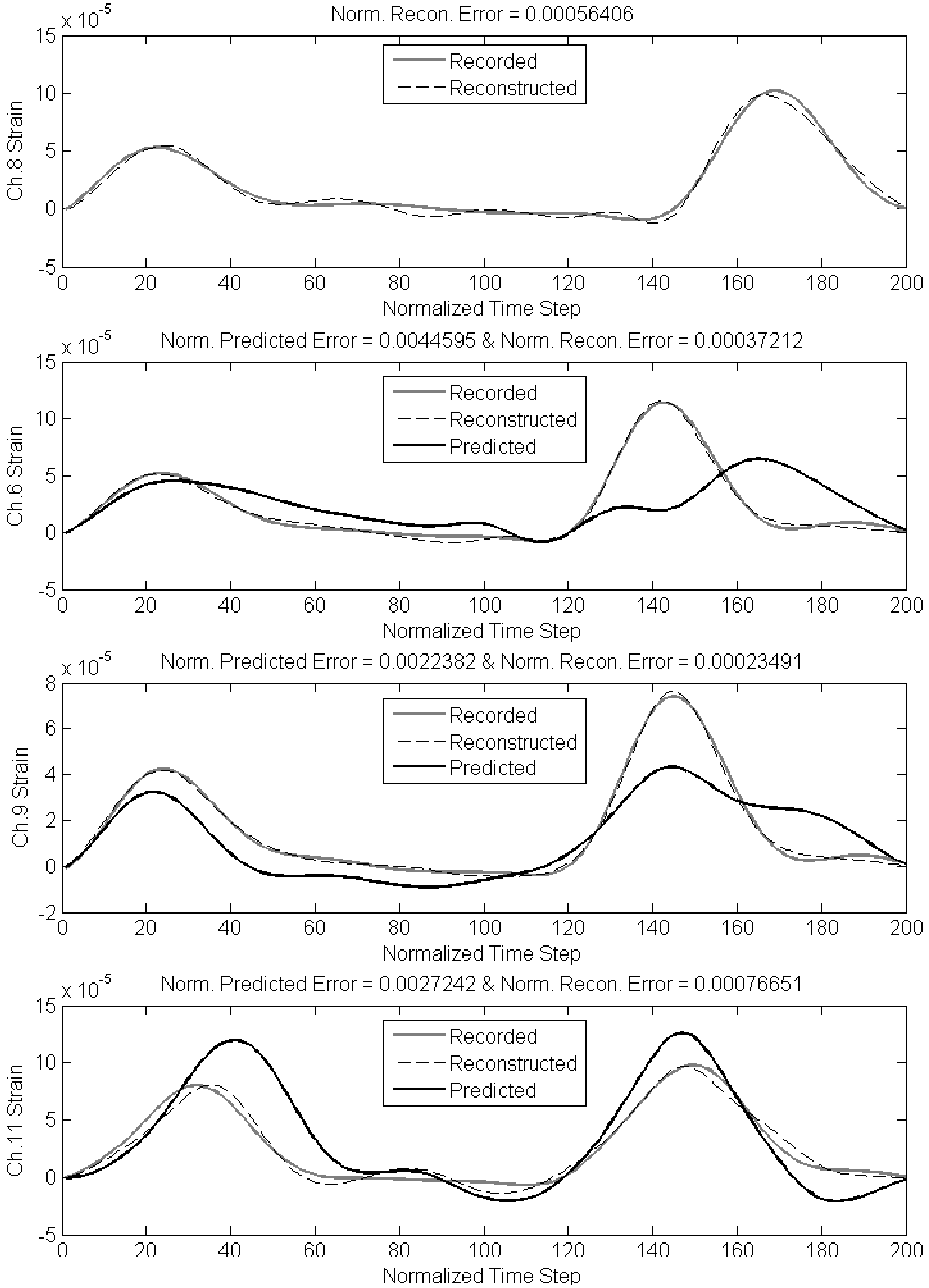

Norm. Recon. Error = 0.00029447

Norm. Predicted Error = 0.0054166 & Norm. Recon. Error = 0.0028191

Norm. Predicted Error = 0.00213 & Norm. Recon. Error = 0.00030323

Norm. Predicted Error = 0.0028506 & Norm. Recon. Error = 0.00010256

Figure 8.85: Strain Time Histories Associated with the Second Outlier



Norm. Recon. Error = 0.00056406

Norm. Predicted Error = 0.0044595 & Norm. Recon. Error = 0.00037212

Norm. Predicted Error = 0.0022382 & Norm. Recon. Error = 0.00023491

Norm. Predicted Error = 0.0027242 & Norm. Recon. Error = 0.00076651

Figure 8.86: Strain Time Histories Associated with the Third Outlier

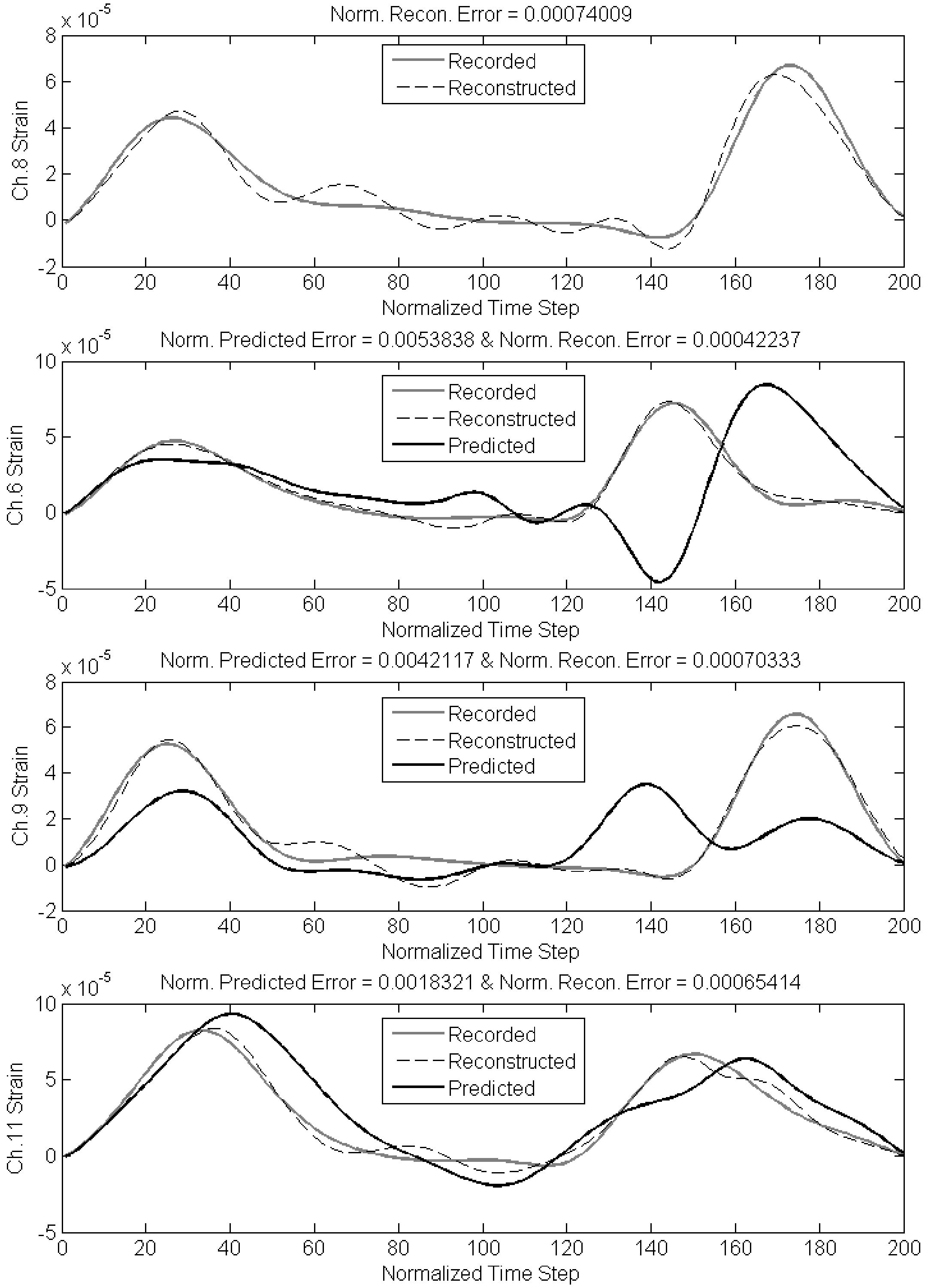


Figure 8.87: Strain Time Histories Associated with the Fourth Outlier

8.5 Summary

Speeds and then wheelbases were calculated using the recorded strain time histories in the labeled data subset from the composite bridge-decks. By using the labeled data, it was possible to focus on vehicles with constant wheelbases, which provided an error check in the calculated axle spacing by plotting histograms of the wheelbases for the different vehicle types. Next, peak axle strains recorded over the period of time the monitoring system was operational were considered. The numbers and distributions of vehicles crossing over the bridge decks were examined. This information allows for accurately modeling the probability distributions of traffic loads, which without this more detailed load data is typically modeled with random distributions.

The same predictive neural networks used in Chapter 7 for damage detection on the finite element model data were employed on the recorded data from the composite decks to look for changes in the response of the bridge-decks. This was done by considering only 2-axle traffic with peak strains typical of campus buses. The data from strain gage 8 was used as input, and three neural networks were constructed to predict strains in neighboring gages 6, 9, and 11. These three neural networks were trained using 1000 patterns recorded in December 2003 and tested on all similar records recorded during 2004 (over 39,000 time histories). Principal Component Analysis was again used for feature reduction and cut the number of features input and output by the neural network from 2000 (each time step being a feature) to 9 or fewer. In addition to comparing the error between the measured and predicted (from neural network) strains, a second comparison was done with strain time histories reconstructed using the inverse Principal Component Analysis transformation applied to the extracted PC’s. In general, there was very good correlation between the measured strains and those predicted by the neural networks and reconstructed using the inverse PCA transformation. One notable result was an increase in both errors during the summer months (where traffic patterns are quite different compared to the rest of the year).

9 UCSD Bridge Testbed for Health Monitoring Technologies

Borrowing on lessons learned from monitoring of the UCSD composite bridge-deck panels (Chapter 3), a second bridge testbed is established. It is envisioned that this testbed (the Voigt Drive / Interstate-5 Overcrossing) will become a live laboratory for structural health monitoring technologies, and will be part of the Powell Laboratory activities as a convenient field environment. This two-lane bridge (measuring nearly 300 feet in length) is a skewed, single-column bent, 4-span, reinforced concrete box girder structure (Figs. 9.1-9.3), located just East of the UCSD campus. The construction style is typical of a large number of Caltrans' overpasses. This particular bridge was chosen due to its proximity to the campus, which was and still is an important consideration for developing and employing a hereto untested system. As usual, difficulties were encountered during the initial installation and required frequent trips back-and-forth between the bridge and the campus laboratories. Preliminary data was finally collected for demonstration purposes, as discussed herein.

An additional benefit with this bridge is the access provided into two of the bridge cells through manholes located in each of the spans (Fig. 9.2). This access into the interior of the bridge made it possible to install all of the sensors and data acquisition hardware within the bridge, thereby simplifying the installation and reducing the necessary permissions required for installation as the internal sensor network posed no risk to traffic on or below the overcrossing. With conventional monitoring systems, where sensors are installed on the exterior of the structure, great care must be taken to ensure none of the hardware (sensors, cabling …) detaches from the bridge and ends up in the traffic below.

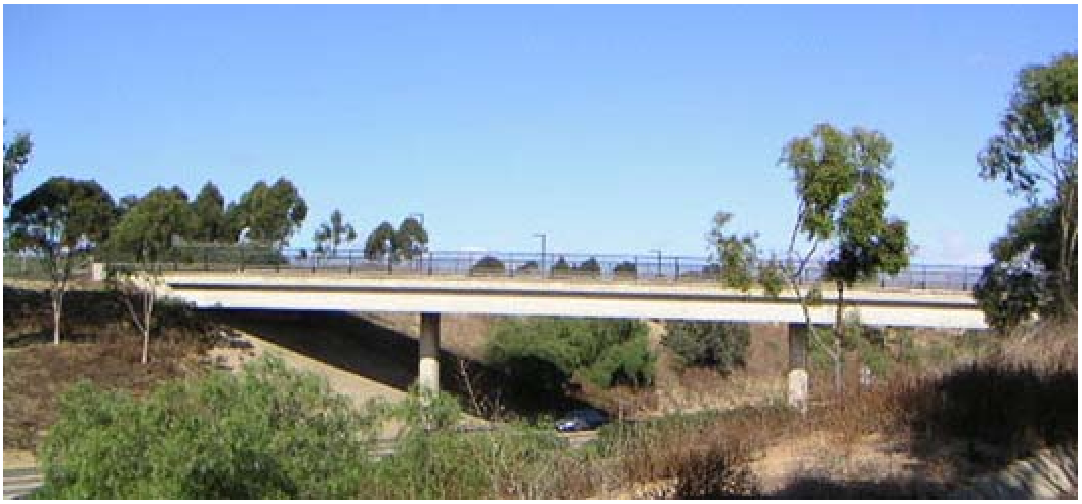


Figure 9.1a: Elevation View of Voigt Drive / I-5 Overcrossing Taken from South-West side of the Bridge

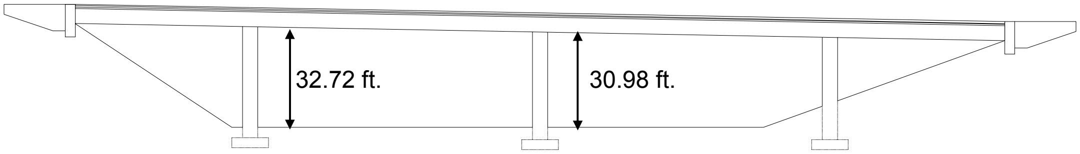


Figure 9.1b: Elevation View of Voigt Drive / I-5 Overcrossing

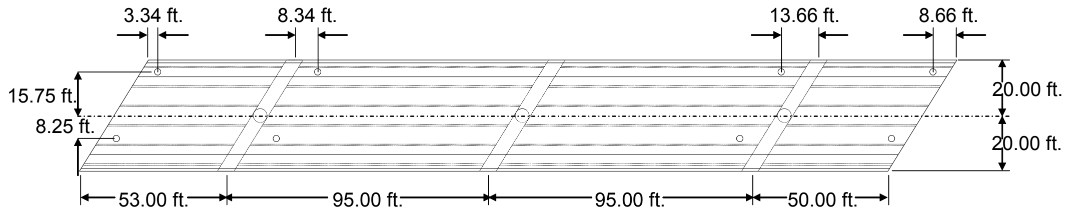


Figure 9.2: Plan View of Voigt Drive / I-5 Overcrossing

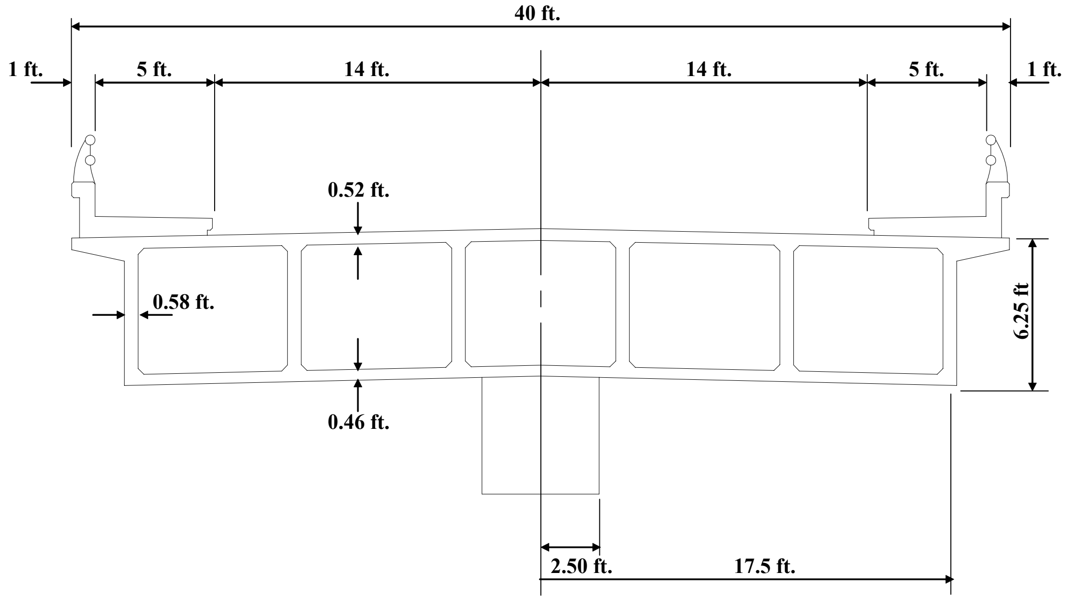


Figure 9.3: Cross Section of Voigt Drive / I-5 Overcrossing

9.1 Structural Health Monitoring System

The proposed testbed for the I-5 overcrossing calls for the installation of an array of sensors inside and around the bridge when completed. Installed sensor types will include accelerometers, fiber optic strain gages, electrical resistance strain gages, thermocouples, thermisters, humidity probes, magnetometers, etc… The dynamic sensors will be utilized to capture the bridge's vertical, horizontal, and torsional dynamic response while the environmental sensors will be used to correlate dynamic response characteristics with environmental factors (such as temperature and relative humidity) and to monitor sensor response with temperature [Worden et al., 2005]. In addition, digital cameras are to be installed to record traffic passing over as well as under the bridge. The overall structural monitoring system architecture is detailed in Figure 9.4.

9.1.1 Initial Dynamic Monitoring Configuration

The initial installation described in this paper consists of a relatively simple configuration involving a total of 29 dynamic sensors. As only a limited number of sensors were to be installed in this phase, it was decided to focus on a single bridge cell, in this case the Northern most cell (Fig. 9.5). This one was chosen for its convenient access through manholes located in the sidewalk on the North side of the bridge (Fig. 9.2). The setup is composed of 20 capacitive accelerometers, spaced 15-feet apart, to measure the vertical response along the length of the bridge (Figs 9.6-9.8). Eight of these sensors are PCB Model 3801 ±3g accelerometers and the remaining twelve are Crossbow CXL01LF1 ±1g sensors. In addition, 8 TML general purpose foil strain gages (Fig.9.9) were installed at the bottom of the interior girder (Fig. 9.10) to monitor the local response of the bridge at the middle of each span as well as at the quarter points of the middle two spans. One thermocouple was attached to the interior girder of the Western span.

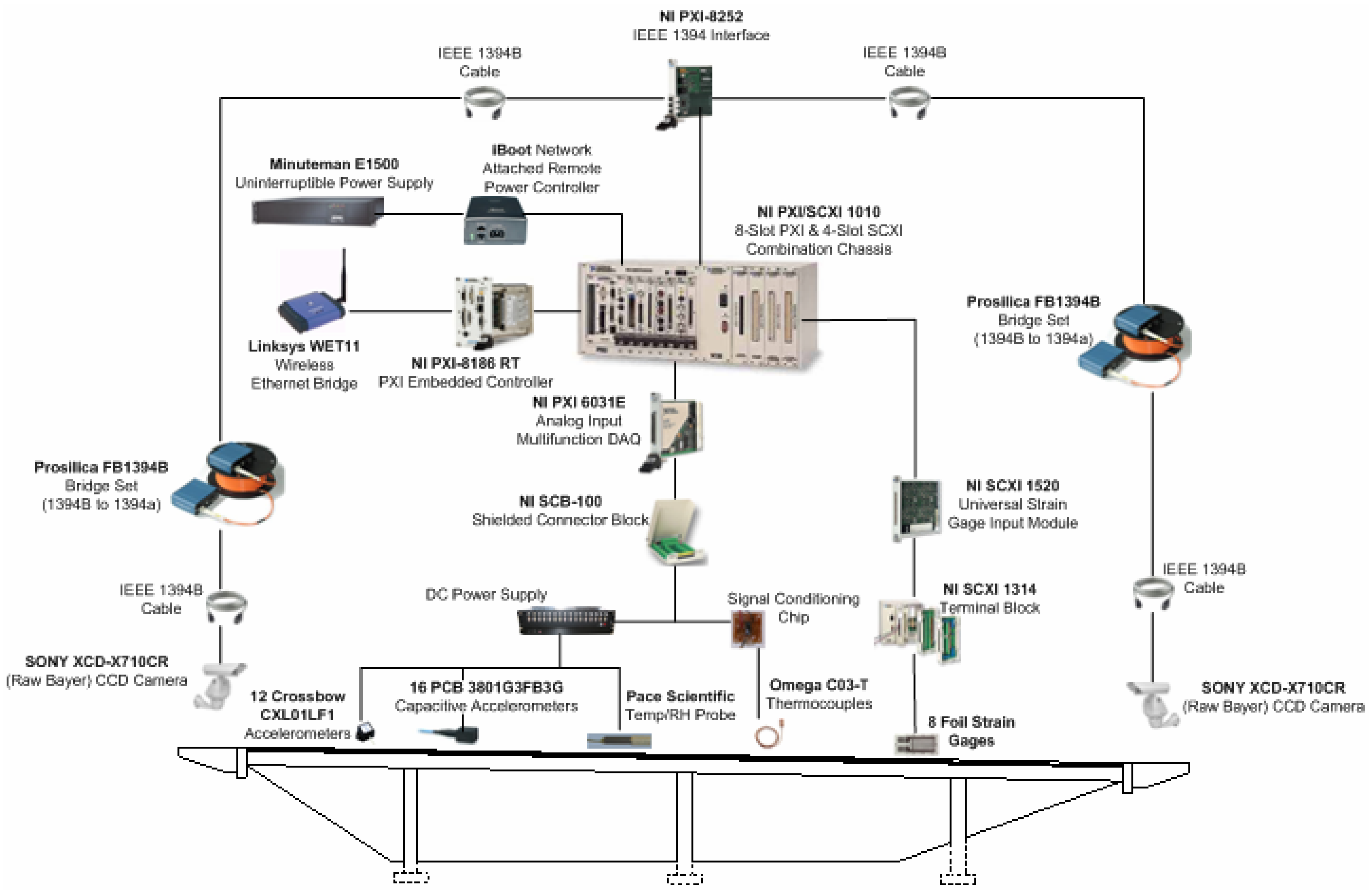


Figure 9.4: Structural Monitoring System Architecture

Figure 9.5: Interior of Northern-Most Cell of Voigt Bridge

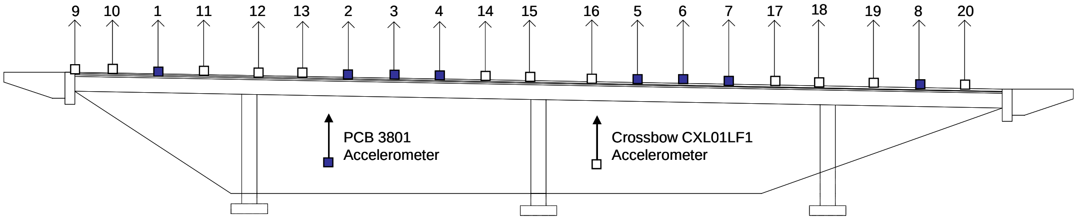


Figure 9.6: Location of 20 Capacitive Accelerometers

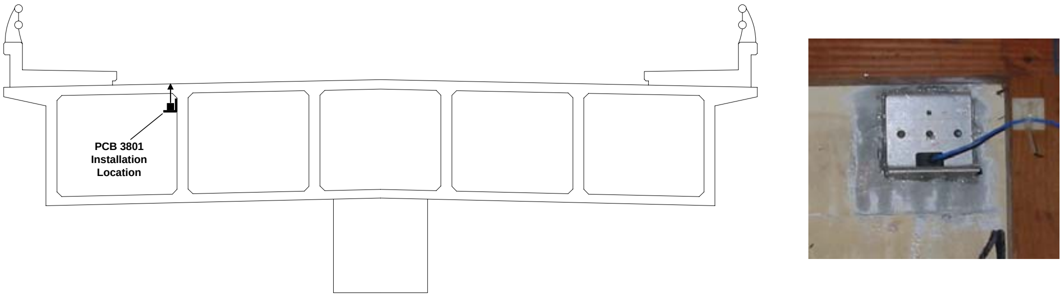


Figure 9.7: Installation details for PCB Model 3801 Capacitive Accelerometers

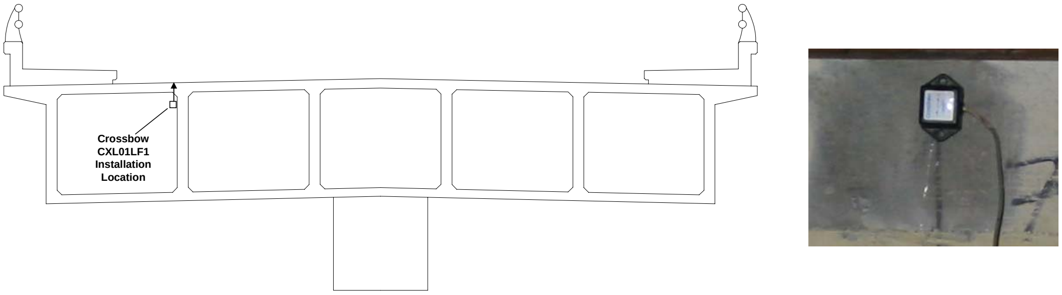


Figure 9.8: Installation details for Crossbow Model CXL01LF1 Capacitive Accelerometers

