DOT/FAA/TC-22/44

Federal Aviation Administration
William J. Hughes Technical Center
Aviation Research Division
Atlantic City International Airport
New Jersey 08405

Machine Learning Solutions for Top-Down Cracking Design of Airport Rigid Pavement

November 2022

Final Report

This document is available to the U.S. public through the National Technical Information Service (NTIS), Springfield, Virginia 22161.

This document is also available from the Federal Aviation Administration William J. Hughes Technical Center at actlibrary.tc.faa.gov.

U.S. Department of Transportation
Federal Aviation Administration

NOTICE

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
DOT/FAA/TC-22/44		

4. Title and Subtitle	5. Report Date
MACHINE LEARNING SOLUTIONS FOR TOP-DOWN CRACKING DESIGN OF AIRPORT RIGID PAVEMENT	November 2022
	6. Performing Organization Code
7. Author(s) Ali Z. Ashtiani, Thomas Paniagua, Timothy Parsons, and Greg Foderaro	8. Performing Organization Report No.
9. Performing Organization Name and Address Applied Research Associates, Inc. 2628 Fire Rd Egg Harbor Township, NJ 08234	10. Work Unit No. (TRAIS)
	11. Contract or Grant No. 692M15-20-T-00033
12. Sponsoring Agency Name and Address U.S. Department of Transportation Federal Aviation Administration Airport Engineering Division 800 Independence Ave., SW Washington, D.C. 20591	13. Type of Report and Period Covered Final Report
	14. Sponsoring Agency Code AAS-100

15. Supplementary Notes
The Federal Aviation Administration Aviation Research Division Contracting Officer Representative (COR) was Dr. David Brill.

16. Abstract
The Federal Aviation Administration (FAA) rigid pavement design process is based on bottom-up cracking failure resulting from tensile stress at the bottom of a flat slab under aircraft loads. The FAA has a long-term goal to add top-down cracking failure mode to the FAA Rigid and Flexible Iterative Elastic Layered Design (FAARFIELD) program. The existing design procedure is not suitable to support design for the top-down cracking failure mode. Critical stresses for rigid pavement design can be calculated by Finite Element Analysis – FAA (FEAFAA), the FAA three-dimensional finite element (3D-FE) program. However, direct use of 3D-FE methods in design software is typically far more time-consuming than is acceptable for design procedures.

The objective of this research was to develop machine learning (ML) solutions to support design of airfield rigid pavements to resist top-down cracking. The ML model is intended to be used as a drop-in replacement for 3D-FE used by FAARFIELD to quickly calculate critical concrete stresses due to aircraft and thermal loads. The model targets rigid pavement design of airfields serving commercial aircraft heavier than 100,000 pounds gross weight. The ML model is a general model that supports individual dual (D), dual-tandem (2D) and dual-tridem (3D) gear configurations as well as a general model for full belly or landing gear configurations. The study proposed a conceptual design method based on cumulative damage factor (CDF) but suitable for top-down cracking design. The input required by the conceptual design method implies that the ML model needs to provide the distribution of transverse stress along the transverse joint and the distribution of longitudinal stress along the longitudinal joint.

A database consisting of 127,000 input-output tuples to train the ML model was developed using finite element methods. The database contains distinct combinations of rigid pavement, thermal, and aircraft gear parameters that were input into FEAFAA 3.0 to determine the resulting stress distribution at the top of the slab. Researchers developed a new artificial neural network (ANN) method that predicts a dynamic functional evaluated over a continuous domain. The model is based on a modular deep learning method. The training operation was performed using backpropagation and the ADAHESSIAN numerical optimization algorithm. The models constructed with the new method are significantly more accurate than previous ML techniques for similar problems. The resulting model is a single model with one interface for all gear types in the data set. The ML model was compiled into a .NET-compatible library suitable for use in a program like FAARFIELD.

17. Key Words	18. Distribution Statement
Rigid pavement, Airport pavement, Pavement design, Machine learning, Artificial Neural Network, Deep learning	This document is available to the U.S. public through the National Technical Information Service (NTIS), Springfield, Virginia 22161. This document is also available from the Federal Aviation Administration William J. Hughes Technical Center at actlibrary.tc.faa.gov.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	74	

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ACRONYMS

σ_{yy}	Longitudinal stress
σ_{xx}	Transverse stress
2D	Dual-tandem gear
3D	Dual-tridem gear
3D-FE	Three-dimensional finite element
AAE	Average absolute error
AC	Advisory circular
ANN	Artificial neural network
API	Application Programming Interface
C/P	Coverage-to-pass ratio
CDF	Cumulative damage factor
CLI	Common Language Infrastructure
CTE	Coefficient of thermal expansion
D	Dual gear
DLL	Dynamic link library
EJS	Equivalent joint stiffness
ELTG	Equivalent linear temperature gradient
FAA	Federal Aviation Administration
FAARFIELD	FAA Rigid and Flexible Iterative Elastic Layered Design
FAASR3D	FAA Structural Response-3D
FE	Finite element
FEAFAA	Finite Element Analysis – FAA
JIT	Just-in-time
L	Longitudinal
LSTM	Long short-term memory
ML	Machine learning
MLP	Multilayer perceptrons
MSE	Mean squared error
NAPTF	National Airport Pavement Test Facility
NCHRP	National Cooperative Highway Research Program
NTIS	National Technical Information Service
P/C	Pass-to-coverage ratio
PCC	Portland cement concrete
PCI	Pavement condition index
PDF	Portable document format
psi	Pounds per square inch
R^2	Coefficient of determination
RMSE	Root mean square error
RRMSE	Relative root mean square error
SCI	Structural Condition Index (load-related component of PCI)
T	Transverse
TG	Temperature gradient
TS	Tandem Spacing
UMAP	Uniform manifold approximation and projection

VB.NET	Visual Basic .NET
XML	Extensible Markup Language

EXECUTIVE SUMMARY

The Federal Aviation Administration (FAA) rigid pavement design process is based on bottom-up cracking failure resulting from tensile stress at the bottom of a flat slab under aircraft loads and does not consider the top-down cracking failure mode. In jointed rigid pavements, top-down cracking is likely to govern the failure when a slab with a negative thermal gradient is loaded near its edges by an aircraft gear. The existing procedure in the FAA airport pavement design program, FAA Rigid and Flexible Iterative Elastic Layered Design (FAARFIELD), does not support design of top-down cracking failure mode. FAARFIELD calculates critical stresses for pavement design using the FAA Structural Response-3D (FAASR3D) finite element library. However, direct use of FAASR3D is time consuming and not practical for design.

This study developed a machine learning (ML) model to support design of airfield rigid pavements to resist top-down cracking. The overall goal of the research was to add the top-down cracking failure mode to the FAARFIELD program. The ML model is intended to substitute the direct three-dimensional finite element (3D-FE)-based computation of concrete stresses at the top of the slab due to aircraft and thermal loads. The ML model targets rigid pavement design of airfields serving commercial aircraft heavier than 100,000 pounds gross weight. The model is developed for a four-layer system of Portland cement concrete (PCC) slab, stabilized base, base/subbase, and infinite subgrade. The ML model is a general model that supports individual dual (D), dual-tandem (2D) and dual-tridem (3D) gear configurations as well as a general model for full belly or landing gear configurations.

The study proposed a conceptual design method based on cumulative damage factor (CDF) but suitable for top-down cracking design. The conceptual design method requires the ML model to quickly return the distribution of transverse stress along the transverse joint and the distribution of longitudinal stress along the longitudinal joint.

Researchers developed a synthetic database consisting of 127,000 finite element simulations to train the ML model. The database contains distinct combinations of rigid pavement, thermal, and aircraft gear parameters that were input into FEAFAA 3.0 to determine the resulting stress distribution at the top of the slab. The database has total of 19 input parameters including pavement properties (layer moduli and thicknesses, joint spacing, and equivalent joint stiffness), thermal load (temperature gradient and coefficient of thermal expansion) and aircraft gear information (gear type, e.g., D, 2D, 3D; gear weight; tire pressure; dual spacing; tandem spacing [if any]; and track spacing [if any]). Gear transverse position was also considered as an input variable to account for aircraft lateral wander. A range was defined for each input parameter to be representative of typical rigid airfield pavement structures and in compliance with the FAA pavement design guidelines. A Monte Carlo simulation was performed to choose the input values for each simulation from the defined ranges. Uniform distributions were used for each input parameter to increase the generalization and accuracy of the trained models across the input space.

Researchers developed a new artificial neural network (ANN) method that predicts a dynamic function evaluated over a continuous domain. The model is based on a modular deep learning method. The model was developed using the PyTorch machine learning. The model has three main modules: Input Encoding, Stress Distribution Prediction, and Uncertainty Prediction. The Input Encoding module takes the input parameters and processes the data into a common latent feature

representation. The latent representation is a highly compressed form of the data that conserves the relevant salient information. The model passes the latent features to the Stress Distribution Prediction module, which outputs the stress predictions. The Uncertainty Prediction module outputs an uncertainty metric that predicts the accuracy of the stress prediction. The training operation was based on backpropagation and the ADAHESSIAN numerical optimization algorithm. A multitask learning formulation of the cost function was used that balances two objectives: 1) minimize the error near the peak stress area, and 2) minimize the mean squared error of the entire stress distribution waveform. The ML model was compiled into a .NET-compatible library suitable for use in a program like FAARFIELD.

The accuracy of the developed model is significantly improved compared to prior work in developing a machine learning model for estimating rigid pavement stresses. The overall root mean square error (RMSE) of testing dataset was 3.95 pounds per square inch (psi). The error analysis indicated that over 90% of errors were less than 5 psi. Also, 92% of errors were greater than -3.95 psi, i.e., the model underpredicts the stress being more than 3.95 psi in only 8% of the cases. Sensitivity analysis determined that error in the model will likely result in less than 0.5-inch of error in the final slab thickness.

The new ML model is significantly more accurate than previous ML techniques for similar problems. The model has several advantages:

- The model is a single model with one interface for all gear types, including full-landing gear configuration, rather than separate models for specific aircraft.
- Instead of just the peak stresses, the model predicts stress distribution, which is critical for accurate pavement design.
- The model is suitable for implementation of top-down cracking design in FAARFIELD.

1. INTRODUCTION

1.1 BACKGROUND

In jointed rigid pavements, top-down cracking is caused by high tensile stresses on the top of a slab. Certain configurations and locations of aircraft gear, high negative temperature or moisture gradient, and transfer of gear load between adjacent slabs through doweled joints can produce top-down cracking (National Cooperative Highway Research Program [NCHRP], 2003). Top-down cracking is likely to govern the failure when a slab with a negative thermal gradient (TG) is loaded near its edges by an aircraft gear. The TGs are caused by daily temperature variations. These gradients result in a difference in temperature between the top and bottom surface of the concrete slab that causes curling in the slab. A negative TG value means the slab is cooler on top and the slab curls upward. The TG tends to produce bending stresses in slabs due to the slab weight together with external constrains, such as foundation reaction, and joint connection between adjacent slabs.

Top-down cracking manifests as cracks that initiate at a longitudinal or transverse joint and propagate to the opposite joint (transverse and longitudinal cracking), or cracks that propagate diagonally to an adjacent joint (corner cracking). National Airport Pavement Test Facility (NAPTF) Construction Cycle 2 exhibited both of these top-down cracking modes during the experiment (Brill et al., 2005).

The Federal Aviation Administration (FAA) rigid pavement design process used by the current version of the FAA Rigid and Flexible Iterative Elastic Layered Design (FAARFIELD) software, version 2.0, is based on bottom-up cracking failure resulting from tensile stress at the bottom of a flat slab under aircraft loads and does not consider the top-down cracking failure mode. Unless otherwise noted, all subsequent references to FAARFIELD are to version 2.0 of the software. The FAA has a long-term goal to add the top-down cracking failure mode to the FAARFIELD program. FAARFIELD calculates critical stresses for pavement design using the FAA Structural Response-3D (FAASR3D) finite element library. FAASR3D is an updated version of NIKE3D program originally developed by the Lawrence Livermore National Laboratory of the U.S. Department of Energy (Guo, 2007) for three-dimensional finite element (3D-FE) analysis of rigid pavements. The stresses that lead to top-down cracking can be computed using the FAASR3D routines built into FAARFIELD, but direct use of 3D-FE methods in design software is typically far more time consuming than is acceptable. A single 3D-FE simulation could take up to 30 minutes on a typical business-class computer.

1.2 PRIOR RESEARCH INTO TOP-DOWN CRACKING DESIGN METHODS

The FAA initiated research to use machine learning (ML) techniques as a rapid alternative to 3D-FE modeling for calculating top-down cracking stresses in a reasonable timeframe for design. Iowa State University developed artificial neural network (ANN) models that estimated the maximum rigid pavement stresses in a slab for arbitrarily located landing gear (Kaya et al., 2017; Rezaei-Tarahomi, et al., 2020). They trained models for gear-loading only and models for combined gear and thermal loading, with separate models for each series of Airbus and Boeing aircraft. The coefficients of determination (R^2) of the models were generally above 0.8, indicating a strong correlation between the ANN and 3D-FE model prediction. Review of the research showed that

this metric describes the average performance of the ANN model but does not address outliers or validity across the input domain. The model validation results showed that many regions of the input domain yielded large inaccuracies, including errors as large as 35 pounds per square inch (psi). This was especially true for the models involving thermal loads and models for lighter aircraft.

1.3 RESEARCH OBJECTIVE

The overall goal of this research program is to develop ML solutions to support design of airfield pavements to resist top-down cracking. The ML model will substitute for the direct 3D-FE-based computation of concrete stresses due to aircraft and thermal loads by quickly returning the critical design stresses for airport rigid pavement top-down cracking failure modes. The model is a general model, supporting multiple aircraft types including those with belly gear. The model targets rigid pavement design of airfields serving aircraft heavier than 100,000 pounds gross weight. The program has three major components:

1. Development of a training database using 3D-FE simulations
2. Development of an ML model as an alternative to the 3D-FE model
3. Implementation of the model as a Visual Basic dynamic link library (DLL) suitable for integration into FARFIELD or other design software

This report documents the development of the training database used to train the ML model, the development of an ML model using Deep ANN methods, and the implementation of the model for use in Visual Basic .NET (VB.NET) applications. The model and library are suitable for use with pavement design methods based on cumulative damage factor (CDF) concepts. It uses the physical parameters of the pavement system, gear loads, and thermal loads as inputs to estimate the critical design stresses as the output.

The training database for an ML model consists of input-output tuples used by the ML method to learn to approximate the relationships between the inputs and the outputs. As such, each record in the database must contain the input parameters expected to be provided to the model and the expected output results from those parameters. To create the training database and to develop the ML model for this research, the following tasks were completed:

1. Identify required output from the model
2. Identify input parameters to the model
3. Identify ranges for each input parameter
4. Perform a Monte Carlo simulation to define the inputs into each finite element (FE) simulation
5. Set up the 3D-FE simulations to run in FEAFAA[1]
6. Run the 3D-FE simulations in FEAFAA
7. Post-process the 3D-FE output from FEAFAA to create the database

[1] FEAFAA is a stand-alone 3D-FE software program developed by the FAA for analysis of jointed concrete pavements. FEAFAA is capable of modeling the complete thermo-mechanical behavior of jointed concrete pavements, including tensile stresses at the top of the slab that cause top-down cracking.

8. Develop ML model based on the created database
9. Compile the ML model to a .NET-compatible library

Based on experience with similar physics-based models, ML experts used engineering judgement to estimate that a database with approximately 100,000 to 120,000 records would be sufficient to train the ML model. A training database with 100,000 records was initially targeted. Development of the database proceeded much quicker than expected, and the database size was increased to 127,000 records. The database contains distinct combinations of rigid pavement structure, TG, and aircraft gear loading that were input into FEAFAA 3.0 to determine the resulting stress distribution at the top of the slab. Results from all 127,000 runs were included in the training database.

2. IDENTIFY REQUIRED MODEL OUTPUT

Researchers assumed the output of the ML stress model developed under this research would be required as input to a top-down cracking design procedure, specifically the stress-related input into the failure model of the rigid pavement design procedure. The exact design procedure for top-down cracking has not been fully defined at the time of writing. To identify the stress-related input into a top-down cracking design procedure, researchers assumed the procedure would be conceptually similar to the existing bottom-up cracking rigid pavement design procedure used by FAARFIELD. Researchers reviewed the design procedure in FAARFIELD and identified input into the failure model along with potential modifications to the procedure necessary to support design for the top-down cracking failure mode. Section 2.1 summarizes the FAARFIELD design procedure, and sections 2.2 and 2.3 discuss potential changes to the procedure to support a top-down cracking mode and the effects on the design of the ML model.

2.1 FAARFIELD 2.0 DESIGN PROCEDURE

FAARFIELD uses an iterative procedure based on CDF, the representation of the amount of the fatigue life of a pavement used up by traffic. A simplified description of the process is:

1. Divide the pavement along its length into a set of longitudinal strips (FAARFIELD uses 10-inch-wide strips)
2. For each aircraft in the design traffic:
    a. Determine the magnitude of the critical stress for that aircraft
    b. Determine the number of allowable repetitions of the critical stress until failure
    c. Distribute repetitions of the critical stress laterally across the pavement into the strips based on volume of traffic, landing gear geometry, critical stress location, and aircraft wander (FAARFIELD uses the concept of pass-to-coverage [P/C] ratio, discussed below, to distribute aircraft traffic across the pavement strips
    d. Calculate the CDF in each strip as the ratio of actual repetitions to allowed repetitions (CDF is discussed in further detail below)
3. Sum the CDF values from all aircraft in the traffic mix in each strip
4. Check the CDF values in each strip
    a. If at least one strip has CDF=1.0 and no strip has CDF>1.0, the design is complete
    b. If at least one strip has CDF>1.0, the pavement is too thin; increase the thickness and go to step 2

c. If all strips have CDF<1.0, the pavement is thicker than required; decrease the thickness and go to step 2

The key concept in the design procedure is that the CDF describes the amount of damage a specific aircraft causes at a specific location in the pavement. This value can be calculated separately for each aircraft for various locations across the pavement, then summed to determine the cumulative damage at each location from all aircraft. CDF is defined as the ratio of the number of applied load repetitions to the number of repetitions to failure as expressed by Equation 1 (Advisory Circular [AC] 150/5320-6G):

$$CDF = \frac{(annual\ departures) \times (life\ in\ years)}{(pass\ to\ coverage\ ratio) \times (coverages\ to\ failure)} \quad (1)$$

AC 150/5320-6G defines the P/C ratio as "the number of passes required to apply one full load application to a unit area of the pavement" (FAA, 2021). FAARFIELD uses the idea of effective tire width to calculate P/C. Kawa (2012) elaborates that the inverse of P/C, the coverage-to-pass (C/P) ratio, is "the probability that any part of the effective tire width covers the center point of a strip," i.e., receives stress from a given pass of an aircraft. P/C accounts for gear geometry and lateral wander of the aircraft. FAARFIELD assumes the effective tire width is directly beneath and the same as the width of the tire contact patch for rigid pavements (Kawa, 2012).

Coverages to failure is the number of coverages a pavement can support before the pavement fails. Coverage is a measure of repetitions of the maximum stress at the bottom of the PCC layer. Coverages to failure is determined from a model developed at NAPTF that relates magnitude of a pavement stress to the number of allowable repetitions of that stress. Failure of a rigid pavement is defined as having a Structural Condition Index (SCI) of 80 or below. SCI is derived from the pavement condition index (PCI) by considering only load-related distresses. An SCI of 80 is consistent with 50 percent of slabs in the pavement exhibiting a structural crack (Brill, 2010).

2.2 ENHANCEMENTS TO CDF DESIGN PROCEDURE TO SUPPORT TOP-DOWN CRACKING

The design procedure in FAARFIELD considers only bottom-up cracking failure without thermal loads and enabling certain assumptions about the nature of the stresses imposed on pavement by aircraft gear. First, the process assumes that any portion of the pavement below the tire width receives the critical stress. Second, while the critical stress is calculated at a joint, the design process does not consider the actual location of a joint within a pavement when calculating CDF. Third, it assumes the critical stresses are equal in both directions, i.e., the transverse stress (σ_{xx}) along the transverse edge is equal to the longitudinal stress (σ_{yy}) along the longitudinal edge. These assumptions are discussed with respect to top-down cracking in the subsequent subsections. The final subsection is a conceptual CDF design procedure that addresses these assumptions. The conceptual design procedure provides enough detail to identify inputs required from the ML model.

