VEHICLE-BORNE AUTONOMOUS RAILROAD BRIDGE IMPAIRMENT DETECTION SYSTEMS

A Dissertation

by

AUSTIN J. ALLARD

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Gary T. Fry
Committee Members,	Luciana R. Barroso
	Peter B. Keating
	Alan B. Palazzolo
Head of Department,	Robin Autenrieth

August 2017

Major Subject: Civil Engineering

ABSTRACT

Timber railroad bridges have been exposed to increasingly large axle loadings accompanied by a steady increase in the amount of railcar traffic over the past 50 years. In addition to mechanical loading, there exists a number of environmental conditions that can deteriorate the timber in these bridges: e.g. insects and decay organisms. The primary inspection method is conducted on a bridge by bridge basis and involves visually examining individual components of the bridge and assessing the damage.

This research examines an automated impairment detection system positioned on a railcar capable of traversing multiple bridges along a track to aid in determining critical bridges that need to be inspected. The technology and techniques presented in this dissertation are envisioned as a potential enhancement to current visual evaluation methods by providing system-wide trending data for human decision makers.

The objective of the research is to develop technology that will autonomously detect structural impairments in timber railroad bridges using data gathered from rail vehicles that cross the bridges. This was accomplished by recording the behavior of a bridge and the motion of a railcar passing over bridge spans. Artificial neural networks, a type of pattern recognition technology, were used to determine relationships between the bridge and vehicle behaviors. The results of a finite element analysis were utilized to train the neural networks to recognize the patterns associated with the bridge and railcar motions. Five different impairment conditions, or simulated damage scenarios, were developed for the training process. This allowed the networks to recognize the patterns

correlating the railcar and bridge data streams. Once the artificial neural networks were successfully trained, new vehicle motions from a field test were presented to the network and the corresponding bridge behavior was predicted. The neural networks were accurate in predicting the maximum chord deflection to within 0.1 inches in 72% tested chords with improved accuracy at faster speeds.

DEDICATION

For my family

ACKNOWLEDGMENTS

I would like to thank my Committee Chair, Dr. Gary Fry, for his advice and recommendations throughout the course of this research. I would also like to thank my committee members, Dr. Luciana Barroso, Dr. Peter Keating, and Dr. Alan Palazzolo, for their support while I was working on this project.

I would like to give a thank you to the research group that has aided me throughout the course of this research. Maysam Kiani, Kimberly Lyons, Johanna Posey, Lisa Rachal, Cynthia Ryan, Sean Smith, and Maryam Tavakoli were all present at the bridge site to aid in collecting the data. They were also involved in the pre-test phase that involved hours of calibrating string potentiometers and soldering connections. A special thank you is given to Amanda Pelton and Johanna Posey for editing and reviewing this dissertation. An additional thank you is offered to Lisa Rachal for illustrating a number of figures present throughout this research. I would like to thank Akram Abu-Odeh, David Allen, Brian Hessler, Nataly Delafuente, and Melissa Martinez, for their time and support.

I would like to give a special thank you to the members of Union Pacific railroad, specifically Eric Epps, for providing a work train for testing and arranging the track time required to safely perform the experiment at Bridge 17.14.

I will be forever grateful for the love and support offered by my family throughout the course of my research. I offer a very special thank you to my wife, Kayla, for giving me the strength to soldier on through the difficult path. Her sacrifices have allowed me to follow my aspirations, so one day, I hope I am able to repay her devotion.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

The research completed in this dissertation was aided by a committee consisting of Dr. Gary Fry, Dr. Luciana Barroso, and Dr. Peter Keating of the Department of Civil Engineering and Dr. Alan Palazzolo of the Department of Mechanical Engineering at Texas A&M University.

The experimental results were acquired with a research group consisting of Maysam Kiani, Kimberly Lyons, Johanna Posey, Lisa Rachal, Cynthia Ryan, Sean Smith, and Maryam Tavakoli. A number of figures depicting the work train and test bridge were illustrated by Lisa Rachal.

The remainder of the work presented in this dissertation was completed by the student independently.

Funding Sources

This work was made possible in part by funding from the USDOT UTC Program under agency ID DTRT13-G-UTC59.

This research was also aided by the financial support of a Richard M. Gehle Memorial Scholarship in Civil Engineering, Jaster-Quintanilla in Historical Preservation Fellowship, and Zachry Fellowship in Civil Engineering.

NOMENCLATURE

a_k	Vertical acceleration time history
b	Cross sectional width
b_k	Bias
b(t)	Step function
c	Distance from the neutral axis to the extreme fiber of the beam
$\frac{d^2y}{dx^2}$	Curvature of beam's neutral surface
δ_l	Average midspan deflection
δ_{net}	Net deflection
δ_{total}	Total deflection
DR	Deflection ratio
ξ	Location of the load from the initial support
ξ_1	Distance from point load to end support
E	Modulus of elasticity
f	Acceleration time history
f_n	Acceleration value at time n
F, $\mathcal{F}$	Fourier transform
$\mathcal{F}^{-1}$	Inverse Fourier transform
h	Cross sectional height
i_i	Indicates East or West chord
in_m	input node
I	Moment of inertia
j_j	Indicates wheel path
k	Window size
k	Number of neurons
l	Number of outputs
L	Span length

m	Number of inputs
M(x)	Bending moment at a location, x, from initial support
n	Current sample being considered
N	Number of time samples
N_x	Number of data points
$\sigma_m(x)$	Maximum flexural stress at a distance, x, from initial support
out_l	Output from neural network system
ρ	Mass density
π	Mathematical constant
P	Load
r^2	Coefficient of determination
s	Current frequency being considered
t	Time
ts	Time step
u_k	Linear combiner output
v_1	Work train velocity
v_k	Transfer function input
$\varphi(\ )$	Transfer function
w_{kj}	Synaptic weights
x	Distance from initial support
X	Value on the X-axis
X_i	Starting data point
y	Deflection
y_k	Output signal from neuron

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

CHAPTER I

INTRODUCTION

I.1 Introduction

Timber bridges constitute a significant portion of the total number of railroad bridges in the United States, accounting for approximately 20 percent of railroad bridges in some northern states and can range from 20 to 40 percent in southern states (Radford et al., 2000). There are nearly 400 miles of timber bridge rail currently used in the United States, and make up roughly 24% of all bridge rail (FRA, 2008). Figure I-1 shows a typical open deck timber trestle railway bridge consisting of bents, stringers, cross ties, and rails. The rails are in direct contact with railcar wheels and are supported by cross ties. The cross ties distribute the load from the rail to two chords of stringers, with each chord having four plies. The stringers span across two bents, with half of the stringers in the chords being continuous over a specific bent. A bent consists of a bent cap, piles, and pile bracing, and is responsible for distributing the load from the stringers to the ground.

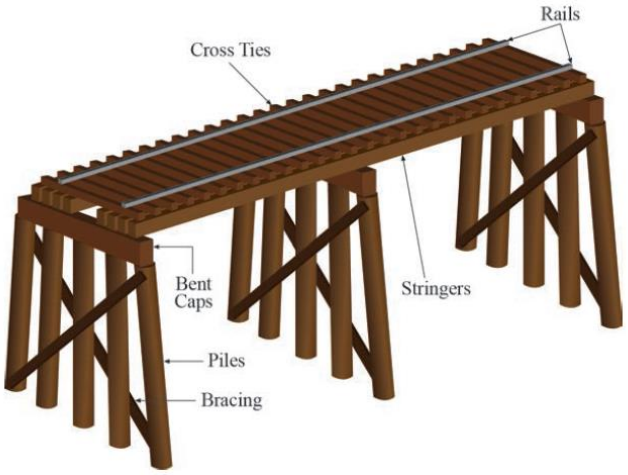

Figure I-1: Open Deck Timber Trestle Railway Bridge

Timber trestle railroad bridges need to be inspected frequently to ensure they can be operated safely. The reason for frequent inspections is a majority of timber trestles have been in service for a long period of time while being exposed to increased axle loads and subjected to various environmental conditions. Prior to 1970, railroads were commonly loaded with rail vehicles with up to 200 kip gross vehicle weight (GVW). Axle loads have steadily increased and now allow 286 kip GVW for nearly all coal traffic (Martland, 2013). On top of these load increases, the number of rail intermodal freight traffic has multiplied over the past 35 years. The yearly volume of containers and trailers shipped on U.S. rail has risen from 3.1 million in 1980 to 13.7 million in 2015 (AAR, 2016). This corresponds to an increase of roughly 3 million containers and trailers shipped every decade. Frequent applications of excessive loads can cause physical damage in timber bridges, such as split stringers and crushed bent caps.

In addition to the increased loading, environmental factors also have a negative impact on the performance of timber bridges. Insect infestations and wood rotting are a few of nature's hazards that can be forced upon the structure. A combination of harmful environmental conditions and increased axle loadings over a long time period has a detrimental effect on the structural behavior of a timber railway bridge.

I.2 **Objective**

The traditional method of rating timber bridges involves manual inspections performed by trained professionals. These qualitative visual inspections can become costly and timely due to the number of bridges that need to be monitored and the availability of the crew required for inspection. Older timber bridges need to be inspected frequently to ensure impairments do not escalate into the failure of the structure. To prevent a structural failure from occurring, all significant impairments must be detected and repaired as quickly as possible. The objective of this research is to develop a technique that is able to enhance these visual inspection methods through the implementation of technology that can detect structural impairments using data gathered from rail vehicles traversing timber bridges. This technology will not only reduce the resources required to monitor timber bridges, but will also provide quantitative data regarding the health of the structure.

I.3 **Methodology**

To accomplish the objective, a structural impairment detection system (SIDS) was created. A SIDS monitors specific characteristics of a structure and relates the values to structural impairments. The proposed SIDS uses pattern recognition capabilities of artificial neural networks to determine damage in a timber railway bridge by analyzing data from a railcar crossing the bridge. Neural networks are a nonlinear parallel processing system that use a method of learning to recognize patterns and trends composing a training data set (Haykin, 1999). The relationships learned from training can then be used to analyze similar data.

The success of the structural impairment detection system is highly dependent on the efficiency of the system of neural networks. The networks must correctly use the input sensor data to determine the operating condition (damage present) of the timber bridges. A number of steps were followed to ensure the effectiveness of the SIDS:

1. A finite element analysis was performed to determine the behavior of an undamaged timber railway bridge.
2. A number of structural impairments were imposed during the finite element analysis, and the change in behavior of the system was recorded.
3. A system of sensors that were susceptible to changes in bridge behavior were mounted on a railcar. An additional set of sensors were attached to the bridge to compare with the SIDS output.

4. The competitive arrays of neural networks were trained to correlate patterns and trends in the bridge and railcar finite element analysis data streams to corresponding simulated impairments.
5. The competitive array of neural networks were tested with new data sets from the test railcar and the results were compared to the corresponding recorded bridge behavior to determine the efficiency of the structural impairment detection system.

The developed SIDS uses a combination of finite element modeling, experimental testing, and artificial neural networks to determine the structural behavior of a timber railway bridge from a railcar data stream. This is accomplished with a system of sensors specifically designed to detect changes in the system's performance relating to bridge impairments. This SIDS does not rely on a single sensor to determine an impairment location, but rather, it uses a number of sensors analyzed in unison to provide the structure's overall health.

I.4 **Timber Railway Bridge Background**

The proposed research presents a new method to monitor the structural health of a timber railway bridge, but there are a number of other number of other methods that have been developed in the past years to determine the health of timber bridges in service for prolonged periods. Emerson, et al. (1998) used ultrasonic waves to inspect and determine decay locations in timber members. Morison, et al. (2002) used a form of impact testing while measuring the frequency response to locate impairments in a bridge. Peterson and

Gutkowski (1997) used a combination of these two techniques. Global dynamic excitation was used to locate critical members in need of testing, and ultrasonic inspection evaluated the critical members. Ritter (1990) and Wipf, et al. (2000) used accelerometers and displacement measurements to monitor the response of a timber bridge subjected to live loading. Babcock, et al. (2006) used finite element analysis to quantify the effects of static loading on a laboratory sized timber bridge.

All of these techniques and visual inspections require an experienced individual or team of individuals with adequate monitoring equipment to be present at the bridge site. Timber railway bridges are still commonly used form of freight and passenger transportation, and require an excessive amount of manpower to monitor to ensure the safety of those who travel on them. The interest in an inspection device located on a railcar traveling over a number of bridges has risen in the past years as technology has continued to improve.

Rakoczy, et al. (2015/2016) has done extensive research in recent years in developing methods to monitor timber bridges with various onboard devices. The research was conducted at the Bridge Deflection Test Facility at the Transportation Technology Center near Pueblo, CO. The bridge tested was able to be modified to mimic various levels of damage located in different spans. Research included using a NUCARS dynamic computer simulation bridge model to simulate the response of a flexible bridge subjected to live train loading. Another study used freight cars equipped with accelerometers and spring displacements, track geometry vehicles, and track deflection measurement vehicles as onboard impairment detection systems. Additional testing was

conducted with T-18 test vehicle and a TUVX 001 loaded hopper modified with an MRail system capable of determining rail deflections. Testing also involved implementing track geometry testing attached to a passenger car. The results of these studies showed promising results for using onboard measurements to locate impairments in a timber bridge at a testing facility.

One disadvantage of this research is it requires a sophisticated set of technology to determine bridge impairments. Additionally, testing was not conducted on a timber bridge in service that had been exposed to a number of years of environmental hazards and continuous train loading. In his Master's thesis, Orsak (2012) conducted an analytical analysis of a bridge subjected to train loading. The results showed 90% accuracy in determining the theoretical impairment condition imposed on the structure. Story and Fry (2014) outlined a method for determining damage present in members of a heel trunnion bascule bridge in service. The process used a combination of finite element analysis, experimental measurements, and artificial neural networks to determine impairments in the bridge. A similar technique was used to determine damage present in a timber railway bridge in this research.

I.5 Impairments in Timber Railway Bridges

Throughout its lifetime, a typical timber railway bridge is subjected to occasional health monitoring that leads to repairing individual members or replacing segments of the bridge. Upgrading parts or segments of a bridge can lead to structures that remain in service continuously for over 100 years (Peterson & Gutkowski, 1997). Long service life

and a number of physical and environmental factors negatively influence the bridge performance. Frequent crossing of trains with increasingly large axle weight subjects the bridges to strenuous fatigue loading. Additionally, most climates can cause the wood in the stringers, cross ties, and bents to swell and possibly rot. Other environmental hazards include insect infestations and attacks by various fungi that can degrade structural members. Evaluating impairments in the timber bridges requires considering how each of these factures uniquely affects the bridge performance.

A physical factor that can influence bridge performance is frequent train fatigue loading that leads to cracks developing within the timber stringers. Over time, the cracks propagate to the surface and spread longitudinally along the length of the stringer. Once the crack reaches the surface and spreads, the stringer is said to have split. A split stringer has a significantly lower flexural stress than an unimpaired member.

Figure I-2 shows the simplified effects of a moving train load on a timber stringer. The boundary conditions are approximated as simply supported with a pin or roller attached to either end. The axle load from the train, P, is approximated as a point load located at a distance ξ from the first bent. ξ increases as the train traverses the bridge. This example neglects the effects of the preceding and following axle loads.

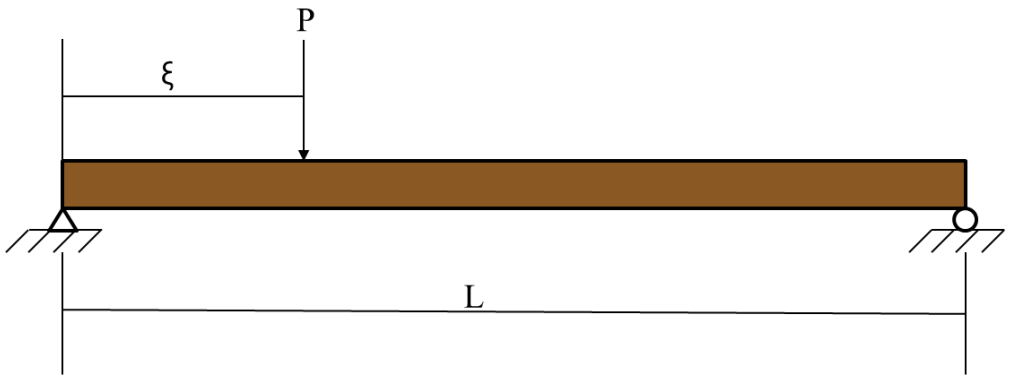

Figure I-2: Simply Supported Beam with a Moving Load

Figure I-3 shows the bending moment, M, and flexural stress distribution at a distance, x, from the first support for a beam loaded as shown in Figure I-2. This loading places the top surface of the beam in compression and the lower surface in tension. Equation I-1 provides a relationship between the bending moment and the maximum flexural stress, σ_m, in the beam. The calculation requires known cross sectional properties such as the distance from the neutral axis to the extreme fiber of the beam, c, and the moment of inertia with respect to the horizontal axis, I.

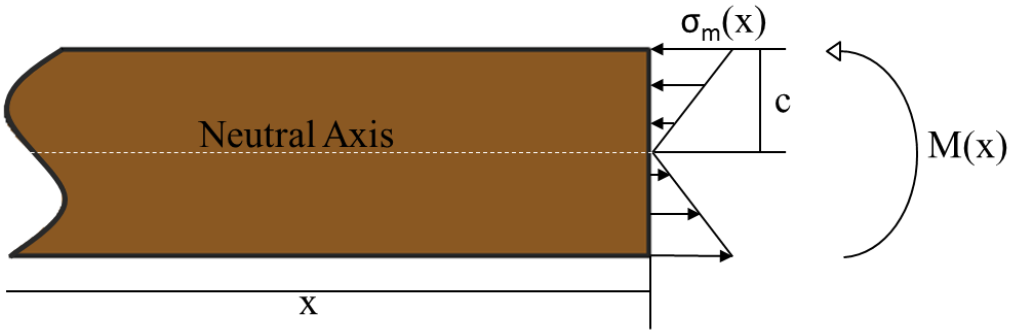

Figure I-3: Bending Moment and Flexural Stress in a Beam

$$\sigma_m(x) = \frac{M(x)c}{I}$$ Equation I-1

A split beam can be approximated by cutting the original beam along the neutral surface to obtain two separate beams as seen in Figure I-4. The height of each beam is now half of its original value, and causes the moment of inertia in each beam to become 1/8th of its initial amount. The total moment resisted by each beam and their respective distances from the neutral axis to the extreme fiber are halved. The flexural stress experienced by a split stringer then becomes two times larger than an unimpaired stringer.

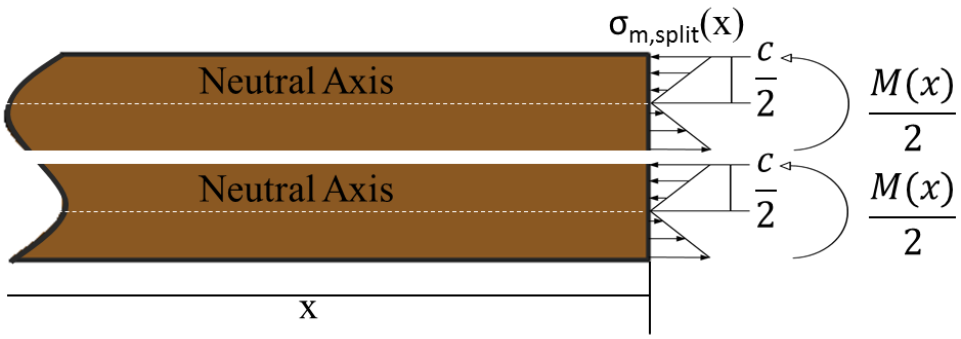

Figure I-4: Bending Moment and Flexural Stress in a Split Beam

In addition to the increased flexural stress in a split stringer, the moment of inertia available in each beam has been reduced by an 1/8th, or a total of 1/4th for the two beams acting together to resist the moment. Overall, this causes a 75% theoretical reduction in the stringers ability to resist the train loads being applied to the bridge. This value is larger than the 62.6% reduction in flexural rigidity measured during experimental testing (Orsak

& Story, 2013). The testing involved subjecting the stringer to four point fatigue loading and measuring the flexural rigidity of a timber stringer before and after the member splits. The difference between the two values comes from the friction in the experimental split beam that was not accounted for in the theoretical analysis. Additionally, the experimental beam may not completely separate into two distinct beams that were assumed during the numerical investigation.

Physical damage experienced by stringers from repeated loading is a common cause of failure, but there are a number of environmental factors that also need consideration. Insect infestations and rotting can create problems with the structures ability to resist load. Humar, et al. (2006) exposed timber samples to various types of wood rot fungi. The research showed that the modulus or elasticity of the untreated wood was reduced by 76% after being exposed to the rotting fungi over eight weeks. The treated samples showed more resistance to effects of the fungi, but there was still a significant reduction in the Young's Modulus. Rotting is one of the top causes of replacements for timber bridges because it will ultimately occur in wood subjected to various weather effects (Forsling et al., 2012). Therefore, it is important to consider environmental hazards when analyzing impairments in timber specimen.

Experimental research has shown split stringers can reduce the moment of inertia of a timber beam by 62.6% and rotting can decrease the modulus of elasticity by 76%. If both of these were to occur on a stringer in a worst case scenario, the flexural rigidity, EI, of the beam would be diminished by 91% of its original value. Beam theory determines the effect of the reduced flexural rigidity on a timber stringer's performance by examining

the governing differential equation for an elastic curve shown in Equation I-2. This is a second order linear differential equation representing the curvature of a neutral surface.

$$\frac{d^2y}{dx^2} = \frac{M(x)}{EI} \qquad \text{Equation I-2}$$

Beam theory can be used with the simply supported beam with a moving load shown in Figure I-2 to determine the equation for the bending moment shown in Equation I-3. M(x) is the bending moment of the beam at a distance, x, from the first support. L is the length of the span and ξ is the distance of the moving load, P, from the first bent. The pointed brackets represent a singularity function where the output is zero if the value inside is negative.

$$M(x) = \frac{P(L-\xi)x}{L} - P\langle x-\xi\rangle \qquad \text{Equation I-3}$$

Equation I-3 can be inserted into Equation I-2 to obtain the equation of the elastic curve. The boundary conditions y(0)=0 and y(L)=0 can be applied to find the values of the integration constants. The deflection of the beam at the midspan (L/2) for the moving load is provided in Equation I-4.

$$y(L/2) = \frac{P}{6EI}\left(\frac{1}{2}(L-\xi)^3 - \langle\frac{L}{2}-\xi\rangle^3 - \frac{3}{8}L^2(L-\xi)\right) \qquad \text{Equation I-4}$$

Equation I-4 shows the midspan deflection of the beam is dependent on the load, span length, load location, and flexural rigidity. The flexural rigidity is inversely proportional to the deflection. If the EI value were to decrease by 50%, the deflection for the load located in the same spot would theoretically double. In a timber railway bridge, there are multiple members responsible for resisting load, so one damaged stringer wouldn't have the same effect on the structure as seen in this theoretical example. However, increased stringer deflections are good indication that there is damage present in the bridge.

I.6 Theoretical Axle Acceleration

Timber railway bridge midspan deflections are able to indicate impairments that may be present in a system. If this value could be recorded over time, then bridge impairments could be detected in a timely manner. However, implementing timber bridge deflection devices in every bridge in the U.S. would be a costly and time consuming expedition. An ideal method would be to relate a measured deflection of the railcar to the midspan bridge deflection. However, measuring the deflections of a moving object requires expensive equipment and specially trained personnel to operate it, so a simpler device is needed for this research. There is a mathematical relationship between the position of an object with respect to time and its acceleration. Accelerometers are relatively inexpensive devices that are used to measure vibrations and accelerations of moving objects.

The theoretical acceleration of a moving object on a bridge can be measured by examining the deflection of a beam. Consider a simply supported beam with a moving point load, P, shown in Figure I-5. The moving load is once again the axle load of a train approximated by a point load on the beam. The previous section described how to obtain the equation for midspan deflection for a moving load. The same process can be used to find the deflection for any point, x, along the span and is shown in Equation I-5.

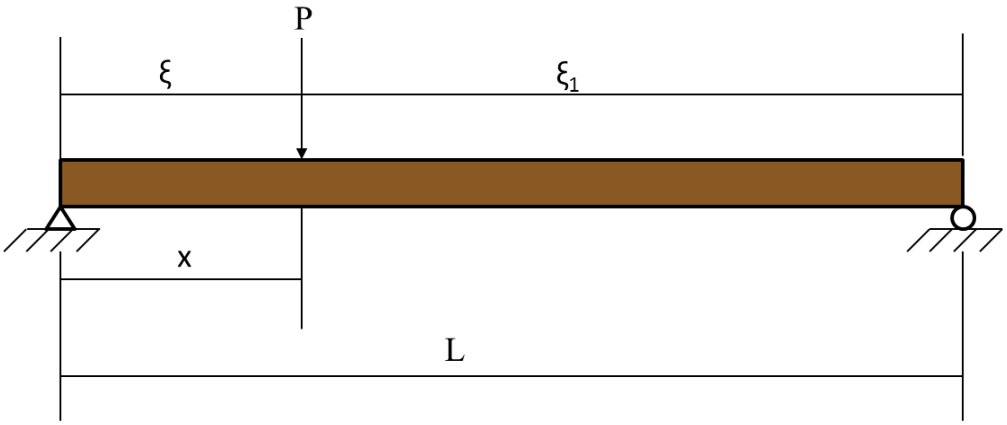

Figure I-5: Simply Supported Beam with Moving Load P

$$\delta(x) = \frac{-P\xi_1 x}{6LEI}\left(L^2 - x^2 - {\xi_1}^2\right) \qquad \text{Equation I-5}$$

Determining the acceleration at the point load, P, require calculated the deflection at the same point. Therefore, ξ is set to equal x, making ξ_1 the span length, L, minus x. Setting x equal to the product of velocity, v_0, and time, t, gives the equation as a function of time. The velocity is assumed to have a constant horizontal value for this calculation.

After substituting in these values, the deflection as a function of time is given in Equation I-6.

$$\delta(t) = \frac{-P{v_0}^2}{3LEI}(L^2t^2 - 2Lv_0t^3 + {v_0}^2t^4) \qquad \text{Equation I-6}$$

The second derivative of Equation I-6 gives the vertical acceleration values as a function of time shown in Equation I-7. The equation can be transformed into a function of position by substituting time with x divided by velocity. The acceleration as a function of the distance from the initial supports is provided in Equation I-8.

$$a(t) = \frac{-P{v_0}^2}{3LEI}\left(2L^2 - 12Lv_0t + 12{v_0}^2t^2\right) \qquad \text{Equation I-7}$$

$$a(x) = \frac{-2P{v_0}^2}{3LEI}(L^2 - 6Lx + 6x^2) \qquad \text{Equation I-8}$$

The above equations show the location of the point load on the beam along with a number of parameters influence the theoretical acceleration. These parameters include the magnitude of the load, the beam span length, and the flexural rigidity of the system. Since the velocity of the point load is squared in the equations, it has a significant impact on the final acceleration value.

The acceleration measured by an axle on a moving railcar along with midspan bridge deflections was used to implement the SIDS. A variety of speeds were considered

to investigate the effect of speed on the impairment detection system. Additionally, various degrees of structural impairment were simulated by modeling the bridge in a computer simulation. These factors were used in conjunction with a system of neural networks to identify patterns relating wheel path accelerations to bridge deflections. The efficiency of the networks was a crucial factor in a successful SIDS implementation.

The following chapters of this dissertation provide further details about the development and testing of the SIDS. Chapter 2 details the design and training of the system of artificial neural networks. Chapter 3 provides the experimental testing procedure, while Chapter 4 defines the finite element analysis computer simulations. Chapter 5 discusses the results of the artificial neural networks and Chapter 6 provides the conclusions and recommendations for future work.

CHAPTER II

ARTIFICIAL NEURAL NETWORKS

II.1 Artificial Neural Networks

The objective of this research is to develop technology that can estimate the condition of a bridge using data gathered from a vehicle that crossed the bridge. Developing a technology capable of predicting the condition of a bridge using data from a vehicle crossing the bridge requires defining a relationship between the vehicle's behavior and the bridge's response. This research employs a competitive array of neural networks to determine a connection between the vertical wheel path accelerations of a work train and the midspan deflections of the bridge it crosses.

Artificial neural networks consist of a system of virtual neurons that are able to imitate the cognitive abilities of living brain cells (Hagan et al., 1996). Computing traditionally involves executing a specified series of programmed instructions to solve a problem. Alternatively, artificial neural networks use information learned from examples to solve problems (Bishop, 1994). During a training phase, the networks are given a set of input and output data. These networks use an iterative process to determine the relationship between the data. When presented with new input data, the neural network can use the learned information to estimate the output. Since neural networks learn from examples, they are able to establish relationships between data when little information is known about the system.

Analytical models can predict the behavior of complex, real-world system. However, they require significant information about the system and intricate calculations. In the case of a bridge, natural and man-made imperfection make it difficult to accurately determine material properties, component wear, and member connections. Additionally, complex interactions within the system make quantifying the dynamic loading of the bridge difficult. Because they rely on trends rather than exact values, neural networks can analyze real-world, analytical problems with the absence of precise information (Bishop, 1994). .

II.2 History of Artificial Neural Networks

Artificial neural networks are a type pattern recognition technology capable of correlating a set of input data to output data. They have evolved throughout history to their current state which has a number of practical applications. In the late nineteenth century, researchers began investigating the use of a man-made neural system to solve problems (Priddy & Keller, 2005). Early work involved determining how neurons in the human brain process information. The idea was further explored when Hebb (1949) developed a psychological learning rule to explain how neurons in the brain react when exposed to a stimulus. Farley and Clark (1954) constructed the first digital artificial neural network with randomly organized neurons designed to perform a simple task.

Rosenblatt (1958) expanded upon the theories further by developing a hypothetical nervous system, referred to as a perceptron, capable of classifying patterns. The research concluded that the efficiency of the perceptron could be improved by increasing the

number of stimuli that the system has the opportunity to learn from. The performance was also enhanced by increasing the number of neurons used in the process.

Werbos (1974) improved existing techniques by developing a new learning rule where the output from the network could be propagated back through the system to improve the results. When the propagation occurs, the synaptic weights of the system are modified to produce a new output. The process is iterated to minimize the error associated to the neural network training. This iteration allowed the data obtained from a dynamic model to have a better fit with real-world samples. This method, now referred to back propagation, is one the of most commonly used artificial neural network algorithms (Priddy & Keller, 2005).

These researchers set the baseline for the theory and motivation of present-day artificial neural network systems. The ideas and theories have been continually modified over the years to construct the advanced models that are currently used today.

II.3 Training Artificial Neural Networks

For the artificial neural network to begin learning the patterns in the data, they must first analyze training input and training output data. The training input is the original data stream presented to the network. The training output is a known value that establishes a target for the network to generate based on the input.

The training input and training output are used during a training process to help the neural networks operate efficiently. This involves providing the arrays with a known input and output combination that allows them the best opportunity to learn the patterns

comprising the data. During the training process, the input training vectors are introduced into the neural network where they are multiplied by internal weights and then inserted into a transfer function, and produces a simulated output. Next, the error between this output and the training output is calculated. If the error is above a given tolerance, the internal weights are adjusted, and the process is iterated until the error is deemed acceptable or other stopping criteria are met. The process, shown in Figure II-1, is called back propagation. The training is considered successful if the final output produced by the network is similar to the target output from the bridge.

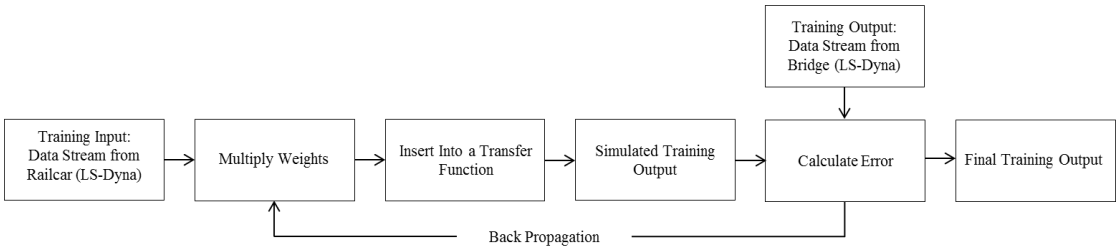

Figure II-1: Artificial Neural Network Training Process by Back Propagation

The vertical wheel path accelerations obtained from the finite element analysis were chosen as the training input for the networks. A wheel path refers to the motion of a particular axle as the train crosses the bridge. The training output is the average chord midspan bridge deflection data obtained from a finite element analysis. The finite element analysis procedure and results are provided in Chapter 4.

The training input and training output vectors, shown in Figure II-2, include a number of test parameters. In the input training vector, the v_1 value provides the train velocity speed. The i_i and j_j parameters indicate the position of the measured acceleration.

The accelerations measured over the east chord of the bridge are labeled i_1 and i_2 is the west chord. The wheel paths of the recorded acceleration are referred to as j_1 through j_4. The vertical acceleration time histories are labeled as a_1 through a_8. For the training output vector, δ_1 and δ_2 refer to the average midspan deflections over the east or west chords of stringers, respectively.

Training Input	Training Output
v_1	i_1
i_1	δ_1
j_1	i_2
a_1	δ_2
i_1	
j_2	
a_2	
.	
.	
.	
i_2	
j_4	
a_8	

Figure II-2: Training Input and Training Output Vectors

II.4 **Artificial Neural Network Architecture**

Artificial neural networks are complex mathematical systems that can be used for a variety of different purposes. Understanding how a neural network system operates requires examining the architecture within the network. The architecture consists of a number of hidden layers. Each of these layers has a number of neurons relating to a

transfer function. Figure II-3 shows an example of an *m*-2-1 neural network system. The system is referred to as *m*-2-1 because it has *m* number of inputs, 2 neurons in the first and only hidden layer, and 1 output node. This is just one example of a neural network system. Artificial neural networks can have multiple hidden layers, each with a different number of neurons.

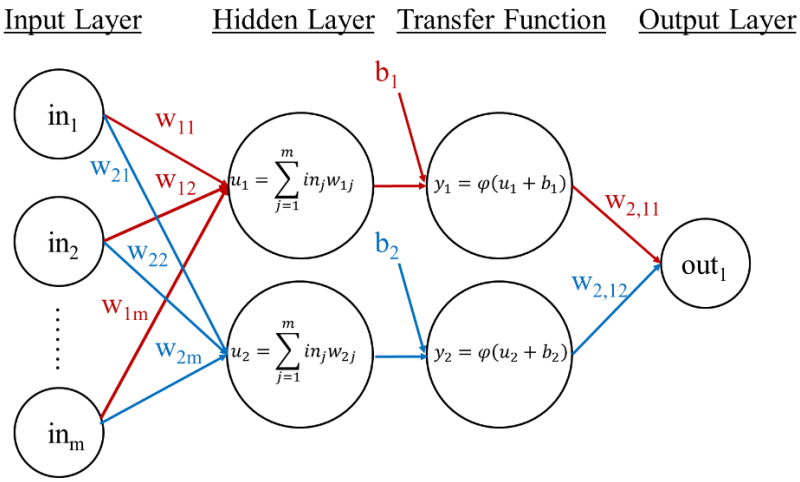

Figure II-3: *m*-2-1 Neural Network System

Figure II-3 shows the basic operating procedure of a neural network system. The *m* number of input nodes inserted into the neural system are labeled as in_m. Every input is multiplied by a unique synaptic weight, w_{km}, before being summed in each of the two neurons in the hidden layer. The linear combiner outputs, u_k, are added to a bias, b_k, before they are inserted into a transfer or activation function, $\varphi(\ )$. The bias can be thought of as an additional synaptic weight that is multiplied by an extra hidden input node with a value of 1. The bias applies an affine transformation of the linear combiner output before it is

inserted into the transfer function. The output signal from each neuron, y_k, is multiplied by another set of synaptic weight and summed to produce the output from the neural network system, out_l.

This neural network system is used in the training procedure explained in the previous section. The output from the neural network system would be compared to the training output. If the error between the two data sets was determined to be too high, the system would use back propagation to adjust the synaptic weights and continue to repeat the procedure to minimize the error.

Defining the number of hidden layers and neurons is one of the key features in constructing the neural network architecture. Another is determining which transfer function would provide the best output. The three transfer functions typically used in neural network architectures are the Threshold or Heaviside Function, a Piecewise-Linear Function, and a Sigmoid Function. The input for the transfer function, v_k, is the sum of the linear combiner output and the bias as seen in Equation II-1.

$$v_k = u_k + b_k \qquad \text{Equation II-1}$$

The limits of the Threshold Function are shown in Equation II-2. The output of the neuron is equal to 1 if the input is greater than or equal to zero, and zero for negative valued inputs. This is known as the all-or-nothing property and is used in the McCulloch-Pitts Model (Haykin, 1999). A visual representation of the Threshold Function is provided in Figure II-4.

$$\varphi(v_k) = \begin{cases} 1, & v_k \geq 0 \\ 0, & v_k < 0 \end{cases}$$ Equation II-2

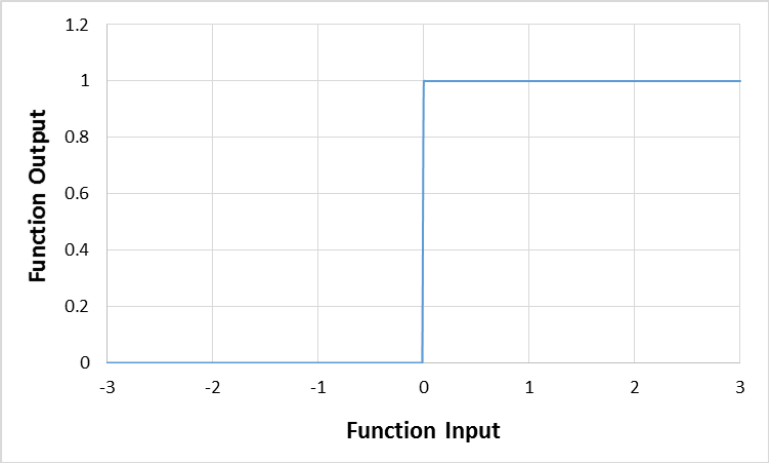

Figure II-4: Threshold Function

The Piecewise-Linear Function, expressed in Equation II-3., has an output of 1 for an input greater than ½, and a value of zero for inputs less than – ½. The output also increases linearly from zero to 1 for values between ± ½. The Piecewise-Linear Function is shown visually in Figure II-5. The function represents a common case where the amplification factor in the linear region is set equal to 1. In other specialized cases, the amplification factor can be adjusted. For example, a very large amplification factor causes the range in the linear region of the inputs to decrease while the output range remains unchanged, resulting in an output similar to a Heaviside Function.

$$\varphi(v_k) = \begin{cases} 1, & v_k \geq \frac{1}{2} \\ v + \frac{1}{2}, & \frac{1}{2} > v_k > -\frac{1}{2} \\ 0, & v_k \leq -\frac{1}{2} \end{cases}$$ Equation II-3

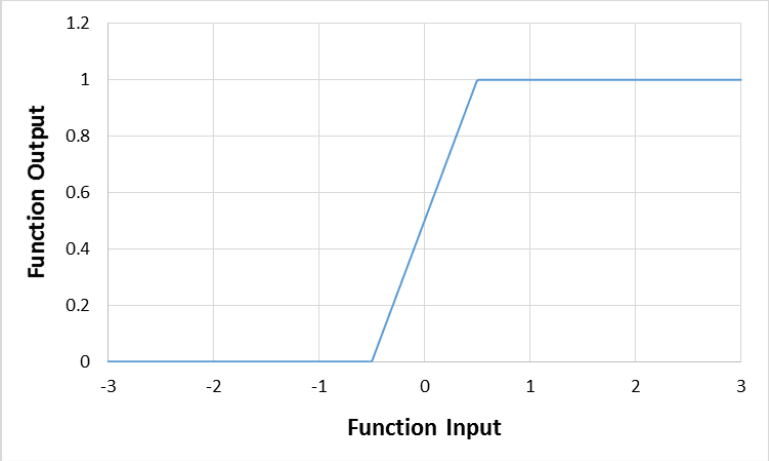

Figure II-5: Piecewise-Linear Function

The most commonly used activation function used in neural systems is the Sigmoid Function. A Sigmoid Function can be represented by the logistic function provided in Equation II-4. This function shows an s-shaped behavior that increases nonlinearly for values ranging from zero to 1 and is plotted in Figure II-6. This function can be modified by the addition of a slope parameter placed in the exponent. Increasing the slope power towards infinity causes the plot to resemble a Threshold Function. The slope of the line where the input is equal to zero has a value of the slope parameter divided by 4.

$$\varphi(v_k) = \frac{1}{1 + \exp(-v_k)}$$ Equation II-4

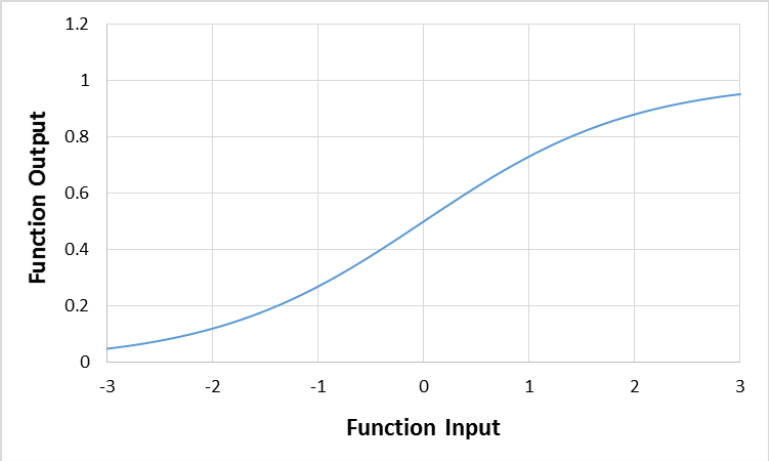

Figure II-6: Sigmoid Function

All of these functions have outputs ranging from zero to 1. It may also be necessary for the outputs to range from -1 to 1. This is accomplished in the Threshold functions by setting negative values to -1 and positive values to 1. Adjusting the Piecewise-Linear Function requires modifying the limits of the linear range from -1 to 1. A hyperbolic tangent function is used as the Sigmoid Function if output limits between -1 and 1 are desired.

II.5 Competitive Array of Neural Networks

The challenge with using artificial neural networks arises when determining the network architecture that will produce desirable results. Neural network architecture consists of a number of hidden layers, and each layer has a specified number of neurons.

It also includes transfer functions and error criteria, and each of these parameters can affect the efficiency of the system. The effectiveness of artificial neural networks can be improved by having several networks compete for the chance to train on the input data stream (Hyland & Davis, 2002). A competitive array of neural networks consists of a number of individual neural networks with unique architectures. The training input vector is inserted into every network and each produces an initial simulated output. A comparator, an additional neural network, examines the simulated outputs and determines which network produced a result that was the most comparable to the target output. Only the winning network will be allowed to train on the input through the adjustment of its internal weights. Additionally, the comparator learns which network yielded this result and the corresponding input training vector related to this output. This allows the comparator to analyze various inputs and determine the appropriate network in the system that would provide the most accurate results. Figure II-7 shows an example with neural network 2 providing the best initial simulated output (indicated by the arrow from the comparator to network 2). This winning network will be allowed to train using back propagation to produce a bridge condition vector similar to the target training output.

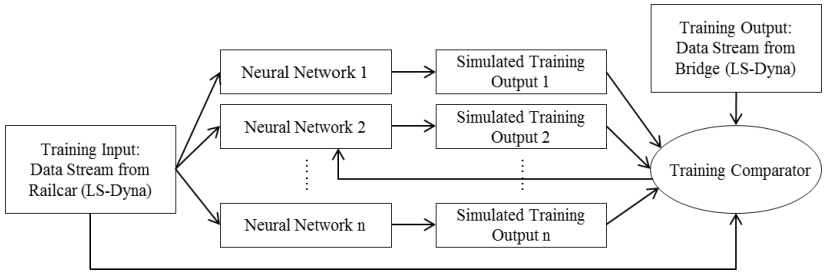

Figure II-7: Competitive Array of Neural Networks Training Process

The winning network is chosen by evaluating the dot product of the target training output unit vector and the unit vectors of the simulated training output from multiple competing networks. The dot product of two unit vectors has a range from -1 to 1. Unity specifies the vectors are in the same direction, while negative unity signifies opposite directions. Therefore, the dot product closes to 1 is chosen as the winning network, and is allowed to continue training using back propagation.

Figure II-8 illustrates an example using the dot product of unit training vectors to determine which network produced the most similar simulated output. In the example, neural network 2 produced the result closest to unity and is therefore the winning network. Note that the simulated vector from neural network 3 has the smallest mean squared error, but produces the dot product furthest from unity. Conversely, the second neural network has the largest mean squared error, but it produced the winning network. This is due to the mean squared error determining which simulated output produced a vector location that was the closest to the target, while the dot product evaluation considers the patterns within each vector to determine the winning network.

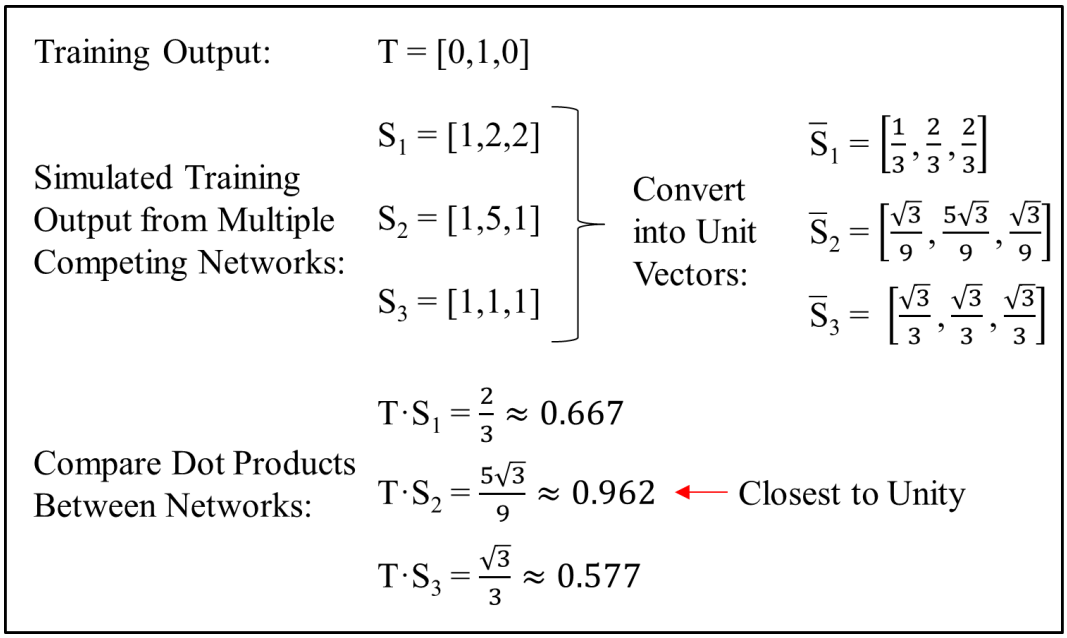

Figure II-8: Dot Product Evaluation for Neural Network System

II.6 **Testing Artificial Neural Networks**

Once neural networks have been trained to recognize the patterns within a provided set of input and output data, they can be tested with new data. The testing done during this research was conducted with wheel path acceleration experimental data provided in Chapter 3. The performance of the artificial neural network system is related to the procedure used during the training process. The more variation in patterns and trends learned while training will be reflected in the system's ability to recognize new patterns. This involves providing the network with the same initial input and output data set, and adding various degrees of noise to the results.

In addition to adding noise to a sample data set to create multiple variations, the training data sets can show different conditions. The inputs and outputs used during this

research included the behavior of a timber railway bridge and the corresponding railcar reaction as it crossed. A bridge with various levels of damage was used to train the artificial neural networks to determine the severity of impairment in the structure.

It is beneficial to provide different conditions for the neural networks to analyze during the training process as the networks learn to recognize the input and corresponding output patterns of five different impairment conditions described in Chapter 4. After completing the training process, the operating phase shown in Figure II-9 was conducted to test the networks. During this process, all internal weights have been established and the comparator network is presented with a new set of input data. The comparator analyzes the signal and presents the data to a neural network that was trained with data containing similar patterns. The networks generate the estimated average chord midspan deflection as the operating output of the system. The estimated deflections can be compared to experimentally measured deflections to determine the accuracy of the neural network method. The capability of artificial neural networks to produce reliable results from the experimental data stems from its ability to generalize. This means the networks can provide realistic outputs when presented inputs that were not seen during the training process (Haykin, 1999).

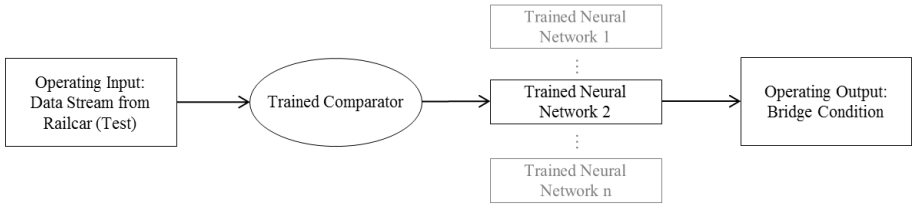

Figure II-9: Neural Network Operating Phase

CHAPTER III

EXPERIMENTAL TESTING

III.1 Experimental Design

This research tested two 15 foot spans of Bridge 17.14, an open deck timber trestle railway bridge in service, to observe the bridge's behavior under live loading. Specific nomenclature defined each stringer according to its span, chord, and ply. Figure III-1 shows the nomenclature of the stringers and bents of Bridge 17.14. Span 7 and span 8, the spans monitored during the tests, correspond to the 7th and 8th span crossed by a southbound train. The western chord was labeled as chord 1 and the eastern chord was chord 2. The ply numbering of the stringers increase from west to east.

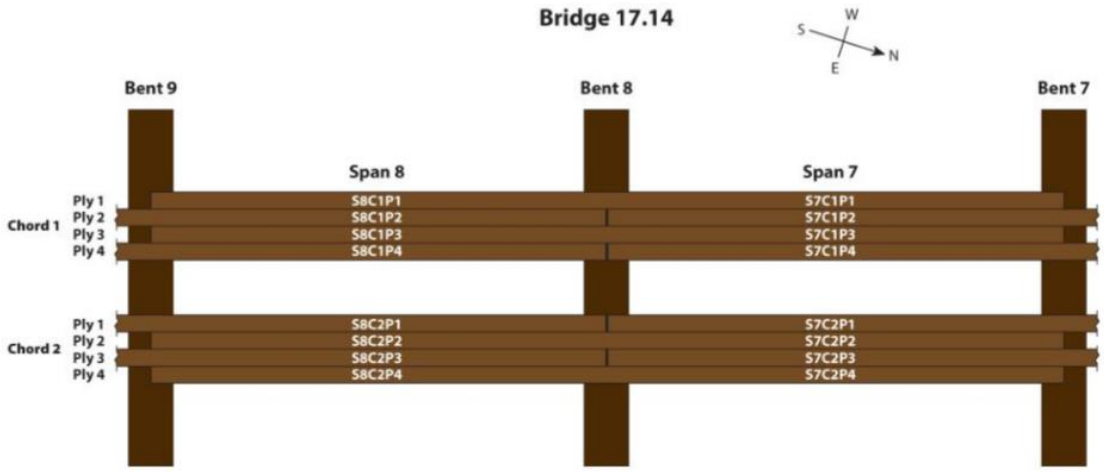

Figure III-1: Bridge 17.14 Nomenclature

A work train consisting of two locomotives followed by two railcars, shown in Figure III-2, provided a dynamic load for bridge testing. The two railcars were loaded

with a large amount of steel to simulate the effects of a loaded railcar on the bridge. These railcars are typically used to determine to effectiveness of the weigh stations that the train would cross along its journey. These railcars commonly travel across timber bridges making them ideal for employing the structural impairment technology.

Figure III-2: Work Train Crossing Bridge 17-14

Observing the motion of the train during loading required instrumenting four wheel paths. A wheel path refers to the horizontal and vertical motion of an axle. Research has shown that impairment detection works the most efficiently when monitoring two trucks connecting two separate railcars opposed to analyzing two trucks on a single railcar (Orsak, 2012). Therefore, the two trucks connecting the two trailing railcars were instrumented to record the motion of the train. This configuration resulted in the longest consecutive load to be placed on a single span during the train loading. The

locations of the wheel paths with respect to the rest of the work train are shown in Figure III-3. Wheel path 1 was the first axle to enter a test span, and wheel path 4 was the last to exit.

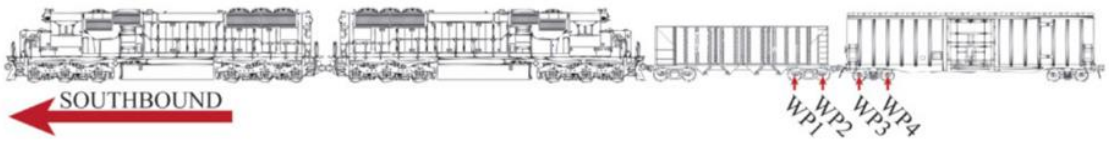

Figure III-3: Work Train

Data was collected with the work train traveling in the southbound direction at speeds ranging from 10 mph to 50 mph collected at increments of 10. Table III-1 shows each speed was repeated twice to provide multiple measurements for every test velocity. Multiple speeds were employed during testing because previous numerical research by Orsak (2012) indicated that faster speeds were more accurate at detecting impairments using artificial neural networks.

Table III-1: Test Speeds

Test	Speed (MPH)
1	10
2	10
3	20
4	20
5	30
6	30
7	40
8	40
9	50
10	50

III.2 **Time Sync Apparatus**

The motion of the wheel paths were observed with two different devices: photoresistors and accelerometers. The photoresistors were used as wheel path positon sensors to track the horizontal progress of the axles as the train traversed the bridge. A photoresistor will decrease in resistance while in the presence of light. The devices were placed in dark tubes so they would not be affected by ambient light from the environment. A 360 lumen flashlight was used to excite the photoresistors. This excitation is seen as a spike in voltage in the results produced by the data acquisition systems. The photoresistor devices and flashlights were placed on wooden posts that were located at bent locations. Another photoresistor and flashlight combination was magnetically attached to the work train. The experimental setup for these devices is shown in Figure III-4.

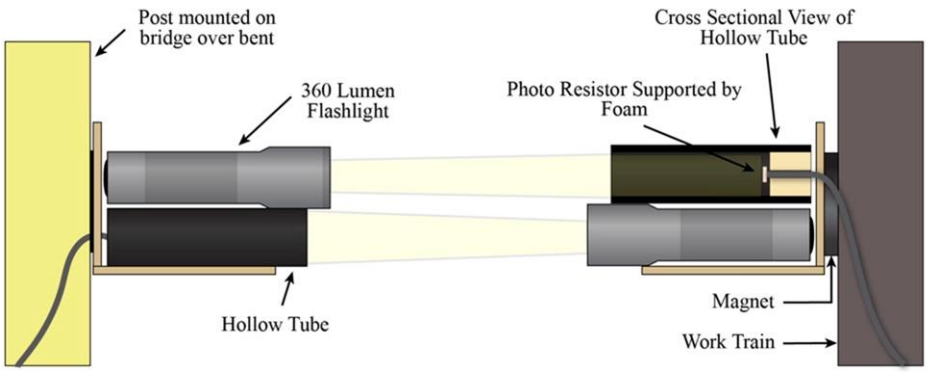

Figure III-4: Photoresistor Experimental Setup

The photoresistors were constructed in the form of a Wheatstone bridge as seen in Figure III-5. The photoresistor was placed into one of the resistor spots, R, on the diagram while the other locations were filled with normal resistors. The excitation voltage, V_{in},

supplied to the circuit was set to 10 V. The resistances in the three resistors was setup to create a balanced bridge when the photoresistor was not in the presence of light. This led the output voltage, Vac, of the system to read a value of zero when the device was in the dark. Once the photoresistor was place in front of a light source, the resistance of the device was decreased, thus creating an unbalance bridge circuit. This imbalance was seen as an increase in the output voltage produced by the system. The amount of voltage increase was proportional to the amount of light being measured by the photoresistor. For example, there were higher voltage readings for direct sunlight compared to fluorescent lighting inside a building. There was also an observable difference seen in the output when pointing the device at the ground and directly at the light source. Dark hollow tubes and high lumen flashlights were needed to allow these devices to be used properly in an outdoor setting.

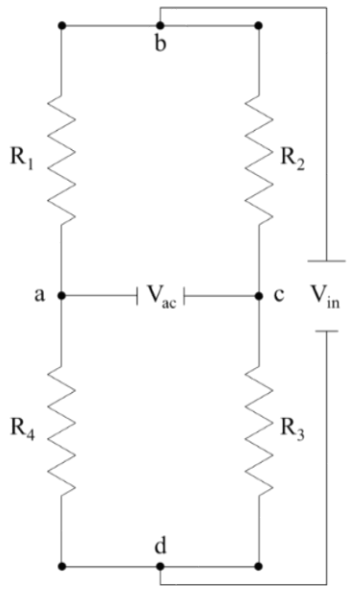

Figure III-5: Wheatstone Bridge Circuit

These devices were required because there were two different data acquisition systems used during testing. A portable Dewetron system was strapped to the work train to measure the vertical acceleration and photoresistor signals. Additionally, a StrainBook was used to record the deflections and photoresistor signals from the bridge. For the neural networks to operate efficiently, the time history of the accelerometers needs to match the time history of the string potentiometers. The internal clocks could be manually set to the same time, but there is a factor of human error that needs to be accounted for. If the time was off by as little as a half a second, the work train could travel nearly 37 feet for a 50 mph test. The use of photoresistors narrowed this error down to inches for any speed.

A photoresistor and flashlight were fastened to three posts and mounted above bents adjacent to the test spans. A second photoresistor/flashlight combination was magnetically attached to the work train above wheel path 1. The flashlights on the posts stimulated the photoresistors on the vehicle and vice versa. This setup allowed the time wheel path 1 crossed each bent to be seen in the results, and was used to determine the velocity of the work train for each test.

III.3 **Deflection Measurement**

String Potentiometers measured the deflection of each stringer. String potentiometers act as variable resistors converting displacement changes to resistance changes. These devices typically consist of a displacement cable, wire spool, rotational spring, and a sensor to monitor the motion. The displacement cable extends from the shell of the device to attach to a moving object. The other end of the cable wraps around a

spool inside the device shell. A torsional spring applies a small force to the spool to remove any slack in the cable. The wiper remains stationary in the device shell and contacts the displacement cable in one location.

The circuit of the device consists of a supplied voltage, a resistor, and a grounding mechanism shown in Figure III-6. As the cable extends and retracts from the transducer, the wiper contacts the cable at a different location along its length. The longer the effective length of the cable the higher the resistance in the circuit. As the resistance changes, the output voltage also changes. The voltage change is linearly proportional to the length of the displaced cable.

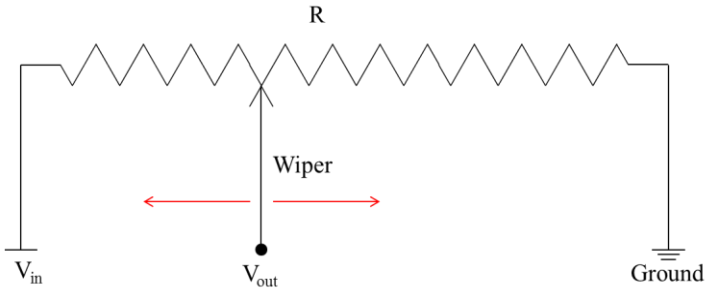

Figure III-6: String Potentiometer Circuit

String potentiometers were placed on the ground below the bridge. The devices were mounted to steel plates to prevent them from sliding or lifting off the ground during testing. The displacement cable was pulled out halfway from the device shell to capture both positive and negative deflections. Stainless steel fishing line was used to connect the string potentiometer cables to a cup hook screwed into the bottom face of the stringers. Since the tensile force was minimal, any change in the wire length was assumed to be

negligible compared to the bridge deflection. Figure III-7 shows two rows of string potentiometers attached to the underside of Bridge 17.14 stringers.

Figure III-7: String Potentiometer Setup Under Span 7

String potentiometers measured both the total and net midspan stringer deflections. The locations of the string potentiometers on the two spans are shown in Figure III-8. The devices located in the center of the stringers record the total midspan stringer deflection while the sensors on the ends are used to calculate the net deflection. The net deflection is an additional measurement that takes the settlement of the supports into account. The net stringer deflection is found by subtracting the average of a stringer's north and south end deflections from the total deflection, as seen in Equation III-1.

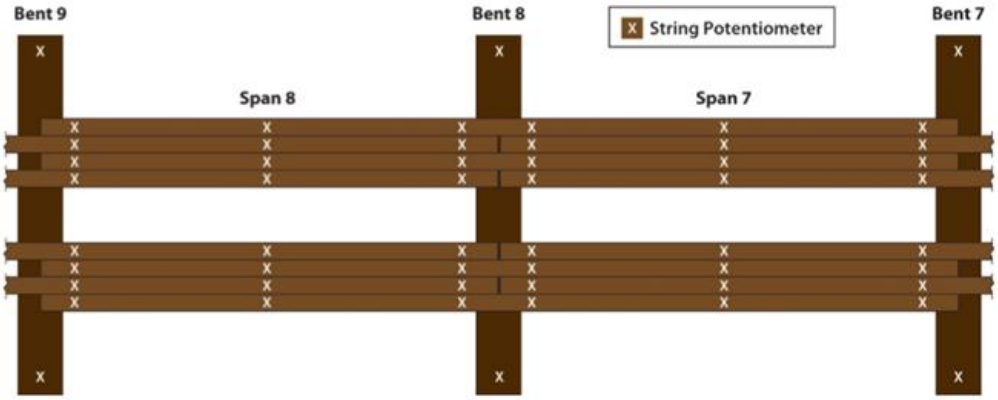

Figure III-8: String Potentiometer Locations

$$\delta_{Net} = \delta_{Total} - \frac{\delta_{North\ End} + \delta_{South\ End}}{2} \qquad \text{Equation III-1}$$

The net defection of a span differs from the total deflection of that span, as seen in Figure III-9. As a rail vehicle traverses the bridge, the stringers will deflect along the span, and the bents compress under the weight of the train. In the figure, d_{ave} represents midspan deflection if the stringer remained completely rigid while the ends deflected. The average deflection and the deflection due to the stringer bending both contribute to the total midspan deflection, δ_{total}. Thus, the net deflection, δ_{net}, can be calculated by determining the difference between d_{ave} and δ_{total}. The net deflection relates to stringer bending, and therefore provides a method for quantifying damage in the beam.

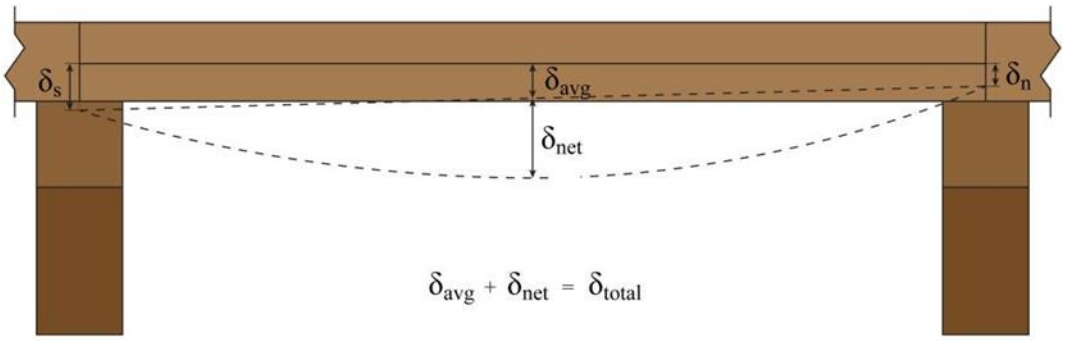

Figure III-9: Side View of Stringer Bending

A data acquisition system recorded the output voltage from the string potentiometers and converted it to a corresponding displacement measurement. A sample rate of 1,000 Hz was used for all stringer deflection measurements during testing. A large sampling frequency was required for this test because high speed railcar velocities were employed.

III.4 Midspan Deflection Results

Figure III-10 shows the time history of the total midspan deflection for span 7 of test 9. The remaining plots can be found in Appendix A. The eight discernable peaks in the plot that correspond to a truck in the work train crossing the midspan. The stringer plies in chord 1 (West chord) were plotted with solid lines, and dotted lines were used for chord 2 (East chord). The four vertical lines in Figure III-10 indicate the time each wheel path is positioned over the midspan. The four stringers in chord 2 all have larger deflection values than those in chord 1.

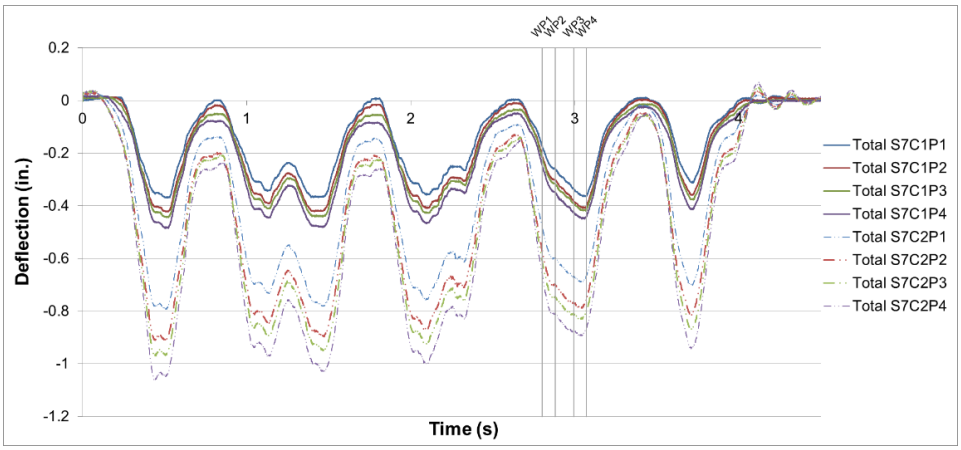

Figure III-10: Work Train Midspan Deflection Time History for Span 7 Test 9

Figure III-11 modifies the time axis so it begins when WP1 enters span 7 and ends when WP 4 exits the span. The time axis for this plot is in sync with the wheel path vertical acceleration plots. The photoresistor wheel path position sensors were used to synchronize these data sets. To accurately compare the bridge impairments produced from the neural networks, the time axis of the data collected from the railcars needed to match the bridge deflection time axis.

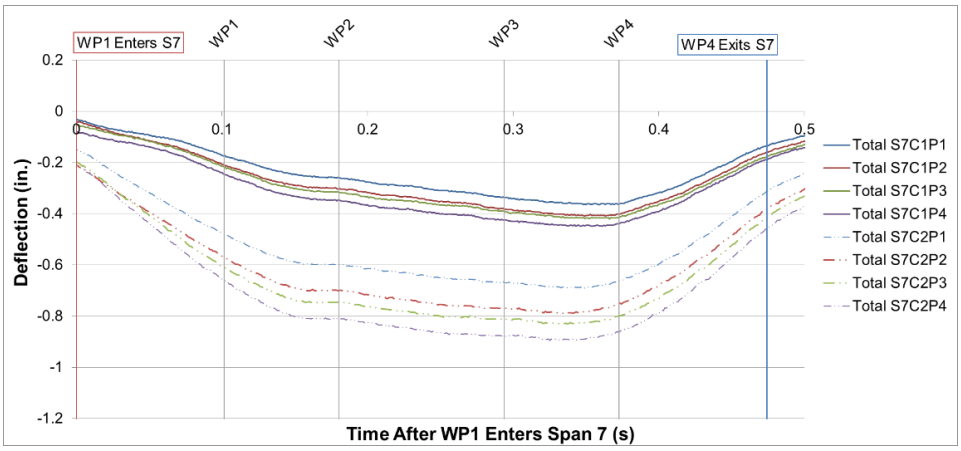

Figure III-11: Work Train Total Midspan Deflection for Span 7

Figure III-12 shows the net midspan stringer deflection for span 7 of test 9. The net deflection takes into account the settlement of the supports, therefore the maximum values for each individual stringer has decreased compared to the total deflections. The two peaks corresponding to the instrumented trucks can be seen in these plots. They occur after the first axle in the truck crosses the midspan but before the second axle reaches that location.

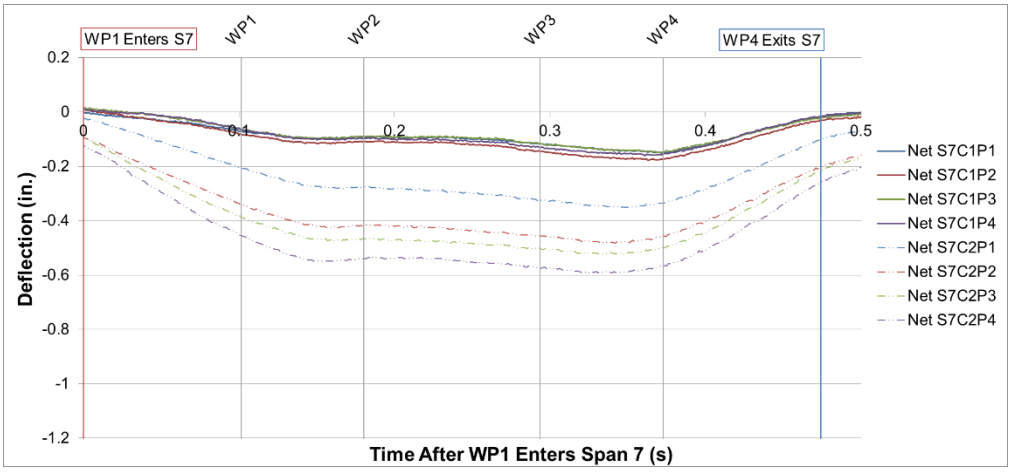

Figure III-12: Work Train Net Midspan Deflection for Span 7

Figure III-13 indicates approximately how the stringers deflect across the cross section when the bridge is loaded with the railcar. The figure shows that there is more deflection located in the east chord of stringers than in the west chord. The value of the stringer deflection also increases from the westernmost stringer in each chord to the easternmost stinger.

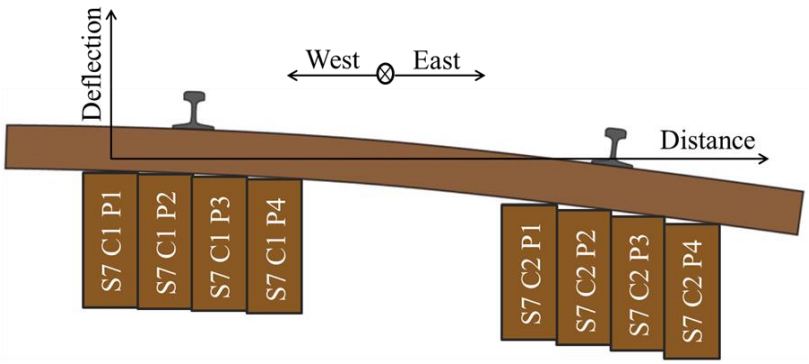

Figure III-13: Cross Sectional Stringer Deflection

A plot of the midspan stringer deflection values compared to their distance from the westernmost stringer at a time of 0.178 seconds is shown in Figure III-14. This period corresponds to the time at which wheel path 2 is positioned over the midspan of the test bridge. The plot indicates that the stringers increase from west to east in a nearly linear manner. The coefficient of determination, r^2, for this plot is 0.987. This indicates that there is a strong linear correlation between the midspan deflection values and the stringer's respective location along the cross section.

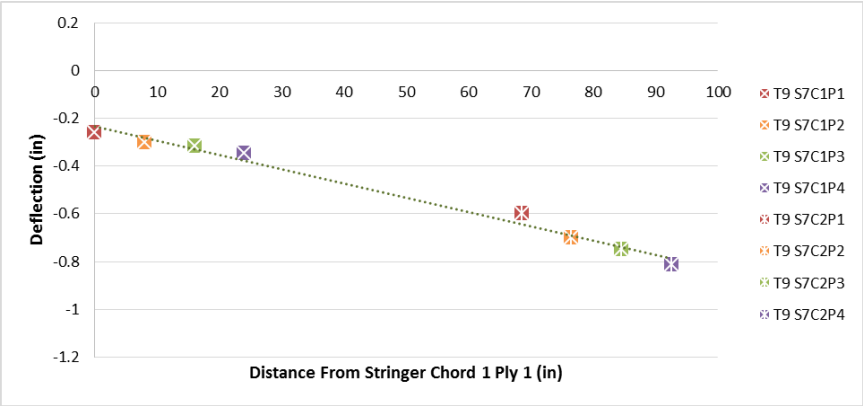

Figure III-14: Stringer Cross Section Deflection at Time = 0.178 s

III.5 Deflection Ratio

There were five different work train velocities used during the experimental testing portion of the research. The purpose of using various test speeds was to determine the effect of speed on the ability of the artificial neural networks to analyze the behavior of the bridge. It was also observed how the differing velocities influenced the measured maximum deflection values. Table III-2 shows maximum average chord deflection values determined from each speed. The average chord deflections were calculated by finding the mean of the four plies of stingers in each chord as seen in Equation III-2. The reason for using the average chord deflections for the comparison are twofold. First, the average chord deflections are used as an output while training the artificial neural networks, and the comparison shows how speed affected the results. Secondly, this measurement indicates how the four stingers in a bridge chord behave as a unit, and eases the process of comparing different spans, speeds, and sides of the bridge.

Table III-2: Maximum Average Chord Total Deflections

Test	Speed (mph)	S7 C1 (in)	S7 C2 (in)	S8 C1 (in)	S8 C2 (in)
1	10	-0.395	-0.929	-0.385	-0.829
2	10	-0.396	-0.926	-0.392	-0.841
3	20	-0.398	-0.923	-0.382	-0.879
4	20	-0.396	-0.929	-0.376	-0.886
5	30	-0.422	-0.884	-0.387	-0.865
6	30	-0.426	-0.884	-0.386	-0.877
7	40	-0.427	-0.883	-0.399	-0.864
8	40	-0.426	-0.901	-0.402	-0.865
9	50	-0.430	-0.934	-0.411	-0.886
10	50	-0.434	-0.941	-0.412	-0.885

$$\delta_{ave} = \frac{\delta_{ply\,1} + \delta_{ply\,2} + \delta_{ply3} + \delta_{ply\,4}}{4}$$ Equation III-2

The table above provides the maximum average chord total deflection values measured for each test. It also shows tests of the same speed have slightly different maximum values measured in each chord. This occurs because there are slight differences in the actual velocity the work train was travelling, and wind speed and direction at the time of the test also influence the results. In addition to the variation in values for similar speeds, the maximum values also tend to increase as the speed increases. This behavior is not consistent for each chord, as there are instances where the measured deflection decreases in faster speeds.

One method of comparing the values measured in one speed to another is to use a deflection ratio. The deflection ratio, DR, is calculated by dividing the maximum deflection measured at a specific speed by the maximum deflection at 10 mph as seen in Equation III-3. The 10 mph values were used as the denominator because this speed had the lowest values for each chord for a majority of the speeds. It was closest the experiment came to a crawl speed, where the dynamic impacts seen in higher speeds would be minimized. Since there are two values for each speed, these values were averaged before Equation III-3 was calculated. This allowed the DR for every speed the chords experienced to be determined. The average DR for all speeds were calculated by averaging the values found from each chord. The deflection ratio that was determined using this process is provided in Table III-3.

$$DR = \frac{Max.\ Chord\ Deflection\ at\ Speed\ x}{Max.\ Chord\ Deflection\ at\ 10\ mph}$$ Equation III-3

Table III-3: Total Defection Ratio

Speed	S7C1	S7C2	S8C1	S8C2	Average
10	1.000	1.000	1.000	1.000	1.000
20	1.004	0.998	0.976	1.057	1.009
30	1.072	0.953	0.995	1.043	1.016
40	1.077	0.961	1.032	1.035	1.026
50	1.093	1.010	1.059	1.060	1.056

Figure III-15 shows the deflection ratio for the chords and the average of the four values. A trend in the DR results can be seen by examining the average value for each speed. As the figure shows, the deflection ratio increases for faster speeds. This implies the timber bridge maximum deflection also increases at higher speeds. The difference between the largest speed and smallest speed maximum deflections amounts to less than 0.05 inches. While faster rail vehicle speeds also increases the deflection, the amount increased isn't as significant as those seen from other factors. The level of damage in the bridge and the load of the rail vehicle both tend have a more meaningful impact on the bridge behavior.

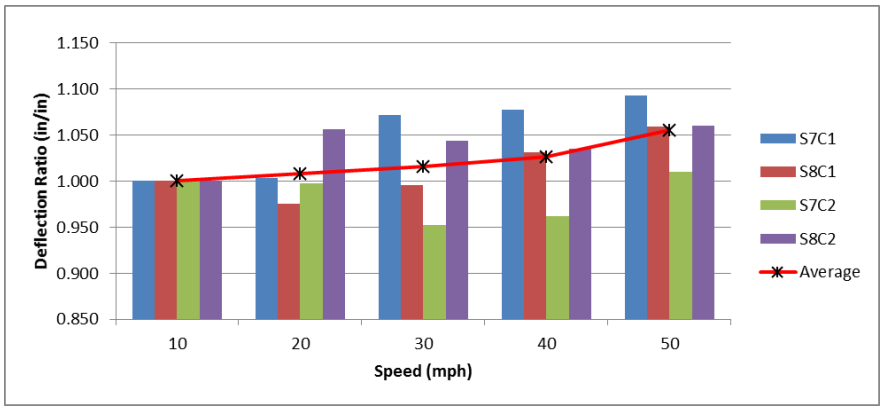

Figure III-15: Total Deflection Ratio

This process was also repeated for the measured net deflections calculated for the timber railway bridge. Table III-4 shows the maximum average chord net deflections for all test speeds. The results are similar to those obtained using the total chord deflections. Slower test speeds tend to show smaller maximum values, while fast work train speed have larger deflection values. The net deflection values measured in this test were found to be approximately half of the total deflection values.

Table III-4: Maximum Average Chord Net Deflections

Test	Speed (mph)	S7 C1 (in)	S7 C2 (in)	S8 C1 (in)	S8 C2 (in)
1	10	-0.151	-0.561	-0.169	-0.431
2	10	-0.153	-0.558	-0.174	-0.436
3	20	-0.152	-0.558	-0.171	-0.457
4	20	-0.149	-0.561	-0.164	-0.462
5	30	-0.160	-0.533	-0.174	-0.453
6	30	-0.162	-0.532	-0.171	-0.459
7	40	-0.163	-0.539	-0.181	-0.450
8	40	-0.163	-0.539	-0.182	-0.452
9	50	-0.166	-0.574	-0.185	-0.463
10	50	-0.168	-0.579	-0.186	-0.463

The net deflection ratio values are provided in Table III-5, and shown visually in Figure III-16. The average net deflection ratios calculated from this research were found to be slightly larger than the total deflection ratios. For example, the 40 mph and 50 mph tests showed net DR values that were 0.78% and 1.42% larger than the total DR values. The average net deflection ratio increased as the pace of the railcar quickened. This is consistent with the results seen in the total deflection ratio plot.

Table III-5: Net Deflection Ratio

Speed	S7C1	S7C2	S8C1	S8C2	Average
10	1.000	1.000	1.000	1.000	1.000
20	0.993	0.999	0.976	1.061	1.007
30	1.059	0.952	1.005	1.053	1.018
40	1.075	0.963	1.058	1.040	1.034
50	1.103	1.030	1.081	1.070	1.071

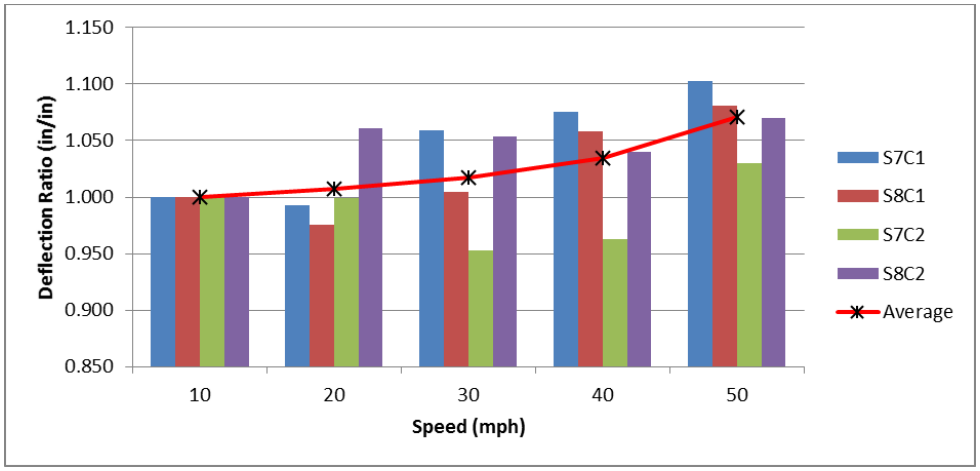

Figure III-16: Net Deflection Ratio

III.6 Vertical Acceleration Measurement

The vertical motion of the four wheel paths were monitored with uniaxial accelerometers. The accelerometers were magnetically attached to the side frame of the railcar truck just above the axle (Figure III-17). This location was chosen to capture the motion of the wheels as the vehicle crossed the bridge. The devices were placed on the side frame because the railcar itself is affected by vibrations that are dampened by the truck springs. Eight accelerometers were used to capture the vertical motion of the vehicle axles. There were two accelerometers placed on each wheel path with one device located on either side of the axle. This allowed for the acceleration to be measured on the east and west sides of the railcar.

Figure III-17: Accelerometer Placement

A variable capacitance accelerometer was used to measure the wheel path accelerations during the experimental test. A typical configuration for a variable capacitance accelerometer is shown in Figure III-18. This type of accelerometer consists of an inertial mass that is anchored to shell of the device by a material with a designed stiffness indicated by a spring in the figure. There are two capacitor plates used in this general configuration. One is mounted on the outside shell and will remain stationary when the accelerometer is subjected to motion. The second capacitor plate is attached to the inertial mass and will move if the device accelerates.

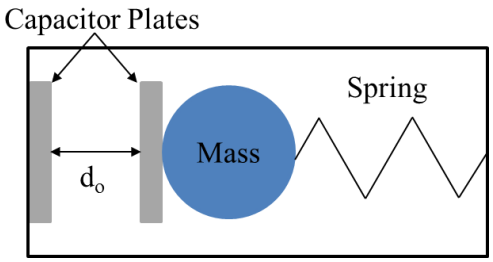

Figure III-18: Capacitive Accelerometer General Concept

The capacitance produced by the accelerometer is a function of the distance between the two capacitor plates. As the distance between the plates decreases, the capacitance produced by the device will increase. While at rest, the original distance between the plates, d_o, is steady and capacitance of the system is constant. When the accelerometer is subjected to motion, the distance between the two plates, d_f, will also change, and the corresponding change in capacitance can be measured. The effect of acceleration on the general configuration of a capacitive accelerometer is shown in Figure

III-19. The change in capacitance can be calibrated to correspond to a change in the measured acceleration.

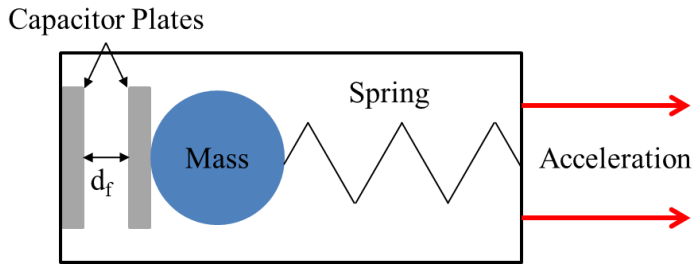

Figure III-19: Capacitive Accelerometer Subjected to Acceleration

The above figures show how a capacitance accelerometer measures acceleration using capacitance. Accelerometers used in industry typically use a number of capacitor plates to measure acceleration values as seen in Figure III-20. In the figure, the inertial mass is connected to the shell of the accelerometer through a number of flexible spring legs. There are also a number of capacitive plates that are connected to the mass and will move when the system accelerates. Stationary capacitive plates are used to create a change in the output capacitance that can be related to a measured acceleration. Using a number of plates in this configuration will improve the accuracy of the measured acceleration to produce more reliable results.

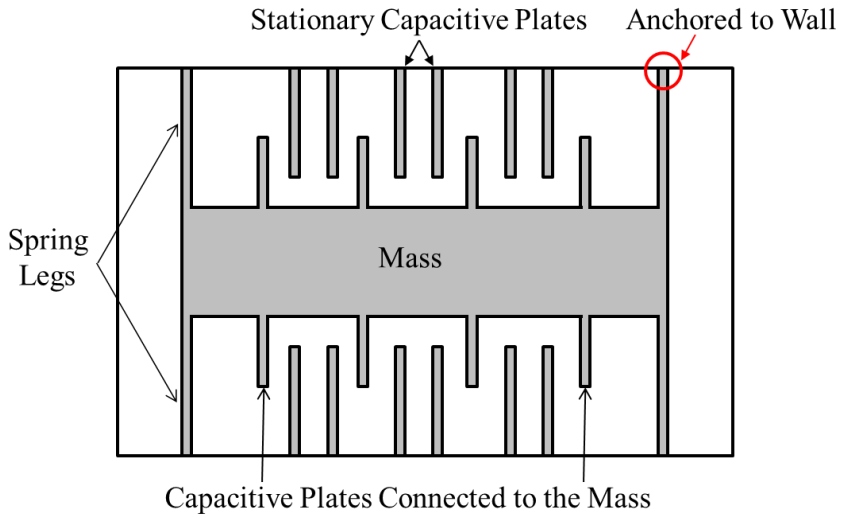

Figure III-20: Variable Capacitance Accelerometer with Multiple Plates

The vertical acceleration of the 4 wheel paths on the work train were measured using 8 of these variable capacitance accelerometers. The acceleration data plots for span 7 of the first 50 mph test are given in Figure III-21. Additional plots of span 8 and the remaining tests are provided in Appendix A. The four wheel paths (WP) are shown in separate plots. The accelerometers positioned over the west chord, or chord 1, were plotted with a blue line and the east chord is indicated in red. The time axis starts when wheel path 1 enters span 7 (or crosses over bent 7), and ends when wheel path 4 exits span 7 (or crosses over bent 8). The time a wheel path is located over a bent is indicated in the plots with colored squares. All acceleration data sets were filtered with a 100 point moving average. This filter helps to reduce the effects of high frequency vibration caused by the train's moving parts, and captures the vertical motion of the wheel paths.

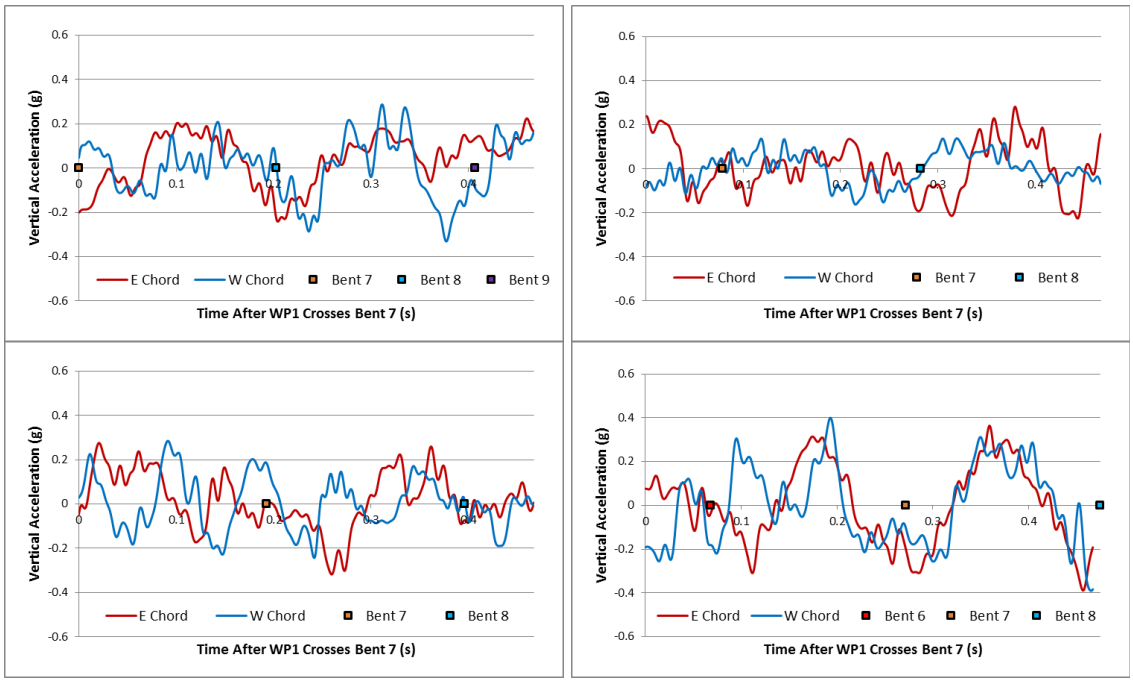

Figure III-21: Span 7 Test 9 Vertical Acceleration: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

III.7 Accelerometer Filtering

All of the wheel path acceleration data signals presented in this research were filtered using a 100 point simple moving average. The equation for implementing a moving average on a data set is shown in Equation III-4. Xi refers to the starting data point for the moving average filter, and N_x is number of points averaged. A simple moving average is often used when analyzing the history of stock prices to better visualize trends within the data, but it has also been used to filter out excess noise from data sets. For stocks, the moving average is calculated by determining the mean of the preceding N_x number of day's final prices. Once a new day begins, the mean is recalculated with the

new N_x number of day's closing prices. The average moves with each new day, and the trend in the data can be observed.

$$Simple\ Moving\ Average = \frac{X_i + X_{i-1} + \cdots + X_{i-(N-1)}}{N_x} \quad \text{Equation III-4}$$

A similar process was used to compute the moving average of the acceleration signals. The formula was adjusted slightly because the current method caused a phase shift in the filtered data. The phase shift was corrected by starting the moving average N/2 points before the point of interest and stopping N/2-1 after the data point. This placed the calculated mean approximately in the center of the moving average. The original simple moving average and phase shift corrected plots compared with sin(2πX) and random noise is provided in Figure III-22. The figure shows the original moving average was able to determine the correct peak and trough values of 1 and -1 respectively. The corrected plot shows accurate peaks as well as determining the X intercept values of 0, 0.5, and 1. The final shape of the plot resembles the original sin(2πX) without noise.

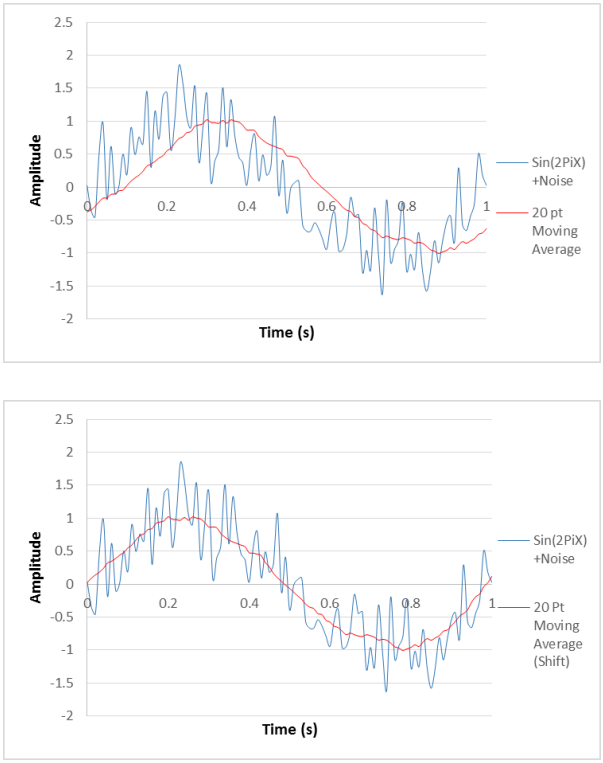

Figure III-22: Simple Moving Average: Original (Left), Corrected (Right)

The moving average filter was chosen for its ability to determine original data signals in the midst of relatively larger noise values. The reason it acts as an efficient filter is a simple moving average is finite impulse response low pass filter with a rectangular window. This filter keeps a majority of the lower frequency vibrations while lessening the effects of the higher frequency accelerations. The motion of the axles as they cross the bridge is in the lower frequency range, and the larger frequencies arise from the moving parts of the railcar. The smaller frequencies are of the most interest for this study, so a low pass filter is and adequate filtering method. The process for implementing a finite impulse response filter with a rectangular window is summarized in the following. An original unfiltered wheel path acceleration signal is provided in Figure III-23. The figure

indicates acceleration spikes ranging approximately from -6g to 8g. It is also difficult to discern any pattern within the data that may show the axle is traversing a bridge. The excessive acceleration values accompanied by the lack of visible patterns and trends would make it difficult for the neural networks to analyze the data as it is. A low pass filter was needed to help reduce the effects of the high frequency accelerations.

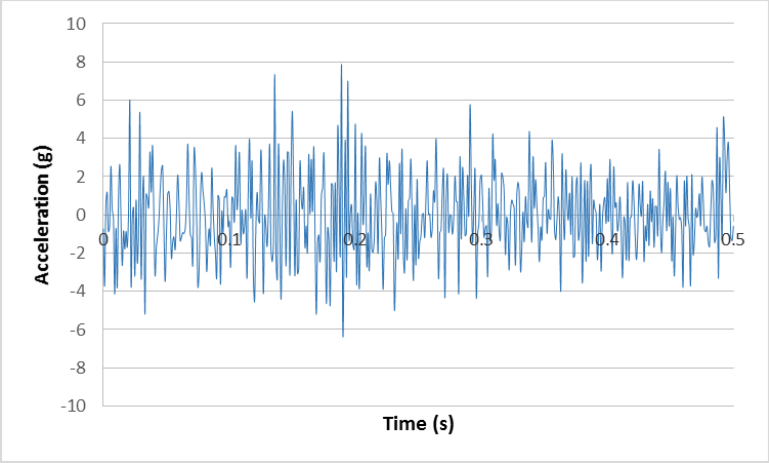

Figure III-23: Unfiltered Wheel Path Acceleration Signal

To apply the finite impulse response filter, the acceleration must be converted from the time domain to the frequency domain using a Fourier transform, F or $\mathcal{F}$. The equation for a Fourier transform is given in Equation III-5. N is the number of time samples, n is current sample being considered, f_n is the acceleration value at time n, and s is the current frequency being considered. The acceleration signal transformed in to the frequency domain is shown in Figure III-24. The figure shows there are relatively large spikes in the 150, 275, and 375 frequency ranges. The accelerations related to the motion of the railcar

and the natural bridge movement are considerably less than these values, so the effects of the larger frequencies needed to be dampened.

$$F(s) = \mathcal{F}\{f(t)\} = \sum_{n=0}^{N-1} f_n e^{\frac{-i2\pi sn}{N}} \qquad \text{Equation III-5}$$

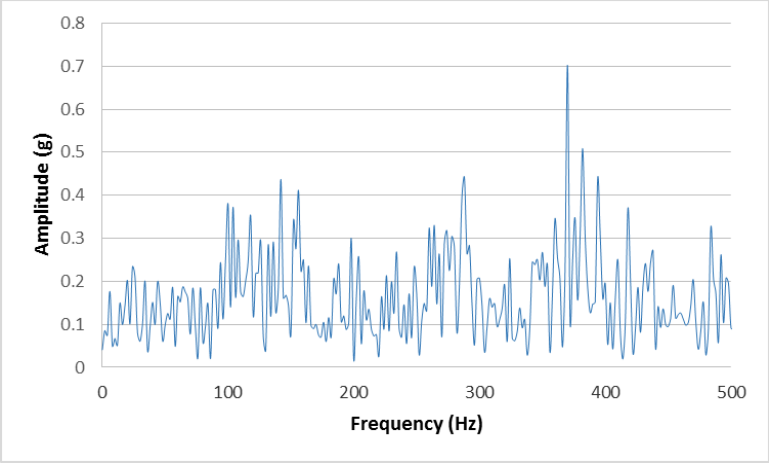

Figure III-24: Acceleration Signal in the Frequency Domain

The finite impulse response filter is applied by multiplying the frequency domain of the acceleration signal by the frequency domain of a rectangular step function. The step function, b, as a function of time, t, is defined in Equation III-6. The function is also dependent on the specified window size, k, and time step, ts. The rectangular step function is plotted in Figure III-25. The window size chosen for this acceleration signal was 100 and a time step of 0.001. Thus, the plot begins at a 1/k value of 0.01 before dropping to zero at 0.099 s. The rectangular shape of the plot can be observed in the figure.

$$b(t) = \begin{cases} \frac{1}{k}, & 0 < t < (k-1) * ts \\ 0, & otherwise \end{cases}$$ Equation III-6

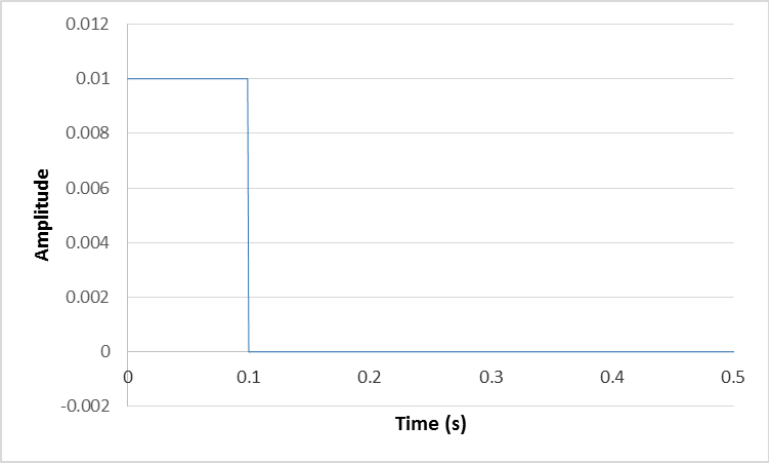

Figure III-25: Rectangular Step Function

The rectangular step function is then transformed from the time domain to the frequency domain using the Fourier transform function in Equation III-5. The step function in the frequency domain is plotted in Figure III-26. The plot begins with an amplitude of 1 and exponentially approached zero at higher frequencies. The plot also spikes towards zero in intervals of 10 Hz along the length. This interval is dependent on the window size used for the filter. The distance between the downward spikes will decrease as the window size increases and vice versa. Therefore, the window size affects how quickly the plot will exponentially approach zero.

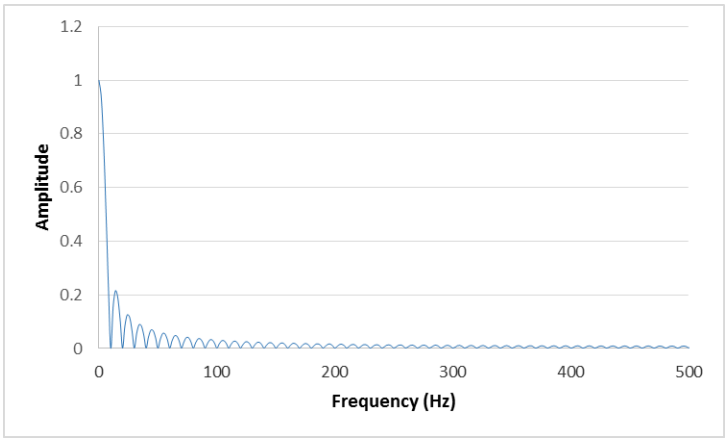

Figure III-26: Rectangular Step Function in the Frequency Domain

The acceleration signal is filtered by multiplying the frequency domain of the acceleration signal by the frequency domain of the rectangular window step function. The filtered acceleration signal in the frequency domain is shown in Figure III-27. The new figure appears similar to the rectangular step function in the frequency domain with the main difference occurring in the peak amplitude values. The filter also effectively lowered the amplitudes of the frequencies in excess of 100 Hz.