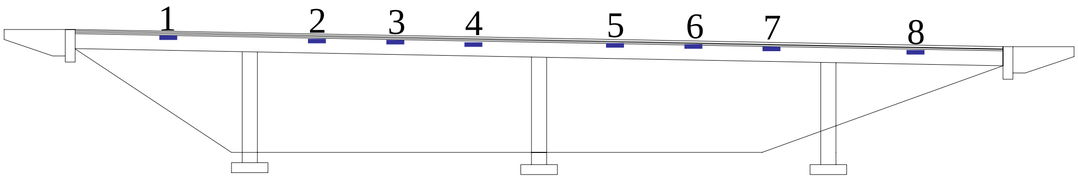


Figure 9.9: Locations of Electrical Resistance Strain Gages

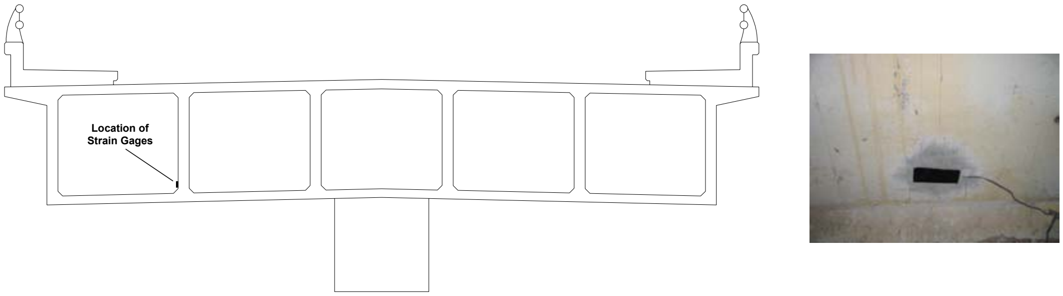


Figure 9.10: Location of Strain Gages, Installed on the Interior Girder (left) and Strain Gage 1 (right) Installed in Western-most Span of the Bridge

9.1.2 Environmental Monitoring

For measuring ambient air temperature, relative humidity, and temperature of the bridge elements, an array of environmental sensors are to be installed inside the Northern-most bridge-cell. To record the temperature of the structural elements of the bridge (i.e. superstructure girders), thermocouples are being attached to the interior girder. For this application, Omega “Cement-On” model CO-T’s (Fig. 9.11a) are used. These are an economical fast response thermocouple for surface temperature measurements with a nominal operational temperature range of -190 to 205°C (-310 to 401°F). Temperature measurements will be important for assessing the long-term performance of the sensors and determine whether further temperature compensation will be required.

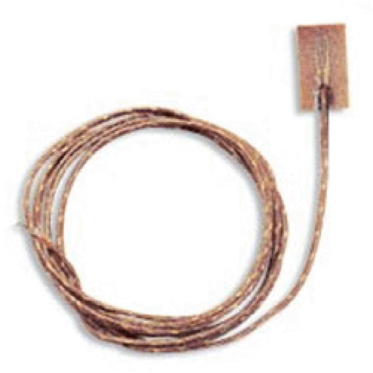


Figure 9.11a: Omega model CO-T

“Cement-On” Thermocouple

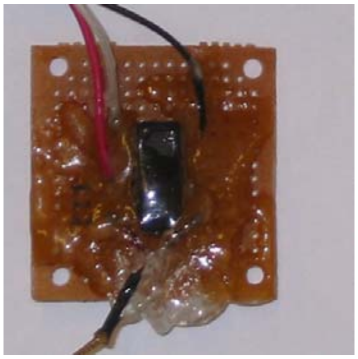


Figure 9.11b: Analog Devices

AD595AQ Thermocouple Amplifier

To measure the voltage generated by the thermocouple (known as the Seebeck voltage), it is not possible to connect the thermocouple directly to the measurement system, as connecting the thermocouple wires to the measurement system creates additional thermoelectric circuits. Therefore it is necessary to utilize signal conditioning for each thermocouple. In this application, an Analog Device AD595AQ monolithic thermocouple amplifier with cold junction compensation is soldered to a circuit board and then coated with epoxy and silicon (Fig. 9.11b) for weatherproofing. In addition to supplying the required signal conditioning, these units can also provide amplification, raising the output voltage from a few mili-volts to ± 10 volts, allowing for long cable runs. For bridge monitoring, it is advantageous to place the conditioning units near the sensors to allow for making the majority of the cable runs with conventional shielded cables, rather than thermocouple wire, thereby reducing cabling costs by over 50%.

To measure the relative humidity and air temperature, dual purpose sensors like the Pace Scientific Temperature/RH sensors (Fig. 9.12) are readily available. This is a precision relative humidity and temperature probe, which outputs an analog output easily measured with standard data acquisition hardware. Another advantage is it does not require any signal conditioning, just a 5 vdc excitation at 2 ma.

The humidity output from the sensor is typically 0.8 vdc at 0% Relative Humidity (RH) and 3.9 vdc at 100% RH. To convert from measured voltage to percent relative humidity, linear scaling is performed with ± 2% RH accuracy from 0 – 95% RH. Temperature compensation for the RH measurements is similarly performed.

Figure 9.12: Pace Scientific Temperature / Relative Humidity Probe

9.2 Data Acquisition

Data from the sensors will be collected on the bridge, using a local data acquisition system, housed within the Northwest corner of the bridge. It is important that this equipment be located within the wireless Internet cloud that will be created at the bridge site. Using a Linksys Wet11B wireless Ethernet bridge with an external antenna mounted on the bridge guard rail, the campus wireless connection is converted to a standard wired Internet connection, which can then passed to the data acquisition system. Once the data is acquired, it will be streamed using the wireless Internet connection back to the SERF building (Fig. 9.14) for further analysis and archiving.

The current system (Fig. 9.13) is built around a National Instruments PXI/SCXI combination chassis with an embedded PXI-8186 real-time controller (with dual-boot capabilities – Windows XP and a real time operating system). For standalone bridge monitoring operation, the data acquisition, local data processing, and data transmission codes are embedded on the PXI controller so they start automatically when the system boots, thereby requiring no human interaction. A NI PXI-6031E (16-bit A/D, 32 differential inputs, 100 kS/s maximum multiplexed sampling rate) board is currently being used for acquiring signals from the accelerometers and environmental sensors, while a SCXI-1520 8-channel universal strain gauge input module is utilized for the eight strain gages' conditioning and acquisition. With the current data acquisition boards, the system is easily expanded to support over 250 channels. Further, if additional sensors types are needed, which have their own requirements (signal conditioning, higher sampling rates) this too is easily managed. For example, to study vehicle/structure interaction, piezoelectric accelerometers are better suited than the capacitive ones currently installed in the bridge. In this case, the ICP signal conditioning can be handled using 8-channel PXI-4472B dynamic signal conditioning modules.

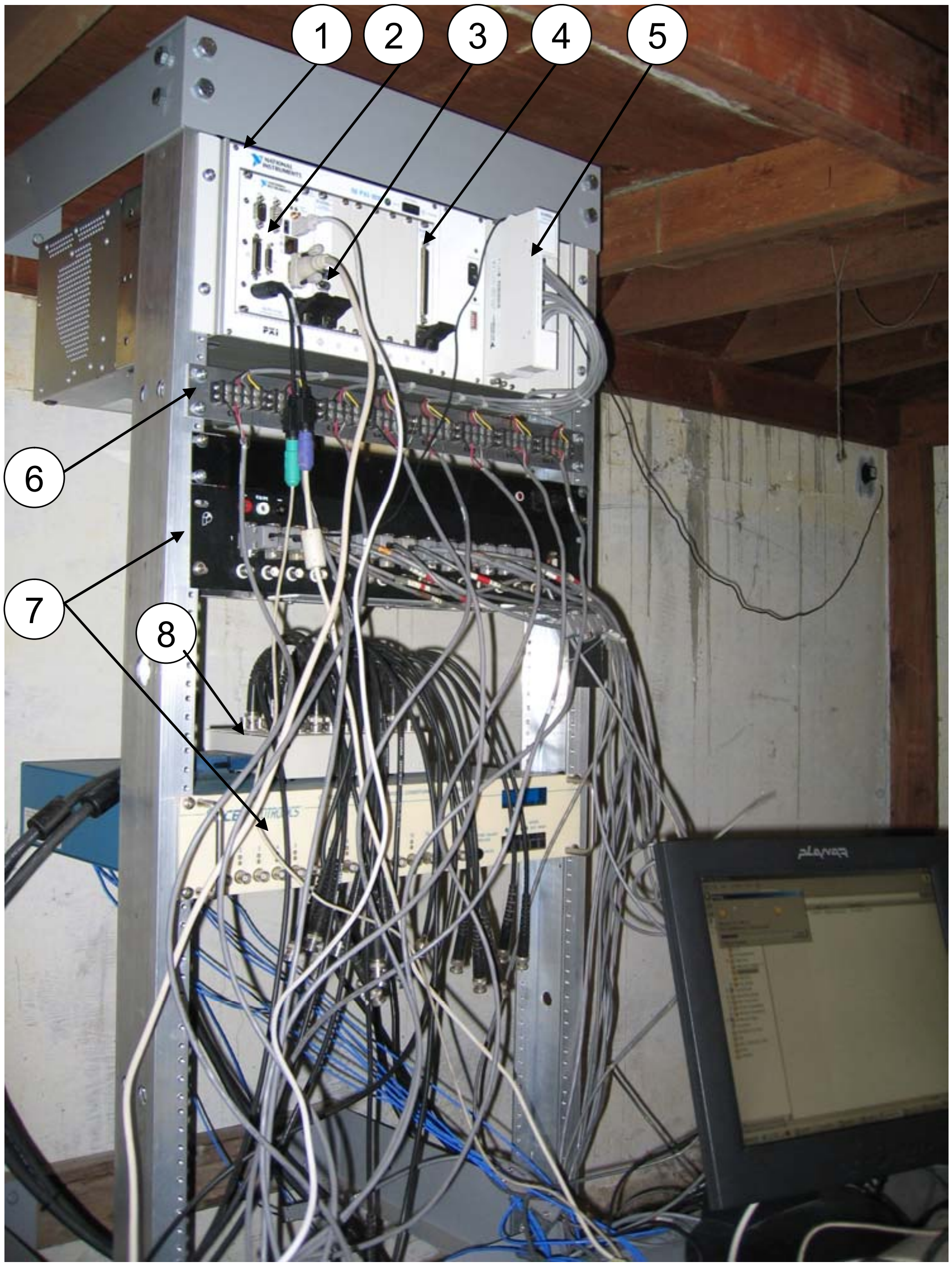


Figure 9.13: Current Data/Image Acquisition System: 1. PXI/SCXI Combination Chassis, 2. PXI-8186 Controller, 3. IEEE 1394 Interface, 4. PXI-6031E Data Acquisition Board, 5. SCXI-1520 Strain Gage Input Module, 6. Strain Gage Terminal Strip, 7. DC Power Supplies for Capacitive Accelerometers, 8. SCB-100 Connector Block.

9.3 Image Acquisition

Two Sony XCD-X710CR IEEE 1394 digital color cameras are being installed on light posts at either end of the bridge (Fig. 9.15). These cameras will be positioned in a manner in which they are capable of monitoring traffic crossing over as well as under the bridge. This allows for correlating the bridge response with the responsible vehicles. The Sony cameras are capable of 800,000 pixel resolution with a maximum 30 frames per second sample-rate. These cameras are connected to the data acquisition computer via fiber optic cables. Fiber-optic technologies were utilized to allow for longer cable runs (upwards of one-mile) than are possible with typical analog and digital cameras. Using National Instruments' LabVIEW software, images are to be acquired from each camera at 5 frames-per-second and time stamped using the data acquisition clock. By building the image acquisition directly into the data acquisition program, highly accurate synchronization between the dynamic sensors and cameras is achieved (much better than those experienced with the network camera installed in the composite bridge-deck testbed, described in Chapter 3). The XCD-X710CR cameras use a CCD with a color mask (commonly known as a "Bayer filter"), which allows the camera to output a "raw color" image as an 8 or 10-bit monochrome image.

To best utilize the limited bandwidth available, the output from the cameras will be streamed to the archiving database in the original monochrome (Bayer) format. Once received by the archiving/analysis computers, the Bayer images (Fig. 9.16a) are decoded to produce uncompressed color images (Fig. 9.16b), which are then passed to the image analysis algorithms. The major advantage is that Bayer-encoded images are 1/3 the size of the same color image. Feature extraction can then be performed on the archived images to extract useful information related to vehicle properties of traffic passing over/under the bridge.

The selection and final placement of the employed cameras were done with the assistance of Professor Mohan Trivedi and Dr. Tarak Gandhi. The simultaneous data / image acquisition and image compression were created in collaboration with Jeff Ploetner.

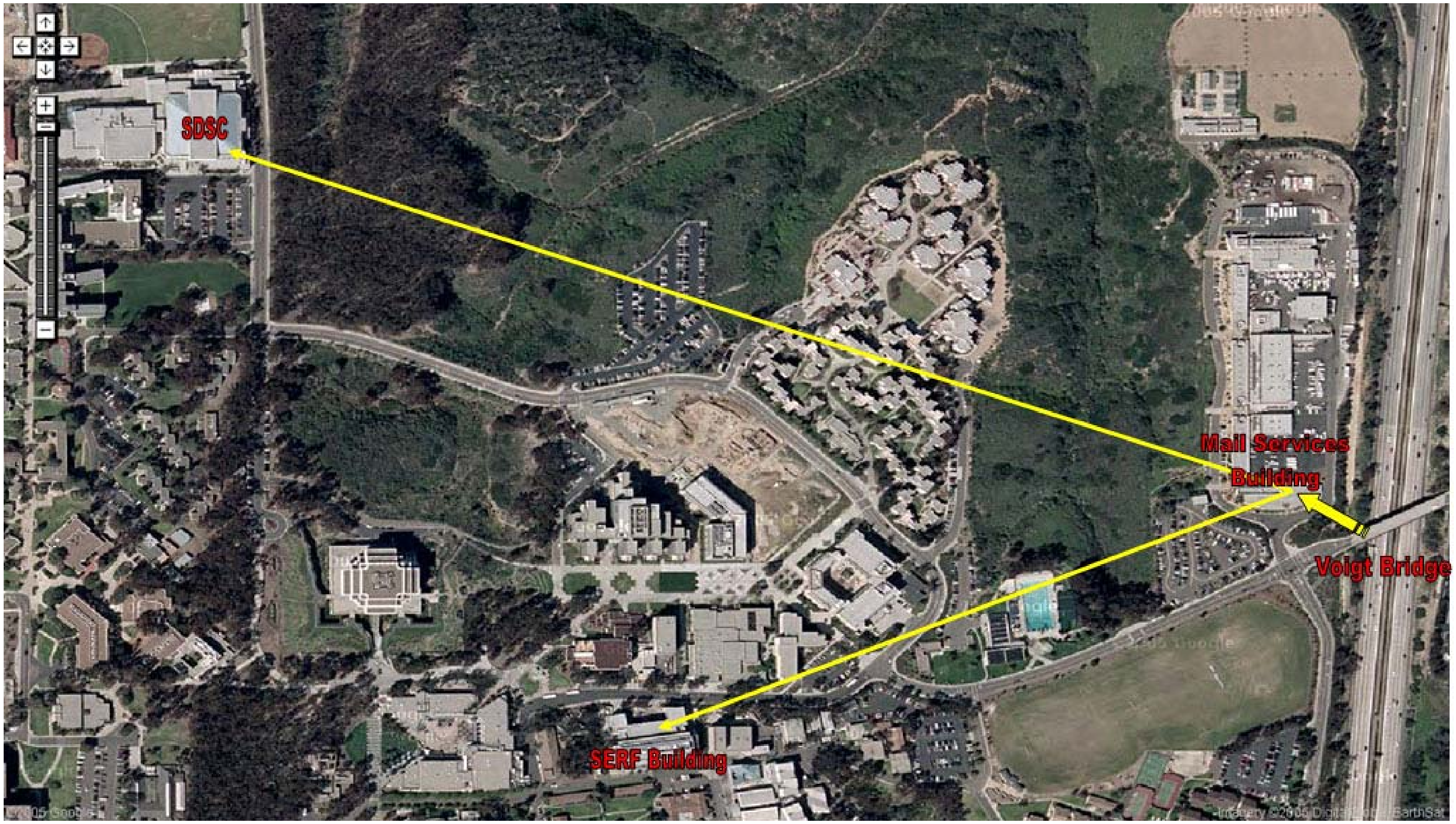


Figure 9.14: Data Streaming Path from Voigt Bridge To SERF Building and San Diego Supercomputer Center

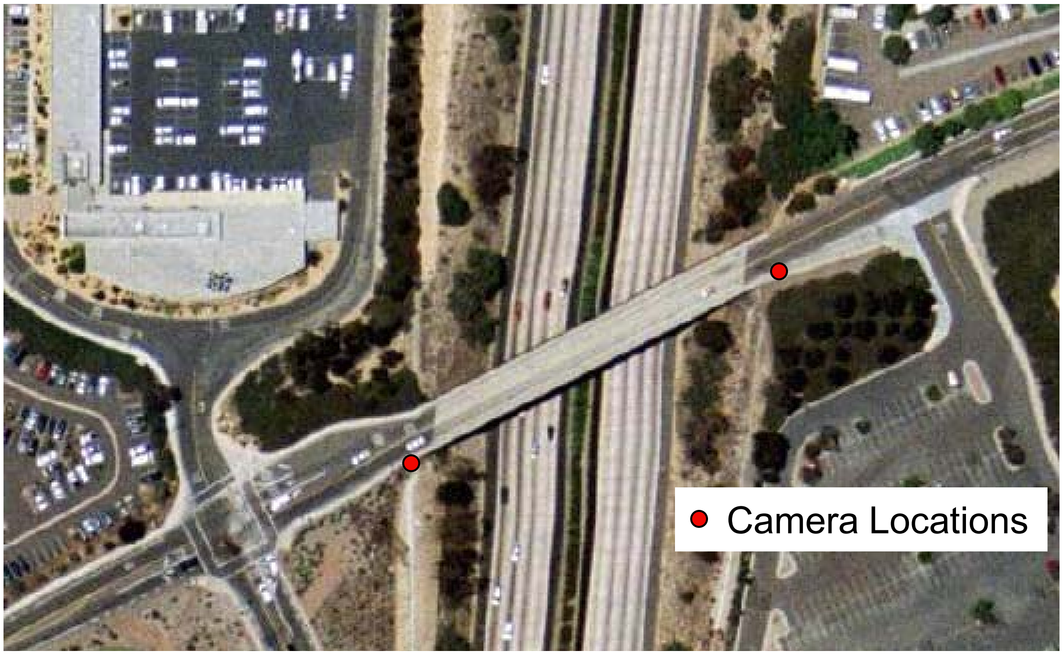


Figure 9.15: Locations of IEEE 1394 Color Cameras

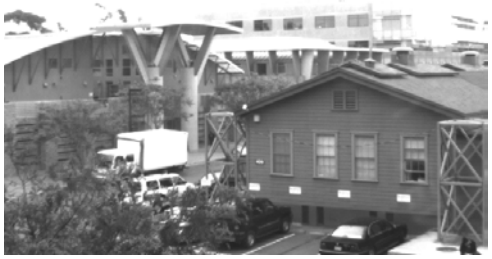


Figure 9.16a: Original Bayer-encoded Image

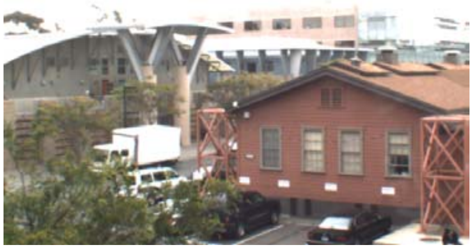


Figure 9.16b: Decoded Color Image

9.4 Internet Connectivity

For streaming all of the data from the bridge, a wireless Internet cloud was created by utilizing an existing wired UCSD Internet connection located in the UCSD Mail Services building (Fig. 9.14). A strong Internet connection has been established on the bridge by installing a wireless router in Southeast corner of the Mail Services Building (approximately 60 meters from the bridge). A Linksys Wet11 wireless Ethernet bridge with external antenna is being set up to bring the wireless Internet connection to the data/image acquisition computer located inside the bridge.

9.5 Data Transmission

For the developing monitoring system, a data acquisition program has been developed to sample the analog signals from the accelerometers, environmental sensors, and strain gages at a rate of 1000 Hz. Once digitized, the signals are placed in a circular buffer. After a predetermined number of samples have been acquired in the buffer, the data is written to disk as a text file in ASCII format. A java-based data transmission program opens the file and streams the data (using TCP protocols) to the SERF building for archiving (Fig. 9.14). Upon confirmation of successful transmission, the data is deleted from the acquisition computer. Should the Internet connection temporarily go down, no data will be lost as the system will continue to acquire data and write it to disk on the local hard drive. Once the connection is restored, the streaming program automatically resumes sending the data. A Java-based loader program has been written to connect from campus to the field computer and read the streamed data, perform basic signal processing, and archive the data within a DB2 database.