2.2.1 Critical Stress Location and P/C Ratio

The P/C ratio is the inverse of probability that a given location will experience stress due to a pass of an aircraft. In FAARFIELD, the P/C calculation assumes that the location of the maximum stress is beneath the location of the aircraft tire (Kawa, 2012). Points beneath the tire width receive the stress, and points outside the tire width do not. The CDF procedure used by FAARFIELD assumes the coverage to failure in each strip is due to the load and the P/C ratio of the same strip, which simplifies the algorithm.

As shown in Figure 1, critical tensile stress at the top of the slab due to aircraft loading can be located far outside the footprint of the landing gear and is not necessarily the same width as the tire or gear assembly.

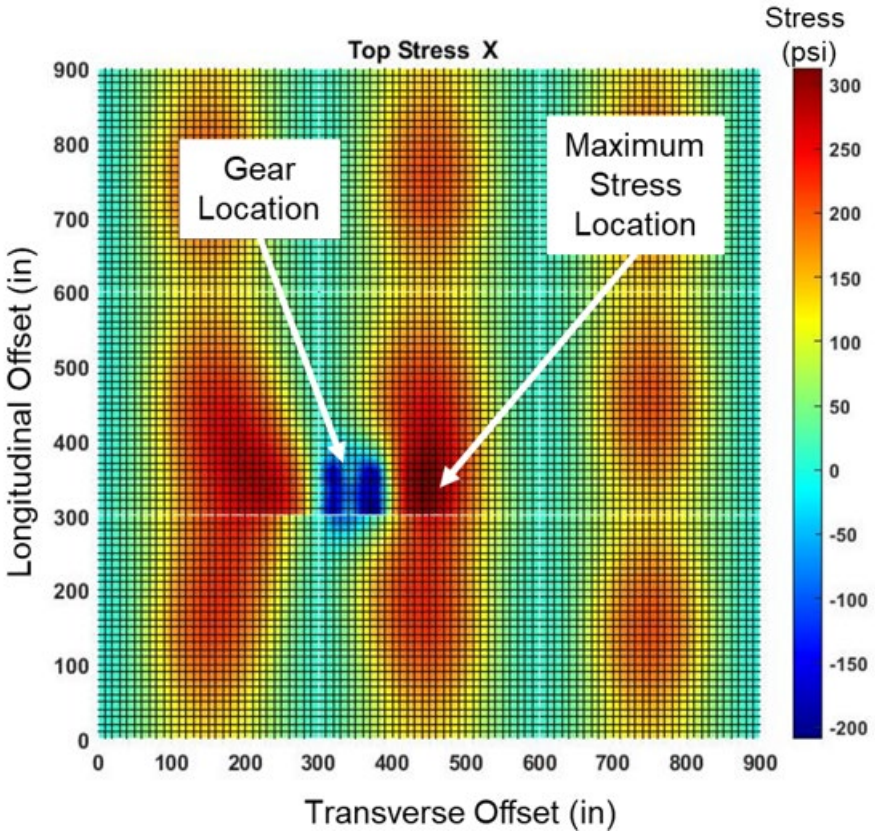


Figure 1. Stresses Outside Gear Footprint, 2D Gear Loading and 25-foot Joint Spacing

Further complicating the calculation of the location of stresses resulting from a pass of a gear, and thus the P/C ratio, the transverse location of the critical stress at the top of a slab can change with the location of the gear with respect to the longitudinal joint, as shown in Figure 2. This indicates that the coverage to failure at each strip results from the load at other strips. When calculating the CDF, a top-down cracking design procedure must therefore consider where the aircraft gear is located, how far away from the gear the critical stress occurs, and the width of the stress bulb.

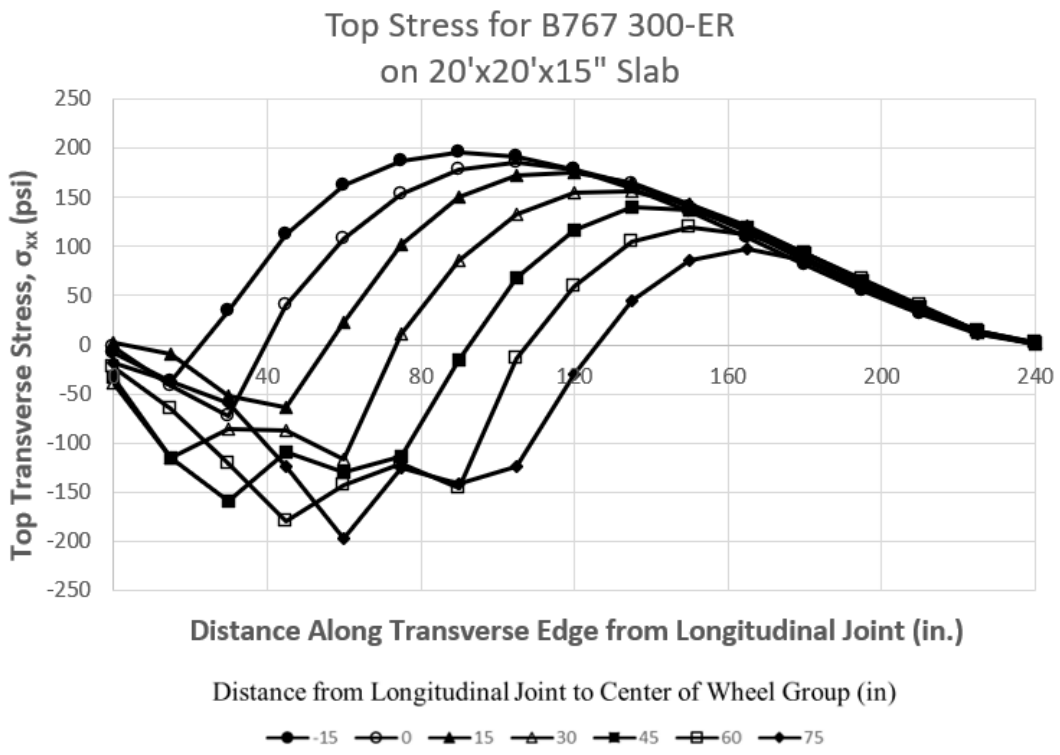


Figure 2. Example Illustrating Variation in Maximum Stress Magnitude and Location with Transverse Gear Location

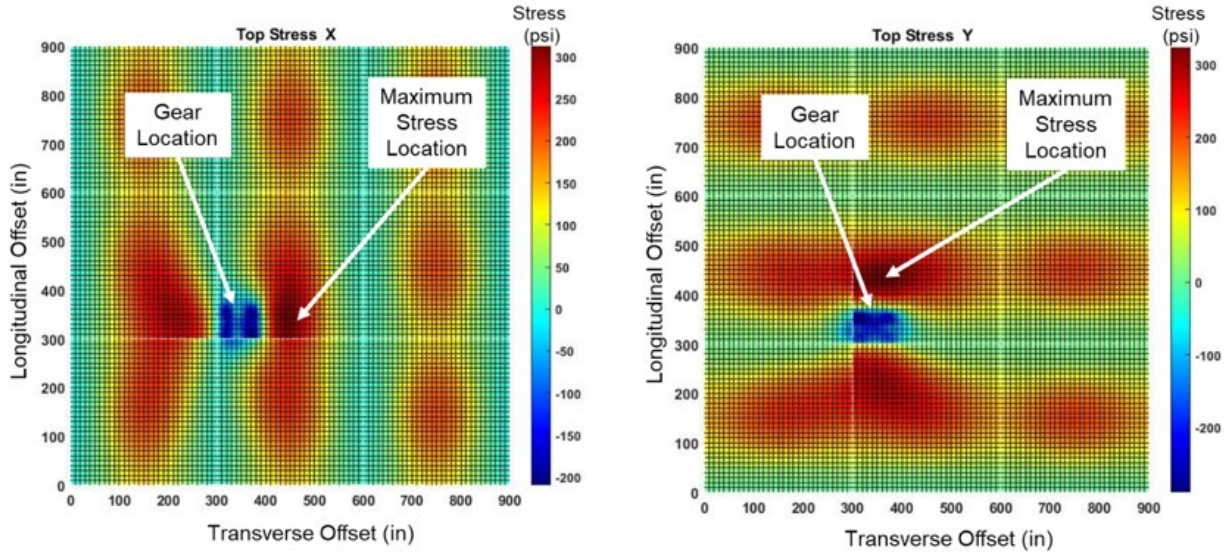
2.2.2 Longitudinal Joint Location and Critical Stress Magnitude

FAARFIELD assumes the critical stress occurs when aircraft gear is adjacent to a joint. It uses a 3D-FE analysis to calculate the stress for this case and uses the result for the critical stress in each strip. No consideration is given to whether the strip is at the edge of the slab or in the center of a slab. The software conservatively assumes that loading in any of the strips results in the magnitude of stress at the bottom of the slab due to an edge-loading condition.

For top-down cracking, the magnitude of the critical stress can change by a factor of 2 depending on transverse gear location within the slab, as shown above in Figure 2. The maximum tensile stresses at the top of the slab occur when the gear location is near the longitudinal edge. However, using the maximum stress in the critical loading condition would be overly conservative. Additionally, since aircraft wander transversely across the pavement, this critical loading condition may not produce the highest level of damage if the aircraft wheels are rarely in this position. Furthermore, stresses that are non-critical but still large enough to cause damage, extend several feet across the pavement, as shown in both Figures 1 and 2. As the gear moves away from the longitudinal edge, the top tensile stress in the longitudinal direction at the slab longitudinal edge will be reduced. However, these lower stresses still contribute to the cumulative damage. This means that the approach of calculating the maximum stress one time and using it as the critical stress in all strips is not valid. The critical stress should be calculated for each strip for each wander position.

2.2.3 Inequality of σ_{xx} and σ_{yy}

FAARFIELD considers the stress component parallel to a joint in the edge-loading condition as the critical stress for bottom-up cracking design. The stress component perpendicular to the joint is small enough that it can be neglected during damage calculations. FAARFIELD considers either 0- or 90-degree orientation of the gear, whichever produces greater stress. The maximum stress parallel to the joint is considered the critical stress for the entire slab. However, for top-down cracking stresses, the gear loading can produce high top tensile stress in both directions, as shown in Figure 3.



(a) σ_{xx} field (b) σ_{yy} field

Figure 3. Comparison of (a) σ_{xx} and (b) σ_{yy} Field at Top of Slab from 2D Gear Loading

Not only may the maximum stress locations in the σ_{xx} and σ_{yy} directions differ, but both may be large enough to cause damage. It follows that a single pass of a gear may contribute damage towards a longitudinal crack in one location and also a different amount of non-negligible damage towards a transverse crack in a different location. The conceptual design procedure must therefore consider σ_{xx} and σ_{yy} separately.

2.2.4 Transverse Joint Location

The moving aircraft gear continuously changes position with respect to the nearest transverse joint, raising the question of how the stress field changes as the aircraft rolls across the slab. Researchers investigated how the magnitude and location of the peak stress evolves as the gear moves. The concern was whether a concept similar to lateral aircraft wander would be required to adequately capture how a taxiing aircraft applies stress to the slab. Results from numerous FE simulations indicated that the maximum values of σ_{xx} and σ_{yy} along the transverse and longitudinal edges occur when the aircraft gear loads are applied near the transverse edge, i.e., when the gear enters the slab. As shown in Figure 4, the top-down cracking stresses were calculated for various longitudinal offsets of a 2D landing gear (B767-300 ER).

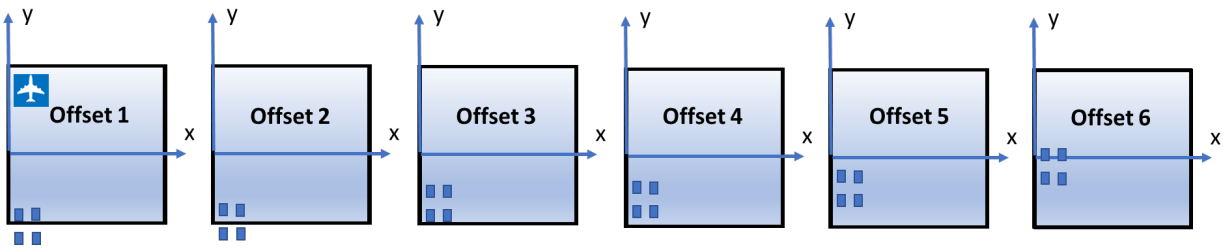


Figure 4. Simulation of a 2D Gear Load Moving Longitudinally

For Offset 1, the gear was positioned such that the front two wheels occupied the slab just forward of the transverse joint, as shown in Figure 4. In Offset 2, the center of the wheel group was positioned directly over the transverse joint, while in Offset 3, all four wheels were on the slab, with the rear wheels positioned along the transverse joint. For offsets 4–6, the gear was advanced by 30-inch intervals from Offset 3. An 18-inch-thick PCC slab was used for the simulation, with TG = -1.0 °F/in. and 25-ft × 25-ft joint spacing, and all joints dowelled. Both σ_{xx} and σ_{yy} along the transverse and longitudinal edges were checked. As a gear moves through the slab during a pass of an aircraft, it goes through each of the positions. Thus, a slab will experience all of the calculated stress states during a single pass. The stress states for each position were plotted for comparison, as shown in Figures 5 and 6.

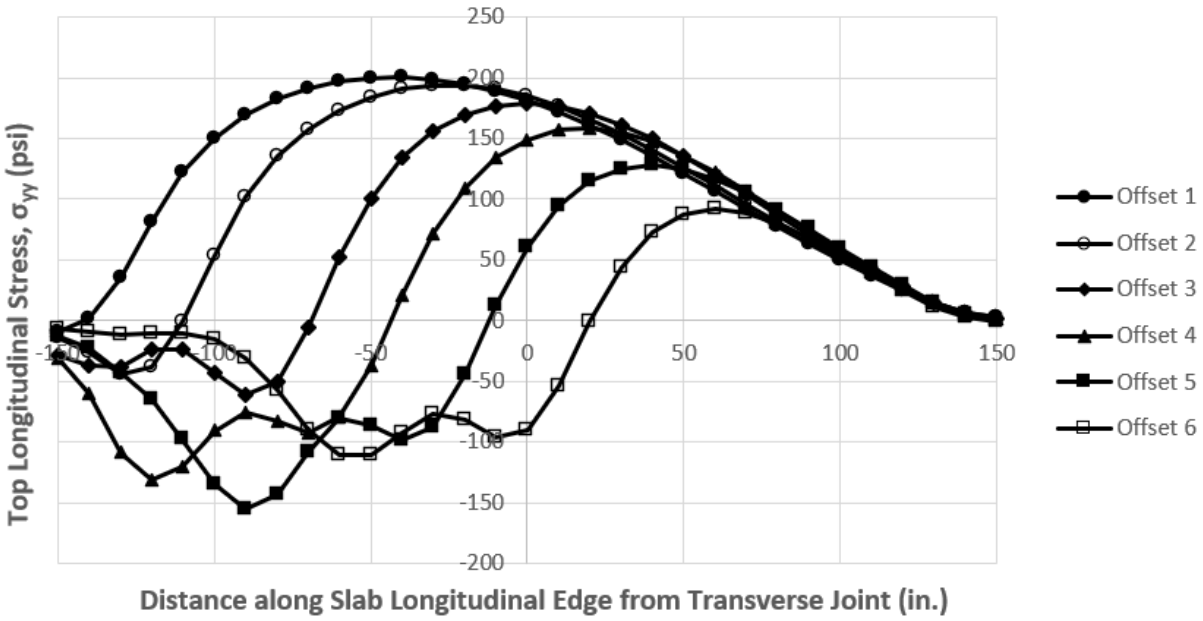


Figure 5. Top-of-Slab Stress Distribution (σ_{yy}) along Longitudinal Edge Due to 2D Gear Load for Various Longitudinal Gear Offsets from Transverse Joint (see Figure 4)

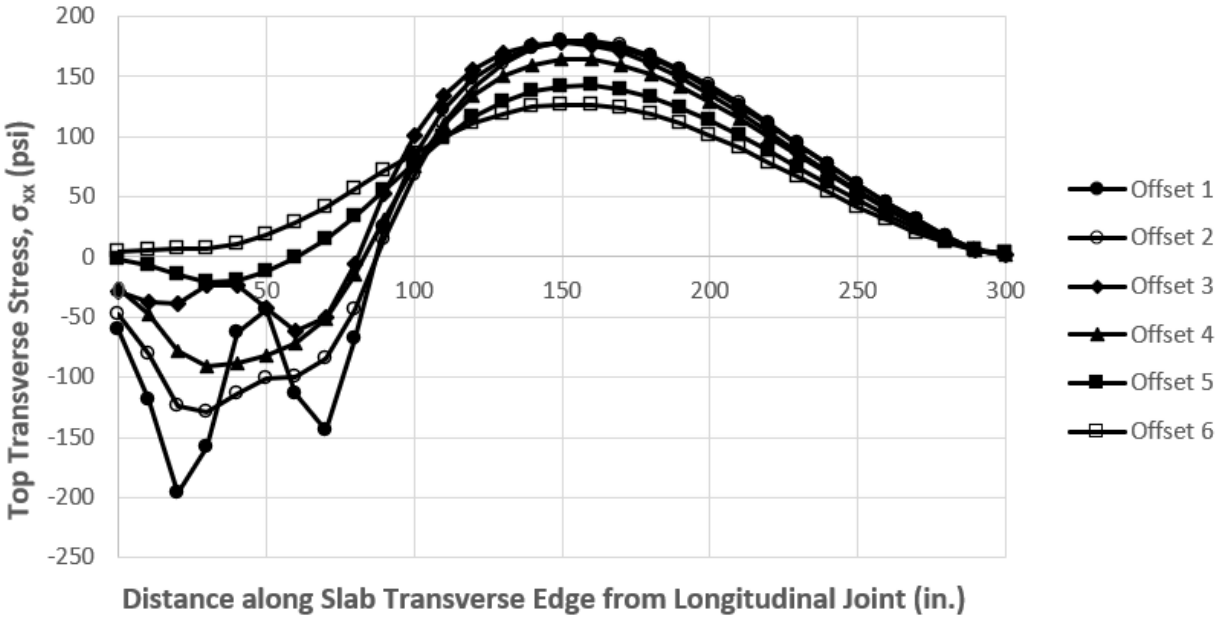


Figure 6. Top-of-Slab Stress Distribution (σ_{xx}) along Transverse Edge Due to 2D Gear Load for Various Longitudinal Gear Offsets from Transverse Joint (see Figure 4)

The critical case for both σ_{xx} and σ_{yy} is Offset 1. It not only produces the highest stress in the slab, but it also produces the highest stress at all locations in the slab for all gear locations. This implies that only the critical location needs to be checked.

The pavement system was also loaded with a moving 3D gear (one belly gear of an Airbus A380 aircraft). Gear offset locations are shown in Figure 7, with stress distributions shown in Figures 8 and 9.

