INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI[®]

ARCHIVAL AND PROCESSING OF STRUCTURAL HEALTH MONITORING DATA

By

Hasan Tahir Qureshi

B.Sc. Civil Engineering (Honors)

University of Engineering and Technology, Lahore, Pakistan

A thesis submitted to the

Faculty of Graduate Studies and Research

in partial fulfillment of the requirements for the degree of

Master of Applied Science

June 2001

Department of Civil and Environmental Engineering*

Carleton University, Ottawa, Canada

Copyright © Hasan Tahir Qureshi, 2001

- * The Master of Applied Science program in Civil and Environmental Engineering is a joint program between Carleton University and the University of Ottawa, which is administered by the Ottawa-Carleton Institute for Civil Engineering.

**National Library
of Canada**

**Acquisitions and
Bibliographic Services**

**395 Wellington Street
Ottawa ON K1A 0N4
Canada**

**Bibliothèque nationale
du Canada**

**Acquisitions et
services bibliographiques**

**395, rue Wellington
Ottawa ON K1A 0N4
Canada**

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-68314-1

Canada

Department of Civil and Environmental Engineering

The undersigned hereby recommends to the
Faculty of Graduate Studies and Research
acceptance of the thesis

**ARCHIVAL AND PROCESSING OF
STRUCTURAL HEALTH MONITORING DATA**

Submitted by

Hasan Tahir Qureshi

in partial fulfillment of the requirements for the degree of
Master of Applied Science

Chair, Department of Civil and Environmental Engineering

Thesis Supervisors

Carleton University
Ottawa, Ontario, Canada

June 2001

ABSTRACT

The Canadian Network of the Centers of Excellence on Intelligent Sensing of Innovative Structures (ISIS) is responsible for the instrumentation and data collection from bridges constructed using innovative construction materials and/or innovative structural concepts. ISIS has initiated a project to create a central archive for data obtained from these instrumented bridges. The data received from the various sources now resides in a central relational database. A world-wide-web interface to the archive has been provided. The archive will enable interested researchers to browse and download this data, using many common formats. Knowledge of the integrity of an in-service structure on a continuous real-time basis will help the civil engineering community in improving the design and in minimizing the cost.

This thesis is mainly concerned with the archival of data collected from these instrumented bridges. Mechanisms for exploring the archive, methodologies for downloading of archived data, and means for updating the archive have been established. Different levels of user-access have also been established in order to facilitate the distribution of data to authorized persons only. Following the creation of relational database, the recorded dynamic data has been processed to find the natural frequency of vibration.

ACKNOWLEDGEMENTS

The author wishes to express his sincerest thanks to his supervisors Professor J. L. Humar and Professor N. Holtz for their guidance, encouragement and support throughout the course of this study.

The scholarship awarded to the author by Carleton University is gratefully acknowledged.

At the end, the author would like to express his sincerest thanks to his wife, parents and family members, teachers and friends for their affection and care extended to him.

**DEDICATED TO MY LOVELY WIFE
BELOVED PARENTS & FAMILY
RESPECTED TEACHERS &
GOOD FRIENDS
FOR ALL THEIR LOVE & SUPPORT**

TABLE OF CONTENTS

ABSTRACT	iii
ACKNOWLEDGEMENTS	iv
DEDICATION	v
TABLE OF CONTENTS	vi
LIST OF FIGURES	ix
LIST OF TABLES	x
LIST OF SYMBOLS	xi

CHAPTER 1 INTRODUCTION

1.1	GENERAL	1
1.2	INTELLIGENT PROCESSING OF DATA FOR DAMAGE DETECTION	5
1.3	LITERATURE REVIEW	7
1.4	OBJECTIVES AND SCOPE	11

CHAPTER 2 MONITORING PROGRAM

2.1	INTRODUCTION	14
2.2	DATA ACQUISITION SYSTEM	14
2.2.1	SENSORS	15
2.2.2	DATA ACQUISITION BOARDS	17
2.2.3	DATA ACQUISITION PROGRAM	17
2.3	CROWCHILD BRIDGE	18
2.3.1	DESCRIPTION OF THE BRIDGE AND TEST PROGRAM	18
2.3.2	INSTRUMENTATION AND DATA ACQUISITION SYSTEM	20
2.3.3	TESTS AND MEASUREMENTS	24
2.4	HEADINGLY BRIDGE (TAYLOR BRIDGE)	27
2.4.1	DESCRIPTION OF THE BRIDGE AND TEST PROGRAM	27
2.4.2	INSTRUMENTATION AND DATA ACQUISITION SYSTEM	29
2.5	SALMON RIVER BRIDGE	30
2.5.1	DESCRIPTION OF THE BRIDGE AND TEST PROGRAM	30
2.5.2	INSTRUMENTATION AND DATA ACQUISITION SYSTEM	31

2.6	WATERLOO CREEK BRIDGE	33
2.6.1	DESCRIPTION OF THE BRIDGE AND TEST PROGRAM	33
2.6.2	INSTRUMENTATION AND DATA ACQUISITION SYSTEM	35
2.7	MONITORING DATA INCLUDED IN THE ARCHIVE	36
2.7.1	CROWCHILD BRIDGE	36
2.7.2	TAYLOR BRIDGE	37
2.7.3	SALMON BRIDGE	38
2.7.4	WATERLOO CREEK BRIDGE	39
 CHAPTER 3 ARCHIVAL OF DATA		
3.1	INTRODUCTION	40
3.2	IMPORTANT ATTRIBUTES OF THE ARCHIVE	42
3.3	BENEFITS OF ARCHIVING	43
3.4	BACK UPS AND VALIDATION OF THE DATA	44
3.5	IMPORTANT OBJECTIVES OF ARCHIVING	45
3.6	INFORMATION PROVIDED BY THE ARCHIVE	48
3.6.1	DOCUMENTATION OF THE INSTRUMENTS	48
3.6.2	IMAGES SHOWING STRUCTURAL SYSTEM AND LOCATION OF SENSORS	49
3.6.3	DATA FILES CONTAINING SENSOR READINGS	49
3.7	FORMATS FOR DATA SUBMISSION	50
3.8	DATA ARCHIVE AND SOFTWARE TECHNOLOGY	54
3.8.1	PRINCIPLES GUIDING THE SELECTION OF SOFTWARE COMPONENTS	54
3.8.2	SELECTED SOFTWARE COMPONENTS	55
3.9	STRUCTURE AND ORGANIZATION OF THE DATABASES	57
3.9.1	TABLE "DataSets"	60
3.9.2	TABLE "Sensors"	60
3.9.3	TABLE "Glossary"	61
3.9.4	TABLE "Images"	62
3.10	ACCESS TO THE ARCHIVE	62
3.11	CLIENT – SERVER ARCHITECTURE	73
3.12	SOFTWARE ARCHITECTURE	79
3.13	CODES WRITTEN FOR THE ARCHIVE	81

CHAPTER 4	PROCESSING OF DATA	
4.1	INTRODUCTION	82
4.2	FORMATS FOR DOWNLOADING OF DATA	83
4.2.1	HTML	84
4.2.2	COMMA SEPARATED VALUES (CSV)	84
4.2.3	TAB DELIMITED VALUES	85
4.3	PROCESSING DYNAMIC DATA	85
4.3.1	OPERATIONAL MODAL ANALYSIS	87
4.3.2	EXPERT SYSTEM PERFORMANCE ANALYSIS SOFTWARE PACKAGE (ESPAN)	89
4.3.3	LMS CADA – X FOR PROCESSING OF DYNAMIC DATA	91
4.4	ANALYSIS OF DATA FOR SALMON RIVER BRIDGE	104
CHAPTER 5	SUMMARY, CONCLUSIONS AND FUTURE WORK	
5.1	SUMMARY AND CONCLUSIONS	111
5.2	FUTURE WORK AND RECOMMENDATIONS	113
REFERENCES		114
APPENDIX A		119
APPENDIX B		141

LIST OF FIGURES

CHAPTER 3 ARCHIVAL OF DATA

Figure 3.1	Layout of the Database	58
Figure 3.2	Access to the Archive	64
Figure 3.3	Opening Page of the Website	65
Figure 3.4	Image viewed by Registered Users	67
Figure 3.5	Data for Salmon River Bridge	68
Figure 3.6	Data of Table1, Salmon River Bridge	70
Figure 3.7	Data of Table Glossary, Salmon River Bridge	71
Figure 3.8	Data of Table Sensors, Salmon River Bridge	71
Figure 3.9	Client-Server Architecture	74

LIST OF TABLES

CHAPTER 2 MONITORING PROGRAM

Table 2.1	Crowchild Bridge, General Information	19
Table 2.2	Summary of the Tests and Measurements on the Crowchild Bridge	24
Table 2.3	Taylor Bridge, General Information	28
Table 2.4	Salmon River Bridge, General Information	30
Table 2.5	Waterloo Creek Bridge, General Information	34
Table 2.6	Monitoring Data Included in the Archive	36

CHAPTER 4 PROCESSING OF DATA

Table 4.1	Fundamental natural frequency of Salmon River Bridge	110
-----------	--	-----

LIST OF SYMBOLS

R = Initial resistance

ΔR = Resistance change

ϵ = Applied strain

C_{β} = Compensation coefficient of the sensor

C_{ϵ} = Strain calibration coefficient

C_t = Correction coefficient of temperature

ϵ_f = Measured strain reading at a given time

ϵ_l = Initial reference strain reading

ϵ_t = Measured temperature reading at a given time

ϵ_o = Initial reference temperature reading

1.1 GENERAL

Bridges are important lifeline structures for our society. It is therefore important that they are maintained in a good state of repair. Several factors may lead to damage of bridge structures. These include damage caused by dynamic loading such as seismic or wind excitation. Other factors responsible for the damage to bridges include creep effect, pre-stress losses, temperature fluctuations, permanent settlements, crack initiation and propagation. Early detection of damage in bridge structures is of obvious importance. This is particularly so for the reasons outlined below.

Aging infrastructure facilities, particularly those constructed from reinforced concrete, are deteriorating at a rapid rate. The rate of deterioration overwhelms the rate at which these structures are being repaired, rehabilitated or replaced by new ones. The US Federal Highway Administration (FHWA) has reported that over 135,000 (24%) highway bridges are rated as structurally deficient, and are definite candidates for repair, rehabilitation or replacement.

One major cause of deterioration in bridge infrastructure is the corrosion of steel, induced by environmental effects and de-icing salts. This has led the civil engineering community to consider the adoption of innovative designs and/or materials, which could mitigate the usual problems of deterioration resulting from corrosion.

One method that is gaining popularity in the construction industry involves the use of Advanced Composite Materials such as carbon, glass and fiber reinforced plastics. Advanced Composite Materials are being used in :

1. The repair and strengthening of reinforced beams and slabs.
2. Repairing of corrosion and earthquake damaged reinforced columns.
3. Development of corrosion resistant reinforcing bars and prestressing tendons.

An innovative design concept that has been motivated by the need to solve concrete deterioration due to reinforcement corrosion in bridge involves the use of steel-free deck.

Recent advances in the use of composite materials and innovative construction techniques have given rise to a need for careful assessment of the integrity of large civil structures. Many of these new design concepts and materials have not been as extensively tested and researched upon as the more conventional alternatives. Moreover their application is often beyond the scope of the current design codes.

Currently, evaluation of the health of civil engineering facilities such as bridges is usually performed at frequent intervals by means of site inspections. A disadvantage of this method of evaluation is that it does not have the ability to respond to potential problems in an efficient manner. Visual inspection is often unable to detect physically hidden structural damage. This may sometimes lead to catastrophic damage to the structure.

With current advancements in computation and communication methods, a significant amount of interest has evolved in developing new diagnostic technologies for intelligent monitoring of the health of a structure, for detecting damage, and for evaluating the integrity of the structure in real-time with minimum human involvement. This is relevant for both existing and new structures. The interest in intelligent monitoring has prompted the development of methods that can be used to identify the changing properties of instrumented structural systems through time.

As stated above, visual inspection may not always reveal the extent of damage and may also fail to detect milestone events in the life of the structure. These milestone events include crack initiation and propagation, and permanent settlements. The damage to a structure can have its origin in creep effect, pre-stress losses, influence of yearly temperature cycles, corrosion of steel, permanent settlements, de-lamination of fiber-reinforced plastics etc. Hence, the use of vibration based damage detection has gained popularity. It is now a well-known fact that any localized damage is responsible for reduction in stiffness and an

increase in the damping of a structure. Reduction in stiffness corresponds to a decrease in the natural frequency of a structure. This fact has been used by many researchers to detect damage.

Periodic monitoring by sensors can also be used to bridge the knowledge void and compensate for the lack of field experience related to the expected performance of the new innovative designs and/or materials adopted recently in bridge engineering. Use of distributed sensors to monitor the health condition of in-service structures becomes feasible if sensor signals can be interpreted accurately to reflect the in-situ condition of the structures through real-time data processing. Advancements in sensor technology offer unique opportunities in this regard and hence the entire system can be integrated and automated to perform real-time inspection and damage detection.

The essence of Structural Health Monitoring (SHM) technology is to develop autonomous systems for the continuous monitoring, inspection and damage detection of structures with minimum labor involvement. The results of structural conditions could be reported through a local network or to a remote center automatically.

In SHM, the change in reading recorded at the same location at two different times is used to identify the condition of the structure. Hence the time history data is crucial for this technique. The accuracy of the identification depends strongly upon the sensitivity of sensors and the interpretation algorithm. Assuming that

accurate sensors are available, the success of SHM will depend on the ability of the interpretation software. In addition it requires that a careful record be maintained of the sensor data. In a project of even modest size such data can be quite voluminous. Unless carefully archived, the interpretation of such data can become quite difficult, particularly because damage detection requires comparison of data obtained at frequent intervals during the history of the structure.

1.2 INTELLIGENT PROCESSING OF DATA FOR DAMAGE DETECTION

During the past two decades, a significant amount of research has been conducted in the area of non-destructive damage evaluation (NDE) based on the changes in the dynamic properties of a structure. All the NDE methods developed to-date can be classified into one of four levels according to their performance. These performance levels include :

1. Level-I. Methods that only identify if damage has occurred.
2. Level-II. Methods that identify if damage has occurred and simultaneously determine the location of damage.
3. Level-III. Methods that identify if damage has occurred, determine the location of damage as well as estimate the severity of the damage.

4. **Level-IV.** Methods that identify if damage has occurred, determine the location of damage, estimate the severity of damage, and evaluate the impact of damage on the structure.

The type of data collected for damage detection can be categorized as follows :

1. **Dynamic test data.** This data includes the measurement of accelerations and strains at a fast scan rate to monitor the response of structure to different excitation forces. The source of excitation forces can be earthquake, wind, moving water, passing traffic, pedestrians etc. Dynamic data is processed to find out the mode shapes and frequencies of the structure in response to different forces of excitation.
2. **Static test data.** This data includes the measurement of strains and displacements under controlled conditions. Loading by a controlled vehicle of known specifications is used to collect this data. The data is helpful in determining the lateral load distribution among girders, temperature induced stresses, and the behavior/effectiveness of pre-stressing forces.
3. **Environmental data.** This data includes the measurement of temperature, wind velocities, ice pressures etc. This data helps in determining the response of the structure to different environmental conditions.

The focus of the present work is to develop a methodology for the archival of the data described above. The work also covers processing of ambient vibration dynamic data to determine the natural frequencies of the system. A commercially available software for the analysis of ambient vibration data has been used for this purpose.

1.3 LITERATURE REVIEW

An archive is a collection of data or records pertaining to a specific source. Database is a generic term commonly used to describe a structured collection of data or records. Databases can take many forms including unstructured full text, images, statistics or a mixture of data sources. A Database System is a software application that the user employs to define, create, manage and analyze a database. A Data Model is the theoretical model by which data is structured. Common data models include relational, network and object-oriented. Data modeling is a methodology for structuring data for use in a database system.

The present work is concerned with the creation of relational database systems. The term Relational database describes a particular type of data model which structures data into individual tables, each made up of fields, which are linked together through a system of key fields. A Table is a set of tabulated information within a database. A Field is a precise element of information within a table taking the form of a column. Record is a complete item of information within a table taking the form of a row. A record contains one or more descriptive fields.

Most relational database systems can import and export ASCII data (American Standard Code for Information Interchange). ASCII is an international standard allowing computers to exchange and display character-based data.

Structured Query Language (SQL) is a computer language developed for use with relational database systems. This language has an extensive set of commands that allow the user to define, manage and analyze/query the database. Most desktop applications now include an SQL component.

A great deal of literature exists on the properties and application of relational databases and Structured Query Language, for example :

<http://www.mysql.com/>

<http://sal.kachinatech.com/H/1/MYSQL.html>

<http://www.edm2.com/0612/msql7.html>

<http://www.microsoft.com/TechNet/Access/technote/ac101.asp>

As stated earlier, the main focus of the present work is the creation of a relational database with world-wide-web access. The later requires the application of HyperText Markup Language (HTML), a widely used document format. It uses tags as part of a general framework for describing a document structure primarily for the world-wide-web.

Again, extensive literature exists on HTML, for example:

<http://www.utoronto.ca/webdocs/HTMLdocs/NewHTML/htmlindex.html>

<http://vzone.virgin.net/sizzling.jalfrezi/>

<http://www.w3.org/MarkUp/>

<http://www.cwru.edu/help/introHTML/toc.html>

Only a few links have been mentioned here. Additional links can be found by using any one of the commonly available search engines.

Another aspect of the present work is the processing of ambient vibration data to determine the natural frequencies of vibration.

The concept of obtaining modal parameters of structures from measurements of ambient excitation dates back to 1960s. Akaike (1969) was among the first to study the use of Auto Regressive-Moving Average (ARMA) type models to analyze systems with ambient excitation. Later on, another procedure called Random Decrement was developed to process ambient-excitation data. This led to the first use of a time-domain parameter extraction scheme, as proposed by Ibrahim and Mikulcik (1973).

