

Spring 2011

Development of a structural parameter estimation program for finite element model updating

John A. Welch

University of New Hampshire, Durham

Follow this and additional works at: <https://scholars.unh.edu/thesis>

Recommended Citation

Welch, John A., "Development of a structural parameter estimation program for finite element model updating" (2011). *Master's Theses and Capstones*. 645.

<https://scholars.unh.edu/thesis/645>

This Thesis is brought to you for free and open access by the Student Scholarship at University of New Hampshire Scholars' Repository. It has been accepted for inclusion in Master's Theses and Capstones by an authorized administrator of University of New Hampshire Scholars' Repository. For more information, please contact nicole.hentz@unh.edu.

**DEVELOPMENT OF A STRUCTURAL PARAMETER ESTIMATION PROGRAM FOR FINITE
ELEMENT MODEL UPDATING**

BY

JOHN A. WELCH

BS, University of New Hampshire, May, 2007

THESIS

**Submitted to the University of New Hampshire
in Partial Fulfillment of
the Requirements for the Degree of**

Master of Science

In

Civil Engineering

May, 2011

UMI Number: 1498976

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

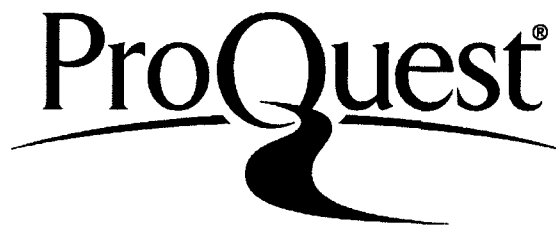
In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



UMI 1498976


Copyright 2011 by ProQuest LLC.

All rights reserved. This edition of the work is protected against unauthorized copying under Title 17, United States Code.



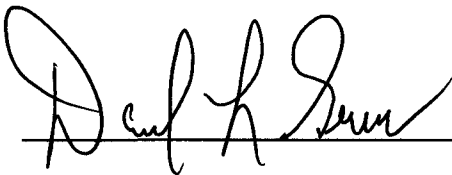
ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

This thesis has been examined and approved.



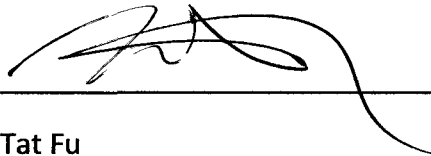
Thesis Director, Erin S. Bell

Associate Professor of Civil Engineering



David L. Gress

Professor of Civil Engineering



Tat Fu

Assistant Professor of Civil Engineering

May 6 2011

ACKNOWLEDGMENTS

This research was funded by the National Science Foundation. My advisor, Dr. Erin Bell, provided me the support and motivation needed to meet the goals of my research, while challenging me along the way. Your dedication helped me to reach not only the goals of my research, but my personal goals of higher education. Jesse Sipple was a great research partner. More than that, you were a great classmate and friend. Thank you for making my graduate studies an enjoyable experience. Finally, I want to thank my friends and family for their support throughout this long adventure. Mom and Dad thank you for always encouraging me to succeed and for supporting me along the way. I love you and couldn't have done this without you. Thank you all for your encouragement and support.

Table of Contents

ACKNOWLEDGMENTS.....	iii
List of Tables	vii
List of Figures	vii
List of Equations.....	ix
ABSTRACT.....	xi
Development of a Structural Parameter Estimation Program for Finite Element Model Updating	xi
CHAPTER 1	1
Introduction	1
1.1 Motivation and Social Need	2
1.2 Literature Survey.....	5
1.3 Bridge Condition Assessment	11
1.4 State of the Art.....	12
1.4.1 Instrumentation	12
1.5 Goals and Major Contributions of the Research.....	16
CHAPTER 2	21
MUSTANG	21
2.1 Parameter Estimation	22
2.1.1 Programmed Error Functions.....	24
2.1.2 Minimization of the Error Function	29
2.1.3 Future Modal Error Functions.....	31
2.2 SAP® API	32
2.3 Programming Logic.....	33
2.4 Key Modular Subroutines.....	35
2.5 Static Stacking	36
Chapter 3	38

Program Verification Models.....	38
3.1 Cantilever Beam	39
3.1.1 One Element Unknown	39
3.1.2 Two Element One Unknown	41
3.1.3 Two Element Two Unknowns	42
3.1.4 Two Element Cantilever with an Internal Spring Hinge.....	44
3.1.5 Two Element Cantilever with a Spring Support	45
3.1.6 Two Element Cantilever with a Pin Connection and End Spring Support	46
3.2 Published Parameter Estimation Models.....	47
3.2.1 Two Dimensional Truss	48
3.2.2 Two Dimensional Bridge Frame	51
3.3 Future Work and Conclusions	61
CHAPTER 4	63
Field Example Bent 57	63
4.1 Background	63
4.2 Model	66
4.3 Simulated Data	67
4.4 Field Data	70
4.4.1 Collected Results	70
4.4.2 MUSTANG Results	70
4.5 Future Work	71
Chapter 5	72
Rollins Road Bridge	72
5.1 Background	73
5.2 April 2008 Field Test.....	77
5.2.1 Data Quality.....	81
5.2.2 Environmental Effects	81
CHAPTER 6	84
Rollins Road Modeling and Parameter Estimation.....	84

6.1 Rollins Road Bridge Model, Sipple (2008).....	85
6.1.1 Removal of Effects Due to Prestressing and Dead Load.....	87
6.1.2 Results Sipple (2008).....	88
6.1.3 Challenges for MUSTANG.....	90
6.2 Simulated Data Runs.....	91
6.2.1 Creating a MUSTANG Compatible Model.....	91
6.2.2 Bridge Model for MUSTANG.....	95
6.2.3 Parameter Estimation with Simple Model.....	96
6.3 Discussion.....	98
CHAPTER 7.....	100
Conclusion.....	100
7.1 Contribution.....	100
7.1.1 MUSTANG – Verification by Published Results.....	101
7.1.2 MUSTANG – Field Collected Data.....	102
7.2 Observations and Areas of Importance.....	102
7.2.1 Element Types.....	102
7.2.2 Loading.....	103
7.3 Future of MUSTANG.....	103
7.3.1 Future Programming.....	105
Works Cited.....	109
APPENDICES.....	111
APPENDIX A – TWO DIMENSIONAL BRIDGE PARAMETER ESTIMATION, CASE 1..	112
APPENDIX B – TWO DIMENSIONAL BRIDGE PARAMETER ESTIMATION, CASE 2..	113
APPENDIX C – TWO DIMENSIONAL BRIDGE PARAMETER ESTIMATION, CASE 3..	114
APPENDIX D – TWO DIMENSIONAL BRIDGE PARAMETER ESTIMATION, CASE 4..	115
APPENDIX E – BENT 57 FIELD DATA.....	116
APPENDIX F – BENT 57 DATA FILE.....	117

List of Tables

Table 1: One Element Pine Cantilever Results.....	41
Table 2 : Two Dimensional Truss Parameter Estimation Data, Sanayei et al. (1991) .	49
Table 3: Comparison of MUSTANG Results to Published Results from Sanayei et al. (1991)	50
Table 4 : Initial Structural Parameters and Damage Scenarios, Sanayei et al. (1999)	53
Table 5: SAP Model Load Case Locations.....	53
Table 6 : Two Dimensional Bridge Parameter Estimation Summary.....	60
Table 7: MUSTANG Iterations for Bent 57, Simulated Data.....	69
Table 8 : Rollins Road 2000 Inspection Report (NHDOT 2007)	74
Table 9: Simple Bridge Results	98

List of Figures

Figure 1: Graphical representation of parameter estimation adapted from Sipple (2008)	7
Figure 2: SHM Instrumentation Tsing Ma Bridge (Xu et al. 2007)	9
Figure 3: SAP2000® Bridge Modeler Wizard	14
Figure 4: Bridge Modeler Information	15

Figure 5: Bridge Maintenance Schematic	16
Figure 6: MATLAB® to SAP® Link	17
Figure 7: Bent 57, C.A. Underpinning, Boston MA (Bell et al. 2008)	19
Figure 8: Rollins Road Field Test.....	20
Figure 9: MUSTANG Flow Chart	23
Figure 10: Error Function Vectorization, Bell et al. (2003)	29
Figure 11: One Element Pine Cantilever Model.....	40
Figure 12: Two Element One Unknown Pine Cantilever	42
Figure 13: Two Element Cantilever Two Parameter Unknown Deflected Shape	43
Figure 14: Two Element Cantilever with Internal Unknown Spring	44
Figure 15: Two Element Cantilever with Spring Hinge Support.....	45
Figure 16: Two Element Cantilever with End Vertical Spring Support	46
Figure 17: Two Dimensional Truss, DOF, Sanayei et al. (1991).....	48
Figure 18 : Two Dimensional Bridge Frame, Sanayei et al. (1999).....	52
Figure 19: SAP Two Dimensional Bridge Frame Model, LC1	54
Figure 20: Two Dimensional Bridge Parameter Estimation Convergence, Members Case 1	55
Figure 21: Two Dimensional Bridge Parameter Estimation Convergence, Springs Case 1	55
Figure 22: Two Dimensional Bridge Parameter Estimation Convergence, Case 2	56
Figure 23: Two Dimensional Bridge Parameter Estimation Convergence, Case 3	58
Figure 24: Two Dimensional Bridge Parameter Estimation Convergence, Case 4	59

Figure 25 : Boston Central Artery / Big Dig Construction (Photo Courtesy of PBS)	64
Figure 26 : Bent 57 Load Test Setup.....	65
Figure 27 : Bent 57 SAP2000® Model.....	67
Figure 28: Bent 57 Gauge Locations, Santini-Bell et al (2008)	68
Figure 29: Deflected Shape for Bent 57, Load Case 1.....	69
Figure 30: Rollinsford, New Hampshire (Google Maps ®).....	73
Figure 31: Rollins Road Bridge.....	76
Figure 32: Trooper Huddleston (NH State Police) Mobile Weigh Station	78
Figure 33: Loaded Truck, Survey Team in Bucket Truck, and Technical Support	80
Figure 34 : Girder 3 Raw and Empirical Data (Sipple 2008).....	82
Figure 35 : Rollins Road Bridge SAP2000® Model.....	88
Figure 36: Simple Bridge Model (SAP 2011)	96
Figure 37: Simple Bridge Wheel Loads (SAP 2011).....	97

List of Equations

Equation 1 : Force-Displacement Relationship	24
Equation 2 : Static Stiffness Error Function.....	25
Equation 3: Static Stiffness Force Comparison.....	25
Equation 4 : Static Condensation	26
Equation 5 : Static Condensation Result	26

Equation 6: Static Flexibility	27
Equation 7: Static Flexibility Error Function	27
Equation 8: Static Strain Error Function	27
Equation 9: Strain Calculation	28
Equation 10 : Objective Function	29
Equation 11 : Linearized Error Function	30
Equation 12 : Sensitivity	30
Equation 13 : Scalar Objective Function.....	30
Equation 14 : Minimized Scalar Objective Function	31
Equation 15: Static Stacking	36

ABSTRACT

Development of a Structural Parameter Estimation Program for Finite Element Model Updating

By

John A. Welch

University of New Hampshire, May, 2011

The condition of America's infrastructure is highlighted by major collapses and overcrowded roadways remind us that our infrastructure is aging and in need of effective maintenance. The American Society of Civil Engineers report card for 2009 graded the nation's bridges as a C. In this period of renovation, rebuilding and limited funding, it is important to use the latest technologies to help make America's roadways safe and establish efficient management protocols. This research develops a program for the purpose of pairing structural health monitoring systems with the power of structural modeling, for the use of model updating and parameter estimation, can help to create a smarter and more efficient method of bridge health monitoring and management. A current and accurate analytical bridge model can help owners assess structural needs as they arise. A first step towards this goal is the creation of a program that utilizes field measurements, bridge inspection reports, analytical structural modeling and the powerful computer based structural model updating methods for bridge condition assessment.

CHAPTER 1

Introduction

The purpose of this research is to create a model updating parameter estimation program. This research takes up the task of creating such a program in a manner that allows for expansion given future research and methods. It also looks into what type of structural modeling will best suit parameter estimation. Throughout the process it becomes evident that bridge instrumentation and modeling is most effective when incorporated from the beginning during the design process. This represents a paradigm shift in bridge management. Monitoring and modeling has occurred at the end of a structures life to ensure safe function. Including these from the beginning of a bridge design will provide a powerful tool for bridge owners.

1.1 Motivation and Social Need

This research was funded by a National Science Foundation (NSF) career grant, titled: Integrating Structural Health Monitoring, Intelligent Transportation Systems and Model Updating into a Bridge Condition Assessment Framework. The driving idea behind this NSF project is to take these elements of bridge design and monitoring and combine them into a useful tool for determining a bridge system's health under working loads and environmental conditions. In practice vast amounts of data related to bridge performance are collected but are maintained by separate entities. There is limited, if any, sharing or combining of information. Combining these sets of data creates an extensive base of information from which structural conditions and integrity could be assessed. The post processed data produces a measured bridge response that is compared to a set of analytically predicted responses to determine the condition of the bridge. Structural health monitoring (SHM) instrumentation is used to record conditions, including environmental loading, that occur within a structure. Intelligent transportation systems (ITS) instrumentation can be used to determine traffic conditions and the loading that cause the SHM responses collected by field instruments. The predicted response of the model under these loads can be used to develop the framework necessary to properly determine a bridge's structural integrity and physical condition.

The 1956 Interstate Highway Program expanded the U.S. highway system to include over 500,000 bridges. At the time many of these bridges were designed based

on a design life of fifty years. There was no comprehensive structural health monitoring system included in the original design. Currently many of these bridges have exceeded or are rapidly approaching the end of their useful life. Because of this, many bridges are now in need of major structural repairs and rehabilitations, or complete replacement. Just as in the 1950's and 1960's, a major re-construction effort is critical to the continued performance and safety of our infrastructure.

The American Society of Civil Engineers has released their report card for 2009. The ASCE report card grades the nation's bridges as a C. The report card continues to state the twenty six percent of the nation's bridges are classified as either structurally deficient or functionally obsolete. According to the United States Department of Transportation, "Structurally deficient means there are elements of the bridge that need to be monitored and/or repaired. The fact that a bridge is "deficient" does not imply that it is likely to collapse or that it is unsafe. It means they must be monitored, inspected and maintained." Without immediate response to these structural deficiencies, serious structural concerns will develop with these bridges. "A functionally obsolete bridge has older design features and, while it is not unsafe for all vehicles, it cannot safely accommodate current traffic volumes, and vehicle sizes and weights." (ASCE 2005) Urban areas are seeing the largest increase these numbers. The ASCE estimates that an annual investment of seventeen billion dollars is required to see significant improvement in the nation's bridges. Currently only 10.5 billion dollars are being invested annually. (ASCE 2009)

Recent structural failures of transportation components have focused the public interest on our infrastructure and its structural health. These failures have spotlighted our bridge reliability and structural integrity assessment protocols. The Mississippi River Bridge, I-35W, not only highlighted the need for consistent bridge inspection, it also illustrated a situation where an in place structural health monitoring system combined with a model updating parameter estimation program could have forewarned of the overstressed state of the bridge and the bridge's impending failure.

President Obama has signed the American Recovery and Reinvestment Act of 2009. This act allocated 26.6 billion to the states for highway investment. (FHWA) A little more than thirteen billion has already been issued and 3,870 transportation projects have been authorized. (FHWA June 3rd, 2009) Given the need for reconstruction and the newly available funding, now is an ideal time to incorporate structural health monitoring systems in the bridge design and construction process from the beginning. Structural health monitoring systems can also be applied to bridges that are being retrofit.

Research into structural health monitoring and the post-processing of collected field measurements for transportation system management has significantly increased in response to the growing demand to evaluate the structural integrity of United States' highway bridges. An in place structural health monitoring system can provide useful data in determining the structural integrity of a bridge. It can be used as a tool

that supplements visual inspection as well as paint a system-based picture of a bridges health to the bridge owner.

Structural health monitoring systems provide a multitude of data. This data needs to be post-processed in order to be useful to a department of transportation or bridge owner. Camera based measurements are an example of this. Software is required to extract measurement data from captured images. Once the data has been post-processed it then can be compared to a base model to determine structural health. Manual parameter estimation is time consuming and difficult. An integrated system that passes data from SHM and ITS sources to a program capable of parameter estimation and model updating would be a great time saver.

1.2 Literature Survey

Parameter Estimation is the inverse to direct structural analysis. With structural analysis elements, the physical properties and behavioral parameters are known. The physical properties are comprised of area, moment of inertia, and modulus of elasticity. The parameters include axial stiffness (EA), rotational stiffness (EI), and torsional rigidity (GJ). The next step is to apply loads to these elements. Due to the complexity of physical structures, such as bridges, this is usually done with the aid of a computer finite element modeling and structural analysis program. The response of the structure is then calculated and compared to acceptable limits.

Parameter estimation operates in the reverse order. The parameters are close estimates intended to resemble the structure's actual properties as built in the field. The purpose of parameter estimation is to take these estimates and modify them until they reflect their "true" values. Parameter estimation takes into account the behavior of the entire system by using the bridge response, whereas a bridge inspection can focus on the status of one element and not how it affects the entire system. Loads are applied to the structure through the use of a nondestructive field test. During this field test the structural response is measured in the form of displacements, rotations and strains. These measurements are taken at critical locations along the structure. The data are post processed and compared to the finite element model's predicted response. The difference in parameter is calculated using published error functions. These error functions combined with a model updating algorithm determine the "true" parameters of the structure.

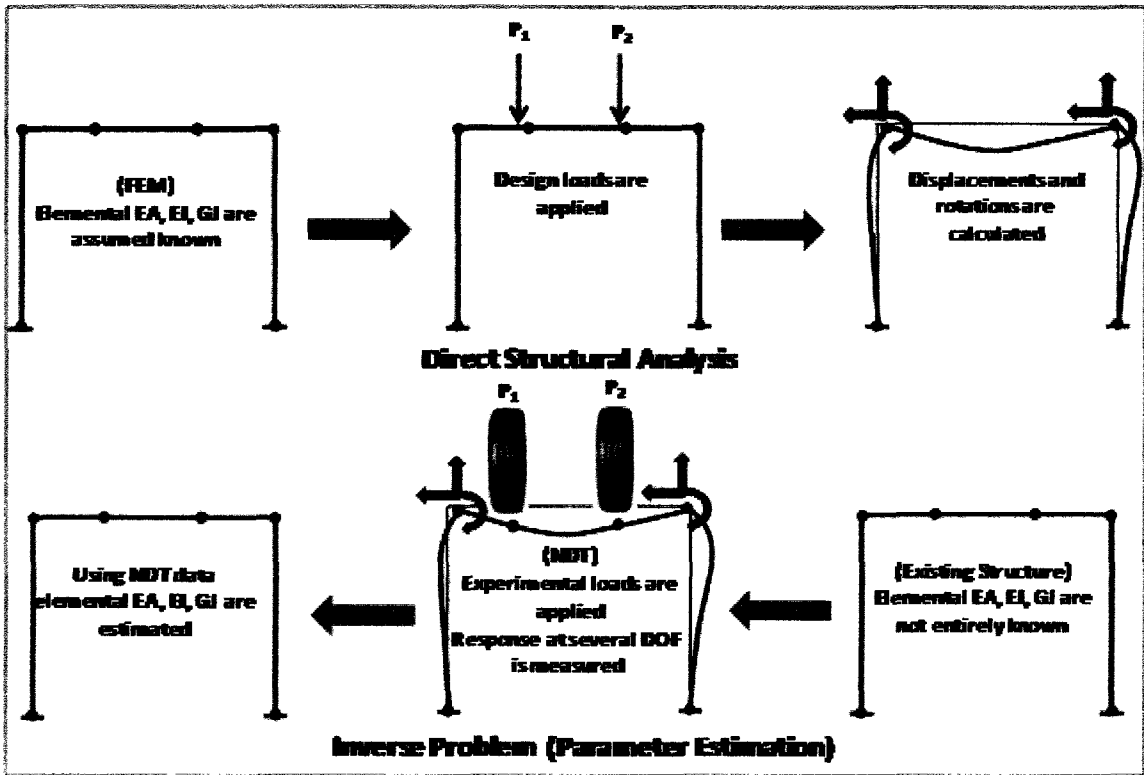


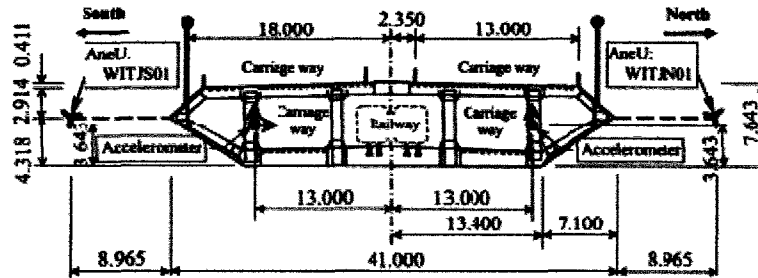
Figure 1: Graphical representation of parameter estimation adapted from Sipple (2008)

According to Sanayei et al. (1999), "Parameter estimation is the process of reconciling an analytical model of a structure with nondestructive test (NDT) data using optimization methods." Parameter estimation shows how the structure behaves in the field as opposed to in the theory of design. Parameter estimation could be used effectively with a structural health monitoring system to create an accurate representation of a bridge, or other structures, in service. This process is an effective way to identify changes in structural stiffness's that can not be observed by visual inspection. Parameter estimation works by adjusting the mass and stiffness of members associated with field measurements until the model deflections reflect the field observations. Comparing these adjusted values with the set of design drawings

can reveal a range of damage from degradation of components to impending/actual failure. Parameter estimation can be affected by field measurement error and modeling inaccuracies. Statistical methods are used to mitigate error.