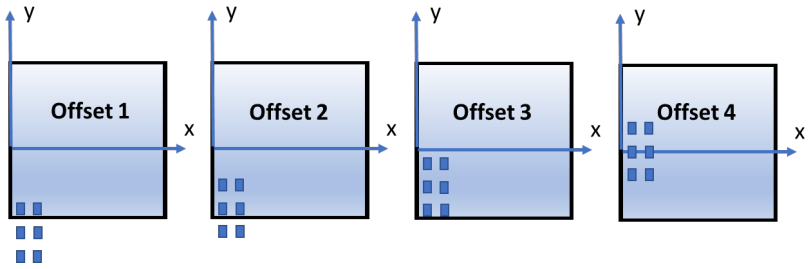


Figure 7. Simulation of a 3D Gear Load Moving Longitudinally

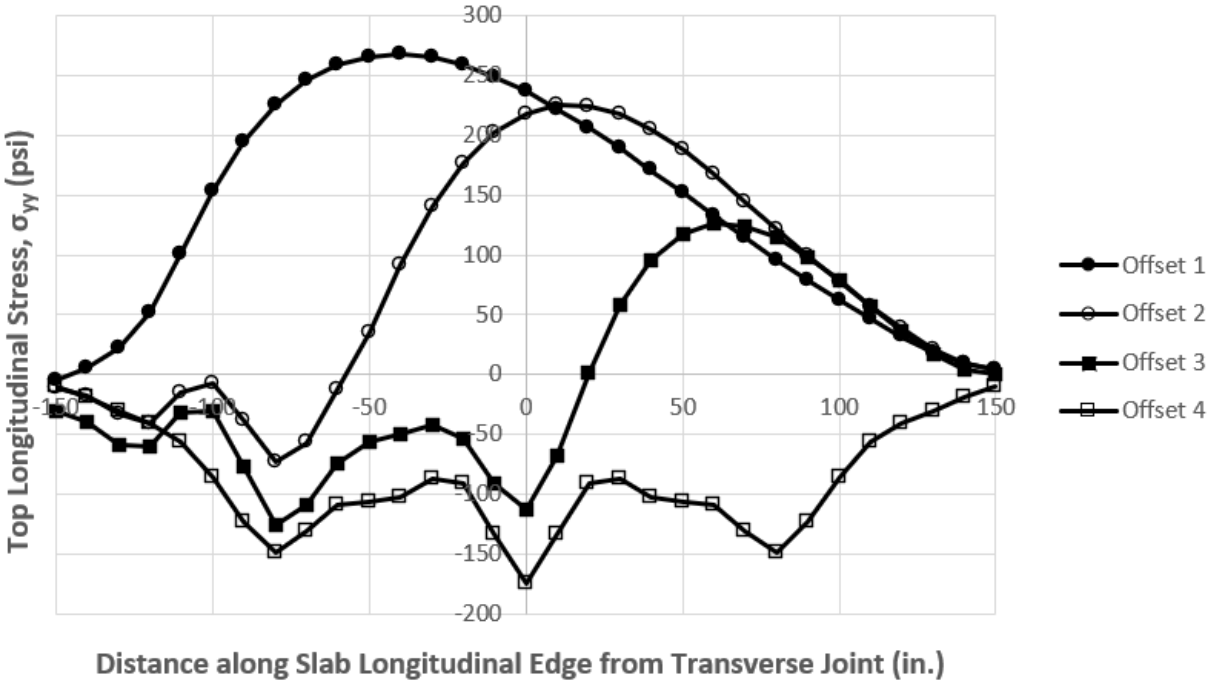


Figure 8. Top-of-Slab Stress Distribution (σ_{yy}) along Longitudinal Edge Due to 3D Gear Load for Various Longitudinal Gear Offsets from Transverse Joint (see Figure 7)

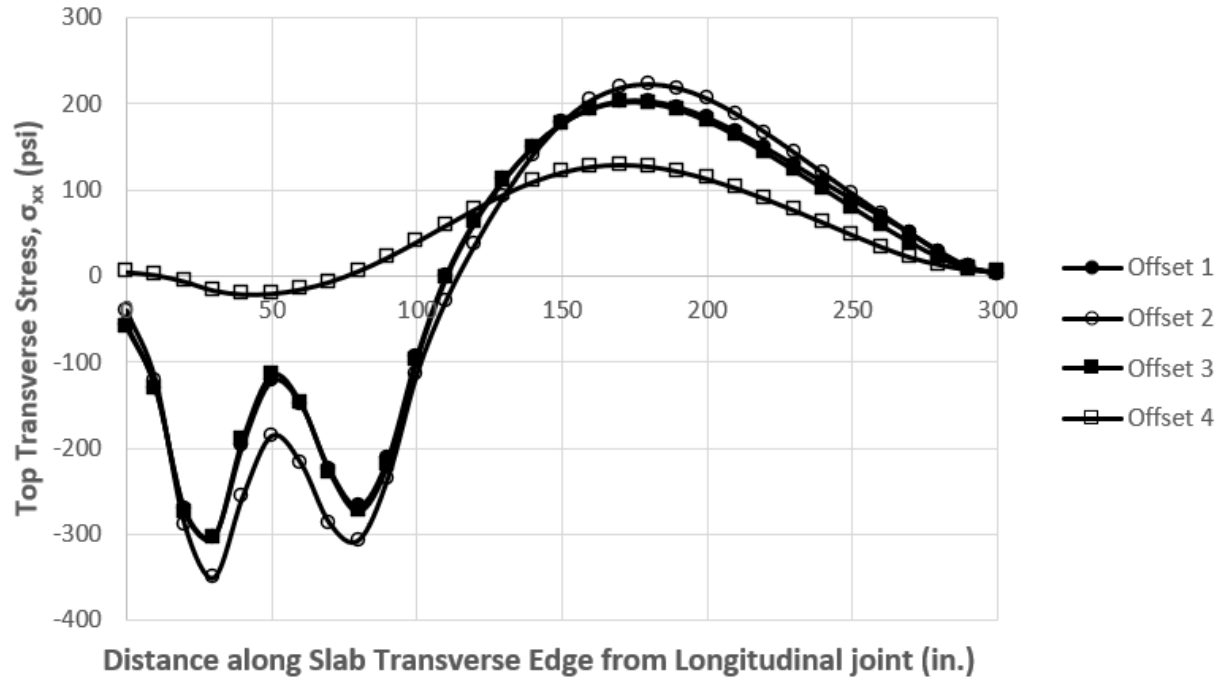


Figure 9. Top-of-Slab Stress Distribution (σ_{xx}) along Transverse Edge Due to 3D Gear Load for Various Longitudinal Gear Offsets from Transverse Joint (see Figure 7)

As shown in Figure 8, gear Offset 1, with the front wheels on the transverse edge, is the critical loading condition producing the largest peak tensile stress σ_{yy}. Longitudinal Offset 2 can create larger tensile stresses σ_{yy} than Offset 1 in some locations; but the controlling damage will accrue in the location of the peak stress from Offset 1, and damage from other offsets can be neglected. The critical location for σ_{xx} is Offset 2, with the center wheels on the transverse joint. Note that for σ_{xx}, Offset 4 causes the highest stress in the interval between approximately 75 in. to 135 in. (distance measured from slab transverse edge), as shown in Figure 9. However, the stress in this interval is less than half of the peak stress and is unlikely to contribute to damage.

Results from these examples suggest that simulating a gear rolling longitudinally along the pavement will not be necessary to obtain the critical tensile stresses at the edges. This means that only the transverse wander needs to be considered in the conceptual CDF method for top-down cracking. The maximum edge stresses can be determined by fixing the gear location in the longitudinal direction at the critical locations.

To further elaborate this concept, a pavement system was loaded with three gear loads: D (B737-800), 2D (B767-300 ER), and 3D (A380 belly gear), at the critical locations (Offset 1 in Figures 4 and 7 above). All three gears were placed at a 25-inch offset from the longitudinal joint, as shown in Figure 10.

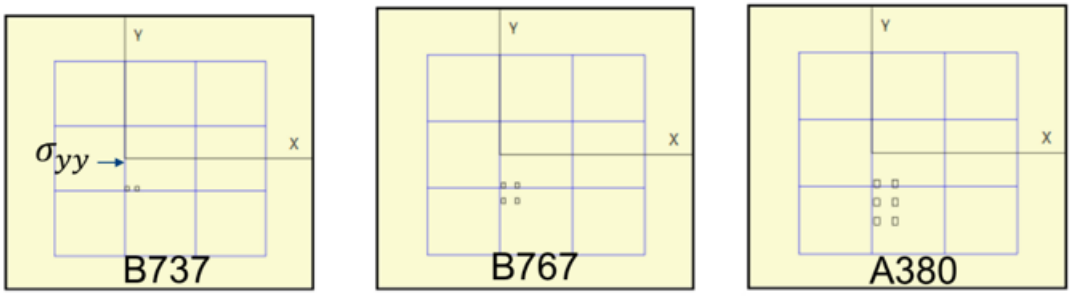


Figure 10. Gear Configuration Setup in FEAFAA

A 14-inch-thick PCC slab was used for the simulation, with TG = -1.0 °F/in., 20×20-ft joint spacing, and all joints dowelled. Figure 11 plots the top stress (σ_{yy}) along the longitudinal edge for each gear load. The peak stress occurs at nearly the same locations, but at different magnitudes, for the three cases. This indicates that placing all the gear types at the critical Offset 1 will ensure the stress accumulation can be captured at a critical location.

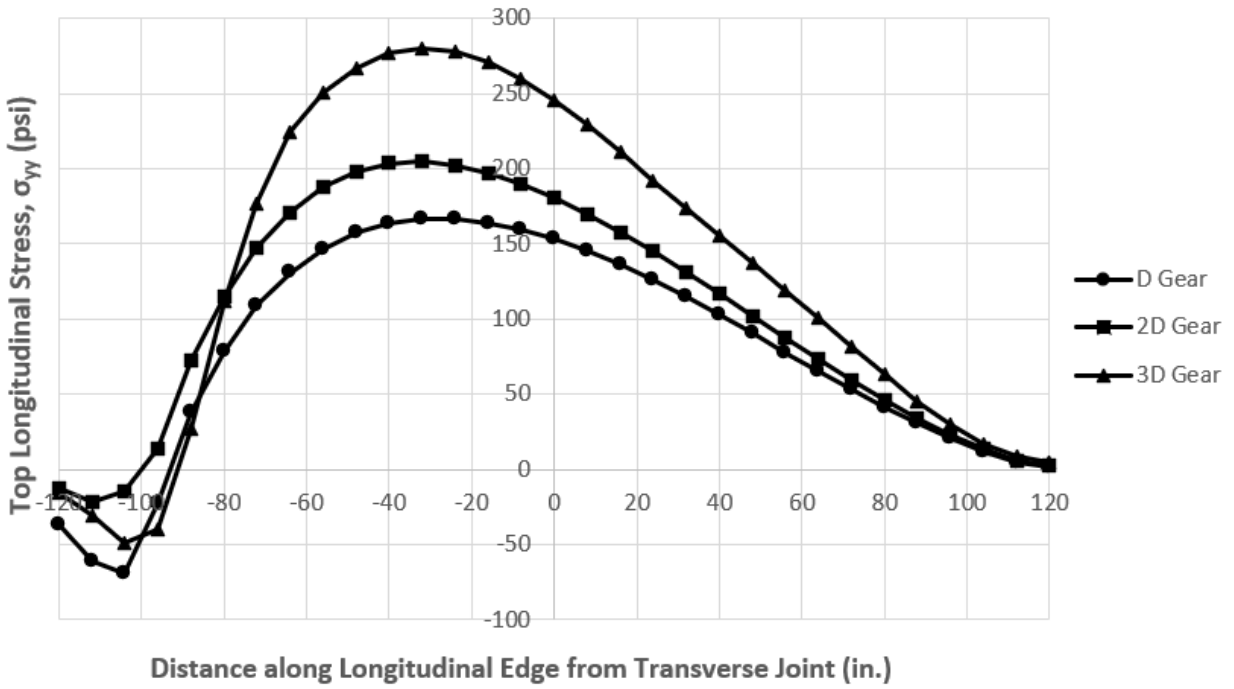


Figure 11. Top-of-Slab Stress Distribution Along Longitudinal Edge at Critical Offset 1

2.3 CONCEPTUAL CDF PROCEDURE FOR TOP-DOWN CRACKING

The purpose of the concept-level procedure presented in this section is to provide enough information to determine what information the ML model under development will be required to provide. It is not intended as a turn-key solution, and additional development may be required for efficient implementation. Researchers modified the existing CDF-based design procedure to address the issues in the previous section. A significant change is breaking the implicit connection between wheel locations and design strips used by the current method. The recommended conceptual CDF-based design procedure to support design for the top-down cracking failure mode is as follows:

1. Divide the pavement along its length into a set of longitudinal strips, and across its width into a set of transverse strips, as shown in Figure 12. The number of strips can be determined based on the design requirements. The user must enter the transverse locations of the longitudinal joints and the longitudinal spacing of the transverse joints.

2. For each aircraft in the design traffic:

    a. Distribute the physical gear position transversely across the pavement location to account for aircraft wander. Place the wheel in the critical longitudinal position.

    b. For each transverse gear position:

        i. For each longitudinal strip:

1. Determine the magnitude of the critical stress σ_{xx} for that aircraft in that position in that longitudinal strip.

2. Determine the number of allowable coverages for this strip based on the critical stress σ_{xx} determined in step 2.b.i.1.

3. Calculate the number of coverages for this longitudinal strip based on the probability the gear will be in this transverse position.

4. Calculate the CDF_{xx} for this longitudinal strip for this aircraft in this transverse position as the ratio of the coverages calculated in step 2.b.i.3 to the allowable coverages calculated in step 2.b.i.2.

ii. For each transverse strip:

1. Determine the magnitude of the critical stress σ_{yy} for that aircraft in that position in that transverse strip.

2. Determine the number of allowable coverages for this strip based on the critical stress σ_{yy} determined in step 2.b.ii.1.

3. Calculate the number of coverages for this transverse strip based on the probability that the gear will be in this transverse position.

4. Calculate the CDF_{yy} for this transverse strip for this aircraft in this transverse position as the ratio of the coverages calculated in step 2.b.ii.3 to the allowable coverages calculated in step 2.b.ii.2.

3. Sum the CDF_{xx} calculated in step 2 in each longitudinal strip for all aircraft.

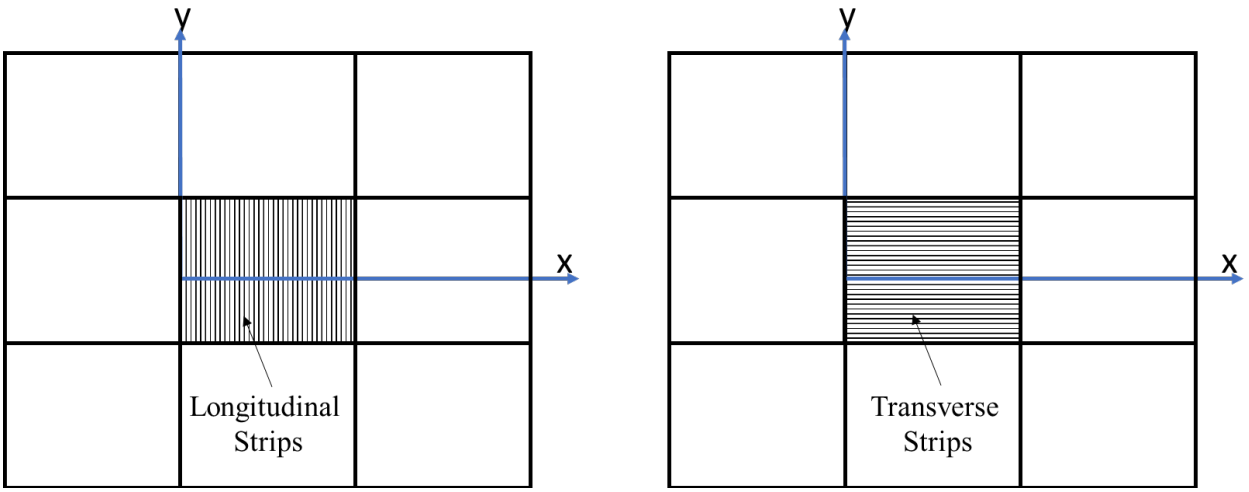
4. Sum the CDF_{yy} calculated in step 2 in each transverse strip for all aircraft.

5. Check the $CDF_{xx,i}$ values in each longitudinal strip i and the $CDF_{yy,j}$ values in each transverse strip j.

a. If at least one strip i or j has CDF=1.0 and no strip has CDF>1.0, the design is complete.

b. If at least one strip i or j has CDF>1.0, the pavement is too thin. Increase the thickness and return to step 2.

c. If all strips have CDF<1.0, the pavement is thicker than required. Decrease the thickness and return to step 2.



(a) (b)

Figure 12. Longitudinal (a) and Transverse (b) Strips in Conceptual CDF Design Procedure

2.4 RECOMMENDED MODEL OUTPUT

By inspection of the conceptual CDF-based top-down cracking design procedure, it becomes apparent that the required design input is more complex than the one used in FAARFIELD. All the inputs required by FAARFIELD are required by the new procedure, plus some additional information. For example, joint spacing, which is a design choice and defined by a user, is required to calculate the critical stresses. Steps 2.b.i.1 and 2.b.ii.1 indicate that the critical stress must be calculated for each strip for each transverse gear position of each aircraft; therefore, calculation of a single critical stress and using it in all strips is not appropriate. This implies that the ML model must provide more than just the single largest stress in the slab. It must provide the stress distribution across the slab. Since the top-down cracking is likely to initiate from the slab joints, the ML model needs to provide the distributions of σ_{yy} along the longitudinal joint and σ_{xx} along the transverse joint.

3. IDENTIFY ML MODEL INPUTS

Input to the ML model was assumed to be the same as the input into a 3D-FE model capable of producing the same output as the anticipated ML model. Finite element models require pavement configuration, thermal load parameters, and gear configuration to determine top-down cracking stresses.

Pavement layer moduli and thicknesses, joint spacing, and equivalent joint stiffness (EJS) were considered as input variables. They are fundamental parameters in FE modeling of a rigid pavement and their values typically vary from pavement to pavement. To reduce model complexity, only a single pavement structure was considered (a four-layer system), meaning that the number of layers is not an input into the ML model. Pavements supporting aircraft targeted by the model (those greater than 100,000 pounds) must use a four-layer system per AC 150/5320-6G. The value of Poisson's ratio was not included as a direct input to the ML because its value is relatively constant for each of the pavement materials considered.

The 3D-FE program FEAFAA models joints as linear elastic vertical spring elements between adjacent slabs (Brill, 1998). The spring elements transmit shear and vertical loads between the slabs. The spring constant is defined as the EJS. The EJS can be modified to represent either doweled or non-doweled joints. The ML model used the same EJS value for both transverse and longitudinal joints.

FEAFAA uses the concept of equivalent linear temperature gradient (ELTG) as the input quantifying the temperature variation through the slab thickness. Based on its use in FEAFAA, ELTG was included as an input to the ML model.

The ML model is a general model that supports individual dual (D), dual-tandem (2D) and dual-tridem (3D) gear configurations as well as full belly or full landing gear configurations. Gear configuration parameters were modeled after the data included in the FAARFIELD aircraft library. Fully describing an aircraft gear requires knowledge of the gear type (i.e., D, 2D, or 3D), gear weight, tire pressure, dual spacing, tandem spacing (if any), and track spacing (if any). Dual, tandem, and track spacing are defined in Figure 13.

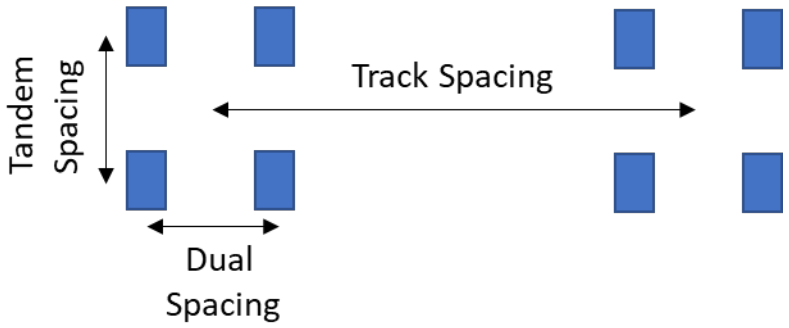


Figure 13. Dual, Tandem, and Track Spacing Parameters

In FEAFAA, the gear location with respect to the origin is defined by *X-offset* and *Y-offset*, as illustrated in Figure 14 for (a) an individual gear truck, and (b) a full landing gear consisting of 2 trucks.

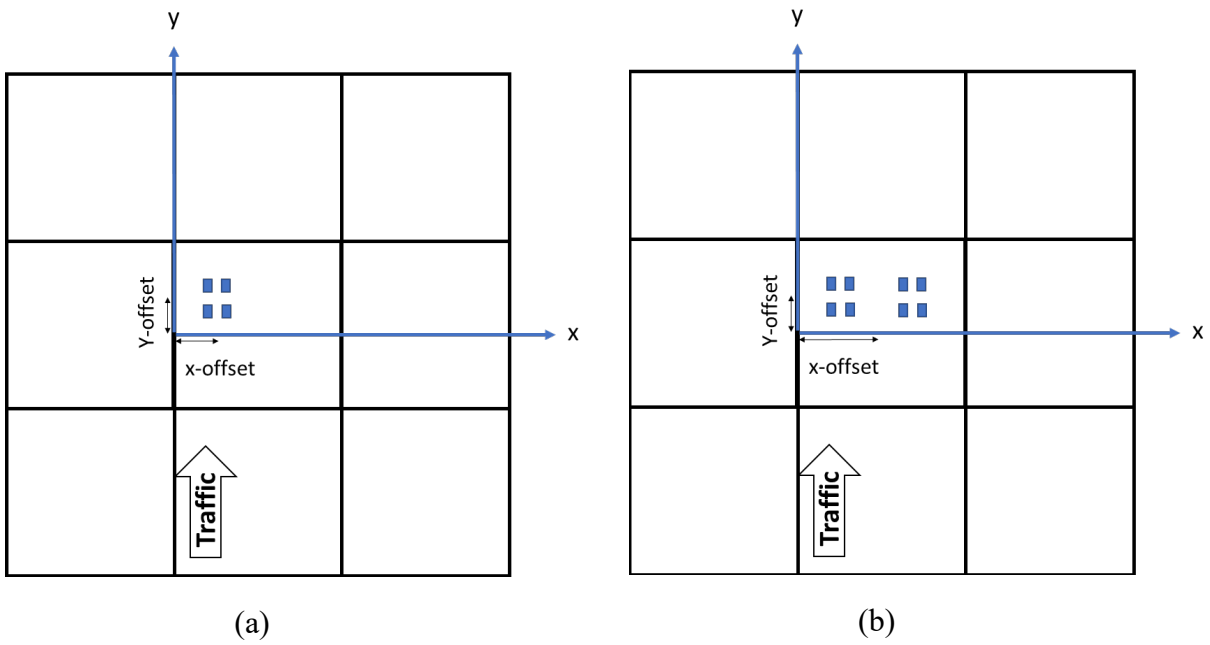


Figure 14. *X-offset* and *Y-offset* with Respect to the Origin for (a) Individual and (b) Full Landing Gears

X-offset is measured from the origin to the center of the wheel group of an individual gear, or to the center of the gear group for a full landing gear. *Y-offset* is measured from the origin to the center of the tandem group in the longitudinal direction. These definitions were retained for the ML model. Because the conceptual design procedure requires computing stresses with the gear at various wander positions, the transverse gear position (*X-offset*) was included as an input to the ML model. However, as discussed in section 2.2.4, the longitudinal position of the gear can be fixed at critical locations to compute the stress distribution along the slab edges. Therefore, the *Y-offset* variable was not used as an input variable for ML training.