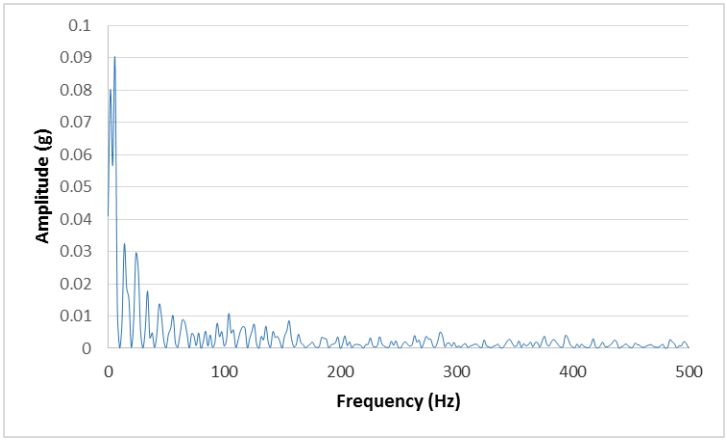

Figure III-27: Filtered Acceleration in the Frequency Domain

The filtered acceleration frequency data was then converted back to the time domain using the inverse Fourier transform ($\mathcal{F}^{-1}$) shown in Equation III-7. Similar to a simple moving average, the finite impulse response filter has a slight phase shift that occurs during the procedure. The phase shift can be corrected by using Equation III-8. The filtered acceleration signal in the adjusted time domain resulting from the inverse transform is shown in Figure III-28. There are now observable peaks and troughs in the data that indicates the railcar traversing the bridge. The amplitudes of the peaks and troughs now range from 0.25g to -0.3g. This is a significant reduction in the 8g and -6g spikes observed in the unfiltered data.

$$f(t) = \mathcal{F}^{-1}\{F(s)\} = \frac{1}{N}\sum_{s=0}^{N-1} F_s e^{\frac{i2\pi sn}{N}} \qquad \text{Equation III-7}$$

$$t_{corrected} = t - \frac{k-1}{2N} \qquad \text{Equation III-8}$$

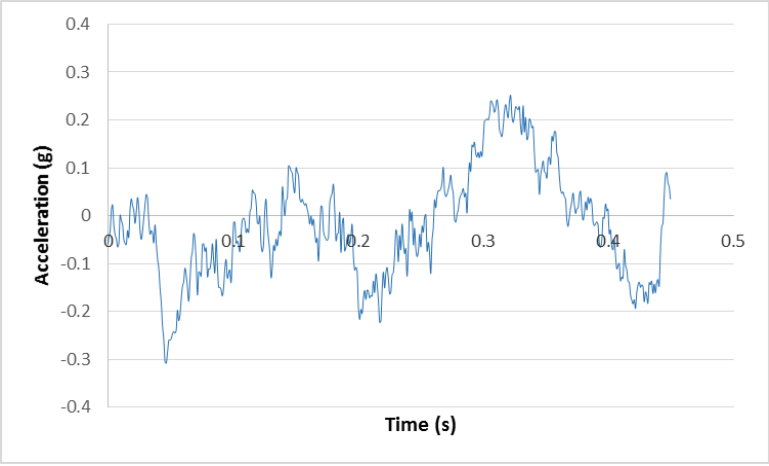

Figure III-28: Filtered Wheel Path Acceleration Time History

III.8 Vertical Acceleration Speed Comparison

In addition to observing how differing work train velocities affected the behavior of the bridge, the vertical wheel path accelerations from the vehicle were also compared. Table III-6 shows the maximum measured acceleration for each test, span, and chord of the bridge. The maximum value was determined during the period of time in which the axle containing the accelerometer was traversing the span. Since there were four instrumented wheel path axles crossing each chord for every test, these values were averaged to give the values seen in the table.

Table III-6: Maximum Vertical Acceleration Speed Comparison

Test	Speed (mph)	S7 C1 (g)	S7 C2 (g)	S8 C1 (g)	S8 C2 (g)
1	10	0.026	0.034	0.058	0.046
2	10	0.033	0.046	0.026	0.063
3	20	0.106	0.088	0.085	0.104
4	20	0.051	0.086	0.077	0.115
5	30	0.120	0.207	0.186	0.174
6	30	0.106	0.138	0.179	0.148
7	40	0.131	0.252	0.107	0.203
8	40	0.108	0.161	0.117	0.196
9	50	0.211	0.240	0.236	0.233
10	50	0.186	0.296	0.249	0.241

Table III-6 provides the maximum acceleration values for every test, span, and chord monitored during the experiment. A majority of the tested spans show chord 2 or the east side of the bridge experienced larger accelerations than chord 1. This behavior can also be seen in the maximum chord acceleration data shown in Table III-7. These

values were determined by averaging the maximum accelerations of both chords for all test speeds. This table shows that chord 2 tends to have a larger peak acceleration than chord 1 for all vehicle velocities. This behavior is related to maximum deflections observed in the previous section. The east side of the test bride experienced larger deflections than the west, and this uneven behavior is detected in the acceleration data. The maximum acceleration values also see significant increases as the work train speed increases. There were slight increases in deflections with changes in speed, but none were as dramatic as those seen in Table III-7. The wheel path acceleration values are largely dependent on the velocity of rail vehicle, and also have the ability to distinguish various levels of bridge deflections based on this data.

Table III-7: Maximum Average Chord Acceleration Comparison

Speed (mph)	Chord 1 (g)	Chord 2 (g)
10	0.036	0.047
20	0.080	0.098
30	0.148	0.167
40	0.116	0.203
50	0.220	0.252

CHAPTER IV

COMPUTER SIMULATION

IV.1 Finite Element Mesh

A finite element analysis using LS-Dyna computer software was used to simulate the field test results and train the artificial neural networks. Figure IV-1 shows a five span finite element mesh and the corresponding simulated train travel direction. The third span was used to calculate the total and net midspan stringer deflections. This span was chosen because it is influenced by loads placed on the adjacent spans. Five spans were used for this analysis because the length from wheel path 1 to wheel path 4 is greater than the span length of the bridge. The trailing wheel path would be positioned in span 1 while the leading wheel path enters the span of interest. Five spans are able to account for the effect of every axle load on the behavior of the bridge.

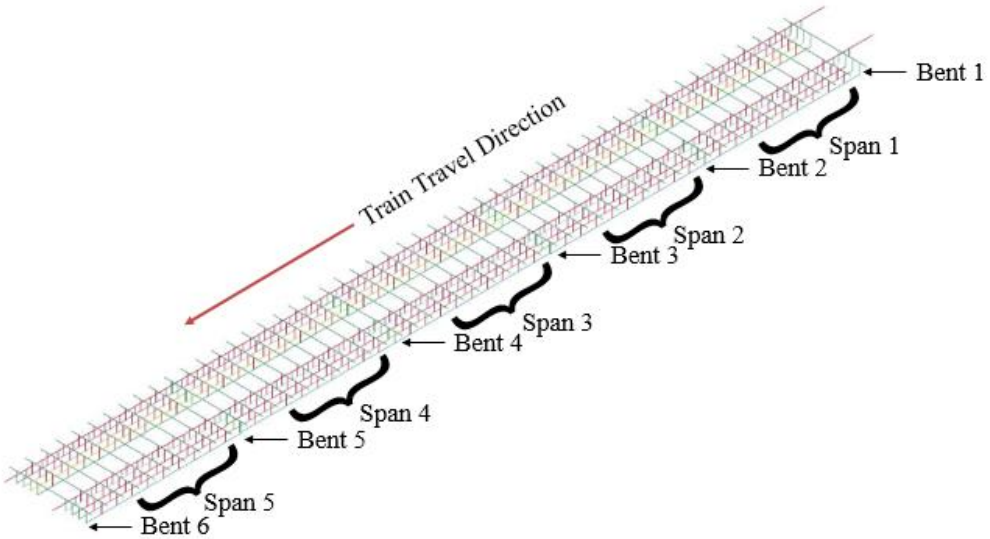

Figure IV-1: Five Span Finite Element Mesh

The mesh consists of rails, cross ties, and stringers modeled with linear elastic Hughes-Liu beam elements with cross sectional integration and were connected with discrete beam link elements shown in Figure IV-2. The structural properties of these elements are similar to those from the open deck timber railway bridge used during testing and are provided in Table IV-1. Each stringer covered two spans and included staggered continuity seen in the test bridge. Figure IV-2 shows stringers 2 and 4 of the western (top) chord are continuous over bent 5, and stringers 1 and 3 show two stringers meeting at slightly translated locations. Similar observations can be made about the stinger in the eastern (lower) chord except the opposite stringers are continuous.

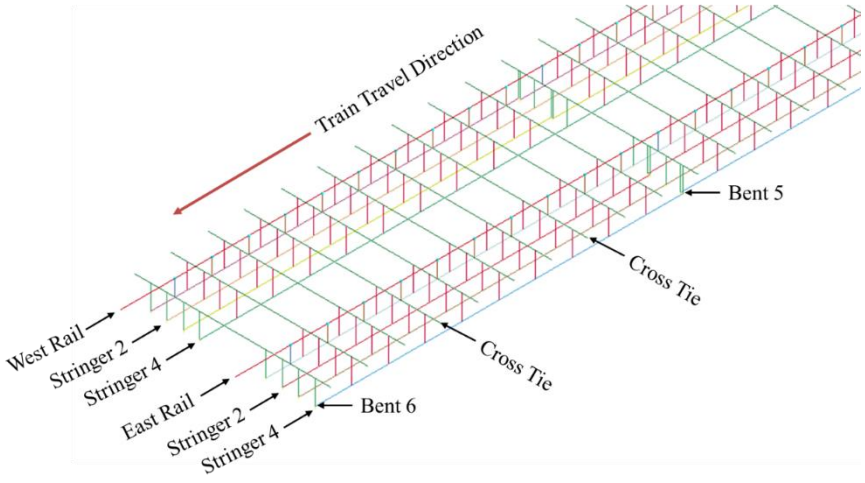

Figure IV-2: Finite Element Mesh (Zoomed)

Table IV-1: Material Properties

Material	E	b	h	$E*I_{horiz}$	ρ
	(kip/in^2)	(in)	(in)	($kip*in^2$)	($kip*s^2/in^4$)
Stringer	1100	8	16	3003733	3.28E-08
Cross Tie	1100	7	8	328533	3.28E-08
Rail	29000	3.2	5.7	1419689	7.35E-07

The reaction of the bents were simulated with a link element fixed at one end and attached to the stringers at the other. This was needed because a simply supported boundary condition caused excessive midspan uplift values when a load is applied to an adjacent span. This phenomenon was not witnessed in the experimental bridge deflection data. Additionally, a fixed support restrained the bent such that no effect from the adjacent loading was seen on the midspan. A support that lies between these two common boundary conditions was needed to accurately depict the behavior observed during experimental testing. The rotational stiffness of the links were adjusted such that there would be a slight uplift in the midspan deflection results. Additionally, the longitudinal stiffness of the elements were adjusted to account for the support settlement seen in the field test.

IV.2 **Simulated Loading**

The vehicle load was simulated with nodal masses that were ramped up and down at wheel locations of the moving work train. Since the two trucks connecting the railcars were instrumented with motion devices, only the connection seen in Figure IV-3 was analyzed. The two railcars were filled with steel scrap metal. However, the total weight

of these cars was unknown, so an assumed value of 80 kips per axle was used for the leading truck and 86 kips per axle for the trailing truck. The effects of the remaining six trucks were assumed to have a negligible impact on the behavior of the wheel path acceleration and midspan deflection, so they were neglected from the analysis. Additionally, it was assumed the wheel remains in direct contact with the rail. Therefore, the wheel path accelerations were approximated with vertical acceleration values of the rail at wheel locations.

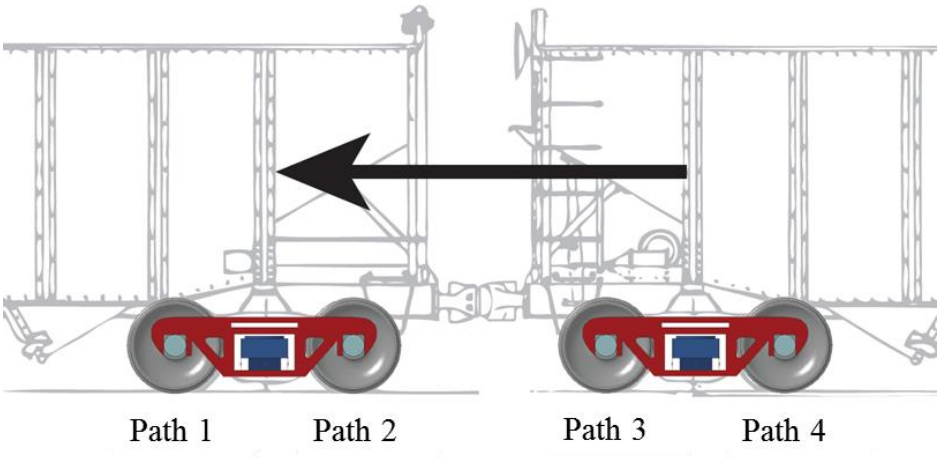

Figure IV-3: Simulated Railcar Loading

IV.3 **Applied Damping**

A global damping value was applied to the bridge during the finite element analysis to lessen the vibrational effects of the dynamic loading on the structure. Determining the correct damping value involved comparing the maximum midspan deflection values of the 50 mph experimental test to the 10 mph test. Testing showed that

as the speed increased, the total deflection also increased, thus, a deflection ratio (DR) could be determined. The DR is the ratio of the maximum midspan deflection measured at a specific speed to the maximum deflection at 10 mph. It was found that the damping value did not have a significant effect on the maximum deflection at slower speeds. Therefore, the global damping value was adjusted by trial and error until the DR from the analytical analysis matched the experimental value.

IV.4 Impairment Conditions

Five levels of simulated damage labeled impairment conditions, or ICs, were imposed during the finite element analysis. A summary of these impairment conditions are provided in Table IV-2. The purpose of the ICs were to provide variability to the artificial neural networks during the training process. The networks analyzed the test data, and predicted the bridge deflection by determining which IC most closely resembled the behavior from the test.

Table IV-2: Impairment Condition

Impairment Condition	EI Reduction	
	West Chord	East Chord
IC1	0%	0%
IC2	25%	50%
IC3	0%	75%
IC4	50%	25%
IC5	75%	0%

IC 1 is a damage scenario simulating a new bridge with no damage present. Damage was inflicted in the four remaining impairment conditions by reducing the flexural rigidity, or EI, in the east or west chord of stringers. As discussed in Chapter 1 the flexural rigidity is a combination of the Young's modulus, E, and moment of inertia, I, of a structural member. Structural damage, such as shear cracking, will reduce the moment of inertia of a member. Environmental damage, such as rotting or insect infestations, could diminish a beam's Young's modulus. Both of these properties affect a member's ability to resist bending or deflection, so both need be considered when evaluating damage in the stringers.

IV.5 Computer Simulation Results

The tested timber railway bridge showed very large deflections in the east chord and relatively small deflections in the west chord, and this behavior is similar to the results in IC 3. Figure IV-4 shows the total midspan stringer deflections obtained from the finite element analysis for IC 3 during a 50 MPH test. The deflection values corresponding to the four plies of stringers in the eastern chord of the plot are roughly twice as large as those in the western chord.

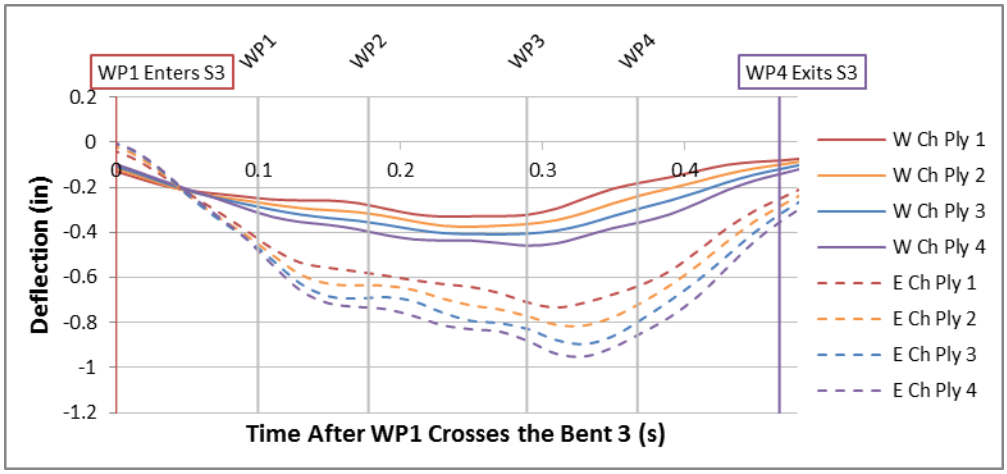

Figure IV-4: Total Midspan Stringer Deflection for IC 3 50 MPH Test

The artificial neural networks were trained to produce the average chord midspan deflections as an output. Figure IV-5 shows a plot of the average chord deflections of the field test compared to the finite element analysis results. The deflections were obtained by averaging the values of the four stringers in a chord throughout its time history. The figure shows that IC 3 compares favorably to the average chord deflections measured during test 9. The calculated deflection from the east chord of the computer starts closer to zero than the experimental test. However, the deflection values corresponding to the time when the train axles are positioned over the bridge midspan are similar. The similarity also remains after the train has begun to leave the span.

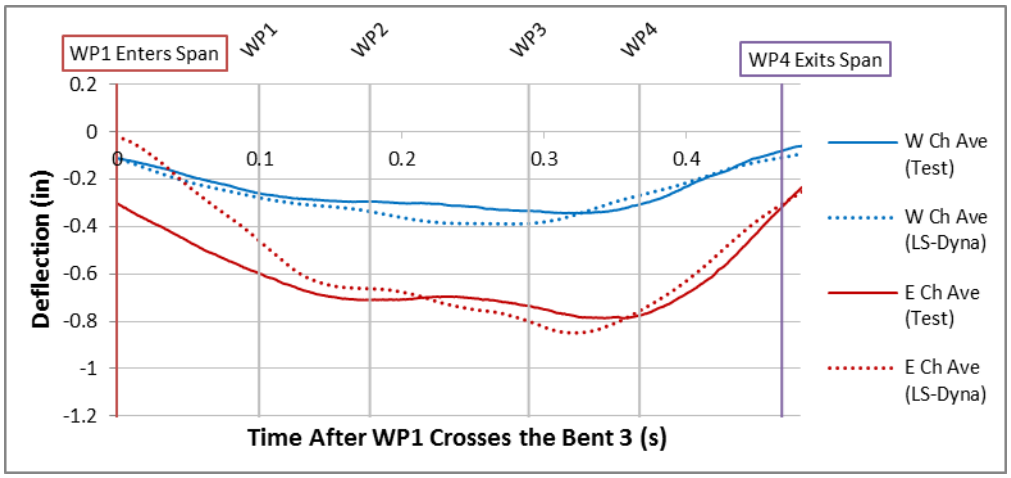

Figure IV-5: Span 7 Test 9 Average Chord Deflection and IC3 Computer Simulation Comparison

The net stringer midspan deflections were calculated from the finite element mesh. The net deflections were determined by subtracting the average of the end deflections from the total midspan deflection. The average chord net deflections from the computer simulation for IC3 is compared to the experimental deflections for span 7 test 9 in Figure IV-6. The figure shows the net deflections for impairment condition three were similar to the experimental measurements for the east and west chords. There is some initial uplift in the east chord of the computer simulation and the maximum deflection is slightly larger than test values. However, the overall patterns and trends seen in the computer simulation are similar to those observed in the experimental analysis. This similarity is useful for obtaining efficient results from the neural network analysis.

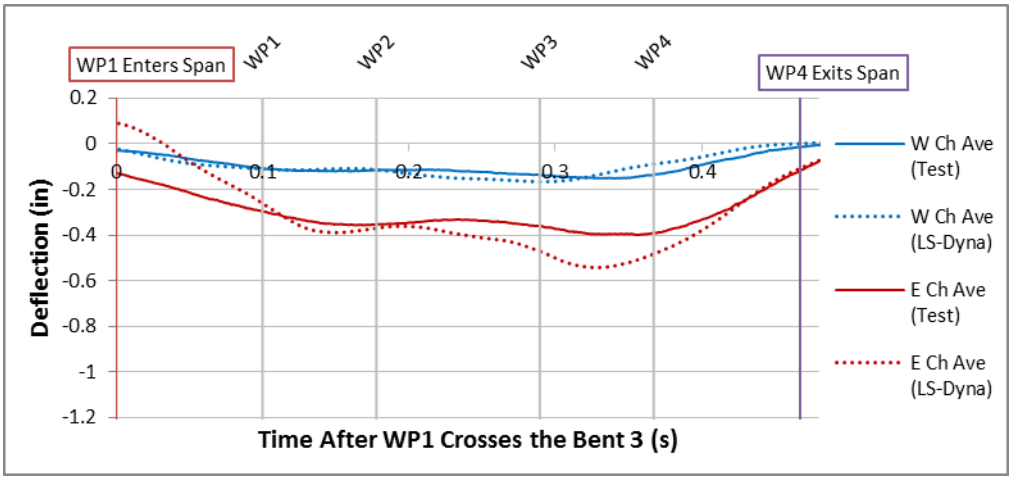

Figure IV-6: Span 7 Test 9 Average Chord Net Deflection and IC3 Computer Simulation Comparison

Four wheel path vertical acceleration plots comparing the east and west chord for an IC 3 50 mph test are given in Figure IV-7. The plots show different shapes and maximum values for each wheel path; however, there are a few noticeable similarities. In this impairment condition, the eastern chord deflects more than the western chord, and this behavior affects the acceleration values. The peak accelerations in the east chord are all greater than the west chord. Additionally, the peaks occur around the midspan while the dips appear approximately at bent locations.

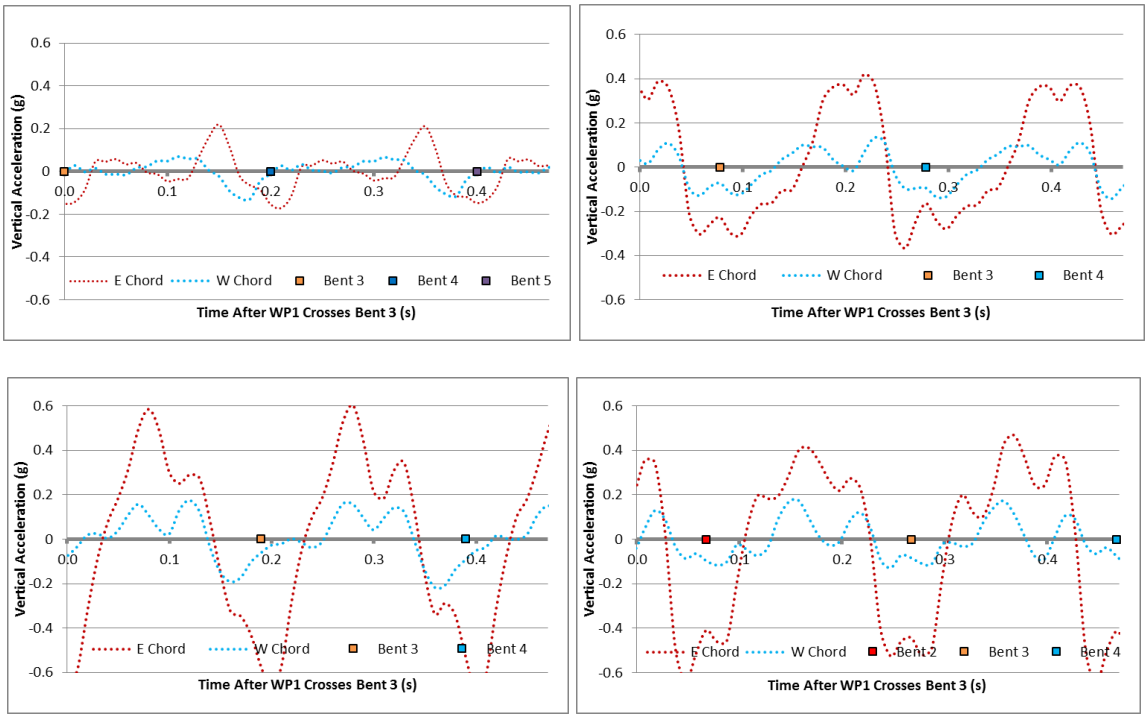

Figure IV-7: 50 MPH IC 3 Vertical Acceleration: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

IV.6 **Computer Simulation Speed Analysis**

In addition to modifying the computer simulation to provide the results of the five impairment conditions, the finite element analysis was also conducted for the five different work train speeds used during experimental testing. Table IV-3 provides the maximum average chord deflection for each test speed and impairment condition for the east chord (chord 2) and the west chord (chord 1). The table shows that the impairment condition has a large influence on the total deflection value for each chord. The maximum deflection values for IC3 and IC5 are nearly twice as large as those from the condition with no damage (IC1). This behavior is seen in each of the five speeds for this simulation.

Table IV-3: Maximum Average Chord Total Deflection for Computer Simulation

Speed (mph)	Impairment Condition	West Chord (in)	East Chord (in)
10	1	-0.394	-0.394
10	2	-0.403	-0.545
10	3	-0.344	-0.774
10	4	-0.545	-0.403
10	5	-0.774	-0.344
20	1	-0.400	-0.400
20	2	-0.411	-0.555
20	3	-0.349	-0.782
20	4	-0.555	-0.411
20	5	-0.782	-0.349
30	1	-0.408	-0.408
30	2	-0.411	-0.561
30	3	-0.351	-0.785
30	4	-0.561	-0.411
30	5	-0.785	-0.351
40	1	-0.408	-0.408
40	2	-0.415	-0.558
40	3	-0.359	-0.786
40	4	-0.558	-0.415
40	5	-0.786	-0.359
50	1	-0.434	-0.434
50	2	-0.449	-0.605
50	3	-0.393	-0.848
50	4	-0.605	-0.449
50	5	-0.848	-0.393

The maximum average chord net deflections were also calculated for each impairment condition and speed. Table IV-4 provides the maximum net deflection values for the east and west chords in the finite element mesh. The results from the net deflections are similar to those seen in the total chord deflection table. The impairment condition has

a larger impact on the net defections compared to the total values. The largest deflections in IC3 and IC5 are nearly three times as large as those in IC1.

Table IV-4: Maximum Average Chord Net Deflection for Computer Simulation

Speed (mph)	Impairment Condition	West Chord (in)	East Chord (in)
10	1	-0.185	-0.185
10	2	-0.212	-0.318
10	3	-0.168	-0.518
10	4	-0.318	-0.212
10	5	-0.518	-0.168
20	1	-0.186	-0.186
20	2	-0.212	-0.318
20	3	-0.168	-0.515
20	4	-0.318	-0.212
20	5	-0.515	-0.168
30	1	-0.183	-0.183
30	2	-0.204	-0.313
30	3	-0.156	-0.510
30	4	-0.313	-0.204
30	5	-0.510	-0.156
40	1	-0.183	-0.183
40	2	-0.207	-0.317
40	3	-0.166	-0.526
40	4	-0.317	-0.207
40	5	-0.526	-0.166
50	1	-0.177	-0.177
50	2	-0.208	-0.324
50	3	-0.166	-0.541
50	4	-0.324	-0.208
50	5	-0.541	-0.166

The deflection ratio for the computer simulation for the maximum midspan chord deflections was calculated for the five test speeds. The ratio is determined by dividing the deflection at a specific speed by the value measured at 10 mph. Since there are five impairment conditions measured, each of these values was averaged to obtain the deflection ratio presented in Table IV-5. The table shows that the deflection ratio increases as the speed of the railcar increases, and this behavior is similar to the experimental deflection ratio results.

Table IV-5: Deflection Ratio for the Computer Simulation

Speed (mph)	West Chord (in)	East Chord (in)
10	1	1
20	1.015	1.015
30	1.023	1.023
40	1.027	1.027
50	1.109	1.109

The maximum acceleration values for each speed, impairment condition, and chord measured during the computer simulation are provided in Table IV-6. The table shows the impairment condition has a significant impact on the maximum acceleration recorded in each chord. The maximum values seen in IC3 and IC5 are approximately three times as large as those seen in the undamaged condition (IC1). This observation is similar to the effect the impairment condition has on the maximum deflection.

Table IV-6: Computer Simulation Maximum Acceleration Comparison

Speed (mph)	Impairment Condition	West Chord (g)	East Chord (g)
10	1	0.007	0.007
10	2	0.008	0.014
10	3	0.006	0.023
10	4	0.014	0.008
10	5	0.023	0.006
20	1	0.027	0.027
20	2	0.032	0.048
20	3	0.024	0.073
20	4	0.048	0.032
20	5	0.073	0.024
30	1	0.044	0.044
30	2	0.056	0.091
30	3	0.043	0.159
30	4	0.091	0.056
30	5	0.159	0.043
40	1	0.079	0.079
40	2	0.089	0.142
40	3	0.079	0.256
40	4	0.142	0.089
40	5	0.256	0.079
50	1	0.127	0.127
50	2	0.155	0.248
50	3	0.138	0.427
50	4	0.248	0.155
50	5	0.427	0.138

Table IV-7 shows the acceleration ratio for each speed used while testing the computer simulation. The acceleration ratio is calculated in the same manner as the deflection ratio. It is used to examine the effect of speed on the maximum acceleration values. The table shows there are significant leaps in the maximum acceleration values

for every incremental speed increase. The 50 mph acceleration ratio, for instance, is nearly 19 times as large as the 10 mph value. This is a substantial difference compared to the modest 10.9 percent difference in deflection ratio values for the same speeds. The table also shows the acceleration values for the east and west chords have identical values. The impairment conditions were designed to allow for different deflection values for each chord. However, impairment conditions were also intended to have some symmetry with IC2 and IC3 having the opposite values of IC4 and IC5. Additionally, the two tables indicate the measured values are highly sensitive to impairment conditions and changes in speeds.

Table IV-7: Acceleration Ratio for Computer Simulation

Speed (mph)	West Chord (g)	East Chord (g)
10	1	1
20	3.59	3.59
30	6.59	6.59
40	11.09	11.09
50	18.84	18.84

IV.7 **Neural Network Training**

The array of artificial neural networks were trained using LS-Dyna finite element analysis results. The wheel path accelerations were the inputs and the average chord midspan deflections were the outputs. To successfully train the neural networks, a number of input and output data sets need to be presented to the array to help the networks determine the patterns and trends comprising the data. This is difficult to accomplish

using only the results from the finite element analysis since the program can be run multiple times, but the produced results will not differ. This behavior is not witnessed in experimental testing. There will be variations within the input and output data sets caused by nature and slight human error (e.g. wind gusts and different train speeds).

The data from the finite element analysis results is post processed to simulate the variability that occurs in field testing. The wheel path vertical acceleration data produced from the LS-Dyna software were manipulated by adding random noise throughout its time history as seen in Figure IV-8. This additional noise simulates the extra signal measured from the experimental test that is caused by the high frequency vibration from moving parts. The noise is not as severe as the results from the field test, but the effect on the output is similar.

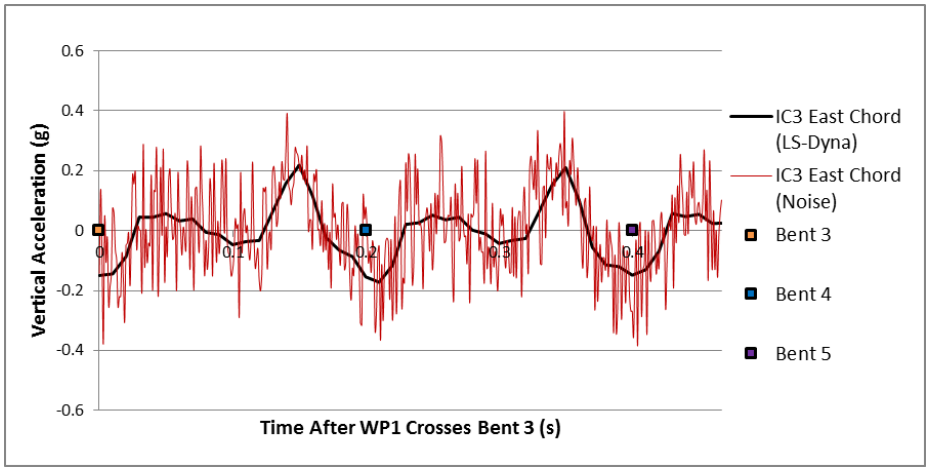

Figure IV-8: Wheel Path 1 IC 3 Vertical Acceleration with Noise

The addition of noise shown in Figure IV-8 makes the results produced from the finite element analysis closer to those from the experimental testing. Since the field test

produced accelerations with large spikes, the data was filtered with a moving average filter to alleviate the effects of the higher frequency vibrations. To make the inputs similar to the field test results, the LS-Dyna vertical accelerations were filtered with a 10 point moving average filter shown in Figure IV-9.

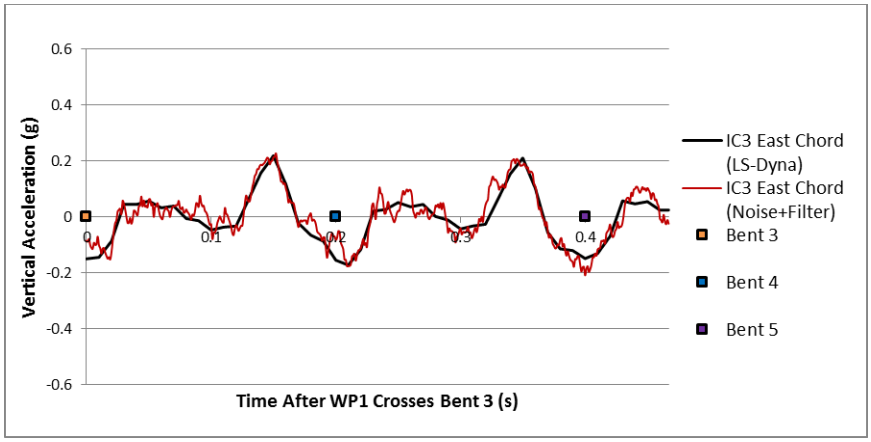

Figure IV-9: Wheel Path 1 IC 3 Vertical Acceleration with Filtered Noise

The wheel path accelerations are not the only experimental parameter that varies during field testing. The average chord midspan deflections also show slight variations during repetitive tests. The variations are different than those experience by the accelerometers, as they tend to increase and decrease uniformly opposed to the extra noise seen in the accelerations. The uniform shift in average chord deflections is shown in Figure IV-10.

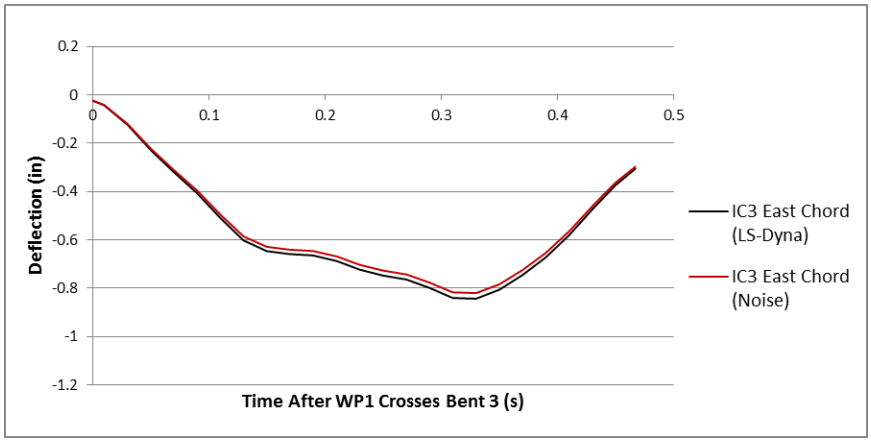

Figure IV-10: Average Chord Midspan Deflection with Uniform Noise

Additional noise was applied to the wheel path accelerations and bridge deflections from the finite element analysis results. The noise was used to help simulate the randomness that may occur during experimental testing. It was also useful for providing additional tests to apply to the artificial neural network training. The process described above was repeated 10 times for each chord and every impairment condition. It was also imposed on all four wheel paths to provide variability in the wheel path accelerations. The extra tests were beneficial in aiding the neural networks in recognizing the patterns in the data and determining the inconsistency that may be associated with each data set.

IV.8 Finite Element Analysis Verification

A number of steps were taken to ensure the results of the finite element analysis were as accurate as possible. In addition to using reasonable material properties and appropriate boundary conditions in the final result, a simple two span system was compared to results from theory and a quasi-static finite element analysis performed on a

program called SAP2000. The output from the theoretical analysis were compared with 2 span LS-Dyna finite element analysis results at a speed of 2 mph to minimize the dynamic effects.

The theoretical deflections were determined by beam theory described in Chapter 1 and were computed using a Matlab script. The deflections for this analysis were analyzed at the load location as it moved along the beam. The wheel path deflections, δ, for the theoretical analysis of a moving load, P, were found using Equation IV-1. The deflections of the rail and stringers attributed to a unit load are indicated by w_r and w_s. Equation IV-2 provides the deflections of a continuous two span simply supported beam with x at the load locations. The reaction at the first support, v_0, and the middle support, R_b, along with the initial slope, Θ_0, are the unknowns in the equation. Equation IV-3 and Equation IV-4 provide the values of the unknowns when the load is located in the first and second spans.

$$\delta = \frac{P * w_r * w_s}{w_r + w_s} \qquad \text{Equation IV-1}$$

$$w(x) = \frac{v_0 x^3}{6EI} + \frac{R_b \langle x - \frac{L}{2} \rangle^3}{6EI} + \theta_0 x \qquad \text{Equation IV-2}$$

$$For\ x \leq \frac{L}{2} \quad \begin{cases} v_0 = P\left(1 - \frac{5x}{2L} - \frac{2x^3}{L^3}\right) \\ R_b = \frac{5P}{L^3}(3L^2 - 4x^2) \\ \theta_0 = -\frac{Px}{48LEI}(7L^2 - 24Lx + 20x^2) \end{cases}$$ Equation IV-3

$$For\ x > \frac{L}{2} \quad \begin{cases} v_0 = \frac{P}{2L^3}(L - x)(L - 2x)(3L - 2x) \\ R_b = -\frac{P}{L^3}(L - x)(L^2 - 8Lx + 4x^2) \\ \theta_0 = -\frac{P}{48LEI}(L - x)(L - 2x)(3L - 2x) \end{cases}$$ Equation IV-4

The boundary conditions were approximated as simply supported for each of these systems, and two 14 foot spans were imposed. This research was implemented before the experimental testing was conducted so approximate values for the work train and the bridge were used. Similar to the actual experiment, the connection between two railcar trucks was used as the applied loading. The distance between wheel paths 1 and 2 along with wheel paths 3 and 4 was approximated as 6 feet. The distance between wheel paths 2 and 3 was assumed to be slightly longer with a distance of 6.67 feet. 78.75 kips was assumed as the load for each of the four wheels in the system. A flexural rigidity of 2.83x106 kip-in^2 and 2.99x106 kip-in^2 was used for each rail and stringer, respectively. These values were also used during the SAP2000 and LS-Dyna finite element analyses.

SAP2000 is a type of finite element software capable of modeling 2D and 3D structures and systems. A 3D two span bridge model was created using this software and is shown in Figure IV-11. The figure shows the rails, cross ties, and stringers were all

meshed during the construction of this model. These elements were connected with stiff spring elements to transfer the loads from the rails to the remaining members. Pin supports were applied as the boundary conditions at the bent locations. A quasi-static analysis was applied to this system, as a load was placed on the rails at the wheel locations and the deflections were recorded. The loads were then moved and the process was repeated to gather numerous data points as the four axles crossed the two spans.

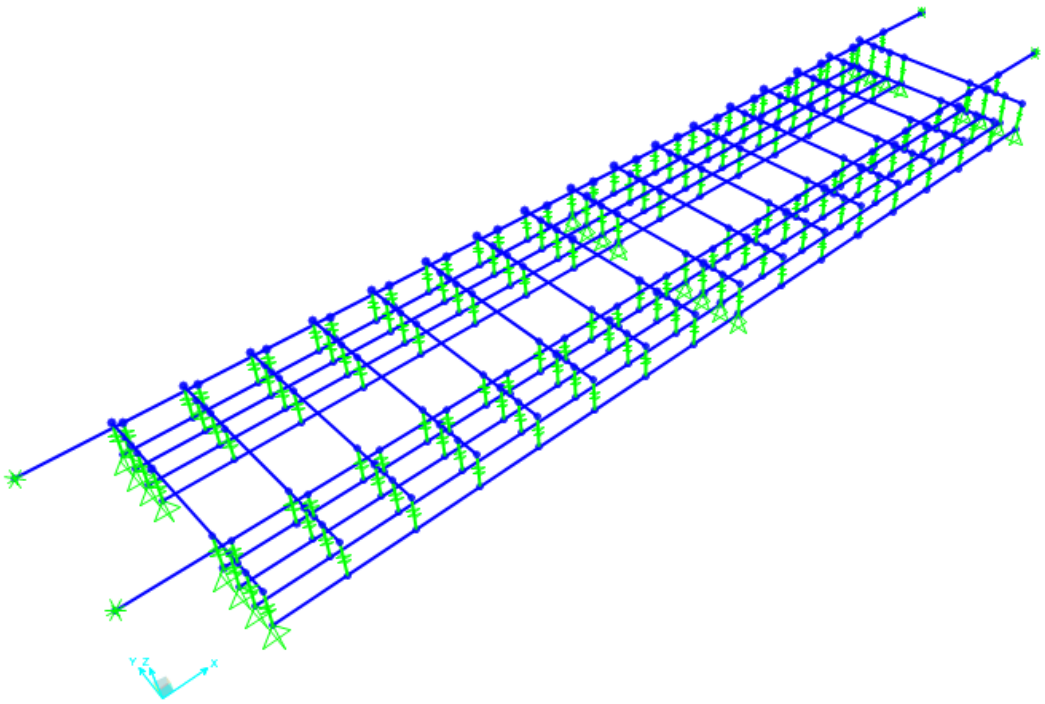

Figure IV-11: SAP2000 Finite Element Analysis Model

LS-Dyna was the third method used for comparing the deflections produced by a simulated passing train. LS-Dyna is capable of performing dynamic finite element analyses of 2D and 3D structures. This method differs from the previous two because this

program can determine the effect of speed on the performance of the system. The Matlab and SAP2000 programs are independent of speed, so they would not produce different results for increased train speeds. For comparison, the LS-Dyna mesh was run with a train speed of 0.4 mph to minimize the dynamic effects from the increased velocity. The two span finite element model is shown in Figure IV-12. The rails, cross ties, and stringers were modeled with beam elements and connected with rigid link elements. The train load and speed was implemented using nodal masses ramped up and down at the wheel locations. Additionally, simply supported boundary conditions were applied to the model.

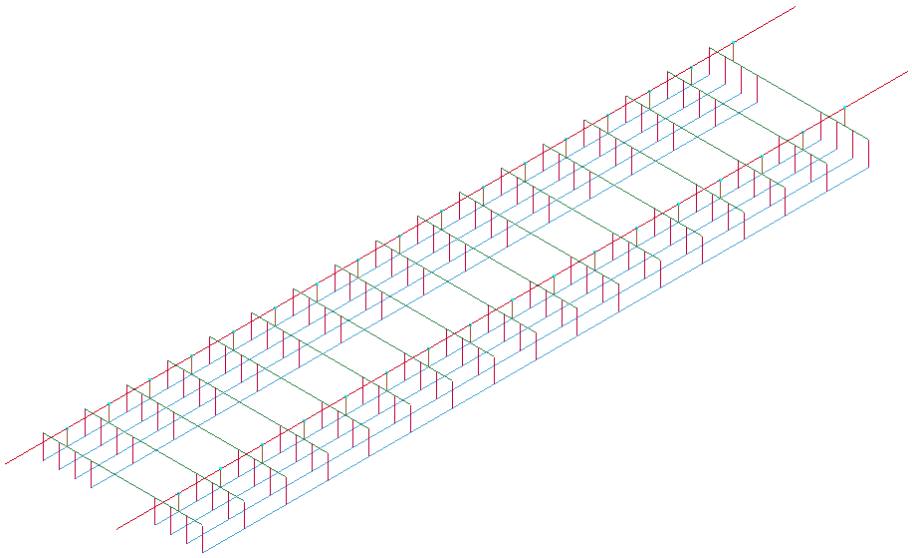

Figure IV-12: LS-Dyna 2 Span Finite Element Analysis Mesh

Each of the three methods of determining the deflections used a two span system with beam elements with the same flexural rigidity. Figure IV-13 shows the deflection comparison of the three methods for the first wheel path traveling in the first bent. Since

the Matlab and SAP2000 methods are not affected by time, the horizontal axis is converted into the wheel's distance from the first bent. The figure shows each of the methods produce and an almost identical deflection plot. They all reach a peak value of approximately 0.25 in when the wheel is positioned slightly past the midspan. There is also a minor hump witnessed when the wheel is 6 feet from the first bent. This corresponds to the time where the second axle would be entering the span.

Figure IV-13: Deflection Comparison for 2 Span Bridge

These results show the original LS-Dyna model provided comparable results to theoretical beam theory and a quasi-static finite element analysis. There were a number of additional modifications made to this original mesh to create the result presented in this research. Some of these include adjusting the mass and speed of the railcar, bridge dimensions, and boundary conditions. However, it is believed these adjustments improved

the final result and made for a more accurate system. This section verified the LS-Dyna program as a trustworthy program that can be useful for modeling both simple and complex structures.

CHAPTER V

RESULTS

V.1 Artificial Neural Network Results

The competitive array of neural networks was trained using the wheel path accelerations and average chord midspan bridge deflections obtained from the LS-Dyna finite element analysis results. The networks were then provided the acceleration data from a field test to estimate the condition of the bridge by recognizing patterns comprising the input signals. Figure V-1 shows a plot depicting the output estimated by the neural network compared to the experimentally measured average chord deflections for bent 7 of test 9. This figure helps depict the accuracy of the array of neural networks. The chord deflections found using the neural network method are similar to those measured during testing. The value of the maximum deflections are comparable, but the locations of the peak show a slight shift. This occurs because the artificial neural networks were trained to recognize the patterns from finite element analysis results. The shift in the deflections from the neural network output was also seen in the computer simulation results.

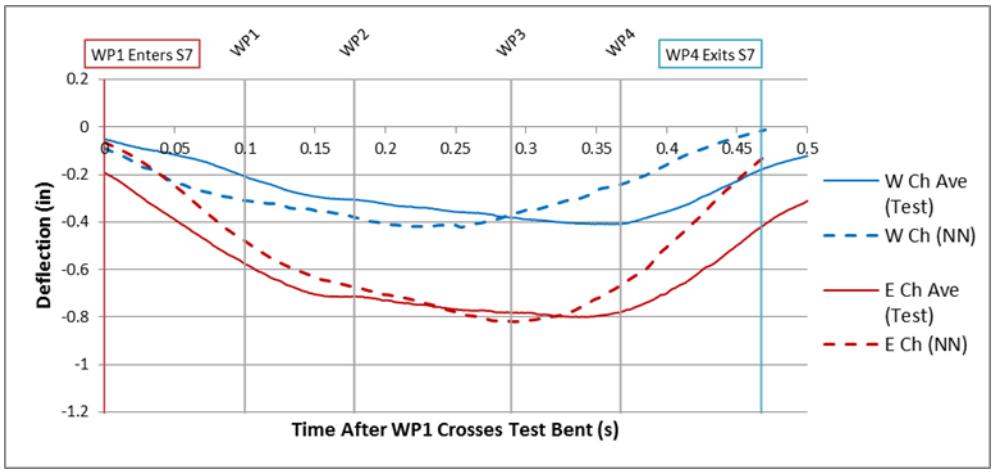

Figure V-1: Span 7 Test 9 Neural Network Output and Experimental Deflection Comparison

The average chord net deflections were also determined from the array of artificial neural networks. The net deflections subtract the compression of the supports from the total deflection. Thus, the net deflection produced smaller values compared to the total deflection. Figure V-2 shows a comparison plot between the measured net deflections and predicted deflections in span 7 test 9. The plots of the deflections produced from the neural networks are similar to the experimentally recorded deflections. The shift in the peak deflections that was seen in the total deflection comparison is also observed in this figure.

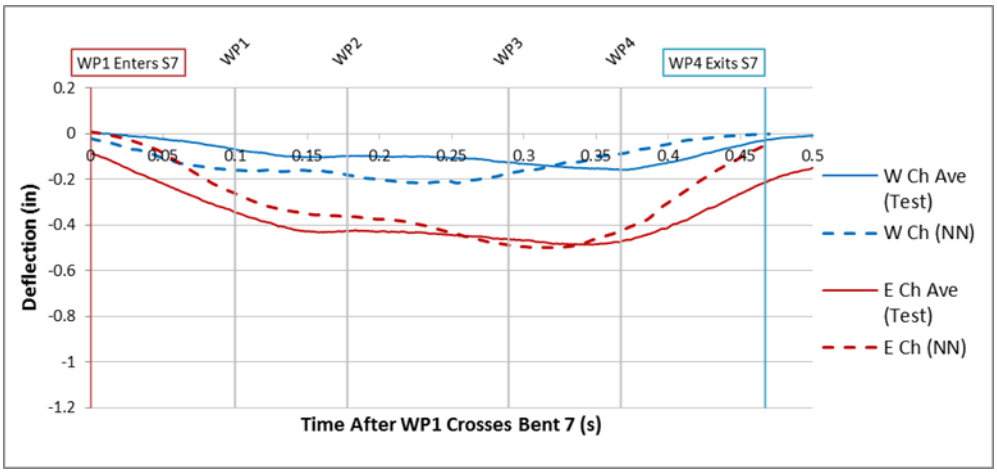

Figure V-2: Span 7 Test 9 Neural Network Output and Experimental Net Deflection Comparison

The maximum average chord deflection values for the neural network and experimental results are summarized in Table V-1. The results are provided for each span, chord, and test that was conducted. The difference between the two values are provided to quantify how close the predicted results were from the experimental values. It also shows whether the array of networks over or under predicted the measured values. The percent difference is calculated to show the error between the two measured values. The string potentiometer displacement transducers have been properly calibrated and were deemed to have accuracy to within 0.01 inches with respect to the actual deflection of the stringers. The neural network deflections are a new method, so most of the error in the percent difference is attributed to the predicted results.

Tests 1 and 2 were measured with a train velocity of approximately 10 mph and the speed incrementally increased to 50 mph. The results in the table show the tests with smaller deflection values tend to have larger errors. Therefore, the error tends to be comparatively larger in the west chord of a span. This is due to the west chord having a

smaller initial deflection, so a small difference between the measured and predicted results will cause a relatively larger percent difference. In every experimental test, the maximum deflection in the east chord was noticeably larger than the west chord. This is a unique characteristic of Bridge 17.14, and it needed to be determined if the proposed method could capture this behavior. Table V-1 shows that for all tests and spans, the neural network method predicted there would be more deflection in the east chord than the west. The difference between the two values is not always as large as those seen in the test, but it is accurate for a majority of the spans.

A comparison between the maximum average chord net deflations for the experiment and neural network method is shown in Table 5-2. The table also includes the difference and percent difference between the results. Similar conclusions can be drawn from the net deflections as those from the total deflections. The difference and percent difference columns show larger error values for early tests with slower speeds. There are a few instances (such as test 1 and span 7) where the maximum deflection in the west chord was predicted to be slightly larger than the east chord. This was not seen in the table showing the total deflection values.

Table V-1: Maximum Chord Total Deflections

Test	Span	Chord	Test (in)	NN (in)	Difference	Percent Difference
1	7	East	-0.86	-0.58	0.28	38.9%
1	7	West	-0.37	-0.54	-0.18	38.6%
1	8	East	-0.73	-0.85	-0.12	15.4%
1	8	West	-0.35	-0.79	-0.44	77.0%
2	7	East	-0.87	-0.70	0.17	21.9%
2	7	West	-0.37	-0.08	0.29	126.8%
2	8	East	-0.74	-0.51	0.23	36.8%
2	8	West	-0.36	-0.42	-0.06	14.3%
3	7	East	-0.88	-0.79	0.09	10.7%
3	7	West	-0.37	-0.49	-0.11	26.7%
3	8	East	-0.79	-0.84	-0.05	5.7%
3	8	West	-0.32	-0.44	-0.12	30.8%
4	7	East	-0.89	-0.83	0.06	6.6%
4	7	West	-0.37	-0.43	-0.06	15.0%
4	8	East	-0.79	-0.77	0.02	2.9%
4	8	West	-0.32	-0.45	-0.13	33.7%
5	7	East	-0.79	-0.86	-0.07	9.1%
5	7	West	-0.40	-0.39	0.00	0.6%
5	8	East	-0.84	-0.85	-0.02	1.9%
5	8	West	-0.32	-0.41	-0.09	24.1%
6	7	East	-0.79	-0.79	0.00	0.3%
6	7	West	-0.40	-0.39	0.01	1.3%
6	8	East	-0.84	-0.78	0.06	7.9%
6	8	West	-0.32	-0.42	-0.10	27.3%
7	7	East	-0.76	-0.82	-0.06	7.4%
7	7	West	-0.41	-0.46	-0.04	10.3%
7	8	East	-0.79	-0.82	-0.03	3.8%
7	8	West	-0.34	-0.46	-0.12	28.9%
8	7	East	-0.77	-0.83	-0.06	7.9%
8	7	West	-0.41	-0.46	-0.04	10.1%
8	8	East	-0.79	-0.87	-0.08	9.4%
8	8	West	-0.35	-0.44	-0.09	23.3%
9	7	East	-0.80	-0.82	0.02	2.7%
9	7	West	-0.41	-0.42	0.01	3.4%
9	8	East	-0.79	-0.83	0.04	5.3%
9	8	West	-0.34	-0.42	0.07	19.0%
10	7	East	-0.81	-0.83	0.02	2.2%
10	7	West	-0.41	-0.43	0.02	5.0%
10	8	East	-0.78	-0.83	0.05	5.7%
10	8	West	-0.34	-0.42	0.08	20.2%

Table V-2: Maximum Chord Net Deflections

Test	Span	Chord	Test (in)	NN (in)	Difference	Percent Difference
1	7	East	-0.52	-0.31	0.21	49.3%
1	7	West	-0.14	-0.38	-0.24	92.3%
1	8	East	-0.36	-0.51	-0.15	34.8%
1	8	West	-0.16	-0.57	-0.41	112.3%
2	7	East	-0.53	-0.37	0.15	34.4%
2	7	West	-0.14	-0.05	0.08	88.0%
2	8	East	-0.36	-0.28	0.08	24.6%
2	8	West	-0.17	-0.23	-0.07	34.9%
3	7	East	-0.53	-0.42	0.12	25.0%
3	7	West	-0.14	-0.28	-0.14	66.7%
3	8	East	-0.40	-0.48	-0.09	19.8%
3	8	West	-0.15	-0.23	-0.08	44.2%
4	7	East	-0.54	-0.48	0.06	11.7%
4	7	West	-0.14	-0.23	-0.09	47.1%
4	8	East	-0.39	-0.45	-0.06	14.2%
4	8	West	-0.14	-0.23	-0.09	46.4%
5	7	East	-0.47	-0.48	-0.01	1.3%
5	7	West	-0.15	-0.21	-0.06	30.8%
5	8	East	-0.43	-0.50	-0.07	14.9%
5	8	West	-0.14	-0.23	-0.09	49.6%
6	7	East	-0.47	-0.48	0.00	1.0%
6	7	West	-0.15	-0.22	-0.06	34.0%
6	8	East	-0.43	-0.50	-0.07	14.9%
6	8	West	-0.13	-0.21	-0.08	45.8%
7	7	East	-0.45	-0.48	-0.03	5.6%
7	7	West	-0.16	-0.23	-0.07	34.3%
7	8	East	-0.40	-0.48	-0.08	18.3%
7	8	West	-0.15	-0.25	-0.09	46.6%
8	7	East	-0.46	-0.47	-0.01	3.0%
8	7	West	-0.16	-0.23	-0.07	37.5%
8	8	East	-0.40	-0.48	-0.08	18.2%
8	8	West	-0.15	-0.23	-0.08	42.5%
9	7	East	-0.49	-0.50	0.01	2.4%
9	7	West	-0.16	-0.21	0.06	30.4%
9	8	East	-0.40	-0.51	0.11	23.6%
9	8	West	-0.15	-0.21	0.06	34.1%
10	7	East	-0.49	-0.50	0.00	0.7%
10	7	West	-0.16	-0.21	0.06	29.7%
10	8	East	-0.40	-0.51	0.11	24.2%
10	8	West	-0.15	-0.21	0.06	33.5%

Table V-3 shows the average error between the measured and predicted chord deflections for each test. The average difference was determined by taking the mean of the absolute value of the difference for each chord measured in Table V-1. The absolute value was used because there were both positive and negative difference values and this could lead to lower average values than actually seen in the testing. The average percent difference is also shown in the table, and was calculated in a similar manner. The neural network method was accurate to within 1/10 of an inch in speeds 20 mph and over for the total deflection table.

Table V-3: Average Error for Total Chord Deflections

Test	Speed (mph)	Ave. Difference (in)	Ave. Perc. Difference
1	10	0.25	42.5%
2	10	0.19	50.0%
3	20	0.09	18.5%
4	20	0.07	14.5%
5	30	0.05	8.9%
6	30	0.04	9.2%
7	40	0.06	12.6%
8	40	0.07	12.7%
9	50	0.04	7.6%
10	50	0.04	8.3%

Table V-4 shows the same columns as the table above, but was calculated using the net chord deflections given in Table V-2. This table follows a similar pattern that occurred in the preceding tables. The calculated error tends to decrease as the speed of the work train increased. The speeds with the lowest errors happened when the rail vehicle

was traveling 30 and 50 mph. It was accurate to within 1/10 of an inch in speeds in excess of 30 mph in the net deflection table.

Table V-4: Average Error for Net Chord Deflections

Test	Speed (mph)	Ave. Difference (in)	Ave. Perc. Difference
1	10	0.25	72.2%
2	10	0.10	45.5%
3	20	0.11	38.9%
4	20	0.07	29.9%
5	30	0.06	24.2%
6	30	0.05	23.9%
7	40	0.07	26.2%
8	40	0.06	25.3%
9	50	0.06	22.6%
10	50	0.06	22.0%

V.2 **Approximate Stringer Midspan Deflection**

The neural networks are trained to produce a single average deflection plot for each chord. However, there are four stringers in each chord, and they all have different values throughout the time history. This can be remedied by comparing the deflection values of the stringers at a specific time. These deflections can be plotted with the distance from the first stringer as the horizontal axis. The plot for the test deflections are given in Figure V-3 and shows the stringers deflecting in a nearly linear manner.

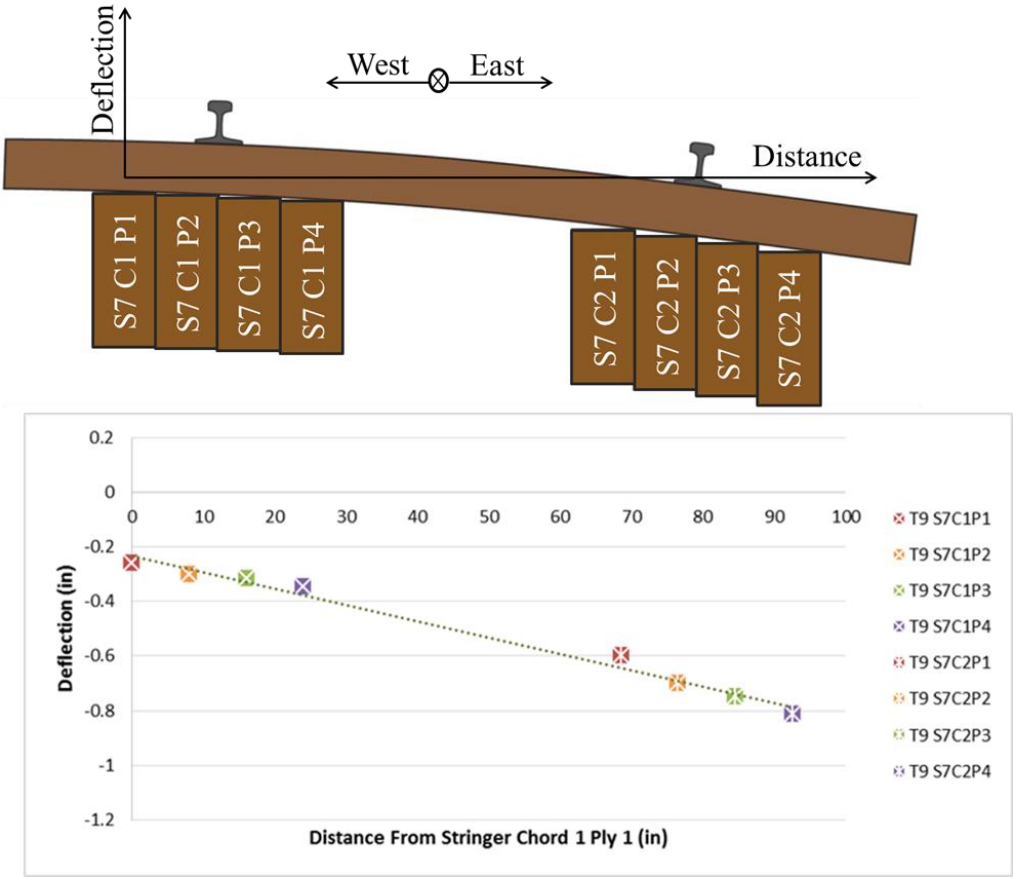

Figure V-3: Span 7 Test 9 Total Midspan Deflection Values at Time = 0.178s

A similar method can be used to determine the approximate stringer deflections based on the average chord midspan deflection. The array of artificial neural networks give the average deflections for the east and west chords as the output. These two plots can be used to determine a linear relationship based on the stringer location. A plot showing the average chord midspan deflection values for the east and west chords at a time equal to 0.33 seconds is shown in Figure V-4. This corresponds approximately to the time at which the maximum deflection from the work train occurred. The figure shows a linear deflection relationship can be determined from the two values located on the plot.

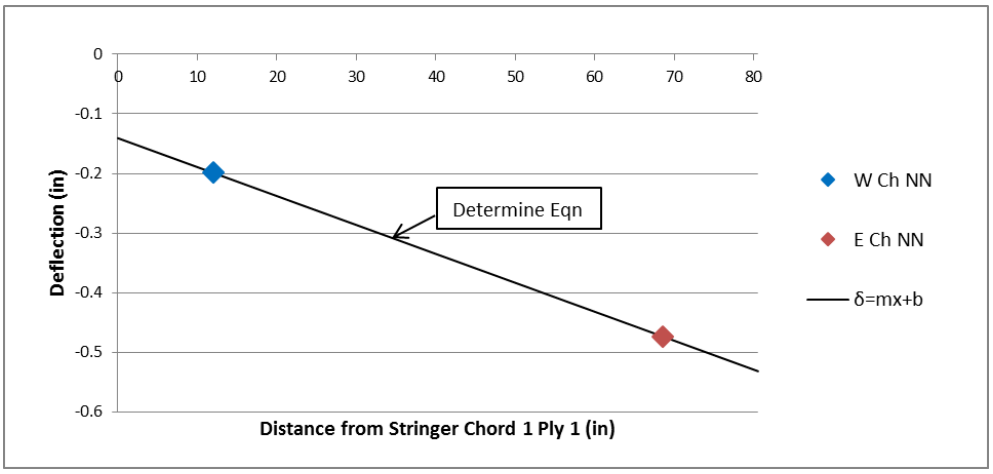

Figure V-4: Average Chord Midspan Deflection Values at Time = 0.33 s

Since the distance between each stringer is known, the linear relationship can be used to approximate the values of the stringers based on the average chord deflections. The average chord deflections are assumed to be located in between the second and third ply in each chord. The calculated stringer deflections are provided in Figure V-5.

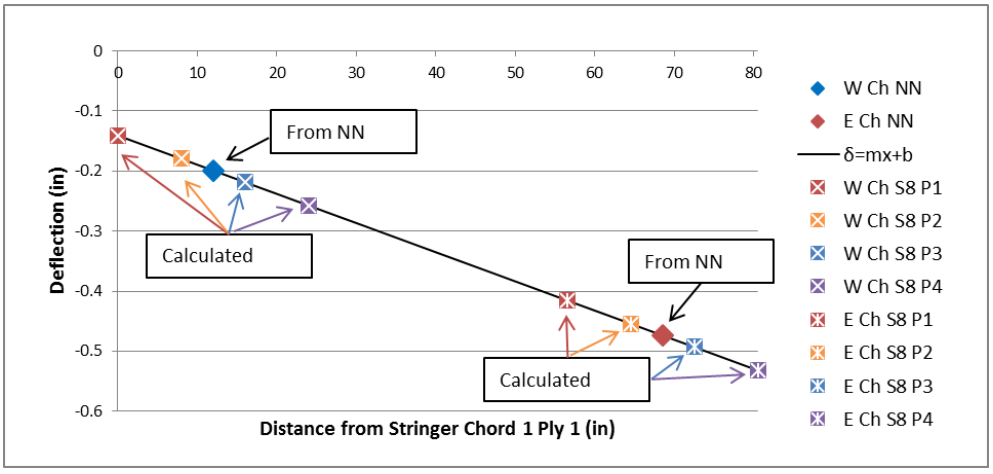

Figure V-5: Approximate Stringer Deflection at Time = 0.33 s

The previous figures show the process for calculating the approximate stringer deflections for a specific time. The assumption of a linear cross sectional stringer displacement can be applied throughout the entire time history. Figure V-6 shows a comparison between the experimentally measured and predicted stringer deflections. This allows for a direct comparison between the values measured in the field to those obtained from the artificial neural network analysis.

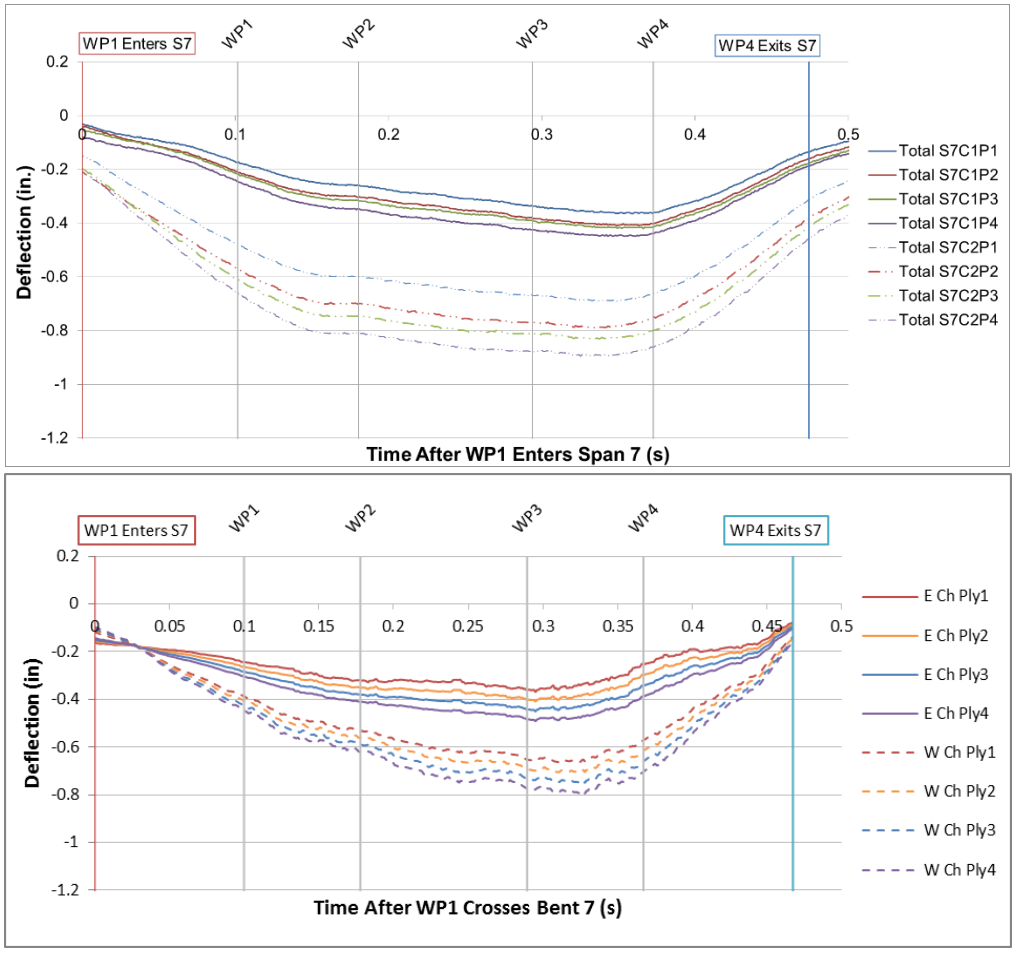

Figure V-6: Span 7 Test 9 Stringer Time History Comparison: Experimental (Top), Predicted (Bottom)

CHAPTER VI

CONCLUSIONS

VI.1 Summary

This research developed a preliminary technology to autonomously detect structural impairments in timber railroad bridges using data gathered from a rail vehicle crossing the bridges. Developing and testing this technology required experimental tests, finite element analyses, and artificial neural networks.

An open deck timber trestle bridge in service was dynamically tested with a work train consisting of two train locomotives followed by two loaded railcars. Ten crossings were made in the southbound directions with the speed of the vehicle ranging from 10 mph to 50 mph. The behavior of span 7 and span 8 of the bridge were recorded with string potentiometers positioned at the midspan and near the supports of each stringer. Additionally, accelerometers on each side of four axles captured the wheel motion of the east and west sides of the train. Lastly, wheel path position devices located above the first wheel path of the rail vehicle and on posts mounted above the bents tracked the progress of the trucks along the tracks, provided the train speed, and synchronized the data signals from the two data acquisition devices.

A finite element analysis using LS-Dyna computer software modeled the test bridge and imposed various damage scenarios to provide theoretical acceleration and deflection data. The mesh included the rails, cross ties, and stringers modelled as beam elements and connected by rigid links. Discrete beam elements with various stiffnesses

approximated the bents. These elements were fixed at one end and attached to the stringer elements at the other. Ramping the magnitude of nodal masses up and down along the rails at the wheel locations simulated the load from the train crossing the bridge. Multiple simulations modeled loading from the train speeds used during the experimental test. Impairment conditions ranged from no damage to excessive damage in one chord and little damage in the other. Deflections from the center span of the mesh provided the total and net midspan deflections. The acceleration on the east and west chords of the bridge at the wheel locations provided the wheel path accelerations of the four axles.

The finite element analysis provided training data for an array of artificial neural networks. Artificial neural networks are a pattern recognition technology capable of correlating sets of input and output data. The neural networks were trained using the finite element results with a method called back propagation. Back propagation modifies the synaptic weights of the system to determine patterns and trends relating the input data, i.e. wheel path accelerations, and output data, i.e. average chord midspan deflections. After determining the synaptic weights, the network was tested by supplying it with the vertical wheel path accelerations from the field test and comparing its results to the measured test deflections. The networks were not provided the test deflections during the training nor testing process.

VI.2 Conclusions

- Asymmetric deflection occurs when there is more significant damage in one chord than the other. The test bridge deflected roughly twice as much on the east side as the west side due to split stringer observed in the east chord but not the west.
- The total and net deflection ratio results increase for faster train speed. However, the maximum deflection between the fastest and slowest speed differs by less than 0.05 inches for the total deflection and 0.03 inches for the net deflection, a modest difference compared to the effects of impairment and vehicle load.
- A 100 point moving average filter applied to the wheel path accelerations is able to reduce the effects of high frequency vibrations and provide the motion of an axle over the bridge for neural network training.
- Vehicle speed and bridge impairments influence the magnitude of the wheel path accelerations. The accelerations increase by approximately 475% from the 10 mph to the 50 mph. The accelerations over the eastern chord, the chord with split stringers, were on average 31% larger than the west chord.
- Computer simulations can represent the behavior of a timber bridge under train loading. A simulated impairment condition with a 75% reduction in the flexural rigidity in the eastern stringers and no damage in the western stringers produced results comparable to the field measurements.
- The use of a competitive array of neural networks improves the efficiency of the detection method compared to using an individual artificial neural network.

Training the system with a multiple networks of various architectures produced more accurate results than a single network.

- Training artificial neural networks with wheel path acceleration signals from a finite element model with large noise filtered with a moving average improves the accuracy of the system.
- Neural networks can analyze bridges with asymmetric deflections. The test bridge is unique in that it had greater damage in the east chord than the west. The neural network array was able to predict larger deflection in the east chord than the west for a majority of the tests and spans.
- Neural networks can approximate both the net and total deflections. With 1/10 of an inch used as an acceptable error, the networks correctly estimated 29 out of 40 (72.5%) of the total deflection tests, and 31 out of 40 (77.5%) of the net deflection tests.
- The accuracy of the system improved at faster speeds. However, 30 mph was the optimal speed for this analysis as 100% of its estimated deflection values were within 0.1 inches of the experimental measurements.

VI.3 **Potential Future Work**

Testing showed that experimental results, finite element analysis, and artificial neural networks can be used together to create a system capable of predicting the behavior of a timber railway bridge based on the motion of a railcar crossing the bridge. Additional research should be conducted to further validate the proposed technology. Future work

can also improve the accuracy and efficiency of this technology, or extend it to new applications.

Testing different bridges with various span lengths and damage levels will validate the neural network output against multiple bridge inputs. Additionally, since this research investigated only four spans of the test bridge, testing multiple spans with a vehicle traveling at a predetermined speed will further justify the use of this technique in practice. Since this research indicated that the accuracy improved with speed, the maximum safe operating speed could be used for future tests.

A 100 point moving average filter smoothed the accelerometer data. Determining the window size involved adjusting the value and recording the effect on the resulting acceleration plot. A moving average produces the same results as a low pass finite impulse response filter with a rectangular window of the same size. Various filters can be investigated to determine the best choice for this application.

This research employed a variable capacitance accelerometer because their ability to measure the motion of objects subjected to vibration. However, the vibrations from moving parts measured during the test proved to be lower than expected, allowing for a more sensitive accelerometer for future work.

Constructing the neural network architectures is one of the most crucial steps to ensure accurate and efficient results. Future work should focus on evaluating each type of network and establishing criteria for the neural network architecture. This includes analyzing the number of hidden layers, the amount of neurons in each layer, and stopping criteria.

The neural network output produced the average chord midspan deflections. The average deflection was chosen because it could be used for bridges with various configurations and stringer locations. However, there may be a need for the deflection of each stringer to be determined from the neural network output. In this case, future research could focus on refining the training method to produce the deflections of each stringer as an output. This method would need to be evaluated to ensure it would produce accurate results for a number of timber bridge configurations.

REFERENCES

AAR. (2016). Rail Intermodal Keeps America Moving. *Association of American Railroads*, 1-5.

Babcock, S. A., Gutkowski, R. M., Charlie, W. A., & Balogh, J. (2006). *Support Motion Effects in a Timber Trestle Bridge: Physical and Analytical Modeling.* (Master of Science), Colorado State University, Fort Collins, Colorado.

Bishop, C. M. (1994). Neural Networks and Their Applications. *Review of Scientific Instruments*, *65*(6), 1803.

Emerson, R. N., Pollock, D. G., McLean, D. I., Fridley, K. J., Ross, R. J., & Pellerin, R. E. (1998). *Nondestructive Tesing of Large Bridge Timbers*. Paper presented at the Eleventh International Symposium on Nondestructive Testing of Wood, Madison, WI.

Farley, B. G., & Clark, W. A. (1954). Simulation of Self-Organizing Systems by Digital Computer. *IRE Trans. on Infor. Theory, PGIT-4*, 76-84.

Forsling, H. P., Balogh, J., Gutkowski, R. M., & Radford, D. W. (2012). Load tests of damaged railroad timber stringers repaired by shear spiking. *Proceedings of the ICE - Structures and Buildings*, *165*(6), 309-321.

FRA. (2010). Bridge Safety Standards. *Federal Register*, *75*(135), 41282-41309.

Gutkowski, R. M., Shigidi, A. M. T., & Tran, A. V. (2003). *FIELD INVESTIGATION OF A STRENGTHENED TIMBER TRESTLE RAILROAD BRIDGE.* Colorado State University, Fort Collins, CO.

Hagan, M. T., Demuth, H. B., & Beale, M. H. (1996). *Neural Network Design*. Boston, MA: PWS Publishing Company.

Haykin, S. (1999). *Neural Networks- A Comprehensive Foundation*. Singapore: Pearson Education.

Hebb, D. O. (1949). *The Organization of Behavior: A Neuropshychological Theory*. New York, NY: John Wiley & Sons.

Humar, M., Bučar, B., & Pohleven, F. (2006). Brown-rot decay of copper-impregnated wood. *International Biodeterioration & Biodegradation, 58*(1), 9-14. doi:10.1016/j.ibiod.2006.03.003

Hyland, D. C., & Davis, L. D. (2002). Toward Self-Reliant Control for Adaptave Structures. *Acta Astronautica, 51*(1-9), 89-99.

Martland, C. D. (2013). Introduction of Heavy Axle Loads by the North American Rail Industry. *Journal of the Transportation Research Forum, 52*(2), 103-125.

Morison, A. M., Evensen, H. A., Ligon, J. B., Erickson, J. R., Ross, R. J., & Forsman, J. W. (2002). *TIMBER BRIDGE EVALUATION: A GLOBAL NONDESTRUCTIVE APPROACH USING IMPACT GENERATED FRFs*. Paper presented at the 20th International Modal Analyasis Conference.

Orsak, J. P. (2012). *THEORETICAL STRUCTURAL IMPAIRMENT DETECTION SYSTEM FOR TIMBER RAILWAY BRIDGES.* (Master of Science), Texas A&M University, College Station, TX.

Orsak, J. P., & Story, B. A. (2013). Vehicle-Bourne Autonomous Railroad Bridge Impariment Detection Systems. *Structural Health Monitoring 2013: a Roadmap to Intelligent Structures - Proceedings of the 9th International Workshop On Structural Health Monitoring, 2*, 1713-1720.

Peterson, M. L., & Gutkowski, R. M. (1997). Evaluation of the structural integrity of timber bridges. *NDT&E International, 32*, 43-48.

Priddy, K. L., & Keller, P. E. (2005). *Artificial Neural Networks: An Introduction* (Vol. Tutorial Texts in Optical Engineering). Bellingham, WA: SPIE Press.

Radford, D. W., Peterson, M. L., & VanGoethem, D. (2000). *COMPOSITE REPAIR OF TIMBER STRUCTURES.* Colorado State University, Fort Collins, Co.

Rakoczy, A. M., & Otter, D. (2015a). Bridge Condition Evaluation Using Track Geometry Systems. *Technology Digest*(TD-15-007), 1-4.

Rakoczy, A. M., & Otter, D. (2015b). Bridge Condition Evaluation Using Track Modulus Measuring Systems. *Technology Digest*(TD-15-006), 1-4.

Rakoczy, A. M., Otter, D. E., Malone, J. J., & Farritor, S. (2016). Railroad Bridge Condition Evaluation Using Onboard Systems. *Journal of Bridge Engineering, 10.1061/(ASCE)BE.1943-5592.0000881*(04016044), 1-12.

Rakoczy, A. M., Shu, X., & Otter, D. (2015a). Vehicle/Bridge Interaction Modeling and Validation - Part 1. *Technology Digest*(TD-15-004), 1-4.

Rakoczy, A. M., Shu, X., & Otter, D. (2015b). Vehicle/Bridge Interaction Modeling and Validation - Part 2. *Technology Digest*(TD-15-005), 1-4.

Ritter, M. A. (1990). *Timber Bridges:Ddesign, Construction, Inspection, and Maintenance*. Washington, DC: USDA Forest Service.

Rosenblatt, F. (1958). The Perceptron: A Probablistic Model for Information Storage and Organization in the Brain. *Psychological Review, 65*(6), 386-408.

Story, B. A., & Fry, G. T. (2014). A Structural Impairment Detection System Using Competitive Arrays of Artificial Neural Networks. *Computer-Aided Civil and Infrastructure Engineering, 29*(3), 180-190.

Werbos, P. J. (1974). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences.* (Doctor of Philosophy), Harvard University.

Wipf, T. J., Ritter, M. A., & Wood, D. L. (2000). *Evaluation and Field Load Testing of Timber Railway Bridge*. Paper presented at the Fifth International Bridge Engineering Conference, Tampa, FL.

Supplemental Sources Consulted

AREMA. (2015). *2015 Manual for Railway Engineering*. American Railway Engineering and Maintenance-of-Way Association. Landover, MD.

Barai, S. V., & Pandey, P. C. (1996). Time-Delay Neural Networks in Damage Detection of Railway Bridges. *Advances in Engineering Software, 28*, 1-10.

Catbas, F. N., Gul, M., Zaurin, R., Terrell, T., Dere, Y., Ansley, M. H., . . . Grimmelsman, K. A. (2009). Structural Health Monitoring of Bridges:

Fundamentals, Application Case Study and Organizational Considerations. *Structures Congress*, 108-118.

Chang, C. C., Wang, M. L., & Xu, Y. G. (2000). Structural Damage Detection Using an Iterative Neural Network. *Journal of Intelligent Material Systems and Structures, 11*(1), 32-42.

Fang, X., Luo, H., & Tang, J. (2005). Structural damage detection using neural network with learning rate improvement. *Computers & Structures, 83*(25-26), 2150-2161.

Fausett, L. (1994). *Fundamentals Of Neural Networks*. Englewood Cliffs, NJ: Prentice Hall.

FHWA. (2015). *Deficient Bridges by State and Highway System*. Federal Highway Administration. Office of Bridge Technology.

Lee, J. W., Kirikera, G. R., Kang, I., Schulz, M. J., & Shanov, V. N. (2006). Structural health monitoring using continuous sensors and neural network analysis. *Smart Materials and Structures, 15*(5), 1266-1274.

LS-PrePost. (2014) (Version 4.2): Livermore Software Technology Corporation.

MATLAB. (2012) (Version 7.14.0.739): The Mathworks Inc.

Moreu, F., Li, J., Jo, H., Kim, R. E., Scola, S., Spencer, B. F., & LaFave, J. M. (2015). Reference-Free Displacements for Condition Assessment of Timber Railroad Bridges. *Journal of Bridge Engineering, 21*(2).

Moreu, F., & Nagayama, T. (2008). *Use of wireless sensors for timber trestle railroad bridges health monitoring assessment*. Structures Congress. ASCE.

Rachal, L. N. (2016). *Determination of Bent-Cap and Stringer Deflections for Timber Railway Bridges Under Live Load.* (Master of Science), Texas A&M University, College Station, TX.

SAP2000. (2015) (Version 18): Computers and Structures Inc.

Sohn, H., Farrar, C. R., Hemez, F. M., Shunk, D. D., Stinemates, D. W., Nadler, B. R., & Czarnecki, J. J. (2004). *A Review of Structural Health Literature: 1996-2001*. Technical Report. Los Alamos National Laboratory.

Story, B. A. (2012). *Structural Impairment Detection Using Arrays of Competitive Artificial Neural Networks.* (Doctor of Philosophy), Texas A&M University, College Station, TX.

Uppal, A. S., Fry, G. T., & Byers, W. G. (2000). *Fatigue Strength of Solid Sawn Timber Railroad Bridge Stringers*. Research Publication TD-00-019, Association of American Railroads/ Transportation Technology Center: Pueblo, CO.

VDOT. (2014). *State of the Structures and Bridges Report: Fiscal Year 2014*. Richmond, VA: Virginia Department of Transportation.

Yeung, W. T., & Smith, J. W. (2005). Damage Detection in Bridges Using Neural Networks for Pattern Recognition of Vibration Signatures. *Engineering Structures, 27*(5), 685-698.

APPENDIX A

SUPPLEMENTAL FIGURES

A.1 Accelerometer Data

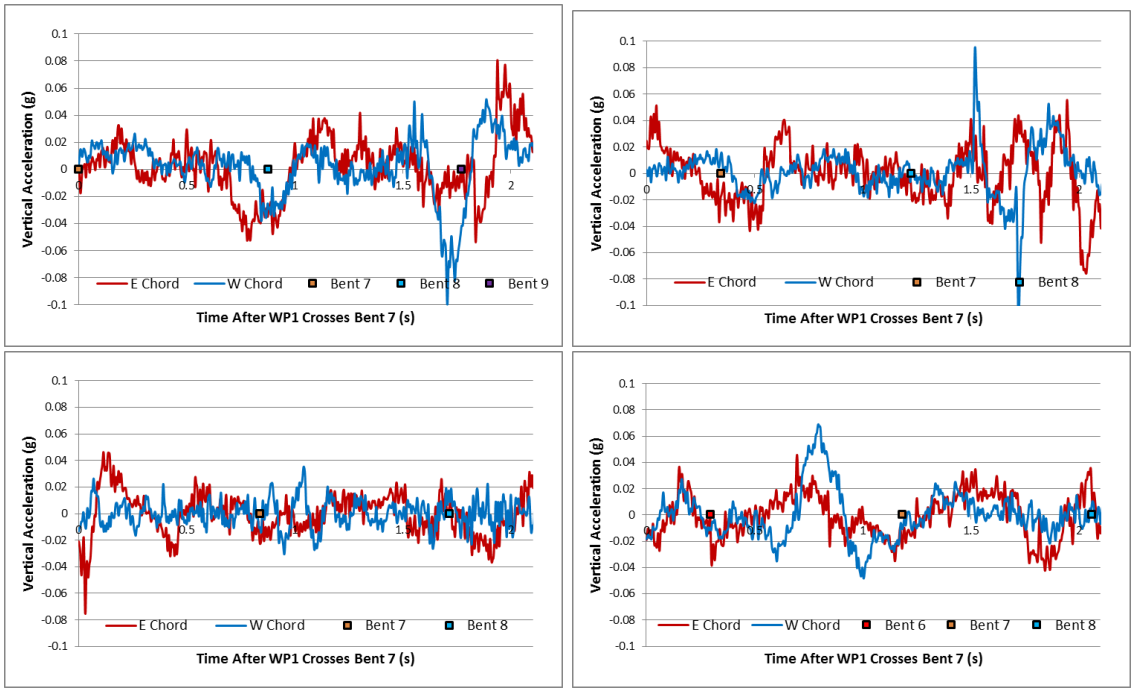

Figure A-1: Span 7 Test 1 Vertical Acceleration: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

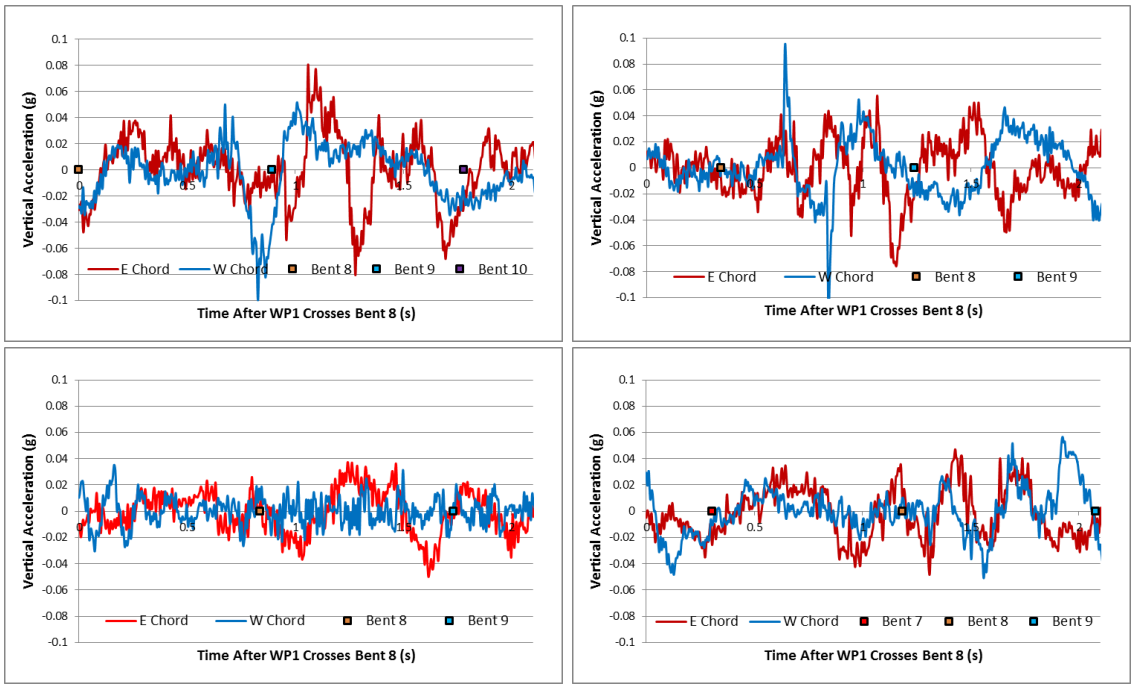

Figure A-2: Span 8 Test 1 Vertical Acceleration: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

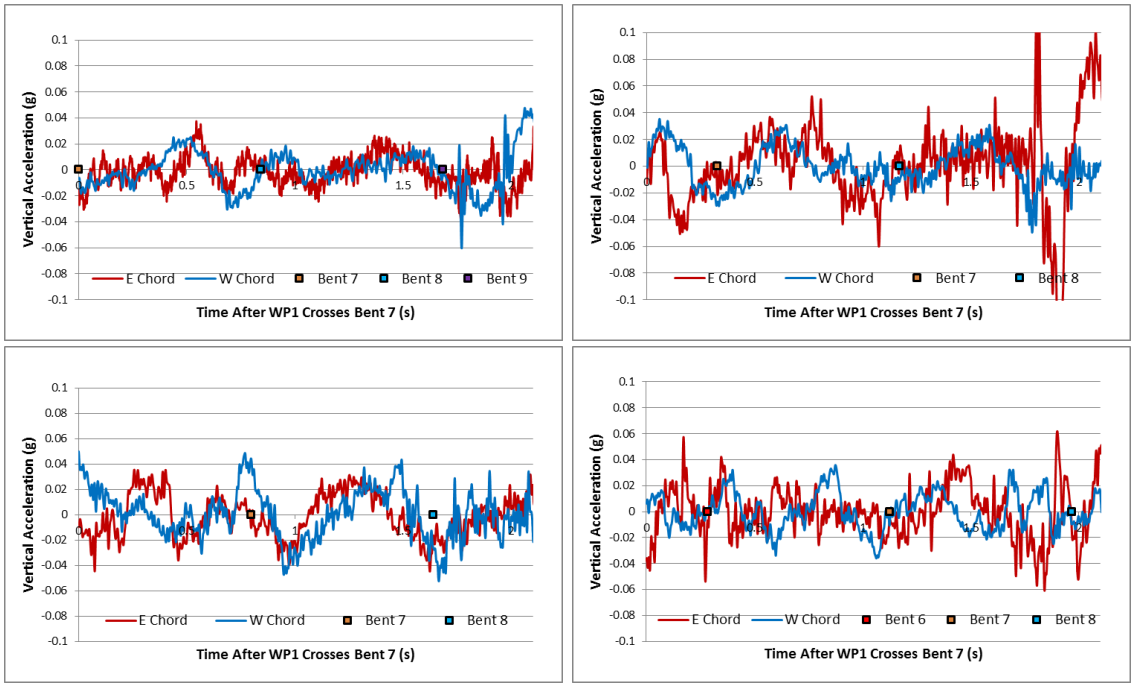

Figure A-3: Span 7 Test 2 Vertical Acceleration: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

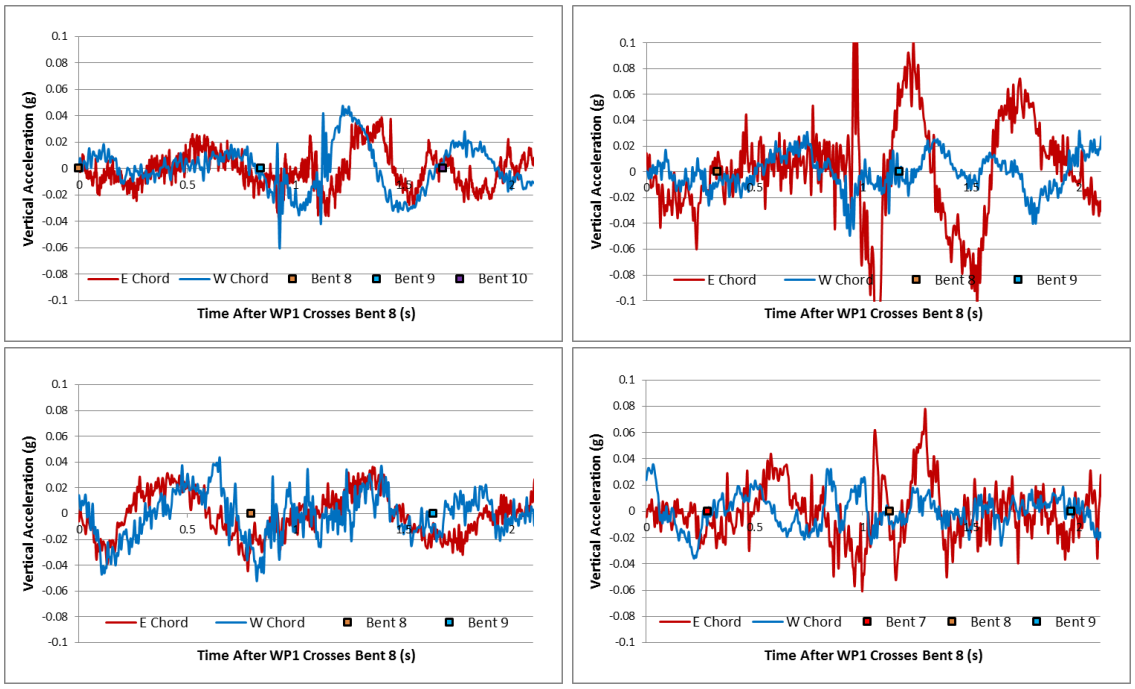

Figure A-4: Span 8 Test 2 Vertical Acceleration: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

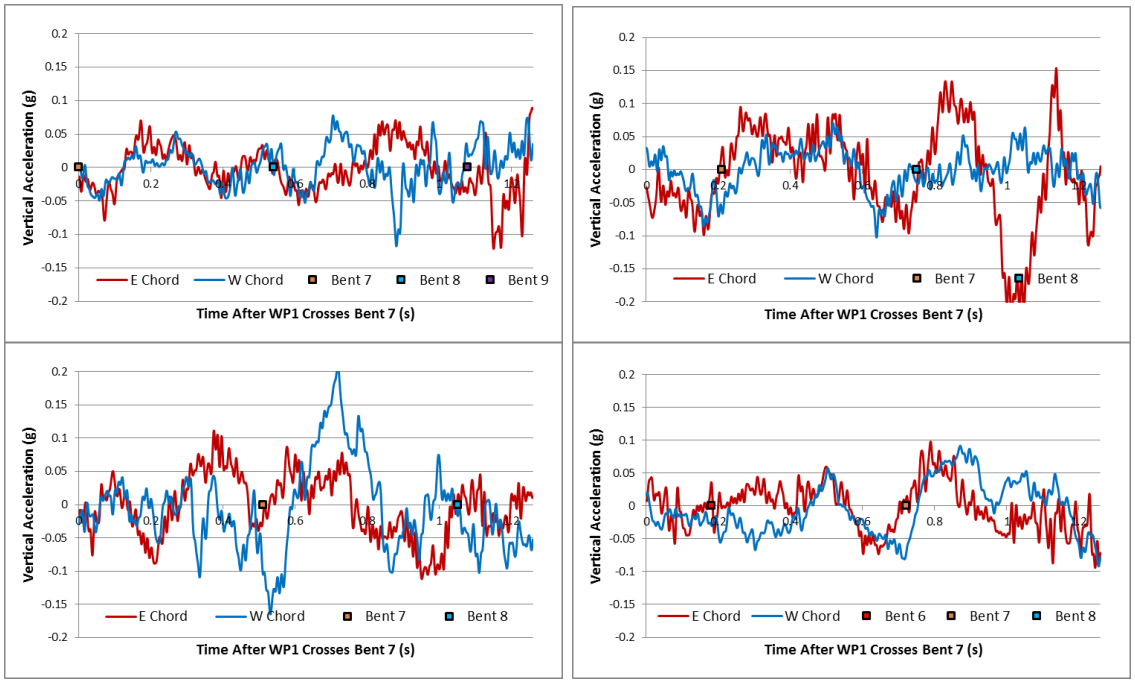

Figure A-5: Span 7 Test 3 Vertical Acceleration: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

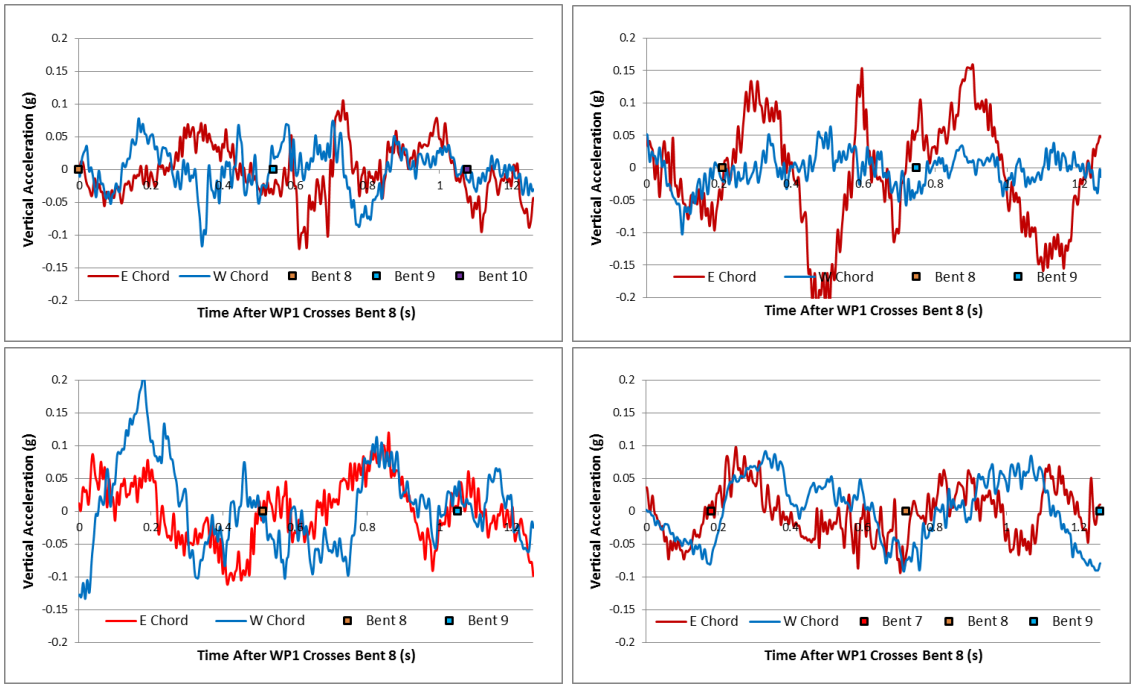

Figure A-6: Span 8 Test 3 Vertical Acceleration: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

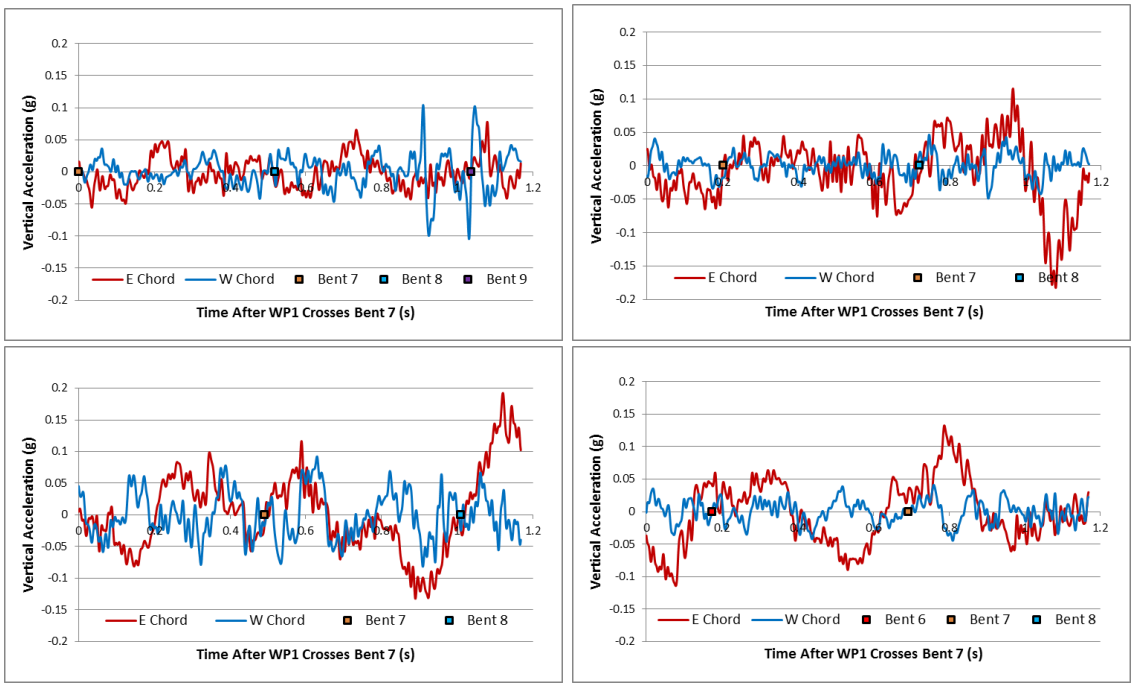

Figure A-7: Span 7 Test 4 Vertical Acceleration: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