Structural health monitoring systems use a wide variety of instrumentation to capture a bridge's response. A good example of this is the Tsing Ma Bridge in Hong Kong. This bridge is fitted with a wind and structural health monitoring system (WASHMS). The WASHMS is comprised of anemometers, accelerometers, level sensors, and strain gauges. Anemometers measure wind speed and direction. The Tsing Ma Bridge supports the transportation of automobiles as well as trains. The instrumentation was used to collect data during Typhoon York, which struck September 16, 1999. This Typhoon presented an opportunity to capture bridge response under several different types of loading. The first case loading was no train passing over the bridge and cars banned from travel due to high wind. The second case there was one train passing over. In the third case there were two trains running in opposite directions. The fourth case had two south trains and one north train. All load cases had high cross winds. (Xu et al. 2007) These different load applications and measurement of response create a well defined system for comparison to a base model and parameter estimation. Such a well defined problem would be impossible without a comprehensive SHM system in place. Parameter estimation has a higher likelihood of success given more data to process. Any unknown measurements are removed from the parameter estimation process via inverting matrices. This introduces error into the parameter estimation process. A comprehensive SHM system

of this nature coupled with a structural model and model updating program would be a useful tool for condition assessment.



(a) Positions of sensors on cross section of bridge deck

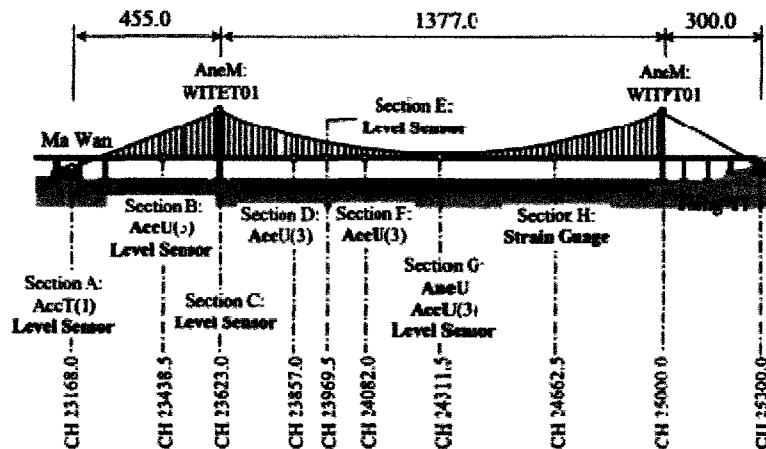


Figure 2: SHM Instrumentation Tsing Ma Bridge (Xu et al. 2007)

Watson et al. (2007) present an excellent example of using SHM and ITS to measure bridge response due to nondestructive load testing. The test used a combination of GPS receivers, digital video cameras, anemometer, and temperature measurements. The GPS measurements and video images were synced to within 1s of each time stamp. The data was compared to an analytical model created in Space Gass. The model underestimated the measured field displacements. This is where a model updating program would prove to be of value by adjusting the model's

parameters to reflect field measurements and give the bridge owner a more accurate picture of the bridge's state. Underestimated deflections signify overestimated strength and can suggest the structure is weaker than predicted.

An area that would benefit greatly from the development of an integrated structural health monitoring system, intelligent transportation system, and model updating program would be the field of transportation asset management. Asset management uses available data from information systems in combination with financial and economic analysis tools to maximize physical performance of capital assets. It also takes into consideration the operation and maintenance costs associated with the assets. (Gifford et al. 2003) For transportation asset management the capital assets would include roads and bridges. A wealth of information would be available from the combined monitoring and modeling system. All of this data could be used to compare performance of certain materials and methods to their associated costs and benefits to the overall infrastructure system. Gifford points out the usefulness of a well developed cost-benefit relationship given the current economic status and funding reductions facing infrastructure.

1.3 Bridge Condition Assessment

Routine bridge inspections are required to identify bridge elements in need of repair and elements in need of routine maintenance. Bridge inspections take place every twenty four months on “healthy” bridges. Given a bridge’s rating these inspections can be required in shorter intervals. The inspections are a visual process and can only identify visible deficiencies or damage in visible elements. Damage that is not immediately apparent can occur at any time between the inspections. An overloaded truck can cause serious damage that would not be apparent at the time of overload. This is where an integrated monitoring system would prove invaluable. A weigh in motion station at the bridge approach could measure an overloaded truck and trigger a warning within the monitoring system. This system could wirelessly notify the owner, or group in charge of monitoring the bridge, of the overload situation. The owner could then acquire ITS and SHM real time data and compare it to the modeled response of the “healthy” bridge. Any discrepancies could indicate damage. The severity of this damage would determine the appropriate response. This could vary from requiring immediate inspection to moving the next scheduled inspection to a sooner date. It is also possible that no action would be required. Without an interactive system and model updating protocol in place it is possible that the damage from an overloaded truck could go unnoticed for two years.

1.4 State of the Art

Parameter estimation is not a new concept. There are several programs that can take field data and perform parameter estimation on finite element models. PARIS©, Sanayei et al. (1998), is a program that was created at Tufts. Paris© uses MATLAB® to create the finite element model and to perform the parameter estimation. Paris© uses matrix algebra to assemble stiffness matrices for a model comprised of truss, frame, partially restrained frame, and/or spring elements in two or three dimensions. A data file takes in the elements properties, joint coordinates, and element connectivity. PARIS© is capable of accepting post-processed field measurements or using “true” parameters to create simulated data for parameter estimation. Static and modal analysis can be performed, Sanayei et al. (1998).

DIAMOND® is also a MATLAB® based parameter estimation program. This program is capable of modal analysis. It can identify the modal properties of a finite element model when it is subjected to a dynamic load. DIAMOND® performs damage detection using strain energy or flexibility analysis algorithms, Los Alamos et al. (1997).

1.4.1 Instrumentation

This research has lead to the development of the MATLAB® based program MUSTANG (Model Updating STructural ANalysis proGram.) MUSTANG takes advantage of the computational capabilities of MATLAB®. MATLAB® is a powerful computational software package. It is capable of handling the massive matrix algebra required for

parameter estimation of finite element models and linking with structural modeling software such as SAP2000®.

SAP2000® is a user-friendly structural finite element modeling program capable of structural analysis and design. Options in SAP2000® include modeling with various element types, performing analysis given many different types of load combinations, and being controlled by the application programming interface (API). Loading can vary from simple point loads to complex thermal loading. The key component of SAP2000® is its open API. This allows SAP2000® to share model information and analysis results with other programs. The SAP2000® API allows other programs to call its functions and run the program remotely. This allows a user to write a program in a compatible language and allow that program to run without any user interaction. One very useful component of SAP2000® is its bridge modeler. Bridges are very complex structures with different types of elements and connections. The bridge modeler helps to simplify the modeling process and provide an accurate visual representation that can help verify the models accurate representation of the bridge being modeled.

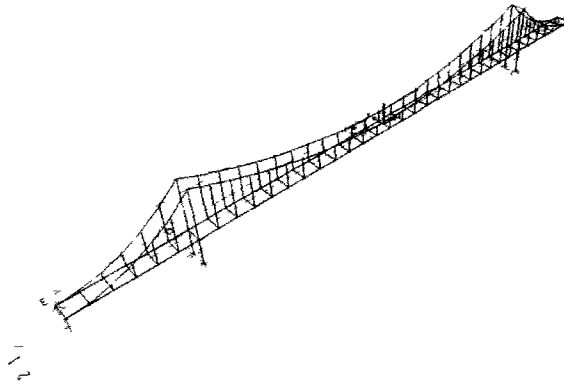


Figure 3: SAP2000® Bridge Modeler Wizard

MATLAB® and SAP2000® are industry partners. This means that the MATLAB® programming language is compatible with the SAP® API. This allows MUSTANG to call SAP® and run models, extract data, calculate updated parameters, and send these parameters back to SAP® to update the model. This updated model now more accurately reflects the “true” field parameters. A bridge’s health can now be determined from this updated model.

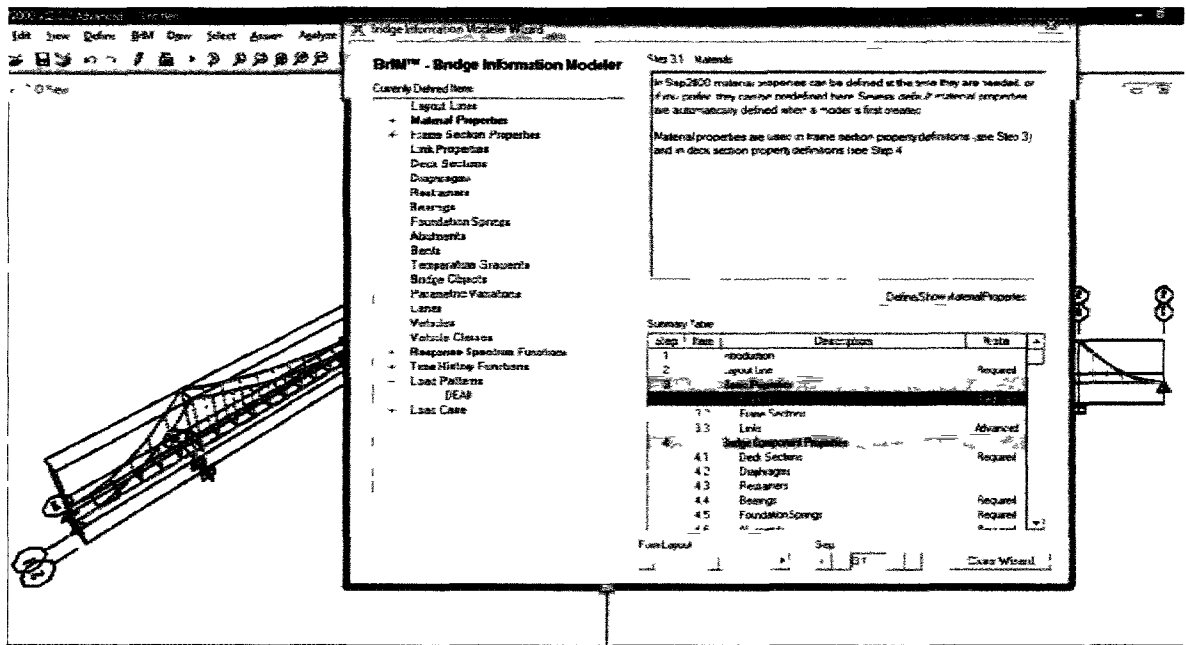


Figure 4: Bridge Modeler Information

1.5 Goals and Major Contributions of the Research

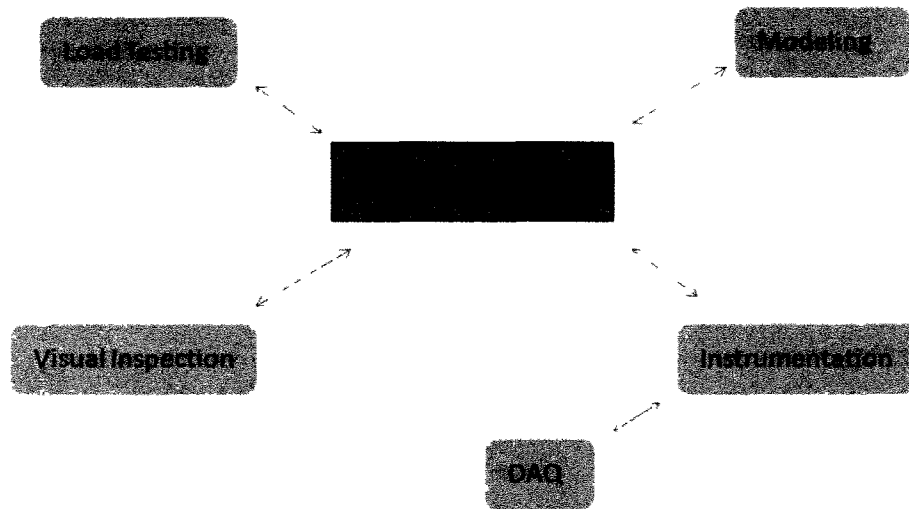


Figure 5: Bridge Maintenance Schematic

The purpose of this research is to create a robust model updating program using parameter estimation and finite element modeling for civil engineering purposes. This model updating program will combine two powerful computation and modeling software packages, MATLAB® and SAP2000®. The SAP2000® API will facilitate the link between the two industry partner software packages. The modeling and analysis will be done using SAP®. The parameter estimation computations and initiation of data transfer is programmed within the MATLAB® program. MUSTANG is a MATLAB® based modular code that utilizes SAP® API functions. The capability of the SAP® API functions includes extracting model information such as node locations, units, and load cases. The API is also capable of extracting element properties such as

moment of inertia, area, and other relevant structural properties. Additionally the API is capable of running load cases, extracting results, modifying elements or individual member properties, and saving updated models. The results extracted include the stiffness matrix, mass matrix, and nodal displacements. This is a short list of the potential API functions.

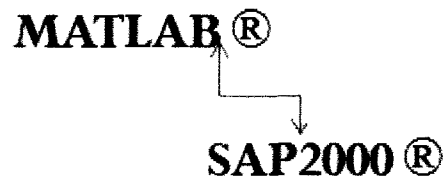


Figure 6: MATLAB® to SAP® Link

Parameter estimation scenarios using MUSTANG are present for both simple and complex structures. As with any new program or method, MUSTANG and its methods must be verified. Hand calculations are calculated and compared to MUSTANG's parameter estimation of the same scenarios. These comparisons show proof of concept that MUSTANG can perform simple model updating given a finite element model and a set of measurements. For the more complex cases that are not practically modeled by hand, there are several published cases of parameter estimation available for comparison, Sanayei et al. (1991). These cases have detailed models and results associated with them that will serve the purpose of independent

verification for MUSTANG. These results will serve as a benchmark for the validity of MUSTANG's parameter estimation capabilities using simulated data.

In addition to simulated, MUSTANG is capable of using field data for the purpose of parameter estimation. During the construction of the Big Dig, transfer bents were used to create an eight to ten lane cut-and-cover tunnel for excavation while the Central Artery in Boston remained in service. As part of graduate research at Tufts University a nondestructive field test was performed on a bent known as Bent57, Bell et al. (2008). The resulting data from this field test was entered into the Paris© program in order to estimate the rotational stiffness of the moment connections. The paper resulting from this will be used to verify MUSTANG's ability to use field data to perform parameter estimation.



Figure 7: Bent 57, C.A. Underpinning, Boston MA (Bell et al. 2008)

The final aspect of this research will be to attempt to perform parameter estimation using MUSTANG on the Rollins Road Bridge model. This bridge is located in Rollinsford, New Hampshire and has a structural health monitoring system in place. The SHM system in place measures strain and temperature. The goal of this SHM system was to measure performance of its FRP reinforcement. The goal was not to create a well defined system for parameter estimation. The bridge was modeled by Sipple (2008) in an effort to estimate bearing pad stiffness. MUSTANG will attempt to use simulated data to match the manual parameter estimation performed by Sipple

(2008) in an effort to show MUSTANG's capability of handling large and complex structures.



Figure 8: Rollins Road Field Test

CHAPTER 2

MUSTANG

The purpose of MUSTANG is to create a bridge between complex analytical parameter estimation and complex structural modeling to create a model updating parameter estimation algorithm. First MUSTANG was used to solve more basic problems, such as beams and frames with simulated data. With a modular core in place, MUSTANG has the potential for solving more complex models with simulated or field data. These models include bridges and field measurements obtained from field tests. Different error functions can be added as they emerge with future research.

2.1 Parameter Estimation

MUSTANG is programmed to perform parameter estimation using static error functions. An error function is a mathematical method to determine the change in parameter based on the difference of model responses versus the actual response of a structure/finite element model. These error functions include static stiffness, flexibility, and strain. There are many more published and well documented error functions than those that have been programmed into MUSTANG. These three were chosen due to the nature of the models being run. There are always new ways being invented to perform parameter estimation. This is why MUSTANG is programmed to be modular, easily modified and expanded. The main body of MUSTANG calls the user specified error function which is written in its own m-file. An m-file is a MATLAB® formatted file that can be run on its own, or be called from another m-file. Variables can be passed into and returned from an m-file upon it being called and run.

The process of parameter estimation begins with the original design. In order to ensure an accurate representation, modeling must be involved with parameter estimation from the beginning of the design process. This ensures that every detail and change makes its way into the model. Having a well defined system to start will help to compensate for the inherent errors of collecting data in the field. When applying this process to a structure that has already been built, engineering judgment takes a key role. The engineer must use all available resources to create the most accurate model possible. This includes a visual inspection of the bridge as well as a close review of the as-built drawings. A nondestructive field test is performed in order to capture the

response of the bridge. Once an accurate model is created and data are collected/post processed, MUSTANG can begin working towards estimating the “true” parameters of the structure. The user inputs the measured degrees of freedom, applied loads, unknown parameters, and selects the desired error functions into a data file. From here MUSTANG will perform parameter estimation until the convergence limits have been satisfied. The user then must review the results. Always engineering judgment will be used to accept or reject the results of parameter estimation.

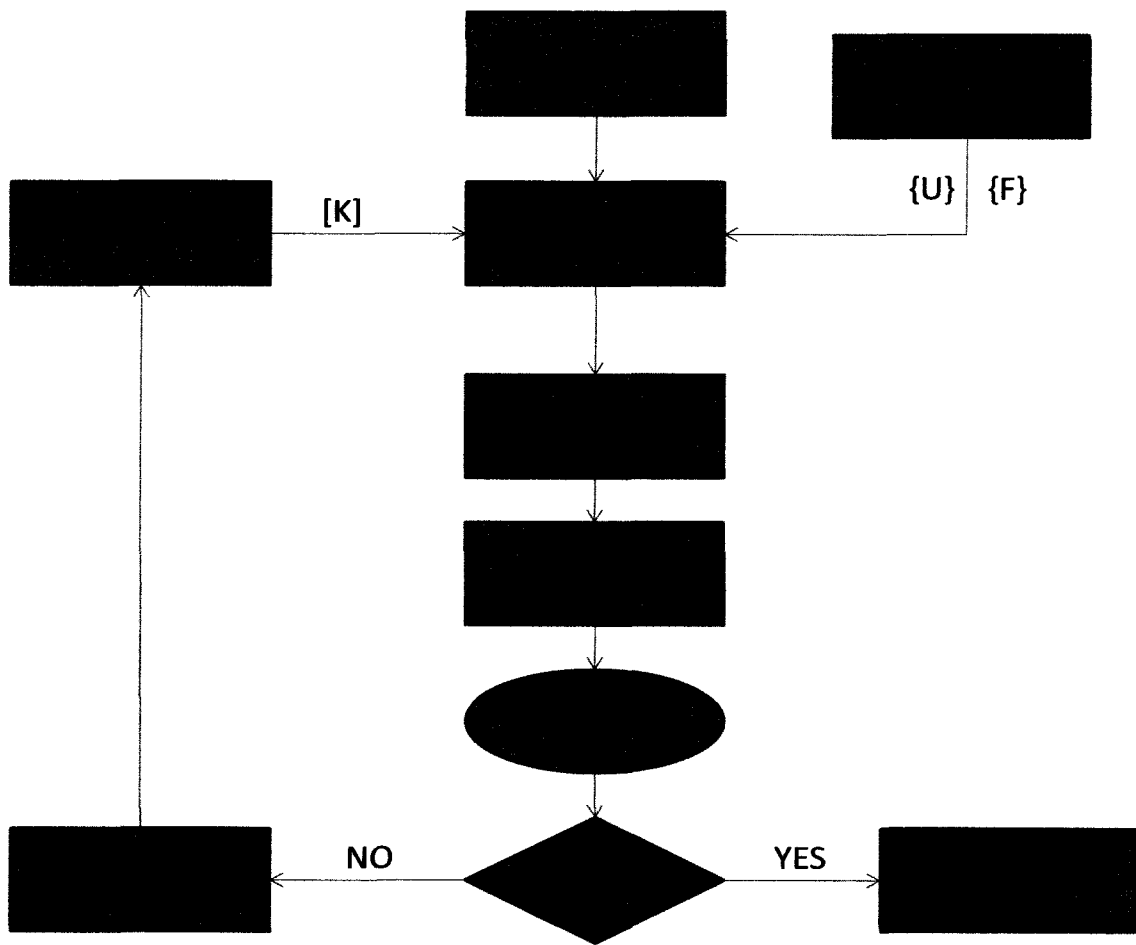


Figure 9: MUSTANG Flow Chart

2.1.1 Programmed Error Functions

There are three error functions that have been programmed in MUSTANG to perform parameter estimation. The first of these three is static stiffness. This error function utilizes nodal displacements and rotations along with the element's stiffness matrix to determine the change in parameter. An elements stiffness matrix is comprised of physical properties including area, moment of inertia, and modulus of elasticity. A structures stiffness matrix can contain millions of values. For a hand calculation, parameter estimation is only possible for very simple models. Utilizing Matlab®, MUSTANG is able to manipulate these enormous matrices in milliseconds. For linear elastic structures the force-displacement relationship is highlighted by Equation 1.