Several other approaches have been developed for the analysis of ambient vibration data. These approaches include peak-picking from Power Spectral Density (PSD) functions, Maximum Entropy Method (MEM), Auto Regressive-Moving Average (ARMA) models, random decrement processing coupled with

time-domain parameter extraction, the use of cross-correlation functions with time-domain parameter extraction, and direct use of raw data with time-domain parameter extraction.

As stated earlier, reduction in stiffness corresponds to a decrease in the natural frequency and an increase in the damping of a structure. Adams, Walton, Flitcroft and Short (1975) used this fact to detect the presence of cracks in fiber-reinforced plastics.

In recent years the application of dynamic in-situ testing for determining the modal parameters of bridges has gained popularity. Many papers have been published in this regard. Some of these papers include Kato and Shimada (1986), Mazurek and DeWolf (1990), Salane and Baldwin (1990), Biswas, Pandey and Samman (1990), Pandey, Biswas and Samman (1991), Hearn and Testa (1991), Alampalli, Fu and Dillon (1997), Wahab and Roeck (1998).

The idea of the use of innovative design and/or materials has also gained popularity in the recent years. Many structures have been constructed using innovative design and/or materials. Some of the papers published in this field include Mufti, Jaeger, Bakht and Wegner (1993), Bakht and Mufti (1996), Newhook and Mufti (1996), Newhook, Mufti and MacDonnell (1997), Bakht and Mufti (1997), Mufti, Jaeger and Bakht (1997), Bakht and Mufti (1998), Mufti (1998), Mufti and Newhook (1998), Mufti, Newhook and Mahoney (1999).

1.4 OBJECTIVES AND SCOPE

The Canadian Network of the Centers of Excellence on Intelligent Sensing of Innovative Structures (ISIS) has actively promoted the use of innovative materials and/or innovative structural systems in bridge structures. As a result, a number of bridges have been built across Canada that use such materials or systems. A major concern in the use of such innovative materials and/or innovative structural concepts is that there is a lack of experience on the long-term performance of such innovative materials and concepts. Hence, a number of bridges using such materials and concepts have been instrumented with a view to monitor their health. This instrumentation has been carried out under the auspices of ISIS, which also has the responsibility of collecting the instrument readings and processing them for damage detection.

To assist in the task of damage detection and health monitoring ISIS has taken the initiative of creating a central archive for the long-term maintenance of data obtained from the instrumented bridges. The focus of the present work is to design and develop a central database for the archival of data. The thesis also deals with the implementation of some available techniques for the processing of dynamic data obtained from ambient vibrations with a view to determine the frequencies of the structures.

The main objectives of the work are highlighted below :

1. Development of archival database.

The data received from the various bridges has to be converted to a common format and is to reside in a central relational database. This database should contain all the related information about the instrumentation and the recording of the data. The archive has to be created assuming that it is to continue for a number of years. Hence, the technology used for the creation of this database should be able to adapt to future demands.

2. Develop mechanisms for exploring the archive.

Authorized users should be able to explore the archive and to extract data from it for their own analysis. A world-wide-web interface to the archive must be provided for this purpose.

3. Develop methodologies for downloading of archived data.

Data should be downloadable in different formats including HTML (HyperText Markup Language), CSV (Comma Separated Values), Tab Delimited, UFF (Universal File Format) and XML (Extensible Markup Language), to serve users with different requirements.

4. Develop means for updating of the archive.

The archive has to be updated on regular basis. Data collectors, with proper permission, should be able to load data directly into the archive.

5. Implement methods of processing dynamic data.

Analysis of the collected dynamic data is to be performed using CADA-X to find the natural frequency of an example bridge, and the results are to be compared with those obtained by using another software called ESPAN.

2 MONITORING PROGRAM

2.1 INTRODUCTION

This chapter provides a general description of the structures being monitored, and the instrumentation and data acquisition systems used for the monitoring program. The information included here was provided by project leaders responsible for monitoring and is included for the sake of completeness and to characterize the nature of the data to be archived. At the start of this chapter a brief description of the different types of sensors used in the monitoring program is provided. This will help in throwing some light on the sensing technology used for obtaining dynamic, static and environmental data from instrumented bridge structures. A more detailed description of the instrumentation used for bridge monitoring is available in the ISIS manual (ISIS, 1999).

2.2 DATA ACQUISITION SYSTEM

A system that collects the required data from the structure being monitored is called a Data Acquisition System. There are two common data acquisition systems.

1. Conventional Data Acquisition System.
2. Computer-based Data Acquisition System.

A Conventional Data Acquisition system comprises sensors and readout units. In this system the operator has to manually record the data by visually reading it from the readout units. For general applications a Computer-based Data Acquisition system, which can automatically record data, must be used.

Following is a brief description of the components of a typical Data Acquisition System used for bridge monitoring :

1. Sensors
2. Data Acquisition Board
3. Data Acquisition Programs

2.2.1 SENSORS

Sensors are the basic components of a monitoring system. Selection of sensors is based on the specific needs of the project. Some of the issues that need to be taken into consideration include initial cost, size limitations, installation cost, required sensitivity, susceptibility to weather and chemicals.

Obviously long-term reliability of sensors is important since the structures last for a long period of time and monitoring must continue over the lifetime. If a sensor is no longer usable it should be possible to easily replace it. Replacement of sensors when necessary ensures that the data is reliable. Details on the long-term reliability can be found in a manual produced by ISIS (1999).

Commonly used sensors for bridge monitoring are :

1. Strain Gauges
2. Temperature Sensors
3. Accelerometers

1. STRAIN GAUGES

The commonly used strain gauges are :

- a) Foil Strain Gauges
- b) Vibrating Wire Gauges
- c) Fibre Optic Gauges

2. TEMPERATURE SENSORS

The types commonly used for civil engineering applications are :

- a) Resistive Temperature Sensors
- b) Vibrating Wire Temperature Sensors
- c) Fibre Optic Temperature Sensors

3. ACCELEROMETERS

Accelerometers used for civil engineering applications are :

- a) Piezoelectric Accelerometers
- b) Spring-mass Accelerometers

2.2.2 DATA ACQUISITION BOARDS

Signal conditioners produce analog signals. Hence they cannot be directly connected to a computer. The purpose of using a data acquisition (DAQ) board is to convert the analog signals into digital signals. Once converted to digital signals, they can then be recognized by a computer. Basic specifications, which are available on most data acquisition products are :

- a) Number of Channels
- b) Sampling Rate
- c) Resolution
- d) Input range
- e) Gain

2.2.3 DATA ACQUISITION PROGRAM

Data Acquisition (DAQ) Program provides specific instructions to the computer. These instructions include how often and when to scan the DAQ board, and instructions on how to process the collected data. Some suppliers provide DAQ programs with their DAQ boards. These programs are usually general and may not fit the requirements of the specific project. Therefore one may need to develop one's own DAQ program.

As stated earlier, a number of bridge structures have been instrumented under the intelligent data processing project of ISIS. Data from four such bridges has been incorporated in the data archive developed in the present work. The four bridges included in the present work are :

1. Crowchild Bridge
2. Headingly Bridge (Taylor Bridge)
3. Salmon River Bridge
4. Waterloo Creek Bridge

2.3 CROWCHILD BRIDGE

2.3.1 DESCRIPTION OF THE BRIDGE AND TEST PROGRAM

Crowchild Trail/University Drive Bridge is located in Northwest Calgary. This bridge is a two-lane one-way traffic overpass. It has three continuous spans of 29.830, 32.818, and 30.230 m. The traffic load on the bridge increased in the last twenty years and hence the superstructure of the bridge was replaced in June 1997. What makes the bridge special is the fact that the new superstructure is the first continuous-span steel-free bridge deck in the world. The salient features of the bridge are presented in Table 2.1 (Afhami and Cheng, 1999)

TABLE 2.1: CROWCHILD BRIDGE, GENERAL INFORMATION

Name	Crowchild Bridge
Location	Calgary, Crowchild Trail and University Avenue
Opened	1997
Spans	Three continuous spans: North span=29.830 m Interior span=32.818 m South span=30.230 m
Structural System	Cast-in-place concrete deck supported by steel I girders. North and interior spans with steel-free concrete deck; south span with concrete deck having steel reinforcement.

The new superstructure is composed of five 900 mm deep steel girders. The concrete slab deck has been reinforced using polypropylene fibre. The girders have been braced using four evenly spaced cross frames in each span. In addition to these, transverse steel girders join the girders at the supports. The deck has no internal steel reinforcement and is 9030 mm wide, and 275 mm thick. In overhanging cantilevers and at the regions of interior supports, glass fibre reinforcement is used. The side barriers are reinforced with prefabricated glass fibre reinforcing grid, NEFMAC (New Fiber Composite Material for Reinforcing Concrete).

In June and July 1997, 105 strain gauges and 5 thermistors were installed on the bridge. Fifty-three of these sensors were embedded in concrete. Ten cable transducers and four accelerometers were temporarily used in tests conducted in

1998. These ten cable transducers and four accelerometers were in addition to the instruments used in 1997.

In August 1997, static truckload and ambient vibration tests were carried out, while the bridge was still closed to traffic. In September and October 1997 further tests were carried out to find out the mechanical effects of temperature change; moreover, dynamic tests were also carried out under passing traffic.

In June 1998 some preliminary measurements were made under passing traffic. In August 1998, static and dynamic truckload tests, and ambient vibration tests were carried out. Crack patterns formed in the bridge deck were also mapped. This mapping was carried out in August 97, August 98 and in June 99.

2.3.2 INSTRUMENTATION AND DATA ACQUISITION SYSTEM

Sensors installed in 1997

While the bridge was still under construction, eighty-six electrical strain gauges, seventeen electrical embedded strain gauges, five electrical temperature probes and two fiber optic strain sensors were installed in the north span. To monitor the temperature profile of the deck, four thermistors were used. Another thermistor was used to measure the air temperature.

Seventeen embedded strain gauges were used to monitor the strain distribution in the deck in the transverse direction. These strain gauges were embedded in five pre-cast blocks. Three of these blocks were in the positive moment region while the rest two were in the negative moment region.

Out of the eighty-six electrical strain gauges, thirty were installed on NEFMAC reinforcement, shear studs of steel straps, glass fiber reinforcement and stainless steel studs. The remaining fifty-six gauges were installed on straps, girders and cross frames.

The performance of the steel straps was monitored using eighteen gauges. Twelve of these gauges monitored the straps in the longitudinal direction of the bridge. The remaining six gauges were installed on straps. These gauges were used to monitor the lateral distribution of strains in the strap at two different sections. One of these sections was in the positive moment region while the other was in negative moment region.

To monitor the strains in the shear studs of the strap, six strain gauges were used. These gauges were located at 8430 mm from the centerline of bearing at the north abutment.

To monitor steel girders, thirty-four strain gauges were used. Three gauges were installed in the positive moment region and three gauges were installed in the negative moment region of the webs of each of the five girders. Thus, a total of thirty gauges were used to monitor load sharing among the girders and the moment distribution. The remaining four strain gauges were installed on the flanges of girder no. 1 to monitor the warping of the girders.

Four strain gauges were used to monitor the response of cross frames. Two strain gauges were installed on NEFMAC at the barriers. Another two were installed on a stainless steel stud at the barriers.

To monitor the glass fiber reinforcement, six gauges were installed at the overhanging cantilevers and fourteen gauges were installed at pier No. 1.

Two fiber optic sensors were installed on the glass fiber reinforcement at the same section as the electrical strain gauges.

Evaluation of field instrumentation

After construction of the bridge, 100 of the 103 electrical strain gauges, all of the thermistors, and both fibre optic sensors were functional. The 3 non-functional gauges were located in the side barriers. Two of these gauges were located on the stainless steel stud and 1 on the NEFMAC.

Additional sensors used in 1998

In order to measure deflection and acceleration ten cable transducers and four accelerometers were temporarily installed in 1998. Three of these accelerometers were placed evenly on the first girder in the north span. The fourth accelerometer was installed on girder no. 2 at a section about 7.5 m from the north abutment.

Cable transducers 1 through 5 were respectively attached to the steel girders 1 through 5. This attachment was made at a section 14.43 m from the centerline of the north abutment.

Cable transducers 6 through 9 were attached to the steel girders 1 through 4. However, this attachment was made at a section 24 m from the north abutment.

The 10th cable transducer was attached to Girder no. 5 in the middle span. This attachment was made at a distance of 11.6 m from centerline of pier No. 1.

2.3.3 TESTS AND MEASUREMENTS

Table 2.2 provides a summary of the tests and measurements carried out on the Crowchild Bridge.

TABLE 2.2 : SUMMARY OF THE TESTS AND MEASUREMENTS ON THE CROWCHILD BRIDGE

Date	Test or Measurement	Performed by	Measurements
Aug 14, 1997	Static Load Test	U of Alberta	Strain
		City of Calgary	Deflection
Aug 15, 1997	Ambient Vibration Test	U of British Columbia	Natural Frequencies
Sep 25&26, 97	Temperature Change	U of Alberta	Load Related Strains
Oct 27, 1997	Passing Trucks	U of Alberta	Strain
June 3, 1998	Passing Trucks	U of Alberta	Strain
Aug 24, 1998	Static Load Test Dynamic Load Test	U of Alberta	Strain, Deflection, Vibration
Aug 24, 1998	Ambient Vibration Test	U of Alberta	Natural Frequencies

Static load test on August 14, 1997

A static load test was carried out on August 14, 1997 in order to measure deflections of the bridge under heavy traffic loads. This test was carried out before the bridge was opened to traffic. Two trucks were used to produce nine different loading cases. Each of the trucks was loaded to 355 kN (80000 lbs). Both trucks were used in six of these load cases by placing them side-by-side.

However, a single truck was used in the remaining three load cases. Strain gauge readings were obtained for each of the different load cases.

Ambient vibration measurements of 1997

An ambient vibration test was performed on August 15. This test was conducted while the bridge was still closed to traffic. Accelerations produced in the bridge were recorded by a series of accelerometers installed for the purpose.

Thermal effect

Strain readings were taken on the afternoon of September 25 and on the morning of September 26 in order to determine the significance of the restrained thermal effects. These reading were taken when there was no traffic on the bridge. The readings had to be corrected for the thermal effect on the completion gauge as well as for the thermal output error of the active gauges.

Dynamic measurements under heavy vehicles, 1997

On October 27, 1997 dynamic measurements were made. There was insignificant response to car traffic. However, passing trucks produced measurable response. A ten second window of strain data at a scan rate of 1000 readings per second was obtained for passing trucks.

Dynamic measurements under heavy vehicles, 1998

On June 3, 1998 dynamic measurements of strains were recorded using heavy trucks. The main purpose of this test was to evaluate the performance of field instruments after having experienced the first winter.

Static load test, 1998

On August 24, 1998 static load tests were carried out using a single truck loaded to 395 kN. The steering axle weighed 42 kN. The middle and rear axles weighed 153 kN and 200 kN respectively. The strains produced by the truckload were recorded. As well, displacements of the girders were measured by using LVDTs installed for the purpose. The truck was taken off the bridge after every 4-5 readings. This was done in order to avoid correction for thermal effects.

Dynamic truckload test, 1998

On August 24, 1998 dynamic truckload test was carried out using the same truck that was used in the static test. The right wheels of the truck were placed midway between Girders 1 and 2, as done earlier in the static test. The truck was passed at four different speeds: 15, 30, 40, and 55 km/hr. All the dynamic measurements were scanned at 500 Hz. Accelerations were measured by using accelerometers installed for the purpose.

Ambient vibration test, 1998

On August 24, 1998 the second ambient vibration test was conducted using four accelerometers. Eight sets of readings were gathered for vertical modes. Duration of each set was 200 seconds and it was scanned at a rate of 200 readings per second.

2.4 HEADINGLY BRIDGE (TAYLOR BRIDGE)

2.4.1 DESCRIPTION OF THE BRIDGE AND TEST PROGRAM

The Taylor Bridge is considered to be the world's largest highway bridge reinforced with fibre-reinforced polymer (FRP). Four girders, portions of the deck slab, and the barrier walls are reinforced with carbon fibre reinforced polymer (CFRP) and glass fibre reinforced polymer (GFRP). The bridge is being remotely monitored using fibre optic sensors embedded in the girders, the deck slab, and the barrier wall to provide continuous information on the health and structural performance of the bridge.

The salient feature of the bridge are presented in Table 2.3

TABLE 2.3: TAYLOR BRIDGE, GENERAL INFORMATION

Name	Headingley Bridge (Taylor Bridge)
Location	Manitoba Provincial Road 334, Assiniboine River
Opened	28 October, 1997
Spans	5 equal spans, 33 m each
Structural System	8 prestressed concrete girders in each span, 4 girders prestressed with CFRP tendons and reinforced with CFRP stirrups.

The bridge is 165.1m long and consists of 40 prestressed concrete AASHTO type girders. A portion of the deck slab is reinforced by CFRP reinforcement. Glass fibre reinforced plastic (GFRP) has been used to reinforce the barrier walls. These barrier walls are connected to the deck slab with double-headed stainless steel bars.

The monitoring program included testing of a large-scale model of a bridge girder totally reinforced and prestressed with carbon FRP. In addition, a full-scale portion of the bridge deck slab reinforced with carbon fibre reinforced polymer (CFRP) was tested up to failure under simulated traffic loads.

2.4.2 INSTRUMENTATION AND DATA ACQUISITION SYSTEM

Fibre Bragg Grating (FBG) sensors have been installed to monitor the strains in the deck slab, in the CFRP reinforcement of the girders, and in the GFRP reinforcement of the barrier walls.

