Perpetual Pavement Analysis for the Marquette Interchange Instrumentation Project

Nicholas James Hornyak
Marquette University

PERPETUAL PAVEMENT ANALYSIS FOR THE MARQUETTE INTERCHANGE INSTRUMENTATION PROJECT

by

Nicholas J. Hornyak, B.S., M.S.

A Dissertation submitted to the Faculty of the Graduate School,
Marquette University,
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy

Milwaukee, Wisconsin

May 2010

ABSTRACT
PERPETUAL PAVEMENT ANALYSIS FOR THE MARQUETTE
INTERCHANGE INSTRUMENTATION PROJECT

Nicholas J. Hornyak, B.S., M.S.

Marquette University, 2010

With the emergence of the mechanistic-empirical pavement design process, development of pavements rely on the structural response and fatigue characteristics of pavement materials due to traffic loads. In the past, pavement design has been almost entirely based on empirical data. One area of interest in designing hot-mix asphalt (HMA) pavements is the fatigue life of the pavement, which has been shown to be dependent on the horizontal strain in the pavement.

This research is focused on measuring the structural response of a pavement located within the Marquette Interchange Project in order to analyze fatigue behavior with great detail. Virtually all variables which affect the life of HMA pavements were measured and analyzed in accordance with the structural data. Assumptions that were historically used in design were verified and suggestions regarding the structural response of the pavement are given. These include results from analyses of load pulse duration, analytical stress/strain predictions, and strain influence.

ACKNOWLEDGMENTS

Nicholas J. Hornyak, B.S., M.S.

I am grateful to all of the people who helped and guided me throughout the years of this work. Without their support this project could not have been completed.

My advisor, Dr. James Crovetti, has been my mentor for many years at Marquette and his knowledge and experience has been a priceless gift. I was given the opportunity to work with him and it has been one of the greatest learning experiences in my life. I thank him for his time, his gift of learning, his friendship, and his patience.

I also want to thank my committee members, Dr. Baolin Wan, Dr. Christopher Foley, Dr. Raymond Fournelle, and Dr. Saeed Karshenas. Together, their vast knowledge has helped guide me throughout this work and also throughout my time at Marquette.

David Newman and Jay Schabelski have been great friends and played a large role in building the experimentation project. Both have been a large part in developing my laboratory skills and always offered a helping hand and guidance when needed.

The experimentation could not have been possible if not for the Wisconsin Highway Research Program and its members. In addition, I would like to thank the contractors who helped coordinate the execution of the instrumentation project during construction of the Marquette Interchange Project: WisDOT, Payne & Dolan, Walsh Construction, Collins Engineers, Outdoor Lighting, HNTB, and TAPCO.

Lastly I want to thank my family for their love, support, and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF EQUATIONS

Chapter 1 - Introduction

Traffic volume in the United States has been growing at a steady pace since the car was introduced. It was immediately apparent that the ability to move across large distances with little effort was of huge benefit. This ability has been the foundation for the economic and technological growth in United States for decades and will continue for decades to come.

A younger Dwight D. Eisenhower participated in military convoy across the United States shortly after World War I. The purpose was to field test Army vehicles and to measure the ease of moving an Army across the continent. During the Second World War Eisenhower served as an Allied Commander in Europe, overseeing the defeat of the Nazi army. He was fascinated at the mobility of the Nazi army, due largely to the autobahn system of roadways. When Eisenhower returned to the States and took office as the nation's president, he persuaded congress to pass the Federal-Aid Highway Act of 1956, the birth of the interstate highway system (Pfeiffer, 2006).

Today, all citizens of the U.S. enjoy the benefits of a high quality transportation system thanks to the efforts of Dwight D. Eisenhower and other supporters of the system. However, in 2006, the system celebrated its 50th anniversary, and many of our roadways are deteriorating and need rehabilitation. Present traffic volumes are higher than engineers expected when developing the system, and the cost of delays to today's motorist is large.

These costs are well understood in today's economy, and civil engineers are charged with building new facilities with more capacity, better performance, safer and longer design lives while still keeping designs economical.

Pavement design ideology has undergone changes since the first highways were built, and there is now a revolution taking place. In the past, pavements were designed using empirical data from a handful of road tests, experience, and rules of thumb. Since computers have become common place, more detailed design procedures based on modern engineering concepts are being implemented. This new design process is currently referred to as the Mechanistic-Empirical (M-E) pavement design process. It incorporates basic material mechanics principles while still maintaining some of the empirical relationships of past and present pavement research.

1.1 - Problem Statement

Most asphalt pavements will eventually deteriorate over time to a condition where the roadway is unsafe or unusable. Typical failure modes are bottom-up fatigue cracking, rutting, thermal cracking, and top-down fatigue cracking. Pavement engineers have been designing new facilities with higher quality materials and longer lasting performance. However, the determination of service life for a particular design has been difficult to accurately predict. These poor predictions generally come from the variable nature of the materials used for construction and traffic uncertainty.

Of particular interest is a new concept called perpetual pavement, sometimes also called long-life pavements. The design philosophy of perpetual pavements is to limit the average maximum tensile strain due to an 18 kip axle load during July temperatures to

prevent the bottom-up mode of failure. The specific limiting values of strain are disputable, but generally fall around the 70 micro-strain range. This strain threshold corresponds to stresses that are near the endurance limit of the pavement structure.

It is understood that while this only prevents one mode of failure, other failure modes can be accommodated by good mix designs and simple preventative maintenance and rehab work. While these other modes of failure are possible, they should be limited to a sacrificial wearing surface. The main structural component of the pavement remains intact and when needed a removal and relay of the wearing surface can restore quality.

These pavements have become mainstream in the state of Wisconsin, but the predicted and actual performance has not been analyzed. Furthermore, the maximum tensile strain, a main limit state during design, has not been accurately measured in actual pavement structures. This research aims to answer these questions concerning perpetual pavements.

1.2 - Objectives/Significance of Work

This research highlights and analyzes the in place performance of a perpetual asphalt pavement section located along I-43 NB within the north leg of the Marquette Interchange project in downtown Milwaukee, Wisconsin. Numerous dynamic pavement sensors were installed into this test section and continually recorded. The data produced is valuable because it allows pavement engineers to analyze dynamic responses year round under real traffic.

Most pavement test tracks and test facilities developed used closed coursed traffic or simulated traffic loading. It has been difficult to capture the ‘real’ effects of real

traffic due to technological limitations of both hardware and software used in measuring. In addition, most simulated testing facilities take only dynamic pavement response measurements periodically, not continuously, leaving gaps of information that could lead to erroneous conclusions.

Novel methods have been developed in this work to analyze the massive amount of data collected. Both the real data and theoretical calculations were then meshed together as an in depth comparison between predicted and actual pavement responses. In addition many other interesting observations and relationships have been uncovered and are discussed.

The benefits of understanding how perpetual pavements respond over time to traffic loads are invaluable. This project provides engineers a window into the pavement aging process, showing how the pavement responds dynamically to repeated loads throughout seasonal variations. At a minimum, the state of Wisconsin has gained data to help pavement engineers design more reliable and cost effective pavements. It is hoped that this data and research will benefit the pavement engineering community as a whole.

1.3 - Organization of Research

This project is comprised of five main sections which are the following: literature review, instrumentation, data processing and analysis, pavement modeling, results and conclusions.

The literature review highlights past research regarding pavement life prediction, performance, and response, and formulates a basis for the need for this research. Instrumentation covers all aspects of the pavement test section, detailing the specifics on

the location, orientation, and measurement type of each sensor and how the data was acquired and stored.

Data generated from the test section was stored and subsequently post processed to gather all of the pertinent information. The data processing and analysis section covers these processes and thoroughly explains the algorithms and data mining techniques used to find the important information.

In order to fully analyze the pavement test section, a finite element model was built and used to compare analytical and measured pavement response to loads. A fully automated program was written to accomplish this and implements a finite-element model for the analysis.

Finally, the results from the modeled and measured pavement responses were combined for comparison to understand the pavement behavior. Final conclusions regarding the study are presented in an organized manner.

Chapter 2 - Literature Review

There have been many attempts to accurately model pavement performance in order to predict pavement service life. These attempts include studying pavements on macro- and meso- level scales. Research at the macro-level scale is composed of full scale test tracks and accelerated testing facilities. The meso-level scale consists of research that has been performed on smaller material specimens studying properties such as cracking mechanisms, binder film behavior, healing, etc. under controlled laboratory testing.

The following is a summary of research that pertains, in most respects, to the research conducted for this project. Some of the most recent work is presented and older work is also included to provide a complete understanding of the progression of the pavement design theory and state of practice.

2.1 - Asphalt Pavement Design Principles

The American Association of State Highway and Transportation Officials (AASHTO) have made large contributions to the practice of pavement design and research. The first edition (1961) of the design guide (known as the "AASHO Interim Guide for the Design of Rigid and Flexible Pavements") used pavement models created from data taken from a full scale highway experiment known as the AASHO Road Test. This experiment helped formulate a design process which was largely, if not completely, empirically based. This design process was carried through to the 1972 and 1993 versions of the design guide. The 1993 design guide is still in use today but is quickly being replaced by the Mechanistic-Empirical Pavement Design Guide (MEPDG). The

concept of mechanistic-empirical design implies elements of both empiricism as well as the principles of mechanics.

2.1.1 - AASHTO Design of Pavement Structures (AASHTO, 1993)

For the purposes of this research and understanding its importance, the salient features of the current pavement design methods need to be understood. In pavement design it is important to consider all factors which affect the performance of a pavement. The guide makes 9 distinct considerations for pavement design (AASHTO, 1993);

1. Pavement performance
2. Traffic
3. Roadbed soil
4. Materials of construction
5. Environment
6. Drainage
7. Reliability
8. Life-cycle costs
9. Shoulder design

These considerations are basic in nature, but cover the factors that dictate the life of a pavement in service and should be included in any thorough research.

Pavement performance includes functional performance, structural performance, and safety, of which the design guide focuses on the former two. The structural performance considers the pavement's ability to carry loads and the associated distress (fatigue cracking, rutting, thermal cracking, etc.) that result from use. Functional

performance is strictly confined to how well the pavement serves the user and typically centers on a driver's level of comfort while in vehicles.

This performance concept formulates the basis for the estimating the life of a pavement and the serviceability of a pavement versus time can be used to measure performance. The unit of measure for performance by AASHTO is given as the present serviceability index (PSI) which is a function of pavement distress and roughness, the measurement of which is dominated by roughness. A pavement begins its life with a measured PSI value, which decreases with time as the pavement undergoes distress from repeated loading. Over time, the pavement will have a PSI value lower than an acceptable limit, at which point the life of the pavement has been consumed.

One important product of the AASHO Road Test was the development of axle load factors and the effect that traffic has on pavement performance and its associated deterioration. The AASHO experiment compared the damaging effects that different axle loads have on the pavement and the individual damage relative to a standard 18-kip equivalent single axle load (ESAL). Equivalent axle load factors (EALF) were developed to convert the incremental damage from any axle repetition into the equivalent number of ESALs to cause the same amount of damage.

For instance, AASHTO recommends an EALF = 0.26 for a 13-kip axle load (assuming a referenced pavement strength and terminal serviceability), which roughly implies, 4 repetitions of a 13 kip axle load equates to the same damage caused by one repetition of an 18 kip single axle load (AASHTO, 1993). These load factors have been developed for single, tandem, and tridem axle groups for various load magnitudes. The traffic that the pavement will endure is then broken down into the various load groups

and converted to an estimated number of ESALs for the design period. Pavement fatigue models are then used to estimate the number of repetitions to failure and the two values are compared.

The road bed material properties are considered in flexible pavement design through the use of the resilient modulus, M_R, of the soil. The resilient modulus of a soil is similar in concept to the Young's modulus of elasticity. However, during testing of the soil, the specimen is subjected to a range of repeated compressive loads under varying confining stress loads. The resilient modulus is defined as the stiffness of the material after a predefined number of conditioning load repetitions and is the ratio of the deviator stress to the recoverable strain of the soil.

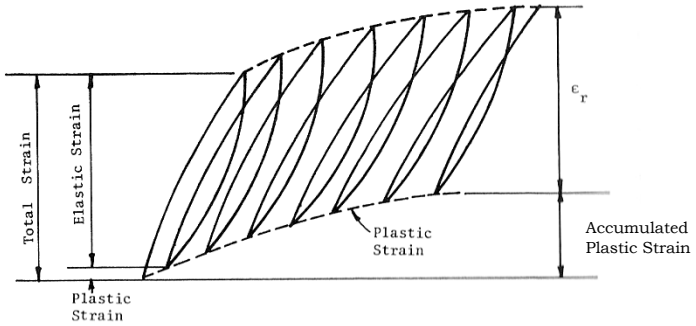


Figure 2-1 - Recoverable strain during resilient modulus testing.

$$M_R = \frac{\sigma_d}{\varepsilon_r} \qquad \textbf{(2-1)}$$

where M_R = resilient modulus, psi

σ_d = deviator stress, psi

ε_r = recoverable strain, in/in

In the 1993 Design Guide, the resilient modulus of the road bed materials is used directly in the design computations. Additionally, M_R values may fluctuate with environmental conditions, and average seasonally adjusted values are typically used.

In addition to the road bed (native) materials described above, the mechanical properties of the sub-base, base, and surface layers of the pavement structure are needed for design. In general, the bound and unbound materials are characterized by their M_R values, which are converted to layer coefficients used to calculate a Structural Number (SN) for the particular pavement design. The SN is an index that is used to determine the total depth of the pavement structure and is based on assigned layer strength coefficients and thicknesses. The SN is used directly in computations to calculate the allowable load repetitions for the pavement.

Two major environmental factors affect the material properties of the pavement structure; temperature and moisture. In regards to bituminous pavements, the stiffness of the bound layers decreases as temperature increases, resulting in higher stresses being imparted to the layers below. At lower temperatures, the stiffness of bituminous materials increases resulting in lower stresses imparted to the layers below, but the material also has a tendency to crack. In addition to this, the continuous temperature fluctuations can lead to thermal cracking from expansion and contraction.

The performance of native and other unbound layers in a pavement structure may be susceptible to moisture or a combination of both moisture and temperature changes. Highly plastic soils, such as clays, can be significantly weakened (reducing stiffness values) in the presence of high moisture contents. Under freezing conditions with the presence of moisture, soils can become frozen and can have large increases in stiffness,

while some other soils can also be classified as “frost susceptible”. During periods of below freezing temperatures, ice lenses may form causing these soils to deform leading to heaving and frost weakening.

Wind, solar radiation, ambient air temperature, rainfall amount, etc. are all contributors to the particular effect on a pavement’s structural performance. In the design of the pavement structure, these environmental effects are typically accounted for by creating a realistic variation in the stiffnesses of the pavement layers throughout the seasonal analysis periods.

Precipitation is mitigated by designing drainage for the pavement system and is very important in reducing the effect that moisture may have on the pavement materials. In design, drainage factors are applied when computing the SN for a given pavement structure, but these are based heavily on empiricism.

In addition, shoulders have been shown to increase the performance of certain pavement designs by reducing moisture intrusion and providing support of lateral movement of the other layers of the pavement structure. The 1993 Design Guide makes no provisions for the benefits of shoulders, but recommends including beneficial effects based on field observations and experience. A guide for the design of shoulders is included however.

Encompassing all of these design variables, is life cycle cost and reliability – two important aspects of pavement design. As costs rise, the stakes are higher for governmental agencies to provide suitable facilities that provide the best performance. Reliability is used to predict the performance of a design which rest on the accurately predicted traffic volumes, material properties, and environmental conditions. The 1993

Design Guide provides provisions for including reliability into the pavement design process, while, with respect to life cycle costing, it is left to the engineer to predict the construction and future maintenance costs.

2.2 - Full Scale Experiments

Numerous full scale experiments have been executed to try to represent the pavement responses under "real world" conditions (Timm et al., 2004). Some of the very first experiments produced the empirical data that some design guides were based upon. However, as pavement behaviors were better understood and as the technology to measure these pavement behaviors became more efficient and economical, these experiments have been repeated in greater detail and scope. The research, which is the basis of this work, is an example of this type of experiment.

2.2.1 - NCAT Structural Experiment

The National Center for Asphalt Technology (NCAT) at Auburn University test track was started in 2000 and has produced excellent research concerning asphalt technology. The track consisted of forty-five flexible pavement test sections, each 200 feet long, and was continuously loaded with FHWA Class 9 vehicles with controlled axle configurations and weight. The trucks were used for the sole purpose of applying load repetitions to the pavement and were driven on the track for eighteen hours a day. The test track was an example of an accelerated performance testing facility (consuming 10 to 15 years of design life in 2 years). Within the numerous test sections are a variety of

different research activities (Brown et al., 2002) (Freeman et al., 2001) (Timm & Priest, 2004)

In 2004, sensors were installed in eight sections of the NCAT test track for measuring dynamic pavement responses. The eight sections selected were constructed of asphalt with varying structures and asphalt mix designs. Asphalt strain gauges were installed as the primary source of data for pavement analysis. Along with these instruments, earth pressure cells (of two different types), vertical compression gauges, soil moisture (time domain reflectometry) probes, and temperature probes were installed to provide supplemental data.

The installation of the sensors was a success, with only a few gauges not surviving the installation. It was suspected that failure was due to damage to the sensor leads during HMA compaction. Low speed data was recorded for the environmental sensors such as temperature and soil moisture. The outputs of strain sensors and earth pressure cells were recorded at high speed under traffic from the calibrated test vehicles. The data was analyzed in a piecewise manner, taking the information that was considered most crucial.

The information taken from the study was used to calibrate the pavement design processes to the local variables. The stated objectives of this particular research were to validate mechanistic pavement models, develop transfer functions for typical asphalt mixtures and pavement cross-sections, study the dynamic effects on pavement deterioration, and evaluate the effect of layer thickness and polymer modification on structural performance. One of the most important results from the project was the

calibration of pavement design parameters for local conditions (Priest & Timm, 2006) (Priest, 2005) (Timm & Priest, 2004).

Wheel wander is a phenomenon that takes place on virtually every roadway and is defined by the natural meandering of vehicles transversely across the pavement lane. This occurs naturally since a driver has a limited ability to drive in a straight line given the natural conditions of the environment such as pavement cross-slope, smoothness, wind direction and speed, vertical and horizontal alignment, etc. This traffic wander is important to pavement analysis because it affects how much damage the pavement undergoes at one particular location. Evidence for the magnitude of this effect was found by researchers studying pavement fatigue under heavy load simulators in which the pavement life was significantly shorter when the load was confined to one path (Buiter et al., 1989) (Timm & Priest, 2005).

The researchers at the NCAT test track found the wheel wander to be normally distributed with standard deviations between 7.6 and 8.0 inches. These results were found to agree with past research (Buiter et al., 1989) (Timm & Priest, 2005). The NCAT researchers found a strong relationship between lateral offset and peak tensile measurements, where greater offsets values resulted in lower strain.

Strain measurements in the NCAT test were consistent with those obtained in other past research. The typical strain responses at the bottom of the pavement due to a moving vehicle loads was noted. For a sensor installed to measure longitudinal strain, this almost always consisted of a compressive strain followed a large tensile strain followed by a smaller compressive strain. For the case of a sensor installed to measure transverse strain, the measurements almost always showed a tensile strain with no

compressive strains preceding or following the peak tensile strain (Timm et al., 2006) (Priest, 2005). These observations coupled with peak strain data were used to support theories on modes of fatigue failure.

2.2.2 - MnROAD

The MnROAD program was sponsored by the Minnesota Department of Transportation and carried out by researchers at the University of Minnesota. The program involved studying behavior at both a test track for controlled loading and a portion of Interstate 94 for loading under real traffic conditions. The focus of the project was very broad and covered many aspects of pavement and highway design. Of interest to this research was the work done to better understand the structural response of different flexible pavements. Some outcomes of the project included calibrating pavement models to the local conditions in the region, thus improving the accuracy of their pavement design procedures. The research also helped shape a mechanistic-empirical design process (Bao, 2000).

To measure the structural response of both PCC and HMA pavements, over 4,500 sensors were installed into the pavement structures. Of these, 1,151 were used to measure dynamic pavement response. Amongst the numerous sensors were asphalt strain sensors and earth pressure cells. These two sensor types were the main resource for acquiring the dynamic load response of the pavements. Many of the other sensors used were focused on acquiring environmental information for the supporting layers below the asphalt, such as temperature, moisture content, and pore water pressure (Baker et al., 2002) (Strommen, 2002).

The strain sensors were arranged in groups of three and spanned across a wheel-path. Some were placed to measure transverse strain while others were placed to measure longitudinal strain, although no reasons were given for why an orientation was used. Previous pilot studies had been carried out but were mainly focused on the type of instruments utilized and not necessarily with the location and arrangement patterns of the sensors.

Optim Electronics MEGADAC data acquisition systems were used to collect the data. Acquisition was done at set time intervals and not necessarily taken continuously (Koubaa & Stolarski, 2002) (Worel, 2006) (Lau & Alouini, 2002).

Researchers did note during the project that numerous sensors eventually failed, crippling the effort. They also reported that they needed more data consisting of additional axle configurations to use in creating and calibrating models. Work is ongoing at the MnROAD project site, but research regarding structural response has ended.

2.2.3 - AASHO Road Test

Shortly after the end of both World Wars, the number of vehicles on the roadway quickly increased. It has estimated that the number of registered vehicles, both automobiles and trucks, tripled between the years of 1919 and 1929 and doubled between 1945 and 1955 (Highway Research Board, 1961). The increased traffic during these post-war periods started to wear on roadways and very little maintenance was being done during war-time to maintain facilities.

In addition to this increase in traffic volume, the roadways were not designed with heavy loads in mind – most traffic consisted of lighter vehicles. However, transporting

cargo over the road was becoming a popular method of moving goods, which meant the roads sustained more stressful loads. In 1930 only 35 states enforced a weight limit on axle loads and by 1932 it was recommended by the American Association of State Highway Officials that all states should adopt a weight limit of 16,000 lb per axle for vehicles with high pressure tires. It was estimated that there were only about 13 axles per 1000 vehicles that weighed over 18,000 lbs between 1936-37 and 86 axles per 1000 vehicles in 1945. Coupled with the fact that the number of trucks on the road had nearly doubled in the same time period, the roadways experienced significant distress.

It was obvious that a better design method was needed to construct quality pavements, and regulation needed to be in place to limit loads. One interesting problem that was faced was how to select an optimum vehicle size and complimentary pavement and bridge structures. An economic study was conducted by collaborative efforts between vehicle manufacturers and engineers to find an optimum vehicle size. In the end, experiments were needed to generate the necessary pavement performance information.

The first test was known as Road Test One-MD and was focused on applying controlled loads to rigid pavements that were representative of the majority of existing pavements. This test would serve as the first basis for determining weight limits and vehicle size. Following Road Test One-MD was another experiment known as the WASHO Road Test which shared similar objectives to Road Test One-MD but focused on analyzing flexible pavement types.

Ultimately more tests were planned, but the scope of the proposed plan grew. A larger spectrum of loads was needed to understand the effects of both light and heavier

loads. In addition, the number of cross sections needed to be expanded to cover designs typically constructed across the country. The culmination of the recent findings and the interest in expanded road testing led to the development of the AASHO Road Test. The basic experimental plan was presented in 1952, and by early 1956 the first contracts for structural steel were awarded.

The plan included the construction of 6 traffic loops outside Ottawa, Illinois. Four of the loops were loaded with heavier tractor-trailer trucks, one loop was loaded with light truck traffic and the last loop was used for static, creep speed (~2 MPH), and vibratory loads. The last loop was meant to be relatively unloaded and serve as a control to observe the effects of environment alone.

The technical objectives of the test, as stated by the National Advisory Committee (Highway Research Board, 1961) were the following:

1. To determine the relationship between the number of various axle loads and groups on pavement performance.
2. To determine significant affects of various axle loads and configurations on bridges.
3. To address special interests such as the effects of shoulders, base materials, different tire types/pressures, military loading, etc.
4. Document all maintenance activities needed to keep the pavement in satisfactory condition.
5. Provide useful information in evaluation of existing facilities, design of new facilities, and future research through the development of instrumentation, test procedures, data, etc.

The result of the research project was a series of reports published after the close of the planned test traffic. The compiled data would eventually lead to design practices and recommendations for pavement designs, published in 1961 as the *Interim Guide for the Design of Pavement Structures*. Since data comparable to that of the Road Test was not available from any other sources, the design recommendations were heavily based on results of the experiment. Eventually the interim guide underwent minor revisions and slowly evolved to the 1993 version of the pavement design guide.

2.2.4 - Kansas Experiment

The state of Kansas along with researchers from Kansas State University initiated a research project in 2005 to measure dynamic structural responses of four different perpetual pavements.

The research involved designing four separate pavement cross sections and designing an instrumentation plan for each of the test sections. The objectives of the experiment were to validate the endurance limit of the pavements, evaluate the cost-effectiveness, and finally to compare measured horizontal strains in bottom of the asphalt layer with those calculated analytically from linear elastic models (Romanoschi et al., 2006) (Romanoschi et al., 2008).

The instrumentation included Texas Measurements (model PML-120-2L, field modified with additional anchors) strain sensors, Geokon earth pressure cells, and temperature sensors to monitor asphalt temperatures. The test sections were loaded with a test vehicle on multiple occasions. Eight of the sensors contained in one of the test sections were destroyed during the construction of the pavement and were later

retrofitted. The retrofit was done by removing a 12 in diameter core, applying a strain gauge to the underside of the core and then reinstalling the core in to the pavement using epoxy. The various instruments were recorded with a National Instruments data acquisition system and the recorded strain signals were analyzed with the aid of a spreadsheet program.

The test sections were loaded with a single axle dump truck (FHWA Class 5) loaded with material and weighed on a static scale (11,000 lbs – front axle, avg. and 19,000 lbs – rear axle, avg.). In addition to this test vehicle, a rolling wheel deflectometer (RWD) was used to stimulate the gauges and measure deflections. The instruments were sampled at 3 kHz while only storing the average of ten measurements.

They found that strains measured in the same section varied as much 30 to 60%. They attributed this to construction and dynamic loading effects. They also report that measured strains were highly influenced by vehicle speed, with much higher strains induced at slower speeds. However, the effect due to increasing speeds was reduced as speeds increased. Three different speed regimes were used during the tests: 20-25 mph, 40-45 mph, and 55-60 mph.

The authors also noted that strains were highest for the thinnest pavement design and steering axle strains were measured at about 50 to 70 % of the measured rear axle strains. For a single axle with dual tires, the transverse strains were always larger than the corresponding longitudinal strains for all cases (temperature, speed, and pavement cross section). Linear elastic analysis of the test sections with the same loading scenario resulted in the calculated longitudinal strains being larger than the transverse, contradicting what was measured in the field. The authors note that this observation may

be due to the chosen instrumented sections, which are close to the shoulder of the roadway.

As expected, higher strains were recorded during the warmer months, confirming that pavement temperature has a significant effect on the pavement response. Measured strains under dual tire loading, also show that transverse strains were typically larger than the longitudinal strains. The authors attribute this to the pavement not recovering completely after the first axle passes over the gauge.

Everstress (an FE pavement analysis software package) model results for the pavements showed that the measured transverse strain were almost twice as large as those calculated by Everstress. In addition, measured longitudinal strains were about half of the computed strains and the measured vertical stress about one-third of that calculated.

2.3 - Fatigue Testing

During the late 1950's, a great deal of research was being conducted on the fatigue behavior of metals, and the science was quickly extended to asphalt. Continuum damage mechanics is a study devoted to the accumulation of "damage" a material undergoes during fatigue. Most notably for the purposes of HMA materials, damage accumulation in regards to fatigue failure is of most interest, and the most widely accepted concept was proposed by M.A. Miner (Miner, 1945). Miner's Rule allows the fraction of damage in a material to be calculated as the ratio of the number of applied load repetitions to the amount of allowable load repetitions (allowable in regards to failure) and is stated mathematically in Equation 2-2.

$$D = \sum_{i=1}^{k} \frac{n_i}{N_i} \qquad \textbf{(2-2)}$$

where D = damage

k = number of different stress levels

n_i = actual number of load repetitions for stress level k

N_i = allowable number of load repetitions at stress level k

As an example of the rule, suppose an A36 steel fatigue specimen is subject to tensile fatigue testing at some level of constant stress amplitude. At the beginning of the test, the damage is 0 and at the end of the fatigue test (that is, at failure), the damage is 1 (100%). The damage is often indicated by the elements decrease in stiffness throughout each loading cycle – implying that small cracks have formed and are growing (the decrease in stiffness only occurring after cracking has started), effectively decreasing the section of the specimen. In this particular case, failure of the specimen could be represented by the complete fracture through the specimen – damage at the failure state would be 100%.

It is important to remember that higher stress levels during a fatigue test will cause the allowable number of load repetitions to decrease and the opposite will happen for lower stress levels. It also follows from this that at higher stress levels, the unit damage per cycle is greater than that at lower stress levels.

Many mechanistic-empirical design procedures today apply Miner's hypothesis to estimate the accumulated fatigue damage for pavements. The application of this damage theory resulted in a beam fatigue test for compacted HMA specimens, pioneered by Monismith and Deacon (Deacon, 1965) (Tayebali et al., 1996) (Monismith & Deacon,

1969). The testing apparatus was designed to continuously and consistently bend a prism of compacted HMA cut to the required dimensions. The deflections of the beam are measured during testing and the stiffness of the beam is then back-calculated using the deflection and loading. Failure of the HMA specimen is typically considered when the stiffness has reach 50% of its initial stiffness.

Fatigue testing of many different HMA specimens has resulted in a handful of transfer functions which relate a level of strain to the allowable number of load repetitions. The most popular were developed by Shell Oil (Shell, 1978) and the Asphalt Institute (Asphalt Institute, 1991), however an updated model has been given in the new Mechanistic Empirical Pavement Design Guide (MEPDG) and is based on a National Cooperative Highway Research Program (NCHRP) study (Witczak et al., 2003). All models follow the general form shown below in Equation 2-3.

$$N_f = Ck_1\left(\frac{1}{\varepsilon_t}\right)^{k_2}\left(\frac{1}{E}\right)^{k_3} \quad \textbf{(2-3)}$$

Where:
- N_f = number of load repetitions to fatigue cracking
- ε_t = tensile strain at the critical location
- E = dynamic modulus of the material
- k_1, k_2, k_3 = laboratory regression coefficients
- C = laboratory to field adjustment factor

The testing protocol recommends a specific loading sequence, but there has been much discussion on the effects of different loading scenarios. A load cycle consists of a steadily increasing load up to the maximum followed by a subsequent release which is

finally followed by a short rest period (load frequency of 1-2 Hz with a 0.1 second load time).

Deacon conducted extensive fatigue testing of HMA specimens using the developed fatigue equipment and found that under a controlled-stress mode of testing, there was a linear relationship between the logarithm of the mean fatigue life and the logarithm of the stress level.

It was also observed that the rate of loading on the specimens had a profound effect – faster loading rates were associated with a shorter fatigue life. Similarly, longer load durations were shown to shorten fatigue life. Longer load durations also resulted in lower stiffnesses (Deacon, 1965). Deacon postulated that the two aforementioned observations can be attributed to the visco-elastic nature of HMA materials.

In addition to these, Deacon observed that, for his specimens, a larger initial stiffness modulus resulted in a longer fatigue life. A higher specific gravity also indicated a longer fatigue life. Ultimately the testing conducted by Deacon indicated that the most general damage determinant was the initial maximum principal tensile strain in the bituminous binder.

However, Deacon's initial testing was done at a rather low-cycle fatigue and very high-cycle fatigue had yet to be studied at the time. High-cycle fatigue has been stated as being probably the most difficult phenomenon to study within solid mechanics. This consequence results in many failures caused by high-cycle fatigue – the difficulty in studying stemming from the very small and hard to measure micro- or nanoscale defects initiating much below the engineering yield stress (Lemaitre & Desmorat, 2005). It has also been noted that for high-cycle fatigue, the experimental data always possesses a

large amount of scatter and that a factor of 10 on the number of cycles to rupture on similar tests is normal. Furthermore, a probability analysis should be used where possible when studying high-cycle fatigue in order to integrate reliability into design (Lemaitre & Desmorat, 2005).

2.3.1 - Endurance Limit

With the increasing emphasis on extended life HMA pavements, also called perpetual pavements, the existence of a fatigue endurance limit (FEL) has been postulated to exist for HMA materials, and initial data supports this theory. The FEL follows from observations that fatigue behavior at low strain levels does not follow the same relationship as that of the materials subjected to "normal" stress levels, and that there exist a strain limit for HMA materials below which no damage occurs (Carpenter et al., 2003).

This concept was originally developed for metals where below a particular stress amplitude level, the plot of a stress amplitude versus cycles to failure plot became a horizontal line, indicating an infinite fatigue life. Monismith and McLean first proposed the FEL for HMA materials to be around 70 microstrain for typical HMA mixtures.

The significance of the FEL for HMA materials would be such that a pavement structure thickness design would depend only on limiting the maximum tensile strain to a value at or below the FEL strain. In addition to this, the pavement structure would require little or no consideration of a traffic analysis. A minimum thickness to limit the strain below the FEL would be all that is necessary. There is no longer a need estimate

consumed fatigue life because the life is essentially infinite (for the bottom-up fatigue mode of failure that is) (Carpenter et al., 2003).

Fatigue testing conducted by Carpenter et al., has clearly established the existence of the FEL (Carpenter et al., 2003). A survival analysis conducted using the Long Term Pavement Performance database indicated that the FEL theory appears to be valid and that the FEL is a property of the HMA material (Von Quintus, 2006). Another study by Prowell et al. (2006) suggests that fatigue testing should be conducted in excess of 10 million cycles to accurately determine the FEL strain limit. They also suggest a FEL strain limit of approximately 100 microstrain. Different modeling techniques have been applied in order to create predictive tools based off of conducting far shorter fatigue test durations (Prowell & Brown, 2006).

2.3.2 - Dissipated Energy Theory

As a further look into finding a unifying approach to characterize HMA materials, researchers began studying the energy dissipated during various load cycles of the traditional fatigue test. It was hypothesized that understanding the nature in which energy was dissipated would allow for a unifying characteristic of the accumulation of damage. After all, it is understood that the hysteresis curve observed in stress-strain plots of a material being loaded and unload represents a loss of energy.

Working off of previous research, Carpenter et al. began studying the dissipated energy during a traditional fatigue test by measuring the stress and strain in the sample during the test (Ghuzlan & Carpenter, 2000) (Ghuzlan, 2001).

The researchers defined the dissipated energy as the area of the hysteresis loop formed during the repeated loading cycles – the area of which represents energy that has been dissipated by the accumulation of damage within the material, heat generation, etc. The researchers posit that no damage occurs as long as the area and shape of the stress-strain hysteresis loop is the same. The supporting reason is that the area below the stress-strain path represents the structural integrity of the sample – if the area and shape remain identical then the structural integrity is not lost. However, when the total amount of structural integrity is changed, then damage has been accumulated (Ghuzlan, 2001) (Ghuzlan & Carpenter, 2000).

The unifying law is then based on calculating the rate of change of the dissipated strain energy between each loading cycle. They found an interesting behavior, which is defined in three different stages of the test specimen. During the initial stage, there is a very rapid decrease in the rate of change of this strain energy release (the strain energy is actually being absorbed by the material). The second stage is a very steady stage where the rate of change is nearly constant (which they refer to as the plateau stage). The third stage is characterized by a sudden increase in the rate of change up until the final failure of the specimen.

Many different beams were fatigued tested at different stress and strain levels in both stress and strain controlled fatigue tests. They found an excellent relationship between the plateau value (of the ratio of dissipated energy) and the number of cycles to failure (i.e. when stiffness was reduced by 50%). They extended this to both normal and low strain-damage levels. This relationship was independent of test conditions and

mixtures and supports the idea that the energy dissipating mechanism is unique and defined (Carpenter et al., 2003).

The original concept of the fatigue test was to measure the stiffness of the material for every load repetition done in similar fashion for fatigue of metals. Classical damage mechanics uses this approach to measure the extent of damage being accumulated in the material. As the material undergoes load repetitions, micro-fractures develop in the material which grow larger with each load cycle. The lost of contact of this material leads to a reduction in stiffness. The severity and quantity of these small fractures can be measured on the macro-level scale by computing the stiffness of the test specimen.

However, at low strain levels, the researchers noticed that a non-linearity was present in the relationship between the rate of damage accumulation and strain level during testing. Testing at low strain levels (below ~100 microstrain), the amount of damage done to the sample changes drastically and becomes a minimum. This nonlinearity and breakpoint further suggest the existence of the FEL. However, a small amount of damage is still accumulated, and it has been proposed that healing is the source of the FEL (Carpenter et al., 2003).

Healing has been described as a continual process by which there is a return of structural integrity back into the material, repairing the damage done and restoring its load carrying ability. This process is most evident when there is a rest period being repeated loading. This process provides the connection between the dissipated strain energy concept and the FEL.

An HMA specimen possesses a certain amount of healing potential, thus a strain level would exist that the damage being done to the sample is equal to the healing potential of the specimen, with the net outcome that no damage has accumulated. At or below this tensile strain threshold, the pavement would have an essentially infinite life (Carpenter et al., 2003) (Shen & Carpenter, 2005).

A related study on HMA with rich bottom bases (RBB) showed that the addition of 0.5% binder over the control mix did increase the initial stiffness of the specimen (Carpenter & Shen, 2006). It was proposed that the additional binder content in the mix would increase the fatigue resistance. Contrary to this, the study showed that the additional binder content had little effect on the FEL and supports a conclusion that healing potential is a property of the specific binder and that different HMA mixtures using the same binder should exhibit similar FEL strain levels.

2.4 - Advances in Modeling Techniques

Modeling of asphalt pavements has been a focal point in order to properly create mathematical models to accurately predict behavior and ultimately pavement life. In the past the modeling process was sometimes considered ineffective due to the long computation times necessary to make accurate predictions. Faster computers have finally brought these numerical analysis solutions to the forefront and integrated into pavement analysis.

The boundary element method (BEM), element free Galerkin, and the layered elastic analysis (LEA) have all been applied to represent pavement structures. The finite

element method (FEM) has become prominent because of its adaptability, versatility and popularity amongst other disciplines (which lends to the availability of software).

In general, three different modeling approaches have been used to model pavements: 2 dimensional, 2-dimensional axi-symmetric (3-D solution), and a full 3 dimensional analysis. Within these three classifications, use of different material models have been applied to extend the general first order, linear elastic analysis to a more refined analysis type such as non-linear material models (which could include stress dependencies).

One such study, by Cho et al. (1996), aimed to determine which of these popular modeling techniques was the most appropriate. In their research they directly compared different FEM models for a given pavement structure. In their study they also varied the aspect ratio and size of each individual element. Their work consisted of using a 2-D plane strain, 2-D axi-symmetric, and 3-D model. They ran the models and compared them with solutions from BISAR (BItumen Stress Analysis in Roads), a LEA program develop for pavement analysis.

The pavement analyzed consisted of an infinitely thick subgrade overlain by 8 inches of PCC with a 4 inch HMA overlay. The pavement was subjected to a 9000 lbs wheel load placed on circular area with a 5.9 inch radius (approximately 80 psi). The models were compared based upon the maximum deflection of the pavement structure and the maximum tensile stress in the PCC pavement. All analyses were carried out on a Cray supercomputer. They found that in general the 2-D model appeared to be too stiff, however, the load was applied to a single node and was not representative. The axi-symmetric model performed extremely well when elements were about 1 inch by 1 inch.

The applied load was circular with a uniform contact pressure, which is consistent with actual conditions. The 3-D dimensional solution was actually carried out in half- and quarter-symmetric fashion due to limits of the FE software (ABAQUS) with load applied at a single point. They found that the relative magnitudes of stresses and displacements to be inconsistent (i.e. stresses too high with small displacements or vice-versa). They concluded that this was likely due to the application of the point load and a refined load distribution was developed and applied to a rectangular area. This improved load application generated results closer to a reasonable solution, but still the relative magnitudes of stresses and deflections were somewhat unreasonable, leading to the conclusion that much more consideration must be given to proper load modeling.

The authors ultimately concluded that the axi-symmetric model was adequate given small enough element sizes and that the 3-D model could be adequate, but would require more consideration when designing the model.

Chapter 3 - Research Method

This research is concerned with studying a section of HMA pavement located on an urban highway. More definitively, the pavement structure was designed to meet the requirements of a perpetual pavement, also called an extended life pavement, and the focus of this research centers on studying the structural response to traffic loads.

An intricate system of state-of-the-art measurement devices and computer systems were combined to accurately measure the pavement response due to live traffic loads. This project is unique in that responses from live traffic are recorded under real conditions, contrary to the controlled conditions that past research projects have studied.

This chapter highlights the important factors in the performance and life of an HMA pavement and illustrates the experimental procedures and equipment.

3.1 - HMA Performance Variables

Many different factors affect the performance of any pavement system. The most important of these features have been accounted for in most design procedures such as those accounted for in the AASHTO 1993 design guide and new Mechanistic-Empirical Pavement Design Guide (MEPDG). It only follows that a thorough analysis of a pavement system should include information, or data, on these variables.

These variables can be viewed in four general subgroups; traffic, pavement design, environmental, and material variables. These four categories act together to form the complex problem of accurately predicting pavement performance.

The amount of traffic can vary greatly on an hourly, daily, weekly, monthly and yearly basis for a given facility. For instance, rush hour traffic patterns can increase the

frequency of loads, but they can also slow traffic down, creating loads that have much longer durations. At other hours it may be the case where there are periods of heavy truck traffic, and yet other instances when traffic is reduced and consists of light, high velocity passenger vehicles.

The traffic variable can also include the type of vehicles making up the total amount of traffic and can be used to describe the spectrum of individual loads applied to the pavement system. Heavier loads stress the pavement system much more than lighter vehicles, and the higher stresses lead to a higher rate of damage to the pavement and hence the need to accurately characterize the traffic.

Material properties logically play a large role in the performance of a pavement system. It is necessary to understand and measure material properties to perform an analysis. Just as Young's modulus is crucial for the analysis of reinforced concrete, so are the stiffness properties for HMA or sub-grade materials.

Coupled with the material properties, is the design of the pavement system. In order to analyze a pavement system, a thorough knowledge of the physical dimensions must be known, such as layer thicknesses, lanes widths, etc. In addition, other physical boundaries or discontinuities, such as construction joints, may be required for a thorough analysis to model stress concentrations.

Environmental variables can influence the performance of a pavement by altering the other aforementioned properties. Environmental factors heavily influence the material properties of HMA and exclusion of these factors in the analysis would be an oversight. HMA materials are temperature dependent with high temperatures resulting in lower stiffness values and low temperatures causing stiffness to increase. In addition,

moisture in the unbound material layers can have a profound effect on the stiffness of those materials, especially fine grained silts and clays.

This research was concerned with measurement of all these variables as well as measurement of the pavement responses. This collected data could then be used to perform a thorough analysis of the pavement system. Past research projects have shared goals similar to this project, and a thorough review helped guide the development of the experiment – the details of which are defined in the following sections.

3.2 - Bottom-Up Fatigue HMA pavement

Of particular interest to this research is the progression of bottom-up fatigue cracking. This mode of failure has been prevalent in the past and represents one of the major forms of structural distress for HMA pavements. This mode of failure has been characterized by a crack initiation at the bottom of HMA layers with subsequent stress cycles causing the crack to grow and eventually propagate to the surface. The crack results in a loss of section, causing other distresses to accumulate much faster. In addition, the crack allows surface moisture to penetrate into the subbase and subgrade materials, which may cause a weakening of those layers leading to more distress.

The means to characterize the fatigue behavior of HMA has been on-going as discussed in Chapter 2. Many variations of a fatigue test have been developed, but Deacon and Monismith pioneered the development and analysis of a fatigue test that has gained the most attention (Monismith & Deacon, 1969). The test consists of bending a rectangular prism of compacted HMA until failure in either a controlled stress or controlled strain mode. The load is typically applied at third points along the length of

the beam with pinned supports at each end (four point bending), causing a region of constant moment at the center of the beam. The load is applied for a duration of 0.1 seconds followed by a 0.9 second rest period (rest periods can vary based on the frequency of testing), although many other loading variations have been applied in other studies. Upon release of the load, the specimen is typically pushed back to its position before the load cycle re-starts. Measurements during the test include the applied load, deflection, and strain at the center of the beam.

Failure criteria for laboratory fatigue testing typically are based on two different testing scenarios. Constant stress type testing is typically used for testing pavements where the HMA thickness is greater than 6 inches. Failure of the sample is clearly defined by complete cracking of the specimen. Constant strain type testing is typically applicable to relatively thin pavements when the thickness is less than 2 inches. Failure in these types of tests is not well defined, so it is common to assume failure when the stress has reached 50% of the initial stress (Huang, 1993). Fatigue testing for pavement thicknesses between 2 and 6 inches demonstrate characteristics of the former and latter, however the criteria for failure can be left to the engineer. It is common to use the 50% reduction in stress for failure criteria, especially when performing a high-cycle fatigue test.

Many fatigue tests were conducted on various HMA mixtures, and the analysis of the data showed that the maximum tensile strain in the HMA specimens was the best predictor of fatigue life (Deacon, 1965). In the fatigue of metals, fatigue behavior is often illustrated by plotting stress amplitude versus the number of cycles to failure, with the plot being referred to as an S-N curve. The same can be done for HMA fatigue

testing; however, controlled strain tests usually require stress to be replaced by the strain level at testing. What was found is that the relationship can be nearly approximated by a straight line on a log-log plot of the data as shown below in Figure 3-1.

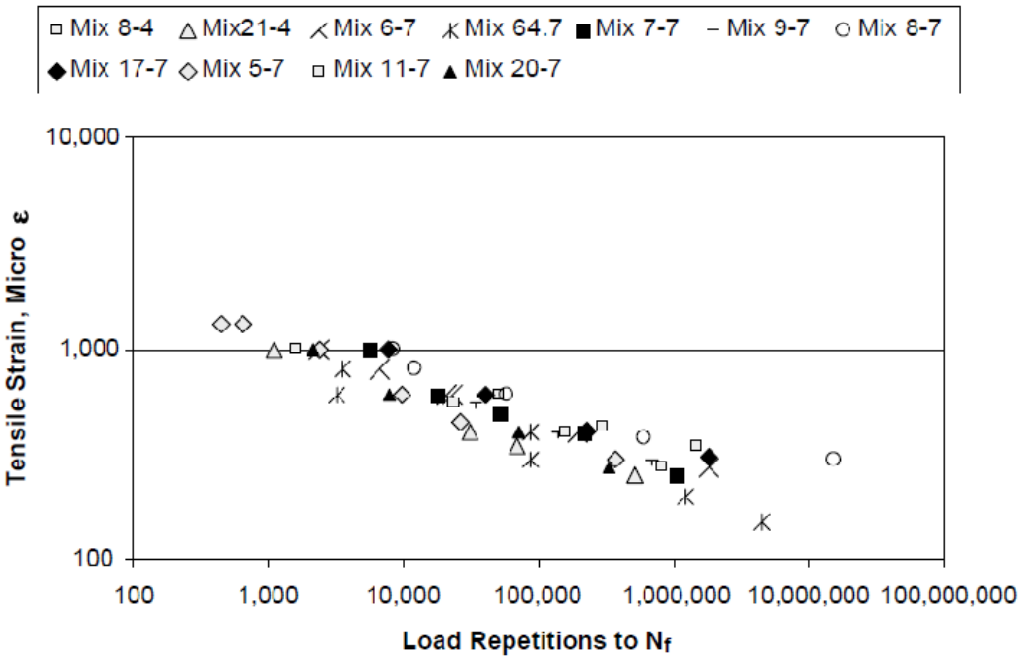


Figure 3-1 - HMA fatigue S-N curve.

The regression equation of this relationship is sometimes referred to as a transfer function because it transfers the strain level in the HMA to the number of repetitions to failure. The form of this relationship has been given in the previous chapter, but bears repeating.

$$N_f = Ck_1\left(\frac{1}{\varepsilon_t}\right)^{k_2}\left(\frac{1}{E}\right)^{k_3} \qquad \textbf{(3-1)}$$

where N_f = number of load repetitions to fatigue cracking

ε_t = tensile strain at the critical location

E = stiffness of the material

k_1, k_2, k_3 = laboratory regression coefficients

C = laboratory to field adjustment factor

This relationship by itself does not allow the fatigue life of pavement to be predicted, but Miner's principle allows the damage to be calculated. Because of the nature of the traffic distributions and the associated loads of each, the induced strain in the pavement is not always the same (and also contains a significant amount of variability). Hence it is necessary to calculate each load range and the associated fraction of fatigue damage, summing up the total damage over time.

For each load range, there is some strain induced in the pavement structure which correlates to an allowable cycles to failure of the HMA. Smaller strains, lead to larger fatigue lives, and vice versa for larger loads. One load repetition at some tensile strain level, ε_t, causes N_f^{-1} amount of damage, where N_f is the allowable number of load repetitions at the particular strain level. Summing up all load repetitions for all strain level gives the predicted amount of fatigue damage. When the ratio of applied repetitions to the allowable repetitions approaches unity, the defined threshold for considering failure is approached (e.g. 50% cracking of wheel path). The preceding explanation is mathematically expressed below in Equation 3-2.

$$D = \sum_{i=1}^{k} \frac{n_i}{N_i} \qquad \textbf{(3-2)}$$

where D = damage

k = number of different stress levels

n_i = actual number of load repetitions for stress level k

N_i = allowable number of load repetitions at stress level k

3.3 - Current Fatigue Models

Fatigue testing by a number of researchers has produced a handful of different of fatigue models. Although none of them can be universally applied to any pavement, their development has been instrumental in developing the latest model used by the 2002 Design Guide calibrated to historical data.

The Shell Oil model was developed covering a wide range of HMA mixes, binders, and testing conditions. Two models were recommended for both constant strain and constant stress laboratory fatigue testing (Bonnaure et al., 1980). The models are given below in Equation 3-3 and Equation 3-4.

Constant stress

$$N_f = \left[0.0252\ PI - 0.00126\ PI \times V_b + 0.00673\ V_b - 0.0167\right]^5 \left(\frac{1}{\varepsilon_t}\right)^5 \left(\frac{1}{E}\right)^{1.4} \qquad \textbf{(3-3)}$$

Constant strain

$$N_f = \left[0.17\,PI - 0.0085\,PI \times V_b + 0.0454\,V_b - 0.112\right]^5 \left(\frac{1}{\varepsilon_t}\right)^5 \left(\frac{1}{E}\right)^{1.8} \quad \textbf{(3-4)}$$

where N_f = number of repetitions to failure

ε_t = tensile strain at the critical location

E = stiffness of the material

V_b = percentage of binder in mix by volume

PI = penetration index of binder

The Asphalt Institute also developed a similar model which was based on laboratory data produced from the constant-stress test. The model, Equation 3-5, is mathematically similar, except for the regression constants. In addition a correction factor, C, was introduced to account for field conditions based on material volumetric relations (Asphalt Institute, 1982).

$$N_f = 0.00432C\left(\frac{1}{\varepsilon_t}\right)^{3.291}\left(\frac{1}{E}\right)^{0.854} \quad \textbf{(3-5)}$$

where $C = 10^M$

$$M = 4.84\left(\frac{V_b}{V_a + V_b} - 0.69\right)$$

V_b = percentage of binder in mix by volume

V_a = percentage of air voids in mix by volume

During the development of MEPDG, the aforementioned fatigue relationships were calibrated by changing the model parameters to match historical pavement performance data. This provided a fatigue relationship which was based on data recorded for pavements found in the U.S. and takes into account local conditions. Sample pavements were taken from the Long Term Pavement Performance (LTPP) database for the study. This updated fatigue cracking model was based upon using 94 new pavement test sections and 42 overlay test sections. Simulations were run for each test section using the MEDPG design software using modification factors on two aforementioned models. Calibration factors were added to mathematical models and these factors were varied over a range of values. This updated model with calibration factors is shown in Equation 3-6 (National Cooperative Highway Research Program, 2004).

$$N_f = \beta_{f1} k_1 \left(\frac{1}{\varepsilon_t}\right)^{\beta_{f2} k_2} \left(\frac{1}{E}\right)^{\beta_{f3} k_3} \quad \textbf{(3-6)}$$

Where: $\beta_{f1}, \beta_{f2}, \beta_{f3}$ = calibration parameters

The calibration factors were varied in value from 0.8 up to 2.5 based upon past literature concerning fatigue testing. The data was input into the program and the predictions of fatigue cracking were produced. The results along with actual field measurements from the LTPP database were compared and optimized to generate a calibrated model. The recommended model is a version of the model given by the Asphalt Institute calibrated for the test section data and is given in Equation 3-7.

$$N_f = 0.00432 \times \beta_{f1} \times C\left(\frac{1}{\varepsilon_t}\right)^{3.9492}\left(\frac{1}{E}\right)^{1.281}$$

$$\beta_{f1} = \beta'_{f1} \times k'_1$$

and **(3-7)**

$$k'_1 = \frac{1}{0.000398 + \frac{0.003602}{1 + e^{(11.02 - 3.49 h_{ac})}}}$$

where β_{f1} = parameter for thin asphalt pavements

h_{ac} = total thickness of asphalt layers, inches

3.4 - Marquette Interchange Instrumentation Project

The Transportation Research Center at Marquette University was awarded a project on behalf of the Wisconsin Highway Research Program (WHRP) to instrument a section of a perpetual pavement along Interstate 43 just north of downtown Milwaukee, Wisconsin. The section of highway is located just north of what is more commonly known to Milwaukeeans as the Marquette Interchange. The megaproject to reconstruct the Marquette Interchange had been continuing since 2004 and was finished by 2008.

As a means to make a lasting investment, an HMA perpetual pavement was selected as the pavement of choice for the North Leg contract. The final pavement design incorporates a 13 inch (325 mm) HMA pavement built upon multiple aggregate layers placed over the native soils.

The breakdown of the pavement structure is listed in Table 3-1 and Table 3-2. The design philosophy of the HMA perpetual pavement is to minimize tensile strains at the bottom of the HMA layer asphalt to maximize the fatigue endurance of the pavement.

In concept, the onset of bottom-up tensile fatigue cracking should be delayed for a period of 40 years or more.

Table 3-1 - Breakdown of asphalt layers.

Description	Thickness, in (mm)	Mix Size, in (mm)	Binder Type, PG
SMA	2 (50)	0.5 (12.5)	70-28
E30	7 (180)	0.75 (19.0)	64-22
C2	4 (100)	0.75 (19.0)	64-22

Table 3-2 - Breakdown of aggregate layers.

Description	Thickness, in (mm)
Open Graded	4 (100)
Dense Graded	6 (150)
Select Crushed	18 (460)

As a means to confirm the design process and philosophy, the pavement section was instrumented to record key pavement response data in real-time under live traffic conditions. In addition, multiple other measurement devices were integrated to supplement the pavement response data. All measured data is stored in a remote database and represents a comprehensive set of response data for a heavily trafficked urban freeway.

3.4.1 - Project Objectives

The main of objective was to fully instrument a perpetual pavement to verify the design assumptions. This implied that many sensors would be used to measure dynamic pavement responses under loading. Installation of the sensors presented itself to be a great challenge. In similar research projects instrumentation was done on test tracks or low volume roads; however, this project required meeting tight construction schedules

and installing instruments, sometimes, in a fraction of the time that would normally be required.

The data that is recorded will be used to correlate back to the design parameters as a check of the design assumptions. The first phase of the project consisted of setting up the system of instruments and the necessary hardware and software to acquire the data. Additional phases included laboratory testing to characterize the materials used for construction (resilient modulus, dynamic modulus, etc) and to create automated techniques for analyzing the mass of data.

3.4.2 - Instrumentation Plan

The Marquette research team developed the instrumentation plan which was presented and accepted by the Wisconsin Highway Research Program (WHRP). Past research that contained similar elements to this project were studied when formulating this plan. Of specific interest was instrumentation done within the MnROAD project sponsored by the Minnesota Department of Transportation and also the work done at the National Center for Asphalt Technology (NCAT) test track. Both of these projects contained instruments which were similar to those used in this research project - namely strain sensors, earth pressure cells, moisture sensors, etc.

3.4.3 - Project Location

The Marquette Interchange is located in downtown Milwaukee at the junction of Interstates 43 and 94. The interchange was originally constructed in 1968 and by the year 2000 carried 300,000 vehicles per day and 7 million annual visitors to downtown

festivals and attractions. The reconstruction of this interchange was done to enhance safety, widen ramps, create right-hand only entrances/exits and extend merge lanes. The northern portion of the interchange, termed the north-leg, includes a 1.5 mile (2.4 km) segment of Interstate 43 between Wells Street and North Avenue. The northbound lanes of I-43 were scheduled for reconstruction during the 2006 construction season and were selected for instrumentation via WHRP research grant 0092-06-01 (Hornyak et al., 2007).

The specific instrumentation location was selected considering installation costs, traffic patterns, and utility access. It was desired to locate the instrumentation section away from any areas that may have a lot of traffic weaving due to entrance and exit ramps. However, since the setting is highly urbanized, very few sections were distant from such areas and some amount of weaving is inevitable. Another important aspect was the physical layout of the landscape surrounding the test section. A control cabinet was required to house the hardware used to collect the sensor data and it would also be necessary for personnel to have easy ingress and egress from the site for future maintenance activities that may be necessary. Excessive distances between the data acquisition equipment and sensors were also unacceptable. Of greatest importance for the proper location was safety – both for personnel working at the cabinet and for the traveling motorist. Lastly, the proper utilities would be needed for the project. Fortunately, the segment of I-43 was surrounded with many lights, ramp meters, and other devices used for traffic management and power supply was not an issue.

The location chosen for instrumentation is located just before (upstream) of the entrance ramp merge from Fond Du Lac/McKinley Avenue, as shown in Figure 3-2. This location provided excellent protection from traffic and an acceptable distance from

the sensor locations. In addition, the site could be accessed from the on-ramp rather than the mainline. The lane closest to the curb-line was chosen to keep the distance from the hardware and instruments a minimum. Within Figure 3-2, the three rectangular areas denoted “A”, “B”, and “C”, indicate the location of the instrumented pavement, the control cabinet, and the roadside sensors respectively.

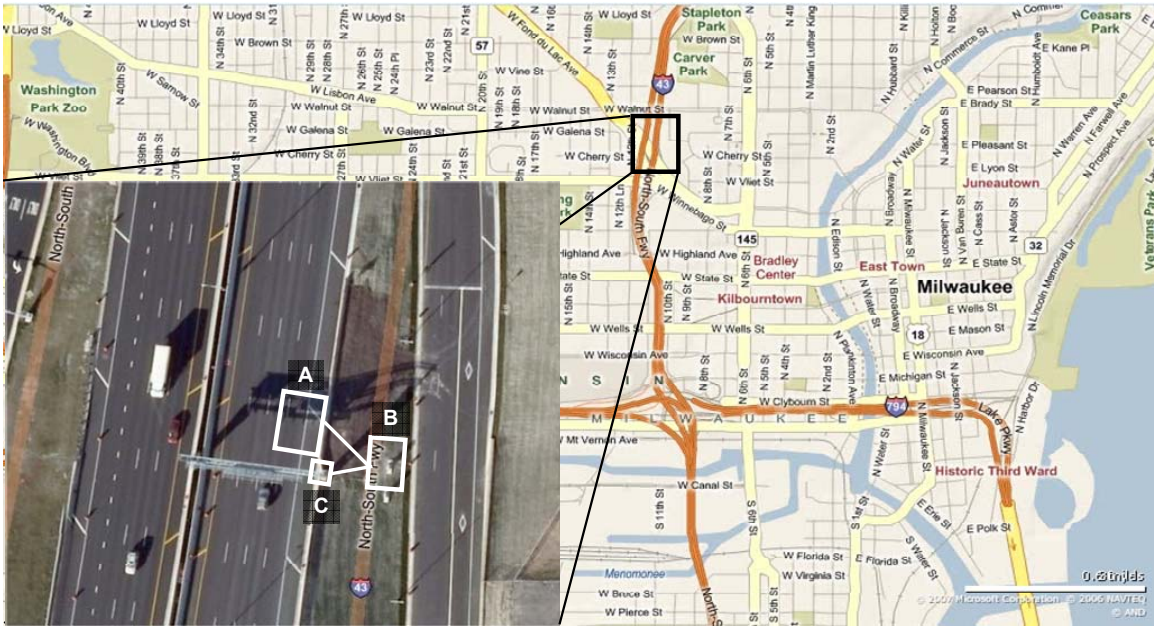


Figure 3-2 – Instrumented test section location.

3.4.4 - Instrumentation

As mentioned, the primary objective of the pavement instrumentation was to measure load-induced strains at the bottom of the HMA layer; however, many other sensors were included to help benefit researchers by providing structural and environmental data necessary for a thorough fatigue analysis. The list of sensors includes strain gauges, earth pressure cells, moisture probes, pavement temperature probes, a weigh-in-motion (WIM) system, a wheel-wander grid, and numerous environmental

sensors. The strain sensors, earth pressure cells, and soil moisture probes were calibrated at Marquette University prior to installation to provide the most accurate measurements possible.

3.4.4.1 - Asphalt Strain, Subgrade and Base Pressure

The general layout of the strain and pressure sensors is provided in Figure 3-3. As shown, a redundant strain gauge setup was accomplished by using three 8-sensor arrays of strain sensors, two of which were composed of strain sensors manufactured by CTL Group and the third comprised of sensors manufactured by Dynatest Incorporated. Four earth pressure sensors (2 redundant groups), manufactured by Geokon Incorporated, were installed to measure vertical pressures in the base and subgrade layers. As shown in Figure3-3, the pressure sensors are located between the strain gauge arrays. There are two pressure sensors at each installation location; one positioned approximately 3 inches (75 mm) below the top of the native subgrade and one approximately 1.5 inches (37 mm) below the top of the dense-graded aggregate base material.

3.4.4.2 - Subgrade Temperature and Moisture Content

Six ECH$_2$O EC-5 moisture probes manufactured by Decagon Devices (Decagon Devices Inc.) and six soil temperature probes manufactured by Romus Incorporated (Romus Inc.), were installed in the vicinity of the pressure cells, three of each per location. Pairs of temperature and moisture sensors were installed at depths of approximately 6, 12 and 24 inches (150, 300, 600 mm) below the top of the native subgrade soil.

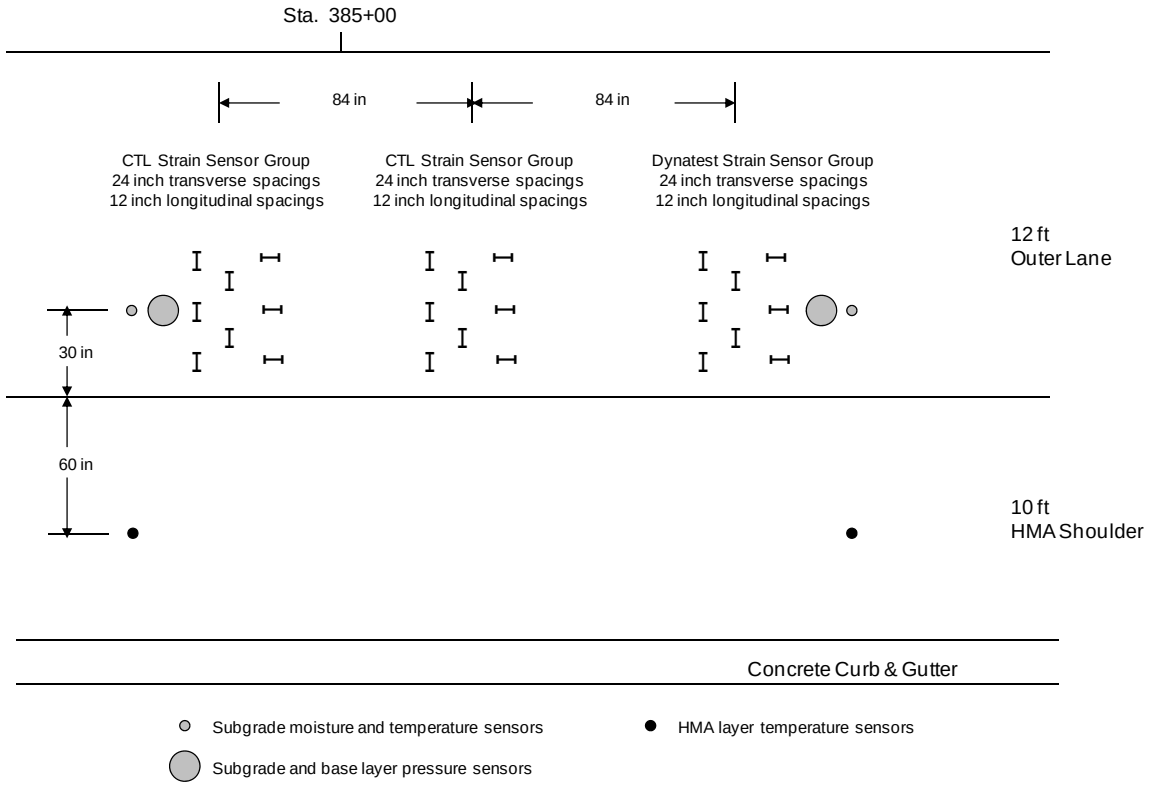


Figure 3-3 - Layout of the strain gauge arrays and earth pressure cell locations.

Two temperature gradient probes were installed to measure pavement temperatures at 1 inch (25 mm) intervals within the HMA layer. To supplement the temperature gradient data, an infrared thermometer was mounted at the roadside and aimed at the pavement in the vicinity of the sensors, providing pavement surface temperature.