Due to bandwidth limitations, it will be impossible to continuously stream all of the images acquired by both cameras. Performing image compression prior to transmitting the images was deemed unsuitable as it would worsen the quality of the results from the image processing algorithms, and local processing was also discarded due to the burden that would be placed on the data acquisition computer's processor. Therefore, it was decided to use a trigger, generated by the analog sensors, to tag images with vehicles crossing the bridge. Only these tagged images will be transmitted.

9.6 Remote Access

Remote access to the data acquisition computer is provided using Windows Remote Desktop Connection and National Instruments LabVIEW webserver. Through the webserver, small changes can be made to the data acquisition parameters (for example, changing sample rates, sensor calibration constants, buffer size …). To make changes to the data acquisition program, including overwriting the current version, Windows Remote Desktop Control will be utilized.

To access the data acquisition computer on the bridge, both of the previously mentioned programs require the system to be operating properly. Should the acquisition computer crash and access become unavailable, a Data Probe iBoot network attached remote power controller is installed with the PXI controller (Fig. 9.4). This device allows authorized users to cycle the power on the controller, rebooting and restoring the system. To ensure proper authorization, access is restricted to users with registered IP addresses.

9.7 Shakedown Test Performed on January 7, 2006

9.7.1 Test Setup

On January 7, 2006, a test was made with the twenty installed accelerometers (Fig. 9.6) and the two Bayer cameras. In this experiment, the cameras had not yet been permanently installed onto the light posts. Therefore, they were placed on tripods and placed on the sidewalk, near the data acquisition computer. The cameras were connected to directly to the data/image acquisition computer (through the open manhole in the sidewalk) using only firewire cables. As the cable runs for the cameras were under twenty feet, the optical repeaters to convert from firewire (IEEE-1394) to fiber optic (1394b) cable were not necessary. One of the cameras was set on the sidewalk and angled to capture traffic crossing over the bridge while the second one was attached to the safety fence and monitored traffic passing under the bridge on Interstate-5 (Fig. 9.17).

Prior to conducting this test, the pathway between the motherboard and hard drive on the PXI-8186 controller installed in the bridge shorted out and an alternate controller and chassis were used. The PXI-6031E data acquisition board and PXI-8252 IEEE-1394 interface board were removed from the PXI/SCXI chassis and installed into a smaller 4-slot PXI-1031 chassis. This new system was configured in the laboratory and then set inside the bridge (Fig. 9.18). Within Figure 9.18, the

PXI-1031 is located on the bottom right side of the picture while the DC power supplies for the capacitive accelerometers are mounted on the rack (left side) along with the original PXI/SCXI combination chassis. Because of very low signal-to-noise ratios, the strain gages were not used in this test. Debugging of these gages is ongoing and the replacement with / inclusion of piezoelectric and fiber optic strain gages is being explored.

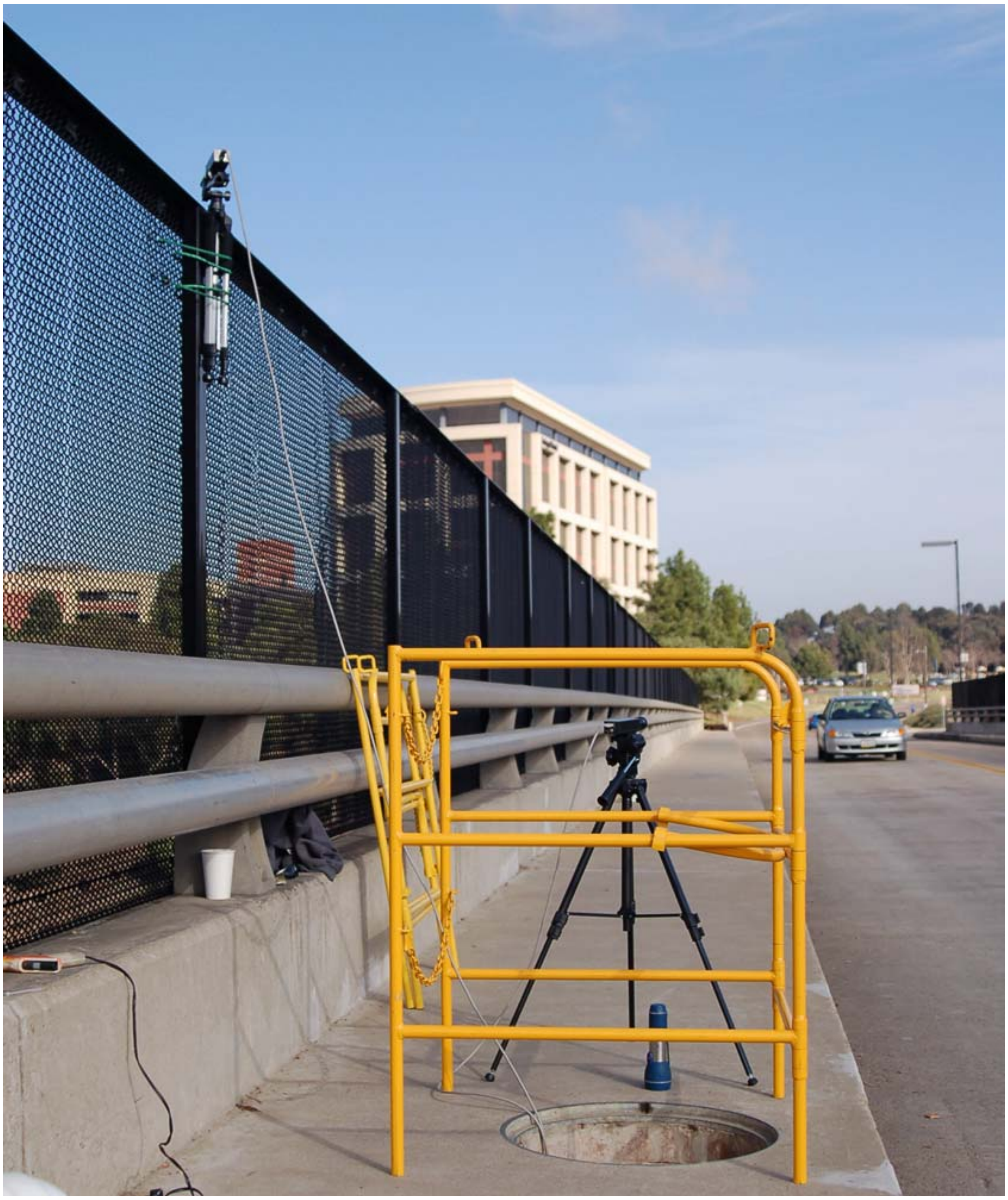


Figure 9.17: Location of the 2 Bayer Cameras

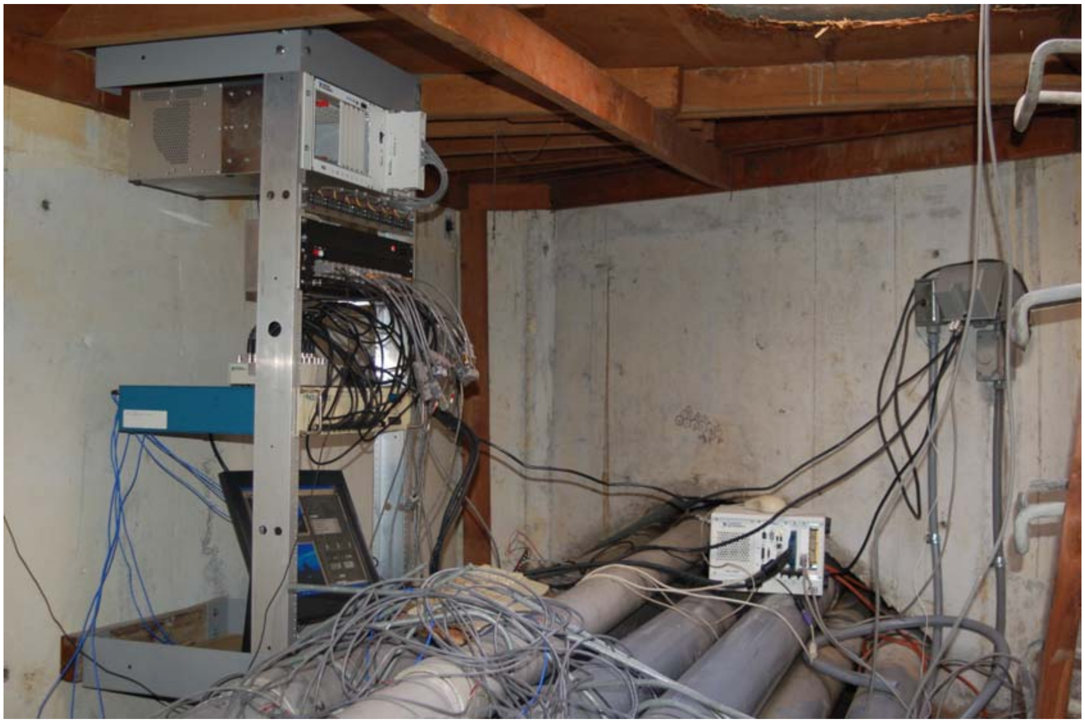


Figure 9.18: Data/Image Acquisition Hardware Installed within Bridge.

9.7.2 Data/Image Acquisition

Great care was taken in selecting the date and time of day to conduct this test. It was decided the best time to run the experiment was on a weekend during a campus holiday when traffic was expected to be at a minimum. This allowed for measuring the bridge response for individual vehicles rather than with multiple vehicles crossing simultaneously, as is usually the case. During the 33 minutes the system was recording data, 128 vehicles were captured. The number and types of vehicles are summarized in Table 9.1. Of particular interest in this data set are the three large vehicles. These were a garbage truck heading east and two westbound campus shuttle buses.

Table 9.1: Summary of Vehicles Recorded Crossing the Bridge

Vehicle Type	**Number Traveling West Bound**	**Number Traveling East Bound**	**Total Number Crossing Bridge**
Car	40	36	76
SUV	16	17	33
Van	3	4	7
Pickup Truck	5	4	9
Large Vehicle	2	1	3

In this experiment, accelerations were recorded using a 1000 Hz sampling rate and images were captured with a nominal 4 Hz acquisition rate. In total, 8167 images were recorded by each of the cameras. Samples of the recorded images from both of the cameras are shown in Fig. 9.19.

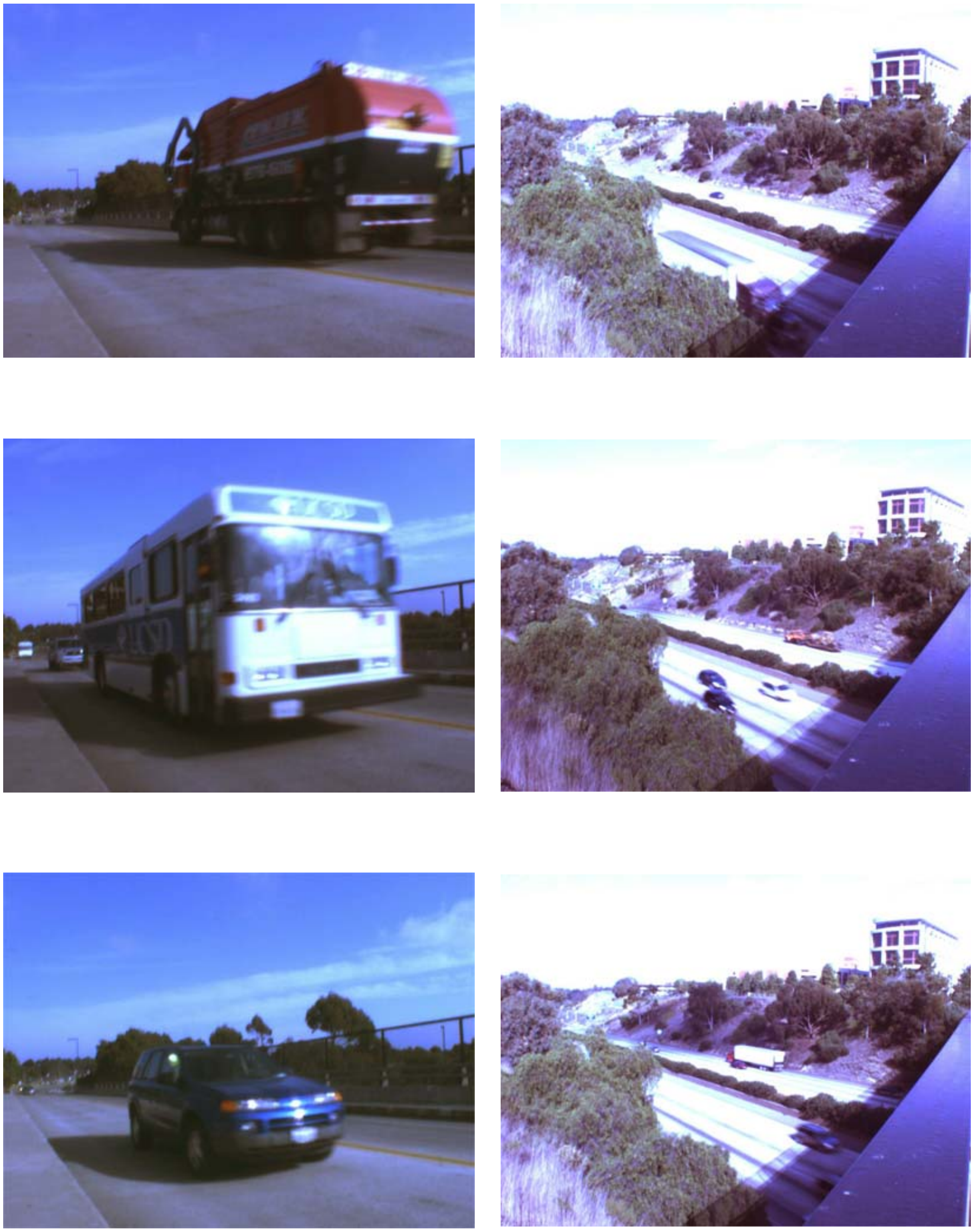


Figure 9.19: Sample of Images Recorded by Bayer Cameras

9.7.3 Qualitative Data Analysis

9.7.3.1 Ambient Vibration Analysis

A preliminary, and somewhat qualitative, analysis of the recorded data was performed and the results are presented herein. First, the response of the bridge subject to ambient loading conditions was considered by investigating measurements taken when no vehicles were crossing the bridge. Even during the relative calm of the weekend, this proved surprisingly difficult. From Figure 9.20, it can be seen that during the thirty plus minutes it took to conduct this experiment, 128 vehicles were captured. This makes for approximately one vehicle every 15 seconds.

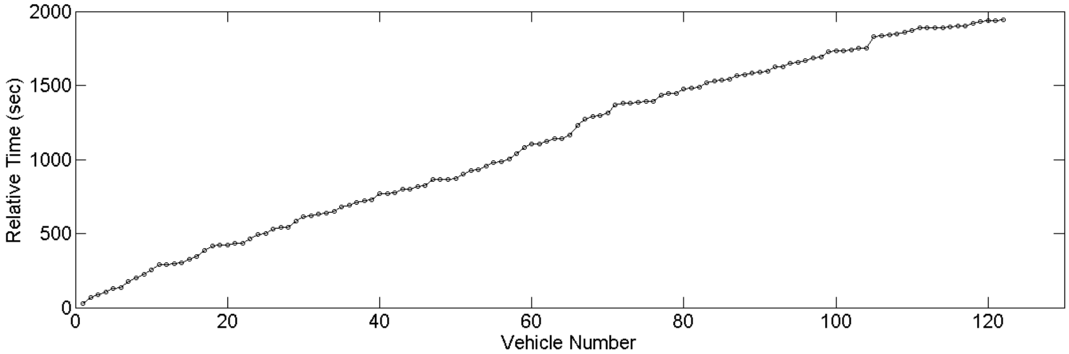


Figure 9.20: Relative Time (from Start of Test) for Each Vehicle Crossing

Two periods were selected and analyzed. The first was a 64 second period (from t = 1761 to 1825 sec), which was the longest duration recorded with no vehicles crossing the bridge. Acceleration time histories recorded by the PCB accelerometers (channels 1-8 in Fig. 9.6) are shown in Figure 9.21 along with the corresponding Fourier Amplitude Spectra (FAS) from all 20 accelerometers in Fig. 9.22. While many modes (at least 10) are present in this data, the lack of a suitable number of similar records makes it virtually impossible to perform the necessary averaging to do a modal extraction.

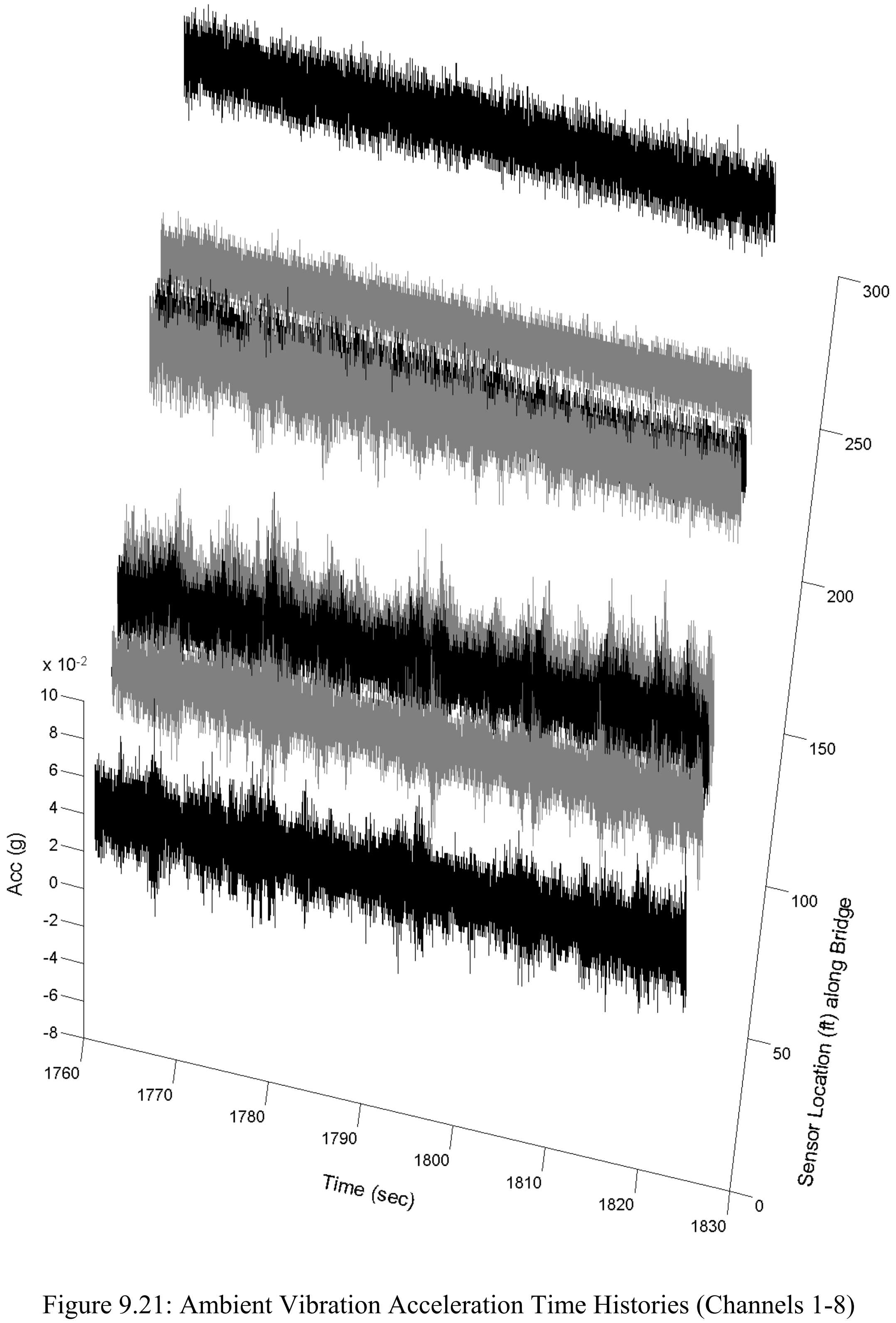


Figure 9.21: Ambient Vibration Acceleration Time Histories (Channels 1-8)

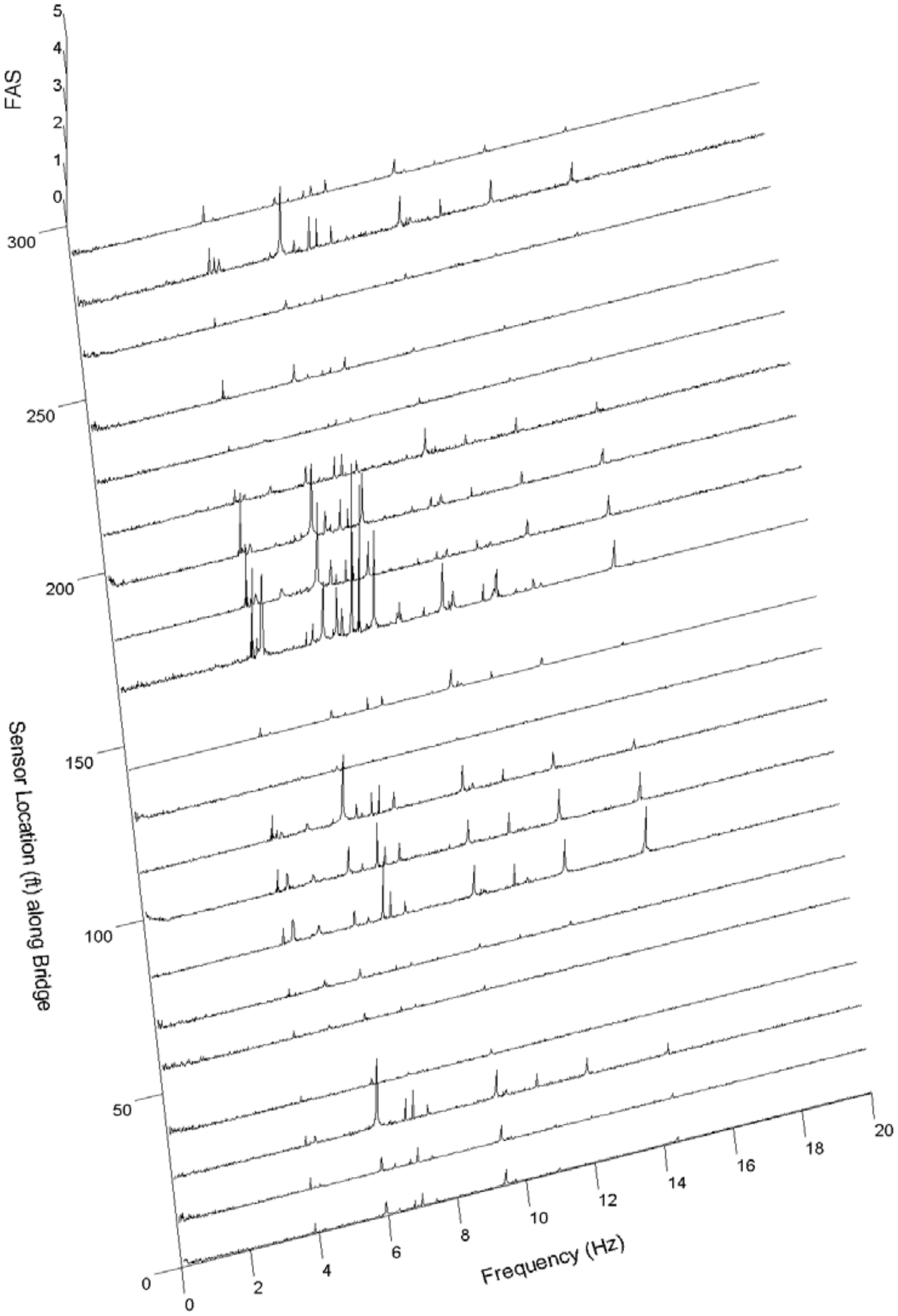


Figure 9.22: Fourier Amplitude Spectra from 65-Second Ambient Acceleration Time Histories