Sixty-three FBG sensors and two multi-bragg sensors were installed on the CFRP reinforcing bars. These sensors were installed at different locations. As the initial prestressing strain was quite high, most of these sensors were installed after the tendons had been tensioned.

Twenty-six electric strain gauges were installed in addition to the FBG sensors. Although proper measures were taken to seal the strain gauges, 60% of the gauges stopped working properly. The cause of this malfunctioning was excessive moisture resulting from steam curing of the concrete girders.

Twenty-two temperature sensors were installed to take temperature readings at various sections.

2.5 SALMON RIVER BRIDGE

2.5.1 DESCRIPTION OF THE BRIDGE AND TEST PROGRAM

The structural form of Salmon River Bridge is concrete slab-on-steel-girder construction with two simply supported spans of 31.2 m each. Conventional steel reinforced deck technology has been used in one span in the eastbound section of the structure, while the other span has been constructed using the innovative steel-free deck technology. The salient features of the bridge are presented in Table 2.4 (Newhook and Mufti 1996; Mufti and Newhook 1998; Mufti 1998; Mufti, Newhook and Mahoney 1999)

TABLE 2.4: SALMON RIVER BRIDGE, GENERAL INFORMATION

Name	Salmon River Bridge
Location	Trans Canada Highway 104 between Truro, on the west, to New Glasgow, on the east, NS
Opened	5 December, 1995
Spans	Two parallel structures, east bound and west bound. Two simply supported spans of 31.2 m in each structure.
Structural System	Composite concrete deck on 6 steel girders in each span. One span in eastbound structure has steel-free deck.

The deck is constructed of polypropylene fibre reinforced concrete with no internal continuous reinforcement. The deck is supported on six steel girders. These girders are laterally restrained by a series of steel straps, which have been welded to the top flanges of adjacent girders.

The curb/parapet also involves an innovative design, as fibre reinforced polymer (FRP) reinforcement is used in the curbs to provide durability similar to the steel-free deck. Two horizontal layers and one vertical layer of a 150x150 mm NEFMAC glass FRP grid are used.

2.5.2 INSTRUMENTATION AND DATA ACQUISITION SYSTEM

The mid-span of each girder in the steel-free deck span has been instrumented with foil strain gauges. These foil strain gauges are welded near the top and bottom flanges. Additional gauges have been installed on the two outer girders.

Foil gauges have also been installed at a number of straps. Three fibre optic strain gauges have also been installed in the glass FRP grid. The gauges are Bragg grating type sensors and were embedded in the FRP elements during the construction stage.

Field assessment

Field assessment of the steel-free deck has focused on the following :

1. Load sharing among the girders.
2. Fatigue performance of the welded strap connections.
3. Participation of the curbs in carrying live loads.

Load sharing among girders

The service load strains were found to be well within the acceptable limits. In the initial design of the bridge an assumption was made that the girders would behave as composite sections under live load. This assumption is seen to hold good.

Strap fatigue

Transverse steel straps were welded to the top flange of the girders. Calculations showed that fatigue governed the design of the connections. It was concluded that strap welding did not adversely affect the girder design as the top flange of the girder was in compression.

The maximum value of tensile strain noted in the straps was found to be 50 micro-strain. This strain is equivalent to a stress of about 10 MPa. Tests carried out in the laboratory showed that failure would occur when the strap stress reached 350 MPa. Stress recorded in the straps showed that the service load stresses were far below the ultimate load levels.

Curb participation in the live load

As mentioned earlier, three sensors were installed in the NEFMAC GFRP grid in the curbs and parapets. Two of these three sensors were functional after the completion of construction process. It was observed that the relative difference in strain levels was less than 10 micro-strain. This indicated that the original design

assumption holds good and the curb has been isolated from taking part in the bending behavior of the deck.

Dynamic data, 1996

Dynamic data under ambient vibrations was first collected in 1996. Two data sets were recorded on August 16th. Eleven data sets were recorded on November 6th and eight data sets were recorded on December 5th. The data comprised strain readings measured at a fast scan rate.

Dynamic data, 1997

Dynamic data under ambient vibrations was next collected in 1997. Seven data sets were recorded on February 4th and eight data sets were recorded on April 25th.

2.6 WATERLOO CREEK BRIDGE

2.6.1 DESCRIPTION OF THE BRIDGE AND TEST PROGRAM

The Waterloo Creek Bridge is located in the Fanny Bay area, southeast of the City of Courtenay at the intersection of the new Inland Island Highway alignment and Waterloo Creek. The bridge carries four lanes of traffic. The bridge structure consists of two independent, single-span, cast-in-place concrete deck structures. The decks are supported by prestressed concrete I-girders. Each deck has been designed to carry two lanes of traffic.

The Southbound structure is built using traditional steel reinforced concrete deck technology, while the northbound structure is constructed using steel-free FRC deck technology. The length of each bridge deck is 24.4 m. Each deck is 12.2 m wide and has a ten degrees skew. Salient features of the bridge are presented in Table 2.5 (Ventura and Tsai, 1998,1999)

TABLE 2.5: WATERLOO CREEK BRIDGE, GENERAL INFORMATION

Name	Waterloo Creek Bridge
Location	Fanny Bay Area, Courtenay, Vancouver Island, BC
Opened	March 1998
Spans	Span = 24.4 m
Structural System	Two independent, parallel, single-span, cast-in-place concrete decks supported by prestressed concrete I-girders. The northbound structure is constructed with a steel-free FRC deck, while the southbound structure is built with a traditional steel reinforced concrete deck. Both bridge decks are 24.4 m in length, 12.2 m wide and have a ten degrees skew.

The bridge deck is 190 mm thick. The concrete slab has no internal steel reinforcement but is reinforced with polypropylene fibre. Five 1473 mm deep prestressed concrete I-girders support each deck. The girder spacing is 2810 mm. The foundation of each abutment consists of twelve 460 mm diameter open-ended steel pipe piles.

2.6.2 INSTRUMENTATION AND DATA ACQUISITION SYSTEM

A total of fifty-three sensors were installed in the northbound structure during construction. These sensors included 32 foil strain gauges, 5 micro-measurement embedment strain gauges, 4 fibre optic sensors, 10 TML embedment strain transducers and 2 smart rod sensors.

A total of eleven sensors were installed in the southbound structure during construction. These sensors included 6 TML embedment strain transducers and 5 micro-measurement embedment strain gauges.

The data acquisition system used consisted of the following components:

1. Sensor connector array panel (custom made)
2. Signal conditioning: 16 amplifier modules (custom made)
3. Data acquisition: 16-channel LS4 Data Logger of the Lakewood Systems Ltd. and 8-channel StressNet System of the Arbutus Cove Systems Inc.
4. A Pentium 166 MMX portable computer.
5. Strain indicator -a model P-350 of Vishay Instruments.

Twenty-four sensors were installed at different locations to measure strains. The data acquisition systems referred in item 3 above were used to collect the data. Both of these systems were installed on October 15, 1998. Since its installation the Lakewood data logger has been working properly. However, the Arbutus Cove StressNet system failed to deliver any useful measurements.

2.7 MONITORING DATA INCLUDED IN THE ARCHIVE

Table 2.6 shows the data contained in the databases created for each bridge structure.

TABLE 2.6: MONITORING DATA INCLUDED IN THE ARCHIVE

Bridge	Sensor Tables	Data Sets	Images
Crowchild	7	88	13
Taylor	3	44	6
Salmon River	2	36	5
Waterloo Creek	5	None	16

Brief descriptions of the data included in the databases created for the above-mentioned bridges are presented in the following sections.

2.7.1 CROWCHILD BRIDGE

Seven sensor tables, in the database, describe the instrumentation used for the monitoring program. Table1 provides information about the characteristics and locations of the foil gauges. Table2 provides information about the thermistors. Table3 provides information about cable transducers. These transducers were temporarily installed in 1998. Table4 provides information about the

specifications of the trucks used in 1997 static tests. Table5 provides information about the accelerometers, which were also temporarily installed in 1998. Table6 provides the deflection readings in mm for the static load test carried out in 1997. Table7 contains definitions of some of the terms used in the previous tables.

At present the archive contains 88 data sets. Fifty-three data sets were recorded in 1997 and thirty-five data sets were recorded in 1998. The various conditions under which the data was recorded include :

- a) No traffic, temperature induced strain
- b) Dynamic measurements under passing truck
- c) Static load test
- d) Ambient vibration
- e) Dynamic data under passing truck

2.7.2 TAYLOR BRIDGE

Three sensor tables describe the instrumentation used for the monitoring program. Table1 provides information about the characteristics and locations of Fibre Optic Grating Sensors (FOS) and Temperature Sensors. Table2 provides information about the connection details between Fibre Optic Sensors and the Temperature Sensors. Table3 contains definitions of some of the terms used in earlier tables.

At present the archive contains 44 data sets. Thirty-one data sets were recorded in 1998, eight data sets were recorded in 1999 and five data sets were recorded in 2000.

In a majority of data sets the data submitted by the project leaders has no time stamp.

2.7.3 SALMON BRIDGE

Two sensor tables describe the instrumentation used for the monitoring program.

Table1 provides information about the characteristics and locations of foil strain gauges. Table2 contains definitions of some of the terms used in the previous tables.

At present the archive contains 36 data sets. Twenty-one data sets were recorded in 1996 and fifteen data sets were recorded in 1997.

The data included in this database is dynamic data recorded under ambient vibrations excited by traffic load. Time history is available for this data. This data has been processed in the present work to find out the natural frequencies of vibration.

2.7.4 WATERLOO CREEK BRIDGE

Five sensor tables describe the instrumentation used for the monitoring program. Table1 provides information about the location of sensors in the northbound part of the structure. Table2 provides information about the location of sensors in the southbound part of the structure. Table3a and Table3b are cross-reference tables. These tables relate data channels of the Lakewood Data Logger and the Arbutus Cove StressNet system with the sensors installed in the bridge. As mentioned earlier, the Lakewood data logger has been working properly, but the Arbutus Cove StressNet system has failed to deliver any useful measurements.

Table4 provides a list of compensation coefficients and strain calibration coefficients for the monitored sensors.

It is expected that data will soon be recorded from the installed sensors, which include fiber optic sensors and foil strain gauges. Once the data has been recorded, it will be included in the archive.

3 ARCHIVAL OF DATA

3.1 INTRODUCTION

In keeping with its objectives, the Canadian Network of the Centers of Excellence on Intelligent Sensing of Innovative Structures (ISIS) has assisted in the use of innovative materials and/or innovative structural concepts in a number of bridge structures across Canada. There is a serious lack of experience on the long-term performance of such innovative materials and concepts. Keeping this in mind, a number of bridges incorporating innovative materials and/or innovative structural concepts have been instrumented with a view to monitor their health. This instrumentation has been carried out under the auspices of ISIS. Health monitoring and damage detection requires intelligent processing of instrument data. Often such processing involves comparison of measurements obtained at regular intervals of time over the life of the structure. The measured data must therefore be preserved over a long duration of time and stored in a form that it is easily accessible and understood by different users involved in health monitoring. Archival of data is therefore an important aspect of health monitoring (Humar, Holtz, Qureshi, 2001). The primary focus of the present work is the design and development of such an archive.

This chapter describes the process of archival of the data collected from different sources. It presents the objectives of archival, information provided by the archive, software technology adopted, organization & architecture of the database, access to the archive, and the extent of the software developed.

Typical instrumentation consists of a small number of sensors, usually 50 to 100, of various kinds (temperature, strain, displacement and acceleration). Depending on the bridge and the stage of monitoring, data may be collected by one or more of the following techniques :

- 1. Data may be collected at regular intervals under normal traffic conditions using an automated or semi-automated data acquisition system. This data may be submitted to the archive on a regular basis.**
- 2. Static load tests may be scheduled and performed periodically. These tests usually involve closing the bridge to normal traffic, stationing a truck at different positions, and taking a full set of readings corresponding to each position.**
- 3. Dynamic tests may be conducted by using accelerometers. Accelerometers may be manually or automatically triggered to record data at a high scan rate, when the bridge is undergoing vibrations induced by ambient forces, such as traffic or wind.**

3.2 IMPORTANT ATTRIBUTES OF THE ARCHIVE

As stated earlier the present work deals with the creation of a central archive for the long-term collection and maintenance of data obtained from the instrumented bridges. For the purpose of archival the data received from various sources is converted to a common format and made available to a central relational database. The choice of relational database was based on the fact that the data to be archived was essentially in the forms of tables. A relational database is the most effective form for storing tabular data. A world-wide-web interface to the archive has been developed and provided. The archive is designed to enable interested researchers to browse the data in the archive, to view the relevant documentation and to download data for their own analysis.

The archive has been established assuming that it will be maintained and used for a number of years, perhaps long past the time of the original investigation. The interface to the archive allows authorized bridge managers to submit data to the archive. It also allows other researchers to explore the archive, and to extract data from it, using several common formats. Data fields for download can be prepared in a small number of different formats. These formats include comma-separated values (CSV) for ease of importing them into spreadsheets and local databases, and universal file format (UFF) for dynamic data. The other formats available to the user are HTML (HyperText Markup Language), Tab Delimited values, and XML (Extensible Markup Language). At this time UFF and XML have not been implemented.

The archive must be capable of storing readings obtained from a variety of sensors ranging from the traditional strain gauges to fiber optic gauges of different kinds. The sensor data may be generated either through periodic static field tests or dynamic measurements of ambient vibrations. It may also be generated through continuous monitoring.

3.3 BENEFITS OF ARCHIVING

Archival of data offers several benefits as listed below. For example, archived data

1. reinforces open scientific inquiry, since the data is available to many researchers at the same time;
2. encourages diversity of analysis and opinions by different researchers having access to the same data;
3. promotes new research and allows for the testing of new or alternative methods, enabling the use of data in ways that the original investigators had not envisioned;
4. improves methods of data collection and measurement through scrutiny by others;
5. invites response to the discoveries made by others.

3.4 BACK UPS AND VALIDATION OF THE DATA

No database is perfectly clean. Errors may slip through the initial safety net or there may be unforeseen problems that might appear after the data input has been completed. In some cases it may even lead one to conclude that a particular dataset is unusable. Databases should be scanned at regular intervals during the input phase. It is a good idea to run the scan again once the input is completed. In some instances, the discovery that the database software had truncated a field from real data to integer data was discovered after the input of data in a test table. The fields had to be adjusted for these data sets but the loss of time was minimal because of the policy of regular data checks. Because of the somewhat unstable nature of software and hardware systems, all validation procedures should be accompanied by a comprehensive and sensible back up. Some key principles to remember include :

1. Regular back-ups of the database.
2. Back-ups in both the format of the used software and in a neutral format such as ASCII.
3. Back-up the data to a medium that is entirely isolated from the principal hardware currently being used.
4. While making back-ups, check that the copies are actually usable and avoid continuous over-writing of old back-ups with new ones.
5. Carefully check the first and the last 5 to 10 percent of the data records created. After making sure that those data records are in good shape, choose random records for quality-control checks.

3.5 IMPORTANT OBJECTIVES OF ARCHIVING

In designing the archive for this work, several important objectives and attributes of an archive were kept in mind. These are summarized below :

1. Open-source alternatives should be used so that the archive is not tied down to any particular commercial solution. This helps in keeping the costs down. It also ensures to make it easier to change solutions if the situation demands it.
2. The archive should be designed for a longer period of time. It should be able to survive for 10 to 20 years or even more.
3. The time and resources invested in the creation of digital resources can easily be placed in jeopardy because hardware and software become obsolete. Hence, systems that are robust and less likely to become completely obsolete should be selected.
4. Digital resources need to be preserved through changing technologies so that they will be available in the future. The extent to which a digital resource can be preserved is mainly dependent on the decisions taken during the data creation process.
5. It is very important that the technology used to create and maintain the data should be able to adapt to changing trends. The archive should be able to adapt to anticipated changes in future demands and technology.

6. The data included in the archive needs to be readily accessible to other researchers so that they are able to explore the archive and to extract data from it for their own analysis.
7. The archive should be updated on regular basis. Means for updating the archive should always be available. Data collectors, with proper permission, should be able to load data directly into the archive.
8. Good documentation optimizes the preservation process to its maximum. It ensures that the researchers can use the archived data to the full extent. It also ensures that there is less likelihood of incorrect use of the data. Moreover, it certainly helps if the user wishes to return to the dataset for further analysis at some stage in the future.
9. Explanatory material is essential for further, informed use of a dataset. Full understanding of the dataset and its contents, cannot be achieved without this material.
10. Information should be provided about the data collection methods. This information should describe the instruments used and the methods employed. It is always useful to include information on quality control methods employed.
11. Information that describes the structure of the dataset should be available. Key to this type of information is a detailed document that describes the structure of the dataset. This document should also include information about relationships between individual files or records. For relational

models, a diagram showing the structure of the dataset should be constructed.

- 12.** Information related to the provenance of the dataset should be maintained. This information is useful in recording the history of the data collection process, changes and developments that occurred, both in the data themselves and the methodology.
- 13.** It is essential that the structure, form and organization of data collection be described fully. This information should include:
 - a)** List of files and tables with information about their contents, number of records and fields, and the way in which they relate both to each other and to the source.
 - b)** List of field names used in each file with information about the characteristics of each field, including name, contents, field length, data type, and information about the way in which the fields relate to each other and to the source.
 - c)** File format that has been used for collecting the data.

A majority of these objectives have been met.

3.6 INFORMATION PROVIDED BY THE ARCHIVE

The data archive provides salient information about each of the bridge project included in the monitoring program. This information includes the title of the bridge, its location, a brief description of the structural system and the date of opening of the bridge. For each project covered by the archive the following information is provided :

1. Documentation of the instruments.
2. Images showing details of the structural systems and location of gauges.
3. Sensor data obtained from periodic and continuous monitoring, static tests, and dynamic tests under controlled vehicle passage and/or ambient vibrations.