3.4.4.3 - Weigh-In-Motion (WIM) and Wheel Wander

WIM and wheel wander systems were purchased from ECM Incorporated and installed directly after the strain gauge arrays. The wheel wander system is composed of three PK piezo strips manufactured by ECM arranged in a reverse "N" pattern as shown

in Figure 3-4. The wander system provides data necessary for the determination of the lateral placement of passing wheels, often referred to as wheel-wander, as well as wheel travel speed.

The WIM system was installed immediately downstream of the wheel wander strips. The WIM system utilizes quartz piezo sensors manufactured by Kistler Corporation (Kistler) and a Never-Fail inductive traffic loop (Never-Fail Loop Systems Inc.). The quartz piezo sensors were each positioned within the outer wheel path to provide redundant measures of the wheel loads affecting the HMA strain and base pressure sensors. The WIM system also provides data indicating the vehicles class, axle spacings, and speed.

In addition, a closed circuit camera was installed by the roadside to provide a snapshot image of each vehicle passing over the instrumented test section. This provides a check that the vehicle passing over the test section is consistent with the WIM and strain/pressure data.

3.4.4.4 - Environmental

Three different sensor types have been installed to provide environmental data. A vented enclosure and temperature sensor measures the ambient air temperature, and a standard anemometer provides a measure of wind speed. Two pyranometers have been installed to measure the solar radiation at the site (which can be used to predict pavement surface temperature). Together, these environmental sensors provide supplemental data useful for comprehensive pavement fatigue analyses.

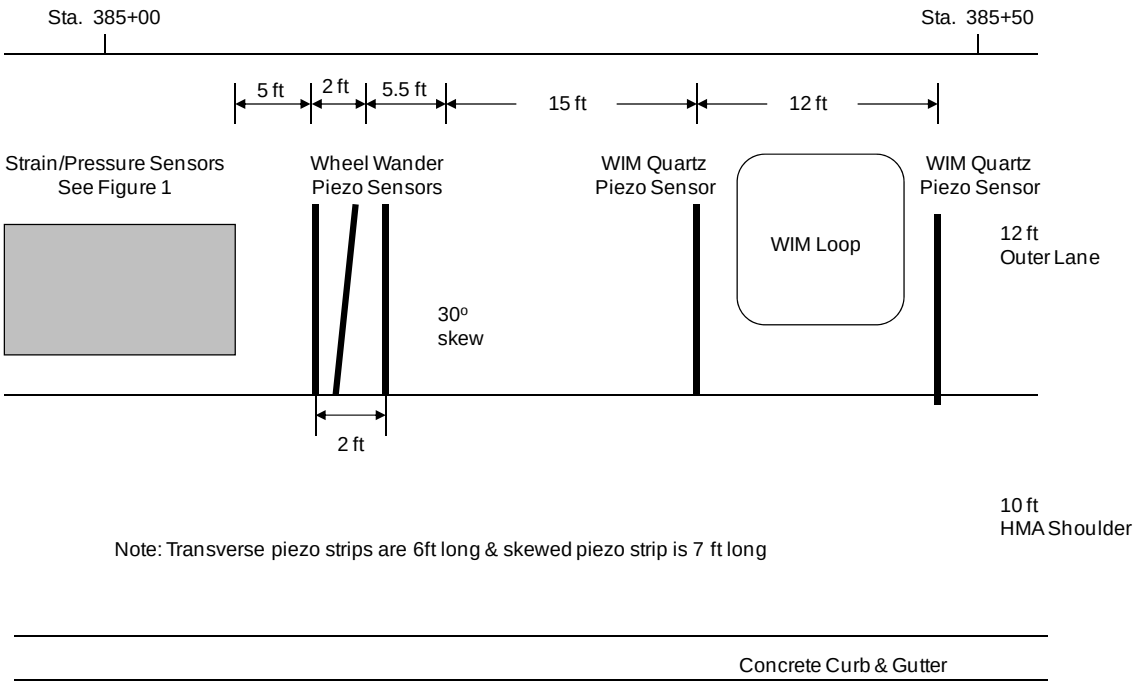


Figure 3-4 - Layout of the sensors used for wheel-wander and WIM.

The installation of these sensors was a crucial step in this project and failure to achieve a reasonable survival rate of the instruments would compromise the project. The details of the installation procedures, materials sample locations, etc. can be found in the project report (Hornyak et al., 2007). In addition the report details the calibration procedures used to verify the accuracy and precision of the instruments used. The measurement error within the instruments without calibration were significant – some errors in the strain sensors were found to be as high as 15 to 20% (Hornyak et al., 2006).

3.5 - Data Analysis Software

Due to the large amount of data being received from the project, it would be nearly impossible to analyze all of the data by hand. Because of this reality, computer programs were developed to automate the process of data extraction. Of particular

interest for this research project, and asphalt pavement research in general, is the maximum tensile strain induced in the pavement by a passing wheel load. The development of this software took place under Phase II of the Marquette Instrumentation Project (Hornyak & Crovetti, 2008).

The first task was to create a method by which the maximum tensile strain could be taken from the raw strain signals and then matched accordingly with the appropriate wheel loading. However "real" data contains many differences from the idealized case of data collection which complicates this task. Figure 3-5 is a flow-chart of the general process used for matching the peak strain data, load-times, and peak pressures with the associated wheel event.

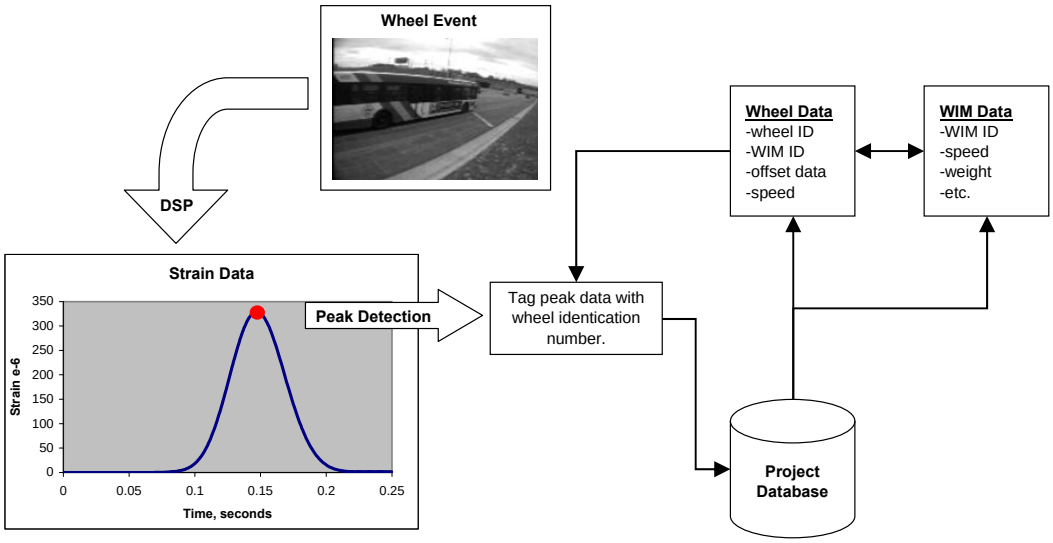


Figure 3-5 - Data collection process.

The difficulties in analyzing the data can be viewed at two different levels - the first level consisting of the strain signal as a complete entity. One aspect that is different from the idealized situation is the fact that there may be more than one peak in each

recorded strain signal. For instance, a tandem axle group may create two peaks that are closely spaced, especially at high speeds. Another concern is instances where the strain signal was recorded improperly, and the peak may be missing, partially recorded or instances where the wheel event is missing entirely. In general, it is not surprising to find anomalies within the data and each of these issues must be dealt with in the programming code. The causes of the error can be related to numerous things such as the vehicle merging into another lane, moving too slow or fast, vehicle bounce, and so on.

The second level is viewed on a smaller scale, as individual data points. An example of this sort of complication is signal noise, and how to distinguish noise from peak strains occurring for very light wheel loads.

The data analysis process for sorting out these peak events was coded and is described in various components. This keeps the coding understandable, manageable and easier to follow than one single file containing all of the code.

3.5.1 - Database Architecture and Program Flow

The database for the acquired data contains four tables which consist of the wheel data, WIM data, strain and pressure data, and environmental data. Each row of the WIM table represents one vehicle while each row of the wheel table represents an individual wheel event. Each row of the strain and pressure table represents one output data point from each of the 29 sensors (25 strain and 4 pressure), which were sampled at a rate of 1 kHz. The number of data rows for each wheel event is dependent on the travel speed of the wheel.

During acquisition, the recording software enters a data buffer and samples the strain and pressure sensors while the load is positioned anywhere within a sixteen foot segment of pavement, beginning at a point 8 feet before the sensor to a point 8 feet after the sensor location.

For a wheel travel speed of 60 mph (88 fps), the wheel would be within this 16 foot measurement window for a period of 16/88 = 0.1818 seconds. Thus, for the 1 kHz sample rate, a total of 182 samples would be stored. As travels speeds diminish, the size of the recorded data buffer increase. An identifier from the WIM system is tagged to the wheel data when the event is triggered.

The environmental table consists of rows of data taken every fifteen minutes and has no correlation to individual wheel/vehicle events other than the time at which they were taken. A graphical relationship of the data structure is shown in Figure 3-6 (excluding environmental data).

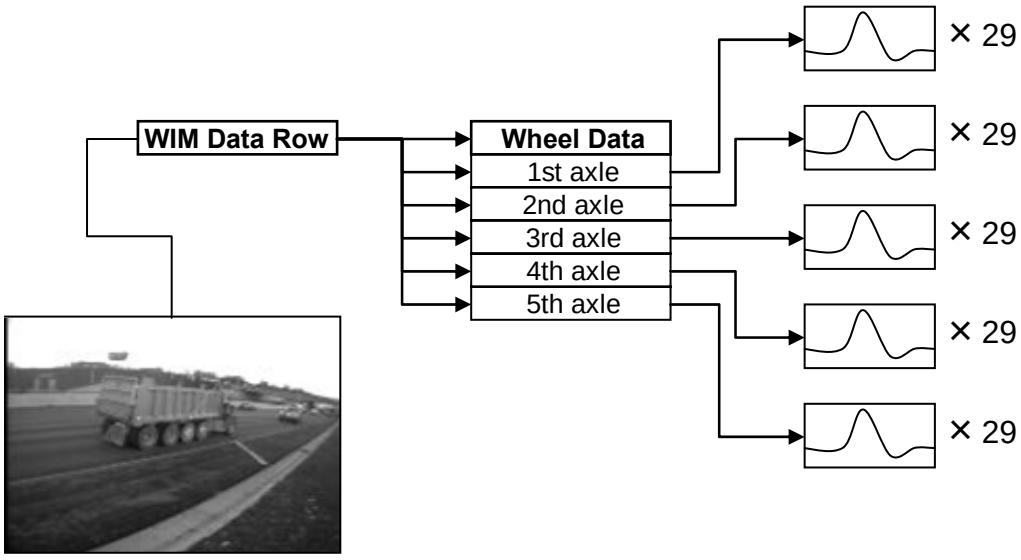


Figure 3-6 - Data relationship.

The most logical procedure to analyze the data based on this architecture corresponds to Figure 3-6. The analysis begins with one row of the WIM table (one particular vehicle) and caches the data row into memory. It then uses a key (WIM ID) unique to the WIM table and queries the wheel table, selecting all rows that possess the same key. The returned rows consist of the specific wheels for that particular vehicle. Finally the unique wheel identifications for individual wheels are used to query the appropriate strain and pressure data and then analyzed. The results of the analysis algorithms are then stored into another database for future use.

3.5.2 - Analysis Modules

As stated before, the analysis package was designed in separate modules. Considering the programming language used in this case, C#, they are often referred to as 'classes' which contain the methods to carry out the work. The significant modules, or classes, used here are for the following: peak detection in signals, aggregation of the peaks amongst all of the sensors, matching the correct peak with each associated wheel event, calculation of the load times, calculation of the area under the signal trace.

In addition to these, another module was developed to rebuild strain signals for axle groups which were closely spaced together – in essence taking the short recordings and 'splicing' them together to form one continuous signal.

There are many other incidental modules which had to be created in order for the program to operate, however they are not essential to this dissertation and information on

these classes is not provided here. These classes are responsible for operations such as opening database connections etc.

3.5.3 - Signal Regeneration

The function of the signal regeneration module is to take strain data for closely spaced axle groups and combine them into one long data stream. The development of this module actually came after difficulties arose while trying to match peak tensile strains with specific wheel events.

The system was designed so that when an axle of a vehicle enters the test section, the data acquisition system sets off events which were triggered by the wheel wander piezo sensors. When a wheel was detected, the data acquisition system entered the data stream being generated and selected data from the strain and pressure sensors for the window of time the wheel was in the test section.

Due to the limitations of the software, the recorded signals were not always uniform. These differences are illustrated below in Figure 3-7 for two FHWA class 4 vehicles recorded only a few hours apart and with similar offsets. The wheel weights for the slow and fast vehicles are 5.45 and 4.1 kips, respectively. Although the axle loads (or wheel loads rather) are somewhat similar, the length of the recorded signals are significantly different.

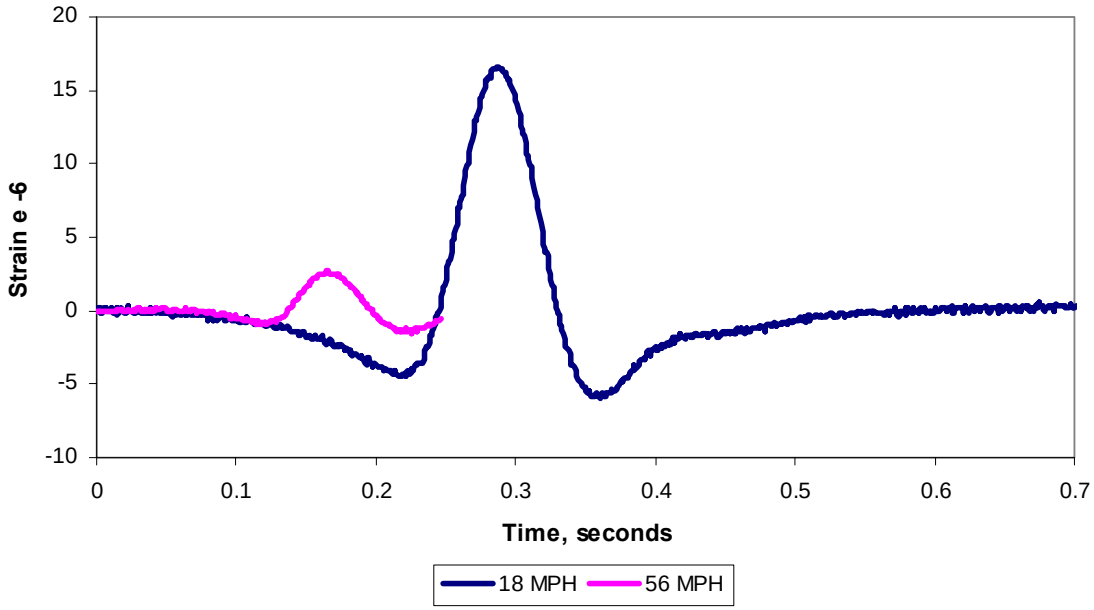


Figure 3-7 - Strain versus vehicle speed.

If the particular vehicle has relatively short axle spacings (say a tandem or tridem axle group), it is possible that the windows of this data collection for the axles may overlap, as shown in the strain signals in Figure 3-8 for a quad axle dump truck. Note that in Figure 3-8 the 3rd and 4th axle recordings are nearly identical and that the 4th recording actually contains data earlier than the 3rd recording.

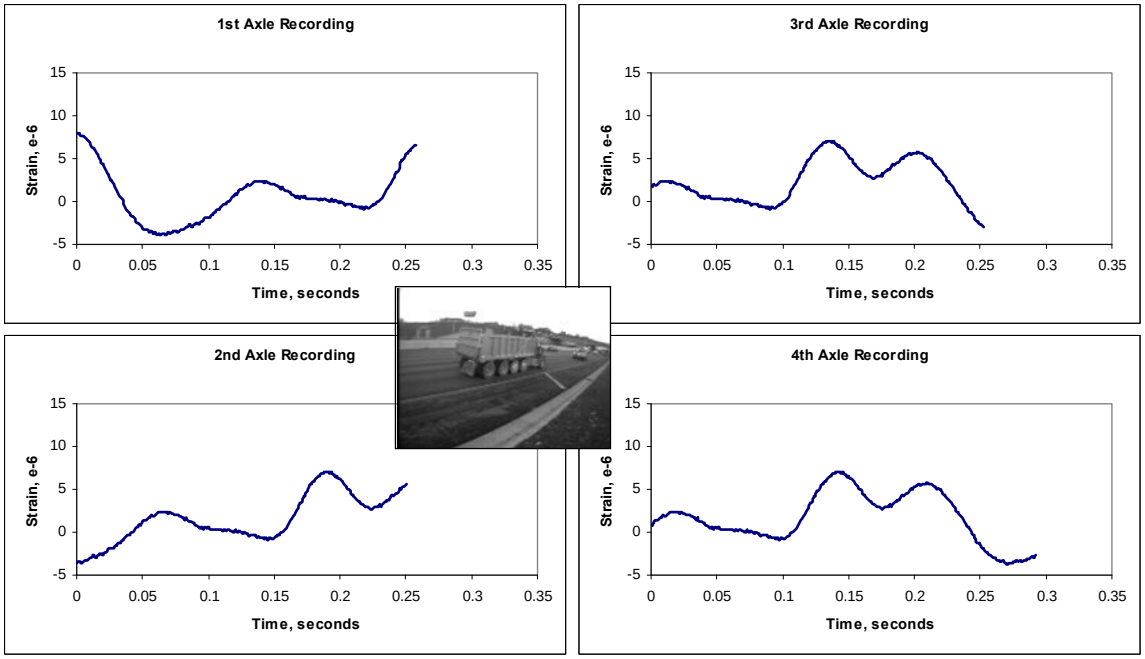


Figure 3-8 - Quad axle strain signals.

Depending on the type of vehicle and the axle configuration, it is possible to have nearly identical strain recordings for two or more wheel events. This presents an issue for analysis because there may be two peak tensile/pressure values in one data recording, thus making a row-by-row analysis of the data not possible. If it were to be done, it would be too difficult to differentiate which peak strain/pressure value corresponded to each axle.

In addition, there is the possibility of shadowing the change in strain/pressure caused by the wheel event. Since we are looking for the change in strain in the sensors an average of the first few data points is taken and subtracted from the rest of the data. For the purpose here, this subtracted value is referred to as the baseline. The 'shadowing' problem that could occur is that for strain recordings containing multiple axles, it is possible that the axle event just before the current may cause a peak in the beginning of the data trace. If the first few data points are averaged to try to obtain a baseline value, it

will compute a much higher (or lower) value than it should, thus possibly providing peak strains that are erroneous.

Because the acquisition system is simply storing data from a memory buffer, the same data points can exist in multiple strain/pressure signals. In the event that there are large overlaps in the signal traces, there may be a large number of data points which are identical. The chaotic nature of the signal due to the noise and from the wheel event itself causes a large enough series of data points to be considered unique. This unique sequence can then be used to search consecutive data traces for a sequence which is exactly the same.

The algorithm used only five data points from only one gauge to do this with almost perfect accuracy. The first five data points from a signal are subtracted from a moving array of data points from the previous signal. If the subtraction of the two arrays equals an array of zeros, then the data points from the newer signal are added to the older signal. This process is carried out for each wheel. However, since the timing for each strain gauge is identical, the remaining 28 instrument signals are added together based on the location found from the first gauge.

The final result of this module is a complete set of strain/pressure signals comprised of signals generated by multiple wheels. Depending on the nature of the recordings, this process may yield no results. In this event the rest of the program handles the analysis in a separate manner.

3.5.4 - Peak Detection

The main function of the peak detection process is to search each strain/pressure signal recording and select the peak tensile strains. A simple process was used to accomplish this goal. Each row of the tabled data represents one peak detected in the signal recording and the number of peaks is limitless.

A common method for finding peaks in data signals is to perform a linear regression of a set of points and subsequently analyze the slope. When the slope equals zero, it is assumed there is a peak at that location. A tool in the software package LabVIEW by National Instruments contained this process and was found to be quite accurate but the simplified process was easier to deploy and control programmatically.

The peak detection module for this project operated in a simple manner. The strain/pressure signals were comprised of data points recorded every 0.001 seconds. First a submitted strain trace was normalized to a baseline value by subtracting the average of the first 20 points from the entire signal. Next, beginning at the first data point, a group of five points are averaged. Then another group of five data points are averaged at a specified distance of 10 data points away (i.e. calculating the centroid of data points at two distinct location spaced apart a reasonable distance). These two averaged values are then subtracted. If the difference is within a specified threshold, then the data point is stored as a peak.

There are different properties which could be set in the algorithm to maximize accuracy. The number of data points to use in the averaged values and the spacing between them can be adjusted as well as the threshold for the difference between them. A thorough analysis had been conducted to obtain the best values. In addition to these an

overall threshold has been applied to filter out signal noise, thus anything smaller then about +1 με (tension) is not recorded as a peak. A time spacing threshold has also been implemented, otherwise it was possible that multiple points meet the search criteria within the same peak pulse.

It should be noted that the signal was manipulated before the actual peak detection. Each data point is multiplied by a factor of 5 to increase the effects of values greater than plus or minus one. This helps to eliminate the effects of signal noise and make actual peaks more pronounced. Also, only the absolute value was used when comparing the difference in the average values; thus a 'valley' would be processed as a 'peak' and the data point is stored with its original sign.

3.5.5 - Load Time and Area Integration

Two other important pieces of information that were taken from the data was the time of loading and the area underneath the tensile (compression) portion of the strain (pressure) signals.

For a particular strain signal, the algorithm required the location of the peak data point of interest which was generated by the peak detection module. The load time is calculated in the program by entering the normalized strain/pressure signal and moving backwards through the data until the strain value is < 0 and records the location (or time). The program then re-starts at the peak value, moving forward through each data point looking for the first data point to drop below zero, storing the location of the point. The load time is the product of 0.001 seconds and the number of data points between the two stored values (based on a sample rate of 1000 Hz). Using the two values found for the

load time, the program iterates through the data points and calculates the area under the strain/pressure curve using the trapezoid rule.

It should be noted that for certain axle configurations, namely tandems and tridems, it was possible that the area and load time calculated for the axles in the group were exactly the same (i.e. same data for each axle in the group). These particular values represent the load time and area for all three combined axles because the strain signals might not return to a base value. An example of such a signal is shown in Figure 3-9 – the strain signal represents the tandem axle group for the class 6 vehicle.

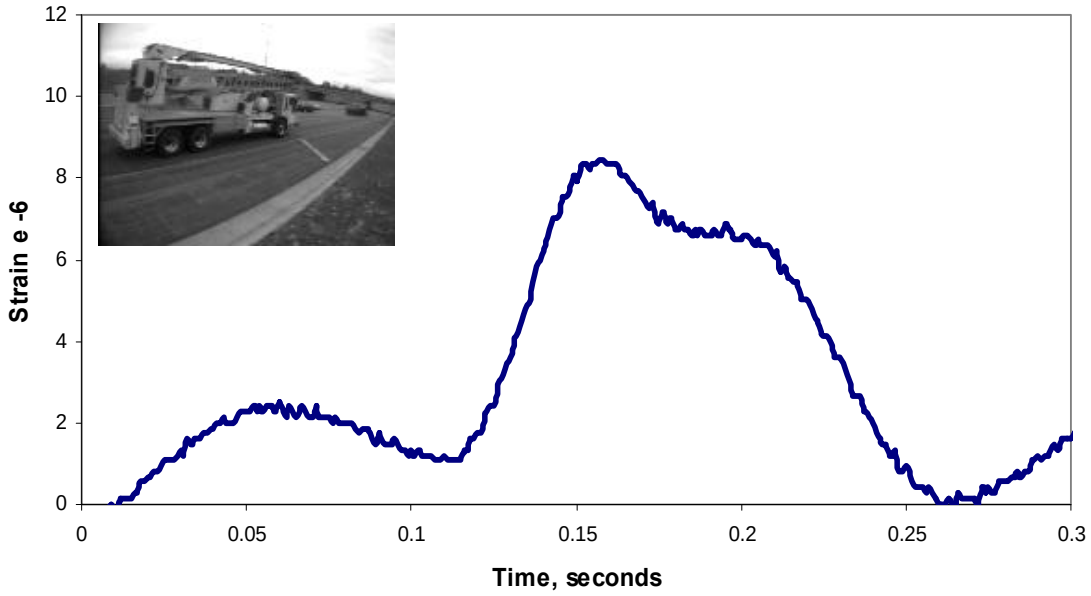


Figure 3-9 - Tandem axle strain signal.

When this module finishes, it returns two tables – one containing load time data and the other area data, each row in the tables representing an individual wheel/axle. The two tables each contain 30 columns; the first column is the unique wheel ID, and the other 29 represent data for each of the 25 strain gauges and 4 pressure cells.

3.5.6 - Matching Peak Strains and Pressures to Wheel Events

The previous three modules must be tied together in some fashion to yield meaningful data. The fourth module was responsible for creating the relationship between the previous three modules. This final module ultimately returns three tables of data which contain the peak strain/pressure, area, and load time data – each row associated with wheel identifications that match those in the original data tables. These three tables are then stored in a new database for querying at a later time.

This final module has three main routines contained within itself. The first routine (represented by steps 1 and 2 in Figure 3-10) gathers all of the wheel and WIM data for a particular vehicle which typically consists of one row of WIM data and numerous rows of wheel data (corresponding to each wheel on the vehicle that was recorded).

The second routine takes data from the peak detection program and sorts them into logical groups based on time (step 3 in Figure 3-10). The peak detection module takes a raw strain signal and detects and records peaks within a signal. Since there are 25 strain sensors and 4 pressure cells, the peaks that are detected do not occur at precisely the same time, but need to be sorted accordingly (i.e. the raw signals are not synchronized in time due to the variations of the physical locations of the sensors). This particular routine runs the peak detection on a signal and then organizes the results into a table where each row represents a relative instance of time. The result is a table of strain/pressure peak values where each row represents a particular instance in time.

Lastly, the third routine analyzes time signatures, matches the peak data with corresponding wheel events, and institutes the time of loading and area routines (step 4 in Figure 3-10). There are several different analysis paths the program uses based upon the corresponding FHWA vehicle class.

From the WIM data, the time spacing between axles is calculated based on speed and axle spacing. The program runs through the sorted and organized peak data looking for the same time signature and stores those particular values tagging them with the appropriate wheel identification. In the event that the program fails to find a proper time signature match, there are default routines to employ a generalized process for storing peak data.

When this module finishes, it returns three data tables containing the area, time of loading, and peak strain/pressure data. From here another set of code handles saving the data to a database for use at a later time (step 5 in Figure 3-10).

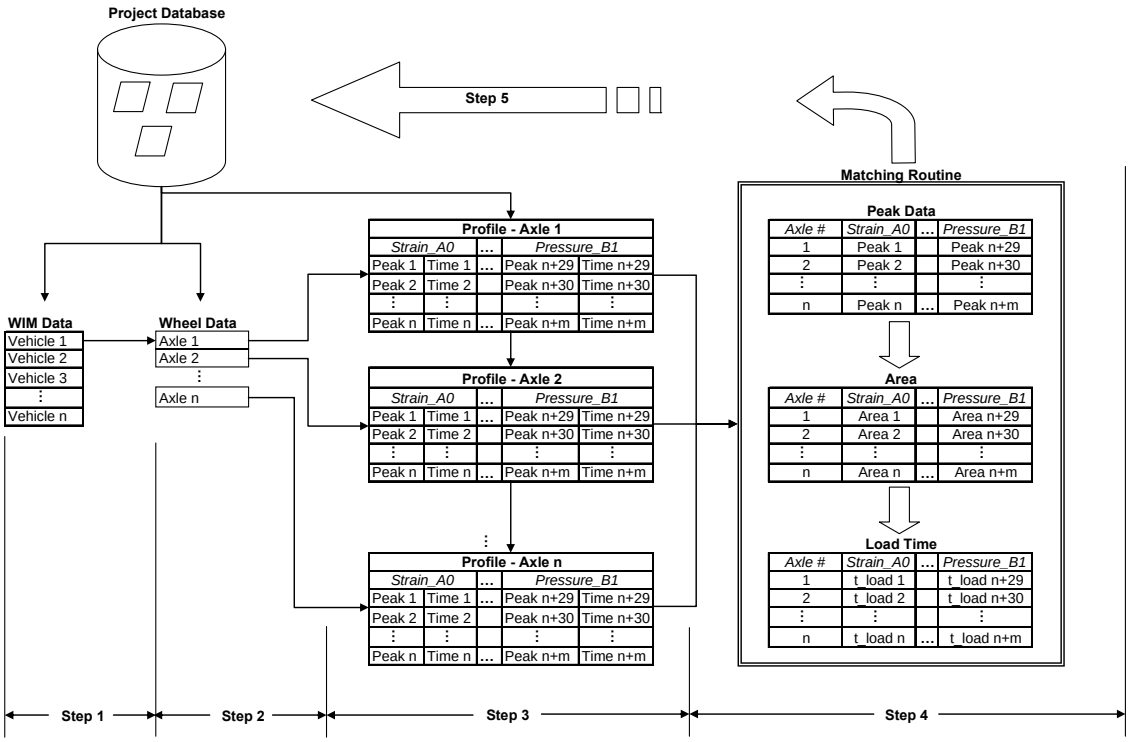


Figure 3-10 - Data analysis program flow

With these modules working together, a completely automated system has been set up to iterate through every row of WIM data and store the analysis data. The program has been developed to analyze the data one week at a time. Depending on the number of wheel events that cause significant strain measurements, the analysis may take up to several hours to process one week of recorded data.

3.5.7 - Analysis Performance

The tools developed to analyze the data and generate meaningful information were checked to ensure accurate results were being obtained. To accomplish this, a series of vehicles were randomly selected from the database and the peak values, area under the

curve, and load times were computed manually. The values were compared against those generated by the analysis program.

One-hundred randomly selected vehicles were chosen from FHWA vehicle classes ranging from class 4 up to class 10 vehicles, which represented the most common vehicle types. All of the axles for the selected vehicle were included in the comparison, but only one strain signal was randomly selected for each axle to account for any localized effects from gauge calibration factors, wheel offset, etc.

It should also be noted that one week of data (42nd week of 2007), was used to sample the vehicles and that this week represented a time where ambient temperatures were moderately warmer and tensile strains were moderately higher. This was done to maximize the number of applicable strain measurements for analysis.

Four vehicles were selected from each class (from class 4 to class 10) which comprised 90 individual axles. In some cases axles were missed by the data acquisition system. However no preferential treatment was given to vehicles with omitted axles. Furthermore, nothing was done to account for these missed axles in terms of checking the performance of the analysis program. The performance of the data acquisition process and the analysis process is completely independent.

From these 90 axles, a strain signal from the first 5 senors in the first sensor array (the 'A' array) was randomly selected to be included in the evaluation. (This was done because during the development of the software, a limited set of strain gauges was used to simplify preliminary coding. The first four gauges consisted of longitudinal gauges which present a greater challenge for analysis due to the common compression-tension-compression pavement behavior.) The actual peak strain value was picked manually for

each of the 90 strain traces. For the case of axle groups containing successive peaks, the value of interest was matched with the proper corresponding axle based on inspection of the signal. Any strain below the threshold of +1 με (tension) was recorded as zero to be consistent with the analysis program.

Figure 3-11 is a plot of actual peak tensile strain (from inspection of raw signals) versus the detected peak strain (selected by algorithm) for each sampled strain trace. Note that most of the sampled strains were lower than 12 με and the results are quite reasonable ($R^2 = 0.94$). In general the detected and recorded peak values (both strain and pressure) will be slightly lower than the actual value. This is due to the fact that a moving average is applied to the signal before analysis – the amount of error is roughly 0.1 με and should be small enough to ignore.

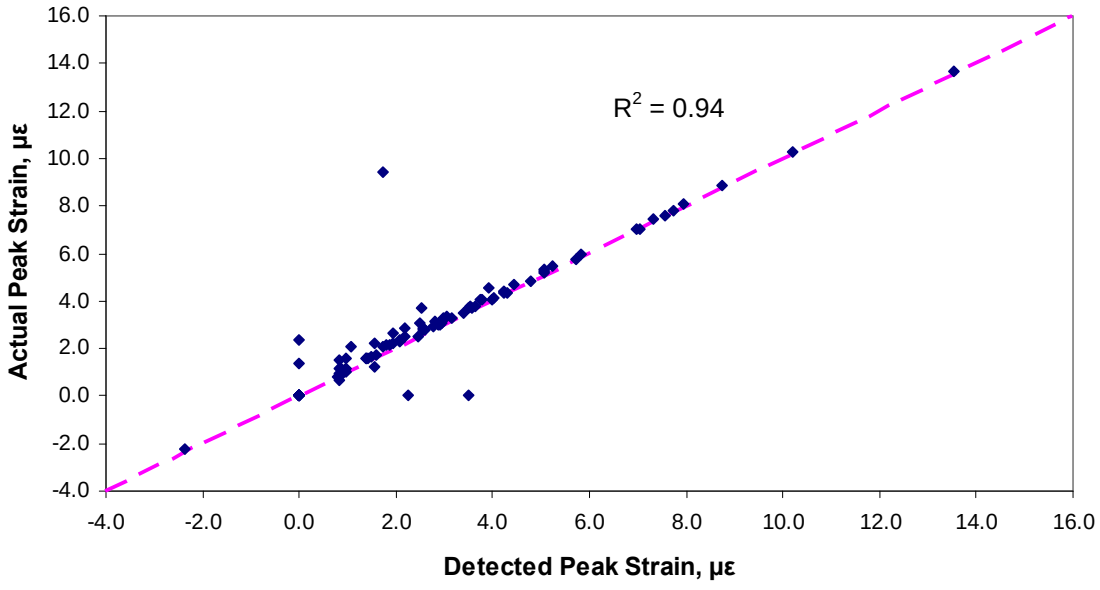


Figure 3-11 - Actual versus detected peak tensile strains.

Accuracy at these relatively low strain values provides some comfort in future analyses were strains become substantially larger. The algorithm used will become increasingly more accurate due to the inherent balancing of the simplified slope analysis used for peak detection. With low strain values signal noise can sometimes become as large as the strain peak, possibly causing misinterpretation of the signal and hence the need to set a lower bound threshold of 1 με.

As described in section 3.4.4 of the peak detection process, strains smaller then +1 με (tension) are filtered out. However, the plot in Figure 3-11 above shows values below this threshold. This is possible because the peak detection process analyzes the absolute values of the signals so all extremes, valleys and peaks, are detected and because the signals are scaled up by a factor of 5. Because of this it is possible that a negative (compressive) strain peak can be included in the result set. In addition, the program locates the peaks on the representative set of signals (sensors) and locates peaks on the remaining sensor signals based on the window of time from the representative sensors.

The algorithm for selecting peaks from the strain has also been applied to the pressure data. No formal check has been made, but close inspection shows that the algorithm operates in similar manner in regard to the strain pulses. In general the pressure cell instruments have much less signal noise and hence the peak detection algorithm operates more efficiently.

The area under the curve and load time calculations are based directly off of the raw or regenerated signals and the peak strain/pressure data as stated in section 3.4.4. Due to this, the accuracy of these values is directly dependent on the performance of the peak detection. Random checks of these calculations show that the algorithms perform as

designed; any errors in calculations are caused by errors in the peak detection process. Post processing of the data can be used to discard erroneous data since the flawed results contain either a set of zeros or extremely large, and clearly unreasonable, values.

3.6 - Pavement Modeling

Since experimental measurements were made in the field, the pavement was modeled as closely as possible in order to test the validity of making predictions for the pavement. This also served to test current practices used for pavement design and develop a prediction model.

The results of the modeled pavement can be used to analyze specific parameters such as fatigue accumulation in the pavement as well as other assumptions made during pavement design. Of interest here is the magnitude of tensile strain at the bottom of the pavement at distances away from the load center. The assumed distribution of tensile strains from analytical models may be significantly different than the experimental distribution. This discrepancy among the influence areas may have implications for calculation of fatigue damage accumulation across the width of the pavement.

In addition, the same scenario can be stated for vertical pressures and the stress influence under analytical and actual conditions. The permanent vertical deformation of the pavement through fatigue is outside of the study of this work, but could be integrated in the future. However, the load pulse durations are of interest for this work and the Method of Equivalent Thickness (MET) is applied and can be compared to that found in the field.

The modeling approach used here covers the aforementioned areas of interest and was designed to provide the pertinent information required for this work. The following discussion describes the finite-element (FE) model developed, environmental, material and load data integration. The entire FE program was written in the C# computing language.

In pavement analysis it is common for the compressive stresses and strains to be considered positive. However for this analysis tensile strains are considered positive to be more consistent with classic mechanics exercises. Furthermore, the axisymmetric analysis utilizes a polar notation to describe stress and strain – Figure 3-12 shows the notation for stress/strain used in this analysis.

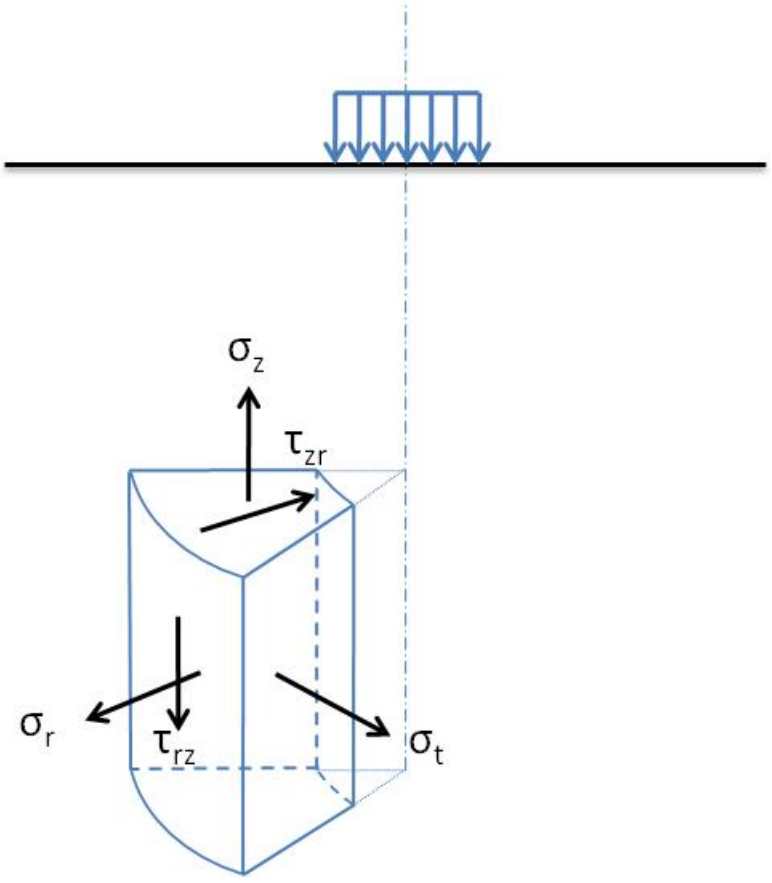


Figure 3-12 - Stress/strain notation.

3.6.1 - FE Model

The most accurate representation of a pavement structure would be a three-dimensional model – taking into account the real geometry. This model could account for discontinuities in the pavement such as paving boundaries, cracks, shoulders, et cetera as well as an exact representation of applied loading. However an almost equally accurate model can be developed using two-dimensional techniques reported by other researchers (Cho et al., 1996). Larger, three-dimensional models require much more computational time and it was not feasible considering the number of iterations performed in this research.

The technique applied for this work used an axi-symmetric model – a two-dimensional model in which is revolved around a selected axis and solved using linear-elastic analysis. This provides three-dimensional results, but only requires two-dimensional calculations. There are limitations of this type of model; the biggest of which is how the loading is applied to the model. Figure 3-13 is a graphical representation of the model developed for the pavement system.

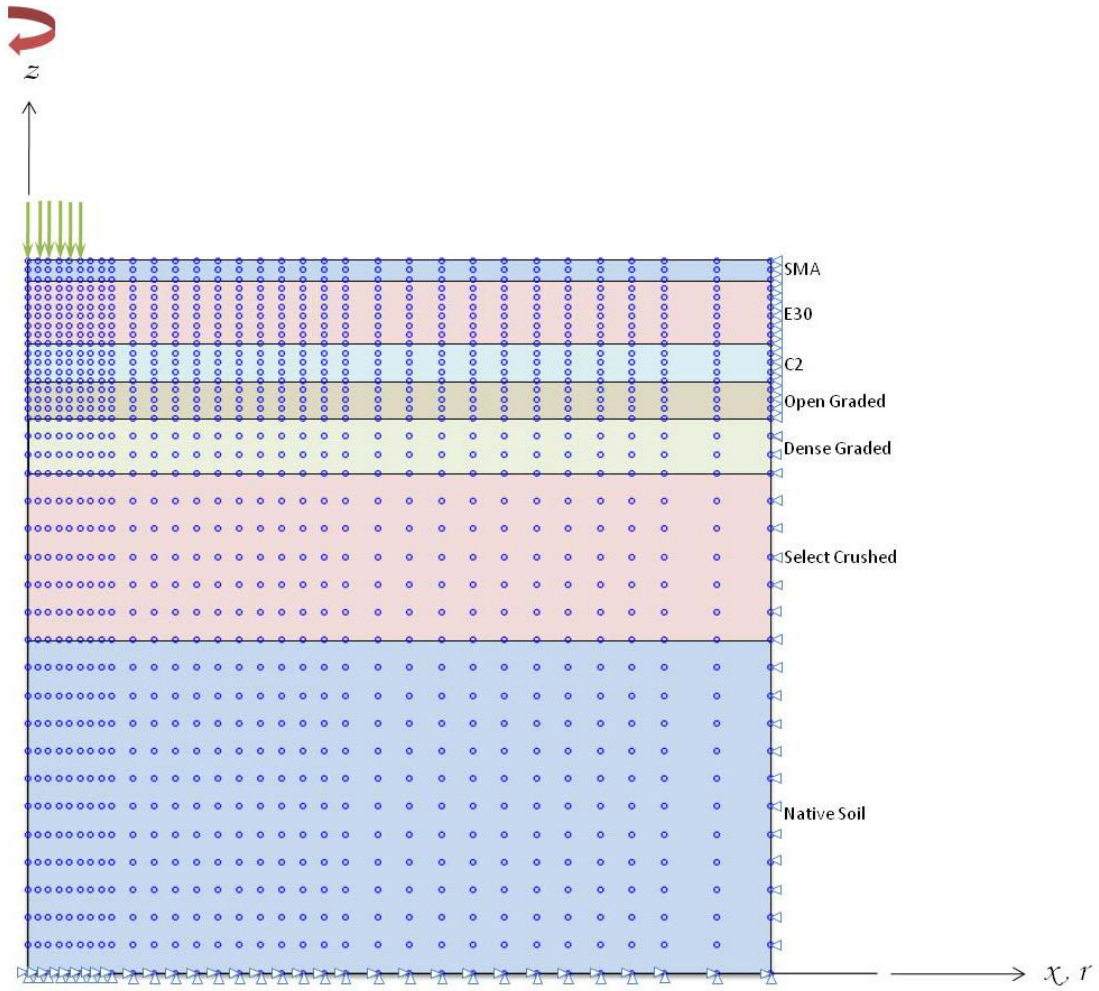


Figure 3-13 - Axi-symmetric pavement model used for analysis – 1178 elements.

3.6.1.1 - Load Application

The loading is applied in the upper left hand corner of the model. The vertical axis is the axis of symmetry and the loading represents half of the applied wheel-load. In this arrangement, the applied wheel loading is assumed to be circular which is a simplification of the actual contact area from common vehicle tires. A more accurate representation of the loaded area may in the form of a rectangular area with two semi-circular areas flanking each longer side as shown in Figure 3-14. However, for two- and three-dimensional finite element analyses this representation is commonly converted to a

rectangular area. It is believed that errors due to the idealization of the loaded contact areas very small and is an adequate representation (PCA, 1984).

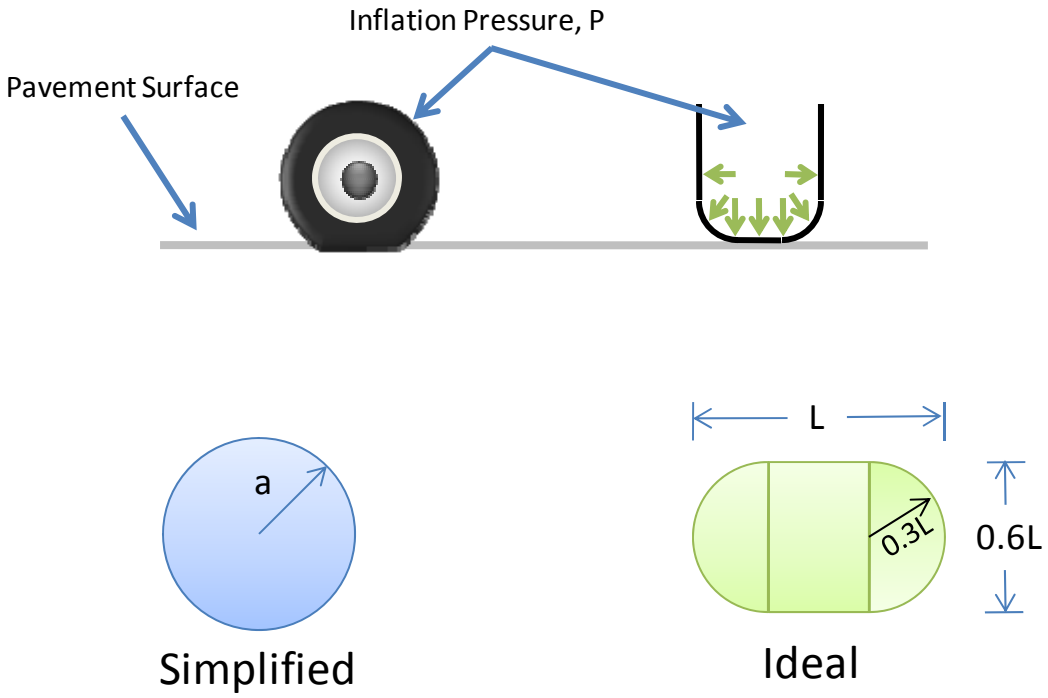


Figure 3-14 – Contact areas and contact pressure.

The above contact areas are assumed to have a uniform pressure applied over the entire area. It is further assumed that the contact pressure is equivalent to the inflation pressure of the tire. As more loading is applied, the amount of deformation of the tires increases proportionately and the contact area increases. It is assumed that there is no increase in the inflation pressure (although in reality there may be a very small increase) and thus the uniform pressure applied to the loaded area is equal to the inflation pressure.

For the idealized loading used in this research, the contact area and contact radius can be calculated given the inflation pressure of the tire and the load in Equation 3-8 and Equation 3-9.

$$Contact\ Area = \frac{P}{p} \quad \textbf{(3-8)}$$

$$a = \sqrt{\frac{P}{\pi p}} \quad \textbf{(3-9)}$$

Where: P = wheel load

p = tire inflation pressure

a = contact radius

The element used for the FE model is a four-node isoparametric quadrilateral with two degrees of freedom per node (x- and y- displacement). The isoparametric element differs from a standard quadrilateral by using the same parameters that define the shape function to also define the geometry.

Due to the limitations of the model, loads can only be applied to the nodes in the system. As stated before, wheel loads are modeled as pressures so a logical method was developed to redistribute the applied pressures into individual loads which were then applied to the proper nodes during analysis.

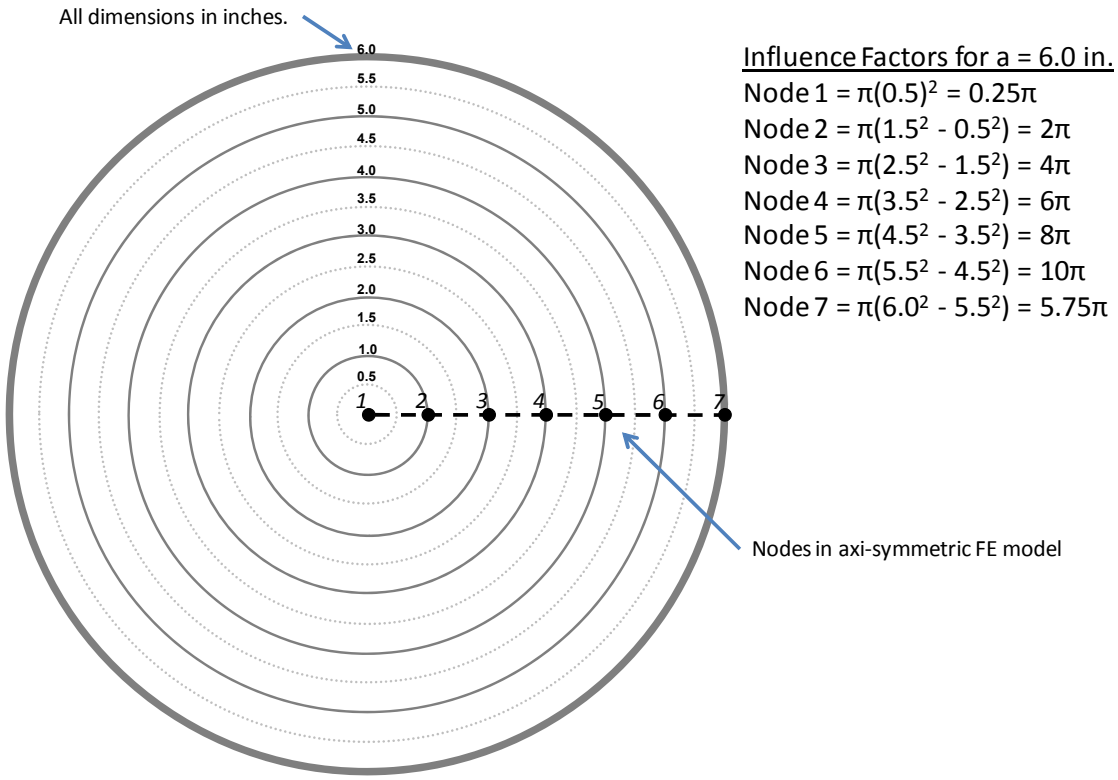


Figure 3-15 - Influence factors for circular load with a 6 inch contact radius.

The above figure illustrates the method used to allocate loads to the nodes in the model. The outermost ring represents the overall contact area of the loaded tire. The dashed concentric circles represent the boundaries between the influence areas to be distributed and the solid circles represent the centerline of the influence area. The dashed horizontal line represents the section that the FE model represents and also the location of the nodes in the model. The total load to be applied to any such node is simply the product of the influence factor (shown in Figure 3-15) and the contact pressure for a given contact radius.

It should be noted the total load is being applied at one location, even though the model is being analyzed in two horizontal dimensions to account for tandem or tridem

axle groups. For the case of tandem and tridem axle groups, the loads are superimposed using the principle of superposition and the cumulative strains and stresses are calculated.

The computer code has been written to handle the loading incrementally – given the wheel load, an appropriate load distribution scheme is selected to apply loads to the nodes in the model. In this process, the contact radius for each vehicle wheel is calculated using the load and tire pressure, the routine then drops into distinct cases based on the contact radius and applies the loading to the proper nodes in the force vector. As the computed contact radius increases, the number of nodes loaded increases, starting from the axi-symetric boundary outward along the modeled surface. For example, for a contact radius of 2 inches, only two nodes are loaded (on the axis of symmetry and the next adjacent surface node.). For a contact radius of 6 inches, six of the surface nodes are loaded, starting from the axis of symmetry and the next five adjacent surface nodes.

3.6.1.2 - Model Development and Adequacy

The finite element mesh was chosen through careful observations of the stresses and strains at critical locations in the model. Inaccurate results may be generated if the model represents a pavement area that is too small or if the element sizing and aspect ratios are inadequate. In order to quantify the size and mesh refinement, multiple models were constructed, solved and analyzed.

The first constructed model contained 322 elements and modeled 38 inches of the half-space and depth of 50 inches. Refinement in all models is focused on the areas near the load application, while the mesh is designed progressively coarser near the

constrained boundaries. The final element/nodal geometry are shown below in Figure 3-16.

For the comparison of the different models, average values were used for the material properties and are tabulated in Table 3-3. The stiffnesses for the HMA materials were placed on the lower range of typical values while applied load was 100,000 lbs with a contact radius of 4 inches. This loading is relatively high; however this was done to amplify the calculated values (in conjunction with the low HMA stiffnesses) since using more realistic values would have resulted in low values of strain and stress. Allowing larger values of stress and strain prevents errors from being accumulated in the numerical computations. Since the model is linear elastic, amplifying the load by a factor of 100, simply results in the solution being amplified by the same factor. This scenario is only done for testing the adequacy of the models.

Table 3-3 - Material properties for model development.

Material	Stiffness, psi	Poisson's Ratio
SMA	100,000	0.35
E30	100,000	0.35
C2	100,000	0.35
Open Graded	15,000	0.35
Dense Graded	15,000	0.35
Select Crushed	25,000	0.35
Native Soil	10,000	0.35

In the analysis program developed for this research, the loads from vehicles are scaled up by a factor of 10,000 and the solutions are reduced as necessary. The results from the program are reported in units of microstrain ($\mu\varepsilon$) and pounds per square inch (psi). During initial testing it was found that small passenger vehicle loads in conjunction

with high HMA stiffnesses during periods of low air temperatures, caused computation errors to be generated during analysis while using double precision. Double precision normally provides 16 significant digits – the direct mathematical computation when involving units of microstrain (10^{-6}) can often lead to computation errors.

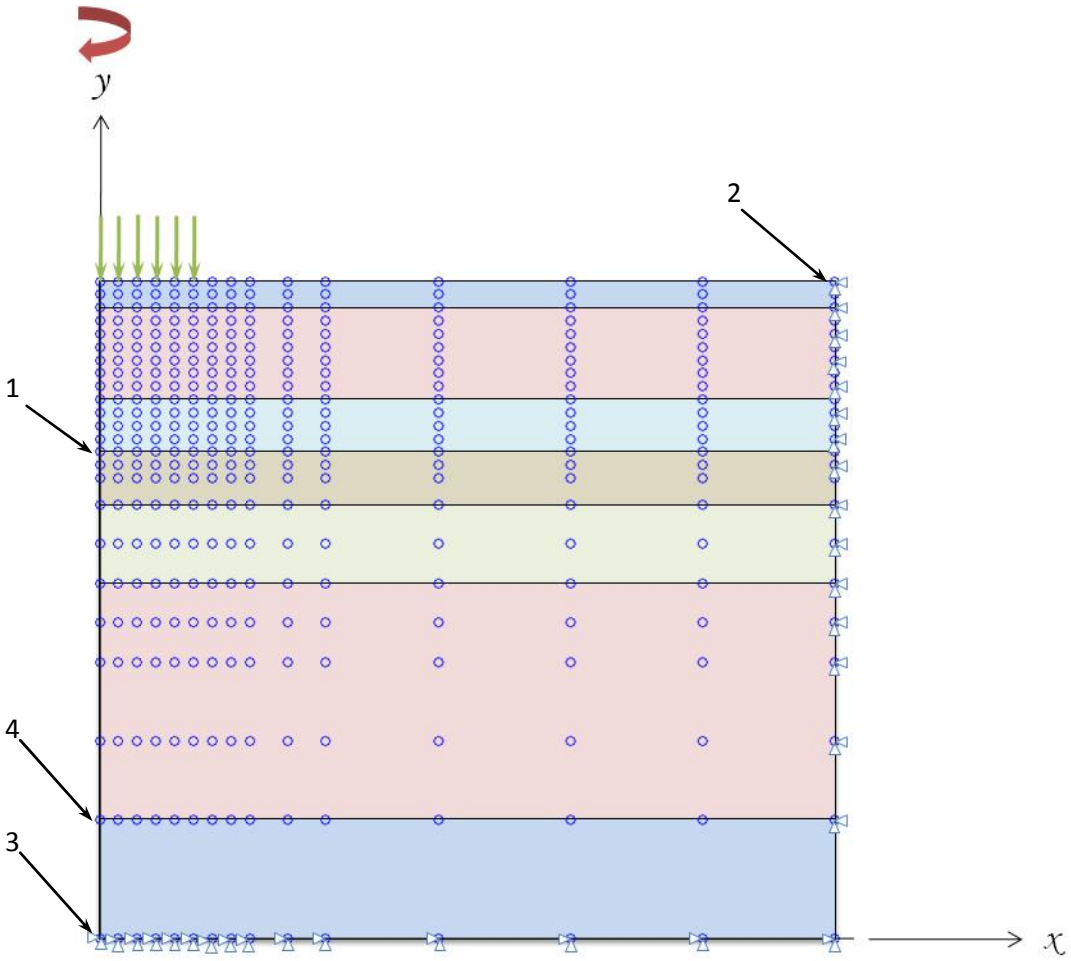


Figure 3-16 - 322 node mesh. Note some constraint symbols were omitted from upper right corner for clarity and that both the x- and y-displacements are constrained on the vertical face.

The degrees of freedom on the vertical edge, along the right side of the model, were constrained in the x-direction. In reality the surrounding pavement constrains most movement in the horizontal directions, similar to that of a plane-strain condition. The

vertical deflection of a real pavement will be substantial, but decreases with distance away from the applied load. The lack of constraint on the vertical side models this effect.

The bottom of the model was also constrained from both vertical and horizontal movement. The bottom of the pavement system is founded on the native soil layers. In reality, there is an amount of confinement placed on the pavement system which becomes greater with depth. This confinement restrains displacement – the constraints placed at the model boundaries should reasonably represent this situation.

Preliminary analysis of the 322 node model showed that there were significant stresses along the right vertical edge of the model, indicating that the constraints were influencing the results. The results of the analysis were compared to results from JULEA, a layered elastic analysis (LEA) program using the same loading, material properties, and geometry. This LEA package was used as a baseline for comparison to the FE model.

The horizontal tensile strain at the bottom of the HMA layer and directly under the load from the FE model was much lower than the strains computed from JULEA. Vertical deflections at the surface and vertical stresses in the base and native soil layers were also lower than those calculated by JULEA. These were indications that the model was too coarse and physical extents did not encompass a large enough area.

The model was increased to encompass a larger volume and the mesh refinement was adjusted as needed until the results at the locations of interest converged. The number of elements in the model increased from 322, to 522, 680, 912, 1178, and 2000. In the evolution of the models, the locations of interest used for comparing results were the horizontal radial tensile strain at the bottom of the HMA layer (under the load), the

radial strains at the upper right (restrained) corner, the radial strain at the bottom left corner of the model directly under the load, and the vertical pressure on top of the native soil layer directly under the load. These locations are labeled in Figure 3-16 as 1, 2, 3, and 4 respectively.

The final model chosen for analysis contains 1178 elements and measures 70 inches from the center of the applied load to the edge of the model (2 × 70 = 140 inches total width). The pavement is modeled to a depth of 77 inches below the surface. Results of the observations during model selection are shown in Table 3-4 below.

Table 3-4 – Finite element model results.

Mesh	Location			
	Radial Strain at Bottom of HMA Location #1	Radial Strain at Upper Right Corner Location #2	Radial Strain at Bottom Left Corner Location #3	Vertical Pressure on Top of Native Soil Location #4
322	2.7749E-03	6.3852E-04	4.7880E-09	NA
522	2.7505E-03	6.1574E-04	2.2233E-09	-2.0668E+01
680	2.7965E-03	4.4076E-04	1.1603E-09	-1.6129E+01
912	2.8063E-03	2.0174E-04	8.3634E-10	-1.6246E+01
1178*	2.8112E-03	2.0078E-04	8.3934E-10	-1.6297E+01
1178**	2.8028E-03	1.5379E-04	7.5912E-10	-1.8394E+01
2000†	2.8076E-03	7.3650E-04	8.5232E-10	NA

- All units in/in and psi.

* The width of the model was expanded by 30 inches and the depth increased by 27 inches.

** The constrained vertical boundary was modified to release/allow vertical deflections.

† This model had the same geometry as the others, however the model was meshed with 1 in. × 1 in. elements throughout the entire model.

During the analysis of the different models, it was observed that a considerable amount of stress accumulated at the upper right corner of the model (although not shown in the table above). With the vertical displacement degree-of-freedom fixed along the vertical edge, the stress at the upper right corner was 17.4 psi. The addition of allowing the vertical displacements on the right edge reduced this stress to only 1.1 psi. Constraining the vertical displacements appeared to make the model too stiff and it was

concluded that allowing the vertical displacements is a better representation of the pavement structure.

3.6.1.3 - FE Analysis Code

The finite code written for the analysis was based off of existing code written by Kwon and Bang (Kwon & Bang, 1997), but was heavily modified to suit the needs of this research. Additionally, the original code was written in MATLAB, but was adapted into a C#/.NET project for this research. This translation between computing languages offered the benefits of the easy integration with databases and storage needed for processing the data.

The original intent of the programming for this research was to use MATLAB code run as a component of a C# project. The limitations of the interoperability eventually led to the entire project being translated to C#. The C# code is available in Appendix A.

The mesh which was selected for the analysis program consisted of the 70 inch wide and 77 inch deep model which contained 1178 elements. The advantage of the models containing less elements is strictly one of computation speed. Current computing power limits the extent of rapid computations of large models with a high degree of refinement. This is why there was an extended effort in selecting an adequate model and mesh. The current model can complete a single linear elastic analysis in under 30 seconds.

The conceptual flow of the program is shown below in Figure 3-17. The program downloads the raw data from the Marquette Interchange database and begins

iterating through each vehicle in the database. The vehicle speed and environmental data are used in computing HMA stiffnesses and the vehicle weight data is used for distributing loads on the FE model (using an inflation pressure of 35 psi for FHWA Class 4 vehicles and below and 80 psi for FHWA Class 5 vehicles and up). Once the FE computations are solved, the results are passed back to another database for storage. The individual modules for computing material properties are discussed in the following sections for each respective material.

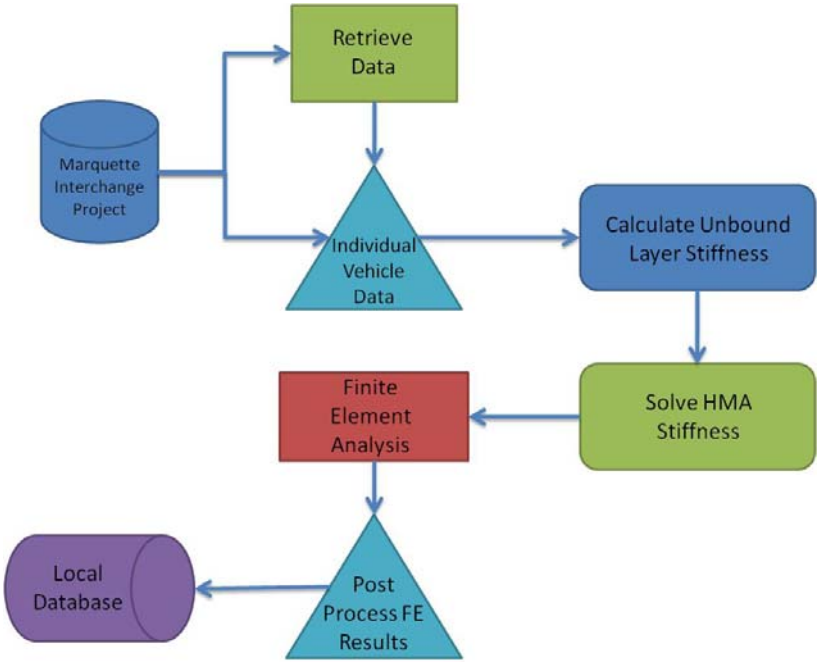


Figure 3-17 - Finite element analysis conceptual flow.

A large portion of the program deals with importing data from the Marquette Interchange Project MySQL database (MySQL/Sun Microsystems, Inc.) and subsequently storing results back to another MySQL database. The details of the

database transactions are available from MySQL and contained within the code in appendix A.

3.7 - Material Modeling

This section details all aspects of the materials used in construction of the test section and the relevant material properties used for pavement modeling. During construction of the pavement, representative samples were taken from the location of the test section. These samples were retained and subsequently analyzed under another research phase of the instrumentation project sponsored by the Midwest Regional University Transportation Center (Crovetti et al., 2008).

All of the retained material samples were tested according to the requirements of the Level I analysis in the MEPDG design guide (National Coorperative Highway Research Program, 2004). The materials tested included the three HMA mixtures, three unbound aggregates, and samples of the native soils extracted from the site.

The data from the materials characterization study were used to develop the constitutive material properties necessary to accurately model the pavement system. Models were constructed to match, as closely as possible, the variations in the material properties in response to environmental changes (temperature, moisture, etc.). The generated data was then used in construction of the FE model used for the analysis.

3.7.1 - Unbound Layers

Three unbound subbase materials were used in the construction of the pavement structure and were placed on the native soils layer. The three materials were quite

different in terms of gradation and composition – two were manufactured from recycled concrete crushed on-site and the other was manufactured from quarried limestone.

The pavement design called for 18 inches of select crushed material placed on top of the native soil, overlain by an additional 6 inches of a dense graded crushed material followed by a 4 inch open graded layer with the HMA topping off the structure. The select crushed and dense graded materials were produced from recycled concrete while the open graded material was manufactured from virgin limestone. The general strength properties are given in Table 3-5.

Table 3-5 - Summary of unbound aggregates.