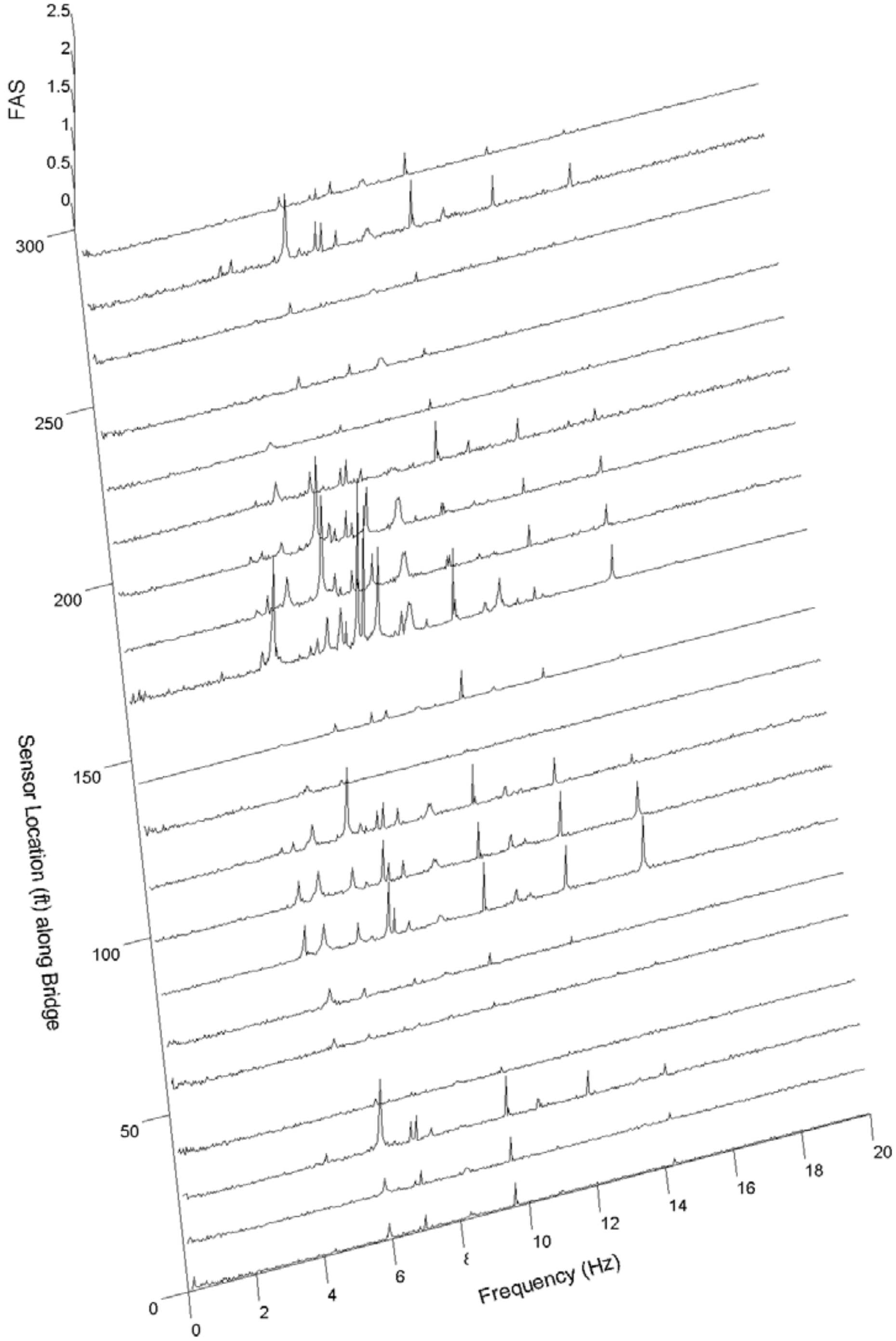


Figure 9.23: Fourier Amplitude Spectra from 30-Second Ambient Acceleration Time Histories

9.7.3.2 Traffic Induced Vibrations

Next the response of the bridge was analyzed for traffic induced vibrations. Two sets of images (taken when the vehicle passes over accelerometers 1, 3, 6, and 8) are shown for a Westbound and Eastbound car (Fig. 9.24). The corresponding Channel 1-8 acceleration time histories are shown in Fig’s. 9.25 and 9.26. From these figures it is observed that each span only demonstrates a strong response while the vehicle is located on that span. Next, two sets of time histories, each 400-seconds long, are plotted in Figures 9.27 and 9.28. A spectrogram of the data in Fig. 9.28 was generated using a Short Time Fourier Transform (STFT) and is shown in Figure 9.29. Of particular interest are the large amplitudes at high frequencies (specifically at 225 Hz in Channels 1 and 6 and 425 Hz in Channel 3). As these frequencies are well out of the dynamic ranges of the PCB 3801 and Crossbow CXL01lF1 accelerometers, it was decided to instead focus on the low frequency dynamics of the structure (Fig’s. 9.30 and 9.31). To remove the high frequency content from the time histories, an anti-aliasing lowpass (FIR) filter was applied to the data. Next, the data was decimated to change the time increment from 0.001 seconds to 0.005 seconds. These operations were performed using the “resample” function in Matlab. The original channel 1-8 acceleration time histories (Fig. 9.32) and resampled time histories (Fig. 9.33) are shown for a typical westbound car. The corresponding Fourier Amplitude Spectra are shown in Fig. 9.34.

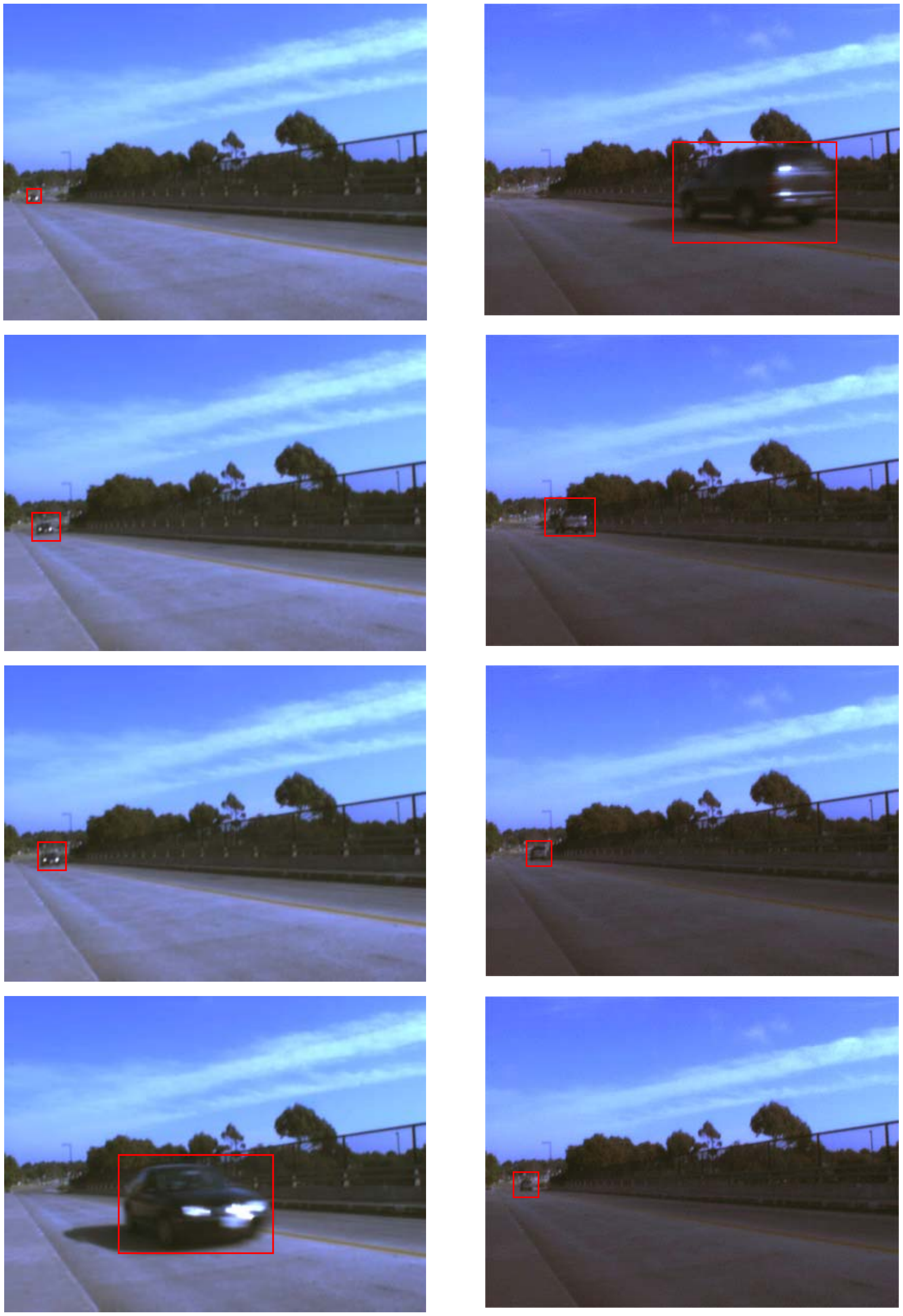


Figure 9.24: Recorded Images for 2 Typical Cars Taken when the Vehicle Reaches Channels 8, 6, 3, 1 (Top to Bottom).

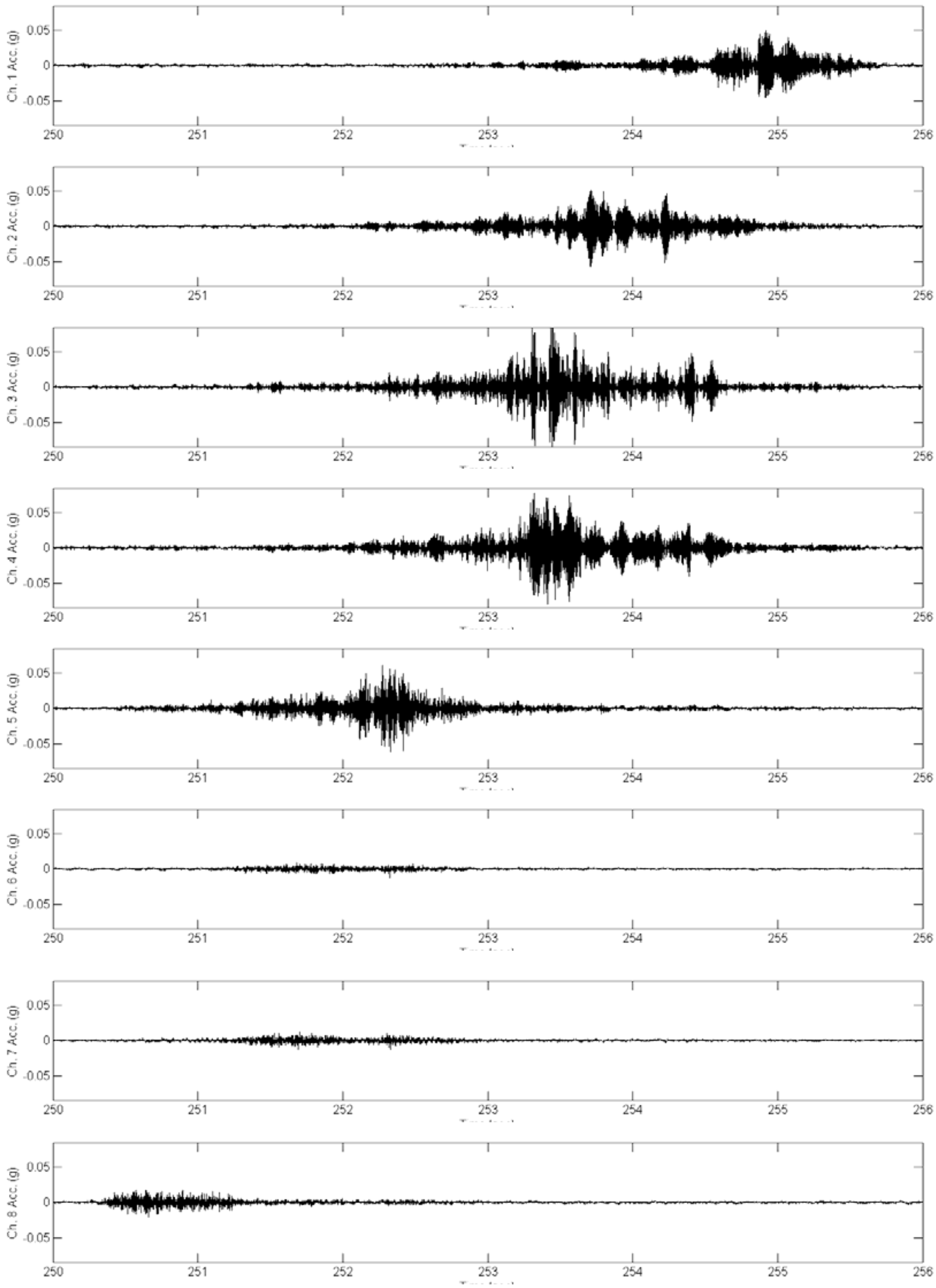


Figure 9.25: Channel 1-8 Acceleration Time Histories for the West-Bound car in Fig. 9.23

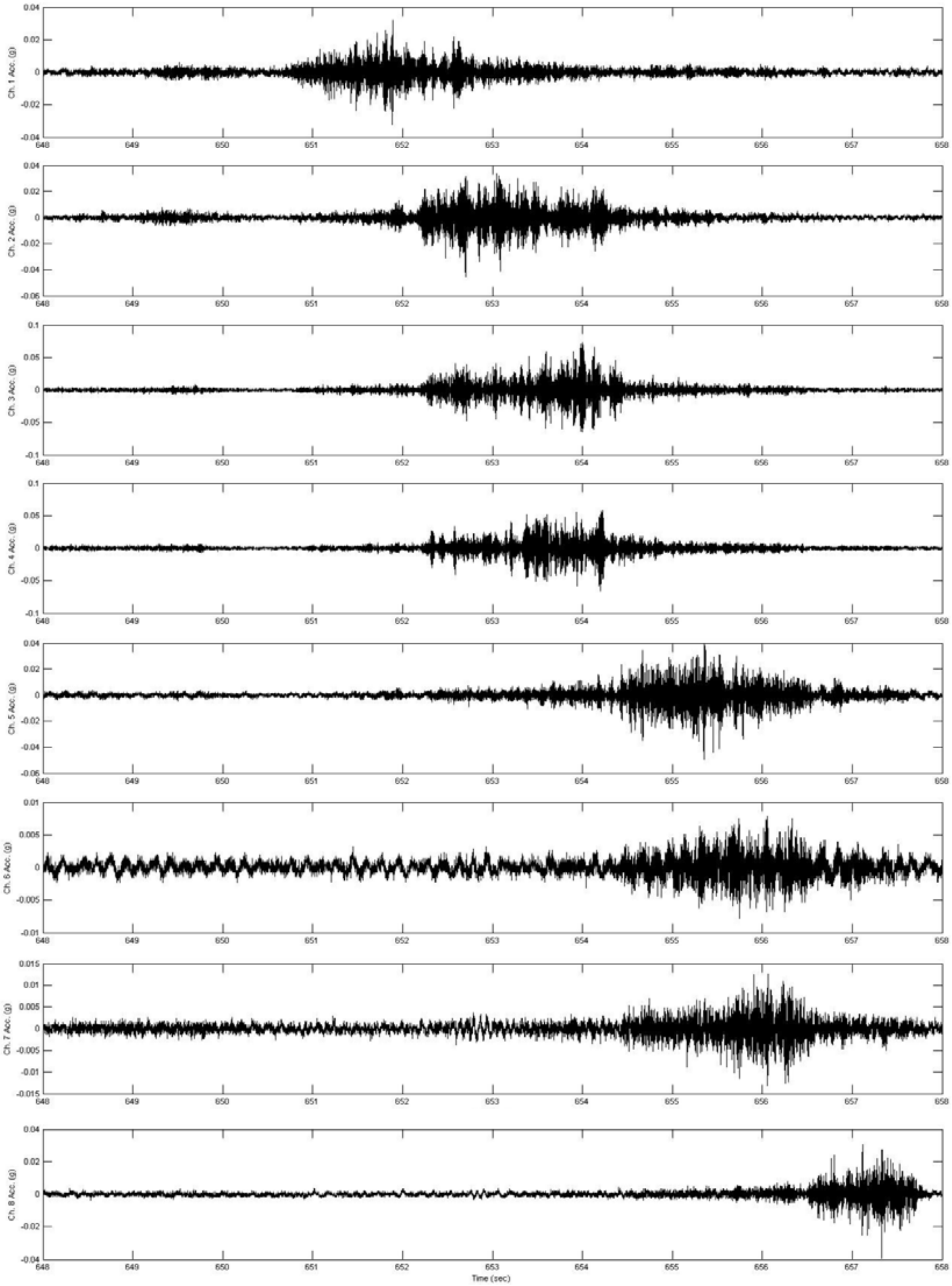


Figure 9.26: Channel 1-8 Acceleration Time Histories for the East-Bound car in

Fig. 9.23

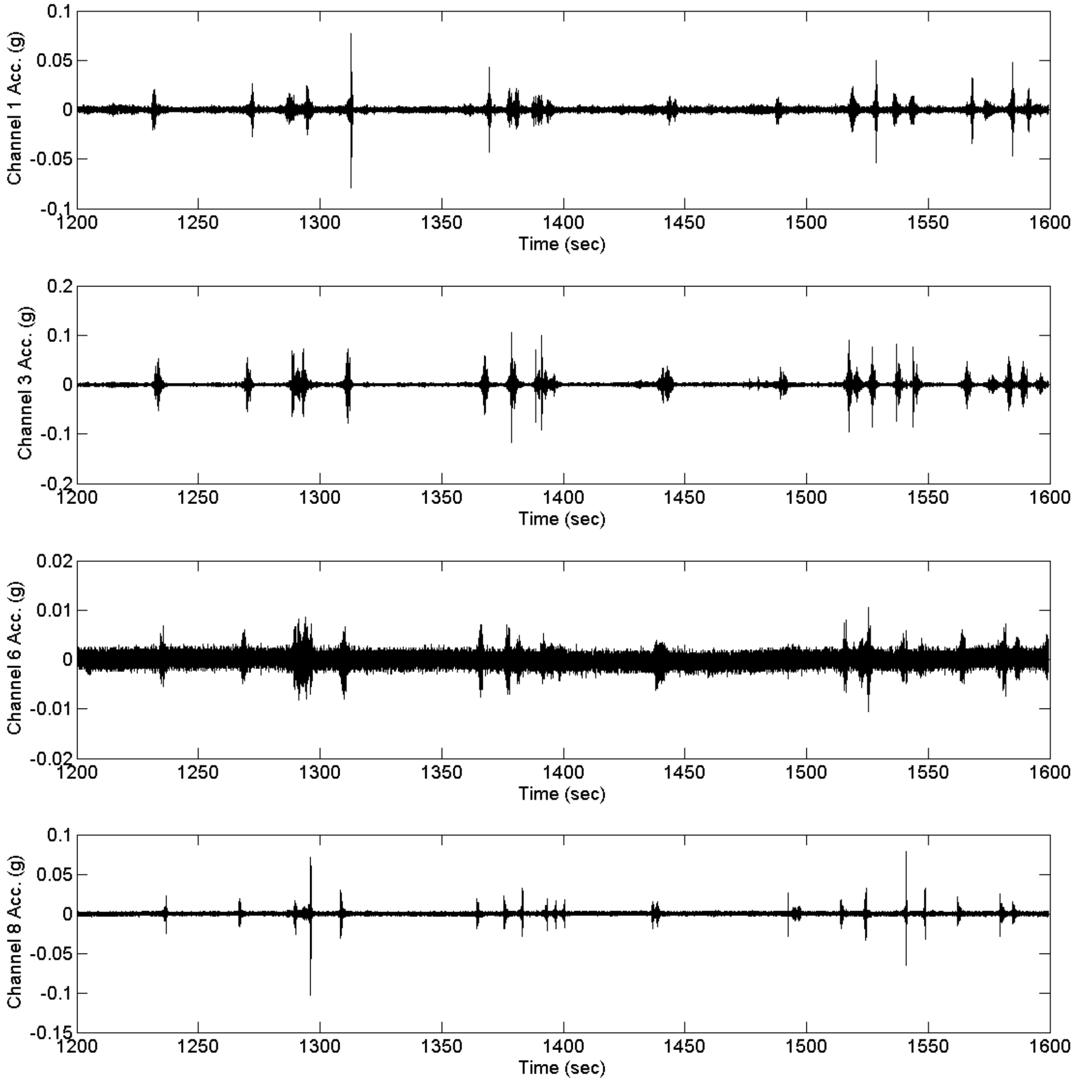


Figure 9.27: Acceleration Time Histories for Channels 1, 3, 6, 8 (Fig. 9.6)