The data is collected from project leaders in standard electronic forms developed for keeping the data in a uniform format. These formats are described in a subsequent section.

3.6.1 DOCUMENTATION OF THE INSTRUMENTS

The documentation related to the instruments includes the following:

1. Sensor number / ID.
2. Sensor type – temperature, strain, displacement, acceleration.
3. Number of sensors used per section/location.
4. Number of sections used in the monitoring program.
5. Wheatstone bridge type used for strain sensors.

6. Reference temperature.
7. Temperature compensation co-efficient.
8. Strain calibration coefficient.
9. Offset values used for fiber optic gauges.
10. Gauge factor.
11. Gauge length.
12. Exact location of the sensor at the bridge component – Abutment, Girder, Steel Straps, Permanent Bracing, Deck, Parapet.
13. Units used for measurement.

3.6.2 IMAGES SHOWING STRUCTURAL SYSTEM AND LOCATION OF SENSORS

These images include the following :

1. Plans and cross-sections of the structure.
2. Drawings showing the location of the various sensors.
3. Photographic images.
4. Electronic images.

3.6.3 DATA FILES CONTAINING SENSOR READINGS

This data forms the heart of the archive, and contains the following information :

1. Unprocessed sensor readings.
2. Description of the loading conditions and the instruments used.
3. Date and time at which the data was recorded.

The sensor readings submitted by the project leaders are referred to as unprocessed. However some basic processing needs to be carried out by the project leaders before submission of the data. Such processing includes, for example, correction of strain gauge and optic sensors for temperature, and filtering of dynamic data to remove noise from the signals. The submitted data is then ready for further processing such as plotting of strain and stress profiles, calculation of induced moments, distribution of load sharing among girders, and determination of frequencies and mode shapes.

3.7 FORMATS FOR DATA SUBMISSION

The data submitted by project leaders needs to be in an organized form. In order to maintain uniformity in the data submitted a series of submission-formats have been prepared. This will not only reduce the burden of initial processing required by the administrators of the archive but will also help in reducing the time required for uploading of the submitted data into the archive.

In some cases the data submitted by the project leaders was in Excel format. These Excel files had to be first converted to text format because the data can be added to MySQL database using only text files. In some cases the size of the data recorded was larger than the capacity that Excel can handle in one file. Hence, the data was stored in 2 Excel files. Such data was first converted to text format and then combined into one text file.

The following table-formats have been suggested for the submission of data.

General Information:

Name	
Location	
Date Opened	
Spans	
Structural System	

Table 1: Fibre Optic Grating Sensors, Characteristics and Location

Sensor no	Gauge Factor GF	Thermo-optic coefficient β_0	Thermal expansion coefficient of the substrate α_s	Reference temp	Offset strain	Temp sensor	Location

Table 2: Fibre Optic Smart Rod Sensors, Characteristics and Location

Sensor no	Gauge Factor GF	Thermo-optic coefficient β_0	Thermal expansion coefficient of the substrate α_s	Reference temp	Offset strain	Temp sensor	Location

Table 3: Embedment Strain Transducers (TML), Characteristics and Location

Sensor no	Reference Strain μ	Reference temp converted to strain μ	Compensation coefficient C_{β} $\times 10^{-6} / ^{\circ}\text{C}$	Strain calibration coefficient C_{ϵ}	Location

Table 4: Foil Strain Gauges, Characteristics and Location

Sensor no	Wheatstone bridge type	Dummy gauge	Temp. Sensor	Reference Strain μ	Reference temp $^{\circ}\text{C}$	Location

It is a well-known fact that strain gauges are affected by change in temperature. Hence temperature compensation has to be applied after data has been recorded. The purpose of using a dummy gauge is to avoid the application of temperature compensation. When a dummy gauge is used in the Wheatstone bridge circuit this gauge counterbalances the effect of temperature and hence temperature compensation need not be applied to the sensor readings. In other cases the project leaders are requested to submit temperature compensated data.

Table 5: Temperature compensated Strain Data obtained from Foil Gauges

Date:

Clock Time at Time Zero:

Time	Sensor FG	Sensor FG	Sensor FG	Sensor FG	Sensor FG	Sensor FG	Sensor FG	Sensor FG	Sensor FG

Table 6: Micrometer Embedment Strain Gauges, Characteristics and Location

Sensor no	Wheatstone bridge type	Dummy Gauge	Temp. Sensor	Reference Strain μ	Reference temp. converted to strain C	Location

Table 7: Temperature compensated strain and temperature data obtained from TML sensors, Foil Strain Gauges, Micrometer Embedment Gauges and Fiber Optic Sensors

Date:

Time	Strain Sensor $\mu\epsilon$	Temp Sensor C	Strain Sensor $\mu\epsilon$	Temp Sensor C	Strain Sensor $\mu\epsilon$	Temp Sensor C	Strain Sensor $\mu\epsilon$	Temp Sensor C

Table 8: Strain data obtained during Controlled Vehicle Test

Date:

Test No.

Location of the front axle of test vehicle	Strain Sensor $\mu\epsilon$	Strain Sensor $\mu\epsilon$	Strain Sensor $\mu\epsilon$	Strain Sensor $\mu\epsilon$	Strain Sensor $\mu\epsilon$	Strain Sensor $\mu\epsilon$	Strain Sensor $\mu\epsilon$	Strain Sensor $\mu\epsilon$

JPEG format was suggested for the submission of image files due to reason that their size is small and this makes it easier to view them while browsing the website.

3.8 DATA ARCHIVE AND SOFTWARE TECHNOLOGY

This section deals with the process leading to the selection of various components of the software developed for the data archive. The principles guiding the selection are described first. This is followed by a description of the software components actually selected for the present project.

3.8.1 PRINCIPLES GUIDING THE SELECTION OF SOFTWARE COMPONENTS

The following important criteria need to be satisfied in the selection of software components :

1. Available open-source alternatives should be used so that the archive is not tied down to any particular commercial solution. This will help to keep the costs down and will also make it easier to change solutions if the situation demands it.
2. It is expected that the archive will have a long life, perhaps 10 to 20 years or even more. Many of the original collectors of the data may no longer be actively involved with the project after a number of years. Consequently systems that are robust and less likely to become completely obsolete should be selected.

3. **Computing technology will evolve a great deal in the future. Hence it is very important that the archive should be able to adapt to these anticipated changes.**

3.8.2 SELECTED SOFTWARE COMPONENTS

Keeping in view the criteria outlined in the previous section the following software components were selected :

1. **Linux Operating System.** Over the past few years, this has proved to be a highly reliable, low-cost solution on which to run server software. Moreover support is also easy to find. Probably more servers on the Internet run Linux than any other operating systems and this number is constantly increasing.
2. **Apache Web-Server.** This web-server is not controversial in any way. It is extremely reliable, efficient, extensible, very well supported (and free). Netcraft surveys have shown that this software powers 60% of the millions of web servers on the Internet. It is almost 3 times as popular as its nearest competitor (the propriety Microsoft IIS).
3. **PHP and Python Programming Languages.** Programming languages are used to write software that will support the database access and dynamic generation of HTML pages for the web. In the present case PHP will be used for the simpler pages – ones that require only a few computations or

database accesses and fairly simple page generation, while considering its superior and simpler object-oriented facilities Python will be used for complex tasks. Python programs tend to be more reliable, efficient and maintainable than any of the common alternatives (Perl, ASP, JSP and PHP).

4. **MySQL Relational Database Software.** Two alternatives present themselves, namely MySQL and Postgresql. While MySQL tends to be used more to support database driven web content, Postgresql has a number of features, such as objects and transactions, that make it more suitable for this application. Preliminary tests have shown that either will adequately handle the millions of records envisioned during the early years. As the archive grows, it may become necessary to switch to more powerful systems such as Inspire or Oracle.

All of the above choices are well proven and very portable. Linux is perhaps the most portable of operating systems. Newer CPU hardware often runs Linux before they run any other operating system. All of the rest of the software (Apache, PHP, Python, Postgresql) run on all common hardware and operating systems, and it is expected that this will continue. This gives the project tremendous flexibility to move to a different computing environment, should the situation demand it.

3.9 STRUCTURE AND ORGANIZATION OF THE DATABASES

The structure of the databases is shown in Figure 3.1. It comprises a database manager and secondary databases for each of the bridges being monitored. The database manager is called IPRM. This name was derived by using the first letter of each word of the project title, “Intelligent Processing and Remote Monitoring”. Any query directed by user’s browser is first sent to database IPRM. After recognizing the nature of the query, the database IPRM points it in the direction of the related bridge database. That database then handles the query by recognizing the nature of information in question. That information is then retrieved, processed and sent back to database IPRM. Database IPRM then sends that information back to the browser of the user where it appears on the user’s monitor. At the present stage of the project the secondary databases include the following :

1. Crowchild
2. Salmon_River
3. Taylor
4. Waterloo_Creek

The basic structure and organization of all these four databases is the same. Database Salmon_River will now be used as an example to describe the various tables in the databases and the associated information provided by them.

FIGURE 3.1 : LAYOUT OF THE DATABASE

Following is a list of all the different types of tables present in database
 Salmon_River :

Name		Contents
1.	DataSets	this table provides information about all data sets.
2.	Sensors	this table provides information about branch tables containing information related to the sensors used in the monitoring program.
3.	Glossary	this table provides information about the different short-names and symbols used for column headings in the tables.
4.	Images	this table provides information about the image files.
5.	Varies	tables containing information about the different types of sensors used in the monitoring program. These tables include information such as the bridge type, temperature compensation co-efficient, location of sensors etc. Table "Sensors" contains the names of these tables.
6.	Varies	tables containing the actual sensor readings obtained from the different sensors used in the monitoring program. Table "DataSets" contains the names of these tables.

The information contained in these tables is further explained below.

3.9.1 TABLE “DataSets”

The following table shows the first two rows of the table “DataSets” :

Id	DSName	TableName	Date	Description
1	sr96081604.txt	SR96081604	1996-08-16	Truck on Left Lane (18 wheeler)
2	sr96081605.txt	SR96081605	1996-08-16	Cement truck on Right Lane

where

Column “id” is the Identity Number.

Column “DSName” contains the external name of the data file containing the dataset. The *.txt extension specifies that it is a text file.

Column “TableName” contains the name of the corresponding database table for the DataSet.

Column “Date” contains the date on which this data was recorded.

Column “Description” provides information about the loading conditions and/or any other relevant information about the data file.

3.9.2 TABLE “Sensors”

The following table shows the first row of the table “Sensors” :

TableName	Type	Description	id
Table1	Foil Strain Gauge	Table1: Foil Strain Gauges, Characteristics And Locations	1

where

Column “TableName” contains the name of the Sensors table.

Column “Type” represents the type of sensors described in the Sensors table.

Column “Description” contains a brief description of the Sensors table.

Column “id” is the Identity Number.

3.9.3 TABLE “Glossary”

The following table shows the first two rows of table “Glossary” :

name	long_name	description
sens_num	Sensor Number/ID	This is the common identifier of the sensor.
B_type	Bridge Type	Wheatstone bridge circuit.

where

Column “name” is the short name used in the sensor tables.

Column “long_name” is the actual long name being referred to in the first column.

Column “description” provides additional information about each sensor.

3.9.4 TABLE “Images”

The following table shows the first row of table “Images” :

id	Image	Thumbnail	Description
1	gagelocl.jpg	gagelocl-tn.jpg	Strain Gage Locations Feb 05, 1996

where

Column “id” is the Identity Number.

Column “Image” is the name of the image file.

Column “Thumbnail” is the name of the thumbnail image.

Column “Description” provides information about the image.

3.10 ACCESS TO THE ARCHIVE

There are essentially four different groups of users accessing the archive. Each group has different access requirements, as shown in Figure 3.2, and described in the following paragraphs :

1. **Providers Of Data.** These are the project leaders who are responsible for the collection of data in the first place. They must be able to document and upload data sets for inclusion in the archive. The data submitted by the providers is essentially raw but must have undergone some basic processing such as, for example, correction of sensor readings for temperature effects and filtering of dynamic data to eliminate noise.

2. **Administrators Of The Archive.** These are the persons that ensure that uploaded data is sufficiently documented and reliable. They are responsible for the conversion of data from the supplied format into the relational database format that is maintained by the archive. In certain cases of demonstrated competence and adherence to standards, some providers will be able to do most or all of these administrative tasks themselves. At present none of the administrative task has been delegated to the project leaders.
3. **Registered Users.** Only those who have registered with the administrators and have received their own login names and passwords are allowed unrestricted browsing of the database and the retrieval and downloading of data.
4. **Unregistered Users.** A small portion of the site has unrestricted access and can be read, anonymously, by anyone with Internet access. This portion contains general descriptions of the project and contacts for fuller access. The vast majority of the site, however, has restricted access. Unregistered users have access only to the unrestricted portion.

All of the above mentioned groups of users have been provided with a web-based interface to access the archival system. Each group of the users has a different starting point appropriate to their level of access.

FIGURE 3.2 : ACCESS TO THE ARCHIVE

The following paragraphs present details of the interface between the users, the web browser and the archive. The first page of the browser is a fairly standard opening page, as shown in Figure 3.3

FIGURE 3.3 : OPENING PAGE OF THE WEBSITE

The opening page provides the user with some basic information in the following order :

- **About the Archives.** This link provides a brief introduction about the archives. It throws light on the proposal for the creation and maintenance

of the archive. It also describes the archival procedure and the future activities of the project.

- **Adding Data to the Archives.** This link provides information about the basic structure and organization of the archive. It enables users, with appropriate level of permission, to upload the recorded data into the archive. At the present stage such permission has not been granted to any user. This will be done only after the users have had sufficient exposure to the archive.
- **Extracting Data from the Archives.** This link provides information about the different formats available for downloading the data. This link also provides help to the user for downloading the data in different formats.
- **The Data Archives.** This link provides information about the actual Data Archive after the users have properly identified themselves by using the username and password assigned to them.

Figure 3.4 shows an image of the screen as seen by group 3, that is, registered users, after they have properly identified themselves. This screen is accessed when the user selects “The Data Archive” from the page shown in Figure 3.3

FIGURE 3.4 : IMAGE VIEWED BY REGISTERED USERS

The description of the interface is as follows :

- **Crowchild Bridge:** Calgary, Crowchild Trail and University Avenue.
This link provides complete information about the Crowchild Bridge.
- **Headingley Bridge (Taylor Bridge):** Manitoba Provincial Road 334, Assiniboine River. This link provides complete information about the Taylor Bridge.
- **Salmon River Bridge:** Trans Canada Highway 104 between Truro, on the west, to New Glasgow, on the east, NS. This link provides complete information about the Salmon River Bridge.

- **Waterloo Creek Bridge: Fanny Bay Area, Courtenay, Vancouver Island, BC.** This link provides complete information about the Waterloo Creek Bridge.

Suppose that the user clicks the link for the Salmon River Bridge. Clicking this link will lead to the window shown in Figure 3.5

The screenshot shows a web browser window with the address <http://dataarchwest.mscanada.com>. The page title is "Salmon River Bridge". The content is organized into three main sections:

Salmon River Bridge

Name:	Salmon River Bridge
Location:	Trans Canada Highway 104 between Truro, on the west, to New Glasgow, on the east, NS
Opened:	5 December, 1995
Spans:	Two parallel structures, east bound and west bound. Two simply supported spans of 31.2m in each structure.
Structural System:	Composite concrete deck on 6 steel girders in each span. One span in eastbound structure has steel-free deck.

Sensors

Set	Type	Description
Table1	Foil Strain Gauge	Table1: Foil Strain Gauges, Characteristics and Locations
Table2	Miscellaneous	Table2: Miscellaneous terms.

Data Sets

Data Set	Date	Description
r96081604.txt	1996-08-16	Truck on Left Lane (18 wheels); Dalhousie Univ.

FIGURE 3.5: DATA FOR SALMON RIVER BRIDGE

The page starts with general information about the Salmon River Bridge, which includes the location, date of opening, description of the structural system.

Table “Sensors” lists all of the tables containing information about the sensors used in the monitoring program. The table “Sensors” also provides information about the type of sensor used, location of the sensor and the units of measurement.

Table “DataSets” lists all the data sets available in the database Salmon_River. It also provides information about the date of recording of data and a brief description of the data set.

At the bottom of the page is a table called “Images”. This table provides information about the different images available in the database. The users are able to view a thumbnail of the images, which provides them a general idea of the images and allows them to select a particular image by double clicking on the thumbnail.

Suppose that the user requests information about sensors used in the monitoring program by clicking on “Table1”. Clicking this link will lead to the screen shown in Figure 3.6

Salmon River Bridge: Sensors

Table1: Foil Strain Gauges, Characteristics and Locations

<u>sens_num</u>	<u>b_type</u>	<u>location</u>	<u>units</u>
FG01	1A4	Strap	micro-strain
FG02	1A4	Girder 1, bottom, mid-span	micro-strain
FG03	1A4	Girder 1, mid-height, mid-span	micro-strain
FG04	1A4	Girder 1, top, mid-span	micro-strain
FG05	1A4	Strap between girders 1 and 2	micro-strain
FG06	1A4	Strap	micro-strain
FG07	1A4	Strap	micro-strain
FG08	1A4	Girder 2, bottom, mid-span	micro-strain
FG09	1A4	Girder 2, mid-height, mid-span	micro-strain
FG10	1A4	Girder 2, top, mid-span	micro-strain
FG11	1A4	Strap between girders 1 and 2	micro-strain
FG12	1A4	Strap	micro-strain
FG13	Unknown	Unknown	Unknown

FIGURE 3.6 : DATA OF TABLE1, SALMON RIVER BRIDGE

The screen in Figure 3.6 shows a list of all the sensors used in the monitoring program. It also provides information about the sensors including bridge type used for foil gauges, location of the sensor and the units of measurement. The column headings have been abbreviated in order to keep the width of the column to a minimum. Clicking the column heading opens a pop-up window that explains the column heading. For example, the pop-up window shown in Figure 3.7 will appear if the user clicks the column heading “sens_num”.