Material	Resilient Modulus, psi	Poisson's Ratio
Select Crushed	30,000	0.35
Dense Graded	Varied by Season	0.35
Open Graded	Varied by Season	0.35

3.7.1.1 - Select Crushed

The select crushed material (SCM) was placed directly on top of the native soils and was designed to serve as an improvement to the native soils. The particle size of this material ranged from 4 to 6 inches in diameter. The gradation for this material according the Wisconsin DOT (WisDOT) specification is given in Table 3-6.

Table 3-6 - WisDOT SCM gradation.

Sieve Size	Percent Passing
5-Inch	90-100
1.5-Inch	20-50
No. 10	0-10

The large particle diameter unfortunately made the task of determining strength properties quite difficult and no data was produced from the characterization report to support specific resilient modulus values. However, careful consideration was used to estimate the resilient modulus and Poisson's ratio of the material. In addition, since the material is open graded and the pavement system contained adequate drainage there should be little variation in this strength parameter due to variations in moisture content in the system.

The resilient modulus used for this study was set at 30,000 psi which matches that used in the materials characterization report. The material is composed of recycled concrete which should have strength similar to that of virgin limestone particles. Similar materials are known to be stress-stiffening materials which imply that the stiffness of the material increases with increases in confining stress. The top of this layer is located 23 inches below the pavement surface which should provide a significant amount of confining stress. In addition, the material was placed in lifts and rolled to achieve maximum density.

Due to these observations and based on experience, a resilient modulus of 30,000 psi for this material should be reasonable. Furthermore, large deviations in this value should still provide reasonable FE solutions since the layer is located deep in the structure. This reduces the influence of this particular layer on the analytical solutions for the HMA layers.

The Poisson's ratio assigned to this material, as well as the other unbound aggregates, is based upon experience and historical values for similar materials. The ratio is known to be consistently around 0.35 to 0.40 for many granular materials – the

characterization study did not produce these measured values. Small variations in these values are known to have a small influence on the pavement analysis results.

3.7.1.2 - Dense Graded Aggregate

The layer placed on top of the select layer was composed of recycled concrete that was screened after crushing. The material contained both larger diameter (3/4") particles as well as smaller particles from the crushing process. This material closely appears to be an ASTM D448 #67 or 68 size number; however the additional small diameter material prevents definite classification. The material was reported to be classified as Poorly Graded Gravel with Sand under the Unified Soil Classification System (USCS) (Crovetti et al., 2008). Additional material parameters and gradation are shown below in Table 3-7 and Figure 3-18 below.

Table 3-7 - Dense graded material properties.

Bulk Specific Gravity	2.396
Apparent Specific Gravity	2.594
Absorption, %	5.47
Max Dry Unit Weight, pcf	121.1
D_{10} , mm	0.55
D_{30} , mm	3.6
D_{60} , mm	8.5
C_u , mm	15.45
C_c , mm	2.77

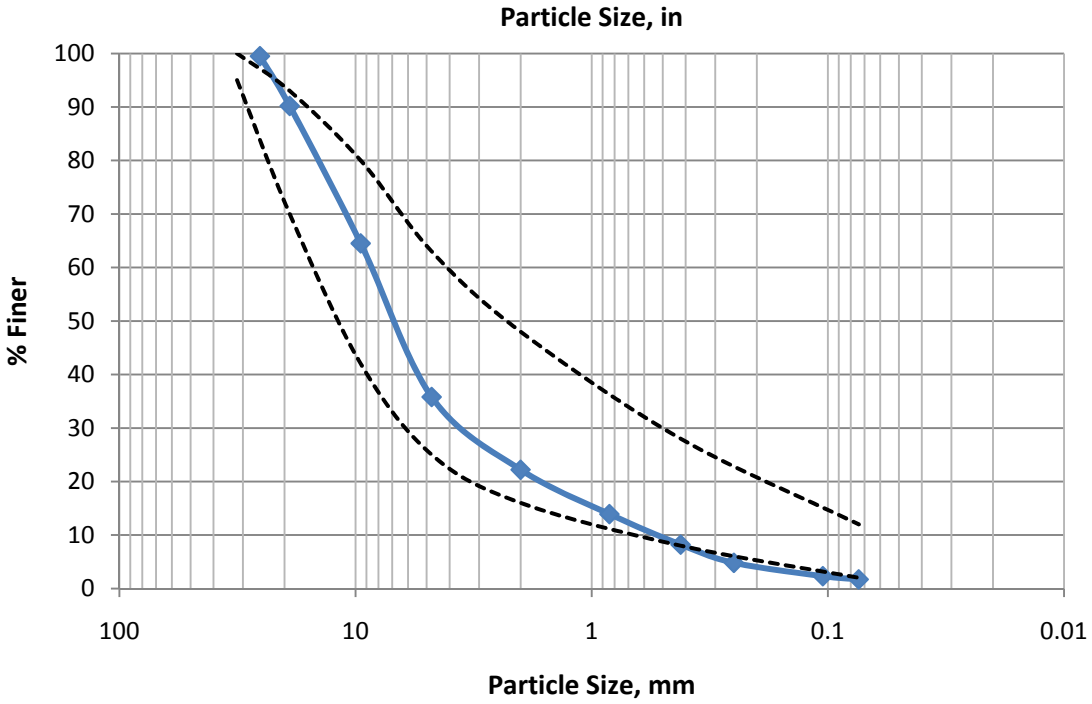


Figure 3-18 - Dense graded particle size distribution.

Resilient modulus testing was conducted on this material, and it was determined that a stress-stiffening model best represented the material. Equation 3-10 is the model generated from the laboratory data. The stress stiffening model implies that there is an increase in stiffness with increases in the bulk-stress (confinement stress). This model was used to generate a value for the stiffness - the dynamic effects from moving wheel loads (causing increases in bulk stress as the load passes) was simplified by including an average stress due to a static load (4.0 psi) into the bulk stress value.

$$M_R = 5{,}741\theta^{0.342} \qquad \textbf{(3-10)}$$

where M_R = resilient modulus, psi

θ = bulk stress, psi

The stress increases from moving wheel loads was considered on an average basis. Since the material layer is below the 13 inch HMA layer and 4 inches of open grade base course (OGBC), the increases in stress due to a wheel load are reduced. An average value of the increase in vertical stress based upon the standard 18-kip axle load was computed to be approximately 5 psi using the layered-elastic analysis program JULEA. This additional increase in vertical stress was included when calculating the bulk-stress.

The bulk stress represents the amount of confinement on the material which includes stresses from the weight of material above the point of interest, including lateral pressure, and stresses caused by loads applied to the pavement surface. This stress scenario is depicted in Figure 3-19.

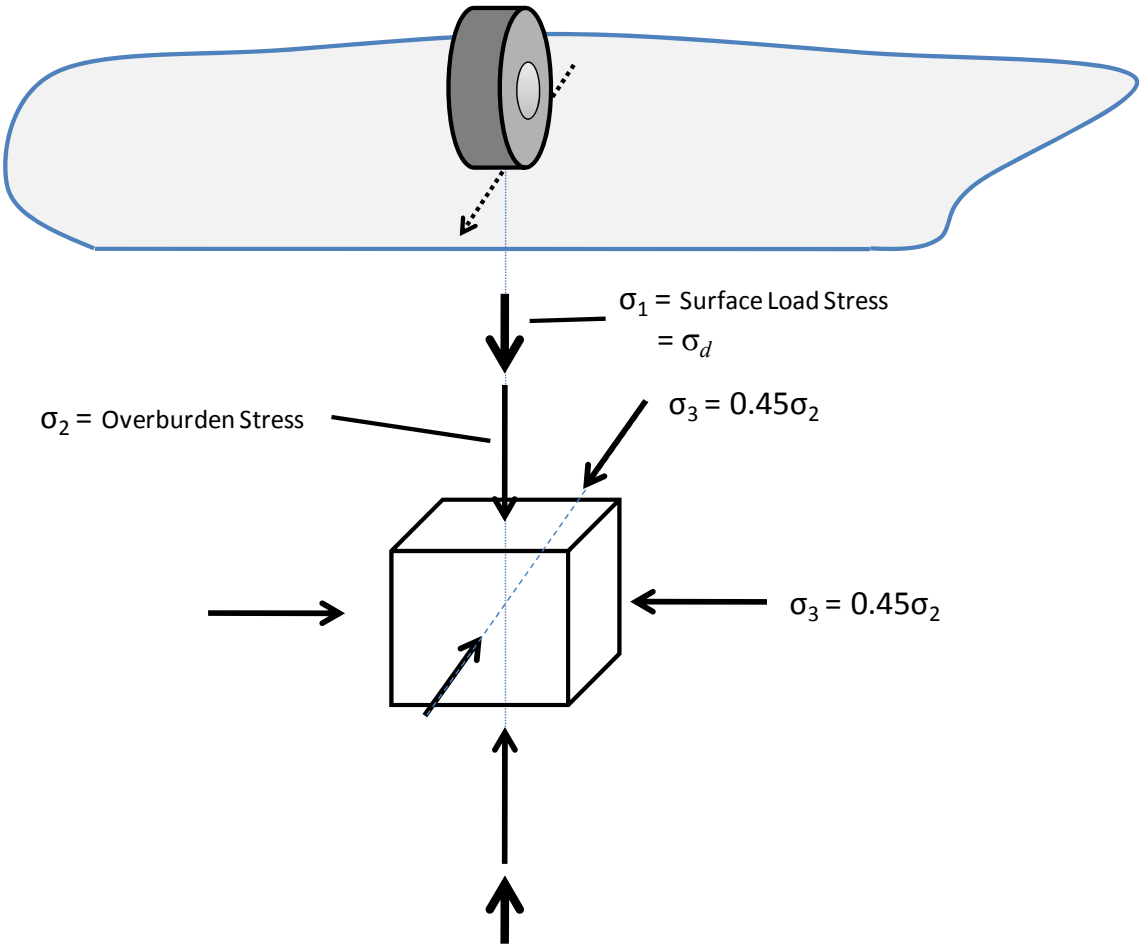


Figure 3-19 – Pavement structure stress state.

The bulk stress is defined by Equation 3-11 below.

$$\theta = \sigma_d + \sigma_2 + 2(k_0\sigma_2) \quad \textbf{(3-11)}$$

where k_0 = lateral earth pressure coefficient
= assumed 0.45

Using the assumed at-rest coefficient of lateral earth pressure, k_0, simplifies to:

$$\theta = \sigma_d + 0.9\sigma_3 \quad \textbf{(3-12)}$$

For computations of resilient modulus for all unbound materials, the actual unit weights of the materials were used. For the calculation of confinement pressures, the stress was calculated at mid-depth of each layer. Table 3-8 shows the computations for estimation of the stresses used for calculation of the bulk stress for all materials. The resulting bulk stress states were then used to estimate the resilient modulus.

Table 3-8 - Stress states and resilient modulus values.

Material		Unit Wt. or Gs	Thickness	Vertical Press.		Lateral Pressure	Wheel Load	M_R
		PCF	in	pcf	psi	psi ($k_0 = 0.45$)	psi	psi
HMA		2.478	13	167.51	1.16	NA		
Open		128.4	4	188.91	1.31	3.2904	6	4616
Dense		121.1	6	240.59	1.67	2.5518	4	12942
Select		125.0	18	364.61	2.53	2.4894	3	30000
Native	Upper	144.2	6	530.46	3.68	2.5577	2	12132
	Lower	146.2	6	603.56	4.19	2.7861	2	5607

The dense graded aggregate layer is relatively permeable and is rather moisture insensitive. It is likely that there will be a small variation in the stiffness of this material with changes in the moisture content of the layers in conjunction with temperature. The pavement structure has a drainage system, and it is unlikely that the layers will become saturated. In the future, it is possible that the unbound aggregate layers could become saturated if the drainage system is not maintained or accumulation of fine materials causes blockages in the underdrains. Saturation would have the effect of lowering the effective vertical stress (the particle-on-particle stresses) and reduce the bulk stress state, thus reducing the resilient modulus.

Since it is likely that there will be a small change due to moisture and temperature, the stiffness was varied slightly with seasonal changes. Stiffness variations

were accomplished in a generalized manner for the dense and open graded layers while the native soils were varied using models developed by past research.

Based on the summary of the soil moisture generated from the soil moisture probes, a particular month was chosen to represent the highest stiffness and another was chosen to represent the weakest month. The soil moisture data is summarized in Figure 3-20 – the precipitation data was taken from weather station data located at General Mitchell International Airport (located in the local Milwaukee area).

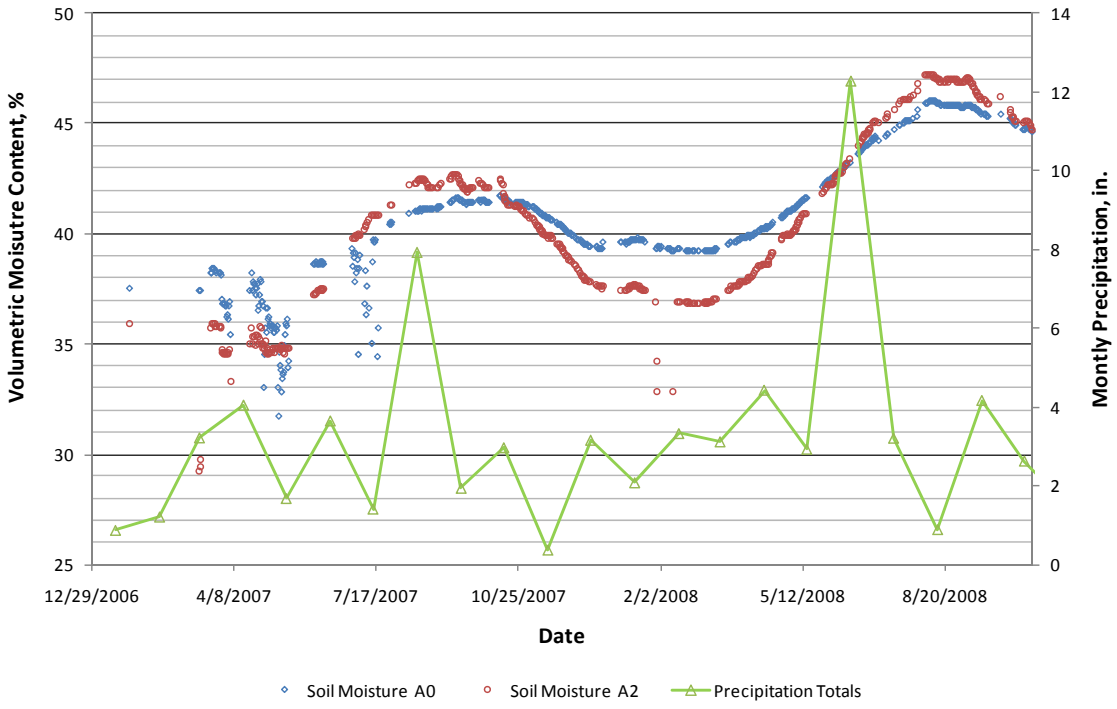


Figure 3-20 - Summary of soil moisture and precipitation at the test section.

Based on the soil moisture data from the test section it was determined that the month with the lowest moisture content (and the peak stiffness) was February while the wettest month (and lowest stiffness) was August. A reduction of 2% of the peak strength per month was used for setting up the uniform gradient for stiffness variation. This

represents approximately 10% reduction in stiffness over a 12 month period. This estimate of variation was developed by investigating design recommendations by the Asphalt Institute for subgrade strength.

For example, starting from the peak month, 2% of the peak strength was subtracted from the previous month's stiffness until reaching the weakest month. From the weakest month, the stiffness was increased at the same rate until reaching the peak stiffness again. Table 3-9 contains the calculated values based on this variation method - these values have been used for use in the FE analysis.

Table 3-9 - Dense graded resilient modulus values.

Month	Peak Stiffness, psi	Reduction Factor	M_R
January	12,942 ×	0.98	12683
February		1.00	12942
March		0.98	12683
April		0.96	12424
May		0.94	12165
June		0.92	11906
July		0.90	11647
August		0.88	11388
September		0.90	11647
October		0.92	11906
November		0.94	12165
December		0.96	12424

3.7.1.3 - Open Graded Aggregate

The open graded aggregate layer was placed on top of the dense graded aggregate layer just prior to placement of the first HMA layer. This layer was primarily designed to drain moisture out of the system and does not provide the same supporting strength that the dense graded layer does, mainly due to the lack of aggregate interlock. This material

is composed of crushed virgin limestone material produced in the local area and closely resembles an ASTM D448 Size #67 material containing a small amount of additional material in smaller size ranges. The material was classified as a Poorly Graded Gravel with Sand under the USCS system (Crovetti et al., 2008). Additional material parameters and gradation are shown in Table 3-10 and Figure 3-21.

Table 3-10 – Open graded material properties.

Bulk Specific Gravity	2.743
Apparent Specific Gravity	2.809
Absorption, %	1.34
Max Dry Unit Weight, pcf	128.4
D_{10} , mm	2.75
D_{30} , mm	5
D_{60} , mm	9
C_u , mm	3.27
C_c , mm	1.01

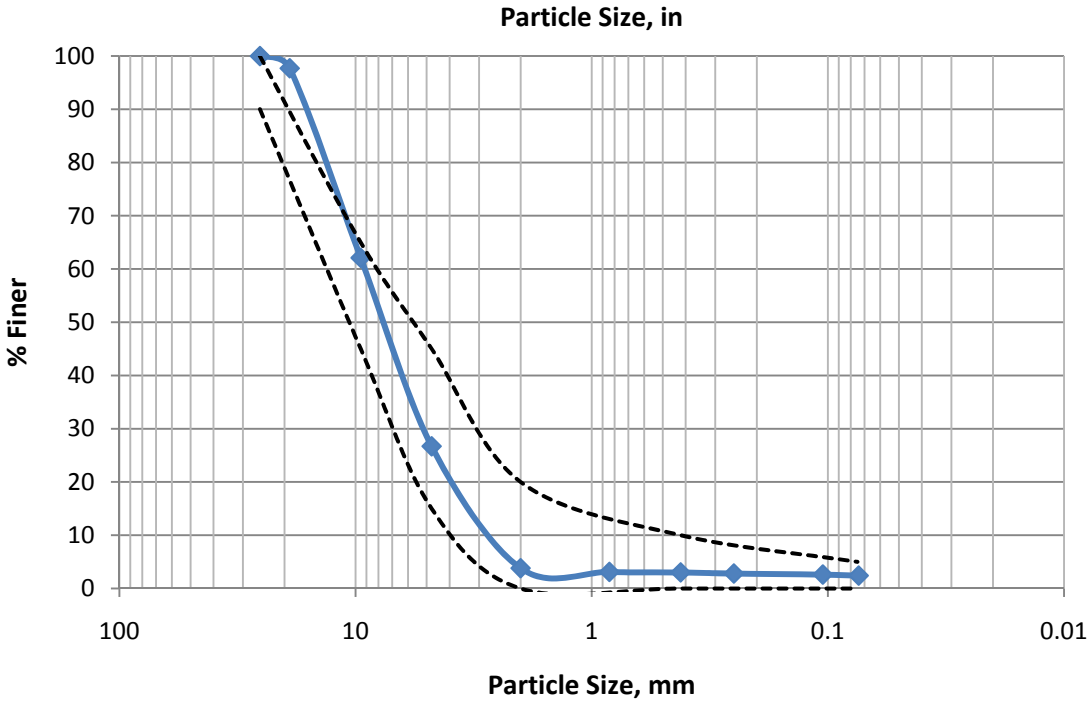


Figure 3-21 - Dense graded particle size distribution.

The stress-stiffening model in Equation 3-13 was produced from resilient modulus testing on the open graded aggregate samples. The model possesses the same form as the dense graded, but with different regression parameters.

$$M_R = 1{,}295\theta^{0.483} \quad \textbf{(3-13)}$$

where M_R = resilient modulus, psi

θ = bulk stress, psi

The aforementioned assumptions for the dense graded aggregate layer apply to this material as well; i.e., it is assumed that the layer will not be saturated for the time period under study and there will be a slight variation in the stiffness of the material due small changes in moisture content. The material is modeled as linear elastic and the stiffness is varied in the same fashion as the dense graded layer. Additionally, the same peak and weak months apply based on the summary of the moisture contents. The resulting resilient modulus values for analysis are tabulated in Table 3-11.

Table 3-11 - Open graded resilient modulus values.

Month	Peak Stiffness, psi	Reduction Factor	M_R
January	4616 ×	0.98	4523
February		1.00	4616
March		0.98	4523
April		0.96	4431
May		0.94	4339
June		0.92	4246
July		0.90	4154
August		0.88	4062
September		0.90	4154
October		0.92	4246
November		0.94	4339
December		0.96	4431

The bulk stress was calculated from the unit weights of the materials above the layer and an average vertical stress due to wheel loads of 6 psi. The unit weights, depths, and calculations for each material and the resulting resilient modulus value can be found in Table 3-8 (pg. 88). Similar to the dense graded aggregate layer, the at-rest lateral earth pressure coefficient, $k_{o,}$ was assumed to be 0.45.

3.7.1.4 - Native Soils

The native soil layer was exposed during the demolition of the existing pavement structure and the proposed vertical alignment required excavation of the native soils. The select crushed material was placed directly on top of this native layer. Due to site access restrictions, samples of the soils were not taken directly from the test section. Instead samples were taken adjacent to the areas that were instrumented at representative depths. Samples were taken from the upper 24 inches of the native soil layer. Upon excavation,

the soils varied considerably in the rather small area from clayey to silty in texture (see Table 3-12).

Since there was a degree of variation of the types of soils, additional resilient modulus testing was conducted to quantify any differences in stiffness behavior. Moisture-density curves were developed for native soils as well as the subbase layers (moisture density data for subbase layers have been omitted from this paper) and are important to helping to define seasonal changes in stiffness characteristics.

Table 3-12 - Native soil properties.

Soil	Liquid Limit, %	Plasticity Index, %	Shrinkage Limit, %	Specific Gravity	Optimum Moisture Content, %	Max Dry Unit Weight, pcf	USCS Classification	Group Index	AASHTO Classification
1A	17	3	14.445	2.705	10	127.5	ML / CL-ML Silt with Sand / Silty Clay with Sand	0	A-4
1B	24	8	2	2.870	12.5	119	CL / CL-ML Lean Clay / Silty Clay with Sand	5	A-4
2	NP	NP	NP	2.725	10.5	118.55	ML Sandy Silt	0	A-4

*NP – Non-plastic

Resilient modulus testing of the soils resulted in two different models being developed for the upper and lower layers. Soils 1A and 2 were taken at the higher elevations and were similar in composition while soil 1B was taken deeper and contained more clay. Since soils 1A and 2 were similar, the results of laboratory testing were combined and averaged, while soil 1B was treated independently.

The characterization report recommended stress-stiffening models for both of the native soil layers. An analysis was done using a stress-softening model, but it was confirmed that stress-stiffening model was most adequate. The models developed for both layers were significantly different in terms of stiffness with the lower layer showing a reduced stiffness (Equations 3-14 and 3-15).

Upper soil layer

$$M_R = 6{,}085\theta^{0.290} \quad \textbf{(3-14)}$$

Lower soil layer

$$M_R = 1{,}595\theta^{0.510} \quad \textbf{(3-15)}$$

where M_R = resilient modulus, psi
θ = bulk stress, psi

The calculation of the bulk stress and the resulting stiffness for both layers is shown in Table 3-8 (pg. 88). A deviator stress of 1 psi due to wheel loads was included in the estimate of stiffness. The stiffness of these layers are dependent upon the moisture content of the soils and a model developed for MEPDG was used to model the seasonal variations in stiffness.

The model (Equation 3-16) developed for MEDPG was based off of past work by (National Cooperative Highway Research Program, 2004) and was calibrated using pavement sections taken from the Long Term Pavement Performance (LTPP) database. The model uses a stiffness ratio and the degree of saturation to predict the resilient modulus of a soil – as the degree of saturation increases, the stiffness of the soil decreases.

$$log\frac{M_R}{M_{Ropt}} = a + \frac{b-a}{1+e^{[\beta+k_S(S-S_{opt})]}} \quad \textbf{(3-16)}$$

where M_R = resilient modulus, psi

M_{Ropt} = resilient modulus at optimum moisture content, psi

S = degree of Saturation

S_{opt} = degree of Saturation at optimum moisture content

$a = Min\left\{log\left(\frac{M_R}{M_{Ropt}}\right)\right\}$

$b = Max\left\{log\left(\frac{M_R}{M_{Ropt}}\right)\right\}$

$\beta = ln\left(-\frac{b}{a}\right)$

k_s = regression parameter

The literature suggested values a, b, β, and k_s for both course and fine grained materials – the values for the fine grained materials has been implemented in this study and are shown below in Table 3-13. The resilient modulus testing for the soils in this study were conducted at the optimum moisture content (determined from moisture-density analyses) and the volumetric moisture content of the soils is measured at the test section. With this information, the resilient modulus can be predicted at a degree of saturation, S.

Table 3-13 - Resilient modulus prediction equation parameters.

Parameter	Value
a	-0.5934
b	0.4
β	-0.3944
k_s	6.1324

The materials characterization produced the gravimetric optimum moisture content and the soil moisture probes implemented at the site produce moisture data in terms of volumetric moisture content - however the prediction model uses the degree of saturation for predictions. In addition, the in-place properties of the soil must be known in order to determine the relative compaction in the field and volumetric properties required for calculating the degree of saturation, S.

Soil properties were measured in-place via nuclear density testing during installation of the earth pressure cells. The bulk density, dry density and moisture content were measured and recorded (Table 3-14). These values were used to estimate the in-place volumetric and gravimetric properties of the soil. Overall, the average dry density of the soils were close to the maximum dry unit weight determined from the moisture-density results from the characterization study and it was assumed that the resilient modulus values for the soils represented the stiffness of the soils in the field.

Table 3-14 – In-place measured soil properties.

Property	Soil 1A and Soil 2 - Upper Soils			Soil 1B - Lower Soils
	Test #1	Test #2	Average	Test #1
Unit Weight, PCF	143.7	144.6	144.15	146.2
Moisture Unit Weight, PCF	19.8	21.3	20.55	18.8
Moisture Conent, %	16.0%	17.3%	16.6%	14.8%
Dry Unit Weight, PCF	123.9	123.3	123.6	127.4

Using the measured field values in Table 3-14 the porosity of the soils were estimated as closely as possible – though there is likely some error due to lack of precision of nuclear density testing. The upper and lower soil porosities were estimated to be 33.0 % and 30.1% respectively. The porosity was integral for calculating the degree of saturation from the volumetric moisture content measured in the field and was calculated using Equation 3-17. The degree of saturation at the optimum moisture content from the characterization report was found to be 74.8 % and 71.1 % for the upper and lower soils, respectively. Table 3-15 is a summary of the calculated values used for implementation.

$$S = \theta\left(\frac{1}{n} - 1\right) \qquad \textbf{(3-17)}$$

where S = Degree of saturation of soil

$$= \frac{Volume\ of\ Water}{Volume\ of\ Voids}$$

θ = Volumetric moisture content of soil

$$= \frac{Volume\ of\ Water}{Volume\ of\ Solids}$$

n = Porosity of soil

$$= \frac{Volume\ of\ Voids}{Total\ Volume}$$

Table 3-15 - In place soil properties.

Soil	n, %	S_{opt}, %	M_{ropt}, psi
Upper	33.0	74.8	11267
Lower	30.1	71.1	4717

The soil moisture data used for estimating the dense and open graded layer resilient moduli was also used to evaluate the variations in stiffness for the native soil layers. Since the soil moisture does not vary greatly on any given day and varies slightly on a monthly basis, a stiffness was selected for each month of the year for use in the FE analysis. The prediction model (Equation 3-16) was used along with the corresponding values from Tabel 3-14 and Table 3-15 to estimate the stiffness for the monthly period of interest. Table 3-16 shows the resulting stiffness profiles for the 12-month period.

The relative deviations in subgrade stiffness have a small effect on the resulting solutions found from FE analysis, mainly due to the depth below the pavement surface (and loading) and the robustness of the pavement structure. Inspection of the soil temperature in the subgrade indicates that freezing temperatures have occurred in only a few instances and lasted briefly.

Table 3-16 - Native soil layer resilient modulus values.

Month	Lower Native Soil Layer M_{ropt} = 4,717 psi, S_{opt} = 71.1%			Upper Native Soil Layer M_{ropt} = 11,267 psi, S_{opt} = 74.8%		
	θ, %	*S%*	M_r	θ, %	*S%*	M_r
January	39.6	91.96%	3625	37.4	75.93%	11806
February	39.2	91.03%	3679	36.9	74.92%	12097
March	39.4	91.50%	3652	37.3	75.73%	11864
April	40.2	93.35%	3550	38.7	78.57%	11084
May	41.6	96.61%	3397	40.9	83.04%	9994
June	43.4	100.79%	3240	43.6	88.52%	8897
July	44.9	104.27%	3136	45.8	92.99%	8192
August	45.9	106.59%	3079	47.0	95.42%	7873
September	45.6	105.90%	3095	46.5	94.41%	8001
October	41.3	95.91%	3428	41.5	84.26%	9727
November	40.6	94.28%	3504	39.8	80.81%	10517
December	39.5	91.73%	3638	37.9	76.95%	11522

The analysis program accesses the above stiffness data, as well as stiffness for all the unbound pavement layers, via a database that contains the stiffness for each month and each material. Timestamps associated with the data from the test section are used to determine the respective month.

It should be noted that temperature was neglected in the stiffness evaluations for all of the unbound layers. This was done to simplify estimations of the stiffness and because the effect of freezing is likely minimal in this system. Since the upper subbase layers are well drained, it is assumed that there is not sufficient moisture to cause a large increase in stiffness because of freezing. Furthermore, as indicated by Figure 3-22, the native soil layer rarely experiences freezing temperatures. These assumptions have been noted and are considered upon analysis of recorded stress and strain measurement from the test section. In the event of frozen layers, the overall increase in section stiffness should cause a reduction in the strain at the bottom of the HMA layers coupled with a reduction in the vertical stress in the base and native soil layers.

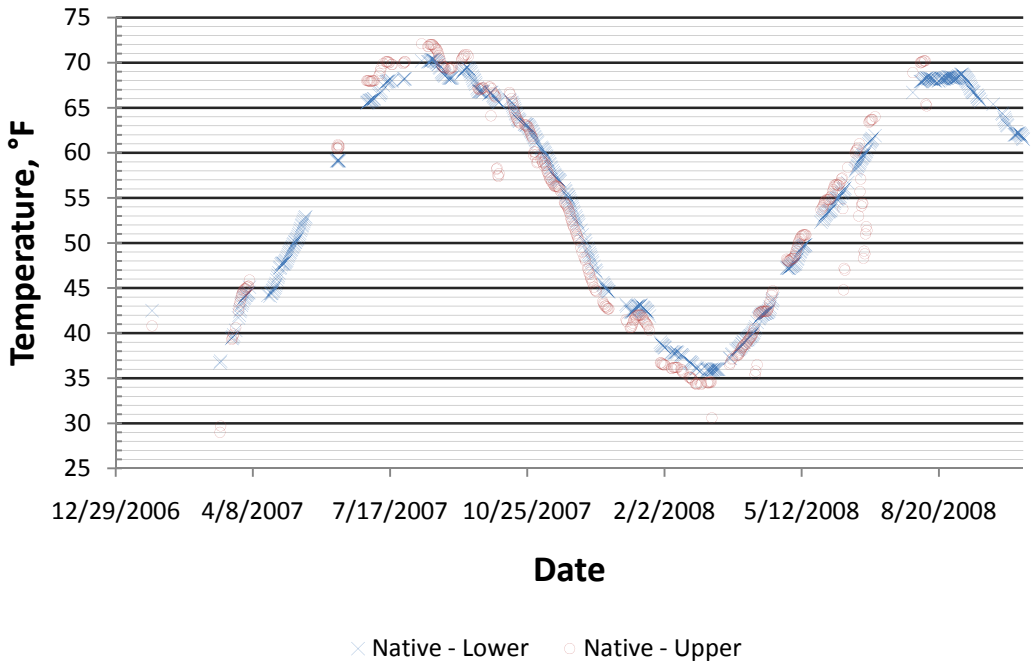


Figure 3-22 - Native soil temperature variation.

3.7.2 - Hot Mixed Asphalt

The stiffness of HMA is typically load rate and temperature dependent. The nature of HMA is quite complex due to the mixture of the aggregates and the bitumen binding agent. This usually limits the measurement of the material strength properties to a macro level, and the individual constituent properties cannot be easily combined to predict the overall material properties. However, the current technology has allowed researchers to model the HMA stiffness throughout a regime of load rates and temperatures with a reasonable degree of accuracy. The downside is that the testing can be quite complicated, time consuming, and costly.

Luckily for this project the materials characterization phase included a full analysis of the HMA materials. Since the pavement consists of 13 inches of HMA (4

compacted layers, 3 distinct layer types), the majority of the pavement strength is derived from the HMA and hence the need for accurate characterization.

3.7.2.1 - Dynamic Modulus

The dynamic modulus takes into account the load rate and temperature dependency of the material. Typically, during modulus testing, a cylindrical HMA specimen is subjected to vertical sinusoidal loading of different frequencies, each set of which is completed under a range of temperatures. During the testing, the load pulse and the resulting strain in the material is recorded - the resulting model (Equation 3-18 to Equation 3-20), referred to as the master curve, can be used to describe the stiffness for a particular rate of loading and a particular temperature. Individual master curves at each temperature are developed during testing. These are then shifted to form one sigmoid, called the master curve. The individual curves are shifted using time-temperature superposition, and the amount of shifting is based on temperature dependency of the material (National Coorperative Highway Research Program, 2004).

$$log(E^*) = \delta + \frac{\alpha}{1+e^{\beta+\gamma[log(t_r)]}} \quad \textbf{(3-18)}$$

where E^* = degree of saturation of soil

t_r = time of loading at reference temperature, Equation 3-19

$\delta, \alpha, \beta, \gamma$ = fitting and regression parameters

$$log(t_r) = \log(t) - c\left[\log(\eta) - \log\left(\eta_{T_r}\right)\right] \quad \textbf{(3-19)}$$

where t_r = time of loading at reference temperature

t = time of loading at temperature of interest

η_{T_r} = binder viscosity at reference temperature

η = binder viscosity at temperature of interest, Equation 3-20

c = regression parameter

$$loglog(\eta) = A + VTSlog(T_R) \quad \textbf{(3-20)}$$

where η = binder viscosity at temperature of interest

T_R = temperature, Rankine

A = regression intercept

VTS = regression parameter of viscosity-temperature susceptibility

The time of loading is transformed from the time domain to the frequency domain by simply taking the inverse of the load frequency. This has recently been the subject of discussion since it is thought that method is inaccurate and a better representation can found by using the angular velocity for conversion or even applying spectral analysis (Al-Qadi et al., 2008) (Dongre et al., 2006) (Thompson et al., 2006)

For a more thorough discussion of master curve development, refer to Part 2, Chapter 2 of the Guide for Mechanistic Empirical Design of New and Rehabilitated

Pavement Structures (National Coorperative Highway Research Program, 2004) and the testing protocol set forth by AASHTO (2003).

Due to the visco-elasticity of the binding agent used in the material, there is a time lag between the induced stress pulse and the resulting strain pulse. This is caused by the damping action of the viscous component of the binding agent. At faster load rates, the viscous damping effect causes the material to possess an increased stiffness. Slower load rates result in the opposite effect, causing a reduction in the stiffness of the material.

At higher temperatures, the viscosity of the binding agent decreases reducing the damping potential of the viscous component of the stiffness. This results in an overall reduction in the stiffness of the material and also effects the magnitude of the effect the load rate has on the material. The dynamic modulus represents these material characteristics.

The materials characterization study included testing of the three HMA mixtures used for construction of the pavement. The bottom layer consists of a four inch thick 19.0 mm binder rich base layer containing recycled asphalt pavement (RAP), a seven inch thick 19.0 mm intermediate layer with standard binder content, and a two inch thick surface layer composed of a 12.5 mm stone matrix asphalt (SMA) mixture (Figure 3-23).

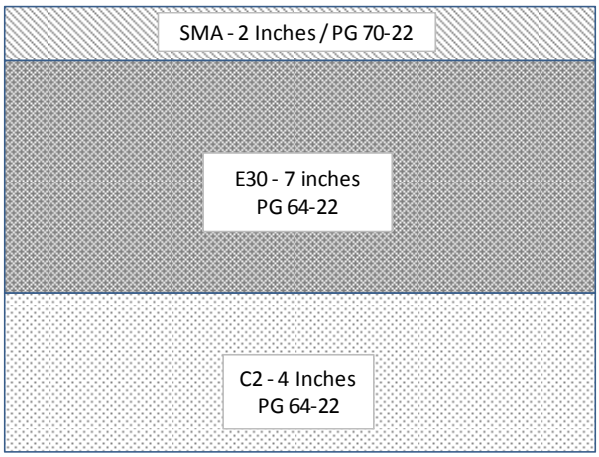


Figure 3-23 - HMA pavement structure and binder types.

The following master curve parameters, shown in Table 3-17, summarize the findings of the dynamic modulus testing conducted on the materials. These constants have been incorporated into the FE analysis program and using Equations 18 through 20. Only the pertinent constants have been included in this work. The complete data, along with data for all of the pavement materials, can found in the materials characterization report (Crovetti et al., 2008). Additionally, complete aggregate gradation, binder data, etc., for the HMA materials are provided in Appendix B.

Table 3-17 - Master curve and binder viscosity parameters.

Mixture	δ	α	β	γ	c
SMA	4.1560	2.4244	-0.3918	0.4794	1.6996
E30	2.5566	4.2125	-1.5623	0.3937	1.6214
C2	1.6841	5.1418	-1.7168	0.3650	1.6905

Asphalt Binder	A	VTS
PG 64-22	9.7732	3.2637
PG 70-22	8.4589	2.7813

3.7.2.2 - Method of Equivalent Thickness

The aforementioned data regarding the development of the HMA stiffness has been integrated into the analysis program to accurately calculate stiffness of the HMA materials for any particular loading and environment.

The MET process, developed by Odemark (Odemark, 1949) can be used to solve for the time of loading based on the relative stiffnesses of the material, assuming a stress distribution through the entire pavement structure. For example,

Figure 3-24 shows a typical pavement structure with an applied load at the surface and stress distribution through the thickness indicated by the dashed lines.

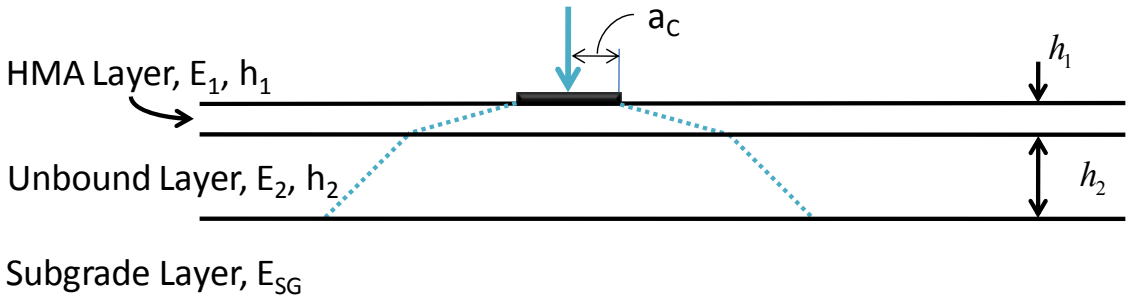


Figure 3-24 - Untransformed section.

The stiffest layers can then be transformed and given a stiffness modulus equal to the subgrade layer, while still maintaining the same overall section stiffness, which results in an increase in the total thickness of the upper layers. The thickness transform is given by Equation 3-21.

$$h_{ie} = h_i \sqrt[3]{\frac{E_i}{E_{SG}}} \quad \textbf{(3-21)}$$

where h_{ie} = equivalent thickness of layer i

h_i = actual thickness of layer i

E_i = stiffness modulus of layer i

E_{SG} = stiffness modulus of subgrade layer

Figure 3-25 represents the resulting pavement structure after transformation. The stress distribution is assumed to be inclined at a 45° – this represents a very broad value of stress distribution given the common shear strength of most soils. Since the upper layers are transformed and are assumed to have the same stiffness as the subgrade, this stress distribution is carried through to the surface of the structure.

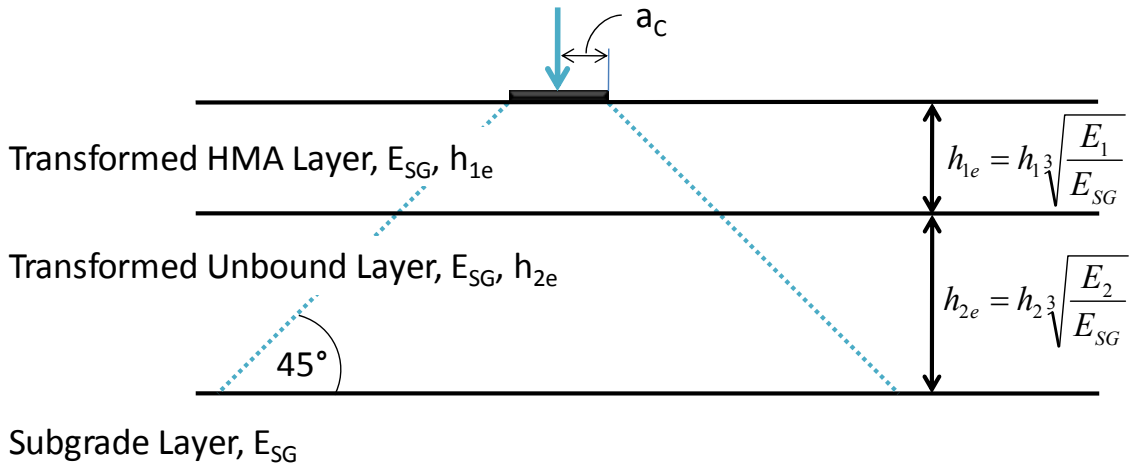


Figure 3-25 - Transformed pavement structure.

Using the assumed stress distribution, the effective length of the influence of the stress on the soil can be calculated at any given depth (referred to as the effective depth) in the structure using the geometry of the transformed section (Figure 3-26). Then

knowing the speed of the moving load, the time that a particular point is influenced by the stress can be calculated by Equation 3-22.

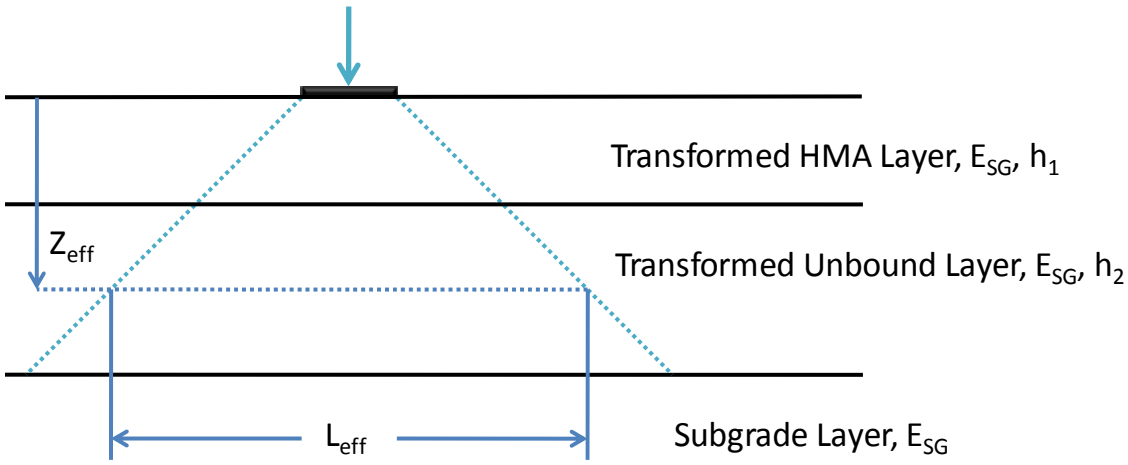


Figure 3-26 - Effective depth and effective length.

$$t = \frac{L_{eff}}{17.6v} \qquad \textbf{(3-22)}$$

where t = Time of loading, seconds

L_{eff} = Effective length, inches

v = Vehicle speed, mph

The process is further complicated by the addition of tandem, triple or other load groups which cause an overlapping of the stresses in the pavement structure as shown in Figure 3-27. This has been handled in MEPDG by applying traffic multipliers to the calculated load times.

In the case of axle groups close to the pavement structure, there is no overlap of stresses, so the time of loading is the same for each axle and there exist two distinct load

pulses (multiplier = 2). Deep in the structure, there will be full overlap of the stresses which results in one large stress pulse (multiplier = 1) with an extended time of loading. In between these two cases, there will be partial overlap. MEPDG has used a scheme that calculates the load time between the two stress pulses and applies a traffic multiplier which is a straight line interpolation between the no-overlap and full-overlap conditions (e.g. between 1 to 2 pulses for a tandem axle, 1 and 3 pulse for a tridem axle, etc.) (National Coorperative Highway Research Program, 2004).

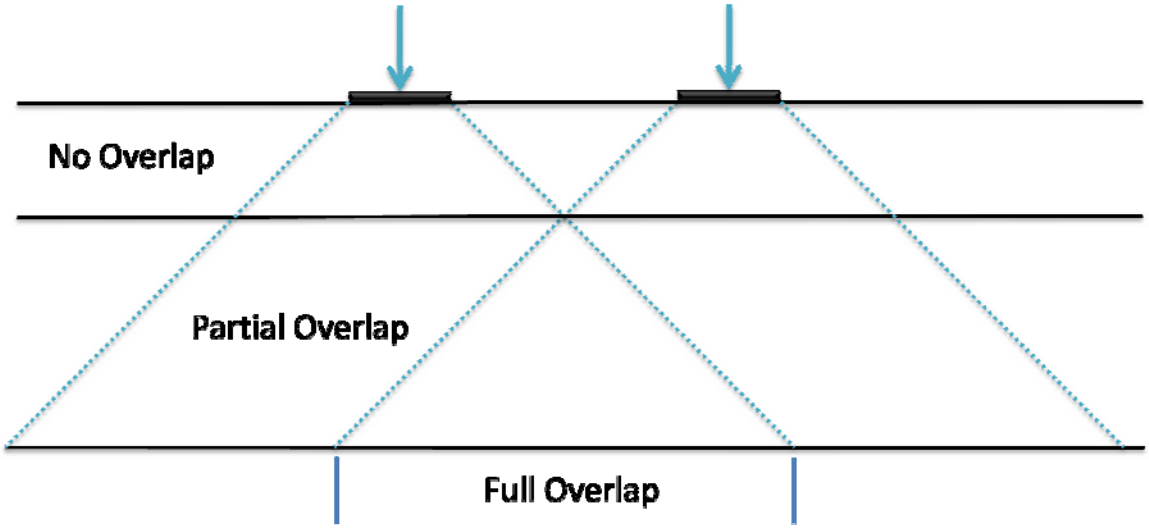


Figure 3-27 - Stress overlap for tandem axles.

Importantly, the researchers found that even while taking into account the overlap conditions, the time of loading is relatively insensitive to axle type as shown in Figure 3-28. However, the traffic multipliers are used for analysis.

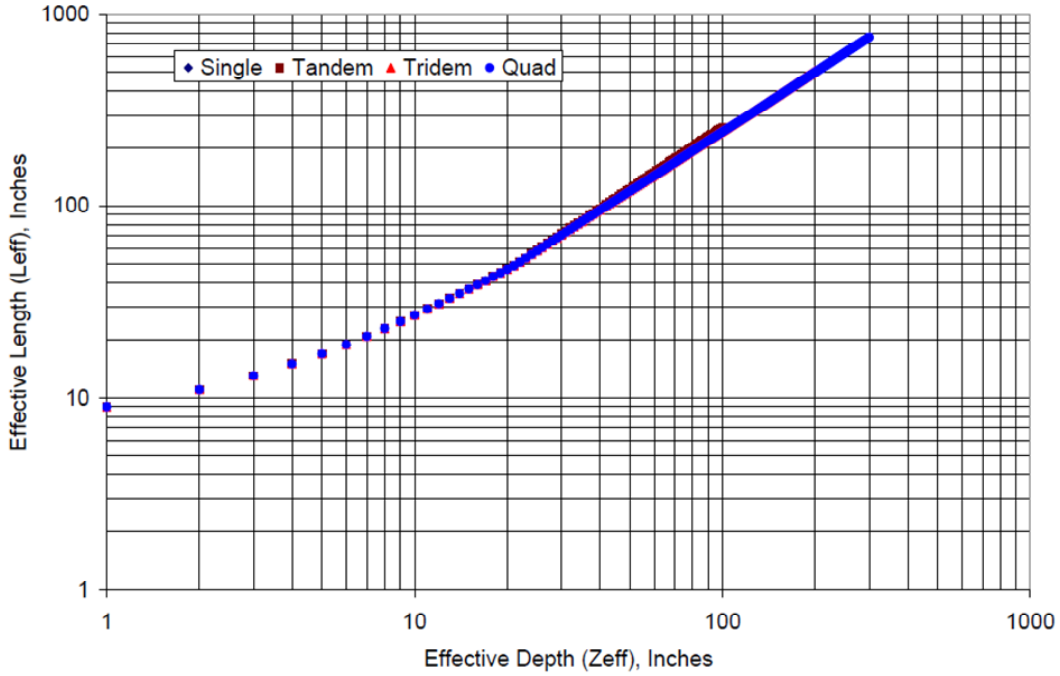


Figure 3-28 - Effective length calculations (National Coorperative Highway Research Program, 2004).

3.7.2.3 - Analysis Implementation

The MET process has been integrated into the analysis program in conjunction with the dynamic modulus computations in order to solve for the HMA stiffnesses. The MET process solves for the load time assuming a single axle based on the findings reported by the MEPDG Appendix CC. However, no multiplier has been applied in the routine since the analysis is done on a per-axle basis. In addition, the actual load times have been calculated from Phase II of the Marquette Interchange Instrumentation Project (Hornyak & Crovetti, 2008) and a preliminary analysis of the load times was recently published (Hornyak & Crovetti, 2009). An investigation into the adequacy of the load times is warranted and is investigated in the following chapters.

The stiffness for the HMA materials is solved in the analysis program following the steps below. The classes within the C# program that carry out the following computations are titled 'DynamicModulus' and 'METIterator' and are contained in Appendix A.

1. Download temperature data from environmental database.
2. Calculate layer temperatures using recorded ambient air temperature and regression model (developed in Chapter 4).
3. Calculate viscosities at reference temperature (70°F) and temperature of interest using Equation 3-20 (pg. 103).
4. Solve for HMA stiffnesses using dynamic modulus data.
5. Calculate time loading using MET.
6. Calculate time of loading at the reference temperature using Equation 3-19 (pg. 103).
7. Calculate stiffness using Equation 3-18 (pg. 102).

The above procedure has one caveat - the MET process relies on the material stiffness to estimate time of loading. This requires and iteration between the dynamic modulus calculations and the time of loading from MET. Initially the process uses a default time of loading of 0.05 seconds to initiate the iteration process. The resulting time of loading is used to repeat steps 4 through 7 until the stiffness converges to 0.1% of the previous iterations value. This can typically be achieved within 10 cycles – no maximum number of iteration cycles has been included in the program.

Because of the distribution of stress through the thickness of the pavement, the entire 13 inch HMA section has been subdivided into one inch intervals. The solver begins with the upper layer, iterates for the solution, and then continues to the following layers. Once the layer is solved, it does not have to be re-solved in the next iteration. The entire iteration of the HMA layer takes milliseconds to solve on an average personal

computer. The stiffness values are then stored and passed on to the finite-element engine for solution.

Chapter 4 - Results

The following sections discuss the results of the data analyses performed and the interpretations and conclusions that can be drawn from the experiment. The focal areas that are discussed are the following: environmental data, structural response of the pavement system, load duration, and fatigue.

4.1 - Environmental Models

Part of the research objectives for this project was to obtain enough environmental data over the course of the project so that future conditions could be modeled. It was also understood that periods would exist where failed equipment may cause gaps in the data. The most important environmental data for pavement analysis, based on past research, are temperatures throughout the pavement structure and the moisture content of the unbound layers.

Temperature has a profound effect on the stiffness of HMA materials and unbound layers can become frozen, increasing stiffness. Moisture has virtually no effect on the properties of HMA materials short term, but unbound layers can become weakened with variations in moisture content.

For these reasons, temperature probes have been installed throughout the pavement structure of the test section. To supplement the pavement temperature data, the ambient air temperature, pavement surface temperature, and shortwave solar radiation are also measured. Soil moisture probes have been installed into the native soils layer. The remaining unbound aggregate layers above the native soils layer and below the pavement are highly permeable and should be virtually moisture insensitive and thus did not require

measurement of moisture content in those zones. Details of the instruments can be found in Chapter 3.

In addition to the need to model future conditions of the pavement test section, periods of time existed where instruments had failed and no data was recorded. The environmental models would serve to provide the best representation of the missing data for the purpose of pavement analysis.

4.1.1 - Model Data

The data selected for the models consisted of a period where all instruments were known to be functioning properly and accurately. The specific dates for the data range from September 26th, 2007 to December 12th, 2007 and include 5,361 samples (each sample representing one row of data which includes temperatures, pyranometer data, et cetera). The ambient air temperatures for this particular period ranged from a low of 7.6 F to 81.1 F.

More data would have been included, but failure of one or more instruments caused difficulties in creating an adequate model. Initial linear regression models suggested that the ambient air temperature, surface temperature, and solar radiation were all significant predictors of the pavement layer temperature. These initial conclusions were somewhat expected given the known factors of heat dissipation through the HMA layers. In addition, these three properties can easily be measured at the test section because they are measured external to the pavement.

Wind speed data is available and was applied to the determination of the pavement surface temperature data. However, the instrument used for measuring surface

temperature incurred damage from the enironment and frequently required service. The quality of some of the data was rendered as questionable and little data was available that could be considered usable.

Model adequacy checks for the linear regression, including wind speed, indicated wind speed was not significant in determining surface temperature. Future measurements may allow a feasible model to be developed and it should be reiterated that pavement surface temperature is crucial for modeling pavement layer temperatures. However, general observations indicate that wind speed may have little effect on the pavement surface temperature for this specific case.

Only HMA layer temperatures and surface temperature were considered for the final environmental model generation. Temperatures are not measured in the unbound aggregate layers and have little effect on these layers. The layers are quite porous and it is assumed that excessive moisture is not present at any time, thus is it reasonable to assume that freezing of these layers is not possible.

The native soil layer on the other hand could be affected by temperature since the soil was silty to clayey in consistency. Subjection to freezing temperatures may cause an increase in stiffness of the layer. Inspection of the recorded temperature data for the native soil layers show that temperatures dropped below 32° F only on rare occasions. For this reason, it can be reasonably assumed that the native soil layer will not become frozen and therefore the stiffness of the material has little dependency on temperature.

4.1.2 - Linear Regression Models

The following models were generated using the data analysis tool-pack contained within Microsoft's Excel spreadsheet software. Other variables were included in the model generation, but the following models represent the best fitting found during the analysis while remaining simple.

4.1.2.1 - Pavement Surface Temperature

The significant predictors for surface temperature were found to be ambient air temperature and solar radiation. The following equation is the model generated.

$$\text{Pavement Surface Temperature} = 0.8907 \times AAT + 0.0135 \times SR + 92.2260 \quad \textbf{(4-1)}$$

where AAT = ambient air temperature, °F

SR = shortwave solar radiation, W/m^2

The regression fit statistics are shown in Table 4-1. The model has a good fit to the data indicated by the high coefficient of regression values. All predictor variables were significant at the $\alpha = 0.05$ level (significance of the regressor). A plot of the data used to generate the model and the predicted values are shown in Figure 4-1.

Table 4-1 – Pavement surface temperature model parameters and statistics.

Predictor	t Stat	P-value
Intercept	58.47	<< 0.001
Ambient Air Temperature	254.59	<< 0.001
Solar Radiation	43.25	<< 0.001
R^2	0.937	
Adjusted R^2	0.937	

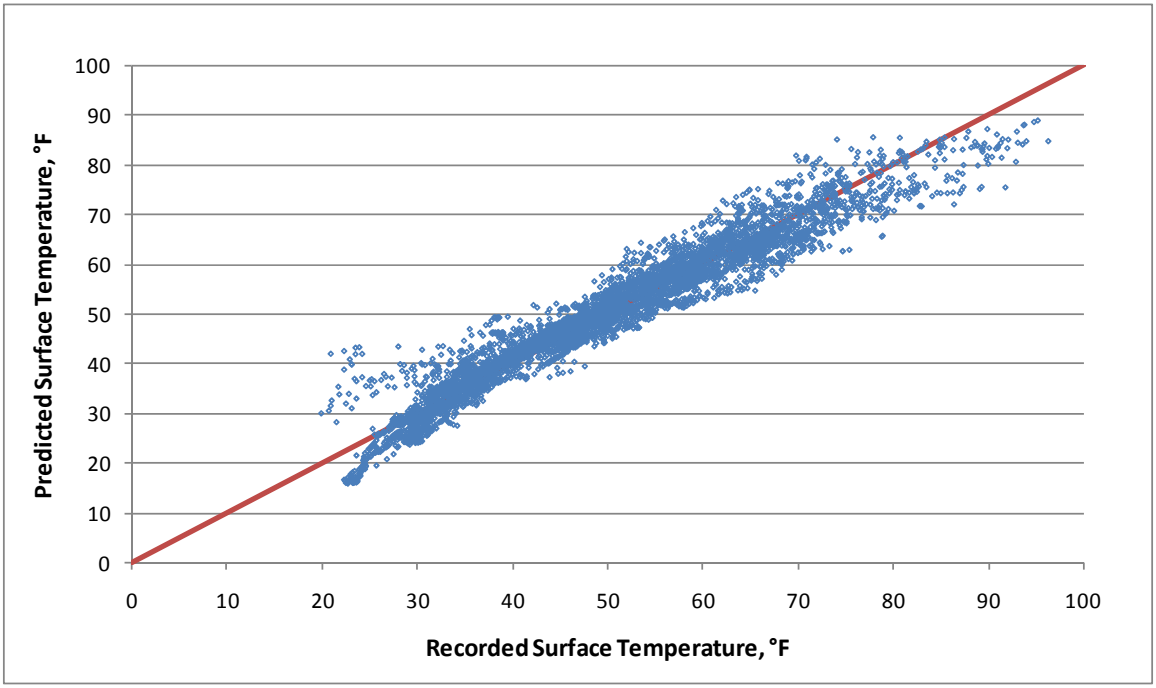


Figure 4-1 – Actual versus predicted surface temperatures.

4.1.2.2 - Pavement Layer Temperature

The layer temperatures are based on values measured at different elevations throughout the thickness of the pavement. The original plans included temperature measurements every inch throughout, however problems with instruments led to a retrofit instrument which measured temperature at the center of each layer. The following linear regression models are the result of the analysis. It was found that ambient air temperature, surface temperature, and solar radiation were significant predictors for each pavement layer temperature. Data used for the models was based on recorded data from the project database. The models take the form of Equation 4-2.

$$T_i = \beta_0 + \beta_1 \times AAT + \beta_2 \times SR + \beta_3 \times PST \qquad \textbf{(4-2)}$$

where T_i = temperature at location *i*, °F

$\beta_0, \beta_1, \beta_2, \beta_3$ = regression coefficients

AAT = ambient air temperature, °F

SR = shortwave solar radiation, W/m^2

PST = pavement surface temperature, °F

The following table contains the regression coefficients and statistics. The high coefficient of regression, R^2, values indicate a good fit to the data. According to the P-values from the regression output, all predictors are significant at the $\alpha = 0.05$ level.

Table 4-2 - Pavement layer temperature model parameters and statistics.

Response Model	Regression Coefficients				R^2	Adjusted R^2
	Intercept	Ambient Air Temperature	Solar Radiation	Pavement Surface Temperature		
Depth = 1.0 inch	14.8342	0.5292	-0.0008	0.4273	0.962	0.962
P-value	<< 0.001	<< 0.001	0.0058	<< 0.001		
Depth = 3.0 inch	18.1805	0.4923	-0.0076	0.4448	0.948	0.948
P-value	<< 0.001	<< 0.001	<< 0.001	<< 0.001		
Depth = 4.0 inch	47.3653	0.4912	-0.0098	0.4237	0.939	0.939
P-value	<< 0.001	<< 0.001	<< 0.001	<< 0.001		
Depth = 7.0 inch	51.6153	0.4541	-0.0149	0.4410	0.911	0.911
P-value	<< 0.001	<< 0.001	<< 0.001	<< 0.001		
Depth = 10.0 inch	72.1011	0.4136	-0.0164	0.4477	0.884	0.884
P-value	<< 0.001	<< 0.001	<< 0.001	<< 0.001		
Depth = 12.0 inch	87.2273	0.3882	-0.0163	0.4520	0.869	0.869
P-value	<< 0.001	<< 0.001	<< 0.001	<< 0.001		

The following figures contain the actual pavement layer temperatures plotted against the predicted values. In addition, the residual plots and normal plots were analyzed for all linear regression models (including pavement surface temperature) and all data appears to be normal distributed.

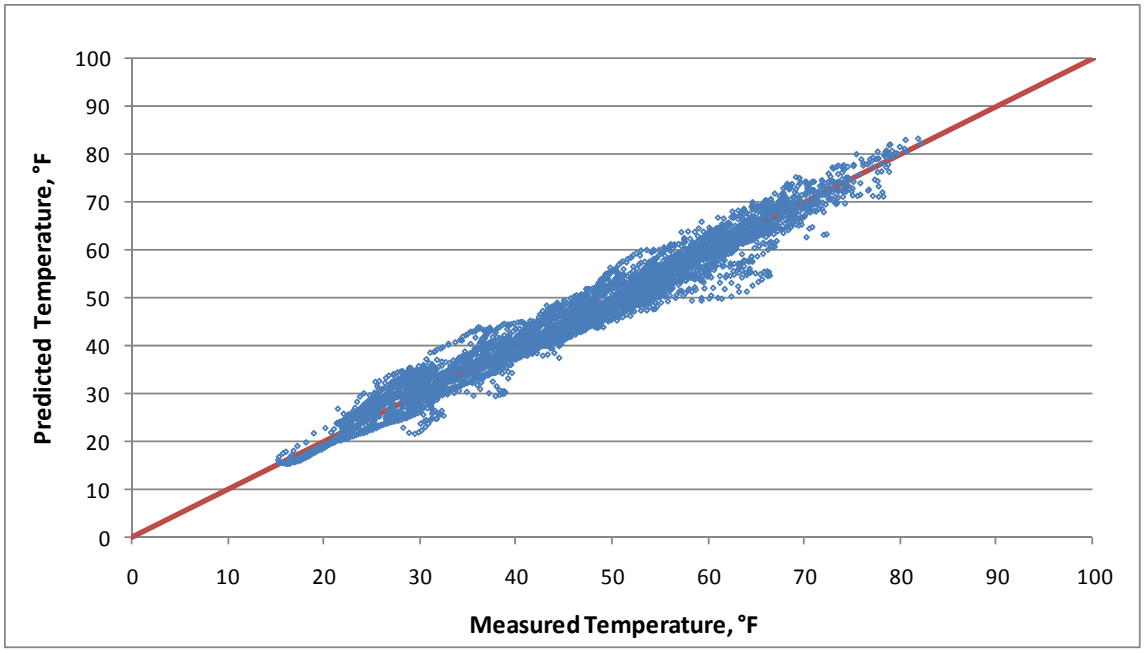


Figure 4-2 – Actual versus predicted pavement layer temperature A0.

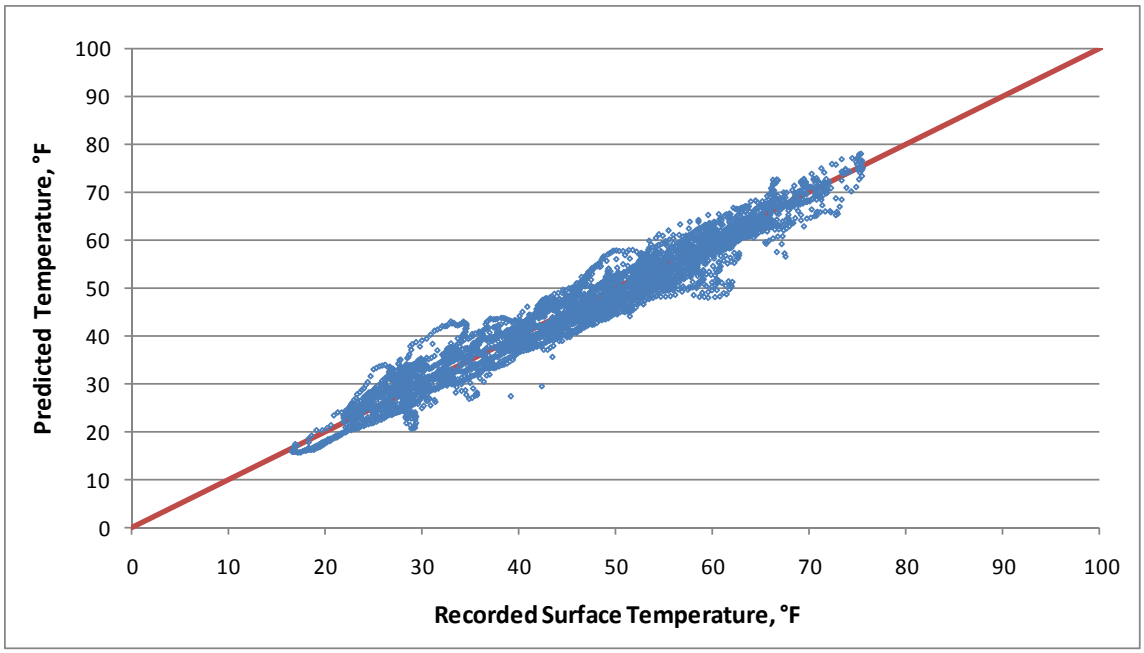


Figure 4-3 - Actual versus predicted pavement layer temperature A1.