Equation 1 : Force-Displacement Relationship

$$F = [k(p)]\{u\}$$

Where force is F , $[k(p)]$ is the stiffness matrix, and $\{u\}$ is the vector of displacements. The manipulation of this equation results in the static stiffness error function, Equation 2. This error function was developed at the University of California, Sanayi and Nelson (1986).

Equation 2 : Static Stiffness Error Function

$$[E_{ss}(p)] = [K(p)]^{Analytical} [U]^{Measured} - [F]^{Measured}$$

The static stiffness error function, Equation 2, is a measurement of the difference between the modeled results and the field measurements. The stiffness matrix $[k(p)]^{Analytical}$ is based on the parameters of the model. The superscript measured, in Equation 2, signifies measured force or displacements. The static stiffness error function calculates the modeled force vector by multiplying the analytical stiffness matrix by the measured response vector. This is then compared to the measured force vector. This is show in Equation 3.

Equation 3: Static Stiffness Force Comparison

$$[E_{ss}(p)] = [F(p)]^{Analytical} - [F]^{Measured}$$

This brings up the problem of having unmeasured degrees of freedom present in a model that is to be used for parameter estimation. It is impossible to measure every degree of freedom in the field. The solution to this problem is static condensation. Known measurements are grouped together and unknown measurements are also grouped together.

Equation 4 : Static Condensation

$$\begin{Bmatrix} F_a \\ F_b \end{Bmatrix} = \begin{bmatrix} k_{aa} & k_{ab} \\ k_{ba} & k_{bb} \end{bmatrix} \begin{Bmatrix} u_a \\ u_b \end{Bmatrix}$$

The measured degrees of freedom have the subscript a, the unmeasured b. Next, using matrix algebra, the error function is manipulated to remove the unmeasured degrees of freedom from the equation. The result from the static condensation is shown in Equation 5.

Equation 5 : Static Condensation Result

$$[E_{ss}(p)] = ([k_{aa}] - [k_{ab}][k_{bb}]^{-1}[k_{ba}])\{u_a\} + [k_{ab}][k_{bb}]^{-1}\{F_b\} - \{F_a\}$$

The b still shows up in this equation in the force vector in the second term. This term zeros out because of the nature of load testing. Only measured loads are applied during a load test; therefore, the force b vector is equal to zero. Dead loads have already caused deflections prior to the measurement equipment being either installed or zeroed out. Therefore the only change in reading occurs from a new applied load.

The static flexibility error function is similar to that of static stiffness. Static flexibility also utilizes nodal displacements and rotations along with the element's stiffness matrix to determine the change in parameter. It is also based on Equation 1, but the stiffness matrix is inverted and multiplied by the force vector. This error function was developed by Sanayei and Saletnik (1996). This is show in Equation 6.

Equation 6: Static Flexibility

$$\{u\} = [k(p)]^{-1}\{F\}$$

This inverted stiffness matrix is known as the flexibility matrix. The static flexibility error function is shown in Equation 7. This equation calculates the difference between the predicted displacements and the measured displacements from the field.

Equation 7: Static Flexibility Error Function

$$[E_{ss}(p)] = [k(p)]^{-1}\{F\}^m - \{u\}^m$$

The next error function is static flexibility using strain data. This varies from the first two error functions because it does not use measured displacements and rotations. Strain measurements from strain gauges are used for parameter estimation with the static strain error function. Strain, ε , is the change in length of an element relative to its initial length. This error function made certain things in MUSTANG easier while at the same time it complicated others. The error function is Equation 8.

Equation 8: Static Strain Error Function

$$E_{sstr}(p) = [\varepsilon_{model}] - [\varepsilon_{measured}]$$

The equation for the model produced strain is the strain displacement matrix, [B], multiplied by the model produced displacements, $\{u_{model}\}$. The [B] matrix relates displacements to strains for frame elements in MUSTANG. The strain calculation is shown in Equation 9.

Equation 9: Strain Calculation

$$\varepsilon_{model} = [B]\{u_{model}\}$$

This is where some difficulty was encountered using the SAP® API. The SAP API does not directly export strain. This makes it necessary to calculate the modeled strain using displacements, rotations, and properties from the model. The B matrix is the strain compatibility matrix. This matrix is unique for different element types. It is comprised of a transformation matrix. This is a matrix that converts coordinates from a global coordinate system to a local coordinate system. The second part of the B matrix uses parameters of the element type to translate rotations and displacements into strain.

Using static strain as an error function there is no static condensation required. This means that there is no need to invert matrices. This eliminates the error that is introduced when inverting a matrix. When assembling the global B matrix, any values that are not measured are zeroed out by the matrix multiplication. This made the programming of this error function in MUSTANG easier.

For all of the different error functions, $E(p)$ is a matrix having dimensions of measured degrees of freedom by number of load cases. This is vectorized to create a vector having the dimensions of total number of measurements by 1. This prepares the error function for minimization. Figure 10 shows the change from matrix to vector.

$$\begin{bmatrix} E(p)_{1,1} & \dots & \dots & E(p)_{1,NSF} \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ E(p)_{NMDOF,1} & \dots & \dots & E(p)_{NSF,NMDOF} \end{bmatrix} \Rightarrow \begin{Bmatrix} E(p)_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ E(p)_{NM} \end{Bmatrix}$$

Figure 10: Error Function Vectorization, Bell et al. (2003)

2.1.2 Minimization of the Error Function

In order to find the difference in parameter, the objective function must first be minimized. This process is the same regardless of error function. The objective function, $J(p)$, is shown in Equation 10.

Equation 10 : Objective Function

$$J(p) = \sum_i \sum_j E(p)_{ij}^2$$

In Equation 10, $E(p)$ is the error function. P is the unknown parameter. $J(p)$ is the square of the Frobenius norm to be minimized. The error function is algebraically linearized, producing Equation 11.

Equation 11 : Linearized Error Function

$$\{E(p + \Delta p)\} \cong \{E(p)\} + [S(p)]\{\Delta p\}$$

Equation eleven does not show the higher order terms as their contribution is negligible. This equation introduces two new terms. $S(p)$ are the sensitivity coefficients with respect to each unknown parameter. These are calculated using Equation 12. They are then vectorized just as the error function. The $\{\Delta p\}$ term is the change in parameter that is the goal of the parameter estimation.

Equation 12 : Sensitivity

$$[S(p_j)] = \frac{\partial [E(p_j)]}{\partial p_j}$$

Equation 12 is substituted into Equation 10 in order to solve for $\{\Delta p\}$. This creates the scalar objective function. The result is shown in Equation 13.

Equation 13 : Scalar Objective Function

$$J(p + \Delta p) \cong (\{E(p)\} + [S(p)]\{\Delta p\})^T (\{E(p)\} + [S(p)]\{\Delta p\})$$

This equation is minimized by taking the partial derivative with respect to $\{\Delta p\}$.

This produces Equation 14, which is used to determine the change in parameter.

Equation 14 : Minimized Scalar Objective Function

$$\{\Delta p\} = -\left([S(p)]^T [S(p)]\right)^{-1} [S(p)]^T \{E(p)\}$$

This minimization technique is based on the least squares algorithm and is used to update the unknown parameters of the model.

2.1.3 Future Modal Error Functions

There are several published error functions that can use modal data to perform parameter estimation. These error functions include modal stiffness and modal flexibility. Modal error functions include the lumped mass matrix and the modal characteristics of phi and omega. The SAP® API can be used to extract the lumped mass matrix with the same logic as is used to extract the stiffness matrix. Modal analysis is valuable in parameter estimation of bridges for several reasons. The first is the determination of the bridges mode shape. Using a bridge model in SAP® it is possible to identify the different modes of vibration present within a bridge. A nondestructive vibration field test can be used to identify a bridge's mode shape in the field. This can be used as an indicator of the bridge's health when compared to a "healthy" model.

Modal error functions are beyond the scope of work for this research. However, they do present an example of how the modular nature of MUSTANG makes them a viable option for future research. Adding this subroutine makes MUSTANG more versatile and increases the tools at the user's disposal.

2.2 SAP® API

The SAP® API is the main reason for choosing SAP® as a modeling program to work with MUSTANG. The API is an open application programming interface. This means that it is capable of sharing data with and able to be controlled by other programs. These programs have to have a compatible interface. MATLAB® has a compatible interface and is an industry partner of SAP®.

MUSTANG uses the SAP® API for several different functions. The first thing MUSTANG does through the API is initiate a connection with SAP®. MUSTANG then opens the user specified model in SAP using various API functions. MUSTANG then uses the API to retrieve various information from the model. This includes coordinate data, element types, active degrees of freedom, and material types. The API is also used to run the model and retrieve the results. MUSTANG uses this information to perform parameter estimation.

The next step in the process is model updating. The user specifies which parameter to update. MUSTANG can perform parameter estimation using area, moment of inertia, and spring stiffness. The change in parameter is calculated with

Equation 14. EP is the matrix calculated from the error function. SP is known as the sensitivity matrix. This matrix takes the partial derivative of the EP matrix with respect to the unknown parameter. MUSTANG turns the stiffness matrix into a matrix of ones and zeros. It can be thought of as a participation matrix. With the change in parameter calculated, MUSTANG sends back the new parameter and changes the model. Then the process starts again and iterates until convergence.

2.3 Programming Logic

The basis for successful parameter estimation is the model. The finite element model is created in SAP® to reflect the structure in the field. SAP® is a powerful and versatile program. This makes the modeling process fairly straight forward and more accurate than simpler, less capable modeling programs. Certain finite elements are used to represent connections or components of the structure. Springs can be used to represent connections that are not fully pinned or fully rigid. They can also be used to mimic soil conditions at a support. Frame elements can be used for flexural members. Shell elements can be used to model the complex deck that spans in two directions. The goal is to get the model to react the way the structure would under the test loads. If an instrument from the field test measures displacement or rotation at a given location, a node must be placed in the model at the same location in order to extract rotations and/or displacements for comparison.

MUSTANG is programmed to read in post-processed field data from text files. This data can be in the form of displacements, rotations, and strains. This field data are the basis for parameter estimation on structures using nondestructive testing. The user data file tells MUSTANG where these measurements are located within the model. MUSTANG then runs the structural model with the API and extracts the modeled results from SAP®. MUSTANG then uses the field data and model data to perform parameter estimation and model updating. One of the benefits to using MUSTANG is that it creates, and saves, a usable model after the parameter estimation is complete.

The parameter estimation and model updating process is iterative. It can run for a certain number of cycles, indicating a poorly defined system, or until a convergence limit is reached. A poorly defined system will continue to run without ever reaching convergence. The user specifies both of these. The convergence limit takes the new change in parameter and compares it to the old change in parameter. If the percent change is less than the user specified limit, the parameter estimation is considered to have converged upon the “true” parameter of the structure. For well defined models, such as frames, convergence can occur with as little as two iterations. For a more complex structure such as a bridge, the process may continue for several hundred cycles before convergence is reached. The iterations happen quickly because SAP® is opened in the operating systems background by MUSTANG without seeing the program interface. After the parameter estimation is complete a model named “___FINAL” is saved and can be opened and used by the user at a later date.

2.4 Key Modular Subroutines

Data File

- Mustang
 - SimDisplacements : Uses simulated values from data file to create “true” measurement set
 - SetModifiers : Sets model element modifiers based on “true” parameters
 - txereader: Retrieves degrees of freedom
 - DisplacementsVector : Retrieves displacements from model
 - ForceVector : Creates vector of user defined forces
 - StiffnessPartition : static condensation for stiffness reader based on measured degrees of freedom
 - txereader
 - ForceSAP: Retrieves nodal forces from model
 - ssSensitivityFrame : Creates sensitivity matrix based on unknown parameters using least squared optimization
 - SetModifiers : Sets model element modifiers based on minimized parameters
 - StiffnessPartition

- Bmatrix3D : creates strain displacement matrix based on user specified measured strains
- txkreader : retrieves stiffness matrix from model
- StaticStiffness : Static Stiffness Error Function
 - txereader
 - txkreader
 - StiffnessPartition
- StaticStrain : Static Strain Error Function
 - txkreader
- StaticFlexibility : Static Flexibility Error Function
- SetModifiers : sets modifiers based on calculated ΔP for unknown parameters

2.5 Static Stacking

Static stacking refers to using more than one type of static error function to perform parameter estimation. The $\{E(p)\}$ matrices from the different user specified error functions are stacked one on top of the other. The same is true for the $\{S(p)\}$ matrices. This allows for different types of measurements to be used with the same model. This creates a more well defined system for the parameter estimation. A more well defined system reduces the time it takes for parameter estimation and increases the quality of the results.

Equation 15: Static Stacking

$$\{E_T\} = \begin{Bmatrix} E_{P1} \\ E_{P2} \\ E_{Pn} \end{Bmatrix}; \{S_T\} = \begin{Bmatrix} S_{P1} \\ S_{P2} \\ S_{Pn} \end{Bmatrix}$$

The process of stacking will allow various SHM and ITS instrumentation to contribute to parameter estimation and model updating in MUSTANG. Digital imaging can calculate displacements used for static stiffness and flexibility error functions. Internal strain gauges will provide strain data for the static strain error function. All of these technologies will help to capture the response of the structure and effectively estimate its parameters as it stands in the field.

Upon completion of modal error function programming, GPS and digital imagery can be used to capture the mode shapes of the bridge. These measurements can then be used in conjunction with static measurements to even further define the system.

Chapter 3

Program Verification Models

In order to program MUSTANG it was necessary to become familiarized with the SAP® API and its various functions. The best way to do this was to take a simple model and begin programming the necessary functions for parameter estimation. This helped in several ways. It required an in depth understanding of the API. There are many API functions that go into MUSTANG and they all need different inputs from the user data file or from data that MUSTANG has retrieved from the model. Using this approach helped to verify the method of programming used as the program was being developed. This prevented MUSTANG from becoming a black box. Function inputs and

returned values were verified as the program was taking shape. The data being extracted from the SAP® models were compared to hand calculated values for validity. Not for purposes of validating SAP®, but rather for making sure MUSTANG was seeing what it thinks it as seeing for data.

3.1 Cantilever Beam

The first verification model used was a cantilever beam. It is a simple example with well defined properties and can be calculated by hand for verification purposes. This was used to determine how to manipulate the SAP® model using the API for the purpose of parameter estimation.

3.1.1 One Element Unknown

The first, and simplest, model was a one element cantilever. This cantilever was comprised of a single 2" by 6" pine section. This two by six was five feet long. The modulus of elasticity used for this pine section was 1600 ksi. The model was configured to ignore the effects of shear. This was done by setting the shear modifiers to zero when creating the two by six in the section designer. At the end of the cantilever a ten pound load was applied in the downward direction. The unknown parameter was the moment of inertia. MUSTANG uses simulated data to run the parameter estimation. The moment of inertia was reduced by twenty five percent. This takes the moment of inertia from 36 in^4 to 27 in^4 . MUSTANG then runs the model and

extrudes the displacement and rotation at node two. These values are the simulated data and are treated as field measurements. The original model was then reopened. MUSTANG then used the static stiffness error function to perform the parameter estimation.

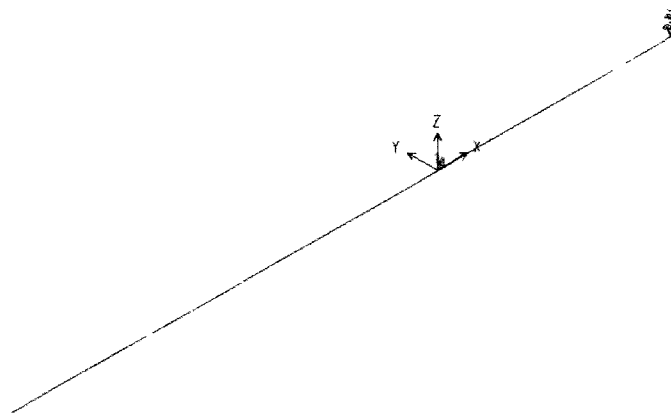


Figure 11: One Element Pine Cantilever Model

The parameter estimation process required only one iteration to reach convergence. This was to be expected with such a simple model. This also matches the hand calculated parameter estimation done with the assistance of the MathCAD software. The EP, SP, and stiffness matrices matched with 0% difference between MUSTANG and the hand calculation. The calculated change in parameter was also a match to the hand calculation. Both methods calculated the “true” moment of inertia

with 0% error. This illustrates that MUSTANG is obtaining the proper information and properly performing the matrix manipulation and algebra required.

Table 1: One Element Pine Cantilever Results

	Hand Calculation	MUSTANG
Number of Iterations	1	1
% Match to Simulated Values	100%	100%

3.1.2 Two Element One Unknown

The next case examined was that of a two element cantilever with one unknown parameter. Both elements consist of the same material properties. They are both the same pine two by six from the one element model. The same twenty five percent reduction in moment of inertia was applied to the second element of the cantilever. The first element has no reduction. MUSTANG runs this and extracts the simulated data. The data file specifies that the unknown parameter was the moment of inertia of the frame object labeled '2'.

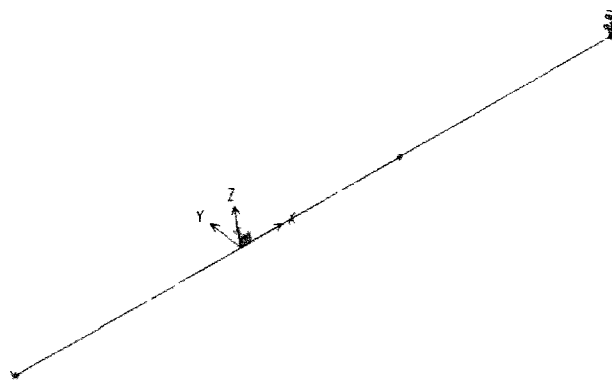


Figure 12: Two Element One Unknown Pine Cantilever

A ten pound load in the negative z direction was applied at the end of element two at node three. The static stiffness error function was used to perform the parameter estimation. The displacements and rotations are measured at nodes two and three. MUSTANG then runs the parameter estimation on this model using the user created data file. This model also converges after one iteration. This shows that the program is capable of handling more than one element.

3.1.3 Two Element Two Unknowns

In real structures there exists the possibility of more than one element having an unknown parameter. This model consists of the same two element cantilever as the previous model. In this model both elements have a reduced moment of inertia. Element one has a fifty percent reduction in moment of inertia. Element two has the

same twenty five percent reduction as it has in the previous models. A ten pound load was applied at node three in the negative z direction.