FIGURE 3.7 : DATA OF TABLE GLOSSARY, SALMON RIVER BRIDGE

Now suppose that the user wants to gain knowledge about one of the data sets. If the user clicks the first data set “SR96081604”, the screen shown in Figure 3.8 appears.

FIGURE 3.8 : DATA OF TABLE SENSORS, SALMON RIVER BRIDGE

The screen begins with a general description of the data contained in Data Set SR96081604. It shows that this data set has 1989 records and 9 columns. After this the first and last five rows of the data are displayed for user's convenience.

After this the user is given the option of downloading the number of rows according to his/her requirement. The user can choose all the rows or can choose a particular number of rows to be downloaded by entering the starting number and the number of rows required.

The user is then provided the choice of selecting the columns to be downloaded. The user can choose all or any number of columns with the help of check boxes.

At the bottom of the screen the user is given a choice for choosing the format for downloading the data. The destination of the data to be downloaded can be either to screen or to a local file. The different formats available are :

- HyperText Markup Language (HTML)
- Comma Separated Values (CSV)
- Tab Delimited
- Universal File Format (UFF)
- Extensible Markup Language (XML)

XML and Universal File Format have not been implemented at this stage.

Once the user has made all of the required selections he/she can press the “Download Selected Data” button to download the desired data.

“Help” buttons have been provided to help the user in case of any confusion.

3.11 CLIENT – SERVER ARCHITECTURE

From the point of view of a person requesting information, the basic architecture of the system is the standard Web client-server model as shown in Figure 3.9. The steps in one cycle of information request and display are approximately the following, with minor variations in some specific cases :

FIGURE 3.9 : CLIENT-SERVER ARCHITECTURE

1. A request starts with the activation of a link in the user's web browser. Activation will typically be by clicking on hyper-link or by pressing a button. That link will have URL associated with it of the form :

`"http://dataarchives.isiscanada.com/archives/bridge.php3?bridge=Salmon_River"`
2. The web browser splits the URL into 3 parts :
 - a) `http://` tells the browser to communicate with the server using the "Hyper Text Transfer Protocol".
 - b) `"dataarchives.isiscanada.com"` is the name of the server on the Internet that is prepared to handle a request for data.
 - c) The browser contacts the server, and sends the remainder of the URL `"/archives/bridge.php3?bridge=Salmon_River"` as a request for information.
3. The server divides the request into two parts at the `"?"`. The first part of the request, `"/archives/bridge.php3"` is examined by the server, and is found to be the name of a "CGI script", that is an executable program (in this case, written in PHP3 programming language). The second part of the request, `"bridge=Salmon_River"` is interpreted as extra data to be made available to the CGI script.
4. The server starts the execution of the CGI script and passes the extra data to it in a form that is easily interpreted by the script.

5. The script “bridge.php3” executes and decodes the data passed to it, and prepares and prints some output in response. In this case it is the Salmon_River database. The script accesses the database, retrieves the requested information about Salmon_River database, and prints the appropriate HTML page.
6. The server then collects all of the text (HTML) printed by the script, and packages it in the format required by the HTTP protocol for transmission to the browser. That data is then sent back to the browser as the answer to the original request.
7. The browser decodes the HTTP data received from the server, and extracts the HTML text from the data stream.
8. The browser then displays the text on the screen, using the HTML instructions to render it in the desired form.
9. The browser then waits for the user to begin the next cycle.

As another example suppose that the user requests information about a data table named “SR96081604” on the database Salmon_River. The associated URL for this request will have the form :

`http://dataarchives.isiscanada.com/archives/dataset.php3?bridge=Salmon_River&table=SR96081604`

This information will be handled in the following manner.

1. The web browser splits the URL into 3 parts :

- a) http:// tells the browser to communicate with the server using the “Hyper Text Transfer Protocol”.**
- b) “dataarchives.isiscanada.com” is the name of the server on the Internet that is prepared to handle a request for data.**
- c) The browser contacts the server, and sends the remainder of the URL
“/archives/bridge.php3?bridge=Salmon_River&table=SR96081604”
as a request for information.**

2. The server divides the request into two parts at the “?”. The first part of the request, “/archives/bridge.php3” is examined by the server, and is found to be the name of a “CGI script”, that is an executable program (in this case, written in PHP3 programming language). The second part of the request, “bridge=Salmon_River&table=SR96081604” is interpreted as extra data to be made available to the CGI script.

3. The server starts the execution of the CGI script and passes the extra data to it in a form that is easily interpreted by the script.

4. The script “bridge.php3” executes and decodes the data passed to it, and prepares and prints some output in response. In this case, the data names a data set in the Salmon_River database. The script accesses the database, retrieves the requested information about the data set in the Salmon_River database, and prints the appropriate HTML page.
5. The server then collects all of the text (HTML) printed by the script, and packages it in the format required by the HTTP protocol for transmission to the browser. That data is then sent back to the browser as the answer to the original request.
6. The browser decodes the HTTP data received from the server, and extracts the HTML text from the data stream.
7. The browser then displays the text on the screen, using the HTML instructions to render it in the desired form.
8. The browser then waits for the user to begin the next cycle.

All of the above requires the cooperation of many pieces of software. All of these are standard, off-the-shelf components, with the exception of the CGI script “bridge.php3”. In the above description, that is the only custom-software component developed specifically for this project.

3.12 SOFTWARE ARCHITECTURE

The entire project required a large number of special software components, at least one for each different kind of request. Each individual component is fairly small and simple, but the interaction and dependencies of a large number of these is quite complex.

The architecture of the software system consists of a number of major modules as described below :

1. **The Access Control Module.** This module validates a particular user and sets certain access rights. Some users have different privileges from others, allowing them to see and/or change different things in the database. This module authenticates users (by prompting for names and passwords) and sets information that limits what those users may do. Some parts of the system have no read access controls (they are essentially publicly readable). The access control module is invoked only when necessary.

2. **The Raw Data Module.** All data submitted to the archive is submitted in 'raw' form (these are essentially raw data files in a form that can be generated by the monitoring team with a minimum of special processing). These are retained by the archive, perhaps in a compressed form. These raw data files must be completely documented; the documentation must include names, dates, sensors read, short description of the test, etc. The documentation must also include notes such as problems encountered,

sensors that were suspected of being non-working, etc. In fact, the documentation must include essentially anything that will help interpret the data 10 years later. This module has tools for uploading the raw files, archiving them and collecting the documentation.

3. **Data Conversion Module.** This is a collection of utilities and components that convert the raw data files to the relational data base form, and from the relational database form to the downloadable formats such as CSV and UFF. Sometimes, special purpose conversions will have to be developed for the occasional raw data set that does not follow the established conventions. These will also be included in this conversion module, as well as being archived with the raw data file to which they are applied. Again, if a raw data file has to be re-processed after 10 years, it must be possible to easily access the original data conversion component that was used.
4. **Browsing Module.** This is the largest set of components. It contains all of the software necessary to explore the database, discovering what is there, and preparing concise displays of that information so that the user may make further selections.
5. **Upload Module.** A set of components for managing the uploading and conversion of raw data files from the investigators.
6. **Download Module.** A set of components for managing the conversion and downloading of selected data.

7. **Documentation Module.** A set of components for managing the collection of documentation that describes a series of data sets.
8. **Help Module.** A set of components that assist users and maintainers of the system.

3.13 CODES WRITTEN FOR THE ARCHIVE

The main codes written for the archive are re-produced in Appendix A. However the User ID and Passwords have been omitted for security reasons. Codes developed to solve specific problems related to the submitted data have not been described here. The need for such codes arose in cases where the submitted data did not follow the specified format. Hence the data submitted by the project leaders had to be converted to a basic format.

4

PROCESSING OF DATA

4.1 INTRODUCTION

The primary objective of archiving the data is to permit an informed user to download the data at anytime and be able to process it to provide indication of the health of the structure. As an example, changes in the health of a structure can be detected by monitoring the load sharing among girders in a slab and girder bridge. The state of load sharing can be assessed by examining the static deflection and stress profile produced by a controlled vehicle load. If the archived data provides sufficient information to enable comparison of deflection and stress profiles at different times but under identical loads, the processing of data is quite straightforward. The focus of archival is then not to provide algorithms for processing of the data but to ensure that data is complete and intelligible. The present project therefore does not deal with processing algorithms, leaving it to the users to use their own tools for processing. The user may want to download data in various formats to suit the processing tool. The different formats in which the data can be downloaded are described here.

While the processing of static data is relatively straightforward, considerable difficulty exists in the analysis of dynamic data, particularly that obtained from ambient vibration tests. An effort has been made in this work to implement some

very recent and sophisticated methods of processing available in a commercial computer software. The results obtained from such processing are described in this chapter.

4.2 FORMATS FOR DOWNLOADING OF DATA

As described in Chapter 3, the user will be able to download data from the archive in the following formats :

- 1. HTML**
- 2. Comma Separated Values (CSV)**
- 3. Tab Delimited**
- 4. Universal File Format (UFF)**
- 5. XML**

The first three formats for downloading have already been implemented; the other two will form part of the future development. Formats HTML, CSV, Tab Delimited & XML are commonly used and well understood formats. The universal file format is suitable for dynamic data. The main reason for the use of UFF is to facilitate sharing of data between engineers and scientists. A detailed description of UFF has been provided by Structural Dynamics Research Laboratory, University of Cincinnati, Ohio. The (URL) address for this information is www.sdrl.uc.edu. Parts of this description are included in Appendix B.

4.2.1 HTML

One of the formats available for downloading data is HTML. As an example of downloading in HTML format consider the first 10 rows of data set SR96081604.

When choosing the HTML option, the user will get the following format on the screen :

Time	FG16	FG17	FG18
0.0000	-18.2700	-16.8800	-7.3900
0.0200	-18.2700	-16.7400	-7.3900
0.0400	-18.2700	-16.7400	-7.2500
0.0600	-18.2700	-16.7400	-7.3900
0.0800	-18.2700	-16.7400	-7.3900
0.1000	-18.4100	-16.8800	-7.3900
0.1200	-18.2700	-16.7400	-7.3900
0.1400	-18.4100	-16.8800	-7.3900
0.1600	-18.4100	-16.8800	-7.3900
0.1800	-18.4100	-16.8800	-7.3900

4.2.2 COMMA SEPARATED VALUES (CSV)

Suppose that the user opts to download the first ten rows of dataset SR96081604 in CSV format. When choosing this option, the user will get following format :

```
"Time","FG16","FG17","FG18"
0.0000,-18.2700,-16.8800,-7.3900
0.0200,-18.2700,-16.7400,-7.3900
0.0400,-18.2700,-16.7400,-7.2500
0.0600,-18.2700,-16.7400,-7.3900
0.0800,-18.2700,-16.7400,-7.3900
0.1000,-18.4100,-16.8800,-7.3900
0.1200,-18.2700,-16.7400,-7.3900
0.1400,-18.4100,-16.8800,-7.3900
0.1600,-18.4100,-16.8800,-7.3900
0.1800,-18.4100,-16.8800,-7.3900
```

4.2.3 TAB DELIMITED VALUES

Suppose that the user opts to download the same data in Tab Delimited format.

When choosing this option, the user will get the following format :

Time	FG16	FG17	FG18
0.0000	-18.2700	-16.8800	-7.3900
0.0200	-18.2700	-16.7400	-7.3900
0.0400	-18.2700	-16.7400	-7.2500
0.0600	-18.2700	-16.7400	-7.3900
0.0800	-18.2700	-16.7400	-7.3900
0.1000	-18.4100	-16.8800	-7.3900
0.1200	-18.2700	-16.7400	-7.3900
0.1400	-18.4100	-16.8800	-7.3900
0.1600	-18.4100	-16.8800	-7.3900
0.1800	-18.4100	-16.8800	-7.3900

4.3 PROCESSING DYNAMIC DATA

A method of health monitoring that is gaining increasing attention is vibration-based damage detection. This method relies on measurements of the vibration characteristics, such as frequencies and mode shapes. A change in these characteristics reflects change in the physical characteristics such as stiffness and mass. Damage detection algorithms permit one to find the location and severity of damage in a structure from the measured changes in vibration characteristics. Several identification methods depend on the measurement of both the input excitation and the output response. These measurements permit the evaluation of frequency response functions (FRF), which are in turn used to find the frequencies and mode shapes.

The following methods are normally used to find the vibration characteristics from FRFs, a process known as modal analysis.

1. Single Degree Of Freedom Method (SDOF)

This method is used to perform Single-Degree-Of-Freedom (SDOF) parameter estimation. The starting point is a set of measured FRFs all relative to one reference DOF. The SDOF methods provide a quick means of identifying a set of modes, but will only produce meaningful results if the measurements satisfy the SDOF assumption. The methods used for performing SDOF are :

- a) Peak picking
- b) Mode picking
- c) Circle fitting

2. Complex Mode Indicator Function (MIF)

This method allows one to identify a modal model for a mechanical system where multiple reference FRFs were measured. The method provides a quick and easy way of determining the number of modes in a system. The methods used for performing Complex MIF are :

- a) LSCE (Least Squares Complex Exponential)
- b) FDPI (Frequency Domain direct Parameter Identification)

3. Time Domain Multi Degree Of Freedom (MDOF)

This method groups together all the functions required to perform a Multi-Degree-Of-Freedom (MDOF) parameter estimation. The starting point is a set of measured FRFs relative to one or more reference DOFs. The MDOF methods provide a high quality model based on techniques that are optimized both in terms of accuracy and speed. The methods used for performing Time Domain MDOF are :

- a) LSCE (Least Squares Complex Exponential)
- b) LSFD (Least Squares Frequency Domain)

4. Frequency Domain Multi Degree Of Freedom (MDOF)

This method groups together all the functions required to perform a Multi-Degree-Of-Freedom (MDOF) analysis based on the Frequency Domain Direct Parameter Identification (FDPI) technique.

4.3.1 OPERATIONAL MODAL ANALYSIS

It should be noted that for heavy structures such as bridges it is often impractical to excite vibrations under a controlled excitation force. The excitation force must be of large magnitude, requiring bulky equipment. Also the bridge must be closed to traffic. To avoid these difficulties ambient vibration excitation must be used in such cases, and the excitation force is therefore not measured. FRFs are not available in such cases, and

special techniques have to be used to find the frequencies and mode shapes from measured output alone. These techniques are generally known as operational modal analysis.

The purpose of operational modal analysis is to extract modal frequencies, damping and mode shapes from data taken under operating conditions, that is, under the influence of natural excitation such as by airflow around the structure, road traffic, water flow etc. The methods used for performing Operational Modal Analysis are :

- a) Least Squares Complex Exponential (LSCE-LSFD method)
- b) Balanced Realization (BR method)
- c) Canonical Variate Analysis (CVA analysis)

Operational Modal Analysis method of processing dynamic data has been used in this work and is dealt with detail in the following sections, which describe two different software packages for determining vibration characteristics from ambient vibration data.

4.3.2 EXPERT SYSTEM PERFORMANCE ANALYSIS SOFTWARE PACKAGE (ESPAN)

Under the auspices of ISIS a system performance analysis software package (ESPAN) was developed for intelligent processing of structural response signals registered by sensors at critical locations of a structure. (Maalej, Karasaridis, Hatzinakos and Pantazopoulou 1998, Maalej, Karasaridis, Hatzinakos and Pantazopoulou 1999). The package ESPAN is used to identify the changing properties of instrumented structural systems through time by being able to analyze both static and dynamic data. ESPAN can provide fast-automated warnings about the performance of a structure. Based on the commercial software MATLAB, ESPAN is fully portable as it can run on any major platform that supports MATLAB.

Current features available in ESPAN

1. Interactive visualization of sensor readings.
2. Estimation of natural frequencies and damping ratios from ambient vibration.
3. Spectral analysis of records using a collection of algorithms.
4. Mode Shapes.
5. Load Sharing Characteristics (Moment and Strain Profiles).
6. Estimation of Pre-stress Losses in prestressed concrete members.
7. Estimation of vehicle velocities.
8. Estimation of the coefficient of thermal expansion of sensor substrate materials.

9. Compression and decimation of data records using wavelet transforms.
10. Signatures of sensors and members given different evaluation criteria.
11. Image viewing diagrams or photos of the structure.
12. Graphical User Interface driven interaction.
13. Temperature compensation.

A system identification procedure was developed for processing the data received from the sensors. This procedure is based on spectrogram estimation. A spectrogram is a three-dimensional plot in which the x-axis, y-axis and z-axis indicate time, frequency and power spectral density (PSD). ESPAN is capable of producing discrete-time spectrogram estimates using one of the following methods :

1. **Periodogram (PER) based approach.** This method is based on the classical periodogram estimator. This method reveals all the spectral components of the signal, and is particularly useful in providing good estimation of the PSD for long data sequences. However, high variance is observed with short data records. Also, noise affects this method severely.
2. **Yule-Walker (YW) method.** This method is based on the parametric approach, and assumes auto-regressive modeling of the signal. Parametric approach often exhibits superior performance than a non-parametric approach.
3. **Burg method.** Burg's method is also based on the parametric approach, and assumes auto-regressive modeling of the signal.

A more detailed description of the methods can be found in Maalej et al (1998, 1999)

System requirements

Hardware - PC Pentium, SunSparc workstation.

Operating Systems - Windows95/NT, Linux, SunOS, Solaris.

Software - MATLAB 5.2

RAM - 16MB minimum.

Recently, commercial software has been developed that incorporates more efficient methods of parameter estimation from ambient vibration data. One of these commercially available software is LMS CADA-X. This software has been used in present work for the processing of dynamic data.