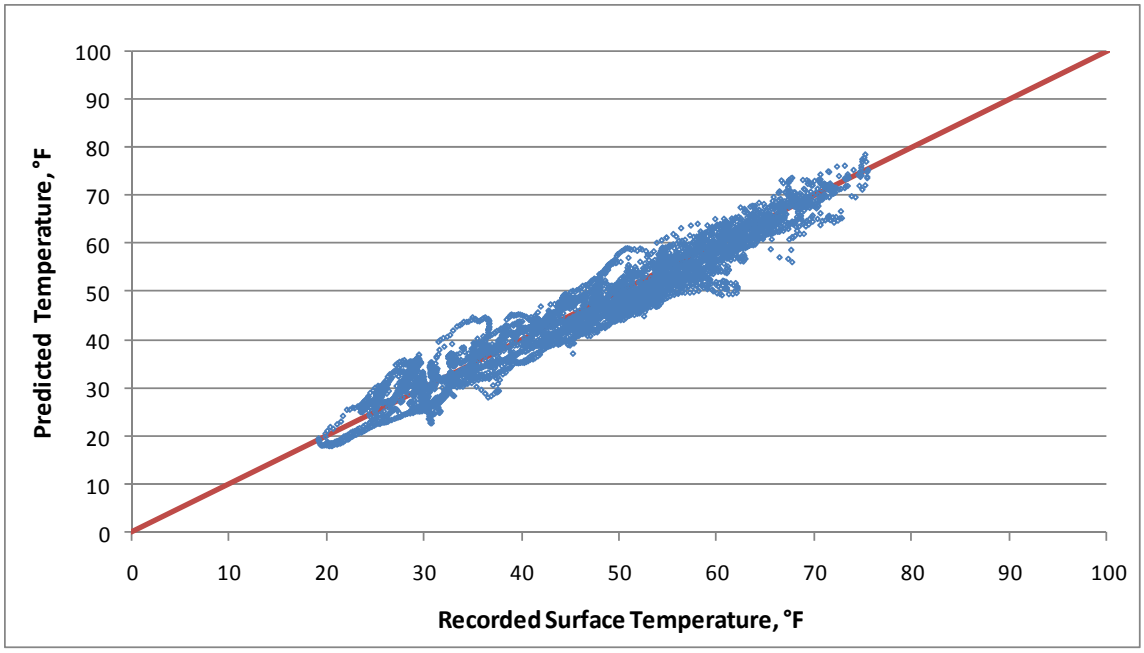


Figure 4-4 - Actual versus predicted pavement layer temperature A2.

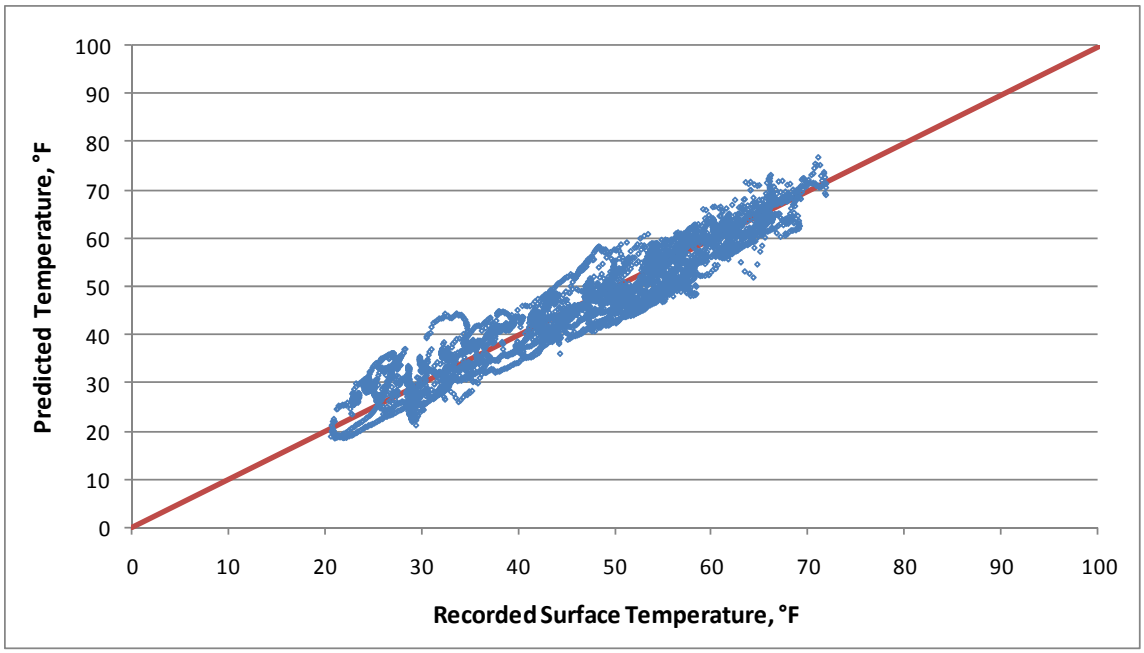


Figure 4-5 - Actual versus predicted pavement layer temperature A3.

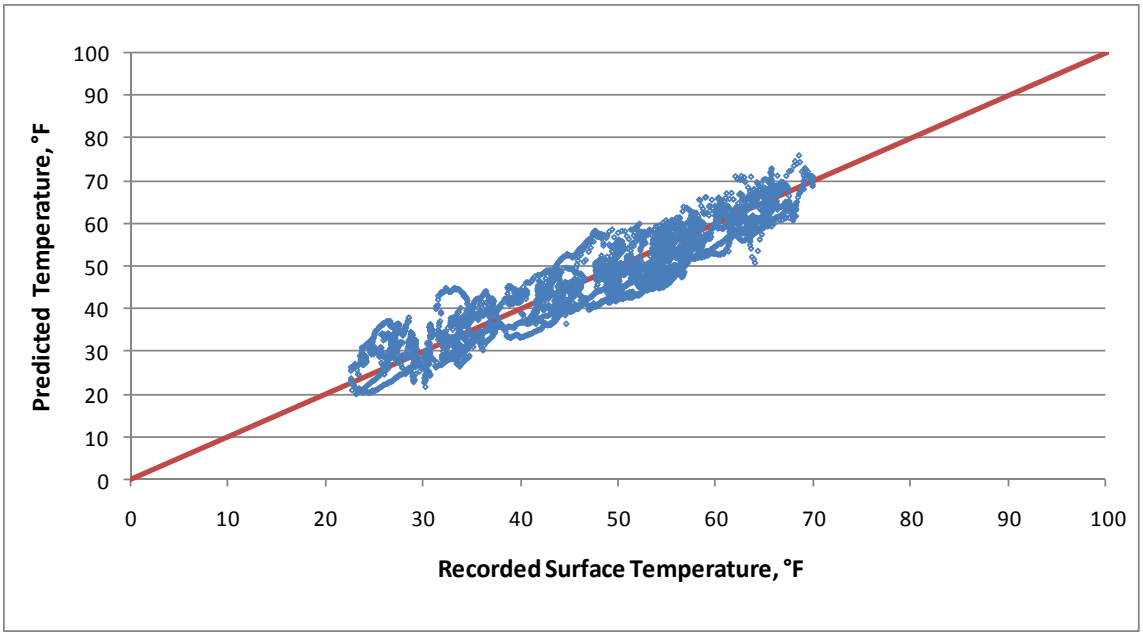


Figure 4-6 - Actual versus predicted pavement layer temperature A4.

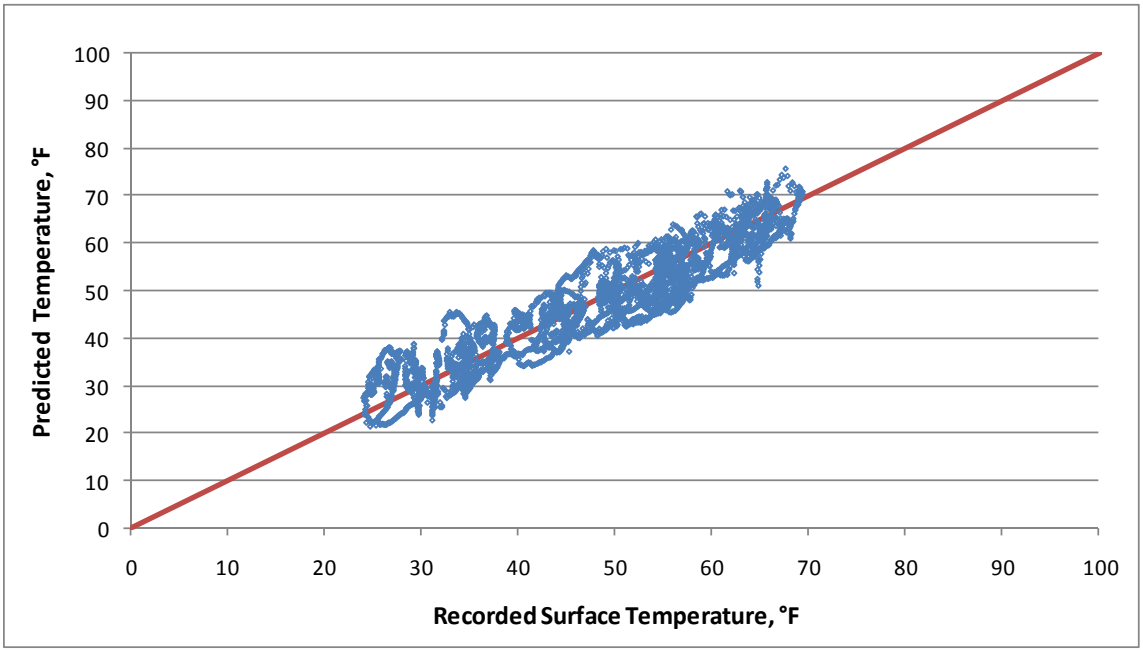


Figure 4-7 - Actual versus predicted pavement layer temperature A5.

The preceding discussion shows that the pavement layer temperature is heavily dependent on the depth below the pavement surface. Two additional models which take depth into account were created and can be used to predicted layer temperatures at any depth. In addition this type of model allows one equation to be used to predict layer temperature rather than using a number of models for each depth.

The models created use the ambient air temperature, pyranometer data, surface temperature, and depth below the pavement surface. The model created using all predictor variables is shown below. This model is not adequate for prediction of layer temperatures within 1-inch of the surface – this region is highly affected by the surface temperature and no experimental data was available to model this region.

$$T_d = 43.8550 + 0.4614 \times AAT - 0.0110 \times SR + 0.4394 \times PST + 0.0762 \times D \quad \textbf{(4-3)}$$

where T_d = temperature at depth d, °F

AAT = ambient air temperature, °F

SR = shortwave solar radiation, W/m^2

PST = pavement surface temperature, °F

D = depth below pavement surface, in.

The associated regression fit parameters are listed in the following table. All regression parameters were found to be significant at the $\alpha = 0.05$ level. The following plot shows the model predictions versus the recorded data.

Table 4-3 – Layer temperature model parameters and statistics.

Predictor	t Stat	P-value
Intercept	46.20	<< 0.001
Ambient Air Temperature	83.11	<< 0.001
Solar Radiation	-68.87	<< 0.001
Surface Temperature	73.35	<< 0.001
Depth	13.72	<< 0.001
R^2	0.912	
Adjusted R^2	0.912	

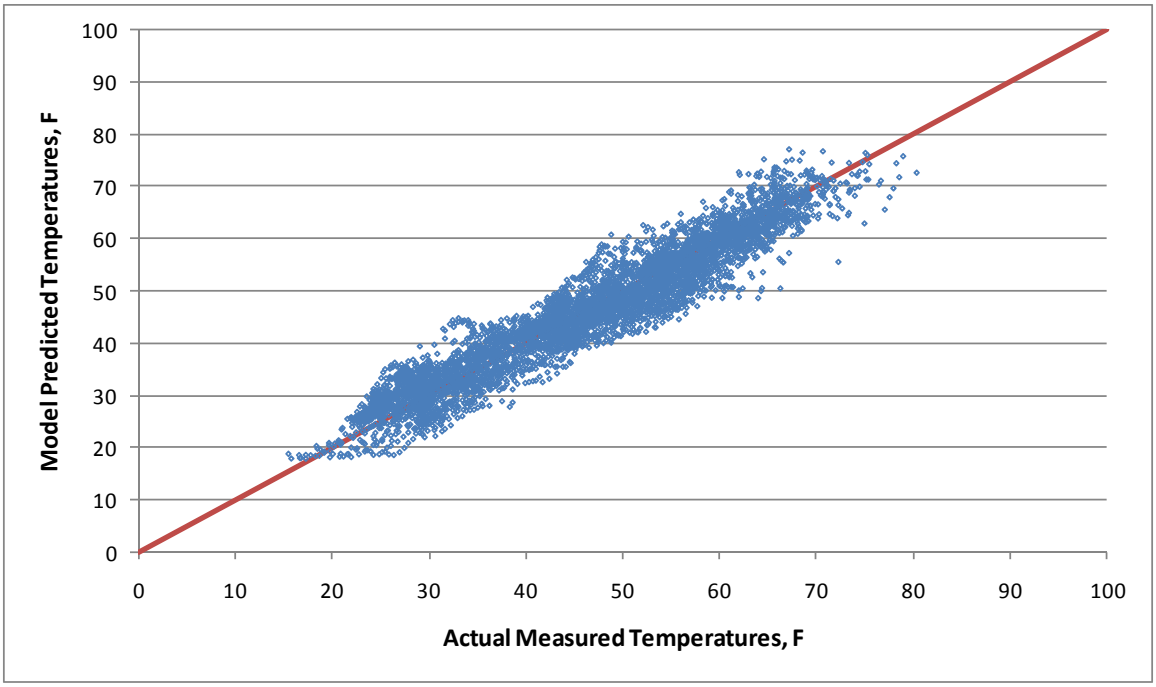


Figure 4-8 – Actual versus predicted layer temperatures for any depth in the HMA layers.

In order to try and simplify the predictions more, a similar model was created which used only the ambient air temperature and the depth below the surface. This was not done only for simplicity, but for the specific reason of predicting layer temperatures where data may have been missing in the Marquette Interchange database. At various points throughout the project, the pyranometers and surface temperature sensors failed

which left a gap in the data. The resulting regression model, fitting parameters and plot are shown below.

$$T_d = 8.6953 + 0.8365 \times AAT + 0.0762 \times D \quad \textbf{(4-4)}$$

where T_d = temperature at depth d, °F

AAT = ambient Air Temperature, °F

D = depth below pavement surface, in.

Table 4-4 – Simplified layer temperature model parameters and statistics.

Predictor	t Stat	P-value
Intercept	102.88	<< 0.001
Ambient Air Temperature	518.17	<< 0.001
Depth	12.48	<< 0.001
R^2	0.893	
Adjusted R^2	0.893	

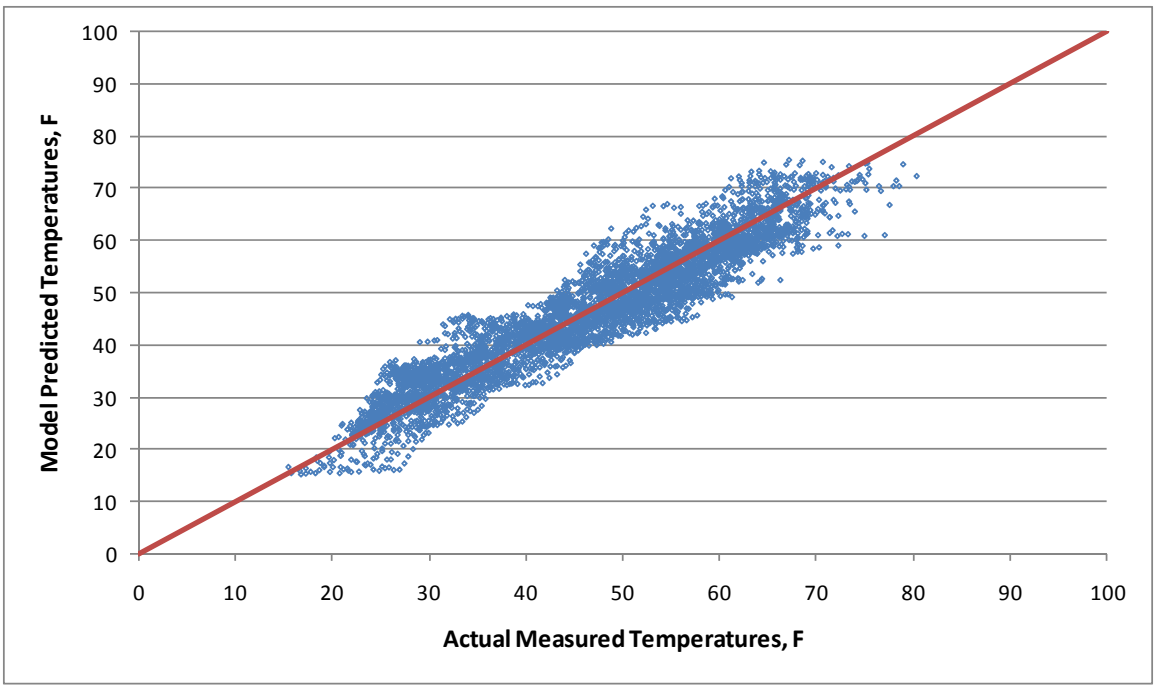


Figure 4-9 – Actual versus predicted layer temperatures for simplified model.

The aforementioned simplified model shows a relatively good fit given the reduced number of prediction variables. The R^2 value for this reduced model is 0.893 while the preceding models all have an average R^2 around 0.910. Given that good performance could be achieved using only the ambient air temperature and depth, the reduced model was chosen for use in analysis of the pavement performance.

This drastically reduced the amount of data-error checking needed in the code to fix missing or erroneous data. The ambient air temperature sensor has been operational throughout the data recording phase of the research and appears to be accurate when compared against other weather data.

As noted before, the layer temperature of the HMA affects the stiffness of the material which requires calculation before any modeling of the pavement structure can

take place. The reduced model (Equation 4-4) has been used in the analysis program for computing performance data.

4.1.2.3 - Combined Model

Since the surface temperature data was not always available, the surface temperature prediction model (Equation 4-1) as substituted into the layer temperature model (Equation 4-3). This allows for calculate of the layer temperature which includes the predicted pavement surface temperature.

$$T_d = 8.4379 + 0.8528 \times AAT - 0.0051 \times SR + 0.0762 \times D \quad \textbf{(4-5)}$$

where T_d = temperature at depth *d*, °F

AAT = ambient air temperature, °F

SR = shortwave solar radiation, W/m^2

D = depth below pavement surface, in.

This model produces very similar results to Equation 4-4, but has a smaller R^2 value of 0.88 compared to an R^2 of 0.90 for Equation 4-4. The additional prediction variable (solar radiation) results from the substitution of Equation 4-1 into the PST variable in Equation 4-3.

4.2 - Load Pulse Duration

The previous chapter detailed the process used to calculate HMA stiffness from the dynamic modulus master curve model. This model requires the load frequency (load

pulse duration) and the temperature in order to predict the material stiffness. One caveat that exists is to properly select, or predict, the load pulse duration at the depth of interest.

Prior work (Hornyak & Crovetti, 2009) initially demonstrated that the load pulse duration determined through the Method of Equivalent Thickness (MET) may have shortcomings. Load pulse duration data from this project was compared to estimated load durations calculated through MET. It was found that the measured load pulse durations were in general smaller (shorter in duration) than the durations approximated by MET for horizontal strain and longer for measured vertical pressure. MET best describes the case of vertical pressure since the vertical load contained no stress reversals whereas horizontal strain contained stress reversal due to the moving load, especially for longitudinal, horizontal strain. The load time extracted for the project data only calculates the duration that the pavement is in tension for the case of horizontal strains.

The likely source of error that causes this discrepancy between the actual and MET load time is partly due to the broad assumption made within MET theory. The two main factors are the transformation of pavement layers to an equivalent section and the assumption that the stress distribution acts at a 45 degree angle from horizontal (Figure 4-10). The transformation of the section thicknesses should be adequate for these computations; however depending upon the strength properties of the materials, the assumed stress distribution angle may not be adequate and can be regarded as a large global assumption of common soil properties.

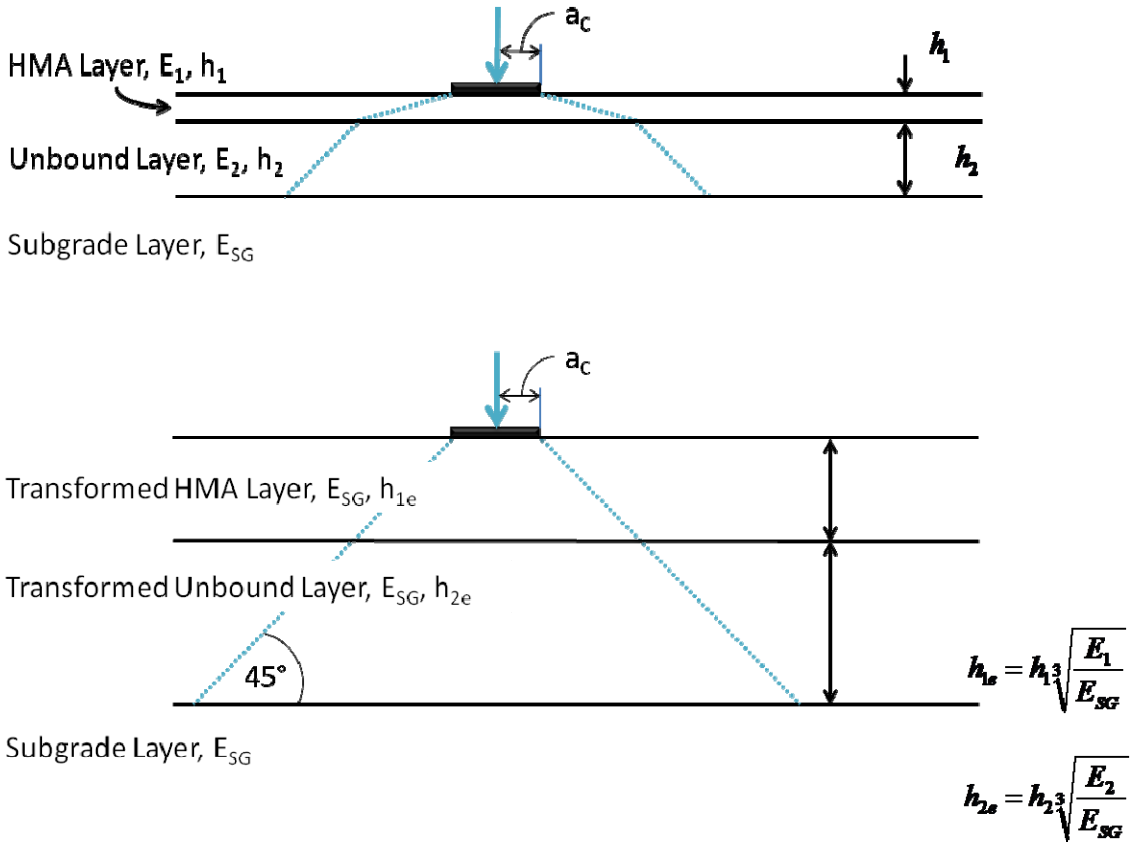


Figure 4-10 - Untransformed and transformed sections for MET computations.

In order to account for these deficiencies, the actual and MET load times have been analyzed and a linear regression has been performed to model the correction. Furthermore, a stress distribution angle has been proposed to more accurately predict the load times for the horizontal strain case.

4.2.1 - Base and Subgrade Pressure

As previously stated for most pavements, the method of equivalent thickness (MET) procedures best describes load pulse duration for the case of vertical pressure in the pavement structure. The effect of a rolling wheel load on the pavement can be thought of in terms of a continuous beam with supports along its length – as shown in

Figure 4-11. Since the HMA contains some tensile strength, there are small displacements occurring in the vicinity of the moving wheel load.

Figure 4-11 - Deflection of a continuous beam.

In contrast, the unbound layers in the structure contain little to no tensile strength. The lack of tensile strength of these layers removes the above effect demonstrated in Figure 4-11 by effectively removing the ability of the material to carry moment/couple forces. However, the shear strength of the materials (both HMA and unbound layers) does allow the material to distribute load through a bridging action.

The earth pressure cells were installed into the subgrade (Figure 4-12) to measure the vertical stress due to the static weight of the pavement structure as well as the vertical loading from passing wheel loads. A total of four cells were installed – two in the subgrade layer (46 inches below pavement surface) and two in the dense graded base layer (20 inches below pavement surface). The pairs were installed with the same horizontal positioning – the two pairs were installed between the strain gauge arrays and were centered in the wheelpath (offset = 0.0 inch).

Figure 4-12 - Earth pressure cell installation.

Figure 4-13 below is a typical recording of vertical pressure in the base and subgrade layers caused by a class 9 vehicle (front axle and tandems on tractor only). Note that the change in the recorded sub-grade pressure is less than the vertical pressure change in the base layer. This is because of the wider distribution of the stress at the greater depth carried by the shear strength of the unbound layers. Also note that the individual axles of the tandem group can be identified in the base plate recording, while the subgrade demonstrates only one stress pulse.

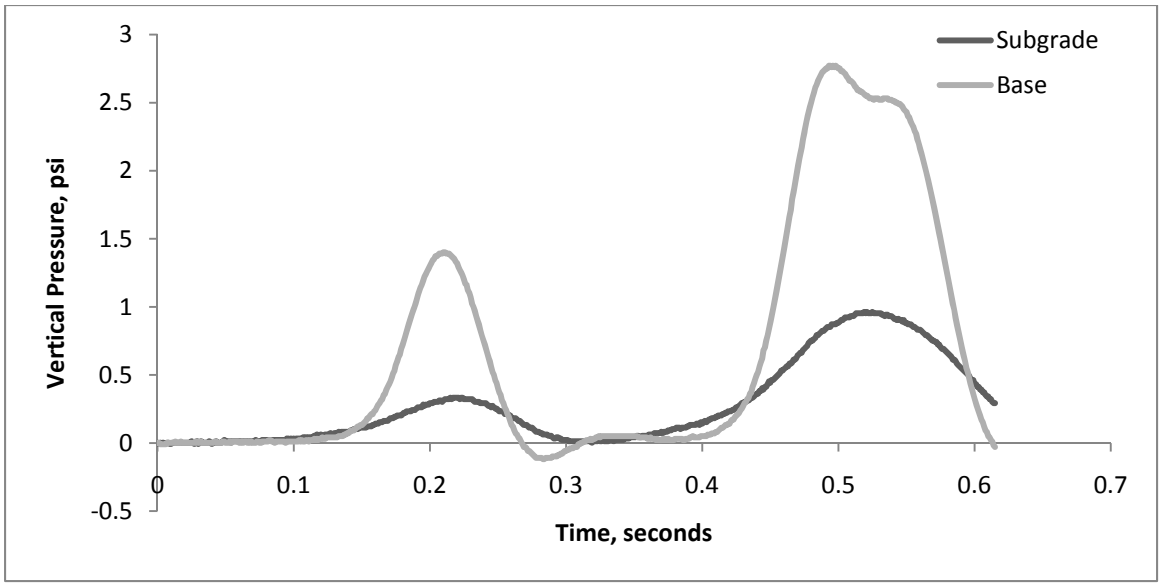


Figure 4-13 - Vertical pressure recordings.

Because the stress from the vehicle loading is distributed over a larger area deeper into the pavement structure, the duration of the stress at any point at a particular elevation is increased (Figure 4-14). In most circumstances the load duration in the unbound layers is not required for determining unbound layer stiffness because the materials are not considered load rate dependent.

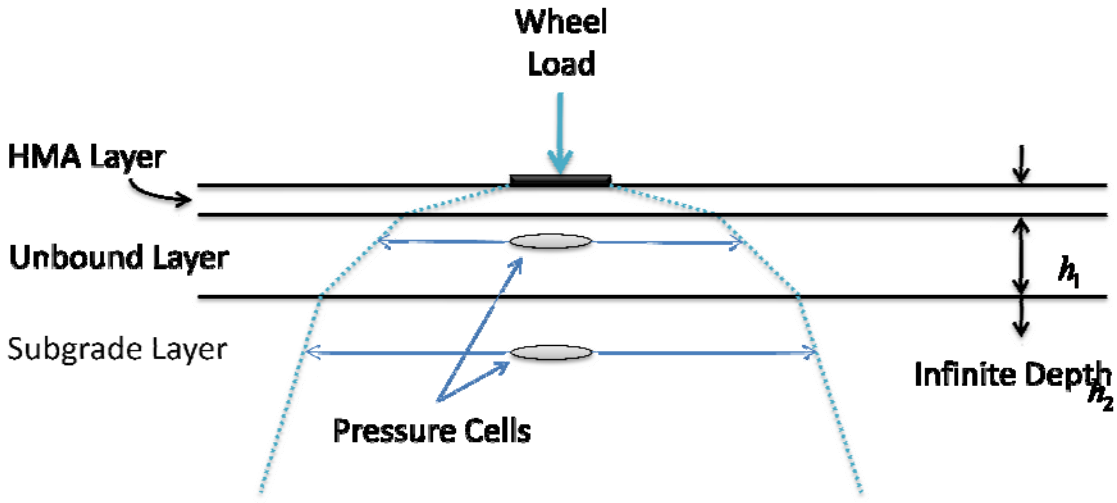


Figure 4-14 - Stress distribution in the pavement structure.

The adequacy of the MET process can be validated by comparing the adequacy for the non-load rate dependent layers. As stated before, the MET process for predicting load/stress durations are suited best for the case of vertical pressure because of the lack large stress reversals (i.e. tension to compression).

For validation of the MET concept, the load pulse durations from collected vehicle data was compared to the load durations computed using MET. Two time periods were used in general for this validation – on from a period of warmer weather (42^{nd} week of 2007) and another from colder temperatures (10^{th} and 11^{th} weeks from 2008).

A two week-long window of data was sampled for the colder temperatures because measured stress and strain magnitudes were significantly reduced due to the increase in HMA stiffness at low temperatures. This limited the number of significant observations found within one week of data (i.e. observations that contained strain measurements greater than 5 με and pressure measurements greater than 0.5 psi). Environmental conditions for week 10 and 11 from 2008 were nearly identical.

The MET load pulse durations for the time periods were not computed directly. The stiffnesses of the HMA layers were solved (iterative procedure) for in the FE analysis and were stored into a database along with the stiffnesses for the unbound layers. The MET load pulse duration calculations were then carried out in a spreadsheet using the computed stiffness data. The load pulse durations for the bottom of the HMA layer (the location of the strain gauges) was completed in the same manner.

Figure 4-15 and Figure 4-16 shows a comparison of the MET load times plotted against the measured load times for the base and subgrade vertical pressures at the warmer temperatures.

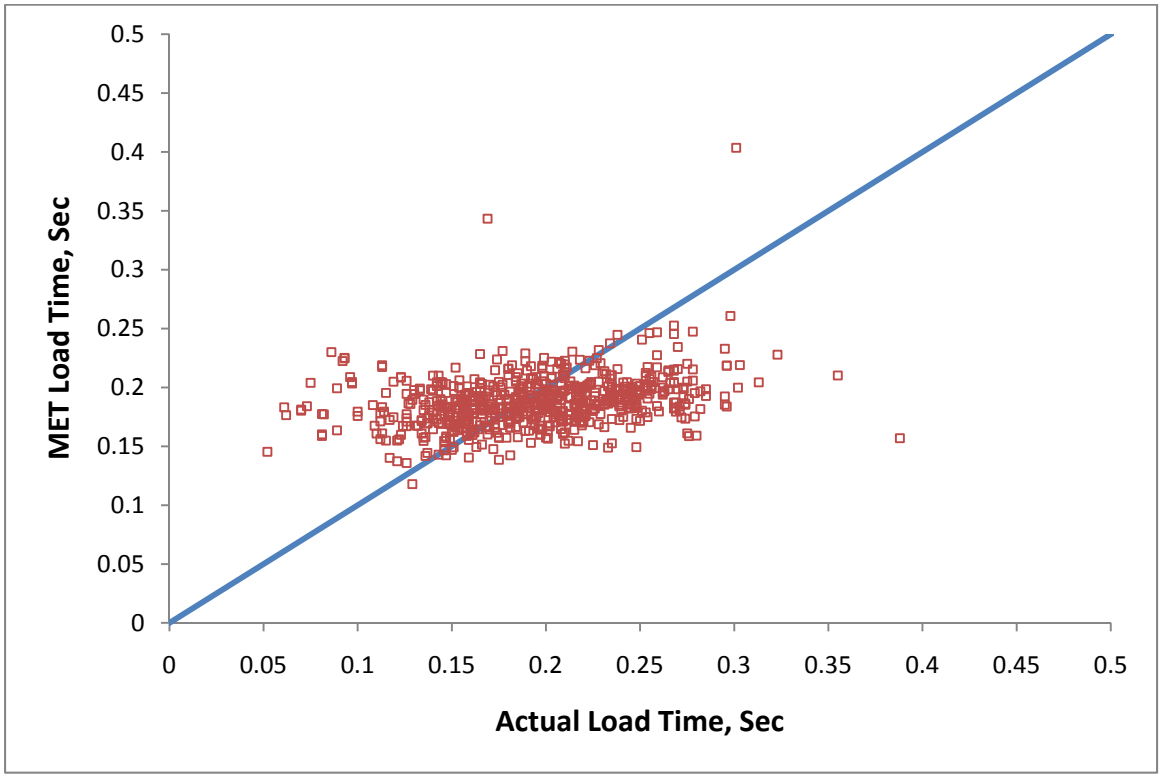


Figure 4-15 - Base actual versus MET load time.

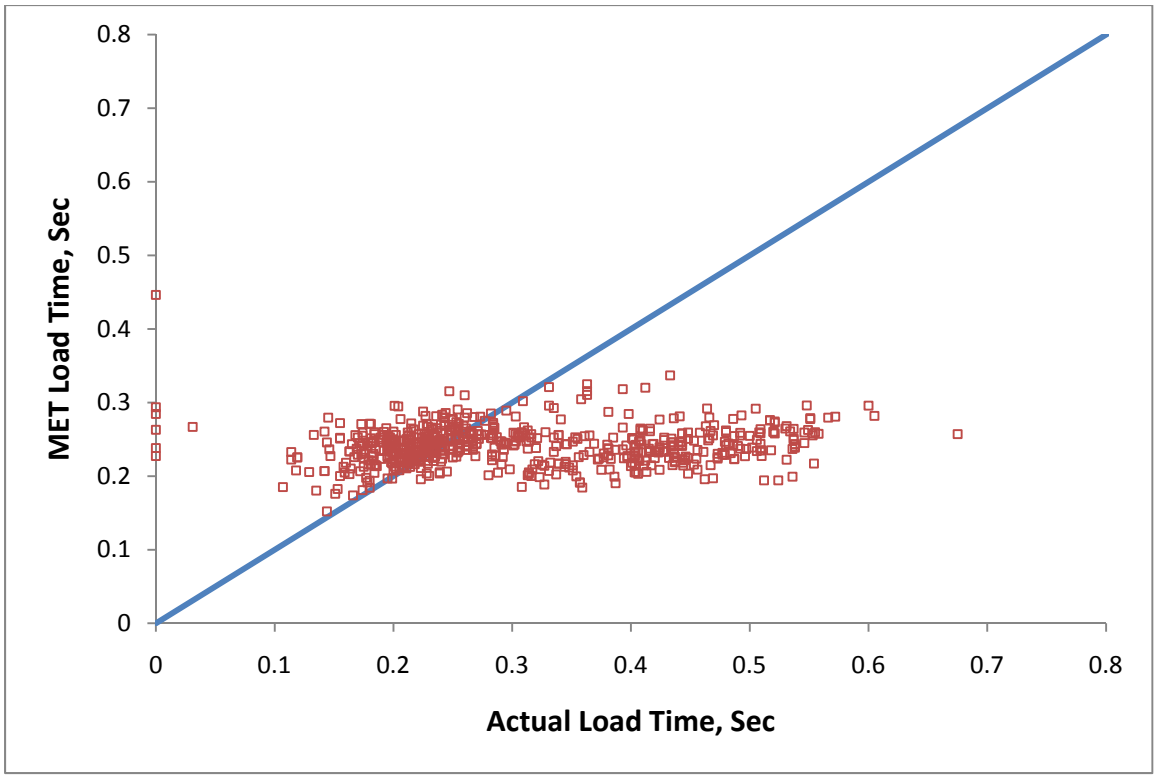


Figure 4-16 - Subgrade actual versus MET load time.

In Figure 4-15 and Figure 4-16, the bulk of the data is generally centralized around the line of equality; there is a trend showing that the MET load times do not contain much variability. The significant variation (scatter) in the data is due to the apparent low variation in the MET load durations, while the measured load durations have much more variation in the range of values.

The apparent insensitivity of the MET computed load times is mostly because the procedure assumes only a single wheel load. In reality there is a significant amount of overlapping of the stresses as discussed in Chapter 3 which causes some load durations to become longer.

Acknowledgment of this effect led to the separation of singles and axle groups (tandems, tridems, etc.) and further extending the load times based on the axle spacing.

Figure 4-17 shows the improvement of the fit of load pulse duration data – the portion of data points that were located to right side of the line of equality have been shift upward. An average axle spacing of 53 inches was used – this spacing was generated by taking the average of the measured axle spacing for tandem axle groups in the data set.

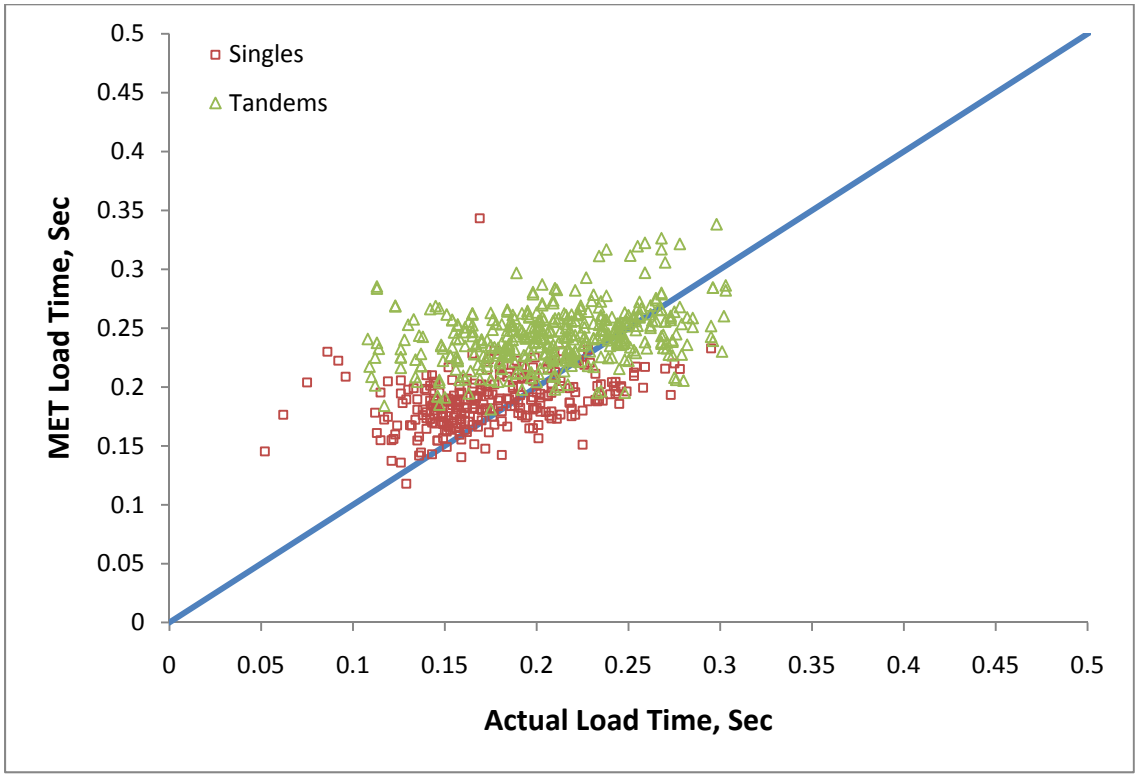


Figure 4-17 – Base singles/tandems load pulse durations.

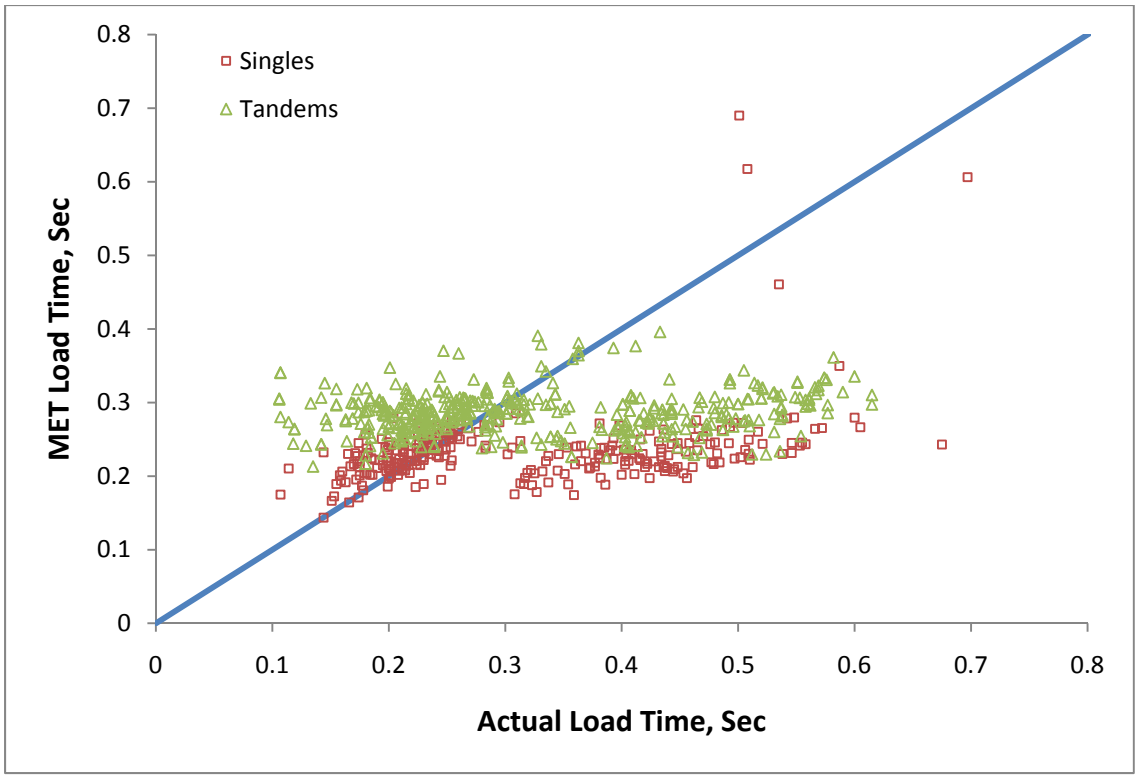


Figure 4-18 – Subgrade singles/tandems load pulse durations.

The preceding figures show that the MET process has crudely predicted the load pulse durations in the base and subgrade. In Figure 4-18 and Figure 4-19 there appears to be a secondary trend in both the single and tandem data points (Figure 4-19 only demonstrates singles for clarity). Around 0.300 seconds for the singles and 0.350 seconds for the tandems, there a slight gap in the actual load times. Upon inspection of the recorded data, it was found that this shift was due to pressure measurements that did not have full stress reversal (i.e. the measurement did not return to the unloaded state between different axles).

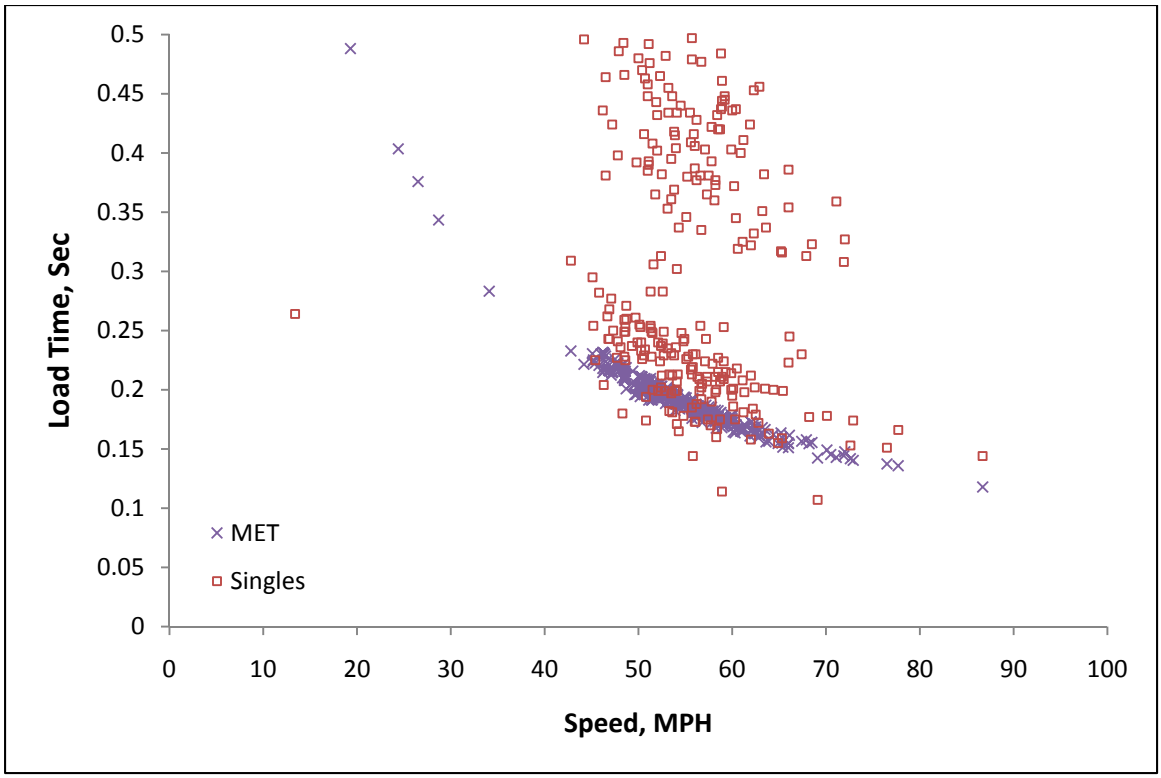


Figure 4-19 - Stress overlap for subgrade pressure.

Figure 4-20 shows pressure recordings from the data set demonstrating the two stress overlap conditions. In the no-overlap condition, the signal returns to the unloaded state, while the overlapped signal does not. For the overlapped condition, the pavement structure is in a constantly loaded state until the vehicle influence passes the area.

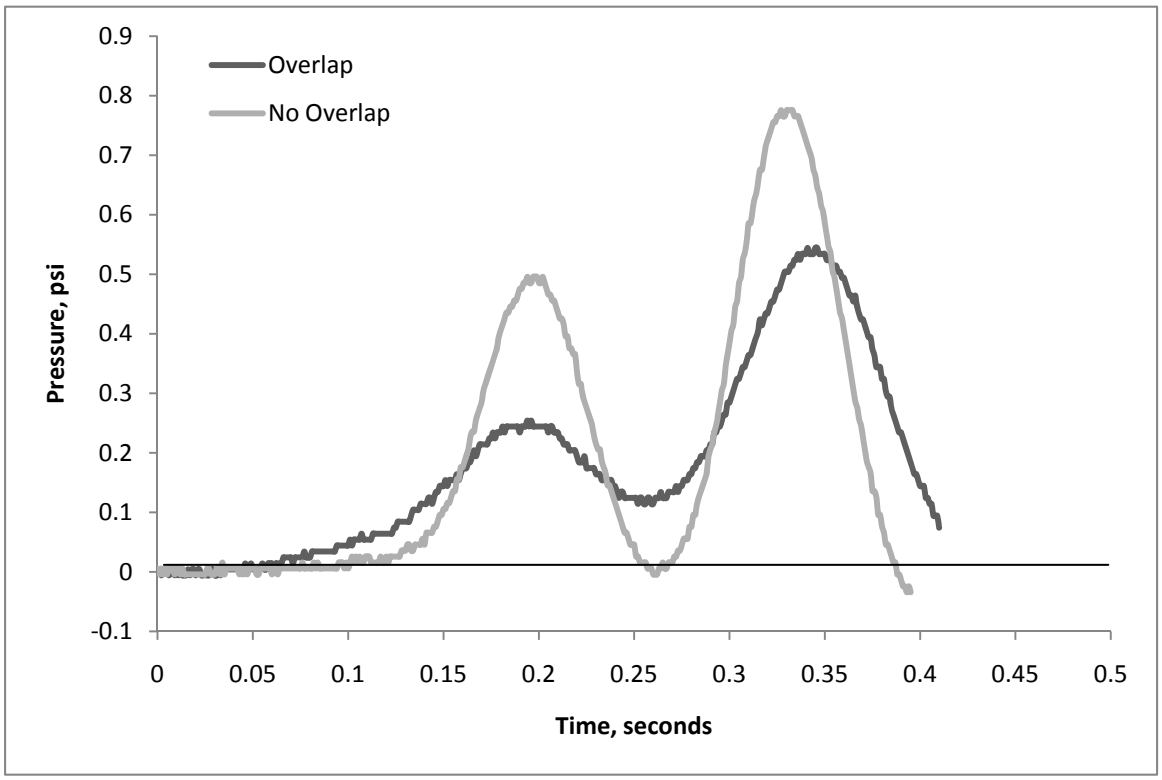


Figure 4-20 - Example of pressure measurements with overlap and no-overlap conditions.

4.2.2 - Horizontal Strain Load Pulse Duration

The MET procedure has been implemented to calculate the load pulse duration in the HMA layers to be compatible with the time-temperature dependent material model for HMA. This load pulse duration is then converted to a load frequency, Equation 4-6, and used in the dynamic modulus model to calculate the HMA stiffness (National Cooperative Highway Research Program, 2004). The MET procedure has been shown in the above analysis for the vertical compression that the load pulse duration can be crudely estimated.

$$f = \frac{1}{t} \quad \textbf{(4-6)}$$

where f = frequency, Hz

t = load pulse duration, seconds

The observed response of pavement under moving wheel loads may imply that the MET procedure for calculating load pulse durations may be inaccurate and over estimating the load pulse durations. Typical strain measurements from other research have shown a considerable stress reversals in the bound layers (Priest, 2005) (Timm & Priest, 2004) (Timm et al., 2004). This effect may increase the rate of straining in the HMA higher than a load scenario with no stress reversals, which has the effect of causing an increased stiffening the HMA layers through the viscoelastic component of the HMA.

Figure 4-21 below is a typical strain response of the HMA pavement from a moving wheel load for transverse and longitudinal horizontal strain. The recordings are from the same class 9 vehicle that caused the pressure recordings used in Figure 4-13 (year-week: 0742; wheel id's: 250183482, 250183483, & 250183484). In the figure, note that the longitudinal strain contains a compression spike before and after each major tensile strain. The transverse strain recording does not exhibit this behavior.

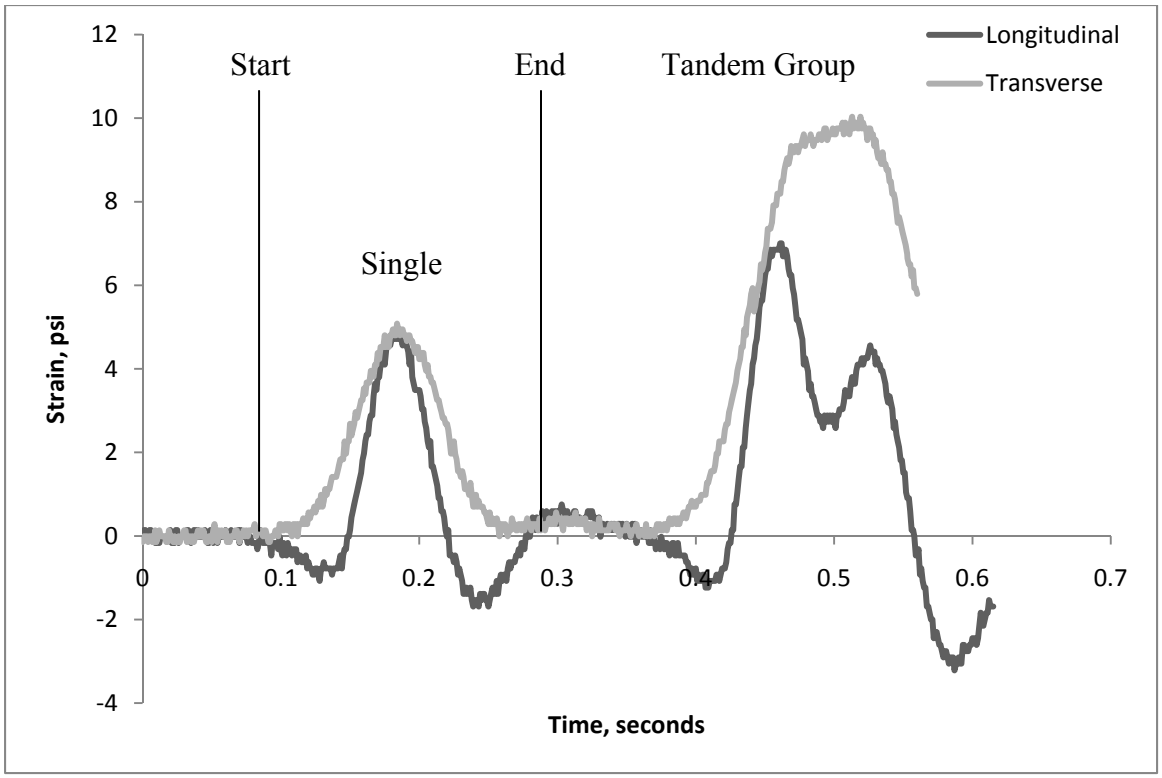


Figure 4-21 - Typical strain response.

Measured from the start to the end of each response, the single axle load times appear to be quite similar as indicated in Figure 4-21. However, the existence of compression in the HMA layer for the longitudinal strain causes the slope of the strain/time data to be much steeper than that of the transverse. This increased slope implies that the rate of strain in the longitudinal direction is faster than in the transverse direction.

Often, HMA pavements that exhibit bottom-up fatigue symptoms typically are evidenced by longitudinal cracking in the wheelpaths. The above observations from Figure 4-21 and the many nearly identical recordings from this project, support and give reason to why longitudinal cracking is a dominant pattern of bottom-up fatigue.

The increased strain rate in the longitudinal direction implies that the stiffness of the HMA in the particular direction is greater than that in the transverse direction due to the viscoelastic nature of HMA (i.e. the material is stiffer at a faster rate of loading). For a particular load, the increased stiffness results in lower strain for the longitudinal direction. This is not witnessed for the front axle of the vehicle in Figure 4-21, however is readily apparent in the strain magnitudes of the tandem axle set that follows. Figure 4-22 shows a typical strain response for single axles – exhibiting transverse strain being larger than the corresponding longitudinal strain.

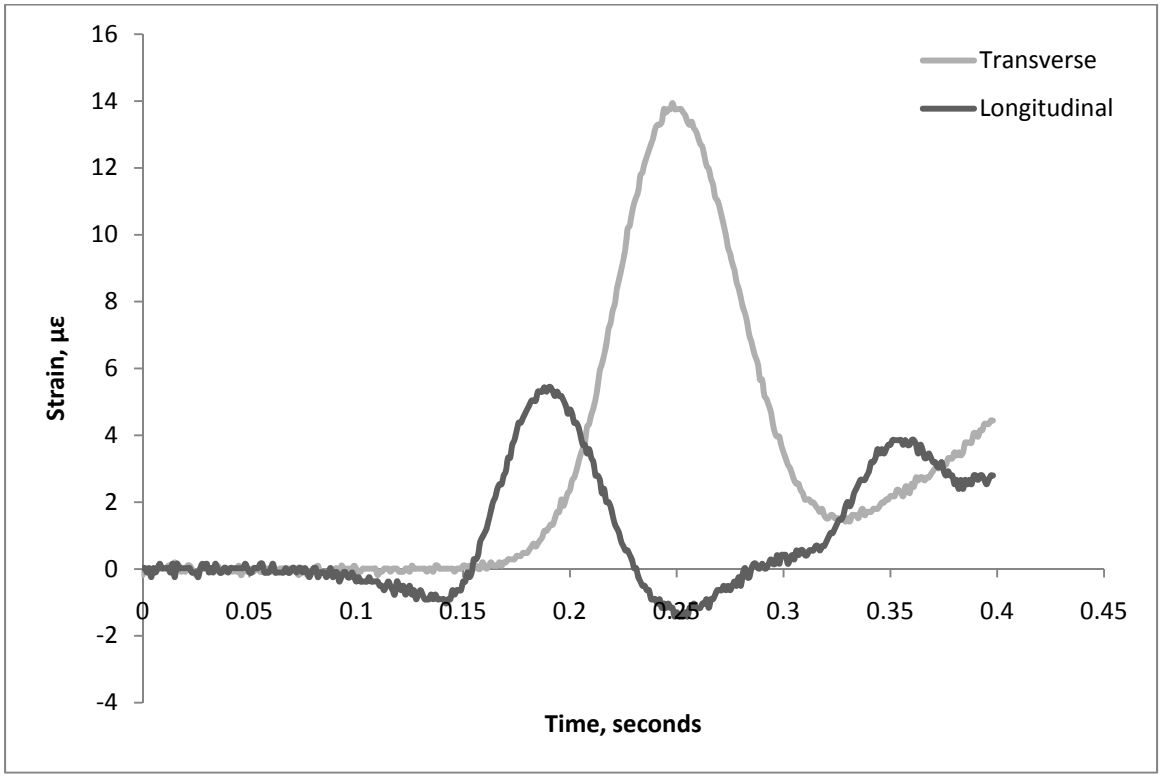


Figure 4-22 - Single axle strain response.

The same comparison of load times completed for the vertical pressure was completed for the horizontal strain to investigate the adequacy of the MET load durations

versus those measured in the field. Figure 4-23 exhibits the results from the date generated for longitudinal horizontal strain.

From inspection of Figure 4-23, it is clear that the MET computed load durations are being over estimated and actual load durations are shorter. Applying the tandem axle spacing moves the cloud of data upwards away from the line of equality, demonstrating a constant error (best fit line through clouds passes through origin). Figure 4-24 is the same data from Figure 4-23, but includes the shift for the tandem axle spacing.

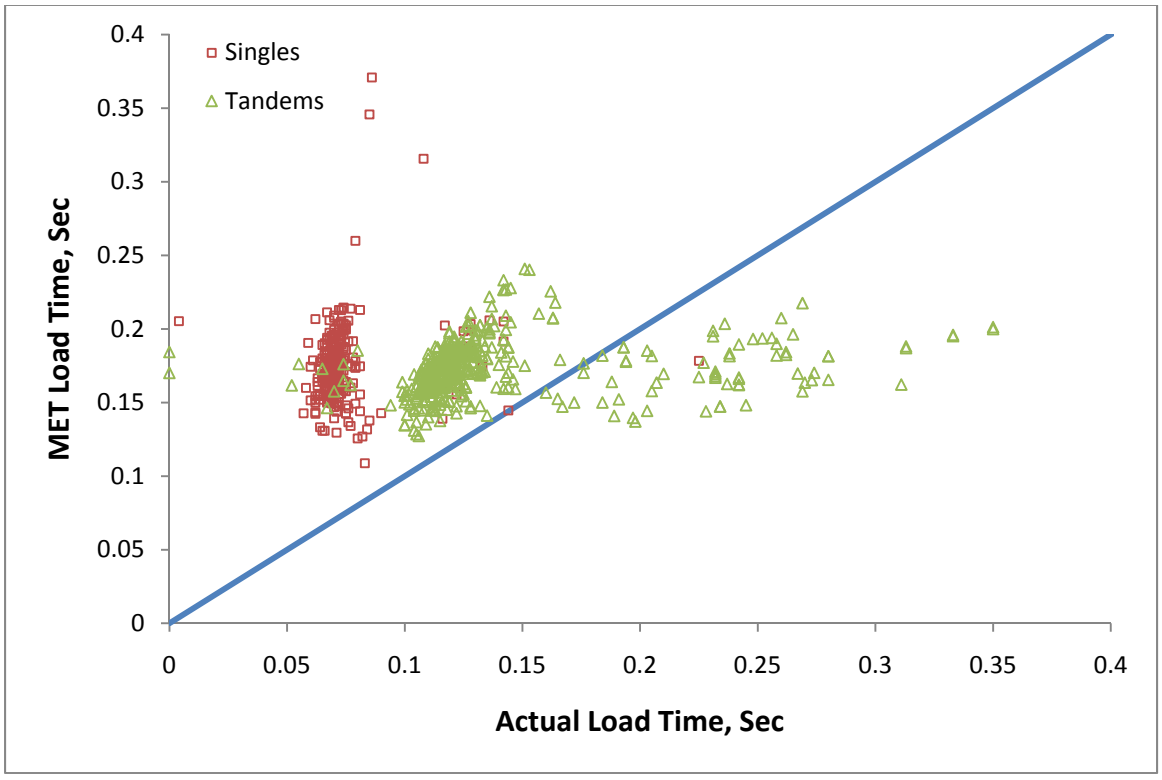


Figure 4-23 - Actual vs. MET predicted load duration from longitudinal horizontal strain – no tandem correction applied.

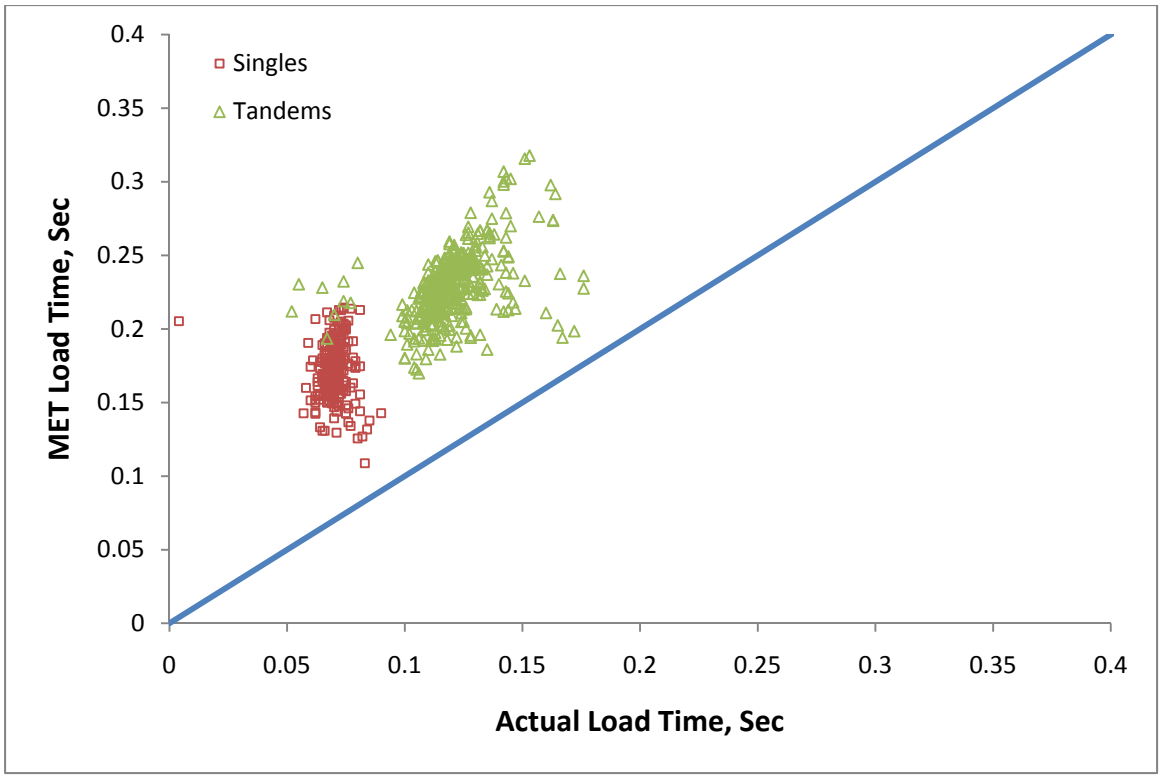


Figure 4-24 - Actual vs. MET predicted load duration from longitudinal horizontal strain including shift for axle groups.

Figure 4-25 below shows the comparison between the MET and measured load durations for the transverse strain measurements (which includes the shift for tandem groups). The data for the transverse load durations contains significantly more scatter than the longitudinal strain data. This is due to the inherent difficulty in defining the start and end of the strain pulse since the signal does not (rapidly) return to an unloaded state , zero, value.

Transverse strain measurements have the tendency to remain in a tensile state and recover to an unloaded state quite slowly. Because of this, only half of the strain pulse is measured and the time is doubled – this assumes the shape of the pulse is symmetric. This method increases the data scatter for the case of tandems (or other load groups) since the middle of the pulse is based on the relative locations of the strain peaks and the

peak detection itself contains some variability. The details of the strain pulse measurement can be found in the Marquette Instrumentation Project Phase II Report (Hornyak & Crovetti, 2008).