4. FINITE ELEMENT SIMULATION EXECUTION

After selecting the input parameters, approximately 127,000 3D-FE simulations were conducted to build the database. For each simulation, values were chosen for each input parameter and entered into FEAFAA, the simulation was run, and the results were recorded.

Simulations used a four-layer pavement system consisting of Portland cement concrete (PCC) slab, stabilized base, base/subbase, and infinite subgrade. The stabilized base was used because FAA standards require it for pavements accepting aircraft of over 100,000 lb. The nine-slab (3×3) system was jointed in both the transverse and longitudinal directions. The top tensile stress results were computed at the edges of the center slab, i.e., the design slab. This pavement configuration has three advantages: 1) it can accommodate the loads from full landing gear of most aircraft, 2) it allows for modeling the effect of adjacent slab loading on the design slab stress calculation, and 3) since the center slab is connected at all its joints, modeling the edge loading and edge stress calculation will be more accurate.

The FE analyses considered a minimum of 30 slab elements in each direction. For cases with slab width larger than 22 feet, 35 slab elements were considered in each direction. With this designation, the maximum mesh size was 8.6 inches, ensuring high accuracy in the FE analysis. Also, since FEAFAA does not accommodate non-uniform mesh, this designation ensures that the mesh size does not exceed the gear tire footprint. Thirty foundation elements were used.

As discussed, longitudinal gear location can be fixed at the critical location for each gear type. For D and 2D gear types, the critical longitudinal location is the same for both σ_{xx} and σ_{yy} calculation. However, for a 3D gear, the critical loading location for σ_{xx} is not the same as the critical loading location for σ_{yy}. Figures 15 and 16 summarize the critical gear location in the longitudinal direction that produces the maximum stress along both the longitudinal and transverse edges, for all individual gear types.

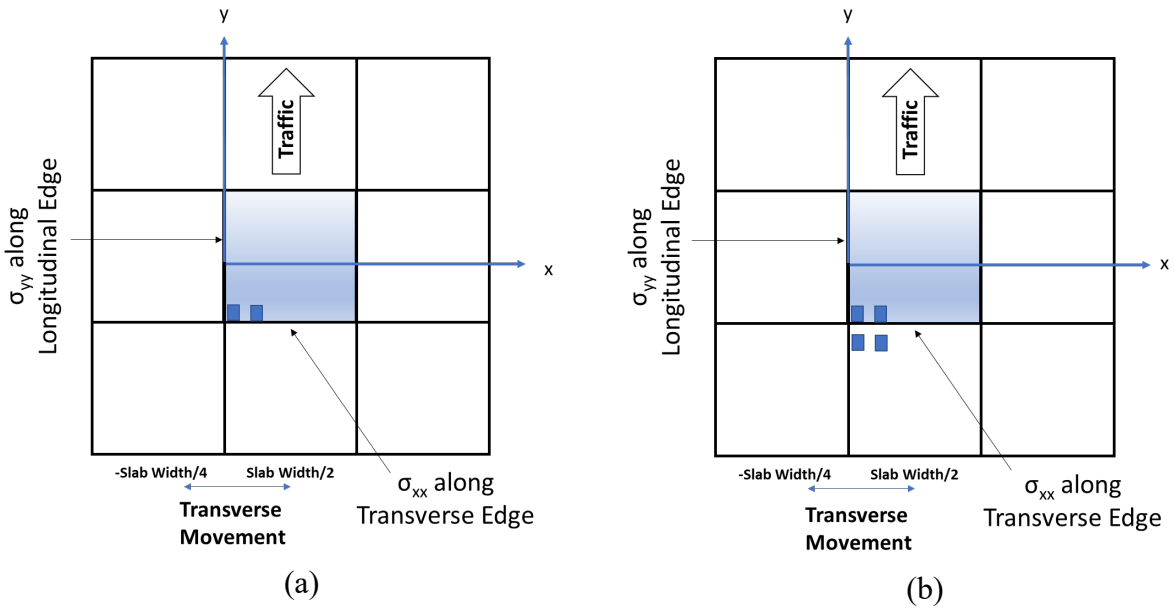
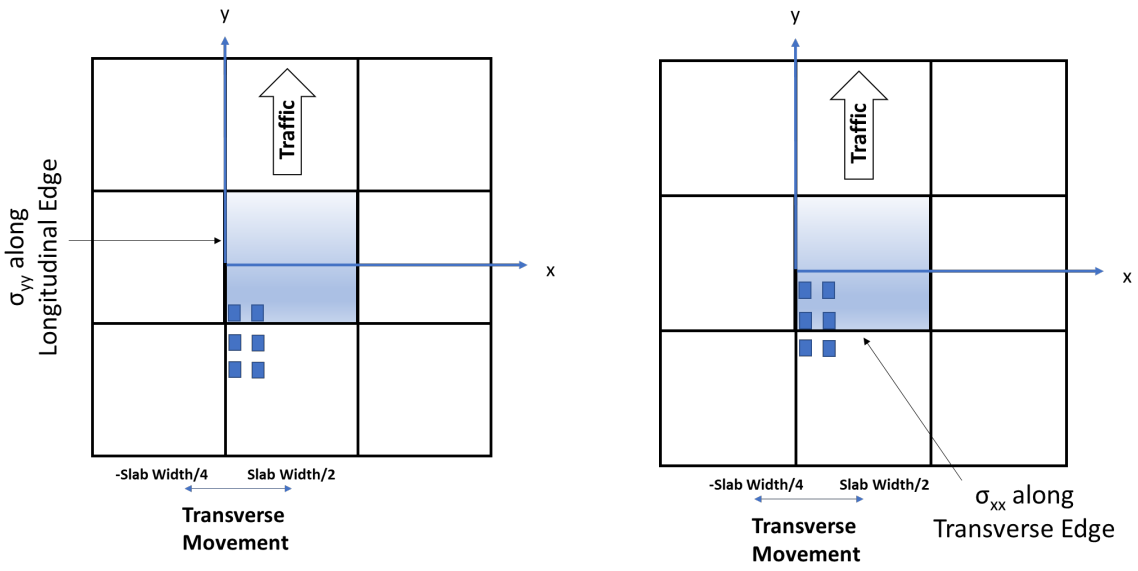


Figure 15. Critical Loading Location of (a) D and (b) 2D Gears in Longitudinal Direction to Produce the Maximum Top Tensile Stress Along both Longitudinal and Transverse Edges



(a) σ_{yy} Along Longitudinal Edge (b) σ_{xx} Along Transverse Edge

Figure 16. Critical Loading Location of 3D Gear in Longitudinal Direction to Produce the Maximum Top Tensile Stress Along (a) Longitudinal and (b) Transverse Edges

The critical gear locations for full landing gears are similar to those for the individual gears, as shown in Figure 17.

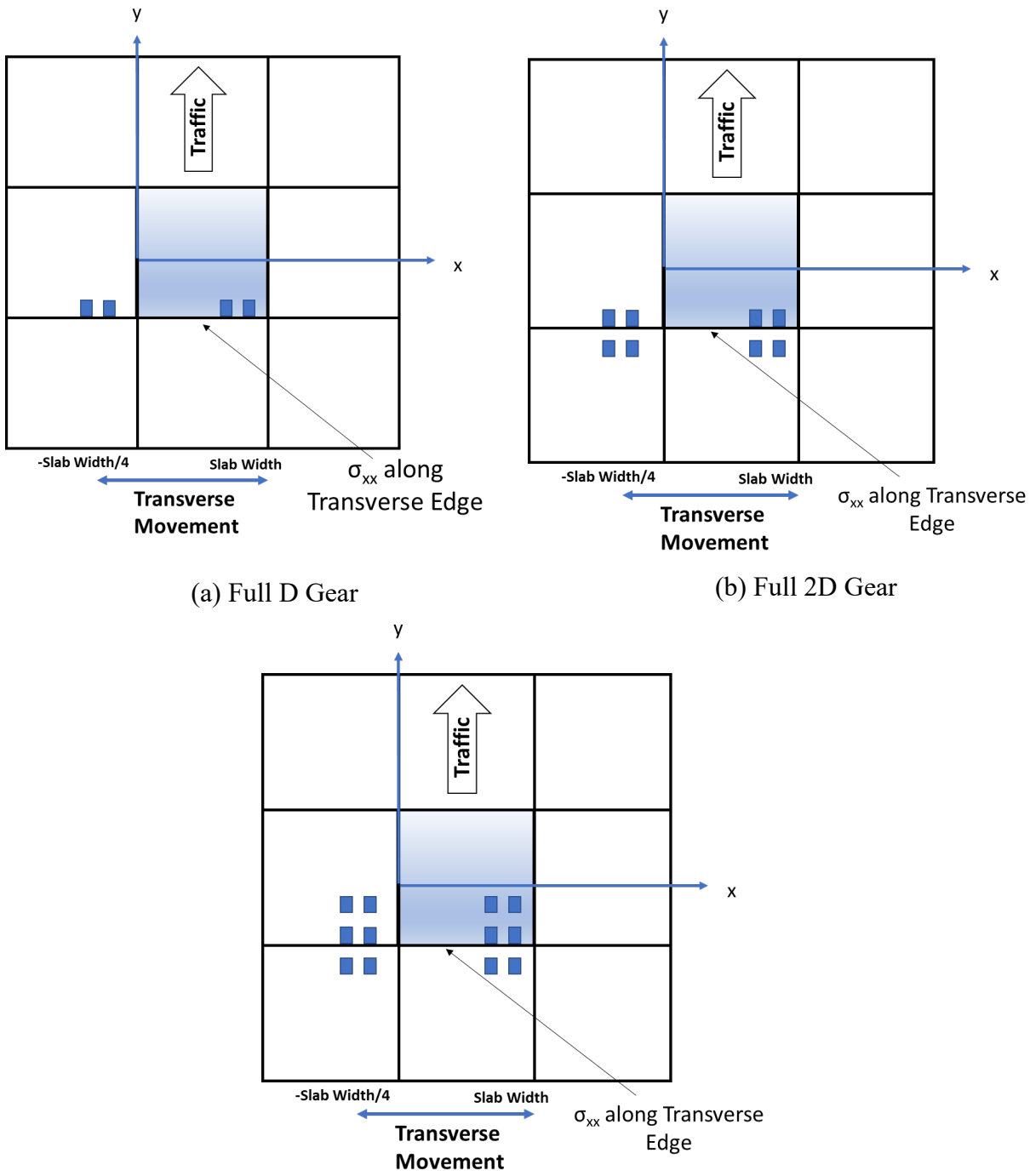


Figure 17. Critical Loading Location of Full Landing Gear in Longitudinal Direction to Produce the Maximum Top Tensile Stress in Transverse Edges

4.1 MONTE CARLO SELECTION OF INPUT PARAMETER VALUES

A Monte Carlo simulation was performed to choose the input values for each simulation. Monte Carlo methods are a class of techniques based on repeated random sampling from a probability distribution to select input values for a complex calculation. They are suitable for simulating physics-based systems such as 3D-FE modeling of rigid pavements that have large amounts of uncertainty and variability in the FE model parameters, and input parameter combinations that are too numerous to sample exhaustively. Uniform distributions were used for each input parameter to increase the generalization and accuracy of the trained models across the input space.

A range was defined for each parameter to establish the boundaries of the uniform distributions. The range of each parameter was selected to be representative of typical rigid airfield pavement structures and in compliance with the FAA pavement design guidelines. Values were generally taken from advisory circulars or the FAARFIELD user manual. The ML models will only be valid over the ranges defined for each input, so researchers extended the range of each parameter to account for potential changes to aircraft and materials in the future.

4.1.1 Parameter Ranges

4.1.1.1 Pavement Properties

The range of values for each pavement property considered by the ML model is shown in Table 1 through Table 3.

Table 1. Ranges of Input Parameters Used for the Training Database

Input Variable	Minimum	Maximum
PCC Modulus (psi)	4,000,000	6,000,000
Stabilized Base Modulus (psi)	100,000	800,000
Base/Subbase Modulus (psi)	20,000	90,000
Subgrade Modulus (psi)	4,500	30,000
PCC Thickness (in.)	8	24
Stabilized Base Thickness (in.)	5	10
Base/Subbase Thickness (in.)	6	30
Slab Width (ft)	(see Table 2)	
Slab Length (ft)	0.85 × Slab Width	1.25 × Slab Width (Not more than 25 ft)
EJS (psi)	110,000	315,000
ELTG (°F/in.)	-2	0.5
CTE (1/°F)	4.0E-06	6.0E-06

Table 2. Limits of Slab Width Based on Slab Thickness

Slab Thickness (in.)	Slab Width (Transverse Joint Spacing)	
	Minimum (ft)	Maximum (ft)
≤ 13	12	20
> 13	12	25

Table 3. Poisson's Ratio for Each Pavement Layer

Pavement Layer	Poisson's ratio
PCC	0.15
Stabilized Base	0.2
Base/Subbase	0.35
Subgrade	0.4

The ranges were determined by extending the limits of the recommended values from the FAARFIELD library and FAA AC 150/5320-6G to cover a wider range of pavement properties. For example, the defined range for the stabilized base layer modulus was extended beyond the minimum value of 250,000 psi (P-301 Soil cement) and the maximum value of 700,000 psi (P-306 Lean Concrete) as recommended in Table 3-2 of AC 150/5320-6G. Although the PCC modulus is fixed at 4,000,000 psi in FAARFIELD, a wider range of modulus values was considered to allow for design of high strength concrete slabs. The minimum layer thickness was determined from Table 3-4 of AC 150/5320-6G. The ranges for joint spacing (slab dimensions) were determined based on the slab thickness to include more realistic cases in the database. Table 3 shows the limits considered for slab width (longitudinal joint spacing) according to the slab thickness. The range for slab length (transverse joint spacing) was determined to be from 0.85 to 1.25 the slab width. EJS values were selected by using the full range of dowel diameters and spacing permitted by Table 3-6 and Table 3-7 of AC 150/5320-6G to determine equivalent EJS values at the extremes of the range. The lower end of the EJS range is appropriate for non-doweled joints, i.e., dummy joints or aggregate interlock.

4.1.1.2 Thermal Gradient

The range considered for the ELTG was between -2 and 0.5 (°F/in.). This covers the typical spectrum of negative temperature gradient in airport pavements and a low positive ELTG that may still contribute to top-down cracking. For an ELTG higher than 0.5 °F/in., the top tensile stresses are too small to contribute to the top-down cracking design. For cases with higher positive ELTG values, the designer may choose to use a value between 0–0.5 °F/in. to perform a more conservative pavement design for top-down cracking. The range considered for coefficient of thermal expansion (CTE) is also shown in Table 1.

4.1.1.3 Aircraft Gear Configuration

Aircraft gear parameter ranges were taken from the FAARFIELD aircraft library. Figure 18 shows box plots of the four main parameters that encode the configuration of each wheel group of a gear: gear weight, tire pressure, dual spacing, and tandem spacing.

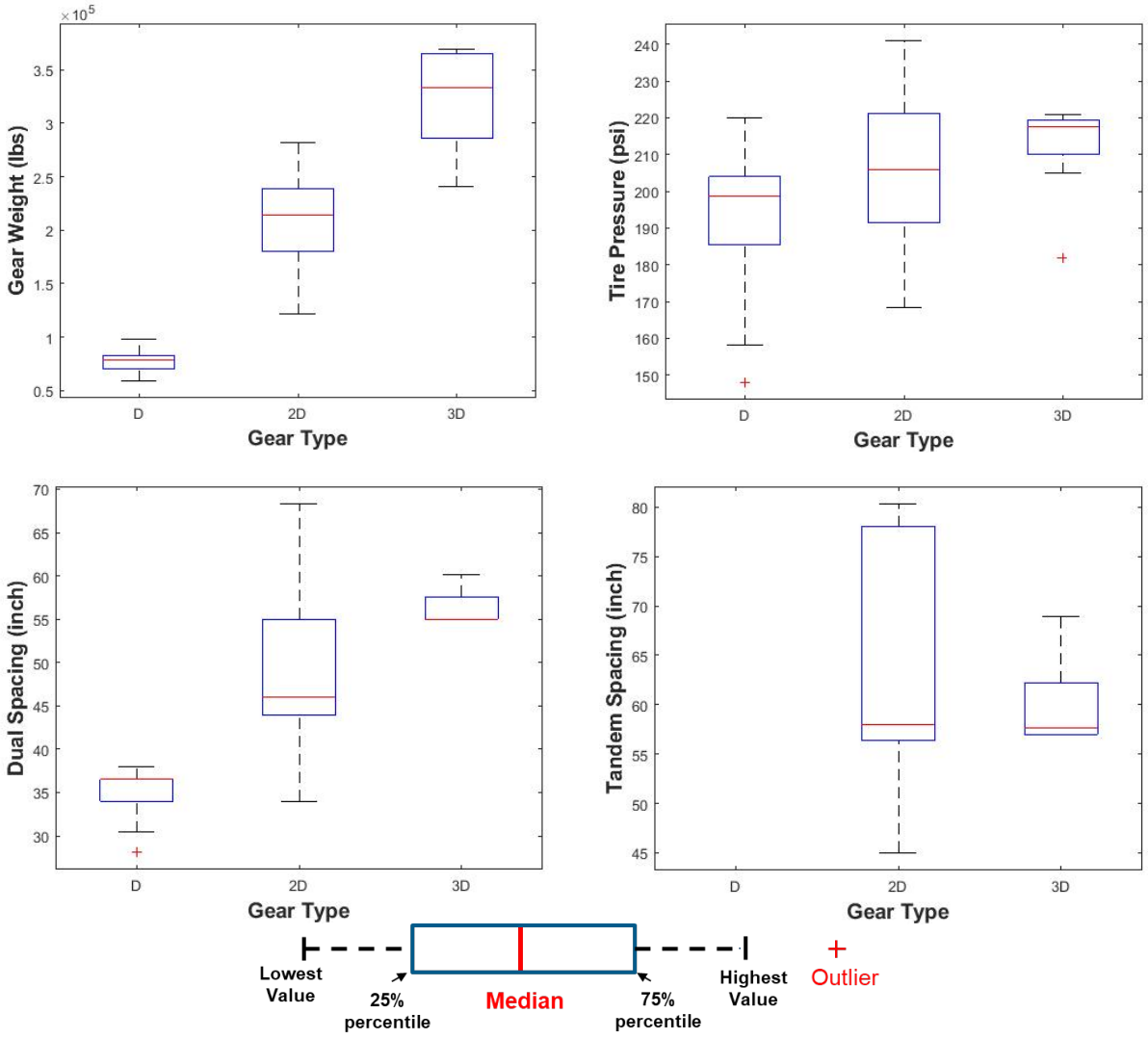


Figure 18. Ranges of the Gear Parameters in the FAARFIELD Aircraft Library

The box plots provide a visual summary of the extent and distribution of gear parameters of all aircraft types in the FAARFIELD library with the gross weight greater than 100,000 lb. The ranges selected are provided in Table 4.