4.3.3 LMS CADA - X FOR PROCESSING OF DYNAMIC DATA

LMS CADA-X software comprises a comprehensive set of modules that assist in modal testing and analysis. The LMS CADA library includes a set of modular programs. The following environments related to modal testing are included in the complete version of LMS CADA-X :

- 1. Fourier Monitor**
- 2. Signature Monitor**
- 3. Time Data Processing**
- 4. Sound Quality**

The following modules are available in the area of data analysis :

- 1. Geometry**
- 2. Modal Analysis**
- 3. Modal Design**
- 4. FRF based substructuring**
- 5. Principal Component Analysis**
- 6. Transfer Path Analysis**
- 7. Experimental SEA**
- 8. Running Modes**
- 9. Acoustic Intensity and Real Time Animation**

The following modules are included in the version of CADA-X available in the Department of Civil and Environmental Engineering at Carleton University :

- 1. Time Data Processing**
- 2. Geometry**
- 3. Modal Analysis**

Program CADA-X was originally developed for HP-9000 series workstations using the Unix operating system. However, LMS has recently developed a version for execution on a PC network. This version is referred to as 3.5B. It uses a graphical user interface called OSF/Motif windows.

Analysis Procedure used in LMS CADA-X

There are 2 stages for the analysis of the recorded data :

- a) Input / Importing of recorded data in CADA-X.**
- b) Analysis of data.**

a) Input / importing data in CADA-X

Following is a brief description of the procedure the user has to follow in order to import the dynamic data :

- 1. The first step is to select a project. The user can “open” an existing project or he can create a new project using the “Create new” option.**
- 2. The dynamic data is imported next. This can be achieved by using either of the following two methods :**
 - a) Using the “Import” pull-down menu. It is possible to import different file formats like “Neutral file”, “Universal file”, “CADA-PC Project” and “RoadRunner files”.**
 - b) Input of dynamic data through *.txt or *.dat file format. To use this method it is necessary to have the file “ascii2bdm.tu” in the same directory as the current project. Click “Time data processing monitor” from “Test” pull-down menu. The following window appears :**

Now in the “Options” pull-down menu click “Commands”.

Then type “upa //C/Directory Name/ascii2bdm” to indicate the exact location of the file ascii2bdm.tu The following window appears :

Now enter the name of the data file to import, the number of the column, number of rows and the result block (the imported file will be stored in the “Result block”). Then press “Apply”. Clicking “Apply” will automatically open “Edit block headers” window, shown below :

Edit block header

Edit header for BPF block

Function type	time record	quality	unknown
Primary signal type	power	secondary signal	power
☐ Geometrical DOF		☐ Geometrical DOF	
Component		Component	
Node		Node	
Direction		Direction	
Identifier		Identifier	
Window type	uniform	Window type	uniform
param 1		param 1	
Case # of channel #		Case # of channel #	
Correction mode		Not applicable	
Amplitude correction factor	1		
Energy correction factor	1		
Acoustic weighting	LINEAR	Q factor	10
X axis	XH #	delta x	1
Y axis	Block size	10001	values
User identification	messages #		
Apply	OK	Reset	Cancel Help

This window contains different fields and a number of action buttons. The contents of all fields are automatically updated to reflect the values related to the current block. Enter appropriate values in the related fields. Click “Apply” and then click “Cancel” to start the importing of data. Click “Cancel” after importing finishes. After this go back to “Time data processing monitor” window. From “Options” pull-down menu, click “Trace List”. The following window appears :

Now use “Trace” pull-down menu to choose “Import from block” option.

The following window will appear :

Click “OK”. The block will now appear with all its associated properties, in the “Trace list” window. Select the block by clicking on it once. Go back to the “Trace” pull-down menu and click “Save as”. Another dialog box appears which prompts you to save the block as a TDF (Test Data File). Supply a TDF name. Pressing “Save” will save the block as a TDF. The data is now stored in the database and is ready for analysis.

b) Analysis of dynamic data

When the input and storage of data is complete, the user is then ready to find the frequencies. The user now needs to switch from “Time data processing monitor” to “Modal analysis”. Click Exit to → Analysis → Modal Analysis as shown below :

Need for Operational Modal Analysis

As stated earlier, traditional modal model identification methods and procedures are based on forced excitation laboratory tests during which Frequency Response Functions (FRFs) are measured. However, the real loading conditions to which a structure is subjected often differs considerably from those used in laboratory testing. Since all real-world systems are to a certain extent non-linear, the models obtained under real loading will be linearized for much more representative working points. Additionally, environmental influences on system behavior will be taken into account.

In many cases, such as traffic and/or wind excitation of civil constructions, forced excitation tests are very difficult to conduct. In such situations operational data are often the only ones available.

Traditional processing of operational data

An accepted way of dealing with operational analysis is based on a peak-picking technique applied to auto- and cross powers of the operational responses. Such processing results in the so-called “Running Mode Analysis”. By selecting the peaks in the spectra, approximate estimates for the resonance frequencies and operational deflection shapes can be obtained. These shapes can then be compared to the laboratory modal results.

The auto- and crosspower peak-picking method requires considerable engineering skill to select the peaks which correspond to system resonance. In addition, no information about the damping of the modes is obtained and the operational deflection shapes may differ significantly from the real mode shapes in case of closely spaced modes.

Application of Operational Modal Analysis

As explained earlier, in ambient vibration testing the excitation force is not measured. Special methods have to be used to find out the frequencies, mode shapes and damping from output data only. The purpose of Operational Modal Analysis is to extract frequencies, mode shapes and damping from operational data.

Over recent years, several modal parameter estimation techniques have been proposed and studied for modal parameter extraction from output-only data.

These include :

1. Auto-Regressive Moving Averaging models (ARMA)
2. Natural Excitation Technique (NExT)
3. Stochastic subspace methods

Three methods are available in CADA-X Operational Modal Analysis module :

- a) **Least Squares Complex Exponential (LSCE-LSFD method)**. This method originates from NExT technique, and works on Auto-correlation and Cross-correlation functions. Operational LSCE-LSFD method should be selected if large data sets are to be used.
- b) **Balanced Realization (BR method)**
This method originates from stochastic subspace technique. This method may be selected if the data sets are not too large.
- c) **Canonical Variate Analysis (CVA analysis)**
This method also originates from stochastic subspace technique. This method is also suitable for cases where the data sets are small.

Additional details on the operational Modal Analysis methods have been provided by Hermans and Van der Auweraer (1999).

The following flow chart shows the procedure used in operational modal analysis.

After clicking “Modal Analysis” in “Time data processing monitor” (as shown above) the following window appears :

1. Click “Licenses” in the “Option” pull-down menu. Then click “Operational Modal Analysis” and then “Close”, as shown in the following figure.

2. Then click “Operational Modal Analysis” from “ASM” pull-down menu.
3. Choose the data source as “From TDF”.
4. The following window appears which prompts the user to choose the TDF.

5. The following Operational Modal Analysis window will appear automatically once the user chooses the TDF for analysis.

6. Select a method for analysis. The default method is “Operational LSCE-LSFD”.

7.

“Pre-processing” icon can then be used for pre-processing of the data. Various available functions are shown below :

8. Pressing “Start pre-processing” button will start the pre-processing process.

9.

Use “Select frequency band” button to select the frequency range.

10.

Then press the “Calculate poles” button to start calculating the poles. Once the poles have been calculated, the user can then start clicking on them to find the frequencies. In CADA-X software, the good poles are represented by the letter “s” (in red color). The user can click on a pole to record the frequency.

4.4 ANALYSIS OF DATA FOR SALMON RIVER BRIDGE

Some of the data collected from Salmon River Bridge was analyzed using the above-mentioned technique. The average value of the frequency observed is mentioned at the bottom of each analysis. Close agreement between the frequencies calculated from different sets of data shows good precision in the calculations performed. Twenty-four data sets were processed for calculating frequency of vibration. Results from these tests have been shown in Table 4.1. Figures from five of these analyses are shown below:

Calculated Frequency : 3.875 Hz

lms

Static Stabilization #1

10/10/10

Calculated Frequency : 3.868 Hz

lms Window Align Format Limits Data

Help

Calculated Frequency : 3.875 Hz

Calculated Frequency : 3.878 Hz

Calculated Frequency : 3.855 Hz

The results obtained using CADA-X are summarized in Table 4.1

TABLE 4.1 : FUNDAMENTAL NATURAL FREQUENCY OF SALMON RIVER BRIDGE

Id	Frequency (Hz)
1	3.829
2	3.830
3	3.824
4	3.869
5	3.875
6	3.873
7	3.875
8	3.878
9	3.830
10	3.831
11	3.825
12	3.868
13	3.853
14	3.857
15	3.872
16	3.828
17	3.824
18	3.826
19	3.870
20	3.875
21	3.865
22	3.827
23	3.856
24	3.855

Earlier work by Maalej, Karasaridis, Hatzinakos and Pantazopoulou (1998) and Maalej, Karasaridis, Hatzinakos and Pantazopoulou (1999), showed that the calculated natural frequency estimates for Salmon River were 3.9 Hz and 3.88 Hz respectively. The above results are in good agreement with the previous work.

SUMMARY, CONCLUSIONS 5 AND FUTURE WORK

5.1 SUMMARY AND CONCLUSIONS

The present work deals with the creation of a central archive for the long-term collection and maintenance of data obtained from the instrumented bridges. For the purpose of archival, the data received from various sources has been converted to a common format and made available to a central relational database. A world-wide-web interface to the archive has been provided. The archive has been established assuming that it is to continue for a number of years.

The archive has been designed to enable interested researchers to browse the data in the archive, to view the relevant documentation and to download data for their own analysis. The interface to the archive allows authorized project leaders/bridge managers to submit data to the archive. It also allows other researchers to explore the archive, and to extract data from it, using several common formats. These formats include comma-separated values (CSV) for ease of importing them into spreadsheets and local databases. The other formats available to the user are HTML (HyperText Markup Language) and Tab Delimited values.

The benefits of archiving, the important objectives associated with the present archive and databases have been described in this work.

Following the creation of the desired database, dynamic data collected from Salmon River has been processed to find the frequencies of vibration. Commercially available software LMS CADA-X was used for this processing.

In conclusion it can be stated that knowledge of the integrity of in-service structures on a continuous real-time basis is an important objective for the managers of the structure as well as for the users. When continuous monitoring is not possible, periodic testing may be carried out. The data obtained from monitoring can be used to provide the bridge engineer a warning if abnormal conditions should occur. This will help in avoiding catastrophic failures. Monitoring also permits efficient scheduling of maintenance and optimum use of resources available for such maintenance.

Structural health monitoring requires comparison of sensor readings taken at successive instances of time over the history of the structure. Evidently, an important requirement for the success of structural health monitoring program is careful archival of the data. Because the archived data would be used over an extended period of time by a series of users, it should be as complete and transparent as possible. Important attributes have been described in this work. The data archive has been designed to satisfy these attributes as far as possible.

5.2 FUTURE WORK AND RECOMMENDATIONS

- 1. The archive should be updated on regular basis.**
- 2. A uniform and standard format needs to be developed for the collection and submission of data collected by the project leaders.**
- 3. The collection and submission formats suggested in the present work need to be expanded so as to include additional important aspects of monitored data.**
- 4. A code for downloading data in Universal File Format needs to be developed.**
- 5. Better correlation between the archived data and the documentation available needs to be established.**
- 6. Further analysis of the archived data should be carried out to understand the response of such bridges to different loading conditions and patterns.**
- 7. A history of the analysis performed on the recorded data, to find out the natural frequency of vibration, needs to be maintained in order to help the researchers compare their results with the previous analysis.**
- 8. A regular replacement of the faulty sensors and/or data acquisition units needs to be maintained.**
- 9. Web technology and software architecture utilized should be further modified in order to help increase the life of the archive.**

REFERENCES

- [1]** Adams, R.D., Walton, D., Flitcroft, J. E. and Short, D. (1975). Vibration Testing as a Nondestructive Test Tool for Composite Materials: American Society for Testing Materials. Composite Reliability, ASTM STP 580, pp. 159-175.
- [2]** Afhami, S., Cheng, R. (1999). Field Instrumentation and Monitoring of Crowchild Bridge in Calgary, Alberta. Department of Civil and Environmental Engineering, University of Alberta, Canada.
- [3]** Akaike, H. (1969). Power Spectrum Estimation Through Autoregressive Model Fitting. Annals of the Institute of Statistical Mathematics, 21, pp. 407-419.
- [4]** Alampalli, S., Fu, G. and Dillon, E. W. (1997). Signal Versus Noise in Damage Detection by Experimental Modal Analysis. Journal of Structural Engineering, ASCE, 123(2), pp. 237-245.
- [5]** Bakht, B. and Mufti, A. A. (1996). FRC Deck Slabs without Tensile Reinforcement. Concrete International, 18(2), pp. 514-530.
- [6]** Bakht, B. and Mufti, A. A. (1997). Concrete Deck Slabs without Tensile Reinforcement. International Seminar on New Trends in Highway Construction, New Delhi, India, Volume II, pp. 59-75.

- [7] Bakht, B. and Mufti, A. A. (1998). Five Steel-free Bridge Deck Slabs in Canada. Journal of the International Association for Bridge and Structural Engineering, IABSE, 8(3), pp. 196-200.**
- [8] Biswas, M., Pandey, A.K. and Samman, M.M. (1990). Diagnostic Experimental Spectral/Modal Analysis of a Highway Bridge. Journal of Analytical and Experimental Modal Analysis. 5(1), pp. 33-42.**
- [9] Hearn, G. and Testa, R. B. (1991). Modal Analysis for Damage Detection in Structures. Journal of Structural Engineering, ASCE, 117(10), pp. 3042-3063.**
- [10] Hermans, L. and Van der Auweraer, H. (1999). Modal Testing and Analysis of Structures under Operational Conditions: Industrial Applications.**
- [11] Humar, J. L., Holtz, N. M. and Qureshi, H. T. (2001). Archival of Data from Monitoring of Innovative Structures. SPIE 8th Annual International Symposium on "Smart Structures and Materials", Newport Beach, California, USA, Paper No. 4337-34, pp. 195-196.**
- [12] Ibrahim, S. R. and Mikulcik, E. C. (1973). A Time Domain Vibration Test Technique. Shock and Vibration Bulletin, No. 43, Part 4.**
- [13] ISIS Canada (1999). Bridge Instrumentation and Monitoring Manual. Project T6.2. Canadian Network of Centers of Excellence in Intelligent Sensing of Innovative Structures, Winnipeg, Manitoba.**

- [14] Kato, M. and Shimada, S. (1986). Vibration of PC Bridge during Failure Process. Journal of Structural Engineering, ASCE, 112(7), pp. 1692-1703.**
- [15] Maalej, M., Karasaridis, A., Hatzinakos, D. and Pantazopoulou, S. J. (1998). Intelligent Monitoring of Instrumented Infrastructure Facilities. 30th International SAMPLE Technical Conference, pp 278-292.**
- [16] Maalej, M., Karasaridis, A., Hatzinakos, D. and Pantazopoulou, S. J. (1999). Spectral Analysis of Sensor Data in Civil Engineering Structures. Computers and Structures, pp. 675-689.**
- [17] Mazurek, D. F. and DeWolf, J. T. (1990). Experimental Study of Bridge Monitoring Technique. Journal of Structural Engineering, ASCE, 116(9), pp. 2532-2549.**
- [18] Mufti, A. A. (1998). Field Performance of a Reinforcing Steel-free FRC Deck Slab for Salmon River Bridge. Proceedings of the 1998 Structural Engineering World Congress, Elsevier Science Limited, paper reference: T138-4.**
- [19] Mufti, A. A. and Newhook, J. P. (1998). Experimental Evaluation of the Steel-free Deck Slab of the Salmon River Bridge. Proceedings of the First Annual Synthetic Fiber Reinforced Concrete Symposium, Orlando, Florida. Synthetic Fiber Reinforced Concrete Association, pp. 133-157.**

- [20] Mufti, A. A., Jaeger, L. G. and Bakht, B. (1997). Field Performance of Steel-free Deck Slabs of Girder Bridges. Proceedings, US-Canada-Europe Workshop on Bridge Engineering, Zurich, Switzerland, July 14-15, pp. 239-246.**
- [21] Mufti, A. A., Jaeger, L. G., Bakht, B. and Wegner, L. D. (1993). Experimental Investigation of FRC Slabs without Internal Steel Reinforcement. Canadian Journal of Civil Engineering. 20(3), pp. 398-406.**
- [22] Mufti, A. A., Newhook, J. P. and Mahoney, M. A. (1999). Salmon River Bridge Field Assessment. Proceedings of the 1999 Canadian Society for Civil Engineering Annual Conference. Volume I, pp. 51-61.**
- [23] Newhook, J. P. and Mufti, A. A. (1996). A Reinforcing Steel-free Concrete Bridge Deck for Salmon River Bridge. Concrete International, 18(6), pp. 30-34.**
- [24] Newhook, J. P., Mufti, A. A. and MacDonnell, R. E. (1997). Steel-free Concrete Bridge Deck. XIIth IRF World Meeting, Toronto, Canada, paper no. 184-E.**
- [25] Newhook, J. P., Mufti, A. A. and Mahoney, M. A., (1999). Salmon River Bridge Field Assessment of Deck Cracking. ISIS Canada Project T3.3. Research Report No. 1, Dalhousie University, Halifax, Nova Scotia.**
- [26] Pandey, A. K., Biswas, M. and Samman, M. M. (1991). Damage Detection from Changes in Curvature Mode Shapes. Journal of Sound and Vibration, 145(2), pp. 321-332.**

- [27] Salane, H. J. and Baldwin, J. W. (1990). Identification of Modal Properties of Bridges. Journal of Structural Engineering, ASCE, 116, pp. 2008-2021.**
- [28] Ventura, C. E. and Tsai, P. C. (1998). Waterloo Creek Bridge Project, Instrumentation Progress Report No 1, Department of Civil Engineering, University of British Columbia Earthquake Engineering Research. Project Number 98-001, Vancouver, British Columbia.**
- [29] Ventura, C. E. and Tsai, P. C. (1999). Waterloo Creek Bridge Project, Field Assessment Report No 2, Department of Civil Engineering, University of British Columbia Earthquake Engineering Research. Project Number 99-03, Vancouver, British Columbia.**
- [30] Wahab, M. M. and Roeck, G. (1998). Dynamic Testing of Prestressed Concrete Bridges and Numerical Verification. Journal of Bridge Engineering, ASCE 3(4), pp. 159-169.**

APPENDIX A

CODES DEVELOPED FOR THE ARCHIVE

A1 index.php3

This code has been written in php3 language. This is the main page which is accessed by the web-browser when a user opens this website. This page acts as a blue-print and controls all the information about the bridge data included in the archive.

```
<?php $page_title = 'Main Menu'; include 'header.inc'?>

<table border="0" width="100%">
  <tr>
    <td width="71%" align=center>
      <font face="Arial, Helvetica, sans-serif" color="#990033" size="4">
        <b>Intelligent Processing and Remote Monitoring</b>
      </font><br>
      <font face="Arial, Helvetica, sans-serif" color="#990033" size="5">
        <b>Data Archives and Intelligent Processing</b>
      </font>
    </td>
    <td width="29%"><A
      HREF="http://www.isiscanada.com/programs/monitoring/index.htm"></A></td>
  </tr>
</table>

<BLOCKQUOTE>
<BLOCKQUOTE>
<BIG><B>
```

```

<UL>
<?php
require( 'common.inc' );
$rs = query( "SELECT * FROM Bridges ORDER BY Name" );
while( $b = mysql_fetch_object($rs) ) {
?>
<LI> <A HREF="bridge.php3?bridge=<?php echo $b->DBName?>"><?php echo
$b->Name?>: <?php echo $b->Location?></A>.
<P>
<?php } ?>
</UL>
</B>
</BIG>
</BLOCKQUOTE>
</BLOCKQUOTE>
<?php include 'footer.inc'?>

```

A2 header.inc

This is a text file. The purpose of this code is to control the header of the page.