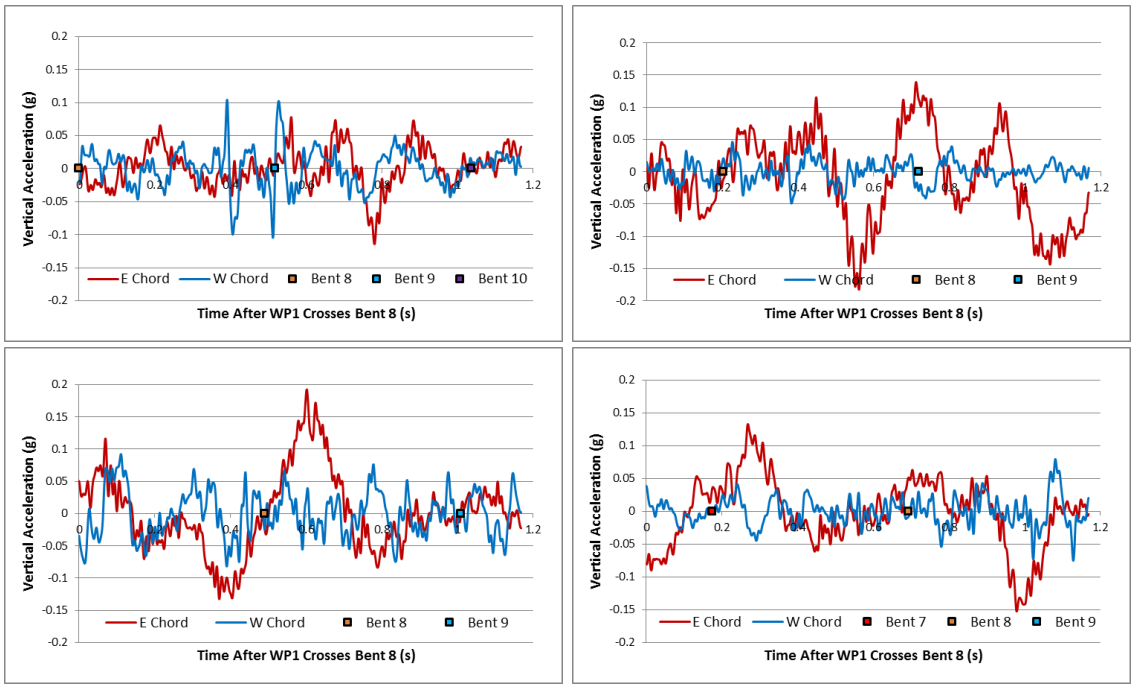

Figure A-8: Span 8 Test 4 Vertical Acceleration: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

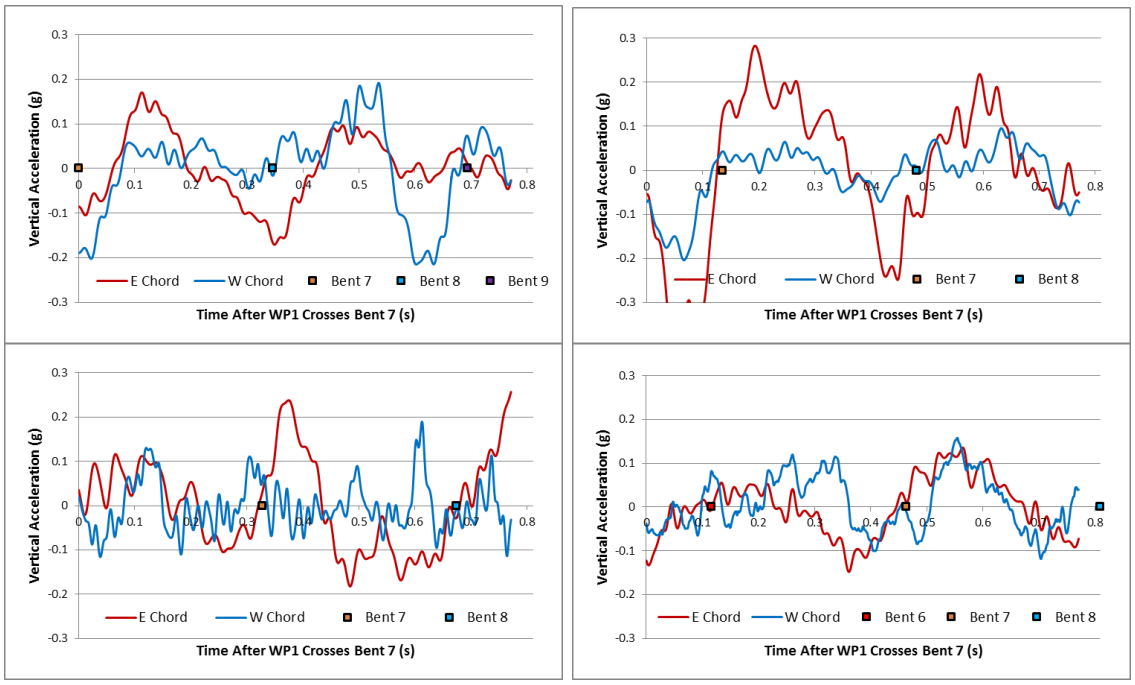

Figure A-9: Span 7 Test 5 Vertical Acceleration: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

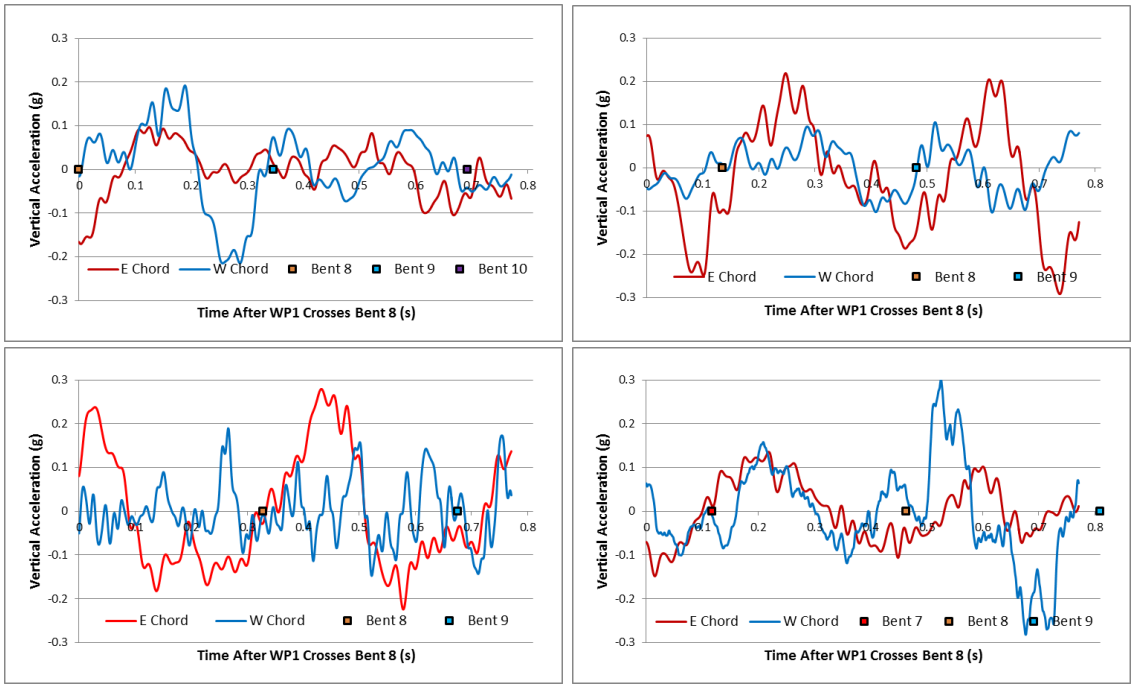

Figure A-10: Span 8 Test 5 Vertical Acceleration: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

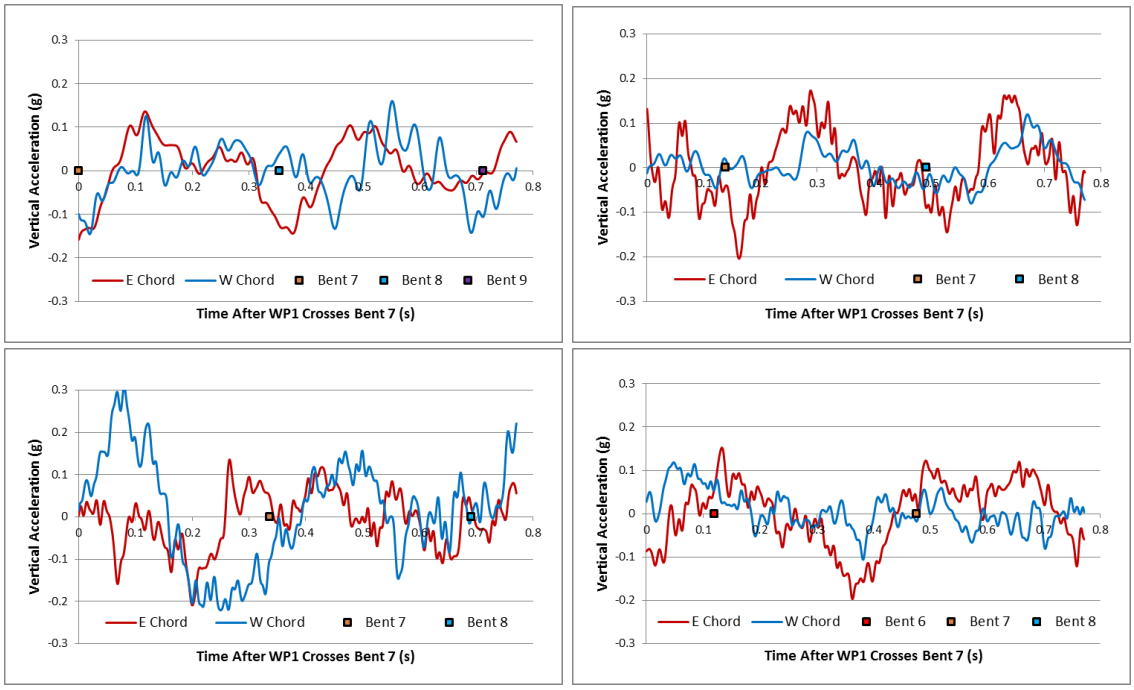

Figure A-11: Span 7 Test 6 Vertical Acceleration: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

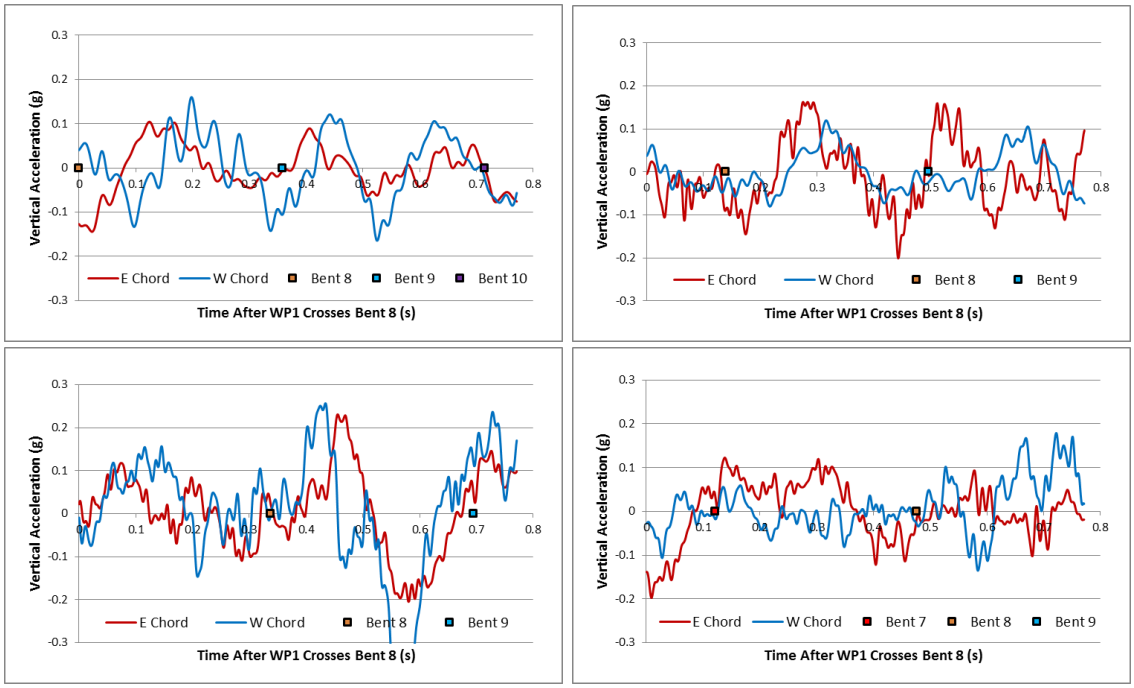

Figure A-12: Span 8 Test 6 Vertical Acceleration: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

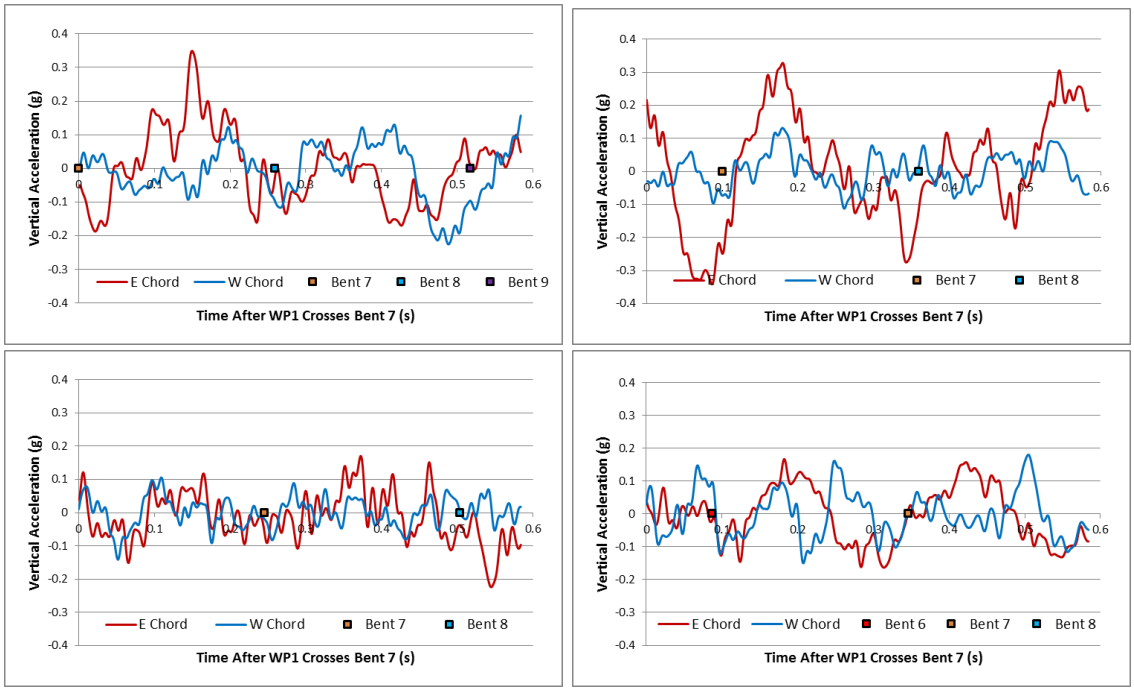

Figure A-13: Span 7 Test 7 Vertical Acceleration: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

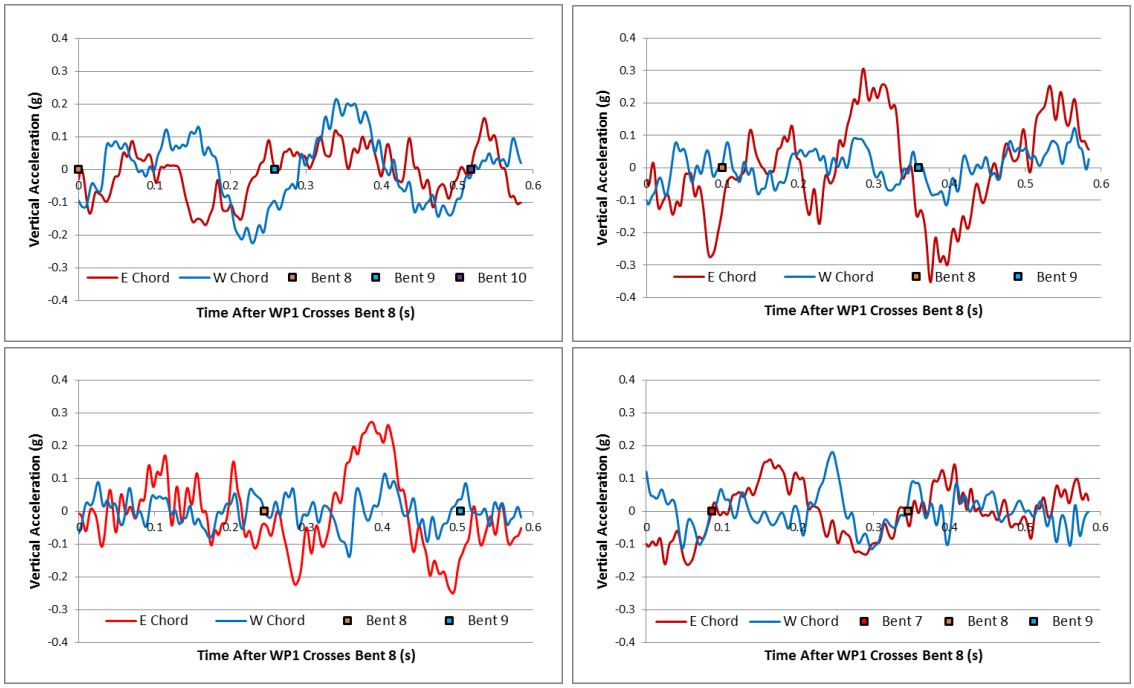

Figure A-14: Span 8 Test 7 Vertical Acceleration: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

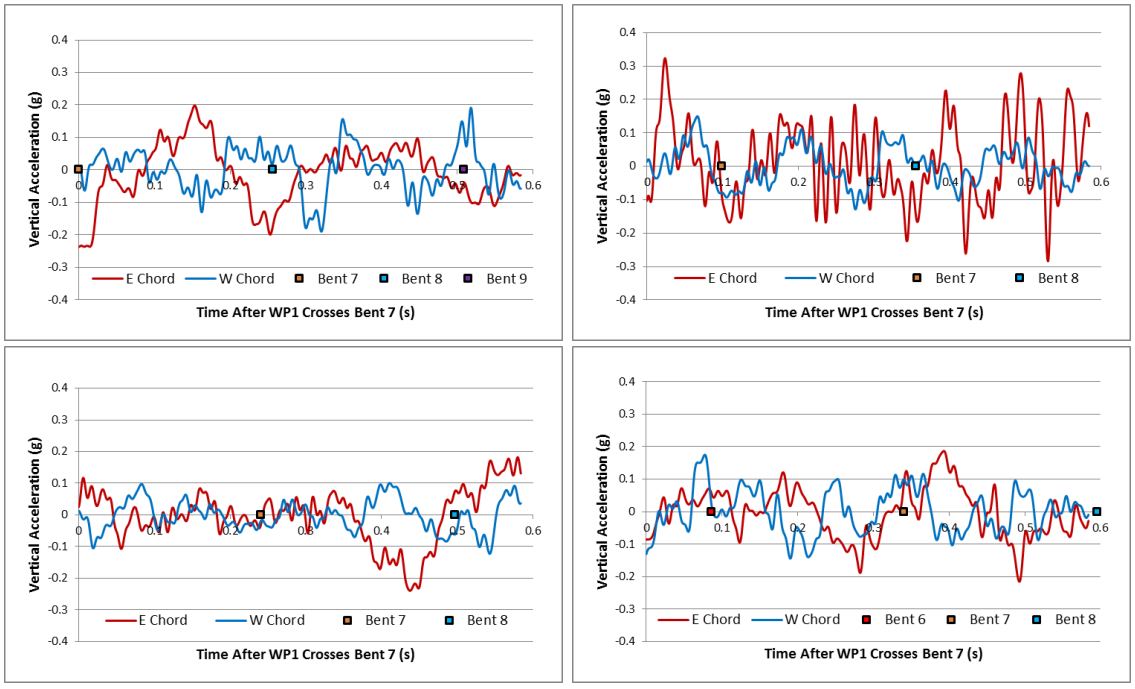

Figure A-15: Span 7 Test 8 Vertical Acceleration: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

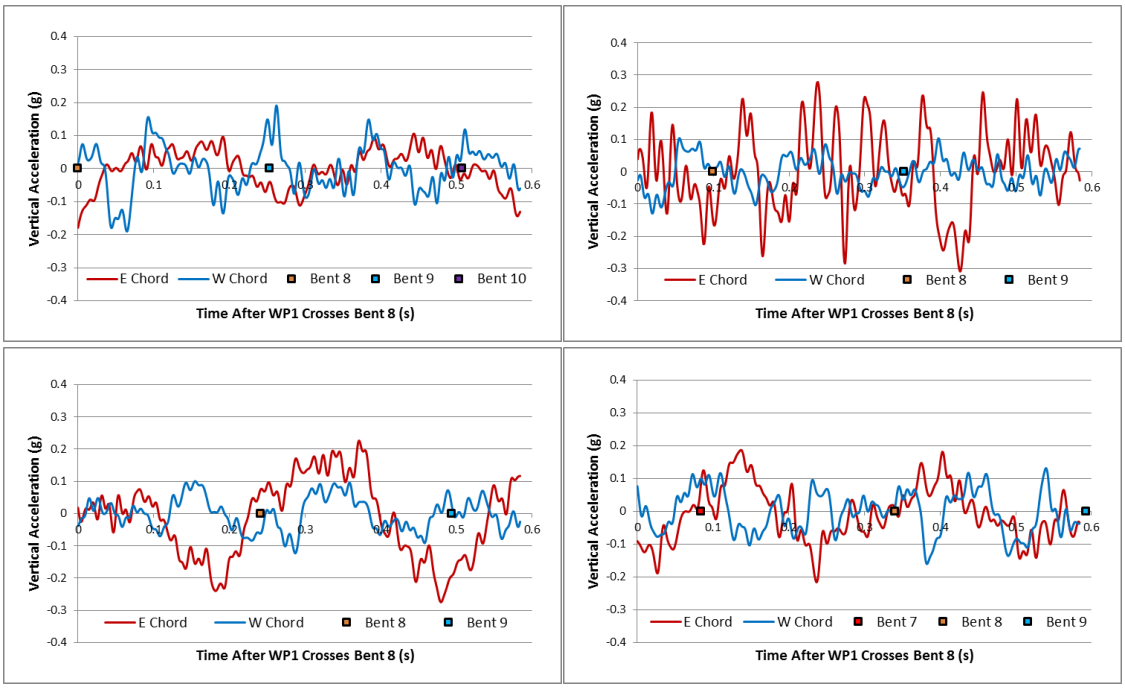

Figure A-16: Span 8 Test 8 Vertical Acceleration: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

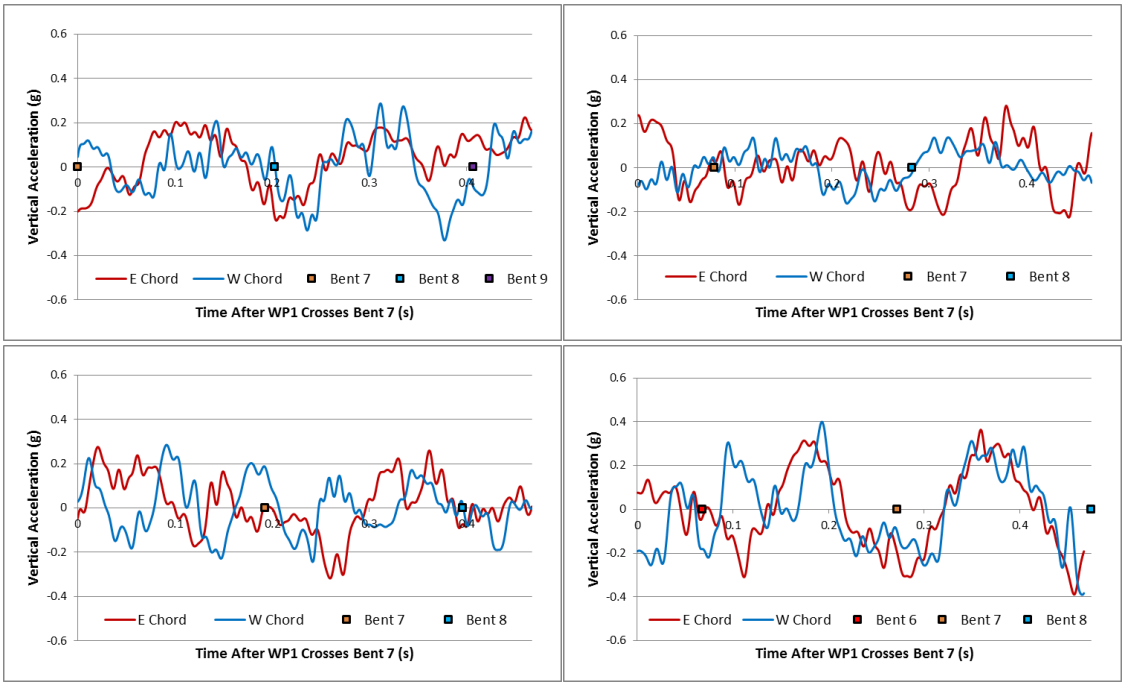

Figure A-17: Span 7 Test 9 Vertical Acceleration: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

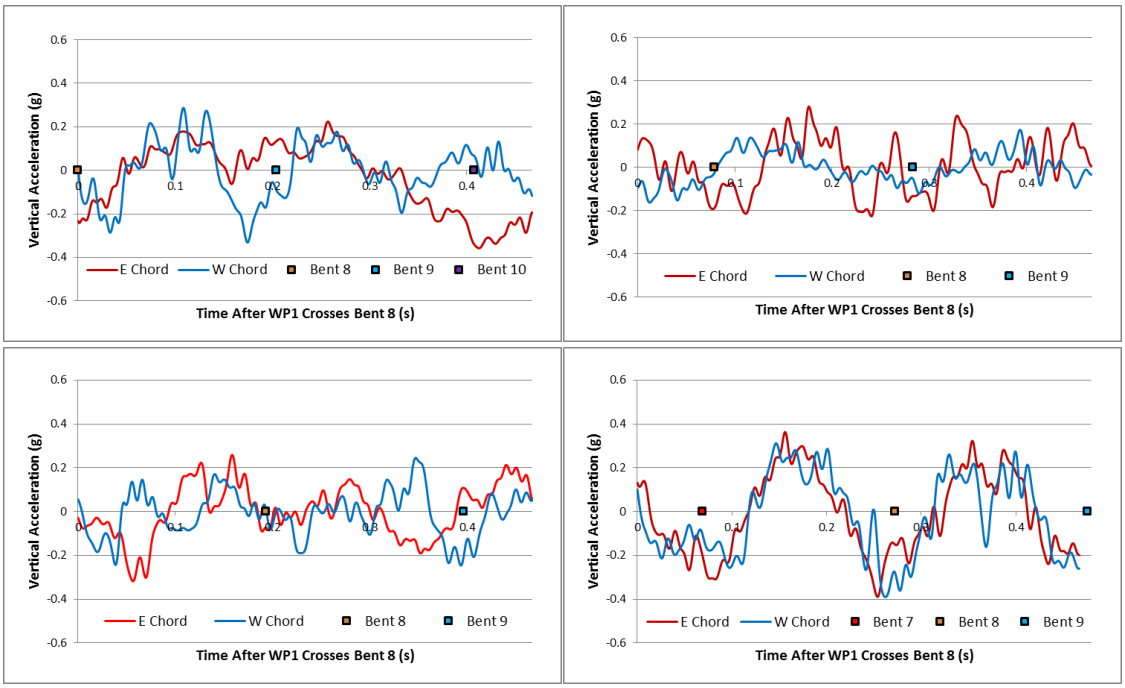

Figure A-18: Span 8 Test 9 Vertical Acceleration: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

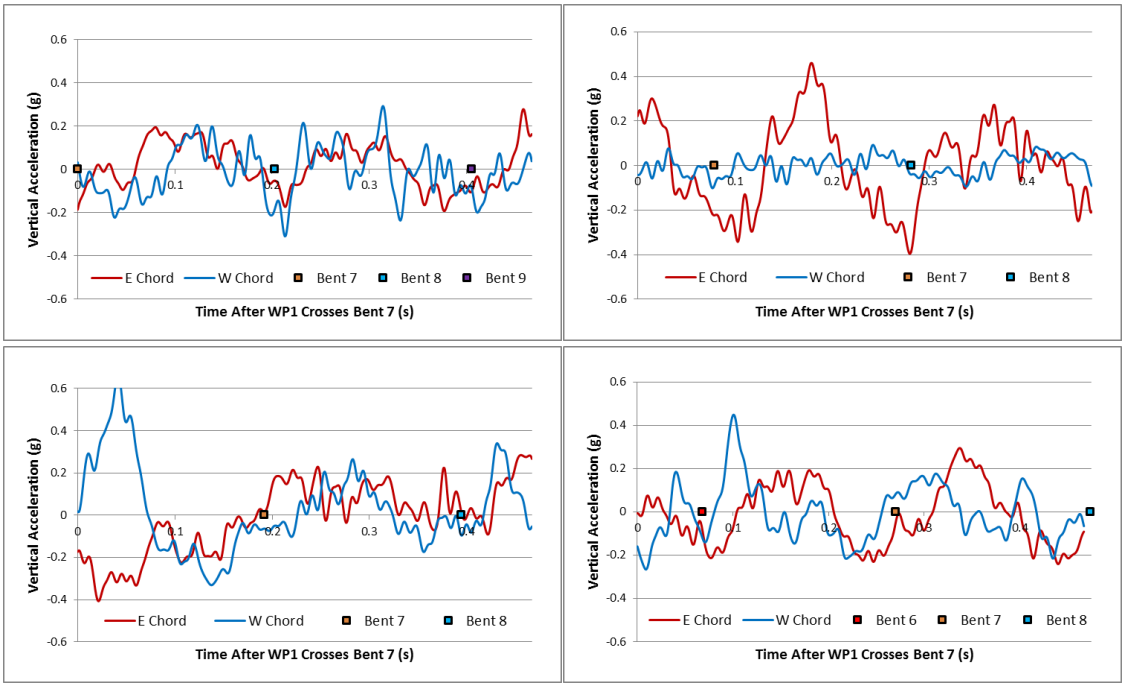

Figure A-19: Span 7 Test 10 Vertical Acceleration: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

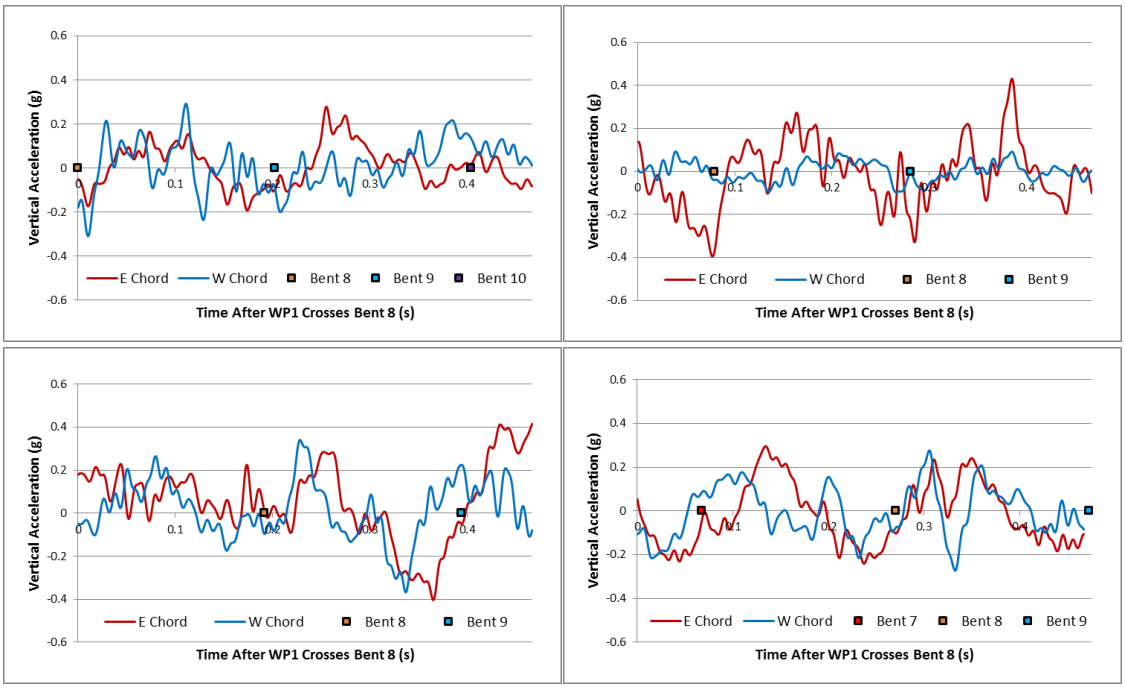

Figure A-20: Span 8 Test 10 Vertical Acceleration: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