Table 4. Selected Ranges of Gear Parameters

Gear Type	Parameter	Minimum	Maximum
D	Gear Weight (lb)	50,000	105,000
	Tire Pressure (psi)	150	240
	Dual Spacing (in.)	28	38
2D	Gear Weight (lb)	120,000	300,000
	Tire Pressure (psi)	170	250
	Dual Spacing (in.)	34	70
	Tandem Spacing (in.)	(see Table 6)	
3D	Gear Weight (lb)	240,000	380,000
	Tire Pressure (psi)	180	250
	Dual Spacing (in.)	55	65
	Tandem Spacing (in.)	1 × Dual Spacing	1.3 × Dual Spacing

Note that data showed that there are some dependencies between the dual and tandem spacing in 2D gear, as shown in Figure 19.

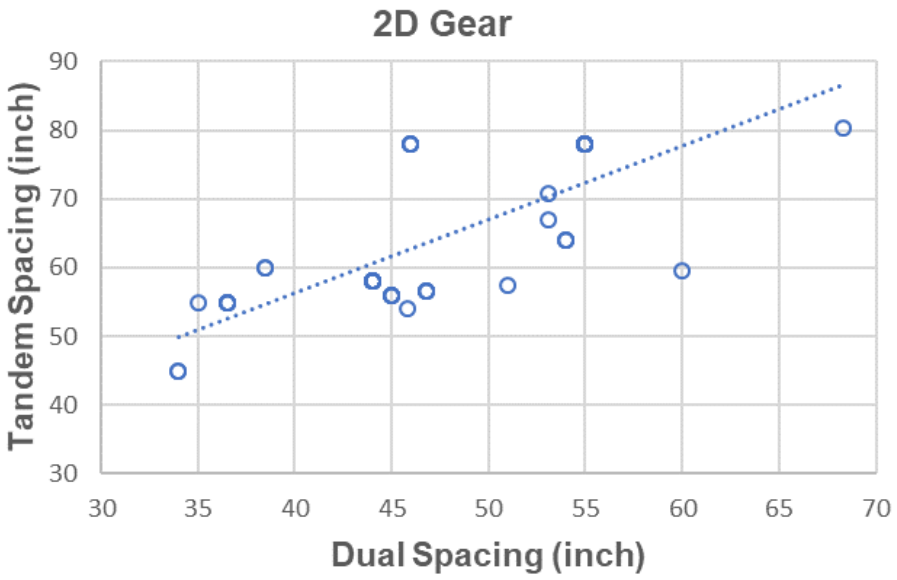


Figure 19. Dual and Tandem Spacing of 2D Gear Types in the FAARFIELD Library

The trend suggests that an increase in dual spacing is typically associated with an increase in tandem spacing. Additionally, the ratio between dual and tandem spacing tends to decrease as the dual spacing increases, as shown in Figure 20.

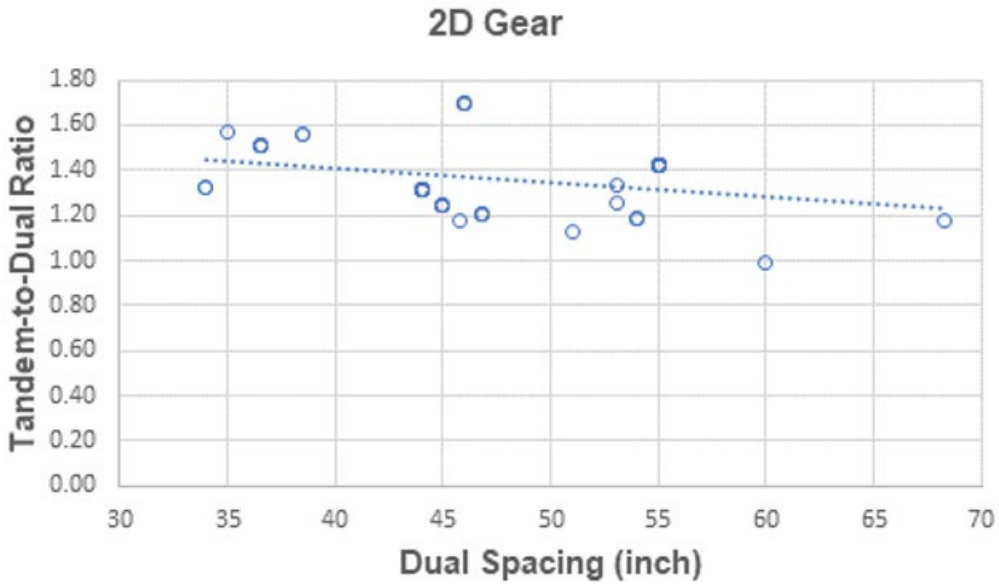


Figure 20. Tandem-to-Dual Spacing Ratio vs Dual Spacing

To include more realistic cases representing an actual aircraft gear, the ranges for tandem spacing was determined based on the dual spacing, as shown in Table 5.

Table 5. Ranges Defined for Tandem Spacing

Dual Spacing (in.)	**Tandem Spacing (in.)**
≤ 50	(1.2 – 1.7) × Dual spacing
50 – 60	(1.1 – 1.45) × Dual spacing
≥ 60	(1.0 – 1.3) × Dual spacing

The ML model is a general model which includes the cases in which multiple wheel groups are on the same slab. This can happen for some of the smaller aircraft and for aircraft with belly gear. Under certain pavement geometry and landing gear configurations, the tensile stress on top of the slab due to a pass of full landing gear is significantly higher along the transverse edge than tensile stress due to an individual wheel group. An individual and a full landing gear of the same type will likely result in almost similar σ_{yy} response along the longitudinal edge. Therefore, the full landing gear configuration is used only for σ_{xx} prediction along the transverse edge. The full landing gear will likely increase the chance of top-down cracking.

Researchers determined that the interaction of the wing and belly gear when on the same slab was unlikely to cause the critical edge stress. FEAFAA simulations of a full A380 landing gear (20 wheels) showed that the combined loading of a wing and belly gear on the same slab produced approximately the same stress distribution as the individual wheel groups, possibly due to their longitudinal offset. A comparison is shown in Figure 21. The FAARFIELD procedure of treating each wheel group as a separate aircraft can likely continue to be used.

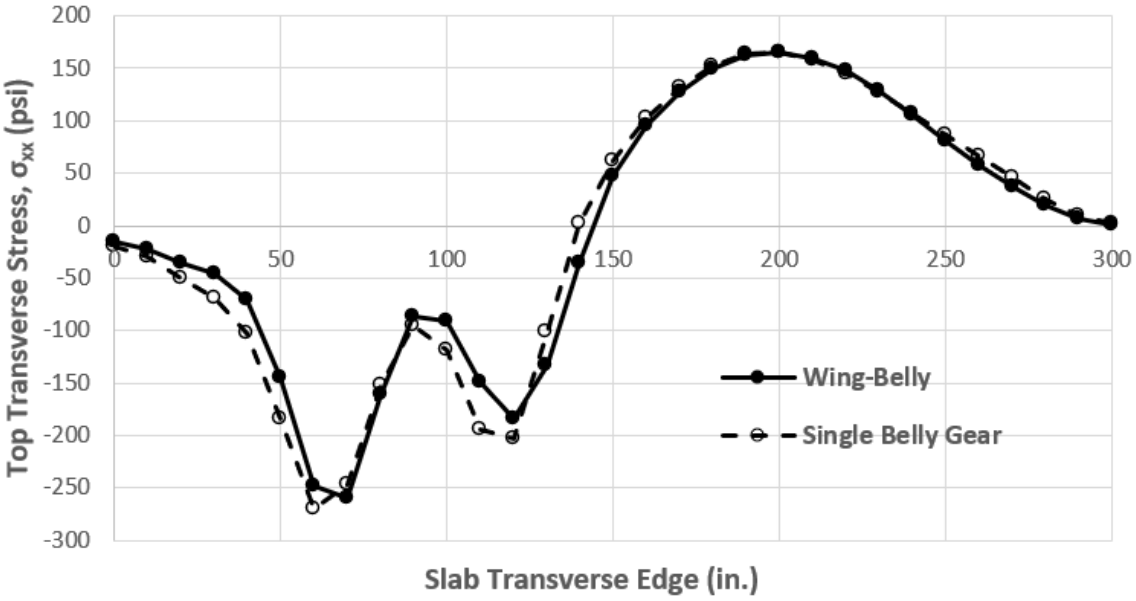


Figure 21. Comparison of Stress Distribution Along Transverse Edge on Top of the Slab Due to an A380 Wing and Belly Gear Versus Only the A380 Belly Gear

Since the interaction of wing and belly gear are unlikely to be the critical case, only aircraft with track spacing (center-to-center of two belly gear or two landing gear) smaller than 25 feet were considered for the full landing gear cases. It is very unlikely that the two landing gear load the same slab when the tracking space is greater than 25 feet. The parameter ranges were reduced when modeling full gear with the expectation that this would reduce the number of training cases needed for an accurate model by reducing the complexity of the input. The information from the following aircraft were considered as the basis to create cases with full landing gear configurations: two 3D belly gear of A-380, two 2D belly gear of B-747, and two D gear of B737 class and A318/A319/A320/A321 class full landing gear. Table 6 summarizes the parameters and their ranges considered for the full landing gear cases.

Table 6. Ranges of Gear Parameters Used for Full Landing Gear

Gear Type	Parameter	Minimum	Maximum
Two D-Gear	Total Gear Weight (lb)	110,000	210,000
	Tire Pressure (psi)	170	240
	Dual Spacing (in.)	28	38
	Track Spacing (in.)	190	300
Two 2D-Gear	Gear Weight (lb)	330,000	490,000
	Tire Pressure (psi)	190	240
	Dual Spacing (in.)	40	52
	Tandem Spacing (in.)	1.2 × Dual Spacing	1.4 × Dual Spacing
	Track Spacing (in.)	130	170
Two 3D-Gear	Gear Weight (lb)	670,000	750,000
	Tire Pressure (psi)	210	240
	Dual Spacing (in.)	56	64
	Tandem Spacing (in.)	1 × Dual Spacing	1.2 × Dual Spacing
	Track Spacing (in.)	185	225

4.1.1.4 Aircraft Gear Transverse Position

The range for aircraft gear transverse position was defined such that it will result in the full output range of σ_{xx} along the transverse edge, including the stresses transferred from the adjacent slab through joint connections. It was determined that loading farther left than a quarter width of the adjacent slab has minimal effect on σ_{xx} in the design slab. The right-hand boundary is defined as the center of the design slab due to symmetry. For full landing gear, the left and right boundaries of X-offset are determined by the location of the left and right gear, respectively. The left boundary is where the center of the left gear is at quarter width of adjacent slab, and the right boundary is where the center of the right gear is at the right edge of the target slab. Figures 22 and 23 graphically shows the aircraft gear transverse position range, and Table 7 provides the formulae to calculate the ranges.

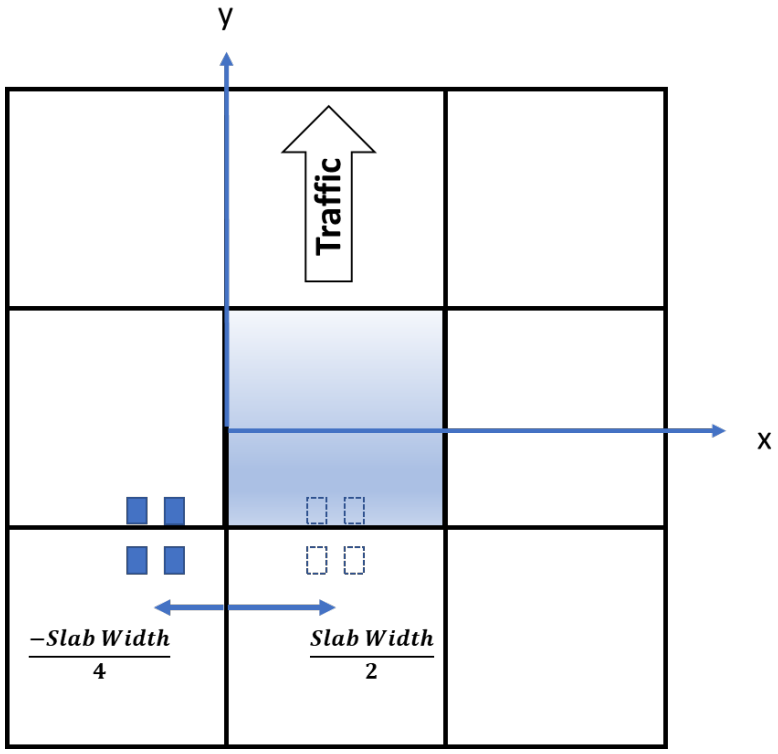


Figure 22. Range of Transverse Movement for Individual Gear

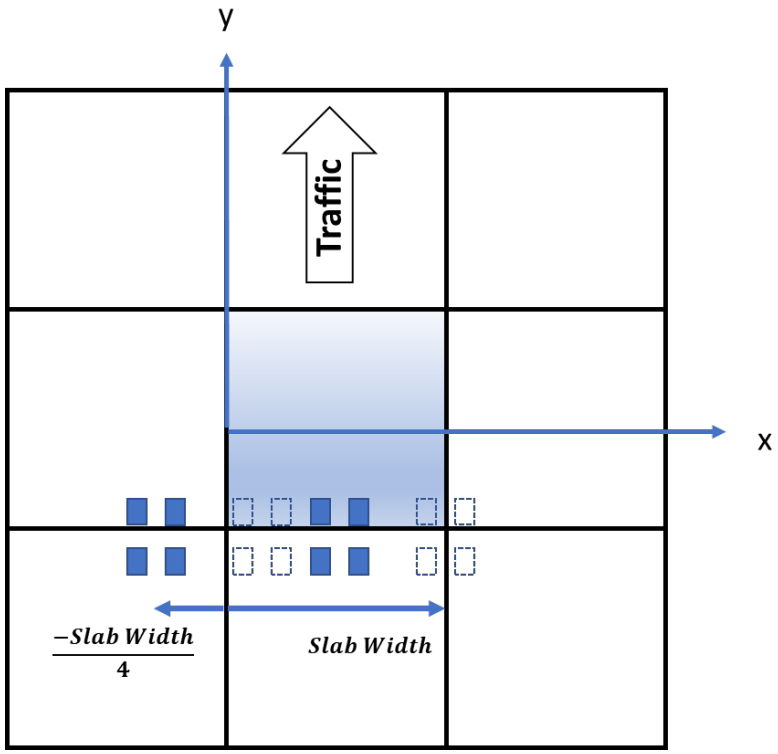


Figure 23. Range of Transverse Movement for Full Landing Gear

Table 7. Gear Transverse Position Range

Parameter	Left Boundary	Right Boundary
Gear X-offset for Individual Gear (in.)	$-\frac{\text{Slab Width} \times 12}{4}$	$\frac{\text{Slab Width} \times 12}{2}$
Gear X-offset for Full Landing Gear (in.)	$\frac{\text{Track Spacing}}{2} - \frac{\text{Slab Width} \times 12}{4}$	$(\text{Slab Width} \times 12) - \frac{\text{Track Spacing}}{2}$

4.1.2 Monte Carlo Process

A GNU Octave[2] script was developed to perform the Monte Carlo simulation. It generated a matrix of input values, with each row of the matrix containing the inputs for one 3D-FE simulation. The script uses the "Mersenne Twister" algorithm as its pseudorandom number generator. Below is the procedure for generation of a case:

1. Randomly select a value for each pavement parameter from the predefined ranges in Table 1, assuming a uniform probability distribution.
2. Randomly select a number between 1 and 3 to determine the gear type (D, 2D, or 3D).
3. Based on the gear type selected in step 2, randomly select values for the four gear parameters from the predefined ranges in Table 4, assuming a uniform distribution of each parameter.
4. Randomly select a transverse position from the range defined in Table 7, assuming uniform distribution for each parameter.
5. Record the results in the Input Matrix, in the appropriate Microsoft® Excel® spreadsheet.

This process was repeated as required to generate the requisite number of simulations. Note that the gear parameters are independently selected, and as such, do not necessarily represent a real aircraft. This approach increases the robustness of the model with very little trade-off in accuracy. It also eliminates the need to retrain the ANN model each time an aircraft with a new gear configuration is released.

The Input Matrix contains 102,000 cases for individual landing gear and another 25,000 cases for full landing gear. Figure 24 is a sample of lines from the Input Matrix.

[2] GNU Octave is a numerical analysis program. Its scripting language is compatible with MATLAB®.

Case ID	PCC Modulus	Base Modulus	Subbase Modulus	Subgrade Modulus	PCC Thickness	Base Thickness	Subbase Thickness	Slab Width	Slab Length	ELTG	CTE	EJS	Gear Type	Gear Weight	Tire Pressure	Dual	Tandem	Gear x-offset
Case1D_LT_1	5367299	309381	32142	9809	20.5	9	7.5	21.5	20	-0.73	4.7E-06	223891	1	61605	204.9	32.4	0	4.8
Case1D_LT_2	5279593	220838	88795	23656	11	9	20.5	14	12	-0.93	4.6E-06	111156	1	80592	160.6	30	0	12.6
Case1D_LT_3	5847213	708784	80996	10391	17	5.5	20.5	24.5	25	-1.19	4.4E-06	219633	1	68292	208.9	30.9	0	-55.3
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
Case2D_LT_7070	4029698	192267	28131	28750	8.5	9.5	9.5	12	12	-1.18	5.8E-06	131673	2	269102	230	61.2	67.2	38.1
Case2D_LT_7071	4971318	788405	74313	29373	20.5	6	26	19	18.5	-1.07	4.2E-06	265708	2	219333	206	66.7	84.6	10
Case2D_LT_7072	4990012	507594	79339	24939	17	8	25	16	18	-1.82	5.6E-06	236854	2	157164	178	43.8	61.6	-21
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
Case3D_L_13606	4258295	637082	66437	5173	21.5	5.5	22	21	21	-0.16	5.1E-06	228311	3	325738	238	57	64.9	107.7
Case3D_L_13607	5921611	380629	28619	25798	8.5	8.5	15.5	14.5	14.5	0.3	4.6E-06	143652	3	372879	220	63.7	67.9	71.5
Case3D_L_13608	5111662	496623	54270	15369	23.5	9.5	23	18.5	19.5	-0.48	5.3E-06	225019	3	242107	242	61.9	72.6	-16.1
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
Case3D_T_14925	5012118	144580	57278	9425	9	8.5	19.5	14	13	-0.75	4.7E-06	139824	3	267131	229	57.2	68	-34.2
Case3D_T_14926	4479396	636205	22908	25233	16	7	28.5	21	20.5	-0.53	5.9E-06	158857	3	274113	222	61	68.7	121.4
Case3D_T_14927	4880231	716293	51333	24462	16	7.5	27.5	17	18	-0.73	5.4E-06	146051	3	325327	194	57.5	60.8	88.6
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.

Figure 24. Sample Cases from the Input Matrix

Each row in the Input Matrix is identified with a case ID. A typical case ID is “Case2D_LT_7286”. “L” and “T” in the ID stands for longitudinal and transverse, respectively. “LT” means that the case can be used for the calculation of stress along both longitudinal and transverse edges. All the cases for the D and 2D gear types contain “LT” in their IDs. Case IDs for full landing gear contain “Belly” in their name, e.g., “Belly2D_LT_3190”. Due to the different critical positions of 3D gear for longitudinal and transverse stress, each case with a 3D gear was good only for either transverse or longitudinal, not both, e.g., “Case3D_L_9100” or “Case3D_T_9335”.

4.2 SETUP FOR FEAFAA SIMULATIONS

FEAFAA uses Extensible Markup Language (XML) format job files to define each 3D-FE simulation. The files contain all the information needed to perform a 3D-FE simulation, including the pavement structure and aircraft configuration. The FEAFAA user normally defines and controls a simulation by modifying these parameters through a graphical user interface, but XML is a text-based format easily generated by scripts. Simulations can be programmatically defined by generating a suitable XML file and loading it into FEAFAA.

Files were generated by taking a template file and modifying the contents based on the values in the input matrix. Researchers created the template files by manually defining a job in FEAFAA with an arbitrary gear configuration and a nine-slab, four-layered pavement system subjected to a thermal load. The template files were saved by clicking on “Save Job” under the “3D Mesh Generation” tab. One template was created for each gear type (D, 2D, 3D). An Octave script then copied the template, modified it to reflect the input parameters in a single row of the Input Matrix, saved the file, and moved to the next row. The script generated one input file for each row of the Input Matrix. The FEAFAA files are included as an electronic attachment to this report.