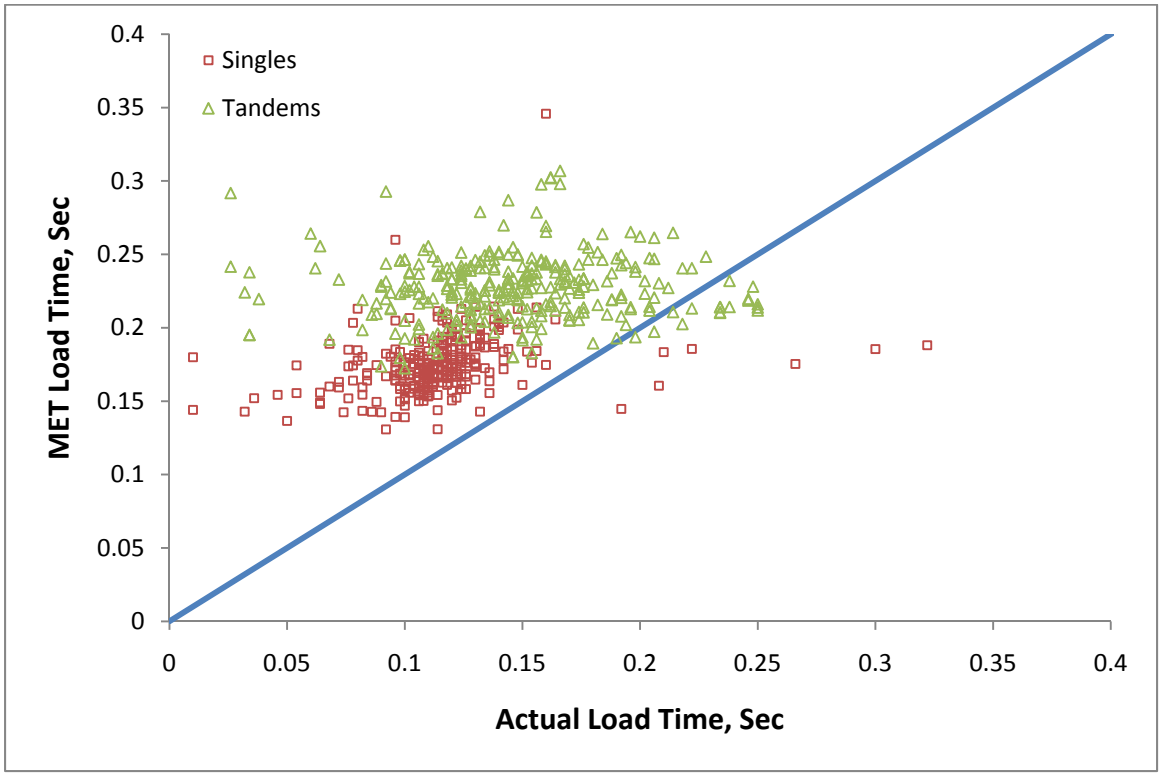


Figure 4-25 - Actual vs. MET predicted load duration from transverse horizontal strain including shift for axle groups.

In addition to the measurement errors generated, the transverse strain is much more sensitive to load location than the longitudinal case. This is explored in more depth in following discussions.

Even with the increased scatter in the data, is it apparent that the load times computed by MET are being rendered too high, but by a smaller margin than those for the longitudinal case. Even though the strain pulses are similar is shape to the vertical

pressure which was found to be somewhat adequate, the load durations are still apparently longer.

The effect of temperature was analyzed to see if there was significant change in the prediction of load pulse duration. The two analysis periods contained drastically different temperature regimes – Table 4-5 below contains the average ambient air temperature for the analysis periods examined.

Table 4-5 - Temperatures for analysis periods.

Month	Air Temperature, F	
	September	March
High	78.2	47.2
Low	41	18.7
Average	59.4	31.1

Figure 4-26 is the same comparison plots as Figure 4-24, but with the two time periods superimposed. It is evident that there is little change in the load duration, both for the predicted and also the measured.

It is also interesting to note that the figure indicates that the MET process is creating an increase in the load times during the colder temperatures. This is likely due to the increase in stiffness of the HMA layers at the colder temperatures which causes an increase in the equivalent thickness, ultimately resulting in longer load pulse durations. In reality this is reasonable since the HMA will carry the loads over a large influential area, however the experimental does not show this affect. The average actual loads times for the colder and warmer temperatures was found to be 0.136 seconds and 0.137 seconds respectively. The average predicted load times for the colder and warmer temperatures was found to be 0.245 seconds and 0.229 seconds respectively.

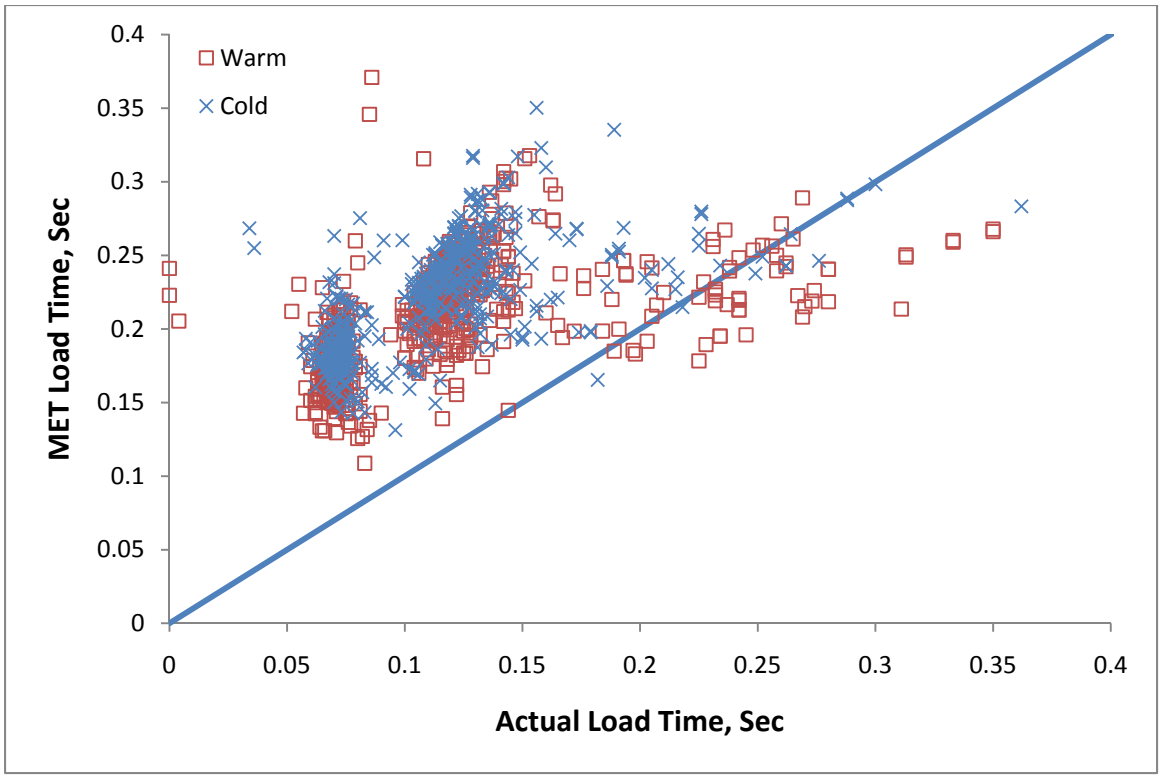


Figure 4-26 - Actual vs. MET predicted load duration at different temperatures.

The errors between the MET predicted and the actual load durations were minimized and an adjustment has been supposed to account for these errors in measurement. The initial analysis used the assumed 45° stress distribution as discussed in Chapter 3. This angle is a broad representation of the ability of the bound and unbound layers to carry induced loads through the shear strength of the materials. For instance, in soil mechanics it is well understood the different modes of failures of common soil types.

For instance, clays can have a tendency for a punching type of failure, especially when the clay is unconsolidated and undrained (U-U state). Consolidated clays, clays with drainage and many granular materials typically exhibit a shear strength and failures can exhibit significant local displacements due the shearing action of the soil. Triaxial

strength testing of these materials confirm this well understood concept (Murthy, 2003). The strength characteristics of common soil types are exhibited the Mohr's plots below in Figure 4-27.

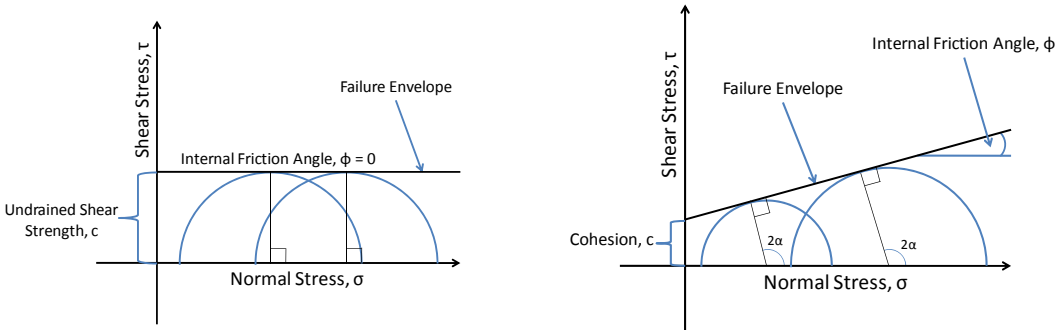


Figure 4-27 – Mohr's circle plots for unconsolidated-undrained state (left) and drained state (right).

The stress distribution angle assumed in MET has a direct impact on the computation of the load duration. Increasing the steepness of this angle from 45° to say 50° or 60° (from horizontal) would effectively reduced the computed load pulse duration.

In order to quantify a corrected angle for the stress distribution, the solver function in Microsoft Excel was used to minimize the error between each individual data pair of measured and MET load pulse durations for the horizontal strains as well as the vertical pressures.

After minimizing the errors, the resulting angle was then used to reevaluate the stiffness computations and the resulting stress/strain responses of the pavement using the FE analysis program. Since the routine to calculate the HMA stiffness based on the MET procedures was integrated to the FE package, only the selected time periods were used for the analysis since each data set took over eight hours to complete.

The data from the 42nd week of 2007 and the 10th and 11th weeks of 2008 were used for the optimization. The optimized stress angles were found by minimizing the square of the errors between the MET and actual load pulse durations. The process was completed for each individual case (i.e. vertical stress, transverse horizontal strain, etc.) and the stress angle for each noted. Table 4-6 below contains the optimized stress angles found for each.

Table 4-6 - Optimize MET stress angles.

Orientation	Stress Angle, °	
	Warm Temperature	Cold Temperature
Longitudinal Strain	68.55	68.29
Transverse Strain	56.45	65.12
Base Vertical Pressure	45.00	45.00
Subgrade Vertical Pressure	45.34	44.48

It is worthy to note the optimized stress angles for the different time periods (and environmental conditions) are strikingly similar with the exception of transverse strain. The similarities between the temperature regimes support the theory that temperature has little influence on load times.

Also, the above load duration analysis showed that the MET and actual load durations for the vertical pressure cases were quite similar. The optimized stress angle is quite close to the assumed 45° angle which further supports that the MET process can be adequate the case of vertical pressure.

The optimized angles for the horizontal strain have a much larger change from the assumed 45° angle. Both angles for horizontal load times were increased which is consistent with the data comparison from above. The difference in stress angles for the transverse strain case is due to the amount of scatter in the data from the warmer

temperature period. During this period of warmer temperatures, the measured strain in the HMA does not return to the unloaded state between axle passes and can typically have a residual strain up to 5 με. During analysis at the warmer temperatures, the program tends to 'over-measure' some axles which are relatively closely spaced and produce long load times because the strain measurement never returns to an unloaded state.

For the strain load time analysis, most of these cases have been removed for analysis because the magnitude of the residual strain is rater low. In Figure 4-23 (pg. 142), the data points that extend horizontally to the right (high measured load times) exhibits these particular 'over-measurements'. However, this assumption of residual strain (also similarly visible in the earth pressure data discussed prior in Section 4.2.1 regarding stress overlap) may warrant further investigation in future research to quantify if this effect has any influence on the fatigue characteristics of the pavement.

It should be noted that the load time calculations should be iterated since the stiffness changes with the change in load duration. However the error should be reasonably small - the updated stress angle was used in the analysis program and the stiffnesses were iterated for the FE analysis.

Below are updated comparison plots the MET and actual load durations for both the horizontal strains and vertical pressures for the 42nd week of 2007 dataset. The data for the figures used the updated stress angles found for each respective case except for the vertical pressure where the stress angle for longitudinal strain was used (this was done since the optimized was relatively close to the assumed 45° and the longitudinal angle held the greatest value of 60° and would thus provide a lower limit of load duration).

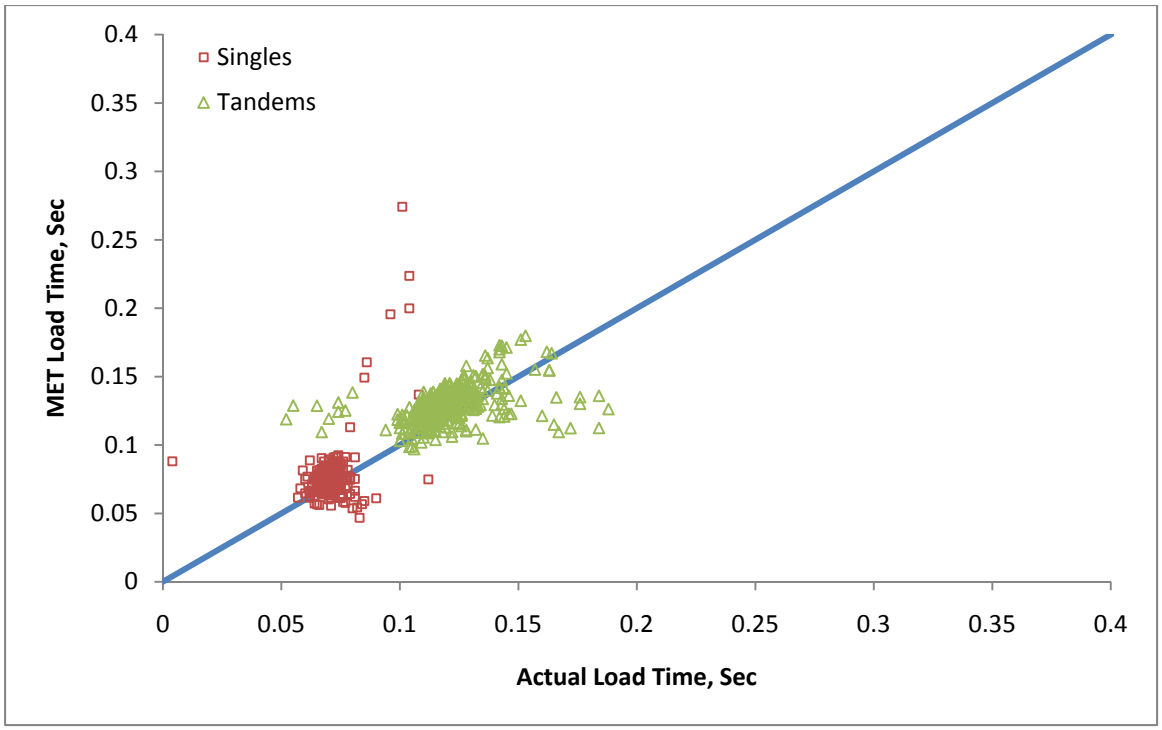


Figure 4-28 – Actual versus predicted load pulse durations for longitudinal strain using the increased stress angle.

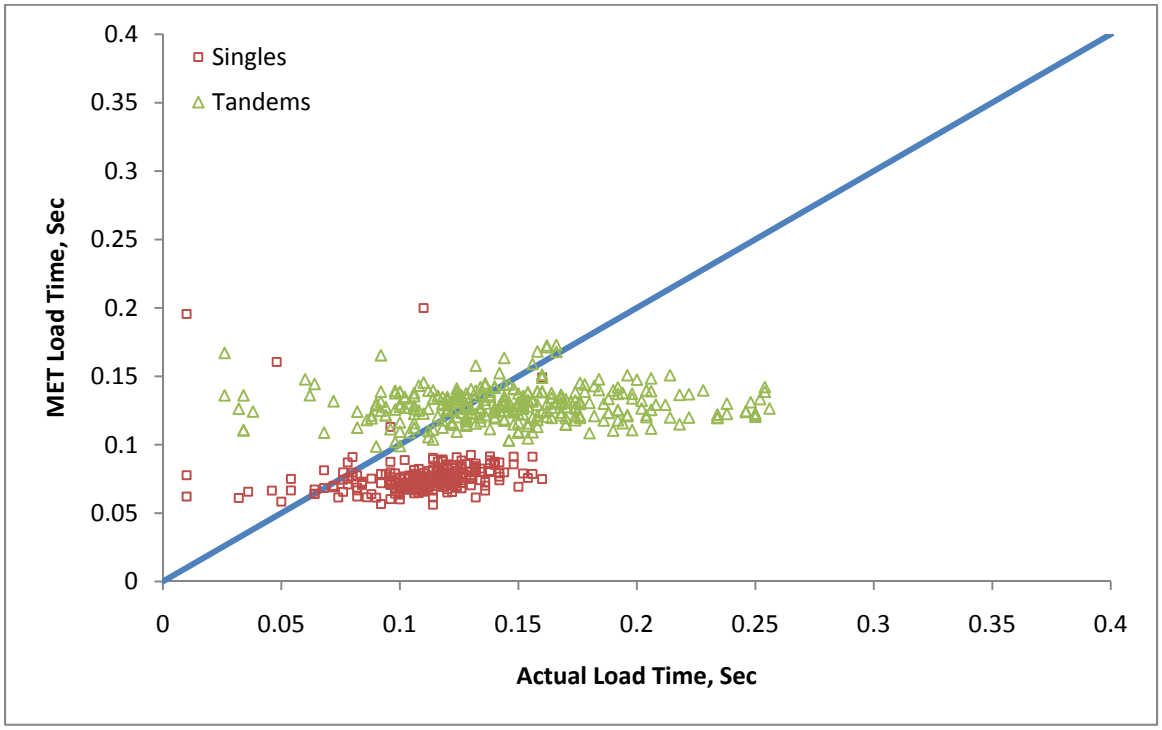


Figure 4-29 - Actual versus predicted load pulse durations for transverse strain using the increased stress angle.

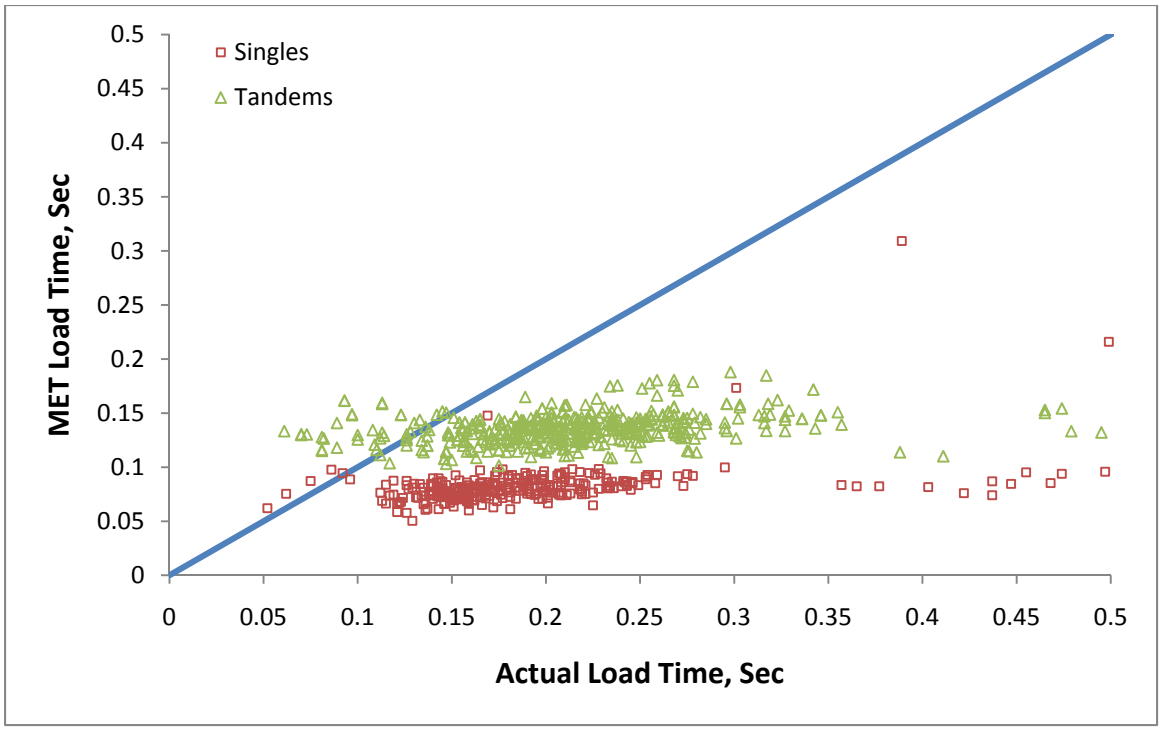


Figure 4-30 - Actual versus predicted load pulse durations for base pressure using the increased stress angle.

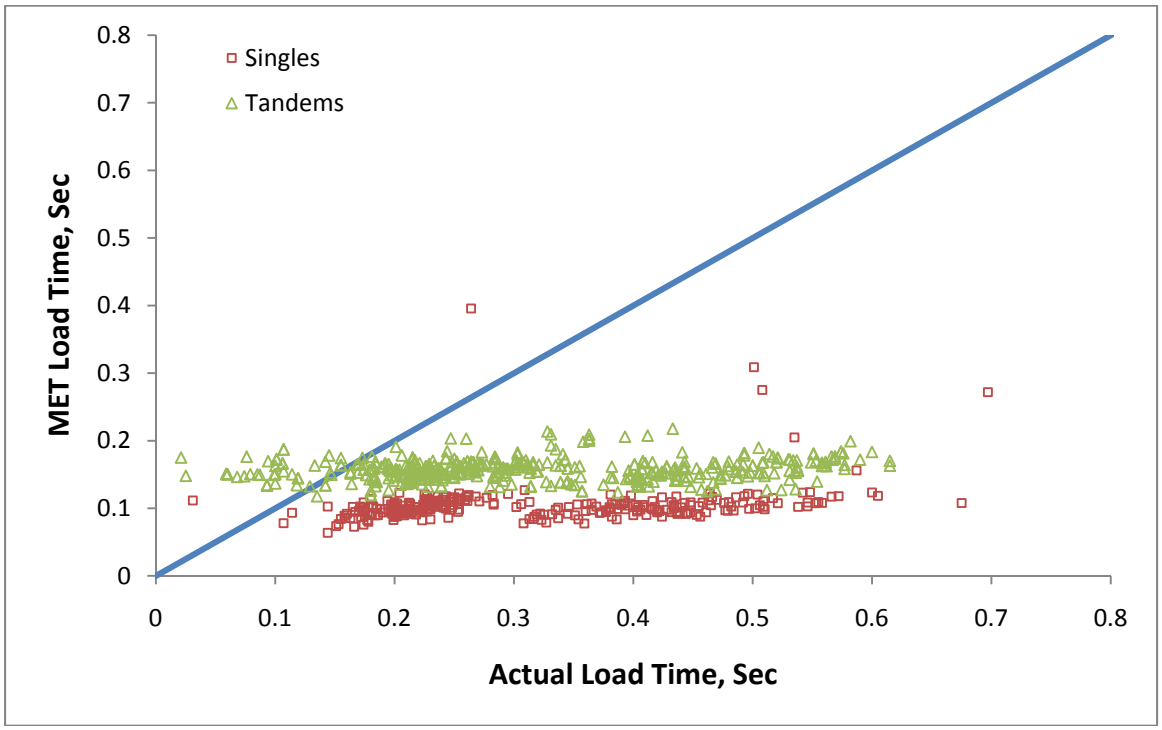


Figure 4-31 - Actual versus predicted load pulse durations for subgrade pressure using the increased stress angle.

It is apparent the optimized stress angle has provided a better prediction of the load pulse durations for the horizontal strain and the vertical pressure are crudely represented. The procedure was completed for data taken from the 10th and 11th weeks of 2008. The load duration data was nearly identical to that of the data taken from the warmer period used to generate Figure 4-28 through Figure 4-31.

In the past, other researchers studying the load pulse durations have described the load pulse as a function of vehicle speed (Barksdale, 1971). Figure 4-32 and Figure 4-33 below are scatter plots of the load pulse durations and speed for both the longitudinal and transverse orientations and both predicted and measured load times. The longitudinal load pulse durations have a much stronger relationship with speed while the transverse contains more variability (due to the detection errors mentioned above). The simple linear regression models applied in Figure 4-32 and Figure 4-33 represent the best fit models for the predicted data.

The prediction model presented Figure 4-32 fit the experimental data quite well for the longitudinal strain case. On the contrary, the transverse model (Figure 4-33) for singles is slightly under-predicting the load times (short in duration), while the model for the tandems fits quite well. Note that the experimental (measured) transverse load times for the singles and tandems (Figure 4-33) lie nearly on top of each other. Figure 4-29 indicated that for the chosen stress angle for analysis, the load pulse durations are shorter in duration than those found experimentally. This was expected since the stress angle found by minimizing the error between measurements was found to be less than 60°. The selected stress angle was used as an average value for both orientations and small discrepancies were expected.

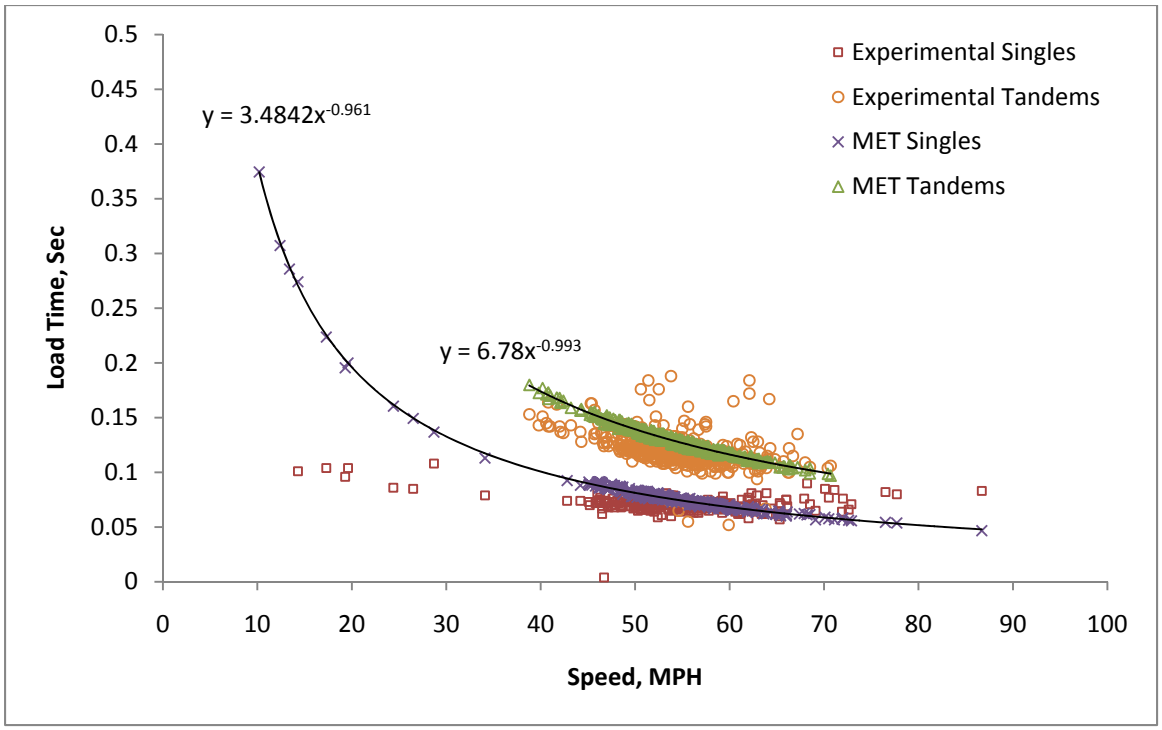


Figure 4-32 - Speed versus longitudinal strain load pulse duration.

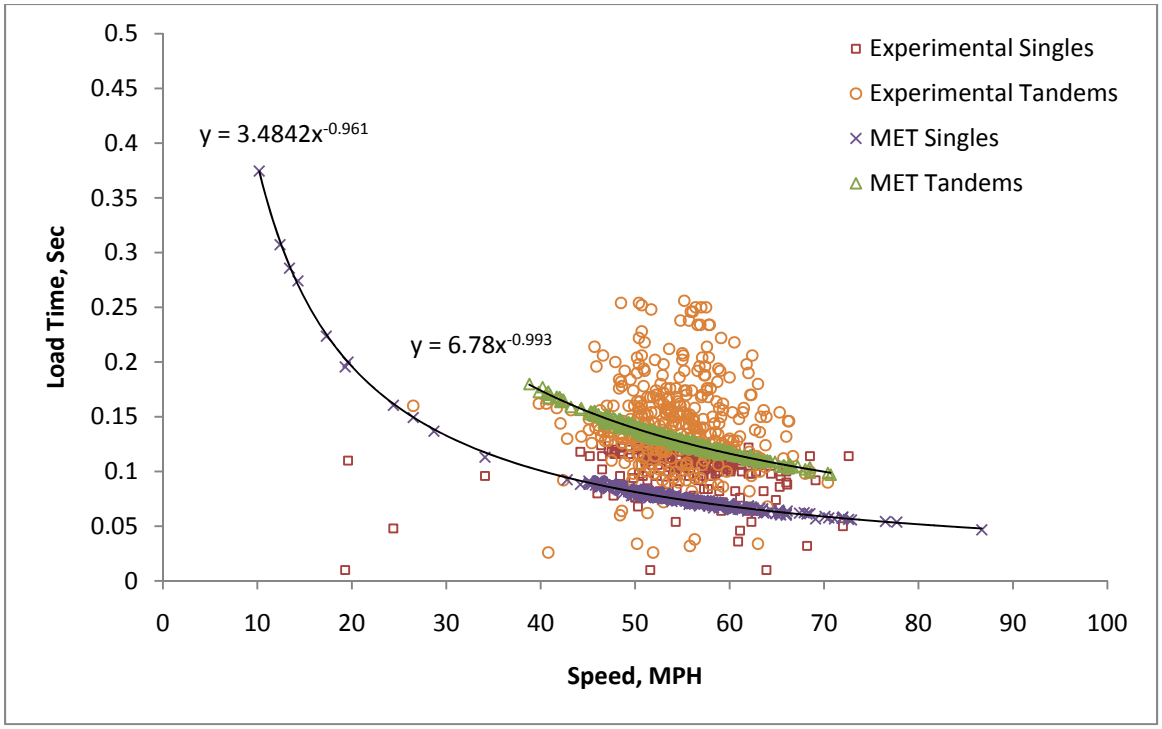


Figure 4-33 - Speed versus transverse strain load pulse duration.

4.3 - Stress/Strain Prediction

The previous discussion on the load pulse duration was focused on testing adequacy of the MET process for determining the load pulse durations which are then used for determining the stiffness of the HMA pavement. The discussion concluded that an increased stress angle used in the MET process can provide a much more accurate prediction of the load pulse duration in the HMA layers.

The following provides an analysis of the FE results compared to the stress/strain measurements made at the test section. Of particular interest to this project were the horizontal strains and the vertical pressures. The preliminary data analyzed used the standard MET procedures with the 45° stress angle. The FE computations were then completed using the updated stress angles found in the section 4.2.2 and the improvement in stress/strain prediction is analyzed. The analytical representation is important for future predictions of the strain, which can then be used to estimate fatigue consumption of the pavement.

Discrepancies found between experimental and the analytical results may be attributed to several different parameters - the biggest of which is the stiffness characteristics of the HMA. Variations in stiffness of the subgrade should be minimal; however the inaccurate prediction of the load pulse duration may lead to larger error in stiffness predictions. Based upon the measured load times for the vertical pressure and corrected load times in the HMA, a better prediction of the strain can be achieved. Other sources of error can be attributed to the FE analysis program which contains a few mild assumptions in regards to load modeling, boundary constraints, etc.

Higher quality strain prediction results in a better quality fatigue prediction. However, this method does not account for irregularities in the pavement structure such as cracks or layer debonding (either partially or fully debonded). These conditions may prevail in the future as the pavement ages. Due to the young age of this pavement, it was a safe assumption that these conditions either do not exist or exist to a very small degree.

4.3.1 - Vertical Pressure

The FE model of the pavement was designed to mimic the actual pavement as closely as possible to allow for accurate predictions of the structural response of the pavement. Since the data from the field was collected and processed, it was possible to make a direct comparison between the actual measurements and those predicted by FE analysis.

Figure 4-34 below is a comparison of the actual versus field measured pressures in the base layer. Only loads that were located laterally within ± 2.0 inches of the instruments centerline were considered. It is apparent from this plot that the predicted pressures at this location are 65% to 70% lower than the actual pressures from the field. A linear regression model has been added to the plot with the y-intercept forced to through the origin. The slope of this regression model essentially provides the average ratio of the FE predicted pressure to the actual.

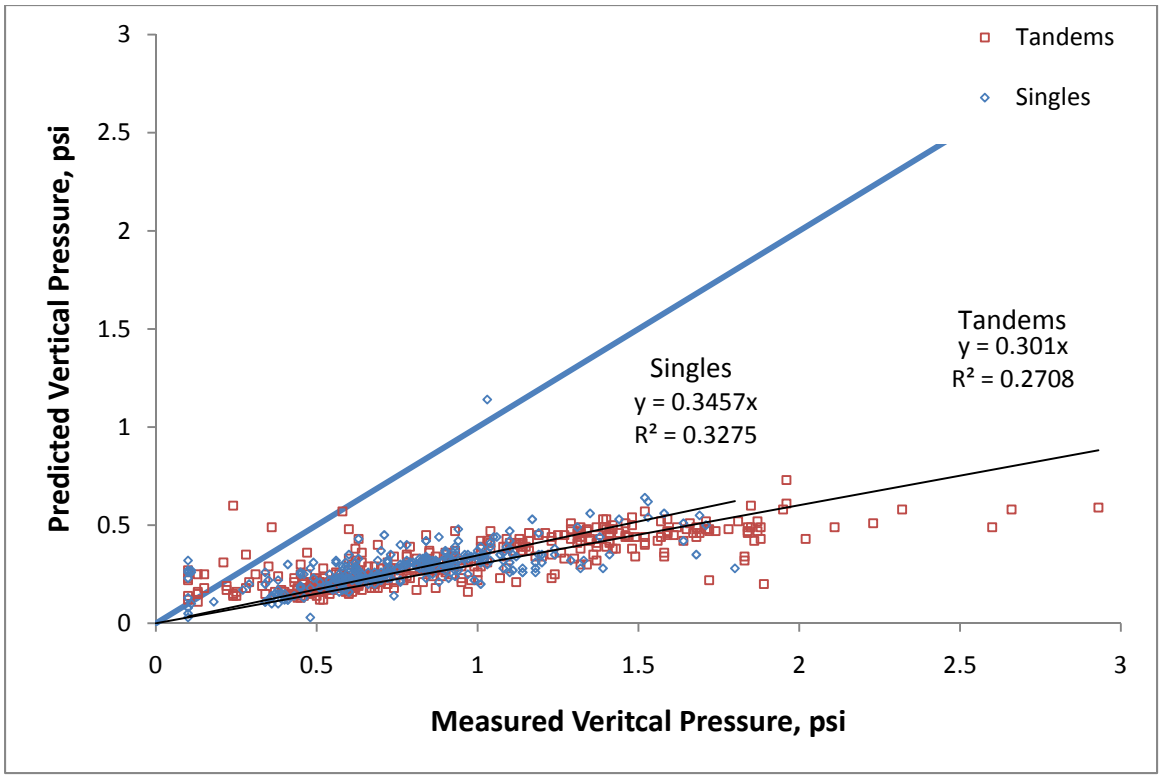


Figure 4-34 - Actual vertical pressure versus predicted for base layer.

Since the MET predicted load times for the horizontal strain were found to be longer than actual, it is expected that the predicted pressures would be higher than actual since the HMA stiffness would be reduced. The data here suggests the predicted and actual stresses are on the contrary with the ratio of the predicted and actual pressure being less than unity.

It should also be noted that all of the pressure cells were installed in each respective layer within a bed of fine, uniformly graded sand which protected the cells from puncture or other damage from the sharp, coarse fractured particles that compose the base layer. Because of the sand bedding, there is the possibility that the higher strength base layer can bridge around the sand/pressure cell discontinuity and actually cause lower pressures to be measured.

In the subgrade/native soil layer where the soil might be weaker than the sand bedding, the opposite effect may occur. In this scenario, the pressure cell and bedding is stiffer than the surrounding material and the combined effect can act as a simple spread footing in which the material directly below the instrument 'feels' less stress due to the distribution (Geokon, Inc., 2004).

In both cases, the instrument installation can have quite an effect on the state of stress around the cell leading to measurement errors. Due to this discontinuity of the stress field, it is possible that the base pressure cell may be producing readings that are slightly under the real stresses. The cells have been reported to produce measurement errors up to 15% of the mean soil stress (Geokon, Inc., 2004). The cells were installed using practices to minimize measurement error.

Figure 4-35 shows the pressures in the native soils layers. In this figure, it is apparent that the predicted pressures are closer to the measured pressures, especially for singles, but still about 40% lower than measured (although there are some significant leverage points within the data). Again the decreased predicted stress could also be due to the relatively weaker subgrade causing the pressure measurements to be higher than the real stress.

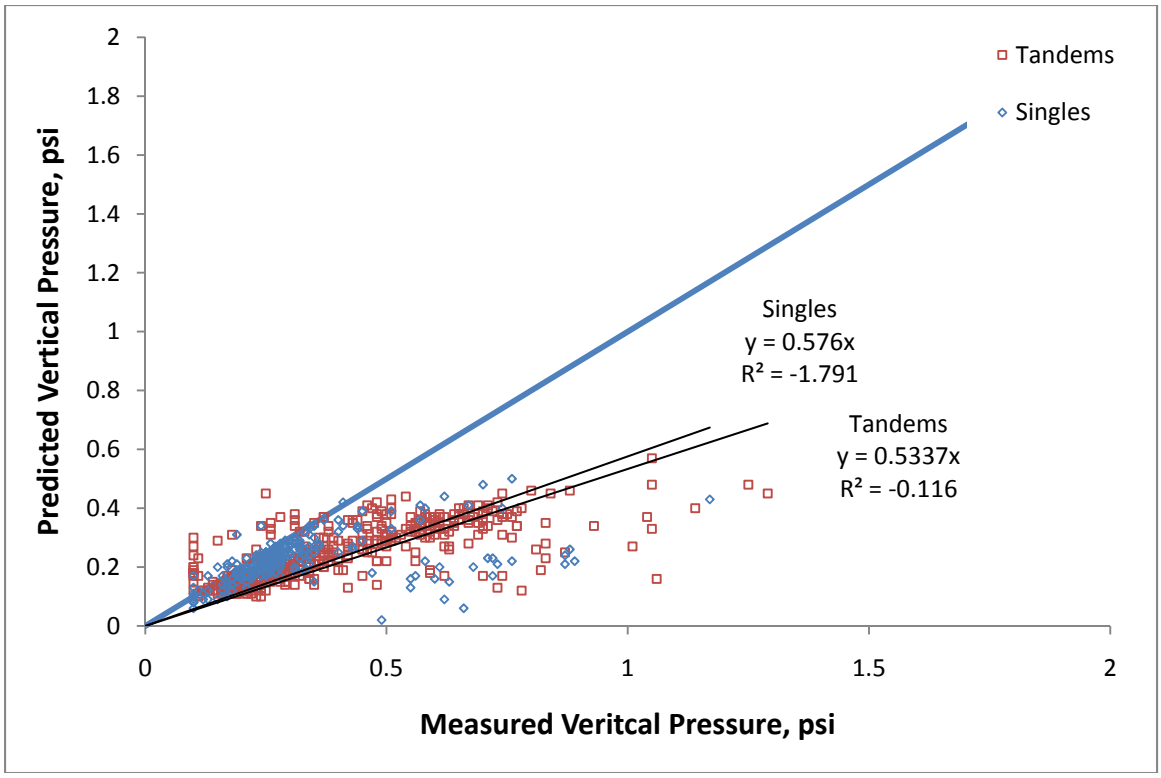


Figure 4-35 - Actual vertical pressure versus predicted for native soil layer.

A likely source of the prediction error for the pressure cell data relates to the FE model developed for the pavement. The mesh has been refined near the vicinity of the load and the mesh becomes progressively coarse moving away from the load – both horizontally and vertically. This may have caused the model to be too stiff at the location of the pressure cells, thus resulting in a lower stress calculation.

4.3.2 - Horizontal Strain

Figure 4-36 and Figure 4-37 below are comparisons of the predicted and actual horizontal strain for both the longitudinal and transverse strain. Similar to the load duration analysis, the data only represents moving loads that were located over the instruments within a range of ±2 inches from the centerline.

It is apparent that in the longitudinal cases, the predicted strains are higher than the measured strains. This is expected since the standard MET procedures were used to generate the material data for the predictions and the load durations are much longer than the actual, applying a lower HMA stiffness which results in higher predicted strains in the HMA. This finding is consistent with the load duration data.

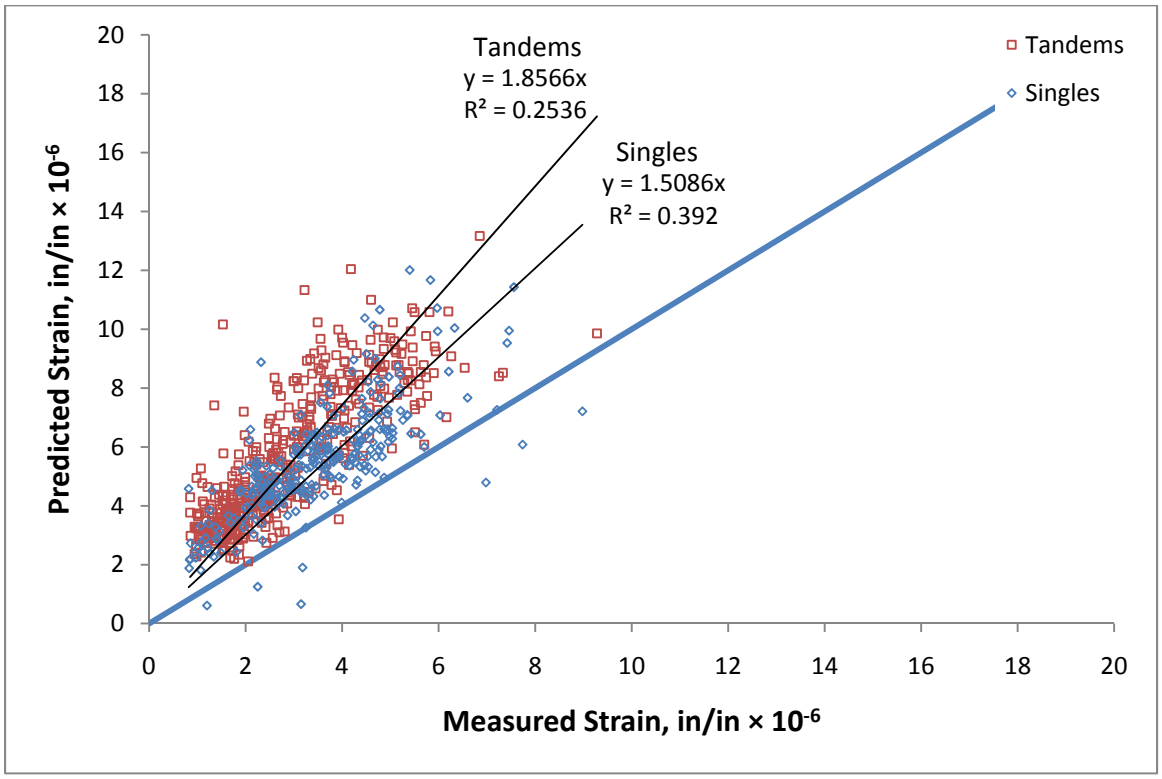


Figure 4-36 - Measured versus predicted longitudinal horizontal strain.

Contrary to the longitudinal case, the predicted transverse strains are nearly in agreement with the measured data. The predictions were actually slightly lower than the measured values – observations of the transverse strain during the experiment typically indicated that the transverse strains were slightly larger than longitudinal strains for the same loads.

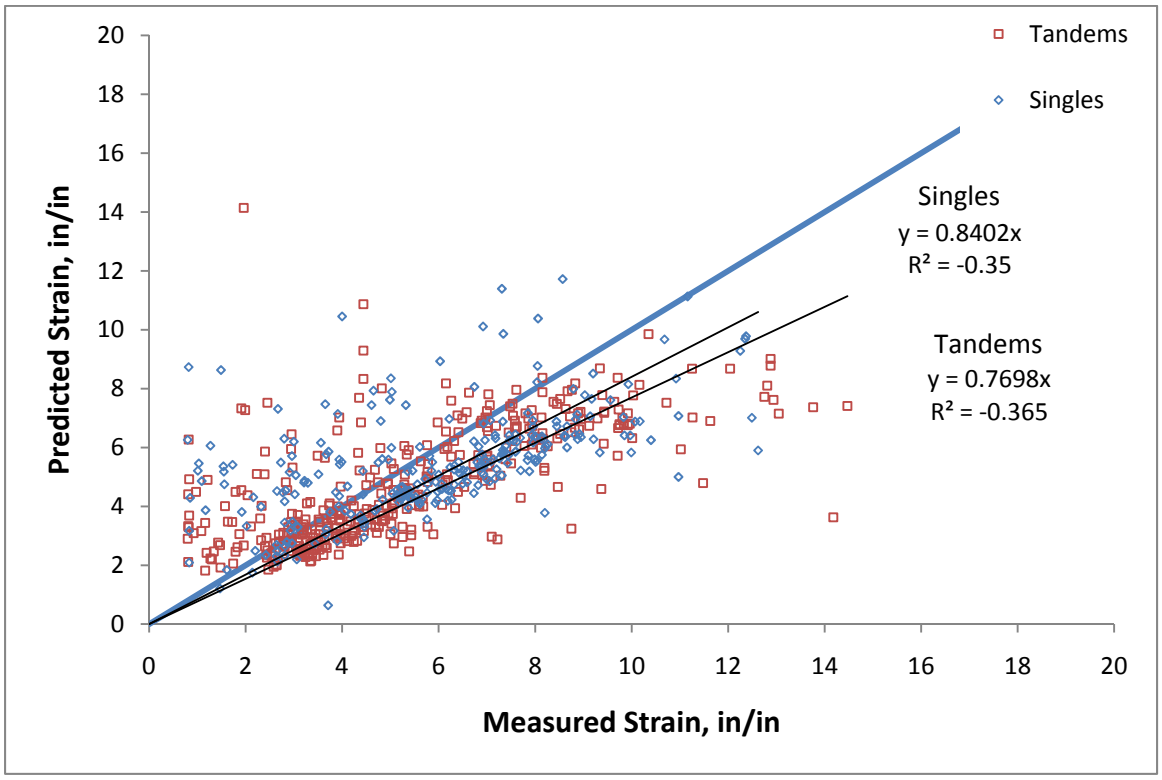


Figure 4-37 - Measured versus predicted transverse horizontal strain.

It is also observed from the transverse and longitudinal data that the disparity between tandems and singles grew. The difference between the ratios for the two cases is larger than those for the vertical pressures and is expected since the observed data indicated that there was disparity between load time durations for singles and tandems. The disparity between load time durations was observed to be larger for the longitudinal

case and the measured strain data reflects this by indicating a larger difference between singles and tandems for longitudinal strains over the transverse strains.

The optimized stress angle found during the load time analysis was implemented into the FE program and the structural responses were recomputed. Figure 4-38 and Figure 4-39 shows the results of the adjustment in load time. The steeper stress angle used for the MET process causes on overall reduction in computed load time durations for the predicted results (because of the stiffening of the HMA layers).

There is an improvement in the predicted results for the longitudinal strain, moving the data closer to the line of equality. However, the predicted transverse strains are also reduced slightly and the predictions are slightly under the measured values. The increase in stress angle has improved predictions for longitudinal strain. However as noted in the discussion on load time, the optimized stress angle for transverse strains was less than the 60° value used overall. Taking this into account, it was expected that the transverse strain predictions would be negatively affected. The same notion can be applied to the vertical pressure predictions as well.

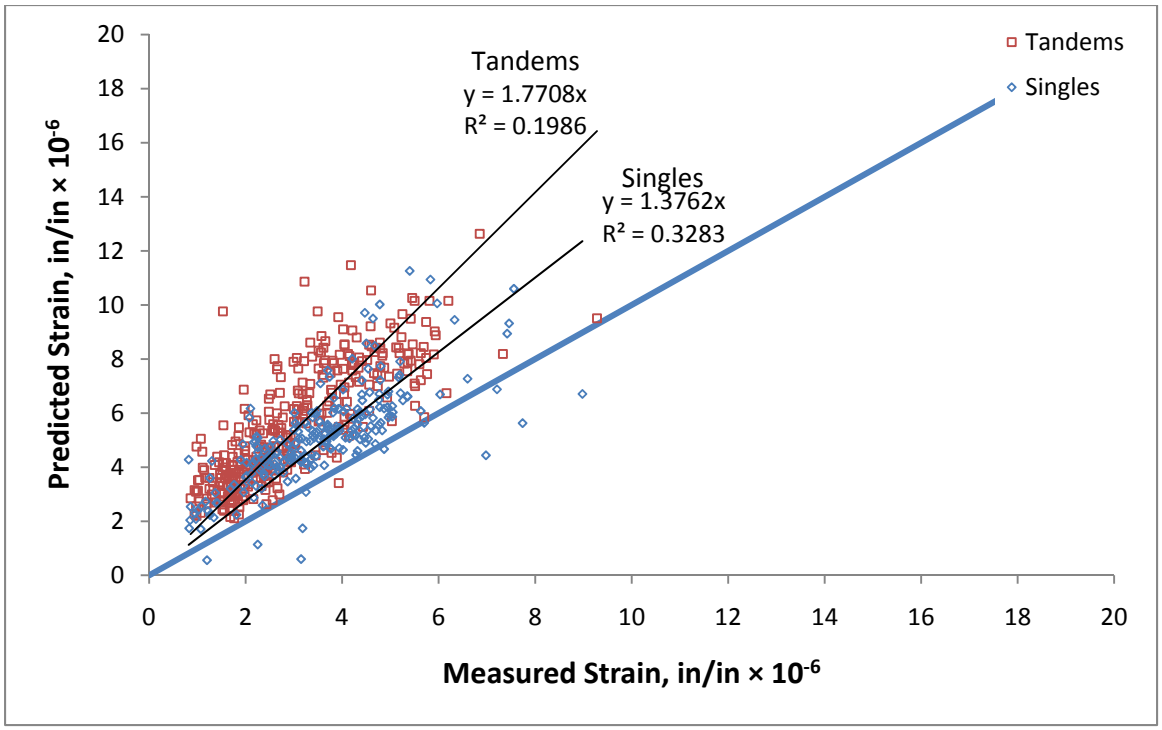


Figure 4-38 - Measured versus predicted longitudinal strain with optimized MET stress angle.

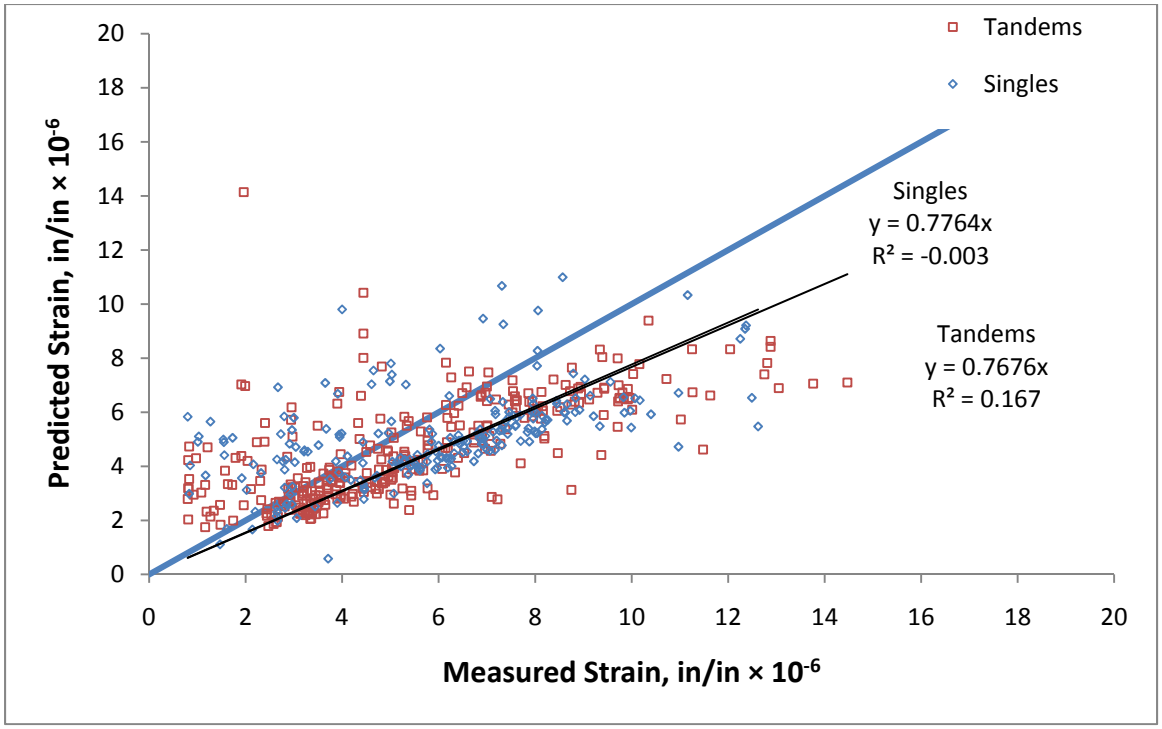


Figure 4-39- Measured versus predicted transverse strain with optimized MET stress angle.

Because of the stiffening of the HMA layers when implementing the updated MET stress angle, the predicted pressures are also reduced by a small amount. Figure 4-40 and Figure 4-41 below show the reduction of the pressures due to the updated stress angle. The subgrade vertical pressure still has better agreement compared to the base course, although predictions are still under predicted. There is practically no change due to the updated MET angle and the predictions are still crude - the possible source of errors within the instruments and the FE model (e.g. modeling of the load, tire pressures, etc.) still hold for this data.

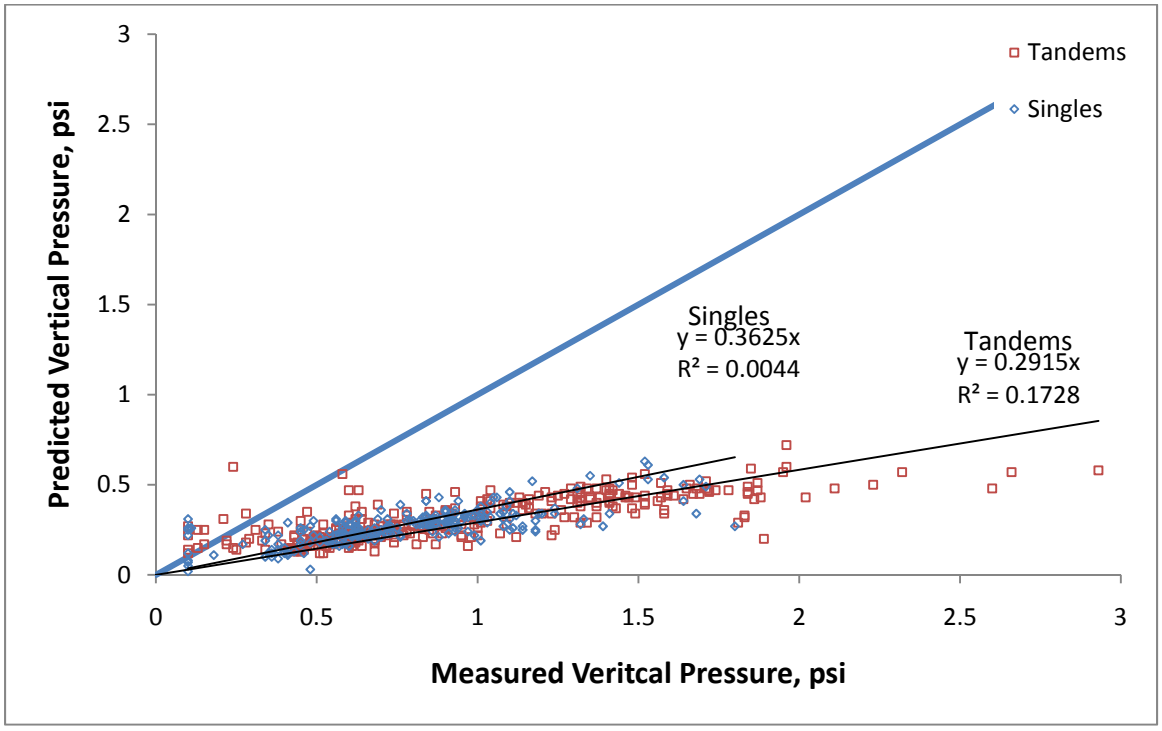


Figure 4-40 – Measured versus predicted base course vertical pressure with optimized MET stress angle.

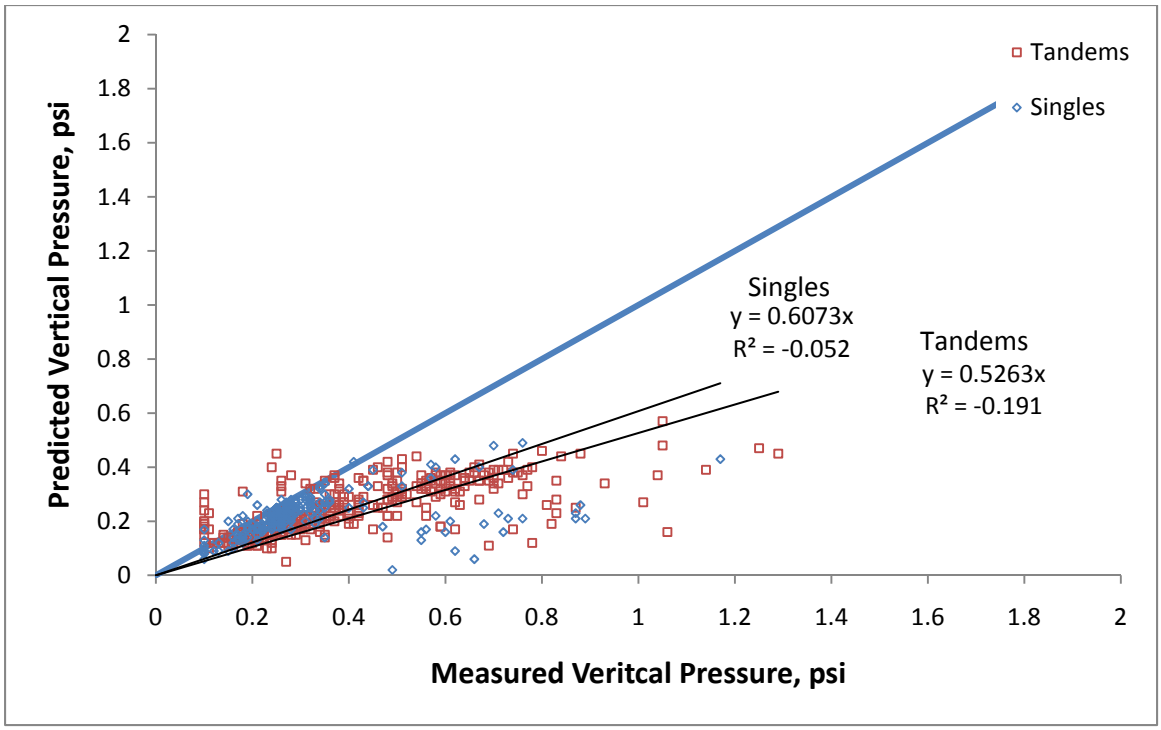


Figure 4-41 - Measured versus predicted base sub-grade vertical pressure with optimized MET stress angle.

In general, increasing the stress angle has benefitted prediction of the longitudinal strains and the affect of the inaccuracies to other predictions is relatively small. This may lead to a conclusion that the modeling of the load pulse may not be as important as other factors in predicting the structural response of this pavement. However, the increases in the stress angle have helped to better model the load pulse durations for future predictions.

There have been discussions that there is an error, or discrepancy, in the conversion of the load pulse duration from the time domain into the frequency domain. It has been suggested that some other conversion method is needed to accurately convert the load pulse duration. One suggestion is that instead of converting the time to a

frequency by simply taking the inverse (as in Equation 4-6) of the load pulse time, the frequency should actually be the angular frequency, stated in Equation 4-7.

$$f = \frac{1}{2\pi t} \quad \textbf{(4-7)}$$

where f = frequency, Hz

t = load pulse duration, seconds

Other conversions have been suggested, although the most dominant has been the angular frequency (Thompson et al., 2006). It should be noted that this discrepancy arises from the difference in the loading schemes used to develop the dynamic modulus master curve for the material and how the material is loaded in service. The dynamic modulus uses a sinusoidal loading (compression only) with no rest periods between each successive load.

In service, a pavement will experience a number of rapid loads which are not consistent in the timing. Furthermore, in between vehicles there may be seconds to minutes of rest between loadings which can have implications on the viscoelastic component of the HMA materials. One of these modulus implications is that the rest periods may allow the viscous component of the stiffness model to return to unloaded state – contrary to repeated loading where the viscous component steadily increases. Additionally the shape of the test specimen and the laboratory loading configurations do not match that of the materials in service.

The angular frequency was implemented into the FE analysis program along with the increased stress angle and the analysis was recomputed for the sample periods once

more. It was observed that there was a shift in the data similar to that in the above analysis for the optimized MET stress angle.

However, it was observed that the computed load time durations did not fit those measured in the field. A better understanding of the effects this change on the dynamic modulus models is warranted for further investigation. It should be noted however, that the change in the FE results were still somewhat reasonable when using the different frequency domain conversion.

The various methods of computing the load durations however only cause a small change in the stiffness of each HMA layer. Since the pavement in this study has a relatively thick HMA layer, it could be expected that small changes in the stiffness will only result in minor changes in the stress/strain predictions (i.e. there is a stiffness gradient through the thickness of the HMA – additional HMA thickness may not cause a linear change in the structural response).

For a thinner HMA layer pavement, there may be a larger influence of the HMA stiffness on the stress/strain computations since the HMA section of the pavement is smaller (i.e. the stiffness gradient through the thickness of the HMA may have a larger influence). In addition, the dynamic modulus model is sigmoidal and flattens near the outer limits of the model used here (and is characteristic of many other HMA materials). This implies that at very fast or very slow load durations, the unit change in HMA stiffness is less than the unit changes near the center of the sigmoidal model. With a thinner HMA thickness, the median load duration would be higher than that of the median load duration for a thicker HMA section. The interaction of changes in the load

duration on the structural responses could be understood by studying both thick and thin pavements utilizing similar materials.

4.3.3 - Other Structural Observations

As indicated in the analysis of the load durations for the horizontal strains, the transverse load times were longer in duration than the longitudinal load times. Keeping in mind the viscoelastic effect of the HMA, the longer load times for the transverse direction should cause the stiffness to be reduced for the transverse direction and thus lead to higher strains in the HMA. Figure 4-42 below is a plot of recorded peak strain values for both the transverse and longitudinal orientations.

Figure 4-42 demonstrates that the strains measured in the transverse direction are indeed larger than those for the longitudinal orientation. According to the applied linear regression model, the transverse strains are about 1.6 times larger than the longitudinal and a linear model forced through the origin indicate that the transverse strains are about 2.0 times larger than the longitudinal orientation. The comparison plot of load times, Figure 4-43 below, confirms the speculation that the load times for the transverse orientation are larger than the longitudinal.

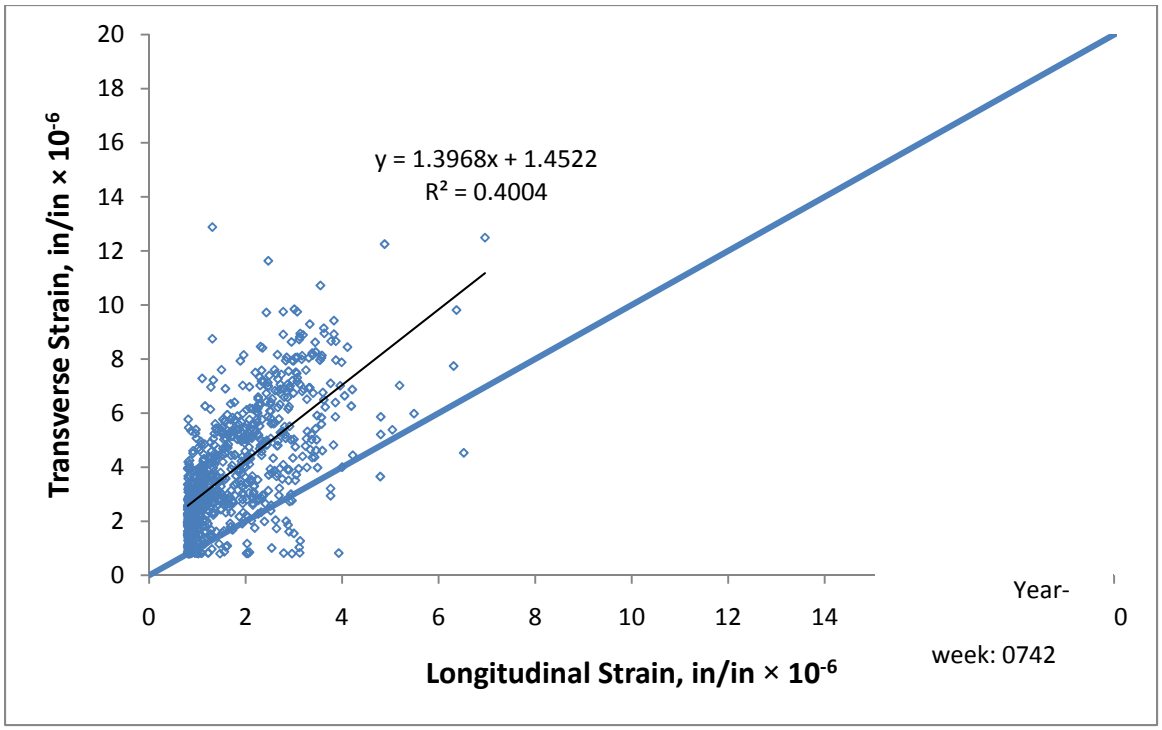


Figure 4-42 - Measured longitudinal and transverse strains.