Figure 13: Two Element Cantilever Two Parameter Unknown Deflected Shape

For this case MUSTANG performed the parameter estimation and model updating using the static stiffness error function. The displacements and rotations were measured at nodes two and three. The active degrees of freedom for this model are displacement in the z direction and rotation about the y axis. MUSTANG runs the parameter estimation on the model using the user created data file. One iteration was necessary to reach convergence. This shows that the program is capable of handling more than one element with more than one unknown.

3.1.4 Two Element Cantilever with an Internal Spring Hinge

The next type of element programmed in MUSTANG was the spring element. The first case examined was that of a two element cantilever with a spring connecting the two elements. The spring was located at node two. The spring in the model was initially set to a rotational stiffness of 10 kips. The simulated data uses a spring with a rotational stiffness of 0.5 kips. The same ten pound load was applied at the end of the two elements, node 3. The same ten pound load was applied at the end of the two elements, node 3.

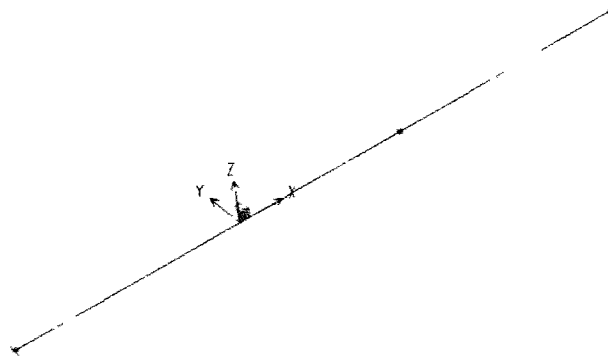


Figure 14: Two Element Cantilever with Internal Unknown Spring

Displacements and rotations are measured at nodes two and three. MUSTANG uses static stiffness to converge in one iteration. The results from MUSTANG matched the hand calculations.

3.1.5 Two Element Cantilever with a Spring Support

The next case had a more practical application to the bridge modeling that MUSTANG will be investigating. A two element cantilever was fixed with a spring support. The spring support was located at node one. All displacements are fixed by setting their spring stiffness, k , to a value that resembles a fixed support. This model use $1e9$ kip-ft. These spring elements are used in a bridge model to represent bearing pads and their relative stiffness. Again a ten pound load was placed at node three. Displacements and rotations are measure at nodes two and three.

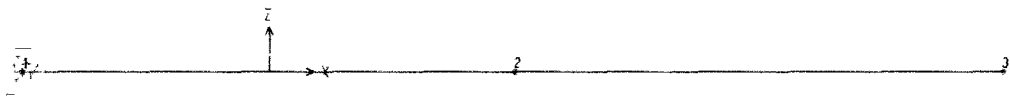


Figure 15: Two Element Cantilever with Spring Hinge Support

The simulated data were run with the rotational stiffness about the global y axis set to fifty kips. The data were collected and treated as field measurements. The model had an initial rotational stiffness of 100 kip-ft. MUSTANG uses the static stiffness function to perform parameter estimation. This simple model takes only one iteration to converge. This matches the hand calculation as well.

3.1.6 Two Element Cantilever with a Pin Connection and End Spring Support

The last of the developmental models was the same two element cantilever, but this model consists of a pinned connection at node one and a vertical spring at node three. This model contains a component that would prove useful in modeling a bridge. The vertical spring can be used to represent an elastomeric bearing pad. This proved useful when modeling the Rollins road bridge. A one kip load was placed on node 3 in the negative z direction. A vertical spring was also located at node three with a ten ft-kip stiffness in the z direction.

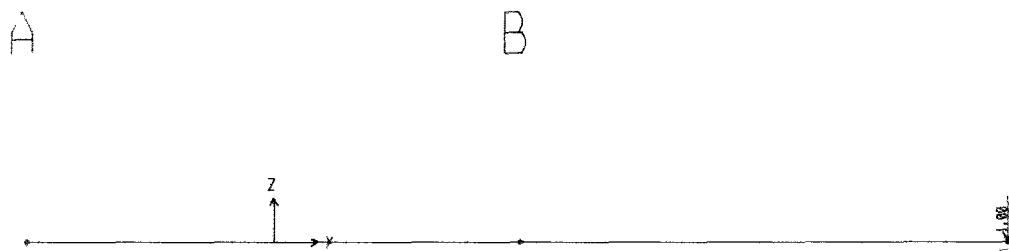


Figure 16: Two Element Cantilever with End Vertical Spring Support

MUSTANG was called by the data file to run parameter estimation using the static stiffness function. Again, only one iteration was required for convergence. This shows that MUSTANG is capable of handling spring elements that resists displacement as well as rotation.

All of the developmental models required an in depth understanding of the SAP® API and its functions. The different functions required to program the error functions required to run the different models all have unique requirements that must be met by a combination of programming and user input into the data file. The development of these models has led to the development of a useful and powerful data file. By adding requirements as model complexity increased, it was shown that the nature of MUSTANG is that of an easily updatable and modifiable parameter estimating and model updating program.

3.2 Published Parameter Estimation Models

The two following verification models are published examples of parameter estimation involving several different element types and boundary conditions. The models include truss, frame, and spring elements with different unknown parameters. The truss elements have area as the unknown parameter. The frame elements use an unknown parameter of moment of inertia. The spring elements have unknown rotational and translational stiffness's. The boundary conditions include pinned connections as well as spring connections. The spring connections are used to model the soil supporting the structure. The two dimensional truss comes from the journal article Damage Assessment of Structures Using Static Test Data, Sanayei et al (1991).

3.2.1 Two Dimensional Truss

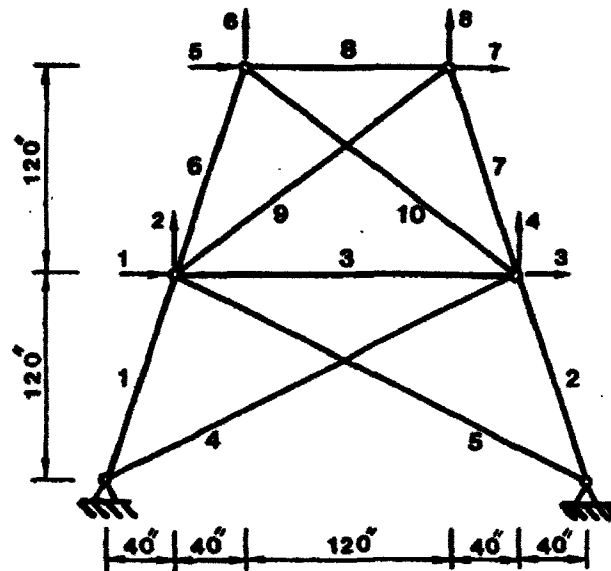


Figure 17: Two Dimensional Truss, DOF, Sanayei et al. (1991)

The first model is a two dimensional, ten member truss. The model comes from the paper Damage Assessment of Structures Using Static Test Data, Sanayei et al. (1991). The truss is two stories. All member connections are pinned. This connection type does not transfer moment. The model is comprised of two active degrees of freedom, U_x and U_y . There are ten different load combinations that are tested. There are different degrees of freedom measured with the different loading combinations, see table 2. A 100 kip load is placed in either the vertical or horizontal direction. Some combinations specify the load to be placed on different nodes. Some of the load combinations involve more than one load case. These situations require stacking, previously discussed, in order to perform parameter estimation on the model's unknown parameters. Table 2 shows the different load combinations with the

different load cases, measured degrees of freedom, and different unknown parameters.

Table 2 : Two Dimensional Truss Parameter Estimation Data, Sanayei et al. (1991)

Case	FDOF	DDOF	NUP	PU	NIM
1	1-8	1-8	10	1-10	36
2	5-8	5-8	10	1-10	10
3	5-8	1,2,5-8	10	1-10	18
4	1-4	1-8	10	1-10	26
5	5,8	1-8	10	1-10	15
6	1-4	1-4	5	1-5	10
7	7	1-8	8	1,2,4,6-10	8
8	7,8	1-6	7	1-6,10	12
9	5-8	1-4	5	1-5	16
10	5,6	1-4	5	1-5	8

The FDOF column lists the degrees of freedom at which the 100 kip load is applied. The DDOF column lists the measured degrees of freedom. The NUP column lists the number of unknown parameters for the given case. The PU column lists the member with the unknown parameter. The only unknown parameter with the truss elements is area. All of the elements begin with an area of 5 in² and a modulus of elasticity equal to 30,000 ksi. The NIM column lists the number of independent measurements. The true area is 3in².

The first column of the table below lists the case number. The next two columns are the number of iterations required for MUSTANG to reach convergence and Sanayei et al. (1991). Both methods reached convergence for all of the load cases.

Table 3: Comparison of MUSTANG Results to Published Results from Sanayei et al. (1991)

Case	MUSTANG Iterations	Iterations (Sanayei)
1	2	2
2	2	2
3	2	2
4	2	5
5	2	2
6	2	2
7	2	2
8	2	2
9	2	2
10	2	2

Both methods show convergence given more than one load case. All cases with the exception of case four reached convergence with two iterations. The change in parameter is reached with only one run. The second run is only required to show that no further changes are required to the parameters. Case four has only one applied load. This provides less data and makes the system less robust. The method of parameter estimation used by Sanayei et al. 1991 required five iterations to achieve convergence. MUSTANG required only two iterations.

This model shows the parameter estimation capabilities of MUSTANG with a two dimensional truss. It illustrates the capacity of MUSTANG to handle multiple unknowns, measured degrees of freedom, and load cases. It also shows the capability of MUSTANG compared with a published example of parameter estimation.

3.2.2 Two Dimensional Bridge Frame

The next model was a two dimensional frame representing a bridge. The model comes from the paper Parameter Estimation Incorporating Modal Data and Boundary Conditions, Sanayei et al. (1999). This model contains frame elements and spring elements. The frame elements consist of W36 X 135 steel girders. The legs consist of W14 X 145 steel columns. The girders are pinned at the beginning and the end of the span. The columns are supported by springs with vertical, horizontal, and rotational stiffness. The degrees of freedom are shown in the figure below.

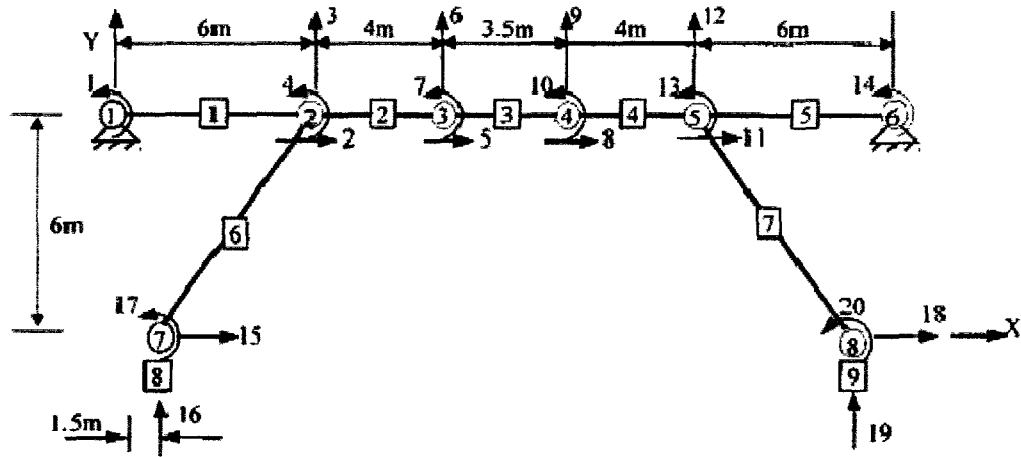


Figure 18 : Two Dimensional Bridge Frame, Sanayei et al. (1999)

Sanayei et al. (1999) uses modal error functions to perform parameter estimation on the bridge model. Even though modal error functions are not yet included in MUSTANG, this model still has many useful purposes as a verification model for MUSTANG. The bridge model can be used effectively to examine different combinations of unknown parameters and element types. This model proved useful in programming the necessary functions for MUSTANG to handle multiple unknown parameters on the same object. The different parameter estimation cases included the area and moment of inertia as an unknown parameter on one object. There are also multiple spring stiffness's unknown on the same spring object. See Table 4 for initial parameters and their associated "true" values.

Table 4 : Initial Structural Parameters and Damage Scenarios, Sanayei et al. (1999)

Bridge component (1)	Structural parameter (2)	Parameter units (3)	Best Guess	Damage Scenario 1		Damage Scenario 2	
			Initial values (4)	True values (5)	Initial/True (6)	True values (7)	Initial/True (8)
Girders (beam) W36 × 135	I	m^4	32.47×10^4	16.24×10^4	2.00	64.94×10^4	0.50
	A	m^2	2.76×10^2	1.28×10^2	2.00	5.12×10^2	0.50
	ρ	$kg\ m^3$	7850	Known	1.00	Known	1.00
Legs (partially restrained frame) W14 × 145	I	m^4	7.12×10^4	3.56×10^4	2.00	14.24×10^4	1.25
	A	m^2	2.75×10^2	1.38×10^2	2.00	5.50×10^2	1.25
	K_{ϕ}	$N\ m\ rad$	1.00×10^8	5.00×10^7	2.00	1.33×10^7	0.75
	ρ	$kg\ m^3$	7850	Known	1.00	Known	1.00
Foundation (soil-substructure superelement)	K_{RH}	$N\ m$	6.00×10^8	4.50×10^8	1.33	4.50×10^8	1.33
	K_{RR}	$N\ m$	0	1.00×10^8	0	1.00×10^8	0
	K_{RB}	$N\ rad$	3.20×10^8	2.40×10^8	1.33	2.40×10^8	1.33
	K_{RT}	$N\ m$	7.00×10^8	5.25×10^8	1.33	5.25×10^8	1.33
	$K_{T\phi}$	$N\ rad$	0	1.00×10^8	0	1.00×10^8	0
	$K_{\phi\phi}$	$N\ m\ rad$	1.50×10^9	1.13×10^8	1.33	1.13×10^8	1.33
	M_{RT}	kg	21420	28560	0.75	42840	0.50
	M_{RH}	kg	21420	28560	0.75	42840	0.50
	M_{ϕ}	$kg\ m^2$	17850	23800	0.75	35700	0.50

The first model tested has all of the parameters unknown. The mass matrix is not included as it only applies to dynamic error functions. A vertical load of -1,000kN is applied at nodes two, three, four, and five. Each applied load is given a separate load case: LC1, LC2, LC3, and LC4. See the table below for load case locations.

Table 5: SAP Model Load Case Locations

Load Case	Force Location
LC1	2
LC2	3
LC3	4
LC4	5

The rotation is measured at nodes one and six. The vertical displacement, horizontal displacement, and the in plane rotation is measured at all of the remaining nodes. In the figure below, a triangle represents a pinned connection and the colored

zigzag lines represent the translational springs. Rotational springs are represented by a straight line at the node. These are difficult to see with the translational springs.

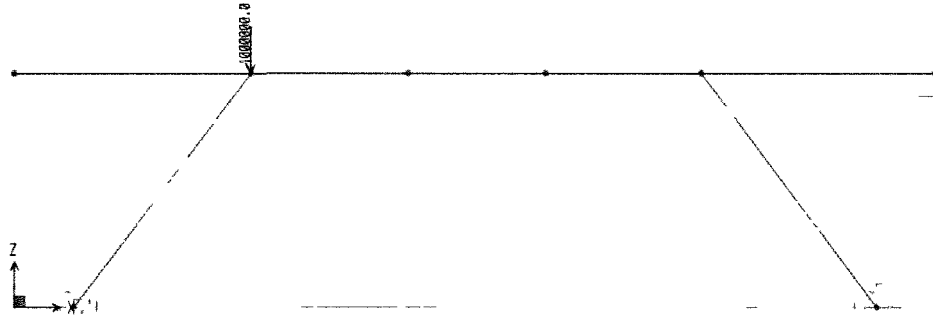


Figure 19: SAP Two Dimensional Bridge Frame Model, LC1

The simulated displacements are based on the frame moment of inertia and area being reduced to fifty percent of their capacity. The vertical, horizontal, and rotational stiffness's are increased by one third. This corresponds to damage scenario 1 in Table 4. The static stiffness error function is used to perform parameter estimation. Six iterations were required by MUSTANG to reach convergence within one percent error for each of the parameters. See **Error! Reference source not found.** on the next page for an iteration by iteration change in parameter. The percent difference column refers to the difference between the adjusted parameter of that iteration and the true value of the parameter, shown in the parameter column. Given a broader allowance for error, the program would require fewer iterations to reach convergence. This model shows the capability of MUSTANG to handle several element types each

with their own unknown parameters. It also displays MUSTANG's capability to support multiple unknown parameter types within one parameter estimation.

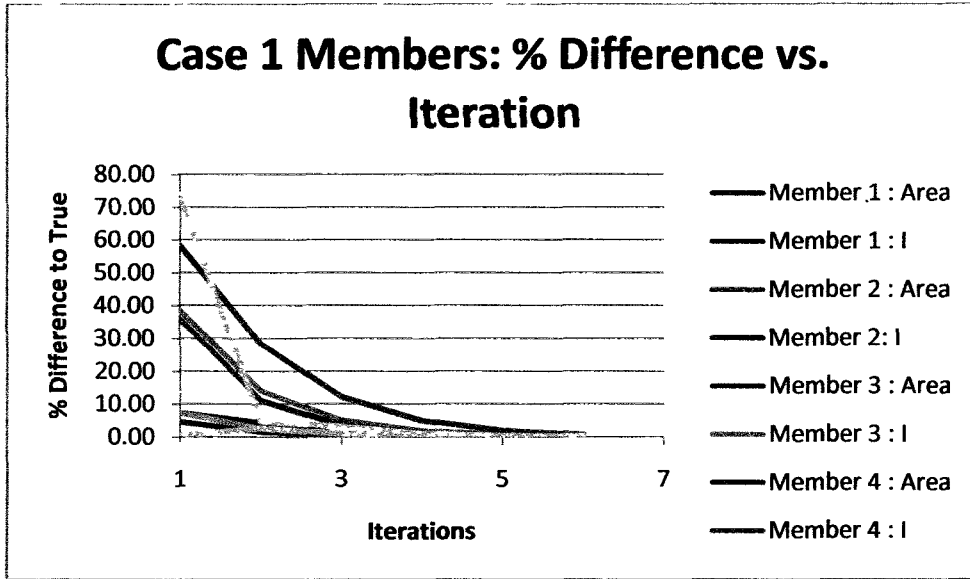


Figure 20: Two Dimensional Bridge Parameter Estimation Convergence, Members Case 1

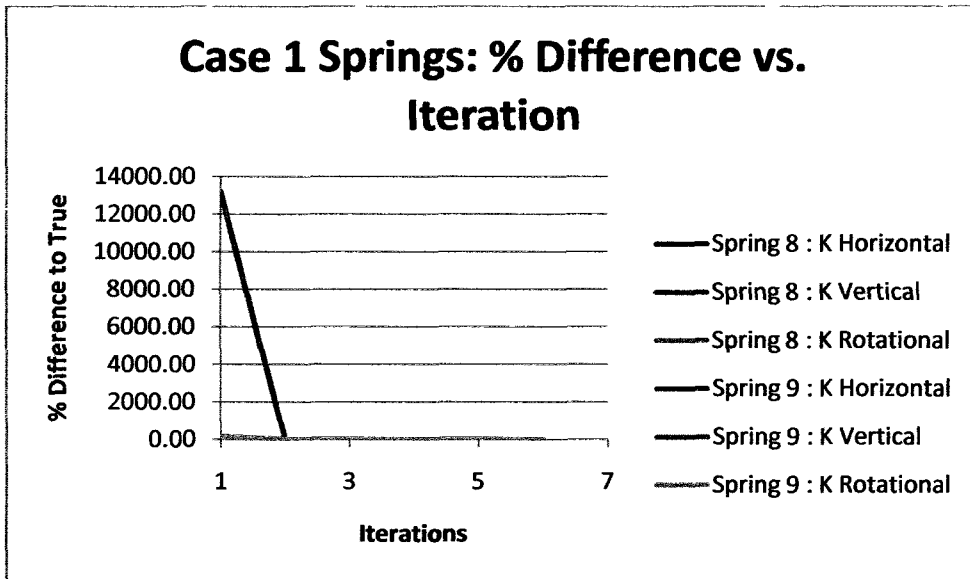


Figure 21: Two Dimensional Bridge Parameter Estimation Convergence, Springs Case 1

The second case for MUSTANG has only one load case. LC1 has a -1,000 KN load in the vertical direction placed at node two. The unknown parameters are translational and rotational stiffness's of the two spring supports. Having only one load case reduces the redundancy of the system. It makes for a lesser conditioned system from which to perform parameter estimation. This will test MUSTANG's capabilities with less information supplied. All degrees of freedom are also measured for LC1.