A.2 Bridge Deflection Data

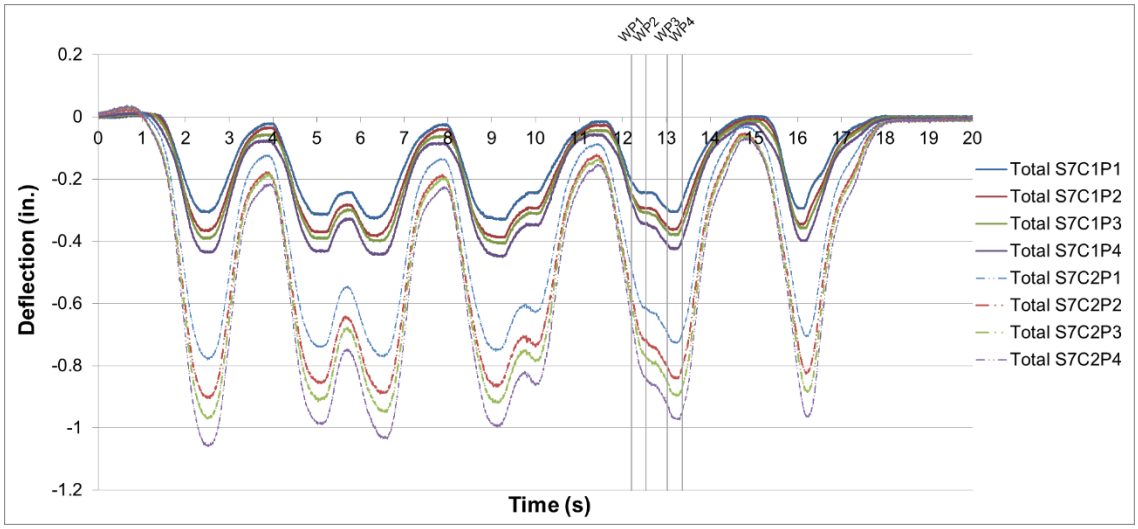

Figure A-21: Work Train Midspan Deflection Time History for Test 1 Span 7

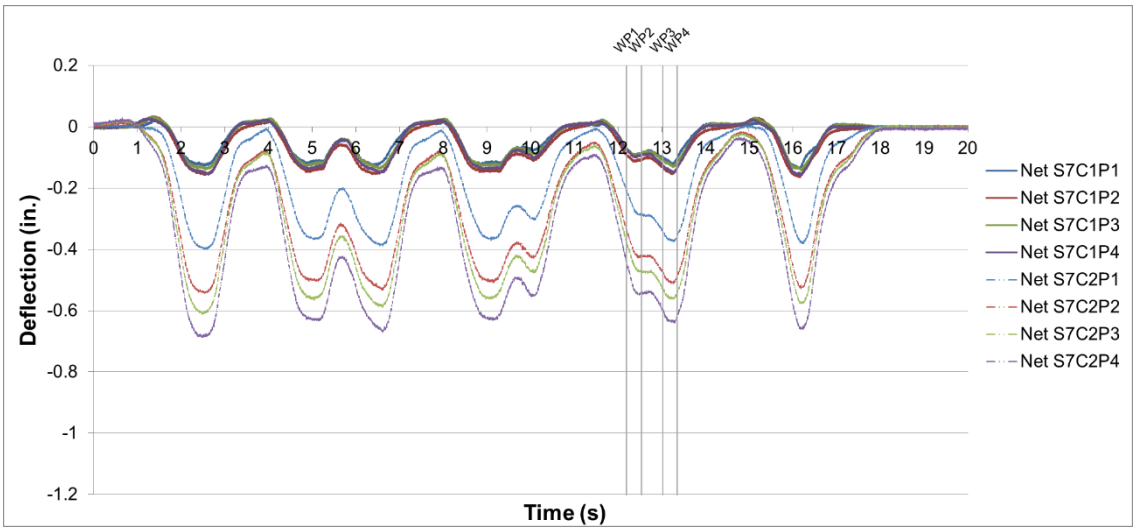

Figure A-22: Work Train Midspan Net Deflection Time History for Test 1 Span 7

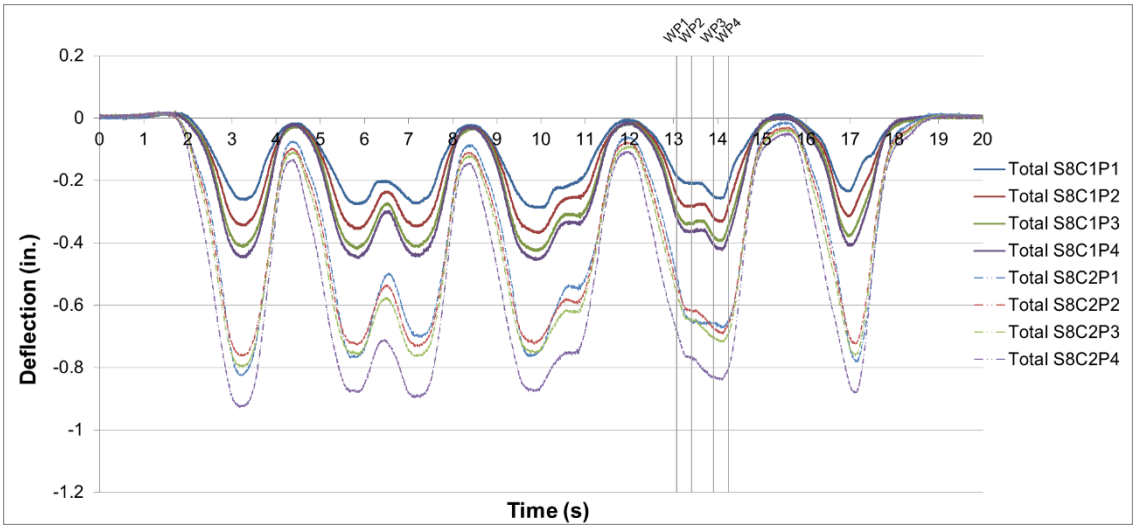

Figure A-23: Work Train Midspan Deflection Time History for Test 1 Span 8

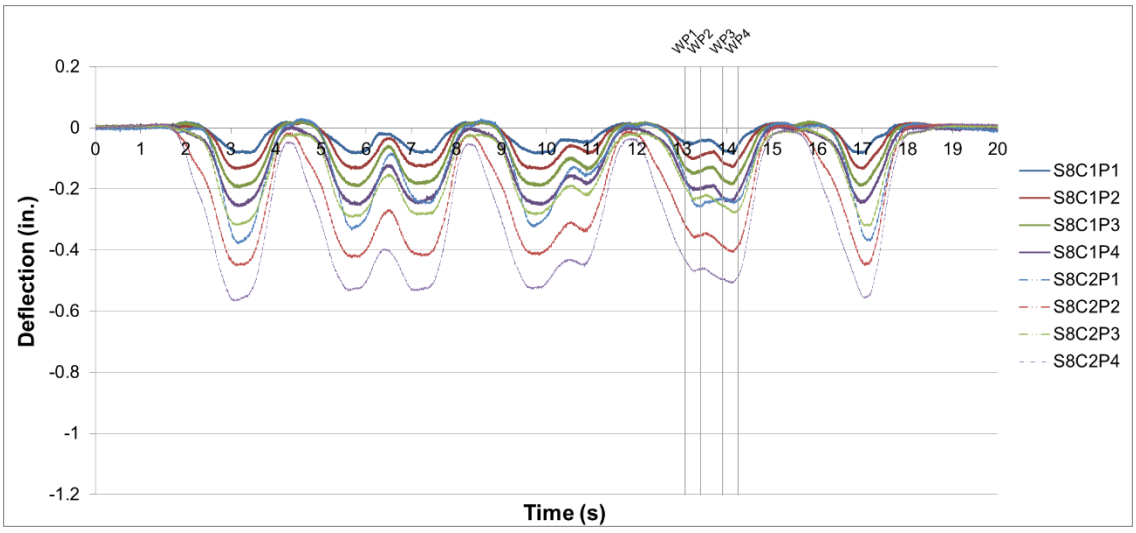

Figure A-24: Work Train Midspan Net Deflection Time History for Test 1 Span 8

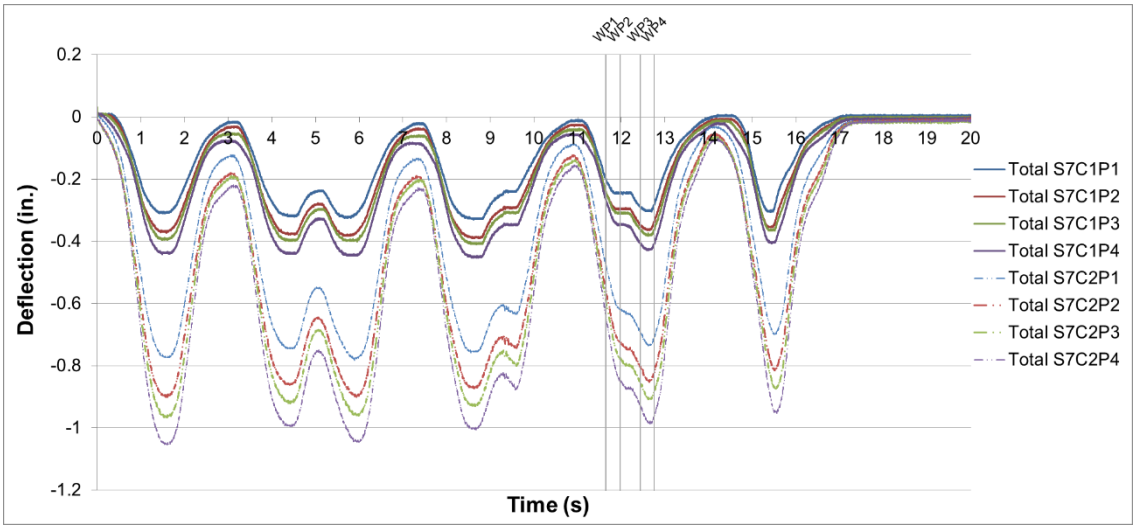

Figure A-25: Work Train Midspan Deflection Time History for Test 2 Span 7

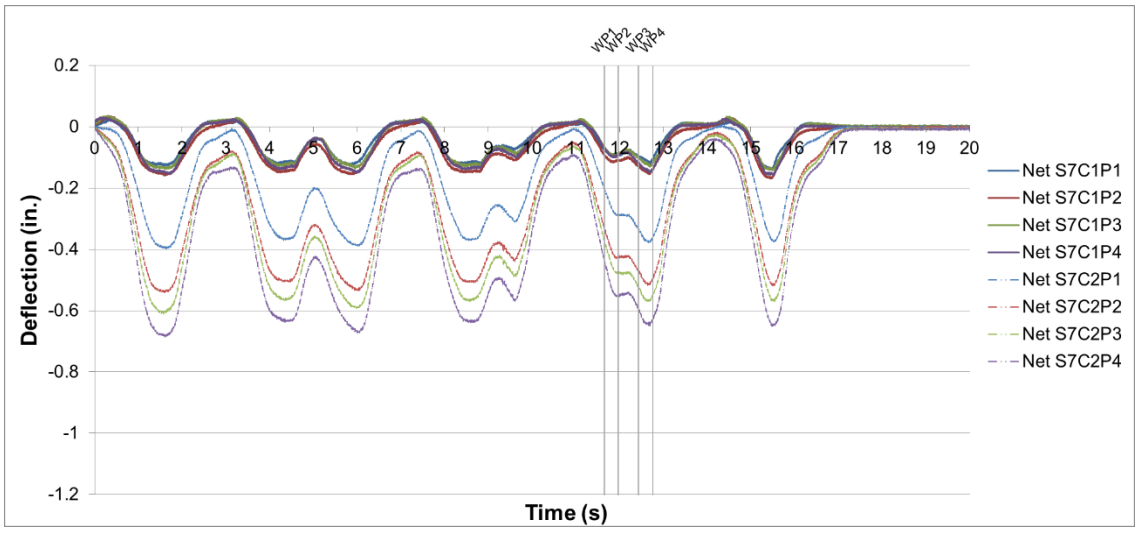

Figure A-26: Work Train Midspan Net Deflection Time History for Test 2 Span 7

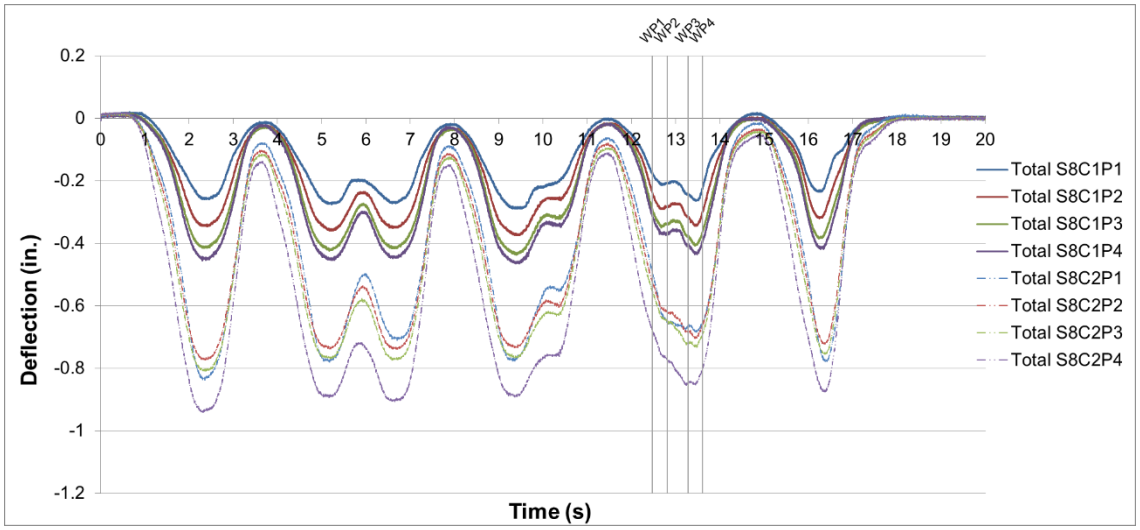

Figure A-27: Work Train Midspan Deflection Time History for Test 2 Span 8

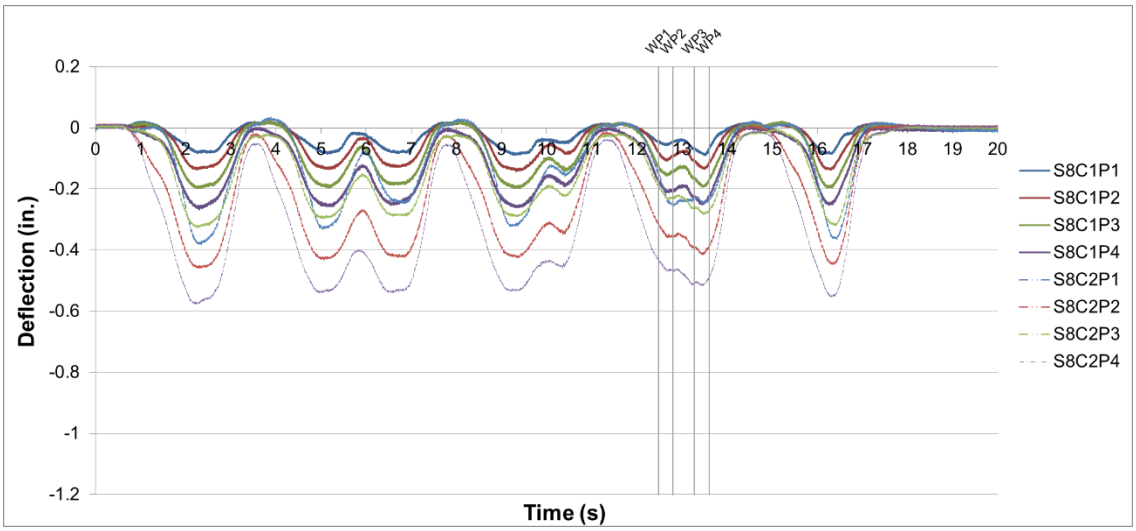

Figure A-28: Work Train Midspan Net Deflection Time History for Test 2 Span 8

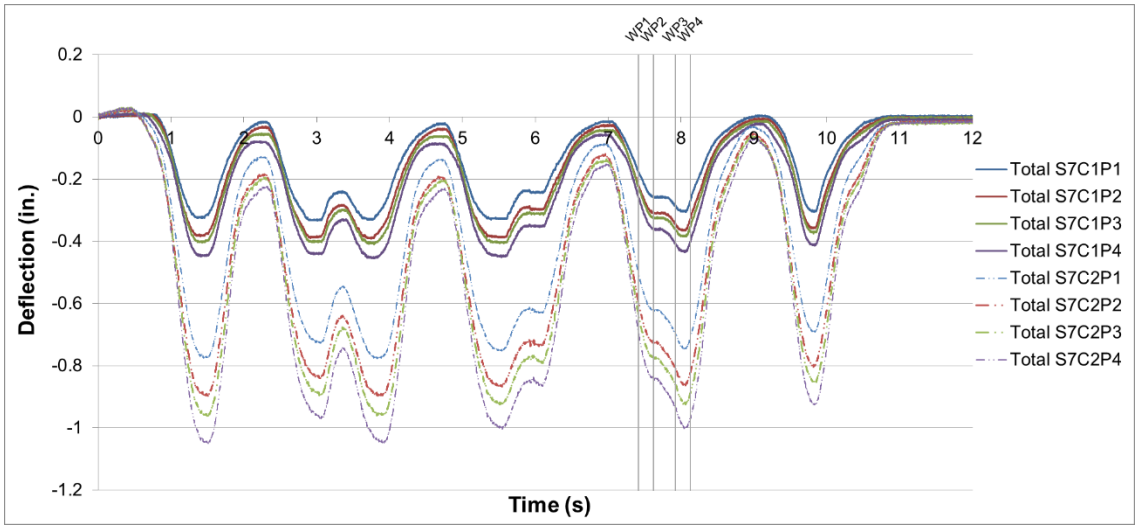

Figure A-29: Work Train Midspan Deflection Time History for Test 3 Span 7

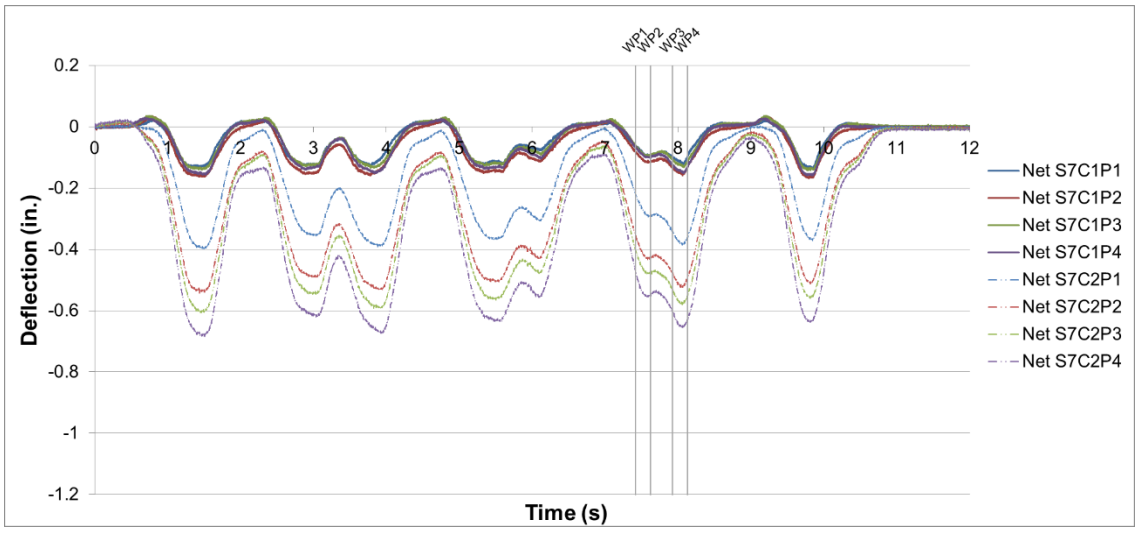

Figure A-30: Work Train Midspan Net Deflection Time History for Test 3 Span 7

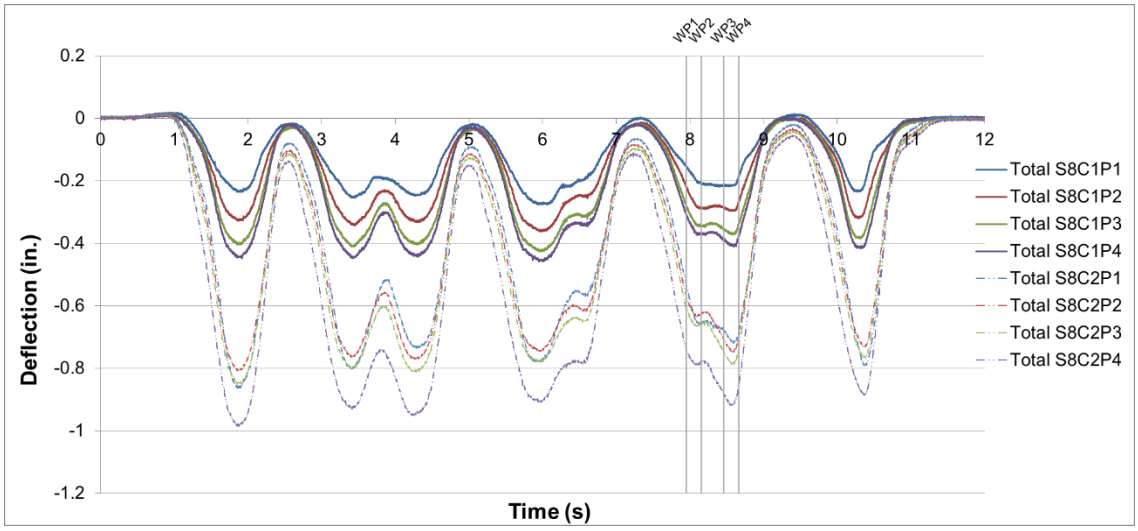

Figure A-31: Work Train Midspan Deflection Time History for Test 3 Span 8

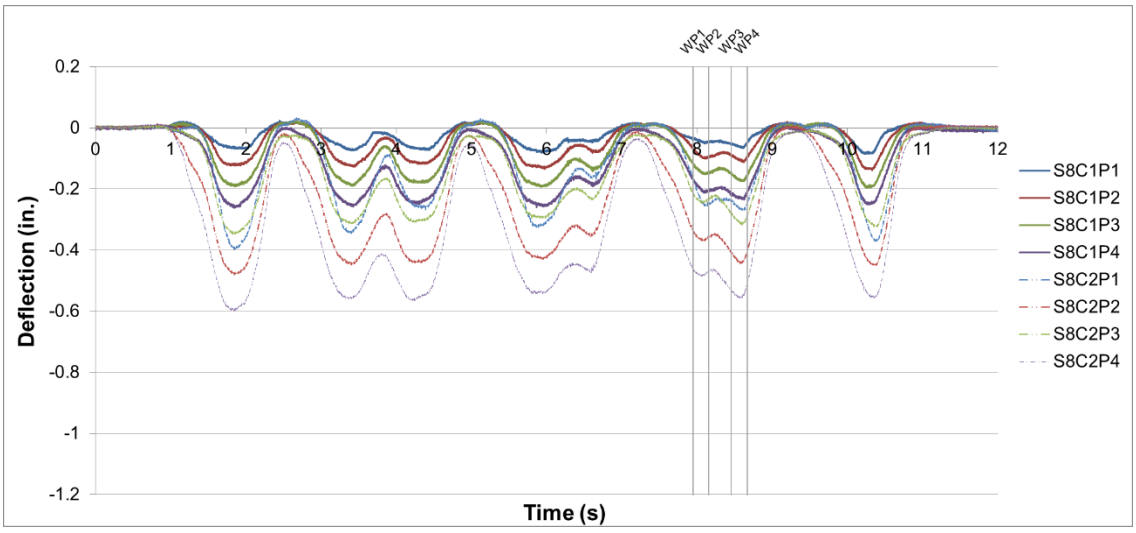

Figure A-32: Work Train Midspan Net Deflection Time History for Test 3 Span 8

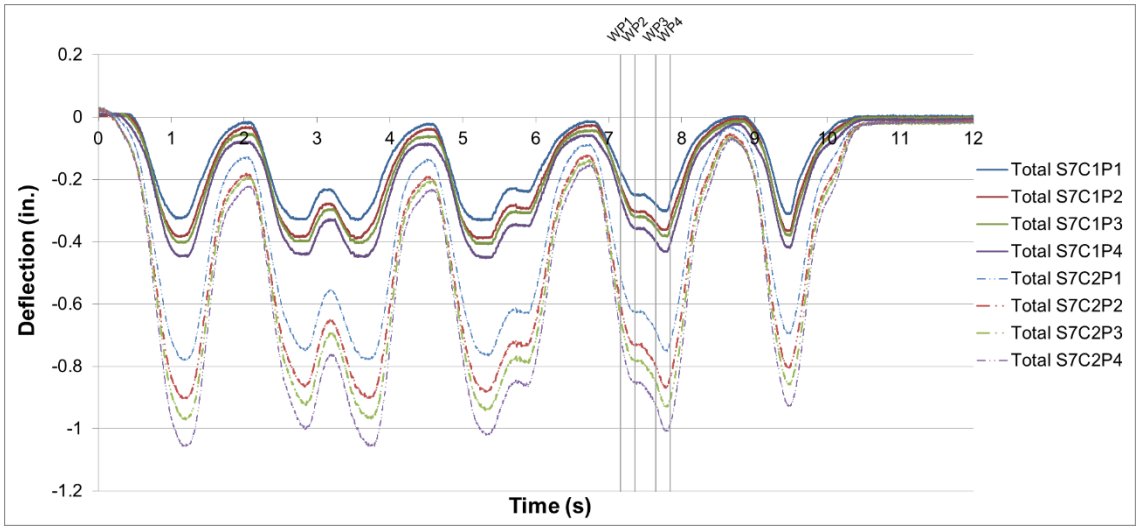

Figure A-33: Work Train Midspan Deflection Time History for Test 4 Span 7

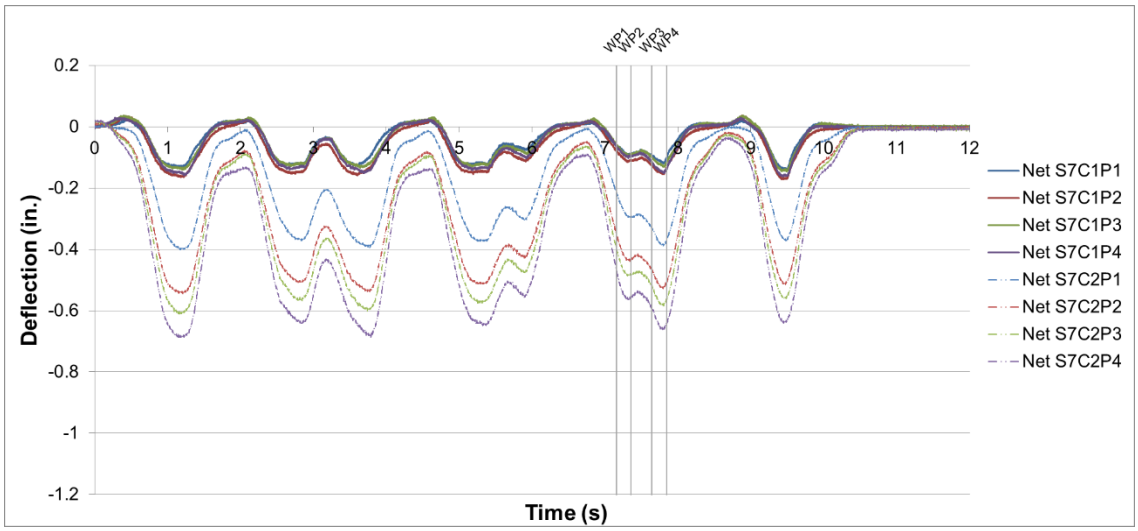

Figure A-34: Work Train Midspan Net Deflection Time History for Test 4 Span 7

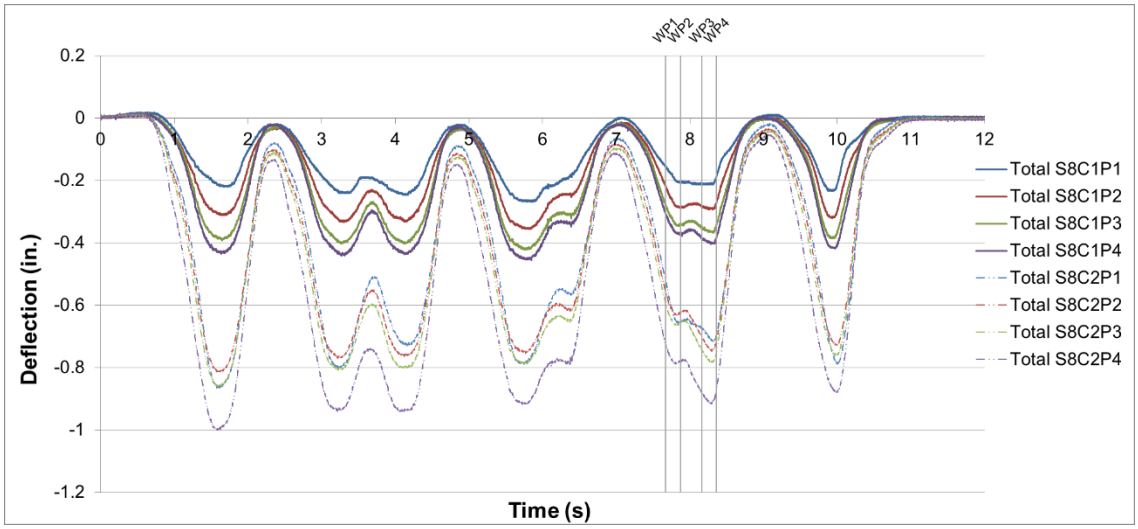

Figure A-35: Work Train Midspan Deflection Time History for Test 4 Span 8

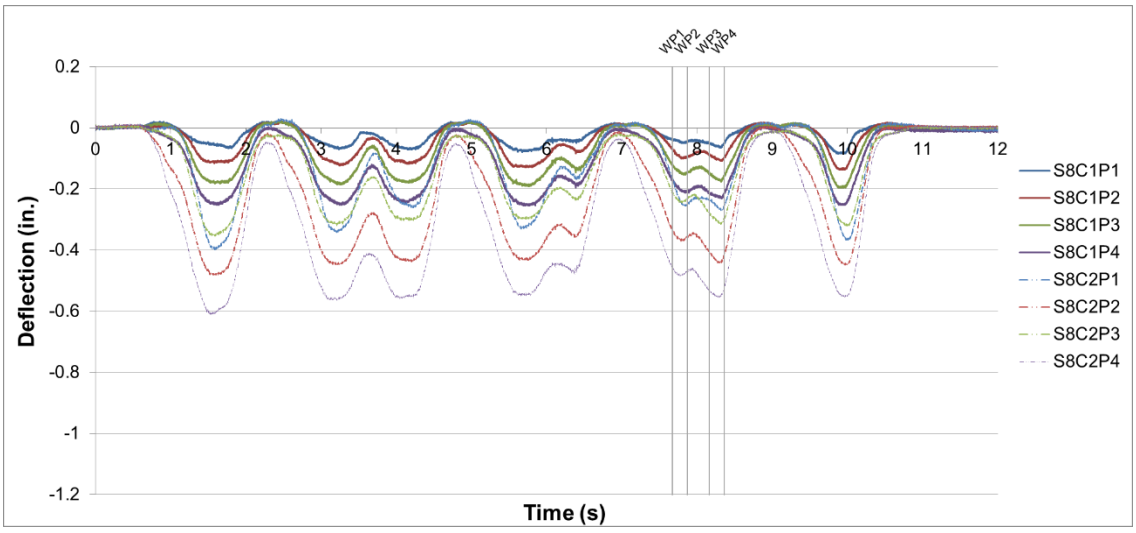

Figure A-36: Work Train Midspan Net Deflection Time History for Test 4 Span 8

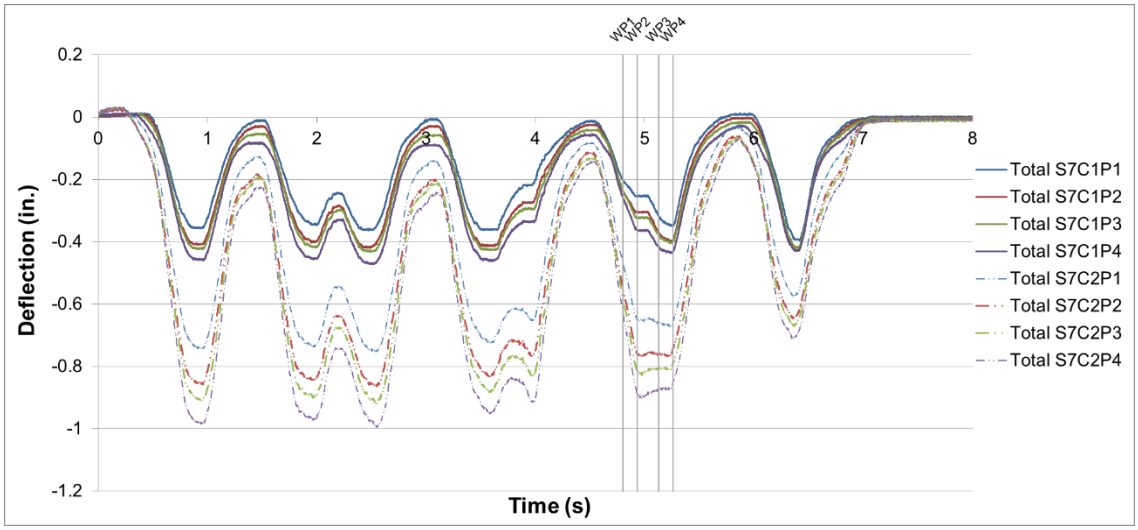

Figure A-37: Work Train Midspan Deflection Time History for Test 5 Span 7

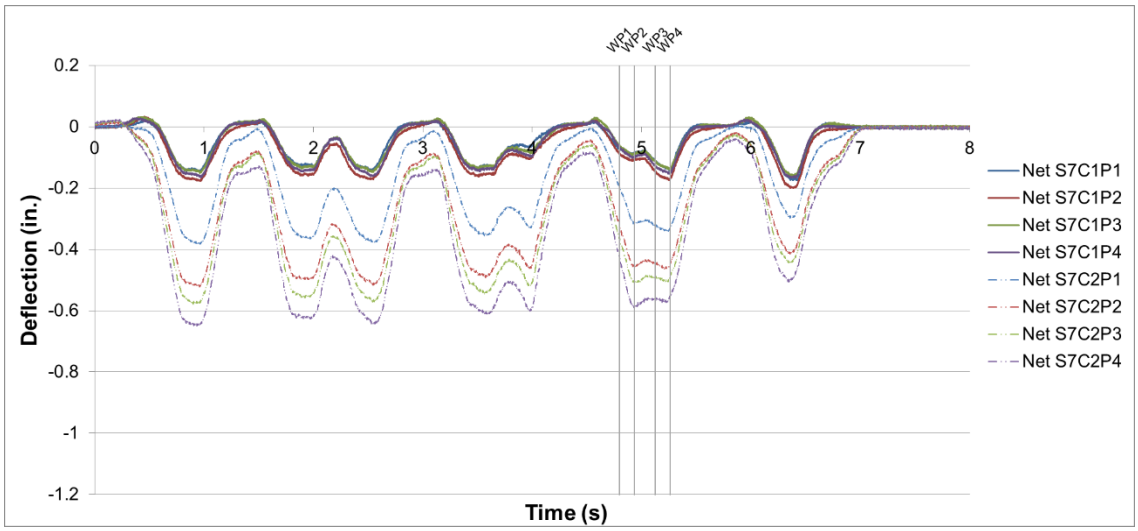

Figure A-38: Work Train Midspan Net Deflection Time History for Test 5 Span 7

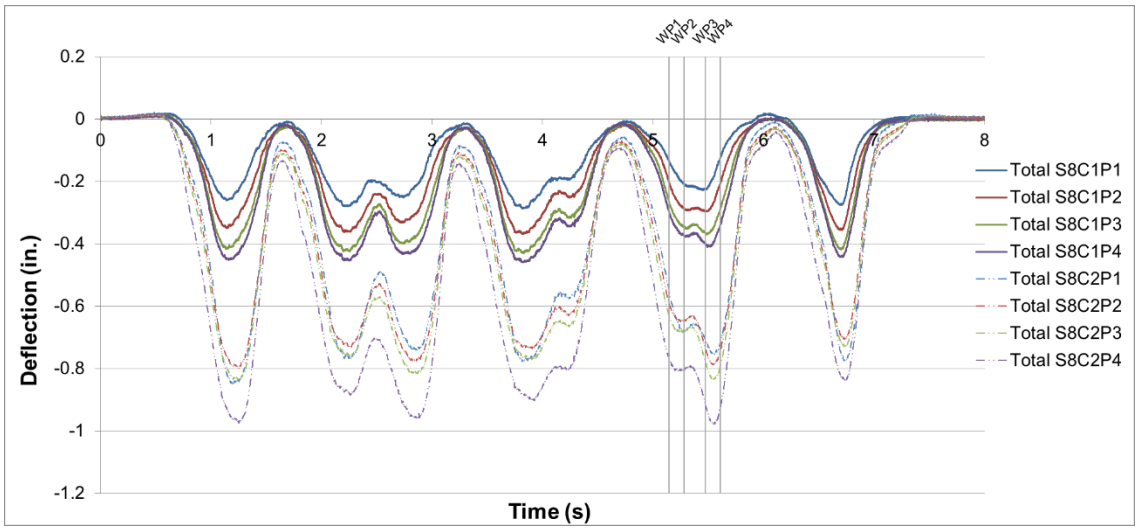

Figure A-39: Work Train Midspan Deflection Time History for Test 5 Span 8

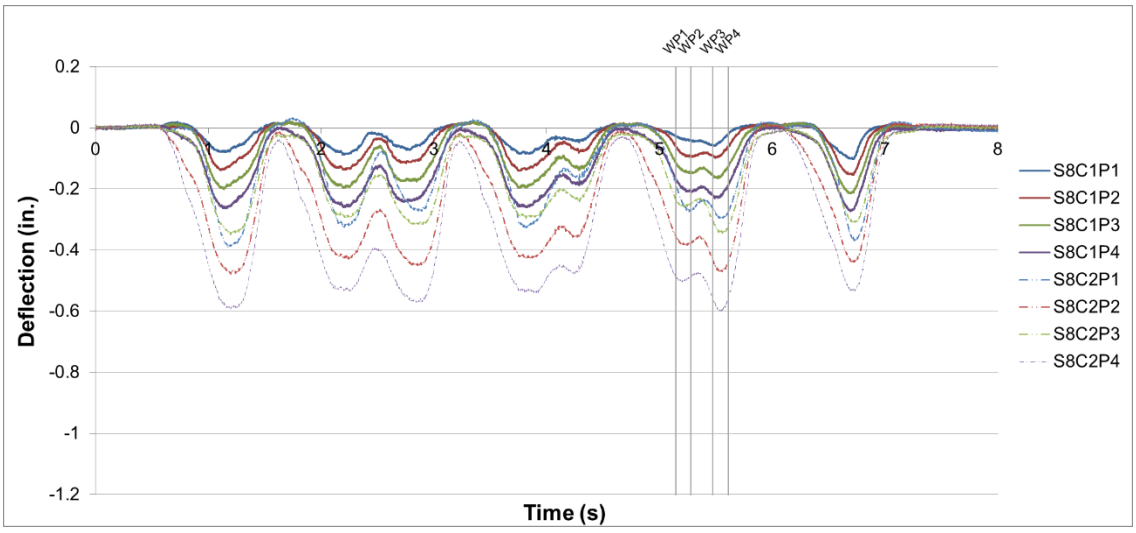

Figure A-40: Work Train Midspan Net Deflection Time History for Test 5 Span 8

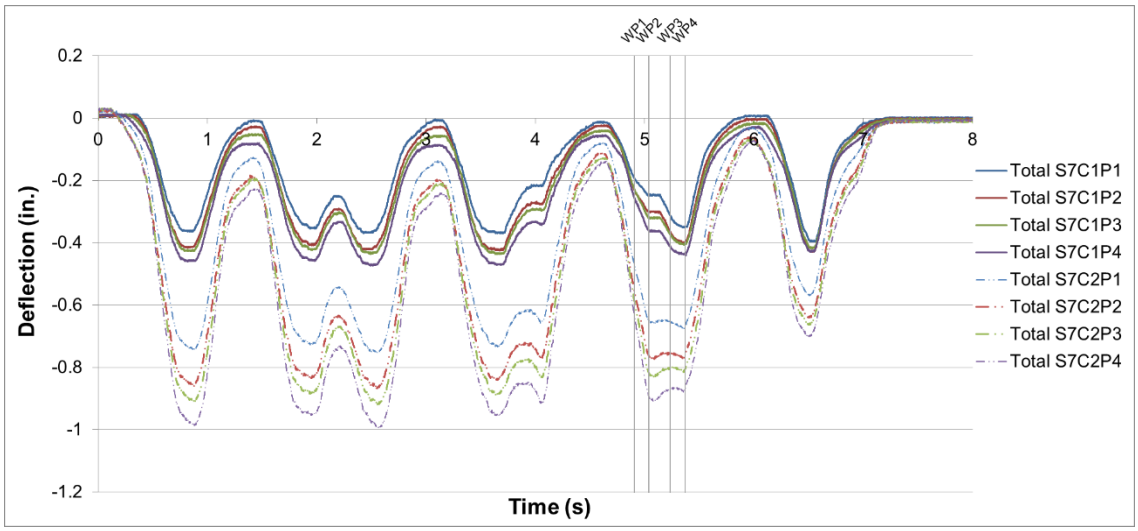

Figure A-41: Work Train Midspan Deflection Time History for Test 6 Span 7

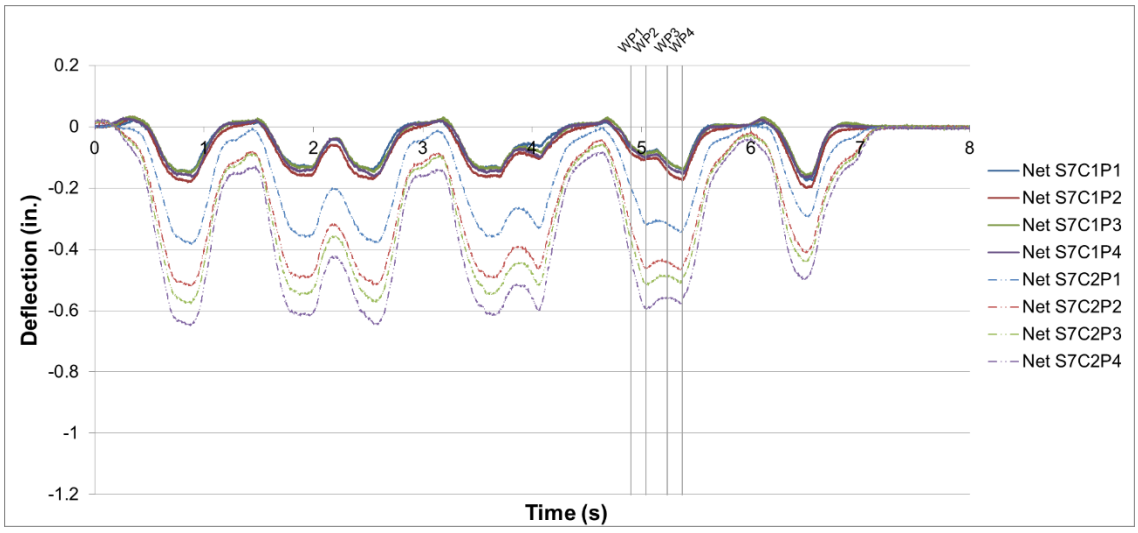

Figure A-42: Work Train Midspan Net Deflection Time History for Test 6 Span 7

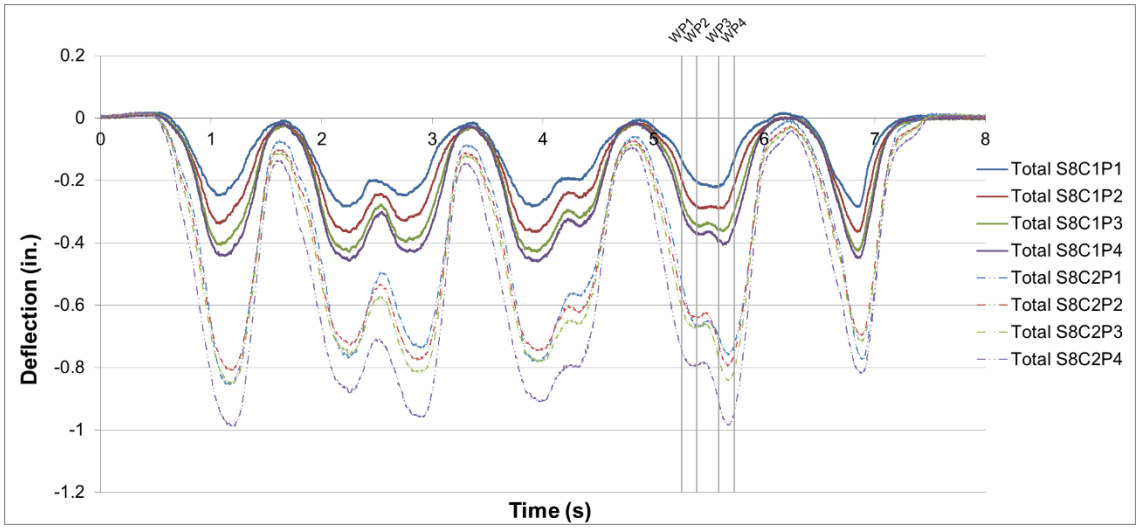

Figure A-43: Work Train Midspan Deflection Time History for Test 6 Span 8

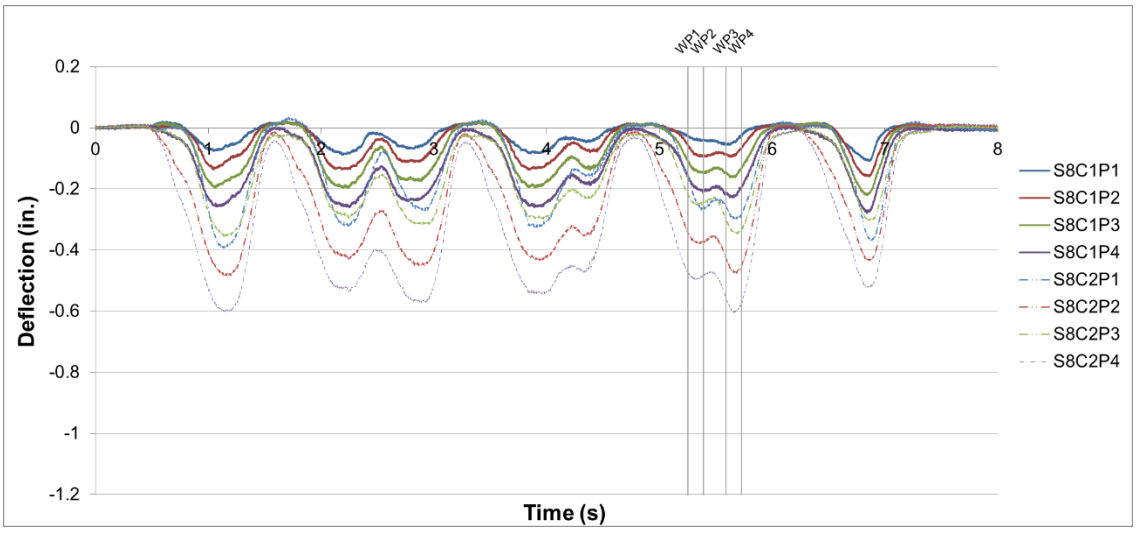

Figure A-44: Work Train Midspan Net Deflection Time History for Test 6 Span 8

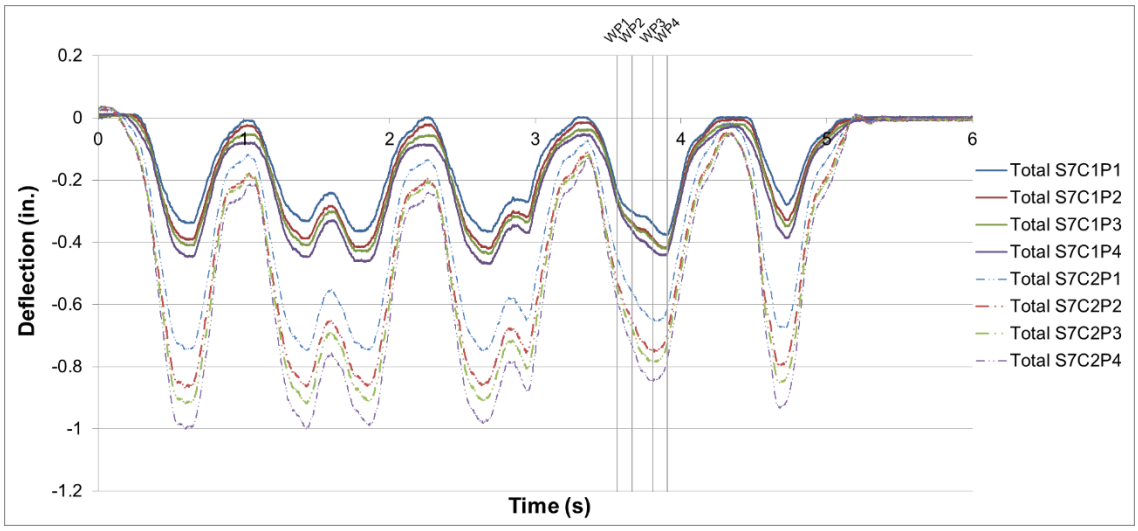

Figure A-45: Work Train Midspan Deflection Time History for Test 7 Span 7

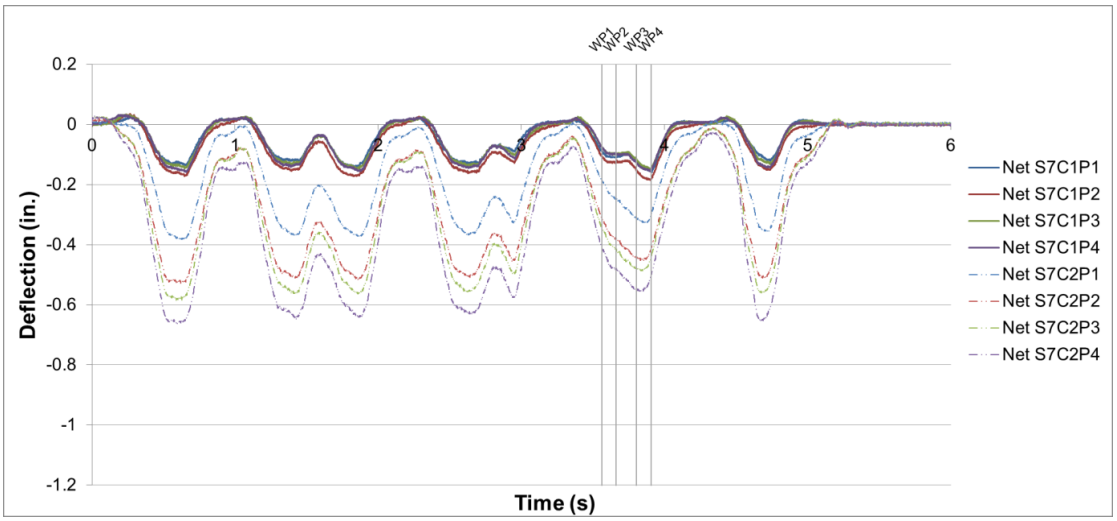

Figure A-46: Work Train Midspan Net Deflection Time History for Test 7 Span 7

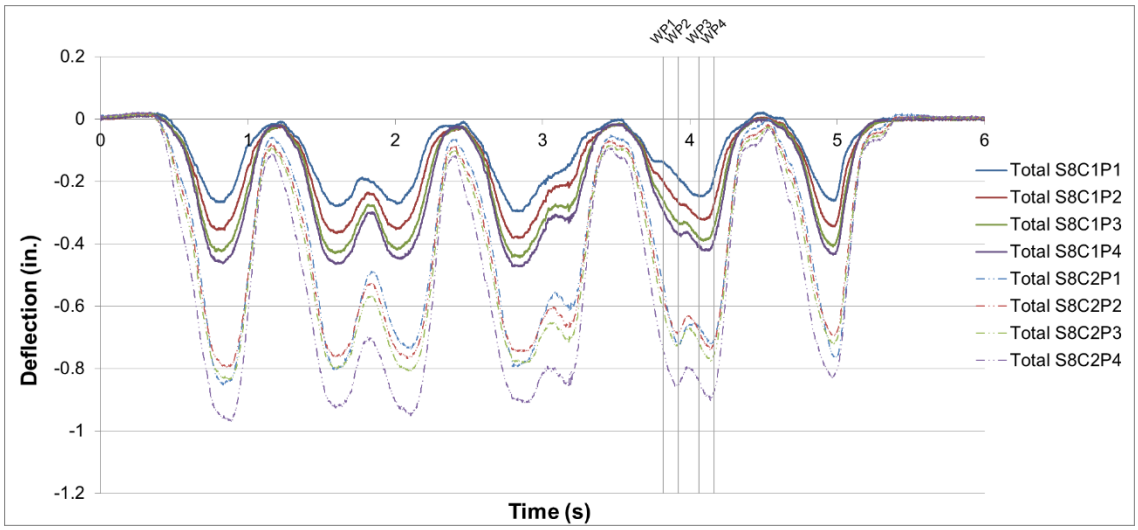

Figure A-47: Work Train Midspan Deflection Time History for Test 7 Span 8

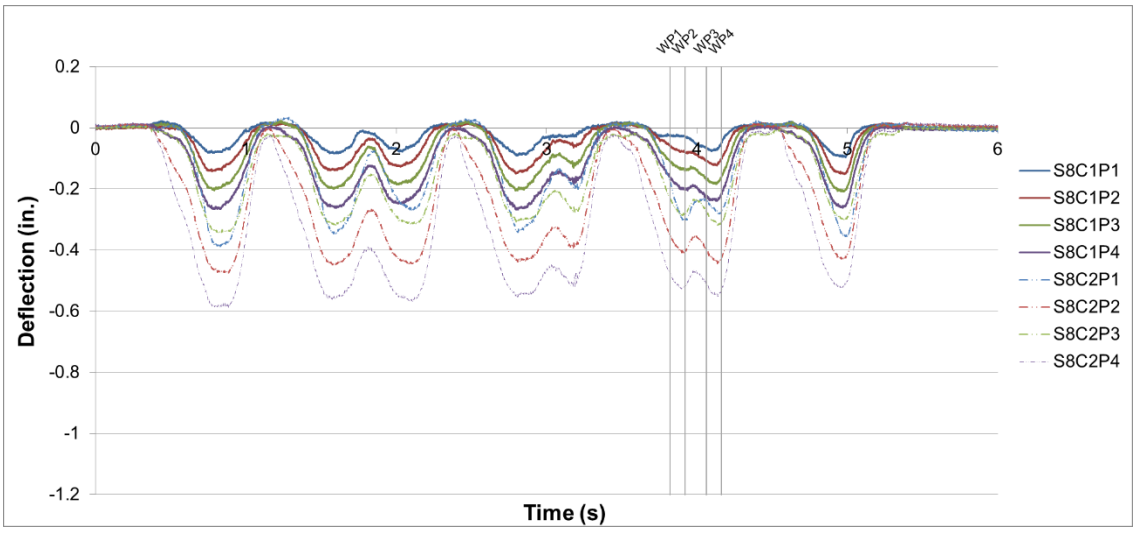

Figure A-48: Work Train Midspan Net Deflection Time History for Test 7 Span 8

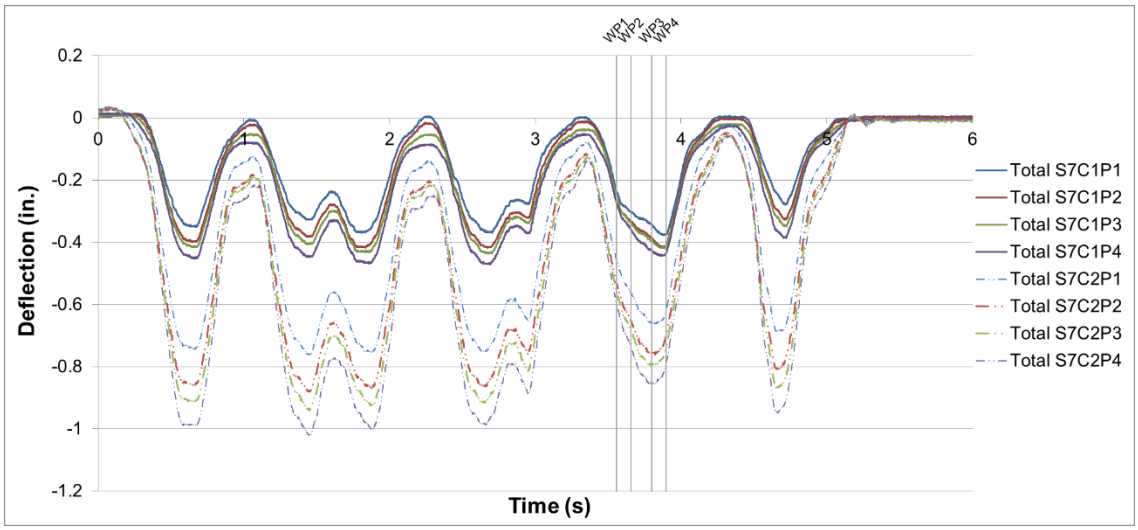

Figure A-49: Work Train Midspan Deflection Time History for Test 8 Span 7

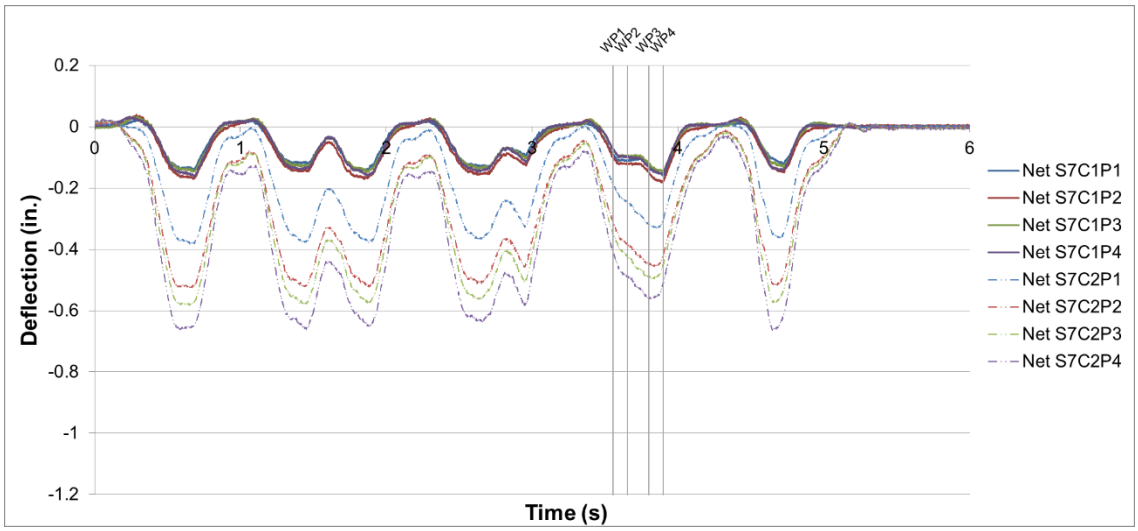

Figure A-50: Work Train Midspan Net Deflection Time History for Test 8 Span 7

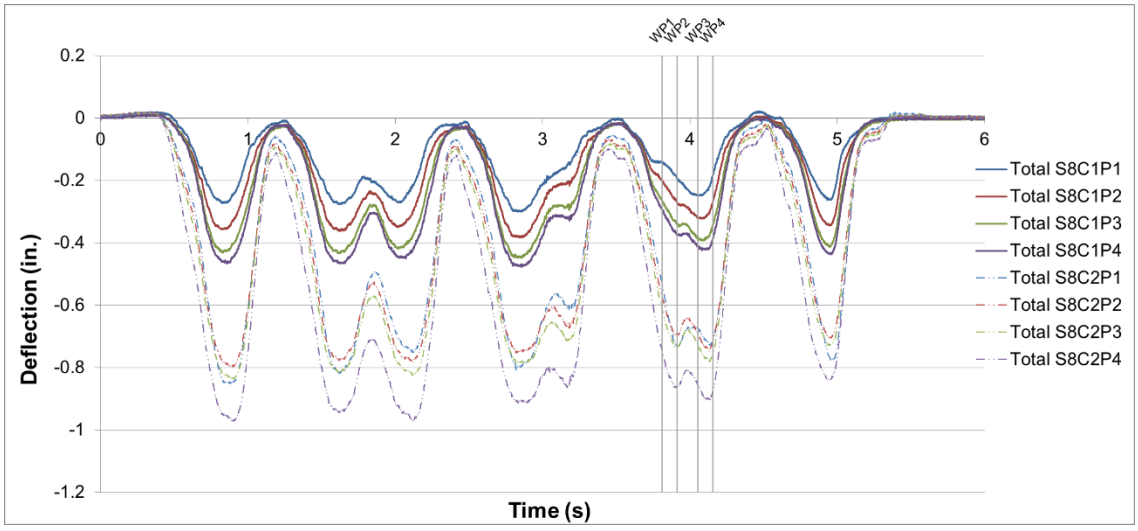

Figure A-51: Work Train Midspan Deflection Time History for Test 8 Span 8

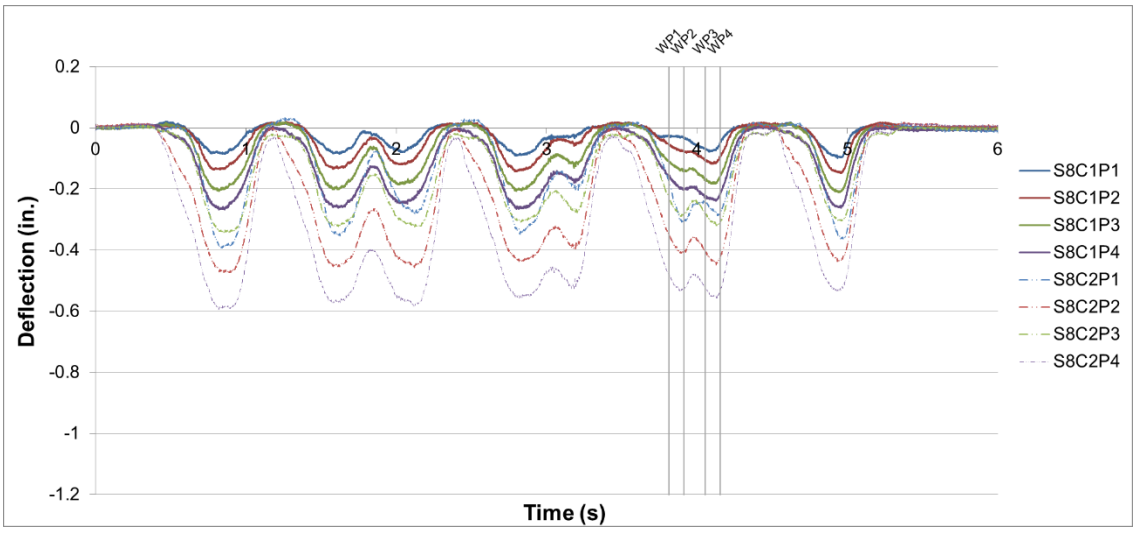

Figure A-52: Work Train Midspan Net Deflection Time History for Test 8 Span 8

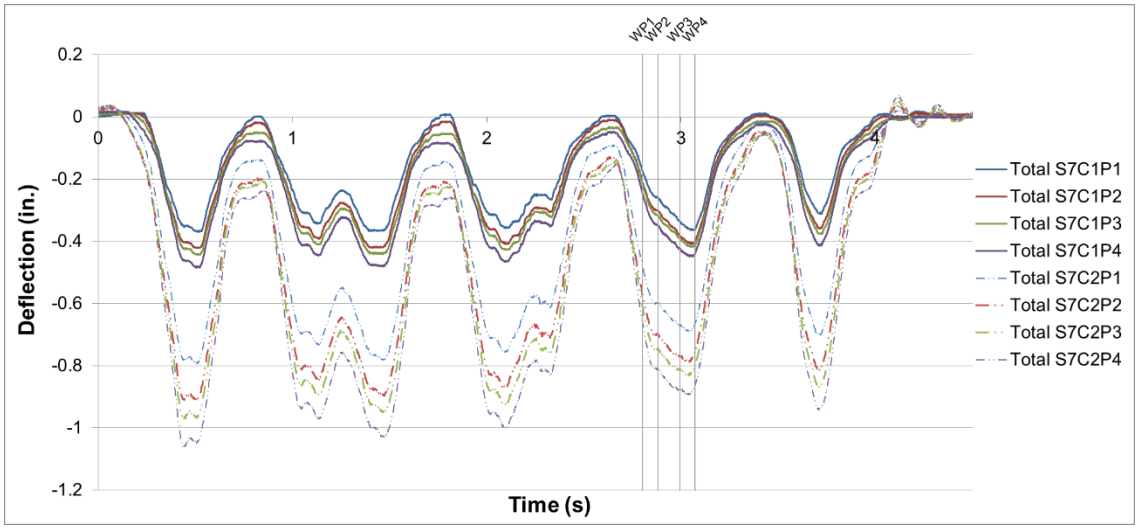

Figure A-53: Work Train Midspan Deflection Time History for Test 9 Span 7

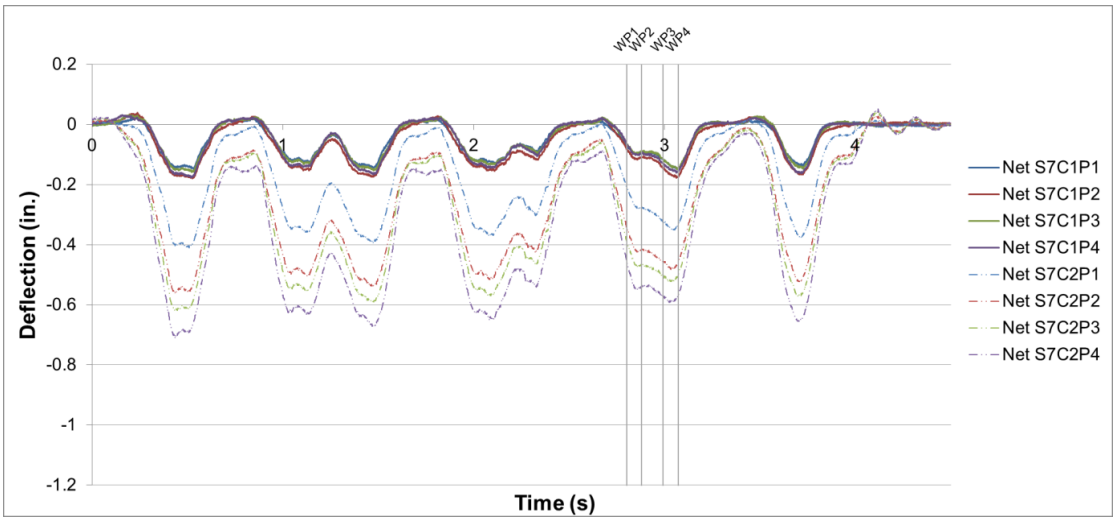

Figure A-54: Work Train Midspan Net Deflection Time History for Test 9 Span 7

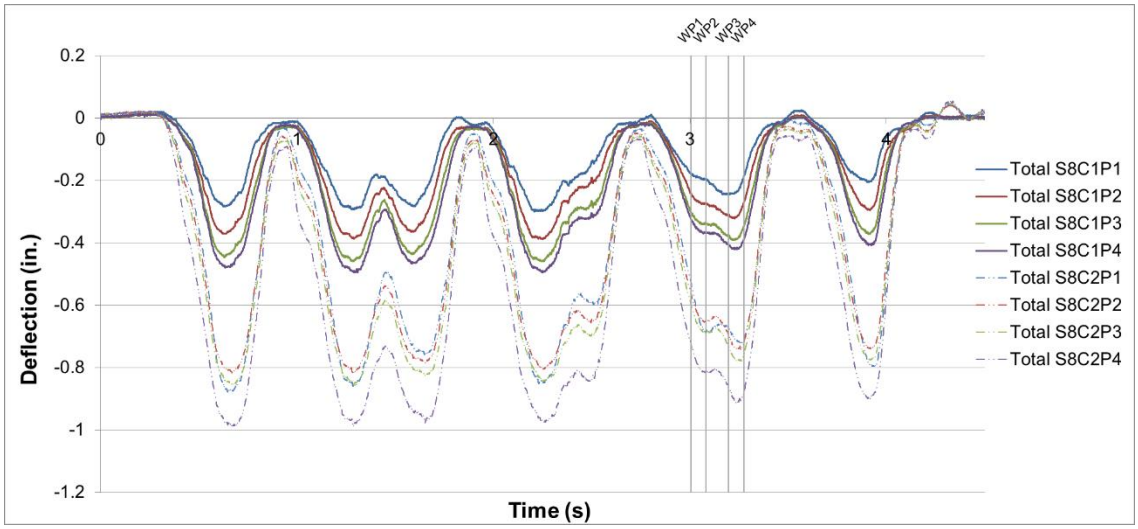

Figure A-55: Work Train Midspan Deflection Time History for Test 9 Span 8

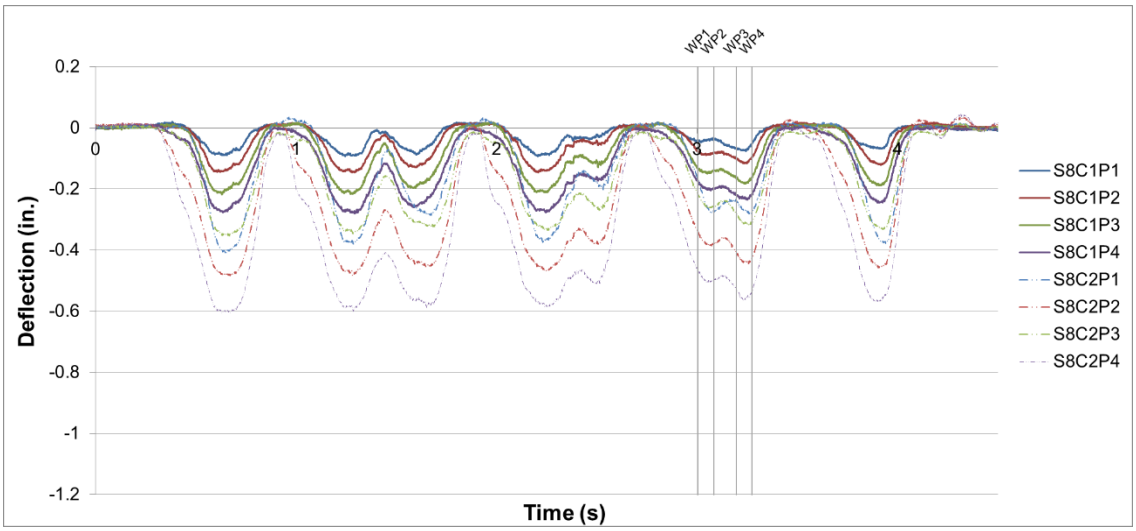

Figure A-56: Work Train Midspan Net Deflection Time History for Test 9 Span 8

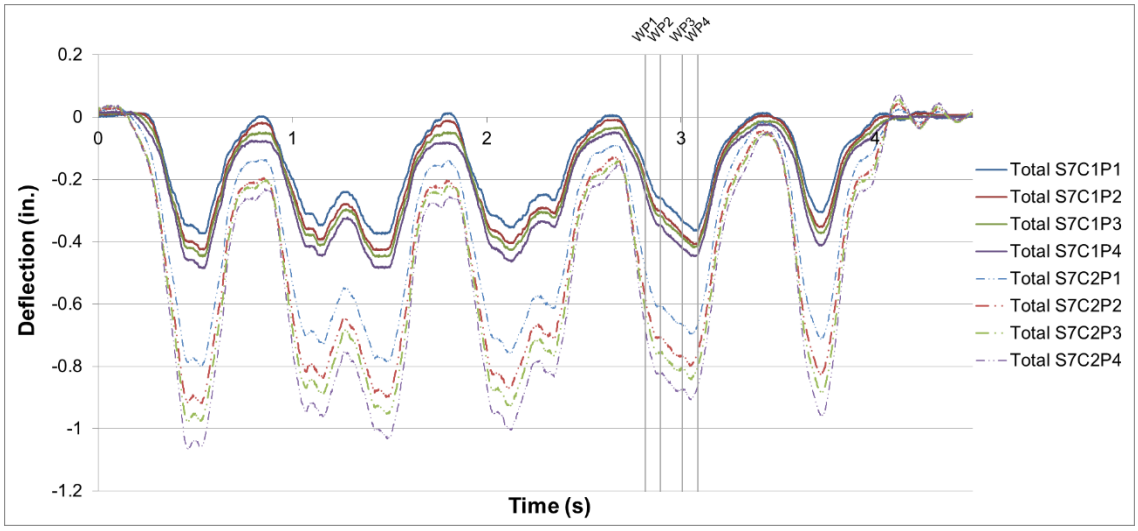

Figure A-57: Work Train Midspan Deflection Time History for Test 10 Span 7

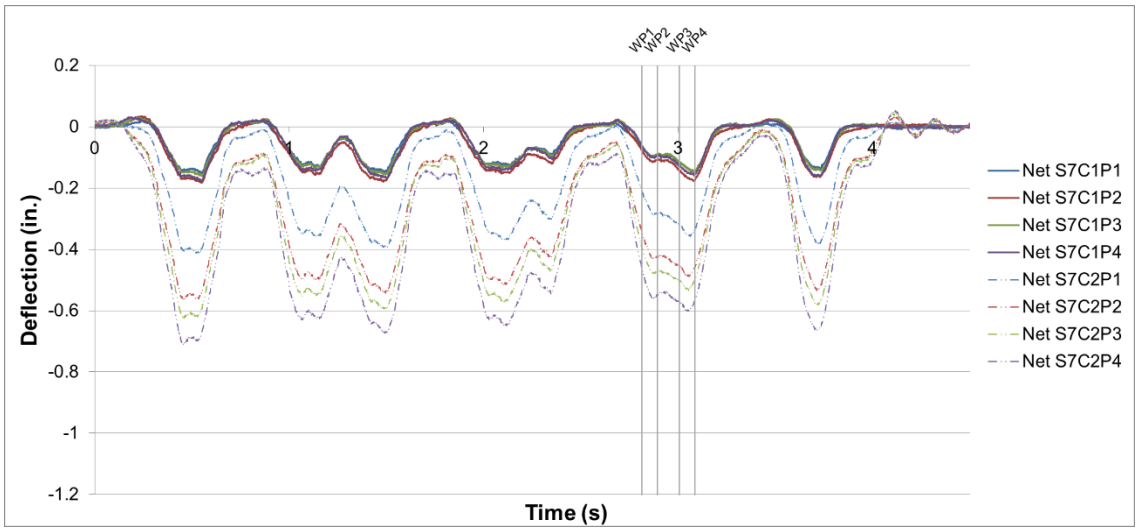

Figure A-58: Work Train Midspan Net Deflection Time History for Test 10 Span 7

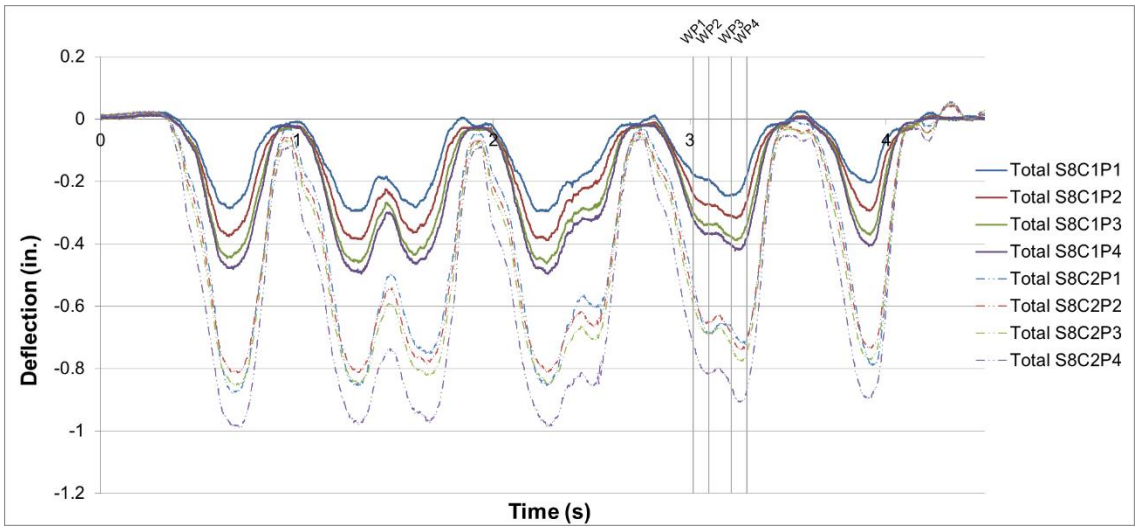

Figure A-59: Work Train Midspan Deflection Time History for Test 10 Span 8

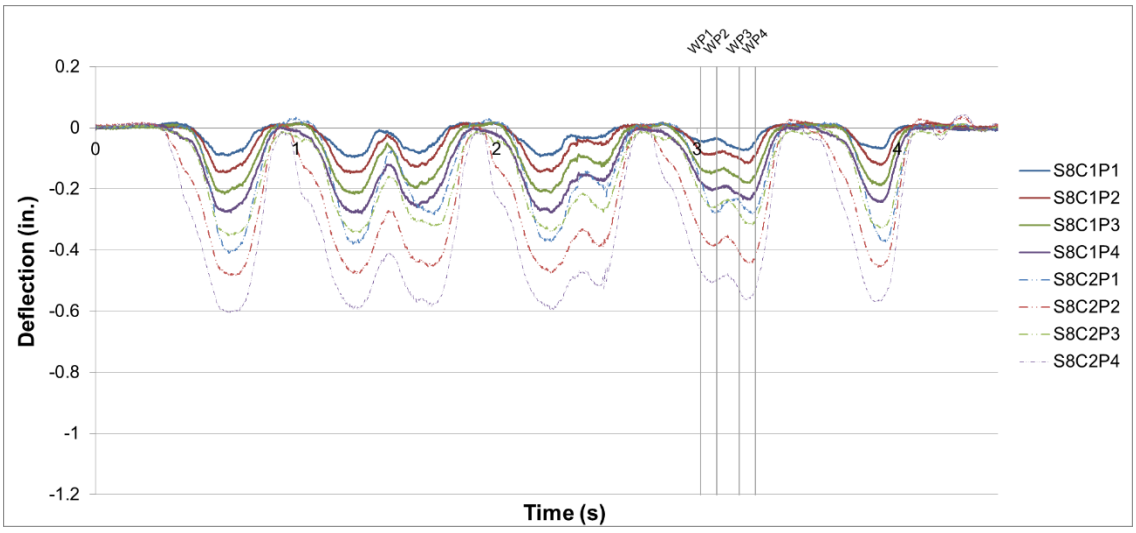

Figure A-60: Work Train Midspan Net Deflection Time History for Test 10 Span 8

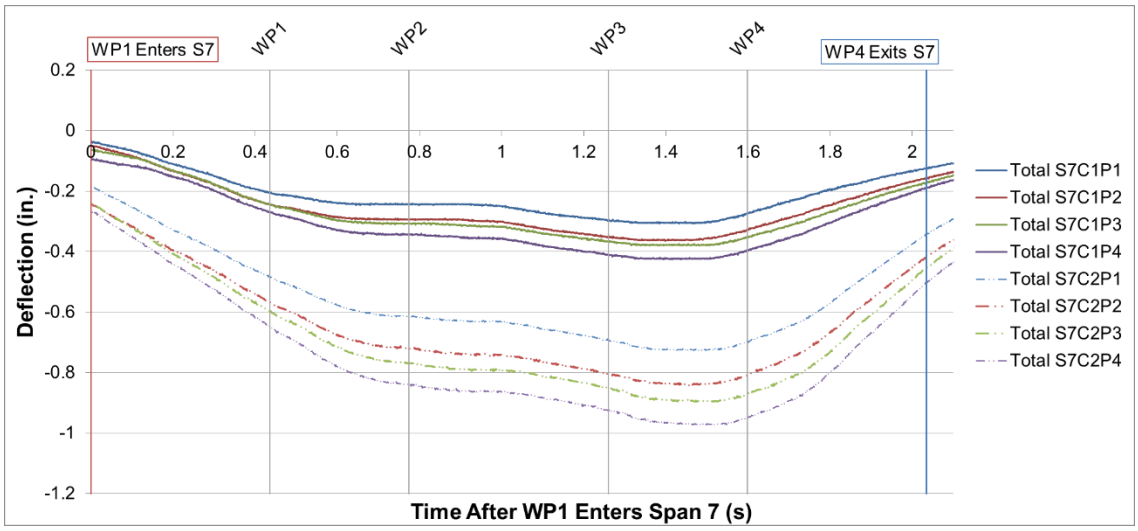

Figure A-61: Span 7 Test 1 Work Train Midspan Stringer Deflection

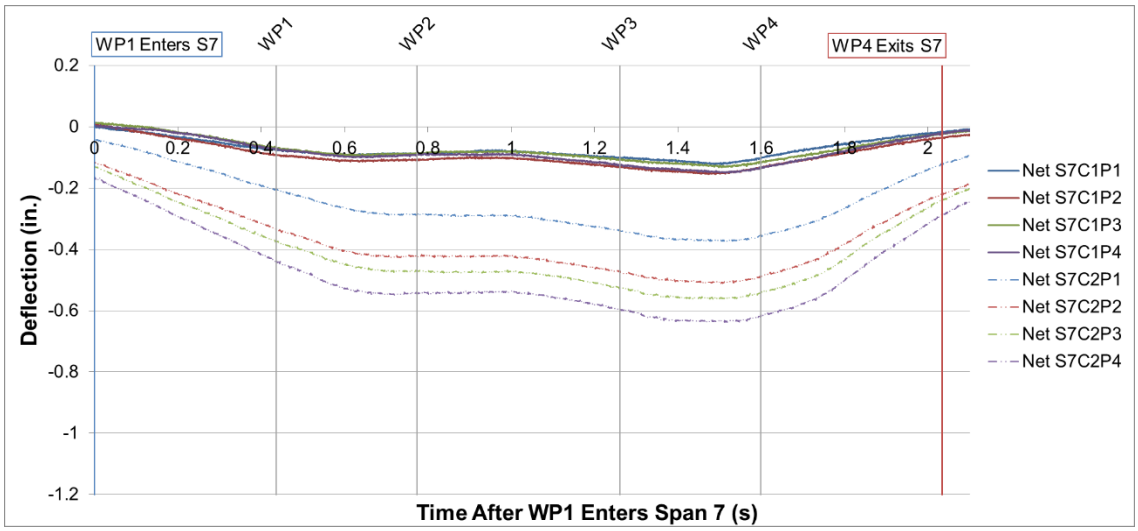

Figure A-62: Span 7 Test 1 Net Stringer Midspan Deflection

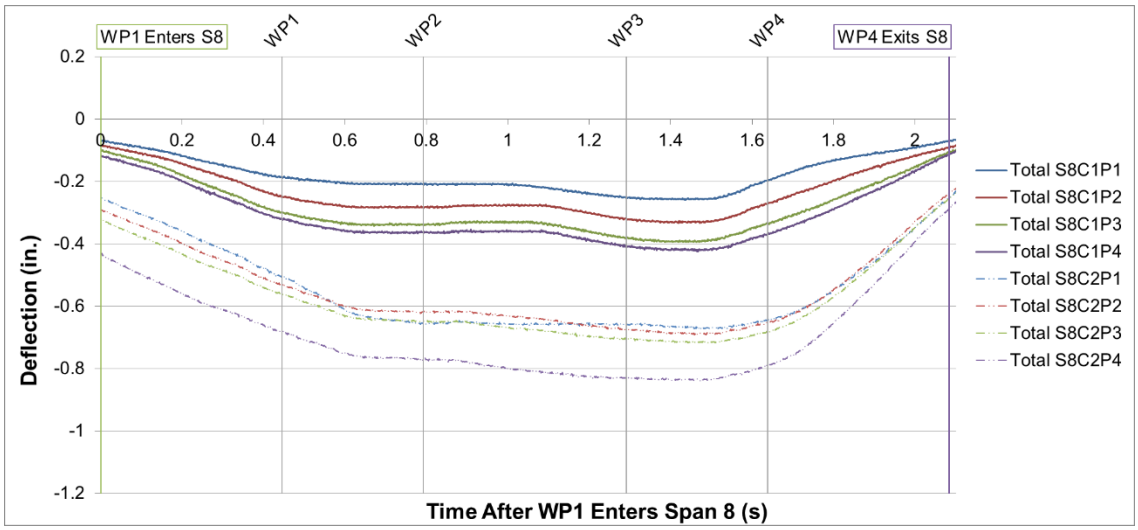

Figure A-63: Span 8 Test 1 Work Train Midspan Stringer Deflection

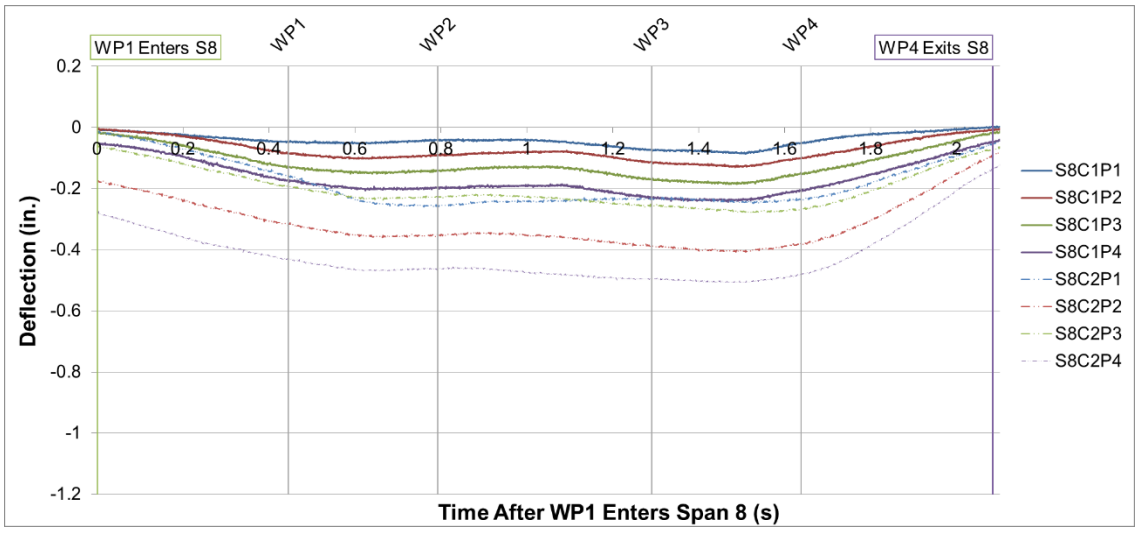

Figure A-64: Span 8 Test 1 Net Stringer Midspan Deflection

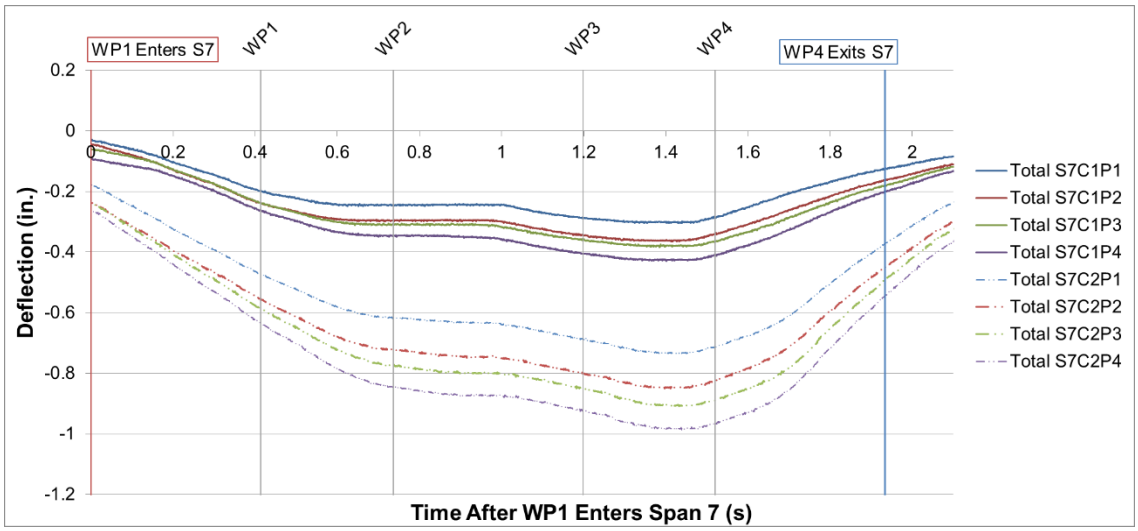

Figure A-65: Span 7 Test 2 Work Train Midspan Stringer Deflection

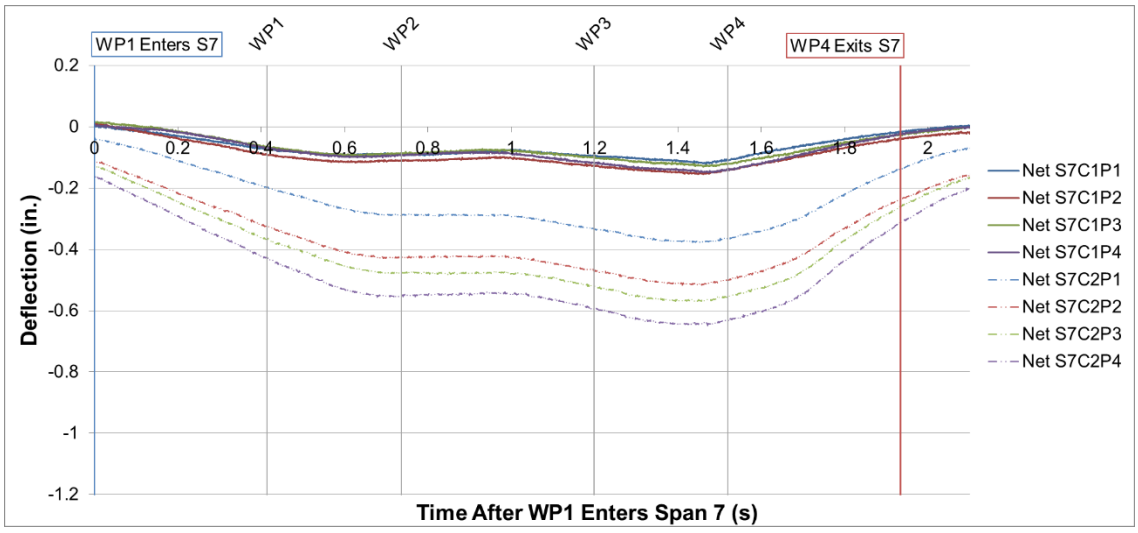

Figure A-66: Span 7 Test 2 Net Stringer Midspan Deflection

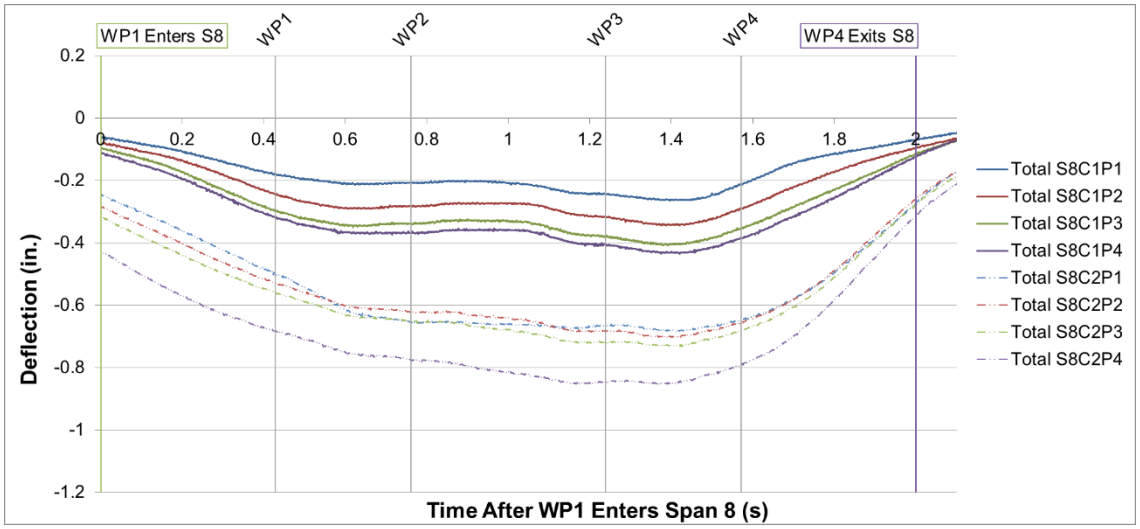

Figure A-67: Span 8 Test 2 Work Train Midspan Stringer Deflection

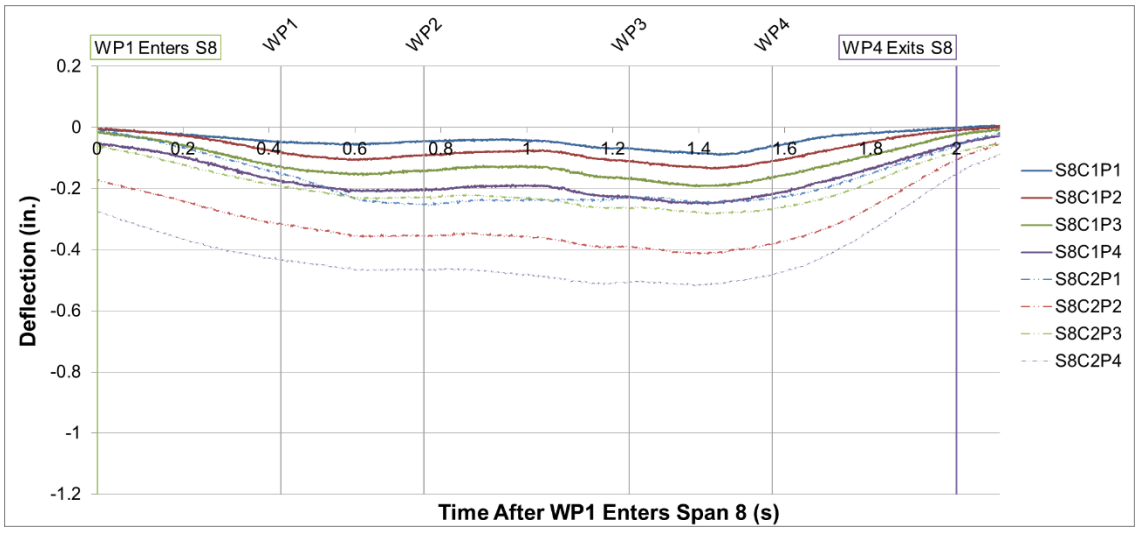

Figure A-68: Span 8 Test 2 Net Stringer Midspan Deflection

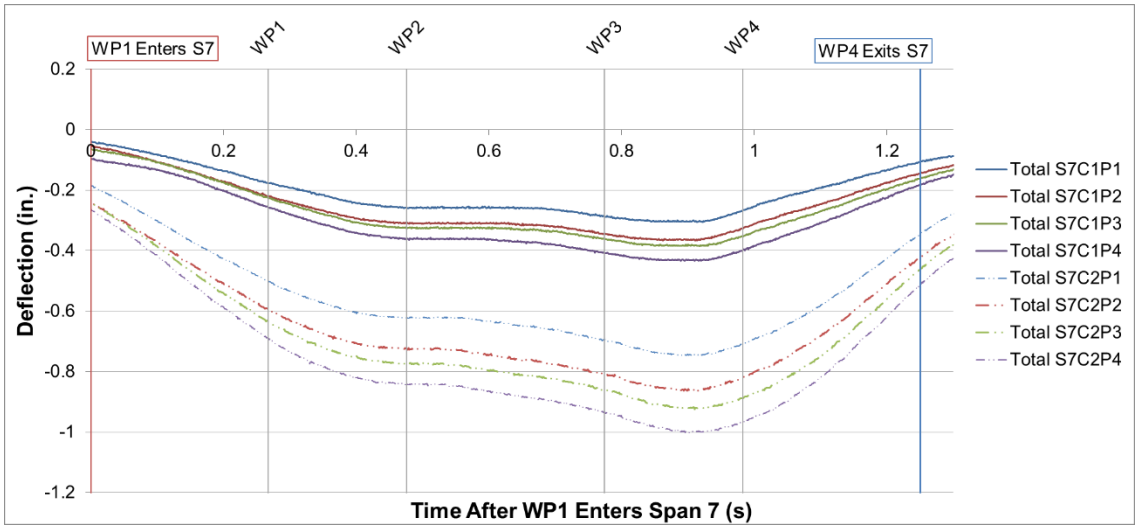

Figure A-69: Span 7 Test 3 Work Train Midspan Stringer Deflection

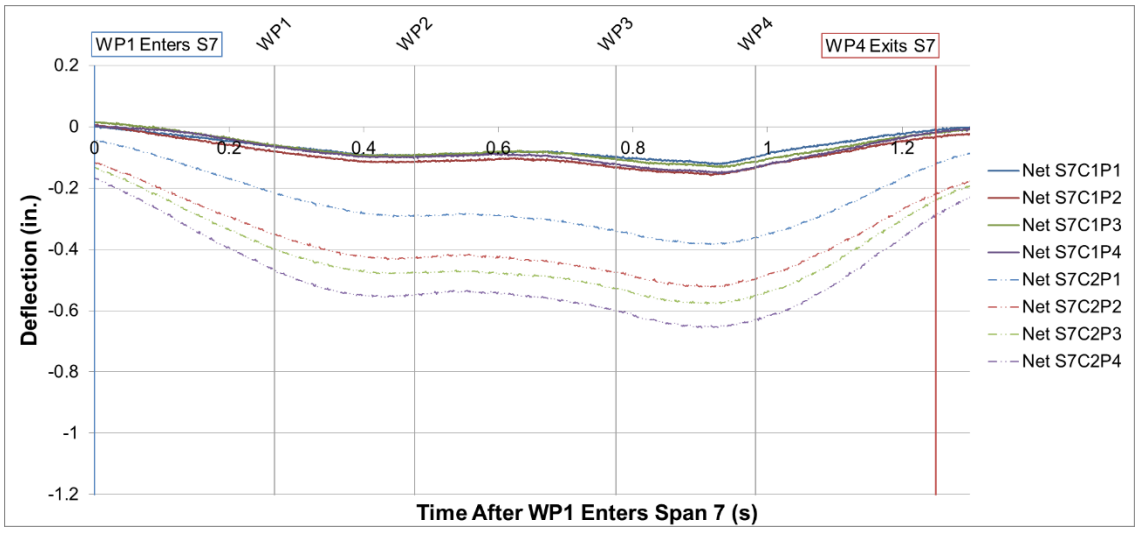

Figure A-70: Span 7 Test 3 Net Stringer Midspan Deflection

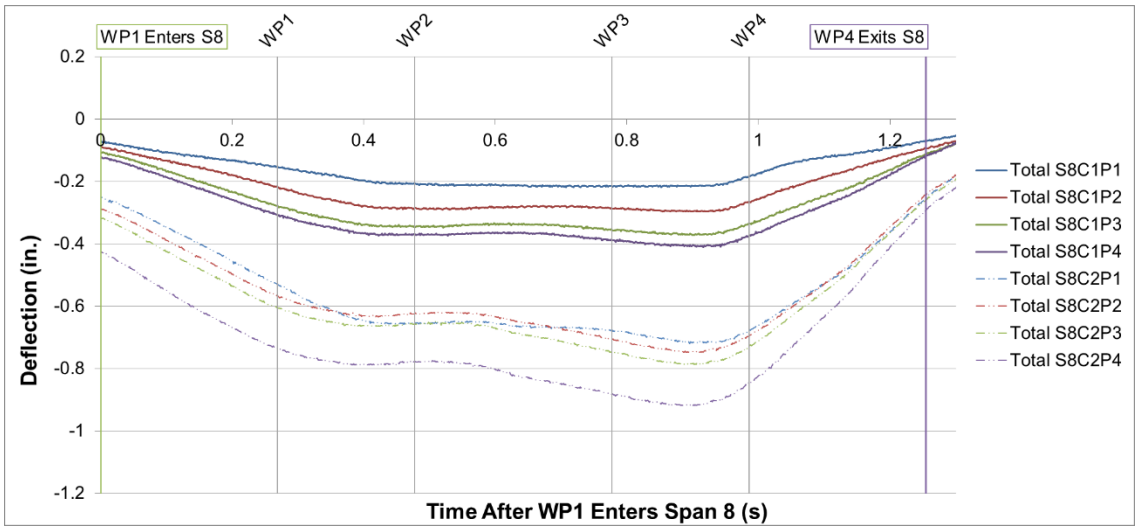

Figure A-71: Span 8 Test 3 Work Train Midspan Stringer Deflection

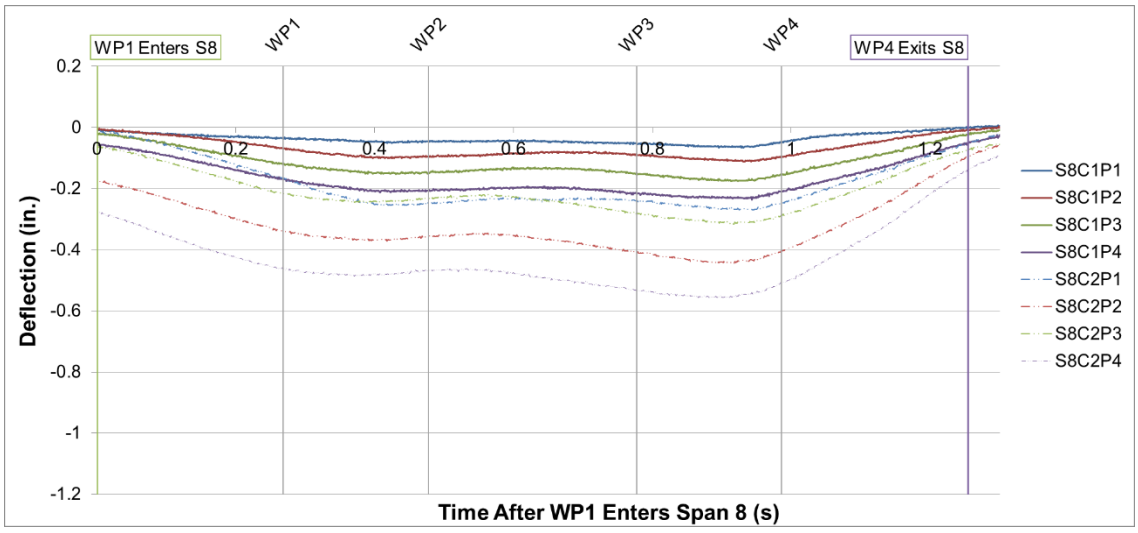

Figure A-72: Span 8 Test 3 Net Stringer Midspan Deflection

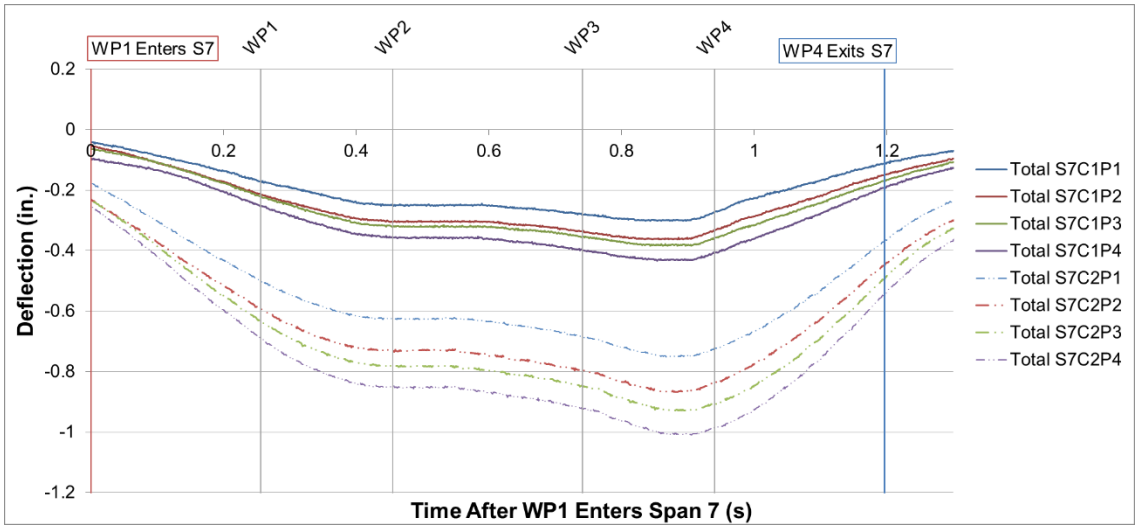

Figure A-73: Span 7 Test 4 Work Train Midspan Stringer Deflection

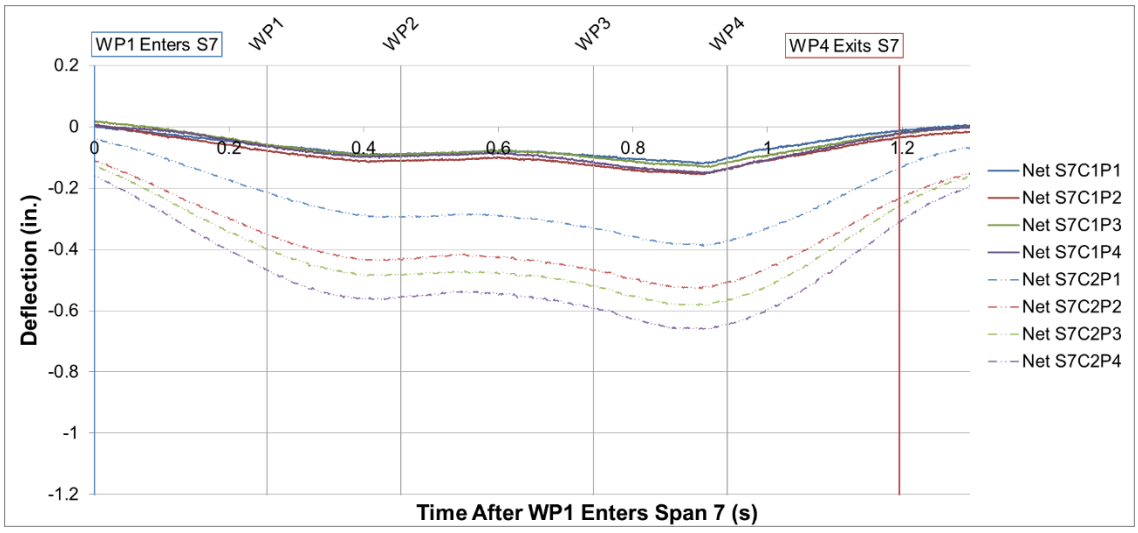

Figure A-74: Span 7 Test 4 Net Stringer Midspan Deflection

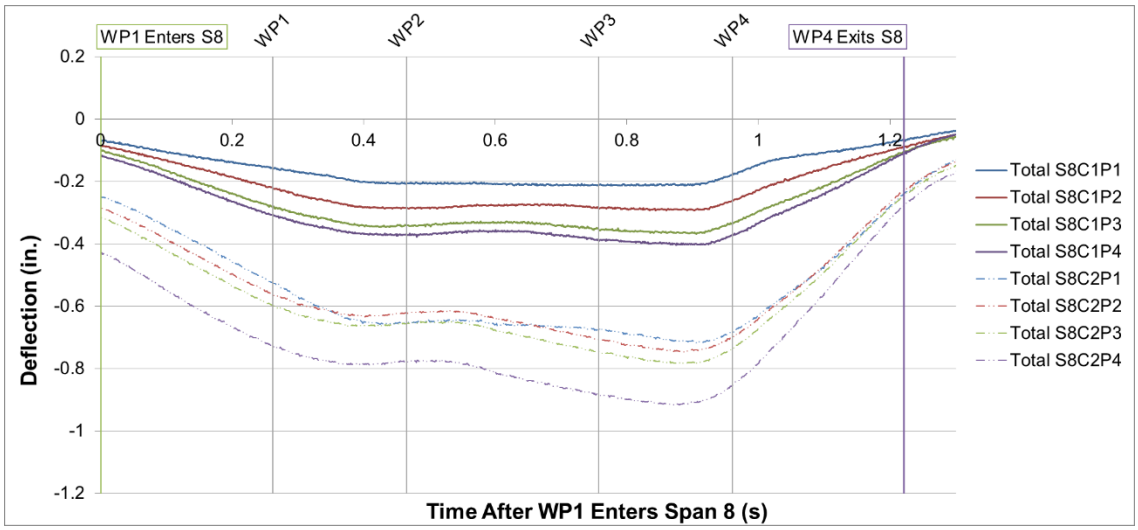

Figure A-75: Span 8 Test 4 Work Train Midspan Stringer Deflection

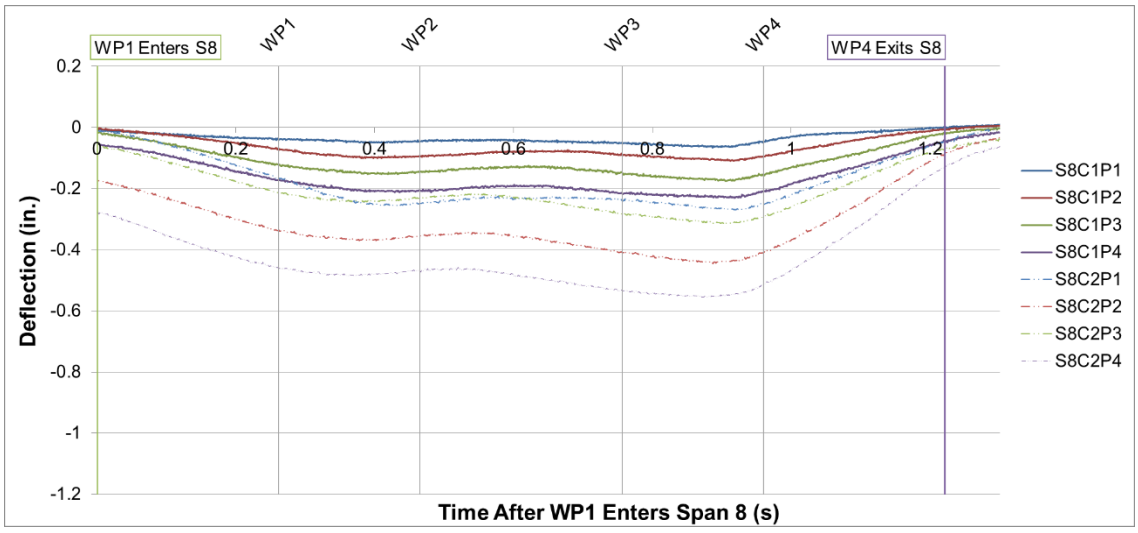

Figure A-76: Span 8 Test 4 Net Stringer Midspan Deflection

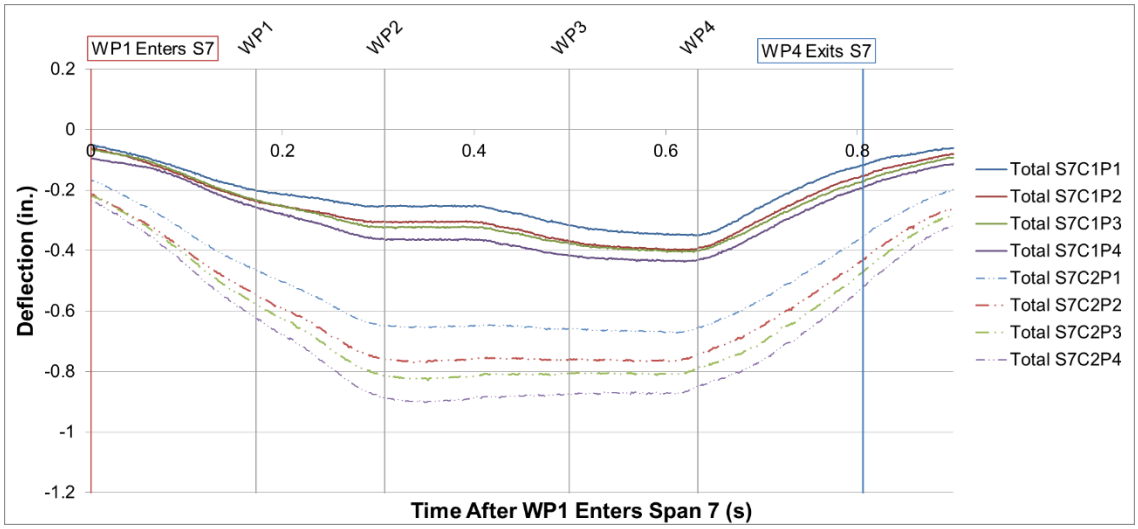

Figure A-77: Span 7 Test 5 Work Train Midspan Stringer Deflection

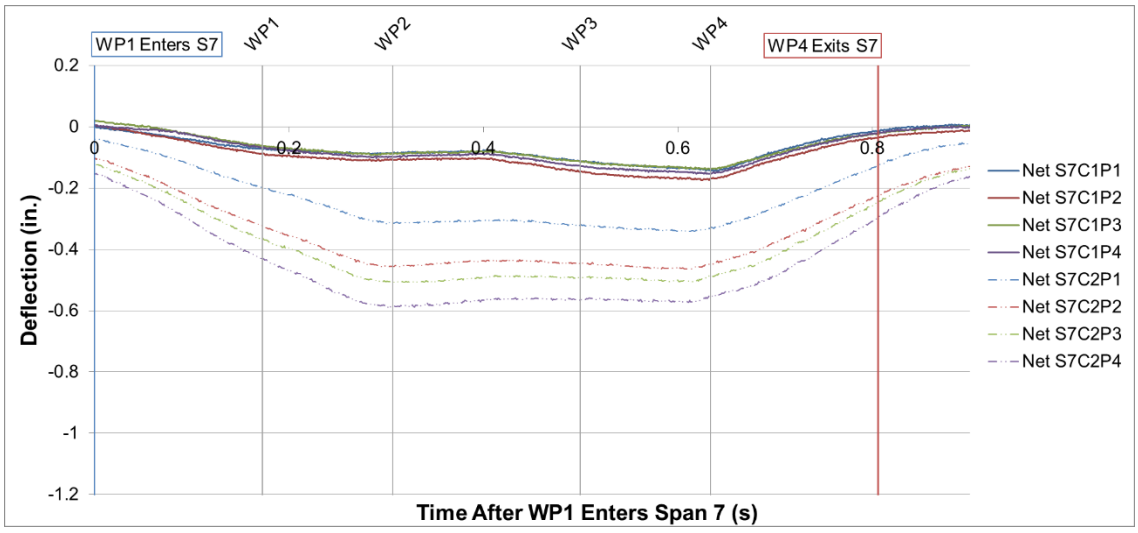

Figure A-78: Span 7 Test 5 Net Stringer Midspan Deflection

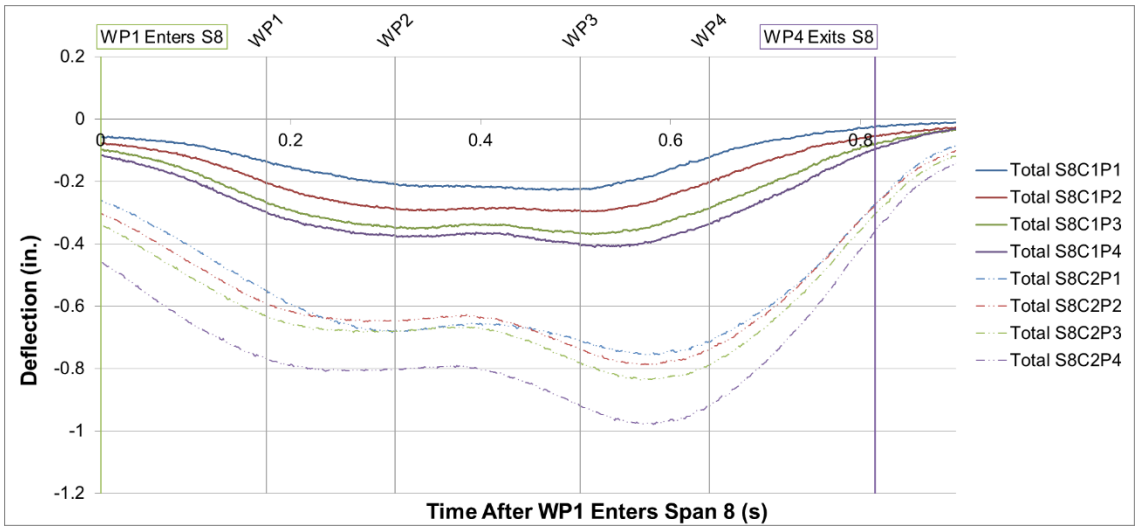

Figure A-79: Span 8 Test 5 Work Train Midspan Stringer Deflection

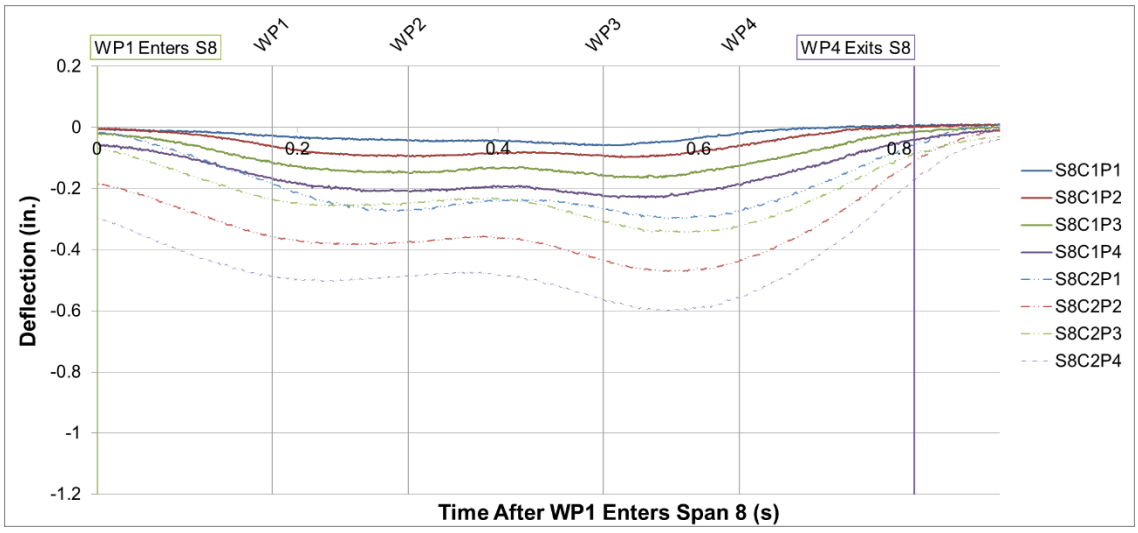

Figure A-80: Span 8 Test 5 Net Stringer Midspan Deflection

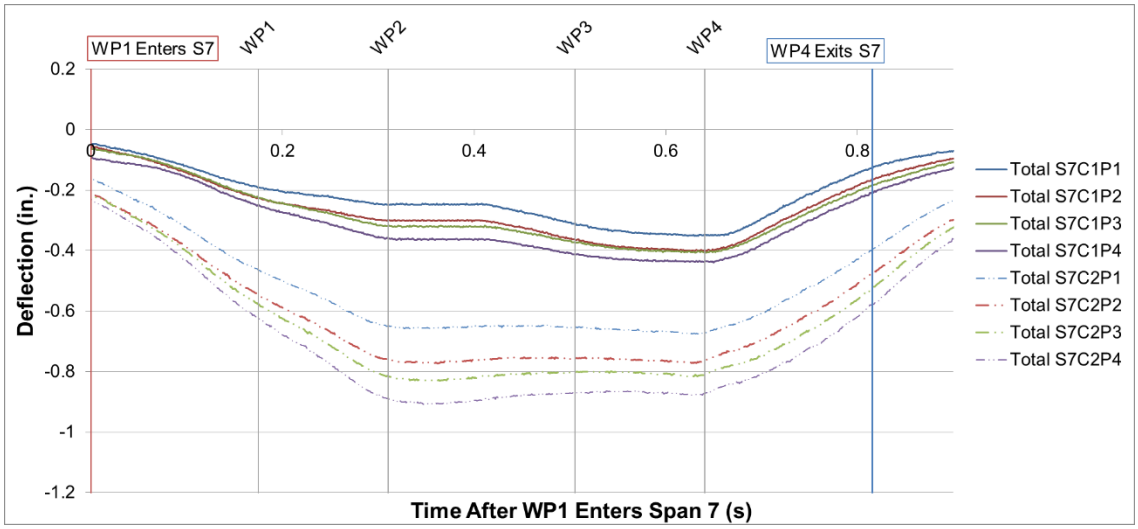

Figure A-81: Span 7 Test 6 Work Train Midspan Stringer Deflection

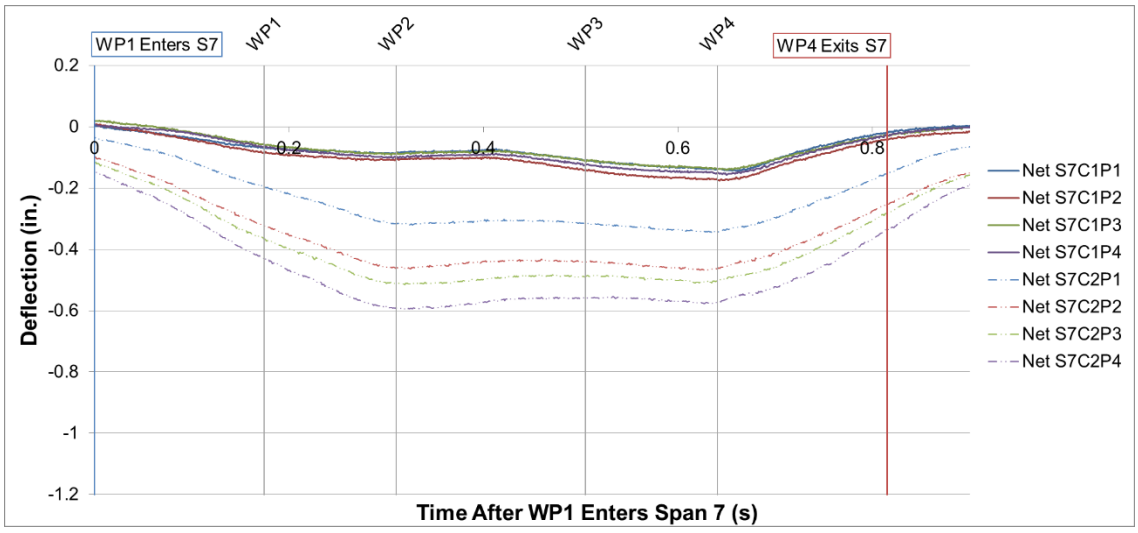

Figure A-82: Span 7 Test 6 Net Stringer Midspan Deflection

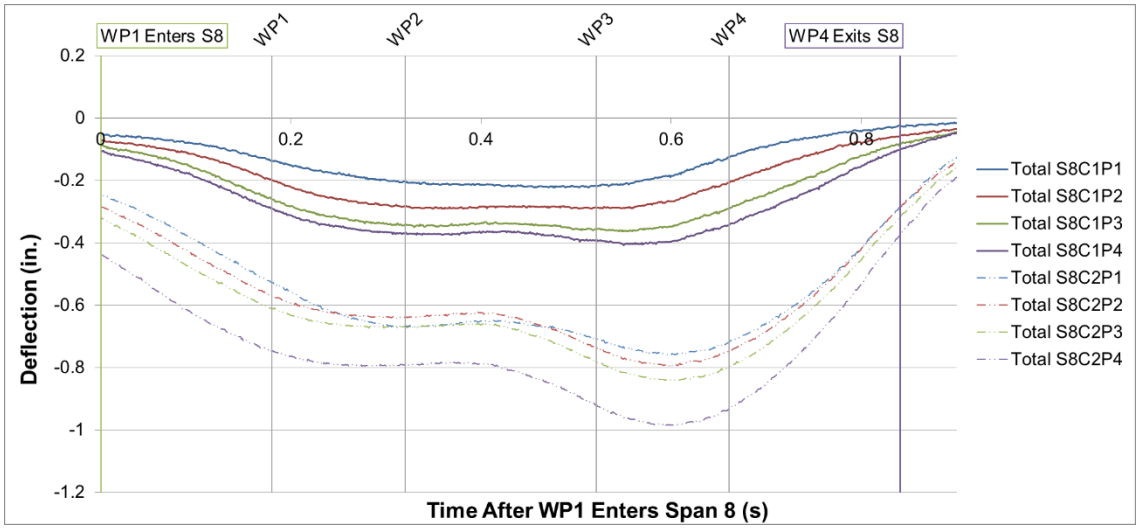

Figure A-83: Span 8 Test 6 Work Train Midspan Stringer Deflection

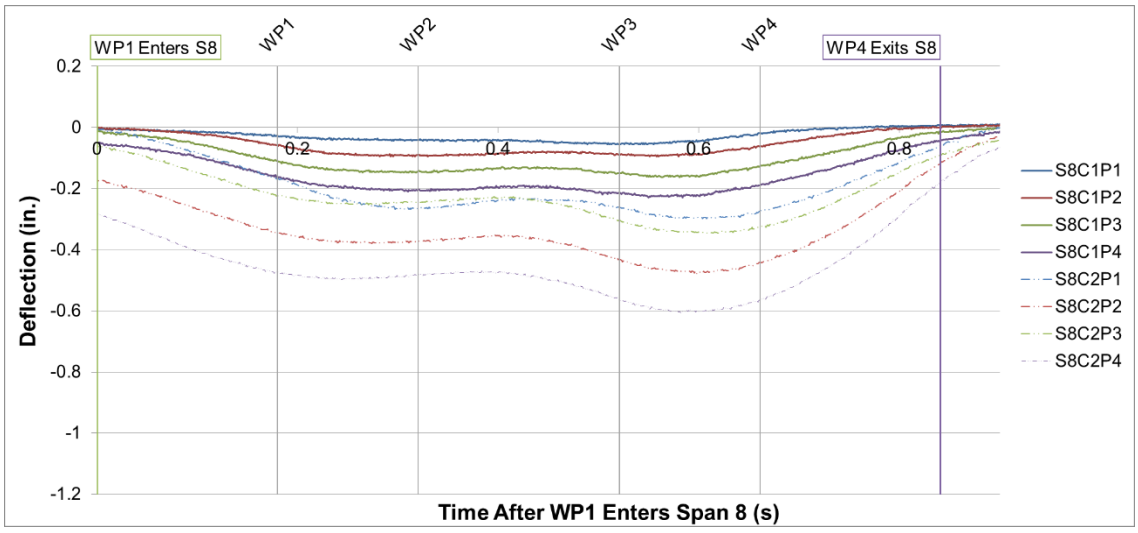

Figure A-84: Span 8 Test 6 Net Stringer Midspan Deflection

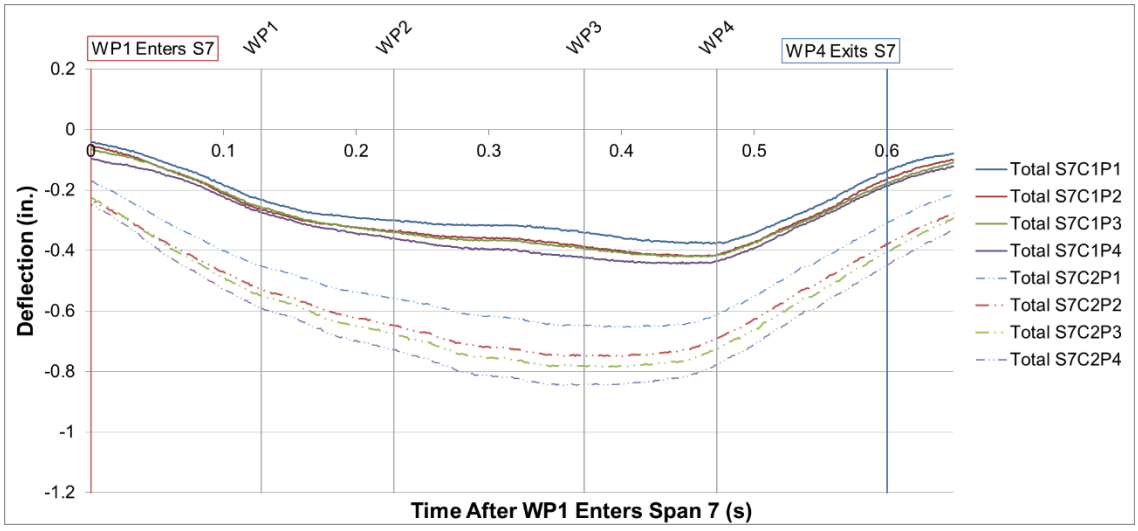

Figure A-85: Span 7 Test 7 Work Train Midspan Stringer Deflection

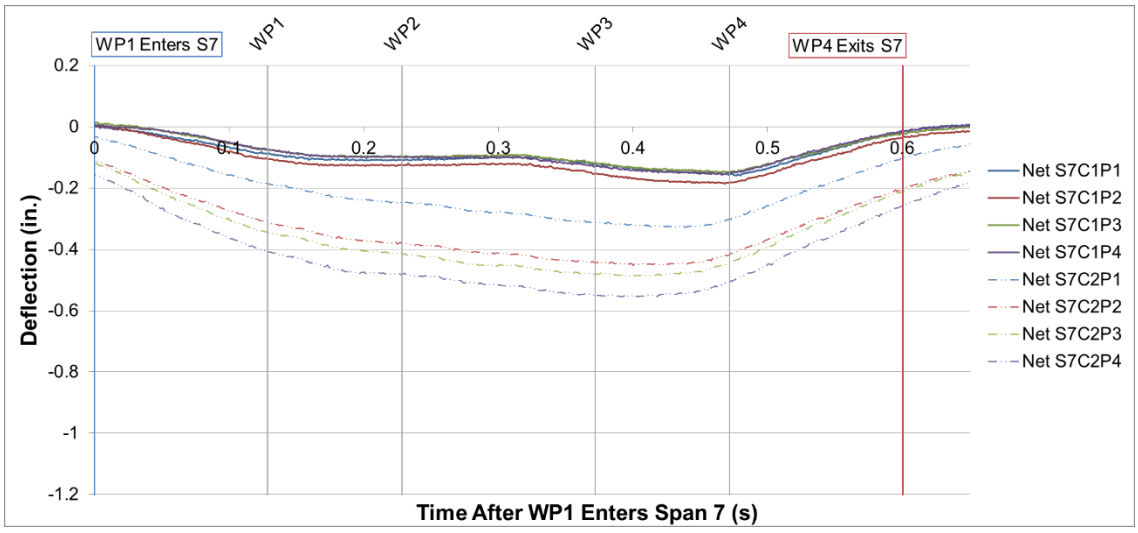

Figure A-86: Span 7 Test 7 Net Stringer Midspan Deflection

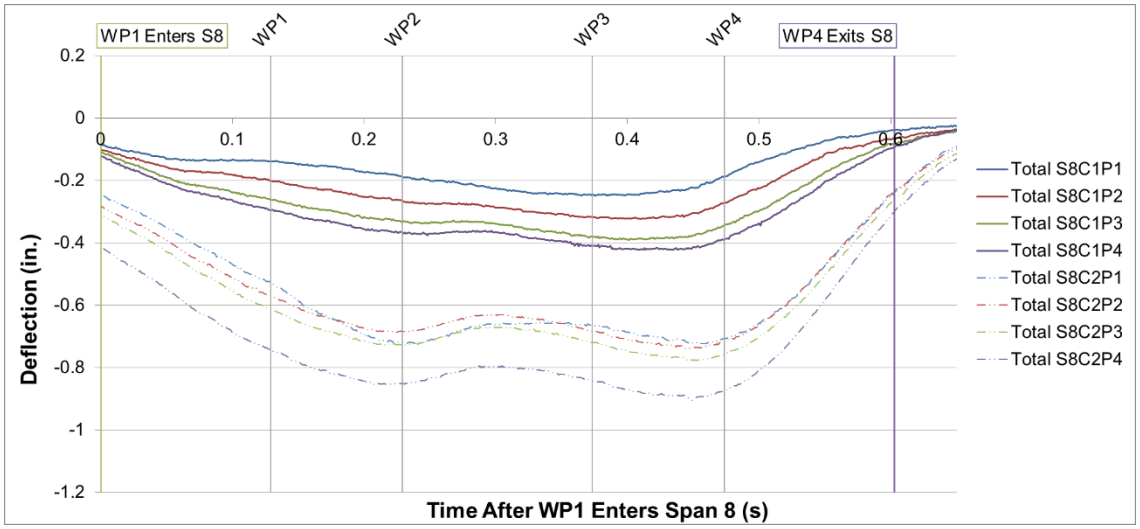

Figure A-87: Span 8 Test 7 Work Train Midspan Stringer Deflection

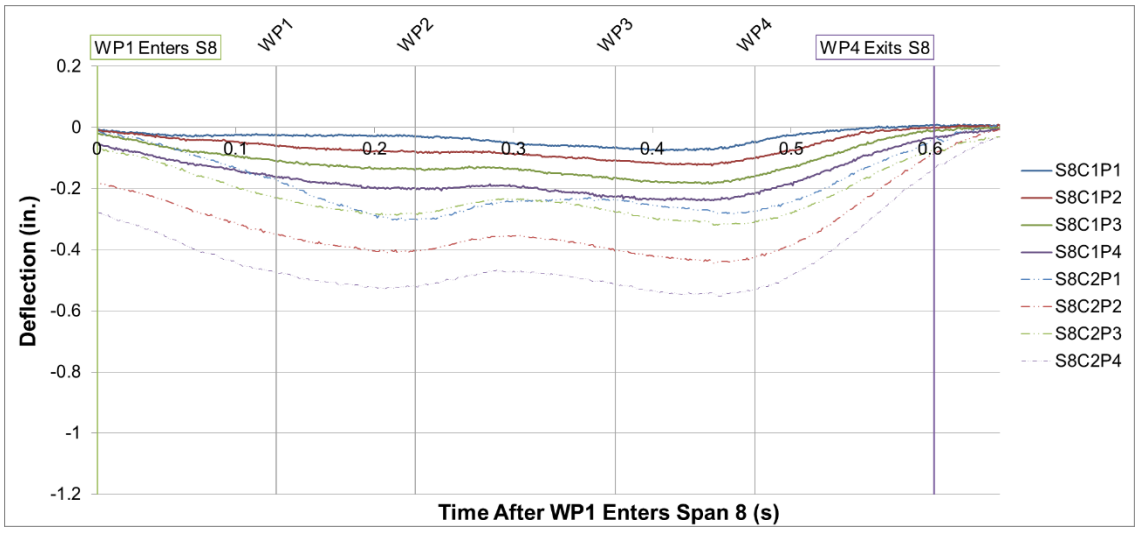

Figure A-88: Span 8 Test 7 Net Stringer Midspan Deflection

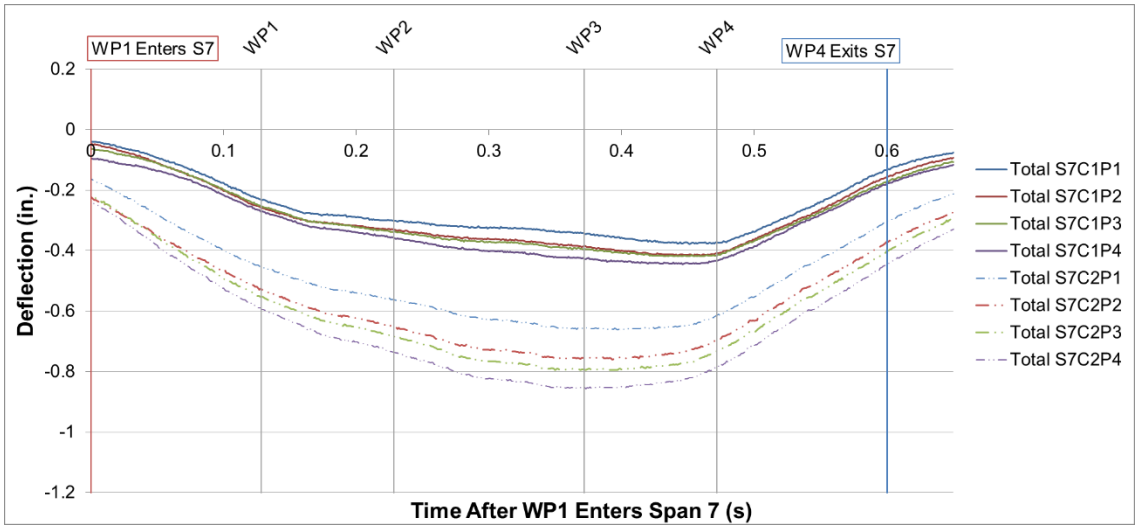

Figure A-89: Span 7 Test 8 Work Train Midspan Stringer Deflection

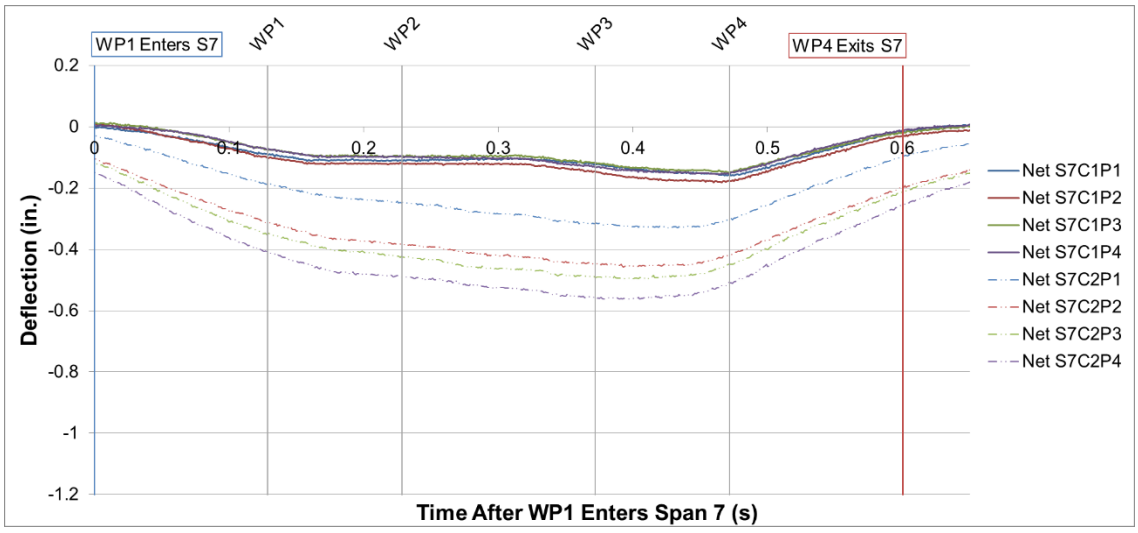

Figure A-90: Span 7 Test 8 Net Stringer Midspan Deflection

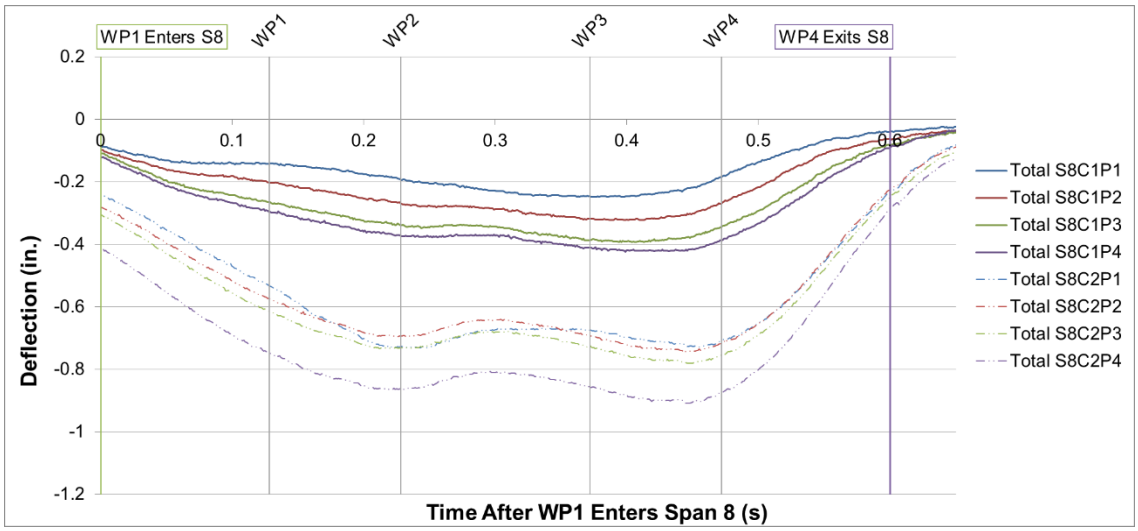

Figure A-91: Span 8 Test 8 Work Train Midspan Stringer Deflection

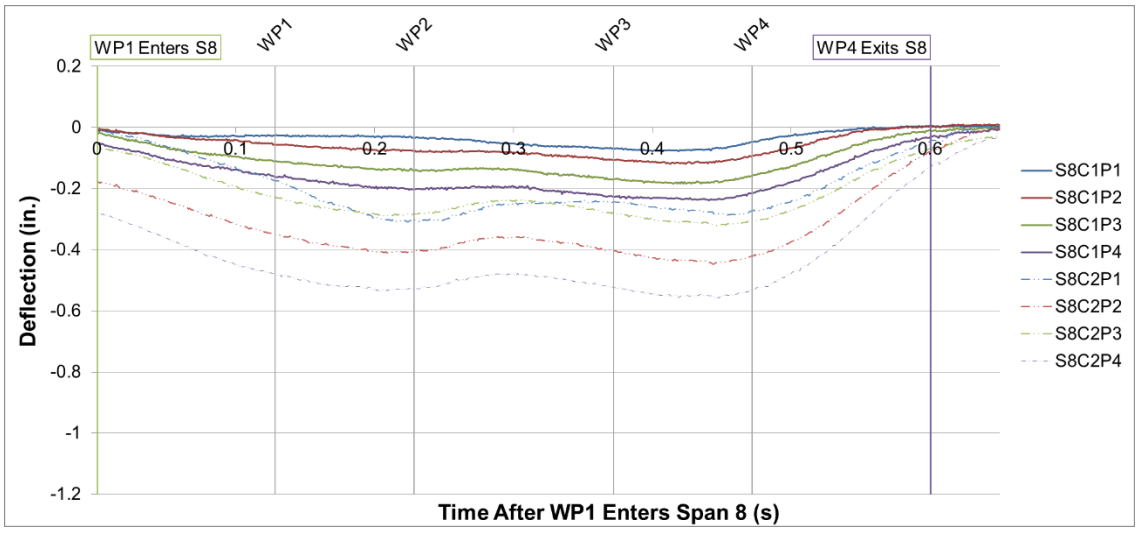

Figure A-92: Span 8 Test 8 Net Stringer Midspan Deflection

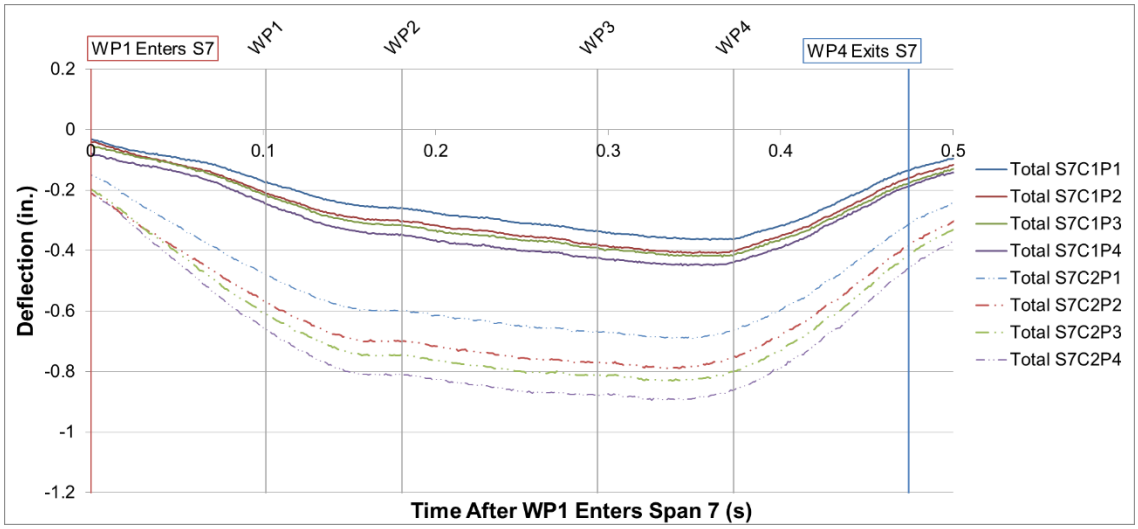

Figure A-93: Span 7 Test 9 Work Train Midspan Stringer Deflection

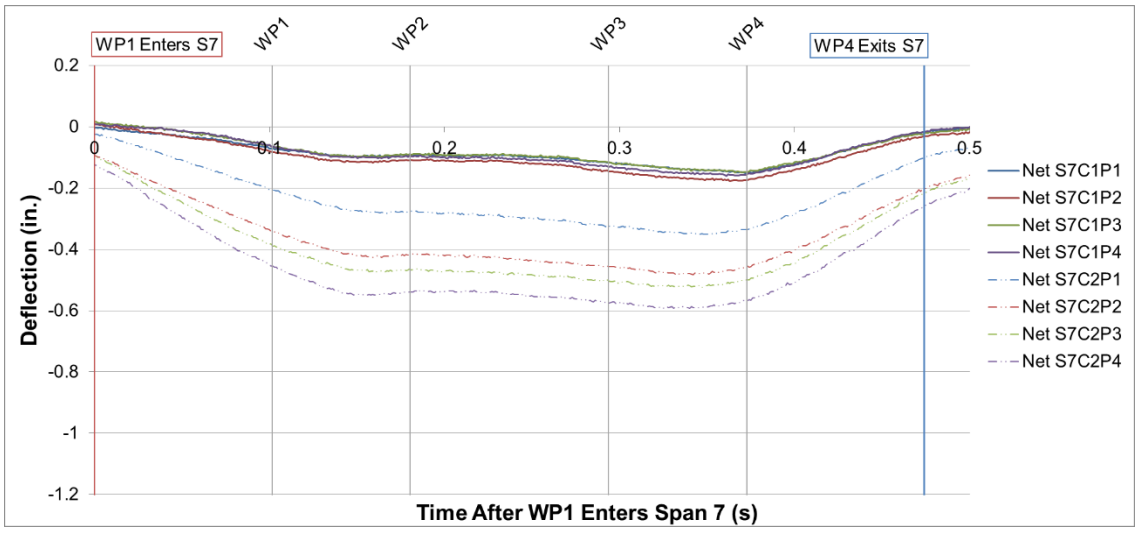

Figure A-94: Span 7 Test 9 Net Stringer Midspan Deflection

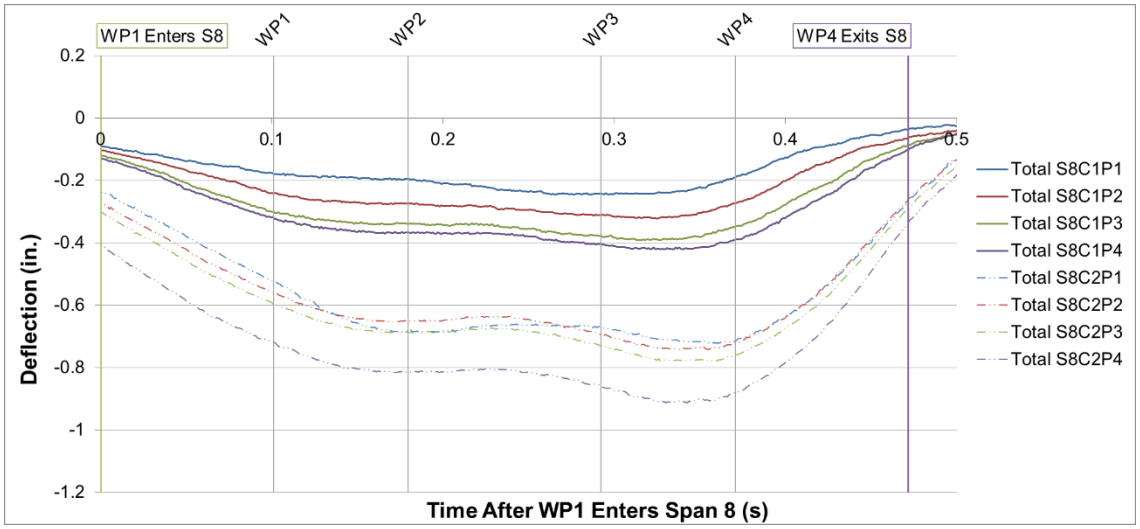

Figure A-95: Span 8 Test 9 Work Train Midspan Stringer Deflection

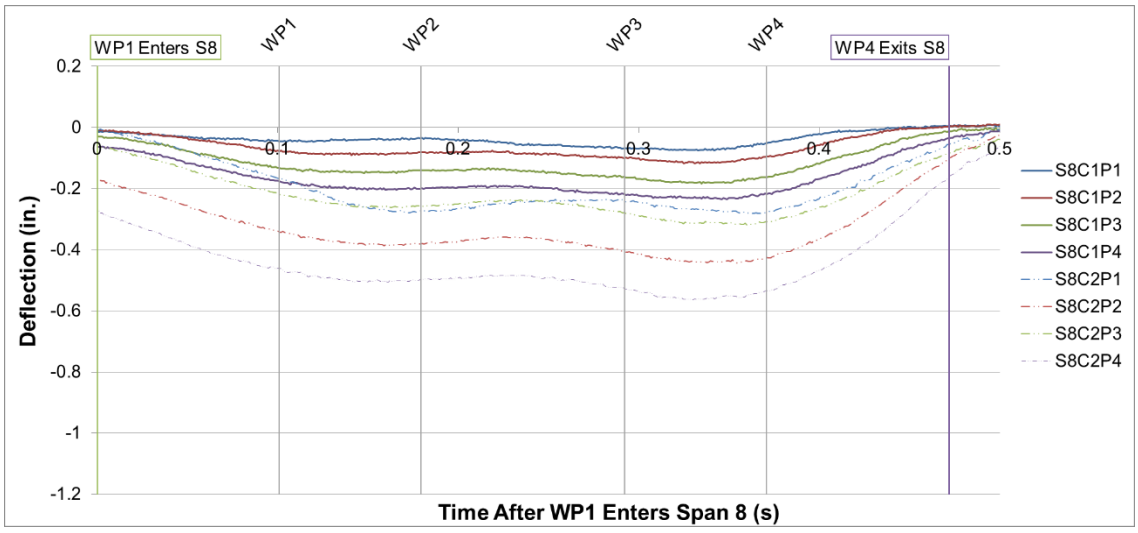

Figure A-96: Span 8 Test 9 Net Stringer Midspan Deflection

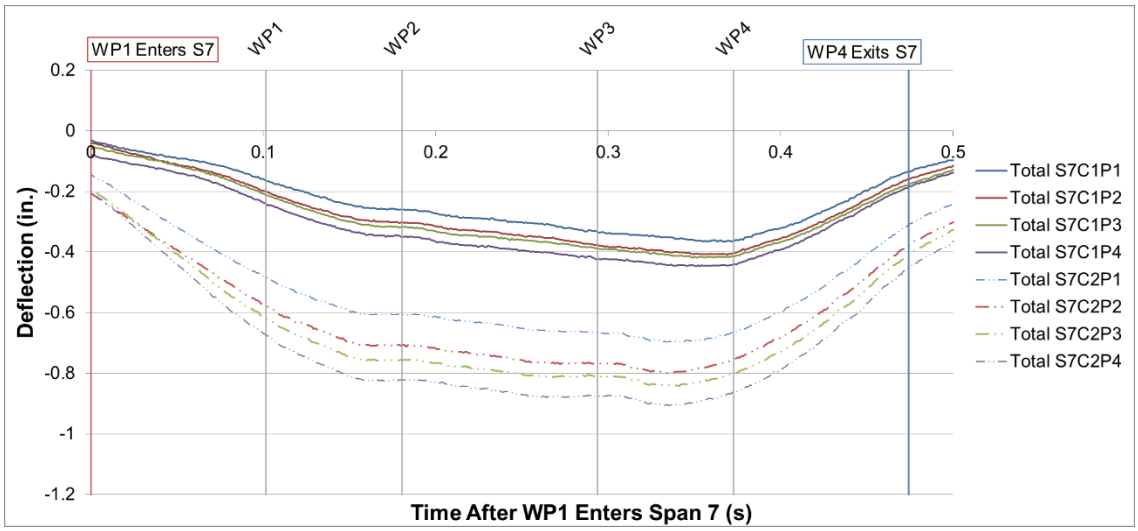

Figure A-97: Span 7 Test 10 Work Train Midspan Stringer Deflection

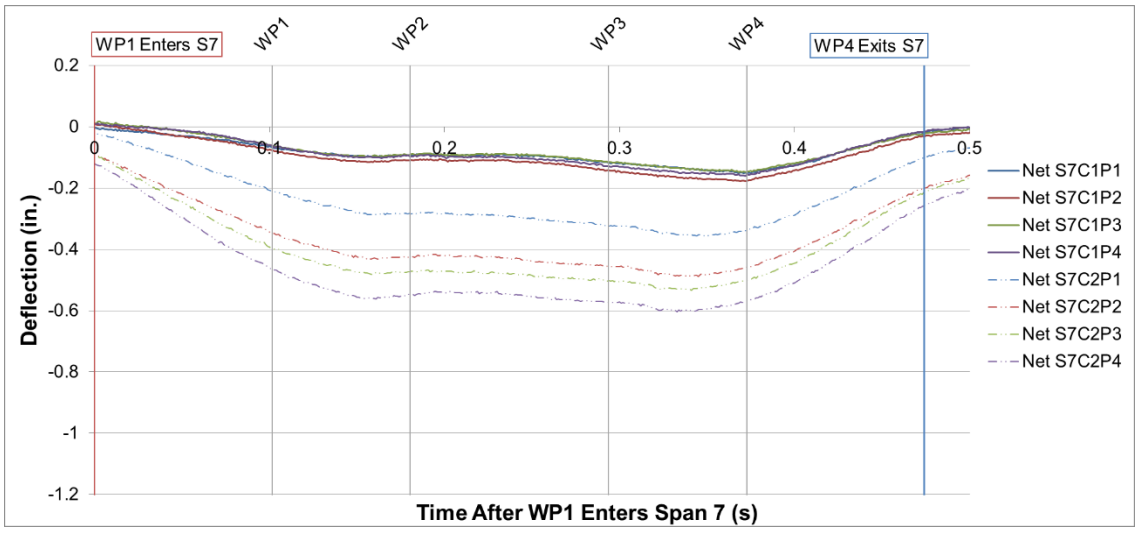

Figure A-98: Span 7 Test 10 Net Stringer Midspan Deflection

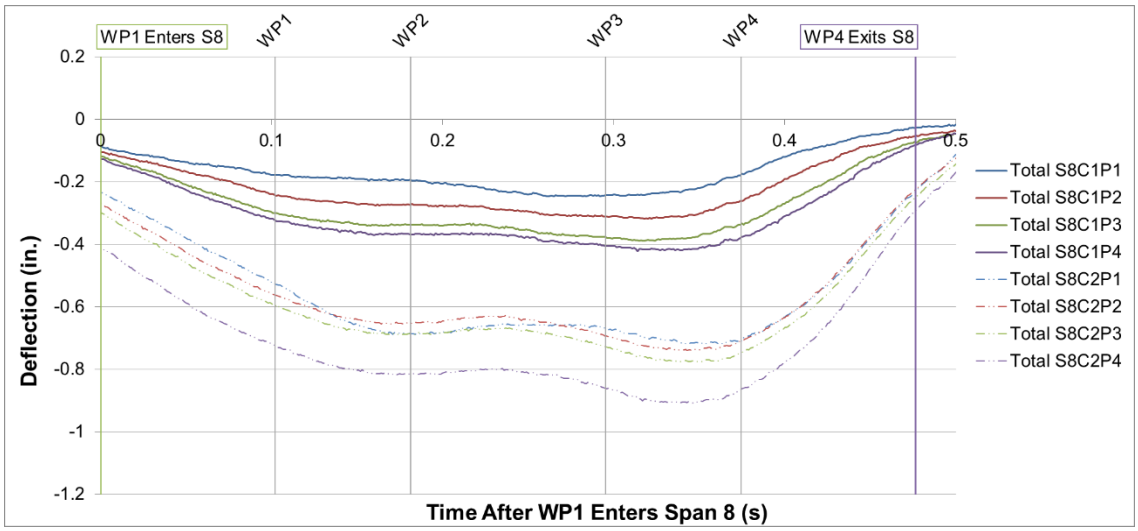

Figure A-99: Span 8 Test 10 Work Train Midspan Stringer Deflection

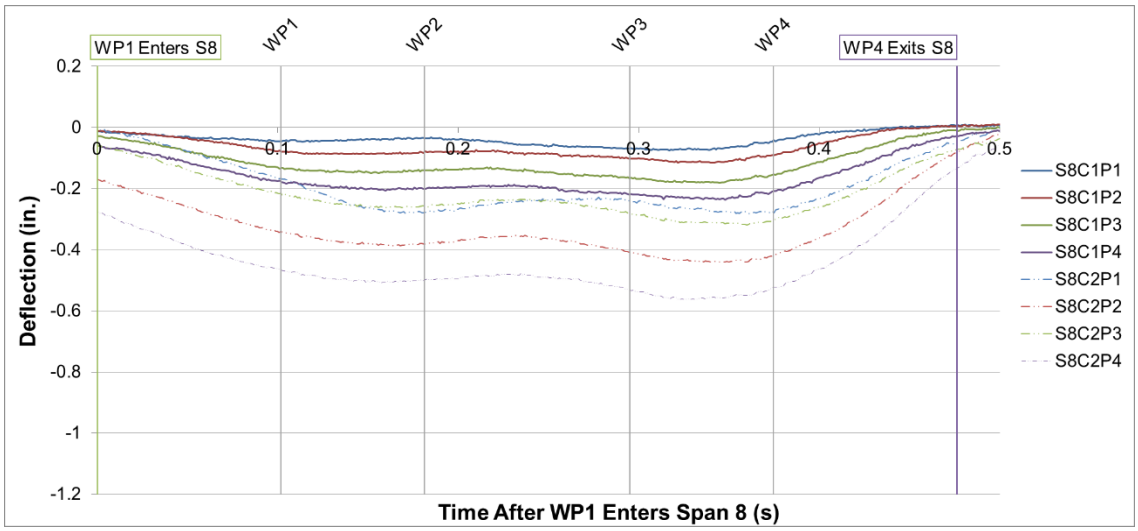

Figure A-100: Span 8 Test 10 Net Stringer Midspan Deflection

A.3 Computer Simulation Data

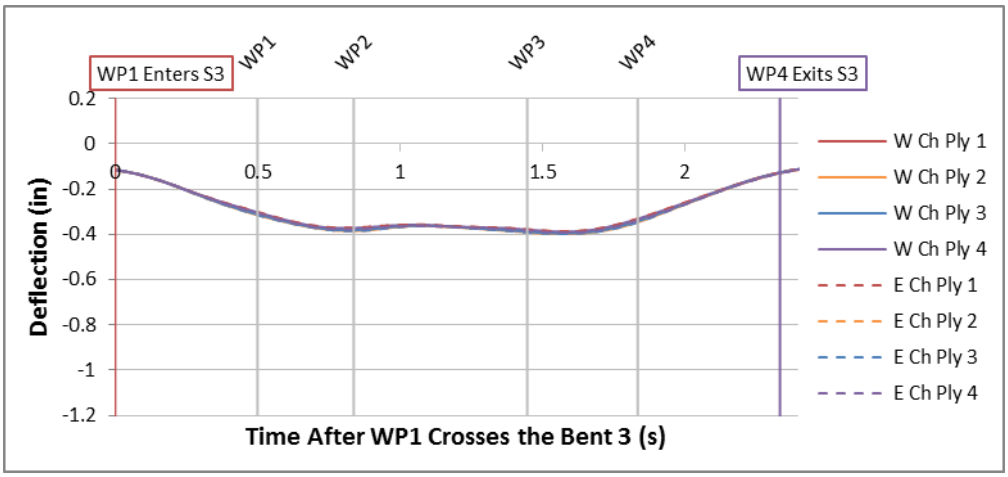

Figure A-101: IC1 10 MPH Total Midspan Stringer Deflections

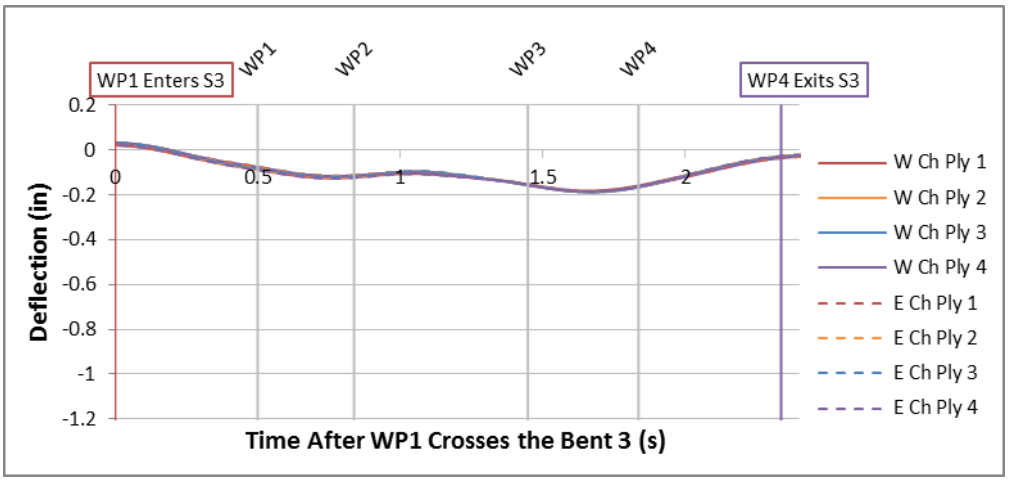

Figure A-102: IC1 10 MPH Net Midspan Stringer Deflections

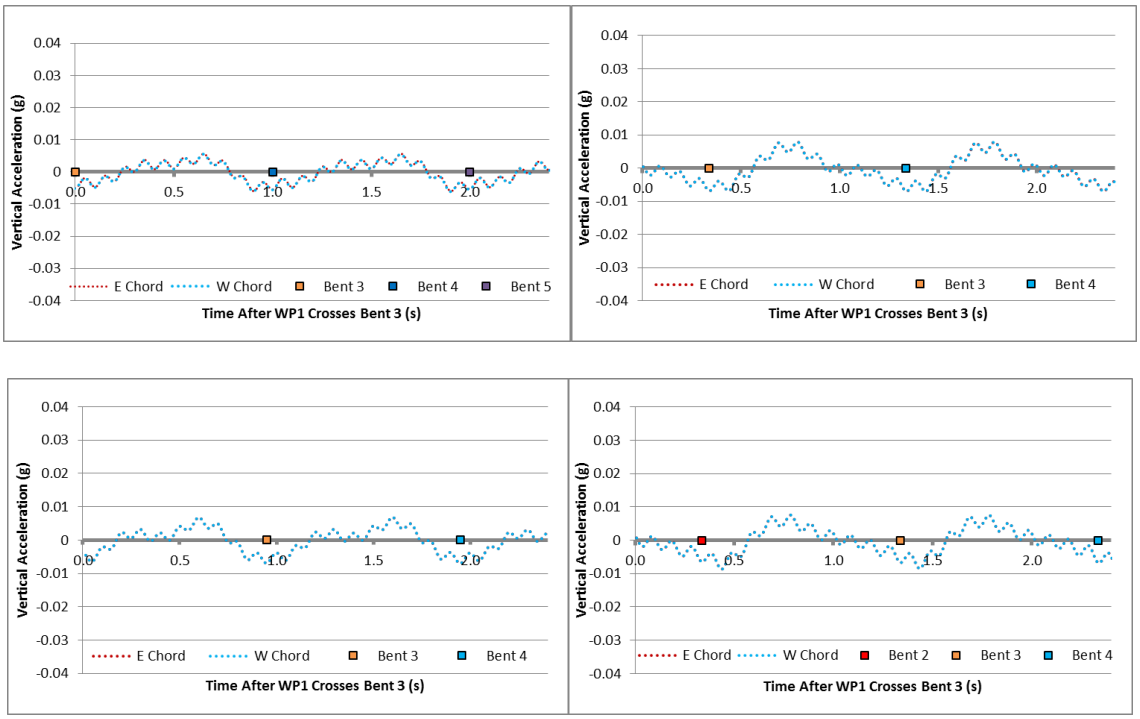

Figure A-103: IC1 10 MPH Wheel Path Accelerations: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

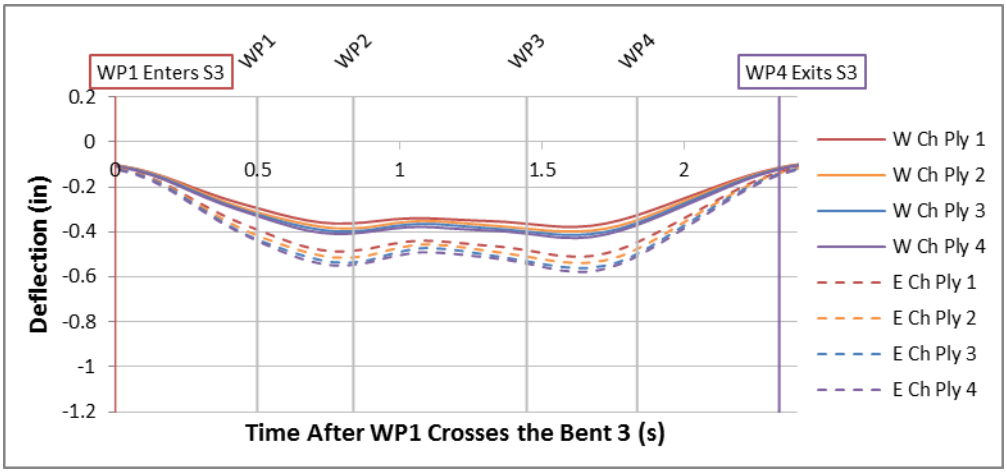

Figure A-104: IC2 10 MPH Total Midspan Stringer Deflections

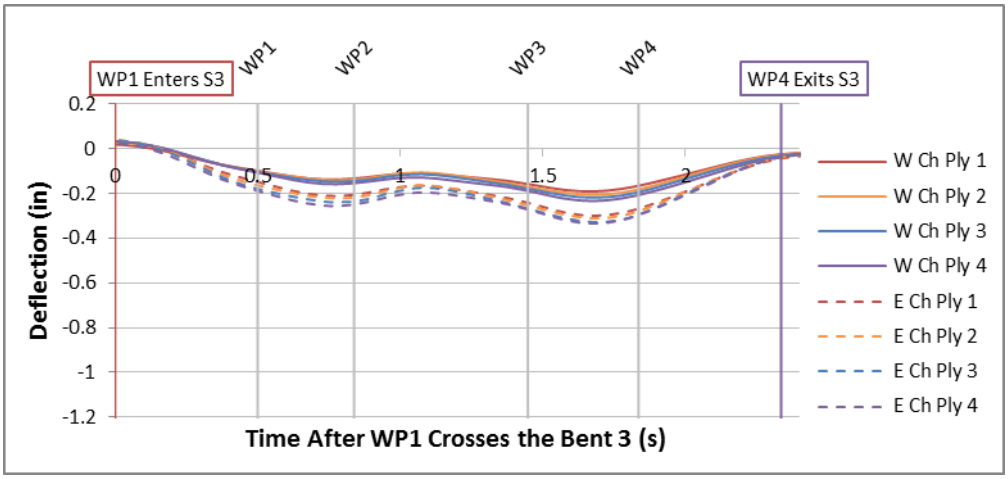

Figure A-105: IC2 10 MPH Net Midspan Stringer Deflections

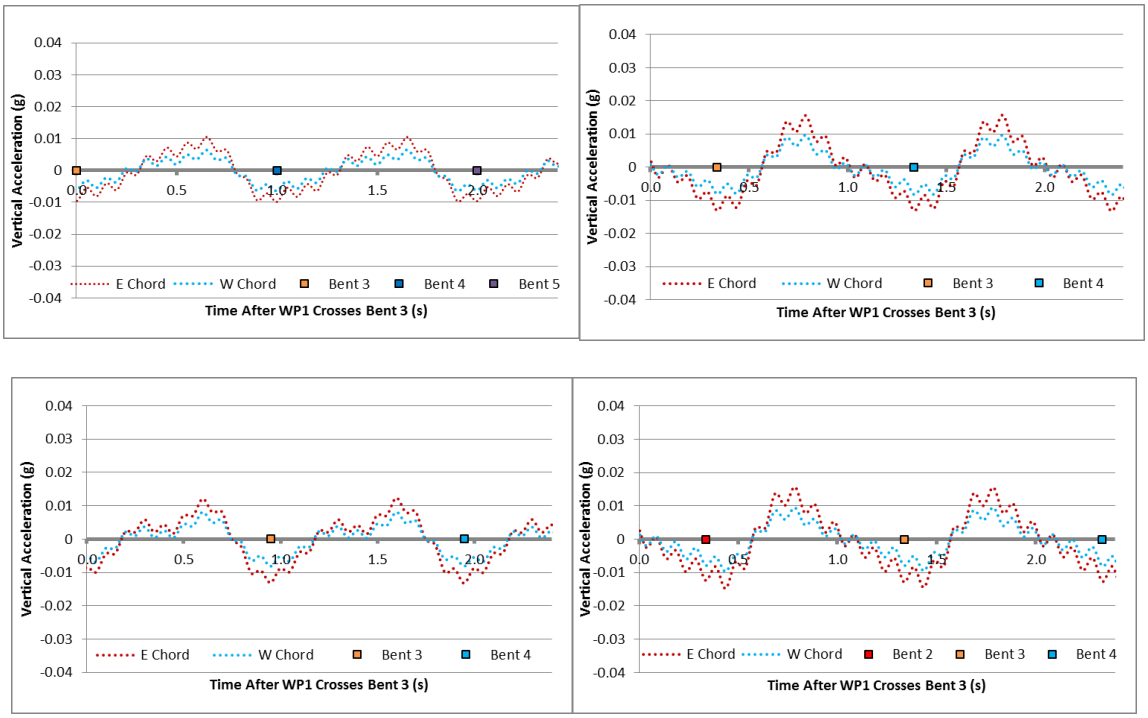

Figure A-106: IC2 10 MPH Wheel Path Accelerations: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