This code gets included in index.php3 during its execution.

```

<html>

<head>
<title>ISIS IPRM Archives: <?php echo $page_title?></title>
</head>

<body marginwidth="0" marginheight="0" topmargin="0" leftmargin="0"
bgcolor="#DFD4B1" background=" ../images/brigbg.jpg" >

```

A3 footer.inc

This is a text file. The purpose of this code is to control the footer of the page.

This code gets included in index.php3 during its execution.

```
<p align="right"><a href="mailto:central@isiscanada.com"></a>&nbsp; <a  
href="mailto:muftia@cc.umanitoba.ca"></a>&nbsp; <a  
href="mailto:jhumar@cee.carleton.ca"></a></p>  
</body>  
</html>
```

A4 common.inc

This is a text file. This code controls the features which are common in the execution of different other codes. This code controls the access to the archive by verifying the username and password. The user is then able to browse the information about the bridges once he or she has been properly identified.

```
<?php /* Hey! Emacs! -*- tab-width:4 -*- */  
  
$host = 'localhost';  
$user = '????';  
$password = '????';  
if( $dbname == "" )  
    $dbname = 'IPRM';  
$me = $SCRIPT_NAME;  
  
if( $dbname == "" ) {
```

```

        die( "Database name not specified.\n" );
    }
    $lid = mysql_pconnect( $host, $user, $password )
        or die( "Unable to connect to server" );
    mysql_select_db( $dbname, $lid )
        or die( "Unable to select DB: $dbname: " . mysql_error() );

    function query( $str ) {
        global $lid;
        $r = mysql_query( $str, $lid );
        if( $r ) return $r;
        echo '<FONT COLOR=RED><B>Error in query: ', $str;
        echo ": ", mysql_error(), "</B></FONT><BR>\n";
        die("");
    }

    if( $bridge != " " ) {
        $rs = query( "SELECT * FROM Bridges WHERE DBName='$bridge'" );
        if( mysql_num_rows($rs) != 1 )
            die( "Unknown Bridge: " . $bridge );
        $BRIDGE = mysql_fetch_object($rs);
        $dbname = $bridge;
        mysql_select_db( $dbname, $lid );
    }

    class NullObject { };
    $common_tables = "";
    $common_sensors = "";
    $common_glossary = "";

    function sensorinfo( $name ) {

        global $common_tables;
        if( !is_array($common_tables) ) {
            $common_tables = Array();           /* build list of all sensor table
names */
            $rs = query( "SELECT * FROM Sensors" );
            while( $t = mysql_fetch_object($rs) ) {
                $common_tables[] = $t;
            }
        }

        global $common_sensors;
        if( !is_array($common_sensors) ) {

```



```

        $common_sensors = Array();          /* build dictionary of all sensors
*/
    for( $i = 0; $i < sizeof($common_tables); $i++ ) {
        $t = $common_tables[$i];
        $rs = query( sprintf( "SELECT * FROM %s", $t->TableName ) );
        while( $s = mysql_fetch_object( $rs ) ) {
            $s->Type = $t->Type;
            $common_sensors[$s->sens_num] = $s;
        }
    }

    $s = $common_sensors[$name];
    if( is_object($s) )
        return $s;
    if( ereg('^T[0-9]', $name) ) { /* special case temp sensors for Taylor. Ugh!*/
        $temp = ereg_replace( "\\$[0-9]+", "", $name );
        $s = $common_sensors[$temp];
        if( is_object($s) ) {
            $s->sens_num = $name;
            return $s;
        }
    }
    if( !is_object($s) ) {
        $common_sensors[$name] = $s = new NullObject();
        $s->sens_num = $name;
        $s->sens_num = $temp;
        $s->location = '?';
        $s->Type = '?';
    }

    return $s;
}

function glossaryinfo( $name ) {

    global $common_glossary;
    if( !is_array($common_glossary) ) {
        $common_glossary = Array();          /* build list of all glossary entries
names */
        $rs = query( "SELECT * FROM Glossary" );
        while( $t = mysql_fetch_object($rs) ) {
            $common_glossary[$t->name] = $t;
        }
    }

    $s = $common_glossary[$name];

```

```

        if( !is_object($s) ) {
            $common_sensors[$name] = $s = new NullObject();
            $s->name = $name;
            $s->long_name = "";
            $s->description = '?';
        }

        return $s;
    }
}
?>

```

A5 bridge.php3

This code has been written in php3 language. This code controls the information about a particular bridge. The information about each bridge comprises of our main components :

1. General information about the bridge.
2. Information about the sensors used for the monitoring program.
3. Data sets recorded during the monitoring program.
4. Images of sensor locations.

This code controls each of the above-mentioned components.

```

<?php
    include 'common.inc';
    $page_title = $BRIDGE->Name . ' Main Menu';
    include 'header.inc';
?>
<P>
<center>
<h2><?php echo $BRIDGE->Name?></h2>
<table border=1 cellspacing=0 cellpadding=4>
    <tr>

```

```

        <td><b>Name:</b></td>
        <td><?php echo $BRIDGE->Name?></td>
    </tr>
    <tr>
        <td><b>Location:</b></td>
        <td><?php echo $BRIDGE->Location?></td>
    </tr>
    <tr>
        <td><b>Opened:</b></td>
        <td><?php echo $BRIDGE->Opened?></td>
    </tr>
    <tr>
        <td><b>Spans:</b></td>
        <td><?php echo nl2br($BRIDGE->Spans)?></td>
    </tr>
    <tr>
        <td><b>Structural System:</b></td>
        <td><?php echo nl2br($BRIDGE->StructSystem)?></td>
    </tr>
</table>
</center>
<P>

```

```

<center>
<h3>Sensors</h3>
<table border=1 cellpadding=4 cellspacing=0 width="50%">
    <tr bgcolor="#E0E0E0">
        <th>Set</th>
        <th>Type</th>
        <th>Description</th>
    </tr>
    <?php
        $rs = query( "SELECT * FROM Sensors ORDER BY id" );
        while( $row = mysql_fetch_object($rs) ) { ?>
            <tr>
                <td><a href="sensor.php3?bridge=<?php echo $bridge?>&table=<?php echo
$row->TableName?>"><?php echo $row->TableName?></a></td>
                <td><?php echo $row->Type?></td>
                <td><?php echo $row->Description?></td>
            </tr>
        <?php
        }
    ?>
</table>
</center>
<P>

```

```

<center>
<h3>Data Sets</h3>
<table border=1 cellspacing=0 cellpadding=4>
  <tr bgcolor="#E0E0E0">
    <th>Data Set</th>
    <th>Date</th>
    <th>Description</th>
  </tr>
  <?php
    $rs = query( "SELECT * FROM DataSets ORDER BY Date,id" );
    while( $row = mysql_fetch_object($rs) ) { ?>
      <tr>
        <td><a href="dataset.php3?bridge=?php echo $bridge?>&table=?php echo
$rowse->TableName?>"><?php echo $rowse->DSName?></a></td>
        <td><?php echo $rowse->Date?></td>
        <td><?php echo $rowse->Description?>&nbsp;  </td>
      </tr>
    <?php
    }
  ?>
</table>
</center>

<?php
  if( $BRIDGE->image_root != " ) {
  ?>
  <P>
  <center>
  <H3>Images</H3>
  <b>[ <a href="image.php3?bridge=?php echo $bridge?>">Thumbnails</a>
  ]</b>
  <table border=1 cellspacing=0 cellpadding=4>
  <?php
    $rs = query( "SELECT * FROM Images ORDER BY id" );
    $n = 0;
    while( $img = mysql_fetch_object($rs) ) {
  ?>
      <tr>
        <th><?php echo ++$n?>.</th>
        <td><a href="image.php3?bridge=?php echo $bridge?>&image=?php echo
$rowse->Image?>"><?php echo $rowse->Description?></a></td>
      </tr>
    <?php
    }
  ?>
  </table>

```

```

</center>
<?php
}
?>
<?php include 'footer.inc'?>

```

A6 dataset.php3

This code has been written in php3 language. This code controls the information provided by each data set. The information provided by each data set includes the date of recording of data, sensors used for recording data, location of these sensors, total number of rows and total number of columns.

```

<?php
    include 'common.inc';
    $page_title = $BRIDGE->Name . ' Data Set';
    include 'header.inc';
?>
<script language=javascript>
function setOn( form ) {
    sensors = form["sensors[]"];
    for( i = 0; i < sensors.length; i++ )
        sensors[i].checked = true;
    return false;
}
function setOff( form ) {
    sensors = form["sensors[]"];
    for( i = 0; i < sensors.length; i++ )
        sensors[i].checked = false;
    return false;
}
function setNot( form ) {
    sensors = form["sensors[]"];
    for( i = 0; i < sensors.length; i++ )
        sensors[i].checked = !sensors[i].checked;
    return false;
}
function help( str ) {

```

```

        window.open("../help/"+str+".html", "_help",
"toolbar=no,menubar=no,scrollbars=yes,width=400,height=300,resizable=yes");
        return false;
    }
</script>
<P>&nbsp;<P>
<center>
<?php
    $rs = query( "SELECT * FROM DataSets WHERE TableName='$table'" );
    $data = mysql_fetch_object( $rs );
    echo "<table width='75%'\><tr><td><h3 align=center>", $BRIDGE->Name,
" Data Set: ", $data->DSName, ": ", $data->Date, ", ", $data->Description,
"</h3></td></tr></table>\n";
?>
</center>
<P>
<form action="download.php3" method="post">
<input type=hidden name="bridge" value="<?php echo $bridge?>">
<input type=hidden name="table" value="<?php echo $table?>">
<center>
<?php
    $cols = Array();          /* get columns in the selected table */
    $rs = query( "DESCRIBE $table" );
    while( $row = mysql_fetch_object($rs) ) {
        $cols[] = $row->Field;
    }

    $rs = query( sprintf( "SELECT count(%)s AS nrec FROM %s", $cols[0], $table
) );
    $row = mysql_fetch_object( $rs );
    $nrec = $row->nrec;
    echo "<H4>Table ", $table, " has ", $nrec, " records and ", sizeof($cols), "
columns. [<A href='\"' onclick='\"return help('download')\"'>Help
Downloading</a>]</H4>", "\n";

    $qcols = $cols;          /* show only a few columns if too many */
    $partial = 0;
    $n = 4;                  /* show this many from each end of row */
    if( sizeof($cols) > (2*$n+1) ) {
        $qcols = Array();
        $partial = 1;
        for( $i = 0; $i < $n; $i++ )
            $qcols[] = $cols[$i];
        for( $i = sizeof($cols)-$n; $i < sizeof($cols); $i++ )
            $qcols[] = $cols[$i];
    }

```

```

echo "<table border=1 cellspacing=0 cellpadding=4>\n";
echo ' <tr bgcolor="#E0E0E0">', "\n";
echo "  <th>&nbsp;</th>\n";
for( $i = 0; $i < sizeof($qcols); $i++ ) {
    if( $spartial && $i == $n )
        echo "  <th>...</th>\n";
    echo "  <th>", $qcols[$i], "</th>\n";
}
echo " </tr>\n";

/* show a few rows from each end */

$m = 5;
if( $nrec <= 2*$m ) {
    $off1 = 0;
    $len1 = $nrec;
    $off2 = 0;
    $len2 = 0;
} else {
    $off1 = 0;
    $len1 = $m;
    $off2 = $nrec - $m;
    $len2 = $m;
}

$rs = query( sprintf( "SELECT %s FROM %s LIMIT %d,%d", join(',', $qcols),
$stable, $off1, $len1 ) );
while( $row = mysql_fetch_row($rs) ) {
    echo ' <tr align=right>', "\n";
    echo sprintf( "  <td>%d</td>\n", ++$off1 );
    for( $i = 0; $i < sizeof($qcols); $i++ ) {
        if( $spartial && $i == $n )
            echo "  <td>...</td>\n";
        echo "  <td>", $row[$i], "</td>\n";
    }
    echo " </tr>\n";
}

if( $len2 > 0 ) {
    echo sprintf( ' <tr><th colspan="%d">...</th></tr>',
sizeof($qcols)+1+$spartial ), "\n";
    $rs = query( sprintf( "SELECT %s FROM %s LIMIT %d,%d",
join(',', $qcols), $stable, $off2, $len2 ) );
    while( $row = mysql_fetch_row($rs) ) {
        echo ' <tr align=right>', "\n";
        echo sprintf( "  <td>%d</td>\n", ++$off2 );
        for( $i = 0; $i < sizeof($qcols); $i++ ) {

```

```

        if( $partial && $i == $n )
            echo "    <td>...</td>\n";
            echo "    <td>", $row[$i], "</td>\n";
        }
        echo "  </tr>\n";
    }
}

echo "</table>";
?>
<P>
<B>Select Rows below:</B>
<table border=1 cellspacing=0 cellpadding=4>
  <tr>
    <th>Choose:&nbsp;</th>
    <td><select name="rows"><option value="all">All Rows</option><option
value="some">Selected Rows (below):</option></select></td>
  </tr>
  <tr>
    <th>Starting Row #:</th>
    <td><input type=text name=startrow value="1" size=6></td>
  </tr>
  <tr>
    <th># of Rows:</th>
    <td><input type=text name=numrows value="<?php echo $nrec?>"
size=6></td>
  </tr>
</table>
<P>
<B>Select Columns below:&nbsp;&nbsp;&nbsp;&nbsp;</b>[<b>
<a href="" onclick="return setOn(document.forms[0])"><b>Select All</b></a>
<b>|</b>
<a href="" onclick="return setOff(document.forms[0])"><b>Select
None</b></a>
<b>|</b>
<a href="" onclick="return setNot(document.forms[0])"><b>Invert
Selection</b></a>
<b>]</b>
<table border=1 cellspacing=0 cellpadding=4>
  <tr bgcolor="#E0E0E0">
    <th>&nbsp;</th>
    <th>Column</th>
    <th>Type</th>
    <th>Units</th>
    <th>Location</th>
  </tr>

```



```

<big><b><input type=submit value="Download Selected Data"> [ <A href=""
onclick="return help('download')">Help</a> ]</b></big>
</center>
</form>

<?php include 'footer.inc'?>

```

A7 sensor.php3

This code has been in written in php3 language. This code provides a list of all the sensor tables present in the archive for a particular bridge. It provides the names of individual sensor tables and a brief description about each table.

```

<?php
    include 'common.inc';
    $page_title = $BRIDGE->Name . ' Sensors';
    include 'header.inc';
?>
<script language="javascript">
function help( str ) {
    window.open("glossary.php3?bridge=<?php echo $bridge?>&name=" + str,
    "_help",
    "toolbar=no,menubar=no,scrollbars=yes,width=400,height=300,resizable=yes");
    return false;
}
</script>
<P>
<center>
<table width="50%">
<tr><td>
<h2 align=center><?php echo $BRIDGE->Name?>: Sensors</h2>
<?php
    $rs = query( "SELECT * FROM Sensors WHERE TableName='$table'" );
    $data = mysql_fetch_object( $rs );
    echo "<h3 align=center>", $data->Description, "</h3>\n";
?>
</td></tr></table>
</center>
<P>

```

```

<center>
<table border=1 cellspacing=0 cellpadding=4>
  <tr bgcolor="#E0E0E0">
    <?php
      $col = Array();
      $rs = query( "DESCRIBE $table" );
      while( $c = mysql_fetch_object( $rs ) ) {
        $col[] = $c->Field;
        echo "  <th><A HREF=\"\" ONCLICK=\"return help('\", $c->Field, '\"'\", \">\",
$ c->Field, \"</A></th>\n";
      }
    ?>
  </tr>
  <?php
    $rs = query( "SELECT * FROM $table" );
    while( $row = mysql_fetch_array( $rs ) ) {
      echo "  <tr>\n";
      for( $i = 0; $i < sizeof($col); $i++ )
        echo "    <td>", $row[$i], "</td>\n";
      echo "  </tr>\n";
    }
  ?>
</table>
</center>

<?php include 'footer.inc'?>