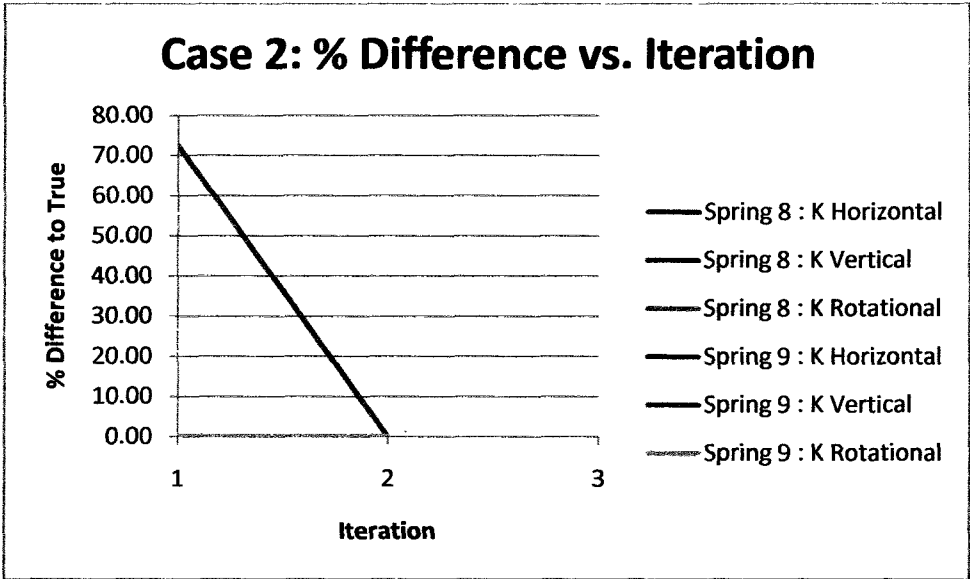


Figure 22: Two Dimensional Bridge Parameter Estimation Convergence, Case 2

This scenario resulted convergence to the true parameters. It required only two iterations to acquire the exact parameters. In the field the “exact” solution does not exist or at least is unknown to the engineer. If MUSTANG were to perform another iteration on this model, or a field model approaching its “true” parameters, it would yield a very low change in parameter. This is the stopping point for the parameter

estimation process. The number of unknowns was drastically reduced from case 1. At the same time the number of measurements was also drastically reduced. This is a useful model to show the efficiency of MUSTANG. It makes sense that as few as two iterations were necessary to reach convergence with this model.

The third case with the two dimensional bridge model focuses on the deck alone. The measured degrees of freedom are one through fourteen. The unknown parameters are the area and moment of inertia of members one through five. This would be an example of a field test given optical deflection along a span. The load cases used are LC1 and LC2. This clusters the data with forces located at nodes two and three. This is not a very well spread data set; therefore, it is not a very well defined system. The movement of the supports is unknown. This makes parameter estimation more difficult. MUSTANG requires seven iterations to estimate the parameters to within one percent of their true values. **Figure 23** shows the results of case 3.

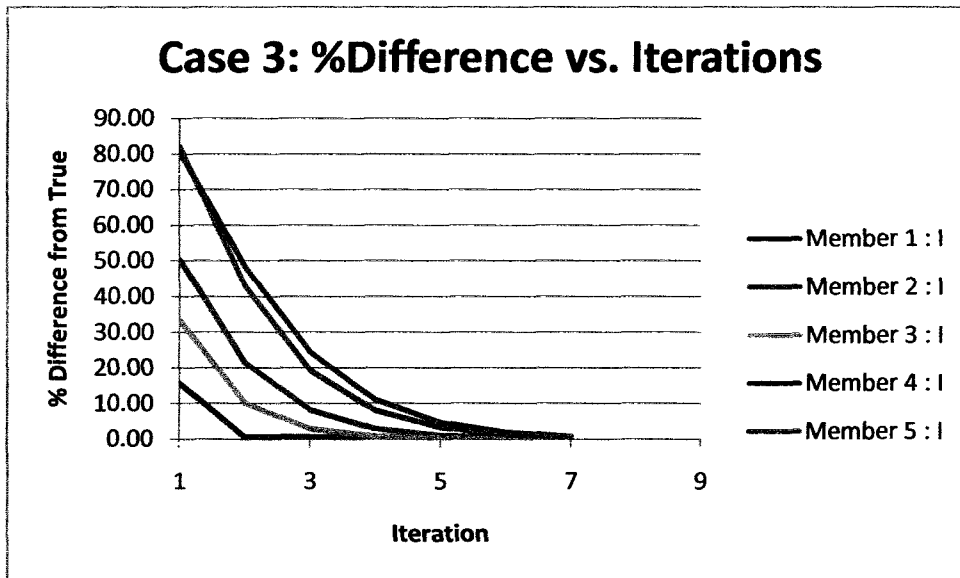


Figure 23: Two Dimensional Bridge Parameter Estimation Convergence, Case 3

The results show that the unknown area converges quicker than the moment of inertia. It takes one iteration to estimate the true value within one tenth of a percent. The moment of inertia takes the next six iterations to converge. The members located closest to the load converge upon their “true” values the quickest. This is logical. There will be greater deflections and rotations closest to the point of loading. This amplifies the effects of the reduced bending capacities and highlights those areas of damage quicker than the members that see less effect from the load cases.

Case 4 could represent an example of a field test focusing on the support of a bridge structure. The unknown parameters are the area and moment of inertia on members six and seven, the support legs. The spring stiffness’s are the other unknown parameters. All degrees of freedom are measured for LC2 and LC3. This presents a

symmetric loading and damage scenario for all unknown parameters. The parameter estimation results in a matching convergence for similar objects. After three iterations all parameters have converged to within one percent of the true values.

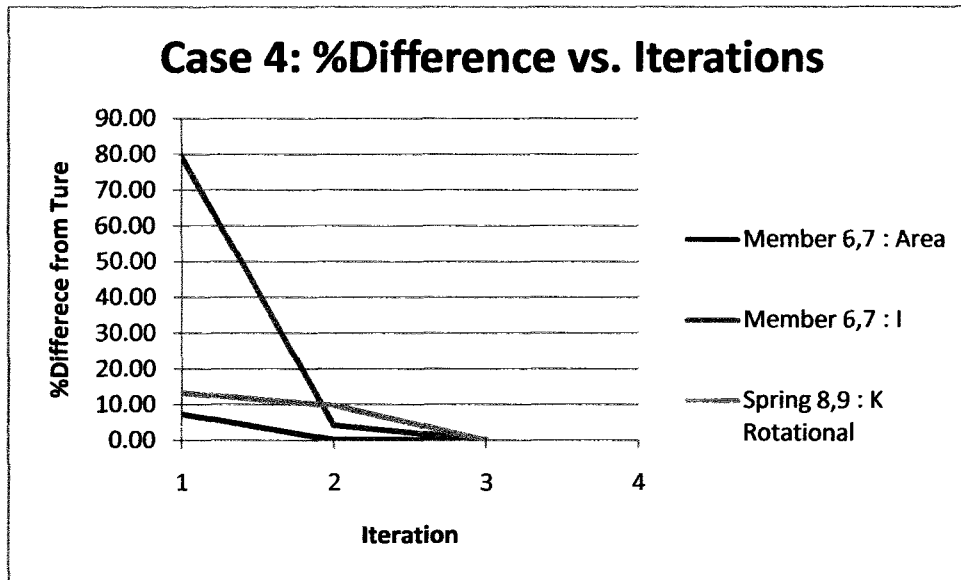


Figure 24: Two Dimensional Bridge Parameter Estimation Convergence, Case 4

The results from the four parameter estimation cases are presented in **Table 6**. The case number is in the first column. The Force Location column contains the node number at which the vertical load is placed. The measure degrees of freedom correspond to the illustrated degrees of freedom in Figure 18. In the unknown parameter column Area is abbreviated A and Moment of Inertia is abbreviated I. The iterations listed are the number of iterations required to converge upon the true parameter to within one percent of its value.

Table 6 : Two Dimensional Bridge Parameter Estimation Summary

MUSTANG Case	Force Location	Measured DOF	Unknown Parameters	Iterations
1	2,3,4,5	all	all	6
2	2	all	spring stiffness's	2
3	2,3	1 - 14	member 1 - 5: A, I	7
4	3,4	all	member 6,7: A, I ; Spring Stiffnesses	3

These two verification models were important in two significant ways. First, the two models helped to optimize the programming of MUSTANG. Several issues arose with both models. When programming the two dimensional truss, problems with the programming loops controlling the stacking of multiple load cases were exposed. Stacking is a key component of MUSTANG. This allows multiple load cases from field tests to be used together for parameter estimation. This creates a better defined system. After this issue was corrected, MUSTANG was able to run all of the analysis cases from the published example Damage Assessment of Structures Using Static Test Data, Sanayei et al. 1991. The two dimensional bridge revealed an error in the way modifiers were stored. If a single object had more than one parameter as an unknown, when the second parameter was being adjusted the first would be overwritten. This was true for springs and frame elements. Once the method of storing adjusted parameters as fixed in MUSTANG, the bridge model was run successfully. The second important aspect of these models was being able to compare MUSTANG to a benchmark. This benchmark was the published results of parameter estimation from

Damage Assessment of Structures Using Static Test Data, Sanayei et al. (1991) and Parameter Estimation Incorporating Modal Data and Boundary Conditions, Sanayei et al. 1999. The two dimensional truss from Sanayei et al. (1991) was modeled as an exact match for parameter estimation using MUSTANG. The published results were directly compared to the results obtained by MUSTANG. The two dimensional bridge from Sanayei et al. (1999) was used as a basis for a useful model. Because the error functions utilized have yet to be programmed into MUSTANG, the methods of parameter estimation were not the same. But the model and simulated damage were both used to create a replicable parameter estimation scenario using static error functions. Both models pushed MUSTANG to develop into a more robust and accurate program.

3.3 Future Work and Conclusions

These results have been attained with other published model updating programs. PARIS® was used to obtain the results for Sanayei et al. (1991). The lacking element of PARIS® is the advanced graphical user input for modeling presented by SAP2000®. Large and complex bridge models are either not possible or very tedious and time consuming using methods that require coordinates and connectivity to be entered manually. This can also lead to error.

Further work is needed to develop MUSTANG into a more robust program. There were no studies conducted of measurement error or modeling error associated with the parameter estimation completed in MUSTANG. The ability to overcome error will be necessary for MUSTANG to be a reliable solution for parameter estimation needs.

CHAPTER 4

Field Example Bent 57

4.1 Background

The first field example to be run by MUSTANG was that of Bent 57. During the construction phase of the Big Dig it was critical to continue traffic flow with minimal interruption through the city of Boston. Route 93 had to continue to use the overpass until the tunnels were opened to traffic. Bent 57 was a moment frame used to support the viaduct during the excavation process and throughout the construction of the Big

Dig, Harrington (1998). This was one of the many bents that supported the underpinning of the overpass.

In the field moment connections can be achieved with either a bolted gusset plate between two members or with a welded plate. Bent 57 has a bolted moment connection. The assumption is that these connections are rigid. A field test was performed to supply data for a parameter estimation to check the validity of this assumption and determine the actual rotational stiffness of these moment connections.

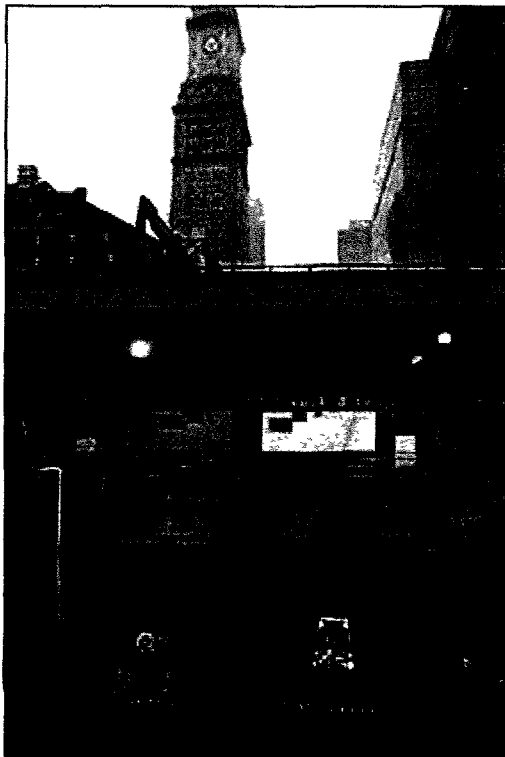


Figure 25 : Boston Central Artery / Big Dig Construction (Photo Courtesy of PBS)

Prior to the 2004 demolition of the Central Artery, a nondestructive load test was performed on Bent 57. The purpose of the load test was to acquire enough data to

perform a parameter estimation on the moment connections between the legs and the cross beam. A crane was used to apply two different load cases to the frame. The first load case consisted of close to a fifty kip vertical load being applied to a pick point at the center of the cross beam. The second load case involves a pulley used to apply a horizontal force of close to twenty kips approximately ten feet above the support on one of the legs. Strains were measured at various locations throughout the moment frame for both load cases. There were four strain gauges on the cross beam connecting the two legs. There were eight strain gauges on the leg that the load was applied to. There were also three tilt meters attached to the frame.



Figure 26 : Bent 57 Load Test Setup

All of the data gathered from the strain gauges for load case one received a rating of one. This rating signified a good data range. All of the readings from the tilt

meters received a rating of two. This rating signifies that the data can be acceptable but that there is a suspect amount of noise present. The percent errors between the measured tilts and the simulated values were too large to consider them as an accurate measurement for parameter estimation. Therefore, the tilt measurements were not included in MUSTANG's data file.

For load case two, strain gauges 1-4 and 11-12 received a rating of one. These gauges have a good data range and are acceptable for use in parameter estimation. Gauges 5-10 have a rating of two. These gauges recorded strains that were more than two hundred percent different from the simulated values obtained using modeled values in SAP. These gauge locations and readings are not included in the parameter estimation.

4.2 Model

Bent 57 was modeled in SAP2000® using a combination of frame elements and spring elements. The frame elements are used to create the structure. The spring elements model the moment connections between the legs and the cross beam. The frame objects are rolled steel sections. The legs are W14 X 145 sections made of A992 GR50 steel. The legs of Bent 57 have the strong axis oriented in the y-axis. The default orientation for the frame's legs has the strong axis oriented along the x-axis. Both legs are rotated 90 degrees within the SAP2000® model in order to accurately represent the Bent 57. The cross beam is a W36 X 300 sections made of A992 GR50 steel.

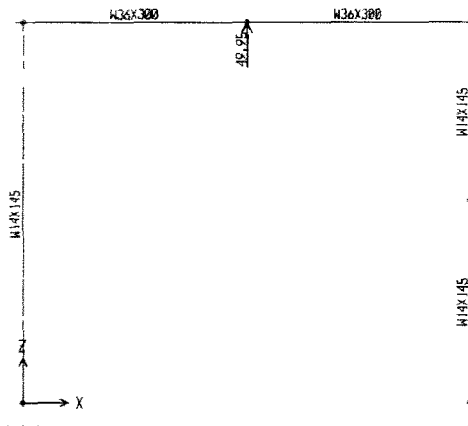


Figure 27 : Bent 57 SAP2000® Model

There are two different types of boundary conditions in the Bent 57 model. The bases of the legs are modeled as fixed to the ground. This results in no rotation or translation occurring at the base of the model. The legs are connected to the cross beam by rotational springs. These springs resist only in plane rotation. The rotational stiffness is used to model the moment connections in the field. The stiffness's are representative of a moment connection's resistance to rotation.

4.3 Simulated Data

The first parameter estimation run by MUSTANG with Bent 57 uses simulated data. This is MUSTANG's first test with the static strain error function. This function uses the strain gauge locations and the frame element type to create a three dimensional strain displacement matrix. This matrix converts measured displacements and rotations into strain measurements at the specified locations.

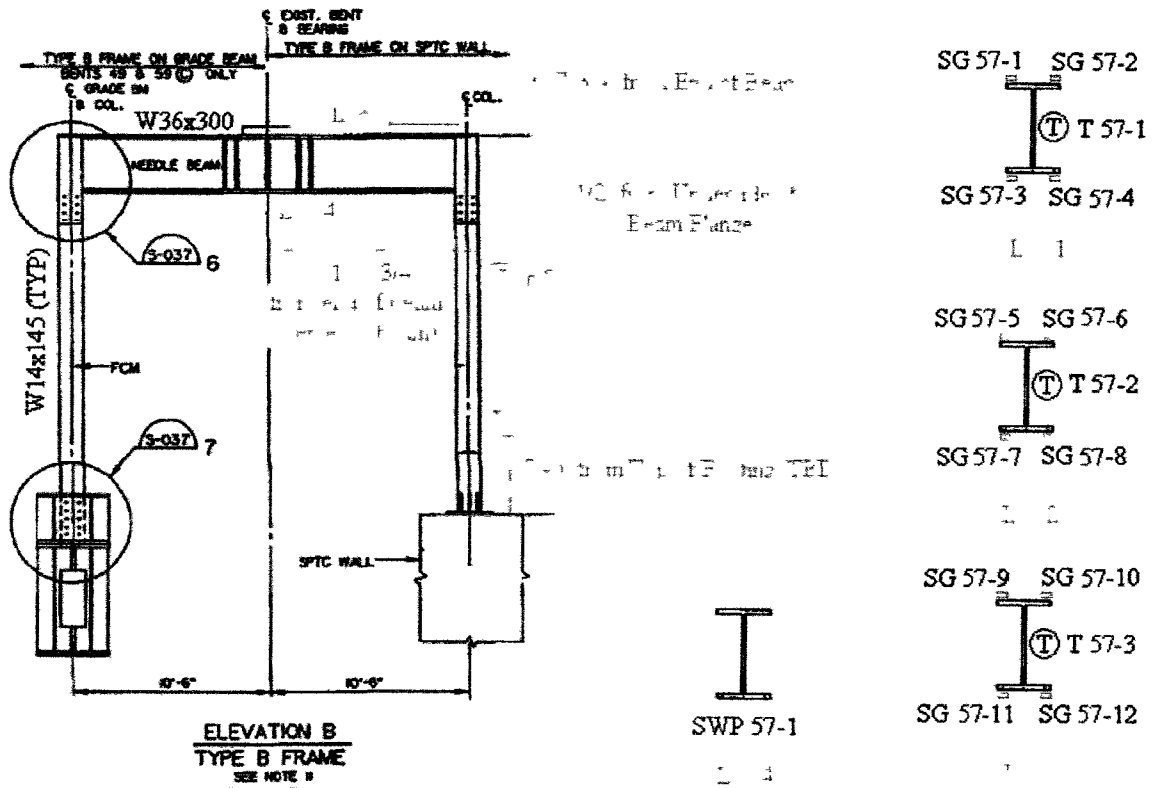


Figure 28: Bent 57 Gauge Locations, Santini-Bell et al (2008)

For load case one all of the gauge locations are included. Even though this is a simulated run, it is intended to represent the field test. All of the gauge locations for the first load case had acceptable amounts of error for parameter estimation. Load case two showed significant error at locations two and three. One pair of gauges at location three recorded acceptable results, these gauges were eleven and twelve. These gauges as well as the gauges from location one were included in load case two.

The rotational stiffness's were set to 9.58×10^7 inch-kips to create the simulated data. The static strain stiffness function was used for both load cases. The simulated strains from each load case were stacked to create a better defined system

for parameter estimation. The two unknown parameters were the rotational stiffness's at joints three and four.

Table 7: MUSTANG Iterations for Bent 57, Simulated Data

Case	Member	Parameter		Iteration 1		Iteration 2	
		True Value	Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True
1	2	9.580E+07	5.000E+07	9.23E+07	3.66	9.53E+07	0.52
	4	9.580E+07	5.000E+07	9.23E+07	3.65	9.53E+07	0.52

MUSTANG required only two iterations to converge upon the “true” rotational stiffness of both springs to within one percent error. Given the symmetry of the model, it makes sense that both springs converge at a similar pace given the low error simulated data. The rapid convergence also indicates that the system is well defined and conducive to parameter estimation.