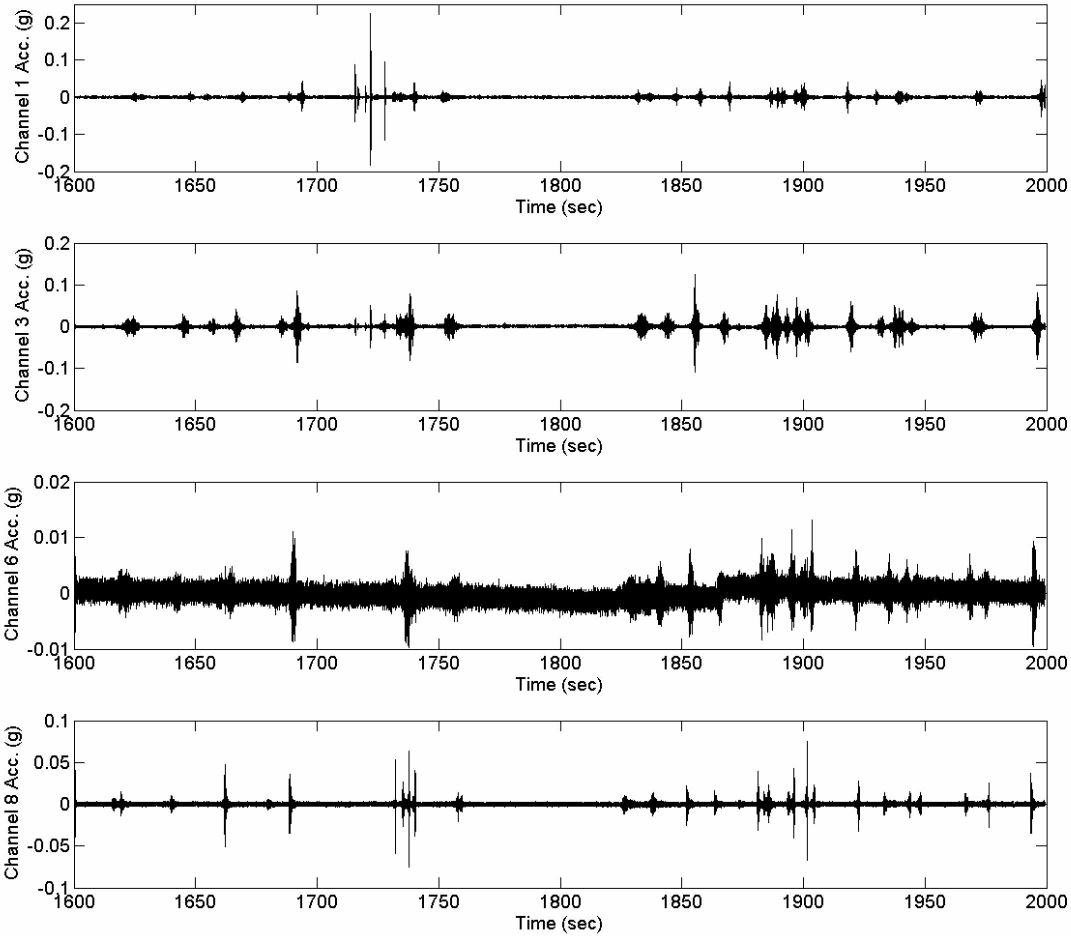


Figure 9.28: Channel 1, 3, 6, 8 Acceleration Time Histories

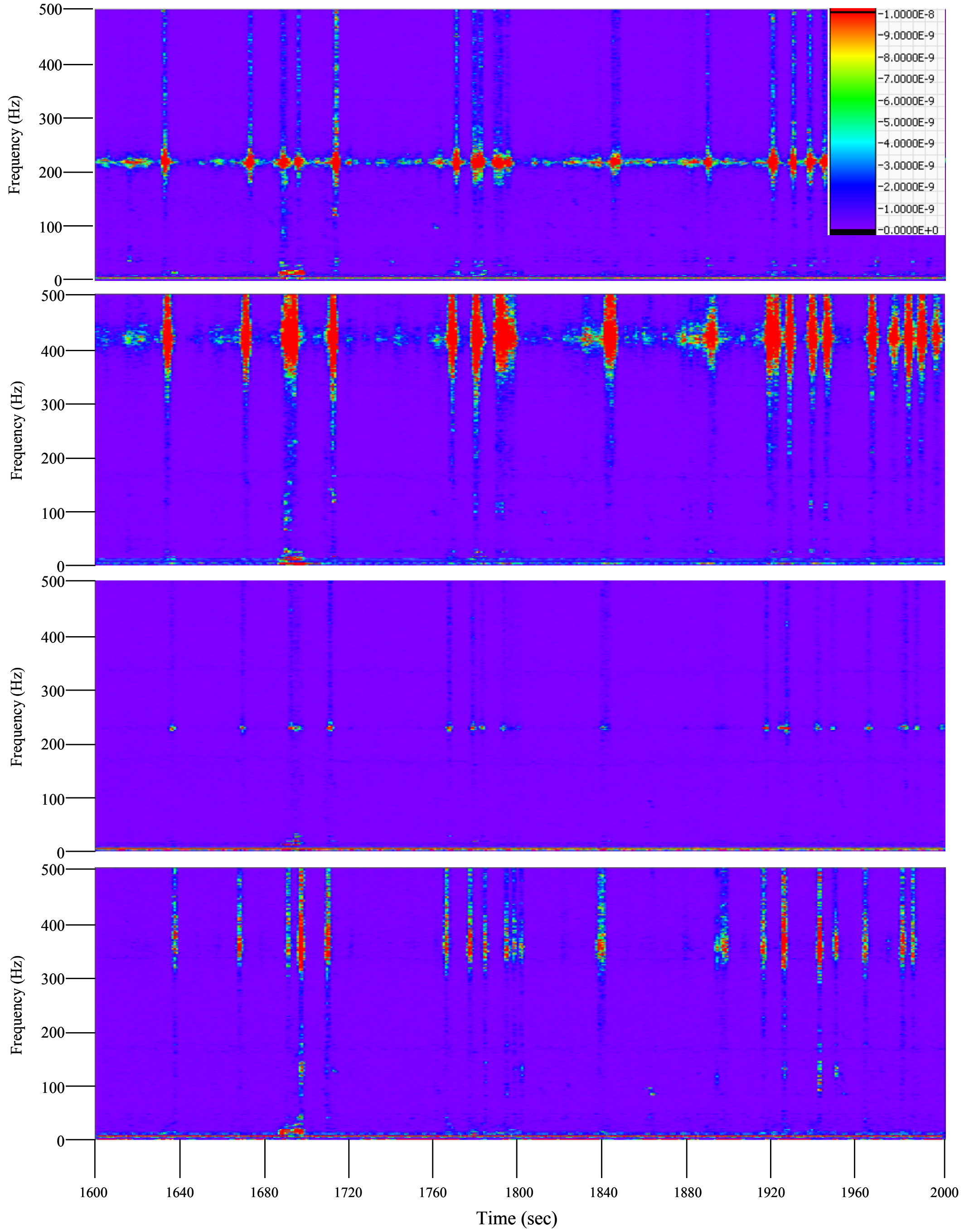


Figure 9.29: Spectrogram of Channel 1, 3, 6, 8 Acceleration Time Histories

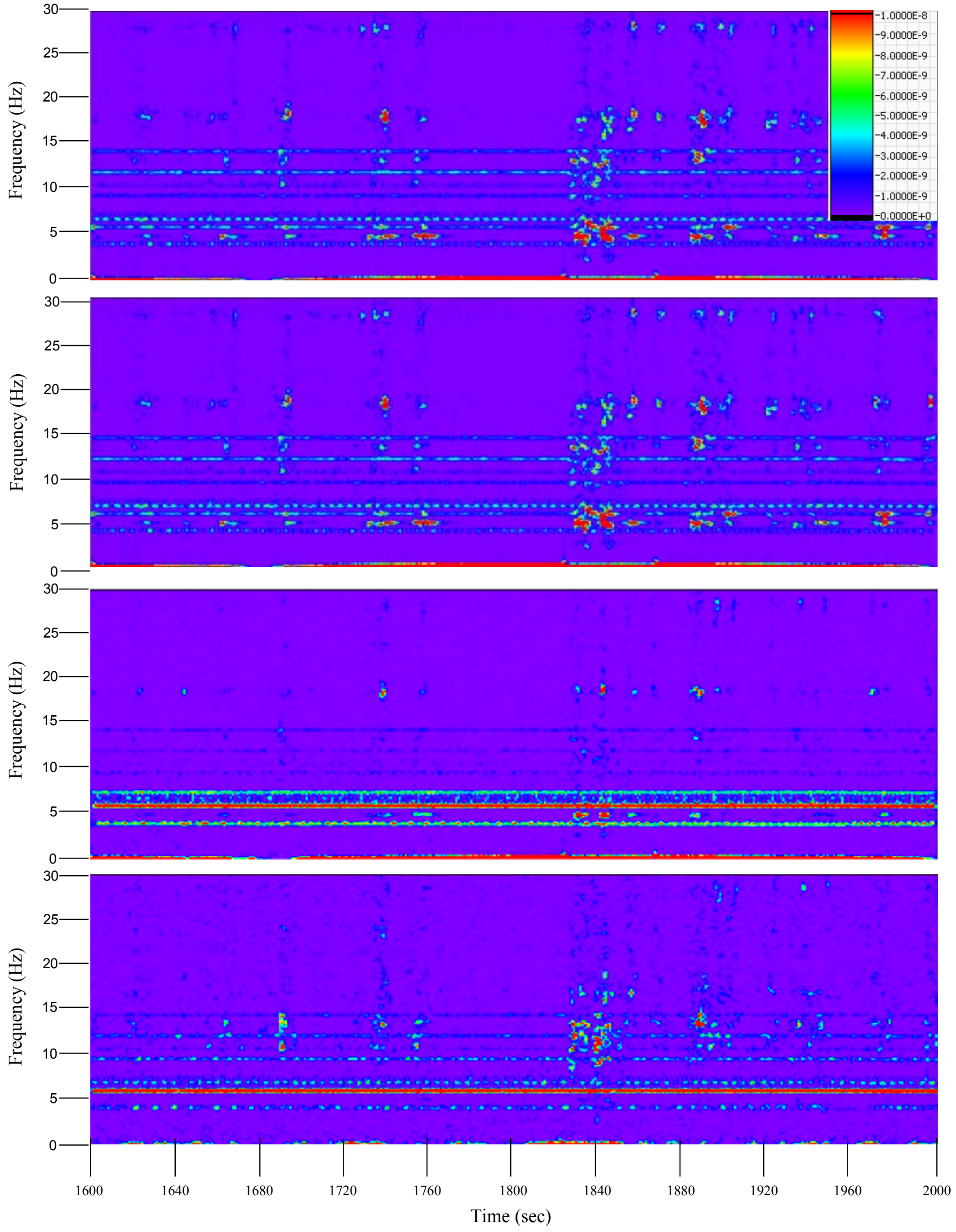


Figure 9.30: Low Frequency Range of Spectrogram (0 – 30 Hz) for Channels 1, 3, 6, 8

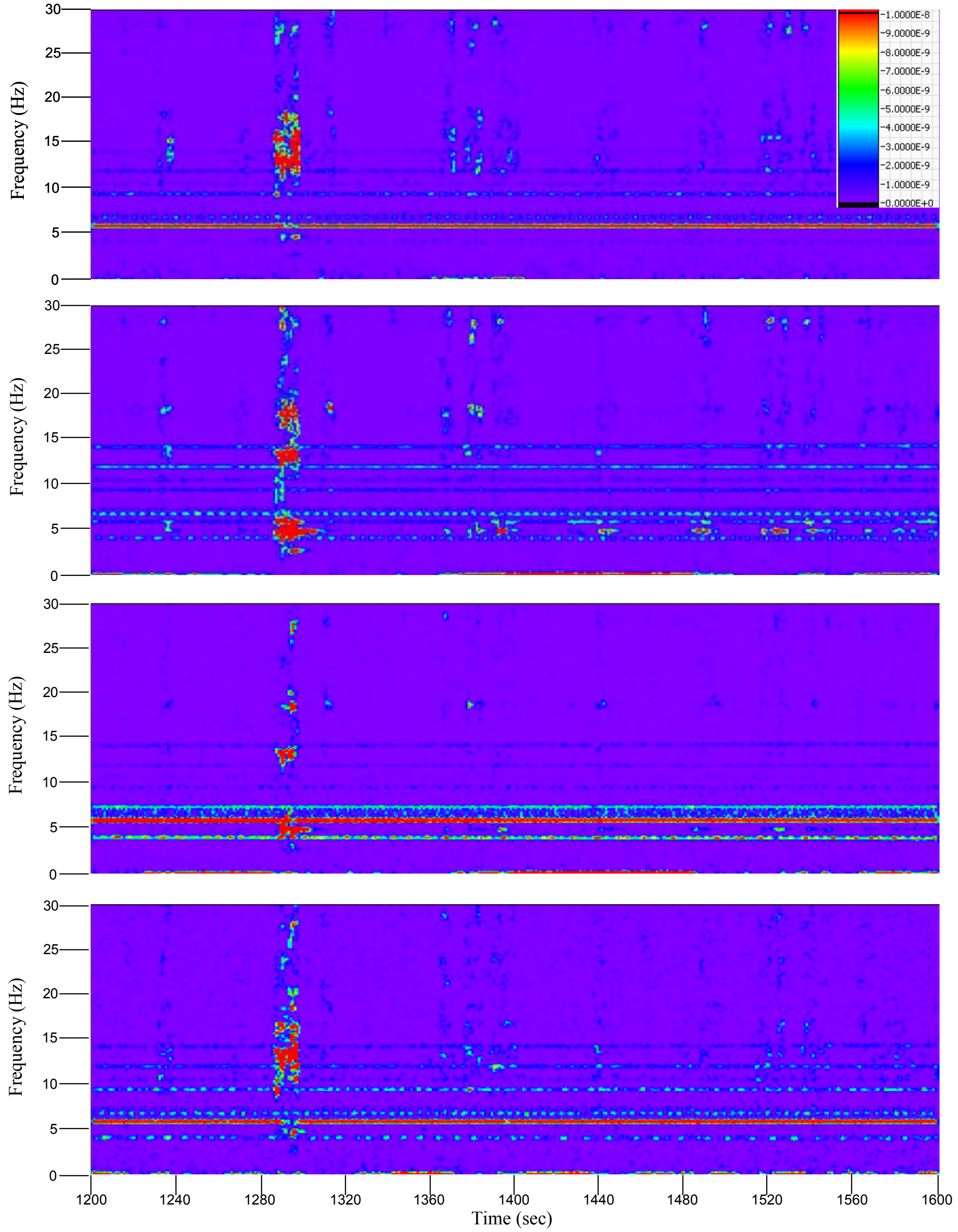


Figure 9.31: Spectrograms of Channel 1, 3, 6, 8 Acceleration Time Histories (top to bottom)

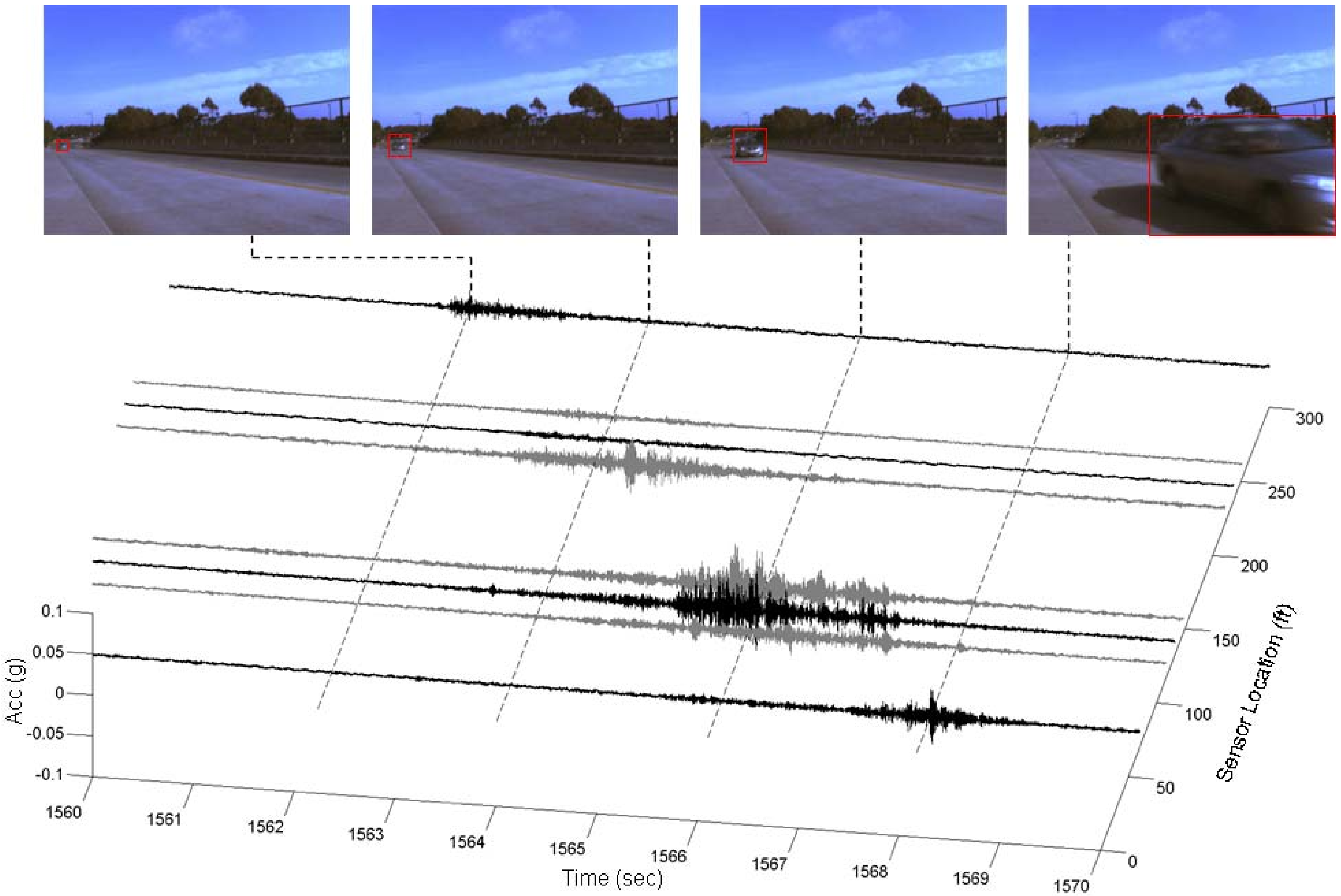


Figure 9.32: Original Recorded Acceleration Time Histories for a Typical Westbound Car

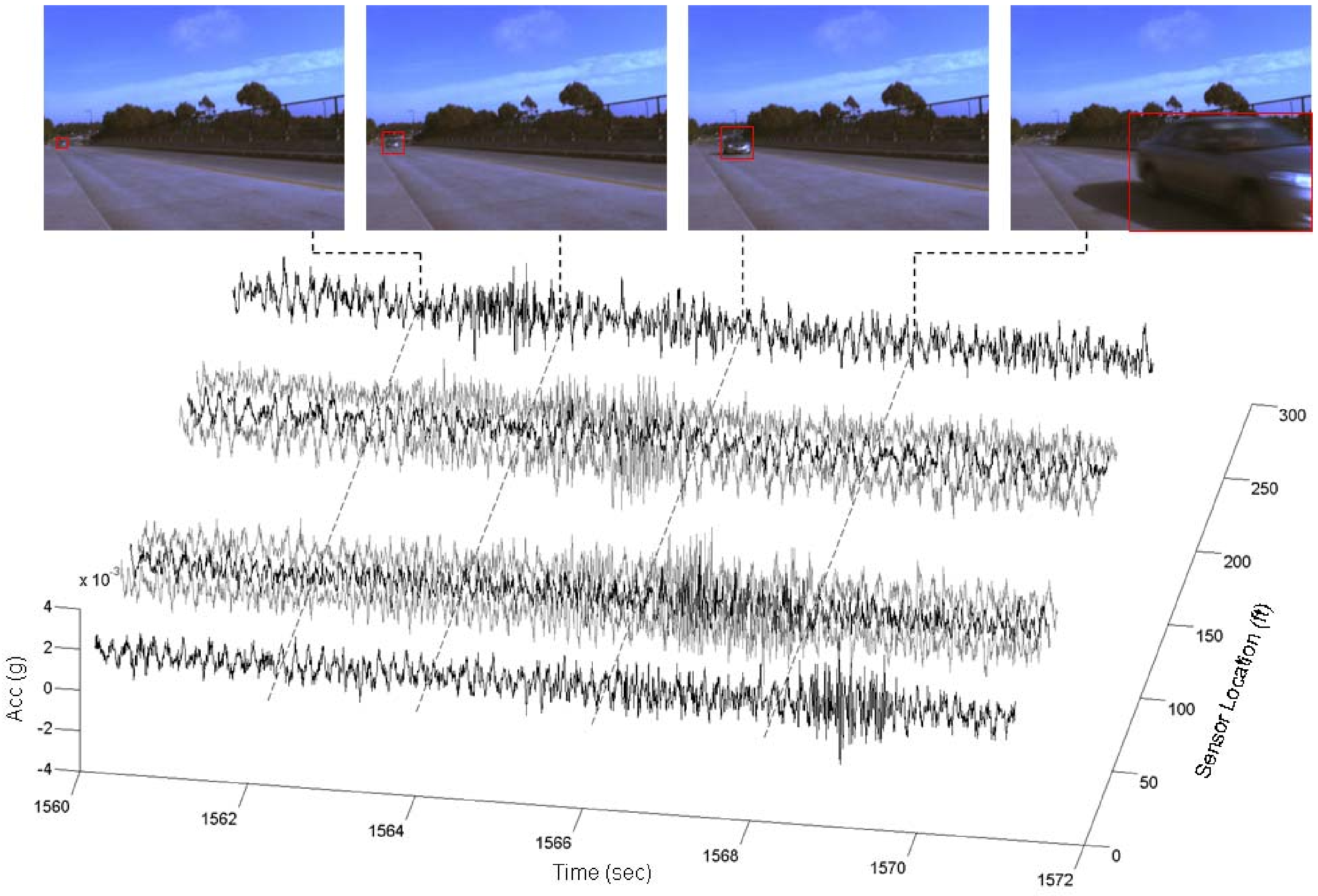


Figure 9.33: Resampled Acceleration Time Histories for a Typical Westbound Car

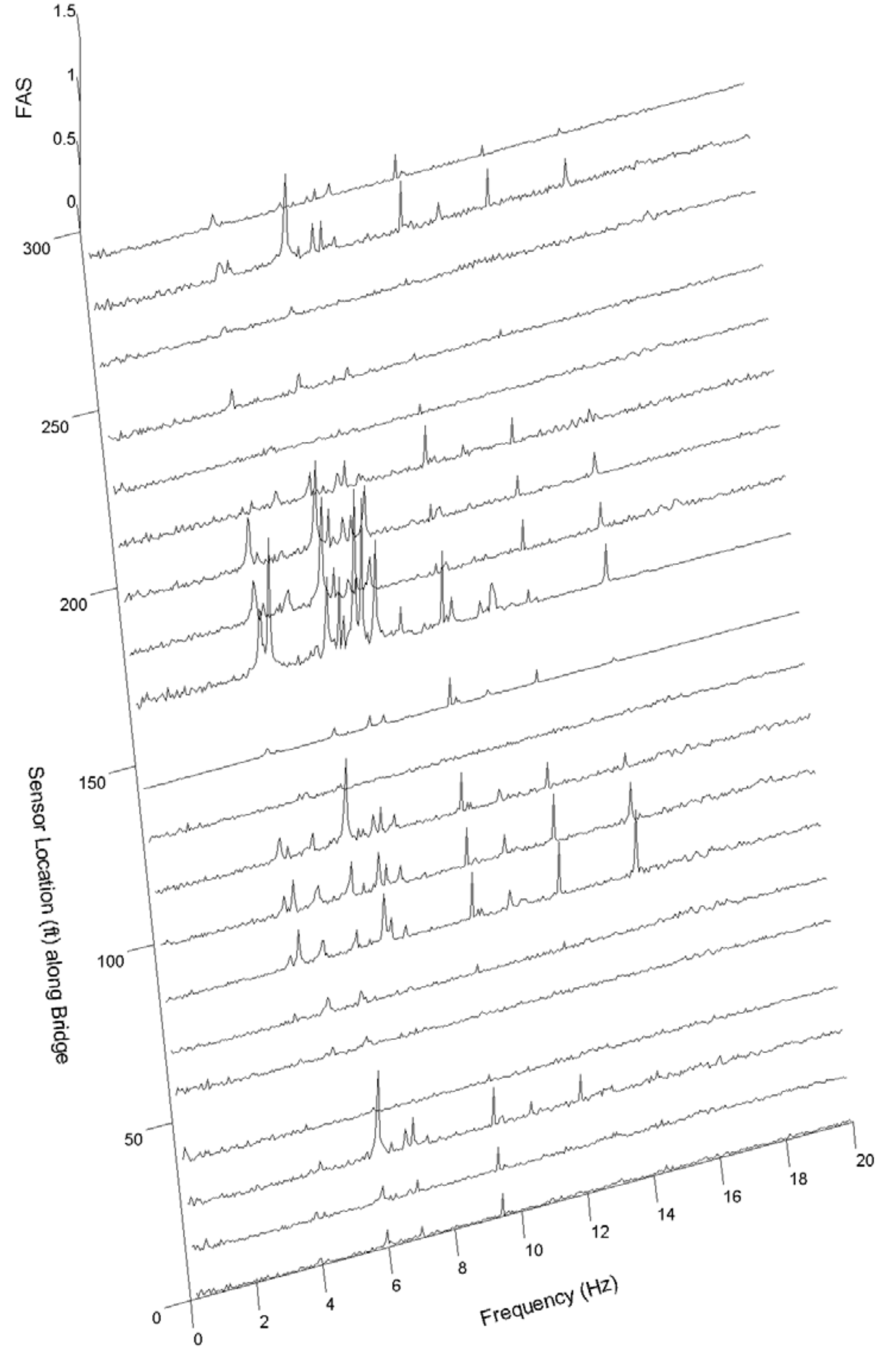


Figure 9.34: Fourier Amplitude Spectra (FAS) for a Typical Westbound Car

A second traffic case was also considered. Based on the spectrograms, an anomaly in the bridge response was observed at approximately t = 1290 sec. From the recorded video, this was found to be a garbage truck (Fig. 9.19) moving east on the bridge and a small SUV traveling westbound. The original acceleration time histories along with the resampled data are shown in Figures 9.35 and 9.36, respectively. The corresponding Fourier Amplitude Spectra is included in Fig. 9.37. From the Fourier Amplitude Spectra, the traffic induced vibrations (forced vibration) is evident in the broad bands being excited, rather than simply vibrating at only a few natural frequencies. Also worth noting is the disruption in the bridge response while the garbage truck travels along the overcrossing. Under normal loading conditions (ambient as well as smaller vehicular loads), a very strong and prominent response at approximately 6.0 Hz is observed. However, when the garbage truck passes over the bridge, the amplitude of this frequency diminishes in favor of others (Fig. 9.31). Once the truck moves off the bridge, the 6 Hz signal quickly resumes.