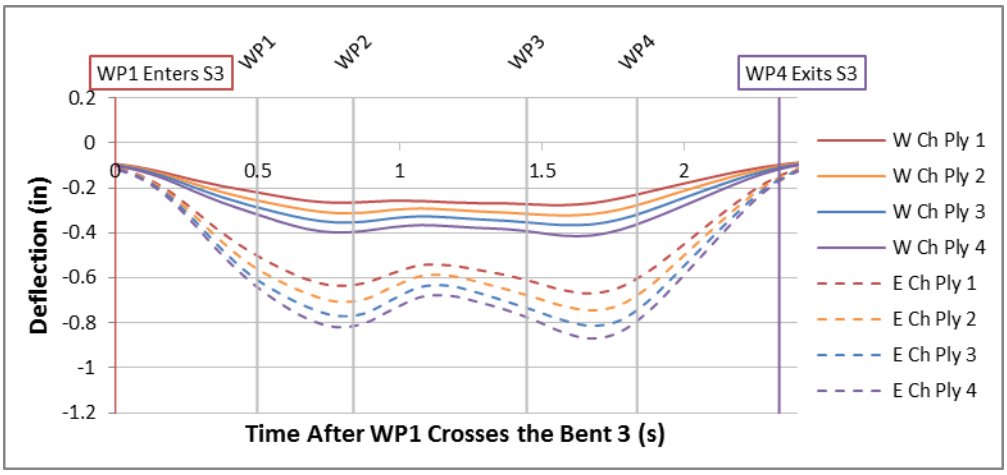

Figure A-107: IC3 10 MPH Total Midspan Stringer Deflections

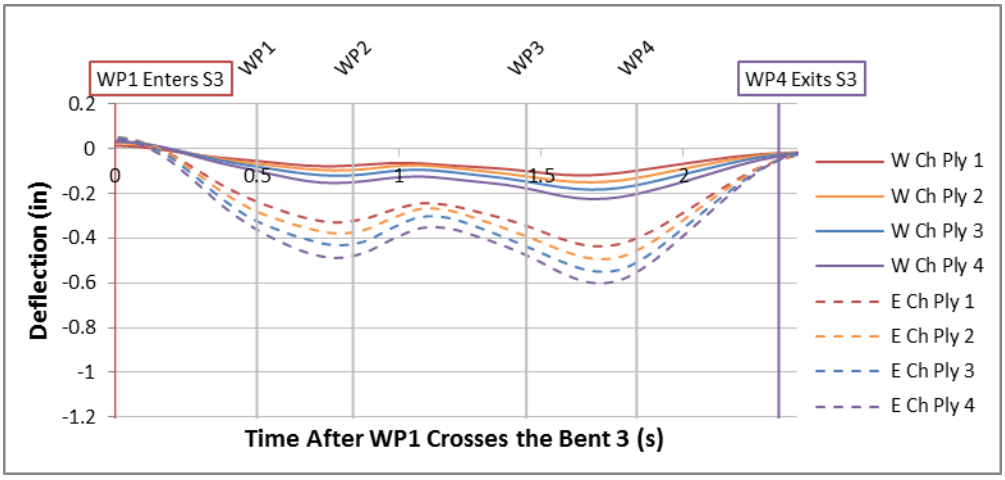

Figure A-108: IC3 10 MPH Net Midspan Stringer Deflections

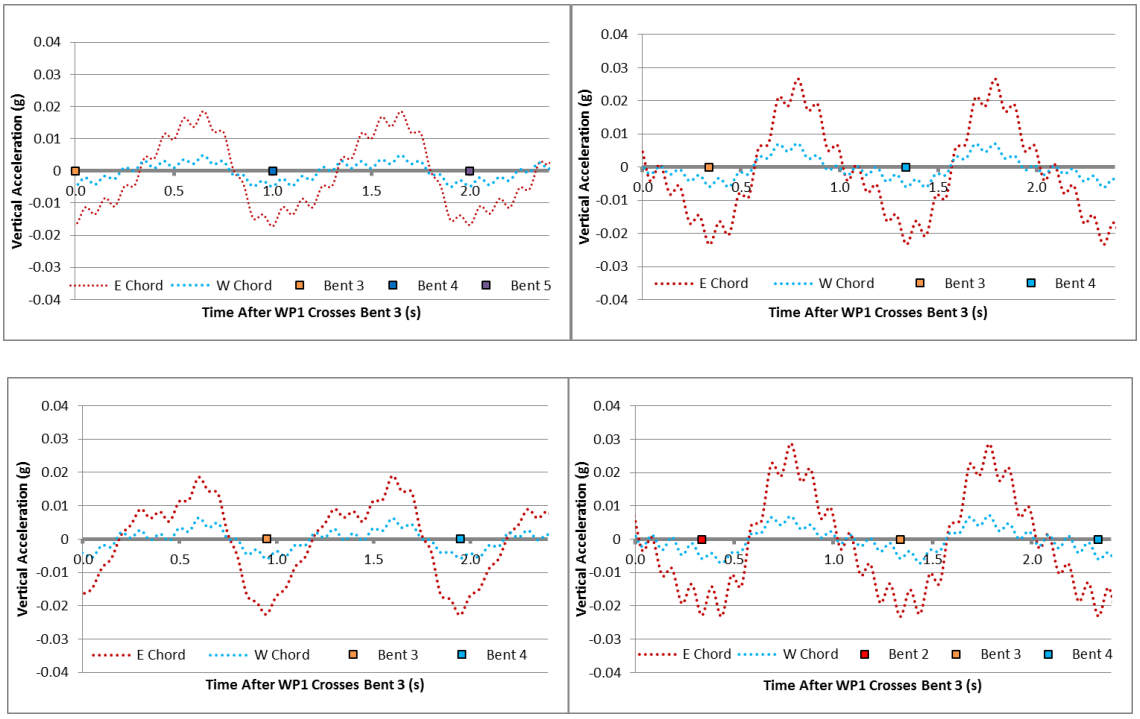

Figure A-109: IC3 10 MPH Wheel Path Accelerations: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

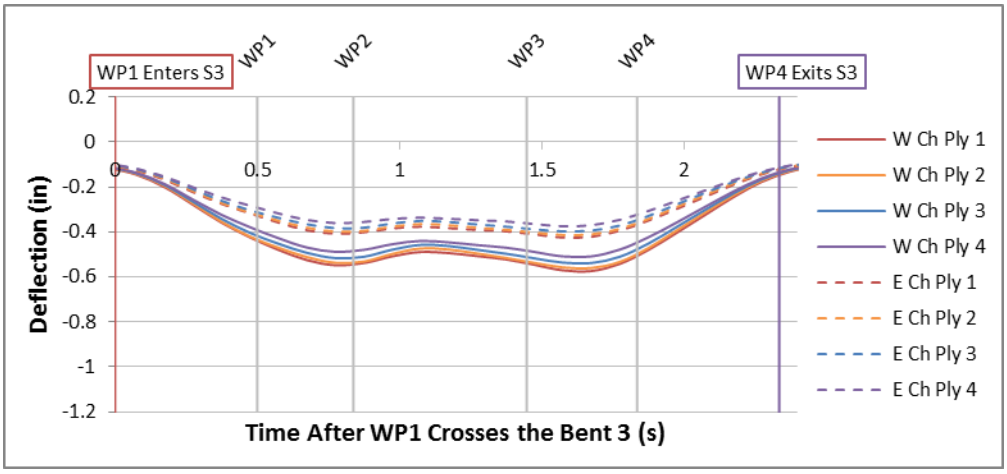

Figure A-110: IC4 10 MPH Total Midspan Stringer Deflections

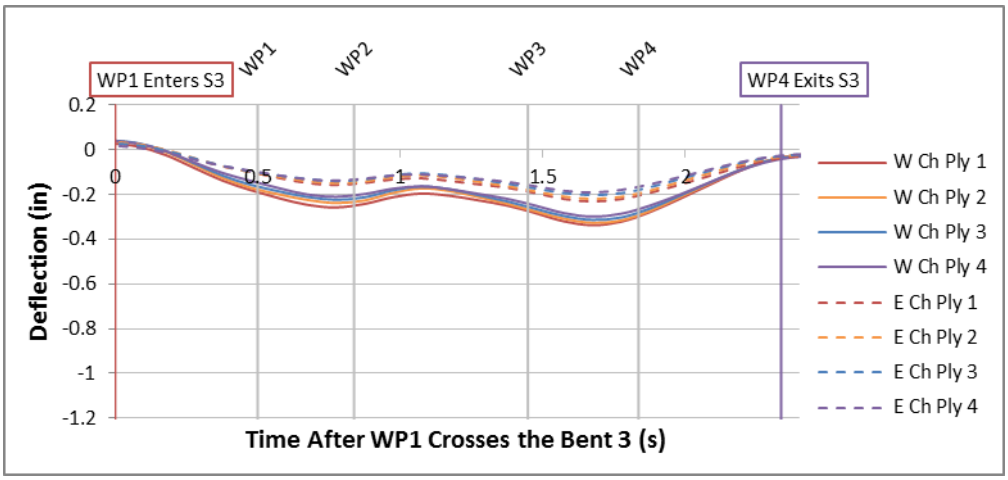

Figure A-111: IC4 10 MPH Net Midspan Stringer Deflections

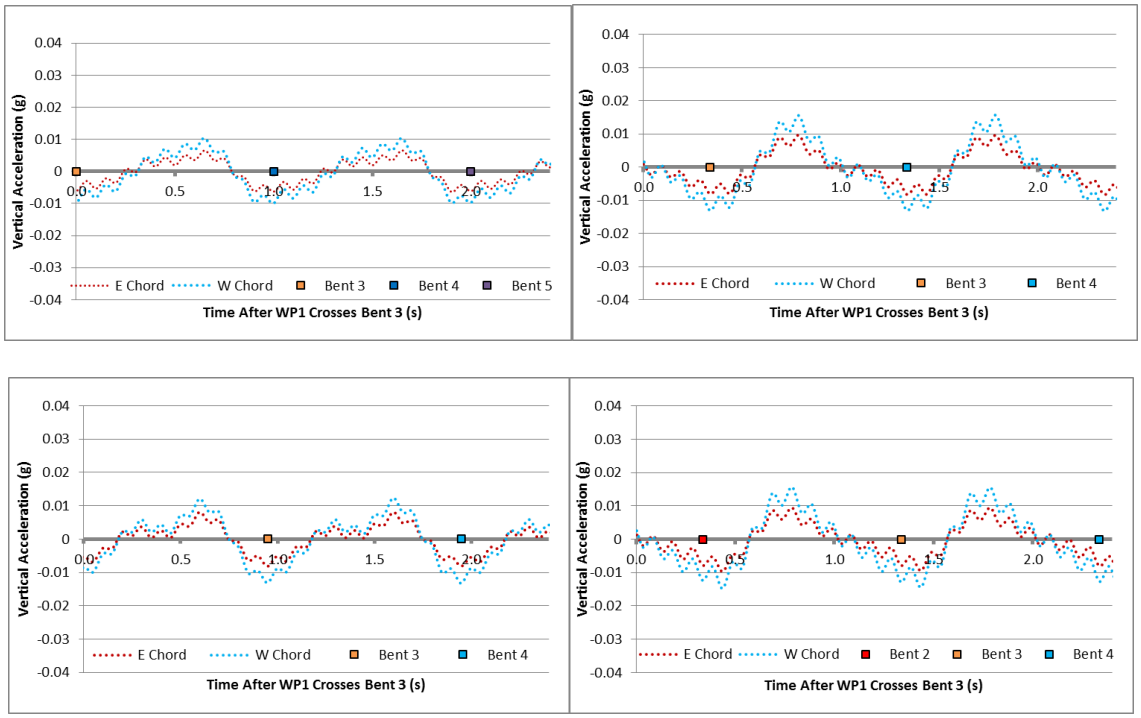

Figure A-112: IC4 10 MPH Wheel Path Accelerations: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

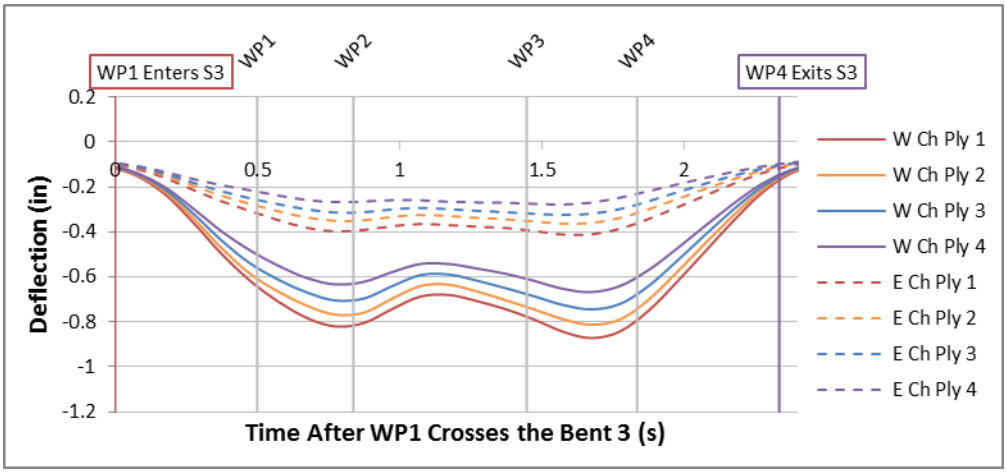

Figure A-113: IC5 10 MPH Total Midspan Stringer Deflections

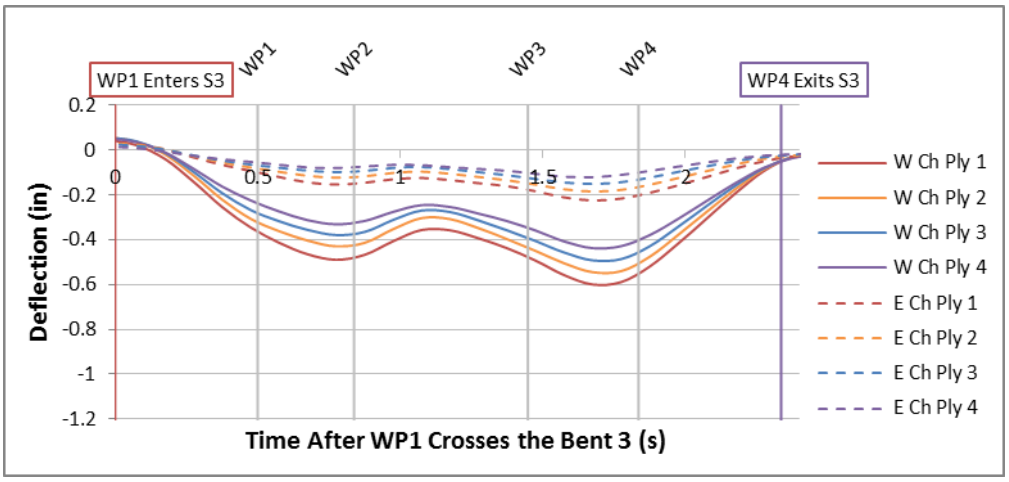

Figure A-114: IC5 10 MPH Net Midspan Stringer Deflections

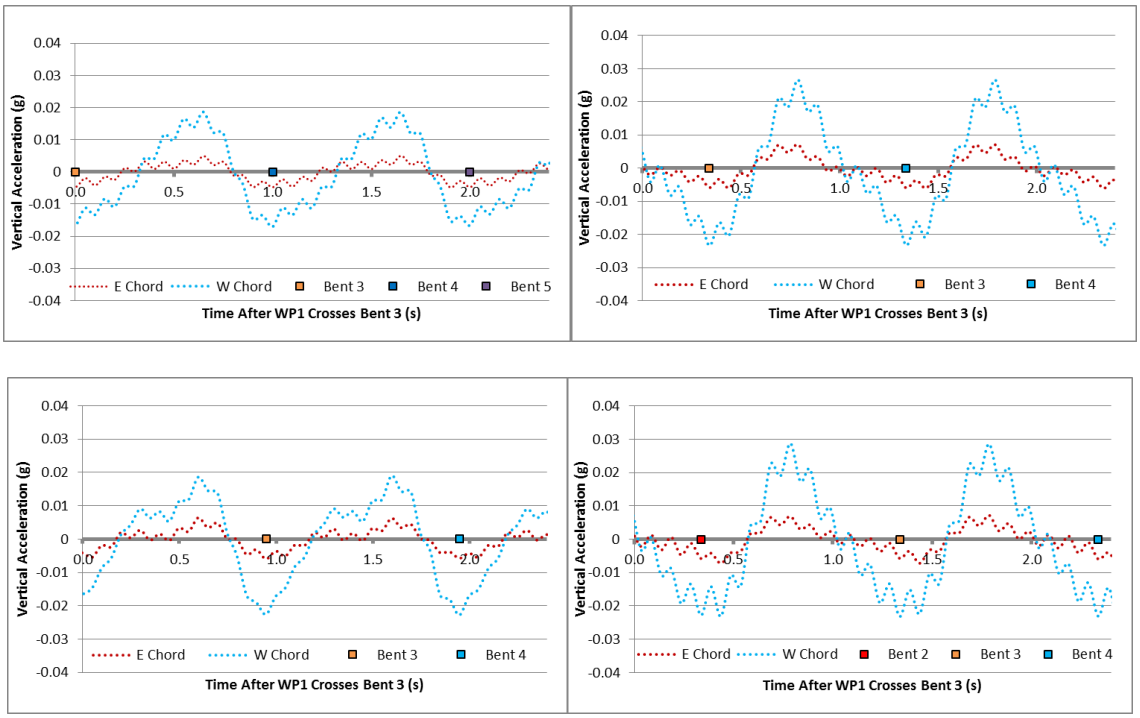

Figure A-115: IC5 10 MPH Wheel Path Accelerations: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

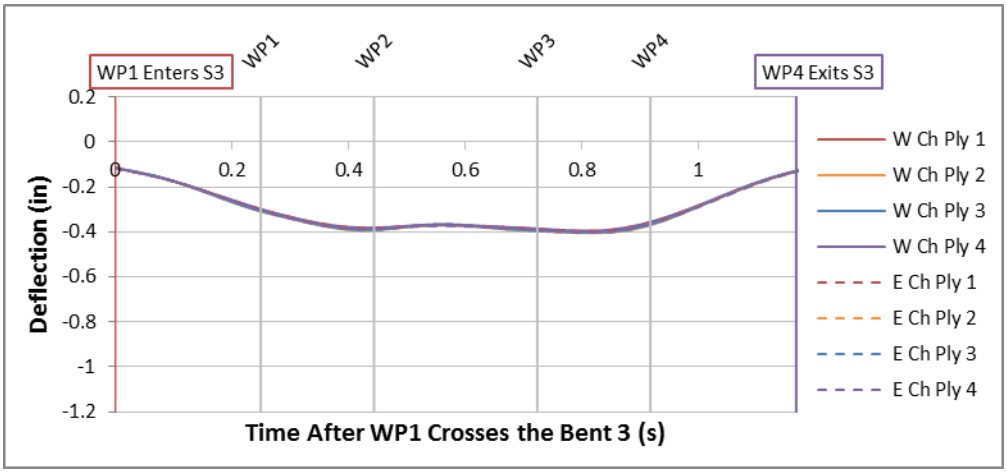

Figure A-116: IC1 20 MPH Total Midspan Stringer Deflections

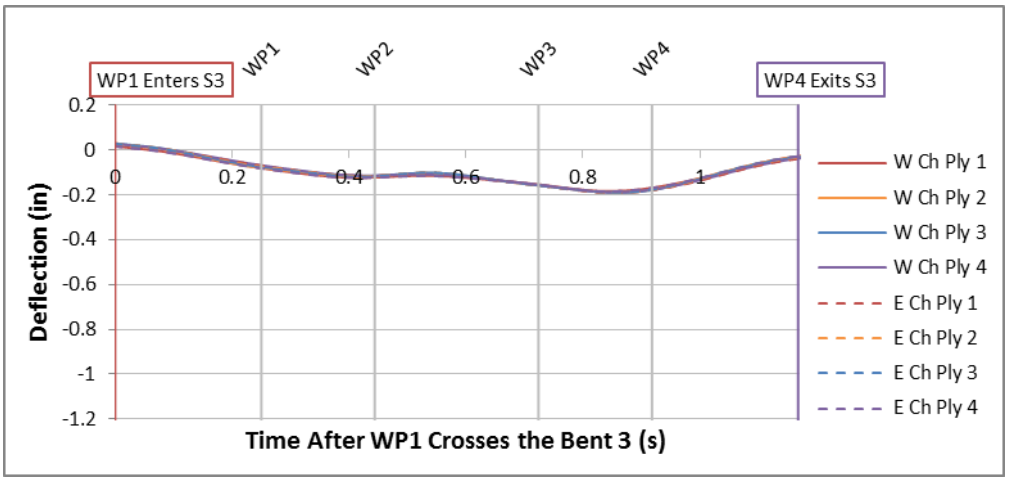

Figure A-117: IC1 20 MPH Net Midspan Stringer Deflections

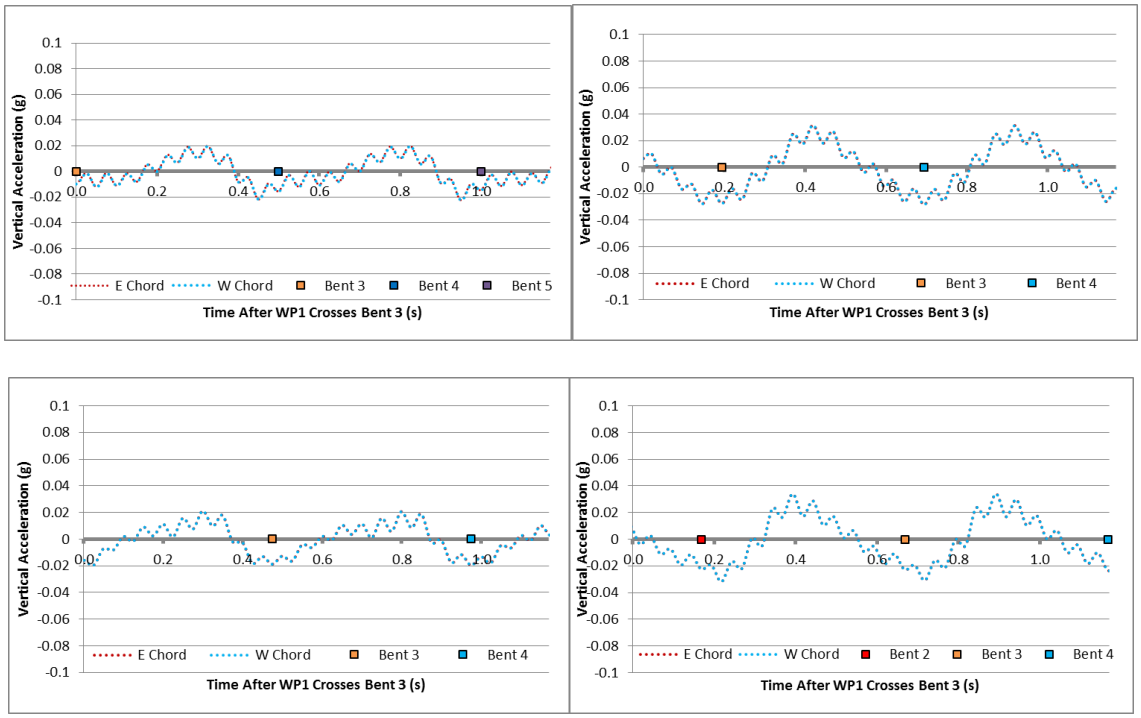

Figure A-118: IC1 20 MPH Wheel Path Accelerations: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

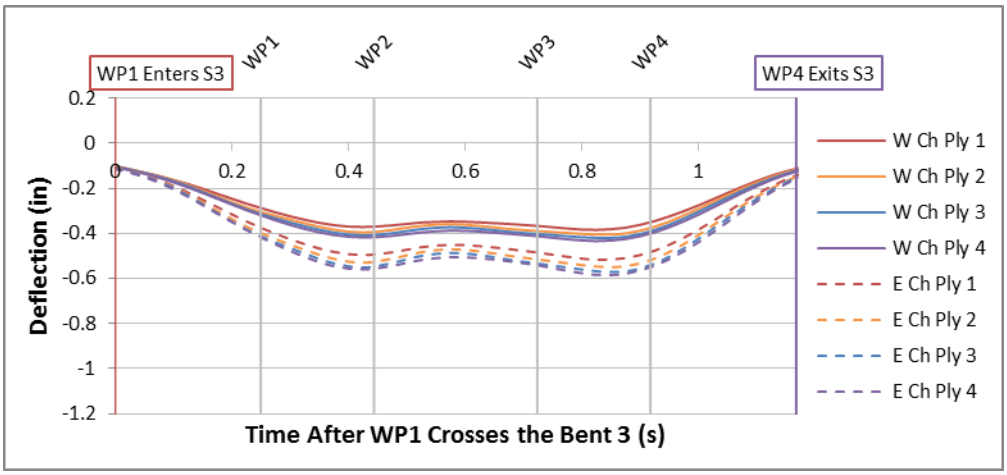

Figure A-119: IC2 20 MPH Total Midspan Stringer Deflections

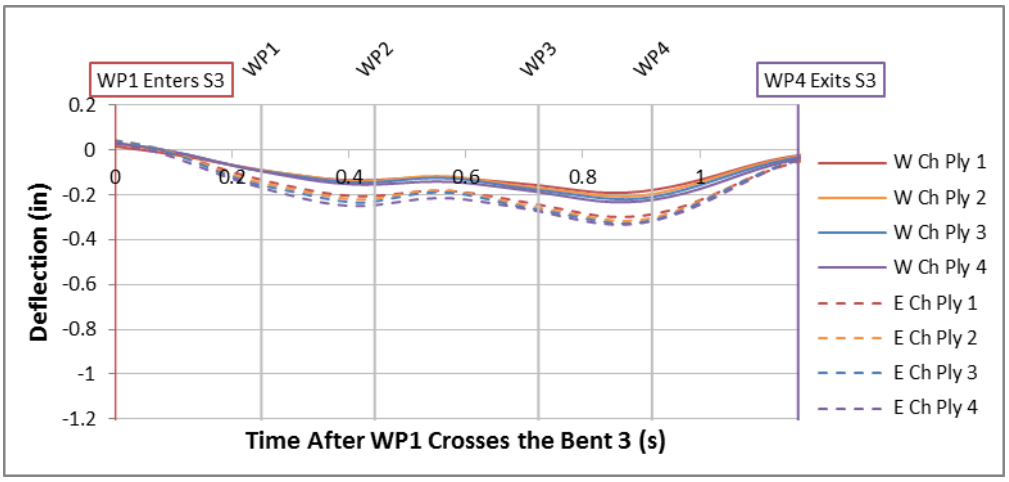

Figure A-120: IC2 20 MPH Net Midspan Stringer Deflections

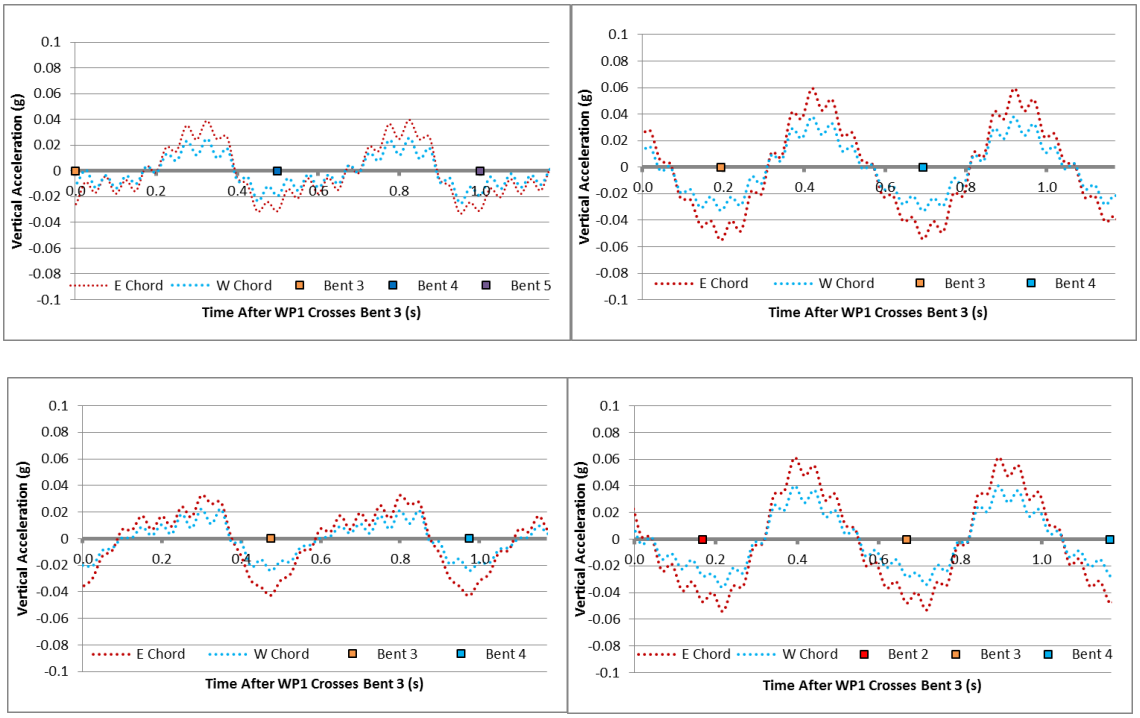

Figure A-121: IC2 20 MPH Wheel Path Accelerations: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

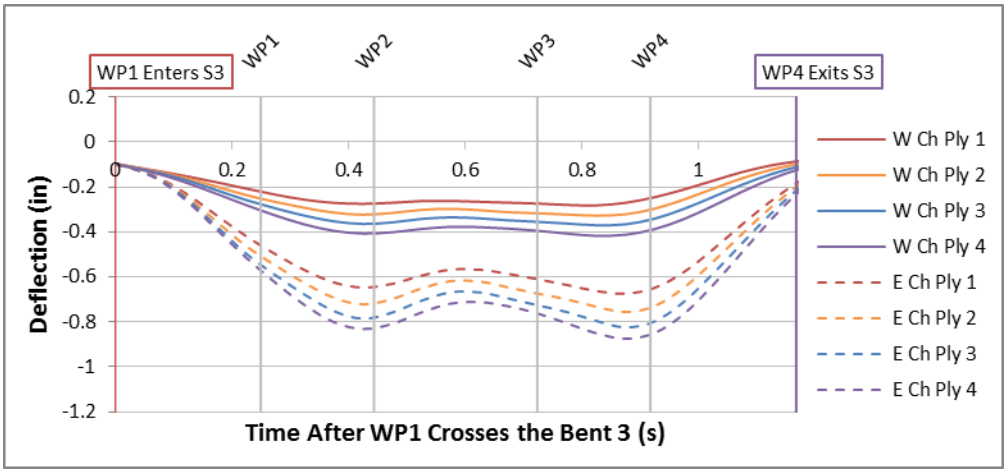

Figure A-122: IC3 20 MPH Total Midspan Stringer Deflections

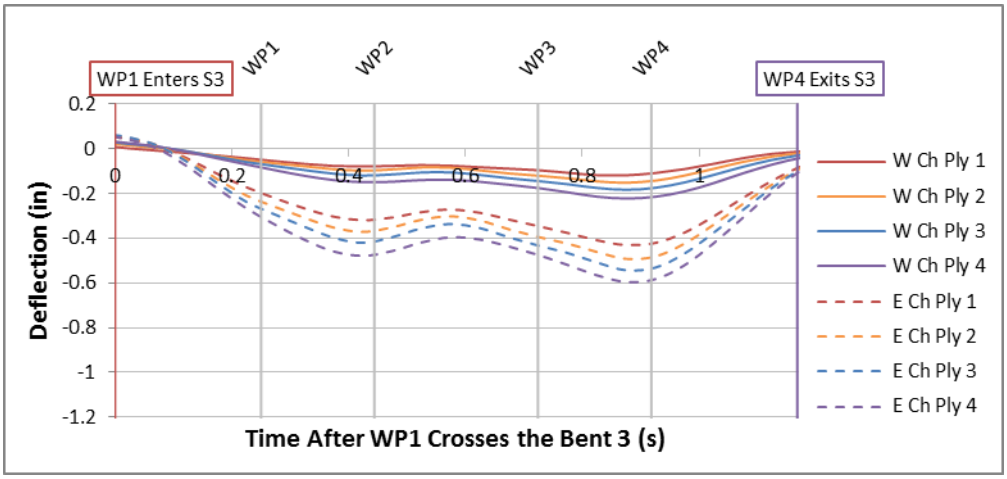

Figure A-123: IC3 20 MPH Net Midspan Stringer Deflections

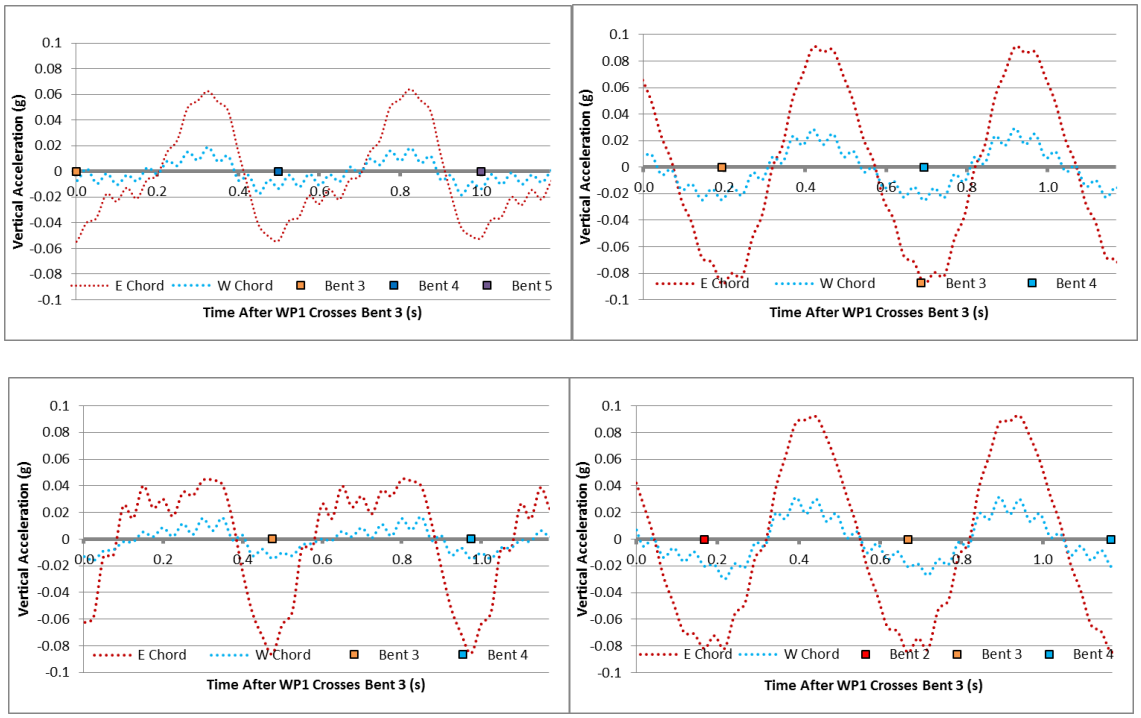

Figure A-124: IC 3 20 MPH Wheel Path Accelerations: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

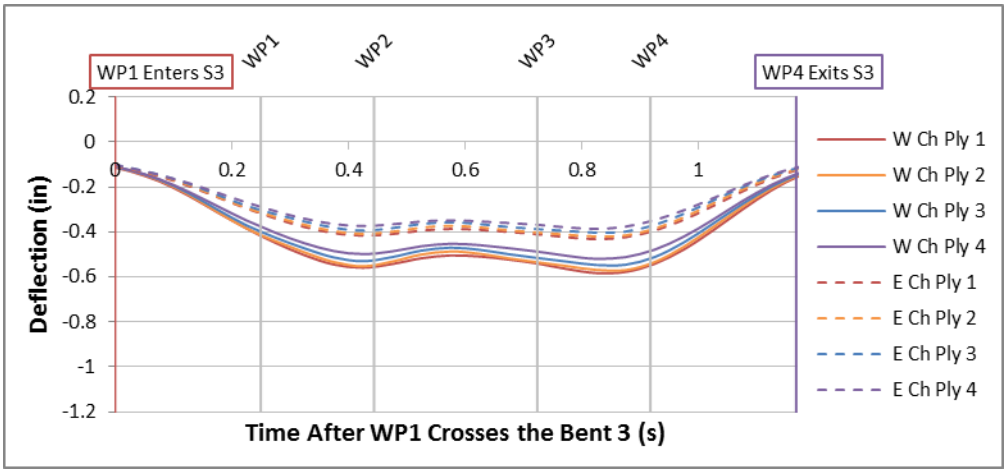

Figure A-125: IC4 20 MPH Total Midspan Stringer Deflections

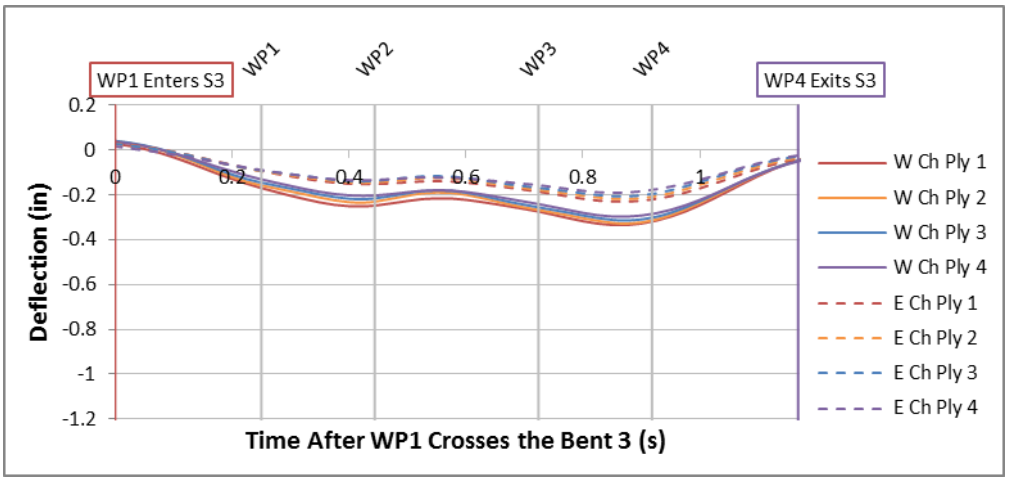

Figure A-126: IC4 20 MPH Net Midspan Stringer Deflections

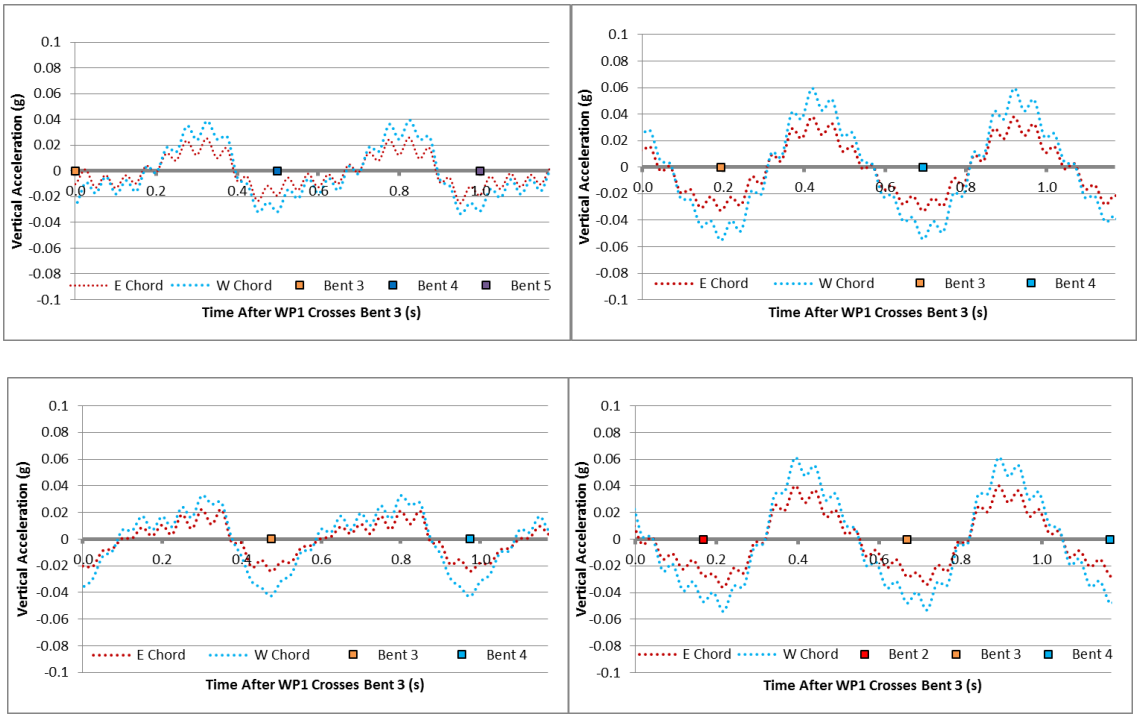

Figure A-127: IC4 20 MPH Wheel Path Accelerations: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

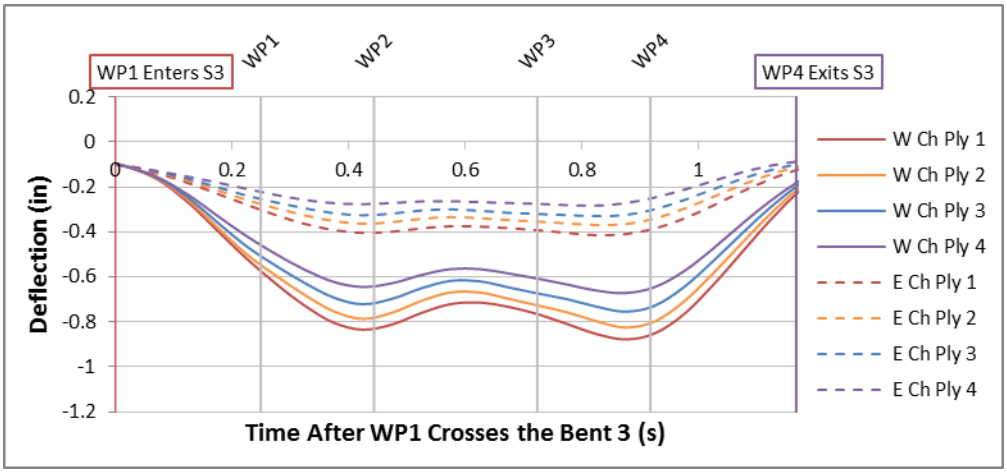

Figure A-128: IC5 20 MPH Total Midspan Stringer Deflections

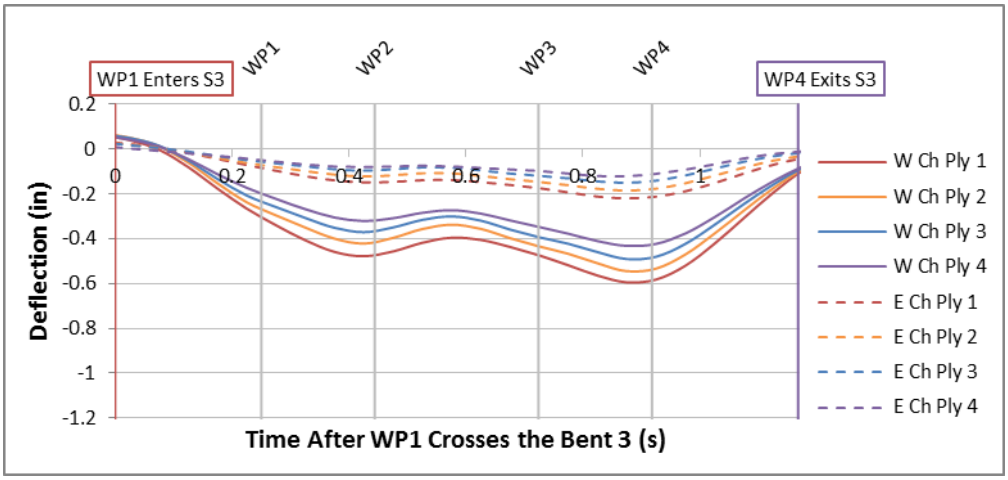

Figure A-129: IC5 20 MPH Net Midspan Stringer Deflections

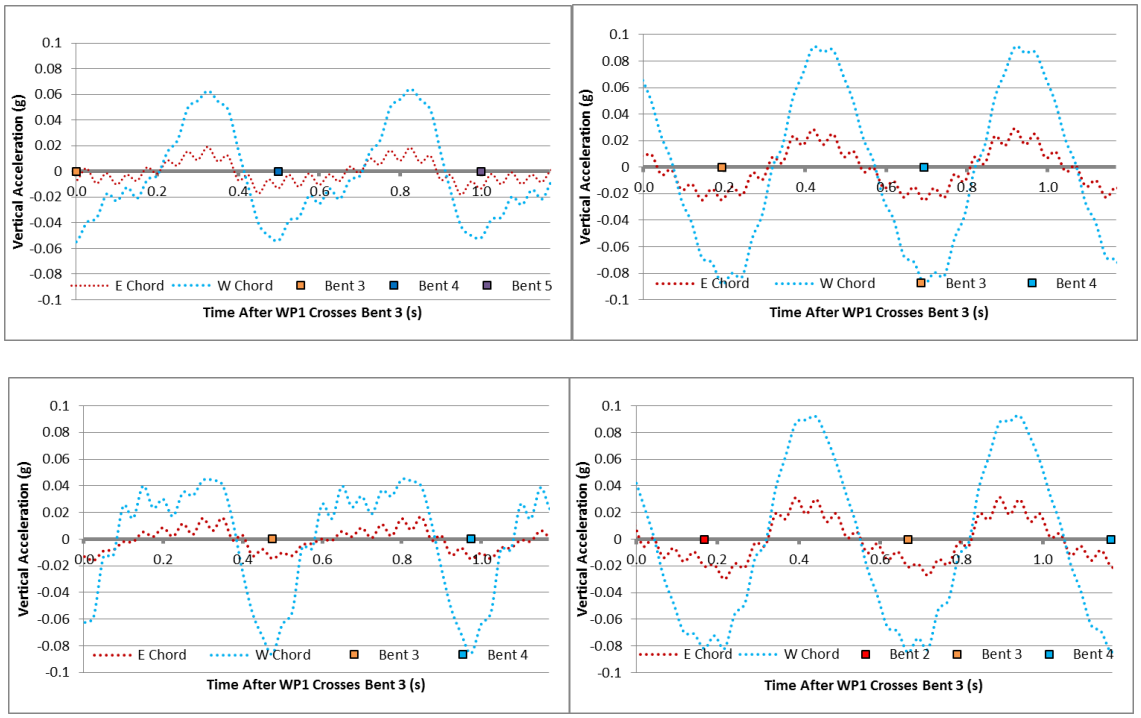

Figure A-130: IC5 20 MPH Wheel Path Accelerations: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

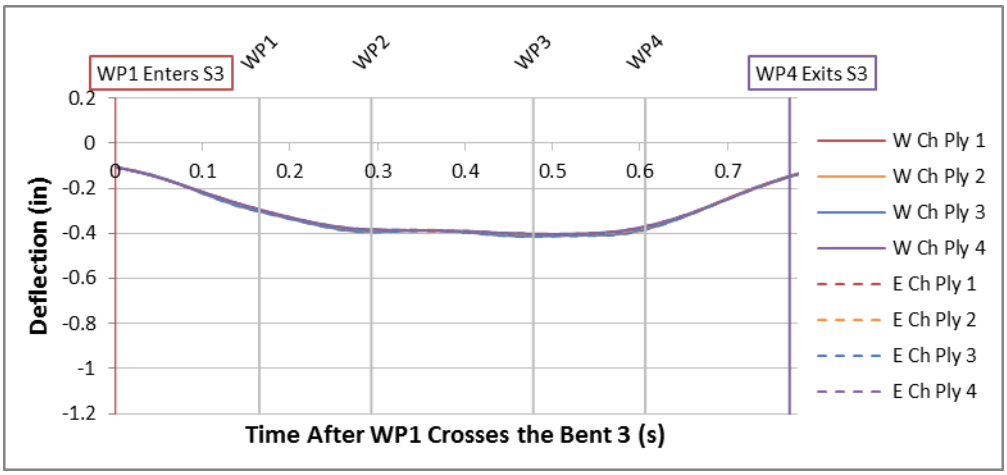

Figure A-131: IC1 30 MPH Total Midspan Stringer Deflections

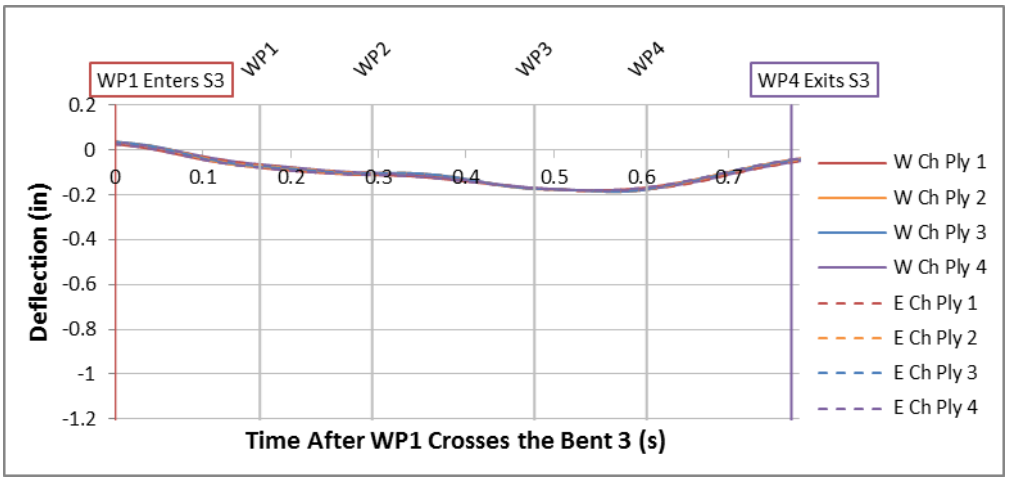

Figure A-132: IC1 30 MPH Net Midspan Stringer Deflections

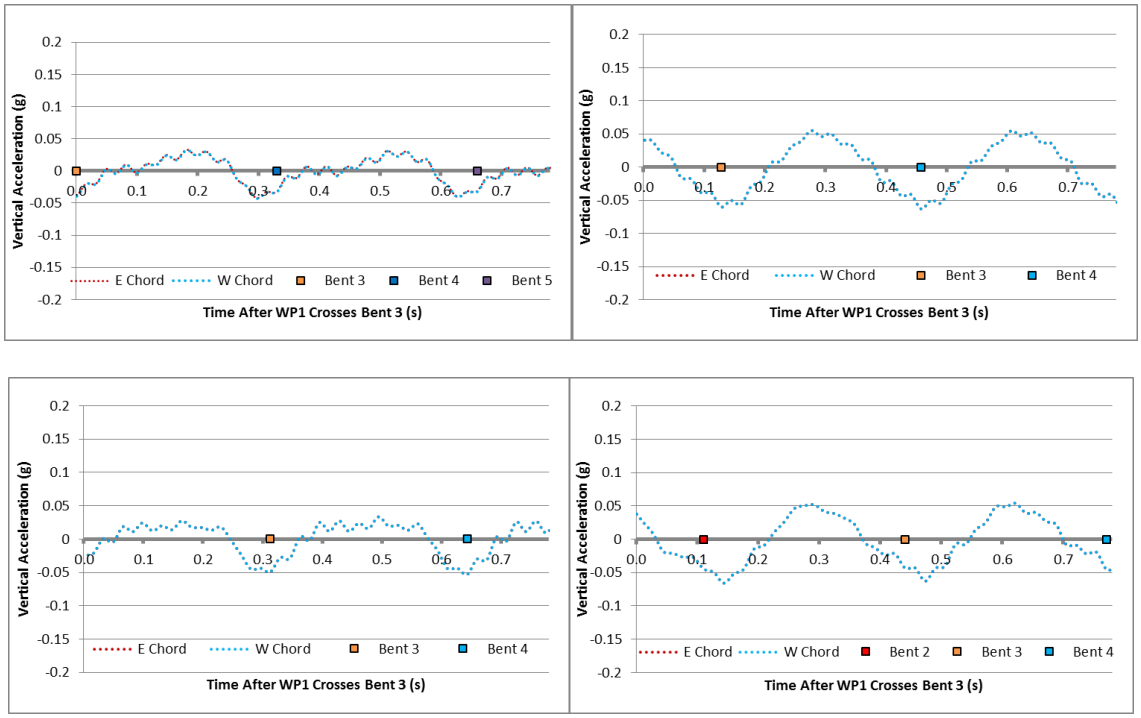

Figure A-133: IC1 30 MPH Wheel Path Accelerations: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

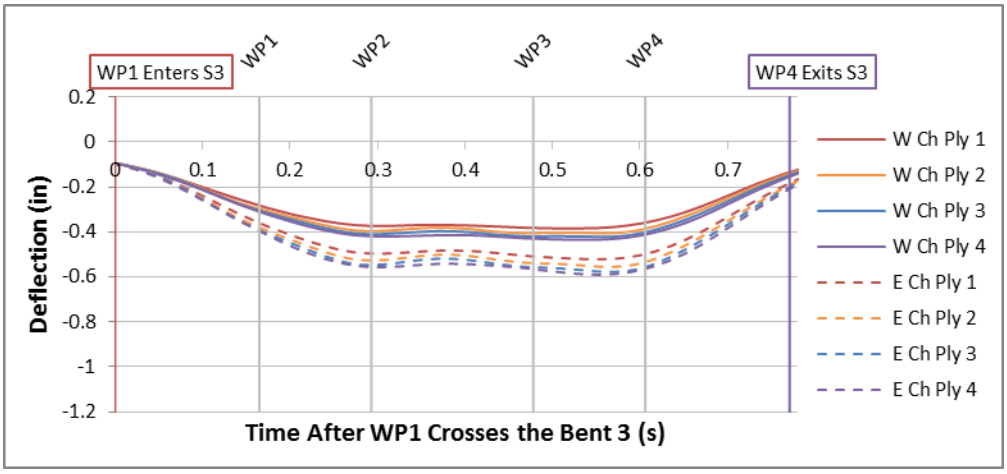

Figure A-134: IC2 30 MPH Total Midspan Stringer Deflections

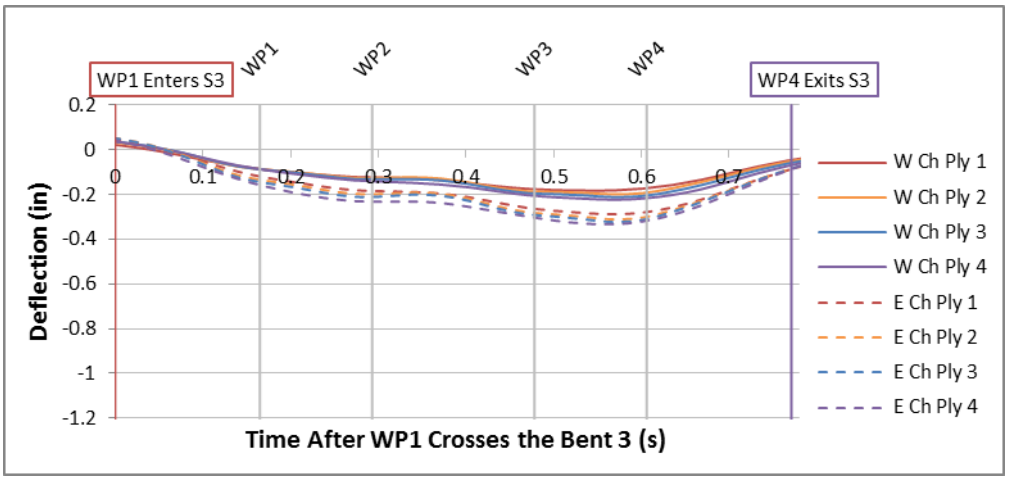

Figure A-135: IC2 30 MPH Net Midspan Stringer Deflections

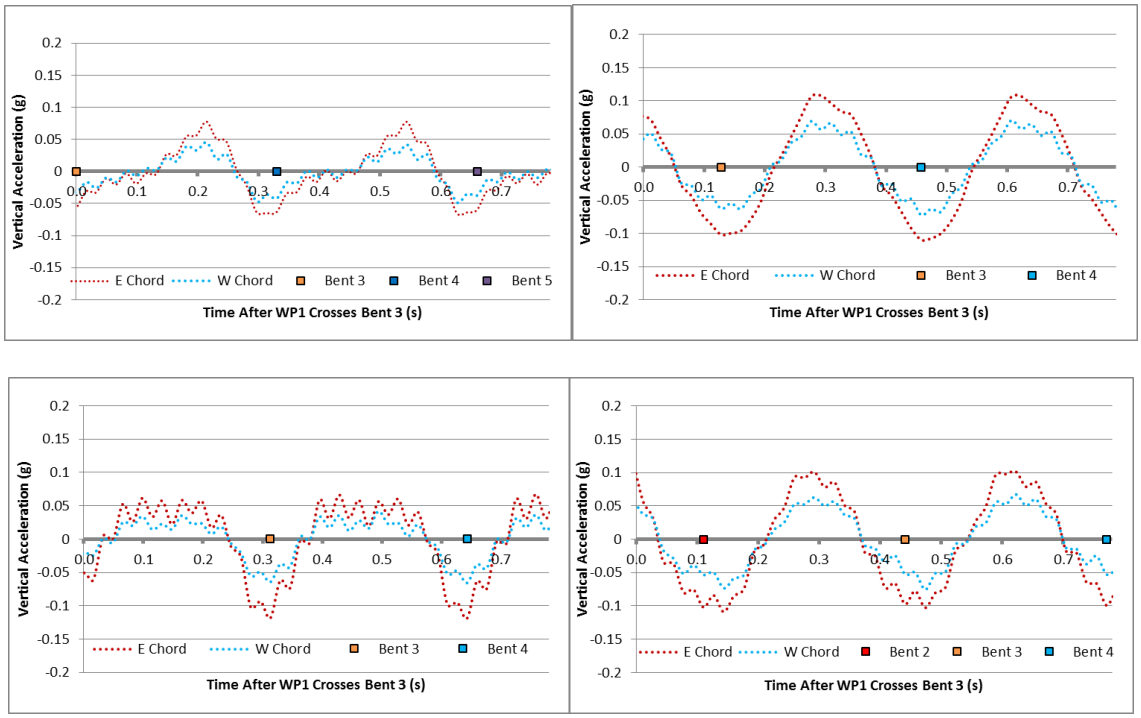

Figure A-136: IC2 30 MPH Wheel Path Accelerations: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

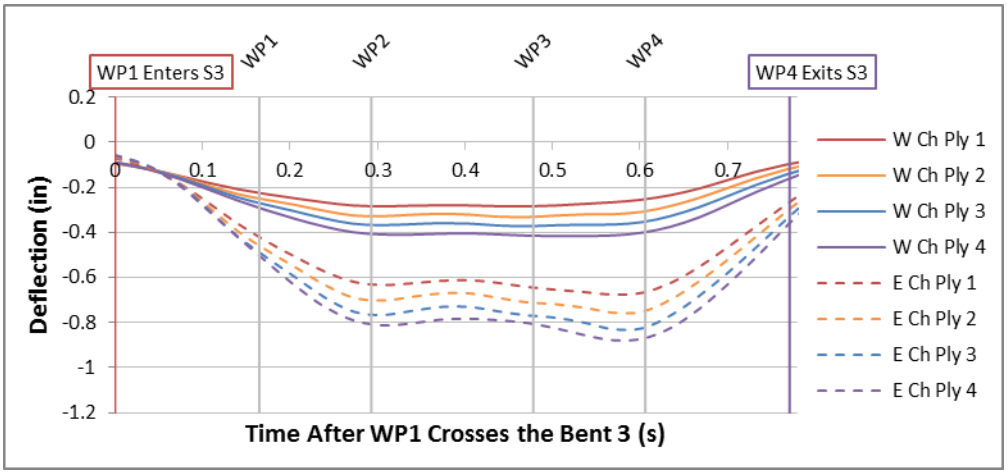

Figure A-137: IC3 30 MPH Total Midspan Stringer Deflections

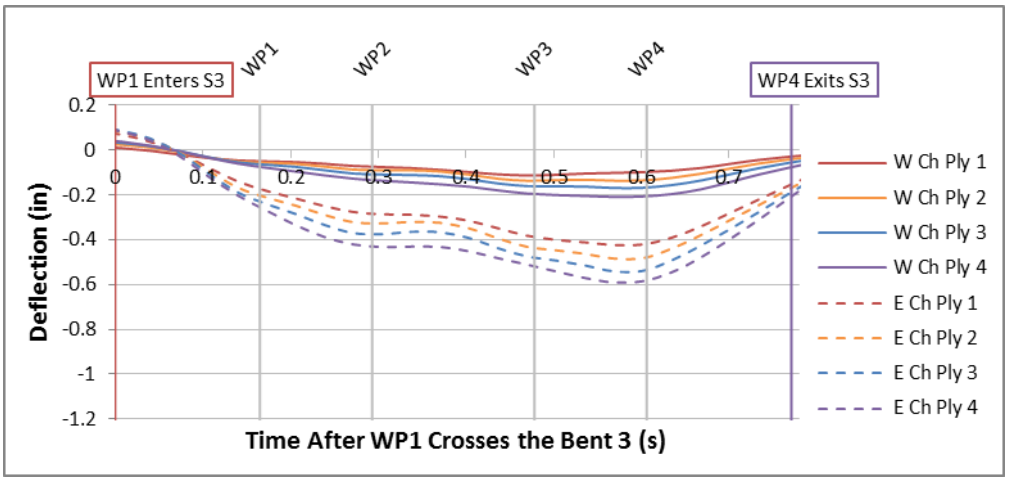

Figure A-138: IC3 30 MPH Net Midspan Stringer Deflections

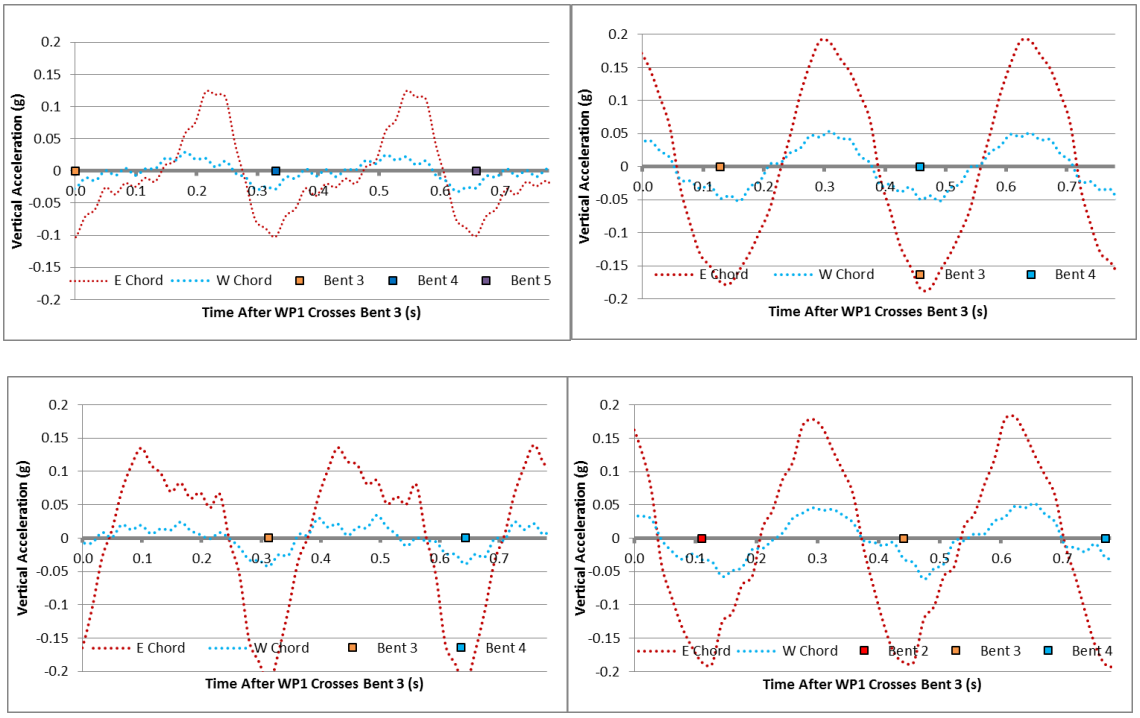

Figure A-139: IC3 30 MPH Wheel Path Accelerations: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

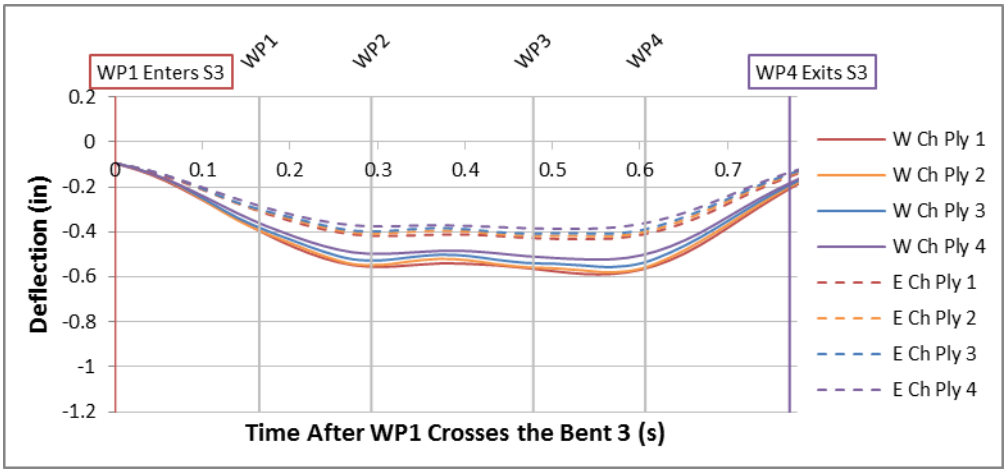

Figure A-140: IC4 30 MPH Total Midspan Stringer Deflections

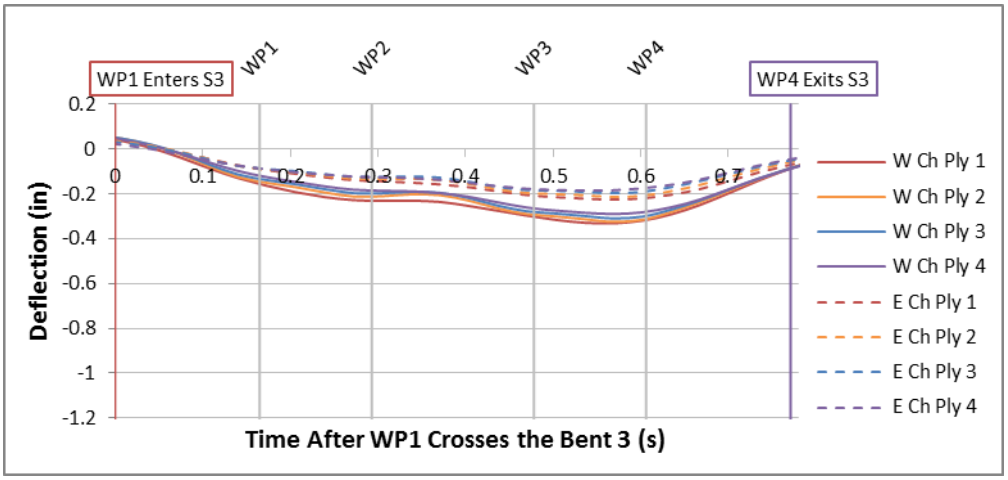

Figure A-141: IC4 30 MPH Net Midspan Stringer Deflections

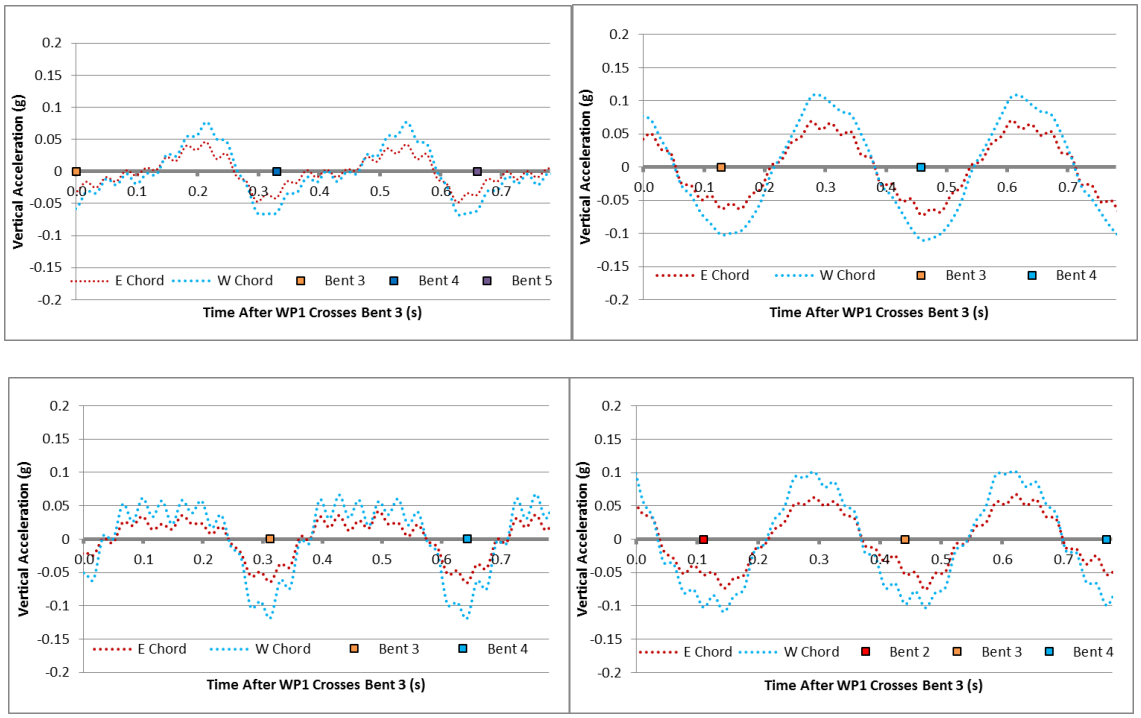

Figure A-142: IC4 30 MPH Wheel Path Accelerations: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

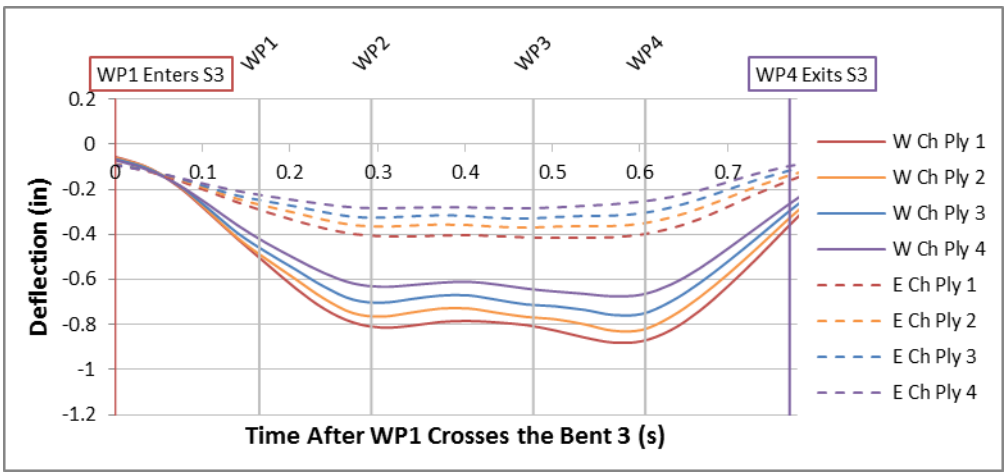

Figure A-143: IC5 30 MPH Total Midspan Stringer Deflections

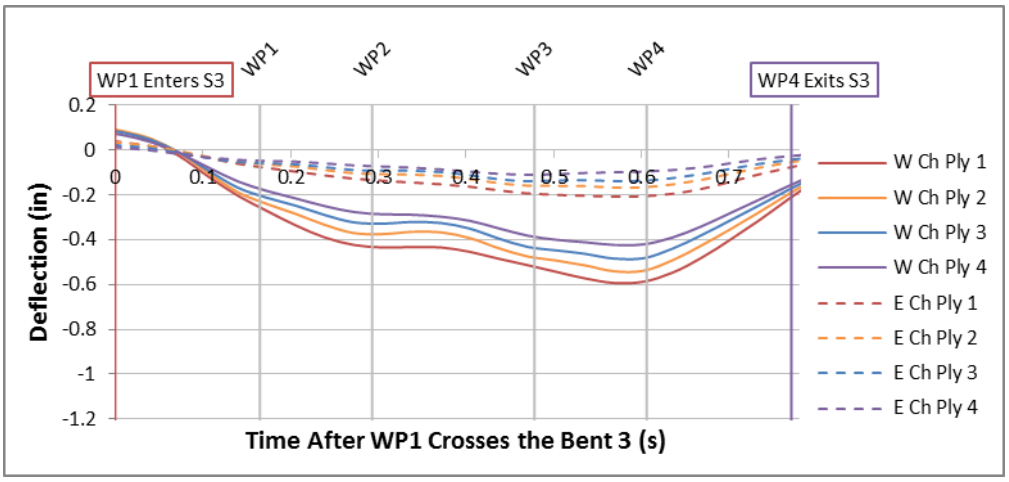

Figure A-144: IC5 30 MPH Net Midspan Stringer Deflections

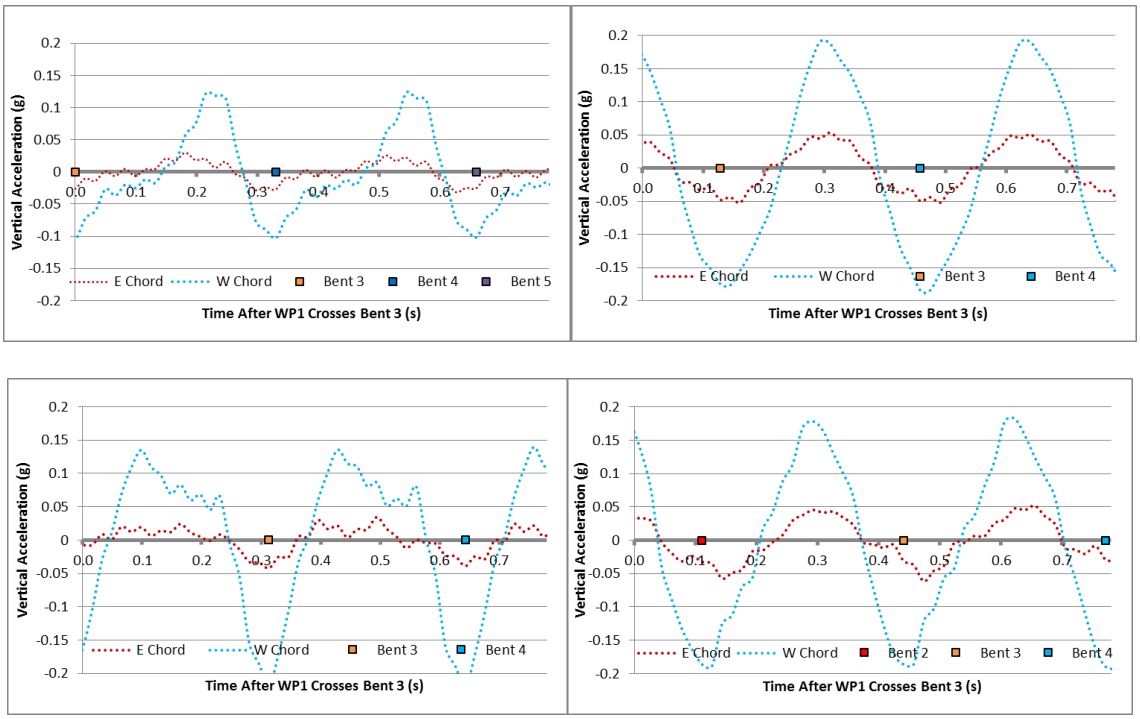

Figure A-145: IC5 30 MPH Wheel Path Accelerations: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

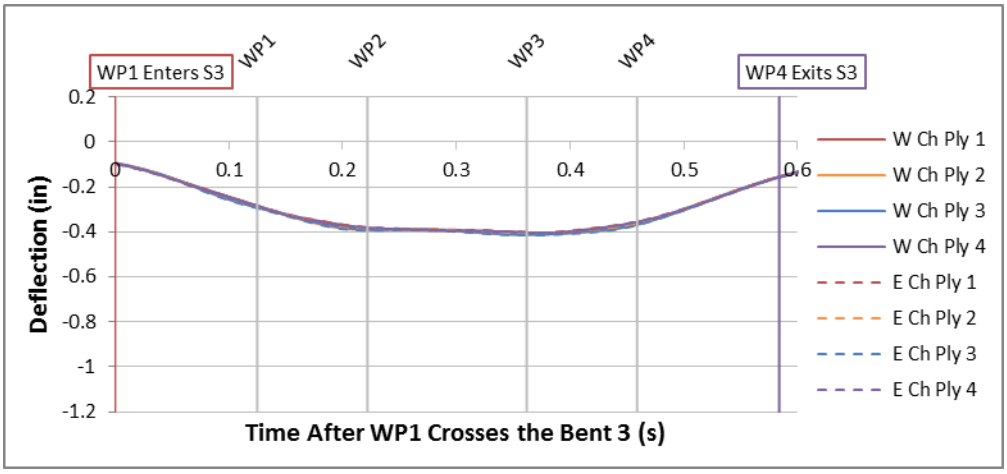

Figure A-146: IC1 40 MPH Total Midspan Stringer Deflections

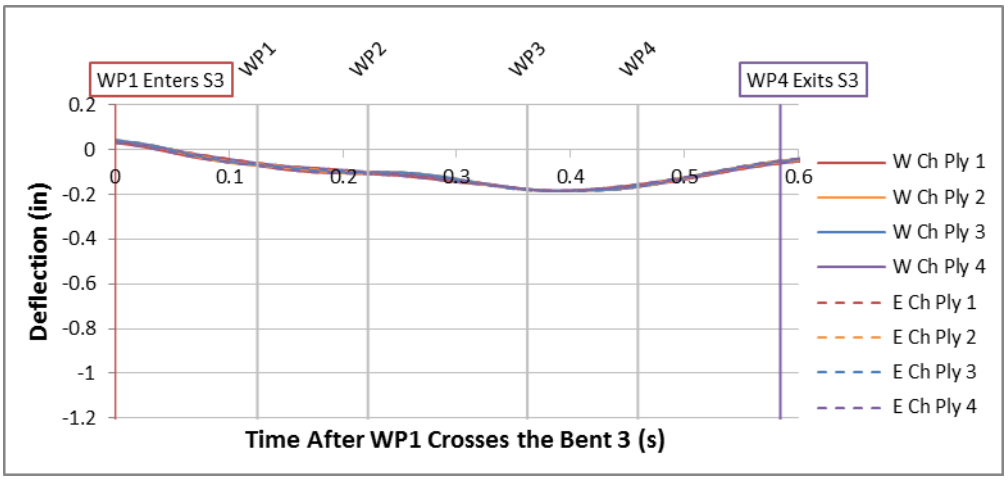

Figure A-147: IC1 40 MPH Net Midspan Stringer Deflections

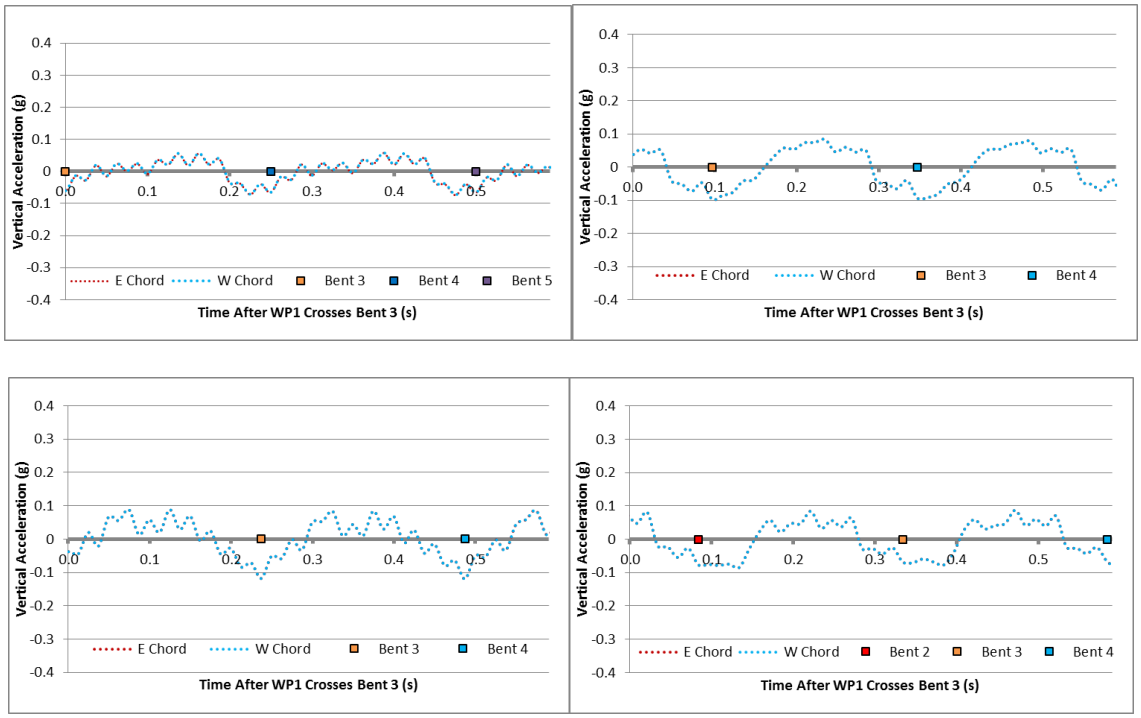

Figure A-148: IC1 40 MPH Wheel Path Accelerations: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

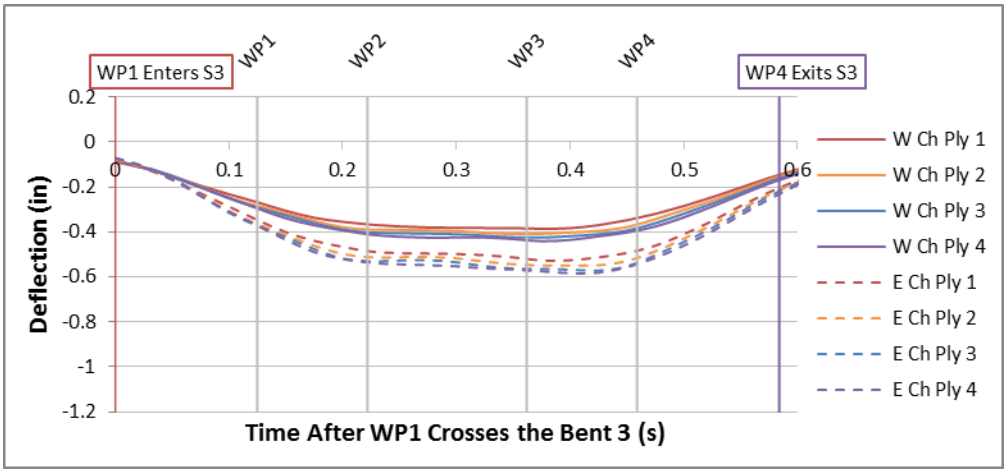

Figure A-149: IC2 40 MPH Total Midspan Stringer Deflections

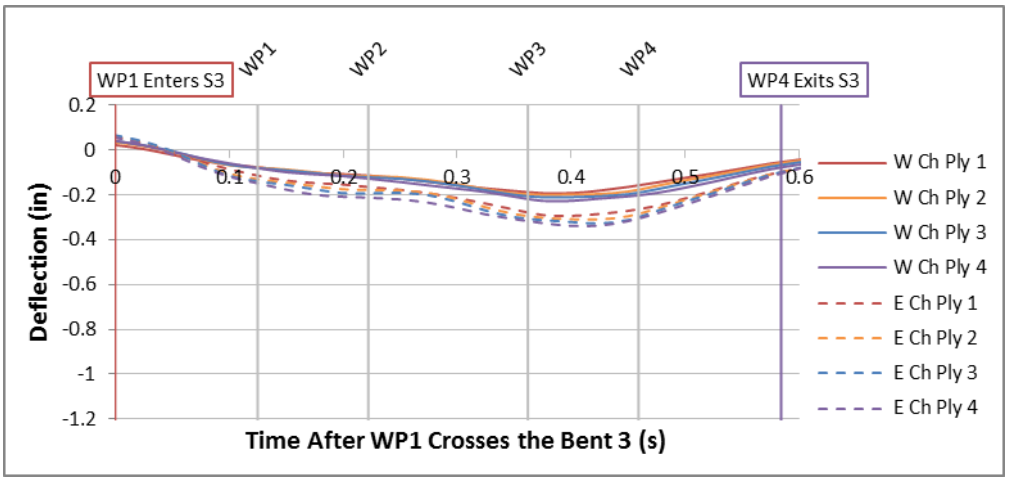

Figure A-150: IC2 40 MPH Net Midspan Stringer Deflections

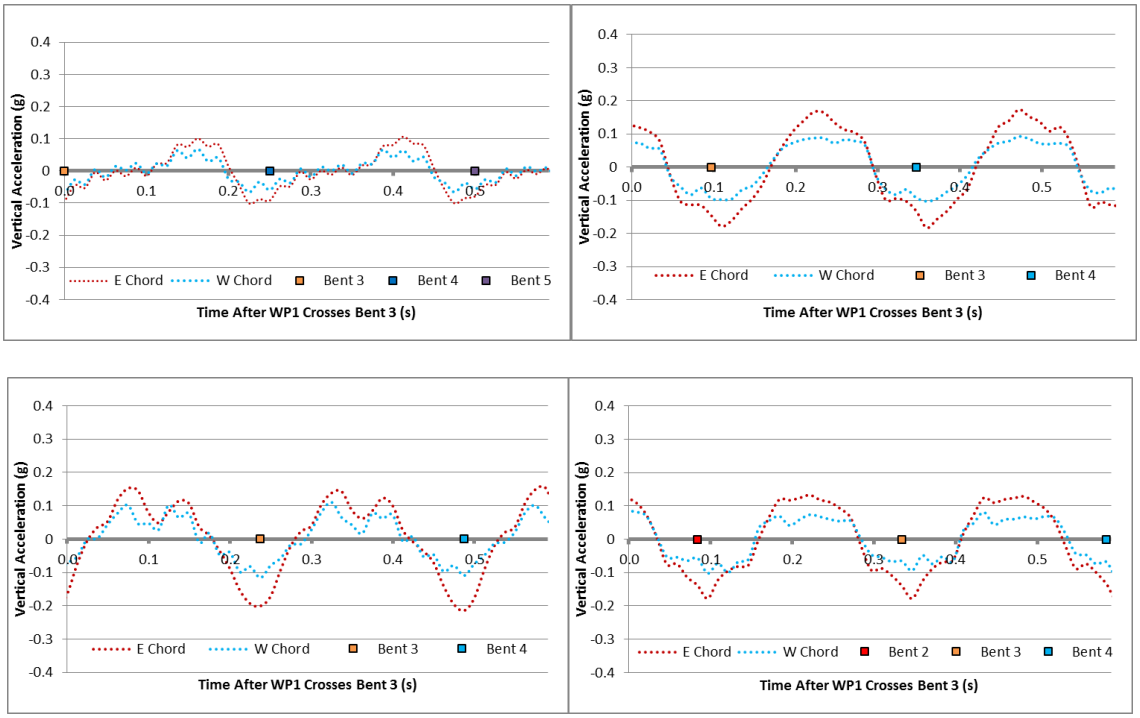

Figure A-151:IC2 40 MPH Wheel Path Accelerations: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

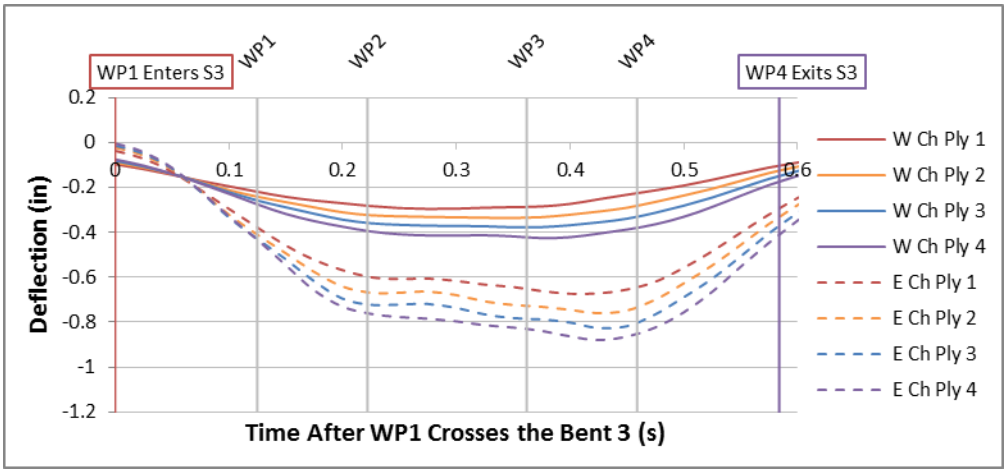

Figure A-152:IC3 40 MPH Total Midspan Stringer Deflections

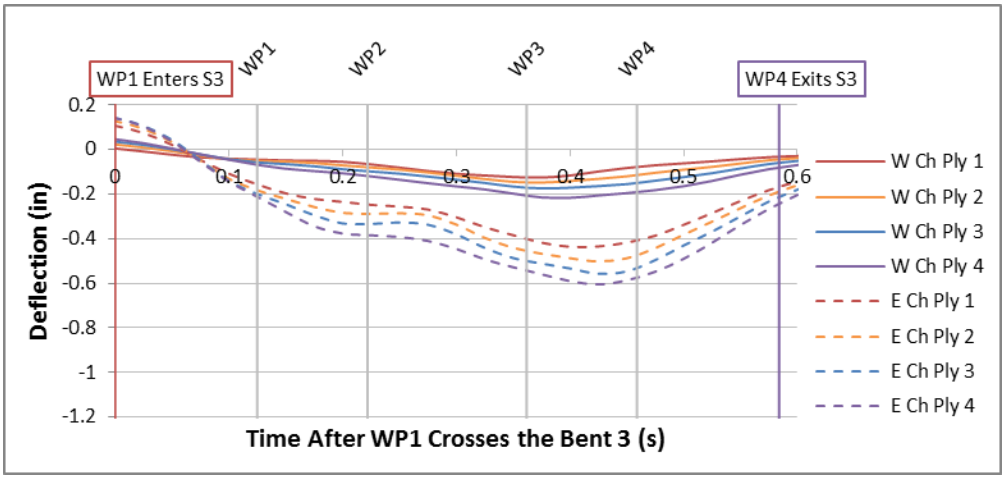

Figure A-153: IC3 40 MPH Net Midspan Stringer Deflections

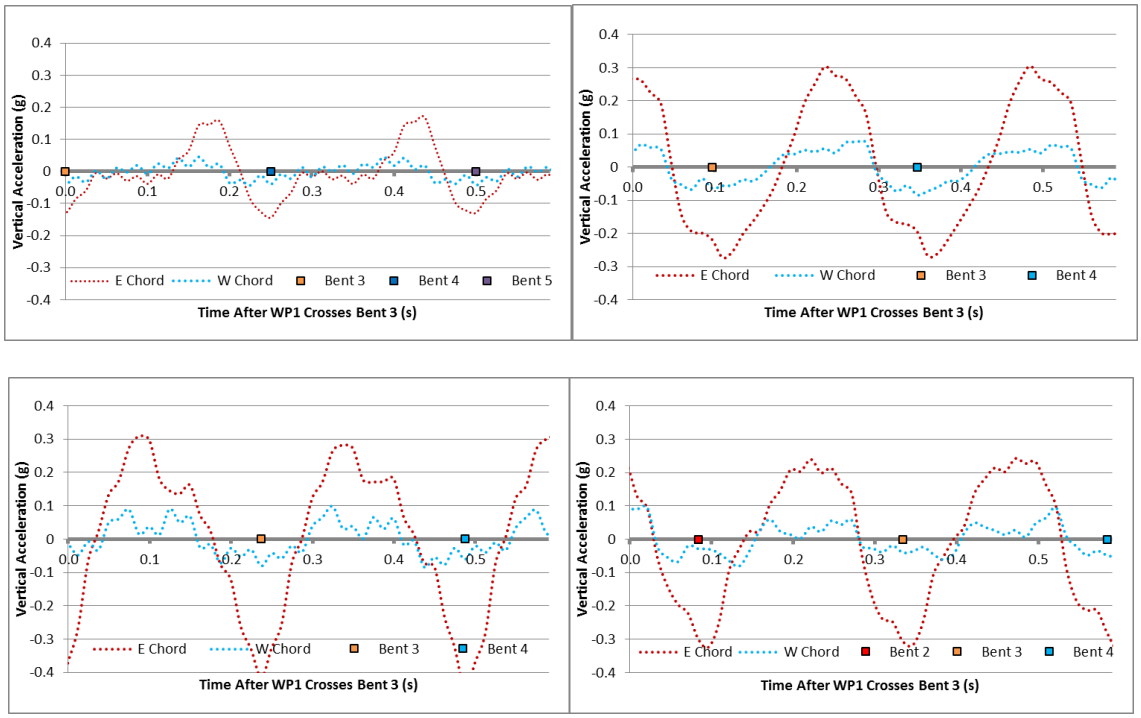

Figure A-154: IC3 40 MPH Wheel Path Accelerations: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

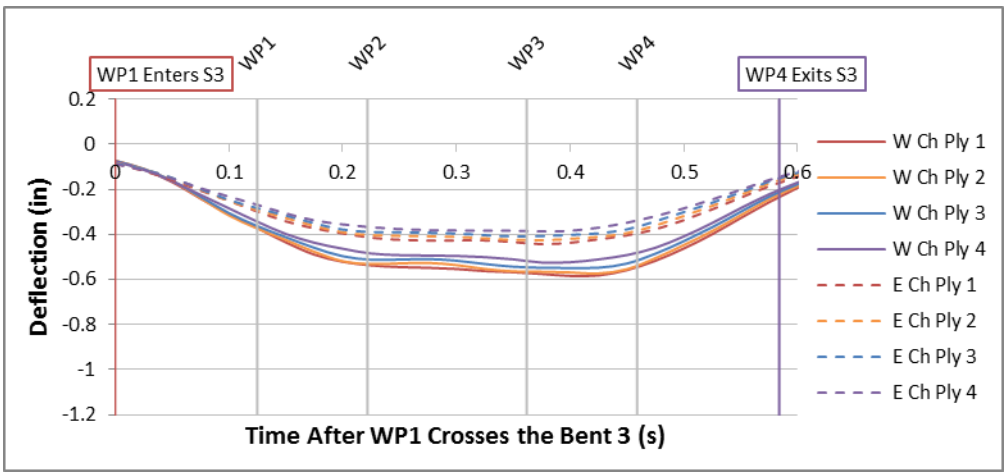

Figure A-155: IC4 40 MPH Total Midspan Stringer Deflections

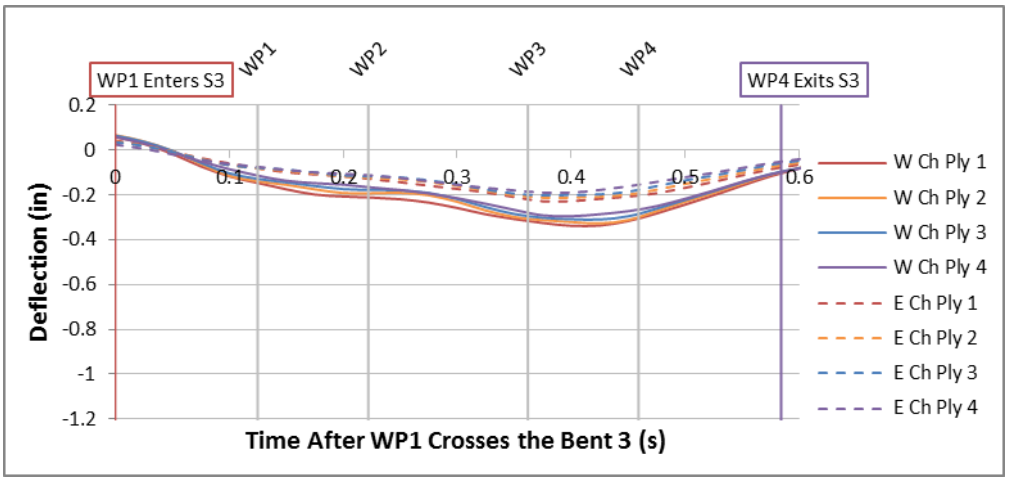

Figure A-156: IC4 40 MPH Net Midspan Stringer Deflections

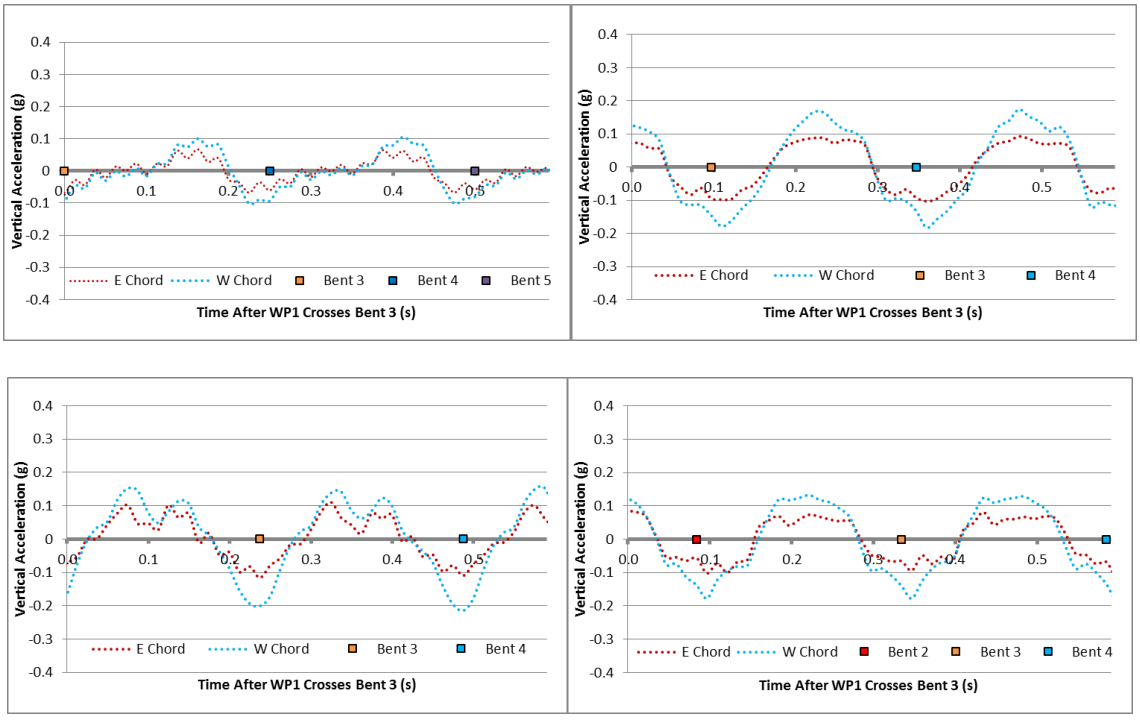

Figure A-157: IC4 40 MPH Wheel Path Accelerations: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