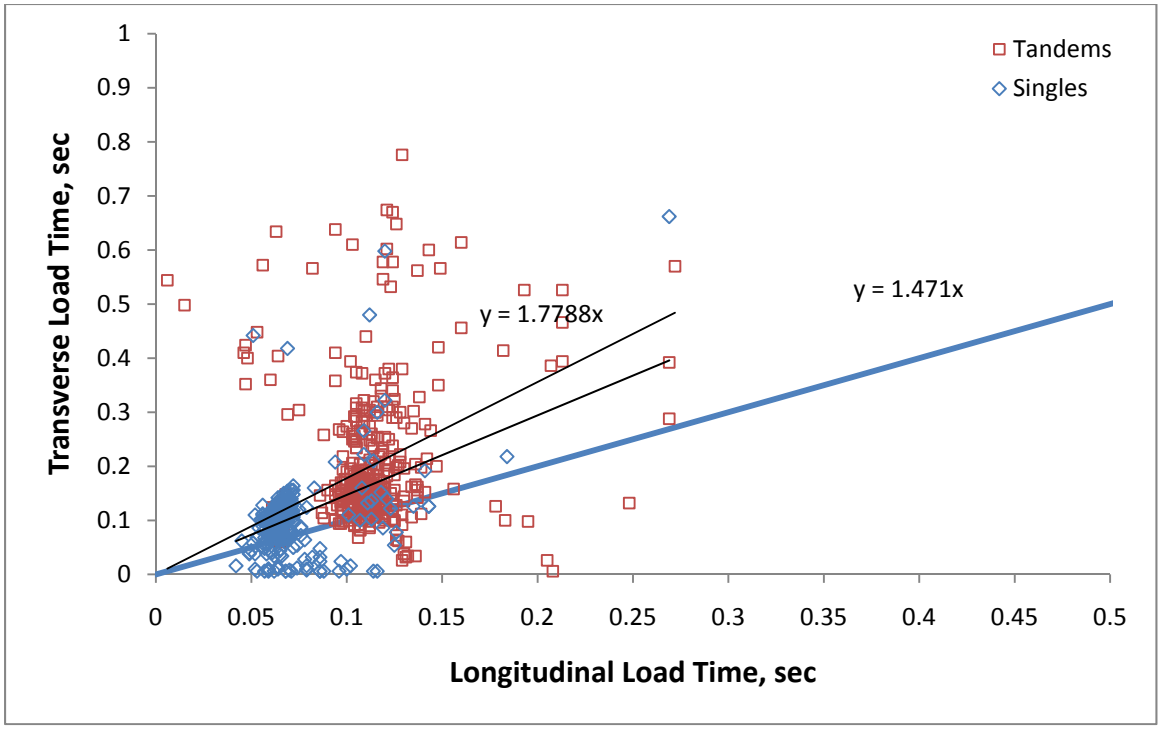


Figure 4-43 - Load time durations for horizontal strain.

4.4 - Fatigue Analysis

This research utilizes real-time pavement performance data that was recorded continuously. It may have been possible to find measured changes in data, indicating the growing quantity of damage in the pavement; however at the time of this report, there is no evidence to suggest that any damage has taken place and this is not surprising given the young age of the pavement and the robust design of the section. It is also unlikely that any of the instruments will survive a length of time to actually measure large changes.

However, what can be accomplished is the creation of a model that describes the distribution of damage across the wheelpath of the pavement. The current version of the MEPDG software uses a defined distribution of strain (and subsequently damage) across the wheel path to estimate fatigue damage (National Cooperative Highway Research Program, 2004). This process is based on assuming a relative path for a given wheel/load type - the strain is used to compute incremental damage across the pavement and the damage is accumulated over the desired design life.

Traffic loads on the pavement are assumed normally distributed to account for wander. Wheel-wander under live traffic has had limited attention, but is highly important, especially at locations where there exist significant traffic weaving/merging. The following analysis of fatigue damage gives attention to this matter.

A model for the accumulation of fatigue across the pavement has been proposed which represents more closely the actual conditions within the pavement. However, many pavements are unique in regards to the wheel path that vehicles/motorists travel-due to geometric conditions (such as super elevation, grade, objects in clear zone, etc.),

environmental conditions and traffic conditions (e.g. congestion, speed, etc.). This particular section of pavement is located downstream of a drop lane section and just before a weaving section for an on-off ramp (Fond Du Lac Ave. on-ramp and North Ave. off-ramp). More information regarding driving habits is required before supposing a universally accepted model or applying this model to pavement sections that vary drastically in the aforementioned conditions.

4.4.1 - Model Development

The representative time period used for the previous studies has been reinstituted for this portion of the analysis (the 42nd week of 2007 and the 10th and 11th weeks of 2008). These two periods contain a good sampling of data which possess high quality since little deficiencies were present within the instrumentation system. In addition, these two time periods represent the temperature extremes that are encountered in the geographical location.

The focus of fatigue analyses for HMA pavements has typically only concerned heavy vehicles with little, or any, regard for passenger type vehicles. For this analysis, only heavier vehicles have been considered, keeping consistency with past research. However, it would be possible for this body of research to understand the effects of passenger vehicles, but due the small strains witnessed in this project it is highly unlikely that any damage is truly accumulated.

Of interest for this research was the influence various loads had on the pavement not only below the load (where the highest stresses are assumed to exist), but also to assess the quantity of damage caused by stress at some distance away from the load

center. The data from this project was used to develop a model which describes the influence of peak strains at various distances away from the location of the load. This strain influence along with the measured wheel wander can then be used to calculate damage across the wheelpath.

To model the influence of peak tensile strains at distances away from the load center, strain ratios were computed for each wheel pass for various load ranges and load types (i.e. singles or duals). The strain ratio (Equation 4-8) was computed as the peak strain at the strain gauge of interest, divided by the maximum peak strain measured for each of the similar gauges (similar in orientation). By definition, unity represents the maximum value of the strain ratio.

$$Strain\ Ratio = \frac{strain}{maximum\ strain} \qquad \textbf{(4-8)}$$

Using the measured wheel wander data (offset measured from the centerline of the wheelpath), the distance of the load from each gauge was computed as shown below in Figure 4-44. The offset distances computed were such that a positive value indicates the load was positioned to the right of the instrument and negative indicates the load was positioned left of the center. It was found there was a distinct relationship between the strain ratios and the distance from the load which was independent of the wheel loads.

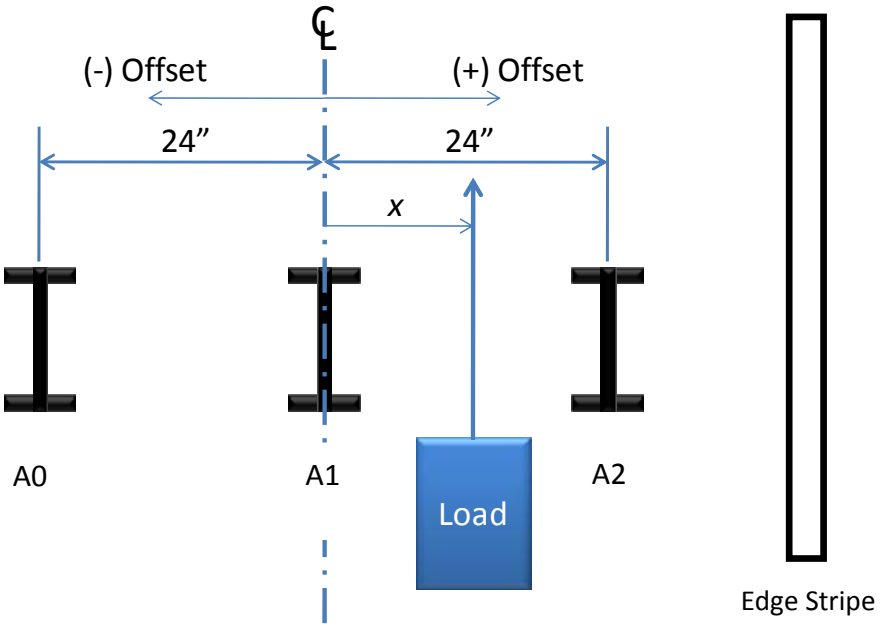


Figure 4-44 - Wheel wander offset notation.

An example strain response is shown below in Figure 4-45 – in this example the center gauge (A1) has a maximum value of about 3.0 με. Gauges A0 and A2 both located 24 inches to the left and the right of the center gauge, respectively, and have measured strains of about 1.5 με each. Thus, the strain ratio for gauges A0 and A1 would be approximately 0.5. This ratio was computed for both longitudinal and transverse strain measurements.

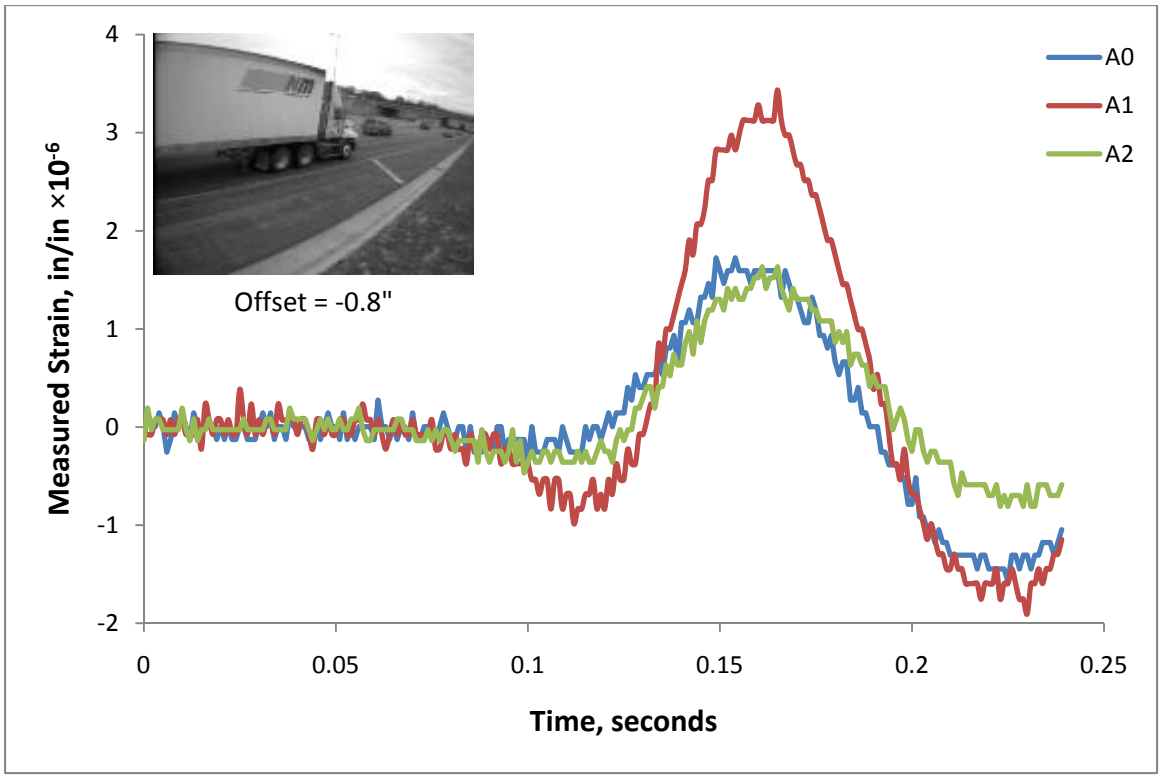


Figure 4-45 - Typical strain response for a class 9 truck.

Figure 4-46 below is a plot of the strain ratios and load distance which demonstrates the relationship between load placement and strain. Strain ratios were computed for the three longitudinal gauges A0, A1 and A2 - strain ratio included gauges A3 and B4 to complete the 12" interval across the wheel path, but the ratios are not shown. Note that data takes a parabolic form, however the shape is not symmetrical and that there is more scatter for load distances that are near the upper limit of the positive distances.

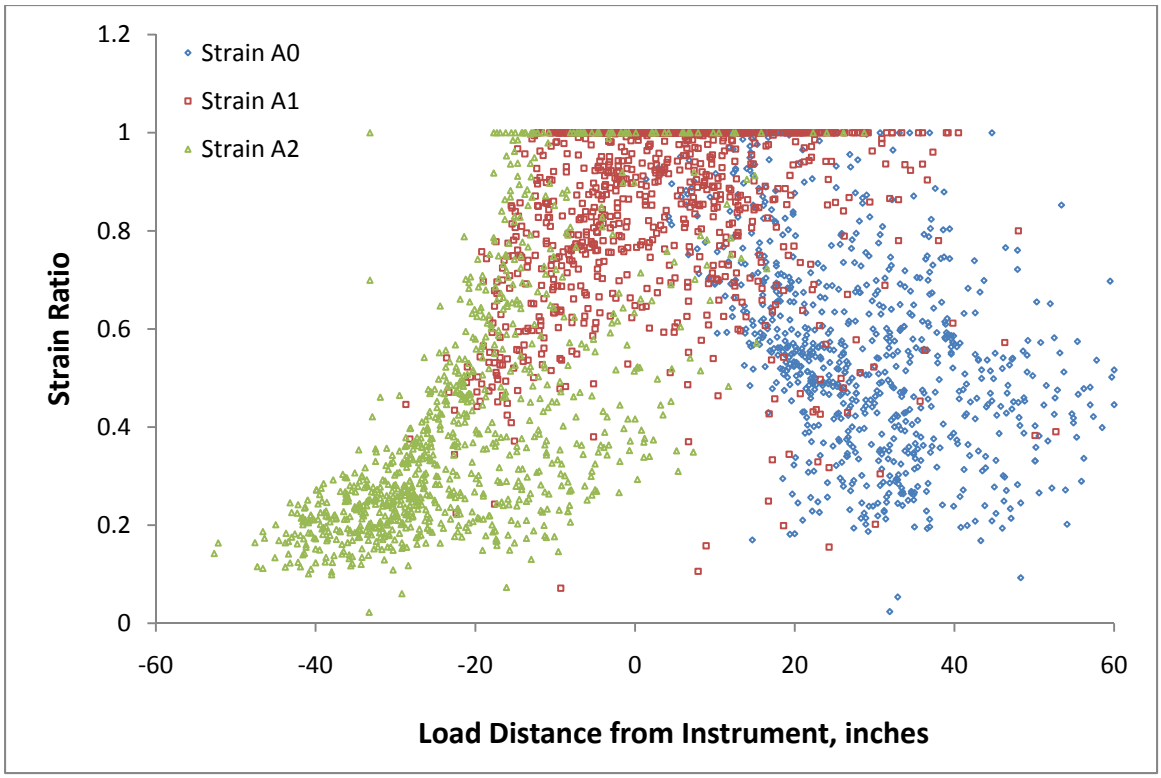


Figure 4-46 - Longitudinal strain ratio versus load distance, singles.

Figure 4-47 below illustrates the geometry of a typical axle load. Based on the notation and definition of the load distance given above, a positive distance value indicates the load is located to the right of the instrument. The instrumented section is located in the right wheel-path and when the load is located far to the right of the instrument, the wheel at the opposite end of the axle become closer in proximity to the instruments. The scatter in the figure above at the positive side is due to the influence of the wheel load from the other side of the axle. Gauge A0 contains the most scatter since it is the closest to the center of the lane (closest to the opposing wheelpath).

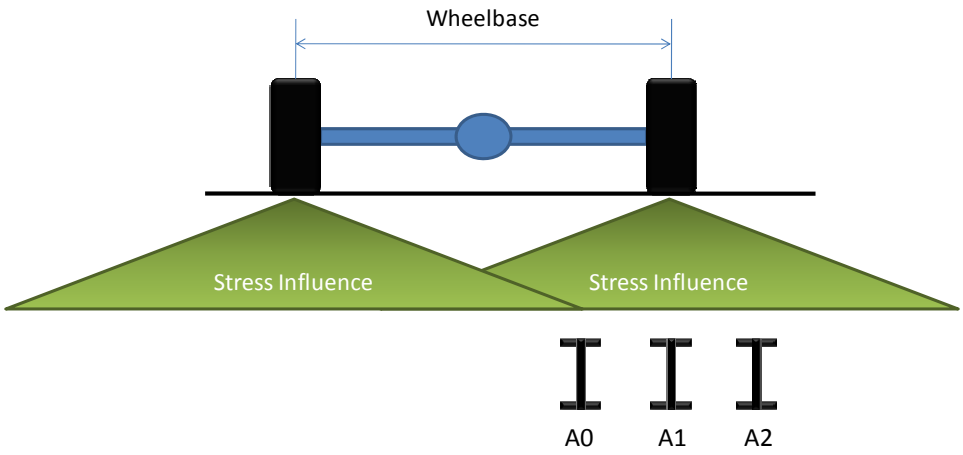


Figure 4-47 - Stress influence from axle loads.

Since the effect of the opposing wheel has a clear influence on the strain the models developed for the strain ratios included this condition. However, to provide an additional model which does not contain this condition (where the opposing wheel does not have an effect), the portion of the distance-strain ratio curve that does not exhibit this effect was essentially mirrored to provide a symmetrical pattern. This was thought to be useful for instances where the opposing wheel loads do not significantly overlap. This may be the case of thinner HMA sections (such as the case of stress overlap for the analysis of load pulse durations above). In addition, a symmetrical model could be used to model closely spaced loads by superimposing the loads on top of each other.

In order to generate meaningful models that could be used for future fatigue analyses, the data was separated into 4 distinct groups resulting in 4 models for each group. Initially, load ranges were used to generate the relationships to enhance the correlation and fit of the models, however reduction in scatter was not significantly reduced and regression analyses indicated the load was not a significant predictor of the strain ratio at the alpha = 0.5% level.

This finding was important since the assumptions during FE analyses was that the pavement behaved in a linear elastic manner. The good relationship found between the strain ratios and load distances indicates that the HMA does behave in a linear elastic fashion for these types and rate of loadings.

Fatigue models were based on 4 cases – longitudinal and transverse for singles and longitudinal and transverse for tandems. A mix of vehicle types was initially used to generate the models, however there was difficulty in separating the actual wheel loads during the preliminary analyses (i.e. difficult to assign measured wheel load to each particular wheel event). Because of this difficulty only class 9 vehicles where used since it can be reasonably assumed that the first load is a single axle and the following axles are tandems (and ignoring the possibility of super-singles instead of dual tires).

Plots of the data generated for the single and tandem longitudinal strain are shown in Figure 4-48 and Figure 4-49 for both the single and dual wheel loads. The data was filtered in addition to the aforementioned vehicle class filter. Only logical values of offset were considered which implies that offset measurements outside of the physical range were eliminated. Measurements outside of the physical range are recorded in the database but represent errors in the data collection due to merging vehicles, slow traffic, and low signal amplitude. In addition there were a number of computed strain ratios of 1.00 for loads that hit a significant distance from the center of the gauge. Many of these instances were associated with errors in the strain peak detection process where a peak value was missed or incorrectly identified and the data associated with these were eliminated.

The bold lines in Figure 4-48 and Figure 4-49 represent the best fit curve found through linear regression of the data points. As noted before, other potential predictors were included in the regression analyses; however it was found that the best representation was found as a function of the load distance from the instrument. More detailed information regarding the regression models is presented further into the discussion.

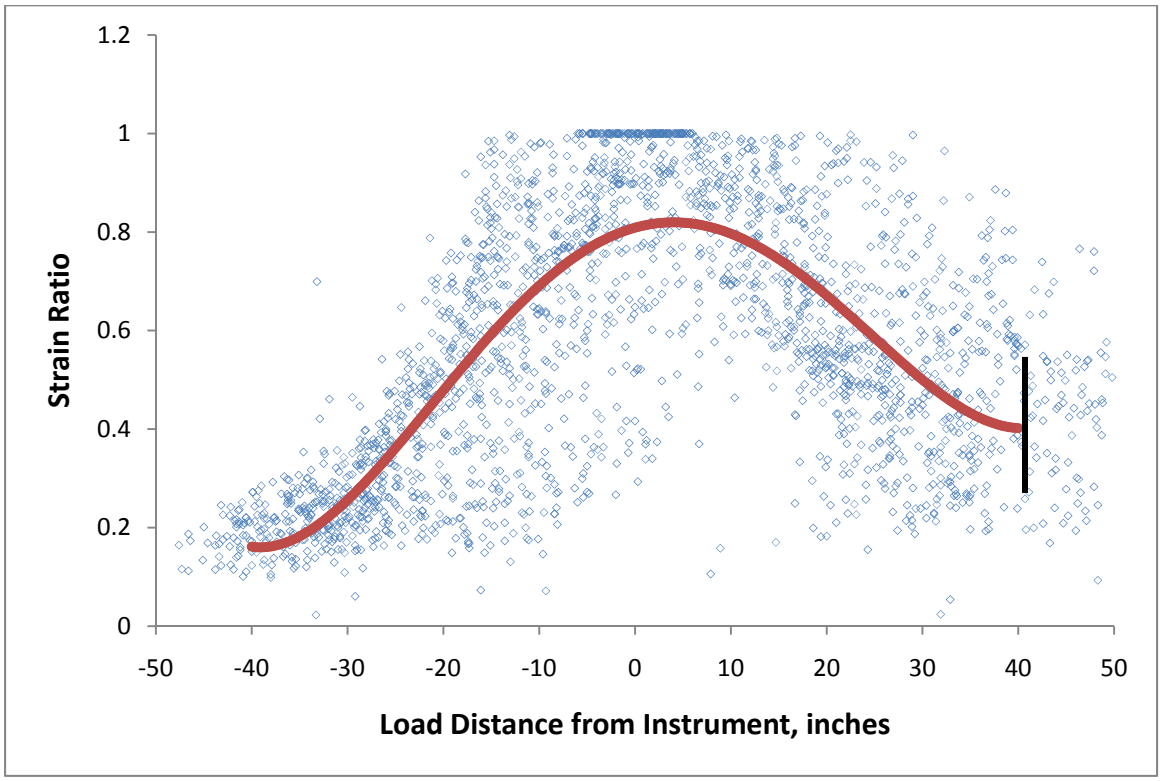


Figure 4-48 - Strain ratios for longitudinal strain, singles.

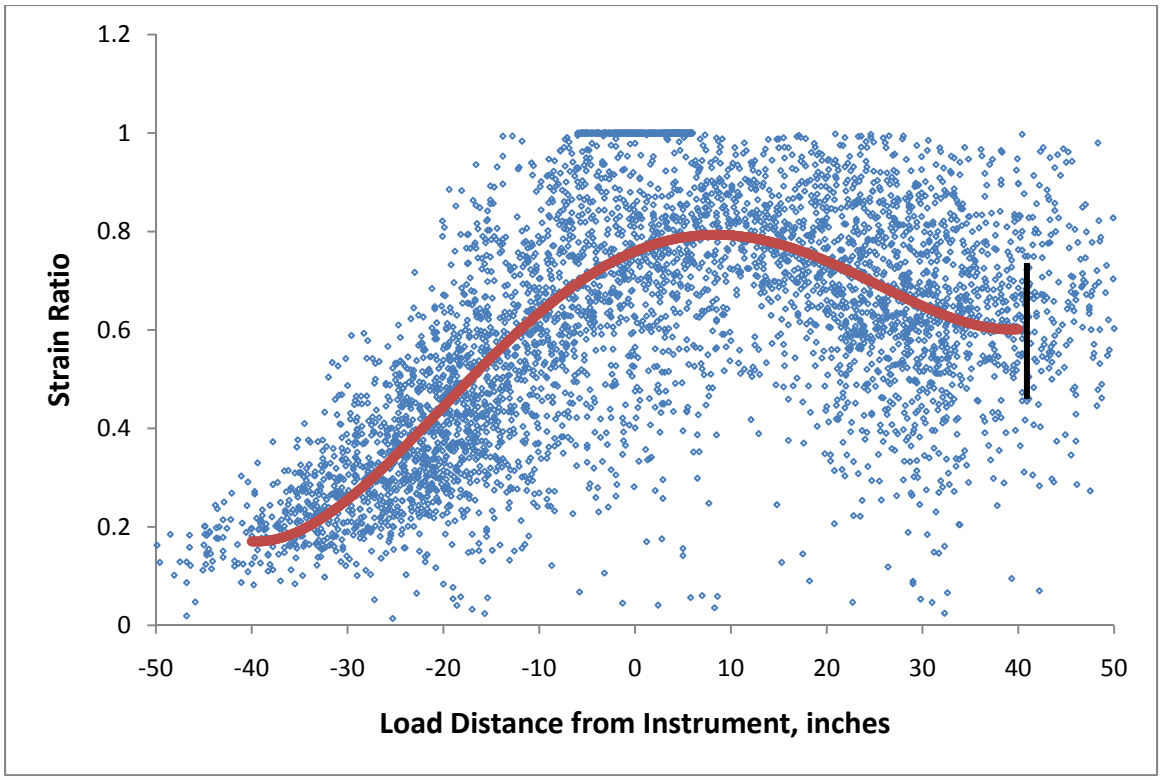


Figure 4-49 - Strain ratios for longitudinal strain, duals.

The distribution between the strain ratios is asymmetric due to the effect of the opposing wheel load at the other end of each axle. These figures only provide the strain influence for one wheelpath and not the entire lane. The strain influence for the entire lane could be theorized by mirroring the above data in Figure 4-48 and Figure 4-49 about the dark vertical line at the right end of the best fit line. However the strain influence for the entire lane cannot be stated as such since the instrumentation did not encompass a large enough area to confidently make a model.

Between the single and dual loads, it is apparent that this effect is amplified. It is suspected this is due to the wider influence area of the dual tires contrary to the narrower stance of the singles. The distribution for the singles is narrower and more symmetrical than that for dual loads, especially when loads hit to the right of the instruments. Figure

4-50 shows the two regression models superimposed on each other, demonstrating the differences between the models.

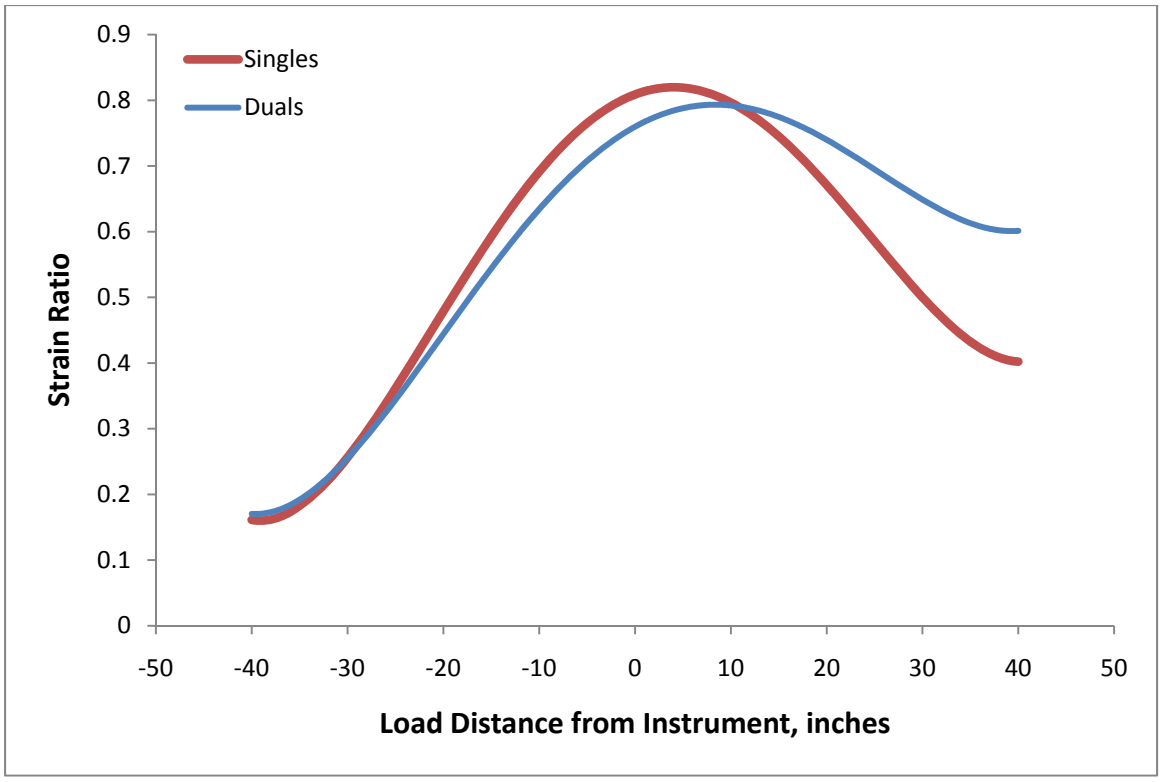


Figure 4-50 - Comparison of best fit regression models for longitudinal strain ratios.

The case of transverse strain ratios was found to be quite different in terms of the behavior of the measured strains. Figure 4-51 and Figure 4-52 provide plots of the strain ratios versus distance for the single and dual load configurations. The is still some mild asymmetry to the data, indicating the effect of the opposing wheel load, however it is much less significant than the longitudinal cases.

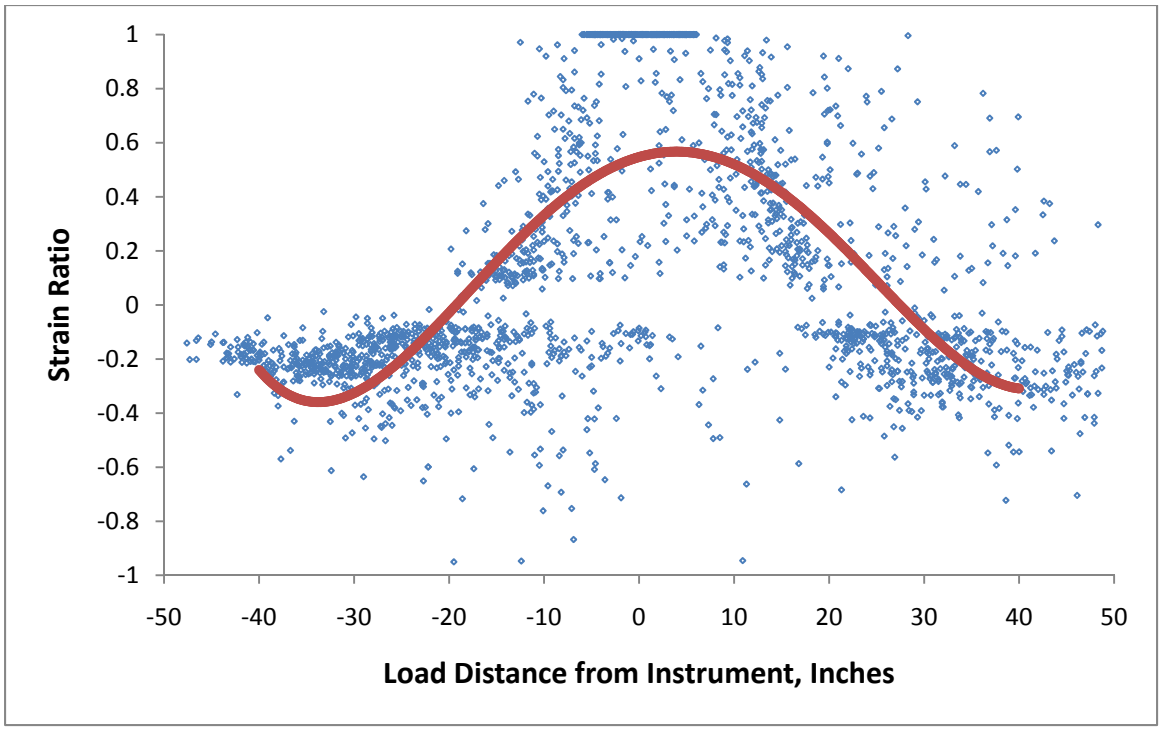


Figure 4-51 – Strain ratios for transverse strain, singles.

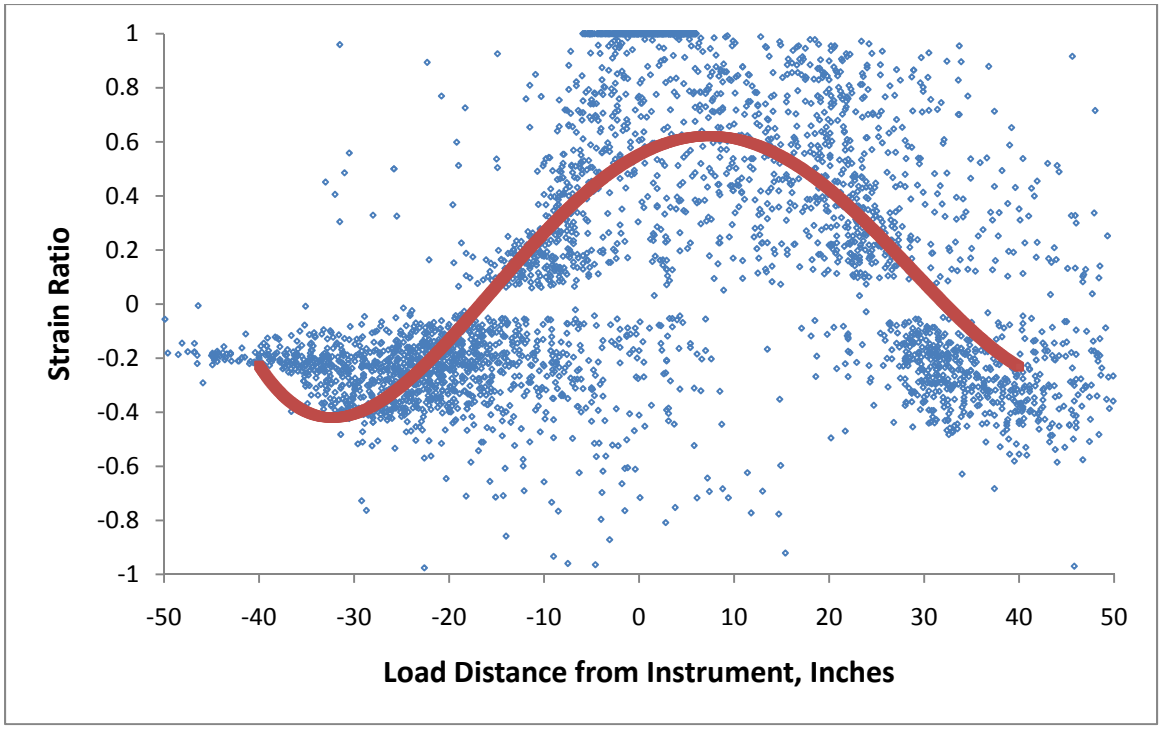


Figure 4-52 - Strain ratios for transverse strain, duals.

What differentiates the transverse cases from the longitudinal cases is the existence of negative strain ratios. This indicates that at distances away from the wheel load, the measured strain values were actually negative (compression). This is not an astounding discovery since it is understood that cracking in HMA pavements can initiate at the top of the pavement and propagate downwards – evidence of this phenomenon. In addition, simple layered elastic analysis (LEA) of loads indicate a small reversal; however, the magnitude of the reversal for LEA is not as large.

The continuous beam analogy used previously in Figure 4-11 can be applied to understand the nature of the pavement deflection under a moving wheel load. The strain reversals are not unique to transverse strain though. The analysis of the strain ratios above only consider the peak values of the strain events – observations of the entire strain signals indicate that strain reversals takes place for longitudinal strains.

As stated previously, most longitudinal strain measurements contain a compression component that precedes and follows the main tensile pulse. This phenomenon is demonstrated in Figure 4-45, most notably for the center gauge, A1. It has been observed that this compression pulse preceding and following the main tensile event is absent for transverse strain measurements. However, there does exist compressive strains at a distance from the load center which is exhibited by the transverse strain signals for a class 9 vehicle below in Figure 4-53.

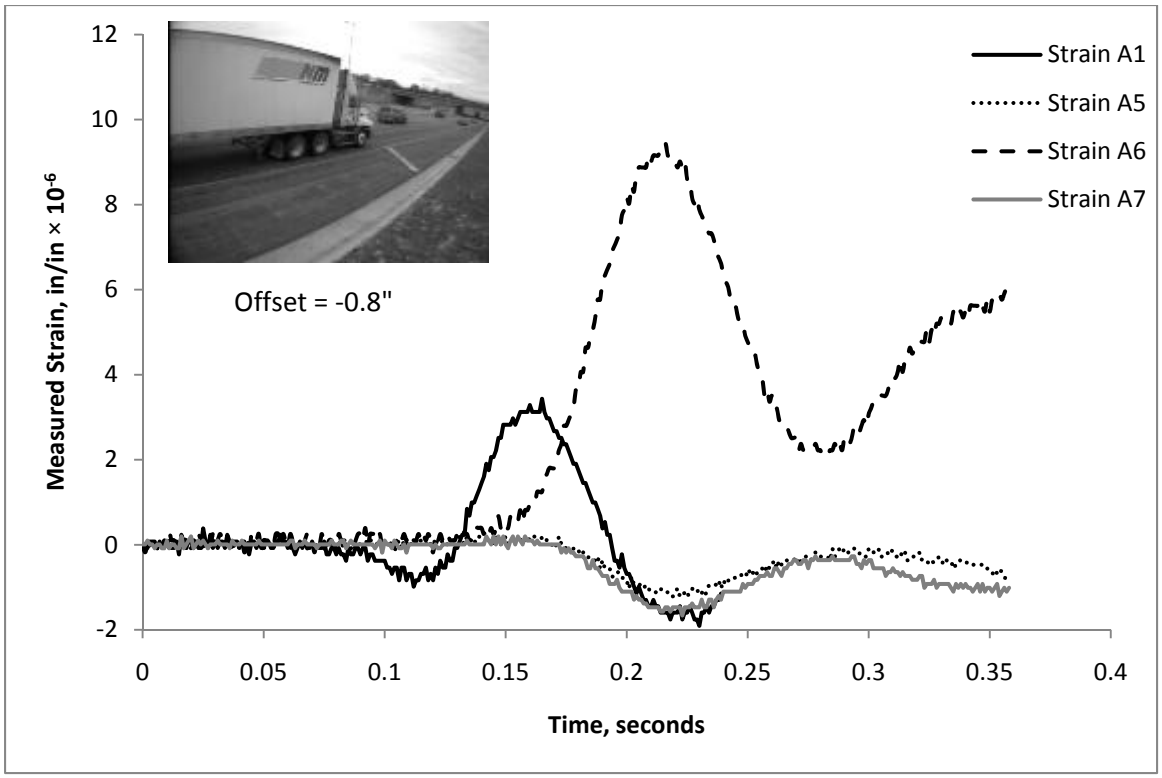


Figure 4-53 - Typical transverse strain response.

The vehicle in Figure 4-53 had a measured offset value of only -0.8 inches which is nearly on top of the center gauge (sensor A1 & A6 in the figure). The transverse strain variation demonstrates that the bottom of the pavement is in tension directly under the load and in compression at the locations of the other gauges (24 inches left and right of the center gauge). After the main strain pulse, there is a slight rise in the signal which is due to the following set of tandem axles. The response in Figure 4-53 is typical for many of the loading situations observed for this pavement.

4.4.2 - Asymmetric Regression Models

The red solid lines in Figures 4-49 through 4-53 represents the regression model that was generated from the data. The resulting regression model parameters are listed below in Table 4-7 and take the form of Equation 4-9. For each case, various predictor variables were included in the regression analysis; namely speed, wheel load, load distance, as well as functions of these variables.

$$Strain\ Ratio = \beta_0 + \beta_1 D + \beta_2 D^2 + \beta_3 D^3 + \beta_4 D^4 \qquad \textbf{(4-9)}$$

where D = load distance from instrument, inches

β_i = regression coefficient

Table 4-7 - Resulting asymmetric regression model variables and statistics.

Response Model	Regression Coefficients					Multiple R^2	Adjusted R^2	R^2
	Intercept	D	D^2	D^3	D^4			
Singles - Longitudinal	0.8089	0.0054	-6.6996E-04	-1.5054E-06	2.1277E-07	0.75	0.56	0.56
P-value	<< 0.001	<< 0.001	<< 0.001	<< 0.001	<< 0.001			
Duals - Longitudinal	0.7598	0.0081	-4.8112E-04	-1.6720E-06	1.5464E-07	0.72	0.51	0.51
P-value	<< 0.001	<< 0.001	<< 0.001	<< 0.001	<< 0.001			
Singles - Transverse	0.5472	0.0101	-1.2573E-03	-6.8484E-06	4.6482E-07	0.71	0.50	0.50
P-value	<< 0.001	<< 0.001	<< 0.001	<< 0.001	<< 0.001			
Duals - Transverse	0.5524	0.0186	-1.1906E-03	-1.1658E-05	4.3873E-07	0.71	0.51	0.51
P-value	<< 0.001	<< 0.001	<< 0.001	<< 0.001	<< 0.001			
* Where D = Load distance from instrument, inches.								

Speed was considered in the regression analysis since the HMA materials are load rate dependent due to viscoelasticity (see Sections 4.2.1 and 4.2.2). In addition, the wheel load was also considered since it was thought that load magnitude might influence the stress field distribution in the pavement. In both of these cases, these predictor variables were found to be insignificant to the regression, although there was some

improvement model fit (multiple R^2 statistic). Since the improvements from these additions were small they were eliminated from the regression for model simplicity.

It was found that the strain ratios were primarily dependent upon the load distance from the instrument. A simple polynomial fit produced the best representation of the data. This relatively simple model can be applied in future analyses with little input. The models were produced using the full spectrum of the strain ratios (i.e. included the asymmetric data caused by the opposing wheel load).

All of the above models listed in Table 4-7 have Multiple R^2 values greater than 0.70 although the coefficient of determination values ranged from 0.50 to 0.56. The potential predictor variables were added and subtracted until the Multiple R^2 values were a maximum with the fewest number of potential regressors (which included potential regressors other than load distance). The above models were selected to best represent the strain ratio as a function of load distance.

Although the models do possess a rather low coefficient of determination, the models represent the average range of the data which is indicated in Figures 34 to 39. The experimental data presented and used in this analysis contains a significant amount of scatter. The data was filtered to reduce the amount of scatter, but was done in such a manner to maintain quality and accuracy.

It was not reasonable to reduce the scatter by removing data in an unjustified manner. The aforementioned methods used to reduce a large portion of the scatter (class 9 vehicles only, removal of loads with illogical offset values, etc.) were felt to be justifiable and reasonable.

Other methods were attempted to reduce the scatter, but either did not improve the relationship or were somewhat unjustifiable. These methods included deletion of data with very low strain ratios that were located over the instrument. For instance, it could be reasonably assumed that the strain ratio should equate to unity when the load is directly over the instrument. An attempted method was to force the strain ratios to unity when the load was located within the zone of influence of the instrument (i.e. load distance from the instrument was < 6 inches, assuming the next closest instrument was 12 inches away).

There was a natural variability that was expected when designing the experiment and was such because gaining the additional precision would require a significant amount of experimental effort. One source of error was generated from the wheel wander system which was found to have a 95% Confidence Interval of approximately 2.0 inches which implies each offset (and hence distance from load) measurement could vary as much as ± 2.0 inches.

The resolution of the strain measurements is also a large contributor to the overall variability of the strain ratios. The instruments are located across the wheel path at 12 inch centers – a wheel passing between two instruments would cause the peak strain to be directly under the load, but the instruments to each side of the instrument would theoretically measure the same strain. The measurement could also technically be the maximum strain and the calculated strain ratios would then be unity for both instruments.

4.4.3 - Symmetric Regression Models

In order to provide models which could be used to represent only one load with no influence from other axles, the data used to produce the asymmetric (‘real’) models

were used to generate symmetrical data that did not include the data affected by opposing wheels.

The data which represented strain ratios for loads that were located to the left of the instrument were retained while data pertaining to loads hitting to the right of the instrument were removed. The resulting one-sided data was then mirrored to represent loads hitting to the right of the instrument. A multiple-linear regression was then performed to create the models listed in Table 4-8.

Table 4-8 - Resulting symmetric regression model variables and statistics.

Response Model	Regression Coefficients			Multiple R^2	Adjusted R^2	R^2
	Intercept	D^2	D^4			
Singles - Longitudinal	0.7655	-8.0191E-04	2.6305E-07	0.77	0.59	0.59
P-value	<< 0.001	<< 0.001	<< 0.001			
Duals - Longitudinal	0.7126	-7.3828E-04	2.4524E-07	0.73	0.54	0.54
P-value	<< 0.001	<< 0.001	<< 0.001			
Singles - Transverse	0.4337	-1.3050E-03	5.4323E-07	0.66	0.43	0.43
P-value	<< 0.001	<< 0.001	<< 0.001			
Duals - Transverse	0.3531	-1.3118E-03	5.6885E-07	0.62	0.38	0.38
P-value	<< 0.001	<< 0.001	<< 0.001			
* Where D = Load distance from instrument, inches.						

A similar polynomial fit was found to best represent the data, although the odd-powered variables were removed due the symmetry (and found to be insignificant regressors during the analysis). The models follow the general form found in Equation 4-10 below.

$$Strain\ Ratio = \beta_0 + \beta_1 D^2 + \beta_2 D^4 \qquad \textbf{(4-10)}$$

Similar to the real models in the previous section, the symmetric models fit the data with a similar degree of accuracy. The amount of scatter in the data is logically the same as that found in the real models and the sources of the variability are also the same.

These models, including the asymmetric real models, represent an average of the data that has been collected and processed. Usage of these models should be able to accurately predict strain ratios for the various scenarios. When used in conjunction with an analysis to produce the maximum strain for a given wheel load, it is then possible to estimate strains at distances away from the load.

These models should only be used for distances ±40.0 inches from the location of interest which is the limit of the regression model. Extrapolating beyond these limits may produce strain ratio predictions with significant error. The polynomial function used to model the strain ratios contain an inflection point near the ±40.0 inch prediction limit and could result in infinitely large strain ratios. Caution must always be exercised when implementing regression models beyond the limits of the data used to develop the model.

4.4.4 - Implementation of Fatigue Model

When computing the fatigue damage of the HMA pavements, the general fatigue model described in Chapter 3 can be used in conjunction with the HMA stiffness and strain to compute an incremental damage for a particular load event. The developed model can be used to generate the strains at all distances across the wheel path. These strains can then be implemented to generate cumulative fatigue damage across the entire wheel path. Each axle pass can then have an associated damage profile which can then be summed up with each progressive axle pass, taking into account wheel wander by

simply superimposing the damage profile for each accumulation summation. The following steps allow the damage to be computed.

1. Given a particular traffic profile and assumed axle load distribution.
2. Solve FE or LEA problem to find maximum tensile strain for a particular vehicle.
3. Calculate strain ratios across the width of the wheelpath/lane.
4. Solve for strain magnitudes across the pavement using strain ratios and max tensile strain.
5. Using a fatigue model, use the computed strains along with the stiffness to develop the incremental damage.
6. Once the incremental damage is solved, steps 2 to 5 can be repeated and the cumulative damaged summed.

The natural wheel wander across the pavement can be easily taken into account by either dividing the pavement into intervals across the width or implementing a Monte Carlo method to randomly select wheel wander from a given distribution. This allows one to tailor the analysis based on different variations of wheel wander.

In future studies it may be interesting to better estimate the stiffness profile of the pavement in question. However this would require a much more in depth experiment to capture the stiffness profile of the pavement and analyze the actual measurements in conjunction with the viscoelastic model presented and used in this study.

The fatigue damage calculation above is unique in that the actual damage profile for a real pavement has been developed and can be implemented for future damage calculations. Previous attempts at modeling this effect use an assumed strain profile based on simple LEA approaches (National Cooperative Highway Research Program,

2004). While the previous attempts have been modeled as accurately as possible, a process and a model have been developed to produce a more realistic approach to accounting for damage for real traffic.

Chapter 5 - Conclusions

This research covers over 3 years of experimentation and development work which provided the data and the tools to generate the information and results. One of the project outcomes, outside of this work, from the instrumentation work was to generate material properties and model the pavement system using analysis programs readily available and compare the results to those generated by the new MEPDG software. The selected pavement design will also serve as a long term experiment, demonstrating to professionals in Wisconsin the potential performance gains of the perpetual pavement philosophy.

The results generated in this research took additional steps to closely analyze the experimental data and develop tools to rigorously process all of the collected data. The results only represent a fraction of the collected data. The following sections describe the conclusions drawn from the results presented in Chapter 4.

5.1 - Environmental

The environmental conditions for pavements is of great importance and must be accounted for in some manner for an accurate analysis. The effects of the environment on pavement systems are well understood and the experiment for this research implemented the tools necessary to accurately monitor the environmental conditions. This was accomplished by using temperature sensors, soil moisture probes, pyranometers, and an anemometer to capture the most important environmental factors.

Of all of the recorded data to date, over 5000 samples (representing over 50 days with temperature ranges from 7° to 81° F) from the data collection system were used to

construct environmental models to predict pavement layer temperatures. These models can be used for future predictions and were implemented into the finite-element program created for this research. The pavement layer temperature is very critical in analysis of the HMA layers since the stiffness of the material is heavily dependent on temperature. The data that was selected was verified and represents a complete set of data with little error. It should be noted that the data is not of a consecutive nature, but were actually separate data samples from different periods of time. This gives samples over a wide range of time periods, providing a good average over different seasons

During the analysis of the environmental data, good relationships for the HMA layer temperature were found. Linear regression models were built of these relationships and were tabulated in Chapter 4. It was found that the pavement surface temperature could be predicted from the ambient air temperature and the solar radiation at the site. This model is only accurate for the surface conditions at the time of the data collection as the pavement texture and color will likely have an impact on the adequacy of this model. Over time, wear on the pavement will likely strip the upper coating of the binder, exposing the rather light colored limestone aggregate. The lighter colors may have a tendency to reflect the visible light radiation, thus making the surface temperature less dependent on the total solar radiation received.

It was also found during the regression analysis that wind speed at the site was not a significant predictor for surface temperature. It was theorized that increased wind speed may cause more heat to be conducted away from the pavement surface. It is possible that the wind speed measured at a distance above the ground surface varies greatly over the wind speed at the pavement surface (due to the viscosity of the fluid). In

addition, passing vehicles will absolutely create air flow over the pavement surface in which case makes the natural wind speed a small factor in heat transfer. This argument may be different on roads with rather low traffic volumes.

A regression analysis of the HMA pavement layer temperatures indicated that the layer temperatures were dependent on the ambient air temperature, solar radiation, and the pavement surface temperature. The regression model statistics indicated an excellent fit to the data; however the models were generated at the specific depths of the instruments (which was done to fill in blanks in the project database). The depth of each instrument was used to generate another regression model that could then be used to predict the layer temperature any depth in the HMA layers. The adequacy was found to be good based on the model fit statistics.

One caveat to these models was that the solar radiation and pavement surface temperature must be known. To further simplify the model, only the ambient air temperature and depth was used to predict layer temperatures. The resultant model was only slightly less adequate than the previous model, but far easier to implement.

5.2 - Load Pulse Duration

Due to the viscoelasticity of the HMA materials, the stiffness of the HMA layers are both temperature and load rate dependent. The Method of Equivalent Thickness (MET) is commonly used to predict the load pulse durations in pavement systems. This duration is then converted to a frequency and used in the dynamic modulus material model in conjunction with the temperature to calculate the stiffness. For each load succession, only one stiffness is used in most analyses even though there is likely a

stiffness variation as the load is passing. This stiffness variation during the loading should be reasonably small, although there is little evidence in support of this and a more in depth analysis may be beneficial.

Since the pressures and strains were measured for each load, it was possible to compute the load durations from the experimental data. These load durations were then compared to those found through MET.

Analysis of the vertical pressure load pulse durations indicated that the MET process can provide reasonable predictions. Data for the vertical pressures contained more scatter than that of the horizontal strain measurements which is likely due to the amount of stress distribution and overlap that occurs at depths below the surface. The data scatter was higher for the lower subgrade than the upper base course pressure. In both cases, little distinction was noticed between the single and tandem axle load groups – indicating the stresses are fully overlapped at the depth of these instruments.

The load pulse duration analysis was seen as a good way to confirm the MET concept since the shape of the stress measurements were uniform and best represents the assumed stress distribution in the MET theory. This is because the typical vertical stress measurements exhibit a compression only response with no stress reversals. Because of the construction of the cells, it might be argued that the cells will not register a tensile event, however the plates are under a constant static stress due to the overburden stress. Any stress that might cause a tensile stress to occur at the cell locations would cause the overburden stresses to be relieved. This was never observed.

A long term relief of the vertical stress may be present due to changes in soil moisture. This effect was not investigated and any observed changes in static pressure

could largely be attributed to temperature changes which affect the cells readout do to expansion and contraction of the materials used to construct the instrument.

The horizontal strains were also analyzed in a similar manner to the vertical pressure load pulse durations. It was observed that the standard MET process was over estimating load pulse durations, especially for the longitudinal strain case. This may have been expected for the longitudinal strain since the responses typically contain compression strains preceding and following the main tensile strain event. The behavior implies that the strain/load rate for this orientation occurs at a faster rate than for other orientations such as vertical pressure or transverse strain (in which the stress reversals are absent from the recordings.) However, predicted load times for the transverse strain measurements were also too long even with the stress reversals absent. Comparison of differing temperature regimes for the measured load times indicated little change between high and low temperatures, while a shift in the predicted load times, however small, is apparently more than those measured.

In order to better predict the longitudinal and transverse strain load times, different methods were attempted to improve these predictions. This included instituting the angular frequency instead of the normal frequency conversion (inverse of the load time) which has been the subject of recent discussions regarding this topic. The best solution found was to modify the assumed 45° stress angle used in the MET process. Steepening the assumed stress angle essentially reduces the load times. The best fit angle was found by simply minimizing the errors between the predicted and experimental data.

The average angle for the base and subgrade vertical pressures from this minimization was found to be nearly the assumed angle of 45°. However, the average

stress angle found for both longitudinal transverse load times was approximately 60°. Re-analyzing the data with the updated stress angle had little to no effect on the predictions of the load times. Ideally the stress angles for both the longitudinal and transverse strain load times would be about 68° and 60° respectfully while the vertical pressures can remain at the standard stress angle (to produce the least amount of prediction error). It is theorized that the ability of the materials to carry and distribute the load is subject to other factors and that the MET process which uses an assumed transformed section does accurately depict this.

The relationship between vehicle speed and experimental load pulse durations is well defined for the longitudinal and transverse strains. The same relationship for vertical pressures is slightly less defined due to the inherent scatter in the data. Linear regression models for load time as function of speed were applied, however these models are only valid for this pavement type. From the data, it is apparent that there is a distinction between the load times for singles and tandems, the strongest evidence for the longitudinal strain load times. As stated before, there is little distinction in regards to the vertical pressures, likely due to the depth of the instruments where full stress overlap of the loads occurs. Observations of the recorded strain responses indicate that at the bottom of the HMA layer, most of the stresses are overlapped for multi-axle loads and the measured strains do not have a significant return to an unloaded state between axles.

5.3 - Stress/Strain Observations

A complete finite-element (FE) program was written to analyze the loads that passed over the test section of pavement and the results stored to a database. All of the

necessary material models and loading types were included in the analysis with some minor assumptions. The FE data analyzed was first computed using the standard MET procedures and then re-analyzed using the optimized stress angle of 60°. The data was then queried joining FE data with the experimental data.

Upon inspection of the data concerning vertical pressures, the FE results were similar to the experimental results. In general though, the measured pressures were lower than the analytical results. The subgrade vertical pressure results agreed more closely with the experimental data over the base layer vertical pressures. However, there does exist an element of error in the experimental results since the pressure plates disrupt the stress field in the soil and can contribute to this mild inaccuracy.

It was observed that for the vertical pressures, there was little distinction between singles and tandems. Since the analysis of the load pulse durations indicated that there was little distinction due to stress overlap, this finding was expected. This reinforces the idea that the MET process can be used to predict the load times; however the user must be careful to understand the extent of the stress overlaps at depths. Since it has been observed that there is an increase in the stress overlap from the strain down to the pressure cells, it is clear that the amount of stress overlap does indeed vary. However, since no data is available above the bottom of the HMA layer in this project, the extent of this gradient can't be realized.

The horizontal strain analysis indicated that the disparity between singles and tandems increases which is in agreement with the findings for the load time. Again this is likely because there is less stress overlap closer to the surface of the pavement.

Though this disparity is greater for the longitudinal strains over transverse which is exactly what was indicated from the load times.

The increased stress angle for the MET process produced FE results with an overall reduction in the analytical strains. This was expected since the increase in load times will result in stiffer HMA layers, thus reducing the responses. Though there was a reduction in the strain, the change was rather lower – this indicates that the load time duration may not have a profound effect on the ultimate stress/strain results.

The angular frequency was introduced to into the FE program and the analysis was re-run. The response in the analytical strains was similar to those found for the increases MET stress angle. The overall change in the stress/strain were again, rather small, however the load times were not consistent with those found experimentally as indicated previously.

These rather small changes in the analytical stress and strain computations indicate that overall the load time does not change the structural response by a large amount even though there is a change in the stiffness of each HMA layer. This may not be the case for thinner HMA layer pavements. In this case, the HMA layers are quite thick relative to other typical pavement sections. It is possible that the unit change in stiffness of the HMA layers in this case causes smaller unit change in the structural response than the same changes for thinner HMA layers. For thinner HMA layers, the same unit change in the load time may cause a much larger unit change in the structural response. This effect can be explained by simple beam theory and the stress equation due to bending (Equation 5-1).

$$\sigma = \frac{My}{I} \qquad \varepsilon = \frac{My}{IE} \qquad \textbf{(5-1)}$$

where σ = stress
ε = strain
M = moment
y = depth to neutral axis
I = moment of inertia
E = stiffness

Because of thickness increases, the moment of inertia will increase according to a power function. A unit change in stiffness for both thick and thin sections creates different unit changes in the computed strain. However, the pavement system contains much more stress interactions then a simple beam and this influence is unknown.

Other observations of the horizontal strain were investigated to confirm observations from the load time and FE analyses. In particular, observations were taken during review of the recorded strain responses and it was noted that in general the transverse strains were typically higher than the longitudinal strains for the same load. Comparison of the data concluded that the transverse strains were indeed higher than the respective longitudinal. During the analysis of the load pulse durations, it was also noted that the durations were shorter in length for the longitudinal direction due in part to the stress reversals. Since the transverse load pulse durations were longer than the longitudinal, it was expected that the strains may be higher due to a lower stiffness in the particular orientation. This supports a theory that the HMA materials may possess different stiffness characteristics for the different orientations under normal traffic loading (i.e. the HMA layers are anisotropic). This observation warrants further study.

5.4 - Fatigue Analysis

The fatigue of HMA materials has been studied closely in the past and has been found to be heavily dependent on the magnitudes of strain induced by the applied wheel loads. The behavior of the fatigue characteristics have been compared and modeled in a similar fashion to the fatigue of metals – increases in the stress magnitude of repeated loading causes a reduction in the fatigue life. This behavior which is well understood in metals has been applied to HMA materials (using strain instead of stress for fatigue life predictions). The complexity of the HMA material itself leads to the study of the material at a macroscopic level whereas fatigue of metals can typically be understood by the crystalline nature of the metal under study.

One particularly important factor in applying the fatigue theory to HMA is correctly modeling the loadings that are applied to the pavement system – both in distribution in load magnitude and also the distribution of the influence of those loads. The purpose of the analysis under study here was to understand the stress distribution of the loadings and develop a method to model these distributions. In the past, it has been common to assume the distribution of the stresses and strains in the pavement system using readily available layered elastic analysis or finite-element analysis. This distribution is important because the pavement will undergo stresses at some distances away from the load center and the cumulative nature of the fatigue life requires the incremental damage to accurately predict the fatigue life consumption. Drivers have a natural tendency to meander within their travel lane, thus adding another variation which must be accounted for.

In order to understand the strain distributions from individual wheel loads, a strain ratio was computed for each load sample. This ratio described the strains laterally across the wheelpath as a fraction of the maximum strain measured (which is found directly under the applied load). A good relationship was found between the strain ratios and the lateral distance away from the load center.

From the data, it was found that the relationship between the strain ratios and the lateral distance was not symmetrical, but was roughly parabolic. The asymmetry of the relationship is theorized to be caused by the wheel loads at the opposing end of the axle. This implies that even though a typical truck may have an axle width of over 90 inches, both wheels can cause a strain response at a significant distance away.

Single and dual wheels were analyzed separately as well as the distributions for both transverse and longitudinal orientations. It was found that for the longitudinal orientation there is a significant difference between the strain-distance relationships for the two load types which is likely due to the wider stance of most dual tires which can spread the load out further. Single tires create a much narrower strain-distance relationship with less influence from the opposing wheels.

In the case of the transverse strain-distance relationship, the data seems to be quite similar between both singles and tandems. This effect could be better understood by providing better resolution in the strain measurements. It is likely that a difference between duals and singles does exist given the geometry of the loads, but the strain measurements here do not provide enough data to make a justifiable conclusion.

Regression analysis of the strain ratio-distance data was conducted to generate models which can be applied to predict strains at any distance knowing only the

maximum tensile strain. In addition, symmetric models were created by using the data not affected by the opposing wheel load. This was done with the intent that individual loads could be superimposed for situations that are not defined by standard axle/load configurations. The linear regression models were found to fit reasonably well given the larger amount of scatter in the data. Different methods of data filtering were attempted to further improve the model fit, but ultimately the precision of the data is governed by the resolution designed into the experiment.

The predictors of the strain ratio were found to be almost solely dependent on the distance from the load. Other potential regressors were included in to the regression analyses, but were found to be insignificant via the regression statistics. Notably, speed and wheel load were two regressors that were included in these analyses due to their known effects on HMA structural response, but both were found to contribute little to the strain ratio prediction. It is possible that additional resolution in a similar experiment may show dependence.

5.5 - Recommendations

The research has raised additional questions which could be investigated to further improve the accuracy of fatigue life prediction and the nature of HMA pavement structural response. In regards to the time-temperature dependence of the HMA materials, there is evidence in support of a stiffness gradient through thickness of the HMA which needs to be better understood. Within the MET procedure to predict load times, the fastest durations occur in the upper layers with an increase at greater depths. It is proposed that this effect could be measured by placing numerous strain gauges through

the thickness of the HMA layers, spaced longitudinally at known distances. The load times could then be measured and compared to the stiffness gradient generated through MET. The analytical stiffness gradient could then be corrected by balancing the stiffnesses to produce a no-stress (neutral axis) of the HMA layers that matches the experimental data.

Since there is a stiffness gradient in the HMA and it has been shown here that the load time may have a rather small effect on the overall analytical response, it would be wise to instrument pavements with different HMA layer thicknesses, but with identical materials. A similar analysis that has been conducted in this research could be reapplied which would hopefully demonstrate the dependence on load times for different HMA layer thicknesses. Thinner HMA sectioned pavements may be more sensitive to changes in the stiffness. In addition, the overall effect of temperature can be studied for thinner HMA sectioned pavements. Temperature was shown for this pavement to have little effect on the load times (i.e colder temperatures should lead to stiffer HMA which should, theoretically, distribute the load further leading to higher load times).

This research experiment was limited to the analysis for bottom-up cracking since strain was only measured on the bottom of the HMA layer. Placing strain gauges near the surface can yield similar models to understand the phenomenon that occurs at the pavement surface, causing top-down cracking.

There is evidence to support the idea that the HMA layers have different stiffnesses for different orientations. A much more advanced analysis might prove beneficial for accurately predicting strain in the HMA layers and could be done by simply recalculating the stiffness for the three major orientations studied here (transverse,

longitudinal, and vertical). The vertical stress/strain load times in the HMA could not be developed for this research, because it was not measured in this experiment. However, inclusion of strain gauges to measure the vertical orientation would allow for this. It is cautioned that using stress cells (similar to the earth pressure cells used the unbound layers in this project) to do this may cause inaccuracies due to the lack of the ability to measure tensile events and because there is a lag between strain and stress due to the material's viscoelasticity (but could analytically be corrected).

The nature of the distribution of the stress and strain on the pavement due to different loads can be better understood by gaining better resolution in strain measurement. This can be done by simply applying more strain gauges across the width the wheelpath. This research used strain gauges on 12 inch centers – adding more (but spaced longitudinally) instruments can help provide this resolution. A more thorough analysis could be done by instrumenting the entire lane, or (more aggressively) the entire cross-section.

The data produced in this work relied heavily on software customized to extract information from the raw data. Development of this is time consuming and refinements can be made, but at a cost. Refinements in the software could vastly improve the amount of usable data and the precision, leading to higher quality.

Ultimately, more experimental data similar to this project is needed because it provides the structural data to validate the analytical models which are prevalent in the mechanistic-empirical design process.