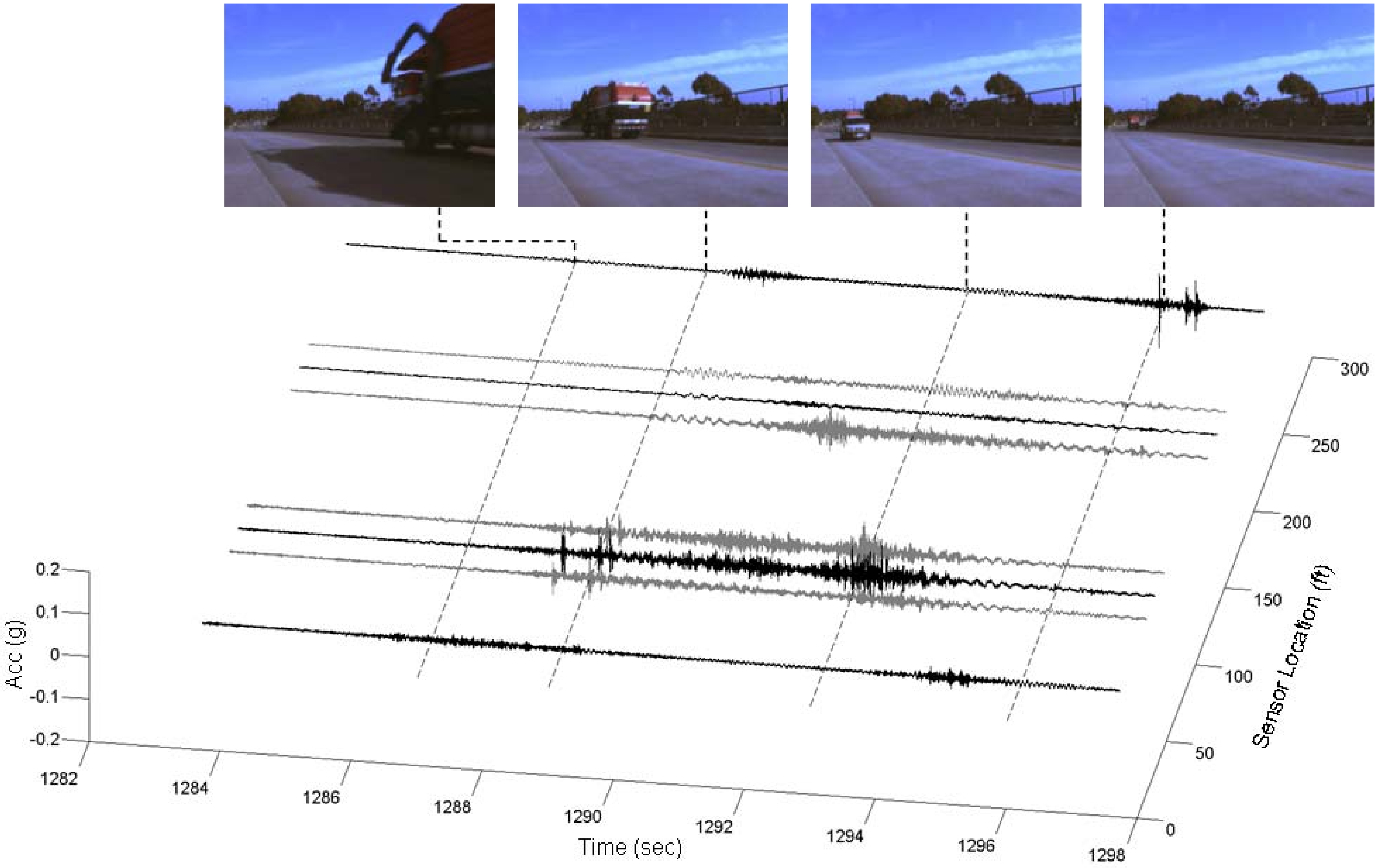


Figure 9.35: Acceleration Time Histories for Channels 1-8 with Corresponding Video

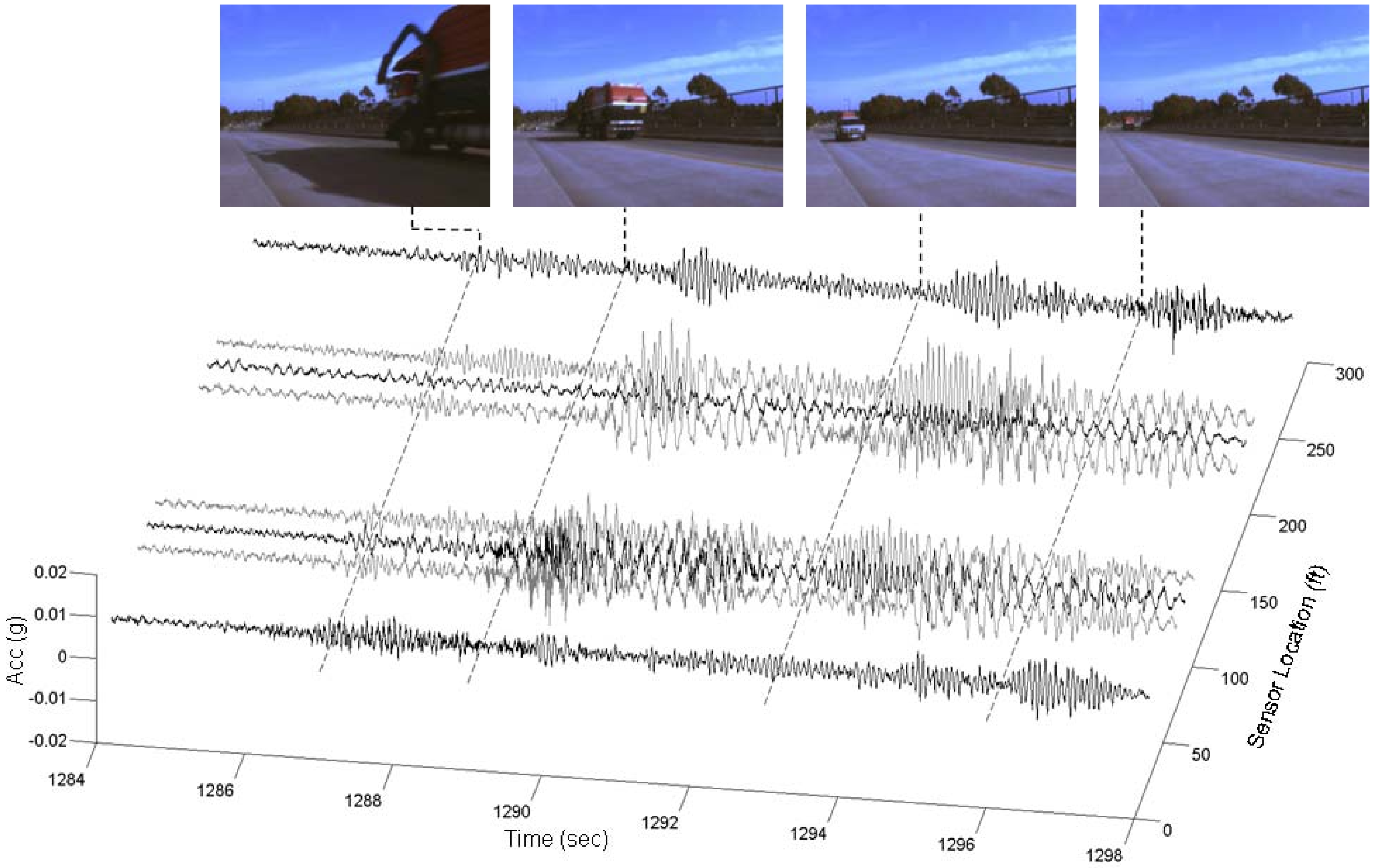


Figure 9.36: Resampled Channel 1, 3, 6, 8 Acceleration Time Histories

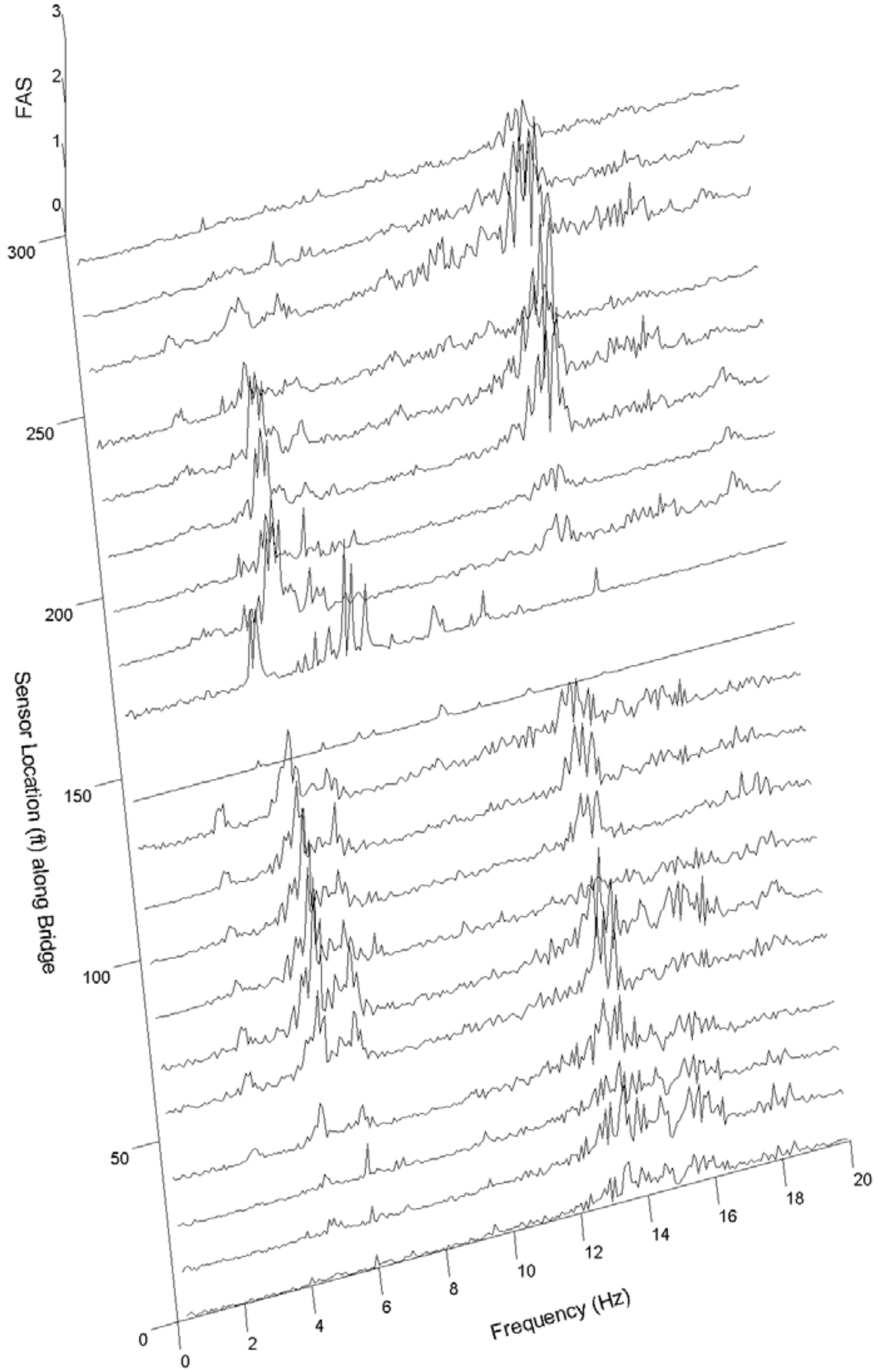


Figure 9.37: Fourier Amplitude Spectra of 20 Accelerometers Under Loading from Garbage Truck

9.8 Summary

Capitalizing on lessons learned from the composite bridge decks, a second bridge testbed has been established on the Voigt Drive / Interstate-5 overcrossing, located next to the UCSD campus, for further testing of health monitoring technologies. This new continuous monitoring system is built around an easily expanded system supporting over 250 channels of sensor data and 3 cameras. Of particular interest is the integration of the image acquisition into the data acquisition computer thereby providing hardware synchronization between the sensors and cameras. A shakedown test was performed on the system using the initial sensor array composed of twenty capacitive accelerometers and two color cameras. Time-synchronized video and acceleration were recorded continuously over a thirty minute period, during which time 128 vehicles were captured crossing the bridge. This data is suitable for performing system identification for determining natural frequencies, mode shapes, and damping ratios as well as for exploring the dynamic response associated with vehicle/structure interaction. This system will soon be made available to collaborators from UCSD and abroad for testing new sensor types, data acquisition/transmission methods, and data mining strategies.

10 Summary and Conclusions

This section starts with a summary of conducted investigations followed by a concise list of dissertation contributions. In Chapter 2, elements composing a state of the art structural monitoring system have been detailed. First, an overview of the dynamic and environmental sensors available for a heterogeneous sensor array was presented. For continuous monitoring applications, where data logging is not practical, PC-based data/image acquisition has been addressed, and a key issue of data/image synchronization was noted. Data transmission protocols for streaming data from a remote bridge site to more suitable location were explored. Finally, since simply storing acquired data as a series of a text files is not a viable option in post-processing data analysis, issues related to database systems were discussed.

A research testbed for applying the aforementioned structural monitoring framework was established and described in Chapter 3. Pc-based data acquisition was employed along with a network camera for continuous monitoring of a series of instrumented composite bridge-decks. Over a three-year period, time synchronized video and strain data have been recorded and archived in a series of databases, and made available for on-line querying. The continuously recorded data was post processed and separated into 439,654 discrete events, each composed of a ten-second strain time history and 40 images of a vehicle crossing the bridge decks. Local processing potential for data reduction and event detection was demonstrated with the peak hourly strain database established for the 2-channel monitoring system.

Several demonstration applications have been made with this testbed. First, image processing and feature extraction were performed on the video data recorded from the composite bridge decks (Chapter 4). For each of the aforementioned strain time histories, image processing was used to extract the picture most clearly showing the vehicle crossing the bridge decks, thereby making it possible to sort and bin the time histories by vehicle type. This operation allowed for examining the distribution of vehicles crossing the decks, useful for studying probability distributions of traffic loads and their effects (e.g., strains). Further, by sorting and binning the recorded traffic, a unique data set was established in which the measured bridge response and images corresponding to the loads (passing traffic) are both available. This data is extremely useful for applying artificial intelligence algorithms for vehicle classification and property estimation as well as for monitoring for changes in the bridge system response over time.

As traffic passing over the bridge decks only traveled at relatively low speeds (majority at or under the speed limit of 25 MPH), it was not possible to train neural networks on the recorded data for vehicle property (speed, wheelbase, axle weights) estimation. Also, vehicle induced strain data taken from a damaged state of the bridge decks was not available, and damaging the decks was not a possibility. Consequently, a computational model of the bridge decks was created to provide strain time histories similar to those recorded on the actual system

(Chapter 5). Good results were obtained using a pilot one-dimensional finite element model in which the interaction between the structure and vehicle were ignored (instead traffic was modeled as a series of moving loads). The generation of the finite element input and load files was implemented into Matlab which allowed for efficient generation and execution of the large numbers of runs (thousands) necessary for the conducted parametric studies and neural network training.

The strain time histories generated from the finite element model were used for vehicle property estimation and damage detection. These operations were first done using the peak axle strains (Chapter 6) and then later with neural networks (Chapter 7). The use of peak middle and quarter span strains allowed for calculating vehicle speed and wheelbase with moderate accuracies. Excellent results were obtained using neural networks, even under random traffic loading. With the neural networks, different methods of feature reduction were explored (including peak axle strains and principal component analysis). It was found that using peak axle strains (along with the corresponding time at which these strains occur) as inputs gave the best results, and accuracies under random traffic loading of 0.42 MPH for speeds, 0.17 ft for wheelbases, 0.07 kips for front axle weights, and 0.13 kips for rear axle weights were achieved.

For damage detection, good results were obtained using data generated by a single vehicle type and the peak axle strains from 16 locations along the length of the finite element model. For the more difficult and realistic scenario of random traffic loading, neural networks had to be employed. Neural networks were able to detect and classify the level of damage, with 90% accuracy, even when damage was allowed to occur in simultaneous locations. By training separate networks to monitor each damage zone and combining the output from each, the amount of training data required is vastly less than using a single network (which requires many more damage scenarios). However, as with all supervised learning techniques, the more refined the classification levels, the more data required for network training. Therefore, an alternate method of damage detection was explored which did not require any training data from the damaged state of the structure (a big advantage when applying the detection methods to real data). This new method used strain time histories from one location on the FE model of the bridge decks to train a neural network to predict an associated time history at another location. Principal Component Analysis was employed to reduce the number of features in the each of the input and target strain time histories from 4000 to less than 10. As the neural network is trained using data exclusively from the undamaged state when damage occurs, the error between predicted and measured (generated) strains within these damaged regions increases. A normalized error was utilized providing correlation between the percent reduction in stiffness (level of damage) and the difference in the predicted and observed strains, even under random traffic loading. Ultimately, changes in stiffness as low as 2% could be consistently detected, providing a significant improvement over the previously employed methods. Again, as this method does not require data from the damaged structure and can be trained using random traffic patterns, it is easily implemented on actual data from the composite bridge decks.

Speeds and then wheelbases were calculated using the recorded strain time histories in the labeled data subset from the composite bridge-decks (Chapter 8). By using the labeled data, it was possible to focus on vehicles with constant wheelbases, which provided an error check in the calculated wheelbases by plotting histograms of the wheelbases for the different vehicle types. Next, the peak axle strains recorded over the period of time the monitoring system was operational were considered. The numbers and distributions of vehicles crossing over the bridge decks were examined. This information allows for accurately modeling the probability distributions of traffic loads, which without this more detailed load data is typically modeled with random distributions.

The same predictive neural networks used in Chapter 7 for damage detection on the finite element model data were employed on the recorded data from the composite decks to look for changes in the response of the bridge-decks (Chapter 8). This was done by considering only 2-axle traffic with peak strains typical of campus buses. The data from strain gage 8 was used as input, and three neural networks were constructed to predict strains in neighboring gages 6, 9, and 11. These three neural networks were trained using 1000 patterns recorded in December 2003 and tested on all similar records recorded during 2004 (over 39,000 time histories). Principal Component Analysis was again used for feature reduction and cut the number of features input and output by the neural network from 2000 (each time step being a feature) to 9 or fewer. In addition to comparing the error between the measured and predicted (from neural network) strains, a second comparison was done with strain time histories reconstructed using the inverse Principal Component Analysis transformation applied to the extracted PC's. In general, there was very good correlation between the measured strains and those predicted by the neural networks and reconstructed using the inverse PCA transformation.

Capitalizing on lessons learned from the composite bridge decks, a second bridge testbed has been established on the Voigt Drive / Interstate-5 overcrossing, located next to the UCSD campus, for further testing of health monitoring technologies (Chapter 9). This new continuous monitoring system is built around an easily expanded system supporting over 250 channels of sensor data and 3 cameras. Of particular interest is the integration of the image acquisition into the data acquisition computer thereby providing hardware synchronization between the sensors and cameras. A shakedown test was performed on the system using the initial sensor array composed of twenty capacitive accelerometers and two color cameras. Time-synchronized video and acceleration were recorded continuously over a thirty minute period, during which time 128 vehicles were captured crossing the bridge. This data is suitable for performing system identification for determining natural frequencies, mode shapes, and damping ratios as well as for exploring the dynamic response associated with vehicle/structure interaction. This system will soon be made available to collaborators from UCSD and abroad for testing new sensor types, data acquisition/transmission methods, and data mining strategies.