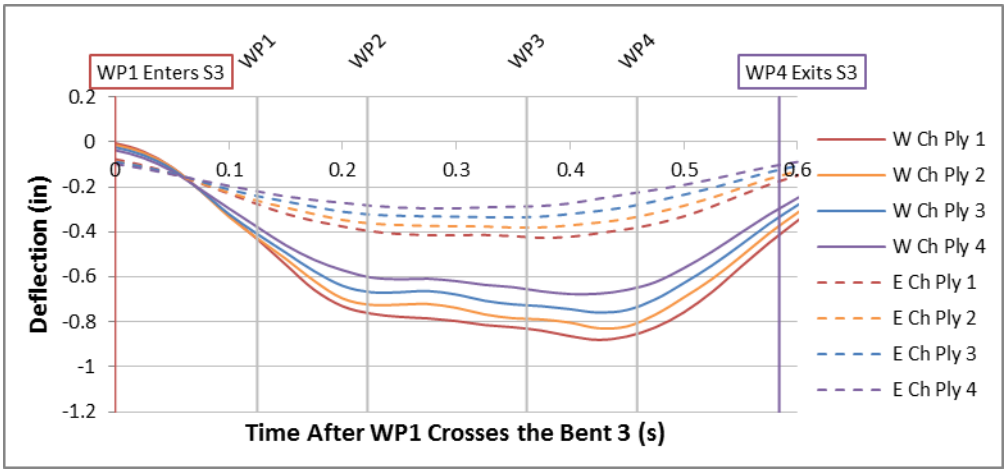

Figure A-158: IC5 40 MPH Total Midspan Stringer Deflections

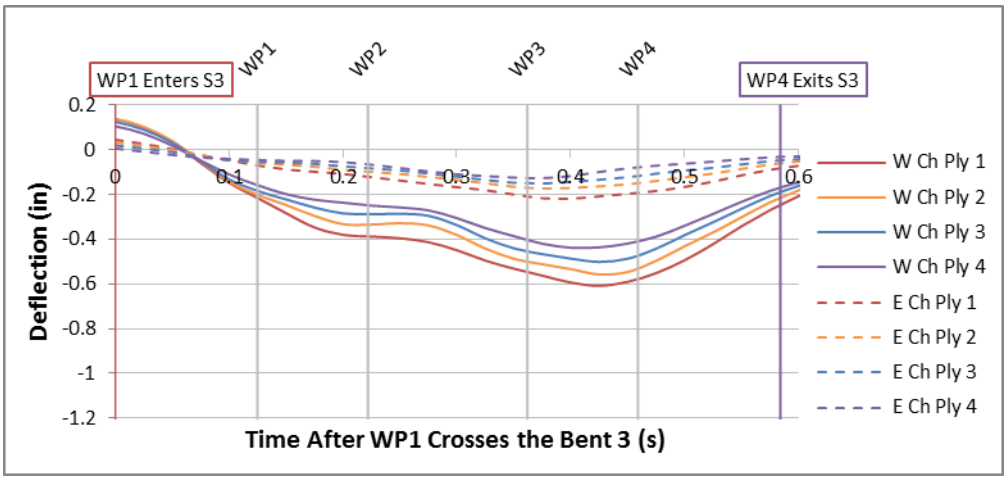

Figure A-159: IC5 40 MPH Net Midspan Stringer Deflections

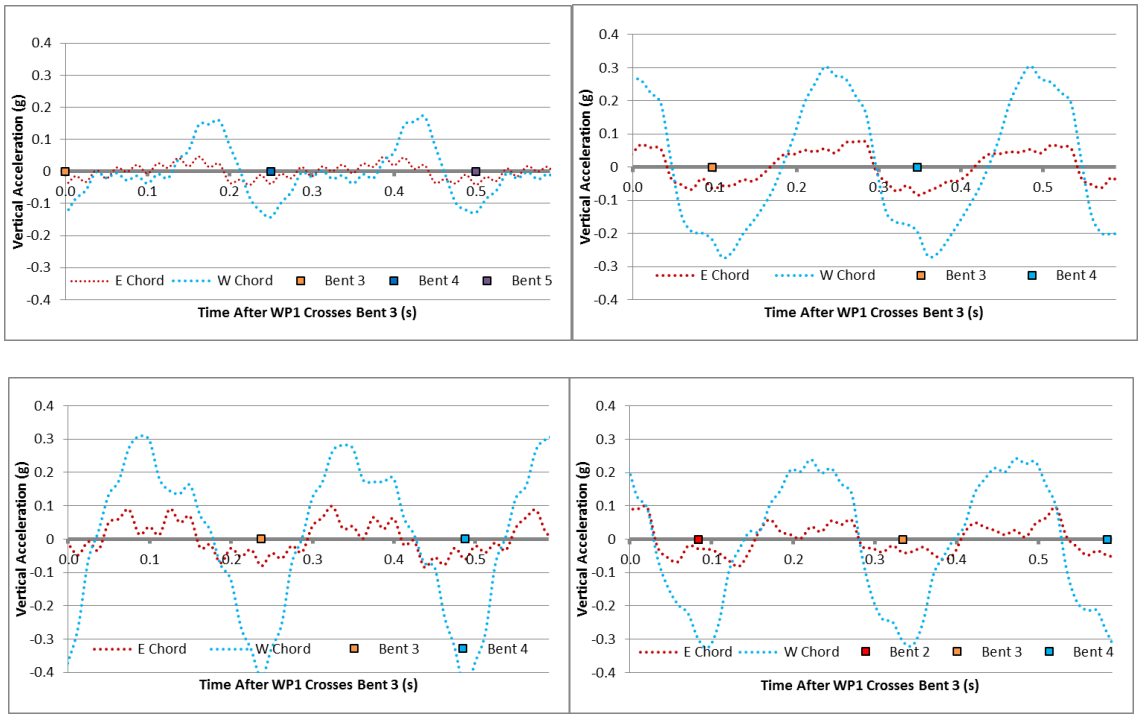

Figure A-160: IC5 40 MPH Wheel Path Accelerations: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

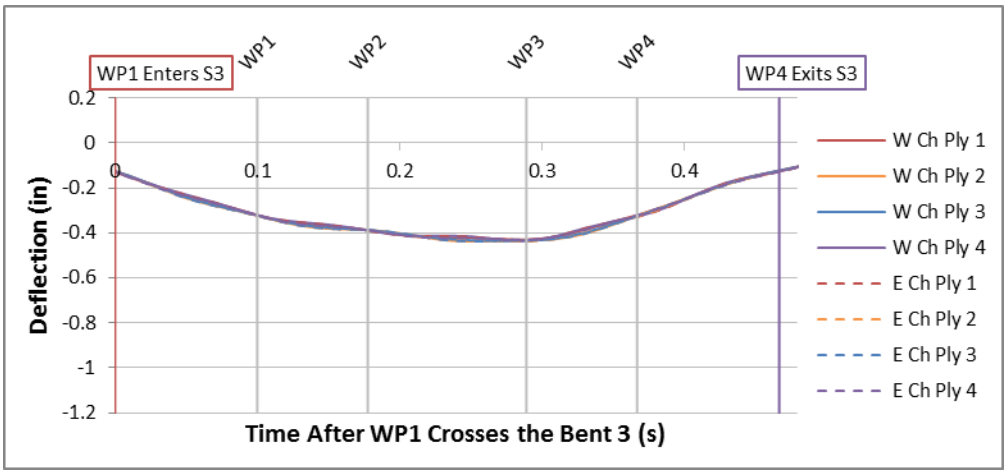

Figure A-161: IC1 50 MPH Total Midspan Stringer Deflections

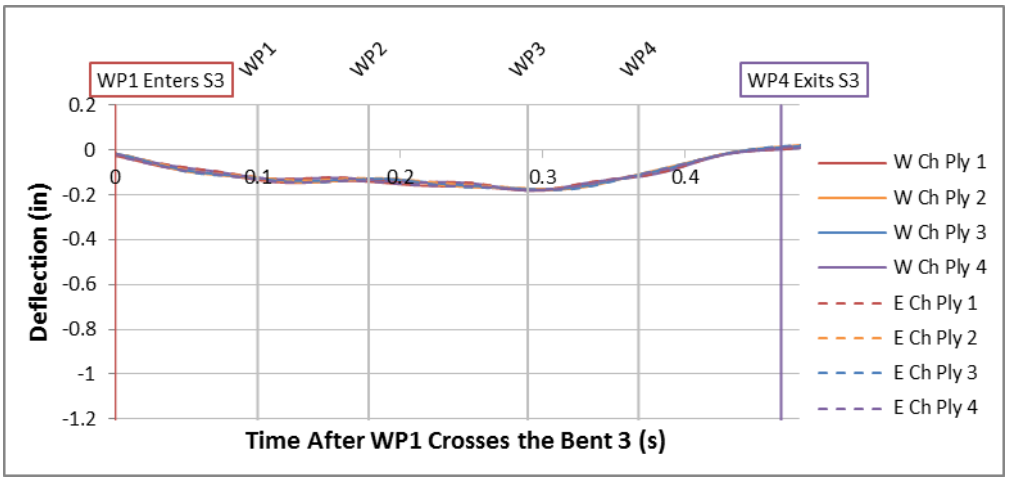

Figure A-162: IC1 50 MPH Net Midspan Stringer Deflections

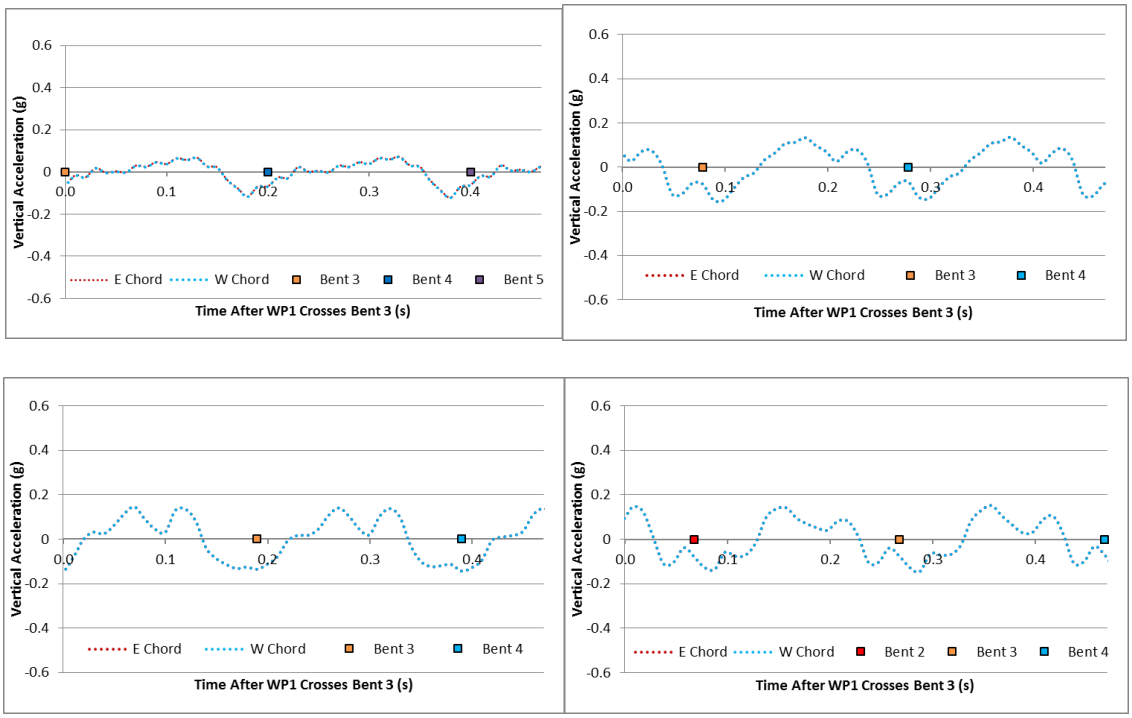

Figure A-163: IC1 50 MPH Wheel Path Accelerations: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

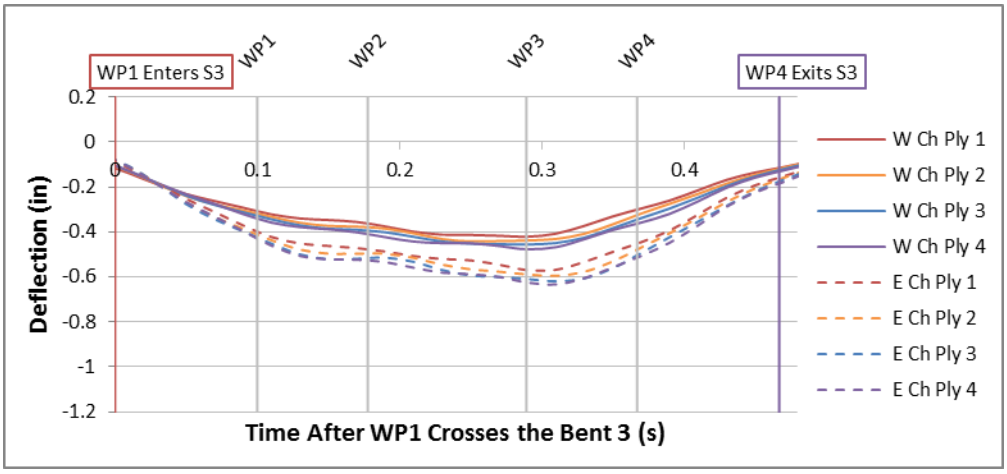

Figure A-164: IC2 50 MPH Total Midspan Stringer Deflections

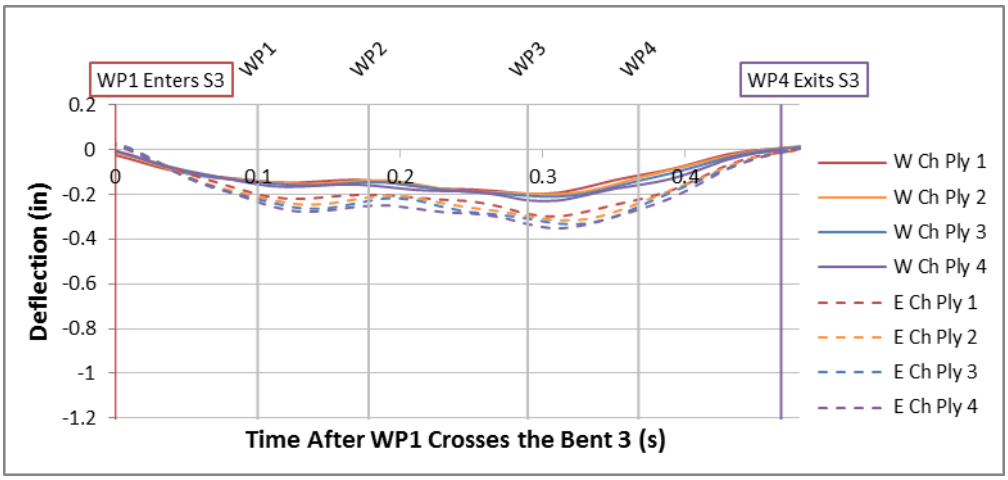

Figure A-165: IC2 50 MPH Net Midspan Stringer Deflections

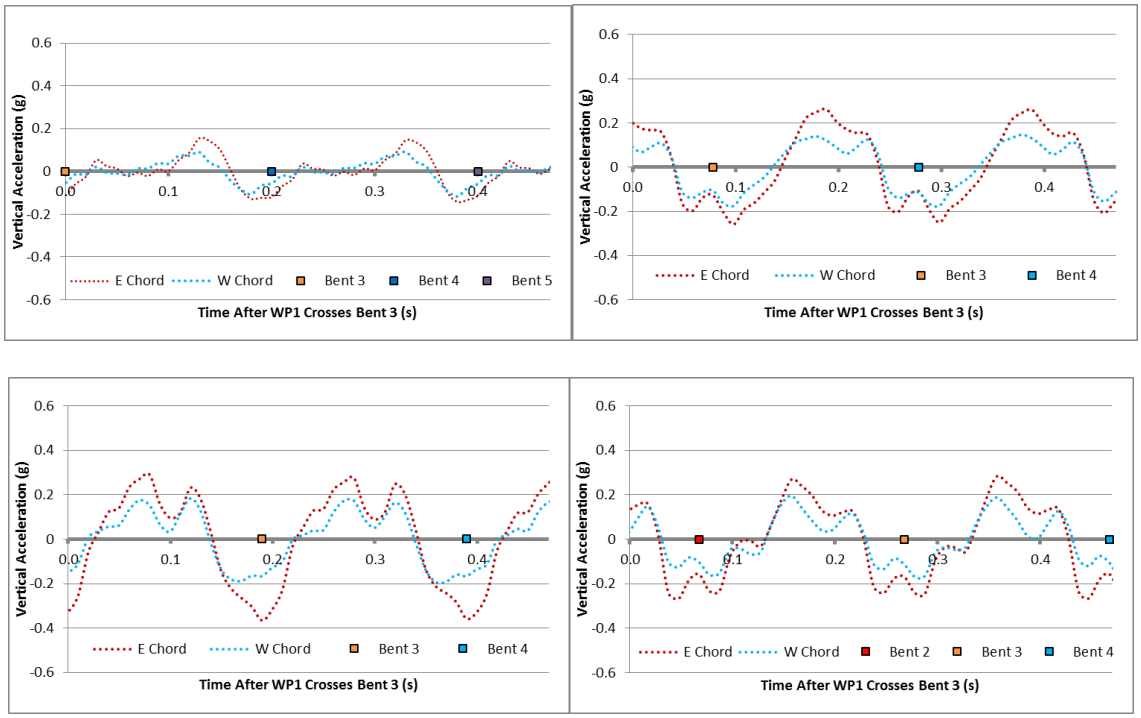

Figure A-166: IC2 50 MPH Wheel Path Accelerations: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

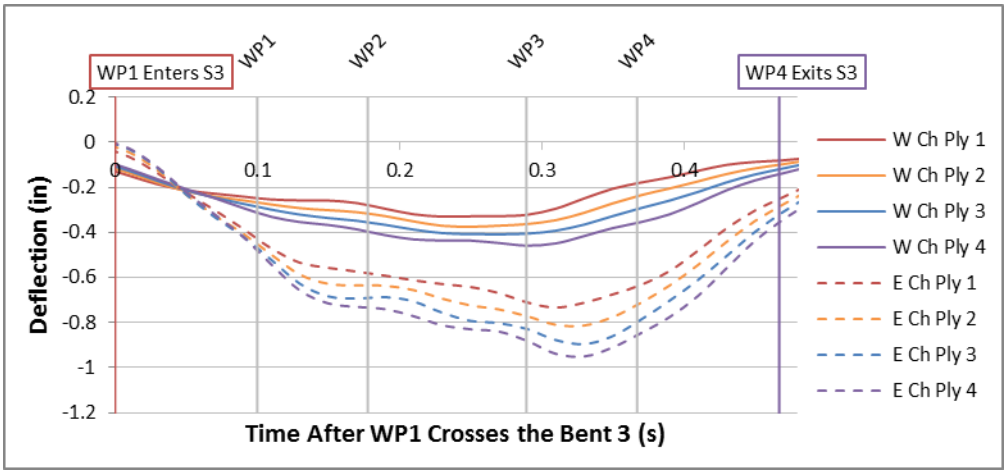

Figure A-167: IC3 50 MPH Total Midspan Stringer Deflections

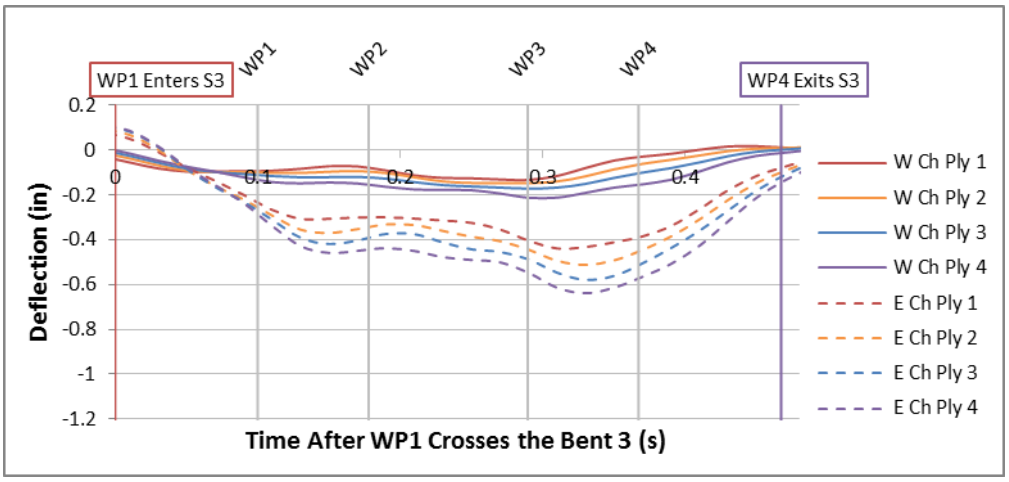

Figure A-168: IC3 50 MPH Net Midspan Stringer Deflections

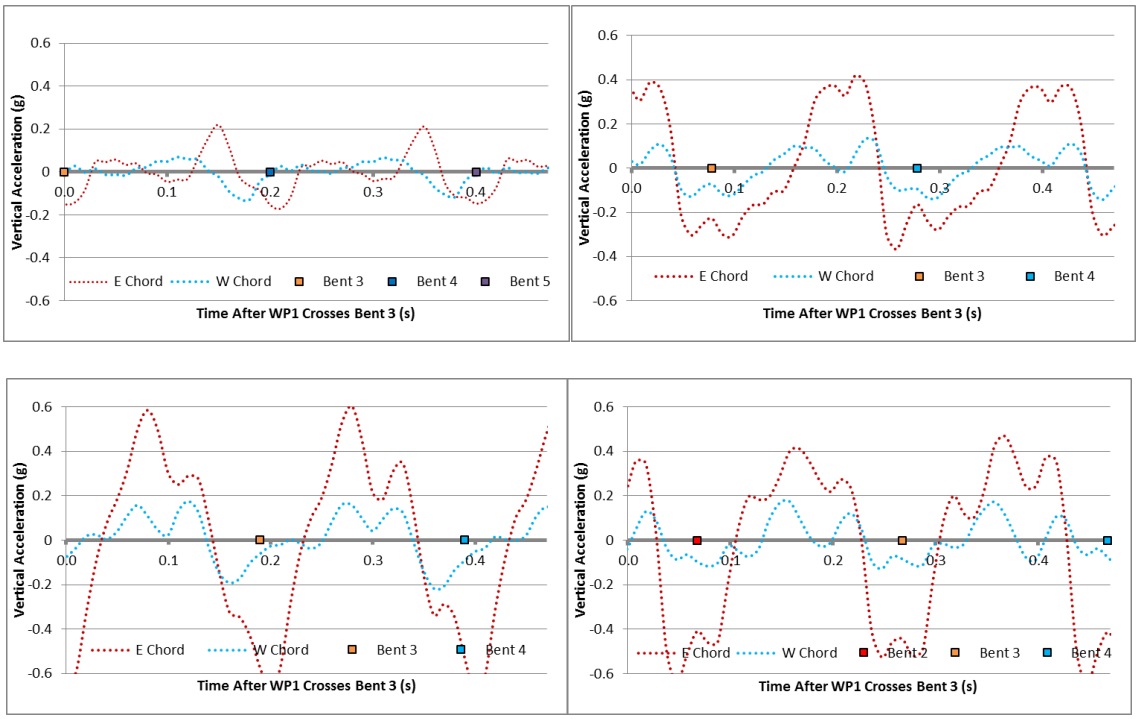

Figure A-169: IC3 50 MPH Wheel Path Accelerations: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

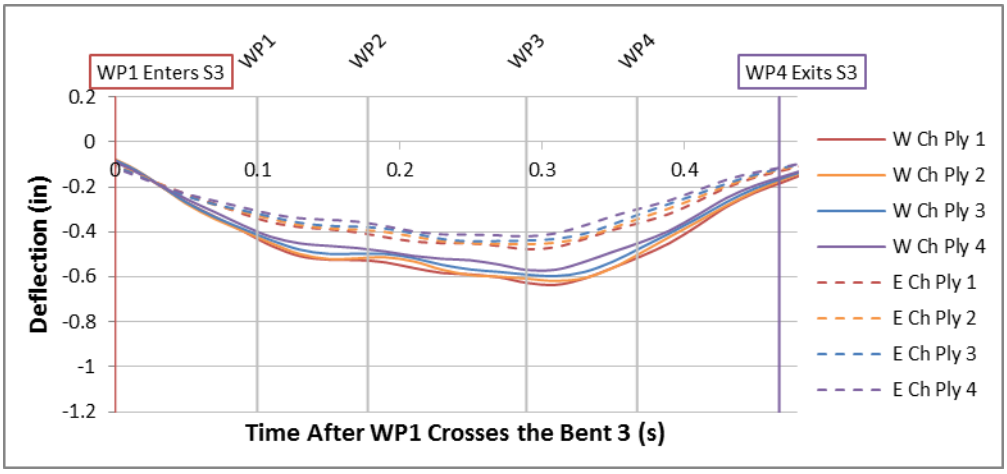

Figure A-170: IC4 50 MPH Total Midspan Stringer Deflections

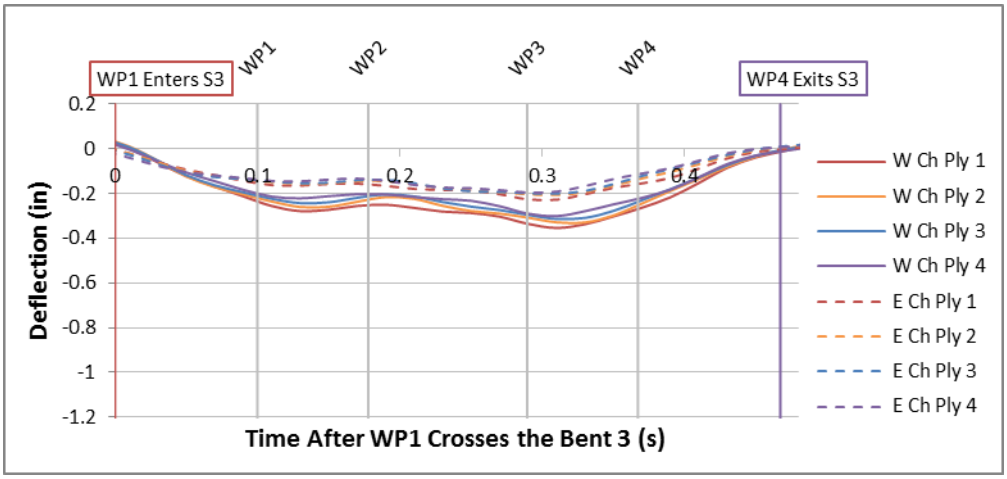

Figure A-171: IC4 50 MPH Net Midspan Stringer Deflections

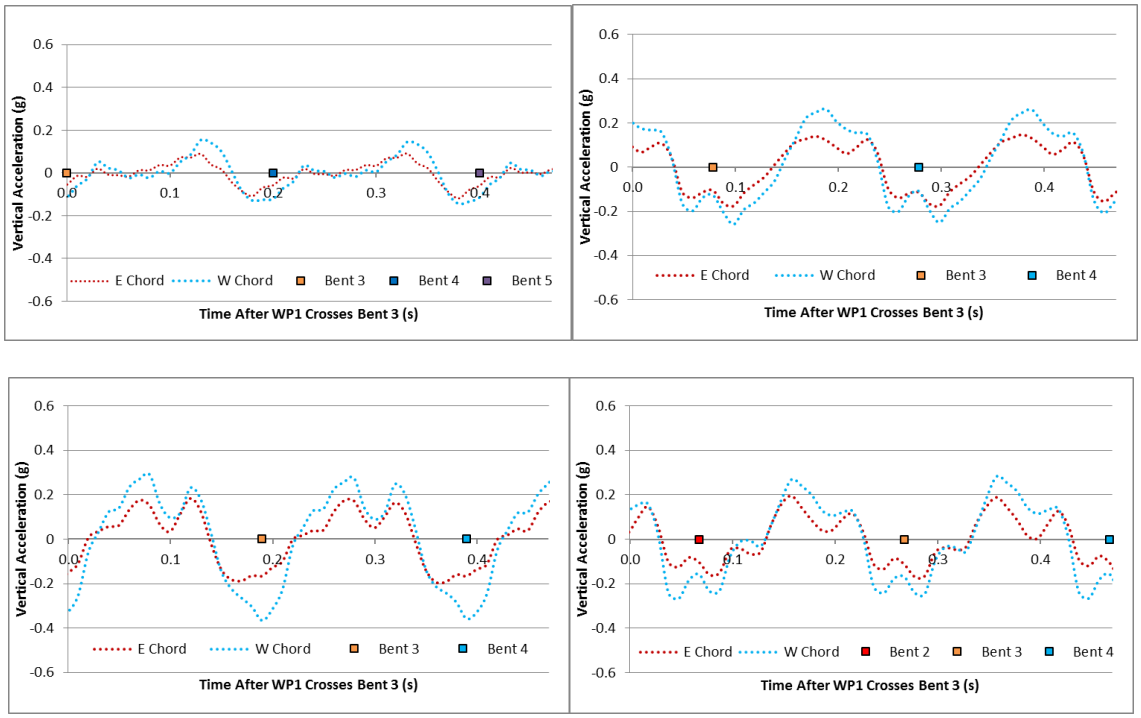

Figure A-172: IC4 50 MPH Wheel Path Accelerations: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

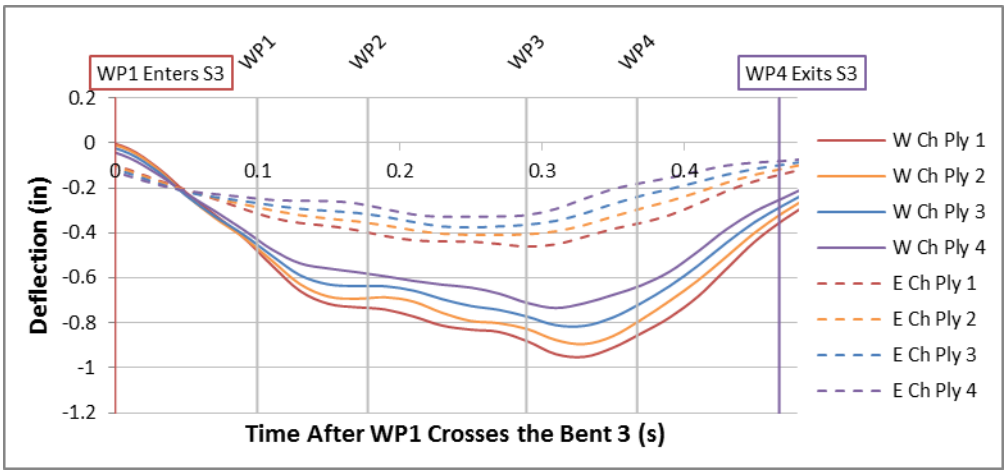

Figure A-173: IC5 50 MPH Total Midspan Stringer Deflections

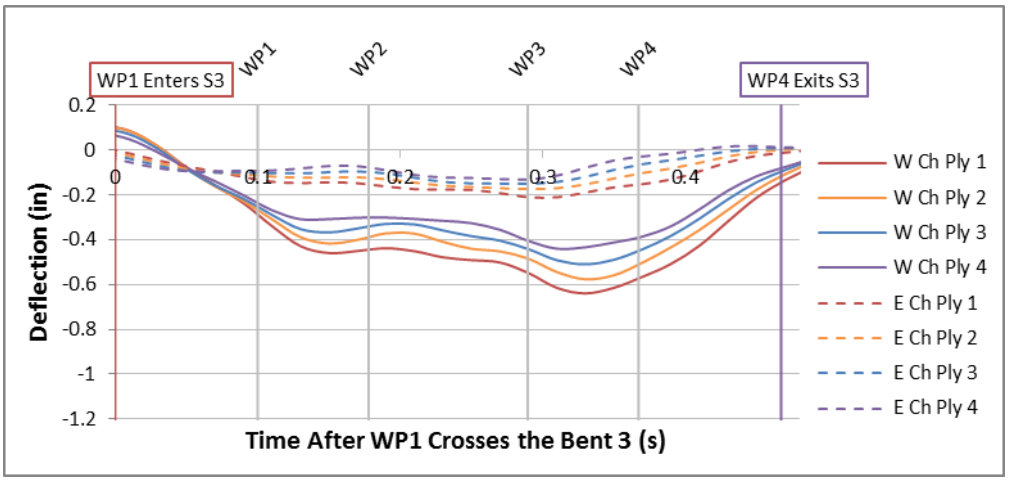

Figure A-174: IC5 50 MPH Net Midspan Stringer Deflections

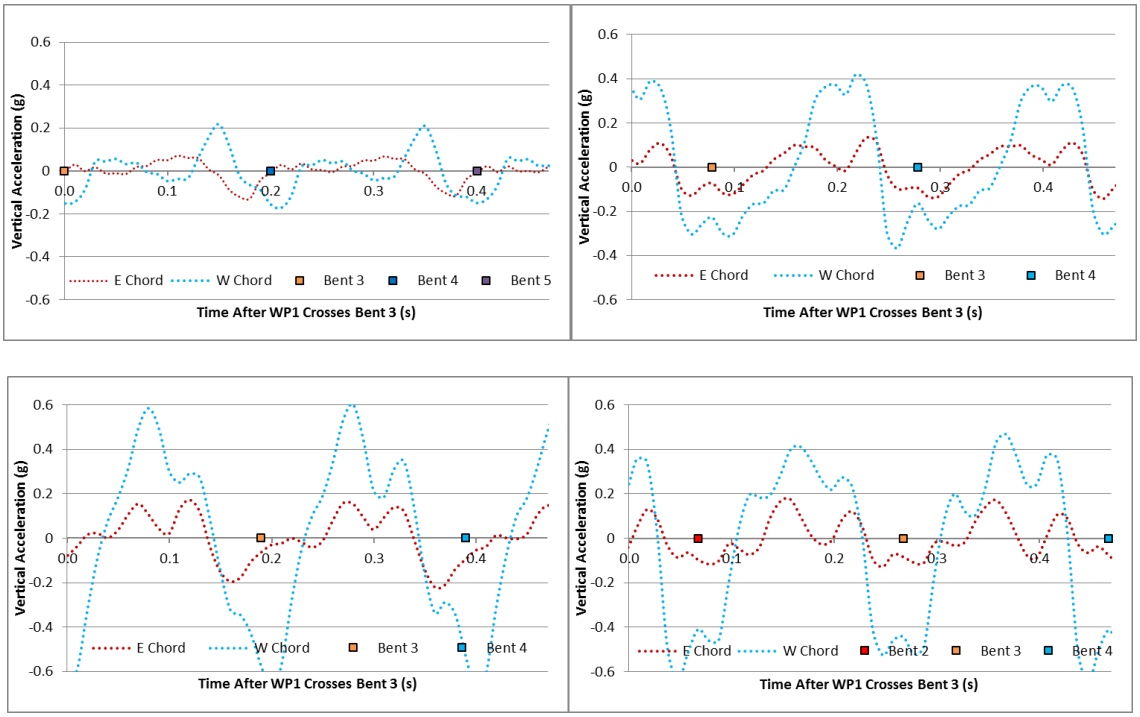

Figure A-175: IC5 50 MPH Wheel Path Accelerations: WP1 (Top Left), WP2 (Top Right), WP3 (Lower Left), WP4 (Lower Right)

A.4 Artificial Neural Network Results

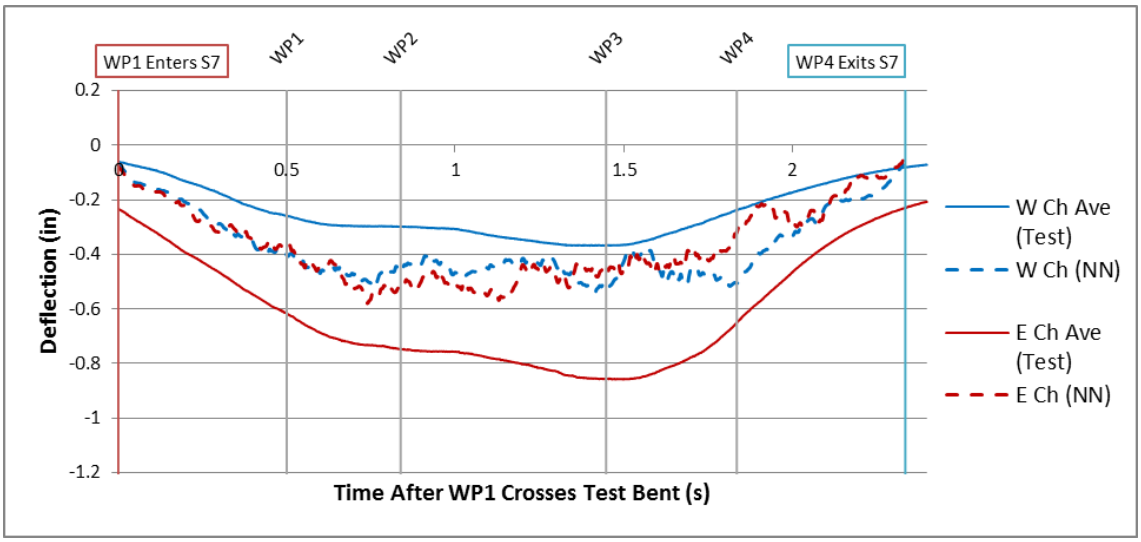

Figure A-176: Span 7 Test 1 Neural Network and Test Average Chord Deflection Comparison

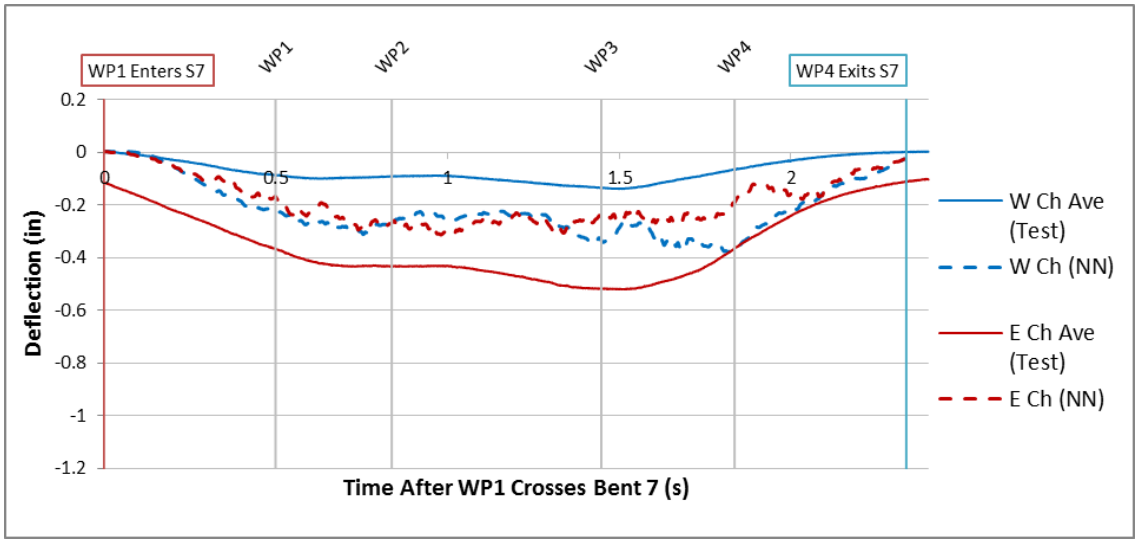

Figure A-177: Span 7 Test 1 Neural Network and Test Average Chord Net Deflection Comparison

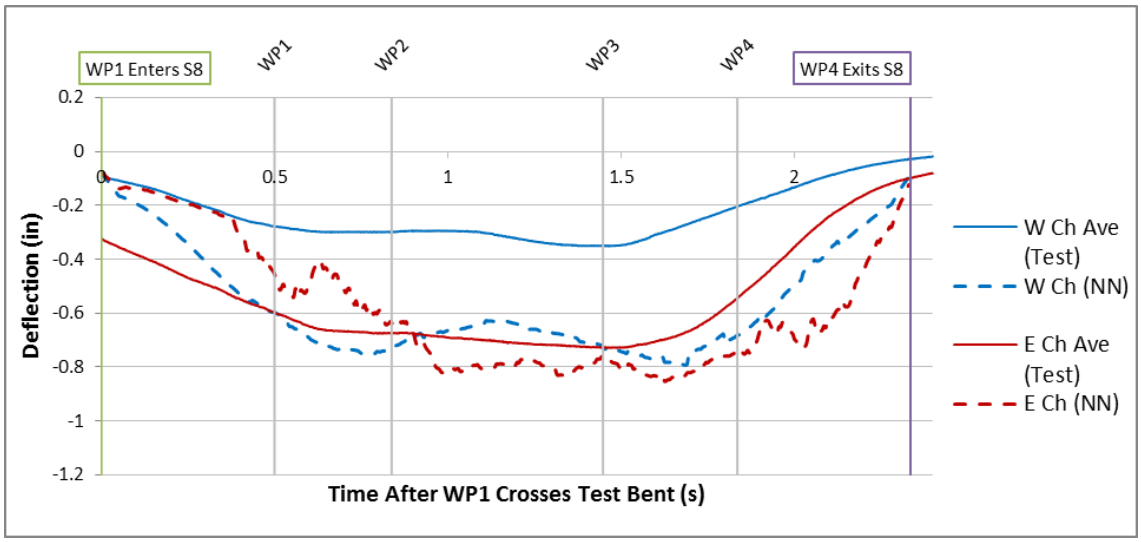

Figure A-178: Span 8 Test 1 Neural Network and Test Average Chord Deflection Comparison

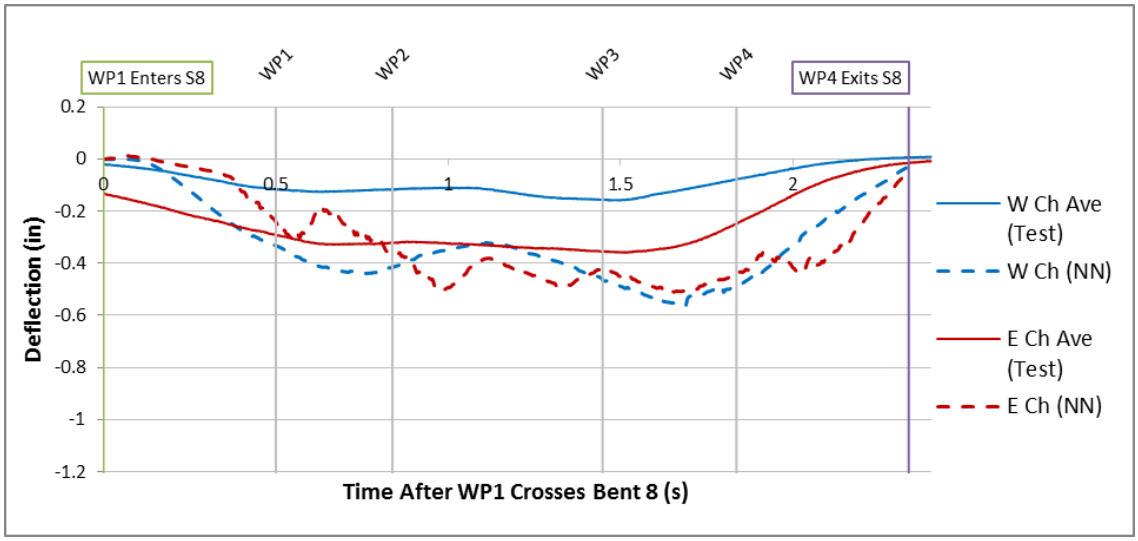

Figure A-179: Span 8 Test 1 Neural Network and Test Average Chord Net Deflection Comparison

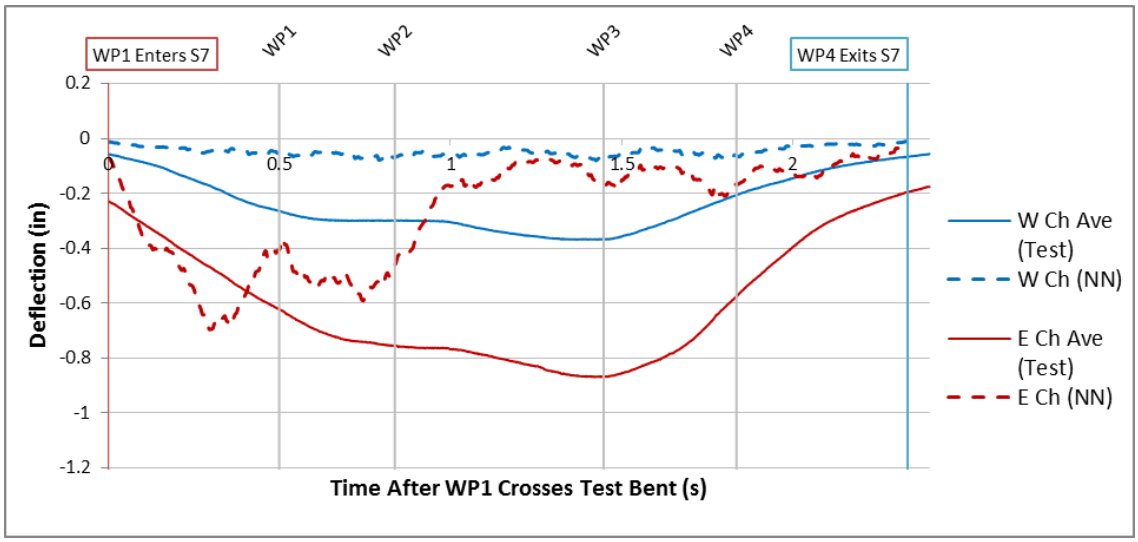

Figure A-180: Span 7 Test 2 Neural Network and Test Average Chord Deflection Comparison

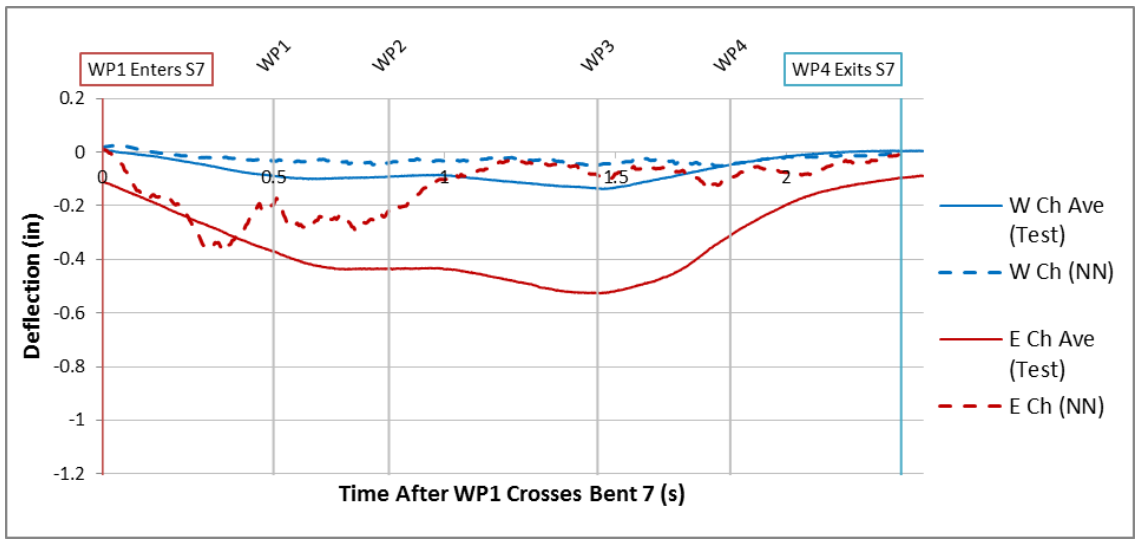

Figure A-181: Span 7 Test 2 Neural Network and Test Average Chord Net Deflection Comparison

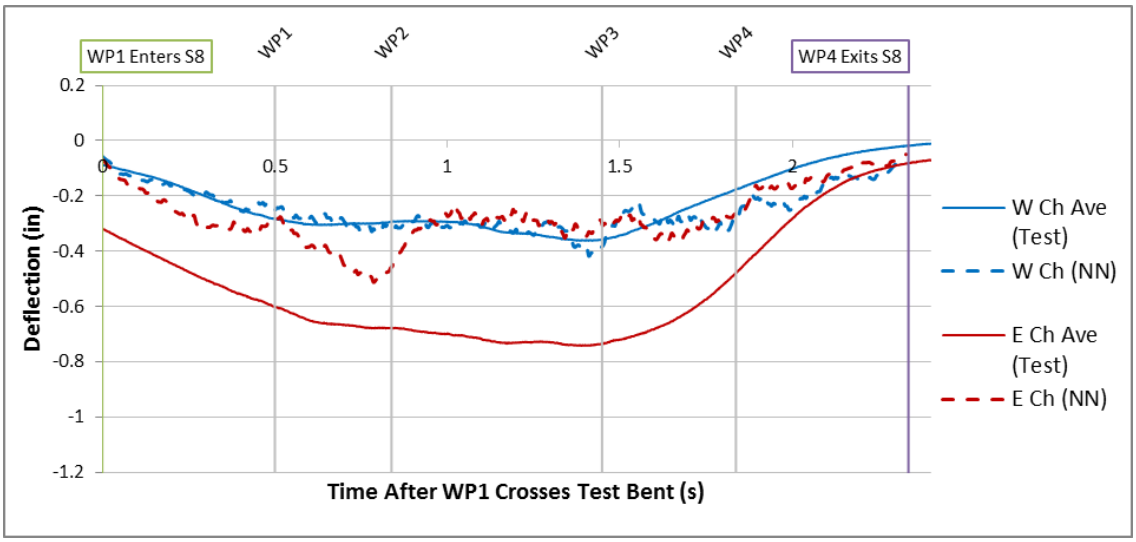

Figure A-182: Span 8 Test 2 Neural Network and Test Average Chord Deflection Comparison

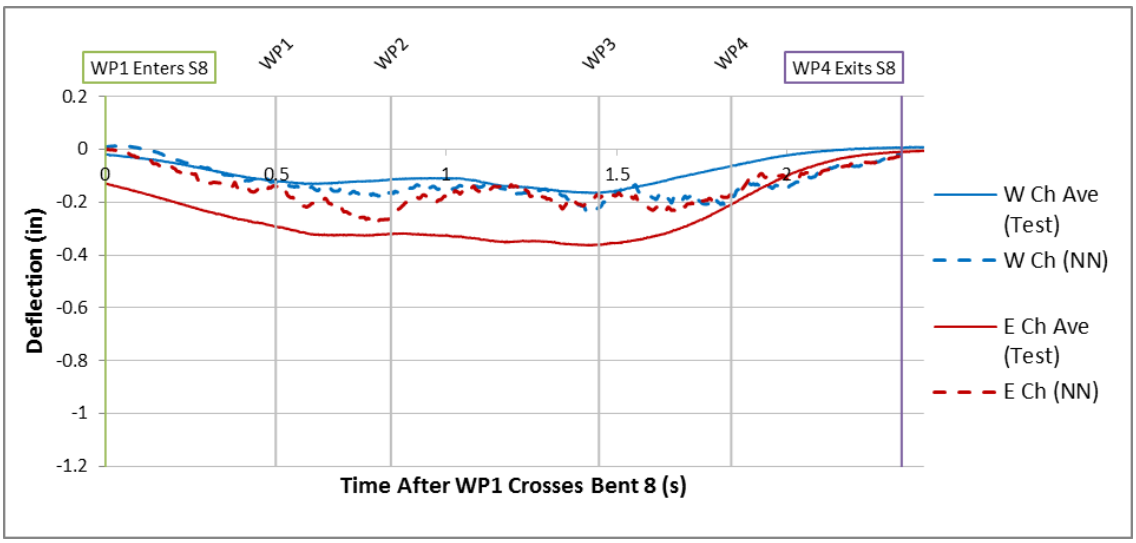

Figure A-183: Span 8 Test 2 Neural Network and Test Average Chord Net Deflection Comparison

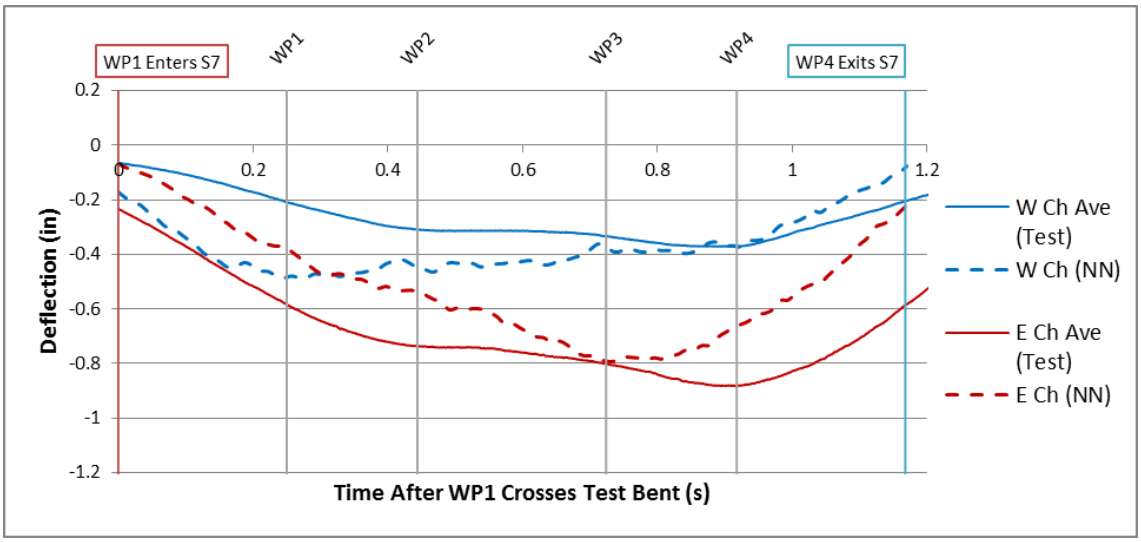

Figure A-184: Span 7 Test 3 Neural Network and Test Average Chord Deflection Comparison

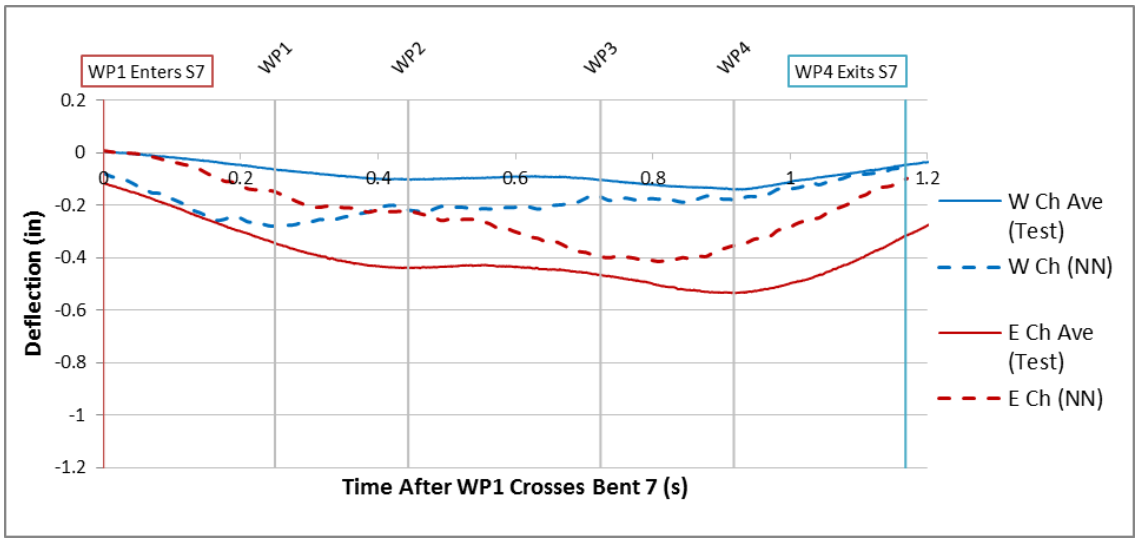

Figure A-185: Span 7 Test 3 Neural Network and Test Average Chord Net Deflection Comparison

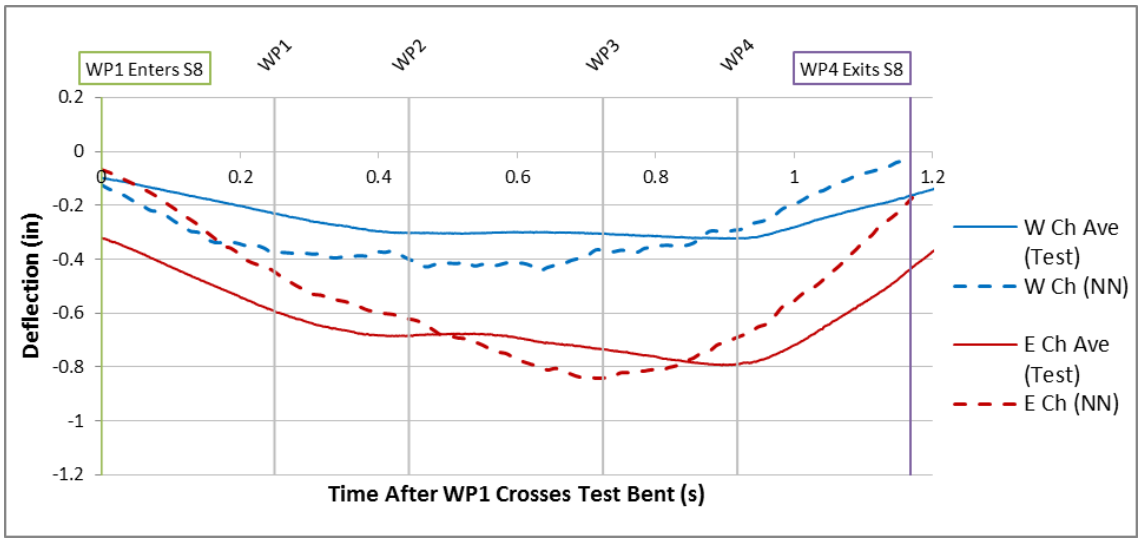

Figure A-186: Span 8 Test 3 Neural Network and Test Average Chord Deflection Comparison

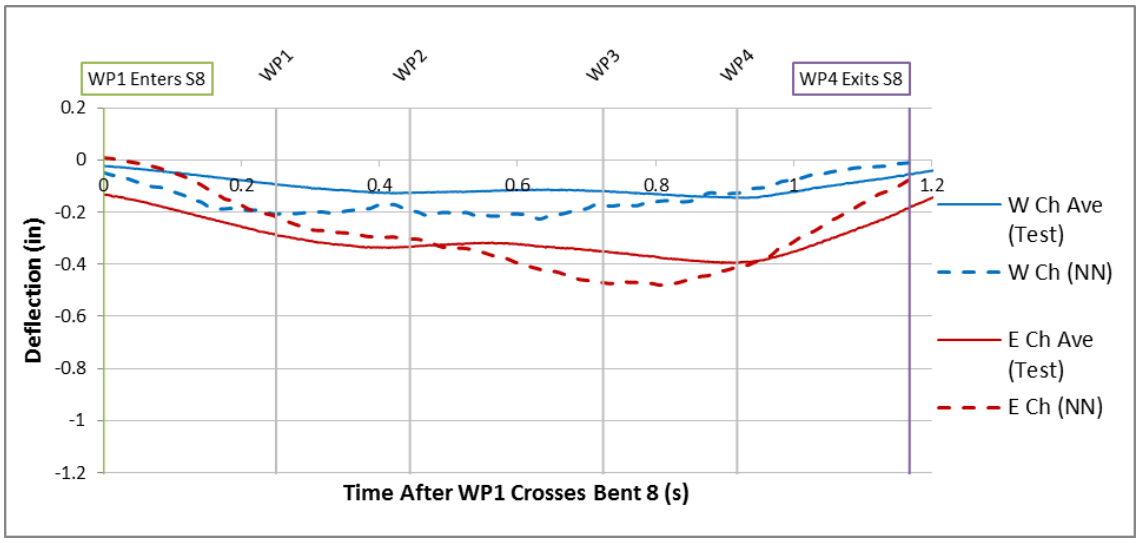

Figure A-187: Span 8 Test 3 Neural Network and Test Average Chord Net Deflection Comparison

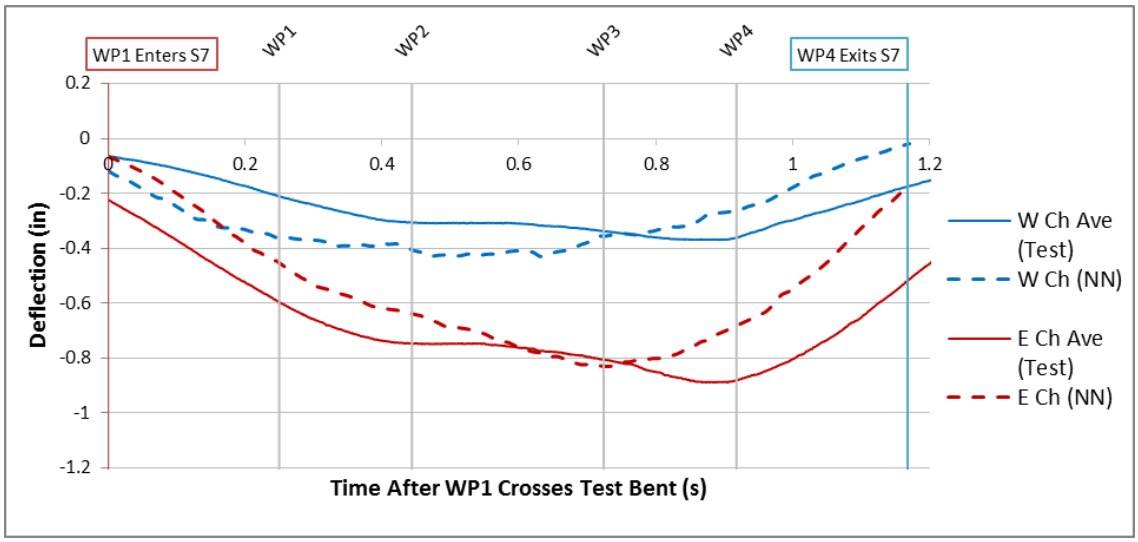

Figure A-188: Span 7 Test 4 Neural Network and Test Average Chord Deflection Comparison

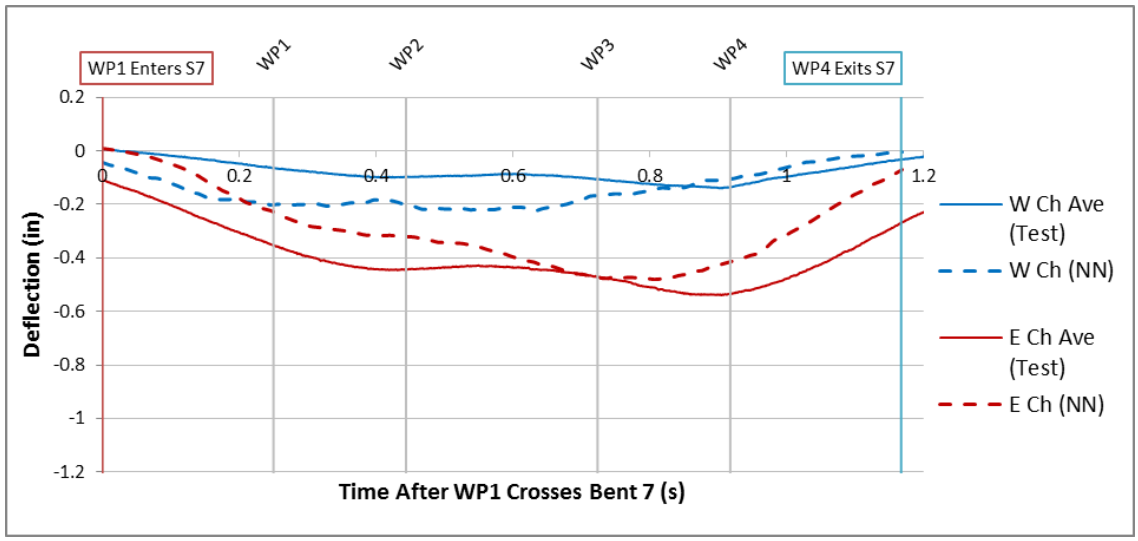

Figure A-189: Span 7 Test 4 Neural Network and Test Average Chord Net Deflection Comparison

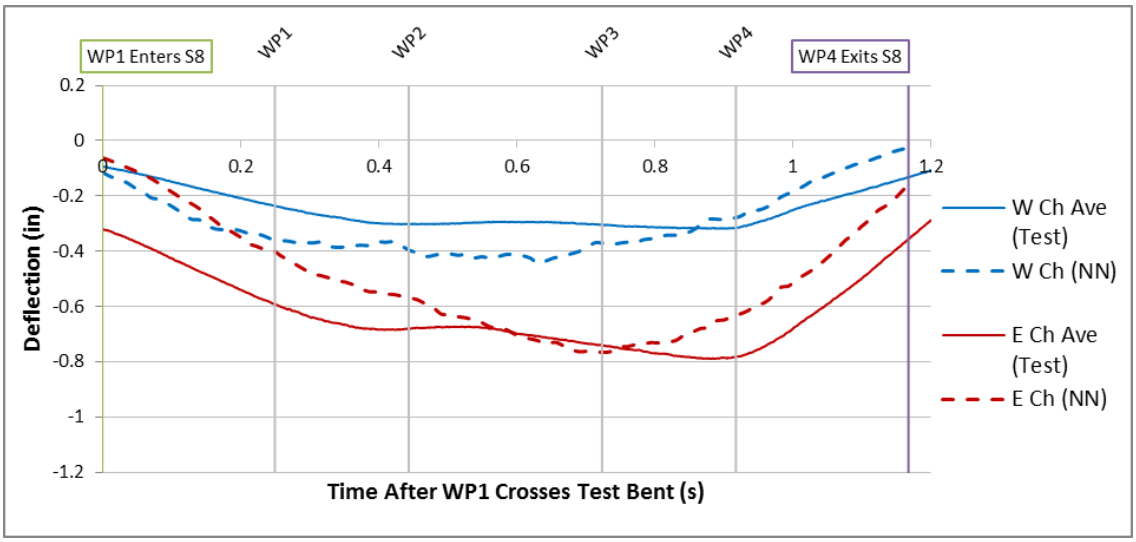

Figure A-190: Span 8 Test 4 Neural Network and Test Average Chord Deflection Comparison

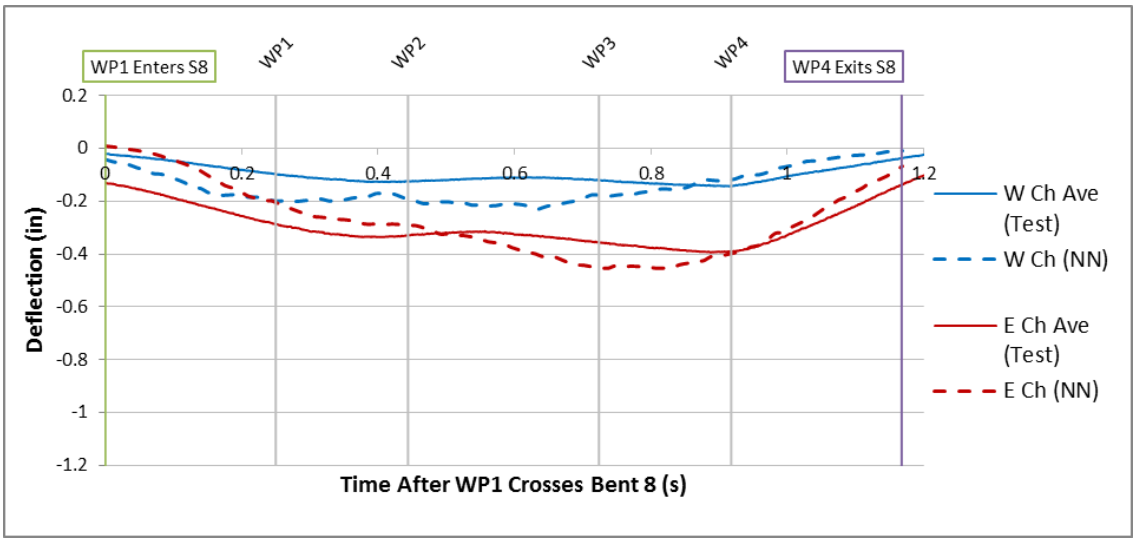

Figure A-191: Span 8 Test 4 Neural Network and Test Average Chord Net Deflection Comparison

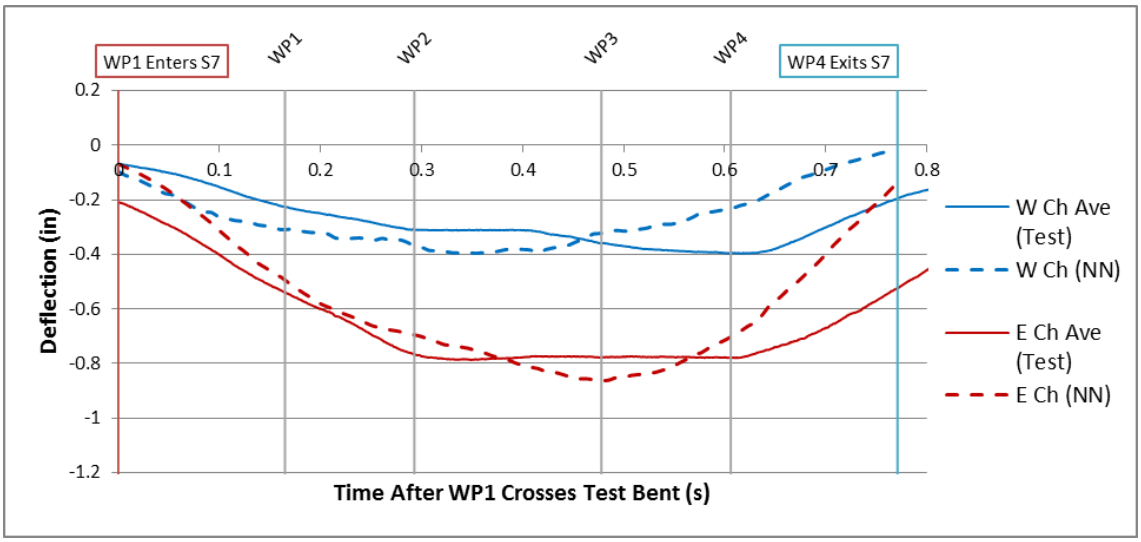

Figure A-192: Span 7 Test 5 Neural Network and Test Average Chord Deflection Comparison

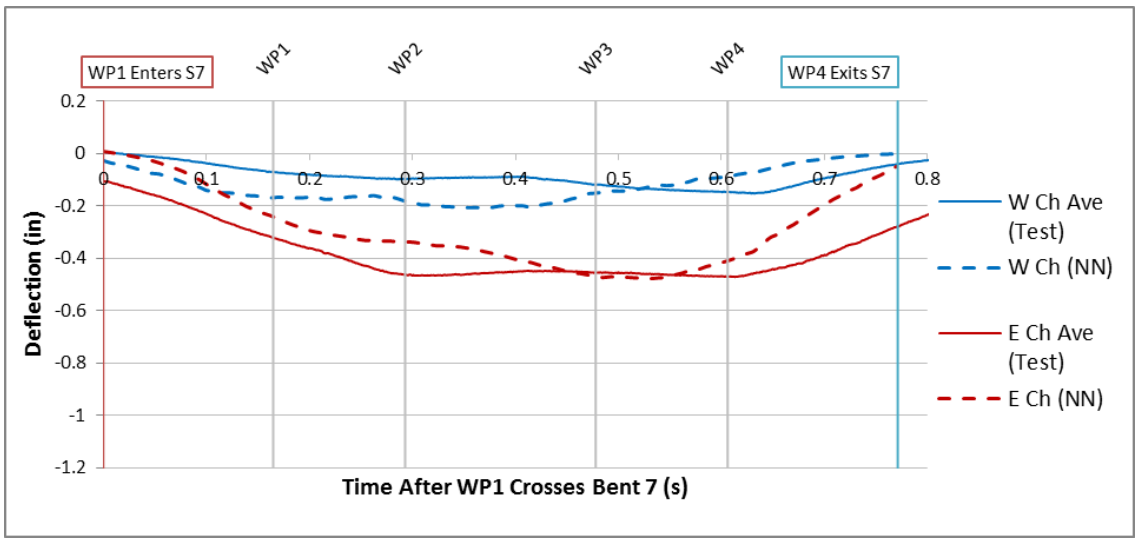

Figure A-193: Span 7 Test 5 Neural Network and Test Average Chord Net Deflection Comparison

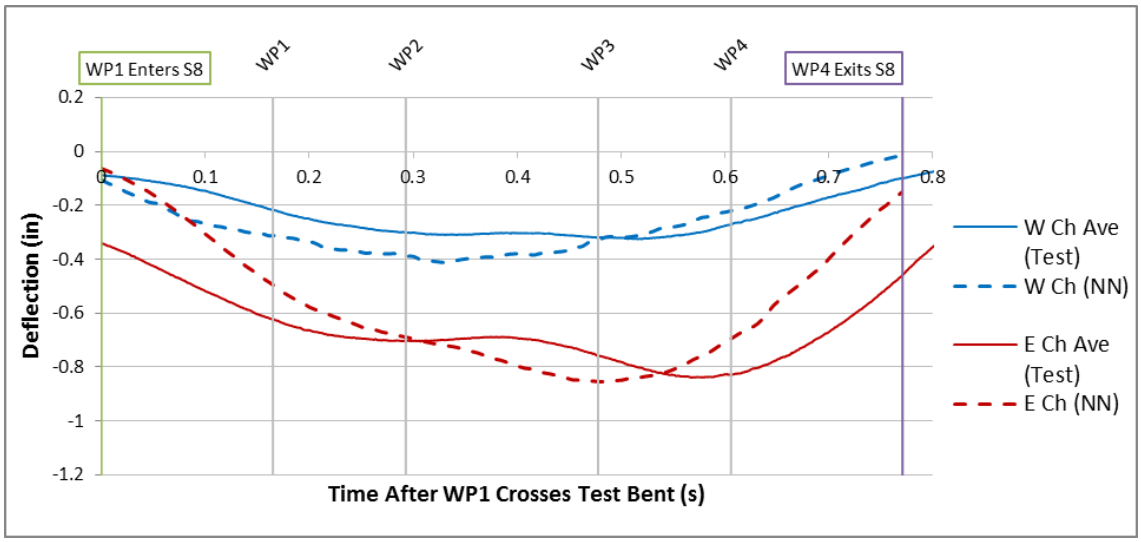

Figure A-194: Span 8 Test 5 Neural Network and Test Average Chord Deflection Comparison

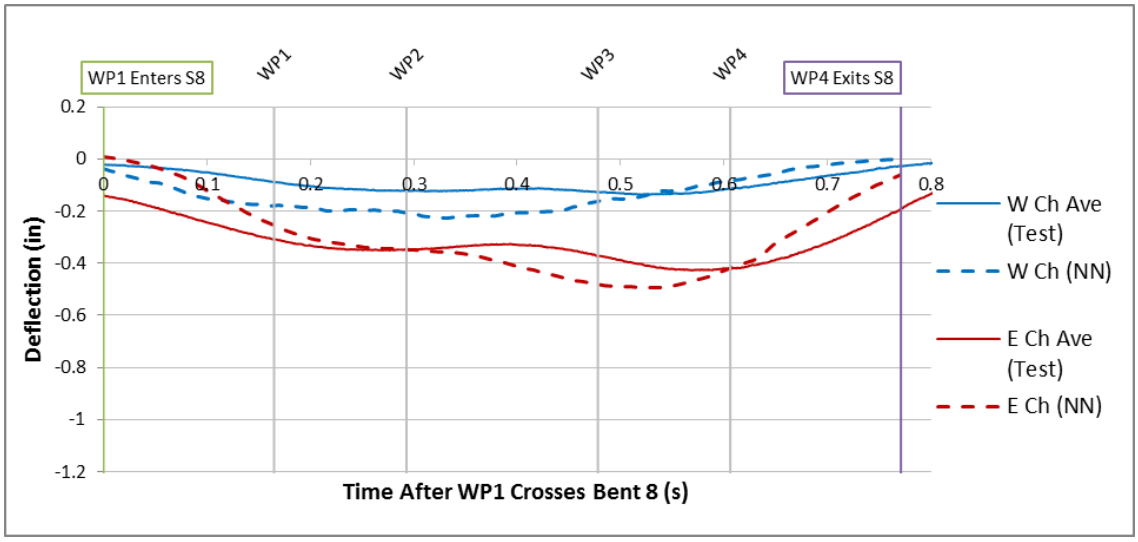

Figure A-195: Span 8 Test 5 Neural Network and Test Average Chord Net Deflection Comparison

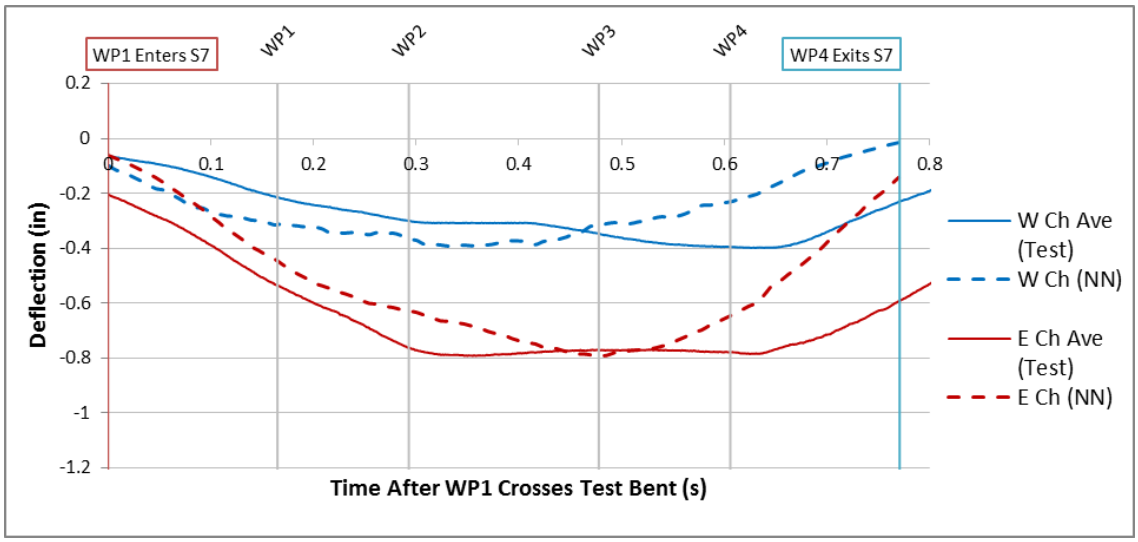

Figure A-196: Span 7 Test 6 Neural Network and Test Average Chord Deflection Comparison

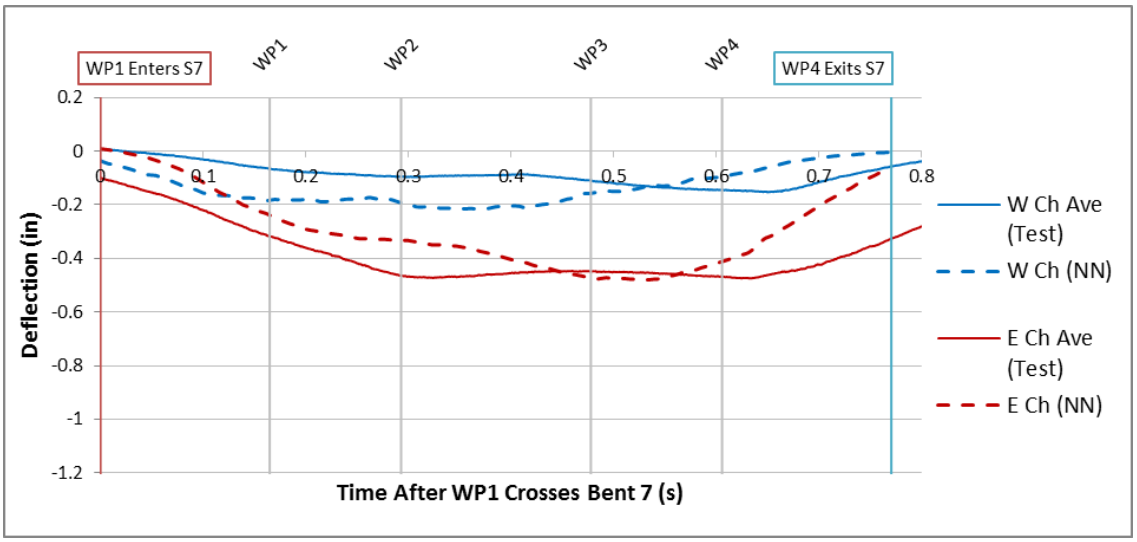

Figure A-197: Span 7 Test 6 Neural Network and Test Average Chord Net Deflection Comparison

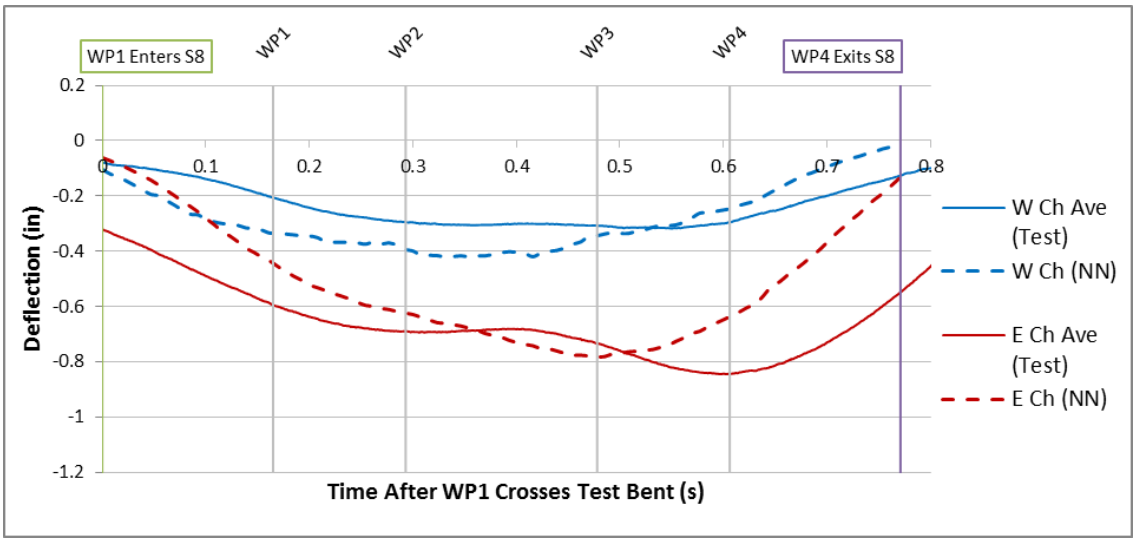

Figure A-198: Span 8 Test 6 Neural Network and Test Average Chord Deflection Comparison

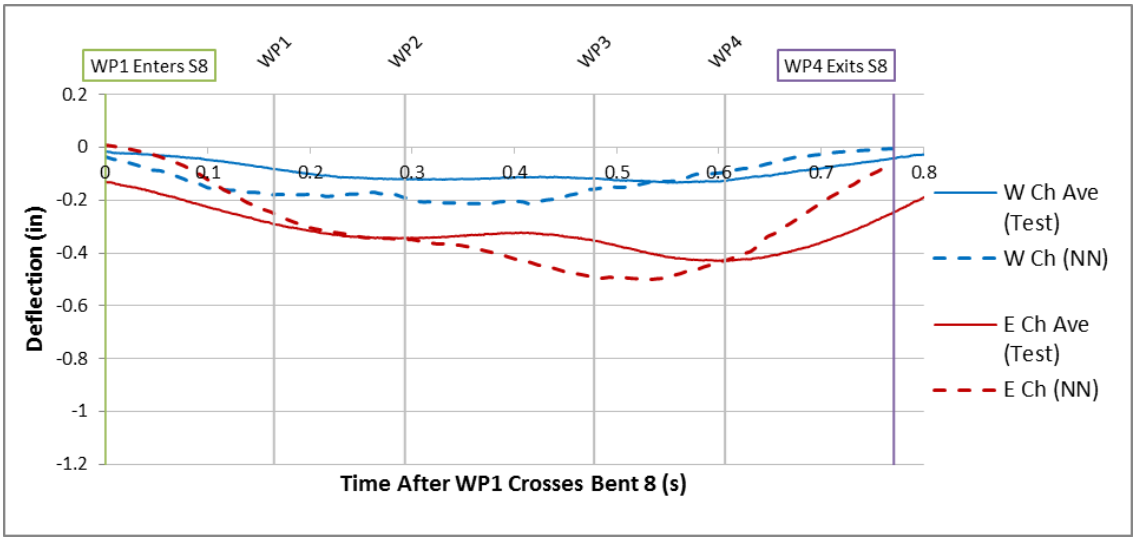

Figure A-199: Span 8 Test 6 Neural Network and Test Average Chord Net Deflection Comparison

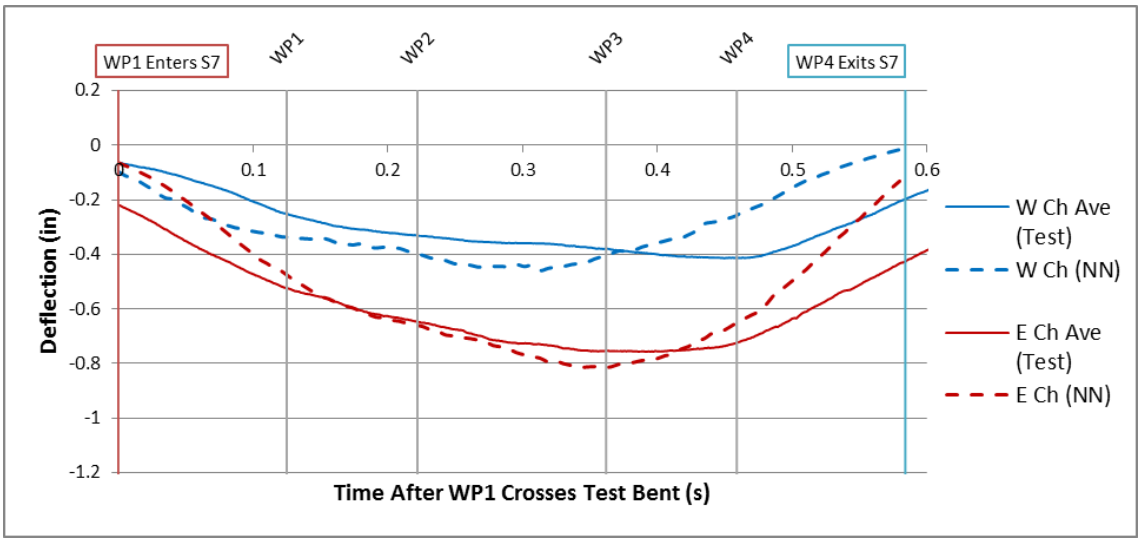

Figure A-200: Span 7 Test 7 Neural Network and Test Average Chord Deflection Comparison

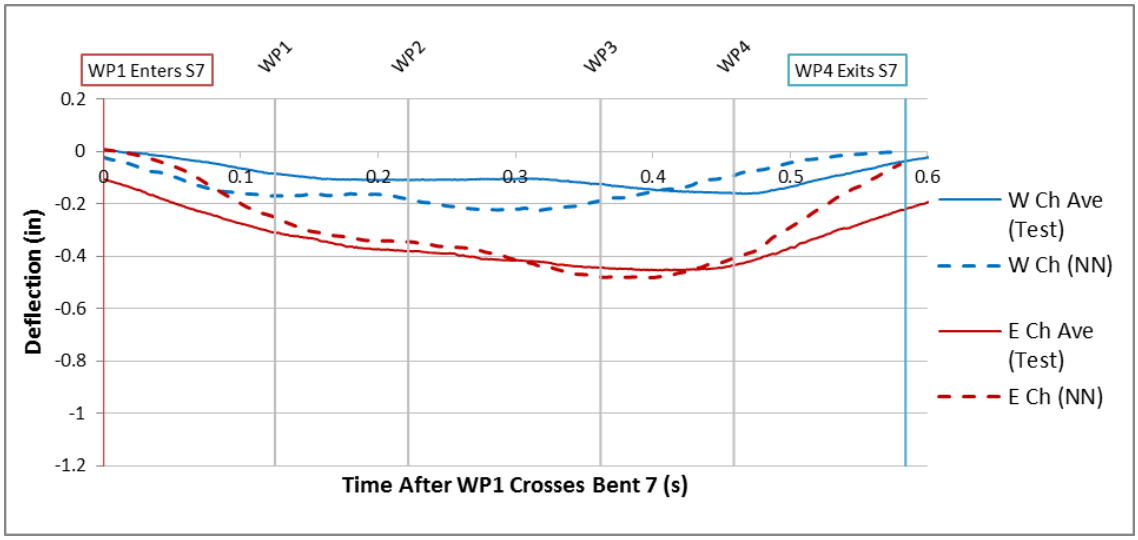

Figure A-201: Span 7 Test 7 Neural Network and Test Average Chord Net Deflection Comparison

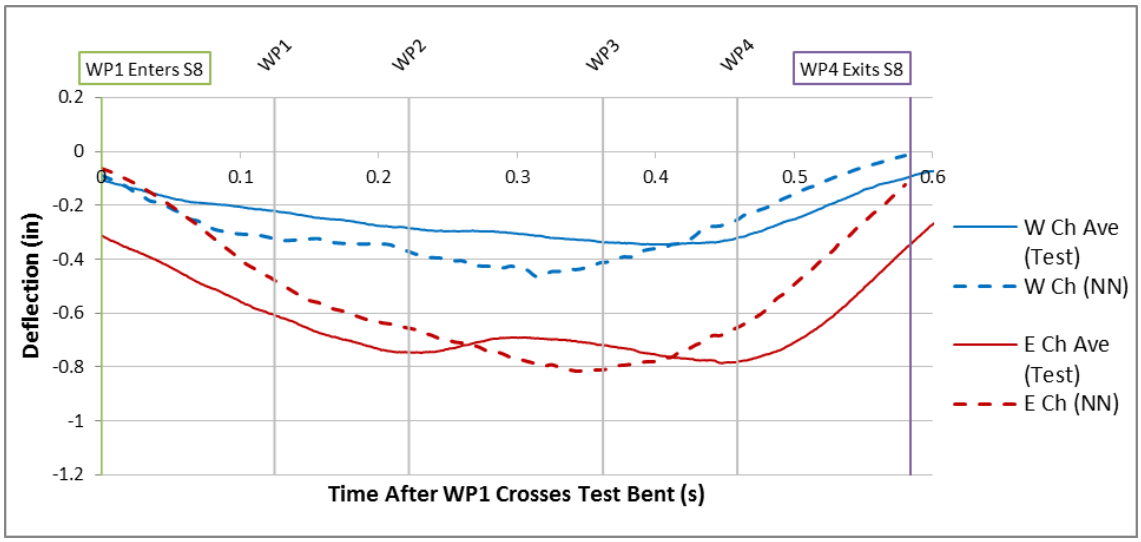

Figure A-202: Span 8 Test 7 Neural Network and Test Average Chord Deflection Comparison

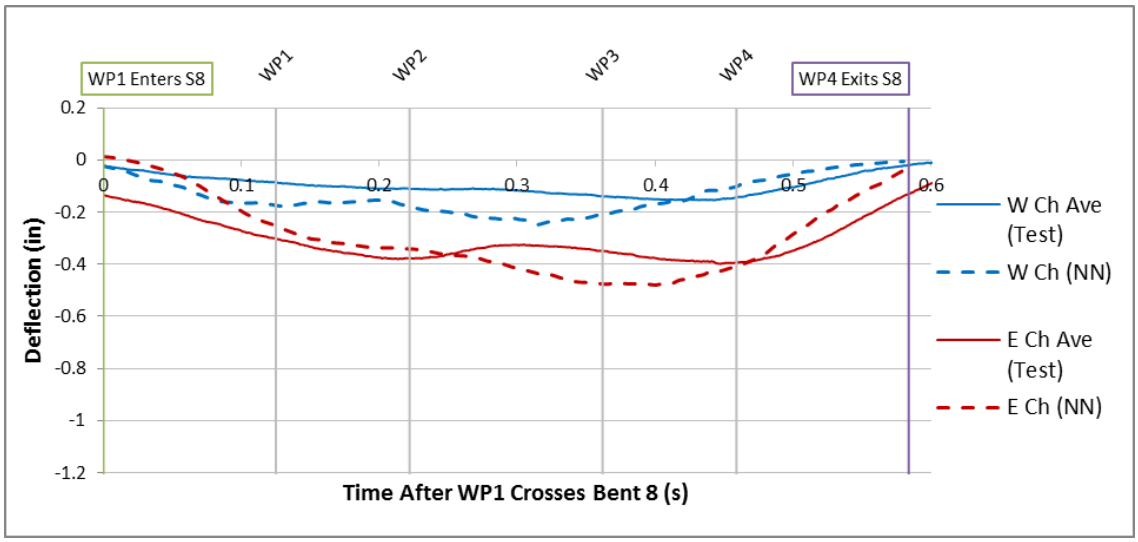

Figure A-203: Span 8 Test 7 Neural Network and Test Average Chord Net Deflection Comparison

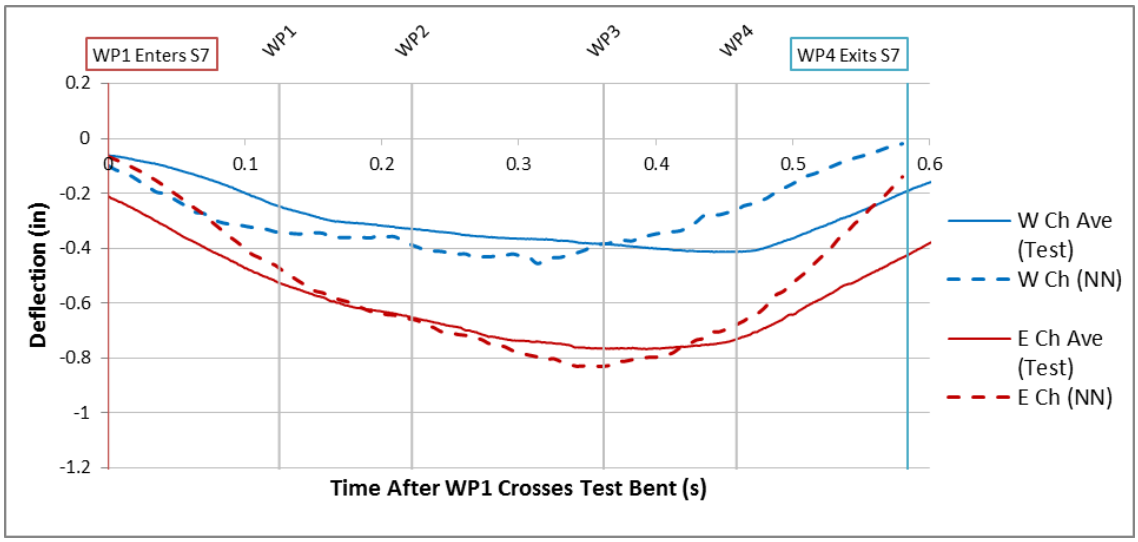

Figure A-204: Span 7 Test 8 Neural Network and Test Average Chord Deflection Comparison

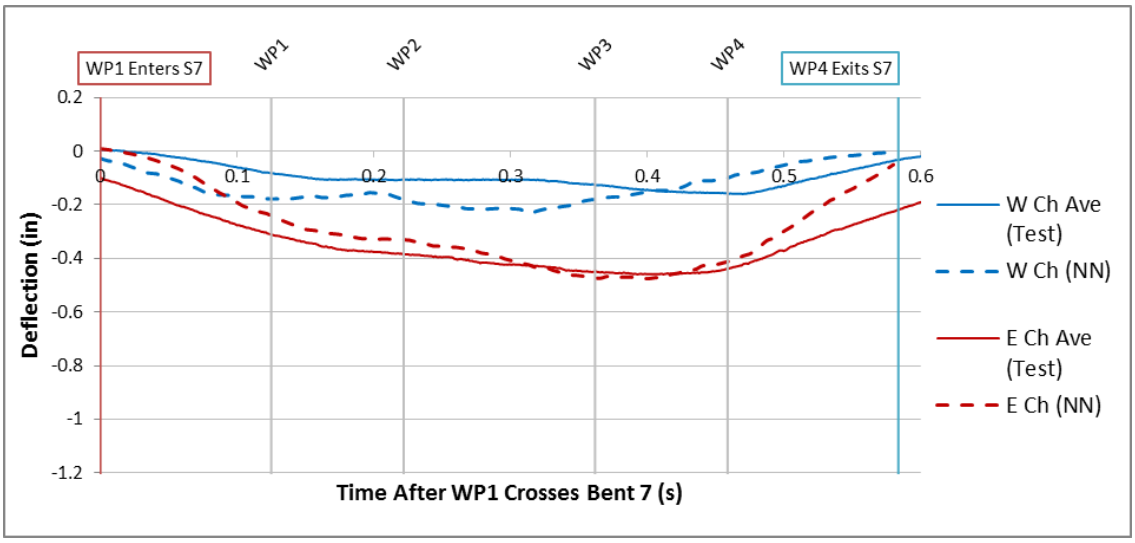

Figure A-205: Span 7 Test 8 Neural Network and Test Average Chord Net Deflection Comparison

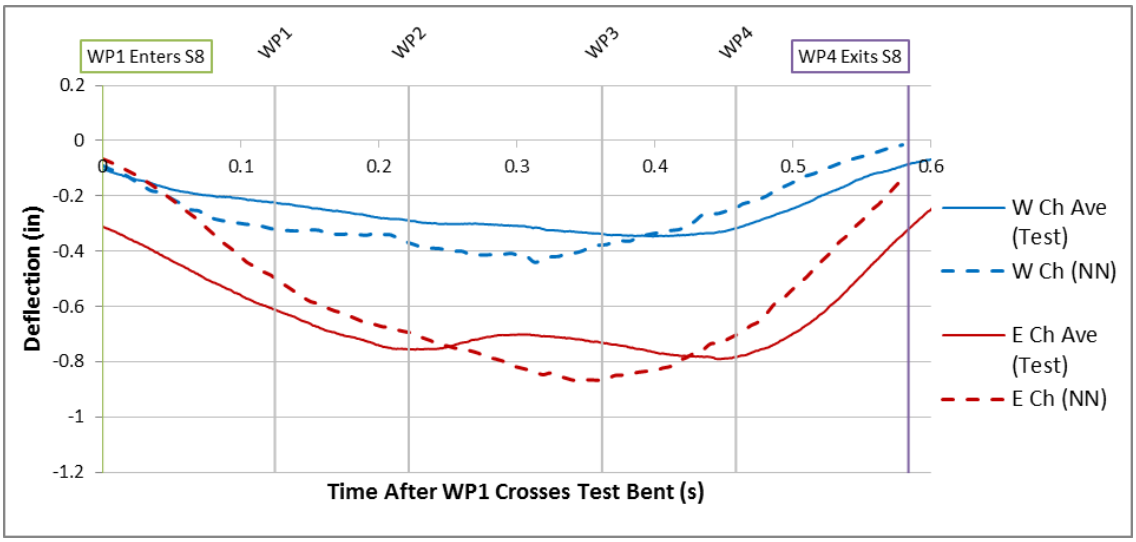

Figure A-206: Span 8 Test 8 Neural Network and Test Average Chord Deflection Comparison