Figure 25 shows examples of the “Airplane Selection” of the job files for a 2D and a 3D gear. The <X> and <Y> tags represent the tire coordinates.

```
<?xml version="1.0" encoding="utf-8"?>
<FEAFAAJobInfo>
  <AirplaneSelectionTab>
    <AirplaneGroup>6.00</AirplaneGroup>
    <AirplaneIndex>0.00</AirplaneIndex>
    <GrossWeight>234233</GrossWeight>
    <PcntOnMainGears>100</PcntOnMainGears>
    <NMainGears>1</NMainGears>
    <NWheels>4</NWheels>
    <TirePressure>222</TirePressure>
    <WheelCoordinates>
      <X>0</X>
      <Y>-31.4</Y>
      <X>0</X>
      <Y>31.4</Y>
      <X>71.7</X>
      <Y>-31.4</Y>
      <X>71.7</X>
      <Y>31.4</Y>
```

(a)

```
<?xml version="1.0" encoding="utf-8"?>
<FEAFAAJobInfo>
  <AirplaneSelectionTab>
    <AirplaneGroup>6.00</AirplaneGroup>
    <AirplaneIndex>1.00</AirplaneIndex>
    <GrossWeight>377985</GrossWeight>
    <PcntOnMainGears>100</PcntOnMainGears>
    <NMainGears>1</NMainGears>
    <NWheels>6</NWheels>
    <TirePressure>227</TirePressure>
    <WheelCoordinates>
      <X>-68.2</X>
      <Y>-28.75</Y>
      <X>-68.2</X>
      <Y>28.75</Y>
      <X>0</X>
      <Y>-28.75</Y>
      <X>0</X>
      <Y>28.75</Y>
      <X>68.2</X>
      <Y>-28.75</Y>
      <X>68.2</X>
      <Y>28.75</Y>
```

(b)

Figure 25. Aircraft Portion of Sample FEAFAA Job Files for (a) 2D and (b) 3D Gears

For each database case, the dual spacing and tandem spacing from the Input Matrix were used to calculate the tire coordinates of individual gear. Full landing gear cases also used track spacing to calculate tire coordinates. In FEAFAA, the gear axis coordinate system is different than the pavement axis coordinate system. Appendix A contains tables illustrating the calculation of tire coordinates of the individual gear and the full landing gear.

As discussed in Section 3, the longitudinal gear position was not considered a training variable, rather its value was fixed according to the critical gear position. This parameter is identified with the tag <GearYoffset> in the XML file. The gear Y-offset for each case was calculated based on the slab length, tandem spacing (TS), and the tire length. The equations used for calculation of the longitudinal gear position are included in Appendix B. A sample FEA.XML file is included in Appendix C.

4.3 BATCH OPERATION OF FEAFAA

To perform the simulations, programmers modified FEAFAA to support batch analysis. The added functions call the FEAFAA buttons code-behind, which simulates a user pressing the correct sequence of buttons in the interface. The functions also allow the job files to be accepted as command line parameters. A separate program with a graphical interface, called "Batch_Controller," was developed in Visual Basic to call the added functions and control sequential execution of job files. The Batch_Controller interface is shown in Figure 26.

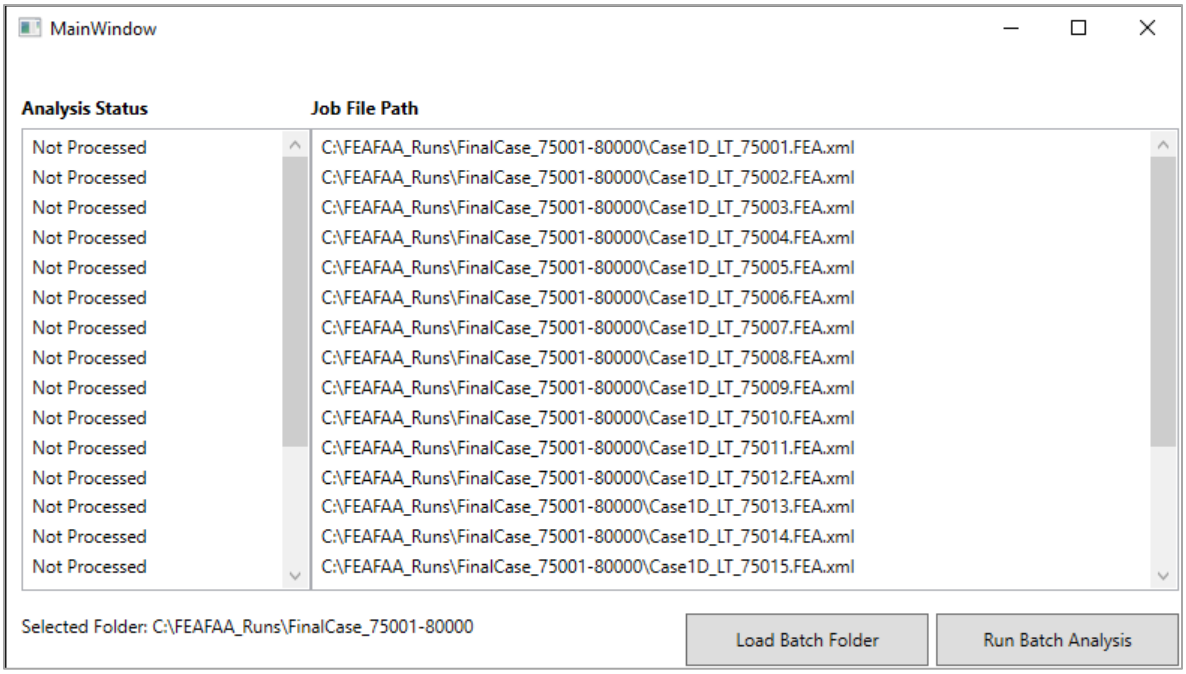


Figure 26. Graphical Interface of Batch Analysis Program

The program accepts the directory containing a group of job files, sequentially loads each one into FEAFAA and executes the job, then saves the output files. Batch_Controller saves FEAFAA output from each case in separate subdirectories called *PrintOut-<JobName>*. While all FEAFAA results are saved for completeness, only two output files were used in this study: "*model_load.dat*" and "*modal_stress_1.dat*". After all the job files are processed, Batch_Controller copies these files from each *PrintOut* folder to a new folder and renames them to reflect the case ID. For example, the output files for "Case2D_LT_1802" were renamed "Case2D_LT_1802_model_load" and "Case2D_LT_1802_modal_stress_1". A cluster of computers was dedicated to completing the FEAFAA simulations required to construct the database. The FEAFAA output files are included as electronic attachment to this report.

4.4 POST-PROCESS 3D-FE OUTPUT INTO A DATABASE

GNU Octave scripts post-processed the 3D-FE stress distribution results into a database format suitable for training ML models. The *model_load.dat* files contain information about the 3D-FE model, including nodal coordinates, load, and boundary conditions. The *model_stress_1.dat* files contain the nodal displacements (coordinates of the deformed pavement elements) and the six components of computed stress extrapolated to the nodes, including bending stresses in the X and Y directions (σ_{xx} and σ_{yy}), compressive stresses in the Z direction (σ_{zz}), and the in-plane and out-of-plane shear stresses (σ_{xy}, σ_{yz},, and σ_{zx}). The post-processing script reads the *model_load.dat* and *model_stress_1.dat* files for each FEAFAA simulation and calculates the nodal stresses along the longitudinal (σ_{yy}) and transverse (σ_{xx}) edges on the top surface of the design slab. Engineers reviewed the processed data to ensure the FE simulations converged properly, and that the resulting stresses were both reasonable and consistent with the project goal. Sample results from "Case2D_LT_7072" are shown in Figures 27 and 28.

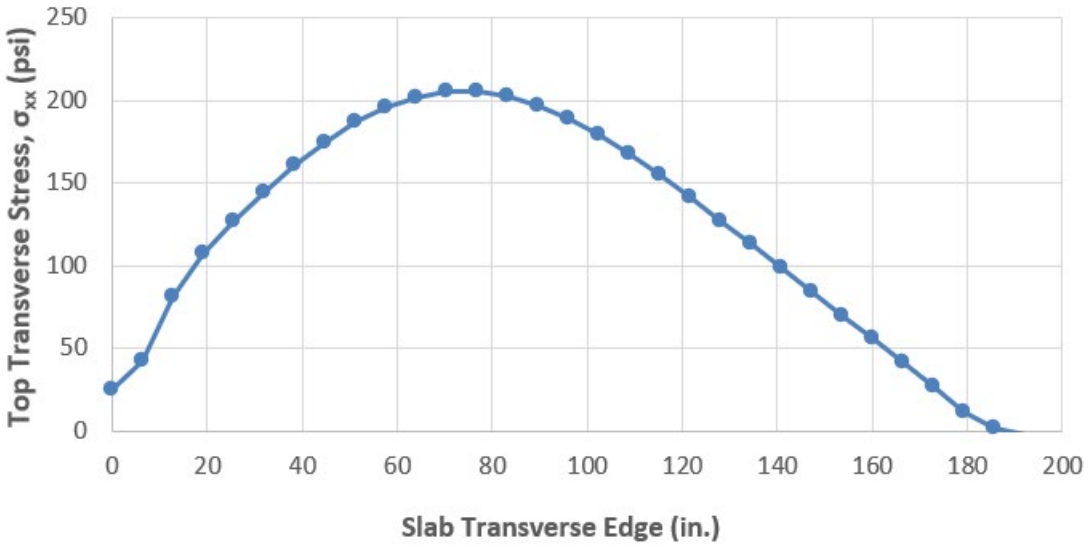


Figure 27. Stress Distribution Along Transverse Edge on Top of the Design Slab in Case2D_LT_7072

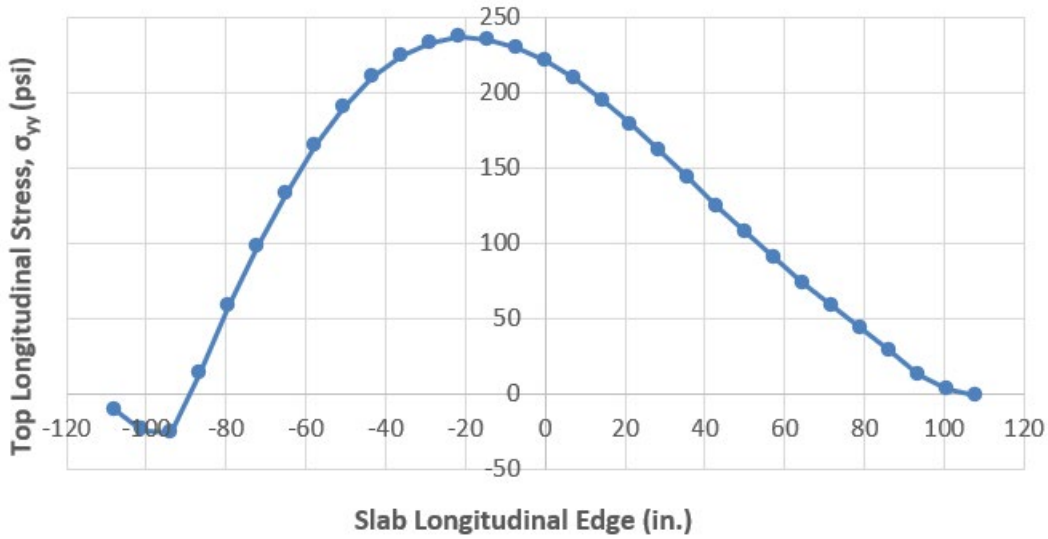


Figure 28. Stress Distribution Along Longitudinal Edge on Top of the Design Slab in Case2D_LT_7072

The gear location for the example is shown in Figure 29.

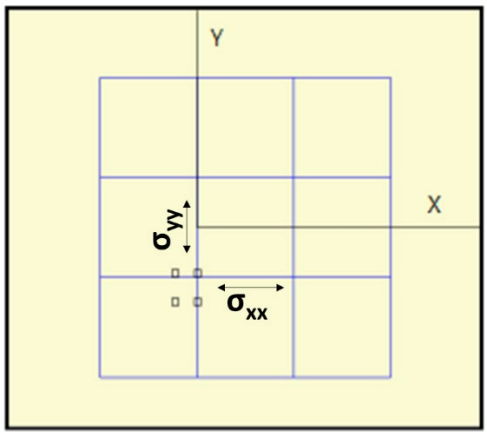


Figure 29. Pavement and Gear Configurations in Case2D_LT_7072

The script writes the calculated stress distributions to an Excel spreadsheet file to construct the database *Output Matrix*. *Output Matrix* contains the nodal coordinates of the transverse and longitudinal edges of the design slab and the calculated stress.

5. MACHINE LEARNING MODEL FOR STRESS DISTRIBUTION ESTIMATION

5.1 MODEL OVERVIEW

Researchers developed an ML model to estimate the stress distribution along slab edges for any gear configuration within the parameter ranges identified in Section 4.1.1.3. The model was trained using the data generated by the finite element simulations described in Section 4. A modular deep learning method was used to improve accuracy over similar works that only employed simple ANNs. The term "deep" refers to the ability of modern ANNs to have architectures comprised of a large number of neuron layers stacked in sequence, compared to classical ANNs' effective limit of only two or three layers. An increased number of layers greatly improves an ANN's representative and processing power. This new approach used a collection of mechanisms specifically chosen both to optimally encode the simulation inputs and to effectively represent and process the information of the underlying physical phenomena.

The presented model is one singular architecture with a common interface for all gear types in the data set. The model encodes all configurations by parameterizing five properties of each aircraft gear and the gear type. Modern ANNs make this possible through recent advancements that have improved their ability to input large sets of diverse information and approximate discontinuous functions. They use explicit "switching" mechanisms that are specialized for the processing of disparate information (e.g., attention mechanisms) (Alom et al., 2019).

The architecture, illustrated in Figure 30, was designed to be fully differentiable, i.e., every mechanism and mathematical operator in the model has a defined gradient function. This allows the backpropagation algorithm that performs the training to be applied end-to-end across all trainable variables of the model.

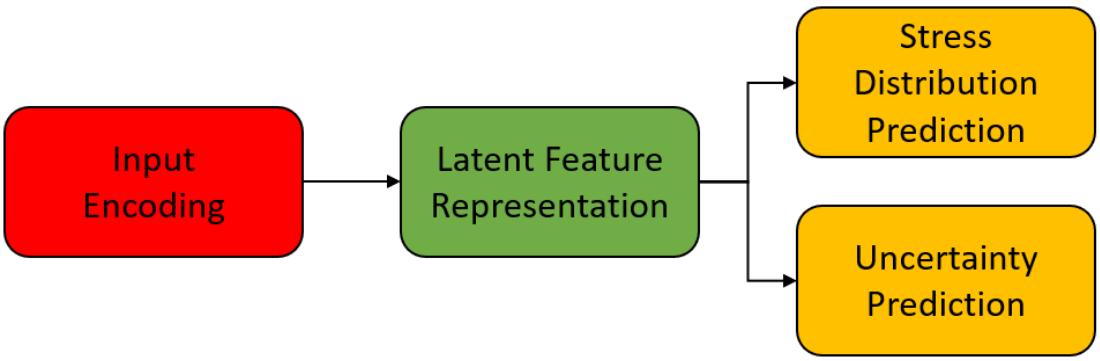


Figure 30. Overview of Machine Learning Model Architecture

The model was developed using the PyTorch open-source machine learning framework. The model has three main modules: Input Encoding, Stress Distribution Prediction, and Uncertainty Prediction. The input encoding module takes the input parameters, regardless of aircraft type and configuration, and processes the data into a common latent feature representation. The latent representation is a highly compressed form of the data that conserves the salient information that is relevant to this problem. The model passes the latent features to the stress distribution prediction module, which outputs the stress predictions, and the uncertainty prediction module, which outputs an uncertainty metric that attempts to predict the accuracy of the stress prediction.

5.2 INPUT ENCODING MODULE

The input encoding module reduces the input data into a latent feature representation. This module has two functions: 1) it compresses the inputs down to the information determined to be most valuable for this problem, and 2) it creates a common interface for the inputs of all aircraft, regardless of landing gear configuration. The architecture of this module is shown in Figure 31.

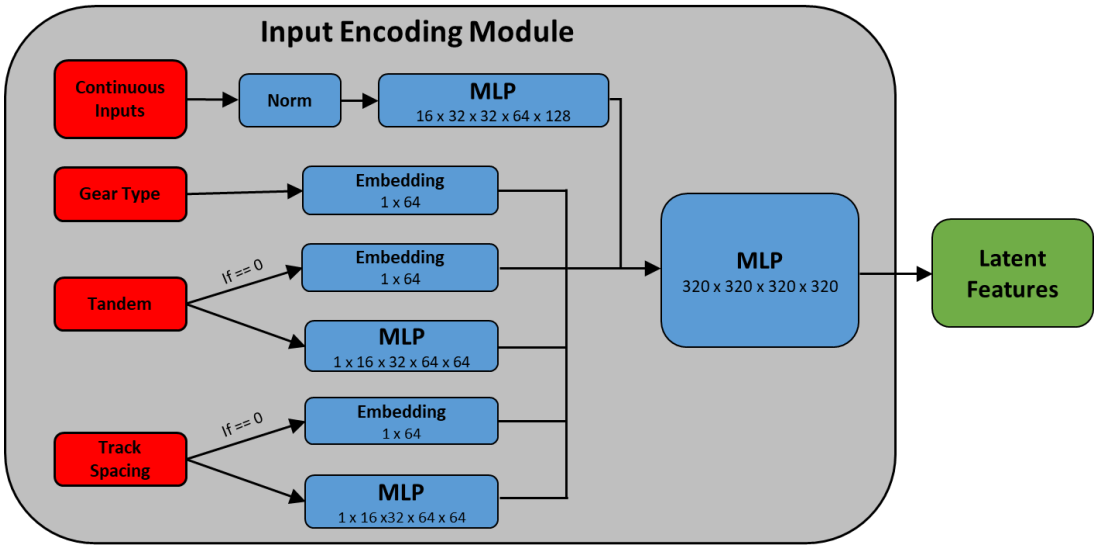


Figure 31. Architecture of Input Encoding Module

Multilayer perceptrons (MLPs), also known as fully connected feedforward neural networks, perform the compression. The model forces the MLPs to reduce the relatively large input space to a much smaller output space, and through training, they learn to do this while preserving the salient information needed for the regression task.

A system of embeddings and MLP operators provide a common interface to ensure that a single model can be used for any aircraft with gear parameters within the defined ranges. An embedding is a trainable continuous tensor used to represent discrete variables. The embedding is used here to translate the discrete gear types, the existence of wheels in tandem, and the existence of a track spacing parameter, to a vector that can be processed by an MLP. Other input variables already represented by continuous values are processed directly by MLPs, including the tandem and track spacings when they exist. The continuous variables are also normalized as per common practices in training neural networks.

The use of a single model is also beneficial to the overall model accuracy. Since the underlying physics is the same between all aircraft, the model exploits a concept called transfer learning to share common knowledge between training cases and improve predictions by generalizing.

5.3 STRESS DISTRIBUTION PREDICTION MODULE

The stress distribution prediction module calculates the final stress distribution along the slab edges from the latent feature vector. This is done through a novel method that trains the model to output the weights of a linear transformation that is applied to the latent features, which then produces the stress predictions. The module is split into two identical paths so that the stress distributions for each dimension (σ_{xx} and σ_{yy}) are predicted separately. This architecture is illustrated in Figure 32.

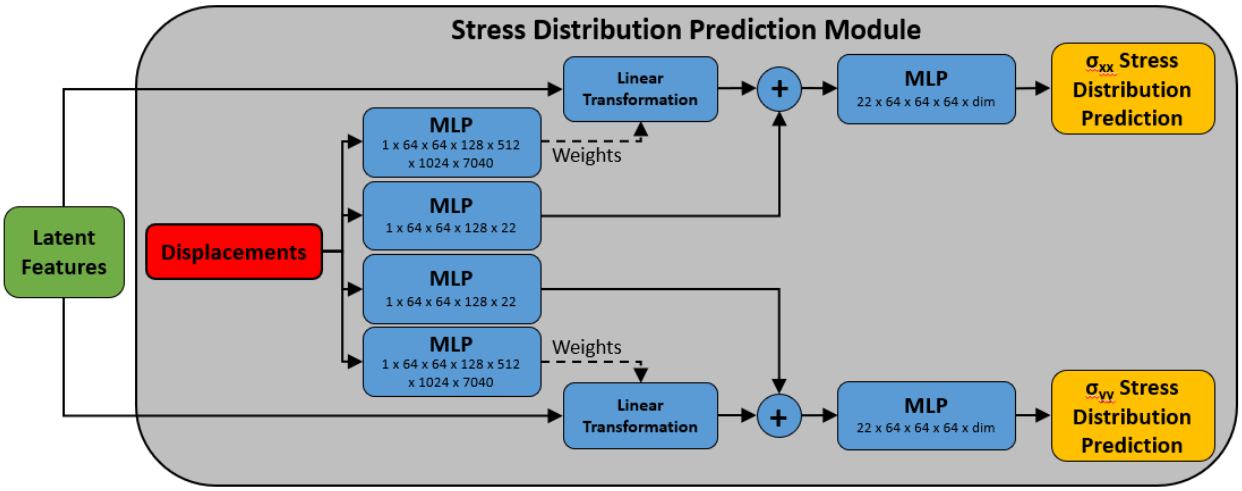


Figure 32. Architecture of Stress Distribution Prediction Module

In standard deep learning regression models, which are typically constructed using MLPs or summations of waves in a basis, e.g., discrete wavelet kernels, the model is typically constrained by the need to specify a fixed number of outputs. This constraint creates a significant obstacle for

the model that reduces its overall power to approximate the target function and thus decreases accuracy. There are alternative methods that do have this flexibility, such as recurrent mechanisms like long short-term memory (LSTM) networks. These methods have a very high complexity and computational cost, and because of this, they are a poor fit for this problem.