10.1 Key Contributions / Findings from this Research

- An integrated system for structural health monitoring has been developed and is ready for deployment at the 1000's of heterogeneous sensors level. This system handles all tasks of monitoring, including data acquisition, data transmission, data archiving, and database querying. In doing so, the following was developed and integrate:
    1. A scalable data acquisition systems, such as the one employed on the Voigt Bridge testbed (Chapter 9), capable of acquiring data from a large number of diverse sensors.
    2. A TCP/IP streaming program which is robust enough of handle variable network speeds and network outages. This program has to be separate from the data acquisition programs (to prevent crashing during network outages) but still able to access the flow of data coming from the acquisition buffers.
    3. A data loader program to buffer the incoming data streams and to upload the data to the database systems.
    4. Methods for synchronizing data from the multitude of sensors and cameras.
    5. Expandable databases to manage the flow of data coming from the bridges and to make all of the data accessible for analysis. Further

derived data from analyses must be capable of being stored into the databases.

This framework has been implemented and tested on the composite bridge-deck panels and Voigt Drive / Interstate-5 overcrossing testbeds.

- The acquisition and use of time synchronized sensor plus video data has been pioneered. Now, in addition to being able to monitor the response of the structure using recorded video, it is possible to define the loads acting on the structure.
- An archived and available database containing over 25,000 sets of traffic induced strain time histories with video have been recorded over a three year period. This data set is composed of peak hourly strain time histories (and video) recorded from two strain gages installed on the bridge-deck panels testbed and contains a wide range of vehicle types, from passenger cars to 5-axle truck and trailers.
- An available database containing approximately 500,000 sets of synchronized traffic induced strain time histories with video have been recorded over a one year period. This data set contains the recorded strain time histories (and video) from sixteen strain gages for every vehicle passing over the composite bridge-deck panels testbed during the time span the monitoring system was operational.
- Using the extracted features from image processing performed on recorded video, a unique data set of 7,561 records was established in which traffic

was binned into 57 vehicle types. This data, recorded over a one year period, is ideal for supervised learning algorithms in that the tedious and time consuming task of data labeling and data cleansing has already been performed. Within this data set, data from two strain gages along with the time synchronized video is made available. This data has many uses in video analysis alone, sensor analysis alone, or in data fusion. Potential applications for this data include vehicle property estimation, traffic classification, and damage detection.

- A second dataset of 4,863 labeled traffic induced strains and video has been created from all the traffic passing by in a one-week period. Here, data from sixteen strain gages along with video is made available. This data is particularly useful for vehicle classification and vehicle property estimation.
- All of this sorted and cleansed strain data with video (along with the original data) is made available on-line for querying and browsing. For researchers interested in utilizing the data, upon request, it can be shipped to them on an external hard drive.
- Within the numerical simulation framework, it has been shown that neural networks performed satisfactorily in providing highly accurate estimates of vehicle speeds, wheelbases, and axle weights.
- A neural network-based damage detection methodology has been constructed, verified on the numerically simulated data, and tested on the recorded data. Strain time histories from one location (sensor) on the

structure are used as inputs for neural networks, which then predict strains at neighboring locations. The error between these predicted and strains and the actual values is monitored. With this method, all of the network training is done with data from the undamaged structure; therefore, when damage occurs, the mappings employed by the neural networks are no longer valid. Consequently, the error between the predicted and measured strains increase and the magnitude of the error is utilized for assessing the extent of the damage. This method was tested on the simulated data from the finite element model with outstanding results.

- The developed integrated monitoring framework has been implemented on the Voigt Drive / Interstate-5 overcrossing testbed. While the initial installation incorporates approximately 30 sensors, this number is expected to quickly grow, and the system is ready to handle thousand-level sensor arrays.
- A key feature of this full-scale, state-of-the-art bridge testbed is the open atmosphere encouraging collaboration with researchers who are interested in the data collected by the pilot sensor array as well as those who have additional sensors to install on the bridge. Current collaborators include:
    1. Professors Kincho Law (Stanford University) and Jerome Lynch (University of Michigan) for wireless data acquisition/transmission in which side-by-side comparisons with the wired sensors will be made.

   2. Professor Magda El Zarki (University of California, Irvine) for data transmission over power lines.
   3. Professor Mohan Trivedi (University of California, San Diego) for image acquisition, image analysis, and assistance in finalizing the camera types and locations.
   4. Professor Ramesh Rao (University of California, San Diego) for air quality and airborne hazard monitoring.

- The currently installed sensor network on the Voigt Bridge testbed provides data suitable for performing system identification as well as for exploring vehicle/structure dynamics. Amongst other possibilities, this data is applicable for:
   1. Determining resonant frequencies, damping ratios, and mode shapes. Changes in these values will be tracked against changes in the environment, including ambient air temperature, temperature of the structural components, and relative humidity.
   2. Calibrating and updating finite element models based on measured data and extracted features [Feng et al., 2004; Feng and Bahng, 1999].
   3. Development and testing of local processing and on-site analysis algorithms with applications in data reduction and event detection.
- From the data recorded in the system shakedown test performed on January 7, 2006, the following observations were made:

1. Traffic passing underneath the bridge was observed to cause vibrations with amplitudes comparable to those from traffic crossing on the bridge. This phenomenon along with possible implications for fatigue analysis will need to be studied further.
2. Under low level excitation (from cars and other passenger vehicles), traffic induced vibrations were confined to the span the vehicle was on, and the bridge vibrates predominantly at its resonant frequencies.
3. When a large vehicle (campus shuttle bus or large truck) passes over the overcrossing, the bridge vibrates at forced frequencies. Large vibrations were also observed in neighboring spans, not just the one the vehicle is situated on. When these vehicles move off of the bridge, the bridge quickly returns to the preferred resonant frequencies (free vibration).

10.2 Cited Publications

Finally, the research presented in this dissertation has been used in the following works so far:

Chang, R., Gandhi, T., and Trivedi, M. M. (2004). "Vision Modules for a Multi-Sensory Bridge Monitoring Approach." *7th International IEEE Conference on Intelligent Transportation Systems (ITSC 2004)*, Washington, DC, October 3-6.

Elgamal, A., Fraser, M., McMartin, F. (2005). “On-line Educational Shake Table Experiments.” *Journal of Professional Issues in Engineering Education & Practice*, ASCE.

Elgamal, A., Fraser, M., and Zonta, D. (2005). “Webshaker: Live Internet Shake-Table Experiment for Education and Research.” *Computer Applications in Engineering Education*.

Elgamal, A., Conte, J.P., Yan, L., Fraser, M., Masri, S.F., El Zarki, M., Fountain, T., and Trivedi, M. (2004). "A Framework for Monitoring Bridges and Civil Infrastructure." (Invited Paper) *Proc. of the 3rd China-Japan-US Symposium on Structural Health Monitoring and Control*, Dalian, China, October 13-16.

Elgamal, A., Conte, J.P., Fraser, M., Masri, S., Fountain, T., Gupta, A., Trivedi, M., and El Zarki, M. (2003). “Health Monitoring for Civil Infrastructure.” *Proc. of the 9th Arab Structural Engineering Conference, (9ASEC)*, Nov. 29 - Dec. 1, Abu Dhabi, UAE.

Elgamal, A., Conte, J.P., Masri, S., Fraser, M., Fountain, T., Gupta, A., Trivedi, M., and El Zarki, M. (2003). “Health Monitoring Framework for Bridges and Civil Infrastructure.,” *Proc. of the 4th International Workshop on Structural Health Monitoring*, Stanford, CA, September 15-17.

Elgamal, A., Fraser, M., and Pagni, C. (2002). “Internet Live Shake-Table Testing for Education in Earthquake Engineeering.” *Proc. of the International Conference on Engineering Education (ICEE-2002)*, Manchester, UK, August 18-22.

Elgamal, A., Zonta, D., and Fraser, M. (2001). “A Pilot Web-Shared Controlling and Monitoring System for Real-Time Assessment of Civil Engineering Structures.” *Proc. of the 2nd ECCE Symposium on Information and Communication Technology in the Practice of Building and Civil Engineering*, Espoo, Finland.

Fraser, M., Yan, L., He, X., Elgamal, A., Conte, J.P., and Fountain, T. (2004). "Simple Neural Network Application for Traffic Monitoring." *Proc. of the International Conference on Computaional & Experimental Engineering and Sciences (ICCES'04)*, Madeira, Portugal, July 26-29.

Fraser, M., Elgamal, A,. Oliver, K., and Conte, J.P. (2004). “Data Fusion Application For Health Monitoring.” *Proc. of the 1st International Workshop on Advanced Smart Materials and Smart Structures Technology,* Honolulu, HI, January 12-14.

Fraser, M., Elgamal, A., Conte, J.P., Masri, S., Fountain, T., Gupta, A., Trivedi, M., and El Zarki, M. (2003). “Elements of an Integrated Health Monitoring Framework.” *Proc. of the 2003 SPIE Conference on Nondestructive Evaluation for Health Monitoring and Diagnostics*, San Diego, CA, March 2-6.

Fraser, M., Elgamal, A., Conte, J.P., Karbhari, V., Zhao, L., and Seible, F. (2002) "Pilot Health Monitoring Framework for Composite Bridge-Deck Panels." *Proc. of the Third Middle East Symposium on Structural Composites for Infrastructure Applications (MESC-3),* Aswan, Egypt, Dec 17-20.

Fraser, M., Elgamal, A., Zonta, D. (2002). "The Webshaker Pilot Project: An Internet Framework For Real-Time Monitoring and Control Systems of Civil Engineering Structures." *Proc. of the 3rd World Conference on Structural Conference*, Como, IT, April 7-12.

Fountain, T., Ding, L., Cotofana, N. (2004) "Web Services in Sensor Networks, The TeraBridge ITR Project." SIO, La Jolla, CA.

Gandhi, T. L., Cheng, R, and Trivedi, M. M. (2006). "Video and Seismic Sensor Based Structural Health Monitoring: Framework, Algorithms, and Implementation." *IEEE Transactions on Intelligent Transportation Systems*, in print.

Trivedi, M. M., Gandhi, T. L., and Huang, K. S. (2005). "Distributed Interactive Video Arrays for Event Capture and Enhanced Situational Awareness." *IEEE Intelligent Systems, Special Issue on AI in Homeland Security*, Volume 20, Issue 5, Sept.-Oct. pp 58 - 66.

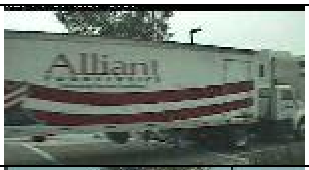
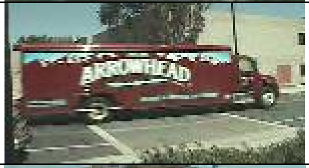
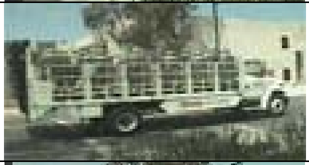
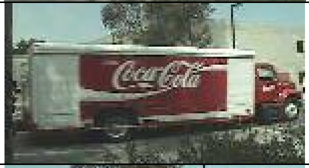
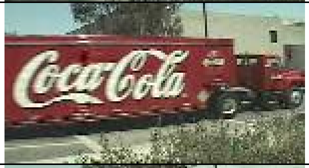
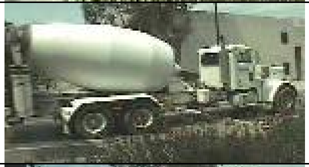
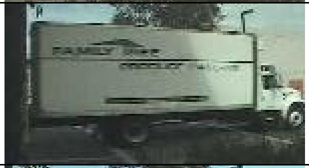
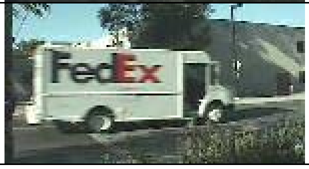
Trivedi, M. M., Prati, A., and Kogut, G. (2002). "Distributed Interactive Video Arrays for Event Based Analysis of Incidents." *5th International IEEE Conference on Intelligent Transportation Systems*, Singapore, pp. 950–956

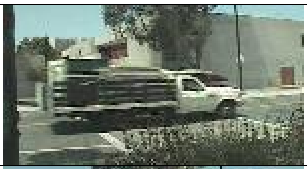
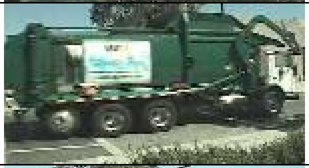
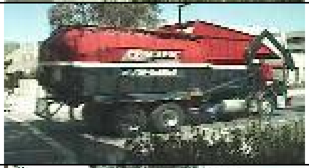
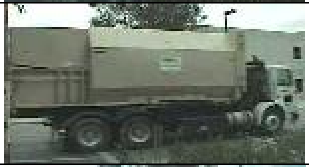
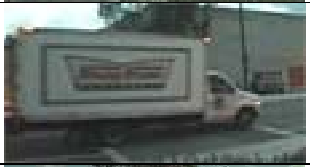
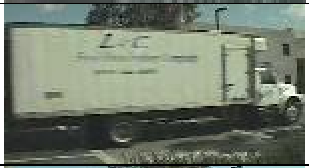
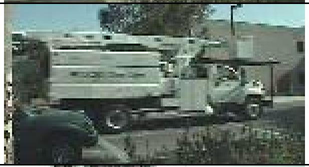
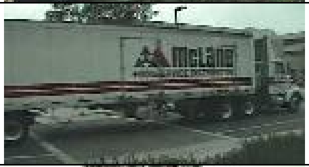
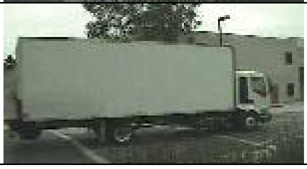
Yan, L., Fraser, M., Oliver K., Elgamal A., Conte J. P., and Fountain T. (2005). "Traffic Pattern Recognition using an Active Learning Neural Network and Principal Component Analysis." *Proc. of the 8th International Conference on the Application of Artificial Intelligence to Civil, Structural and Environmental Engineering*, August 30 – September 2, Rome, Italy.

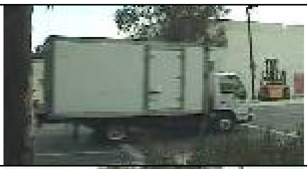
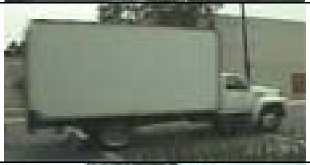
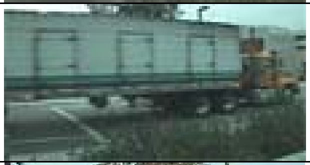
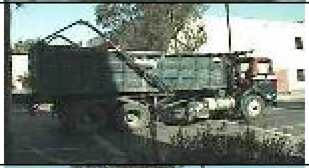
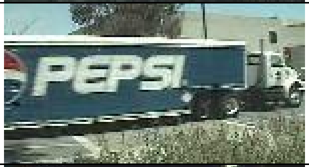
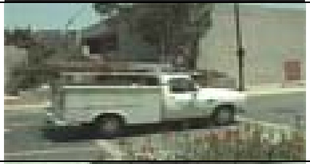
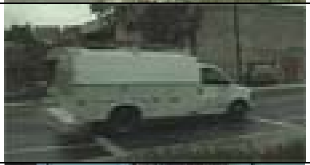
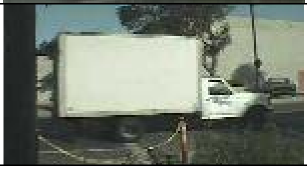
Yan, L., Fraser, M., Elgamal, A., and Conte, J.P. (2004). "Applications of Neural Networks in Structural Health Monitoring." *Proc. of the 3rd China-Japan-US Symposium on Structural Health Monitoring and Control*, Dalian, China, October 13-16.

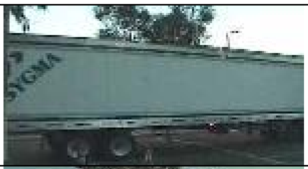
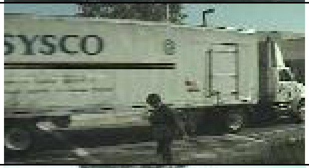
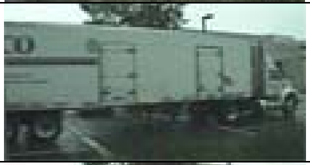
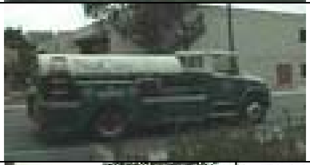
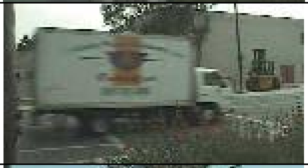
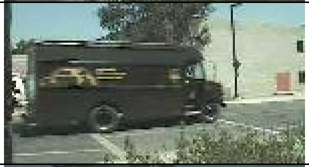
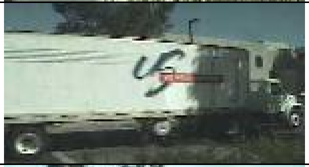
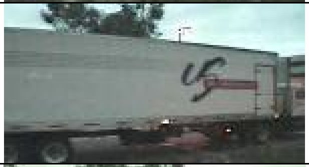
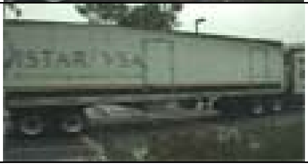
Zonta, D., Elgamal, A., Fraser, M., and Priestley, M. J. N. (2002). "A Vibrational Based Methodology for After-Earthquake Damage Assessment of Multistory Frame-Resistant Buildings." *Proc. of the 12th European Conference on Earthquake Engineering*, London, UK, September 9-13.

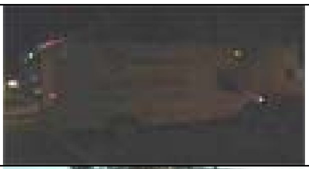
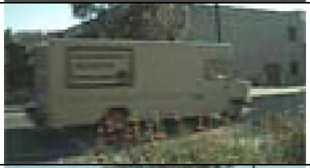
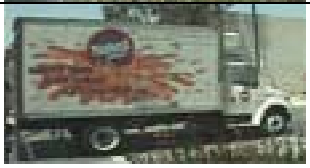
Appendix 1: Fifty Seven Vehicle Type Images for 2-Channel Database (Prepared in Collaboration with Kendra Oliver)

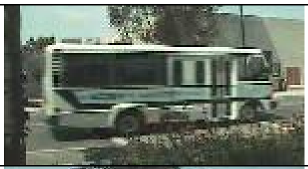
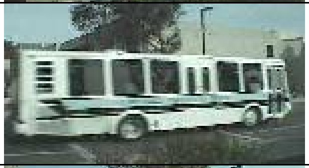
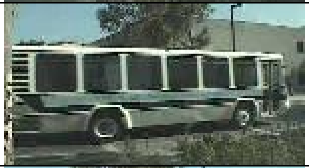
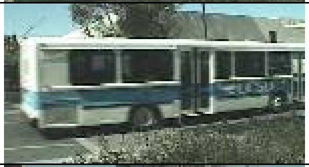
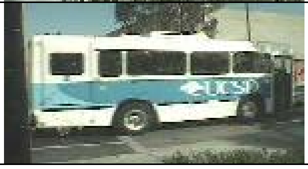
Vehicle Type	Typical Vehicle Image
TRUCKS	
Alliant Trucks	
Arrowhead Trucks	
Chemical Gas Trucks	
Coca-Cola Truck Type 1	
Coca-Cola Truck Type 2	
Concrete Mixer Trucks	
Family Tree Trucks	
FedEx Trucks	

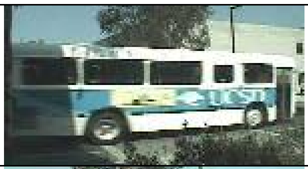
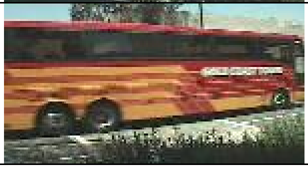
Gardening Trucks	
Green Garbage Trucks	
Red & Black Garbage Trucks	
Tan Garbage Trucks	
Krispy Kreme Trucks	
L & C Trucks	
Lift Truck	
McLane Trucks	
Medium White Trucks Type 1	

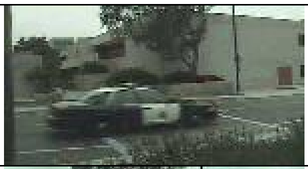
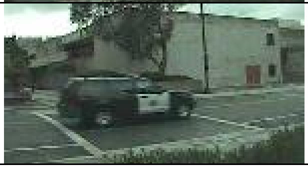
Medium White Trucks Type 2	
Medium White Trucks Type 3	
Multifoods Trucks	
Pacific Waste Trucks 2	
Pepsi Trucks	
Ryder Trucks	
Service Pick-up Trucks	
Service Vans	
Small White Trucks	

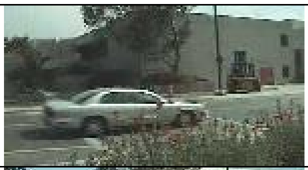
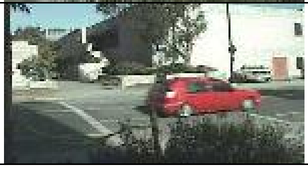
Sygma Truck	
Sysco Trucks Type 1	
Sysco Trucks Type 2	
Tank Trucks	
Tortilla Trucks	
UPS Trucks	
US Trucks Type 1	
US Trucks Type 2	
Vistar/VSA Trucks	

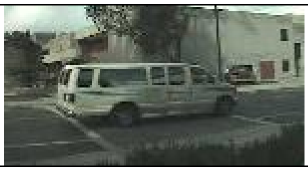
Trucks, *Type 134*	
Trucks, *Type 135*	
Other Trucks	 *Picture Varies with Truck Type*

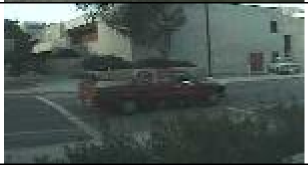
Vehicle Type	Typical Vehicle Image
Buses	
UCSD Bus Type 2	
UCSD Bus Type 5	
UCSD Bus Type 6	
UCSD Bus Type 7	
UCSD Bus Type 8	

UCSD Bus Type 9	
Other Buses	

Vehicle Type	Typical Vehicle Image
UCSD POLICE VEHICLES	
Police Cars	
Police SUVs	

Vehicle Type	Typical Vehicle Image
SMALL PERSONAL VEHICLES	
Cars	
Wagons	

Vehicle Type	Typical Vehicle Image
LARGE PERSONAL VEHICLES	
Vans	

SUVs	
Pick-up Truck	

Vehicle Type	Typical Vehicle Image
UNIQUE VEHICLES	
Forklifts	
Golf Carts	
Unique 5 Axle Trucks	
Fire Trucks	
Other Unique Vehicles	 *Picture Varies with Truck Type*

Table Appendix 1: Vehicle Type Profiles

Vehicle Type	Number of Vehicles Recorded	Number of Axles	Axle Spacing
TRUCKS			
Alliant Trucks	26	4	~
Arrowhead Trucks	14	2	✓
Chemical Gas Trucks	10	2	✓
Coca-Cola Trucks Type 1	7	2	✓
Coca-Cola Trucks Type 2	8	3	~
Concrete Mixer Trucks	14	3	~
Family Tree Trucks	14	2	✓
FedEx Trucks	44	2	✓
Gardening Trucks	20	2	~
Green Garbage Trucks	19	3	✓
Red & Black Garbage Trucks	108	3	✓
Tan Garbage Trucks	22	3	✓
Krispy Kreme Trucks	86	2	✓
L & C Trucks	18	2	✓
Lift Trucks	7	2	✓
McLane Trucks	113	5	~
Medium White Trucks Type 1	42	2	~
Medium White Trucks Type 2	21	2	~
Medium White Trucks Type 3	34	2	~
Multifoods Trucks	36	5	~
Pacific Waste Trucks 2	6	3	~

Pepsi Trucks	19	3	✓
Ryder Trucks	40	2	✓
Service Pick-up Trucks	55	2	
Service Vans	29	2	✓
Small White Trucks	97	2	~
Sygma Trucks	69	5	~
Sysco Trucks Type 1	35	3	~
Sysco Trucks Type 2	25	4	~
Tank Trucks	6	2	✓
Tortilla Trucks	49	2	✓
UPS Trucks	3	2	✓
US Trucks Type 1	17	3	~
US Trucks Type 2	26	4	~
Vistar/VSA Trucks	44	5	~
Trucks, *Type 134*	61	2	✓
Trucks, *Type 135*	15	2	✓
Other Trucks	527	Varies	✗
BUSES			
UCSD Buses Type 2	1	2	✓
UCSD Buses Type 5	497	2	✓
UCSD Buses Type 6	1	2	✓
UCSD Buses Type 7	10	2	✓
UCSD Buses Type 8	100	2	✓
UCSD Buses Type 9	1215	2	✓
Other Buses	44	Varies	✗
LARGE PERSONAL VEHICLES			
Vans	742	2	✗

SUVs	418	2	✗
Pick-up Trucks	441	2	✗
UCSD POLICE VEHICLES			
Police Cars	488	2	✓
Police SUVs	103	2	✓
SMALL PERSONAL VEHICLES			
Cars	350	2	✗
Wagons	13	2	✗
UNIQUE VEHICLES			
Forklifts	11	2	✓
Golf Carts	4	2	✗
Unique 5-Axle Trucks	2	5	✓
Fire Trucks	9	Varies	✗
Other Unique Vehicles	33	Varies	✗

✓ = Identical
~ = Similar
✗ = Different

References

Abdel-Ghaffar A.M. and Housner G.W. (1977). “An Analysis of the Dynamic Characteristics of a Suspension Bridge by Ambient Vibration Measurements.” *Report EERL, 77- 01*, Earthquake Engineering Research Laboratory, California Institute of Technology.