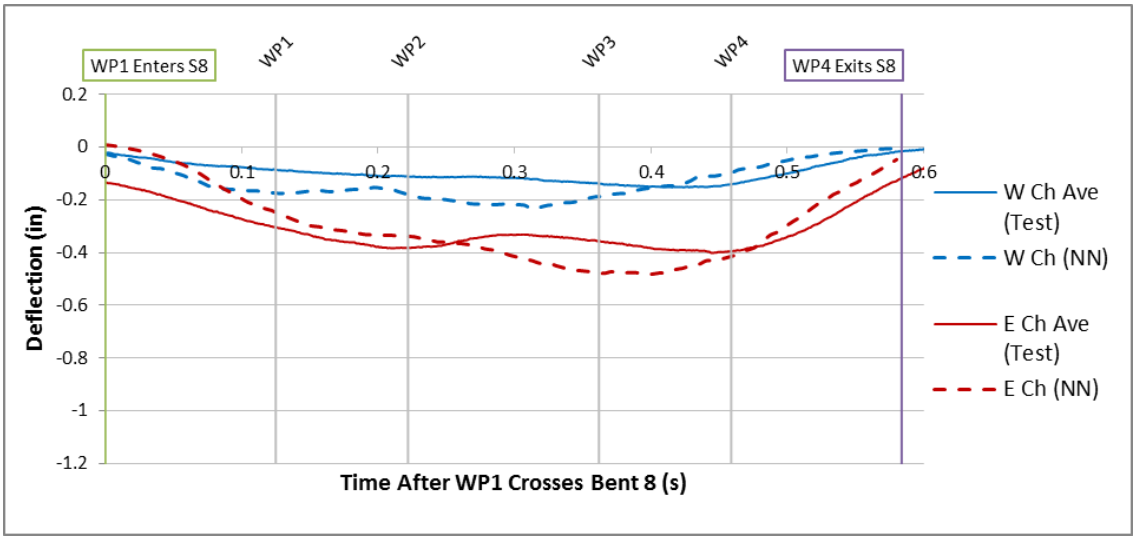

Figure A-207: Span 8 Test 8 Neural Network and Test Average Chord Net Deflection Comparison

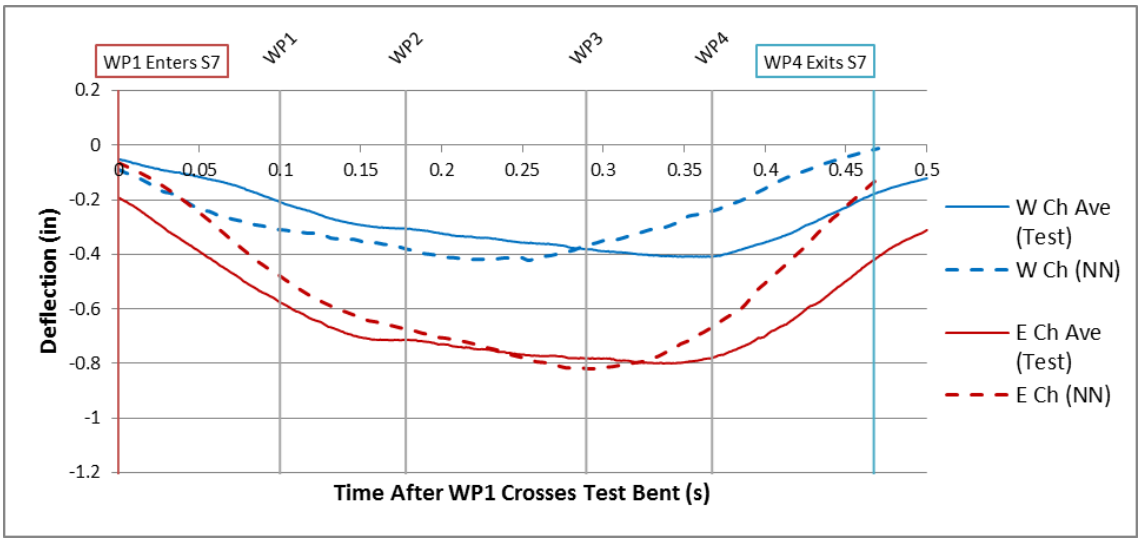

Figure A-208: Span 7 Test 9 Neural Network and Test Average Chord Deflection Comparison

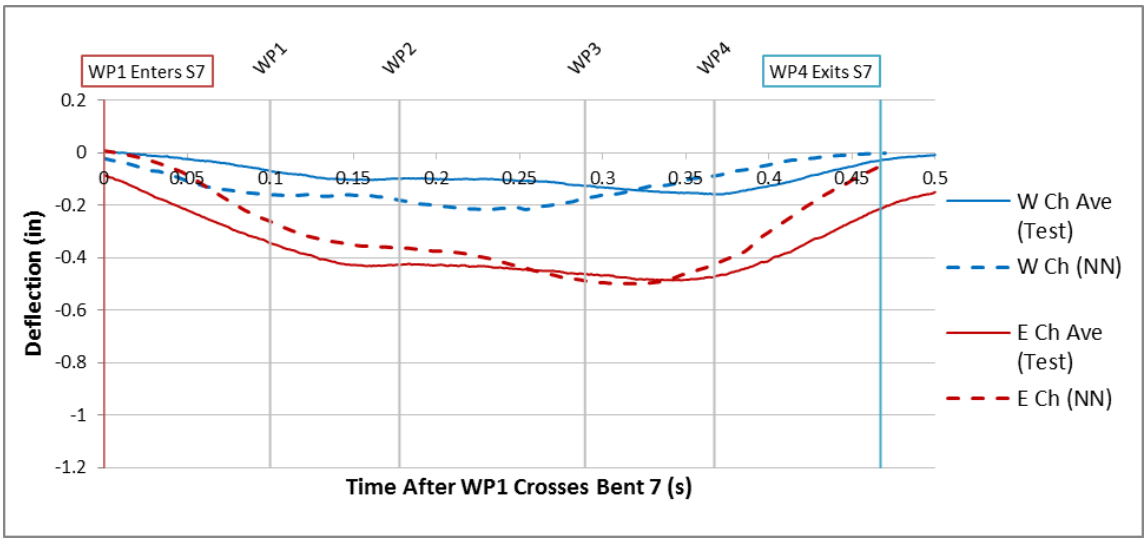

Figure A-209: Span 7 Test 9 Neural Network and Test Average Chord Net Deflection Comparison

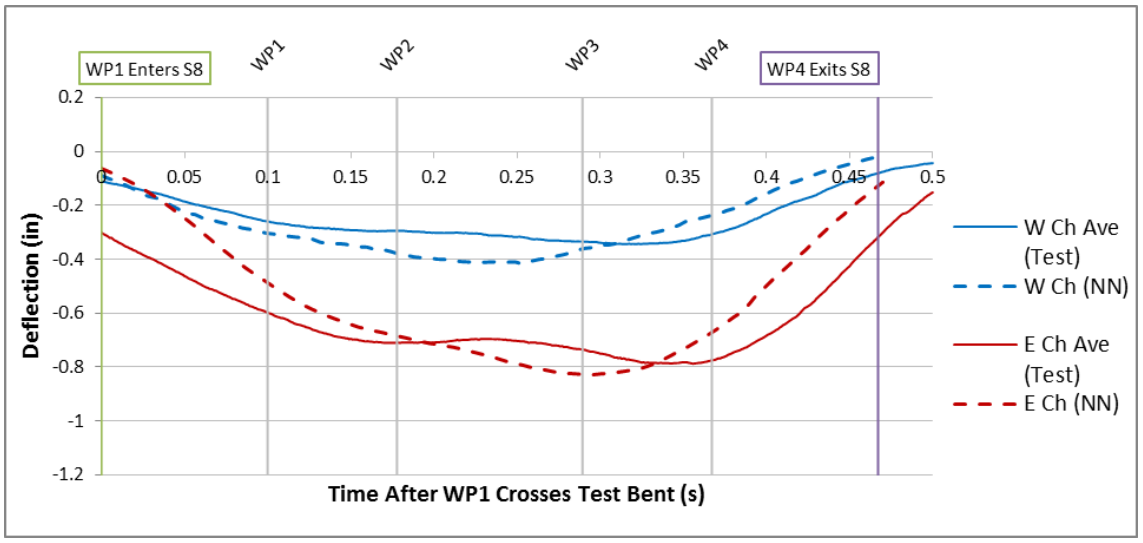

Figure A-210: Span 8 Test 9 Neural Network and Test Average Chord Deflection Comparison

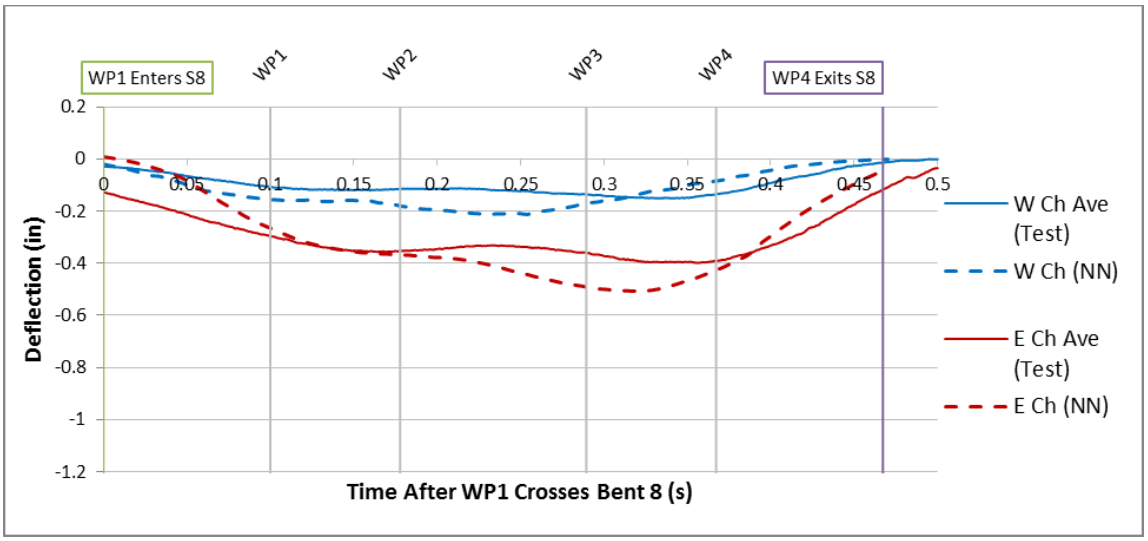

Figure A-211: Span 8 Test 9 Neural Network and Test Average Chord Net Deflection Comparison

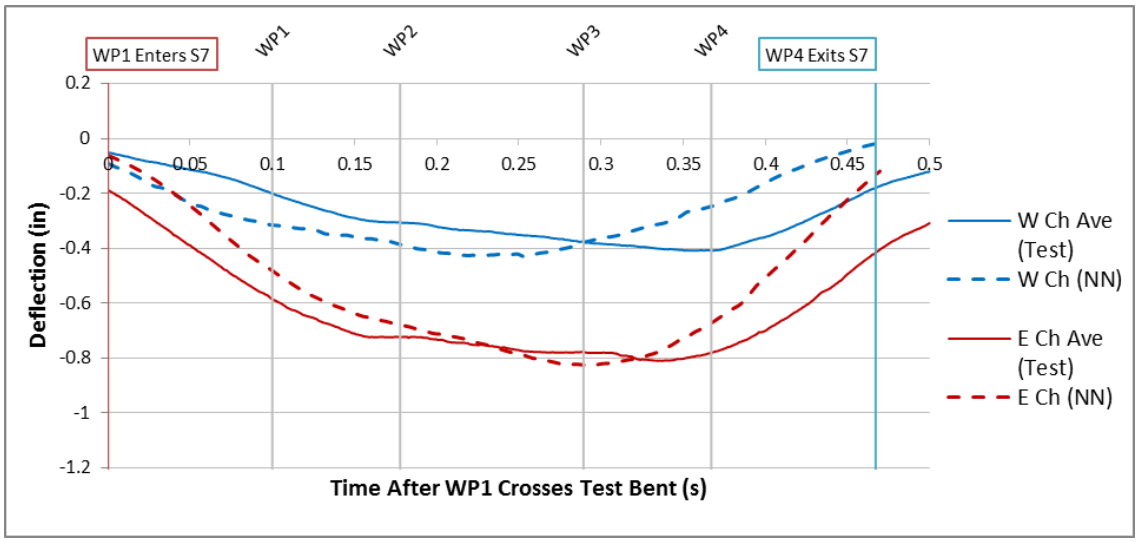

Figure A-212: Span 7 Test 10 Neural Network and Test Average Chord Deflection Comparison

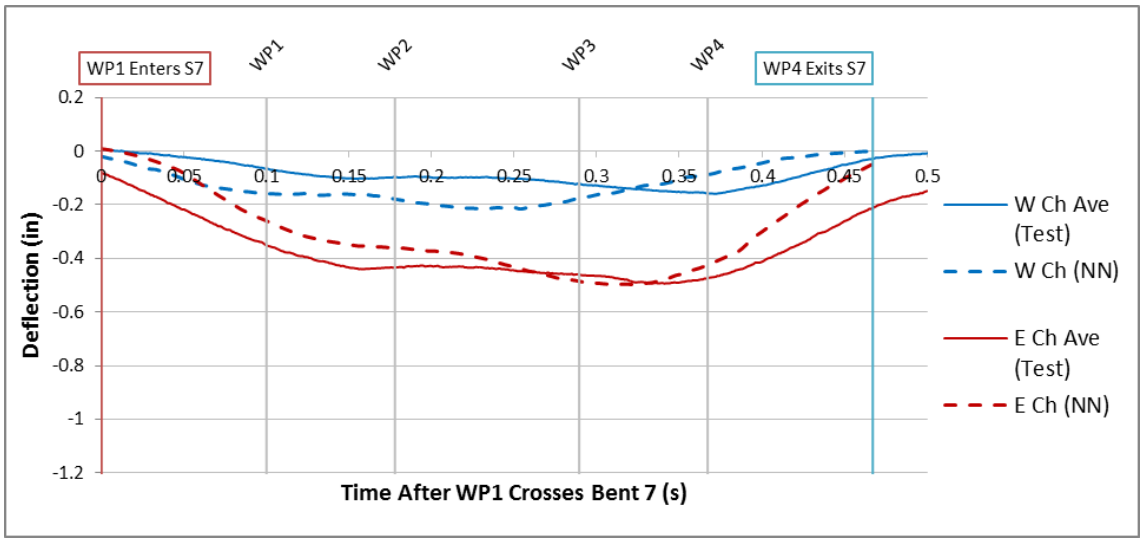

Figure A-213: Span 7 Test 10 Neural Network and Test Average Chord Net Deflection Comparison

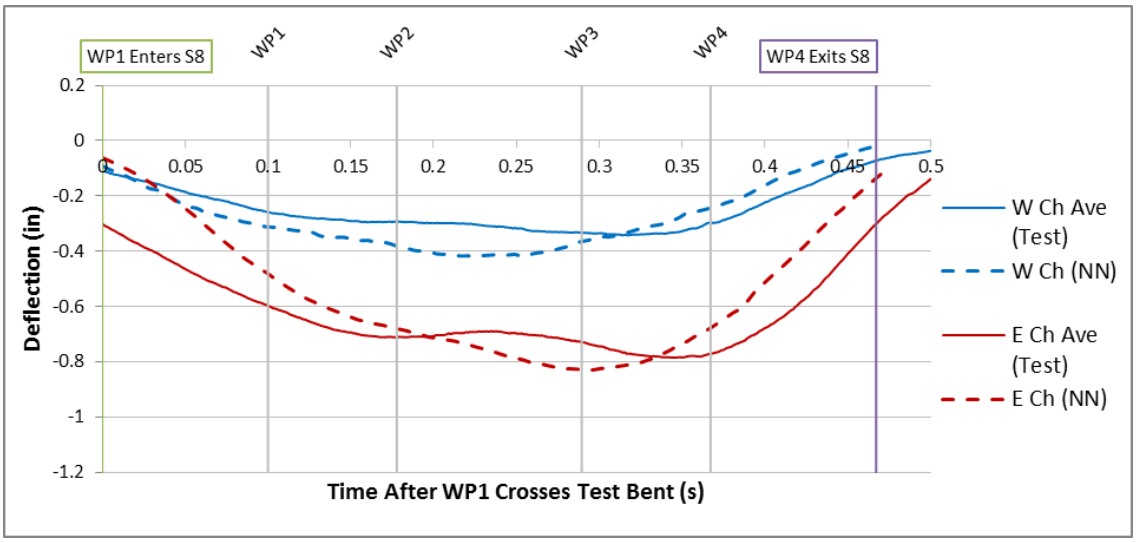

Figure A-214: Span 8 Test 10 Neural Network and Test Average Chord Deflection Comparison

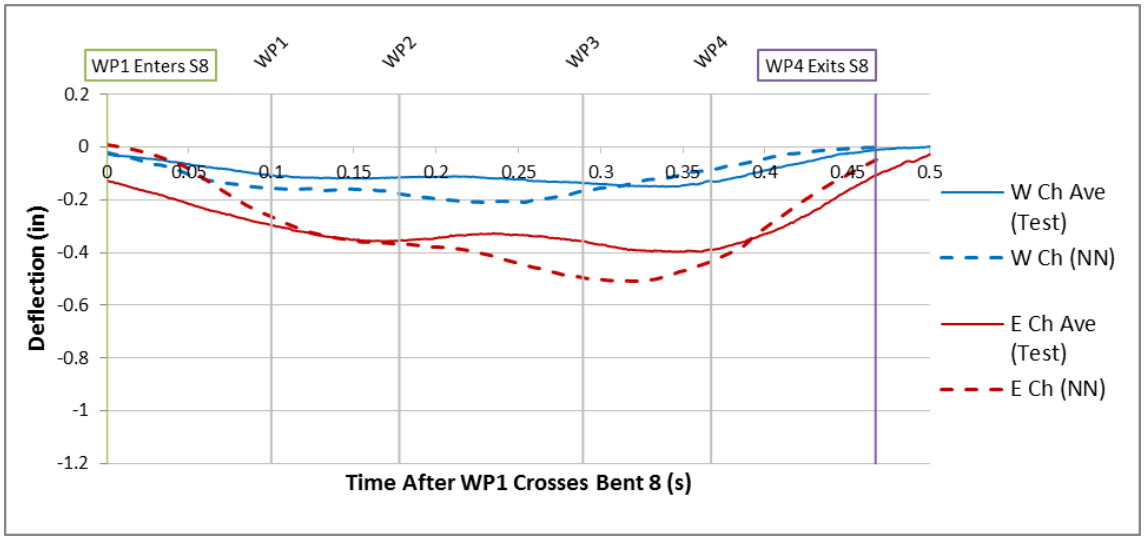

Figure A-215: Span 8 Test 10 Neural Network and Test Average Chord Net Deflection Comparison

A.5 Approximate Stringer Deflection from Neural Networks

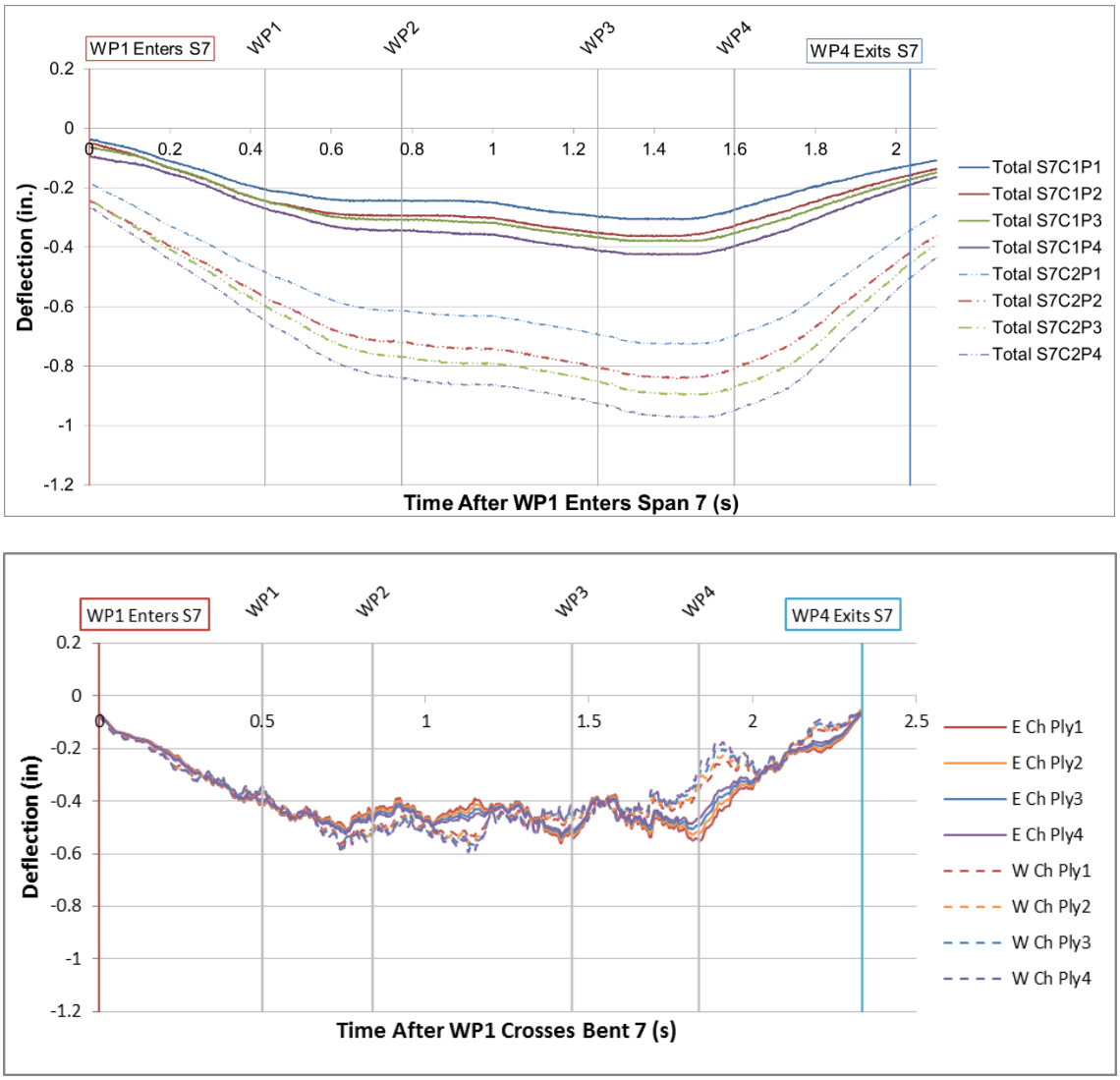

Figure A-216: Stringer Time History Comparison Test 1 Span 7: Experimental (Top), Predicted (Bottom)

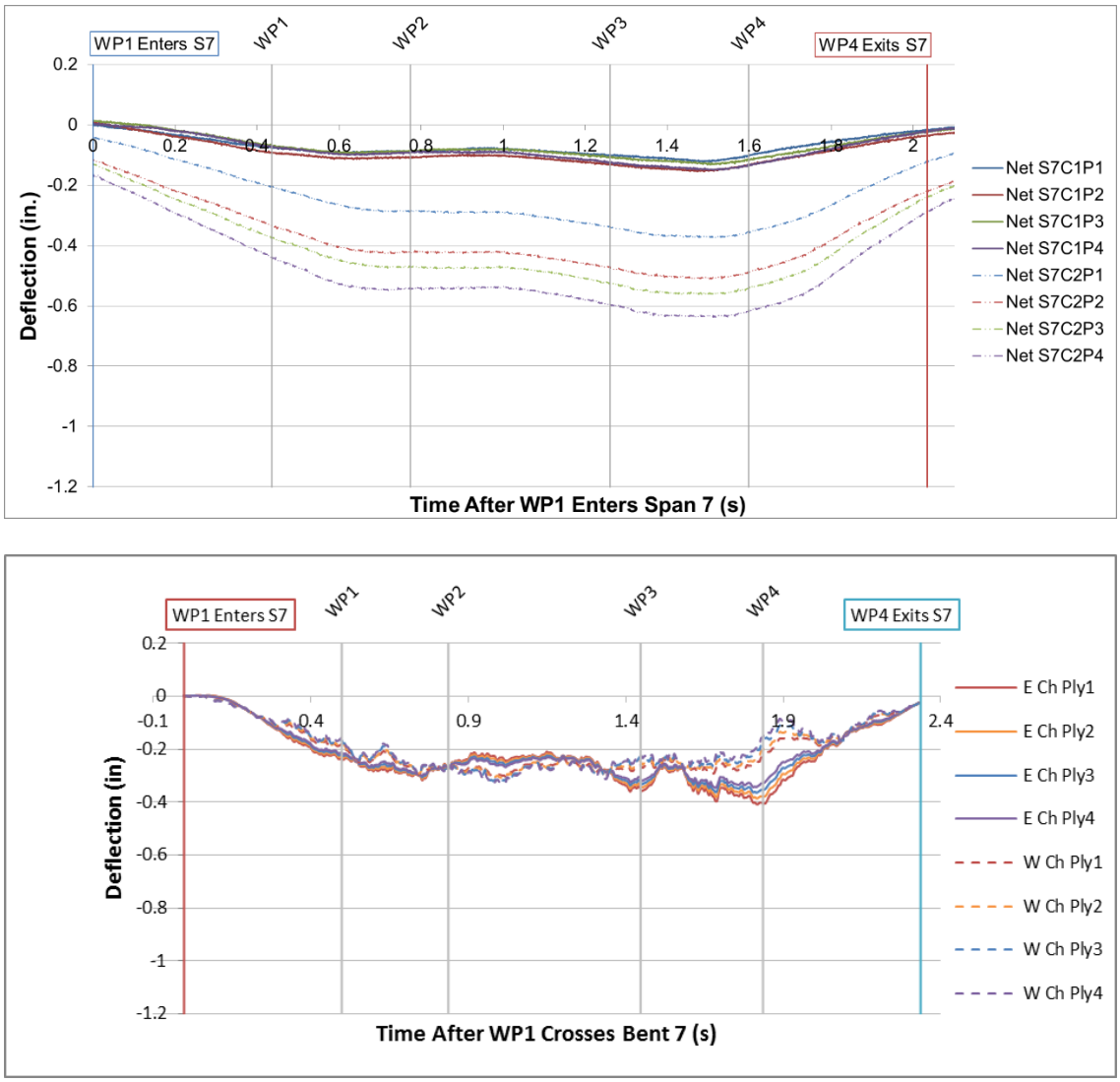

Figure A-217: Net Stringer Time History Comparison Test 1 Span 7: Experimental (Top), Predicted (Bottom)

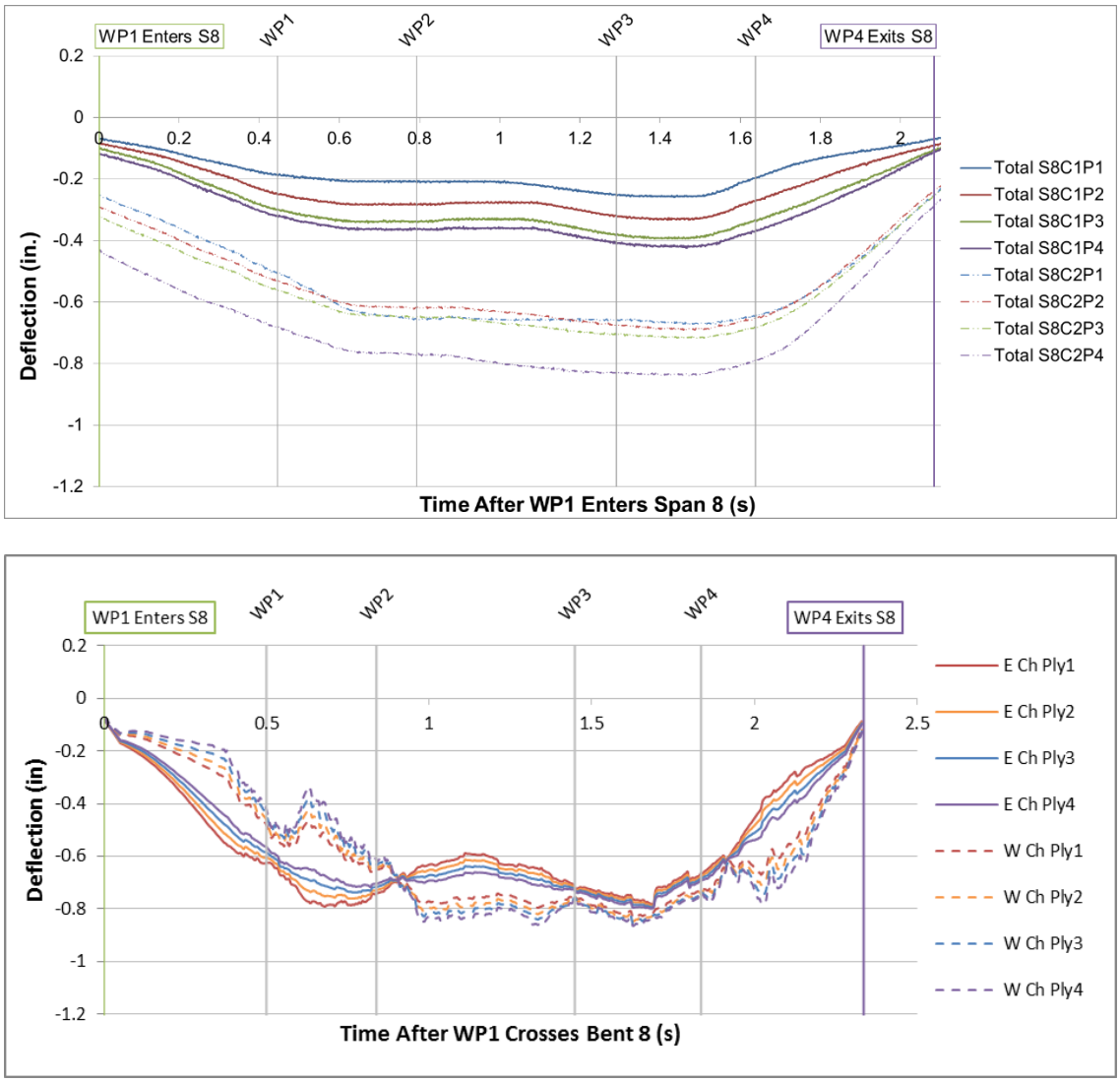

Figure A-218: Stringer Time History Comparison Test 1 Span 8: Experimental (Top), Predicted (Bottom)

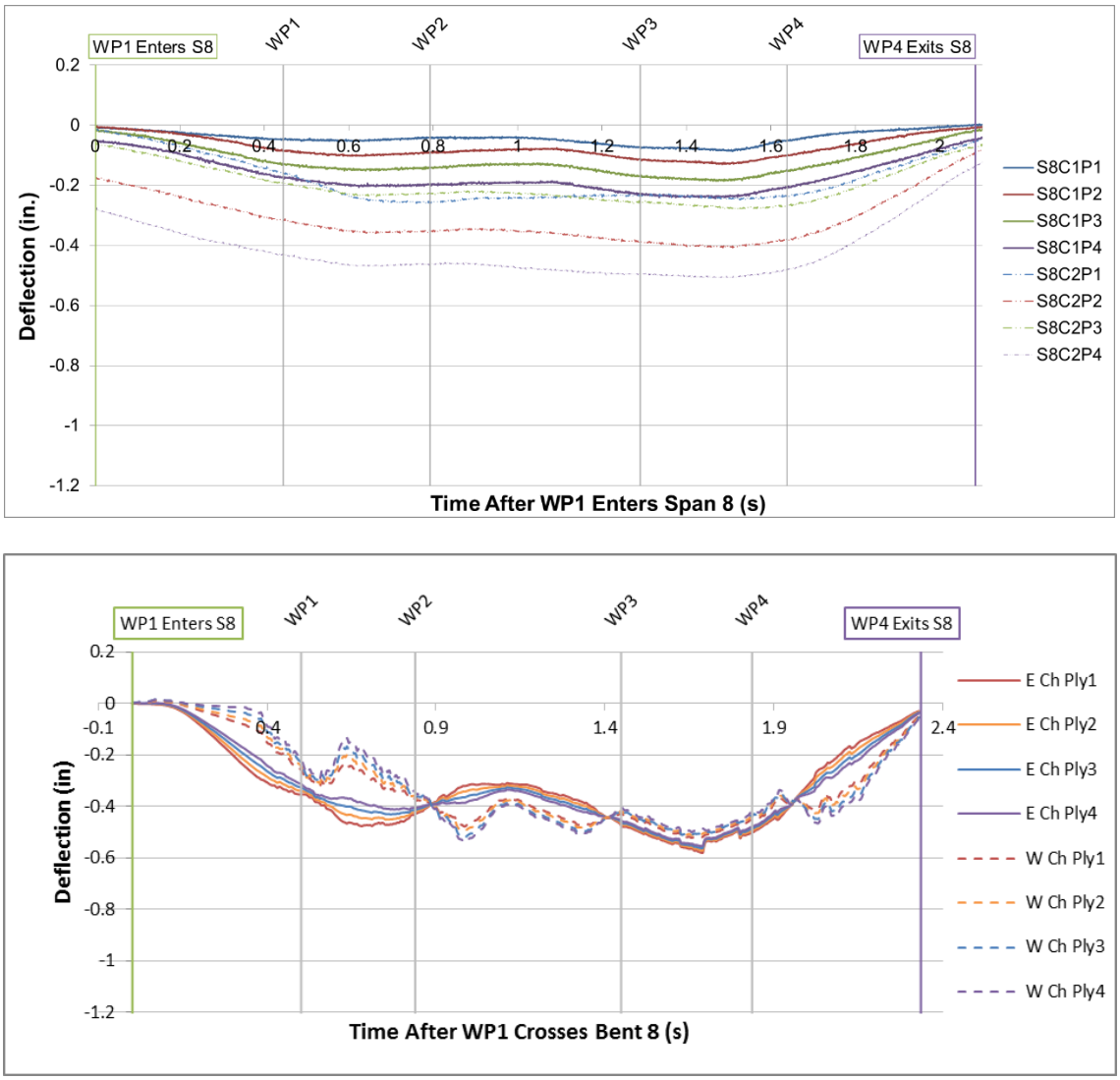

Figure A-219: Net Stringer Time History Comparison Test 1 Span 8: Experimental (Top), Predicted (Bottom)

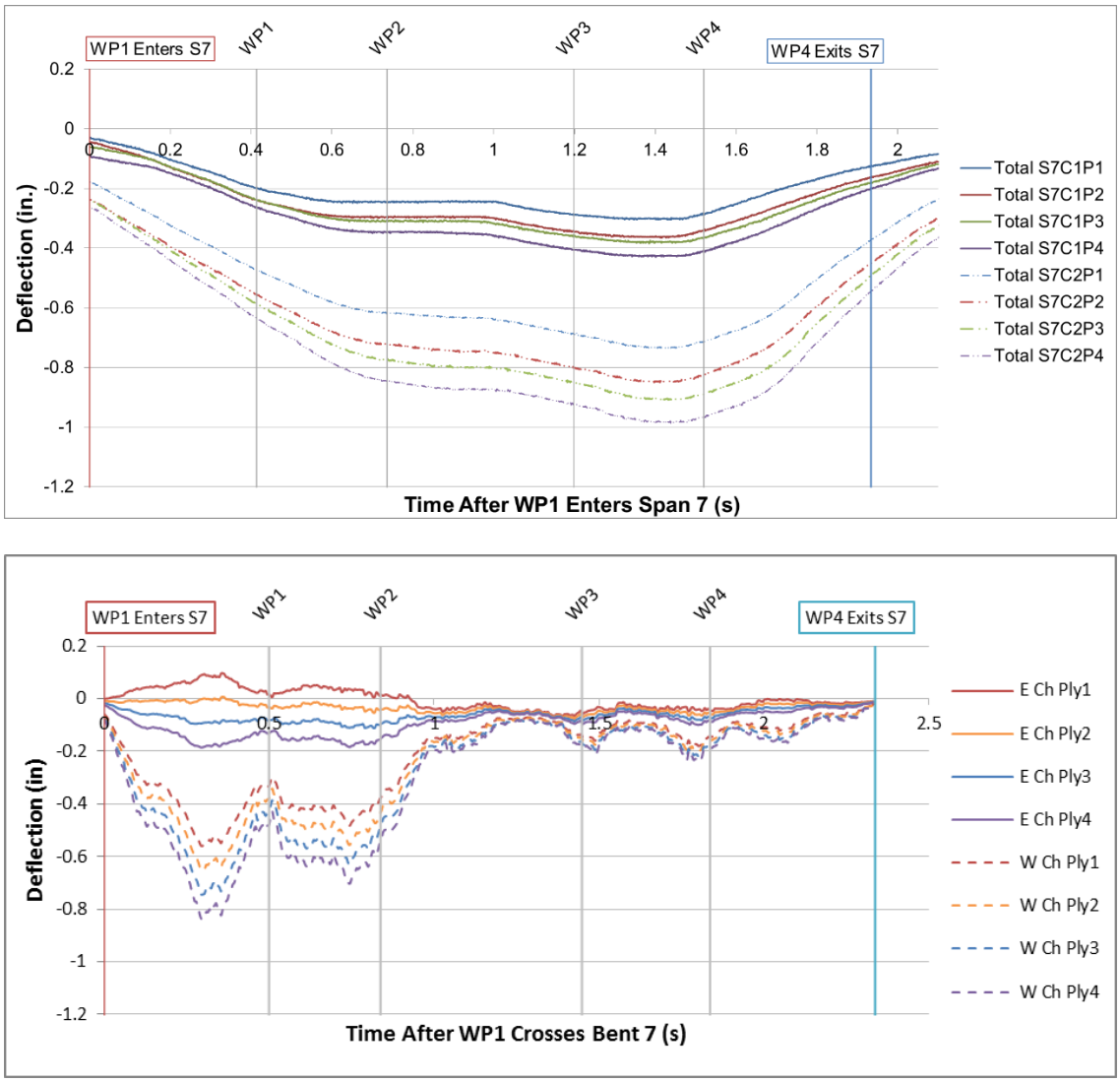

Figure A-220: Stringer Time History Comparison Test 2 Span 7: Experimental (Top), Predicted (Bottom)

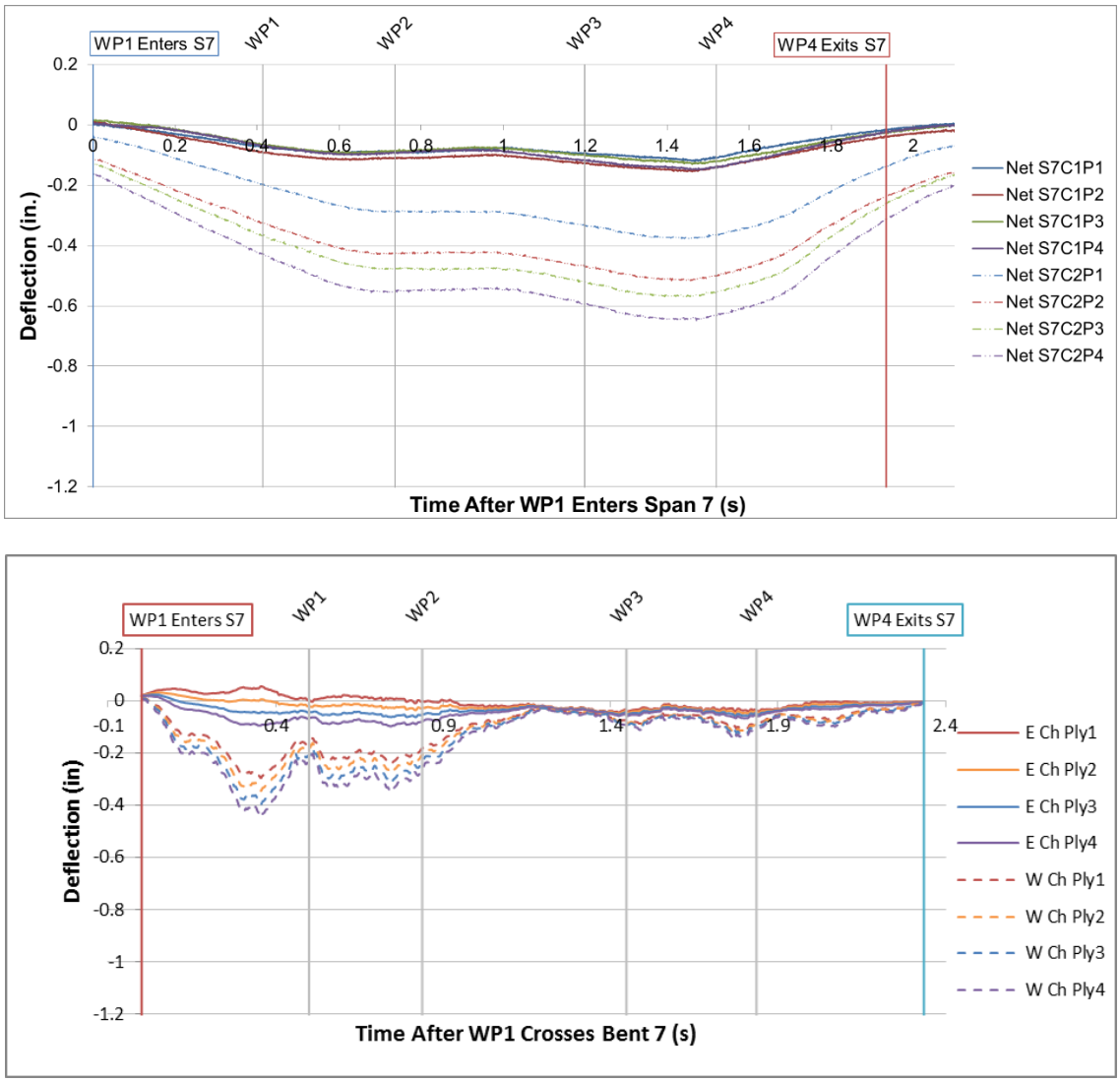

Figure A-221: Net Stringer Time History Comparison Test 2 Span 7: Experimental (Top), Predicted (Bottom)

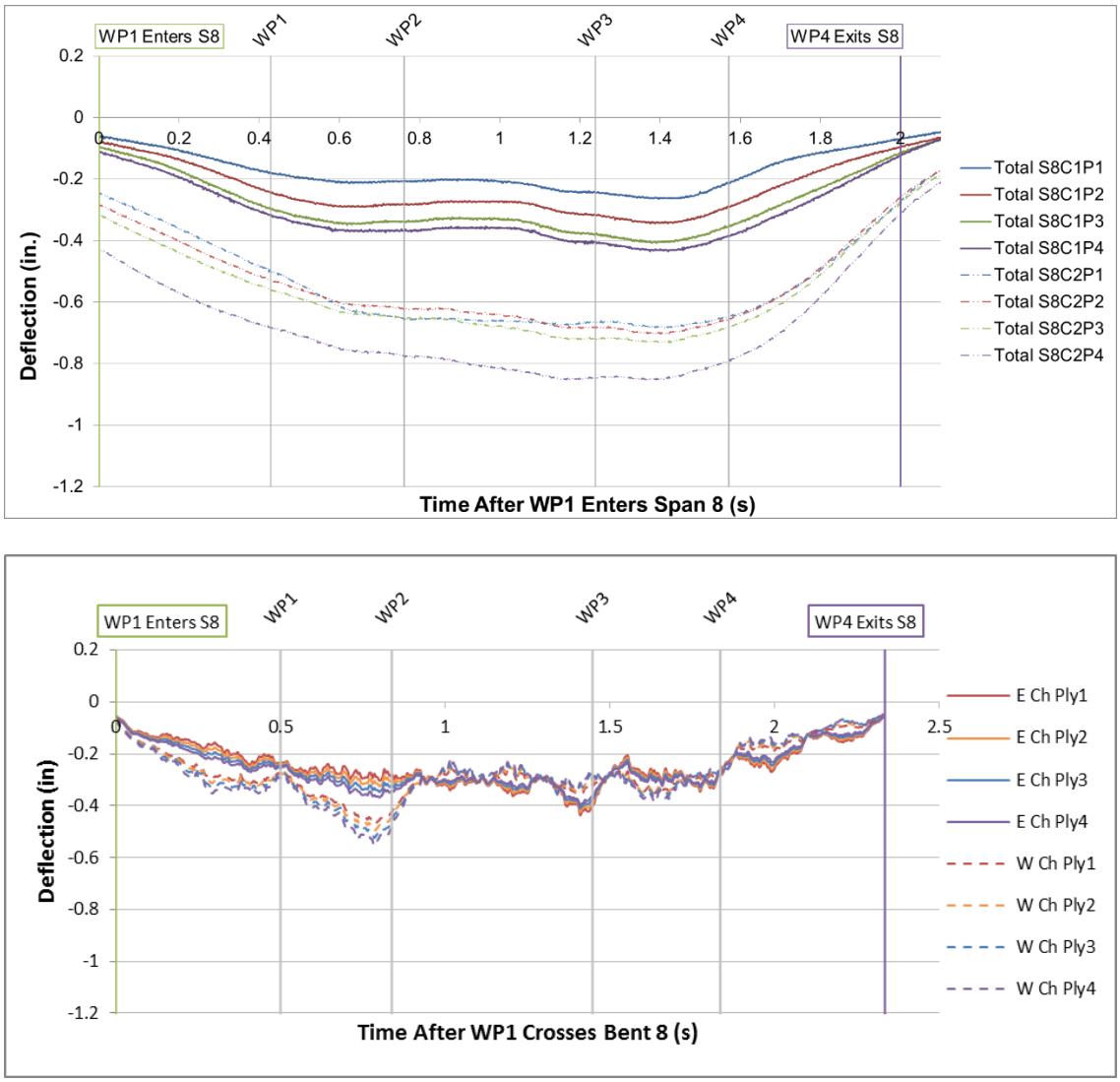

Figure A-222: Stringer Time History Comparison Test 2 Span 8: Experimental (Top), Predicted (Bottom)

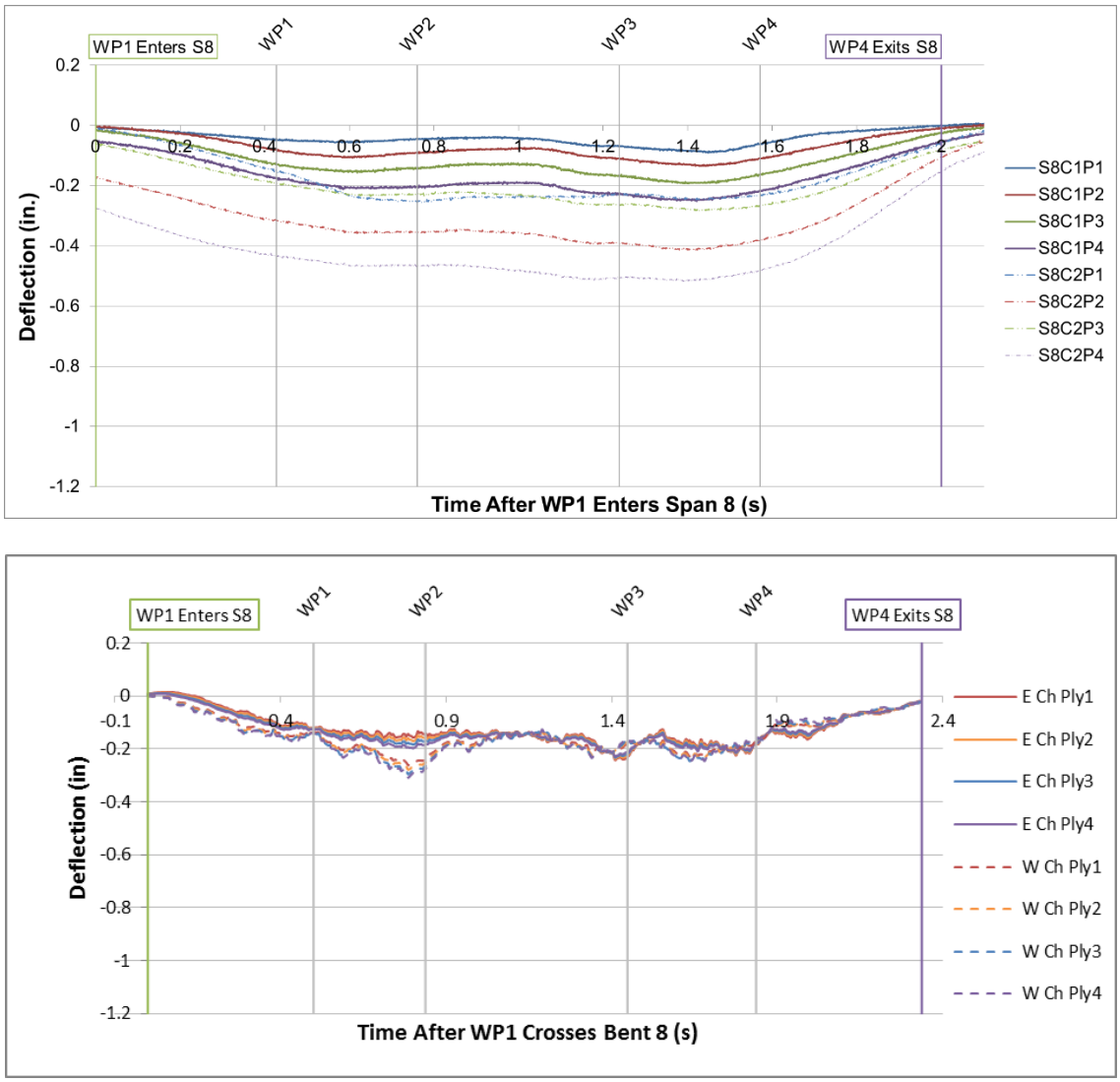

Figure A-223: Net Stringer Time History Comparison Test 2 Span 8: Experimental (Top), Predicted (Bottom)

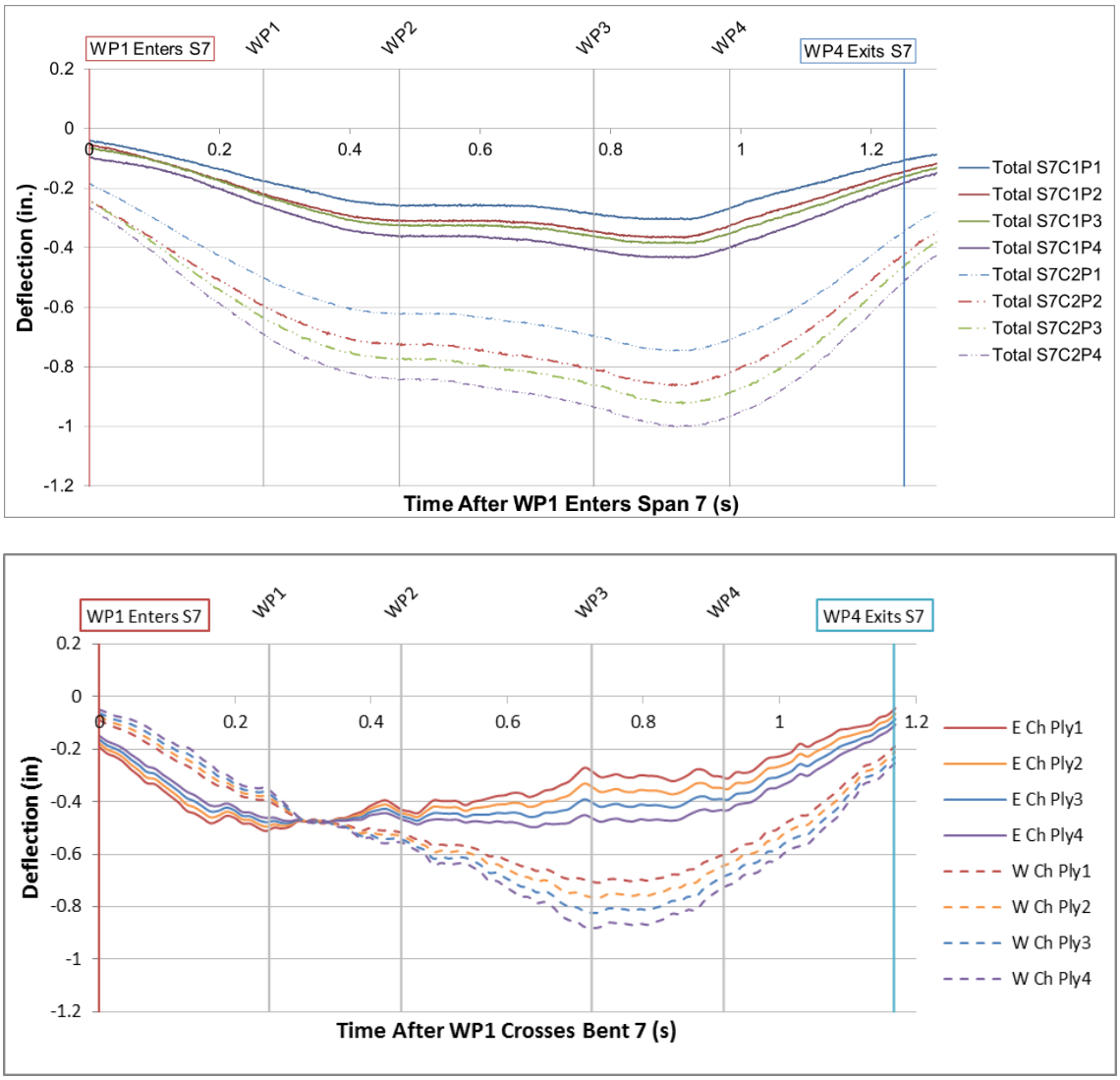

Figure A-224: Stringer Time History Comparison Test 3 Span 7: Experimental (Top), Predicted (Bottom)

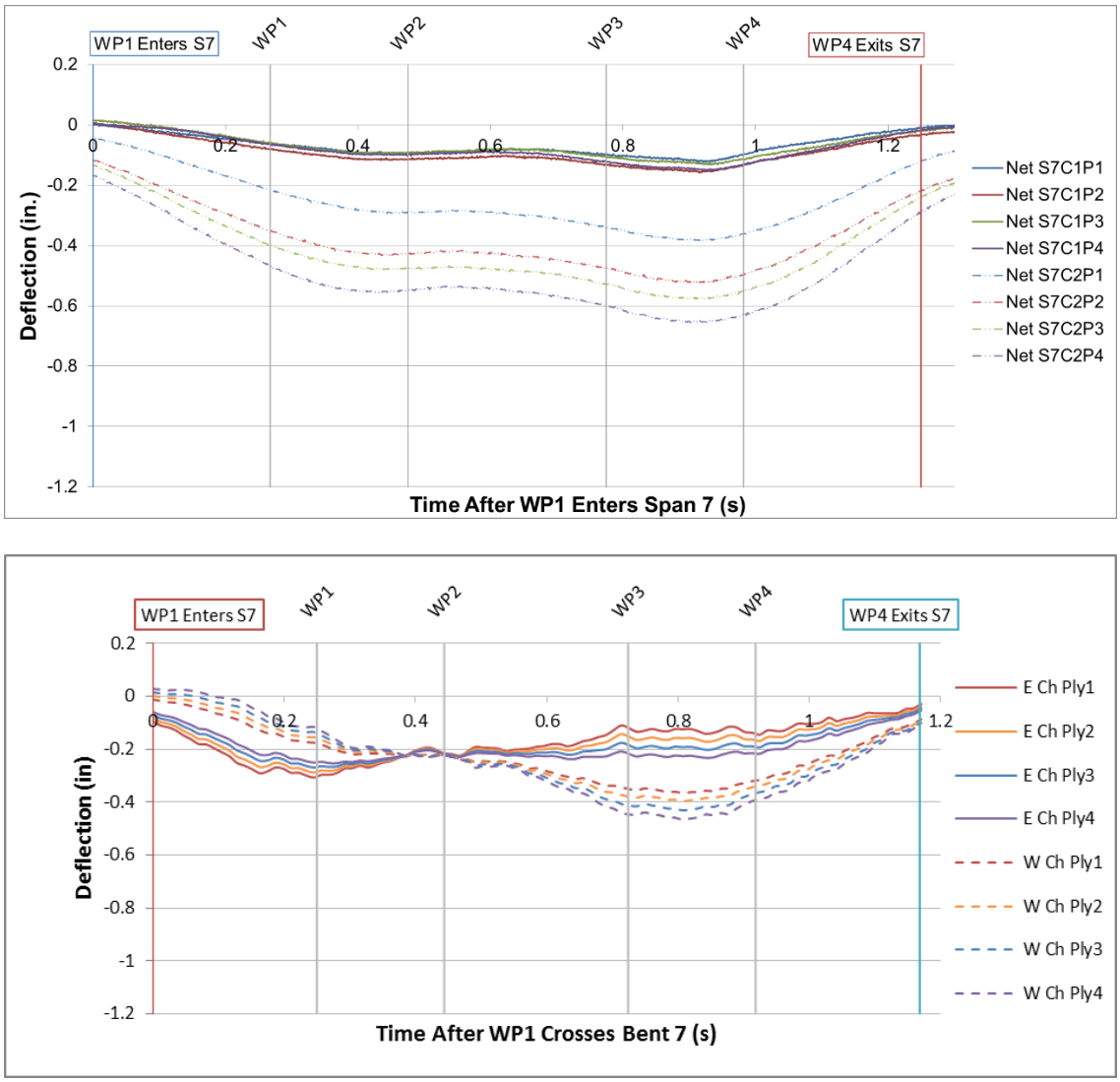

Figure A-225: Net Stringer Time History Comparison Test 3 Span 7: Experimental (Top), Predicted (Bottom)

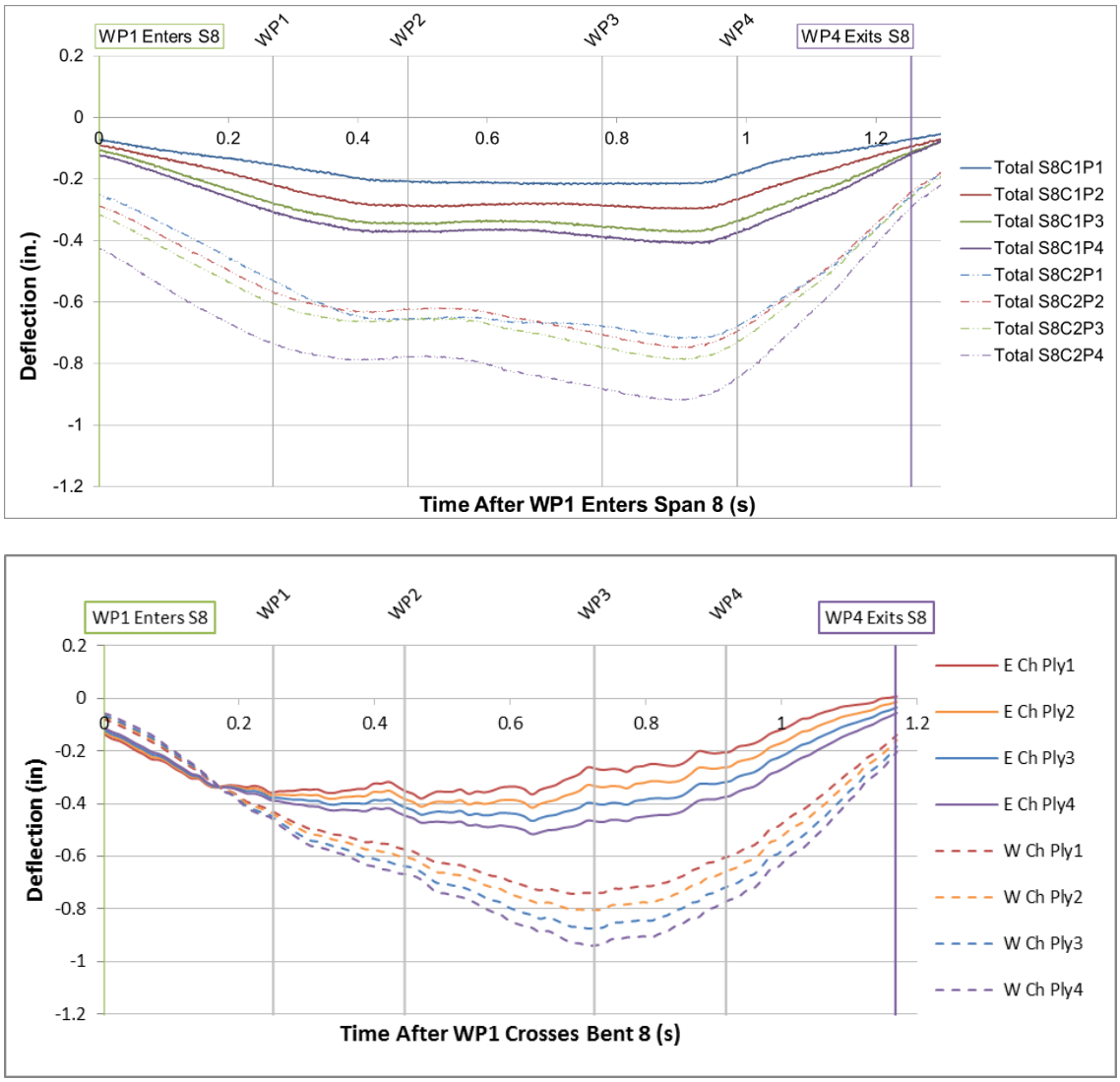

Figure A-226: Stringer Time History Comparison Test 3 Span 8: Experimental (Top), Predicted (Bottom)

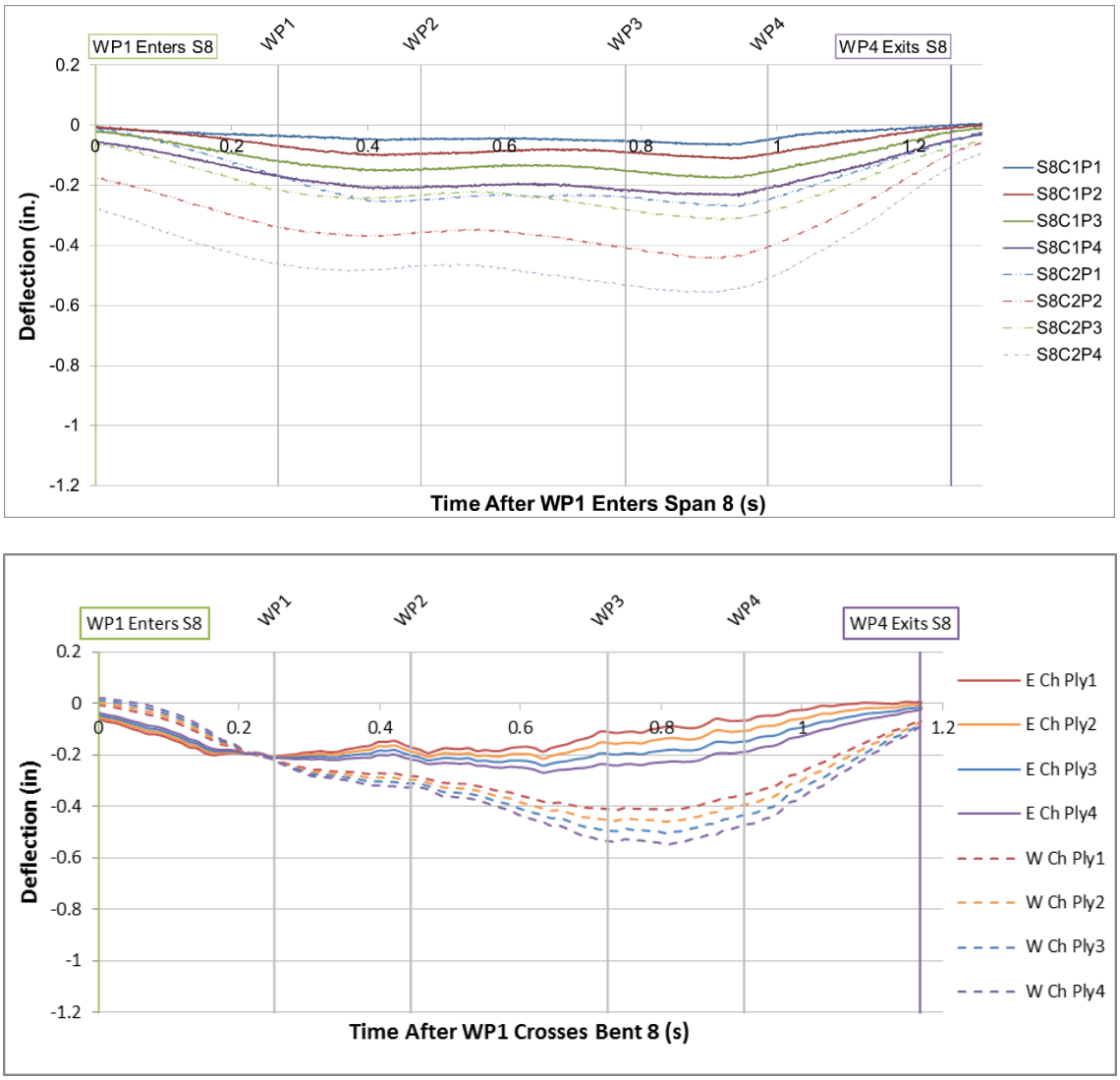

Figure A-227: Net Stringer Time History Comparison Test 3 Span 8: Experimental (Top), Predicted (Bottom)

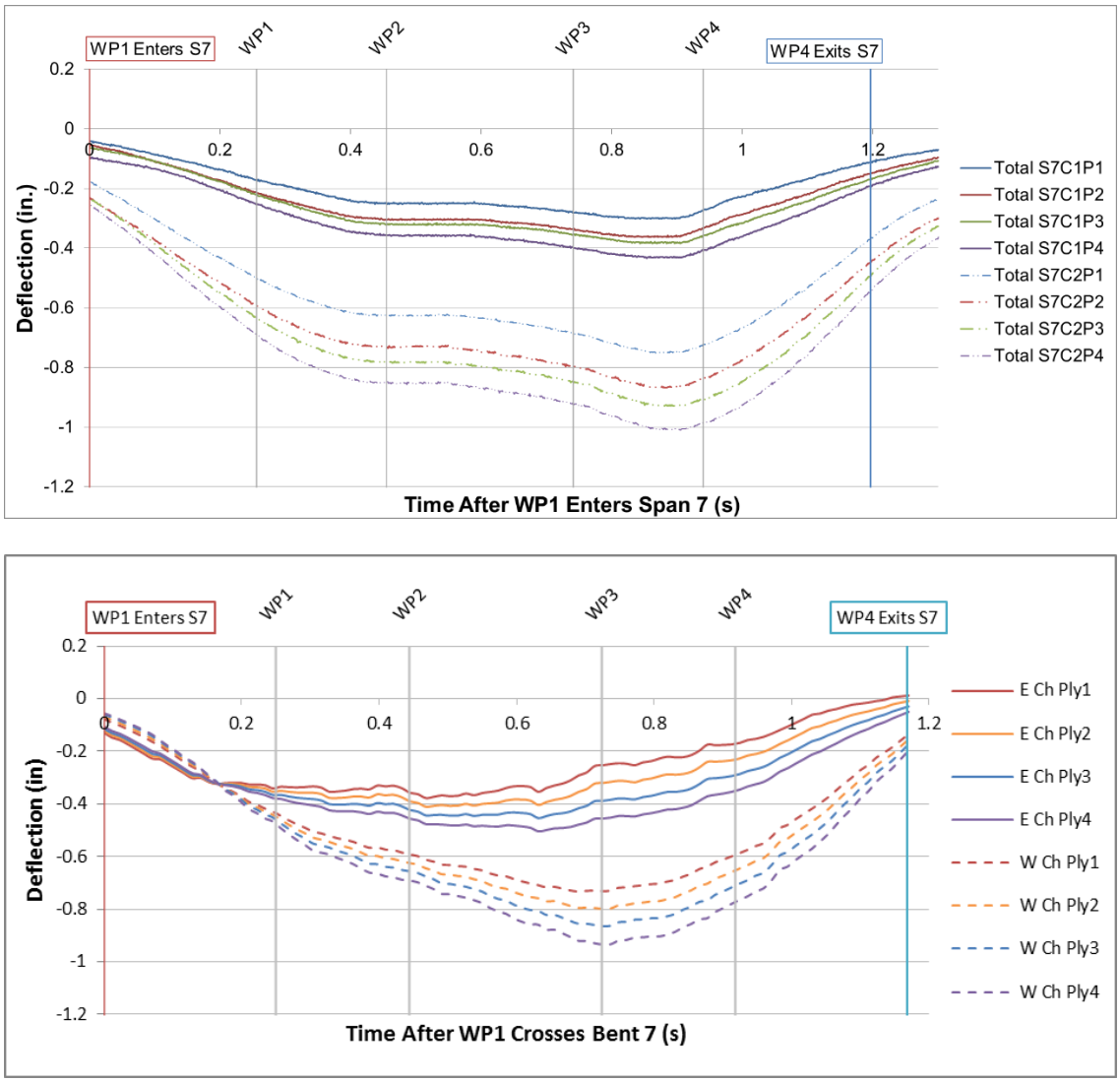

Figure A-228: Stringer Time History Comparison Test 4 Span 7: Experimental (Top), Predicted (Bottom)

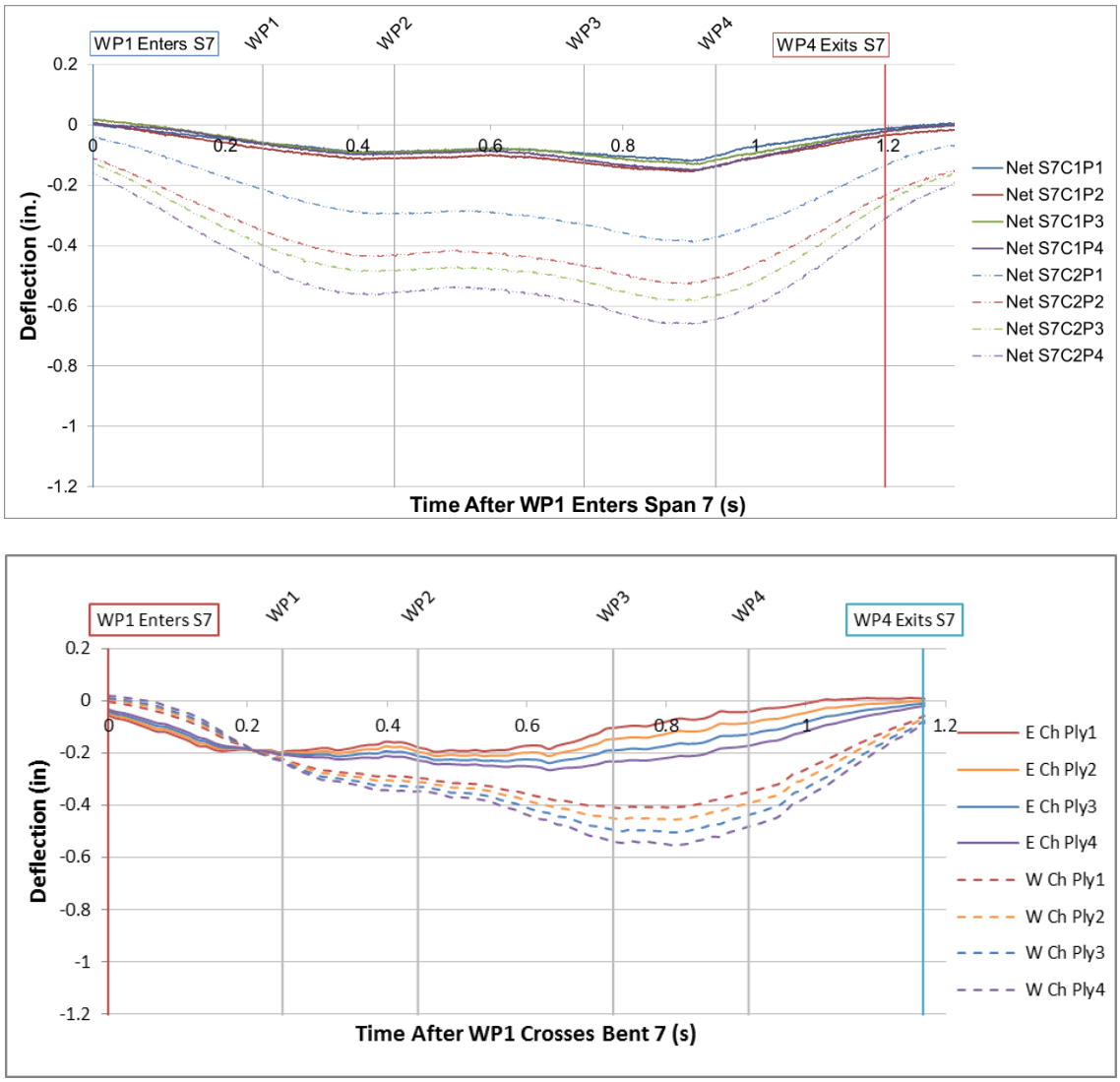

Figure A-229: Net Stringer Time History Comparison Test 4 Span 7: Experimental (Top), Predicted (Bottom)

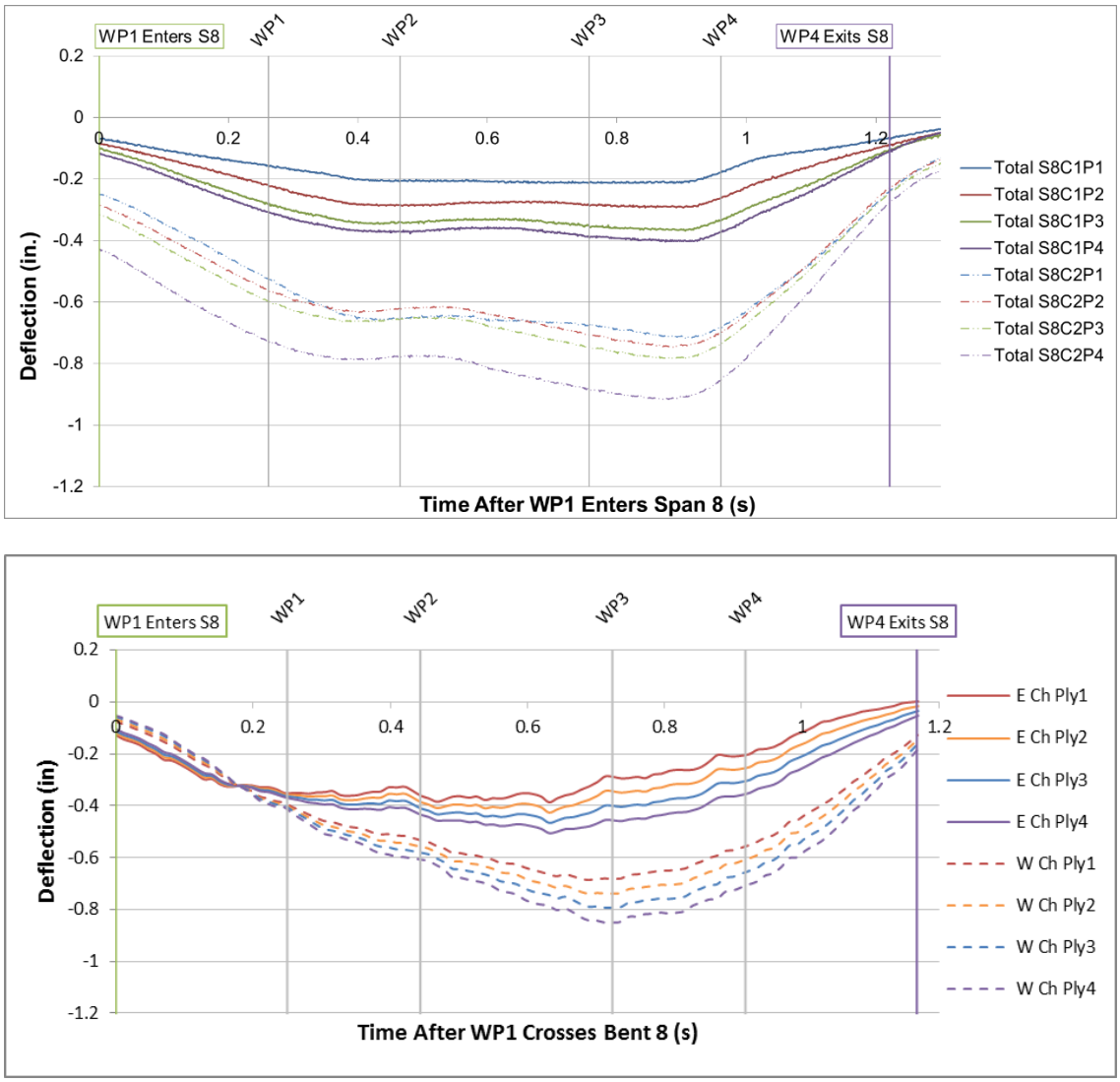

Figure A-230: Stringer Time History Comparison Test 4 Span 8: Experimental (Top), Predicted (Bottom)

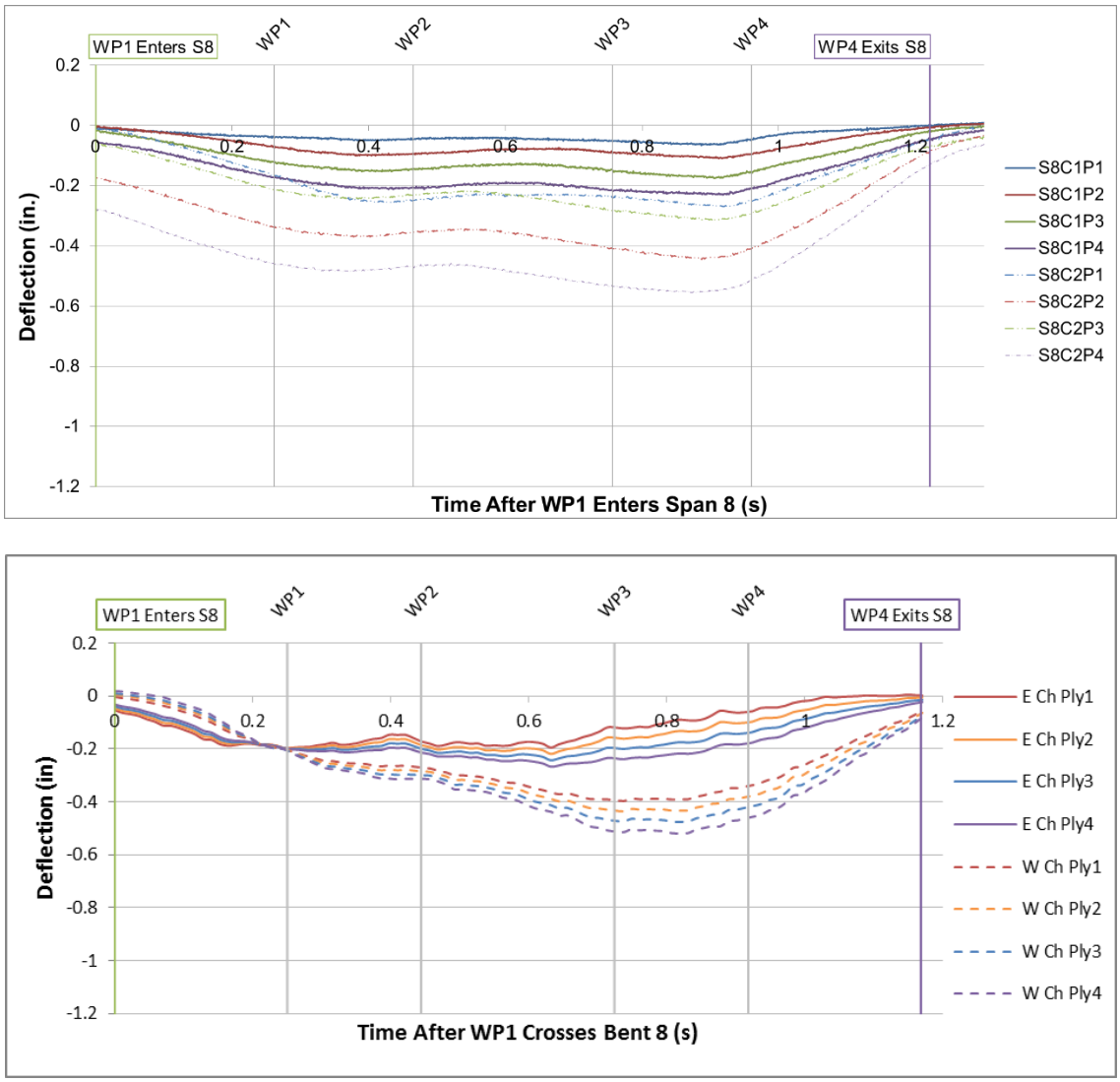

Figure A-231: Net Stringer Time History Comparison Test 4 Span 8: Experimental (Top), Predicted (Bottom)

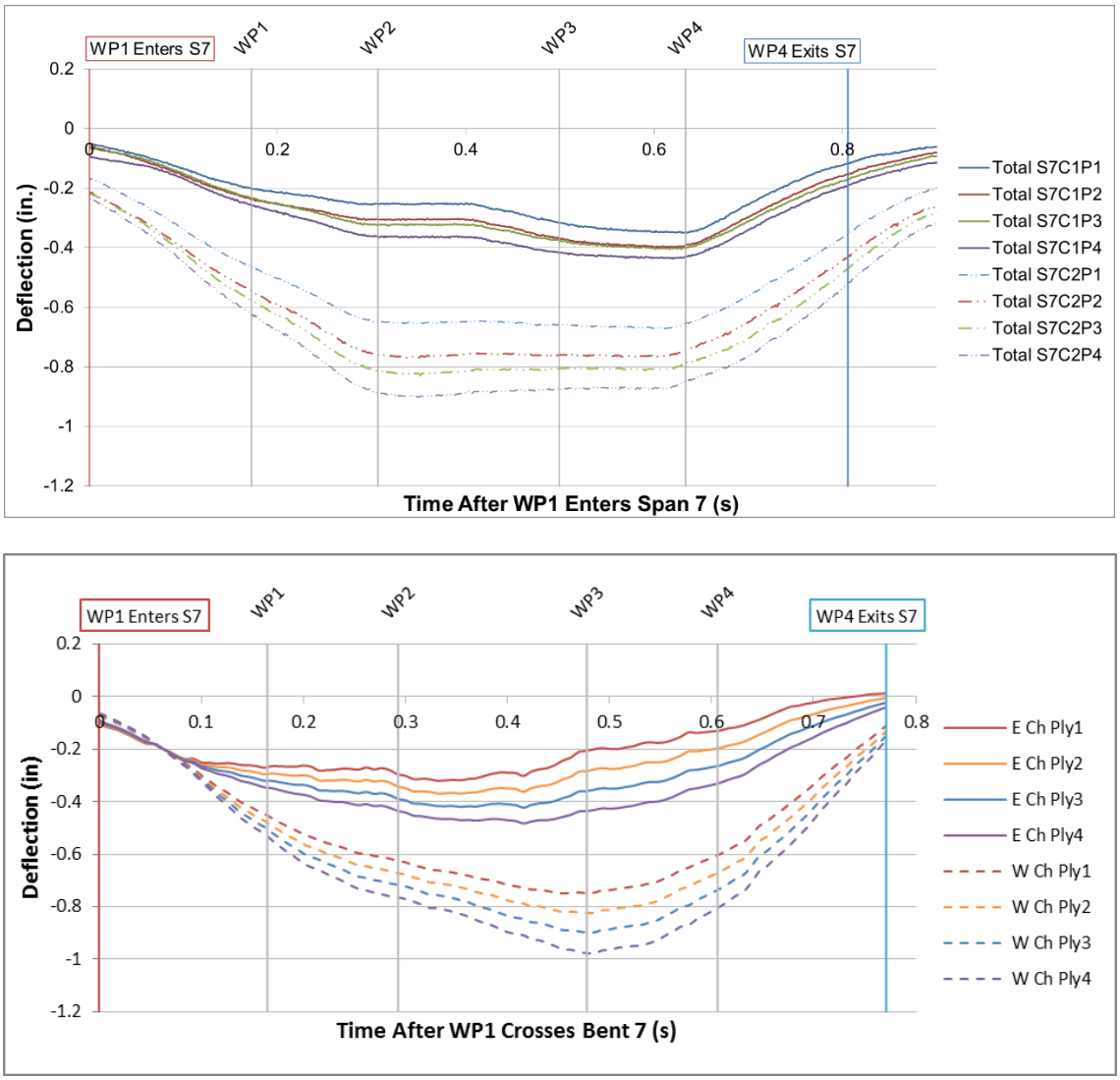

Figure A-232: Stringer Time History Comparison Test 5 Span 7: Experimental (Top), Predicted (Bottom)

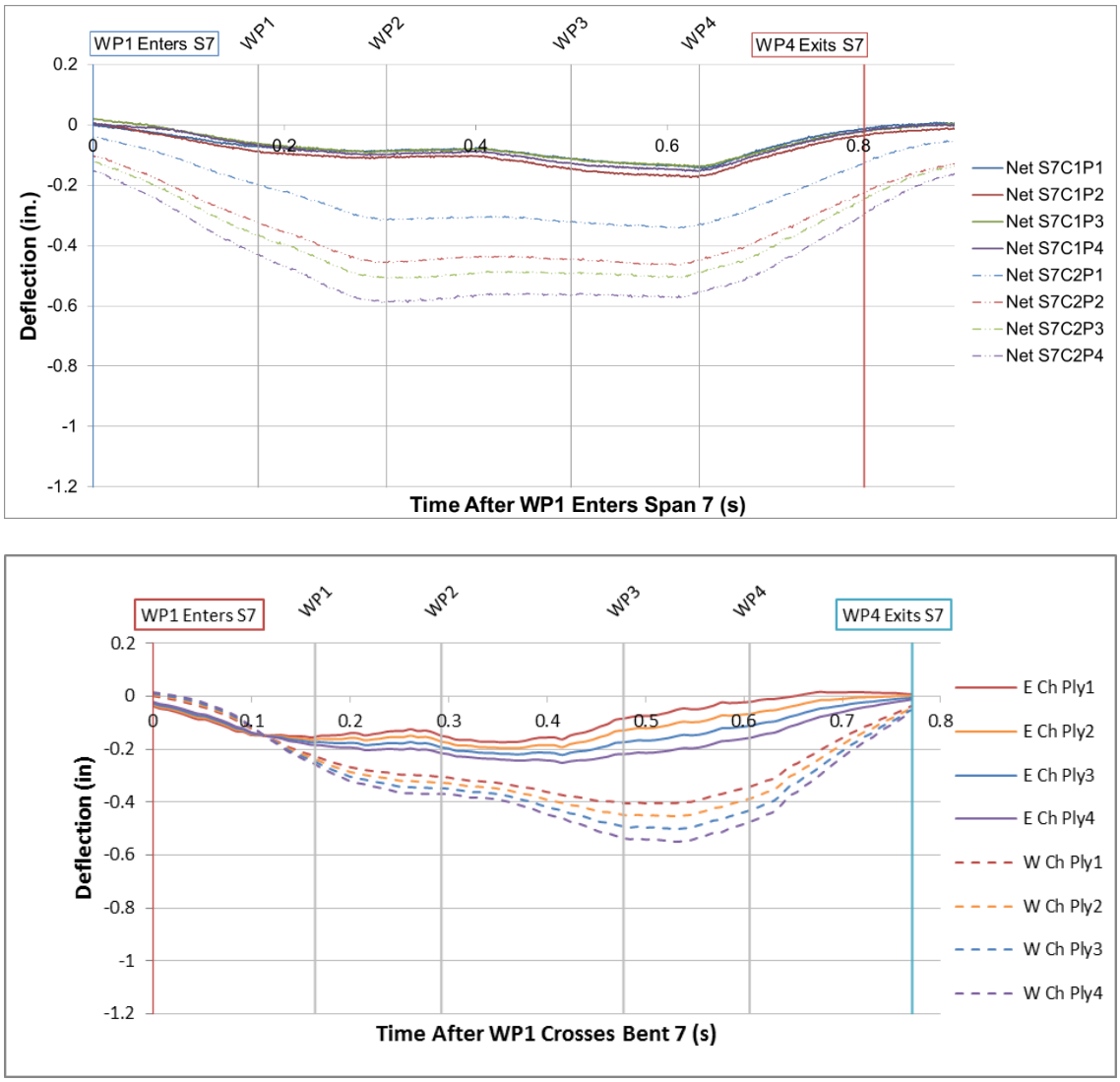

Figure A-233: Net Stringer Time History Comparison Test 5 Span 7: Experimental (Top), Predicted (Bottom)

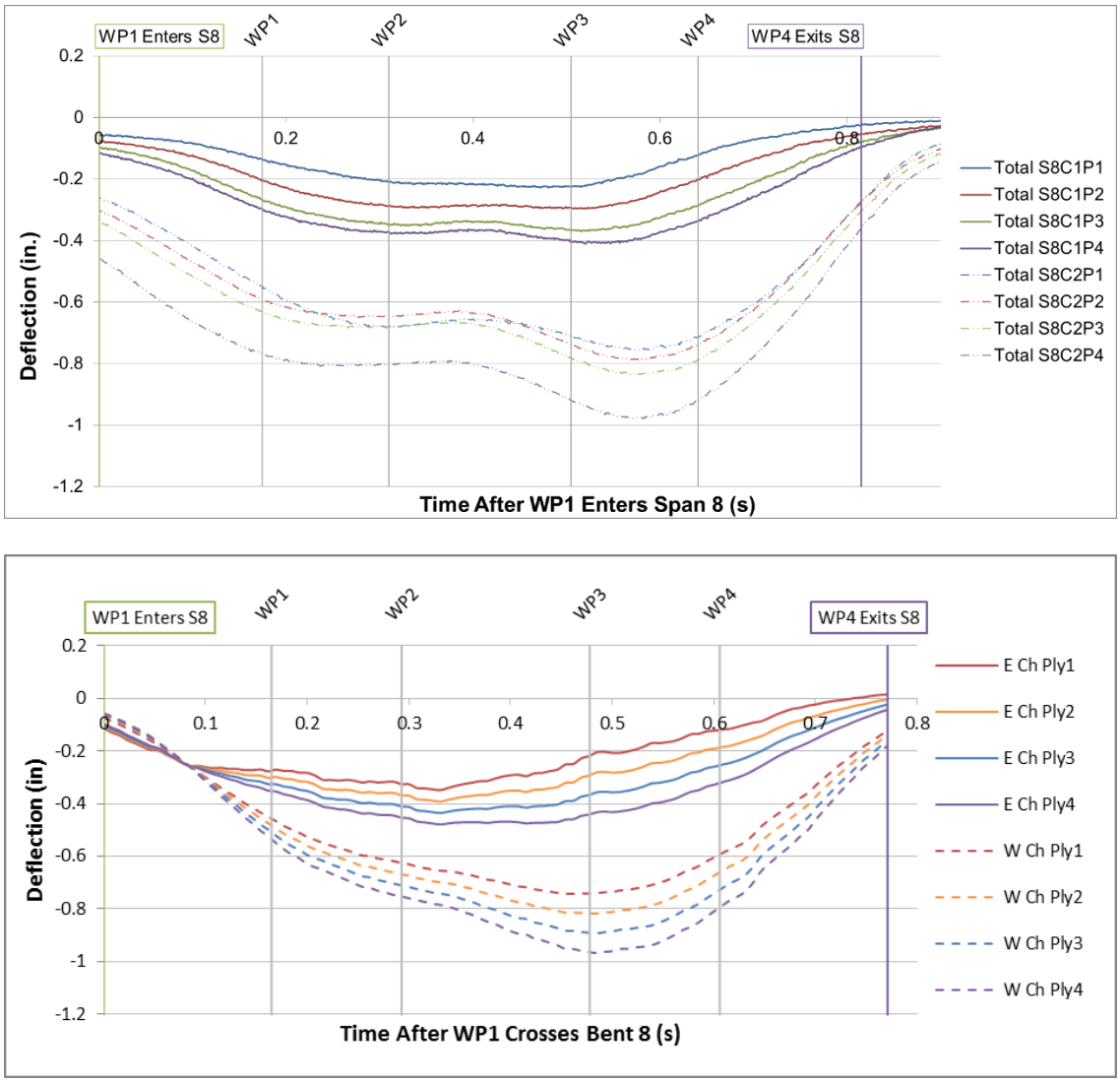

Figure A-234: Stringer Time History Comparison Test 5 Span 8: Experimental (Top), Predicted (Bottom)

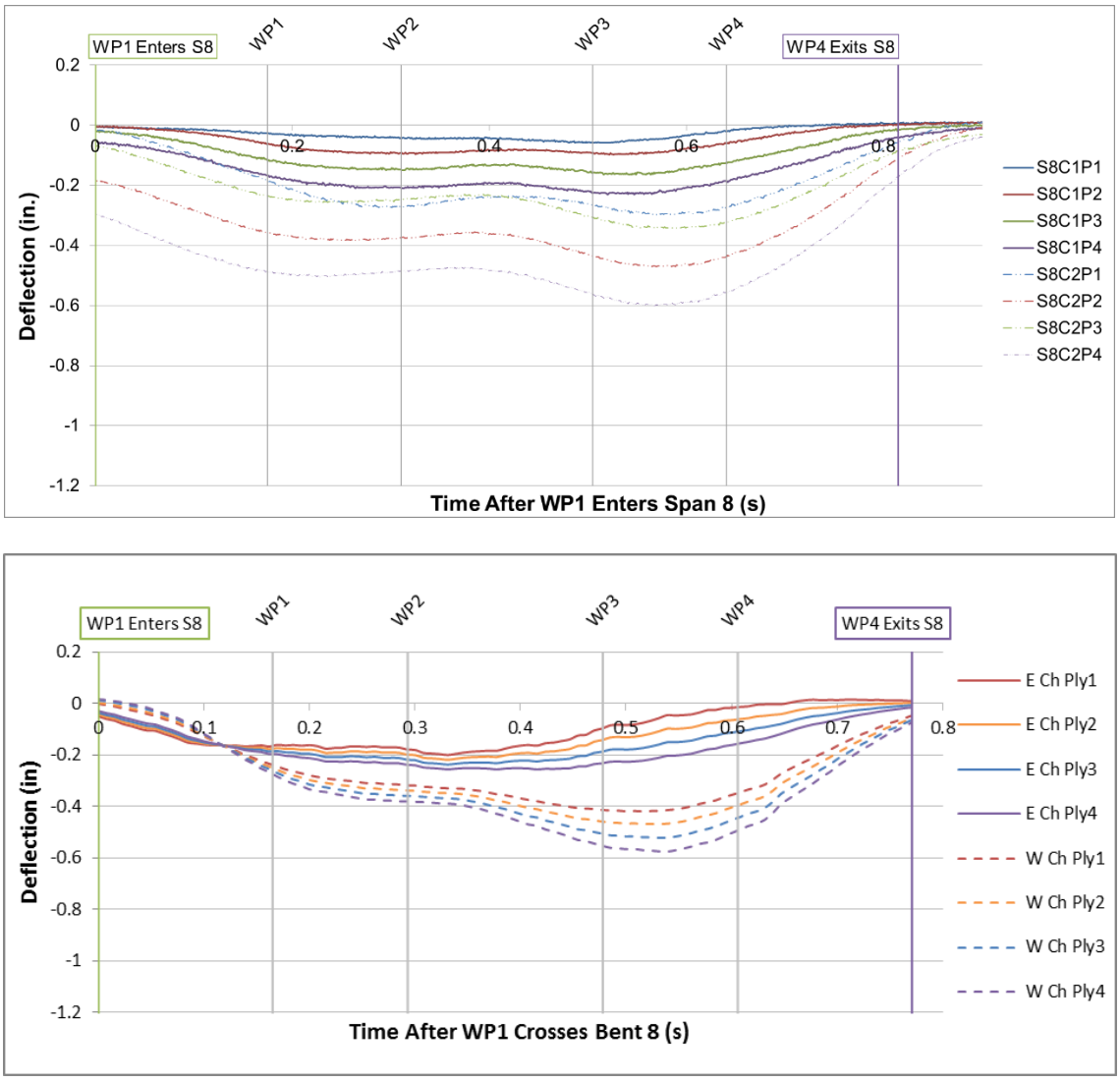

Figure A-235: Net Stringer Time History Comparison Test 5 Span 8: Experimental (Top), Predicted (Bottom)

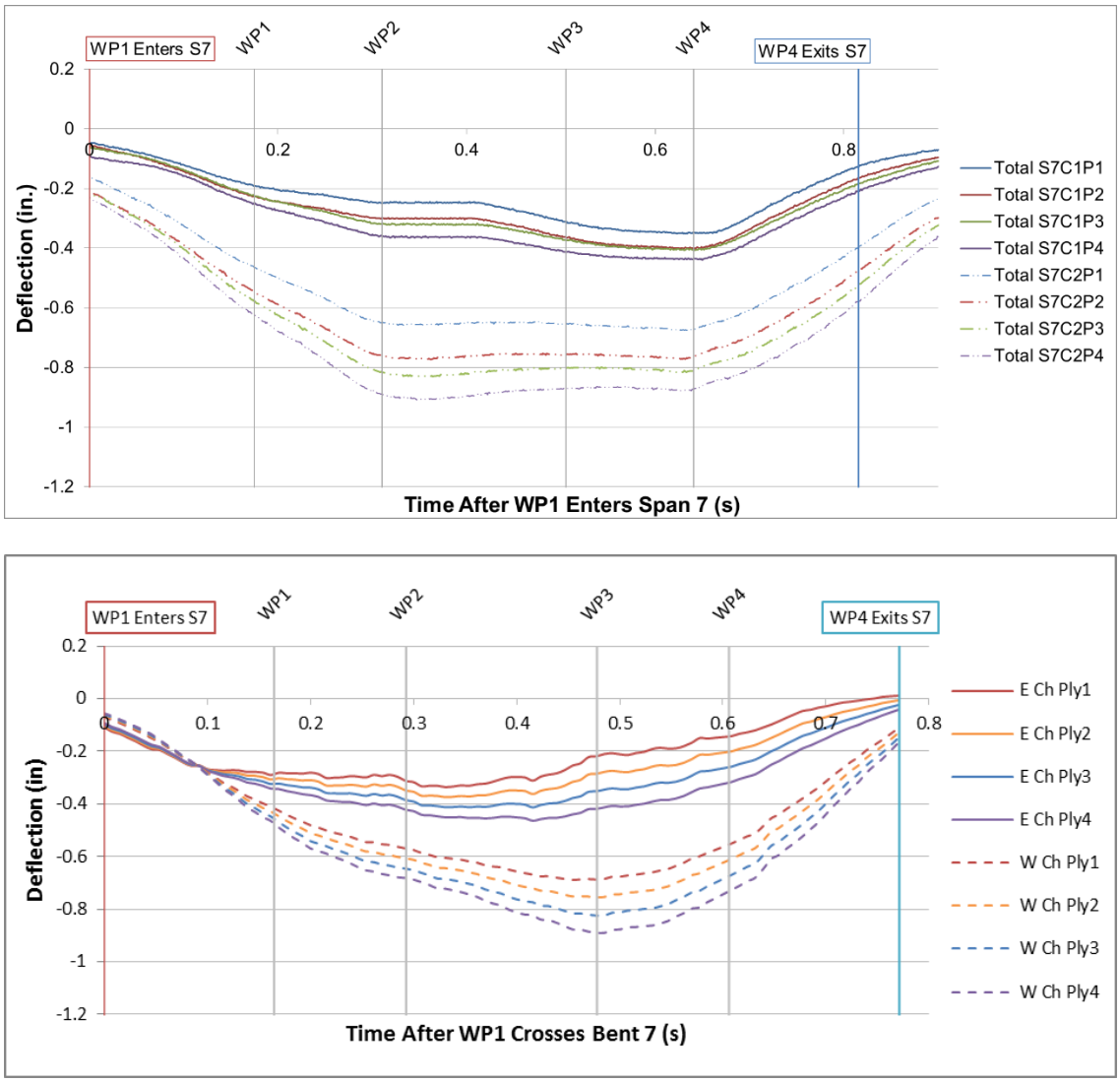

Figure A-236: Stringer Time History Comparison Test 6 Span 7: Experimental (Top), Predicted (Bottom)