The new method presented here instead predicts a dynamic functional that can be evaluated over a continuous domain. Therefore, there is no need for the model to specify a fixed range or grid over the spatial domain on which predictions are made, and impeding operations like interpolation are not necessary.

The standard MLP for discrete predictions often consists of a set of nonlinear layers for feature extraction and a final linear transformation to evaluate the waveform from nonlinear features. If $X \in \mathbb{R}^{B \times F_{in}}$ denotes the inputs, and $Z \in \mathbb{R}^{B \times F_{hidden}}$ are the intermediate nonlinear features, then the standard fixed output MLP defines the waveform prediction $Y = ZW$, where $W \in \mathbb{R}^{F_{hidden} \times T}$ and T is the fixed number of prediction points. In other words, every evaluated point in the waveform is a linear combination of Z.

To allow for a continuous waveform prediction $w(t)$, where *t* is an abstract distance value along the waveform named "displacements" in Figure 32, T must be of a dynamic size and W can no longer be a simple matrix. We recognize that since every column of W corresponds to a specific point in the continuous domain of $w(t)$, every column of W must be a function of t. With this observation, we reformulate the traditional MLP waveform predictor to $Y = ZW(t_{eval})$, where $W(t_{eval}) \in \mathbb{R}^{F_{hidden} \times len(t_{eval})}$ and t_{eval} is a vector containing desired points at which the function is evaluated. Stated differently, $W(t_{eval})$ is a dynamic size matrix that transforms the state space Z into some points in the continuous range of function $w(t)$.

5.4 UNCERTAINTY PREDICTION MODULE

The uncertainty prediction module outputs a supplementary result to provide an accuracy estimate of a given stress distribution prediction. This quantification of uncertainty provides useful information for model improvement by exposing weak spots in the model's mechanisms and problematic patterns in the data that can be addressed. Calculating uncertainty also communicates confidence to the user so that the functional prediction error can be better understood, and if higher accuracy is desired for some samples, this metric can be used to instruct the simulation to fall back to the existing finite element methodology. The architecture of this module, shown in Figure 33, is simple and consists only of a single MLP for each stress distribution dimension.

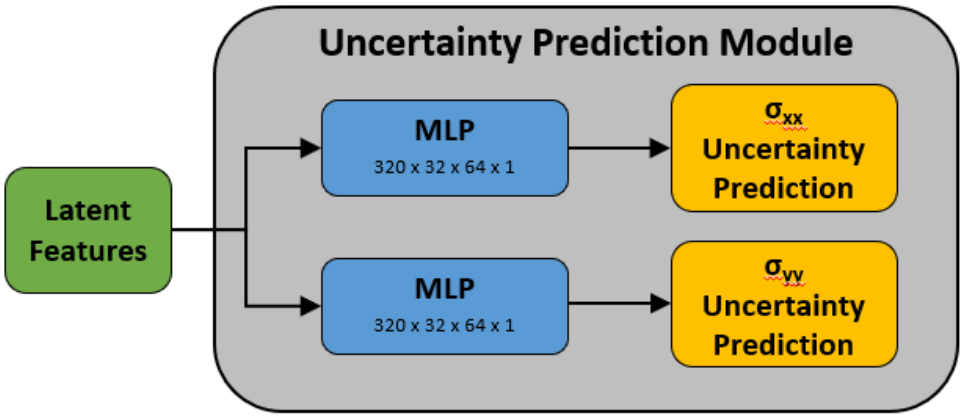


Figure 33. Architecture of Uncertainty Prediction Module

An example result is illustrated in Figure 34 in the form of an uncertainty envelope superimposed on the model prediction and ground truth stress distributions.

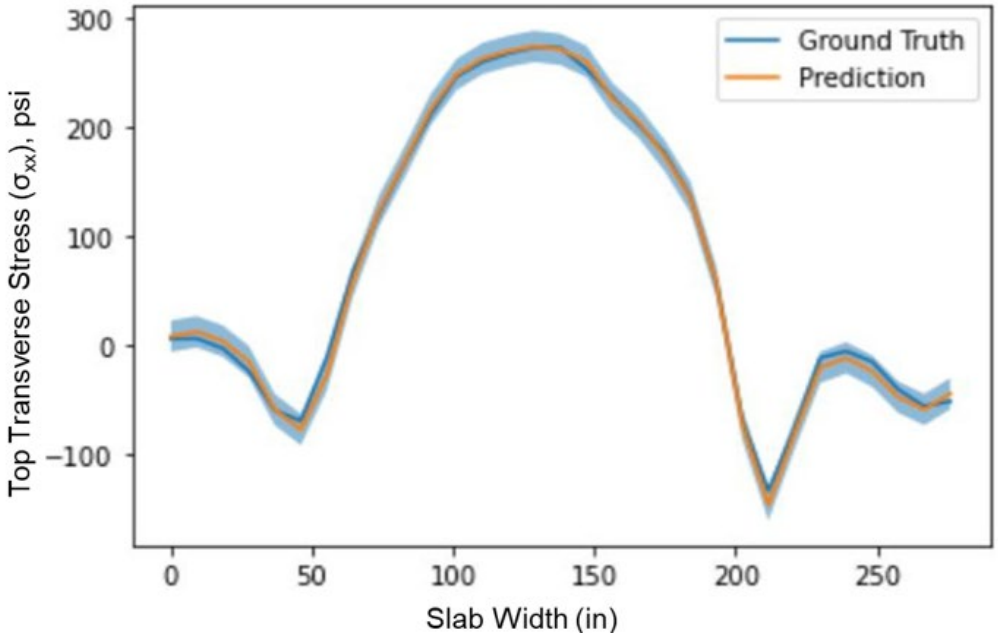


Figure 34. Example of Estimated Uncertainty Envelope Around Stress Prediction

An additional tool called Uniform Manifold Approximation and Projection for dimensionality reduction (UMAP) was also used as a supplement to the uncertainty quantification analysis. This method examines the topology of the training data and creates a mapping that maximizes space between physically different cases (McInnes et al., 2018). UMAP is a general-purpose manifold learning technique for dimensionality reduction that is built upon a framework of Riemannian geometry and algebraic topology, and its performance is superior to that of t-distributed stochastic neighbor embedding (t-SNE) and other popular methods in regard to computational performance and the preservation of the global structure. It works by combining the fuzzy simplicial set representations of local manifold approximations to create a representative topology of the data.

A sample result of this method applied to the latent representation of the training data is shown in Figure 35.

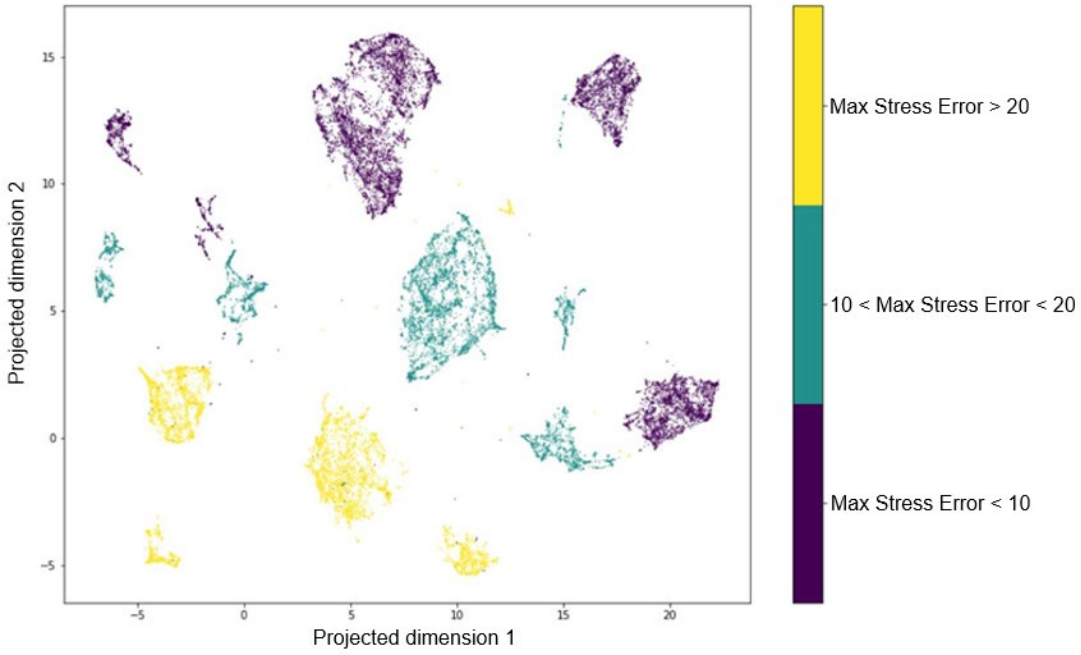


Figure 35. Applying UMAP Approach to Training Data

UMAP analysis shows the data forms clusters that clearly delineate problematic data in the latent representation. Researchers intended to use the information to improve the model for the specific problem cases or to identify cases where uncertainty is high, and a failover method (such as the finite element method) should be used to calculate the design stresses instead of the ML model. Mapping the projected dimensions to physical aircraft parameters was unsuccessful and the technique was abandoned when significant improvements to model accuracy made this approach unnecessary.

5.5 MODEL TRAINING

The model was trained on a subset of the developed database, where 20% of the samples were reserved for testing. The training operation was performed using backpropagation and the ADAHESSIAN numerical optimization algorithm. ADAHESSIAN is a fast second-order stochastic optimization algorithm designed for use in machine learning. It was selected for use here by experimentation with this and other stochastic gradient descent algorithms commonly employed to train deep neural networks (Yao et al., 2020).

A multitask learning formulation of the cost function, i.e., the function that the model seeks to minimize during training, was used to better accommodate the needs of the pavement design problem. It balances two objectives: 1) minimize the error near the peak stress area because those values are more critical in calculating the cumulative damage to the pavement, and 2) minimize the mean squared error of the entire stress distribution waveform to encourage the model to learn a more generalized representation of the underlying physics. A simple weighted linear sum of the

two terms is used to include both objectives, as defined in Equation 2, where w_i denotes the weighting value of the corresponding objective term.

$$J = w_1[\max(\sigma_{xx}) - \max(\hat{\sigma}_{xx})]^2 + w_2\left[\max\left(\sigma_{yy}\right) - \max\left(\hat{\sigma}_{yy}\right)\right]^2 + w_3 \sum_X [\sigma_{xx} - \hat{\sigma}_{xx}]^2 + w_4 \sum_Y \left[\sigma_{yy} - \hat{\sigma}_{yy}\right]^2 \quad (2)$$

5.6 OTHER ATTEMPTED TECHNIQUES

Over the development of this model, several methods were tried but ultimately discarded. They are briefly mentioned here to document the investigative process and explain some of the tools that have the potential to be used in similar physics-based problems.

Before discovering the final approach described above, kernel functions were the main technique examined. By forcing the intermediate signals to be represented by these functions, the model gets assistance by exploiting the expert knowledge that the functions of the physical process should have the same form. Therefore, the space of functions that the model must search during training is greatly reduced, freeing up its available resources to improve accuracy and better generalize across all aircraft cases. Although this method was supplanted by the current one described above, it was still successful and can theoretically be combined to achieve even greater accuracy. Several kernel functions were evaluated, most notably sinusoidal and wavelet functions. For this model, the best function was empirically determined to be the Daubechies 6 wavelet kernel (Kessler et al., 2003).

A related approach called inverse dynamic time warping was also tried. Inverse dynamic time warping is a point-matching algorithm that allows the kernel functions to stretch and contract to better match the true waveform. However, this approach struggles with matching the local minimums and maximums of the stress function.

Lastly, an alternate model was designed to predict the CDF function directly instead of the stress distribution. The accuracy of this was low because the presence of an exponential term in the function increased the sensitivity severely.

6. RESULTS

6.1 STRESS DISTRIBUTION PREDICTIONS

The accuracy of the model presented here is seen to be significantly improved compared to prior work in developing a machine learning model for this problem, as seen in Table 8 (Kaya et al., 2017; Rezaei-Tarahomi et al., 2020), with metrics root mean square error (RMSE), R^2, average absolute error (AAE), and relative root mean square error (RRMSE) defined in Equations 3–6. Note that the σ_{xx} errors are higher because their training data include full-body aircraft cases, which are more difficult to estimate. As explained in Section 4.1.1.3, predicting σ_{yy} is not necessary for the full-body cases.

Table 8. Accuracy Metrics of Presented Model Compared to Previous State-of-the-Art ML Model

Performance Metric	**New Model (total)**	**New Model (σ_{xx} only)**	**New Model (σ_{yy} only)**	**Wavelet Kernel**	**Previous State-of-the-Art [worst model, best model]**
RMSE	3.95 psi	4.91 psi	2.74 psi	8.89 psi	[34.6 psi, 23.4 psi]
R^2	0.997	0.996	0.998	0.985	[0.905, 0.961]
AAE	2.31 psi	2.94 psi	1.73 psi	5.95 psi	[25.2 psi, 17.4 psi]
RRMSE(%)	2.92%	3.51%	2.23%	6.9%	--

$$RMSE = \sqrt{\frac{1}{n} \times \sum_{i=1}^{n} \left(y_i^{prediction} - y_i^{true}\right)^2} \tag{3}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n} \left(y_i^{true} - y_i^{prediction}\right)^2}{\sum_{i=1}^{n} (y_i^{true} - \bar{y}_i^{true})^2} \tag{4}$$

$$AAE = \frac{1}{n} \times \sum_{i=1}^{n} \left|y_i^{prediction} - y_i^{true}\right| \tag{5}$$

$$RRMSE(\%) = 100 \times \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left(\frac{y_i^{true} - y_i^{predicted}}{y_i^{true}}\right)^2} \tag{6}$$

To further illustrate the accuracy of the predictions, two histograms of the errors are presented in Figure 36 (in which darker pixels indicate more samples) and Figure 37.

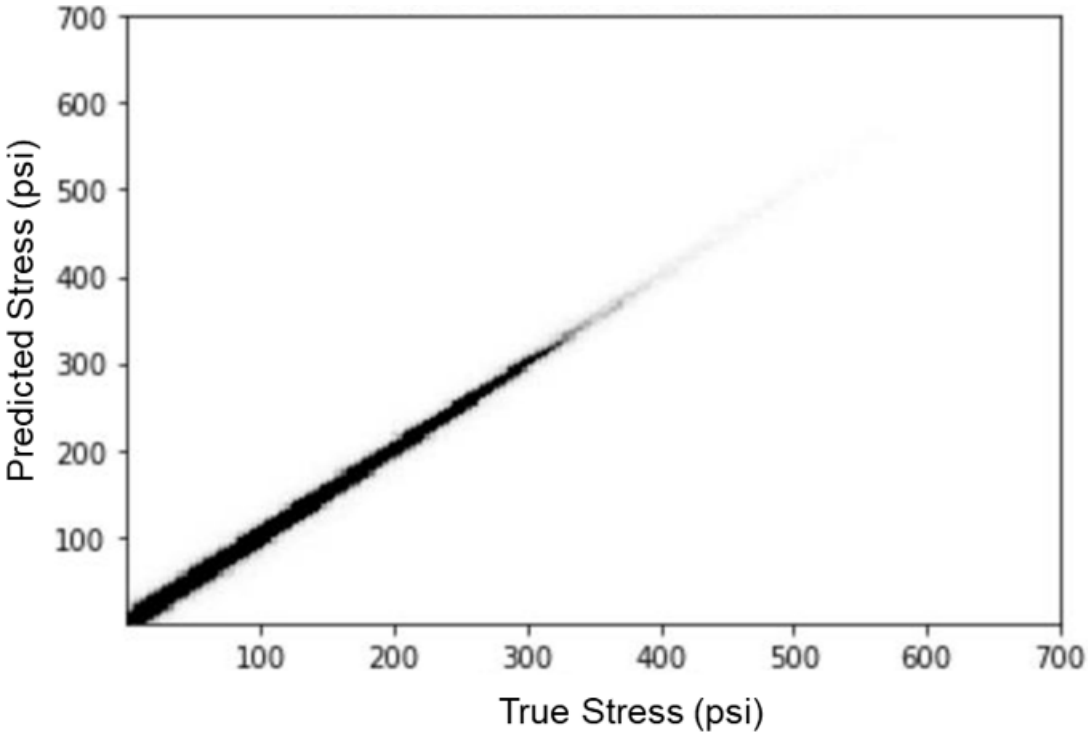


Figure 36. 2D Histogram of Model Stress Prediction Error Distribution

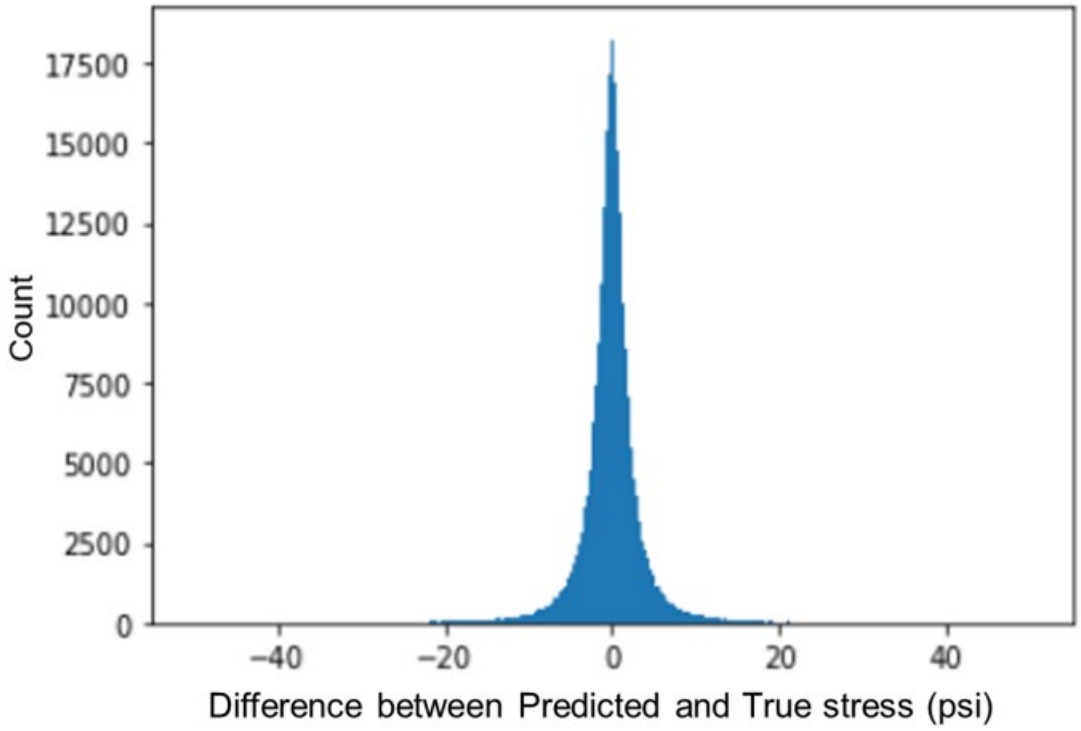


Figure 37. Histogram of Model Stress Prediction Error Distribution for All Test Data

Figures 36 and 37 show the errors present from all individual stress predictions over each predicted stress distribution for the testing data set. Although there are some outliers and poor predictions, over 90% of errors are less than 5 psi. The error analysis also indicates that 92% of errors are greater than -3.95 psi, i.e., the model underpredicts the stress by more than 3.95 psi in 8% of the cases. In addition, an example prediction of an individual stress distribution is shown in Figure 38.