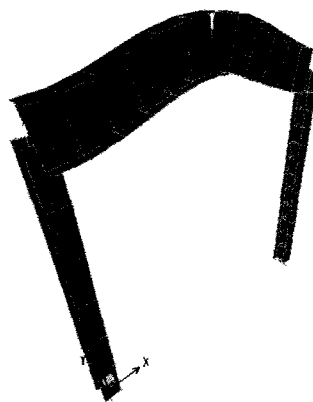


Figure 29: Deflected Shape for Bent 57, Load Case 1

4.4 Field Data

4.4.1 Collected Results

The field data collected for Bent 57 were found to have significant error in several of the measurements. The collected data can be found in Appendix E. During the load test there were failures of the sensors that lead to the error in the data set, Blanchard (2004).

4.4.2 MUSTANG Results

The findings of MUSTANG given the collected data were inconclusive. The left spring had a change in parameter that continued to increase the stiffness of the in plane rotation toward infinity. The right spring had a change in parameter that resulted in the spring stiffness increasing toward negative infinity. In reality a negative stiffness is not possible. Because SAP2000® treats a negative stiffness value as mathematical possibility, the model continues to function. Future parameter limits in MUSTANG will prevent the parameter estimation process from continuing give this result.

This leads to several conclusions. The first is that given the number of bolts used to create the moment connection for Bent 57, a spring stiffness of infinity used to model the frame's moment resistance represents a reasonable assumption. This suggests that the connection is completely resistant to moment and does not yield

under the applied loads. MUSTANG started to show this behavior using the collected field data. However, the negative stiffness next determined shows that MUSTANG is unable to perform valid parameter estimation on Bent 57 using field data. Given previous concerns about data quality and the inconclusive results of MUSTANG, the data is likely not sufficient for parameter estimation. This also illustrated the importance of a well defined system with quality measurement data. The results without significant error failed to create a well defined system from which parameter estimation could take place.

The results of Bent 57 using field data were inconclusive, but they did provide an example where field data was used for parameter estimation. This helped to develop the subroutine for taking in and using post-processed field data for the purpose of parameter estimation.

4.5 Future Work

For future work with Bent 57 it is not recommended that the field data be considered for parameter estimation. Bent 56 was a braced frame that was also used in the Central Artery project. Although Bent 56 was not used in this research, the data shows similar error and is also not recommended for parameter estimation. Both frames could be modeled and used for stiffness comparison in future research.

Chapter 5

Rollins Road Bridge

A key component of bridge condition assessment is the field load test. A load test is necessary to verify the response of a bridge in the field. Field tests, couple with appropriate SHM equipment, can provide a multitude of data. This data can provide a well defined base from which a model updating parameter estimation program can operate. Given the appropriate post processing, MUSTANG can use this type of data to assess a bridges structural condition. Rollins Road presented a firsthand opportunity to observe and take place in a field load test, as well as get a feel for what type of data are produced.

5.1 Background

The Rollins Road Bridge is located in the town of Rollinsford in the state of New Hampshire. Rollinsford is an inland town located northeast of Dover, NH and close to the Maine border, see Figure 30. The Rollins Road Bridge is a simple span overpass that carries Rollins Road over Main Street and the B&M Railroad. Main Street and the railroad track run parallel to each other and create interesting restrictions for the bridge. The daily traffic seen by the bridge can vary greatly. Typical passenger vehicles use the bridge daily as well as tractor trailer trucks.

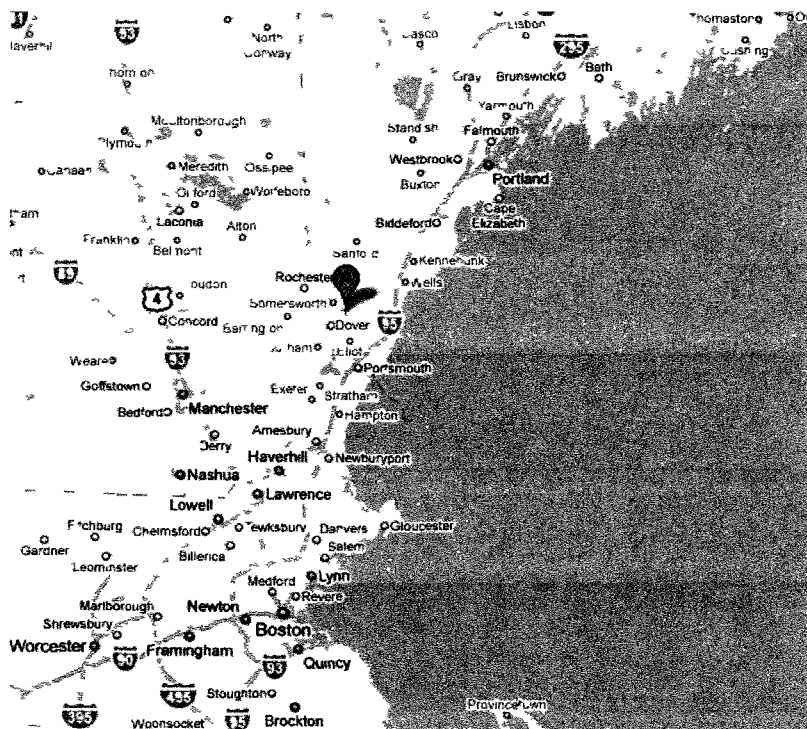


Figure 30: Rollinsford, New Hampshire (Google Maps ®)

The original bridge was built in the 1930's and was comprised of steel stringers and a concrete deck. The four simple spans created a 172 ft total length. The years of deicing treatment during the harsh New England winters had taken their toll on the Rollins Road Bridge. Before its replacement these effects were most visible in the deck. This is evident in the 2000 NHDOT Bridge Inspection Report. This final bridge inspection report showed that the bridge was in need of immediate repair or replacement, see Table 8 Bowman et al (2003).

Table 8 : Rollins Road 2000 Inspection Report (NHDOT 2007)

October 26th 2000 Bridge Inspection	
Deck	3 Serious
Superstructure	4 Poor
Substructure	6 Satisfactory

The bridge was replaced and completed in December of 2000. It was built using funds from the Innovative Bridge Research and Construction (IBRC) program that is administered by the Federal Highway Administration (FHWA). In order to receive funding from the IBRC, a proposed bridge must incorporate high strength and innovative materials as well as include instrumentation to monitor the structure (Sipple 2008). The Rollins Road Bridge utilizes carbon fiber reinforced polymers (CFRP) in the deck and in the precast prestressed girders. The CFRP is used in place of

traditional steel reinforcement. The girders consist of high strength concrete, traditional prestressing strand, and CFRP. The CFRP is used vertically for shear reinforcement. In the deck, the CFRP is placed in a grid pattern. Fiber optic strain gauges and temperature sensors were attached to the CFRP. This was done on site as well as at the precast plant before the concrete was poured. The purpose of the instrumentation was to monitor the performance of the new and innovative materials. The instrumentation plan was not designed with the intention of using the data for overall condition assessment and parameter estimation. The instrumentation is all linked into a data acquisition system, DAQ, which stores the data. The data can be retrieved by calling the DAQ, which is linked to a modem, or by using a laptop on site.

The Rollins Road Bridge has undergone three load tests since its construction in 2000. The first was done fifty six days after the December 2000 construction in order to establish a baseline for the data collected from the instrumentation. This baseline was used to determine the behavior of the deck and girders with the new, undamaged materials. A 75.6 kip truck was used for this load test. The truck was stopped at predetermined locations and the time was marked to correlate with the data collected by the DAQ. The next load test was performed in August of 2001, approximately nine months after the initial load test. This test was also intended to observe the reactions of the CFRP, girders, and deck. This test utilized a 76.9 kip truck. This test was done to ensure that the bridge and its new materials were still performing as expected. These load tests were performed with the intent of proving the capabilities of the CFRP and

high strength concrete in comparison to more traditional bridge construction. The next load test would not take place until April of 2008.

The Rollins Road Bridge was inspected on July 9th of 2007. The NHDOT bridge inspection report scored the deck, superstructure, and substructure all ratings of 9 (NHDOT Bureau of Bridge Design, 2007). The highest possible rating is a ten. This shows that over seven years, the Rollins Road Bridge and all of its innovative materials are still performing as designed and show limited, deterioration or decrease in capacity.



Figure 31: Rollins Road Bridge

5.2 April 2008 Field Test

The next load test for the Rollins Road Bridge took place in April of 2008. Given the research into MUSTANG taking place at the time, this load test tried to focus on utilizing the data for assessing the structural health of the bridge using parameter estimation. Given the excellent marks from the NHDOT inspection in 2007, this load test focused on using the collected data to assess the condition of the elastomeric bearing pads. If the bridge was exhibiting behavior inconsistent with a healthy bridge, these behaviors would be attributed to the bearing pads. Much preparation went into this load test in order to capture as much data as possible. Two DAQ's were used during the load test, the permanent on site and one rented for the load test. In the weeks leading up to the test, several site visits were performed by Sipple (2008) and Welch to determine the status of the strain gauges and temperature sensors. A small number of gauges had been damaged during the construction phase. All of the sensors had an identification number that related to their position within the structure.

This load test was done in similar fashion to the first two. An NHDOT supplied truck was used to apply a controlled load on the bridge. To start the load test a zero load reading was taken using the DAQ. Traffic was stopped from crossing the bridge, courtesy of the Rollinsford Police Department, while the readings were being taken. This reading was to be used as the benchmark from which to compare the load influenced measurements. The truck was unfortunately significantly lighter than the

trucks used for the first two load tests. The New Hampshire State Police used a mobile weigh station to accurately determine the trucks weight. The two axle truck weighed in at 37.4 kips. This was significantly lighter than the trucks used in the two previous load tests. Given the excellent NHDOT rating of the Rollins Road Bridge and its elements, it was determined that any deterioration would take place in the elastomeric bearing pads located at the bridge abutments. Their performance was to be highlighted by performing parameter estimation on the strain data collected from the test.



Figure 32: Trooper Huddleston (NH State Police) Mobile Weigh Station

There were several different types of measurements taking place during the Rollins Road load test. The internal temperature and strain gauges were recording during the duration of the test. Once the truck was stopped on the predetermined loading point, the time was noted on the computer that was recording the data from the DAQ. This was done to sync the loading with the response of the system. The NHDOT was also taking survey measurements on the underside of the girders at the center of the span. They also utilized the zero load reading to calculate their displacements. A bucket truck was used to hoist the crew member to the bottom of the girders to hold the measuring stick.

Optical measurements were taken by two teams representing industry and one group of UNH students, all using different camera systems. The optical measurements focused on girder five, the visible girder in front of the cameras. The UNH students had difficulty extracting any useful data from the test. The data quality was unreliable.

One of the most influential factors of the load test was the change in temperature. The morning started out chilly and overcast, but by mid day and the end of the test it was significantly warmer and sunny. This change in temperature had a dramatic effect on the strain recorded on the bridge throughout the day. The internal temperature sensors recorded the change in temperature throughout the load test. The ambient temperature was also measured above and below the deck. The effects

of the change in temperature and how they were dealt with are discussed further in the following chapter.



Figure 33: Loaded Truck, Survey Team in Bucket Truck, and Technical Support

5.2.1 Data Quality

In order to rule out structural damage to the precast girders and the deck, Sipple (2008) compared the strains recorded from each of the three load tests at specific locations. This also helped to determine which strain gauges were still functioning. A direct data to data comparison was done to compare the bridge response and see how things have changed over the last eight years. Gauges from the CFRP in the deck and the girders were all compared with a zero loading. The difference was caused by varying weather patterns. Weather patterns can have a significant impact on strain readings due to their small magnitude of measurement. This is discussed further in the following section.

5.2.2 Environmental Effects

Environmental effects play a huge role in the daily life of a structure. Structural elements can experience a great amount of strain due to the expansion and contraction of materials. This strain is caused by the change in temperature experienced by the bridge through the day. This trend is shown by viewing a plot of strain vs. time. It was found that the maximum strain induced by the truck was approximately three microstrain. Throughout the test the bridge experienced a change in strain of twenty five microstrain due to thermal loading. Three zero load readings were taken throughout the load test. Using these three readings and the different coefficients of thermal expansion, Sipple (2008) was able to use an empirical correction to remove the strain caused by temperature.

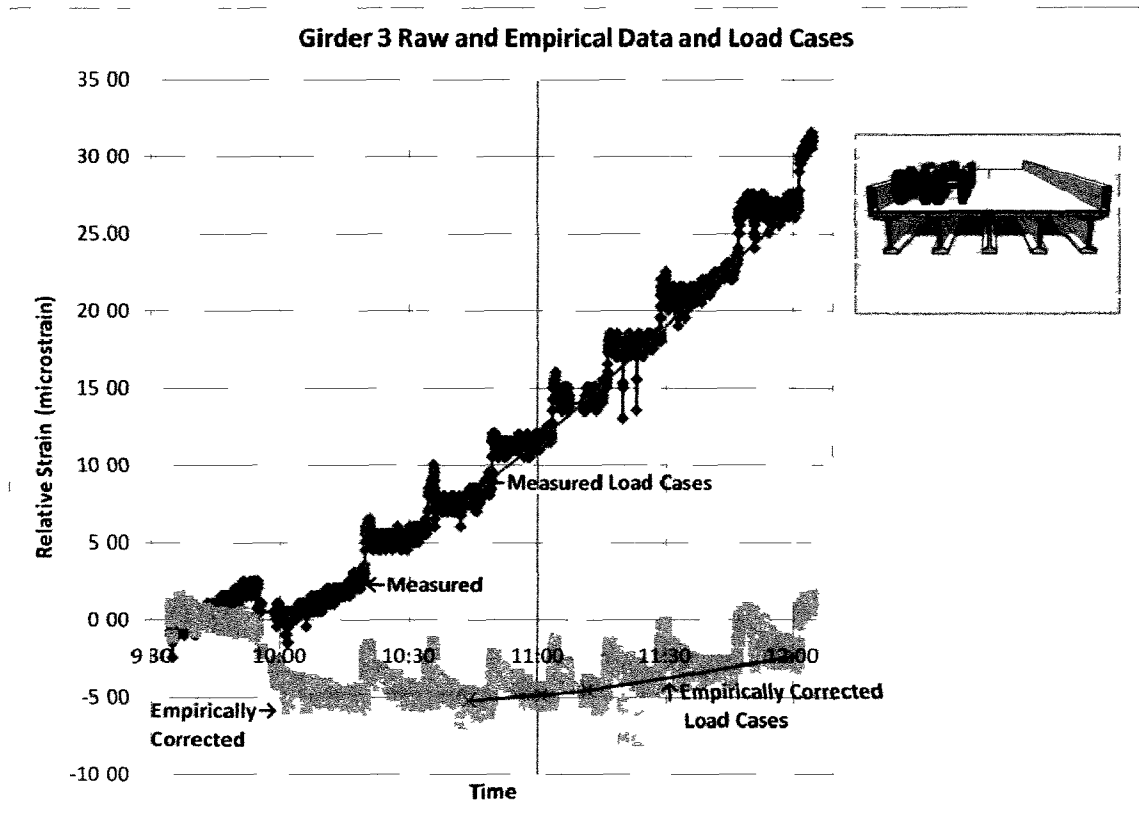


Figure 34 : Girder 3 Raw and Empirical Data (Sipple 2008)

This empirical correction for thermal effects takes place while post-processing data. SAP2000® is capable of applying thermal loads to a structure. The error functions utilized by MUSTANG only work with point loads for parameter estimation. Applying thermal loads to the model would not work in MUSTANG. Therefore, the data are corrected before being used for parameter estimation. The zero load readings were also used to effectively “zero out” the strain gauges. Due to locked in stress from casting and the previously discussed prestressing effects, the fiber optic gauges are constantly reporting a strain, not zero. Using the three zero load readings, Sipple (2008) was able to take the data and start the load test results at zero. The strains due

to the load of the truck are then based at zero. This allows for the effects of the truck on the bridge to be the focus of the load test. The locked in strains and temperature strains are normal for the bridge. The strain induced by a heavy load, a truck load, would be the main cause for concern.

CHAPTER 6

Rollins Road Modeling and Parameter Estimation

Several different approaches to condition assessment are presented with the Rollins Road Bridge. The first is typical visual inspection. A visual inspection can find problems that appear on the surface. The next approach utilized the SHM instrumentation initially installed in the Rollins Road Bridge at construction. This instrumentation was used to verify the performance of the new and innovative structural reinforcement. MUSTANG was developed after the planning and construction phases of the bridge. Problems encountered using the instrumentation

and existing models for the purpose of parameter estimation, illustrate the fact that a shift in thinking is necessary to fully utilize model updating parameter estimation programs, such as MUSTANG.

6.1 Rollins Road Bridge Model, Sipple (2008)

Structural Health Monitoring for parameter estimation can be looked at as having three major components necessary for success. The first component starts in the field with a robust and informative instrumentation plan. This instrumentation provides the measurements from which the condition of the structure is assessed. The middle component is the parameter estimation program, MUSTANG, which takes the field response and compares it to the predicted response of the structure. That predicted response comes from the last component, the model. An accurate model is just as essential to the process as the data from the field. The saying “garbage in garbage out” holds true to the model as well as the field data. If the model is not a true representation of the structure, any conclusions drawn from the parameter estimation will not be valid. An accurate model is a crucial piece for successful parameter estimation. This highlights the importance of including modeling from the beginning of the design process as an effective tool for a DOT or bridge owner.

The Rollins Road Bridge SAP2000® model, created by Sipple (2008), includes the effects from the CFRP, prestressing strand, and the steel reinforced elastomeric bearing pads. The bridge modeler in SAP2000® was used to create the model for the

Rollins Road Bridge (Sipple 2009). The deck was modeled using layered shells. This allowed Sipple (2008) to use a layer to represent the CFRP reinforcement. This layer contained scaled values to represent the parameters of the CFRP grid spread out over the area of the layer. The other shell layers contain the properties of the high strength concrete.

Using the Bridge Modeler in SAP2000® allowed Sipple (2008) to select the appropriate New England Bulb Tees for the girders as well as specify the appropriate strand pattern for the prestressing strand. The girder parameters were used with frame elements in the Rollins Road model.

The elastomeric bearing pads were modeled as translational and rotational springs created by link elements. These were the boundary conditions for the bridge. Using a spring as a boundary condition more accurately models a bridge's support in the field than using either a pinned or fixed connection. The pinned connection has no resistance to moment which would result in excessive rotation in the analysis results. A fixed connection allows for no rotation in the joint. This would result in lower values of deflection at the center of the bridge span and a transfer of moment to the abutments. The stiffness parameters of these springs were determined with a combination of equations from research into the topic (Stanton, Roeder, Mackenzie-Helnwein, White, Kuester, & Craig, 2008). The "true" parameters of these bearing pads was one of the main focuses of Sipple (2008) research.

One of the more complex elements of the Rollins Road Bridge model is the load application. Using SAP2000® BrIMTM, Bridge Modeler, nodes are placed according to a certain programming algorithm. These nodes more often than not will not line up with the applied truck loads. The error functions programmed into MUSTANG require loads to be applied at joints. Sipple (2008) created a finite element mesh to be placed over the bridge deck. This mesh took the distributed wheel loads and distributed the force resultants to the SAP2000® created nodes.

6.1.1 Removal of Effects Due to Prestressing and Dead Load

In order to focus on the structural response due to the truck load, the effects due to prestressing in the New England Bulb Tee's and the superstructure's self weight need to be removed from the results. There are two ways in which SAP2000® models prestressing. One way is to apply a point load at the end of the frame element representing the girder. It does not model the behavior of the prestressed girder. The properties of the frame element do not change if the girder is prestressed or just ordinarily reinforced. The other option is to connect a tendon object at joints along the frame member to represent prestressing strand. This also affects the models behavior without modifying the frame properties. This poses a problem for MUSTANG in its current condition. This will be discussed later.

Sipple (2008) was able to get around this additional strain by creating two models. This first model contained only the prestressing load and self weight of the bridge. The model was run and the strains were recorded for this "zero load" state.

The second model had the prestress forces, the self weight, and the applied truck load. The strains were recorded for this load case. The strains from the first model were then subtracted from the strains for the second model. This left only the strain due to truck load. This was comparable to the adjusted field data that is discussed in the next section.