```

A8 sensorinfo.php3

This code has been written in php3 language. This code provides complete information about a particular sensor table. Typical sensor information includes its type, location, calibration factor and thermal coefficient of expansion.

```

<?php include 'common.inc'?>
<html>
<head>
<title>Sensor Info: <?php echo $bridge?> - <?php echo $sensor?></title>

```

```

</head>
<body bgcolor=white>
<h3><?php echo $BRIDGE->Name?>: Column Heading '<?php echo
$sensor?>'/</h3>
<table border=1 cellspacing=0 cellpadding=5>
  <tr>
    <th>Number/ID:</th>
    <td><?php $s=sensorinfo($sensor); echo $s->sens_num?></td>
  </tr>
  <?php if( $s->location != " ) { ?>
    <tr>
      <th>Location:</th>
      <td><?php echo $s->location?>&nbsp;</td>
    </tr>
  <?php } ?>
    <tr>
      <th>Units:</th>
      <td><?php echo $s->units?>&nbsp;</td>
    </tr>
  </table>
<p>
<center>
<b>[ <a href="" onclick="window.close()">Close Help Window</a> ]</b>
</body>
</html>

```

A9 glossary.php3

This code has been in written in php3 language. This code displays the actual long name for the column heading and also provides a brief description about the abbreviated column-headings used in different tables.

```

<?php include 'common.inc'?>
<html>
<head>
<title>Glossary Info: <?php echo $bridge?> - <?php echo $name?></title>
</head>
<body bgcolor=white>
<h3>Glossary for <?php echo $BRIDGE->Name?></h3>

```

```

<table border=1 cellspacing=0 cellpadding=5>
  <tr>
    <th>Term:</th>
    <td><?php $g=glossaryinfo($name); echo $g->name?></td>
  </tr>
  <tr>
    <th>Full Name:</th>
    <td><?php echo $g->long_name?>&nbsp;</td>
  </tr>
  <tr>
    <th>Description:</th>
    <td><?php echo $g->description?>&nbsp;</td>
  </tr>
</table>
<p>
<center>
<b>[ <a href="" onclick="window.close()">Close Help Window</a> ]</b>
</body>
</html>

```

A10 image.php3

This code has been in written in php3 language. This code controls the different images included in the archive for a particular bridge. Thumbnails of images have been included in the archive to help the user. The user can click any of the thumbnails to view a full-size image.

```

<?php
  include 'common.inc';
  $page_title = $BRIDGE->Name . ' Images';
  include 'header.inc';
?>
<p>
<center>
<h2><?php echo $BRIDGE->Name?></h2>
<p>
<?php

```

```

        if( $image == " ) {
?>
<table border=1 cellspacing=0 cellpadding=4>
  <tr>
    <td><b>Name:</b></td>
    <td><?php echo $BRIDGE->Name?></td>
  </tr>
  <tr>
    <td><b>Location:</b></td>
    <td><?php echo $BRIDGE->Location?></td>
  </tr>
  <tr>
    <td><b>Opened:</b></td>
    <td><?php echo $BRIDGE->Opened?></td>
  </tr>
  <tr>
    <td><b>Spans:</b></td>
    <td><?php echo nl2br($BRIDGE->Spans)?></td>
  </tr>
  <tr>
    <td><b>Structural System:</b></td>
    <td><?php echo nl2br($BRIDGE->StructSystem)?></td>
  </tr>
</table>
</center>

<P>
<center>
<H3>Images</H3>
<table border=1 cellspacing=0 cellpadding=4>
<?php
  $rs = query( "SELECT * FROM Images ORDER BY id" );
  $n = 0;
  while( $img = mysql_fetch_object($rs) ) {
?>
    <tr>
      <th><?php echo ++$n?>.</th>
      <td><a href="image.php3?bridge=<?php echo $bridge?>&image=<?php
echo $img->Image?>">image_root?>/<?php echo $img->Image?>"></a></td>
      <td><a href="image.php3?bridge=<?php echo $bridge?>&image=<?php
echo $img->Image?>"><?php echo $img->Description?></a></td>
    </tr>
  <?php
    }

```

```

?>
</table>
</center>
<?php
    } else { /* image given, load only a specific one */
        $rs = query( "SELECT * FROM Images WHERE Image='$image'" );
        $img = mysql_fetch_object($rs);
        $src = $BRIDGE->image_root . "/" . $img->Image;
        $saltsrc = eregi_replace( "(\\.jpg)$", "-800\\1", $src );
        if( $saltsrc != " " && file_exists($saltsrc) )
            $src = $saltsrc;
    }
?>
<P>
<center>

<h3><?php echo $img->Description?></h3>
<P>
<b>[ <a href="image.php3?bridge=<?php echo $bridge?>">Thumbnails</a>
]</b>
</center>
<?php
    }
?>
<?php include 'footer.inc'?>

```

A11 download.php3

This code has been in written in php3 language. This code controls the different options available for downloading the sensor data. It helps the user to choose the desired format for downloading the sensor readings.

```

<?php /* Hey! Emacs! -*- tab-width:4 -*- */ ?>
<?php
    include "common.inc";

    if( !is_array($sensors) ) {
        if( $sensors == " " )
            die( "No sensors selected." );
        else

```

```

        $sensors = Array( $sensors );
    }
    $cols = join( ',', $sensors );

    $needfooter = 0;
    if( $destination == 'screen' ) {
        if( $format == 'html' ) {
            $needfooter = 1;
            include 'header.inc';
        } else {
            header( "Content-Type: text/plain" );
        }
    } else {
        header( "Content-Type: application/x-" . $format );
        header("Content-disposition: filename=$table.$format");
    }

    $query = "SELECT $cols FROM $table";
    if( $rows != 'all' ) {
        $query .= sprintf( " LIMIT %d,%d", $startrow-1, $numrows );
    }
    $rs = query( $query );
    include "download-$format.inc";

    if( $needfooter )
        include "footer.inc";
?>

```


A12 download-csv.inc

This is a text file. This code controls the downloading of the data in CSV format.

```
<?php
echo "", join("","$sensors), "", "\n";

while( $row = mysql_fetch_row($rs) ) {
    $delim = "";
    while( list($i,$val) = each($row) ) {
        echo $delim;
        $delim = ',';
        if( ereg( '[^+0123456789\\.]', $val ) )
            echo "", $val, "";
        else
            echo $val;
    }
    echo "\n";
}
?>
```

A13 download-html.inc

This is a text file. This code controls the downloading of the data in HTML format.

```
<script language="javascript">
function help( str ) {
    window.open("sensorinfo.php3?bridge=<?php echo $bridge?>&sensor=" +
str, "_help",
"toolbar=no,menubar=no,scrollbars=yes,width=400,height=300,resizable=yes
");
    return false;
}
</script>
<table border=1 cellspacing=0 cellpadding=2>
```

```

<tr bgcolor="#E0E0E0" valign=top>
<?php
    for( $i = 0; $i < sizeof($sensors); $i++ ) {
        $s = sensorinfo($sensors[$i]);
        echo "    <th><A HREF=\"\" ONCLICK=\"return help(\"", $s->sens_num,
        \"\")\">\", $s->sens_num, "</A></th>\n";
    }
?>
</tr>
<?php
    while( $row = mysql_fetch_row($rs) ) {
        echo "    <tr align=right>\n";
        while( list($i,$val) = each($row) )
            echo "        <td>\", $val, "</td>\n";
        echo "    </tr>\n";
    }
?>
</table>

```

A14 download-txt.inc

This is a text file. This code controls the downloading of the data in text format.

```

<?php
    echo join("\t",$sensors), "\n";

    while( $row = mysql_fetch_row($rs) ) {
        echo join("\t",$row), "\n";
    }
?>

```

APPENDIX B

UNIVERSAL FILE FORMAT

A Universal File is a physical file in ASCII format, containing symbolic data in physical records with a maximum record length of 80 characters. On the physical file, data is contained in logical data sets with the following characteristics :

1. The first record of the data set contains “-1” right justified in columns 1 through 6. Columns 7 through 80 of the physical record are blanks.
2. The second record of the data set contains the numeric range, 1 through 32767, right justified in columns 1 through 6. Columns 7 through 80 of this physical record are blanks.
3. The last record is identical to the first record.
4. The specification of data on the remaining records of the data set are totally dependent on the data set type.

For example :

bbbb-1

bbxxx

...

...(data pertaining to the data set type ‘xxx’)

...

bbbb-1

Following is a brief description of UFF-58.

UNIVERSAL DATASET : TYPE 58

Description: Function at Nodal DOF

Record 1: FORMAT (80A1)

Field 1 ID Line 1

NOTE: ID Line 1 is generally used for the Function description.

Record 2: FORMAT (80A1)

Field 1 ID Line 2

Record 3: FORMAT (80A1)

Field 1 ID Line 3

NOTE: ID Line 3 is generally used to identify when the function was created. The date is in the form DD-
MMM-YY, and the time is in the form HH:MM:SS,
with a general Format (9A1,1X,8A1).

Record 4: FORMAT (80A1)

Field 1 ID Line 4

Record 5: FORMAT (80A1)

Field 1 ID Line 5

Record 6: FORMAT (2(I5,I10),2(1X,10A1,I10,I4))

DOF Identification

Field 1	Function Type
0	: General or Unknown
1	: Time Response
2	: Auto Spectrum
3	: Cross Spectrum
4	: Frequency Response Function
5	: Transmissibility
6	: Coherence
7	: Auto Correlation
8	: Cross Correlation
9	: Power Spectral Density (PSD)
10	: Energy Spectral Density (ESD)
11	: Probability Density Function
12	: Spectrum
13	: Cumulative Frequency Distribution
14	: Peaks Valley
15	: Stress/Cycles
16	: Strain/Cycles
17	: Orbit
18	: Mode Indicator Function
19	: Force Pattern
20	: Partial Power
21	: Partial Coherence
22	: Eigenvalue
23	: Eigenvector
24	: Shock Response Spectrum
25	: Finite Impulse Response Filter
26	: Multiple Coherence
27	: Order Function

Field 2 **Function Identification Number**

Field 3 **Version Number, or sequence number**

Field 4 **Load Case Identification Number**

 0 : **Single Point Excitation**

Field 5 **Response Entity Name ("NONE" if unused)**

Field 6 **Response Node**

Field 7 **Response Direction**

 0 : **Scalar**

 1 : **+X Translation** 4 : **+X Rotation**

 -1 : **-X Translation** -4 : **-X Rotation**

 2 : **+Y Translation** 5 : **+Y Rotation**

 -2 : **-Y Translation** -5 : **-Y Rotation**

 3 : **+Z Translation** 6 : **+Z Rotation**

 -3 : **-Z Translation** -6 : **-Z Rotation**

Field 8 **Reference Entity Name ("NONE" if unused)**

Field 9 **Reference Node**

Field 10 **Reference Direction (same as field 7)**

NOTE: Fields 8, 9, and 10 are only relevant if field 4 is zero.

Record 7: **FORMAT (3I10,3E13.5)**

Data Form

Field 1 **Ordinate Data Type**

 2 - **real, single precision**

 4 - **real, double precision**

 5 - **complex, single precision**

 6 - **complex, double precision**

Field 2 **Number of data pairs for uneven abscissa spacing, or number of data values for even abscissa spacing**

Field 3 **Abscissa Spacing**

0 - uneven

1 - even (no abscissa values stored)

Field 4 **Abscissa minimum (0.0 if spacing uneven)**

Field 5 **Abscissa increment (0.0 if spacing uneven)**

Field 6 **Z-axis value (0.0 if unused)**

Record 8: **FORMAT (I10,3I5,2(1X,20A1))**

Abscissa Data Characteristics

Field 1 **Specific Data Type**

0 - unknown

1 - general

2 - stress

3 - strain

5 - temperature

6 - heat flux

8 - displacement

9 - reaction force

11 - velocity

12 - acceleration

13 - excitation force

15 - pressure

16 - mass

17 - time

18 - frequency

19 - rpm

20 - order

Field 2 Length units exponent

Field 3 Force units exponent

Field 4 Temperature units exponent

NOTE: Fields 2, 3 and 4 are relevant only if the Specific Data Type is General, or in the case of ordinates, the response/reference direction is a scalar, or the functions are being used for nonlinear connectors in System Dynamics Analysis.

See "Addendum A" for the unit exponent table

Field 5 Axis label ("NONE" if not used)

Field 6 Axis units label ("NONE" if not used)

NOTE: If fields 5 and 6 are supplied, they take precedence over program generated labels and units.

Record 9: FORMAT (I10,3I5,2(1X,20A1))

Ordinate (or ordinate numerator) Data Characteristics

Record 10: FORMAT (I10,3I5,2(1X,20A1))

Ordinate Denominator Data Characteristics

Record 11: FORMAT(I10,3I5,2(1X,20A1))

Z-axis Data Characteristics

NOTE: Records 9, 10, and 11 are always included and have fields the same as record 8. If records 10 and 11 are not used, set field 1 to zero.

Record 12:

Data Values

Case	Ordinate Type	Precision	Abscissa Spacing	Format
1	Real	Single	Even	6E13.5
2	Real	Single	Uneven	6E13.5
3	Complex	Single	Even	6E13.5
4	Complex	Single	Uneven	6E13.5
5	Real	Single	Even	4E20.12
6	Real	Single	Uneven	2(E13.5,E20.12)
7	Complex	Single	Even	4E20.12
8	Complex	Single	Uneven	E13.5,2E20.12

NOTE: There are 8 distinct combinations of parameters that affect the details of READ/WRITE operations. The parameters involved are Ordinate Data Type, Ordinate Data Precision and Abscissa Spacing. These combinations have not been documented here. The reader is encouraged to visit www.sdrl.uc.edu/UFF2/58.asc to gain further knowledge.

GENERAL NOTES

- 1. ID lines may not be blank. If no information is required, the word "NONE" must appear in columns 1 through 4.**
- 2. ID line 1 appears on plots in Finite Element Modeling and is used as the function description in System Dynamics Analysis.**
- 3. Data loaders use the following ID line conventions :**
 - a) ID Line 1 - Model Identification**
 - b) ID Line 2 - Run Identification**
 - c) ID Line 3 - Run Date and Time**
 - d) ID Line 4 - Load Case Name**
- 4. Coordinates codes from MODAL-PLUS and MODALX are decoded into node and direction.**
- 5. Entity names used in System Dynamics Analysis prior to I-DEAS Level 5 have a 4-character maximum. Beginning with Level entity names will be ignored if this dataset is preceded by dataset 259. If no dataset 259 precedes this dataset, then the entity name will be assumed to exist in model bin number 1.**
- 6. Record 10 is ignored by System Dynamics Analysis unless case = 0. Record 11 is always ignored by System Dynamics Analysis.**
- 7. In record 6, if the response or reference names are "NONE" and are not overridden by a dataset 259, but the corresponding node is non-zero, System Dynamics Analysis adds the node and direction to the function description if space is sufficient.**

- 8. ID line 1 appears on XY plots in Test Data Analysis along with ID line 5 if it is defined. If defined, the axis units labels also appear on the XY plot instead of the normal labeling based on the data type of the function.**
- 9. For functions used with nonlinear connectors in System Dynamics Analysis, the following requirements must be adhered to:**
 - a) Record 6: For a displacement-dependent function, the function type must be 0; for a frequency-dependent function, it must be 4. In either case, the load case identification number must be 0.**
 - b) Record 8: For a displacement-dependent function, the specific data type must be 8 and the length units exponent must be 0 or 1; for a frequency-dependent function, the specific data type must be 18 and the length units exponent must be 0. In either case, the other units exponents must be 0.**
 - c) Record 9: The specific data type must be 13. The temperature units exponent must be 0. For an ordinate numerator of force, the length and force units exponents must be 0 and 1, respectively. For an ordinate numerator of moment, the length and force units exponents must be 1 and 1, respectively.**
 - d) Record 10: The specific data type must be 8 for stiffness and hysteretic damping; it must be 11 for viscous damping. For an ordinate denominator of translational displacement, the length units exponent must be 1; for a rotational displacement, it must be 0. The other units exponents must be 0.**
 - e) Dataset 217 must precede each function in order to define the function's usage (i.e. stiffness, viscous damping, hysteretic damping).**

ADDENDUM A

Length, force and temperature exponents must be supplied for a specific data so that the unit conversion can be correctly performed. i.e. Record 8 Field 1 = 1. For example, if the function has the physical dimensions of Energy (Force * Length), then the required exponents would be as follows:

Length = 1: Force = 1 : Energy = L * F : Temperature = 0

Table - Unit Exponents

Specific Data Type	Direction					
	Translational			Rotational		
	Length	Force	Temp	Length	Force	Temp
0	0	0	0	0	0	0
1	Requires input to fields 2,3,4					
2	-2	1	0	-1	1	0
3	0	0	0	0	0	0
5	0	0	1	0	0	1
6	1	1	0	1	1	0
8	1	0	0	0	0	0
9	0	1	0	1	1	0
11	1	0	0	0	0	0
12	1	0	0	0	0	0
13	0	1	0	1	1	0
15	-2	1	0	-1	1	0
16	-1	1	0	1	1	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0

NOTE: Units exponents for scalar points are defined within System Analysis prior to reading this dataset.

ADDENDUM B

There are 8 distinct combinations of parameters that affect the details of READ/WRITE operations. The parameters involved are Ordinate Data Type, Ordinate Data Precision and Abscissa Spacing. These combinations have not been documented here. The reader is encouraged to visit www.sdrl.uc.edu/UFF2/58.asc to gain further knowledge.