BIBLIOGRAPHY

AASHTO. (1993). *AASHTO Guiide For Design of Pavement Structures.* Washinton, D.C.: American Association of State Highway and Transportation Officials.

AASHTO TP62: Standard Method of Test for Determining Dynamic Modulus of Hot-Mix Asphalt Concrete Mixtures. (2003). Washington, D.C.: American Association of State Highway and Transportation Officials.

Al-Qadi, I., Xie, W., & Elseifi, M. (2008). Freqeuncy Determination from Vehicular Loading Time Pulse to Predict Appropriate Complex Modulus in MEPDG. *Journal of Association of Asphalt Paving Techonologists* , 739-772.

Apogee Instruments Inc. (n.d.). *Silicon-cell Photodiode Pyranometers*. Retrieved 2007, from Apogee Instruments Inc.: http://www.apogeeinstruments.com

Asphalt Institute. (1982). Research and Development of the Asphalt Institute's Thickness Design Manual (MS-1), 9th edition. *Research Report 82-2*.

Asphalt Institute. (1991). *Thickness Design-Asphalt Pavements for Highways and Streets, Manual Series No. 1.* Asphalt Institute.

Baker, H., Buth, M., & Van Deusen, D. (2002). *Minnesota Road Research Project, Load Response Instrumentation Installation and Testing Procedures, Final Report.* Minneapolis: University of Minnesota.

Bao, W. (2000). *Calibration of Flexible Pavement Structural Model Using MnRoad Field Data, M.S. Thesis.* Minneapolis: University of Minnesota.

Barksdale, R. D. (1971). Compressive Stress Pulse Times In Flexible Pavements For Use In Dynamic Testing. *Highway Research Recordd, No. 345, Journal of Highway Research Board of the National Research Council,* pp. 32-44.

Bonnaure, F., Gravois, A., & Udron, J. (1980). A New Method of Predicting the Fatigue. *Journal of the Association of Asphalt Paving , 49,* 499-524.

Brown, E. C., Hanson, D. L., Powell, B., Prowell, B., & Watson, D. (2002). *NCAT Test Track Design, Construction, and Performance, NCAT Report 04-03.* Auburn University: National Center for Asphalt Technology.

Buiter, R., Cortenraad, A., Van Eck, A., & Van Rij, H. (1989). Effect of Transverse Distribution of Heavy Vehicles on Thickness Design of Full-Depth Asphalt Pavements. *Transportation Research Record, No. 1227, Journal of the Transportation Research Board of the National Academies, Washington D.C.*, pp. 66-74.

Carpenter, S. H., & Shen, S. (2006). Fatigue Characteristics of Rich Bottom Bases for Structural Design of Perpetual Pavements. *Proceedings for the International Conference on Perpetual Pavements, Columbus, OH.*

Carpenter, S. H., Ghuzlan, K. A., & Shen, S. (2003). Fatigue Endurance Limit for Highway and Airport Pavements. *Transportation Research Record, No. 1832, Journal of the Transportation Research Board of the National Academies, Washington D.C.*, pp. 131-138.

Cho, Y., McCullough, F., & Weissman, J. (1996). Considerations on Finite-Element Method Application in Pavement Structural Analysis. *Transportation Research Record, No. 1539, Journal of the Transportation Research Board of the National Academies, Washington D.C.*, pp. 96-101.

Crovetti, J. A., Titi, H., Coenen, A., Williams, C., Li, X., & R., M. E. (2008). *Materials Characterization and Analysis of The Marquette Interchange HMA Perpetual Pavement.* Madison: Midwest Regional University Transportation Center.

CTL Group. *Asphalt Strain Gauge ASG-152*. Retrieved 2006, from CTL Group: http://www.ctlgroup.com

Deacon, J. A. (1965). Fatigue of Asphalt Concrete, Ph.D. Dissertation. University of California, Berkely.

Decagon Devices Inc. *EC-5 Soil Moisture Sensor.* Retrieved 2006, from Decagon Devices Inc.: http://www.decagon.com/

Dongre, E., Myers, L., & D'Angelo, J. (2006). Conversion of Testing Frequency to Loading Time: Impact on Performance Predictions Obtained from the M-E Pavement Design Guide, paper

#06-2394. *85th Annual Meeting of the Transportation Research Board.* Washington, D.C.: Transportation Research Board of the National Academies.

Dynatest Inc. *Dynatest PAST II AC Strain Transducer*. Retrieved 2006, from Dynatest: http://www.dynatest.com

Electronique Controle Measure. *ECM-France*. Retrieved 2007, from http://www.ecm-france.com/gb/index.php

Freeman, R. B., Carr, H. T., McEwen, T. V., & Powell, R. B. (2001). *Instrumentation at the National Center for Asphalt Technology Test Track.* US Army Corps of Engineers, ERDC TR-01-9.

Geokon Inc. *Model 3500 Earth Pressure Cell*. Retrieved 2006, from Geokon Inc.: http://www.geokon.com/

Geokon, Inc. (2004). *Model 3500/3510 Earth Pressure Cell Instruction Manual.* Lebanon, NH.

Ghuzlan, K. A. (2001). *Fatigue Damage Analysis In Asphalt Concrete Mixtures Based Upon Dissipated Energy Concepts, Ph.D. Dissertation.* University of Illinois at Urbana-Champaign.

Ghuzlan, K. A., & Carpenter, S. H. (2000). Energy-Derived, Damage-Based Failure Criterion for Fatigue Testing. *Transportation Research Record, No. 1723, Journal of the Transportation Research Board of the National Academies, Washington D.C.*, pp. 141-149.

Highway Research Board of the National Research Council, Division of Engineering and Industrial Research. (1961). *Special Report: The AASHO Road Test, No. 816.* pp. 951-955.

Hornyak, N. J., & Crovetti, J. A. (2009). Analysis of Load Pulse Durations for the Marquette Interchange Project. *Transportation Research Record, No. 2094, Journal of the Transportation Research Board of the National Academies, Washington D.C.*, pp. 53-61.

Hornyak, N. J., & Crovetti, J. A. (2008). *Perpetual Pavement Instrumentation Project Phase II - WHRP 08-04.* Madison, WI: Wisconsin Highway Research Program.

Hornyak, N. J., Crovetti, J. A., Newman, D. E., & Schabelski, J. (2007). Asphalt Pavement Instrumentations: The Quest for Truth, paper #07-3050. *Proceedings of The 86th Annual Meeting of the Transportation Research Board of the National Academies.* Washington, D.C.

Hornyak, N. J., Crovetti, J. A., Newman, D. E., & Schabelski, J. P. (2007). *Perpetual Pavement Instrumentation for the Marquette Interchange Project Phase I - WHRP 07-11.* Madison, WI: Wisconsin Highway Research Program.

Huang, Y. H. (1993). *Pavement Analysis and Design.* Upper Saddle River, New Jersey, USA: Prentice Hall, Inc.

Kistler. *Lineas Quartz Sensor for Weigh-In-Motion*. Retrieved 2007, from Kistler Instrument Co.: http://www.kistler.com/

Koubaa, A., & Stolarski, H. (2002). *Assistance In the Validation of the MnRoads Database, Final Report.* Minneapolis: University of Minnesota, Department of Civil Engineering.

Kwon, Y. W., & Bang, H. (1997). *The Finite Element Method.* Boca Raton: CRC Press.

Lau, W., & Alouini, M. (2002). *Development of Automated Procedures for Dynamic MegaDAC Calibration and Data Recovery, Final Report.* Minneapolis: University of Minnesota, Department of Civil Engineering.

Lemaitre, J., & Desmorat, R. (2005). *Engineering Damage Mechanics.* The Netherlands: Springer-Verlag.

Miner, M. A. (1945). Cumulative Damage in Fatigue. *Journal of Applied Mechanics, Vol. 12, Transactions of the ASME, Vol. 67,* pp. A159-A164.

Monismith, C. L., & Deacon, J. A. (1969). Fatigue of Asphalt Paving Mixtures. ASCE Transportation Engineering Journal, Vol. *95:s,* pp. 317-346.

Murthy, V. (2003). *Geotechnical Engineering, Principles and Practices of Soil Mechanics and Foundation Engineering.* New York: Marcel Dekker, Inc.

MySQL/Sun Microsystems, Inc. (n.d.). *MySQL*. Retrieved 2008, from http://www.mysql.com/

National Cooperative Highway Research Program. (2004). *Guide for Mechanistic-Empirical Design of New and Rehabilitated Pavement Structures.* Washington D.C.: National Research Council.

Never-Fail Loop Systems Inc. *Never-Fail Inductive Loop Systems*. Retrieved 2007, from Never-Fail Loop Systems: http://www.neverfail.com/

Odemark, N. (1949). Investigation as to the Elastic Properties of Soils and the Design of Pavements According to the Theory of Elasticity. *National Swedish Road and Traffic Research Institute, Report 77* .

Pfeiffer, D. A. (2006). Ike's Interstate at 50 - Anniversary of the Highway System Recalls Eisenhower's Role as Catalyst. *The U.S. National Archives and Records Administration, Vol. 38, No. 2.*

Priest, A. (2005). Calibration of Fatigue Transfer Functions for Mechanistic-Empirical Flexible Pavement Design. *M.S. Thesis*. Auburn, AL: Auburn University.

Priest, A., & Timm, D. (2006). *Methodology and Calibration of Fatigue Transfer Functions for Mechanistic-Empirical Flexible Pavement Design, NCAT Report 06-03.* Auburn University: National Center for Asphalt Technology.

Prowell, B. D., & Brown, R. E. (2006). Methods for Determining The Endurance Limit Using Beam Fatigue Tests. *Proceedings of The International Conference on Perpetual Pavements, Columbus, OH*.

Romanoschi, S., Gisi, A., & Dumitru, C. (2006). The Dynamic Response of Kansas Perpetual Pavements Under Vehicle Loading. *Proceedings of The International Conference on Perpetual Pavements, Columbus, OH*.

Romanoschi, S., Gisi, A., Portillo, M., & Dumitru, C. (2008). The First Findings from the Kansas Perpetual Pavements Experiment. *Transportation Research Record, No. 2068, Journal of the Transportation Research Board of the National Academies, Washington D.C.*, pp. 41-48.

Romus Inc. *Romus Incorporated*. Retrieved 2006, from http://www.romusinc.com

Shell. (1978). *Shell Pavement Design Manual.* London: Shell International Petroleum.

Shen, S., & Carpenter, S. H. (2005). Application of the Dissipated Energy Concept in Fatigue Endurance Limit Testing. *Transportation Research Record, No. 1929, Journal of the Transportation Research Board of the National Academies, Washington D.C.*, pp. 165-173.

Strommen, R. (2002). *Dynamic MegaDAC Calibration Procedure and Development Manual, Version 1.1.* Minneapolis: University of Minnesota.

Tayebali, A. A., Deacon, J. A., & Monismith, C. L. (1996). Development and Evaluation of Dynamic Flexural Beam Fatigue Test System. *Transportation Research Record, No. 1545, Journal of the Transportation Research Board of the National Academies, Washington D.C.*, pp. 89-97.

Thompson, M., Carpenter, S., Dempsy, B., & Elliot, R. (2006). Independent Review of the Recommened Mechanistic-Empirical Design Guide and Software. *National Cooperative Highway Research Program, Report 1-40A* .

Timm, D. H., & Priest, A. L. (2004). *Dynamic Pavement Response Data Collection and Processing at The NCAT Test Track, NCAT Report 04-03.* Auburn University: National Center for Asphalt Technology.

Timm, D. H., Priest, A. L., & McEwen, T. V. (2004). *Design and Instrumentation of the Structural Pavement Experiment at the NCAT Test Track, NCAT Report 04-01.* Auburn University: National Center for Asphalt Technology.

Timm, D., & Priest, A. (2006). *Material Properties of the 2003 Test Track Structural Study, NCAT Report 06-01.* Auburn University: National Center for Asphalt Technology.

Timm, D., & Priest, A. (2005). *Wheel Wander at the NCAT Test Track, NCAT Report 05-02.* Auburn University: National Center for Asphalt Technology.

Timm, D., West, R., Priest, A., Powell, B., Selvaraj, I., Zhang, J., et al. (2006). *Phase II NCAT Test Track Results, NCAT Report 06-05.* Auburn University: National Center for Asphalt Technology.

U.S. Department of Transporation Federal Highway Administration. (2004). *Federal Size Regulations For Commercial Motor Vehicles.* Washington, DC: U.S. Department of Transporation Federal Highway Administration.

Von Quintus, H. L. (2006). Application of the Endurance Limit Premise in Mechanistic-Empirical Based Pavement Design Procedures. *Proceedings of The International Conference on Perpetual Pavements, Columbus, OH.*

Witczak, M. W., Zapata, C., & El-Basyouny, M. M. (2003). *Input Data for the Calibration and Validation of the 2002 Design Guide for New Constructed Flexible Pavement Sections.* Development of the 2002 Guide for the Design of New and Rehabilitated Pavement Structures – Technical Report. National Cooperative Highway Research Program (NCHRP), Project 1-37A. Federal Highway Administration.

Witczak, M. W., Zapata, C., & El-Basyouny, M. M. (2003). *Input Data for the Calibration and Validation of the 2002 Design Guide for Rehabilitated Pavement Sections with HMA Overlays.* Development of the 2002 Guide for the Design of New and Rehabilitated Pavement Structures – Technical Report. National Cooperative Highway Research Program (NCHRP), Project 1-37A. Federal Highway Administration.

Worel, B. (2006). *MnROAD Database Guide.* Maplewood: Minnesota Department of Transporation - MnROAD Operations.

Appendix A – C# Code

The following contains the code used to conduct the analysis for this project. The program allows a user a take weigh-in-motion and environmental data and conduct a finite-element analysis for each measured vehicle. The program relies heavily on communication with a MySQL server for the storage of results. The popular MySQL community server and .NET connector (both open source) are available for download from www.mysql.com.

The finite element engine used in this analysis program was adapted from MATLAB code written by Kwon and Bang (1997). The C# language does not include a base class for conducting linear algebra, however the .NET class library Mapack written by Lutz Roeder was used. The Mapack .NET package is available from http://www.lutzroeder.com/dotnet/. In addition, a sparse matrix solver was used to increase the speed of the linear algebra inversion solution. This open source .NET library package is available from dnAnalytics and can be found at http://dnanalytics.codeplex.com/.

The following code can be made availble in electronic form upon request from the author.

Class: Analysis

```
using System;
using System.Collections.Generic;
using System.Collections;
using System.Text;
using Mapack;

namespace MPave
{
    class Analysis
    {
        //  Sampling convention.  1 = (-1,-1);   2 = (-1, 1)
        //   _________            3 = ( 1,-1);   4 = ( 1, 1)
        //  | 2  |  4 |
        //  |____|____|
        //  | 1  |  3 |
        //  |____|____|

        //Linear elastic axisymmetric FE pavement analysis using quad node
        //isoparametric elements.  All layers fully bonded.  Adapted from MATLAB
        //program.
        //Last updated 01/08/2008

        //Hardcoded model data
        int nElements;                              //# of elements in system
        int numNodesElement = 4;                    //# of nodes per element
        int DOF = 2;                                //# of degrees of freedom per node
        int nNodes;                                 //# of nodes in system
        int sDOF;                                   //System degrees of freedom
        int eDOF;                                   //Degrees of freedom per element
        int nGLX = 2;                 //2 X 2 Gaussian-Legendre quadrature (numerical
        int nGLY = 2;                 //integration technique)
        int nGLXY;// = nGLX * nGLY;   //Number of sampling points per
                                      //element (quadrature)

        Matrix rElementStress, rElementStrain,
            tElementStress, tElementStrain,     //Global matrices for
            vElementStress, vElementStrain;     //storing solutions

        double contactRadius, wheelLoad;
        double[] moduli, poissonsRatio;

        //Constructor
        public Analysis(double contactRadius, double wheelLoad, double[] moduli,
            double[] poissonsRatio)
        {
            this.contactRadius = contactRadius;
            this.wheelLoad = wheelLoad * 1000;// 000 - Wheel load amplified to
            this.moduli = moduli;             //prevent small number computations
            this.poissonsRatio = poissonsRatio;
        }

        public void StartAnalysis()
        {
            //
            //For Debugging
            //

            //Console.WriteLine("CR: " + contactRadius.ToString() + "\tWL: " +
            //    wheelLoad.ToString());
            //for (int j = 0; j < moduli.Length; j++)
            //{
            //    Console.WriteLine(moduli[j].ToString());
            //}
            //Console.ReadLine();

            //
            //Initialize mesh and material variables and matrices
            //
```

```
ModelConstants modelConstants = new ModelConstants();
nElements = modelConstants.NumElements;
nNodes = modelConstants.NumNodes;
int[] xValues = modelConstants.XValues;
int[] yValues = modelConstants.YValues;


sDOF = nNodes * DOF;
eDOF = numNodesElement * DOF;
nGLXY = nGLX * nGLY;

rElementStress = new Matrix(nElements, 4);
rElementStrain = new Matrix(nElements, 4);
tElementStress = new Matrix(nElements, 4);
tElementStrain = new Matrix(nElements, 4);
vElementStress = new Matrix(nElements, 4);
vElementStrain = new Matrix(nElements, 4);

//
//Generate Mesh and node connectivity
//

Mesh newMesh = new Mesh(xValues, yValues);

int[,] gCoord = newMesh.CreateModelNodeCoord();
int[,] nodes = newMesh.CreateModelConnectivity();

//Set boundary conditions
BoundaryConditions newBC = new BoundaryConditions(xValues,
    yValues, gCoord);

int[] bCDOF = newBC.ConstrainedDOF;
int[] valuesDOF = newBC.ValuesDOF;

//
//Set force vector
//
ForceVector fVector = new ForceVector(sDOF, contactRadius, wheelLoad,
    yValues, xValues);
double[] forceVector = fVector.GenerateForceVector();

Matrix points2D, weights2D;
FEGLQ2D samplingPointsWeights = new FEGLQ2D(nGLX, nGLY);
points2D = samplingPointsWeights.CreateIntegrationPoints2D();
weights2D = samplingPointsWeights.CreateIntegrationWeight2D();

//
//Initialize material matrices
//
Matrix materialElementCode =
    modelConstants.MaterialCodes;//Matrix for material code pattern
                                 //defined below
ArrayList materialMatrices = new ArrayList();
for (int m = 0; m < moduli.Length; m++)
{
    FEMatIso matMatrixIso = new FEMatIso(moduli[m], poissonsRatio[m]);
    Matrix matMatrix = matMatrixIso.AssembleMaterialMatrix();
    materialMatrices.Add(matMatrix);
}


//
//Compute element matrices and assemble
//

int[] nodalData = new int[4];
double[] xCoord = new double[4];
double[] yCoord = new double[4];

int kkSize = nNodes * 2;
```

```
Matrix kk = new Matrix(kkSize, kkSize);

//Beginning first integration loop...
for (int iElement = 0;
    iElement < nElements; iElement++) //Loop for total # of elements
{
    for (int j = 0; j < numNodesElement; j++)
    {
        nodalData[j] = nodes[iElement, j];
        xCoord[j] = (double)gCoord[nodalData[j], 0];
        yCoord[j] = (double)gCoord[nodalData[j], 1];
    }

    Matrix k = new Matrix(eDOF, eDOF);

    //
    //Begin numerical integration
    //

    for (int intX = 0; intX < nGLX; intX++)
    {

        double xSample = points2D[intX, 0];
        double weightX = weights2D[intX, 0];

        for (int intY = 0; intY < nGLY; intY++)
        {
            double ySample = points2D[intY, 1];
            double weightY = weights2D[intY, 1];

            //Compute shape function
            FEIsoQ4 feIsoq = new FEIsoQ4(xSample, ySample);
            double[] shape, dhdr, dhds;

            //Sampling points for integration
            shape = feIsoq.ComputeShapeFunction();
            dhdr = feIsoq.Computedhdrq4();
            dhds = feIsoq.Computedhdsq4();

            //Compute Jacobian - calculate determinate and inverse
            FEJacob2 jacob2 = new FEJacob2(numNodesElement, xCoord,
                yCoord, dhdr, dhds);
            Matrix jacobian = jacob2.ComputeJacobian2D();
            double detJacobian = jacobian.Determinant;
            Matrix invJacobian = jacobian.Inverse;

            FEDeriv2 derivative = new FEDeriv2(numNodesElement, dhdr,
                dhds, invJacobian);

            double[] dhdx = derivative.dhdxShapeFunction();
            double[] dhdy = derivative.dhdyShapeFunction();

            double xCenter = 0;
            for (int i = 0; i < numNodesElement; i++)
            {
                xCenter = xCenter + shape[i] * xCoord[i];
            }

            Matrix kinematicMatrix;
            FEKineAx kinematicAxi = new FEKineAx(numNodesElement, dhdx,
                dhdy, shape, xCenter);
            kinematicMatrix = kinematicAxi.CreateKinematicMatrix();


            //Compute element matrix

            double pi = Math.PI;
            Matrix kinematicTransposed = kinematicMatrix.Transpose();

            int materialCode = (int)materialElementCode[iElement, 1];
```

```
            k = k + (kinematicTransposed * 2 * pi * xCenter) *
                (Matrix)materialMatrices[materialCode - 1] *
                kinematicMatrix * weightX * weightY * detJacobian;
        }
    }

    FEElementDOF elementDOF = new FEElementDOF(nodalData,
        numNodesElement, DOF);
    int[] index = elementDOF.IndexDOF();

    FEAssemble1 newAssembly = new FEAssemble1(kk, k, index);
    kk = newAssembly.Assemble();
}
//Console.WriteLine("End of first integration loop...");

//Apply boundary conditions
FEApplyConstraints bConditions = new FEApplyConstraints(kk, forceVector,
    bCDOF, valuesDOF);
bConditions.ApplyConstraints();
kk = bConditions.getkk;
forceVector = bConditions.getff;

//
//Solve matrix equation
//
Matrix dispMatrix;

Matrix forceVectorMatrix = new Matrix(forceVector.Length, 1);

//Re-map vectors and matrix for solver
for (int n = 0; n < forceVector.Length; n++)
{
    forceVectorMatrix[n, 0] = forceVector[n];
}

double[,] matrixDouble = new double[kk.Rows, kk.Columns];
for (int n = 0; n < kk.Rows; n++)
{
    for (int m = 0; m < kk.Columns; m++)
    {
        matrixDouble[n, m] = kk[n, m];
    }
}

//
//Solve matrix equation...
//

//MAPACK Inversion - TOO SLOW!
//dispMatrix = kk.Inverse * forceVectorMatrix;


//Mapack Solver - better
//dispMatrix = kk.Solve(forceVectorMatrix);

//
//******************dnAnalytics - FASTEST SOLVER*********************
//
SparseSolver newSolver = new SparseSolver();
double[] dispArray = newSolver.InvertSolver(matrixDouble, forceVector);

//Re-map force vector for matrix algebra
dispMatrix = new Matrix(dispArray.Length, 1);
for (int n = 0; n < dispArray.Length; n++)
{
    dispMatrix[n, 0] = dispArray[n];
}


//
//Element stress computation
```

```
//

Matrix stress = new Matrix(4, 4); //Matrices for storing element
Matrix strain = new Matrix(4, 4); //results at integration points.

for (int iElement = 0; iElement < nElements; iElement++)
{
    int intp = 0;

    for (int j = 0; j < numNodesElement; j++)
    {
        nodalData[j] = nodes[iElement, j];
        xCoord[j] = gCoord[nodalData[j], 0];
        yCoord[j] = gCoord[nodalData[j], 1];
    }

    for (int intX = 0; intX < nGLX; intX++)
    {
        double xSample = points2D[intX, 0];
        double weightX = weights2D[intX, 0];

        for (int intY = 0; intY < nGLY; intY++)
        {
            double ySample = points2D[intY, 1];
            double weightY = weights2D[intY, 1];
            //intp++;

            //Compute shape function
            FEIsoQ4 feIsoq = new FEIsoQ4(xSample, ySample);
            double[] shape, dhdr, dhds;

            //Sampling points for integration
            shape = feIsoq.ComputeShapeFunction();
            dhdr = feIsoq.Computedhdrq4();
            dhds = feIsoq.Computedhdsq4();

            //Compute Jacobian - calculate determinate and inverse
            FEJacob2 jacob2 = new FEJacob2(numNodesElement, xCoord,
                yCoord, dhdr, dhds);
            Matrix jacobian = jacob2.ComputeJacobian2D();

            double detJacobian = jacobian.Determinant;
            Matrix invJacobian = jacobian.Inverse;

            FEDeriv2 derivative = new FEDeriv2(numNodesElement, dhdr,
                dhds, invJacobian);

            double[] dhdx = derivative.dhdxShapeFunction();
            double[] dhdy = derivative.dhdyShapeFunction();

            double xCenter = 0;
            for (int i = 0; i < numNodesElement; i++)
            {
                xCenter = xCenter + shape[i] * xCoord[i];
            }

            Matrix kinematicMatrix;
            FEKineAx kinematicAxi = new FEKineAx(numNodesElement, dhdx,
                dhdy, shape, xCenter);
            kinematicMatrix = kinematicAxi.CreateKinematicMatrix();

            FEElementDOF elementDOF = new FEElementDOF(nodalData,
                numNodesElement, DOF);
            int[] index = elementDOF.IndexDOF();


            //Extract element displacement vector
            Matrix elementDisplacement = new Matrix(eDOF, 1);
            for (int i = 0; i < eDOF; i++)
            {
                elementDisplacement[i,0] = dispMatrix[index[i],0];
```

```
            }

            Matrix eStrain = kinematicMatrix * elementDisplacement;
            double[] eStraind = new double[eStrain.Rows];
            for (int n=0;n<eStrain.Rows;n++) //Map strain to simple array
            {
                eStraind[n] = eStrain[n, 0];
            }

            Matrix eStress = new Matrix(0, 0);

            //Compute element stress
            int materialCode = (int)materialElementCode[iElement, 1];

            eStress =
                (Matrix)materialMatrices[materialCode - 1] * eStrain;

            //Map stress to simple array
            double[] eStressd = new double[4];
            for (int n=0;n<eStress.Rows;n++)
            {
                eStressd[n] = eStress[n,0];
            }

            for (int i = 0; i < 4; i++)
            {
                strain[intp, i] = eStraind[i];
                stress[intp, i] = eStressd[i];
            }
            intp++;

        }
    } //End of integration loops

    double factor1 = 1.8660254d;
    double factor2 = 0.13439746d;

    //Computation into radial stresses and strains
    rElementStress[iElement, 0] = factor1 * stress[0, 0] - 0.5 *
        stress[1, 0] - 0.5 * stress[2, 0] + factor2 * stress[3, 0];
    rElementStress[iElement, 1] = factor1 * stress[1, 0] - 0.5 *
        stress[0, 0] - 0.5 * stress[3, 0] + factor2 * stress[2, 0];
    rElementStress[iElement, 2] = factor1 * stress[2, 0] - 0.5 *
        stress[0, 0] - 0.5 * stress[3, 0] + factor2 * stress[1, 0];
    rElementStress[iElement, 3] = factor1 * stress[3, 0] - 0.5 *
        stress[1, 0] - 0.5 * stress[2, 0] + factor2 * stress[0, 0];

    rElementStrain[iElement, 0] = factor1 * strain[0, 0] - 0.5 *
        strain[1, 0] - 0.5 * strain[2, 0] + factor2 * strain[3, 0];
    rElementStrain[iElement, 1] = factor1 * strain[1, 0] - 0.5 *
        strain[0, 0] - 0.5 * strain[3, 0] + factor2 * strain[2, 0];
    rElementStrain[iElement, 2] = factor1 * strain[2, 0] - 0.5 *
        strain[0, 0] - 0.5 * strain[3, 0] + factor2 * strain[1, 0];
    rElementStrain[iElement, 3] = factor1 * strain[3, 0] - 0.5 *
        strain[1, 0] - 0.5 * strain[2, 0] + factor2 * strain[0, 0];

    //Computation into tangential stresses and strains
    //(perpendicular to plane of model)
    tElementStress[iElement, 0] = factor1 * stress[0, 1] - 0.5 *
        stress[1, 1] - 0.5 * stress[2, 1] + factor2 * stress[3, 1];
    tElementStress[iElement, 1] = factor1 * stress[1, 1] - 0.5 *
        stress[0, 1] - 0.5 * stress[3, 1] + factor2 * stress[2, 1];
    tElementStress[iElement, 2] = factor1 * stress[2, 1] - 0.5 *
        stress[0, 1] - 0.5 * stress[3, 1] + factor2 * stress[1, 1];
    tElementStress[iElement, 3] = factor1 * stress[3, 1] - 0.5 *
        stress[1, 1] - 0.5 * stress[2, 1] + factor2 * stress[0, 1];

    tElementStrain[iElement, 0] = factor1 * strain[0, 1] - 0.5 *
        strain[1, 1] - 0.5 * strain[2, 1] + factor2 * strain[3, 1];
    tElementStrain[iElement, 1] = factor1 * strain[1, 1] - 0.5 *
        strain[0, 1] - 0.5 * strain[3, 1] + factor2 * strain[2, 1];
```

```
            tElementStrain[iElement, 2] = factor1 * strain[2, 1] - 0.5 *
                strain[0, 1] - 0.5 * strain[3, 1] + factor2 * strain[1, 1];
            tElementStrain[iElement, 3] = factor1 * strain[3, 1] - 0.5 *
                strain[1, 1] - 0.5 * strain[2, 1] + factor2 * strain[0, 1];

            //Computation into vertical stresses and strains
            vElementStress[iElement, 0] = factor1 * stress[0, 2] - 0.5 *
                stress[1, 2] - 0.5 * stress[2, 2] + factor2 * stress[3, 2];
            vElementStress[iElement, 1] = factor1 * stress[1, 2] - 0.5 *
                stress[0, 2] - 0.5 * stress[3, 2] + factor2 * stress[2, 2];
            vElementStress[iElement, 2] = factor1 * stress[2, 2] - 0.5 *
                stress[0, 2] - 0.5 * stress[3, 2] + factor2 * stress[1, 2];
            vElementStress[iElement, 3] = factor1 * stress[3, 2] - 0.5 *
                stress[1, 2] - 0.5 * stress[2, 2] + factor2 * stress[0, 2];

            vElementStrain[iElement, 0] = factor1 * strain[0, 2] - 0.5 *
                strain[1, 2] - 0.5 * strain[2, 2] + factor2 * strain[3, 2];
            vElementStrain[iElement, 1] = factor1 * strain[1, 2] - 0.5 *
                strain[0, 2] - 0.5 * strain[3, 2] + factor2 * strain[2, 2];
            vElementStrain[iElement, 2] = factor1 * strain[2, 2] - 0.5 *
                strain[0, 2] - 0.5 * strain[3, 2] + factor2 * strain[1, 2];
            vElementStrain[iElement, 3] = factor1 * strain[3, 2] - 0.5 *
                strain[1, 2] - 0.5 * strain[2, 2] + factor2 * strain[0, 2];
        }
    }

    //
    //Properties
    //

    public Matrix RadialStrain
    {
        get
        {
            return rElementStrain;
        }
    }

    public Matrix RadialStress
    {
        get
        {
            return rElementStress;
        }
    }

    public Matrix TangentialStrain
    {
        get
        {
            return tElementStrain;
        }
    }

    public Matrix TangentialStress
    {
        get
        {
            return tElementStress;
        }
    }

    public Matrix VerticalStrain
    {
        get
        {
            return vElementStrain;
        }
    }

    public Matrix VerticalStress
```

```
{
    get
    {
        return vElementStress;
    }
}

/// <summary>
/// Returns the radial strain at the bottom of the
/// HMA layer in units of microstrain.
/// </summary>
/// <returns></returns>
public double[] GetBottomHMARadialStrain()
{
    double[] radialStrain = new double[19];
    int element = 25;

    for (int m = 0; m < 19; m++)
    {
        if (element == 25)//16
        {
            radialStrain[m] =
                rElementStrain[element, 0] * 1 * 2*1000;
            //Console.WriteLine("Initial: " +
            //    radialStrain[m].ToString());
        }
        if (element == 1203)//538
        {                                                   //29
            radialStrain[m] =
                rElementStrain[(element - 38), 2] * 1*1000;
            //Console.WriteLine("Last: " +
            //    radialStrain[m].ToString());
        }
        if(element != 1203 && element != 25)//538, 16
        {                                                     //29
            radialStrain[m] =
                (rElementStrain[(element - 38), 2] +
                rElementStrain[element, 0] / 2) * 1*1000;
            //Console.WriteLine("Intermediate: " +
            //    radialStrain[m].ToString());
        }
        element += 38;//29
    }

    return radialStrain;
}

/// <summary>
/// Returns the tangential strain at the bottom of the HMA
/// layer in units of microstrain.
/// </summary>
/// <returns></returns>
public double[] GetBottomHMATangentialStrain()
{
    double[] tangentialStrain = new double[19];
    int element = 25;

    for (int m = 0; m < 19; m++)
    {
        if (element == 25)//16
        {
            tangentialStrain[m] =
                tElementStrain[element, 0] * 1 * 2 * 1000;
        }
        if (element == 1203)//538
        {                                                   //29
            tangentialStrain[m] =
                tElementStrain[(element - 38), 2] * 1 * 1000;
        }
        if (element != 1203 && element != 25)//538, 16
        {                                                     //29
```

```
            tangentialStrain[m] =
                (tElementStrain[(element - 38), 2] +
                tElementStrain[element, 0] / 2) * 1 * 1000;
        }
        element += 38;//29
    }

    return tangentialStrain;
}

/// <summary>
/// Returns the vertical pressure at the top of the native
/// soil layer in units of psi.
/// </summary>
/// <returns></returns>
public double[] GetVerticalPressure()
{
    double[] verticalPressure = new double[19];
    int element = 11;

    for (int m = 0; m < 19; m++)
    {
        if (element == 11)
        {
            verticalPressure[m] =
                vElementStress[element, 1] / 1000 * 2;
            //Console.WriteLine("Initial: "
            //    + radialStrain[m].ToString());
        }
        if (element == 1189)
        {
            verticalPressure[m] =
                vElementStress[(element - 38), 3] / 1000;
            //Console.WriteLine("Last: " +
            //    radialStrain[m].ToString());
        }
        if (element != 1189 && element != 11)
        {
            verticalPressure[m] =
                (vElementStress[(element - 38), 3] +
                vElementStress[element, 1] / 2) / 1000;
            //Console.WriteLine("Ping: " +
            //    radialStrain[m].ToString());
        }
        element += 38;
    }

    return verticalPressure;
}



/// <summary>
/// Returns the vertical pressure at the location of
/// the upper pressure plate (dense layer) in units of psi.
/// </summary>
/// <returns></returns>
public double[] GetBaseVerticalPressure()
{
    double[] verticalPressure = new double[19];
    int element = 19;

    for (int m = 0; m < 19; m++)
    {
        if (element == 19)
        {
            verticalPressure[m] =
                vElementStress[element, 1] / 1000 * 2;
            //Console.WriteLine("Initial: " +
            //    radialStrain[m].ToString());
        }
```

```
                if (element == 1197)
                {
                    verticalPressure[m] =
                        vElementStress[(element - 38), 3] / 1000;
                    //Console.WriteLine("Last: " +
                    //    radialStrain[m].ToString());
                }
                if (element != 1197 && element != 19)
                {
                    verticalPressure[m] =
                        (vElementStress[(element - 38), 3] +
                        vElementStress[element, 1] / 2) / 1000;
                    //Console.WriteLine("Intermediate: " +
                    //    radialStrain[m].ToString());
                }
                element += 38;
            }

            return verticalPressure;
        }
    }
}
```

Class: BoundaryConditions

```
using System;
using System.Collections.Generic;
using System.Text;

namespace MPave
{
    class BoundaryConditions
    {
        int[] xValues;
        int[] yValues;
        int[] constrainedDOF;
        int[] valuesDOF;
        int[,] gCoords;


        public BoundaryConditions(int[] xValues, int[] yValues, int[,] gCoords)
        {
            this.xValues = xValues;
            this.yValues = yValues;
            this.gCoords = gCoords;

            SetBoundaryConditions();
        }

        private void SetBoundaryConditions()
        {
            int sizeDOF = (xValues.Length + yValues.Length-1) * 2;

            constrainedDOF = new int[sizeDOF];

            int index = 0;

            //
            //For debugging
            //

            //for (int i = 0; i < sizeDOF; i++)
            //{
            //    Console.Write(constrainedDOF[i].ToString() + ", ");
            //}
            //Console.ReadLine();

            //for (int i = 0; i < sizeDOF; i++)
            //{
            //    Console.Write(valuesDOF[i].ToString() + ", ");
            //}
            //Console.ReadLine();



            //Builds the constrained DOF array for bottom row.
            for (int i = 0; i < gCoords.Length / 2; i++)
            {
                if (gCoords[i, 1] == 0 && gCoords[i, 0] !=
                    xValues[xValues.Length - 1])
                {
                    constrainedDOF[index] = i * 2;
                    constrainedDOF[index + 1] = i * 2 + 1;
                    index = index + 2;
                }
            }

            //Build the constrained DOF array for the right side.
            bool firstRow = true;
            for (int i = 0; i < gCoords.Length / 2; i++)
            {
                if (gCoords[i, 0] == xValues[xValues.Length - 1])
                {
```

```
                    //************For fixed right side**********************
                    //constrainedDOF[index] = i * 2;
                    //constrainedDOF[index + 1] = i * 2 + 1;
                    //*****************************************************

                    //************Fixed corner*****************************
                    if (firstRow)
                    {
                        constrainedDOF[index + 1] =
                            i * 2 + 1;//Fix bottom right corner
                        firstRow = false;
                    }
                    //*****************************************************

                    index = index + 2;
                }
            }

            valuesDOF = new int[sizeDOF];

            for (int j = 0; j < sizeDOF; j++)
            {
                valuesDOF[j] = 0;
            }

            //
            //For debugging
            //

            //for (int i = 0; i < sizeDOF; i++)
            //{
            //    Console.Write(constrainedDOF[i].ToString() + ", ");
            //}
            //Console.ReadLine();

            //for (int i = 0; i < sizeDOF; i++)
            //{
            //    Console.Write(valuesDOF[i].ToString() + ", ");
            //}
            //Console.ReadLine();
        }

        public int[] ConstrainedDOF
        {
            get { return constrainedDOF; }
        }

        public int[] ValuesDOF
        {
            get { return valuesDOF; }
        }

    }
}
```

Class: DynamicModulus

```
using System;
using System.Collections.Generic;
using System.Text;

namespace MPave
{

    //Constructor
    class DynamicModulus
    {

        //Dyanmic modulus material model parameters
        double delta;
        double alpha;
        double beta;
        double gamma;
        double c;
        double A;
        double VTS;

        public DynamicModulus(double delta, double alpha, double beta,
            double gamma, double c, double A, double VTS)
        {
            this.delta = delta;
            this.alpha = alpha;
            this.beta = beta;
            this.gamma = gamma;
            this.c = c;
            this.A = A;
            this.VTS = VTS;
        }

        //Reference viscosity
        private double ReferenceViscosity()
        {
            double temperatureReferenceRankine = 70d + 459.67d;
            double viscosityTrLogLog =
                A + VTS * Math.Log10(temperatureReferenceRankine);
            double viscosityTr = Math.Pow(10d, Math.Pow(10d, viscosityTrLogLog));
            //Console.WriteLine("Reference Viscosity = " + viscosityTr.ToString());
            return viscosityTr;//cP
        }

        //Viscosity
        private double Viscosity(double temperatureFahrenheit)
        {
            double temperatureRankine = temperatureFahrenheit + 459.67d;
            double viscosityLogLog = A + VTS * Math.Log10(temperatureRankine);
            double viscosity = Math.Pow(10d, Math.Pow(10d, viscosityLogLog));
            //Console.WriteLine("Viscosity = " + viscosity.ToString());
            return viscosity;//cP
        }

        //Intermediate calculation of log_Tr parameter
        private double logtr(double temperatureFahrenheit, double timeOfLoading)
        {
            double viscosity = Viscosity(temperatureFahrenheit);
            double viscosityTr = ReferenceViscosity();

            double logLoadTimeRefTemp = Math.Log10(timeOfLoading) - c *
                (Math.Log10(viscosity) - Math.Log10(viscosityTr));

            return logLoadTimeRefTemp;
        }

        /// <summary>
        /// Returns the stiffness, in psi, based on material input data.
        /// </summary>
```

```
        /// <param name="temperatureFahrenheit">The temperature of interest
        /// in degrees F.</param>
        /// <param name="timeOfLoading">The time of loading, in seconds, at
        /// the temperature of interest.</param>
        /// <returns></returns>
        public double CalculateStiffness(double temperatureFahrenheit,
            double timeOfLoading)
        {
            //For debugging/checking
            //Console.WriteLine("temperature = " + temperatureFahrenheit.ToString());
            //Console.WriteLine("time of loading = " + timeOfLoading.ToString());
            //Console.ReadLine();

            double logLoadTimeRefTemp =
                logtr(temperatureFahrenheit,          //additional converts LT
                timeOfLoading /*/ (2 * Math.PI)*/);   //on angular freq.

            double logEStar =
                delta + alpha / (1 + Math.Exp(beta + gamma * (logLoadTimeRefTemp)));
            double EStar = Math.Pow(10d, logEStar);

            return EStar;
        }


    }
}
```

Class: FEApplyConstraints

```
using System;
using System.Collections.Generic;
using System.Text;
using Mapack;

//
//Applies boundary conditions to model
//

namespace MPave
{
    class FEApplyConstraints
    {
        Matrix kk;
        int[] constrainedDOF, valuesDOF;
        double[] ff;

        //Constructor
        public FEApplyConstraints(Matrix kk, double[] ff,
            int[] constrainedDOF, int[] valuesDOF)
        {
            this.kk = kk;
            this.ff = ff;
            this.constrainedDOF = constrainedDOF;
            this.valuesDOF = valuesDOF;
        }

        //Set constrained nodes in global stiffness matrix to zero
        public void ApplyConstraints()
        {
            int n = constrainedDOF.Length;
            int sizeDOF = kk.Rows;
            int c;

            for (int i = 0; i < n; i++)
            {
                c = constrainedDOF[i];

                for (int j = 0; j < sizeDOF; j++)
                {
                    kk[c, j] = 0;
                }

                kk[c, c] = 1;
                ff[c] = valuesDOF[i];
            }
        }

        //Properties
        public Matrix getkk
        {
            get
            {
                return kk;
            }
        }

        public double[] getff
        {
            get
            {
                return ff;
            }
        }
    }
}
```

Class: FEAssemble1

```
using System;
using System.Collections.Generic;
using System.Text;
using Mapack;

//
//Assemble local stiffness matrices into global
//stiffness matrix
//

namespace MPave
{
    class FEAssemble1
    {
        Matrix kk;  //System stiffness matrix
        Matrix k;   //Element stiffness matrix
        int[] index; //d.o.f. vector associated with element

        public FEAssemble1(Matrix kk, Matrix k, int[] index)
        {
            this.kk = kk;
            this.k = k;
            this.index = index;
        }

        public Matrix Assemble()
        {
            int eDOF = index.Length;
            int ii, jj;

            for (int i = 0; i < eDOF; i++)
            {
                ii = index[i];

                for (int j = 0; j < eDOF; j++)
                {
                    jj = index[j];
                    kk[ii, jj] = kk[ii, jj] + k[i, j];
                }
            }

            return kk;
        }

    }
}
```

Class: FEDeriv2

```
using System;
using System.Collections.Generic;
using System.Text;
using Mapack;

//
//Compute shape functions
//

namespace MPave
{
    class FEDeriv2
    {
        int numNodesElement;
        double[] dhdr, dhds, dhdx, dhdy;
        Matrix invJacob;

        public FEDeriv2(int numNodesElement, double[] dhdr,
            double[] dhds, Matrix invJacob)
        {
            this.numNodesElement = numNodesElement;
            this.dhdr = dhdr;
            this.dhds = dhds;
            this.invJacob = invJacob;
        }

        public double[] dhdxShapeFunction()
        {
            dhdx = new double[numNodesElement];

            for (int i = 0; i < numNodesElement; i++)
            {
                dhdx[i] = invJacob[0, 0] * dhdr[i] + invJacob[0, 1] * dhds[i];
            }

            return dhdx;
        }

        public double[] dhdyShapeFunction()
        {
            dhdy = new double[numNodesElement];

            for (int i = 0; i < numNodesElement; i++)
            {
                dhdy[i] = invJacob[1, 0] * dhdr[i] + invJacob[1, 1] * dhds[i];
            }

            return dhdy;
        }

    }
}
```

Class: FEElementDOF

```
using System;
using System.Collections.Generic;
using System.Text;

//
//Indexing
//

namespace MPave
{
    class FEElementDOF
    {
        int[] node;
        int numNodesElement, numDOF;

        public FEElementDOF(int[] node, int numNodesElement, int numDOF)
        {
            this.node = node;
            this.numNodesElement = numNodesElement;
            this.numDOF = numDOF;
        }

        public int[] IndexDOF()
        {
            int elementDOF = numNodesElement * numDOF;
            int k = 0;
            int[] index = new int[elementDOF];

            for (int i = 0; i < numNodesElement; i++)
            {
                int start = (node[i]) * numDOF;

                for (int j = 0; j < numDOF; j++)
                {
                    index[k] = start + j;
                    k++;
                }
            }

            return index;
        }
    }
}
```

Class: FEGLQ1D

```
using System;
using System.Collections.Generic;
using System.Text;

//
//Gaussian-Legendre Quadrature - 1-dimension
//

namespace MPave
{
    class FEGLQ1D
    {
        int nGL;
        double[] points1D;
        double[] weight1D;

        public FEGLQ1D(int nGL)
        {
            this.nGL = nGL;
        }

        public double[] CreateIntegrationPoints1D()
        {
            //Left out other cases, only nGL = 2 is used here.

            points1D = new double[nGL];

            points1D[0] = -0.577350269189626;
            points1D[1] = 0.577350269189626;

            return points1D;
        }

        public double[] CreateIntegrationWeights1D()
        {
            weight1D = new double[nGL];

            weight1D[0] = 1.0;
            weight1D[1] = 1.0;

            return weight1D;
        }

    }
}
```

Class: FEGLQ2D

```
using System;
using System.Collections.Generic;
using System.Text;
using Mapack;

//
//Gaussian-Legendre Quadrature - 2-dimensions
//

namespace MPave
{
    class FEGLQ2D
    {
        int nGLQX;
        int nGLQY;
        int nGL;
        //double[,] points2Dd, weights2Dd;
        Matrix points2D, weights2D;

        public FEGLQ2D(int nGLQX, int nGLQY)
        {
            this.nGLQX = nGLQX;
            this.nGLQY = nGLQY;
        }

        public Matrix CreateIntegrationPoints2D()
        {
            if (nGLQX > nGLQY)
            {
                nGL = nGLQX;
            }
            else
            {
                nGL = nGLQY;
            }

            FEGLQ1D new1D = new FEGLQ1D(nGL);
            double[] pointX = new1D.CreateIntegrationPoints1D();
            double[] pointY = new1D.CreateIntegrationPoints1D();

            //points2Dd = new double[nGL, 2];
            points2D = new Matrix(nGL, 2);

            for (int i = 0; i < nGLQX; i++)
            {
                points2D[i, 0] = pointX[i];
            }
            for (int i = 0; i < nGLQX; i++)
            {
                points2D[i, 1] = pointY[i];
            }

            return points2D;
        }

        public Matrix CreateIntegrationWeight2D()
        {
            if (nGLQX > nGLQY)
            {
                nGL = nGLQX;
            }
            else
            {
                nGL = nGLQY;
            }

            FEGLQ1D new1D = new FEGLQ1D(nGL);
```

```
            double[] weightX = new1D.CreateIntegrationWeights1D();
            double[] weightY = new1D.CreateIntegrationWeights1D();

            //weight2Dd = new double[nGL, 2];
            weights2D = new Matrix(nGL, 2);

            for (int i = 0; i < nGLQX; i++)
            {
                weights2D[i, 0] = weightX[i];
            }
            for (int i = 0; i < nGLQX; i++)
            {
                weights2D[i, 1] = weightY[i];
            }

            return weights2D;
        }
    }
}
```

Class: FEIsoQ4

```
using System;
using System.Collections.Generic;
using System.Text;

//
//Generates integration sampling values
//

namespace MPave
{
    class FEIsoQ4
    {
        double[] shapeq4 = new double[4];
        double[] dhdrq4 = new double[4];
        double[] dhdsq4 = new double[4];
        double rValue, sValue;

        public FEIsoQ4(double rValue, double sValue)
        {
            this.rValue = rValue;
            this.sValue = sValue;
        }

        public double[] ComputeShapeFunction()
        {
            shapeq4[0] = 0.25 * (1 - rValue) * (1 - sValue);
            shapeq4[1] = 0.25 * (1 + rValue) * (1 - sValue);
            shapeq4[2] = 0.25 * (1 + rValue) * (1 + sValue);
            shapeq4[3] = 0.25 * (1 - rValue) * (1 + sValue);
            return shapeq4;
        }

        public double[] Computedhdrq4()
        {
            dhdrq4[0] = -0.25 * (1 - sValue);
            dhdrq4[1] = 0.25 * (1 - sValue);
            dhdrq4[2] = 0.25 * (1 + sValue);
            dhdrq4[3] = -0.25 * (1 + sValue);
            return dhdrq4;
        }

        public double[] Computedhdsq4()
        {
            dhdsq4[0] = -0.25 * (1 - rValue);
            dhdsq4[1] = -0.25 * (1 + rValue);
            dhdsq4[2] = 0.25 * (1 + rValue);
            dhdsq4[3] = 0.25 * (1 - rValue);
            return dhdsq4;
        }
    }
}
```

Class: FEJacob2

```
using System;
using System.Collections.Generic;
using System.Text;
using Mapack;

//
//Compute unit Jacobian
//

namespace MPave
{
    class FEJacob2
    {
        int numNodesElement;
        double[] dhdr, dhds, xCoord, yCoord;
        Matrix jacob2D = new Matrix(2, 2);

        public FEJacob2(int numNodesElement, double[] xCoord,
            double[] yCoord, double[] dhdr, double[] dhds)
        {
            this.numNodesElement = numNodesElement;
            this.xCoord = xCoord;
            this.yCoord = yCoord;
            this.dhdr = dhdr;
            this.dhds = dhds;
        }

        /// <summary>
        /// Computes the Jacobian
        /// </summary>
        /// <returns>Returns the 2 x 2 Jacobian</returns>
        public Matrix ComputeJacobian2D()
        {
            for (int i = 0; i < numNodesElement; i++)
            {
                jacob2D[0, 0] = jacob2D[0, 0] + dhdr[i] * xCoord[i];
                jacob2D[0, 1] = jacob2D[0, 1] + dhdr[i] * yCoord[i];
                jacob2D[1, 0] = jacob2D[1, 0] + dhds[i] * xCoord[i];
                jacob2D[1, 1] = jacob2D[1, 1] + dhds[i] * yCoord[i];
            }

            return jacob2D;
        }

    }
}
```

Class: FEKineAx

```
using System;
using System.Collections.Generic;
using System.Text;
using Mapack;

//
//Compute kinematic matrix
//

namespace MPave
{
    class FEKineAx
    {
        int numNodesElement;
        double[] dhdx, dhdy, shape;
        double radialDist;

        public FEKineAx(int numNodesElement, double[] dhdx,
            double[] dhdy, double[] shape, double radialDist)
        {
            this.numNodesElement = numNodesElement;
            this.dhdx = dhdx;
            this.dhdy = dhdy;
            this.shape = shape;
            this.radialDist = radialDist;
        }

        /// <summary>
        /// Computes the kinematic matrix.
        /// </summary>
        /// <returns>Kinematic matrix</returns>
        public Matrix CreateKinematicMatrix()
        {
            Matrix kinematic = new Matrix(4, 8);
            int i1;
            int i2;
            int index = 0;

            for (int i = 0; i <= numNodesElement; i++)
            {
                i1 = (i - 1) * 2;
                i2 = i1 + 1;

                if (i != 0)
                {
                    kinematic[0, i1] = dhdx[index];
                    kinematic[1, i1] = shape[index] / radialDist;
                    kinematic[2, i2] = dhdy[index];
                    kinematic[3, i1] = dhdy[index];
                    kinematic[3, i2] = dhdx[index];
                    index++;
                }
            }

            return kinematic;
        }

    }
}
```

Class: FEMatIso

```
using System;
using System.Collections.Generic;
using System.Text;
using Mapack;

//
//Assembles/calculates the material matrix
//

namespace MPave
{
    class FEMatIso
    {
        double modulus;
        double poissonRatio;

        public FEMatIso(double modulus, double poissonRatio)
        {
            this.modulus = modulus;
            this.poissonRatio = poissonRatio;
        }

        public Matrix AssembleMaterialMatrix()
        {
            Matrix materialMatrix = new Matrix(4, 4);

            double scalarValue =
                modulus / ((1 + poissonRatio) * (1 - 2 * poissonRatio));

            materialMatrix[0, 0] = 1 - poissonRatio;
            materialMatrix[0, 1] = poissonRatio;
            materialMatrix[0, 2] = poissonRatio;
            materialMatrix[0, 3] = 0;

            materialMatrix[1, 0] = poissonRatio;
            materialMatrix[1, 1] = 1 - poissonRatio;
            materialMatrix[1, 2] = poissonRatio;
            materialMatrix[1, 3] = 0;

            materialMatrix[2, 0] = poissonRatio;
            materialMatrix[2, 1] = poissonRatio;
            materialMatrix[2, 2] = 1 - poissonRatio;
            materialMatrix[2, 3] = 0;

            materialMatrix[3, 0] = 0;
            materialMatrix[3, 1] = 0;
            materialMatrix[3, 2] = 0;
            materialMatrix[3, 3] = (1 - 2 * poissonRatio) / 2;

            materialMatrix = Matrix.Multiply(materialMatrix, scalarValue);

            return materialMatrix;
        }

    }
}
```

Class: ForceVector

```
using System;
using System.Collections.Generic;
using System.Text;

//
//Generates force vectors for models.
//

namespace MPave
{
    class ForceVector
    {
        int sDOF;
        double[] forceVector;
        int[] yCoords, xCoords;
        double area, contactRadius, wheelLoad, contactPressure;

        //Constructor
        public ForceVector(int sDOF, double contactRadius, double wheelLoad,
            int[] yCoords, int[] xCoords)
        {
            this.sDOF = sDOF;
            this.contactRadius = contactRadius;
            this.wheelLoad = wheelLoad / 2;
            this.yCoords = yCoords;
            this.xCoords = xCoords;

            forceVector = new double[sDOF];

            for (int i = 0; i < forceVector.Length; i++)
            {
                forceVector[i] = 0d;
            }
        }
        /// <summary>
        /// Generates the force vector based on y-coordinate
        /// model array based on wheel
        /// load and contact radius.
        /// </summary>
        /// <returns></returns>
        public double[] GenerateForceVector()
        {
            int maxYCoord = yCoords.Length;
            int forceDOF = (maxYCoord) * 2 - 1;
            int shift = maxYCoord * 2;

            if (contactRadius < 1.5)
            {
                area = Math.PI * Math.Pow(1.0, 2);
                contactPressure = wheelLoad / area;

                forceVector[forceDOF] =
                    -0.25 * contactPressure * Math.PI;

                forceVector[forceDOF + shift] =
                    -0.75 * contactPressure * Math.PI;
            }

            if (contactRadius >= 1.5 && contactRadius < 2.5)
            {
                area = Math.PI * Math.Pow(2.0, 2);
                contactPressure = wheelLoad / area;

                forceVector[forceDOF] =
                    -0.25 * contactPressure * Math.PI;

                forceVector[forceDOF + shift] =
                    -2.0 * contactPressure * Math.PI;
```

```
    forceVector[forceDOF + shift * 2] =
        -1.75 * contactPressure * Math.PI;
}

if (contactRadius >= 2.5 && contactRadius < 3.5)
{
    area = Math.PI * Math.Pow(3.0, 2);
    contactPressure = wheelLoad / area;

    forceVector[forceDOF] =
        -0.25 * contactPressure * Math.PI;

    forceVector[forceDOF + shift] =
        -2.0 * contactPressure * Math.PI;

    forceVector[forceDOF + shift * 2] =
        -4.0 * contactPressure * Math.PI;

    forceVector[forceDOF + shift * 3] =
        -2.75 * contactPressure * Math.PI;
}

if (contactRadius >= 3.5 && contactRadius < 4.5)
{
    area = Math.PI * Math.Pow(4.0, 2);
    contactPressure = wheelLoad / area;

    forceVector[forceDOF] =
        -0.25 * contactPressure * Math.PI;

    forceVector[forceDOF + shift] =
        -2.0 * contactPressure * Math.PI;

    forceVector[forceDOF + shift * 2] =
        -4.0 * contactPressure * Math.PI;

    forceVector[forceDOF + shift * 3] =
        -6.0 * contactPressure * Math.PI;

    forceVector[forceDOF + shift * 4] =
        -3.75 * contactPressure * Math.PI;
}

if (contactRadius >= 4.5 && contactRadius < 5.5)
{
    area = Math.PI * Math.Pow(5.0, 2);
    contactPressure = wheelLoad / area;

    forceVector[forceDOF] =
        -0.25 * contactPressure * Math.PI;

    forceVector[forceDOF + shift] =
        -2.0 * contactPressure * Math.PI;

    forceVector[forceDOF + shift * 2] =
        -4.0 * contactPressure * Math.PI;

    forceVector[forceDOF + shift * 3] =
        -6.0 * contactPressure * Math.PI;

    forceVector[forceDOF + shift * 4] =
        -8.0 * contactPressure * Math.PI;

    forceVector[forceDOF + shift * 5] =
        -4.75 * contactPressure * Math.PI;
}

if (contactRadius >= 5.5 && contactRadius < 6.5)
{
    area = Math.PI * Math.Pow(6.0, 2);
```

```
        contactPressure = wheelLoad / area;

        forceVector[forceDOF] =
            -0.25 * contactPressure * Math.PI;

        forceVector[forceDOF + shift] =
            -2.0 * contactPressure * Math.PI;

        forceVector[forceDOF + shift * 2] =
            -4.0 * contactPressure * Math.PI;

        forceVector[forceDOF + shift * 3] =
            -6.0 * contactPressure * Math.PI;

        forceVector[forceDOF + shift * 4] =
            -8.0 * contactPressure * Math.PI;

        forceVector[forceDOF + shift * 5] =
            -10.0 * contactPressure * Math.PI;

        forceVector[forceDOF + shift * 6] =
            -5.75 * contactPressure * Math.PI;
    }

    if (contactRadius >= 6.5 && contactRadius < 7.5)
    {
        area = Math.PI * Math.Pow(7.0, 2);
        contactPressure = wheelLoad / area;

        forceVector[forceDOF] =
            -0.25 * contactPressure * Math.PI;

        forceVector[forceDOF + shift] =
            -2.0 * contactPressure * Math.PI;

        forceVector[forceDOF + shift * 2] =
            -4.0 * contactPressure * Math.PI;

        forceVector[forceDOF + shift * 3] =
            -6.0 * contactPressure * Math.PI;

        forceVector[forceDOF + shift * 4] =
            -8.0 * contactPressure * Math.PI;

        forceVector[forceDOF + shift * 5] =
            -10.0 * contactPressure * Math.PI;

        forceVector[forceDOF + shift * 6] =
            -12.0 * contactPressure * Math.PI;

        forceVector[forceDOF + shift * 7] =
            -6.75 * contactPressure * Math.PI;
    }

    if (contactRadius >= 7.5)
    {
        area = Math.PI * Math.Pow(8.0, 2);
        contactPressure = wheelLoad / area;

        forceVector[forceDOF] =
            -0.25 * contactPressure * Math.PI;

        forceVector[forceDOF + shift] =
            -2.0 * contactPressure * Math.PI;

        forceVector[forceDOF + shift * 2] =
            -4.0 * contactPressure * Math.PI;

        forceVector[forceDOF + shift * 3] =
            -6.0 * contactPressure * Math.PI;
```

```
            forceVector[forceDOF + shift * 4] =
                -8.0 * contactPressure * Math.PI;

            forceVector[forceDOF + shift * 5] =
                -10.0 * contactPressure * Math.PI;

            forceVector[forceDOF + shift * 6] =
                -12.0 * contactPressure * Math.PI;

            forceVector[forceDOF + shift * 7] =
                -14.0 * contactPressure * Math.PI;

            forceVector[forceDOF + shift * 8] =
                -7.75 * contactPressure * Math.PI;
        }

        //For Debugging
        //Console.WriteLine("Wheel LoadE3: " + wheelLoad.ToString() +
        //    " \t Contact Radius: " + contactRadius.ToString());
        //Console.WriteLine("Contact Pressure:  " +
        //    contactPressure.ToString());
        //Console.ReadLine();
        return forceVector;
    }

    //************************322 Element Mesh**************************
    /// <summary>
    /// Generates force vector for 322 element mesh.
    /// </summary>
    /// <returns></returns>
    public double[] GenerateForceVector322()
    {
        if (contactRadius < 1.5)
        {
            area = Math.PI * Math.Pow(1.0, 2);
            contactPressure = wheelLoad / area;

            //322 Element
            forceVector[47] = -0.25 * contactPressure * Math.PI;
            forceVector[95] = -0.75 * contactPressure * Math.PI;
        }

        if (contactRadius >= 1.5 && contactRadius < 2.5)
        {
            area = Math.PI * Math.Pow(2.0, 2);
            contactPressure = wheelLoad / area;

            forceVector[47] = -0.25 * contactPressure * Math.PI;
            forceVector[95] = -2.0 * contactPressure * Math.PI;
            forceVector[143] = -1.75 * contactPressure * Math.PI;
        }

        if (contactRadius >= 2.5 && contactRadius < 3.5)
        {
            area = Math.PI * Math.Pow(3.0, 2);
            contactPressure = wheelLoad / area;

            forceVector[47] = -0.25 * contactPressure * Math.PI;
            forceVector[95] = -2.0 * contactPressure * Math.PI;
            forceVector[143] = -4.0 * contactPressure * Math.PI;
            forceVector[191] = -2.75 * contactPressure * Math.PI;
        }

        if (contactRadius >= 3.5 && contactRadius < 4.5)
        {
            area = Math.PI * Math.Pow(4.0, 2);
            contactPressure = wheelLoad / area;

            forceVector[47] = -0.25 * contactPressure * Math.PI;
            forceVector[95] = -2.0 * contactPressure * Math.PI;
            forceVector[143] = -4.0 * contactPressure * Math.PI;
```

```
            forceVector[191] = -6.0 * contactPressure * Math.PI;
            forceVector[239] = -3.75 * contactPressure * Math.PI;
        }

        if (contactRadius >= 4.5 && contactRadius < 5.5)
        {
            area = Math.PI * Math.Pow(5.0, 2);
            contactPressure = wheelLoad / area;

            forceVector[47] = -0.25 * contactPressure * Math.PI;
            forceVector[95] = -2.0 * contactPressure * Math.PI;
            forceVector[143] = -4.0 * contactPressure * Math.PI;
            forceVector[191] = -6.0 * contactPressure * Math.PI;
            forceVector[239] = -8.0 * contactPressure * Math.PI;
            forceVector[287] = -4.75 * contactPressure * Math.PI;
        }

        if (contactRadius >= 5.5 && contactRadius < 6.5)
        {
            area = Math.PI * Math.Pow(6.0, 2);
            contactPressure = wheelLoad / area;

            forceVector[47] = -0.25 * contactPressure * Math.PI;
            forceVector[95] = -2.0 * contactPressure * Math.PI;
            forceVector[143] = -4.0 * contactPressure * Math.PI;
            forceVector[191] = -6.0 * contactPressure * Math.PI;
            forceVector[239] = -8.0 * contactPressure * Math.PI;
            forceVector[287] = -10.0 * contactPressure * Math.PI;
            forceVector[335] = -5.75 * contactPressure * Math.PI;
        }

        if (contactRadius >= 6.5 && contactRadius < 7.5)
        {
            area = Math.PI * Math.Pow(7.0, 2);
            contactPressure = wheelLoad / area;

            forceVector[47] = -0.25 * contactPressure * Math.PI;
            forceVector[95] = -2.0 * contactPressure * Math.PI;
            forceVector[143] = -4.0 * contactPressure * Math.PI;
            forceVector[191] = -6.0 * contactPressure * Math.PI;
            forceVector[239] = -8.0 * contactPressure * Math.PI;
            forceVector[287] = -10.0 * contactPressure * Math.PI;
            forceVector[335] = -12.0 * contactPressure * Math.PI;
            forceVector[383] = -6.75 * contactPressure * Math.PI;
        }

        if (contactRadius >= 7.5)
        {
            area = Math.PI * Math.Pow(8.0, 2);
            contactPressure = wheelLoad / area;

            forceVector[47] = -0.25 * contactPressure * Math.PI;
            forceVector[95] = -2.0 * contactPressure * Math.PI;
            forceVector[143] = -4.0 * contactPressure * Math.PI;
            forceVector[191] = -6.0 * contactPressure * Math.PI;
            forceVector[239] = -8.0 * contactPressure * Math.PI;
            forceVector[287] = -10.0 * contactPressure * Math.PI;
            forceVector[335] = -12.0 * contactPressure * Math.PI;
            forceVector[383] = -14.0 * contactPressure * Math.PI;
            forceVector[432] = -7.75 * contactPressure * Math.PI;
        }

        return forceVector;
    }

    //**********************522 Element Mesh*****************************
    /// <summary>
    /// Generates force vector for 522 element mesh.
    /// </summary>
    /// <returns></returns>
    public double[] GenerateForceVector522()
```

```
{
    if (contactRadius < 1.5)
    {
        area = Math.PI * Math.Pow(1.0, 2);
        contactPressure = wheelLoad / area;

        forceVector[59] = -0.25 * contactPressure * Math.PI;
        forceVector[119] = -0.75 * contactPressure * Math.PI;
    }

    if (contactRadius >= 1.5 && contactRadius < 2.5)
    {
        area = Math.PI * Math.Pow(2.0, 2);
        contactPressure = wheelLoad / area;

        forceVector[59] = -0.25 * contactPressure * Math.PI;
        forceVector[119] = -2.0 * contactPressure * Math.PI;
        forceVector[179] = -1.75 * contactPressure * Math.PI;
    }

    if (contactRadius >= 2.5 && contactRadius < 3.5)
    {
        area = Math.PI * Math.Pow(3.0, 2);
        contactPressure = wheelLoad / area;

        forceVector[59] = -0.25 * contactPressure * Math.PI;
        forceVector[119] = -2.0 * contactPressure * Math.PI;
        forceVector[179] = -4.0 * contactPressure * Math.PI;
        forceVector[239] = -2.75 * contactPressure * Math.PI;
    }

    if (contactRadius >= 3.5 && contactRadius < 4.5)
    {
        area = Math.PI * Math.Pow(4.0, 2);
        contactPressure = wheelLoad / area;

        forceVector[59] = -0.25 * contactPressure * Math.PI;
        forceVector[119] = -2.0 * contactPressure * Math.PI;
        forceVector[179] = -4.0 * contactPressure * Math.PI;
        forceVector[239] = -6.0 * contactPressure * Math.PI;
        forceVector[299] = -3.75 * contactPressure * Math.PI;
    }

    if (contactRadius >= 4.5 && contactRadius < 5.5)
    {
        area = Math.PI * Math.Pow(5.0, 2);
        contactPressure = wheelLoad / area;

        forceVector[59] = -0.25 * contactPressure * Math.PI;
        forceVector[119] = -2.0 * contactPressure * Math.PI;
        forceVector[179] = -4.0 * contactPressure * Math.PI;
        forceVector[239] = -6.0 * contactPressure * Math.PI;
        forceVector[299] = -8.0 * contactPressure * Math.PI;
        forceVector[359] = -4.75 * contactPressure * Math.PI;
    }

    if (contactRadius >= 5.5 && contactRadius < 6.5)
    {
        area = Math.PI * Math.Pow(6.0, 2);
        contactPressure = wheelLoad / area;

        forceVector[59] = -0.25 * contactPressure * Math.PI;
        forceVector[119] = -2.0 * contactPressure * Math.PI;
        forceVector[179] = -4.0 * contactPressure * Math.PI;
        forceVector[239] = -6.0 * contactPressure * Math.PI;
        forceVector[299] = -8.0 * contactPressure * Math.PI;
        forceVector[359] = -10.0 * contactPressure * Math.PI;
        forceVector[419] = -5.75 * contactPressure * Math.PI;
    }

    if (contactRadius >= 6.5 && contactRadius < 7.5)
```

```
    {
        area = Math.PI * Math.Pow(7.0, 2);
        contactPressure = wheelLoad / area;

        forceVector[59] = -0.25 * contactPressure * Math.PI;
        forceVector[119] = -2.0 * contactPressure * Math.PI;
        forceVector[179] = -4.0 * contactPressure * Math.PI;
        forceVector[239] = -6.0 * contactPressure * Math.PI;
        forceVector[299] = -8.0 * contactPressure * Math.PI;
        forceVector[359] = -10.0 * contactPressure * Math.PI;
        forceVector[419] = -12.0 * contactPressure * Math.PI;
        forceVector[479] = -6.75 * contactPressure * Math.PI;
    }

    if (contactRadius >= 7.5)
    {
        area = Math.PI * Math.Pow(8.0, 2);
        contactPressure = wheelLoad / area;

        forceVector[59] = -0.25 * contactPressure * Math.PI;
        forceVector[119] = -2.0 * contactPressure * Math.PI;
        forceVector[179] = -4.0 * contactPressure * Math.PI;
        forceVector[239] = -6.0 * contactPressure * Math.PI;
        forceVector[299] = -8.0 * contactPressure * Math.PI;
        forceVector[359] = -10.0 * contactPressure * Math.PI;
        forceVector[419] = -12.0 * contactPressure * Math.PI;
        forceVector[479] = -14.0 * contactPressure * Math.PI;
        forceVector[539] = -7.75 * contactPressure * Math.PI;
    }

    return forceVector;
}

//*************************2000 Element Mesh***********************
/// <summary>
/// Generates force vector for 2000 element mesh.
/// </summary>
/// <returns></returns>
public double[] GenerateForceVector2000()
{
    if (contactRadius < 1.5)
    {
        area = Math.PI * Math.Pow(1.0, 2);
        contactPressure = wheelLoad / area;

        forceVector[101] = -0.25 * contactPressure * Math.PI;
        forceVector[203] = -0.75 * contactPressure * Math.PI;
    }

    if (contactRadius >= 1.5 && contactRadius < 2.5)
    {
        area = Math.PI * Math.Pow(2.0, 2);
        contactPressure = wheelLoad / area;

        forceVector[101] = -0.25 * contactPressure * Math.PI;
        forceVector[203] = -2.0 * contactPressure * Math.PI;
        forceVector[305] = -1.75 * contactPressure * Math.PI;
    }

    if (contactRadius >= 2.5 && contactRadius < 3.5)
    {
        area = Math.PI * Math.Pow(3.0, 2);
        contactPressure = wheelLoad / area;

        forceVector[101] = -0.25 * contactPressure * Math.PI;
        forceVector[203] = -2.0 * contactPressure * Math.PI;
        forceVector[305] = -4.0 * contactPressure * Math.PI;
        forceVector[407] = -2.75 * contactPressure * Math.PI;
    }

    if (contactRadius >= 3.5 && contactRadius < 4.5)
```

```
    {
        area = Math.PI * Math.Pow(4.0, 2);
        contactPressure = wheelLoad / area;

        forceVector[101] = -0.25 * contactPressure * Math.PI;
        forceVector[203] = -2.0 * contactPressure * Math.PI;
        forceVector[305] = -4.0 * contactPressure * Math.PI;
        forceVector[407] = -6.0 * contactPressure * Math.PI;
        forceVector[509] = -3.75 * contactPressure * Math.PI;
    }

    if (contactRadius >= 4.5 && contactRadius < 5.5)
    {
        area = Math.PI * Math.Pow(5.0, 2);
        contactPressure = wheelLoad / area;

        forceVector[101] = -0.25 * contactPressure * Math.PI;
        forceVector[203] = -2.0 * contactPressure * Math.PI;
        forceVector[305] = -4.0 * contactPressure * Math.PI;
        forceVector[407] = -6.0 * contactPressure * Math.PI;
        forceVector[509] = -8.0 * contactPressure * Math.PI;
        forceVector[611] = -4.75 * contactPressure * Math.PI;
    }

    if (contactRadius >= 5.5 && contactRadius < 6.5)
    {
        area = Math.PI * Math.Pow(6.0, 2);
        contactPressure = wheelLoad / area;

        forceVector[101] = -0.25 * contactPressure * Math.PI;
        forceVector[203] = -2.0 * contactPressure * Math.PI;
        forceVector[305] = -4.0 * contactPressure * Math.PI;
        forceVector[407] = -6.0 * contactPressure * Math.PI;
        forceVector[509] = -8.0 * contactPressure * Math.PI;
        forceVector[611] = -10.0 * contactPressure * Math.PI;
        forceVector[713] = -5.75 * contactPressure * Math.PI;
    }

    if (contactRadius >= 6.5 && contactRadius < 7.5)
    {
        area = Math.PI * Math.Pow(7.0, 2);
        contactPressure = wheelLoad / area;

        forceVector[101] = -0.25 * contactPressure * Math.PI;
        forceVector[203] = -2.0 * contactPressure * Math.PI;
        forceVector[305] = -4.0 * contactPressure * Math.PI;
        forceVector[407] = -6.0 * contactPressure * Math.PI;
        forceVector[509] = -8.0 * contactPressure * Math.PI;
        forceVector[611] = -10.0 * contactPressure * Math.PI;
        forceVector[713] = -12.0 * contactPressure * Math.PI;
        forceVector[815] = -6.75 * contactPressure * Math.PI;
    }

    if (contactRadius >= 7.5)
    {
        area = Math.PI * Math.Pow(8.0, 2);
        contactPressure = wheelLoad / area;

        forceVector[101] = -0.25 * contactPressure * Math.PI;
        forceVector[203] = -2.0 * contactPressure * Math.PI;
        forceVector[305] = -4.0 * contactPressure * Math.PI;
        forceVector[407] = -6.0 * contactPressure * Math.PI;
        forceVector[509] = -8.0 * contactPressure * Math.PI;
        forceVector[611] = -10.0 * contactPressure * Math.PI;
        forceVector[713] = -12.0 * contactPressure * Math.PI;
        forceVector[815] = -14.0 * contactPressure * Math.PI;
        forceVector[917] = -7.75 * contactPressure * Math.PI;
    }

    return forceVector;
}
```

```
        }
    }
```