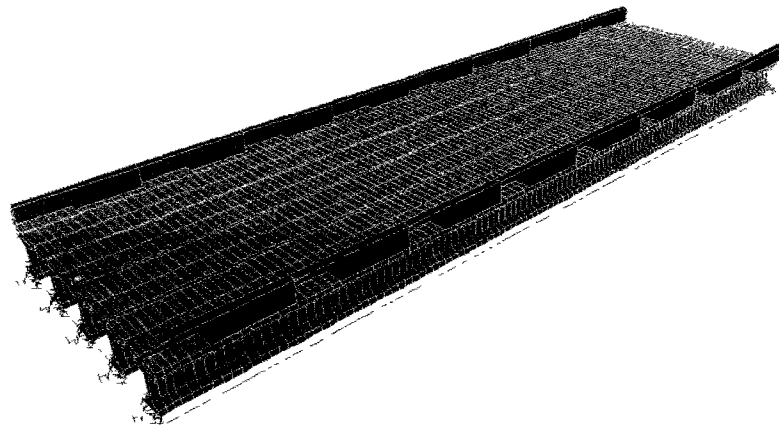


Figure 35 : Rollins Road Bridge SAP2000® Model, Welch (2011)

6.1.2 Results Sipple (2008)

The results of the load test were used by Sipple (2008) to perform manual parameter estimation. This process involved running the initial bridge model in SAP2000® with the assumed parameters. These results then had to be post processed before they could be compared to the load test results. SAP2000® does not calculate

strains for frame elements. This poses a problem for direct comparison to the load test results. This also creates a problem which MUSTANG must overcome by using the transformation matrix and the strain displacement matrix. This extra step can introduce error into the parameter estimation. For the manual parameter estimation, Sipple (2008) used the assumption that the bridge responded in the linear elastic range. This is a reasonable assumption given the weight of the truck compared to the overall strength and redundancy of the bridge. This led Sipple (2008) to the assumption that the strain varied linearly through the girder. Displacements above the given strain gauge were taken and used to calculate the strain. The change in element length was used to calculate the strain at the deck level and the center line of the girder. Sipple (2008) then utilized linear interpolation to calculate the strain at the level of the strain gauge. This was then post processed to remove the effects of prestressing. Then it was finally able to be compared to the field data.

This process creates several problems for MUSTANG in its current state. The instrumentation also creates a problem for direct parameter estimation using field data. MUSTANG typically runs a model, uses the calculated deflections and converts them to strain using the Bmatrix algorithm. The Bmatrix algorithm uses the transformation matrix, the strain displacement matrix, and the strain gauge locations to convert the displacements and rotations into strains. These strains are then directly compared to the post processed strains from the field data.

6.1.3 Challenges for MUSTANG

The data collected at Rollins Road and the model used by Sipple (2008) both presented problems for MUSTANG. The instrumentation used in the bridge does not create a favorable scenario for parameter estimation. The SHM system was intended to verify the performance of the materials, not the structure as a whole. This provides a very small picture of the bridges response during loading. Strain gauges show the behavior of the materials for the purpose of comparison to the more traditional bridge materials.

The strain gauges are never at a true zero or consistent starting point. Strain gauges are highly sensitive to environmental variability. Thermal loads can cause significant strain variation through a day. This can introduce a significant amount of error into the parameter estimation. At the very least it requires an empirical correction; this also can be a source of error. Given that the bridge is dealing with these locked in stresses due to prestressing and self weight, as well as environmentally caused stresses, a zero load reading is used to correct the data further to start the strain data from a zero reading.

This leads to the requirement of having two models to effectively “zero out” the modeled response. This is done to remove the effects of prestressing and dead load in order to have the response, due to the truck load, start at zero. MUSTANG is not set up to do this in an automated fashion. MUSTANG runs the model with the

initial parameters and compares the results to the field data. Using the Rollins Road Bridge Model and the 2008 field test data for parameter estimation in MUSTANG is not currently possible.

6.2 Simulated Data Runs

MUSTANG is not currently programmed to perform parameter estimation on the Rollins Road Bridge using the field data obtained from the April 2008 load test. As a test the Rollins Road Bridge was run in MUSTANG with simulated data to resemble the findings of Jesse Sipple. Running the Rollins Road Bridge model through MUSTANG proved to be a sizeable task. This exercise highlighted several areas of MUSTANG that needed to be updated in order to accommodate parameter estimation using these massive and complex bridge models.

6.2.1 Creating a MUSTANG Compatible Model

Before running MUSTANG some modifications to the model were required. The stiffness pads were modeled using link elements instead of springs. These were replaced with spring elements of the same stiffness.

The first problem encountered when using MUSTANG to run the Rollins Road Bridge model involved importing the degrees of freedom. Some of the referenced degrees of freedom were negative. This caused a reference error. Upon further investigation it was determined that degrees of freedom less than zero within the txe file exist due to restrained nodes. These nodes do not contribute to the stiffness

matrix as their movement is controlled by other nodes. To simulate a bridge deck SAP2000® uses a finite element mesh of area elements. Many of the nodes are constrained to one another in order to emulate the movement of a bridge deck. Using an “if” statement requiring the value be greater than zero otherwise ignored, solved the negative degree of freedom problem.

The next problem was the length of the txe file. After the active nodes are listed, the body constraint names are then listed. MUSTANG was trying to import a string into a matrix. This couldn't be solved by using a cell because it would result in overlapping degrees of freedom that don't actually affect the stiffness matrix. In order to assure that the correct number of nodes was read from the text file, a limit other than the number of lines in the txe file needed to be used. To obtain the correct number MUSTANG runs the model and calls the txk reader, the function used to import the stiffness matrix 'k'. The number of rows, or columns as the matrix is square, is then used to limit the number of lines read by the txe reader. Before accounting for this error, MUSTANG was developing force vectors and displacement vectors that were not the appropriate lengths. These vectors contained duplicate information and could not be multiplied by the stiffness matrix due to their incorrect lengths. Once these two problems were addressed, MUSTANG was capable of running Rollins Road Bridge model.

The Rollins Road Bridge data file set up the bridge model to use the center bearing pad, modeled as a spring, as the unknown element. Only one spring was set to

be unknown to simplify the initial parameter estimation. The initial bearing pad stiffness was set to $1e9$ in-kip in all directions. The true value was set to $1e5$ in-kip in the Global Y direction. The rest remained at $1e9$ in-kip. This would represent the girders bending in the middle and wearing down of the bearing pads rotational resistance in that plane. The four orders of magnitude difference was to make any change noticeable in the model.

The first load case chosen for this test was the 2008LC1 from the model, Sipple (2008). This load case consists of eight point loads representing the different tire loads. The point loads range from as low as 2.34 kips to as high as 7.31 kips and are located close to the center of the bridge. In order to keep things simple only one load case was chosen for the first run.

Measurements were taken at the midpoint of each girder. The measurements used for parameter estimation were the displacement in the z direction and rotation in the y direction. Given the lateral support provided bridge deck, these would be the most pronounced displacements. The static stiffness error function was used to run the parameter estimation. MUSTANG was able to run mathematically but the results were of no use. With the static stiffness error function, when not all nodes are measured it is necessary to invert the stiffness matrix. This includes sensitivity matrix. The stiffness matrix was 7604 rows by 7604 columns. The number of zeros within the sensitivity matrix vastly outnumbered the ones. When the matrix was inverted it was extremely close to singular. The multiplication required to determine a change in

parameter resulted in MATLAB® calculating infinity due to the badly scaled and close to singular matrix.

Adding measured degrees of freedom would not solve the problem with the matrix inversion due to the size of the bridge's stiffness matrix. The next option is to use the static strain error function. The B matrix is full of zeros if a degree of freedom does not contribute to the parameter estimation. The multiplication drops out the terms that do not contribute. The static strain error function does not require the inversion of the sensitivity matrix.

Ten locations per girder were chosen, five on either side of the center of the bridge. The simulated strain gauge location was located at the midpoint of the frame segment length and down eighteen inches from the center. These locations are close to the bottom face of the girder and the middle of its length in order to observe the most deflection.

The first parameter estimation again used only 2008LC1 from the Rollins Road Bridge Model, Sipple (2008). Only one load case was chosen to determine if MUSTANG was capable of running the static strain error function on a model of this size. MUSTANG ran to completion, but the results were inconclusive. The change in parameter was consistently of the order of magnitude of $1e-6$. This shows that the results are invalid. Next MUSTANG utilized all four load cases. The parameter estimation attained similar results. Upon further investigation it was discovered that simply changing the shell modifiers to zero was not eliminating their contribution to

the stiffness matrix for the purpose of obtaining the sensitivity matrix. Further research into shell elements and their contribution to the SAP2000® stiffness matrix is necessary to fully utilize these element types in combination with frame, spring, and other elements to create complex models for the purpose of parameter estimation.

6.2.2 Bridge Model for MUSTANG

In order to illustrate the capabilities and potential for MUSTANG, a parameter estimation friendly bridge model was created. The model consists of five steel girders, W44x285 spanning ninety feet. There are five cross members, W24x146, connecting each girder to the adjacent girder. These represent the stiffness of the deck and provide stability for the top flange of the girders. Each girder is supported at the end by a spring connection resembling a bearing pad. The spring stiffness was set to 1e9.

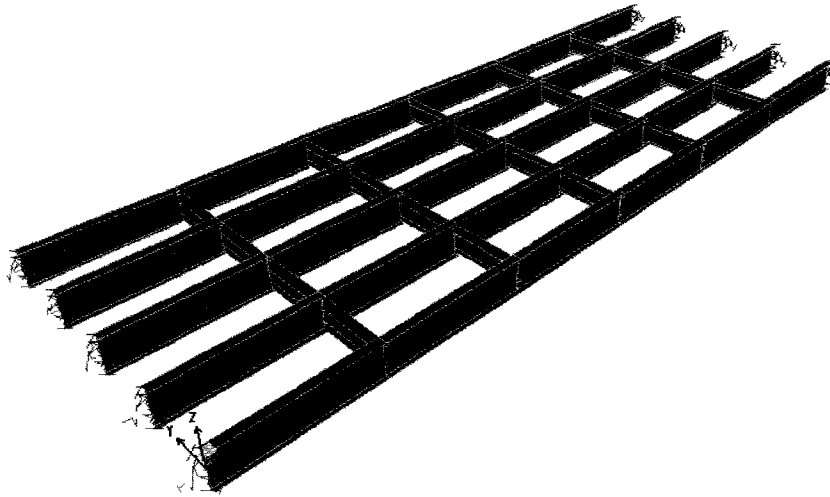


Figure 36: Simple Bridge Model (SAP 2011)

6.2.3 Parameter Estimation with Simple Model

For verification of MUSTANG's ability to process a large structure, the Simple Bridge model was run with MUSTANG. The bearing pads for girder one were chosen as the damaged elements. The rotational stiffness in the plane of the girder was the specified unknown. This would simulate wearing out due to repeated loading. An eight kip point load was placed at nodes three, four, ten and eleven. This was done to represent a truck wheel load. Displacement and rotation measurements were taken at points along girders one and two.

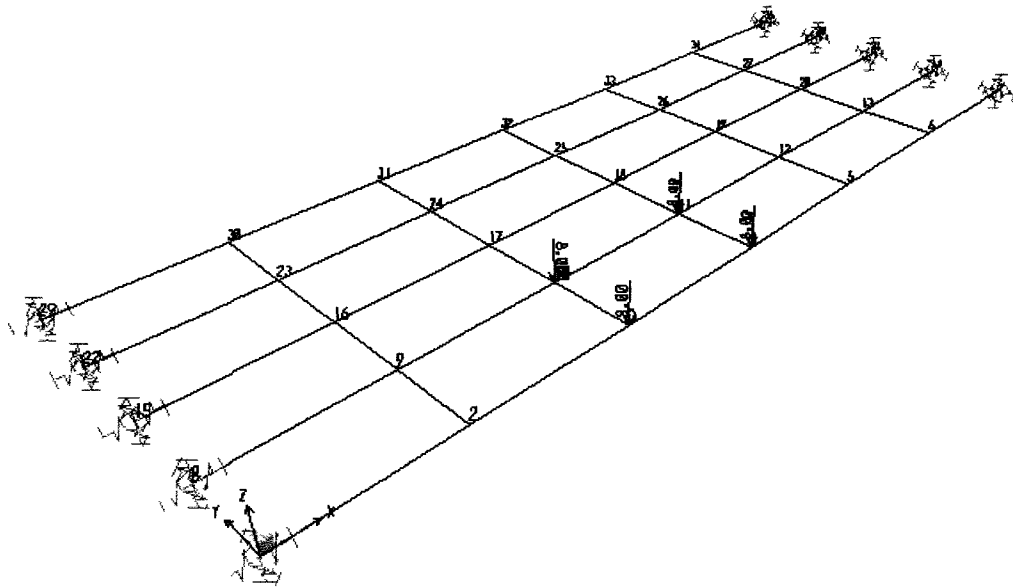


Figure 37: Simple Bridge Wheel Loads (SAP 2011)

The simulated displacements were created by using half of the original rotational stiffness's of the springs at nodes one and eight. The static stiffness error function was used by MUSTANG for parameter estimation. MUSTANG reached convergence with the "true" parameters after two iterations. This shows a well conditioned system.

Table 9: Simple Bridge Results

Case	Member	Parameter		Iteration		Iteration	
		True Value	Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True
1	1	K Rotational		1		2	
		1.000E+05	1.000E+09	1.00E+05	0.00	1.00E+05	0.00
	8	K Rotational		1		2	
		1.000E+05	1.000E+09	1.00E+05	0.00	1.00E+05	0.00

6.3 Discussion

The successful parameter estimation by MUSTANG for the Simple Bridge model highlights several key themes that have been reoccurring throughout this research. One of the keys to successful parameter estimation is a well defined system. For field studies this starts with the placement of the SHM equipment. Strain and displacement measurements taken in critical areas will exhibit the behavior of the entire system and contribute to identifying structural deficiencies. Another reoccurring theme is the need to include structural modeling from the beginning of the bridge design process. An accurate model is a necessary component for parameter estimation. Without a true representation of behavior in the field, accurate conclusions can not be drawn from parameter estimation.

The incorporation of different element types into MUSTANG will be necessary to most accurately model bridge behavior in the field. Future recommended work will include studying the effects of different element types on the stiffness matrix produced by SAP2000®. Adding SHM instrumentation and building structural models after construction can prove to be difficult and limited in providing useful data for

asset management. The time to shift the thinking of bridge management has arrived, given the current period of construction and renovation. Including modeling and instrumentation from the design phase will allow owners to use model updating parameter estimation programs, such as MUSTANG, to manage bridges given real time structural response.

CHAPTER 7

Conclusion

7.1 Contribution

This research focused on creating a parameter estimating and model updating program to be used with physical structures, specifically bridges. MUSTANG utilizes published error functions programmed with the very powerful analytical program MATLAB®. This is used in conjunction with a powerful modeling program, SAP2000®. These programs are used to avoid recreating the wheel as many parameter estimation programs do. Using such an established modeling program allows the entire process of structural health monitoring to start from the very beginning of the design process. An accurate model is necessary to predict the behavior of a yet to be built structure. This

model can now continue to be used in conjunction with a structural health monitoring system for condition assessment. MUSTANG will be able to take the model and the data and use them to determine the health of a system in practically real time. This would eliminate the need to set up a field test for an existing structure. If the structure is older, a model would need to be developed using existing drawings. This is the first introduction of error due to the inherent unknowns of modeling an already built structure.

Having SHM and parameter estimation involved from the beginning of design through service will greatly benefit the owner and change the way bridges are inspected and maintained. If an overloaded truck is detected, an analysis can be done to assess any damage. If a visual inspection notices an area of interest, MUSTANG can specify members and check their status. MUSTANG has the capability to make bridges safer and maintenance more cost effective.

7.1.1 MUSTANG – Verification by Published Results

Utilizing published cases of parameter estimation, Sanayei et al. (1991) and Sanayei et al. (1999), it was possible to develop a model updating parameter estimation program based on published results. These cases not only helped to develop the programming of MUSTANG, but they helped to verify the validity of the programmed error functions utilized by MUSTANG.

7.1.2 MUSTANG – Field Collected Data

Without error free, clean data, MUSTANG was not able to perform parameter estimation on a structure. However, with the consideration of incorporating this field data MUSTANG was programmed to accept field measurements, including strain, displacements and rotations. This will allow future research to input field data for use in parameter estimation.

7.2 Observations and Areas of Importance

7.2.1 Element Types

MUSTANG is capable of performing parameter estimation on SAP2000® models containing several different types of elements. This includes frame, spring, and shell elements. All of these different element types can be manipulated to create the sensitivity matrix. This is done either using property modifiers or, in the case of springs, by changing their properties temporarily. MUSTANG can use members made of either frame or spring element types as an unknown parameter. These two can be used for model updating using parameter estimation. Members of the shell element type are not yet capable of being utilized by MUSTANG as an unknown parameter.

Another problem encountered by MUSTANG was the development of the sensitivity matrix using shell elements connected to frame elements. Using modifiers was not sufficient to zero out the effects of the shell elements. Further research into the effect of shell elements on the stiffness matrix will be necessary to develop the sensitivity matrix given a complex model that contains both element types.

7.2.2 Loading

SAP2000® is capable of many different load types and combinations. The different load types include point loads, distributed loads, body forces, and thermal loading. The error functions programmed into MUSTANG are only compatible with point loads. Currently, models containing loading other than point loads can not be used for parameter estimation. This was evident in the Rollins Road load test results. The thermal effects had to be removed from the data before it could be used for any type of parameter estimation. Although the data were not used by MUSTANG, removing the thermal effects from the reactions allowed a model comprised of only point loads to be created. Even distributed wheel loads need to be translated into point loads before the model can be used in MUSTANG.

The error functions currently utilized by MUSTANG only accept point loads. Field data that is not post-processed to remove temperature strain will be of no use to MUSTANG without the incorporation of a thermal loading into the model. Future research into thermal loading and its effects on the force vector in SAP2000® will be necessary to utilize these sets of complex data from field tests.

7.3 Future of MUSTANG

With every passing year more of our nation's bridges approach the end of their design life span. Harsh environmental conditions and an ever growing population have

deemed many of these bridges functionally obsolete regardless of their physical state. Overloading, harsh environmental exposure, and underfunded infrastructure budgets have rendered many other bridges structurally deficient. In order to keep the public safe, these bridges need to be evaluated for safety. Many bridges will be posted to keep them from experiencing loads that will test the structure. Still many bridges will need to be replaced to meet the demand of a growing nation while keeping its citizens safe. Both of these solutions present excellent opportunities to utilize a structural health monitoring system and a model updating parameter estimation program.

A load test can be performed on a bridge that was constructed without a built in SHM system. The use of optical measuring devices and targets affixed to the bridge provide an excellent measurement tool for a load test. This type of load test can be used on older bridges that are nearing the end of their design life. The results from this type of test can be entered into MUSTANG to determine the health of the structure. The engineer can then use the results to make recommendations to the bridge owner. If the bridge shows little sign of damage then the owner can rest knowing that their bridge is safe. If damage is detected, the severity of it will determine the course of action. If MUSTANG calculates that the response of a certain girder is 65% of what is to be expected, the owner knows that a serious problem exists. Some resolutions to ensure safety on deteriorating bridges include posting weight limits or limiting the number of lanes of traffic.

There is government funding being made available for construction projects across the country. This funding is coming via the American Recovery Act. Many bridges are in need of replacement due to deterioration and/or being unable to handle the volume of traffic. This presents a perfect opportunity to include SHM and ITS instrumentation from the beginning phases of design to the construction and operation of a new bridge. The involvement of these systems from the very beginning will allow engineers to place instrumentation at critical locations along the structure. A proper model of the new bridge should also be developed beginning in the design phase. This would create a well defined system for MUSTANG to perform parameter estimation. Because the structure is new, initially MUSTANG would be used to verify the bridge is behaving properly after construction. It would also be used to check on the bridge in the event of an overloaded truck passing or an accident involving vehicles and the structure.

7.3.1 Future Programming

Given the modular programming approach used to create MUSTANG, it is easy to update the program with new error functions, load types, and element types. There are several functions that can be programmed and added to MUSTANG to make it a more well rounded program.

One of the more useful error functions for parameter estimation on bridges will be using dynamic loading. A bridges mode shape under dynamic loading is a very

useful indicator of the structural health of a structure. The error function is written in its own m file and can be easily added to MUSTANG's list of error functions. This error function will look very similar to those already programmed in MUSTANG.

Another area that requires work is that of error function and parameter normalization. When using strain data and even displacement data for a bridge, the response is several orders of magnitude smaller than the stiffness of the structure. This is not a problem when using similar data. The data retrieved from SAP2000® is very precise and there is not loss of quality when comparing $F - KU$ to the "field" data. Actual field data does not have that kind of precision. Normalization is required to assure that no data quality is lost before or during the parameter estimation.