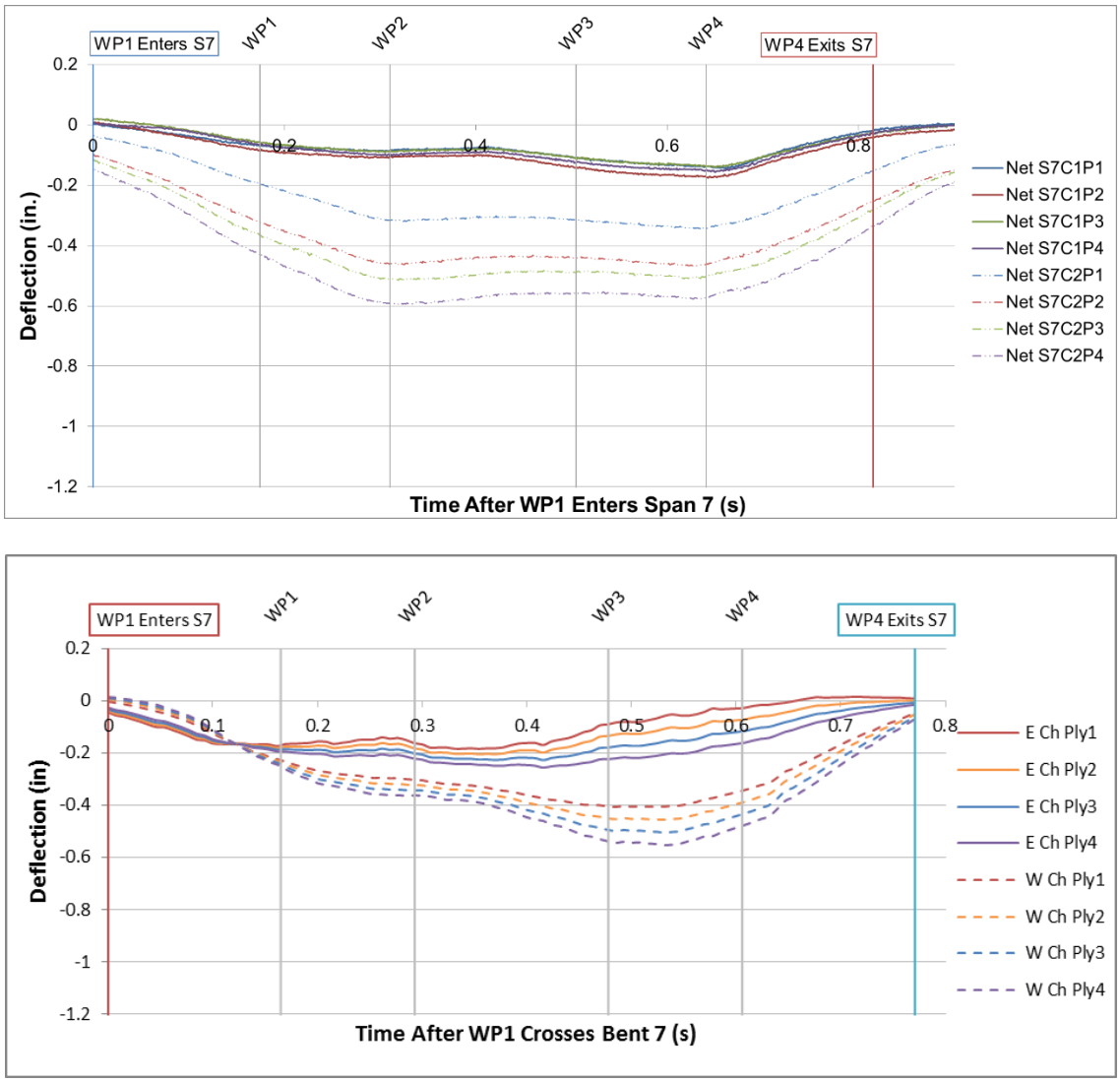

Figure A-237: Net Stringer Time History Comparison Test 6 Span 7: Experimental (Top), Predicted (Bottom)

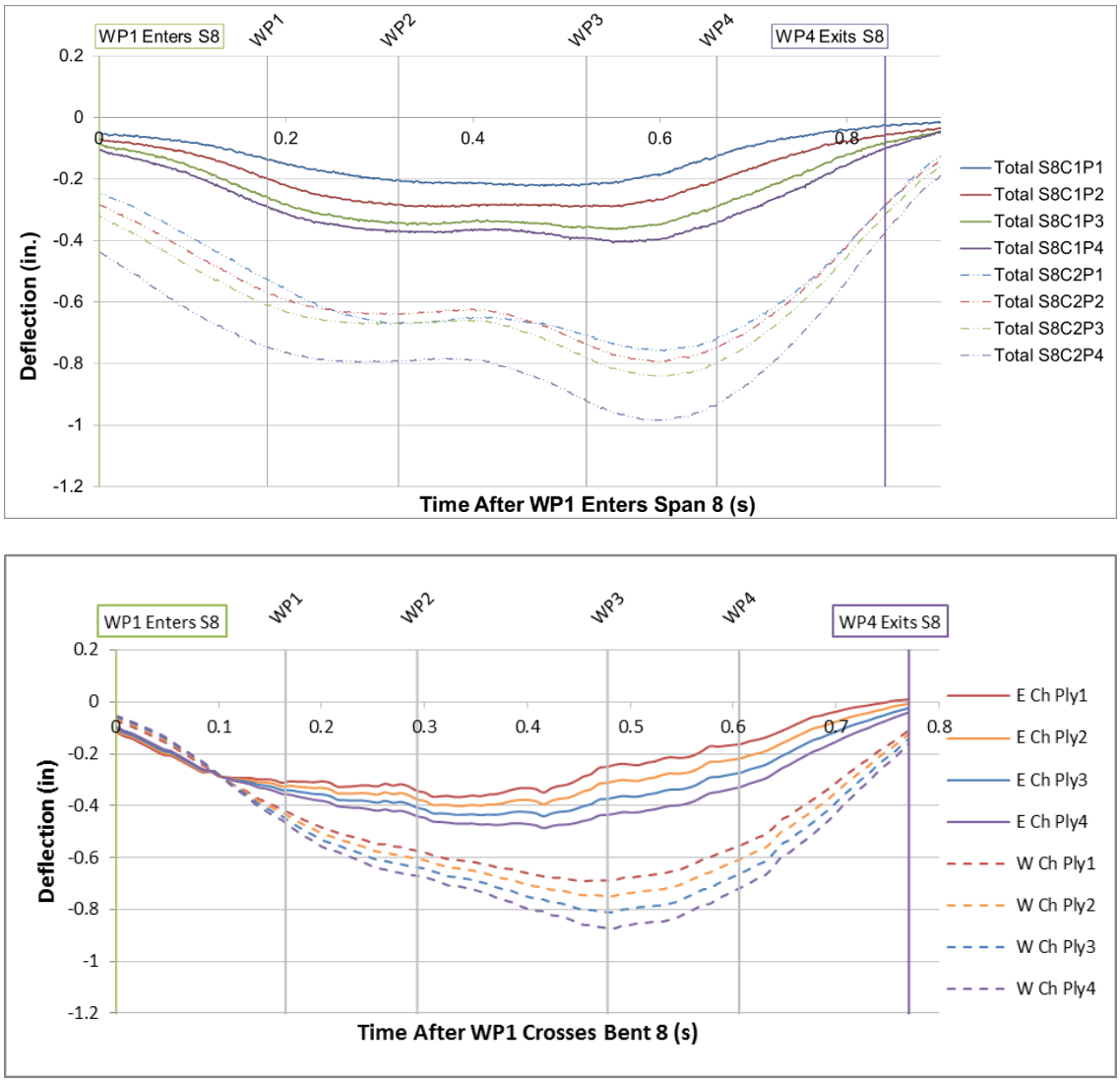

Figure A-238: Stringer Time History Comparison Test 6 Span 8: Experimental (Top), Predicted (Bottom)

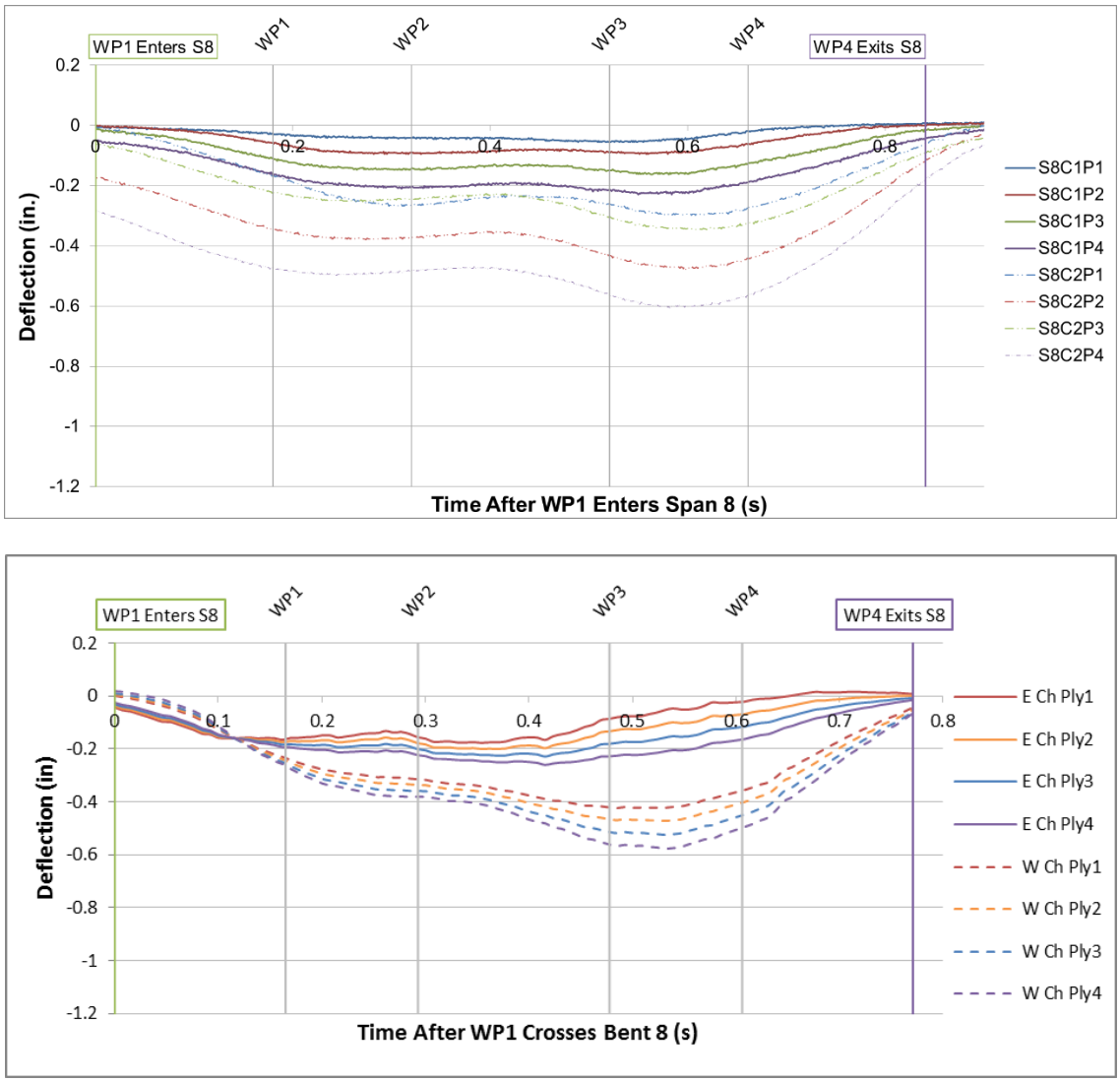

Figure A-239: Net Stringer Time History Comparison Test 6 Span 8: Experimental (Top), Predicted (Bottom)

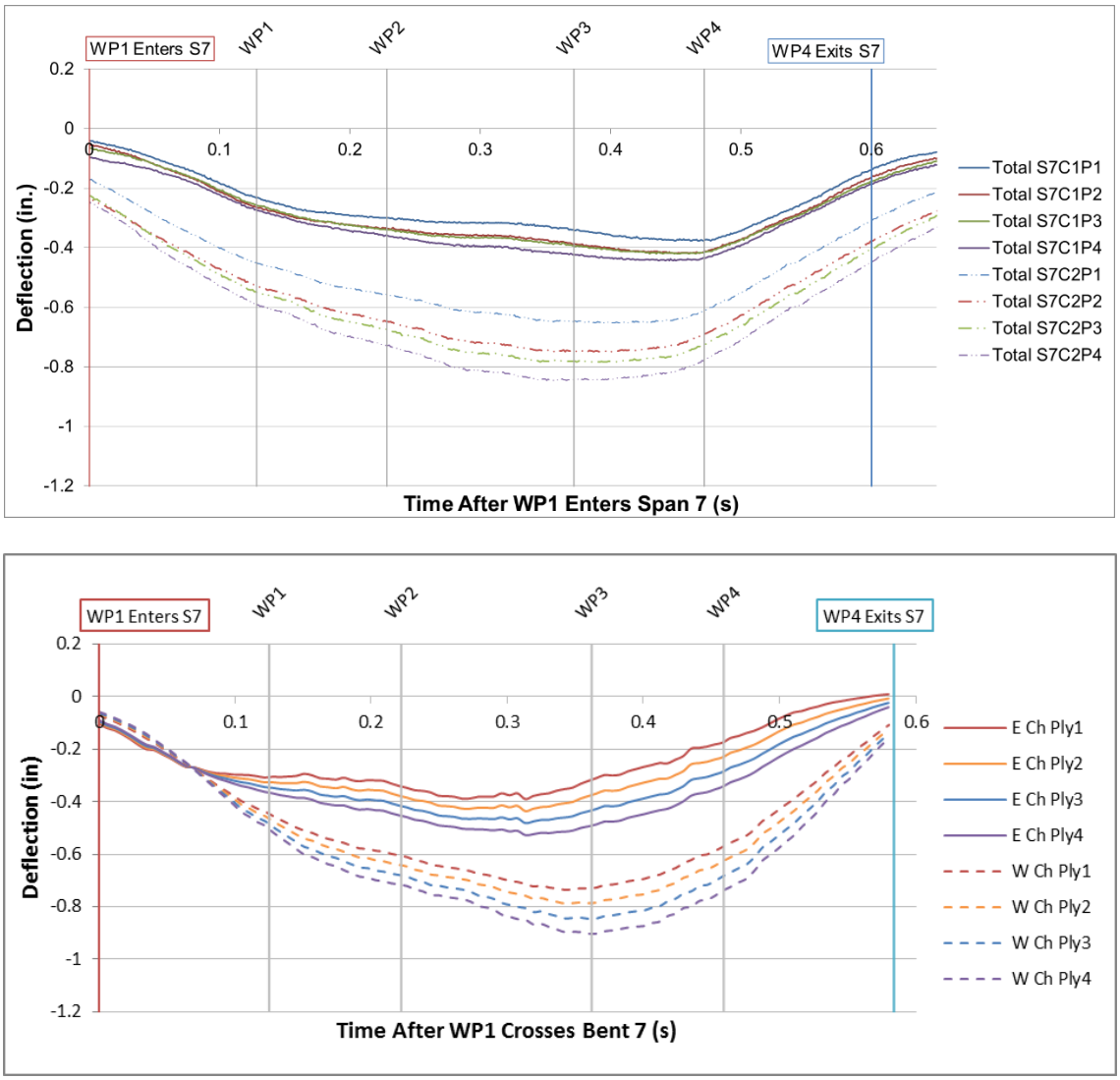

Figure A-240: Stringer Time History Comparison Test 7 Span 7: Experimental (Top), Predicted (Bottom)

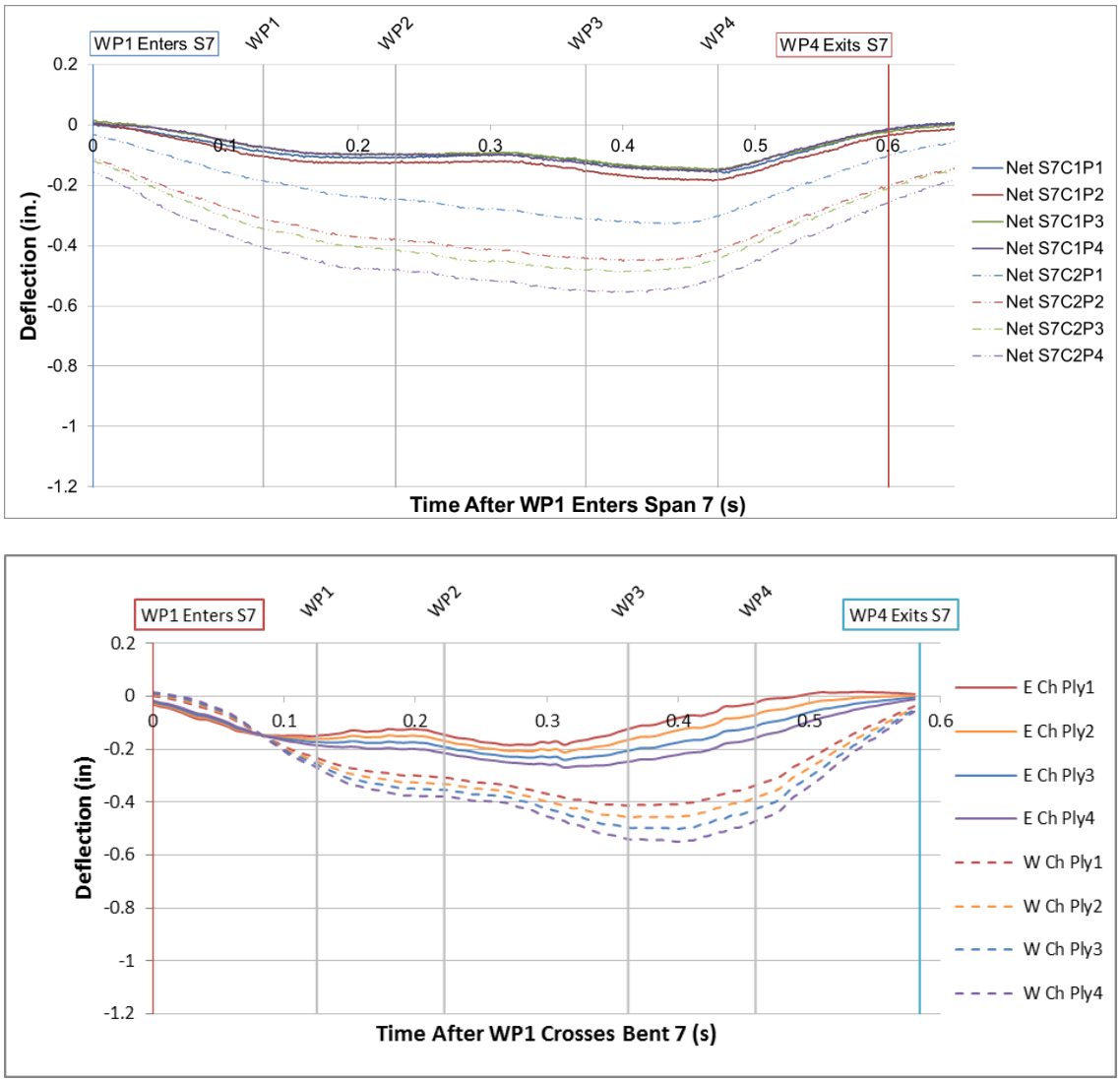

Figure A-241: Net Stringer Time History Comparison Test 7 Span 7: Experimental (Top), Predicted (Bottom)

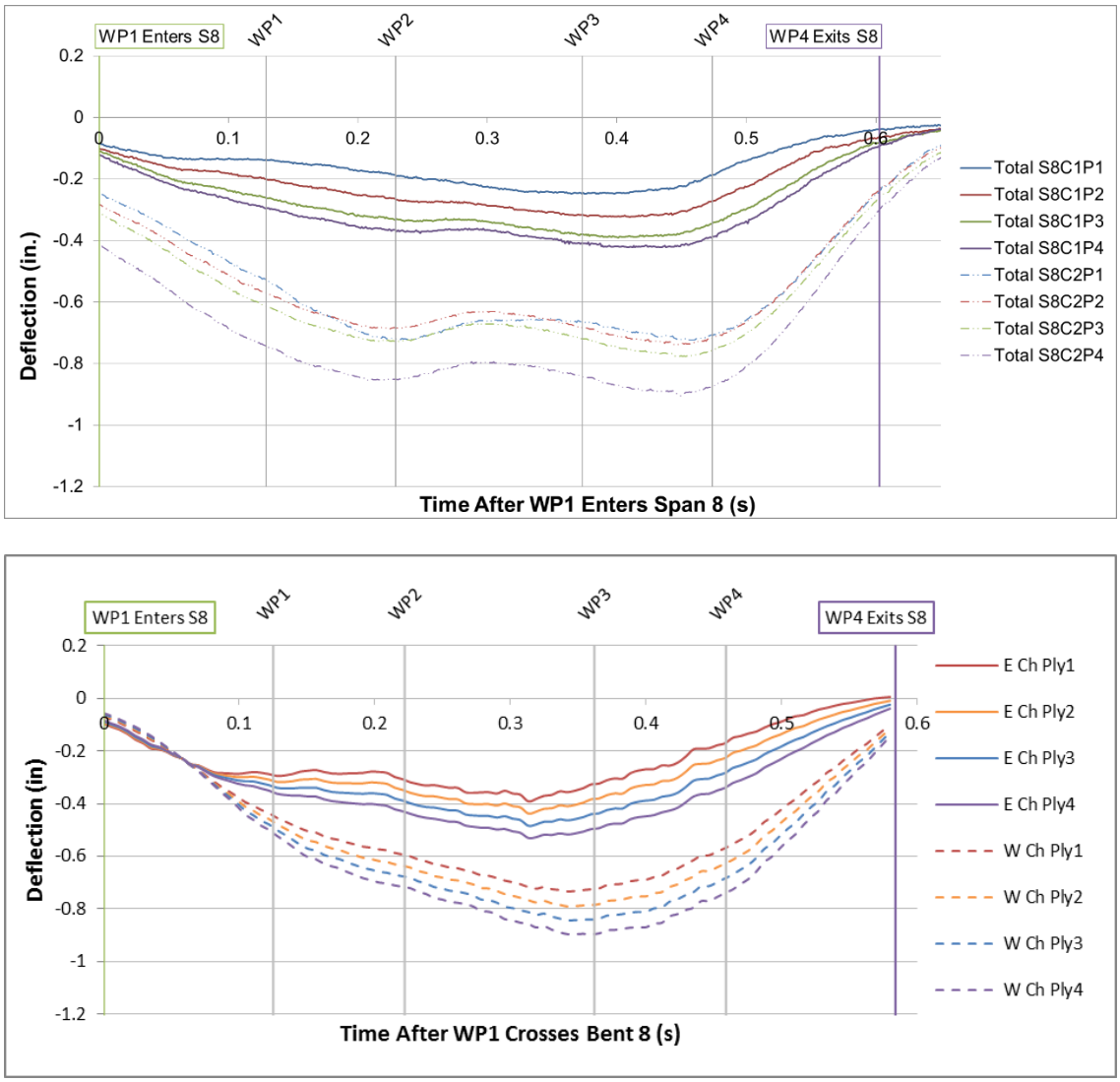

Figure A-242: Stringer Time History Comparison Test 7 Span 8: Experimental (Top), Predicted (Bottom)

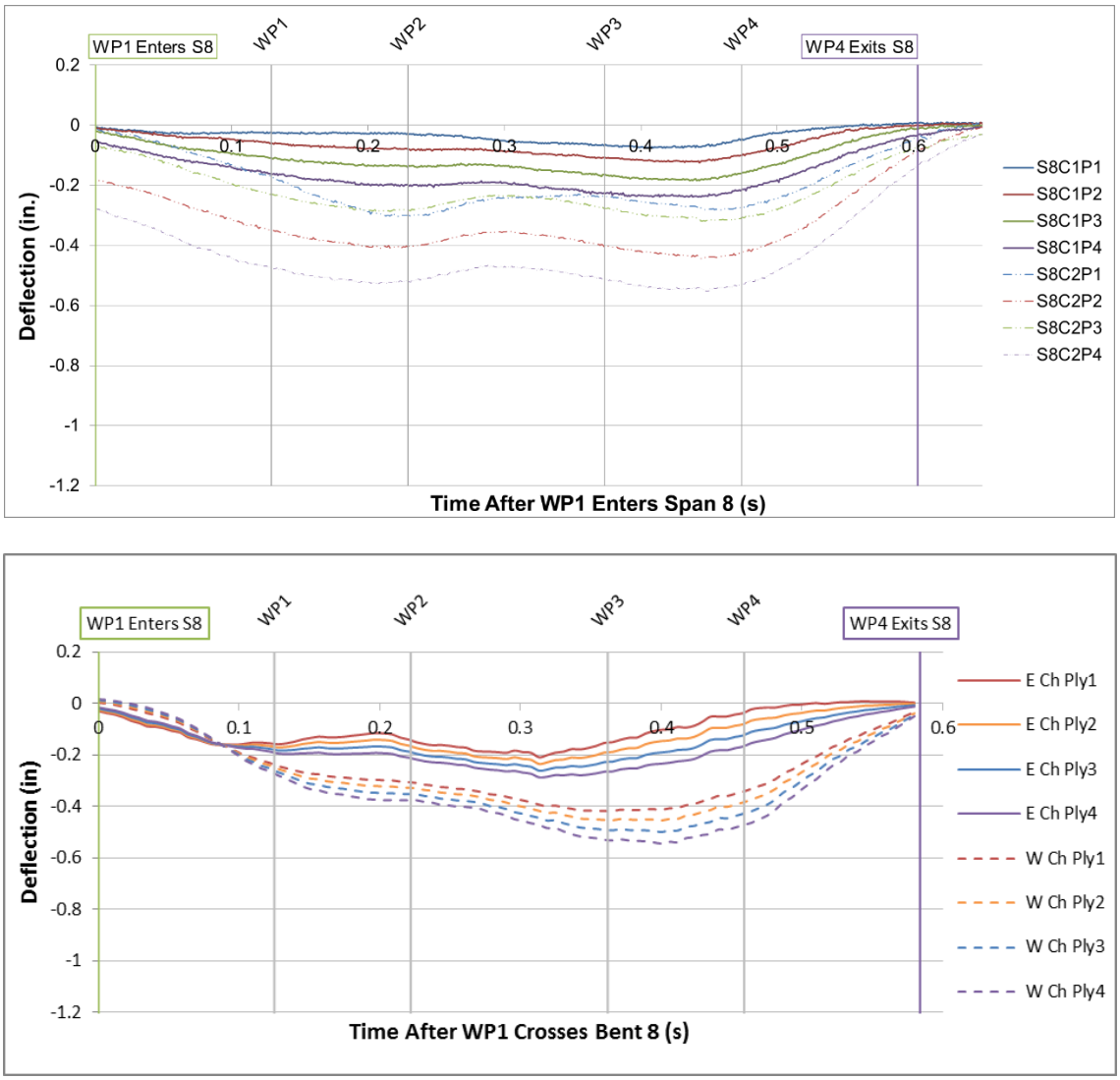

Figure A-243: Net Stringer Time History Comparison Test 7 Span 8: Experimental (Top), Predicted (Bottom)

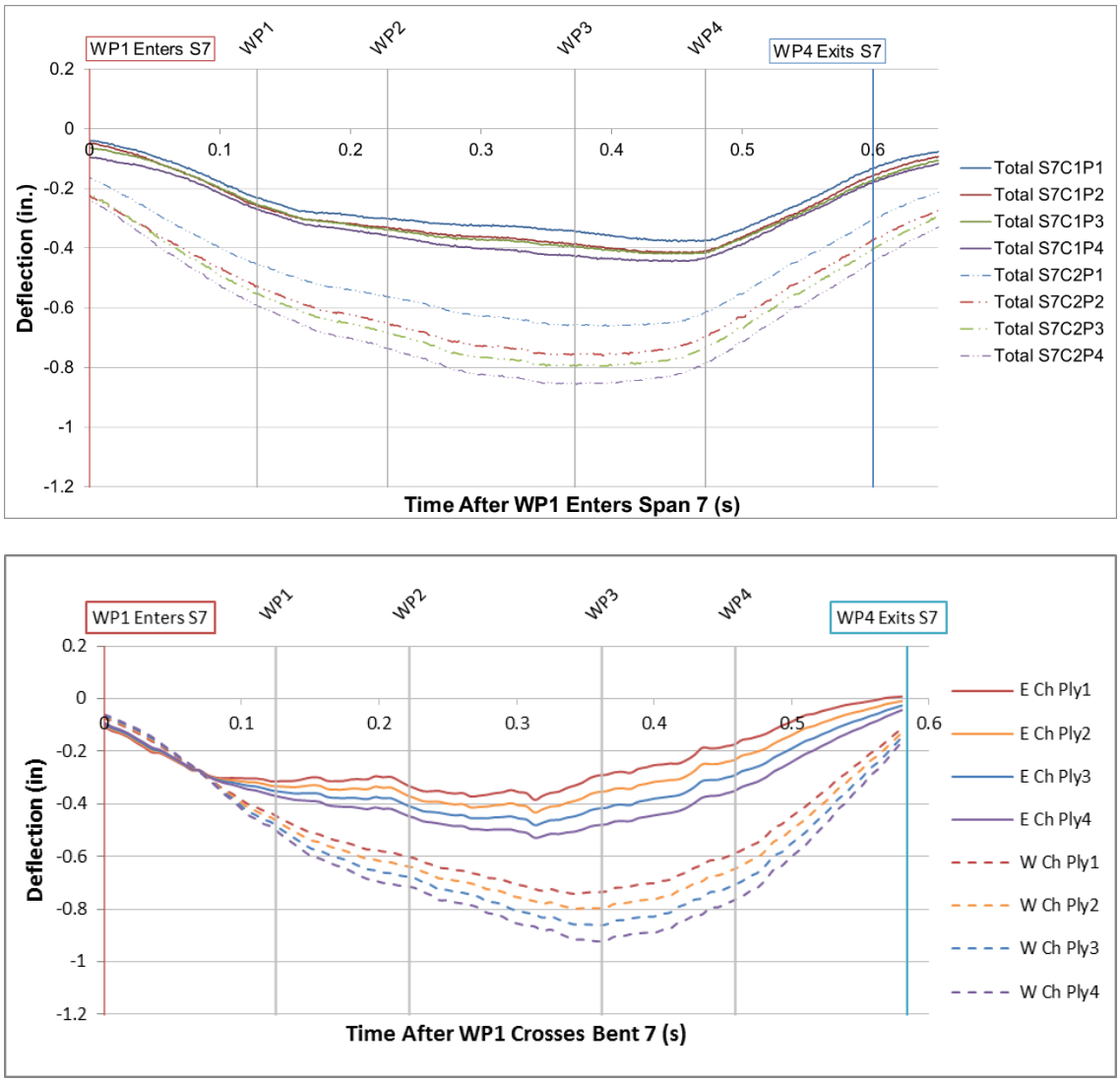

Figure A-244: Stringer Time History Comparison Test 8 Span 7: Experimental (Top), Predicted (Bottom)

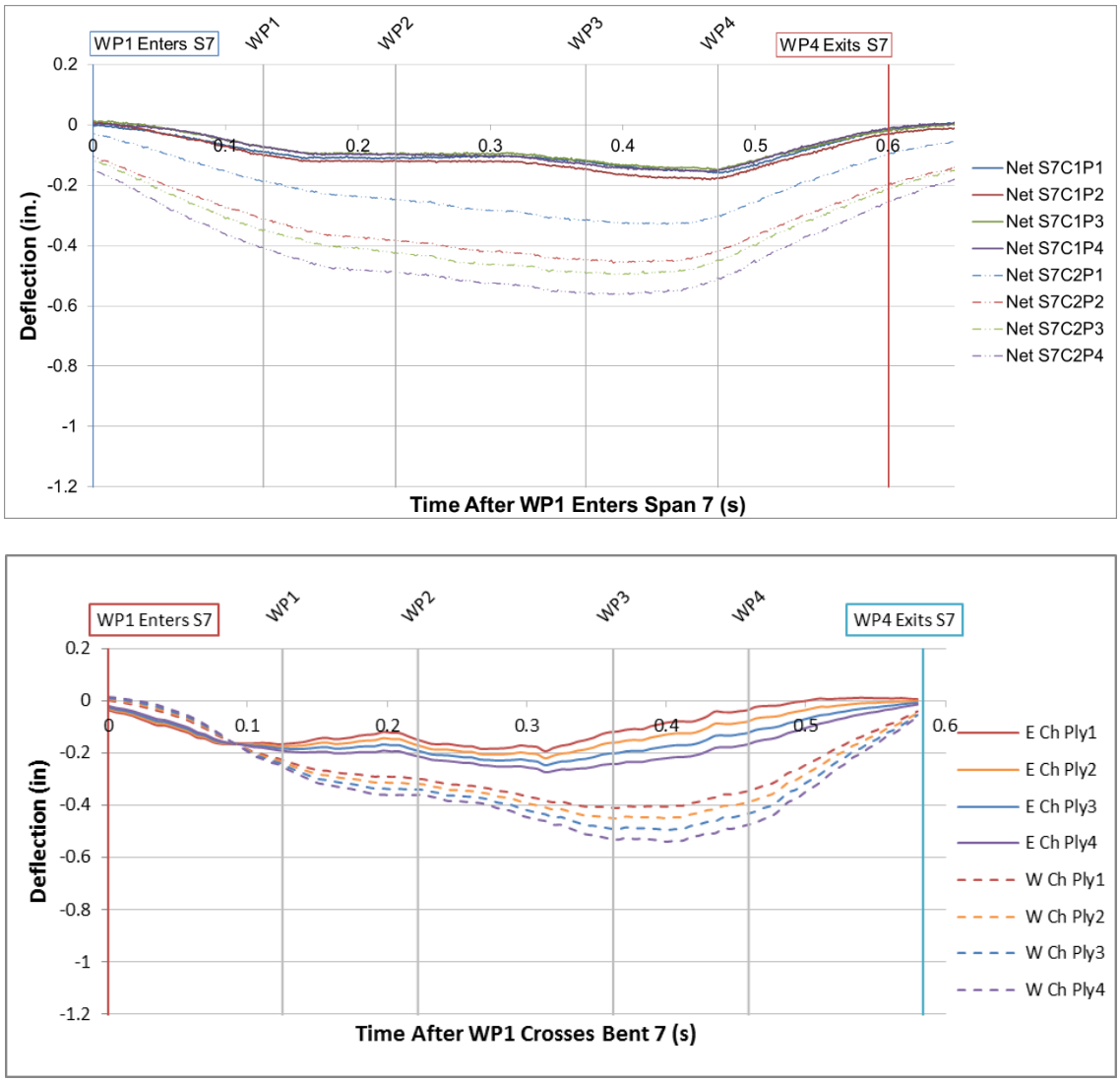

Figure A-245: Net Stringer Time History Comparison Test 8 Span 7: Experimental (Top), Predicted (Bottom)

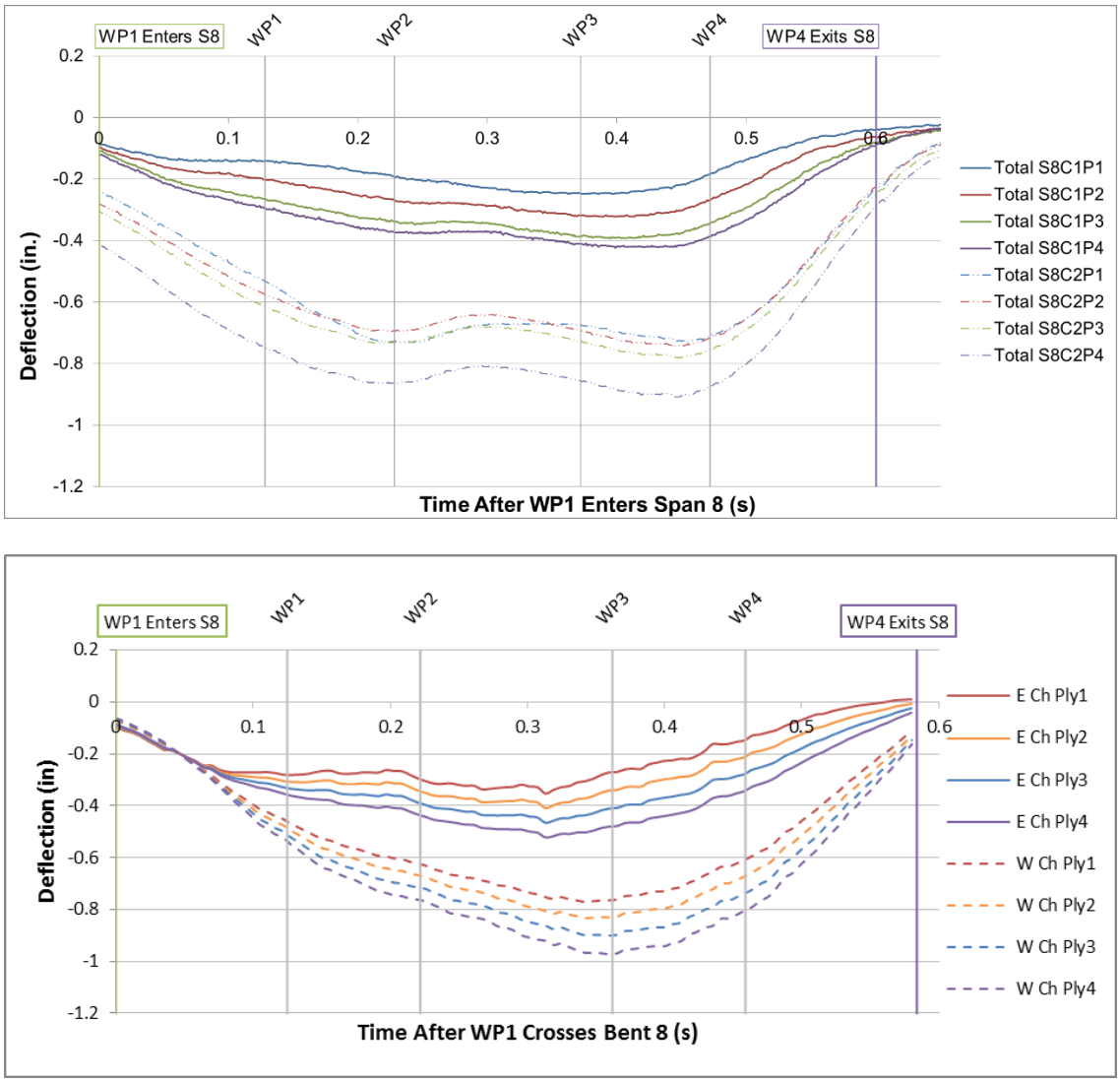

Figure A-246: Stringer Time History Comparison Test 8 Span 8: Experimental (Top), Predicted (Bottom)

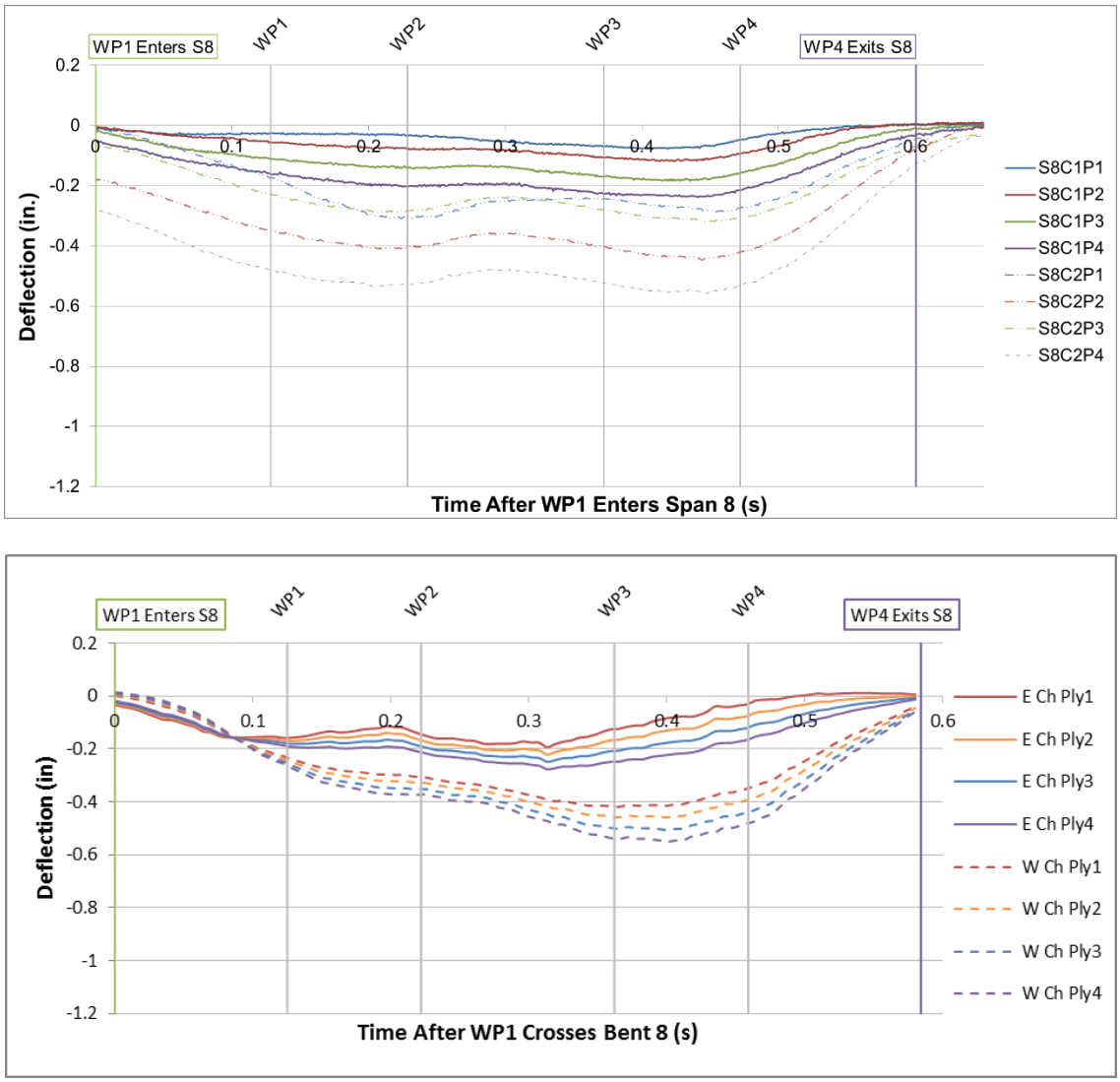

Figure A-247: Net Stringer Time History Comparison Test 8 Span 8: Experimental (Top), Predicted (Bottom)

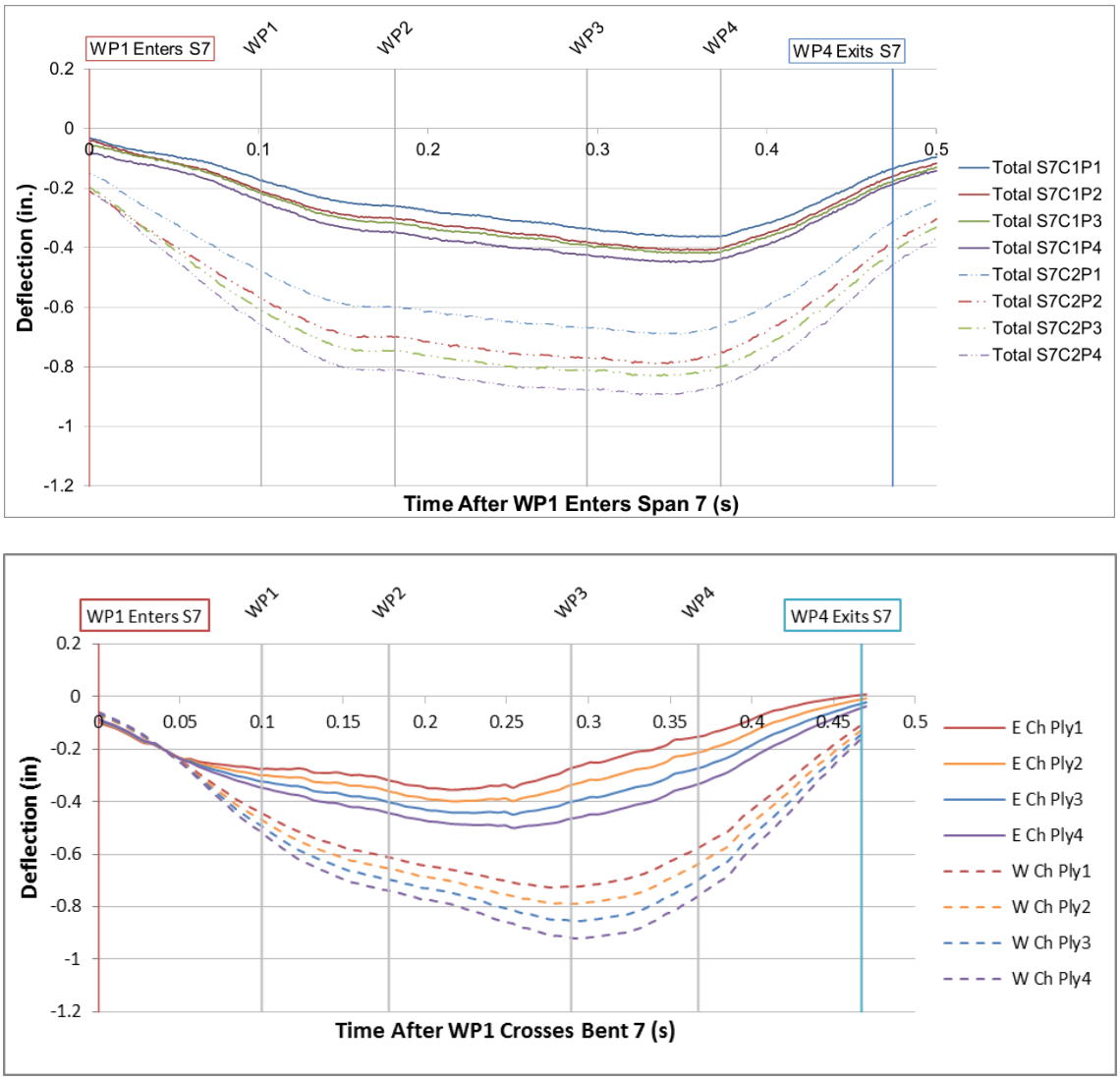

Figure A-248: Stringer Time History Comparison Test 9 Span 7: Experimental (Top), Predicted (Bottom)

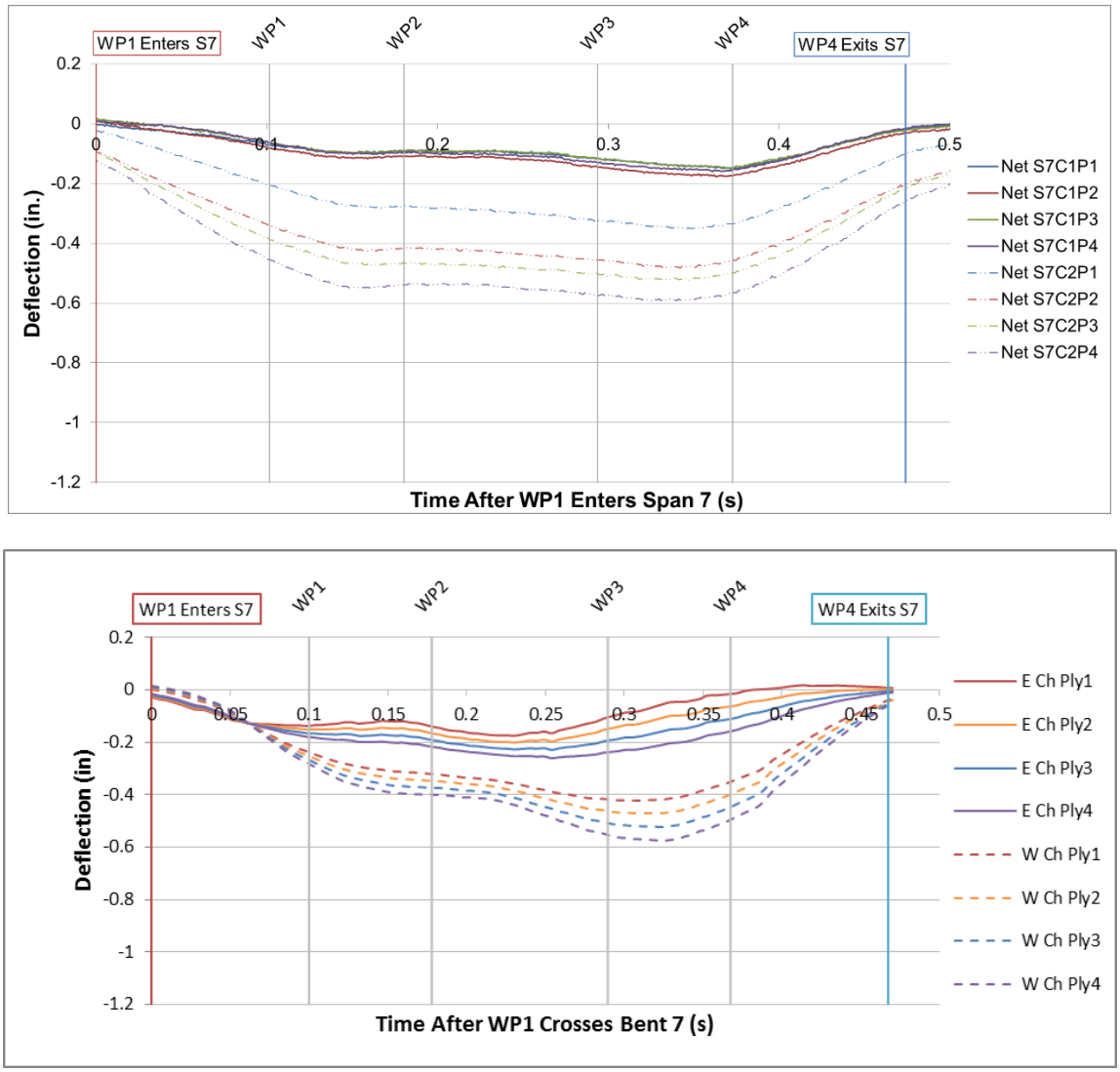

Figure A-249: Net Stringer Time History Comparison Test 9 Span 7: Experimental (Top), Predicted (Bottom)

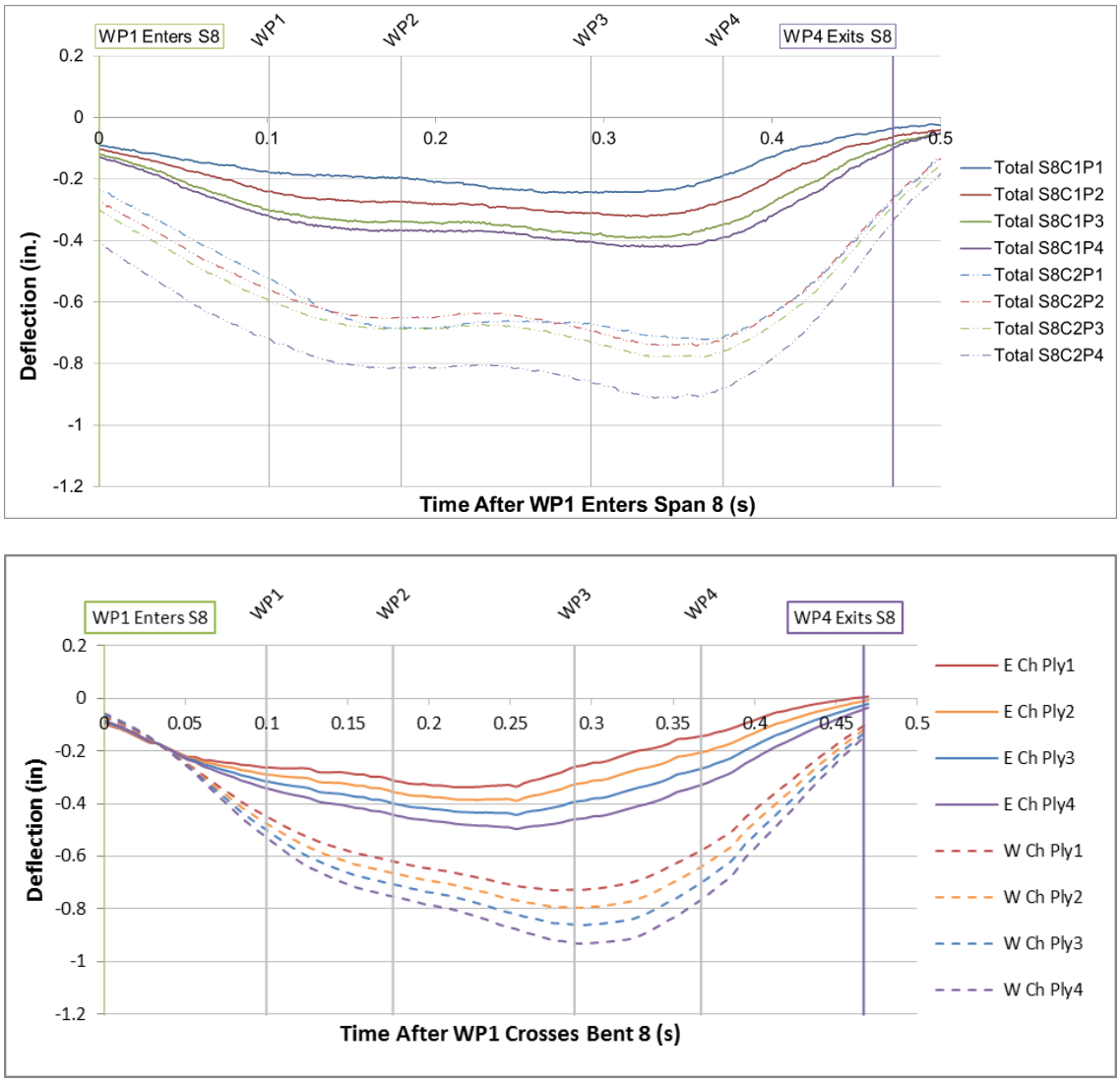

Figure A-250: Stringer Time History Comparison Test 9 Span 8: Experimental (Top), Predicted (Bottom)

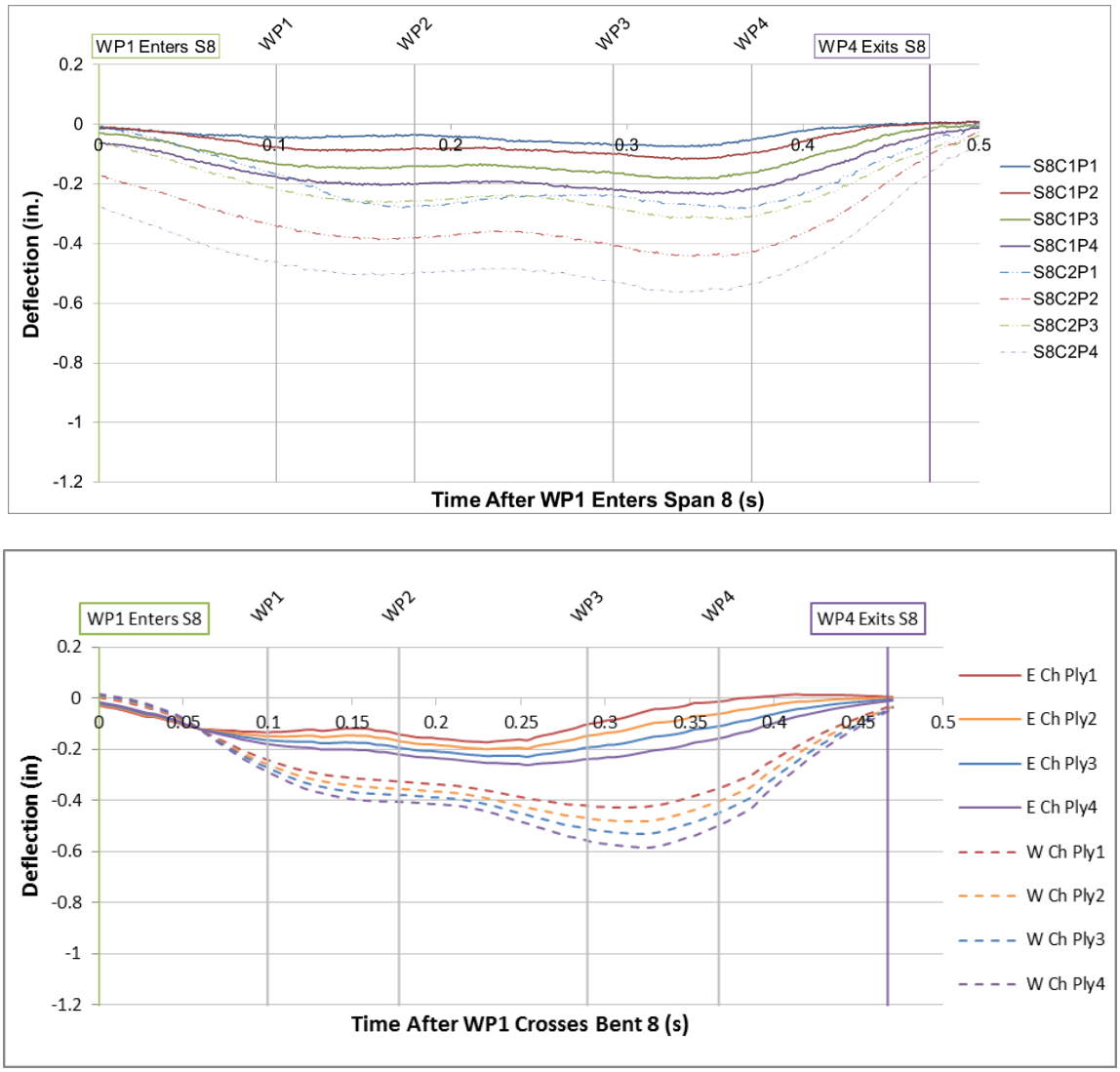

Figure A-251: Net Stringer Time History Comparison Test 9 Span 8: Experimental (Top), Predicted (Bottom)

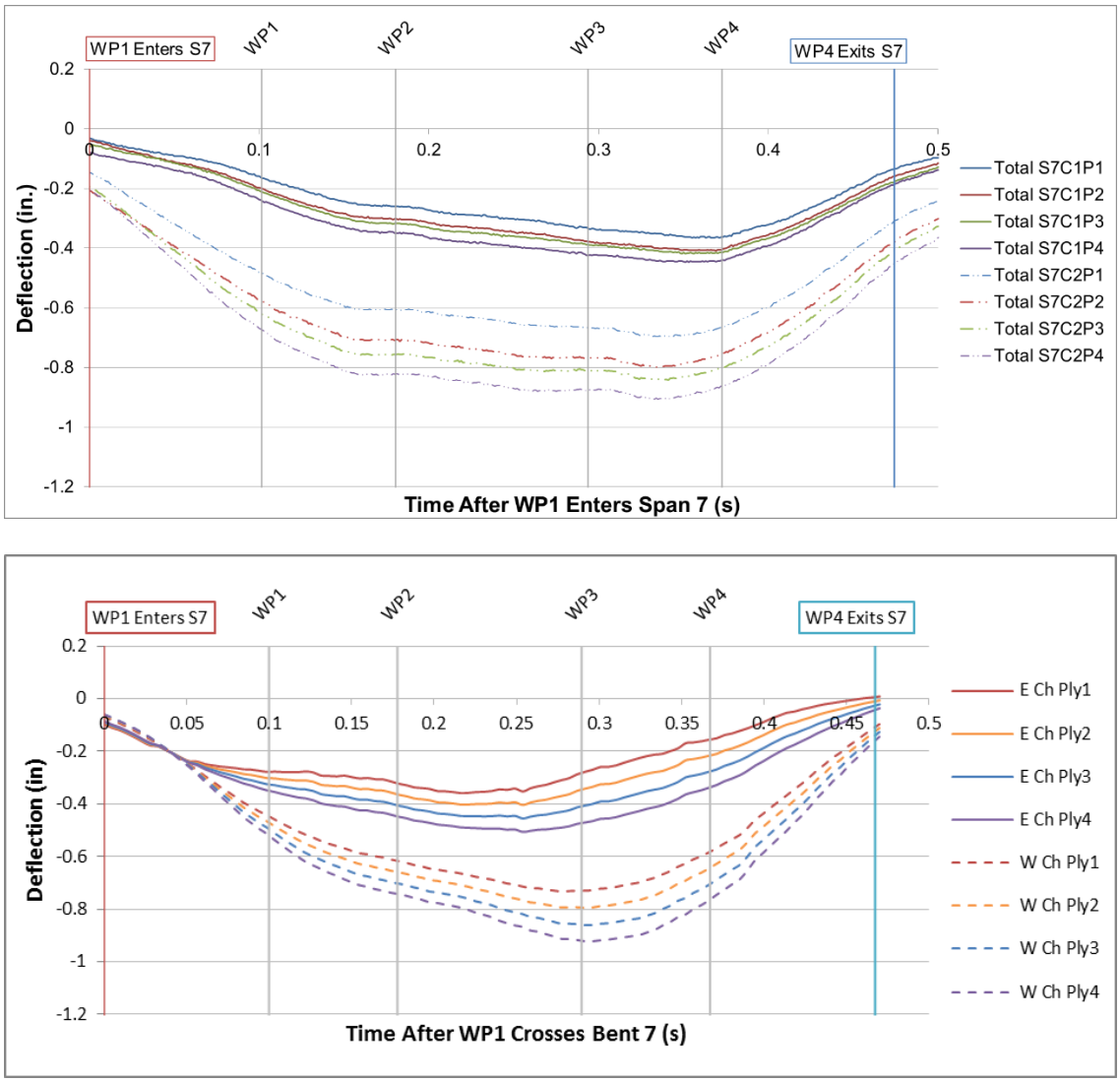

Figure A-252: Stringer Time History Comparison Test 10 Span 7: Experimental (Top), Predicted (Bottom)

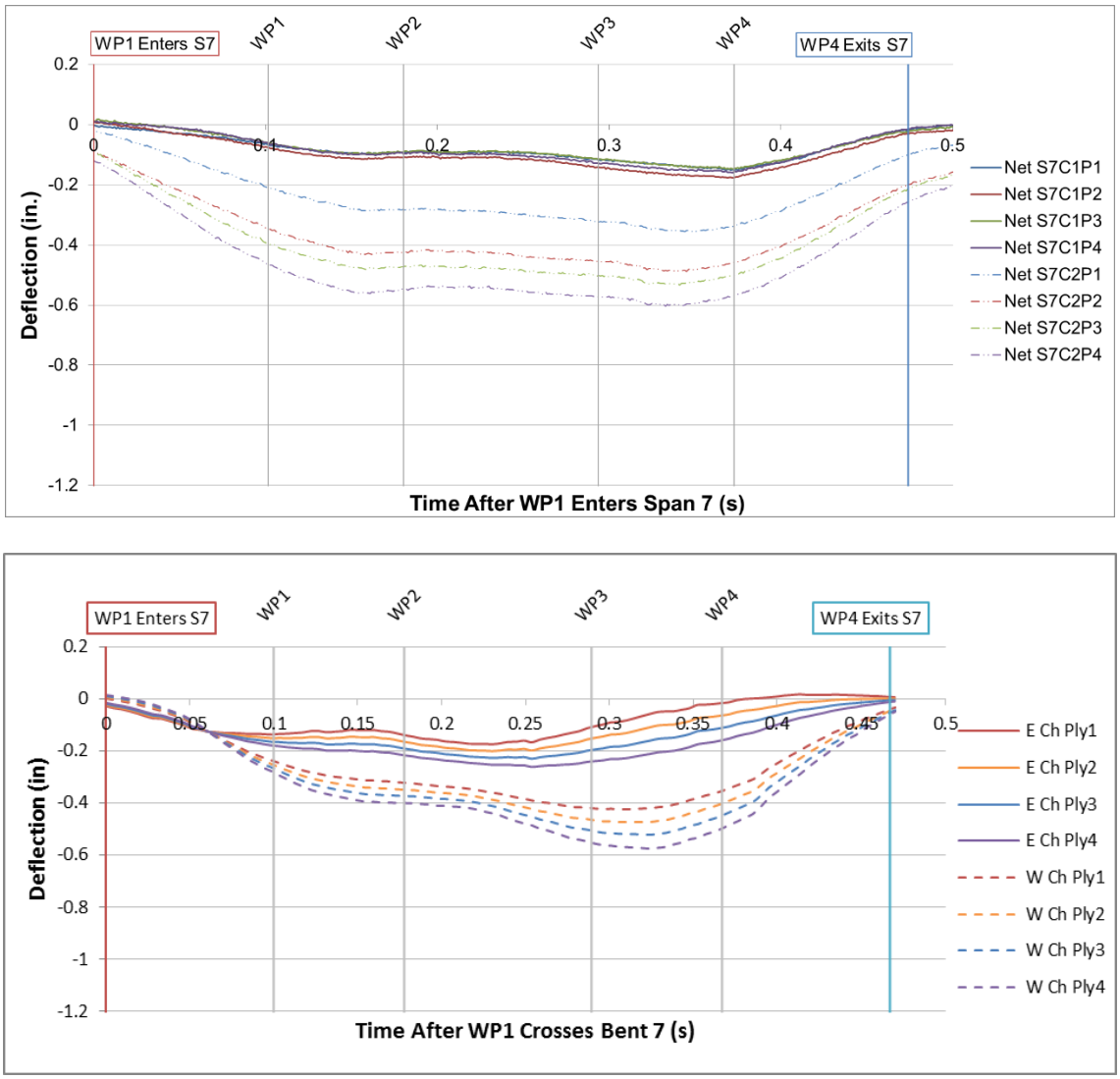

Figure A-253: Net Stringer Time History Comparison Test 10 Span 7: Experimental (Top), Predicted (Bottom)

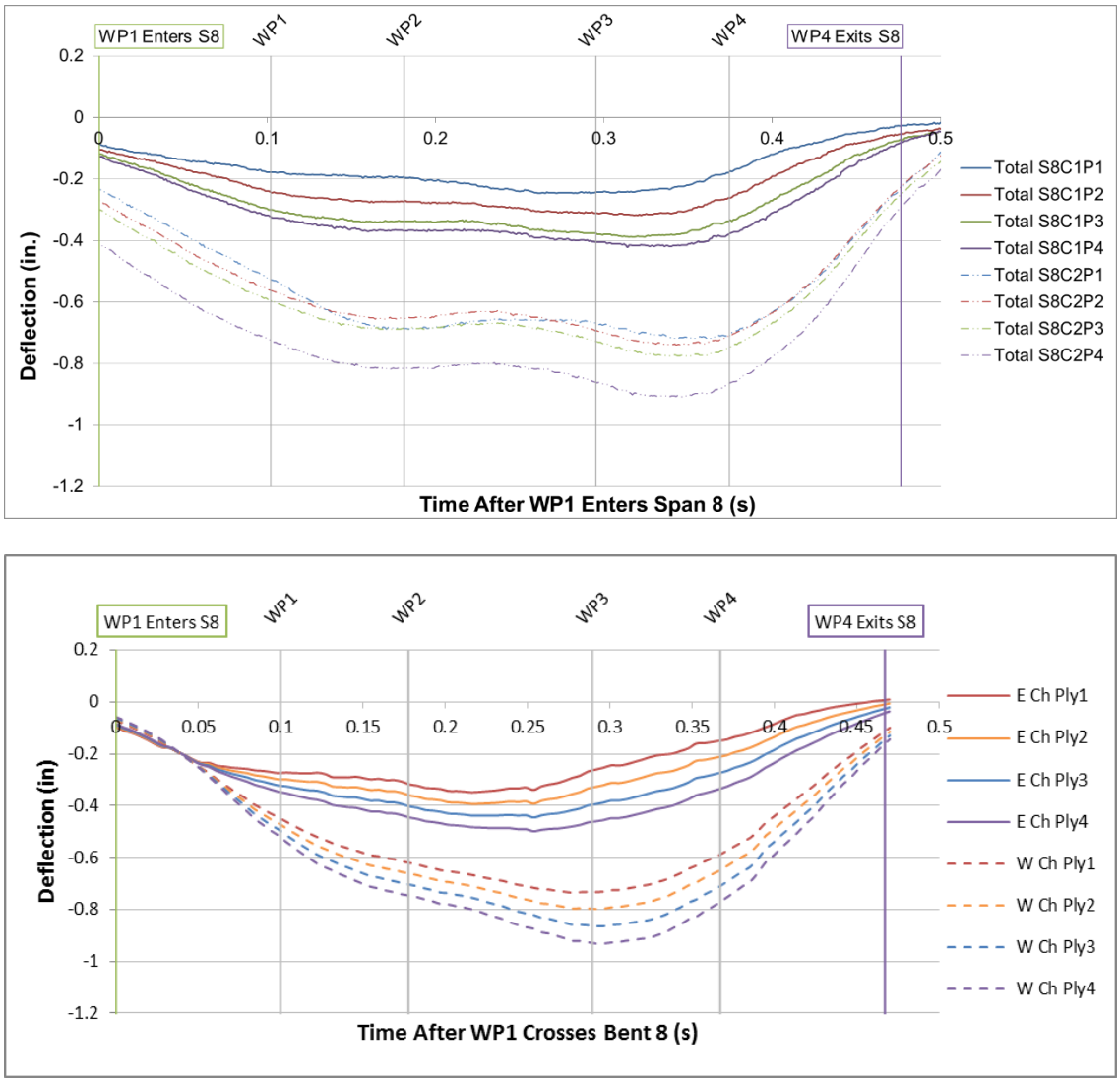

Figure A-254: Stringer Time History Comparison Test 10 Span 8: Experimental (Top), Predicted (Bottom)

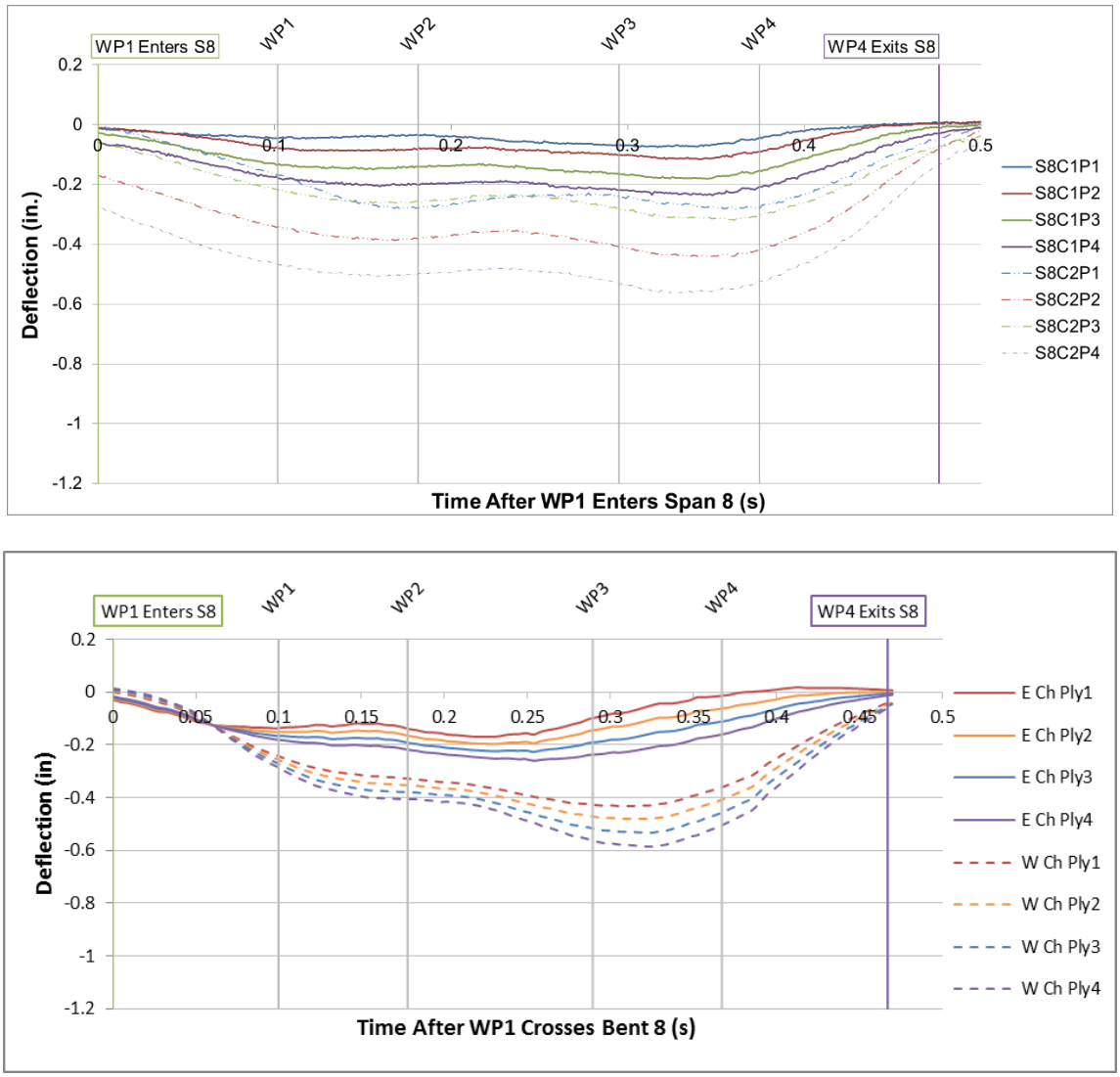

Figure A-255: Net Stringer Time History Comparison Test 10 Span 8: Experimental (Top), Predicted (Bottom)

APPENDIX B

NEURAL NETWORK EXAMPLE

B.1 Neural Network Input Analysis

An example competitive array of neural networks was constructed to analyze the importance of the number of inputs on the performance of the system. Artificial neural networks are able to examine a set of input data and use a series of mathematical operations to correlate it to an output data set. To effectively accomplish this task, the networks must be trained on a provided input and output data set. After the training process has been completed, the networks can then be tested with a new set of input data, and the output will be produced by the system. If the correct output is known for the given set of input, then the performance of the artificial neural networks can be determined. Competitive neural networks makes use of number of neural networks competing in an array to improve the efficiency of the system. There were four artificial neural networks with various architectures used during this example. More information about the training and testing process associated with neural networks is provided in Chapter 2 of this dissertation.

The purpose of this section is to provide an example of how neural networks operate seen in a graphical representation. This example also analyzes how the competitive array of neural networks performs as the number of inputs and outputs the system is trained on is altered. The input for this example are randomly chosen X and Y coordinates with a square window ranging from -2 to 2. Each of these coordinates are

given a designated color and shape depending on their location. These various colors are within specified boundaries to form a picture seen in Figure B-1. The figure shows the target output from the system of networks. The image is meant to resemble a tree with three colors of sky in the background and green grass on the lower layer. The image was constructed with 1,000 different X and Y coordinates and there corresponding colors. The coordinates are the input the neural network and the color and shape in the output. The colors and shapes were represented by a number between zero and 10 in the output training vector.

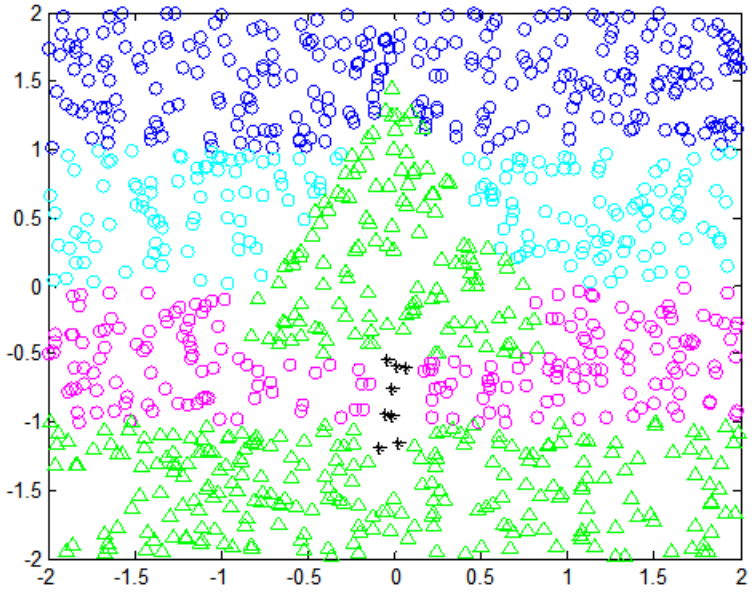

Figure B-1: Neural Network Target Result

The competitive array of neural networks was trained to recognize the patterns correlating the input data to the output data. The networks were first trained with 100 randomly chosen coordinates shown in the left plot of Figure B-2. The blue lines in the

figure indicate the predefined boundaries that determine what color or shape the coordinate should be. The figure shows there is an abundance of white space as there were only 100 data points in this test. After the training was completed, the networks were tested with in the 1,000 input coordinates corresponding to the plot above in Figure B-1. The right side of Figure B-2 shows the results of the neural network testing. The black squares indicates the output produced by the networks was outside the expected range of values. Most of these values are gathered in small sections throughout the plot. When using 100 input data point, the networks had difficulty determining the correct color for some coordinates in each of the regions. The system was 77.4% accurate in estimating the correct output for this test.

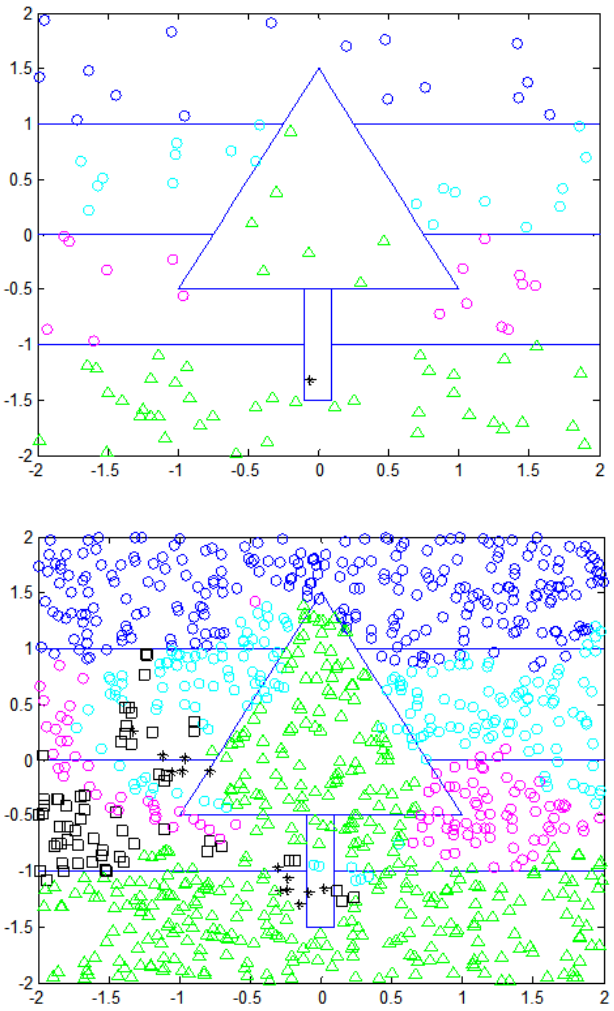

Figure B-2: Example Test 1: Training (Left), Testing (Right)

The second test involved training the networks on 500 different data points. Figure B-3 shows the plot used during training and the plot resulting from testing the data. The competitive array of neural networks were tested with the same set of input coordinates as the first test. The training plot shows most of the white space has been filled and this leads to an improved figure produced from testing. The testing plot shows a majority of the misidentification of the colors comes at the boundaries when more points

are used during training. Most of these errors can be attributed to the training process, as they are located in areas that included rather sizable voids. The accuracy improved dramatically for this test with the network correctly estimating 95.4% of the outputs.

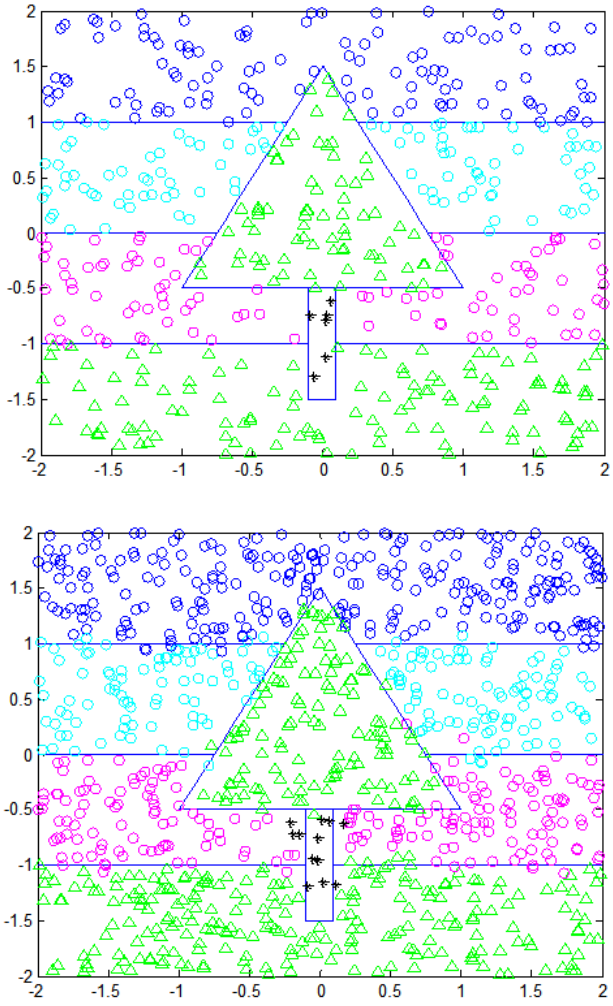

Figure B-3: Example Test 2: Training (Left), Testing (Right)

The third test was similar to the previous two with the exception that 1,000 data points were used for training. This is the same number of points used to the construct the

testing figure. The plot using the training data and the image of the resulting test data is shown in Figure B-4. In this case. The test plot looks nearly identical to the initial correct plot given in Figure B-1. If the lines indicating the boundaries were removed, it would be difficult to identify the incorrectly labeled data points. The accuracy of the networks improved to 96.4% for this test.

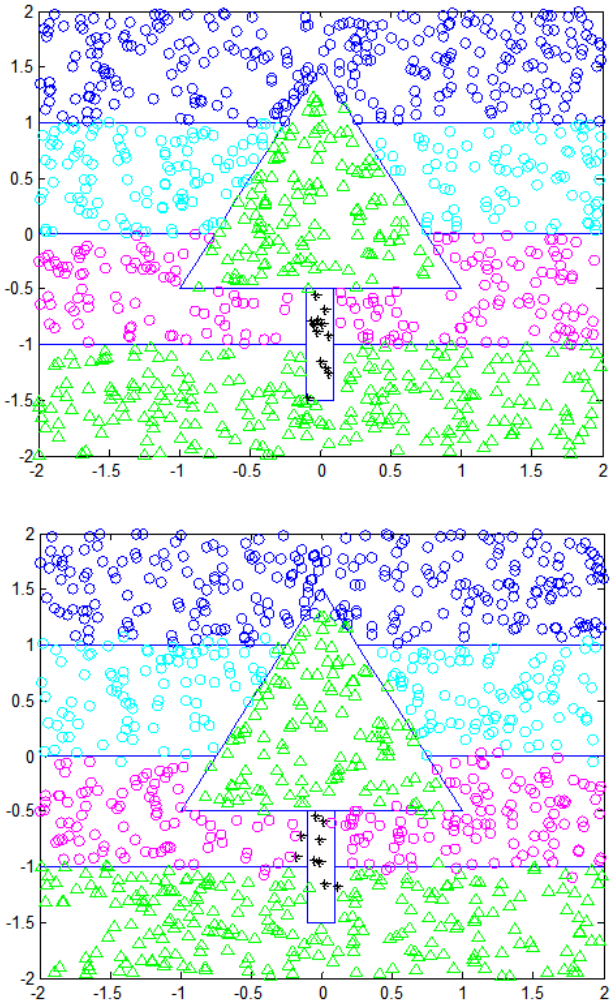

Figure B-4: Example Test 3: Training (Left), Testing (Right)

The competitive array was also tested with 2,000, 5,000, and 10,000 input data points. In each test, the accuracy of the system improved slightly with values of 97.3%, 98.2%, and 98.6%, respectively. The accuracy according to the number of data points the networks were trained with is provided in Table B-1. The table shows there is initially a large jump in accuracy as the number of inputs increases. However, the increase slows dramatically with 1,000 or more inputs trained on the data.

Table B-1: Neural Network Accuracy

Number of Inputs	Percent Correct
100	77.4%
500	95.4%
1000	96.4%
2000	97.3%
5000	98.2%
10000	98.6%

The research showed that using more inputs to train on a specific set of data will improve the accuracy of the competitive neural network system. The disadvantage of using more input vectors arises with the time it takes to train the networks. It only took 20 seconds to train the system using 100 data points. However, that number jumped to over 47 minutes when 10,000 data points were used. That is a substantial increase compared to the difference in accuracy between 10,000 and 1,000 training data vectors. This is a relatively simple system with 2 numbers in the input vector and only 1 in the output. The use more training vectors in a complex system would affect the results

similarly. The number of input vectors becomes another contributing factor in the effectiveness of artificial neural networks, but it comes at a cost of computing time.

B.2 Neural Network Example Matlab Code

```
clear
clc
close all
tic


% Define Limits
a1=-2;
a2=2;
b1=-2;
b2=2;

% Number of Inputs
no=100;

% Assign Random Inputs
 x = a1 + (a2-a1).*rand(no,1);
 y = b1 + (b2-a1).*rand(no,1);
x=x';
y=y';

% Set Boundaries and Plot
figure
m=2;
b=1.5;
x1=-1:.1:0;
y1=1.5+m.*x1;
x2=0:.1:1;
y2=1.5-m.*x2;
plot(x1,y1)
hold on
plot(x2,y2)
hold on
x3=-1:.1:1;
y3=0.*x3-0.5;
plot(x3,y3)
hold on
x4=-0.1:.1:0.1;
y4=0.*x4-1.5;
plot(x4,y4)
hold on
y5=-1.5:.1:-0.5;
x5=0.*y5-0.1;
plot(x5,y5)
hold on
```

```
y6=-1.5:.1:-0.5;
x6=0.*y6+0.1;
plot(x6,y6)
hold on
x7=-2:.1:-0.1;
y7=0.*x7-1;
plot(x7,y7)
hold on
x8=0.1:.1:2;
y8=0.*x8-1;
plot(x8,y8)
hold on
x9=-2:.05:-0.75;
y9=0.*x9;
plot(x9,y9)
hold on
x10=0.75:.05:2;
y10=0.*x10;
plot(x10,y10)
hold on
x11=-2:.05:-0.25;
y11=0.*x11+1;
plot(x11,y11)
hold on
x12=0.25:.05:2;
y12=0.*x12+1;
plot(x12,y12)
hold on
axis([-2,2,-2,2])
for i=1:size(x,2)
    if (y(i)<=m*x(i)+b & y(i)<=-m*x(i)+b & y(i)>=-0.5) | (y(i)<-1 &
x(i)<-0.1) | (y(i)<-1 & x(i)>0.1) | (y(i)<-1.5 & x(i)>=-0.1 &
x(i)<=0.1)
        T0(i)=10;
        plot(x(i),y(i),'g^')
    elseif (y(i)>m*x(i)+b & y(i)>=1 & x(i)<0) | (y(i)>-m*x(i)+b &
y(i)>=1 & x(i)>0)
        T0(i)=8;
        plot(x(i),y(i),'bo')
    elseif (y(i)<-0.5 & y(i)>=-1.5 & x(i)<=0.1 & x(i)>=-0.1)
        T0(i)=5;
        plot(x(i),y(i),'k*')
    elseif (y(i)>m*x(i)+b & y(i)<1 & y(i)>=0) | (y(i)>-m*x(i)+b &
y(i)<1 & y(i)>=0)
        T0(i)=2;
        plot(x(i),y(i),'co')
    else
        T0(i)=0;
        plot(x(i),y(i),'mo')
    end
    hold on
end
```

```
% Inputs for Networks
P0=[x;y];
PT=[P0;T0];

%Scale Data between [.1 .9] (for logsig range)
[P,ps]=mapminmax(P0,-.9, .9) ;
[T,ts]=mapminmax(T0,-.9, .9) ;

% Begin with Zeros
P1=zeros(size(P,1),1);
P2=zeros(size(P,1),1);
P3=zeros(size(P,1),1);
P4=zeros(size(P,1),1);
NN=0;

% Constuct Network Archetecture
net1=newff(P,T,[30 10 5] );
net2=newff(P,T,[15 6 2]);
net3=newff(P,T,[7 3 ]  );
net4=newff(P,T,[4  ] );

% % % Begin Neural Network Training
for i=1:size(P,2)
    %simulate the ith input vector through each network
y1_net1=sim(net1,P(:,i));
y1_net2=sim(net2,P(:,i));
y1_net3=sim(net3,P(:,i));
y1_net4=sim(net4,P(:,i));
    %look at error and choose the smallest to continue trainign the
winning
    %network
y1_error=abs([y1_net1 y1_net2 y1_net3 y1_net4]-T(i));
[mn1,i1]=min(y1_error);
TRNERR=.000000001
TRNEPOCH=500
%Assign each input vector to its winning network
if NN==0
    NN=[i1];
else
    NN=[NN i1];
end
switch i1
    case 1
        if P1==0.
            P1=P(:,i);
            T1=T(i);
        else
            P1=[P1 P(:,i)];
            T1=[T1 T(i)];
        end
        net1.trainParam.epochs=TRNEPOCH;
        net1.trainParam.goal=TRNERR;
        net1.trainParam.showWindow=0;
```

```
        net1=train(net1,P1,T1);
    case 2
        if P2==0.
            P2=P(:,i);
            T2=T(i);
        else
            P2=[P2 P(:,i)];
            T2=[T2 T(i)];
        end
        net2.trainParam.epochs=TRNEPOCH;
        net2.trainParam.goal=TRNERR;
        net2.trainParam.showWindow=0;
        net2=train(net2,P2,T2);
    case 3
        if P3==0.
            P3=P(:,i);
            T3=T(i);
        else
            P3=[P3 P(:,i)];
            T3=[T3 T(i)];
        end
        net3.trainParam.epochs=TRNEPOCH;
        net3.trainParam.goal=TRNERR;
        net3.trainParam.showWindow=0;
        net3=train(net3,P3,T3);
    otherwise
        if P4==0.
            P4=P(:,i);
            T4=T(i);
        else
            P4=[P4 P(:,i)];
            T4=[T4 T(i)];
        end
        net4.trainParam.epochs=TRNEPOCH;
        net4.trainParam.goal=TRNERR;
        net4.trainParam.showWindow=0;
        net4=train(net4,P4,T4);
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


T_new_vec=ind2vec(NN);
sp=0.05;
net_pnn=newpnn(P,T_new_vec,sp);
T_pnn_vec=sim(net_pnn, P);
T_pnn=vec2ind(T_pnn_vec);
T_pnn-NN;
```

```
%Check simulations of Initial Inputs
T_sim=0;
for i=1:size(P,2)
    switch NN(i)
        case 1
          if T_sim==0
              T_sim=sim(net1,P(:,i));
          else
              T_sim=[T_sim sim(net1,P(:,i))];
          end
        case 2
          if T_sim==0
              T_sim=sim(net2,P(:,i));
          else
              T_sim=[T_sim sim(net2,P(:,i))];
          end
        case 3
          if T_sim==0
              T_sim=sim(net3,P(:,i));
          else
              T_sim=[T_sim sim(net3,P(:,i))];
          end
        otherwise
          if T_sim==0
              T_sim=sim(net4,P(:,i));
          else
              T_sim=[T_sim sim(net4,P(:,i))];
          end
    end
end

    T_sim0=mapminmax('reverse',T_sim,ts);

figure
plot(T0,'bs')
hold on
plot(T_sim0,'rx')
hold on
plot(round(T_sim0),'g*')

cnt=0;
for i=1:length(T0)
    if abs( T_sim0(i)-T0(i))<=.4
        cnt=cnt+1;
    end
end
CorrectRat=cnt/length(T0)

figure
m=2;
b=1.5;
x1=-1:.1:0;
y1=1.5+m.*x1;
```

```
x2=0:.1:1;
y2=1.5-m.*x2;
plot(x1,y1)
hold on
plot(x2,y2)
hold on
x3=-1:.1:1;
y3=0.*x3-0.5;
plot(x3,y3)
hold on
x4=-0.1:.1:0.1;
y4=0.*x4-1.5;
plot(x4,y4)
hold on
y5=-1.5:.1:-0.5;
x5=0.*y5-0.1;
plot(x5,y5)
hold on
y6=-1.5:.1:-0.5;
x6=0.*y6+0.1;
plot(x6,y6)
hold on
x7=-2:.1:-0.1;
y7=0.*x7-1;
plot(x7,y7)
hold on
x8=0.1:.1:2;
y8=0.*x8-1;
plot(x8,y8)
hold on
x9=-2:.05:-0.75;
y9=0.*x9;
plot(x9,y9)
hold on
x10=0.75:.05:2;
y10=0.*x10;
plot(x10,y10)
hold on
x11=-2:.05:-0.25;
y11=0.*x11+1;
plot(x11,y11)
hold on
x12=0.25:.05:2;
y12=0.*x12+1;
plot(x12,y12)
hold on
axis([-2,2,-2,2])

correct1=0;
correct2=0;
correct3=0;
correct4=0;
correct5=0;
```

```
cntwrong=0;
cnt1=0;
cnt2=0;
cnt3=0;
cnt4=0;
cnt5=0;

tol=1;
for i=1:size(x,2)
    if abs((T_sim0(i))-10)<=tol
        plot(x(i),y(i),'g^')
    elseif abs(T_sim0(i)-8)<=tol
        plot(x(i),y(i),'bo')
    elseif abs(T_sim0(i)-5)<=tol
        plot(x(i),y(i),'k*')
    elseif abs(T_sim0(i)-2)<=tol
        plot(x(i),y(i),'co')
    elseif abs(T_sim0(i)-0)<=tol
        plot(x(i),y(i),'mo')
    else
        plot(x(i),y(i),'kd')
        cntwrong=cntwrong+1;
    end
    hold on
    if abs(T_sim0(i)-T0(i))<=tol
        if T0(i)==10
            correct1=correct1+1;
        elseif T0(i)==8
            correct2=correct2+1;
        elseif T0(i)==5
            correct3=correct3+1;
        elseif T0(i)==2
            correct4=correct4+1;
        elseif T0(i)==0
            correct5=correct5+1;
        end
    end
    if T0(i)==10
        cnt1=cnt1+1;
    elseif T0(i)==8
        cnt2=cnt2+1;
    elseif T0(i)==5
        cnt3=cnt3+1;
    elseif T0(i)==2
        cnt4=cnt4+1;
    elseif T0(i)==0
        cnt5=cnt5+1;
    end

end


PC1=correct1/cnt1
```

```
PC2=correct2/cnt2
PC3=correct3/cnt3
PC4=correct4/cnt4
PC5=correct5/cnt5
PC=(correct1+correct2+correct3+correct4+correct5)/no
PNC=cntwrong/no

toc
 save example net1 net2 net3 net4 net_pnn


%%%%%%%%%%%%%%%%%%%%Test Networks with New Data%%%%%%%%%%%%%%%%%%%%%%%%
 load example net1 net2 net3 net4 net_pnn
a1t=-2;
a2t=2;
b1t=-2;
b2t=2;

not=1000


%  xt = a1t + (a2t-a1t).*rand(not,1);
%  yt = b1t + (b2t-a1t).*rand(not,1);
% xt=xt';
% yt=yt';
% save input xt yt
load input xt yt

%pattern1
figure
mt=2;
bt=1.5;
axis([-2,2,-2,2])
for i=1:size(xt,2)
    if (yt(i)<=mt*xt(i)+bt & yt(i)<=-mt*xt(i)+bt & yt(i)>=-0.5) |
(yt(i)<-1 & xt(i)<-0.1) | (yt(i)<-1 & xt(i)>0.1) | (yt(i)<-1.5 &
xt(i)>=-0.1 & xt(i)<=0.1)
        T0t(i)=10;
        plot(xt(i),yt(i),'g^')
    elseif (yt(i)>mt*xt(i)+bt & yt(i)>=1 & xt(i)<0) | (yt(i)>-
mt*xt(i)+bt & yt(i)>=1 & xt(i)>0)
        T0t(i)=8;
        plot(xt(i),yt(i),'bo')
    elseif (yt(i)<-0.5 & yt(i)>=-1.5 & xt(i)<=0.1 & xt(i)>=-0.1)
        T0t(i)=5;
        plot(xt(i),yt(i),'k*')
    elseif (yt(i)>mt*xt(i)+bt & yt(i)<1 & yt(i)>=0) | (yt(i)>-
mt*xt(i)+bt & yt(i)<1 & yt(i)>=0)
        T0t(i)=2;
        plot(xt(i),yt(i),'co')
    else
        T0t(i)=0;
```

```
        plot(xt(i),yt(i),'mo')
    end
    hold on
end

P0t=[xt;yt];
PTt=[P0t;T0t];

%Scale Data between [.1 .9] (for logsig range)
[PTt,pst]=mapminmax(P0t,-.9, .9) ;
[TTt,tst]=mapminmax(T0t,-.9, .9) ;
%%Test Sorting here
T_sort1t=sim(net_pnn, PTt);
T_sortt=vec2ind(T_sort1t);

%Check simulations of Test Inputs
T_Testt=0;
for i=1:size(PTt,2)
    switch T_sortt(i)
        case 1
          if T_Testt==0
              T_Testt=sim(net1,PTt(:,i));
          else
              T_Testt=[T_Testt sim(net1,PTt(:,i))];
          end
        case 2
          if T_Testt==0
              T_Testt=sim(net2,PTt(:,i));
          else
              T_Testt=[T_Testt sim(net2,PTt(:,i))];
          end
        case 3
          if T_Testt==0
              T_Testt=sim(net3,PTt(:,i));
          else
              T_Testt=[T_Testt sim(net3,PTt(:,i))];
          end
        otherwise
          if T_Testt==0
              T_Testt=sim(net4,PTt(:,i));
          else
              T_Testt=[T_Testt sim(net4,PTt(:,i))];
          end
    end
end

%[T_Testfinal]=postmnmx(T_Test,mint,maxt);
[T_Testfinal]=mapminmax('reverse',T_Testt,tst);
figure

plot(T0t,'bs')
hold on
plot(T_Testfinal,'rx')
```

```
hold on
plot(round(T_Testfinal),'g*')

cntt=0;
for i=1:length(T0t)
    if abs( T_Testfinal(i)-T0t(i))<=.4
        cntt=cntt+1;
    end
end
CorrectRatt=cntt/length(T0t)

figure
mt=2;
bt=1.5;
x1t=-1:.1:0;
y1t=1.5+mt.*x1t;
x2t=0:.1:1;
y2t=1.5-mt.*x2t;
plot(x1t,y1t)
hold on
plot(x2t,y2t)
hold on
x3t=-1:.1:1;
y3t=0.*x3t-0.5;
plot(x3t,y3t)
hold on
x4t=-0.1:.1:0.1;
y4t=0.*x4t-1.5;
plot(x4t,y4t)
hold on
y5t=-1.5:.1:-0.5;
x5t=0.*y5t-0.1;
plot(x5t,y5t)
hold on
y6t=-1.5:.1:-0.5;
x6t=0.*y6t+0.1;
plot(x6t,y6t)
hold on
x7t=-2:.1:-0.1;
y7t=0.*x7t-1;
plot(x7t,y7t)
hold on
x8t=0.1:.1:2;
y8t=0.*x8t-1;
plot(x8t,y8t)
hold on
x9t=-2:.05:-0.75;
y9t=0.*x9t;
plot(x9t,y9t)
hold on
x10t=0.75:.05:2;
y10t=0.*x10t;
plot(x10t,y10t)
```

```
hold on
x11t=-2:.05:-0.25;
y11t=0.*x11t+1;
plot(x11t,y11t)
hold on
x12t=0.25:.05:2;
y12t=0.*x12t+1;
plot(x12t,y12t)
hold on
axis([-2,2,-2,2])

correct1t=0;
correct2t=0;
correct3t=0;
correct4t=0;
correct5t=0;
cntwrongt=0;
cnt1t=0;
cnt2t=0;
cnt3t=0;
cnt4t=0;
cnt5t=0;

tolt=1;
for i=1:size(xt,2)
    if abs((T_Testfinal(i))-10)<=tolt
        plot(xt(i),yt(i),'g^')
    elseif abs(T_Testfinal(i)-8)<=tolt
        plot(xt(i),yt(i),'bo')
    elseif abs(T_Testfinal(i)-5)<=tolt
        plot(xt(i),yt(i),'k*')
    elseif abs(T_Testfinal(i)-2)<=tolt
        plot(xt(i),yt(i),'co')
    elseif abs(T_Testfinal(i)-0)<=tolt
        plot(xt(i),yt(i),'mo')
    else
        plot(xt(i),yt(i),'ks')
        cntwrongt=cntwrongt+1;
    end
    hold on
    if abs(T_Testfinal(i)-T0t(i))<=tolt
        if T0t(i)==10
            correct1t=correct1t+1;
        elseif T0t(i)==8
            correct2t=correct2t+1;
        elseif T0t(i)==5
            correct3t=correct3t+1;
        elseif T0t(i)==2
            correct4t=correct4t+1;
        elseif T0t(i)==0
            correct5t=correct5t+1;
        end
    end
```

```
    if T0t(i)==10
        cnt1t=cnt1t+1;
    elseif T0t(i)==8
        cnt2t=cnt2t+1;
    elseif T0t(i)==5
        cnt3t=cnt3t+1;
    elseif T0t(i)==2
        cnt4t=cnt4t+1;
    elseif T0t(i)==0
        cnt5t=cnt5t+1;
    end

end


PC1t=correct1t/cnt1t
PC2t=correct2t/cnt2t
PC3t=correct3t/cnt3t
PC4t=correct4t/cnt4t
PC5t=correct5t/cnt5t
PCt=(correct1t+correct2t+correct3t+correct4t+correct5t)/not
PNCt=cntwrongt/not
```