Achler, O. and Trivedi, M. (2004). "Camera Based Vehicle Detection, Tracking, and Wheel Baseline Estimation Approach." *7th International IEEE Conference on Intelligent Transportation Systems* (ITSC 2004), Washington, DC, October 3-6.

Aktan, A. E., Pervizour, M., Catbas, F. N., Barrish, R., Grimmelsman, K., Qin, X., Kulcu, E., Ciloglu, K., Curtis, J., von Haza-Radlitz, G. (2000). “Integrated Field, Theoretical and Laboratory Research for Solving Large Systems Identification Problems.” *Proc. of the International Conference on Advances in Structural Dynamics,* Hong Kong, Dec. 13-15.

Aktan, A. E., Pervizpour, M., Catbas, N., Grimmelsman, K., Barrish, R., Curtis J., and Qin. X. (2001a). “Information Technology Research For Health Monitoring Of Bridge Systems.” *Proc. of the Structural Faults & Repair 2001 Conference*, London, UK, July 4-6.

Aktan, E., Chase, S., Inman, D., and Pines, D. (2001b). “Monitoring and Managing the Health of Infrastructure Systems,” *Proc. of the 2001 SPIE Conference on Health Monitoring of Highway Transportation Infrastructure*, Irvine, CA, March 6-8.

Asbury, S., Glover, M., Humphreys, A., Weiss, E., Mathews, J., and Sol, S. (1997). PERL 5 How-To Second Edition. *Waite Group Press*, Corte Madera, CA.

Bishop C.M. (1995). Neural Networks for Pattern Recognition, *Oxford University Press*.

Bock, Y., Nikolaidis, R. M., and de Jonge, P. J. (2000). “Instantaneous Geodetic Positioning at Medium Distances with the Global Positioning System.” *Journal of Geophysical Research*, Vol. 105, No. B12, Pg 28, 223-28, 253. December.

Bock, Y., Panagiotou, M., Yang, F., Restrepo, J. I., Conte J. (2006). “Shake Table Tests of a Full Scale Reinforced Concrete Wall Building: Real Time 50Hz GPS Displacement Measurements.” *Proc. of SSA Centennial Annual Meeting,* San Francisco, April 18-22.

Broquet, C., Bailey, S. F., Bru¨hwiler, E., and Fafard, M. (1999). ‘‘Dynamic Amplification Factors in Deck Slabs of Concrete Road Bridges.’’ Proc. of the 4th European Conf. on Structural Dynamics (EURODYN), L. Fryba and J. Naprstek, eds., Balkema, Rotterdam, The Netherlands, 795–799.

Broquet, C., Bailey, S., Fafard, M, and Brühwiler, E. (2004). “Dynamic Behavior of Deck Slabs of Concrete Road Bridges.” ASCE Journal of Bridge Engineering, Vol. 9, No 2, pp. 137-146.

Chen, Y. and Feng, M. Q. (2006). “Modeling of Traffic Excitation for System Idntification of Bridge Structures.” *Computer-Aided Civil and Infrastructure Engineering*, Vol. 21, pp. 57-66.

Chopra, Anil K. (2000). Dynamics of Structures: theory and application to earthquake engineering. *Prentice Hall*, New Jersey.

Conte, J. P., Elgamal, A., Masri, S., Fraser, M., Fountain, T., Gupta, A., Trivedi, M., and El Zarki, M. (2003). “Health Monitoring Framework and Structural Analyses,” *UCSD and Caltrans Workshop on Structural Health Monitoring and Diagnostics of Bridge Infrastructures*. March 7.

Conte, J. P., Elgamal, A., Todd, M., Van Den Einde, L., and Fountain, T. (2006). “Acquisition of Dense Sensor Network and Cyberinfrastructure for Auscultation of Bridges and Civil Infrastructure.” *National Science Foundation Major Research Instrumentation Program (MRI) Proposal*.

Doebling, S. W., Farrar, C. R., Prime, M. B., and Shevitz, D. W. (1996). “Damage Identification and Health Monitoring of Structural and Mechanical Systems from Changes in Their Vibration Characteristics: A Literature Review.” *Report No: LA-13070-MS*, Los Alamos National Laboratory.

Elgamal, A., Zonta, D., and Fraser, M. (2001). “A Pilot Web-Shared Controlling and Monitoring System for Real-Time Assessment of Civil Engineering Structures.” *Proc. of the 2nd ECCE Symposium on Information and Communication Technology in the Practice of Building and Civil Engineering*, Espoo, Finland.

Elgamal, A., Conte, J. P., El Zarki, M., Fountain, T., Masri, S., and Trivedi, M. (2002). “ITR: Collaborative Research: An Integrated Framework for Health Monitoring of Highway Bridges and Civil Infrastructure.” *National Science Foundation Information Technology Research (ITR) Proposal*.

Elgamal, A., Conte, J. P., Fraser, M., Masri, S., Fountain, T., Gupta, A., Trivedi, M., and El Zarki, M. (2003a). “Health Monitoring for Civil Infrastructure.” *Proc. of the Ninth Arab Structural Engineering Conference (9ASEC)*, Abu Dhabi, UAE, Nov. 29-Dec. 1.

Elgamal, A., Conte, J.P., Masri, S., Fraser, M., Fountain, T., Gupta, A., Trivedi, M., and El Zarki, M. (2003b). “Health Monitoring Framework for Bridges and Civil Infrastructure.” *Proc. of the 4th International Workshop on Structural Health Monitoring*, Stanford, CA, September 15-17.

Elgamal, A., Conte, J. P., El Zarki, M., Fountain, T., Masri, S., and Trivedi, M. (2003c). “ITR: Collaborative Research: An Integrated Framework for Health Monitoring of Highway Bridges and Civil Infrastructure - NSF Annual Report - Activities and Findings 10/02 to 6/03.” *National Science Foundation Information Technology Research (ITR) Proposal Annual Report*.

Elgamal, A., Conte, J. P., Yan, L., Fraser, M., Masri, S.F., El Zarki, M., Fountain, T., and Trivedi, M. (2004). "A Framework for Monitoring Bridges and Civil Infrastructure." (Invited Paper) *Proc. of the 3rd China-Japan-US Symposium on Structural Health Monitoring and Control*, Dalian, China, October 13-16.

Elgamal, A., Fraser, M., and McMartin, F. (2005a). “On-Line Educational Shake Table Experiments.” *Journal of Professional Issues in Engineering and Practice*, ASCE, 132, 41.

Elgamal, A., Fraser, M., and Zonta, D. (2005b). “Webshaker: Live Internet Shake-Table Experiment for Education and Research.” *Computer Applications in Engineering Education*, Volume 13, Issue 1, Pg. 99-110.

Farrar, C. R., and Jauregui, D. (1996). “Damage Detection Algorithms Applied to Experimental and Numerical Modal Data from the I-40 Bridge.” *Report No: LA-13074-MS*, Los Alamos National Laboratory.

Farrar, C. R. and Doebling, S. W. (1997). “Lessons Learned from Applications of Vibration-Based Damage Identification Methods to a Large Scale Bridge Structure.” *Proc. of the First International Workshop on Structural Health Monitoring; Current Status and Perspectives*, 351-370, Stanford, CA, September 18-20.

Feng, M. Q. and Bahng , E. Y. (1999). “Damage Assessment of Bridges with Jacketed RC Columns using Vibration Test.” *Proc. of the SPIE Conference on Smart Systems for Bridges, Structures, and Highways*, Newport Beach, CA.

Feng, M. Q., Kim, D. K., Yi, J.-H., and Chen, Y. (2004). “Baseline Models for Bridge Performance Monitoring.” *Journal of Engineering Mechanics*, ASCE, Vol. 130, No.5, pp. 562-569.

Foda, M. A. and Abduljabbar, M. A. (1998). “A dynamic green function formulation for the response of a beam structure to a moving mass.” Journal of Sound and Vibration 210(3), 295-306.

Fraser, M., Elgamal, A., Conte, J.P., Karbhari, V., Zhao, L., and Seible, F. (2002) “Pilot Health Monitoring Framework for Composite Bridge-Deck Panels.” *Proc. of the Third Middle East Symposium on Structural Composites for Infrastructure Applications (MESC-3),* Aswan, Egypt, Dec 17-20.

Fraser, M., Elgamal, A., Conte. J. P., Masri, S., Fountain, T, Gupta, A, Trivedi, M., and El Zarki, M. (2003). “Elements of an Integrated Health Monitoring Framework.” *Proc. of the 2003 SPIE Conference on Nondestructive Evaluation for Health Monitoring and Diagnostics*, San Diego, CA, March 2-6.

Fraser, M., Elgamal, A., Oliver, K., Conte, J. P. 2004. "Data Fusion Application for Health Monitoring." *Proc. of the 1st International Workshop on Advanced Smart Materials and Smart Structures Technology,* Honolulu, HI, January 12-14.

Friedman, N. and Russell, S. (2001). “Image Segmentation in Image Sequences: A Probabilistic Approach.” *Proc. of the Thirteenth Conference on Uncertainty of Artificial Intelligence,* Aug.

Garret, J. H., Jr., Ghaboussi, J., Xu, X., and Ranjithan, S. (1992). “Neural Networks and their Application to Civil Engineering.” *Expert System in Civil Engineering – Knowledge Representation,* Allen, R. H., ed., Expert Systems Committee of ASCE, New York.

Gatling, G. (2005). “NIST Update Windows 2000 Clock,” On-line Code from National Instruments. (ftp://ftp.ni.com/contrib/epd/B45EACE3DEE856A4E034080020E74861/NIST_Update_W2k_Clock_.vi).

Ghaboussi, J. (1993). “An Overview of the Potential Applications of Neural Networks in Civil Engineering.” *Proc. of the Structures Congress ’93,* ASCE, Irvine, California, April 19-21.

Ghaboussi, J. (1994). “Some Applications of Neural Networks in Structural Engineering.” *Proc. of the Structures Congress ’94, ASCE,* Atlanta, Georgia, April 24-28.

Gaun, H., Karbhari, V. M., and Sikorsky C. S. (2006). “Web-Based Structural Health Monitoring of an FRP Composite Bridge.” *Computer-Aided Civil and Infrastructure Engineering* **21** pp 39-56.

Hand D., Mannila H., Smyth P. (2001). Principles of Data Mining, MIT Press, Cambridge, MA, ISBN 0-262-08290-X

He, X., Moaveni, B., Conte, J.P., Elgamal, A., Masri, S., Caffrey, J.P., Tasbihgoo, F., Whang, D.H. (2005). “System Identification of New Carquinez Bridge Using Ambinet Vibration Response Data.” *Proc. of the International Conference on Experimental Vibration Analysis for Civil Engineering Structures (EVACES 05),* Christian Crémona, ed., 477-488, Bordeaux, France, October 26-28.

ISIS Canada. (2000). “Strengthening Reinforced Concrete Structures with Externally Bonded Fibre Reinforced Polymers”. *ISIS Canada Design Manual*, Spring, University of Manitoba, Winnipeg, Manitoba, Canada.

Koptenko, S. (2003). “Local Min, Max, Nearest Neighbour.” Resonant Medical Inc., On-line code http://www.mathworks.com.

Law, K. H., Lynch, J. P., Kiremidjian, A. S., Kenny, T. W., and Carver, E., (2005). “The Design of a Wireless Sensing Unit for Structural Monitoring,” *Proc. of The Third Workshop on Structural Health Monitoring,* Stanford University, Stanford CA, September 12-14.

Liu, P. L. and Sun, S. C. (1997). “The Application of Artificial Neural Networks on the Health Monitoring of Bridges.” *Proc. of the First International Workshop on Structural Health Monitoring: Current Status and Perspectives*, 103-110, Stanford, CA, September 18-20.

Liu, S. C. (2001). *Proc. of the 2001 SPIE Conference on Health Monitoring of Highway Transportation Infrastructure,* SPIE, Newport Beach, CA, March 6-8.

Livingston, R. (1999). “FHWA Fiber-Optics Research Program: Critical Knowledge for Infrastructure Improvement.” *Public Roads,* Vol. 63 No. 1, July / August 1999, Federal Highway Administration, Washington, D.C.

Livingston, R. (2004). “The Status of Smart Bridges in the US Federal Highway System.” *Proc of the 1st International Workshop on Advanced Smart Materials and Smart Structures Technology,* Honolulu, Hawaii, January 12-14.

Lus H., Betti R., Longman R.W. (1999). “Identification of Linear Structural Systems using Earthquake-Induced Vibration Data.” *Journal of Earthquake Engineering and Structural Dynamics*, 28:1449 –1467.

Lynch, J. P. and Loh, K. (2005). "A Summary Review of Wireless Sensors and Sensor Networks for Structural Health Monitoring." *Shock and Vibration Digest*, Sage Publications, submitted.

Lynch, J. P. (2005). "An Overview of Wireless Structural Health Monitoring for Civil Structures." *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, The Royal Society, London, in press.

Masri S. F., Smyth, A. W., Chassiakos, A. G., and Caughey, T. K. (1996). “Neural Network Approach to Detection of Changes in Structural Parameters.” *ASCE, Journal of Engineering Mechanics*, 122(4), 350-360.

Mazzoni, S., McKenna, F., and Fenves, G. (2005). “OpenSees Command Language Manual.” (http://opensees.berkeley.edu).

McKenna, F., and Fenves, G. L. (2000). “An Object-Oriented Software Design for Parallel Structural Analysis.” *Proc. of the SEI/ASCE Structures Congress*, Philadelphia, PA, (http://opensees.berkeley.edu).

Melhem, H. G., and Cheng, Y. (2003). “Prediction of Remaining Service Life of Bridge Decks Using Machine Learning.” *ASCE Journal of Computing in Civil Engineering*, Vol. 17, No. 1, pp. 1-9.

Mikik, I., Cosman, P., Kogut, G., and Trivedi, M. M. (2000). "Moving Shadow and Object Detection in Traffic Scenes," *International Conference on Pattern Recognition*, Barcelona, Spain, September 3-8.

Nakamura, M., Masri, S. F., Chassiakos, A. G., and Caughey, T. K. (1998). “A Method for Non-Parametric Damage Detection Through the Use of Neural Networks.” *Earthquake Engineering and Structural Dynamics*, 27, 997-1010.

Oh, S., Ritchie, S. G., and Oh, C. (2001). “Real Time Traffic Measurement from Single Loop Inductive Signatures.” *Institution of Transportation Studies, University of California, Irvine. ITS Working Paper Series*, UCI-ITS-WP-01-15.

Popov, E. P. (1976). Mechanics of Materials. *Prentice-Hall*, New Jersey, 1976.

Prati, A., Mikic, I., Trivedi, M. M., and Cucchiara, R. (2003). "Detecting Moving Shadows: Algorithms and Evaluation." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 918-923.

Raz, O., Buchheit, R., Shaw, M., Koopman, P., and Faloutsos, C. (2004). “Detecting Semantic Anomalies in Truck Weigh-in-Motion Traffic Data Using Data Mining.” *ASCE Journal of Computing in Civil Engineering*, Vol. 18, No. 4, pp. 291-200.

Ridder, C., Munkelt, O., and Kirchner, H. (2000). “Adaptive Background Estimation and Foreground Detection using Kalman Filtering.” *Proc. of International Conference on recent Advances in Mechatronics,* 193-199.

Russell, S. J. and Norvig, P. (1994). *Artificial Intelligence, A Modern Approach*, Prentice Hall, Upper Saddle River, New Jersey.

Saadeghvaziri, M. A. (1993). “Finite Element Analysis of Highway Bridges Subjected to Moving Loads.” *Computers & Structures*, Vol. 49 No. 5, 837-842

Schulz, W.L., Conte, J.P., Udd, E, and Seim, J.M. (2001). “Static and Dynamic Testing of Bridges and Highways Using Long-Gage Fiber Bragg Grating Based Strain Sensors.” *Proc. of SPIE*, Vol. 4330, p. 56.

Smyth, A.W., Pei, J., and Masri, S.F. (2003). “System Identification of the Vincent Thomas Suspension Bridge Using Earthquake Records.” *Journal of Earthquake Engineering and Structural Dynamics*, 32:339-367

Sohn, H., Farrar, C. R., Hunter, N. F., and Worden, K. (2001). “Structural Health Monitoring Using Statistical Pattern Recognition Techniques.” *Journal of Dynamic Systems, Measurement, and Control*, 123, 706-711.

Sohn, H and Farrar, C. R. (2001). “Damage Diagnosis Using Time Series Analysis of Vibration Signals.” *Smart Materials and Structures*, 2001(10), 446-451,

Sohn, H., Farrar, C. R., Hemez, F., Czarnecki, J. J., Shunk, D., Stinemates, D., and Nadler, B. (2003). “A Review of Structural Health Monitoring Literature from 1996-2001.” *Report No: LA-13976-MS*, Los Alamos National Laboratory.

Stalling, J. M., Tedesco, J. W., El-Mihilmy, M., and McCauley, M. (2000). “Field Performance of FRP Bridge Repair.” *ASCE, Journal of Bridge Engineering*, 5/2, 107-113.

Talbot, M. and Stoyanoff, S. (2005). “Full Scale Modal Measurements of the Ile d’Orleans Suspension Bridge.” *Proc. of the International Conference on Experimental Vibration Analysis for Civil Engineering Structures (EVACES 05),* Christian Crémona, ed., 477-488, Bordeaux, France, October 26-28.

Todd, M. D., Chang, C. C., Johnson, G. A., Vohra, S. T., Pate, J. W., and Idriss, R. L. (1999). "Bridge Monitoring Using a 64-channel Fiber Bragg Grating System," *Proc. of 17th Int. Modal Analysis Conf.*, 1719-1725, Orlando, FL.

Turner-Fairbanks Highway Research Center (2005). “Bridge Study Analyzes Accuracy of Visual Inspections,” *http://www.tfhrc.gov/focus/jan01/bridge_study.htm*.

Wahbeh, M., Tasbihgoo, F., Yun, H., Masri, S. F., Caffery, J. P., and Chassiakos, A. G. (2005). “Real-Time Earthquake Monitoring of Large-Scale Bridge Structures.” *Proc. of the 5th International Workshop on Structural Health Monitoring (5th IWSHM)*, pg 190-197, Stanford University, Stanford CA Sept 12-14\.

Wahbeh, A. M., Caffrey, J. P., and Masri, S. M. (2003). “A Vision-based Approach for the Direct Measurement of Displacements in Vibrating Systems.” Journal of Smart Materials and Structures, Vol. 12, pp 785-794.

Wang, M. L. 2005. “Damage Assessment and Monitoring of Long-Span Bridges.” *Proc of the 5th International Workshop on Structural Health Monitoring (5th IWSHM)*, pg 61-79, Stanford University, Stanford CA Sept 12-14.

Worden, K., Farrar, C. R., Manson, G., and Park, G. (2005). “Fundamental Axioms of Structural Health Monitoring.” *Proc of the 5th International Workshop on Structural Health Monitoring (5th IWSHM)*, pg 26-41, Stanford University, Stanford CA Sept 12-14.

Xu, B., Wu, Z. S., and Yokoyama, K. (2002). “Response Time Series Based Structural Parametric Assessment Approach with Neural Networks.” *Proc. of the First International Conference on Structural Health Monitoring and Intelligent Infrastructure (SHMII-1’2003)*, Wu & Abe, eds., 601-609, Tokyo, Japan, November 13-15.

Yan, L., Elgamal, A., Fountain, T., Conte J. P., and Masri, S. F. (2004a). “A PCA-Based Neural Network Approach for Multi-Storey Building Damage Identification.” *Proc. of International Conference on Future Vision and Challenges of Urban Development*, December 20-22, Cairo, Eqypt.

Yan, L., Elgamal, A., Fountain, T., Conte J. P., and Masri, S. F. (2004b). "System Identification for Bridge Foundation Integrity." *Proc. of the International Conference on Geotechnical Engineering*, May, Beirut, Lebanon.

Yan, L., Fraser, M., Elgamal, A., and Conte, J. P. (2004c). "Application of Neural Networks in Structural Health Monitoring." *Proc. of the 3rd China-Japan-US Symposium on Structural Health Monitoring and Control*, October 13-16, Dalian, China.

Yan, L., Fraser, M., Oliver, K., Elgamal, A, Conte, J.P., and Fountain, T. (2005). "Traffic Pattern Recognition using and Active Learning Neural Network and Principal Component Analysis." *Proc. of the Eighth International Conference on the Application of Artificial Intelligence to Civil, Structural, and Environmental Engineering (AICC 2005)*, Rome, Italy, August 30 – September 2.

Yan, L. (2006). "Statistical Inference Approaches for Structural Damage Detection." *Ph.D. Dissertation*, Department of Structural Engineering, University of California at San Diego, La Jolla, CA.

Zhao, L. 1999. "Characterization of Deck-to-Girder Connections in FRP Composite Superstructures." *Ph.D. Dissertation*, Department of Structural Engineering, University of California at San Diego, La Jolla, CA.