Different load types present a unique challenge. Dynamic error functions deal with dynamic loading. Static error functions deal with static point loads. Structure experience loading that is distributed. The loading information obtained from a SHM or ITS system will need to be preprocessed before it is useful to MUSTANG. An algorithm can be programmed for MUSTANG to convert these distributed loads to point loads. These point loads will be applied at actual nodes in the model at the magnitude that will simulate the effects seen by applying a distributed load to structure.

The next type of loading that will need to be accounted for will be thermal loading. An algorithm will need to be programmed to use the recorded temperature effects and the coefficients of thermal expansion in the bridge to determine the strain

and/or displacements due to thermal effects. These displacement or strain vector will then need to be corrected to reflect the structural response with these values removed in order to perform parameter estimation on a structural element due to load.

The next area of focus for the future of MUSTANG will be the element types. Shell elements require additional programming to be utilized for parameter estimation. The shape functions used in SAP2000® are required in order to use strain data for gauges located within a shell element. The shell elements are more complex than frame elements and would be able to paint a more accurate picture of the structural response. Solid elements are also available for modeling using SAP2000®. These elements have not been included in the programming of MUSTANG. Solid elements also represent an opportunity to create a highly accurate model.

Using complex models such as Rollins Road Bridge presents many challenges for parameter estimation. Further investigation is required in order to use these models in conjunction with MUSTANG for parameter estimation. There could be several reasons for the inconclusive results obtained by running the Rollins Road Bridge model. The model was created using SAP2000® Bridge Modeler. This takes away control from the user. There exists the possibility for unwanted boundary conditions, element types, and loading conditions. Further analysis and investigation of the bridge model is required to prepare it for parameter analysis.

The programming of MUSTANG has made it very easy to update. This will be crucial in allowing MUSTANG to reach its full potential. The possibilities are virtually endless for a model updating parameter estimation program linked with an advanced modeling program such as SAP2000®. As more error functions are discovered and published, MUSTANG will be ready and capable of adding them in an effort to create a robust program. A healthy infrastructure is critical to the development of our country and the growth of its people. The use of new technologies will ensure that this infrastructure is safe for all who use it.

Works Cited

- Bell, E.S., Sanayei, M. Javdekar, C.N., & Slavsky, E. (2007). Multiresponse Parameter Estimation for Finite-Model Updating Using Nondestructive Test Data. *Journal of Structural Engineering – ASCE*, 1 067 – 1079
- Blanchard, A. (2004). *Objective Function Investigation and Load Case Selection for Parameter Estimation with Non-Destructive Test Data from Boston's Central Artery*. Medford, MA: Tufts University
- Bowman, M.M (2002). *Load Testing of the Carbon FRP Grid Reinforced Concrete Bridge Deck on the Rollins Road Bridge, Rollinsford, New Hampshire*. Durham, NH: University of New Hampshire
- Gifford, J., Stalebrink, O. (2003) Transportation Asset Management: A Vehicle for Mainstreaming ITS?
- Harrington, P.F. (1998). The challenges of underpinning the central artery. *Civil Engineering Practice*, Fall/Winter, 65-76
- Los Alamos National Laboratories. (1997) DIAMOND. Los Alamos, New Mexico
- NHDOT Bureau of Bridge Design. (2007). *Bridge Inspection Report – Rollinsford 091/085*. NHDOT.
- NHDOT Bureau of Bridge Design. (1999, July). Rollins Road Bridge Over B&M Railroad and Main Street Plans.
- Sanayei, M. (1997). PARIS – PARAmeter Identification System©. Medford, MA: Tufts University

- Sanayei, M. and Nelson, R.B. (1986). Identification of Structural Element Stiffnesses from Incomplete Static Test Data. *Proceedings of SAE Aerospace Technology Conference and Exposition – SAE-861793, Long Beach, CA.*
- Sanayei, M. and Onipede, O. (1991). Damage Assessment of Structures Using Static Test Data. *AIAA Journal*, 29(7), 1174-1179
- Sanayei, M. and Saletnik, M.J. (1996a). Parameter Estimation of Structures from Static Strain Measurements; I: Formulation. *Journal of Structural Engineering – ASCE*, 122(5), 555-562
- Sanayei, M., McClain, J., Wadia-Fasctti, S., and Santini, E. (1999). Parameter Estimation Incorporating Modal Data and Boundary Conditions. *Journal of Structural Engineering – ASCE*, 125(9), 1048-1055
- Santini-Bell, E., Sanayei, M., Brenner, B., Sipple, J., & Blanchard, A. (2008). Nondestructive testing for design verification of Boston's Central Artery underpinning frames and connections. *Bridge Structures – Assessment, Design and Construction*, 4(2), 87 - 98
- Santini, E.M. (2003). *Using Multiple Non-Destructive Test Data Types and Data Sets for Condition Assessment of Bridge Decks*. Medford, MA: Tufts University
- SAP2000 (2010). Version Advanced 14.2.4. Berkeley, California, USA: Computer and Structures, Inc.
- Sipple, J. (2008). *Structural Modeling and Monitoring of the Rollins Road Bridge for Condition Assessment*. Durham, NH: University of New Hampshire
- Y. L. Xu, M.ASCE, W. W. Guo, J. Chen, K.M. Shum, and H. Xia. (2007). Dynamic Response of Suspension Bridge to Typhoon and Trains. I: Field Measurement Results. *Journal of Structural Engineering – ASCE*, 3 – 11

APPENDICES

APPENDIX A – TWO DIMENSIONAL BRIDGE PARAMETER ESTIMATION, CASE 1

Case	Member	Parameter	Iteration	Iteration	Iteration	Iteration	Iteration	Iteration	Iteration				
1	Area	True Value	Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True
		1.76F-07	2.552F-02	1.932F-02	7.97	1.937F-02	4.28	1.741F-02	0.94	1.783F-02	0.26	1.761F-02	0.05
1	Moment of Inertia	True Value	Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True
		1.624F-03	1.247E-03	2.220F-03	55.94	1.804F-03	11.11	1.587F-03	1.45	1.623F-03	0.11	1.620E-03	0.19
2	Area	True Value	Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True
		1.76F-07	2.550F-02	1.930F-02	4.56	1.900F-02	1.54	1.747F-02	0.56	1.782F-02	0.18	1.761F-02	0.04
2	Moment of Inertia	True Value	Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True
		1.624E-03	1.247E-03	2.568E-03	58.19	2.084E-03	28.30	1.822E-03	12.19	1.704E-03	4.91	1.655E-03	1.91
3	Arco	True Value	Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True
		1.28E-02	2.560E-02	1.338E-02	4.56	1.300E-02	1.54	1.287E-02	0.56	1.282E-02	0.18	1.281E-02	0.04
3	Moment of Inertia	True Value	Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True
		1.624E-03	1.247E-03	2.250E-03	39.56	1.804E-03	11.04	1.700E-03	3.07	1.657E-03	1.85	1.620E-03	0.09
4	Area	True Value	Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True
		1.28E-02	2.560E-02	1.649E-02	0.56	1.400E-02	1.24	1.284E-02	0.56	1.280E-02	0.18	1.281E-02	0.09
4	Moment of Inertia	True Value	Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True
		1.624E-03	1.247E-03	2.559E-03	56.15	2.087E-03	28.80	1.822E-03	12.19	1.709E-03	4.91	1.655E-03	1.93
5	Area	True Value	Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True
		1.28E-02	2.560E-02	1.374E-02	7.32	1.312E-02	2.48	1.291E-02	0.84	1.283E-02	0.26	1.281E-02	0.06
5	Moment of Inertia	True Value	Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True
		1.624E-03	1.247E-03	2.208E-03	35.94	1.804E-03	11.11	1.680E-03	3.45	1.642E-03	1.11	1.630E-03	0.37
6	Area	True Value	Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True
		1.28E-02	2.560E-02	1.205E-02	0.38	1.214E-02	2.66	1.294E-02	1.06	1.286E-02	0.50	1.284E-02	0.30
6	Moment of Inertia	True Value	Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True
		1.624E-03	1.247E-03	2.814E-03	73.31	1.548E-03	4.62	1.573E-03	3.17	1.606E-03	1.02	1.610E-03	0.35
7	Area	True Value	Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True
		1.28E-02	2.560E-02	1.205E-02	0.38	1.214E-02	2.66	1.294E-02	1.06	1.286E-02	0.50	1.284E-02	0.30
7	Moment of Inertia	True Value	Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True
		1.624E-03	1.247E-03	2.814E-03	73.31	1.548E-03	4.62	1.573E-03	3.17	1.606E-03	1.02	1.610E-03	0.35
8	K Horizontal	True Value	Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True
		4.50E+08	6.00E+08	4.78E+08	6.22	4.00E+08	2.22	4.53E+08	0.07	4.51E+08	0.22	4.50E+08	0.00
8	K Vertical	True Value	Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True
		1.290E+08	7.00E+08	-6.87E+10	11192.98	1.44E+08	2.67	1.30E+08	0.45	1.27E+08	0.58	1.29E+08	0.19
8	K Rotational	True Value	Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True
		1.130E+08	1.500E+08	6.80E+07	59.87	9.40E+07	16.81	1.10E+08	2.85	1.12E+08	0.88	1.13E+08	0.00
9	K Horizontal	True Value	Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True
		4.50E+08	6.00E+08	4.78E+08	6.22	4.00E+08	2.22	4.53E+08	0.07	4.51E+08	0.22	4.50E+08	0.00
9	K Vertical	True Value	Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True
		1.290E+08	7.00E+08	-6.87E+10	13192.18	1.38E+08	2.67	1.30E+08	0.85	1.27E+08	0.38	1.29E+08	0.19
9	K Rotational	True Value	Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True
		1.130E+08	1.500E+08	-6.80E+07	180.18	9.40E+07	16.81	1.10E+08	2.85	1.12E+08	0.88	1.13E+08	0.00

APPENDIX B – TWO DIMENSIONAL BRIDGE PARAMETER

ESTIMATION, CASE 2

Case	Member	Parameter	Iteration 1		Iteration 2		
2	8	K horizontal					
		True Value	Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True
		4.50E+08	6.000E+08	4.50E+08	0.00	4.50E+08	0.00
		K Vertical					
		True Value	Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True
		5.250E+08	7.000E+08	9.05E+08	72.38	5.25E+08	0.00
	K Rotational						
	True Value	Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True	
	1.130E+08	1.500E+09	1.13E+08	0.00	1.13E+08	0.00	
	9	K horizontal					
		True Value	Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True
		4.50E+08	6.000E+08	4.50E+08	0.00	4.50E+08	0.00
K Vertical							
True Value		Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True	
5.250E+08		7.000E+08	9.05E+08	72.38	5.25E+08	0.00	
K Rotational							
True Value	Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True		
1.130E+08	1.500E+09	1.13E+08	0.00	1.13E+08	0.00		

APPENDIX C – TWO DIMENSIONAL BRIDGE PARAMETER ESTIMATION, CASE 3

Case	Member	Parameter	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5	Iteration 6	Iteration 7
1	Area	True Value	1.28E-02	1.279E-02	1.279E-02	1.279E-02	1.279E-02	1.279E-02	1.279E-02
		Initial Value	2.580E-02	1.279E-02	1.279E-02	1.279E-02	1.279E-02	1.279E-02	1.279E-02
		Moment of Inertia	1.694E-03	1.694E-03	1.694E-03	1.694E-03	1.694E-03	1.694E-03	1.694E-03
2	Area	True Value	1.28E-02	1.279E-02	1.279E-02	1.279E-02	1.279E-02	1.279E-02	1.279E-02
		Initial Value	2.580E-02	1.279E-02	1.279E-02	1.279E-02	1.279E-02	1.279E-02	1.279E-02
		Moment of Inertia	1.694E-03	1.694E-03	1.694E-03	1.694E-03	1.694E-03	1.694E-03	1.694E-03
3	Area	True Value	1.28E-02	1.279E-02	1.279E-02	1.279E-02	1.279E-02	1.279E-02	1.279E-02
		Initial Value	2.580E-02	1.279E-02	1.279E-02	1.279E-02	1.279E-02	1.279E-02	1.279E-02
		Moment of Inertia	1.694E-03	1.694E-03	1.694E-03	1.694E-03	1.694E-03	1.694E-03	1.694E-03
4	Area	True Value	1.28E-02	1.279E-02	1.279E-02	1.279E-02	1.279E-02	1.279E-02	1.279E-02
		Initial Value	2.580E-02	1.279E-02	1.279E-02	1.279E-02	1.279E-02	1.279E-02	1.279E-02
		Moment of Inertia	1.694E-03	1.694E-03	1.694E-03	1.694E-03	1.694E-03	1.694E-03	1.694E-03
5	Area	True Value	1.28E-02	1.279E-02	1.279E-02	1.279E-02	1.279E-02	1.279E-02	1.279E-02
		Initial Value	2.580E-02	1.279E-02	1.279E-02	1.279E-02	1.279E-02	1.279E-02	1.279E-02
		Moment of Inertia	1.694E-03	1.694E-03	1.694E-03	1.694E-03	1.694E-03	1.694E-03	1.694E-03

APPENDIX D – TWO DIMENSIONAL BRIDGE PARAMETER

ESTIMATION, CASE 4

Case	Member	Parameter		Iteration		Iteration		Iteration	
4	6	Area		1		2		3	
		True Value	Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True
		1.28E-02	2.560E-02	1.186E-02	7.36	1.282E-02	0.18	1.282E-02	0.18
		Moment of Inertia		1		2		3	
		True Value	Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True
		1.624E-03	3.247E-03	2.911E-03	79.26	1.692E-03	4.19	1.627E-03	0.17
	7	Area		1		2		3	
		True Value	Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True
		1.28E-02	2.560E-02	1.186E-02	7.36	1.282E-02	0.18	1.282E-02	0.18
		Moment of Inertia		1		2		3	
		True Value	Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True
		1.624E-03	3.247E-03	2.911E-03	79.26	1.692E-03	4.19	1.627E-03	0.17
8	K horizontal		1		2		3		
	True Value	Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	
	4.50E+08	6.000E+08	4.50E+08	0.00	4.50E+08	0.00	4.50E+08	0.00	
	K Vertical		1		2		3		
	True Value	Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	
	5.250E+08	7.000E+08	5.25E+08	0.00	5.25E+08	0.00	5.25E+08	0.00	
9	K Rotational		1		2		3		
	True Value	Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	
	1.130E+08	1.500E+09	9.80E+07	13.27	1.02E+08	9.73	1.13E+08	0.00	
	K horizontal		1		2		3		
	True Value	Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	
	4.50E+08	6.000E+08	4.50E+08	0.00	4.50E+08	0.00	4.50E+08	0.00	
9	K Vertical		1		2		3		
	True Value	Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	
	5.250E+08	7.000E+08	5.25E+08	0.00	5.25E+08	0.00	5.25E+08	0.00	
	K Rotational		1		2		3		
	True Value	Initial Value	New Parameter	%Diff from True	New Parameter	%Diff from True	New Parameter	%Diff from True	
	1.130E+08	1.500E+09	9.80E+07	13.27	1.02E+08	9.73	1.13E+08	0.00	

APPENDIX E – BENT 57 FIELD DATA

Bent 57 Load Case 1, 50k On			Data Quality	
	Degrees		Index	Error
Theta1 Real	5.167E-05	9.018E-07 Radians	2	98.47%
Theta1 SAP		5.893E-05 Radians		
Theta2 Real	-1.240E-04	-2.163E-06 Radians	2	95.95%
Theta2 SAP		-5.342E-05 Radians		
Theta3 Real	-2.774E-04	-4.842E-06 Radians	2	98.20%
Theta3 SAP		-2.692E-04 Radians		
SWP Real		3.842E-02 in	1	34.35%
SWP SAP		2.860E-02 in		
		Average		
SG1	9.1522527 microstrains	✓ 12.8395383	1	
SG3	16.526824 microstrains		1	
Epsilon SAP	7.7720 microstrains			65.2%
SG2	17.269453 microstrains	✓ 26.6142539	1	
SG4	35.959055 microstrains		1	
Epsilon SAP	29.3360 microstrains			9.3%
SG5	41.356839 microstrains	✓ 44.0135952	1	
SG7	46.670352 microstrains		1	
Epsilon SAP	40.5770 microstrains			8.5%
SG6	-6.975736 microstrains	✓ -5.8225133	1	
SG8	-4.669291 microstrains		1	
Epsilon SAP	-3.4700 microstrains			67.8%
SG9	21.804848 microstrains	✓ 22.3960423	1	
SG10	22.987236 microstrains		1	
Epsilon SAP	22.7140 microstrains			1.4%
SG11	-22.18645 microstrains	✓ -21.46172	1	
SG12	-20.737 microstrains		1	
Epsilon SAP	-22.1600 microstrains			3.2%

Data Quality Index Key
 3 - No Data Recorded, Undue amounts of Noise
 2 - Acceptable but Suspect amount of noise
 1 - Good data Range


```

% Load Vector
Node      Fx  Fy  Fz      Mx  My  Mz  LCase#
Load=[  3    0  0  49.95    0  0  0    1
      5   19.48  0  0    0  0  0    2]; %enter zeros if no load
applied

%% Measured Displacement DOF
Node      Ux  Uy  Uz  Rx  Ry  Rz  LCase#
MeasuredDOF=[  2    1  0  1  0  1  0    1
              3    1  0  1  0  1  0    1
              4    1  0  1  0  1  0    2]; %enter 0 if no
measurement, 1 for a measurement
%% Measured Strain DOF
Element #  xbar  ybar  zbar  x  y  z
LCASE
Strain=[  3    82.25  18    0    126  0  1
        1    %Location 3
        3    82.25 -18    0    126  0  1
        1    %Location 3
        4    17.5   0     7    252  10  1
        1    %Location 2
        4    17.5   0    -7    252  10  1
        1    %Location 2
        5    79     0     7    252  10  1
        1    %Location 1
        5    79     0    -7    252  10  1
        1
        3    82.25 -18    0    126  0  1
        2
        5    79     0     7    252  10  1
        2
        5    79     0    -7    252  10  1
        2]; %Location 1

%% Simulated Data Flag
SimulatedStrainFlag=0; % Enter 1 if data is to be simulated, 0 if data
is from a text file
StrainMeasurementFileName={'C:\Mustang2\Measurement\StrainB57finalLC1.
.txt','C:\Mustang2\Measurement\StrainB57finalLC2.txt'}; %include full
path with filename

SimulatedData=1; % Enter 1 if data is to be simulated, 0 if data is
from a text file

if SimulatedData==0
    SIM=0;

MeasurementFileName={'C:\Mustang2\Measurement\DisplacementsBent57.txt'
,'C:\Mustang2\Measurement\DisplacementsBent57.txt'}; %include full
path with filename
else
    MeasurementFileName={0;0};

```

```

    %Element Number: Element Number Within Type, Node Number for
Springs
    %Type: 1=Frame , 2=Shell, 3=Spring
    %Parameter: Frames - 6=Moment of Intertia, 1=Area, 8=Mass FINISH

    %   Element #   Parameter   Type   SimulatedValue/Modifier
SIM=[   2       5           3       9.58e7
       4       5           3       9.58e7];
end

%% Unknown Element Vector

%Element Number: Element Number Within Type, Node Number for Springs
%Type: 1=Frame , 2=Shell, 3=Spring
%Parameter: Frames - 6=Moment of Intertia, 1=Area, 8=Mass FINISH

%   Element #   Type   Parameter
UNK=[   2       3       5
       4       3       5];

%% List All Springs in Model
%   Joint#           Stiffness
SpringList = [   2   0   0   0   0   9e7   0
                4   0   0   0   0   9e7   0];

%ErrorFn Convergence           SS
ConvergenceLimits= [   1e-12 ];

%% Directory Information
SAPModelFileName='Bent57twoLC';
SAPpath='C:\Program Files (x86)\Computers and Structures\SAP2000 14';
SAPmodelPath='C:\Mustang2\SAP12';
LoadCase={'LC1', 'LC2'};%enter loadcase names in order 1,2,... to
agree with Load and Measured Lcase#, must be exact capitalization
GeneralFunction=[ 0 1 0
                  0 1 0];

%% Call to Mustang
PlateThickness=0;
Mustang(Load,MeasuredDOF,MeasurementFileName,UNK, SpringList,Convergen
eLimits,SAPModelFileName,SAPpath,SAPmodelPath,LoadCase,GeneralFunction
,SIM,Strain,SimulatedStrainFlag,StrainMeasurementFileName,PlateThickne
ss)

diary off

```