Class: Mesh

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Collections;

//
//Creates mesh coordinates for rectangular model
//

namespace MPave
{
    class Mesh
    {
        int[] xValues;
        int[] yValues;

        public Mesh(int[] xValues, int[] yValues)
        {
            this.xValues = xValues;
            this.yValues = yValues;
            //xValues =
            //    new int[] { 0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 18, 25, 32, 39 };

            //yValues =
            //    new int[] { 0, 9, 15, 21, 24, 27, 30, 33, 35, 36, 37, 38, 39,
            //        40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50 };
        }

        /// <summary>
        /// Returns an n x 2 array of the global coordinates
        /// for each model node, n.
        /// </summary>
        public int[,] CreateModelNodeCoord()
        {
            int nNodes = xValues.Length * yValues.Length;
            int[,] globalCoord = new int[nNodes, 2];
            int rowCount = 0;

            //Loop building the nodal coordinates
            for (int i = 0; i < xValues.Length; i++)
            {
                for (int j = 0; j < yValues.Length; j++)
                {
                    globalCoord[rowCount,0] = xValues[i];
                    globalCoord[rowCount,1] = yValues[j];

                    rowCount++;
                }
            }

            //
            //For debugging...
            //

            //for (int i = 0; i < nNodes; i++)
            //{
            //Console.WriteLine(globalCoord[i, 0].ToString() +
            //    ", " + globalCoord[i, 1].ToString());
            //    if (i % 100 == 0)
            //    {
            //        Console.ReadLine();
            //    }
            //}
            //Console.ReadLine();
            return globalCoord;

        }
```

```
        /// <summary>
        /// Returns an n x 4 array - each row containing
        /// connected nodes for n-th element.
        /// </summary>
        public int[,] CreateModelConnectivity()
        {
            int nElements = (xValues.Length - 1) * (yValues.Length - 1);
            int[,] elemNodes = new int[nElements, 4];
            int node = 0;

            for (int i = 0; i < nElements; i++)
            {
                if (i % (yValues.Length-1) == 0 && i != 0)
                {
                    node++;
                }

                elemNodes[i, 0] = node;
                elemNodes[i, 1] = node + yValues.Length;
                elemNodes[i, 2] = node + yValues.Length + 1;
                elemNodes[i, 3] = node + 1;



                node++;
            }

            return elemNodes;

            //For debugging...
            //
            //for (int i = 0; i < nElements; i++)
            //{
            //    Console.WriteLine(elemNodes[i, 0].ToString() + ", " +
            //        elemNodes[i, 1].ToString() + ", " +
            //        elemNodes[i, 2].ToString() + ", " +
            //        elemNodes[i, 3].ToString());
            //}
        }
    }
}
```

Class: METIterator

```
using System;
using System.Collections.Generic;
using System.Text;

//
//Iterates through method of equivalent thickness and solves
//until convergence is met.
//

namespace MPave
{
    class METIterator
    {
        //***************Material Constants*************************************
        int[] materialCode = new int[] { 0, 0, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2 };
        double[,] materialParameters = new double[3, 7];
        double[] effectiveDepth = new double[13];
        //**********************************************************************

        double stressAngle = 45 * Math.PI / 180; //Radians
        //Optional angle found through Load Time Analysis = 65 degrees (both)


        //**********************************************************************
        double loadTimeAdjust = 1.0;  //Factor of the load time
        //**********************************************************************

        bool axleGroup = false;
        double[] stiffness, temperature;
        double vehicleSpeed, subGradeStiffness, contactRadius;
        double tandemSpacing = 0;

        public METIterator(double[] temperature, double vehicleSpeed,
            double subGradeStiffness, double contactRadius, bool axleGroup)
        {
            this.temperature = temperature;
            this.vehicleSpeed = vehicleSpeed;
            this.subGradeStiffness = subGradeStiffness;
            this.contactRadius = contactRadius;
            this.axleGroup = axleGroup;
            if (axleGroup)
            {
                tandemSpacing = 53;  //53 inch tandem spacing -
                                     //averaged from WIM data of class 9 vehicles
                                     //120 inch spacing used as a test of sensivity
            }
            InitializeLayerStiffnessData();
        }

        public double[] SolveStiffness()
        {
            //Retrieve initial loading time, t from somewhere as
            //function of vehicle speed
            double loadTime = .05; //This doesn't really affect anything,
                                   //just a starting point

            stiffness = new double[13];

            //
            //Estimate initial stiffness parameters
            //

            for (int i = 0; i < stiffness.Length; i++)
            {
                int material = materialCode[i];
                DynamicModulus modulus =
                    new DynamicModulus(materialParameters[material, 0],
                    materialParameters[material, 1],
```

```
            materialParameters[material, 2],
            materialParameters[material, 3],
            materialParameters[material, 4],
            materialParameters[material, 5],
            materialParameters[material, 6]);

        double layerTemperature = temperature[i];

        stiffness[i] =
            modulus.CalculateStiffness(layerTemperature, loadTime);

        //
        //Begin iteration loops
        //
        bool converged = false;
        double trialDepth = 0;
        double power = 1.0 / 3.0;
        do
        {
            trialDepth = 0;
            for (int j = 0; j <= i; j++)
            {
                double depth = 0;

                if (j == i)
                {
                    double moduluarRatio =
                        stiffness[j] / subGradeStiffness;
                    depth = 0.5 * Math.Pow(moduluarRatio, power);
                    trialDepth += depth;
                }
                else
                {
                    double moduluarRatio =
                        stiffness[j] / subGradeStiffness;
                    depth = Math.Pow(moduluarRatio, power);
                    trialDepth += depth;
                }
            }

            double effLength =
                2 * (contactRadius + trialDepth/Math.Tan(stressAngle))+
                tandemSpacing; //inches

            double updatedLoadTime =
                effLength / (17.6 * vehicleSpeed /** 2 * Math.PI*/) *
                loadTimeAdjust;  //seconds - //'out 2Pi for normal frequency

            double updatedStiffness =
                modulus.CalculateStiffness(temperature[i], updatedLoadTime);

            double convergence =
                (stiffness[i] - updatedStiffness) / stiffness[i] * 100;

            stiffness[i] = updatedStiffness;
            if (Math.Abs(convergence) < 0.1)
            {
                converged = true;
                effectiveDepth[i] = trialDepth;
            }
        } while (converged == false);
    }


    //For debugging
    //Console.WriteLine("Temperature: " + temperature[0].ToString() +
    //    "\tSpeed: " + vehicleSpeed.ToString());
    //Console.WriteLine("Stiffness,Effective Depth");
    //int m = 0;
    //foreach (double d in stiffness)
    //{
```

```
            //Console.WriteLine(stiffness[m].ToString() +
            //    "," + effectiveDepth[m].ToString());
            //    m++;
            //}
            //Console.ReadLine();


            return stiffness;
        }

        //Hard-coded material data from MRUTC study
        private void InitializeLayerStiffnessData()
        {
            //SMA
            materialParameters[0, 0] = 4.1560d;//Delta
            materialParameters[0, 1] = 2.4244d;//Alpha
            materialParameters[0, 2] = -0.3918d;//Beta
            materialParameters[0, 3] = 0.4794d;//Gamma
            materialParameters[0, 4] = 1.6996d;//C
            materialParameters[0, 5] = 8.458896d;//A
            materialParameters[0, 6] = -2.78133d;//VTS

            //E30
            materialParameters[1, 0] = 2.5566d;//Delta
            materialParameters[1, 1] = 4.2125d;//Alpha
            materialParameters[1, 2] = -1.5623d;//Beta
            materialParameters[1, 3] = 0.3937d;//Gamma
            materialParameters[1, 4] = 1.6214d;//C
            materialParameters[1, 5] = 9.773228d;//A
            materialParameters[1, 6] = -3.26367d;//VTS

            //C2
            materialParameters[2, 0] = 1.6841d;//Delta
            materialParameters[2, 1] = 5.1418d;//Alpha
            materialParameters[2, 2] = -1.7168d;//Beta
            materialParameters[2, 3] = 0.3650d;//Gamma
            materialParameters[2, 4] = 1.6905d;//C
            materialParameters[2, 5] = 9.773228d;//A
            materialParameters[2, 6] = -3.26367d;//VTS
        }
    }
}
```

Class: ModelConstants

```
using System;
using System.Collections.Generic;
using System.Text;
using Mapack;

//
//Creates x and y coordinates for mesh and also contains patterning
//for striping material across elements in model.
//

namespace MPave
{
    class ModelConstants
    {
        //*******************1178 Element Mesh - Created 03152009***************
        int[] xValues = new int[] { 0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 14, 16, 18,
            20, 22, 24, 26, 28, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57, 60, 65, 70};
        int[] yValues = new int[] { 0, 3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33,
            36, 39, 42, 45, 48, 51, 54, 56, 58, 60, 61, 62, 63, 64, 65, 66, 67,
            68, 69, 70, 71, 72, 73, 74, 75, 76, 77 };
        int[] materialPattern = new int[] { 18, 18, 18, 18, 18, 18, 18, 18, 17, 17,
            17, 17, 16, 16, 16, 16, 16, 16, 15, 15, 15, 14, 14, 14, 14, 13, 12, 11,
            10, 9, 8, 7, 6, 5, 4, 3, 2, 1 };



        ////*******************522 Element Mesh - Created 01282009**************
        //int[] xValues = new int[] { 0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 14, 16, 18,
        //    20, 25, 30, 35, 40};
        //int[] yValues = new int[] { 0, 3, 6, 9, 12, 15, 18, 21, 24, 27, 29, 31, 33,
        //    34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50 };

        //int[] materialPattern = new int[] { 7, 7, 7, 6, 6, 6, 6, 6, 6,
        //                    5, 5, 5, 4, 4, 4, 4, 3, 3, 3, 3, 2, 2, 2, 2,
        //                    2, 2, 2, 1, 1 };

        ////*************Split Material Layers 522 Elemement Mesh*****************
        //int[] materialPattern = new int[] { 17, 17, 17, 16, 16, 16, 16, 16, 16, 15,
        //    15, 15, 14, 14, 14, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1 };


        //************************322 Element Mesh*****************************
        //int[] xValues = new int[] { 0, 1, 2, 3, 4, 5, 6, 7, 8, 10,
        //    12, 18, 25, 32, 39 };
        //int[] yValues = new int[] { 0, 9, 15, 21, 24, 27, 30, 33, 35,
        //    36, 37, 38, 39, 40, 41,42, 43, 44, 45, 46, 47, 48, 49, 50 };
        //int[] materialPattern = new int[] { 7, 6, 6, 6, 6, 5, 5, 4, 4, 4,
        //                    3, 3, 3, 3, 2, 2, 2, 2, 2, 2, 2, 1, 1 };

        //************************2000 Element Mesh*****************************
        //int[] xValues = new int[] {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
        //    11, 12, 13, 14, 15,16, 17, 18, 19, 20, 21, 22, 23, 24, 25,
        //    26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40};
        //int[] yValues = new int[] {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
        //    11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25,
        //    26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40,
        //    41, 42, 43, 44, 45, 46, 47, 48, 49, 50};
        //int[] materialPattern = new int[] {7,7,7,7,7,7,7,7,7,6,6,6,6,6,6,6,6
        //    ,6,6,6,6,6,6,6,6,6,5,5,5,5,5,5,4,4,4,4,3,3,3,3,2,2,2,2,2,2,2,1,1};


        Matrix materialElementCode;

        public ModelConstants()
        {
            GenerateMaterialCodes();
        }
```

```
//Stripe material codes across all
//elements (vetically from bottom to top)
private void GenerateMaterialCodes()
{
    materialElementCode = new Matrix(this.NumElements, 2);
    int materialCounter = 0;

    for (int m = 0; m < this.NumElements; m++)
    {
        materialElementCode[m, 0] = m;
        materialElementCode[m, 1] = materialPattern[materialCounter];
        materialCounter++;
        if (materialCounter == yValues.Length-1)
        {
            materialCounter = 0;
        }
    }
}

/// <summary>
/// Returns array of x-coordinate values for model nodes.
/// </summary>
public int[] XValues
{
    get
    {
        return xValues;
    }
}
/// <summary>
/// Returns array of y-coordinate values for model nodes.
/// </summary>
public int[] YValues
{
    get
    {
        return yValues;
    }
}
/// <summary>
/// Returns the number of model elements.
/// </summary>
public int NumElements
{
    get
    {
        int n = xValues.Length - 1;
        int m = yValues.Length - 1;

        int numElements = n * m;

        return numElements;
    }
}
/// <summary>
/// Returns the number of model nodes.
/// </summary>
public int NumNodes
{
    get
    {
        int n = xValues.Length;
        int m = yValues.Length;

        int numNodes = n * m;

        return numNodes;
    }
}

public Matrix MaterialCodes
```

```
        {
            get
            {
                return materialElementCode;
            }
        }
    }
}
```

Class: MPaveAnalysis

```
using System;
using System.Collections.Generic;
using System.Text;
using MySql.Data.MySqlClient;
using System.Data;
using System.IO;

//
//Main class for analysis program.  Handles database access and
//distributes data into FE program.
//

namespace MPave
{
    class MPaveAnalysis
    {
        //Class variables
        DataTable WIMDataTable = new DataTable();
        DataTable wheelDataTable = new DataTable();
        DataTable environmentalDataTable = new DataTable();
        DataTable radialStrain = new DataTable();
        DataTable tangentialStrain = new DataTable();
        DataTable verticalPressure = new DataTable();
        DataTable verticalPressureBase = new DataTable();
        DataTable resilientModulus = new DataTable();
        DataTable stiffnessData = new DataTable();
        string yearWeek = "0811";

        //Creates event log
        TextWriter tx = new StreamWriter("E:\\...\\ErrorLog.txt");

        //Set up MySQL connection params
        MySqlConnection myConn, myConn2;
        MySqlDataAdapter da1, da2, da3, da4, da5;
        MySqlCommandBuilder cb1, cb2, cb3, cb4, cb5;
        string connString = "server=localhost;user id=*****;Password=*****;" +
            "persist security info=True;database=analysis;port=****";
        string connString2 = "server=localhost;user id=*****;Password=*****;" +
            "persist security info=True;database=fea_trucks;port=****";

        //Primary key in MySQL DB
        string wheelID;

        //Set poissons ratio for materials
        double[] poissonsRatio =
            new double[]{0.4, 0.4, 0.4, 0.4, 0.4, 0.4, 0.4, 0.4, 0.4, 0.4,
                0.4, 0.4, 0.35, 0.35, 0.35, 0.35, 0.35, 0.35, 0.35, 0.35, 0.35,
            0.35, 0.35, 0.35, 0.35, 0.35, 0.35, 0.35, 0.35, 0.35, 0.35, 0.35,
            0.35, 0.35, 0.35, 0.35, 0.35, 0.35};

        //Stores stiffness data for unbound layers
        double[] resilientModuli = new double[5];

        public void RunAnalysis()
        {
            GetDBData();
            EnvironmentDateTimeFixer();
            int vehicleCount = 0;
            int errorCount = 0;

            foreach (DataRow r in WIMDataTable.Rows)
            {
                string wimID = r["vehicle_number"].ToString();

                string selectFilter = "wim_id = " + wimID;
                string sortFilter = "date_time ASC";

                DataRow[] results = wheelDataTable.Select(selectFilter, sortFilter);
```

```
int numberOfWheels = results.Length;
if (results.Length != 0)
{
    wheelID = results[0]["wheel_id"].ToString(); //Beginning wheel ID

    DateTime dateTime = DateTime.Parse(r["date_time"].ToString());
    int month = dateTime.Month;
    setResilientModuli(month);

    //********Set layer temperatures for vehicle***************
    string envSelectFilter = "date_time >= '" + dateTime + "'";
    string envSortFilter = "date_time ASC";

    DataRow[] envResults =
        environmentalDataTable.Select(envSelectFilter,
        envSortFilter);

    double airTemp =
        Convert.ToDouble(envResults[0]["air_temperature"]) / 10;

    double[] layerTemperature = GetLayerTemperatures(airTemp);

    //********Develop layer stiffness values******************
    double vehicleSpeed = Convert.ToDouble(r["speed"]);
    double subGradeModulus = (resilientModuli[0] + resilientModuli[1]
        + resilientModuli[2] + resilientModuli[3]
        + resilientModuli[4]) / 5;
    //try
    //{
        for (int i = 0; i < numberOfWheels; i++)
        {
            //Get wheelID and determine if it is a single or dual
            int currentWheelID = Convert.ToInt32(wheelID) + i;
            bool single = dualOrSingle(currentWheelID);

            //Grab wheel load for current axle.
            int wheelLoadIndex = i + 1;
            string wheelLoadString =
                "wheel_load_" + wheelLoadIndex.ToString();
            double wheelLoadRaw =
                double.Parse(r[wheelLoadString].ToString());

            if (!single) { wheelLoadRaw = wheelLoadRaw / 2; }

            double wheelLoad =
                wheelLoadRaw * 1000 / 2; //Divide by 2 to
                                         //convert from axle load
                                         //to wheel load
            int vehicleClass = Convert.ToInt32(r["class"]);

            //For debugging.
            //Console.WriteLine("Wheel ID = {0};  WIM ID = {1}",
            //    wheelID, wimID);

            double contactRadius =
                GetContactRadius(wheelLoad, vehicleClass);

            double[] systemStiffness =
                CalculateStiffness(layerTemperature, vehicleSpeed,
                subGradeModulus, contactRadius, currentWheelID);

            if (wheelLoadRaw != 0)
            {
                //*********nter FE Analysis*******************

                //Added wheelLoad divider of 2 because of
                //doubling from half of Contact Area. 20090509
                Analysis newFEA =
                    new Analysis(contactRadius, wheelLoad / 2,
                        systemStiffness, poissonsRatio);
```

```
//********Comment Out To Skip FE***************
newFEA.StartAnalysis();

DataRow newRow = radialStrain.NewRow();
double[] radialHStrain =
    newFEA.GetBottomHMARadialStrain();

DataRow newRow2 = verticalPressure.NewRow();
double[] verticalPres =
    newFEA.GetVerticalPressure();

DataRow newRow4 = verticalPressureBase.NewRow();
double[] verticalPresBase =
    newFEA.GetBaseVerticalPressure();

DataRow newRow5 = tangentialStrain.NewRow();
double[] tangentialHStrain =
    newFEA.GetBottomHMATangentialStrain();


for (int n = 0; n < radialHStrain.Length; n++)
{
    newRow[n + 1] = radialHStrain[n];
    newRow2[n + 1] = verticalPres[n];
    newRow4[n + 1] = verticalPresBase[n];
    newRow5[n + 1] = tangentialHStrain[n];

    if (double.IsNaN(radialHStrain[n]))
    {
        string errTxt =
            "rStrain NaN, Column " +
            n.ToString() + " wheel_id = " +
            wheelID.ToString();

        tx.WriteLine(errTxt);
        tx.Flush();
        newRow[n + 1] = 0;
    }
    if (double.IsNaN(tangentialHStrain[n]))
    {
        string errTxt = "tStrain NaN, Column "
            + n.ToString() + " wheel_id = " +
            wheelID.ToString();

        tx.WriteLine(errTxt);
        tx.Flush();
        newRow5[n + 1] = 0;
    }
    if (double.IsNaN(verticalPres[n]))
    {
        string errTxt =
            "Vertical Press NaN, Column " +
            n.ToString() + " wheel_id = " +
            wheelID.ToString();

        tx.WriteLine(errTxt);
        tx.Flush();
        newRow2[n + 1] = 0;
    }
    if (double.IsNaN(verticalPresBase[n]))
    {
        string errTxt =
            "Vertical Press Base NaN, Column "
            + n.ToString() + " wheel_id = " +
            wheelID.ToString();

        tx.WriteLine(errTxt);
        tx.Flush();
        newRow4[n + 1] = 0;
    }
}
```

```
                            //Console.ReadLine();
                            int wheelIdentification =
                                Convert.ToInt32(wheelID) + i;

                            newRow[0] = wheelIdentification;
                            newRow2[0] = wheelIdentification;
                            newRow4[0] = wheelIdentification;
                            newRow5[0] = wheelIdentification;

                            if (!single)
                            {
                                newRow = SuperImposeLoads(newRow);
                                newRow2 = SuperImposeLoads(newRow2);
                                newRow4 = SuperImposeLoads(newRow4);
                                newRow5 = SuperImposeLoads(newRow5);
                            }

                            radialStrain.Rows.Add(newRow);
                            verticalPressure.Rows.Add(newRow2);
                            verticalPressureBase.Rows.Add(newRow4);
                            tangentialStrain.Rows.Add(newRow5);

                            try
                            {
                            da1.Update(radialStrain);
                            da2.Update(verticalPressure);
                            da4.Update(verticalPressureBase);
                            da5.Update(tangentialStrain);

                            //******End Comment for FE Skip*****************

                                //Update stiffness data
                                DataRow newRow3 = stiffnessData.NewRow();
                                for (int n = 0; n < systemStiffness.Length; n++)
                                {
                                    newRow3[n + 1] = systemStiffness[n];
                                }

                                newRow3[0] = Convert.ToInt32(wheelID) + i;
                                stiffnessData.Rows.Add(newRow3);
                                da3.Update(stiffnessData);

                            }
                            catch
                            {
                                Console.WriteLine(
                                    "Error updating database.  Wheel ID = " +
                                    currentWheelID.ToString());
                            }

                        }
                    }

                Console.WriteLine("Vehicle Count: " + vehicleCount.ToString());
                vehicleCount++;
            }
        }

        Console.ReadLine();
    }

    //
    //Calculate layer stiffness values
    //

    private double[] CalculateStiffness(double[] layerTemperature,
        double vehicleSpeed, double subGradeModulus,
        double contactRadius, int currentWheelID)
    {
        MySQLConn myConn = new MySQLConn(yearWeek, connString);
```

```
        //Determine if the axle is part of a tandem group
        bool axleGroup =
            myConn.GetSingleGroup(currentWheelID.ToString());

        METIterator solveHMAStiffness = new METIterator(layerTemperature,
            vehicleSpeed, subGradeModulus, contactRadius, axleGroup);

        double[] hMAStiffness = solveHMAStiffness.SolveStiffness();

        //Assemble pavement system stiffness array.
        double[] systemStiffness = new double[18];

        for (int m = 0; m < hMAStiffness.Length; m++)
        {
            systemStiffness[m] = hMAStiffness[m];
        }

        systemStiffness[13] = resilientModuli[0]; //Open graded
        systemStiffness[14] = resilientModuli[1]; //Dense graded
        systemStiffness[15] = resilientModuli[2]; //Select
        systemStiffness[16] = resilientModuli[3]; //Native soil - upper
        systemStiffness[17] = resilientModuli[4]; //Native soil - lower

        return systemStiffness;
    }

    //
    //Calculate contact radius for wheel load
    //

    private double GetContactRadius(double wheelLoad, int vehicleClass)
    {
        int pressure = 75; //psi
        if (vehicleClass < 4)
        {
            pressure = 35;
        }
        //Console.WriteLine("Wheel_ID: " + wheelID.ToString()
        //    + "\t Class: " + vehicleClass.ToString());
        //Console.WriteLine("Initial Pressure: " + pressure.ToString());
        double contactArea = wheelLoad / pressure;
        double contactRadius = Math.Sqrt(contactArea / Math.PI);

        //For debugging
        //Console.WriteLine("Pressure = {0}", pressure.ToString());
        //Console.WriteLine("Wheel Load = {0}", wheelLoad.ToString());
        //Console.WriteLine("CA = {0}", contactArea.ToString());
        //Console.WriteLine("CR = {0}", contactRadius.ToString());
        //Console.ReadLine();

        return contactRadius;
    }

    //
    //Calculate layer temperatures based on data and regression models
    //

    private double[] GetLayerTemperatures(double airTemp)
    {
        double[] temperature = new double[13];
        double depth = 0.5; //inches, incremented by 1 in to meet mid-depth
                            //of each HMA layer

        //Console.WriteLine("Air Temp: " + airTemp.ToString());
        for (int t = 0; t < 13; t++)
        {
            double temp = 8.6953 + 0.8365 * airTemp + 0.0720 * depth;

            temperature[t] = temp;
            //Console.WriteLine("Temperature: " +
            //    temperature[t].ToString() + "\tDepth: " + depth.ToString());
```

```
            depth += 1.0;
        }
        return temperature;
    }

    //Fixes format of date and time in environment DB
    private void EnvironmentDateTimeFixer()
    {
        environmentalDataTable.Columns.Add("date_time", typeof(DateTime));

        foreach (DataRow r in environmentalDataTable.Rows)
        {
            DateTime date = DateTime.Parse(r["date"].ToString());
            TimeSpan time = TimeSpan.Parse(r["time"].ToString());

            date = date.Add(time);
            r["date_time"] = date;
        }
    }

    #region GetDBData
    /// <summary>
    /// Gets wheel, WIM, environmental, and load-time data.
    /// </summary>
    private void GetDBData()
    {
        MySQLConn connection = new MySQLConn(yearWeek, connString);
        //wheelWIMDataTable = connection.GetWheelWIMData();
        //loadTimeDataTable = connection.GetLoadTimeData();
        WIMDataTable = connection.GetWIMData();
        wheelDataTable = connection.GetWheelData();
        environmentalDataTable = connection.GetEnvironmentalData();

        myConn = new MySqlConnection(connString);
        myConn2 = new MySqlConnection(connString2);

        da1 = new MySqlDataAdapter();
        cb1 = new MySqlCommandBuilder(da1);
        string selectString = "SELECT * FROM `radial_strain" +
            yearWeek + "`";
        da1.SelectCommand = new MySqlCommand(selectString, myConn2);
        da1.Fill(radialStrain);

        da2 = new MySqlDataAdapter();
        cb2 = new MySqlCommandBuilder(da2);
        string selectString2 = "SELECT * FROM `vertical_pressure" +
            yearWeek + "`";
        da2.SelectCommand = new MySqlCommand(selectString2, myConn2);
        da2.Fill(verticalPressure);

        da4 = new MySqlDataAdapter();
        cb4 = new MySqlCommandBuilder(da4);
        string selectString5 = "SELECT * FROM `vertical_pressure_base" +
            yearWeek + "`";
        da4.SelectCommand = new MySqlCommand(selectString5, myConn2);
        da4.Fill(verticalPressureBase);

        da3 = new MySqlDataAdapter();
        cb3 = new MySqlCommandBuilder(da3);
        string selectString3 = "SELECT * FROM `modulus`";
        da3.SelectCommand = new MySqlCommand(selectString3, myConn);
        da3.Fill(resilientModulus);

        string selectString4 = "SELECT * FROM `modulus" + yearWeek + "`";
        da3.SelectCommand = new MySqlCommand(selectString4, myConn2);
        da3.Fill(stiffnessData);

        da5 = new MySqlDataAdapter();
        cb5 = new MySqlCommandBuilder(da5);
        string selectString6 = "SELECT * FROM `tangential_strain" +
            yearWeek + "`";
```

```
    da5.SelectCommand = new MySqlCommand(selectString6, myConn2);
    da5.Fill(tangentialStrain);

}
#endregion

//Set resilient modulus based on preset monthly averaged values
private void setResilientModuli(int month)
{
    string selectString = "month = " + month;
    DataRow[] selectedRows = resilientModulus.Select(selectString);

    resilientModuli[0] = (double)selectedRows[0]["open"];
    resilientModuli[1] = (double)selectedRows[0]["dense"];
    resilientModuli[2] = (double)selectedRows[0]["rselect"];
    resilientModuli[3] = (double)selectedRows[0]["upper_subgrade"];
    resilientModuli[4] = (double)selectedRows[0]["lower_subgrade"];
}

//Determines if wheel is a dual or single
private bool dualOrSingle(int wheelID)
{
    DataTable singleDual = new DataTable();
    string myQuery = "SELECT * FROM `single_group" + yearWeek +
        "` WHERE wheel_id = " + wheelID;

    MySqlConnection myConnection = new MySqlConnection(connString);
    MySqlDataAdapter myAdapter =
        new MySqlDataAdapter(myQuery, myConnection);

    myAdapter.Fill(singleDual);

    int singleorDual = 0;
    if (singleDual.Rows.Count != 0)
    {
        singleorDual = (int)singleDual.Rows[0]["singleGroup"];
    }

    bool single = true;
    if (singleorDual == 1) { single = false; }

    return single;
}

//Superimposes loads in the case of duals
private DataRow SuperImposeLoads(DataRow dataRow)
{
    double[] superimposedValues = new double[radialStrain.Columns.Count];
    double[] dataRowValues = new double[superimposedValues.Length];
    for (int n = 1; n < superimposedValues.Length; n++)
    {
        dataRowValues[n] = Convert.ToDouble(dataRow[n].ToString());
    }

    superimposedValues[1] = dataRowValues[1] + dataRowValues[11];
    superimposedValues[2] = dataRowValues[2] + (dataRowValues[10] +
        dataRowValues[11]) / 2;
    superimposedValues[3] = dataRowValues[3] + dataRowValues[10];
    superimposedValues[4] = dataRowValues[4] + (dataRowValues[9] +
        dataRowValues[10]) / 2;
    superimposedValues[5] = dataRowValues[5] + dataRowValues[9];
    superimposedValues[6] = dataRowValues[6] + dataRowValues[8];
    superimposedValues[7] = dataRowValues[7] + dataRowValues[7];
    superimposedValues[8] = dataRowValues[8] + dataRowValues[6];
    superimposedValues[9] = dataRowValues[9] + dataRowValues[5];
    superimposedValues[10] = dataRowValues[10] + dataRowValues[3];
    superimposedValues[11] = dataRowValues[11] + dataRowValues[1];
    superimposedValues[12] = dataRowValues[12] + dataRowValues[3];
    superimposedValues[13] = dataRowValues[13] + dataRowValues[5];
    superimposedValues[14] = dataRowValues[14] + dataRowValues[7];
    superimposedValues[15] = dataRowValues[15] + dataRowValues[9];
```

```
            superimposedValues[16] = dataRowValues[16] + dataRowValues[10];
            superimposedValues[17] = dataRowValues[17] + dataRowValues[11];
            superimposedValues[18] = dataRowValues[18] + dataRowValues[12];
            superimposedValues[19] = dataRowValues[19] + dataRowValues[13];

            for (int m = 1; m < superimposedValues.Length; m++)
            {
                dataRow[m] = superimposedValues[m];
            }

            return dataRow;
        }
    }
}
```

Class: SparseSolver

```
using System;
using System.Collections.Generic;
using System.Text;
using dnAnalytics.LinearAlgebra;
using dnAnalytics.LinearAlgebra.Solvers;

//
//Operates the dnAnalytics Sparse Solver
//

namespace MPave
{
    class SparseSolver
    {
        public SparseSolver()
        {

        }

        //Solve AX = B linear system
        public double[] InvertSolver(double[,] matrix, double[] rightHandSideVector)
        {

            IPreConditioner precondtioner =
                new dnAnalytics.LinearAlgebra.Solvers.Preconditioners.Diagonal();

            dnAnalytics.LinearAlgebra.Solvers.Iterative.GPBiCG solver =
                new dnAnalytics.LinearAlgebra.Solvers.Iterative.GPBiCG(precondtioner);

            SparseMatrix lhSideMatrix = new SparseMatrix(matrix);
            Vector rhSideVector = new DenseVector(rightHandSideVector);

            Vector solutionVector = solver.Solve(lhSideMatrix, rhSideVector);

            double[] solutionVectorArray = solutionVector.ToArray();
            return solutionVectorArray;
        }
    }
}
```

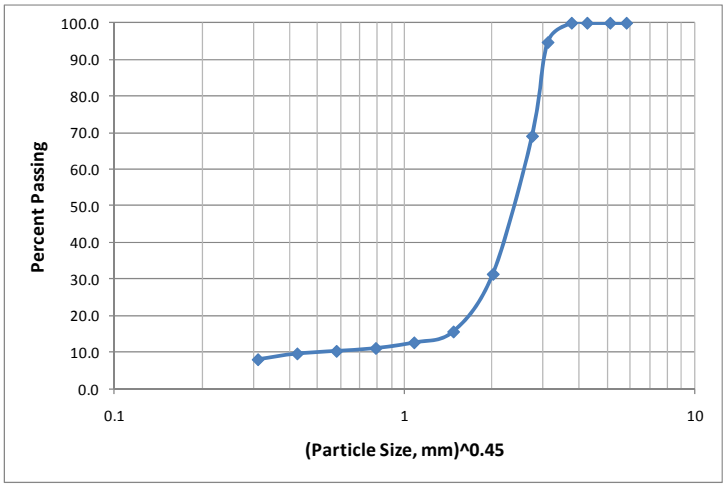
Appendix B - Material Properties

The following contains property data for the materials used to construct the pavement test section. During construction the materials were sampled and stored for testing at a later time. Laboratory testing of these materials was conducted as part of the overall research project and details can be found in the corresponding project report (Crovetti et al., 2008). The following data represents a summary of the pertinent material properties.

B.0 - SMA Mix Design Data – Upper Layer

Mix Type:	SMA
Mix Size:	12.5 mm
Mix Temperature:	155°C - 165°C

Aggregate Data	
Aggregate	% Blend
AGG#1: 1/2" Chip	35.0
AGG#2: 3/8" Chip	23.0
AGG#3: 1/4" Chip	21.0
AGG#4: 1/4" Screenings	16.0
AGG#5: Superfine	5.0



Mix Design Data

Compative Effort	Ni: 8	Nd: 100	Nm: 160		
Binder Data	Grade: 70-28		Srce: CRM, Milwaukee	Gb = 1.025	

Aggregate Gradation

%Blend		AGG#1	AGG#2	AGG#3	AGG#4	AGG#5	JMF
		35.0	23.0	21.0	16.0	5.0	
2	50.0 mm	100.0	100.0	100.0	100.0	100.0	100.0
1-1/2	37.5 mm	100.0	100.0	100.0	100.0	100.0	100.0
1	25.0 mm	100.0	100.0	100.0	100.0	100.0	100.0
3/4	19.0 mm	100.0	100.0	100.0	100.0	100.0	100.0
1/2	12.5 mm	85.0	100.0	100.0	100.0	100.0	94.8
3/8	9.5 mm	15.0	95.0	100.0	100.0	100.0	69.1
#4	4.75 mm	4.2	4.8	45.0	89.6	100.0	31.4
#8	2.36 mm	2.9	2.8	3.2	51.9	100.0	15.6
#16	1.18 mm	2.3	2.4	2.8	35.7	100.0	12.7
#30	0.60 mm	2.1	2.3	2.7	26.8	100.0	11.1
#50	0.30 mm	2.1	2.2	2.6	22.1	100.0	10.3
#100	0.15 mm	2.0	2.2	2.5	18.6	98.6	9.6
#200	0.075 mm	1.9	2.1	2.4	14.7	80.7	8.0
	FAA	0.0	0.0	0.0	48.7	0.0	48.7
	Gsb	2.764	2.741	2.736	2.756	2.717	2.749

**0.3% organic fibers (%wt. of mix) added.

Aggregate Data for Blended Design JMF

Crush 1F/2F:	100.1 / 100.1	Gsb:	2.749	Absorption:	1.1	LA Wear:	0.0 (100)
FAA:	48.7	Gsa:	2.836	Dust %:	1.5		0.0 (500)
SE:	69	Gse:	2.814	Soundness:	0.0	Frz-Thaw:	0.0
Elongated:	6.6 (3:1)						

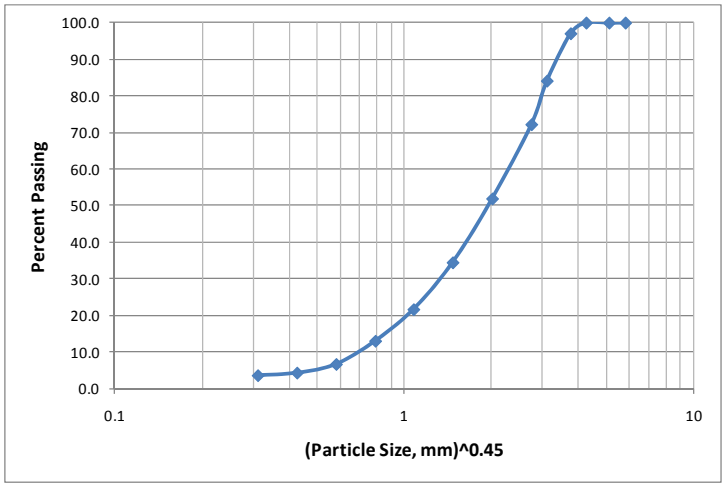
Volumetric Data

	Total Pb	Gmm	Gmb	Va	VMA	VFB	Unit Wt.
JMF	6.1	2.543	2.442	4.0	16.6	75.9	2435

B.1 - E-30 Mix Design Data – Intermediate Layers

Mix Type:	E-30
Mix Size:	19.0 mm
Design ESAL Range (mil):	10 to < 30
Mix Temperature:	135°C - 149°C

Aggregate Source Data	
Aggregate	% Blend
AGG#1: #1 Stone	15.0
AGG#2: 1/2" Chip	15.0
AGG#3: 3/8" Chip	15.0
AGG#4: 1/4" Minus	5.0
AGG#5: MFG'D Sand	40.0
AGG#6: Natural Sand	10.0



Mix Design Data			
Compative Effort	Ni: 8	Nd: 100 Nm: 160	
Binder Data	Grade: PG 64-22	Srce: BP Amoco (PAD)	Gb = 1.020

Aggregate Gradation

%Blend		AGG#1	AGG#2	AGG#3	AGG#4	AGG#5	AGG#6	JMF
		15.0	15.0	15.0	5.0	40.0	10.0	
2	50.0 mm	100.0	100.0	100.0	100.0	100.0	100.0	100.0
1-1/2	37.5 mm	100.0	100.0	100.0	100.0	100.0	100.0	100.0
1	25.0 mm	100.0	100.0	100.0	100.0	100.0	100.0	100.0
3/4	19.0 mm	80.5	100.0	100.0	100.0	100.0	100.0	97.1
1/2	12.5 mm	10.2	84.1	100.0	100.0	100.0	100.0	84.1
3/8	9.5 mm	3.3	17.5	94.1	100.0	100.0	100.0	72.2
#4	4.75 mm	2.2	2.0	1.7	89.5	94.8	86.8	52.0
#8	2.36 mm	1.9	1.9	1.3	59.4	58.8	72.6	34.5
#16	1.18 mm	1.8	1.8	1.2	40.1	32.6	59.5	21.7
#30	0.60 mm	1.8	1.8	1.2	28.4	16.9	41.8	13.1
#50	0.30 mm	1.7	1.8	1.2	22.1	8.5	15.2	6.7
#100	0.15 mm	1.7	1.7	1.2	19.0	5.4	5.7	4.4
#200	0.075 mm	1.6	1.7	1.2	16.6	4.4	4.1	3.7
	FAA	0.0	0.0	0.0	48.4	47.4	41.1	46.3
	Gsb	2.740	2.750	2.770	2.760	2.760	2.660	2.750

**P-0.075mm increased 0.5% at the request of the Project Manager. P-0.075mm JMF = 4.2%

Aggregate Data for Blended Design JMF

Crush 1F/2F:	99.2 / 99.1	Gsb:	2.750	Absorption:	1.0	LA Wear:	0.0 (100)
FAA:	46.3	Gsa:	2.820	Dust %:	0.9		0.0 (500)
SE:	90	Gse:	2.790	Soundness:	0.0	Frz-Thaw:	0.0
Elongated:	3.7 (5:1)						

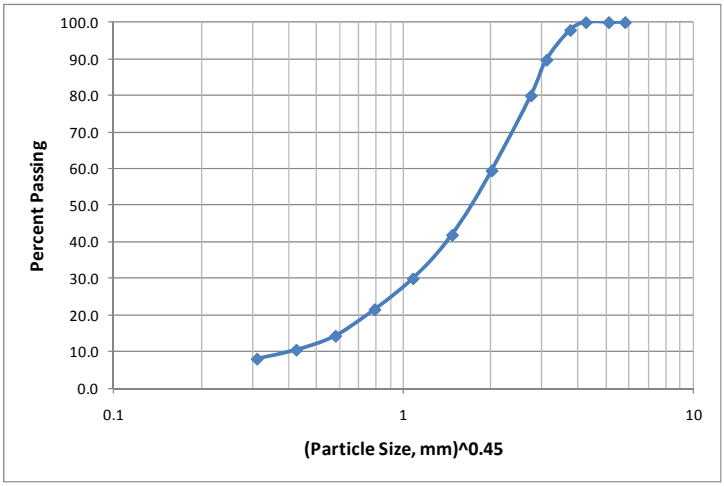
Volumetric Data

	Total Pb	Gmm	Gmb	Va	VMA	VFB	Unit Wt.
JMF	4.6	2.590	2.480	4.0	13.7	70.8	2482

B.2 - C2 Mix Design Data – Lower Layer

Mix Type:	C2
Mix Size:	19.0 mm
Design ESAL Range (mil):	NA
Mix Temperature:	135°C - 149°C

Aggregate Source Data	
Aggregate	% Blend
AGG#1: RAP	30.0
AGG#2: #1 Stone	10.0
AGG#3: 1/2" Chip	10.0
AGG#4: 3/8" Chip	10.0
AGG#5: 1/4" Minus	20.0
AGG#6: MFG'D Sand	15.0
AGG#7: Natural Sand	5.0



Mix Design Data				
Compative Effort	Ni: 0	Nd: 40	Nm: 0	
Binder Data	Grade: PG 64-22	Srce: CRM, Milwaukee	Gb = 1.028	RAP Pb = 4.47

Aggregate Gradation

%Blend		AGG#1	AGG#2	AGG#3	AGG#4	AGG#5	AGG#6	AGG#7	JMF
		30.0	10.0	10.0	10.0	20.0	15.0	5.0	
2	50.0 mm	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
1-1/2	37.5 mm	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
1	25.0 mm	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
3/4	19.0 mm	100.0	78.7	100.0	100.0	100.0	100.0	100.0	97.9
1/2	12.5 mm	100.0	9.4	87.9	100.0	100.0	100.0	100.0	89.7
3/8	9.5 mm	96.0	4.0	11.9	95.6	100.0	100.0	100.0	80.0
#4	4.75 mm	78.0	3.1	1.1	2.1	87.4	89.5	90.7	59.5
#8	2.36 mm	59.1	2.5	1.1	1.5	57.6	55.8	74.7	41.9
#16	1.18 mm	45.0	2.1	1.0	1.4	40.9	32.2	59.4	29.9
#30	0.60 mm	33.6	2.1	1.0	1.3	30.7	18.5	41.1	21.5
#50	0.30 mm	21.4	2.1	1.0	1.3	25.1	10.6	15.8	14.3
#100	0.15 mm	15.2	2.0	0.9	1.3	21.0	6.4	5.9	10.4
#200	0.075 mm	12.1	1.8	0.9	1.2	15.5	4.4	4.3	8.0
	FAA	42.7	0.0	0.0	0.0	49.2	46.5	41.1	45.3
	Gsb	2.673	2.776	2.760	2.762	2.772	2.756	2.653	2.731

Aggregate Data for Blended Design JMF

Crush 1F/2F:	97.9 / 96.8	Gsb:	2.731	Absorption:	0.9	LA Wear:	0.0 (100)
FAA:	45.3	Gsa:	2.800	Dust %:	2.2		0.0 (500)
SE:	74	Gse:	2.793	Soundness:	0.0	Frz-Thaw:	0.0
Elongated:	1.0 (5:1)						

Volumetric Data

	Added Pb	Total Pb	Gmm	Gmb	Va	VMA	VFB	Unit Wt.
JMF	3.2	4.5	2.592	2.494	3.8	12.8	70.3	2487

B.3 - Open Graded Aggregate Material Properties

Table 0-1 – Gradation and properties for Open Graded.

Sieve			% Finer
Mesh	Opening, in.	Opening, mm	
1"	1	25.4	100
3/4"	0.75	19.0	97.7
3/8"	0.375	9.51	62.1
No. 4	0.187	4.76	26.7
No. 10	0.0787	2.00	3.8
No. 20	0.0331	0.841	3.1
No. 40	0.0165	0.420	3
No. 60	0.0098	0.250	2.8
No. 140	0.0041	0.105	2.6
No. 200	0.0029	0.074	2.4

Bulk Specific Gravity	2.743
Apparent Specific Gravity	2.809
Absorption, %	1.34
Max Dry Unit Weight, pcf	128.4
D_{10}	2.75
D_{30}	5
D_{60}	9
C_u	3.27
C_c	1.01

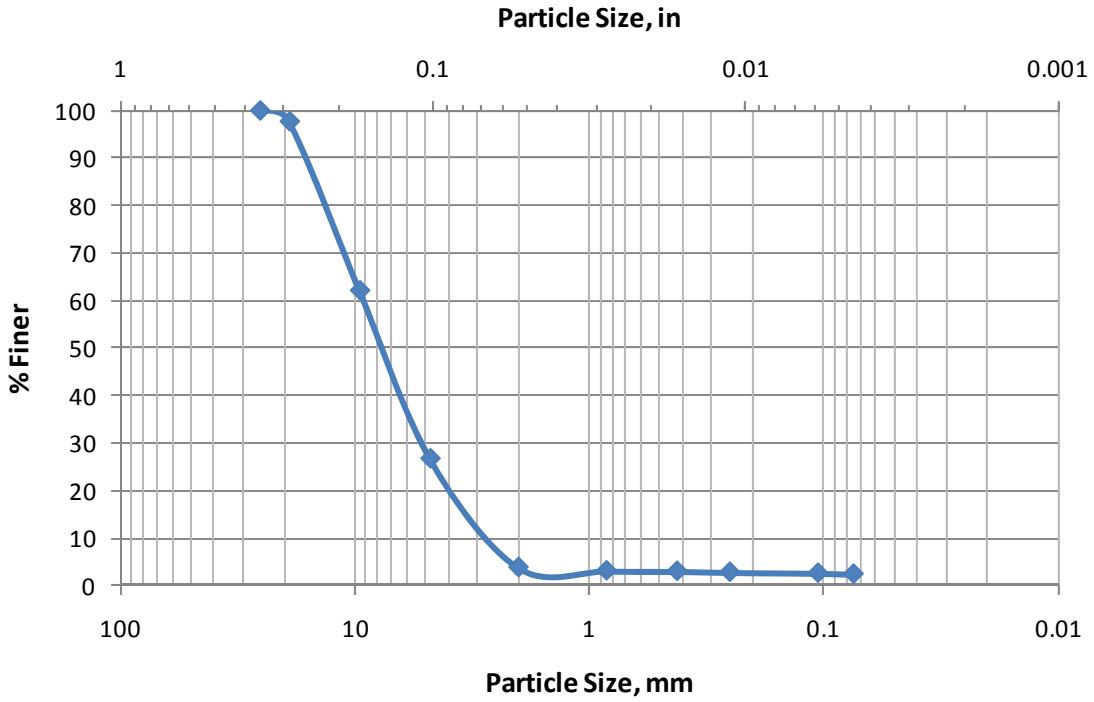


Figure 0-1 - Open Graded particle size distribution chart.

B.4 - Dense Graded Aggregate Properties

Table 0-2 - Gradation and properties for Dense Graded.

Sieve			% Finer
Mesh	Opening, in.	Opening, mm	
1"	1	25.4	99.5
3/4"	0.75	19.0	90.2
3/8"	0.375	9.51	64.5
No. 4	0.187	4.76	35.8
No. 10	0.0787	2.00	22.2
No. 20	0.0331	0.841	13.9
No. 40	0.0165	0.420	8.2
No. 60	0.0098	0.250	4.8
No. 140	0.0041	0.105	2.3
No. 200	0.0029	0.074	1.7

Bulk Specific Gravity	2.396
Apparent Specific Gravity	2.594
Absorption, %	5.47
Max Dry Unit Weight, pcf	121.1
D_{10}	0.55
D_{30}	3.6
D_{60}	8.5
C_u	15.45
C_c	2.77

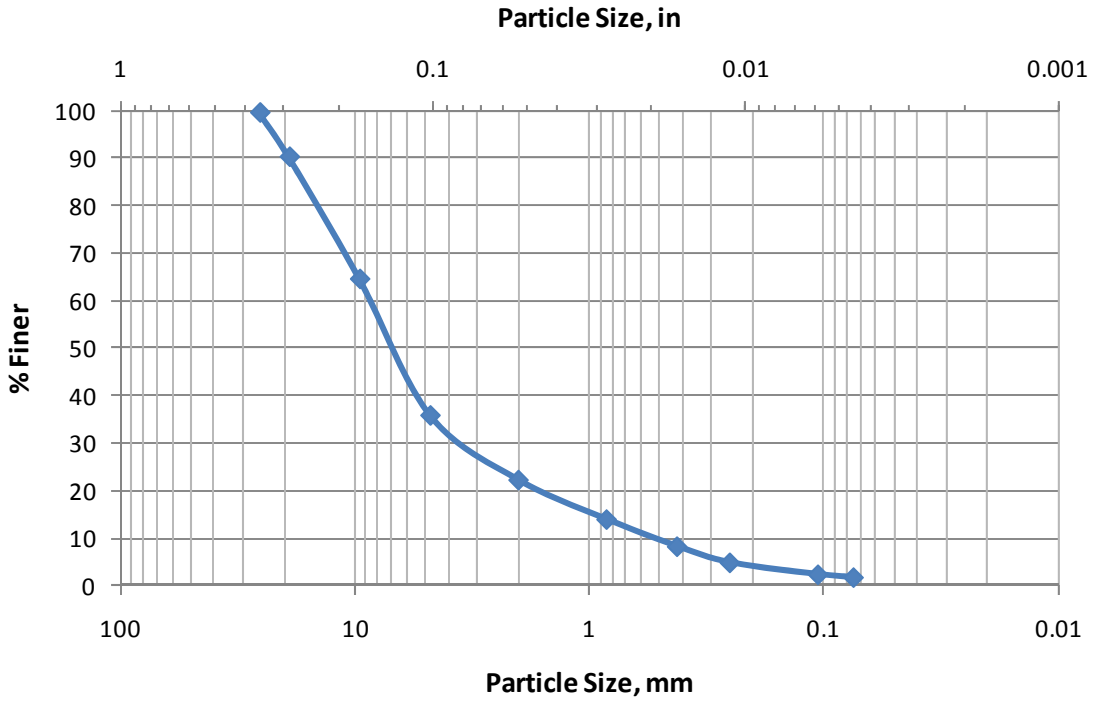


Figure 0-2 - Dense Graded particle size distribution chart.

B.5 - Native Soil Layer Properties

Table 0-3 - General soil properties.

Soil	Liquid Limit, %	Plasticity Index, %	Shrinkage Limit, %	Specific Gravity	Optimum Moisture Content, %	Max Dry Unit Weight, pcf	USCS Classification	Group Index	AASHTO Classification
1A	17	3	14.445	2.705	10	127.5	ML / CL-ML Silt with Sand / Silty Clay with Sand	0	A-4
1B	24	8	2	2.870	12.5	119	CL / CL-ML Lean Clay / Silty Clay with Sand	5	A-4
2	NP	NP	NP	2.725	10.5	118.55	ML Sandy Silt	0	A-4

Table 0-4 - Location and elevation soil properties.

Soil	43NB - Station, ft	Center - Offset, ft.	Elevation, ft.	Specific Gravity		Optimum Gravitmetric Moisture Content, %		Degree of Saturation at Optimum M.C., %		Maximum Dry Unit Weight, PCF		USCS Classification	Group Index	AASHTO Classification
Soil 1A	385+26.43	64.61 R	656.0' to 657.0'	2.69	Average	10.0%	Average	86.4%	Average	128.0	Average	ML	0	A-4
				2.72	2.705	10.0%	10.0%	80.8%	83.6%	127.0	127.5	CL-ML	0	A-4
Soil 1B	385+26.43	64.61 R	655.0' to 656.0'	2.86	Average	12.5%	Average	73.4%	Average	120.0	Average	CL	5	A-4
				2.88	2.870	12.5%	12.5%	68.8%	71.1%	118.0	119	CL-ML	5	A-4
Soil 2	385+40.84	63.67 R	654.4' to 656.4'	2.75	Average	10.5%	Average	65.9%	Average	119.3	Average	ML	0	A-4
				2.7	2.725	10.5%	10.5%	65.9%	65.9%	117.8	118.55	ML	0	A-4

Table 0-5 – Upper layer soils gravimetric and volumetric computations.

Property	Soil 1A and Soil 2 - Upper Soils		
	Test #1	Test #2	Average
Unit Weight, PCF	143.7	144.6	144.15
Moisture Unit Weight, PCF	19.8	21.3	20.55
Moisture Conent, %	16.0%	17.3%	16.6%
Dry Unit Weight, PCF	123.9	123.3	123.6
Specific Gravity	2.715		
Porosity, = V_v / V_T × 100%	= 0.3301 / 1.0 =		33.0%

144.2 PCF			1 ft^{3*}	
0	A	0.0000	0.3301	1.0597
20.6	W	0.3301		
123.6	S	0.7296		
*Assumed volume.				
**Sample assumed saturated.				

Table 0-6 – Lower layer soils gravimetric and volumetric computations.

Soil 1B - Lower Soils		
Property	Test #1	
Unit Weight, PCF	146.2	
Moisture Unit Weight, PCF	18.8	
Moisture Conent, %	14.8%	
Dry Unit Weight, PCF	127.4	
Specific Gravity	2.87	
Porosity, = V_v / V_T × 100%	= 0.3013 / 1.0 =	30.1%

146.2 PCF			1 ft^{3*}	
0	A	0.0000	0.3013	1.0127
18.8	W	0.3013		
127.4	S	0.7114		
*Assumed volume.				
**Sample assumed saturated.				

Table 0-7 – Soil property key for Tables 5 and 6 above.

Total Weight, PCF			Total Volume, ft^3	
Wt. Air	A	Vol. Air	Vol. Voids	Calc. Total Vol.
Wt. H_20	W	Vol. H_20		
Wt. Soil	S	Vol. Soil		

Table 0-8 - Lower soil layer resilient modulus calculations.

Lower - MrOpt = 5,261 psi, Sopt = 71.1%					
Month	**θ - A0**	**S**	*Denom*	*POW*	**Native**
January	39.6	0.919615	4.981882	-0.18939	3625
February	39.2	0.910326	4.737599	-0.18306	3679
March	39.4	0.91497	4.857807	-0.18626	3652
April	40.2	0.933548	5.378567	-0.19846	3550
May	41.6	0.96606	6.46471	-0.21759	3397
June	43.4	1.00786	8.266043	-0.23823	3240
July	44.9	1.042694	10.21313	-0.25235	3136
August	45.9	1.065917	11.7931	-0.26038	3079
September	45.6	1.05895	11.29259	-0.25808	3095
October	41.3	0.959093	6.211291	-0.21372	3428
November	40.6	0.942837	5.664744	-0.20421	3504
December	39.5	0.917292	4.919354	-0.18783	3638

Table 0-9 - Upper soil layer resilient modulus calculations.

Upper - MrOpt = 11,642 psi, Sopt = 74.8%					
Month	**θ - A2**	**S**	*Denom*	*POW*	**Native**
January	37.4	0.759333	2.0378	-0.0118	11806
February	36.9	0.749182	1.9685	-0.0012	12097
March	37.3	0.757303	2.0236	-0.0097	11864
April	38.7	0.785727	2.2424	-0.0392	11084
May	40.9	0.830394	2.6845	-0.0842	9994
June	43.6	0.885212	3.4476	-0.1347	8897
July	45.8	0.929879	4.3186	-0.1705	8192
August	47	0.954242	4.9180	-0.1878	7873
September	46.5	0.944091	4.6561	-0.1808	8001
October	41.5	0.842576	2.8303	-0.0960	9727
November	39.8	0.808061	2.4467	-0.0620	10517
December	37.9	0.769485	2.1122	-0.0224	11522

Table 0-10 – Resilient modulus summary for all unbound layers.

Mr Summary							
Month	**θ - A2**	**S**	**Open**	**Dense**	**Select**	**Native Upper**	**Native Lower**
January	37.4	0.759333	4523	12683	30000	11805	3625
February	36.9	0.749182	4616	12942	30000	12097	3678
March	37.3	0.757303	4523	12683	30000	11863	3651
April	38.7	0.785727	4431	12424	30000	11083	3550
May	40.9	0.830394	4339	12165	30000	9993	3397
June	43.6	0.885212	4246	11906	30000	8896	3239
July	45.8	0.929879	4154	11647	30000	8192	3136
August	47	0.954242	4062	11388	30000	7872	3078
September	46.5	0.944091	4154	11647	30000	8000	3094
October	41.5	0.842576	4246	11906	30000	9726	3427
November	39.8	0.808061	4339	12165	30000	10517	3503
December	37.9	0.769485	4431	12424	30000	11521	3638