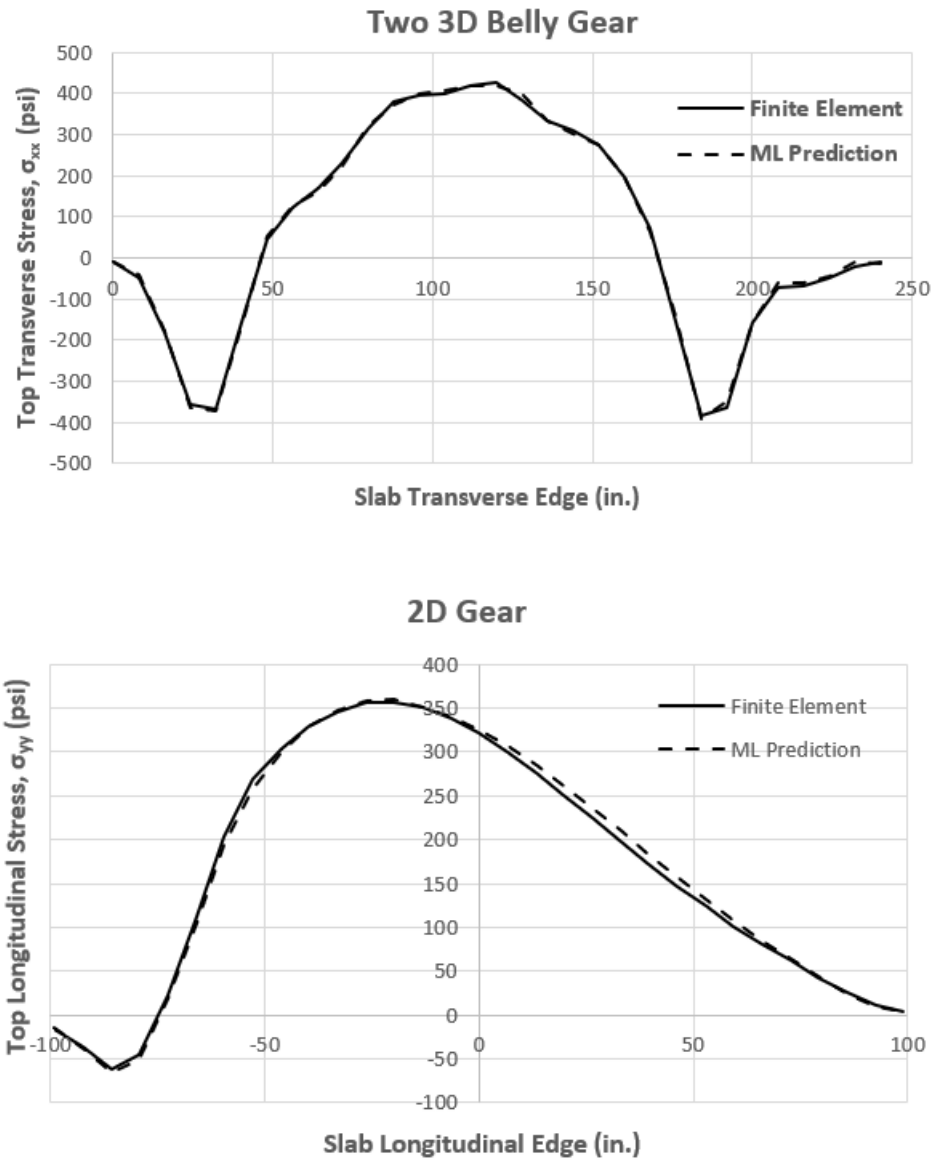


Figure 38. Example Stress Distribution Prediction

6.2 UNCERTAINTY PREDICTIONS

The predictions produced by the uncertainty prediction module were much less accurate than the main stress distribution predictions. Uncertainty quantification analyses of this type normally provide a weak estimate, but still provide value in identifying poorly performing features of the

model. They can also caution the user when they identify a problematic case. An overview of the accuracy is shown in Table 9, where the "error" considered here is the difference between the error predicted by the uncertainty prediction module and the true error of the prediction produced by the stress distribution prediction module.

Table 9. Summary of Accuracy Metrics for Uncertainty Prediction Module

RMSE	R^2	AAE
5.81 psi	0.723	3.11 psi

Further representations of the error distribution are given as a histogram and scatter plot in Figures 39 and 40, respectively. In these results, the percentage of error estimates that were wrong by a magnitude greater than 10 psi was only 0.74%.

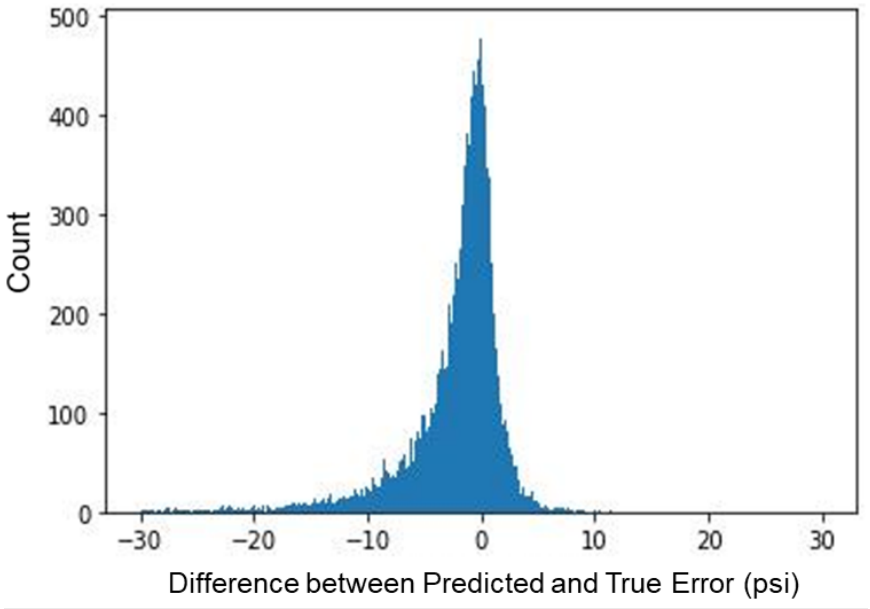


Figure 39. Histogram of Error from Uncertainty Prediction Module

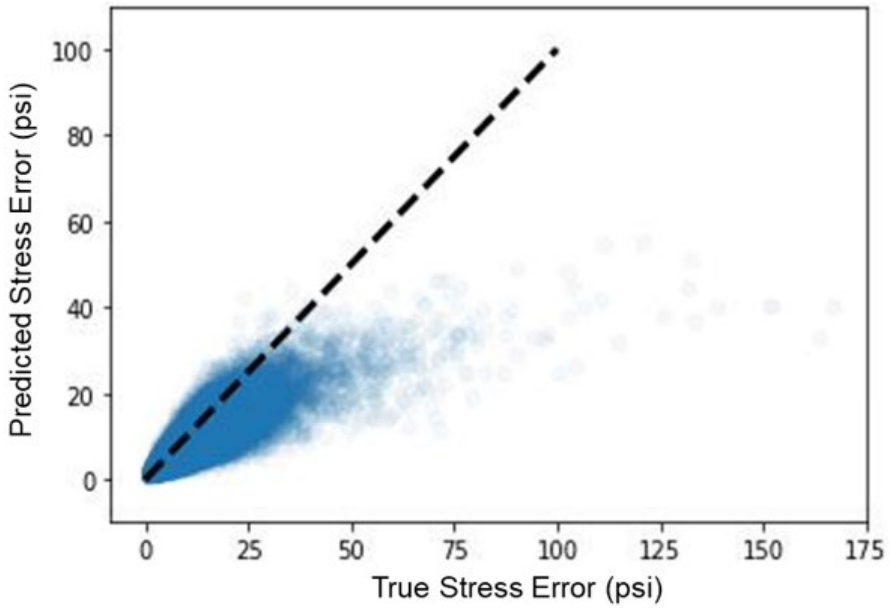


Figure 40. Scatter Plot of Uncertainty Prediction Module Results Compared to True Errors from the Stress Predictions

6.3 SENSITIVITY OF DESIGN THICKNESS TO ERROR IN STRESS PREDICTION

Researchers calculated the effect of error in the ML model results on pavement thickness. A CDF-based design procedure, as outlined in AC 150/5320-6G for bottom-up cracking design, was assumed for purposes of the sensitivity analysis. While the CDF itself is sensitive to small errors in stress predictions from the ML model, analysis determined that design PCC thickness is much less sensitive to error in the stress prediction. To examine the effect of prediction error on thickness, researchers selected a few cases from the database that had relatively high error. The number of passes of ML-predicted stress to achieve CDF = 1 was determined and then was used to calculate the CDF based on FE-calculated stress. For the cases with CDF greater than 1, the exercise used the gear loading from these cases to perform a series of FEAFAA calculations, varying the pavement thickness in FEAFAA until the FE-calculated stress matched the ML-predicted stress. Table 10 shows the results of stress analysis along with the estimation error at the peak stress and RMSE of the stress distribution prediction.

Table 10. Comparison of Predicted Stress from ML Model and FE Analysis

Case ID	PCC Thickness (in.)	Peak Stress (psi)		Error at peak (psi)	RMSE (psi)
		ML	FE		
Case2D_LT_3300	20.0	316.0	323.4	7.4	8.6
Case3D_L_88347	15.5	323.1	329.5	6.3	3.2
Case2D_LT_76666	16.5	348.9	354.8	5.9	5.9
Case3D_T_59560	19.0	365.2	374.7	9.4	6.0

Table 11 shows that the stresses calculated after increasing the PCC thickness by 0.5 inch are comparable to the ML-predicted stress for the thinner slab, thus resulting in a similar CDF.

Table 11. Calculated Stress from FE Analysis After Increasing PCC Thickness

Case ID	Increased PCC Thickness by 0.5-inch	FE-Calculated Peak Stress (psi)
Case2D_LT_3300	20.5	315.4
Case3D_L_88347	16.0	322.1
Case2D_LT_76666	17.0	347.7
Case3D_T_59560	19.5	365.1

The analysis above indicates that even for some of the cases that resulted in the highest errors in predicted stress, the design thickness based on the ML model is within 0.5 inch of the design thickness based on the FE model. This is consistent with the recommendation in AC 150/5320-6G to round the FAARFIELD design thickness to the nearest 0.5 inch. This is also in line with common practice for contractors to increase thickness up to 0.5 inch to ensure they receive full payment for thickness. Given that 90% of stress estimates with the ML model will be within 5 psi of FE-calculated stress, the model will underpredict or overpredict required thickness by 0.5 inch less than 10% of the time.

To further illustrate the effect of prediction error on designed thickness, the stress distribution along transverse edge resulting from FE analysis and predicted by ML for a 19-inch-thick slab under a 3D gear were calculated, as shown in Figure 41.

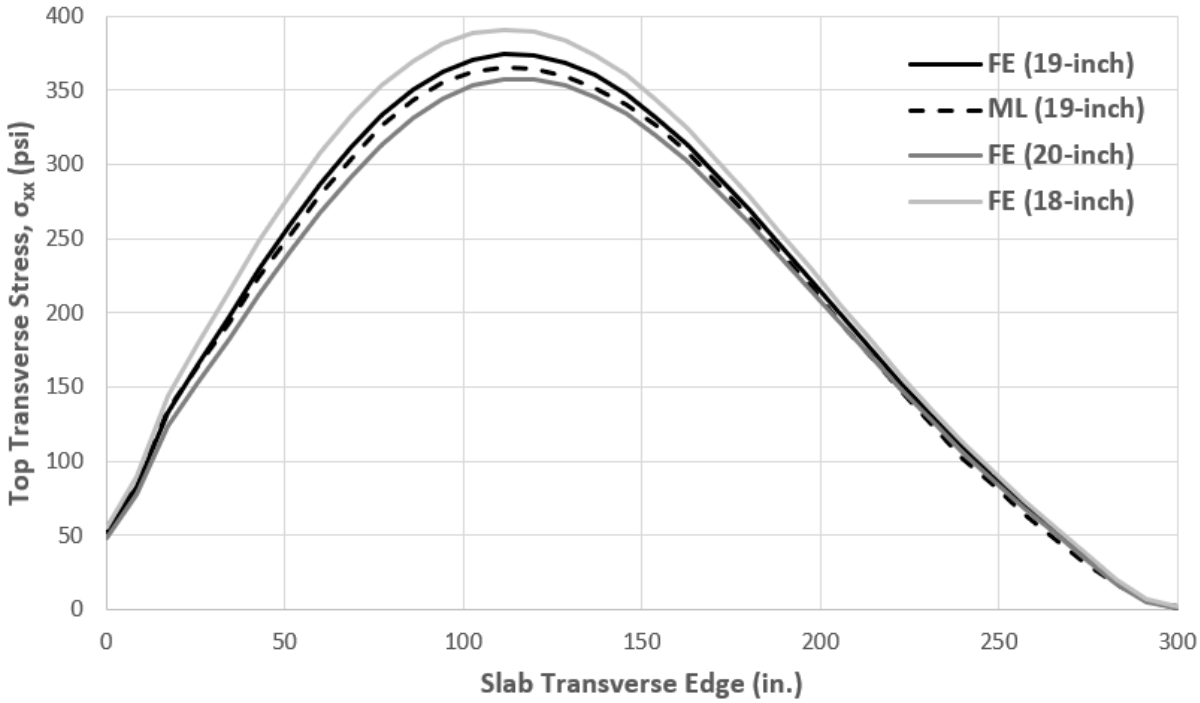


Figure 41. Example Comparing Predicted Stress and FE Stress Due to 5% Change in Slab Thickness

The FE-calculated stress distributions for the same pavement system but with 5% thicker (20-inch) and 5% thinner (18-inch) slab are also shown in Figure 42. The ML-predicted stress for the 19-inch slab is within the limit of FE stresses distribution from the 18-inch and 20-inch slab. This suggests that the difference in stress due to 5% change in slab thickness is likely greater than the error in stress prediction.

7. IMPLEMENTATION OF ML MODEL AS A .NET LIBRARY

The ML model was implemented as a standalone library that can be referenced by .NET applications such as FAARFIELD. The library has four classes:

- PavementData
- AircraftData
- TopDownStressModel
- InferenceResult

PavementData is a class to organize the pavement input data for the ML model. *AircraftData* is a class to organize the aircraft load input data for the ML model. *TopDownStressModel* takes aircraft list (list of AircraftData objects) and pavement structure as arguments and predicts the transverse and longitudinal stresses at the top of a PCC pavement along the edges of a pavement slab using the ML model. *InferenceResult* provides an array of stresses along the transverse and longitudinal joints, the peak stresses in each array, and an estimate of the error in the prediction for each *AircraftData* object.

To make the library available in a .NET language, researchers exported the model from Python using PyTorch's Just-In-Time (JIT) compilation functionality. Pytorch implements its JIT functionality as a computational engine that is decoupled from the Python Interpreter, allowing use of the model without Python. The model's code was adjusted to execute within the subset of instructions allowed in PyTorch's JIT implementation. The code was exported as a computational graph defining the model.

PyTorch uses a C++ application programming interface (API) to execute the model without a Python engine. To allow the execution of the model from a .NET language, a C++/Common Language Infrastructure (CLI) wrapper was created as a bridge between .NET and the C++ API. The C++/CLI bridge implements classes that mirror classes in the C++ library in a way that is interpretable and accessible within a .NET solution. Programmers provide model inputs in a .NET language and the C++/CLI library translates them to calls understood by the PyTorch computational graph. The C++/CLI bridge translates the results of the model back to .NET data types and returns them to the calling program. Objects, methods, and properties of the ML class in the compiled DLL are transparently available to .NET programmers in a .NET environment. Appendix D provides a schema of the library classes and instructions for linking the DLL into a .NET application.

8. SUMMARY

The goal of this research effort was to develop an ML model to be a drop-in replacement for the finite element simulations currently used to calculate top-down cracking stresses.

Researchers reviewed existing design procedures and 3D-FE methods of calculating top-down cracking stresses and determined that they were not suitable to support design for the top-down failure mode. A conceptual design method based on CDF but suitable for top-down cracking design was proposed. The input required by the conceptual design method implies that the ML model needs to provide the distribution of transverse stress σ_{xx} along the transverse joint and the distribution of longitudinal stress σ_{yy} along the longitudinal joint. Researchers also identified that users need to input joint spacing as part of any design procedure that considers top-down cracking. Review of 3D-FE methods identified 19 parameters related to pavement structure, aircraft landing gear, and thermal conditions that are required to calculate the stresses that cause top-down cracking in rigid airfield pavements. These were selected as inputs to the ML model. Transverse gear position must be included as an input to the ML model to account for aircraft wander. Longitudinal gear position need not be considered due to the shape and magnitude of the stress distribution as an aircraft rolls through a slab.

A Monte Carlo simulation was used to define 127,000 combinations of rigid pavement, thermal, and aircraft gear parameters. An Octave script used the output of the Monte Carlo simulation to generate a FEAFAA job file for each combination of parameters. Custom written software loaded each job file into FEAFAA, executed the finite element analysis, and saved the finite element results. Another Octave script post-processed the FEAFAA results into a database called the *Output Matrix* that contains the nodal coordinates of the longitudinal and transverse edge and the associated stress distributions. The results in the *Output Matrix* have the same Case ID as the

associated input in the Input Matrix. Engineers reviewed the *Output Matrix* and determined that it is suitable for training the ML model. These data were compiled into a database.

An ML model was developed from the database to estimate the stress distribution along slab edges for any gear configuration. Several machine learning techniques were tried but ultimately unsuccessful in developing a model with sufficient accuracy. It was determined that the models based on the most promising technique, discrete wavelet kernels, had insufficient accuracy because they were constrained by specifying a fixed number of outputs. Researchers developed a new ML method that predicts a dynamic functional evaluated over a continuous domain. The model is based on a modular deep learning method. The training operation was performed using backpropagation and the ADAHESSIAN numerical optimization algorithm. The models constructed with the new method are significantly more accurate than previous ML techniques for similar problems. The resulting model is a single model with one interface for all gear types in the data set. The new model predicts the entire stress distribution, instead of just the peak stresses. Sensitivity analysis indicated that although CDF can be sensitive to the prediction error, error in the model will likely result in less than 0.5-inch of error in the final slab thickness.

The ML model was compiled into a .NET-compatible library suitable for use in a program like FAARFIELD.

9. REFERENCES

Brill, D. R., Hayhoe, G. F., & Ricalde, L. (2005). Analysis of CC2 rigid pavement test data from the FAA's National Airport Pavement Test Facility, In *The 7th International Conference on the Bearing Capacity of Roads, Railways, and Airfields (CD-ROM)*, Trondheim, Norway.

Brill, D. R. (2010). *Calibration of FAARFIELD rigid pavement design procedure* (DOT/FAA/AR-09/57). https://www.tc.faa.gov/its/worldpac/techrpt/ar0957.pdf

Brill, D. R. (1998). *Development of advanced computational models for airport pavement design* (DOT/FAA/AR-97/47). https://www.tc.faa.gov/its/worldpac/techrpt/ar97-47.pdf

Federal Aviation Administration. (2021). *Airport pavement design and evaluation* (Advisory Circular [AC] 150/5320-6G). https://www.faa.gov/documentLibrary/media/Advisory_Circular/150-5320-6G-Pavement-Design.pdf

Guo E., Ricalde, L., & Kawa, I. (2007). *FAA finite element design procedure for rigid pavements* (DOT/FAA/AR-07/33). https://www.tc.faa.gov/its/worldpac/techrpt/ar0733.pdf

Kawa, I. (2012). *Pass-to-coverage computation for arbitrary gear configurations in the FAARFIELD program* (DOT/FAA/TC-TN12/47). https://www.tc.faa.gov/its/worldpac/techrpt/tctn12-47.pdf

Kaya, O., Rezaei-Tarahomi, A., Ceylan, H., Gopalakrishnan, K., Kim, S., & Brill, D. R. (2017). *Developing rigid airport pavement multiple-slab response models for top-down cracking mode using artificial neural networks*. In Transportation Research Board 96th Annual Meeting, No. 17-05375. https://dr.lib.iastate.edu/handle/20.500.12876/13659

Kessler, B. M., Payne, G. L., & Polyzou, W. N. (2003). *Wavelet notes*. University of Iowa. https://arxiv.org/pdf/nucl-th/0305025.pdf

McInnes, L., Healy, J., & Melville, J. (2018). *UMAP: uniform manifold approximation and projection for dimension reduction*. https://doi.org/10.48550/arXiv.1802.03426

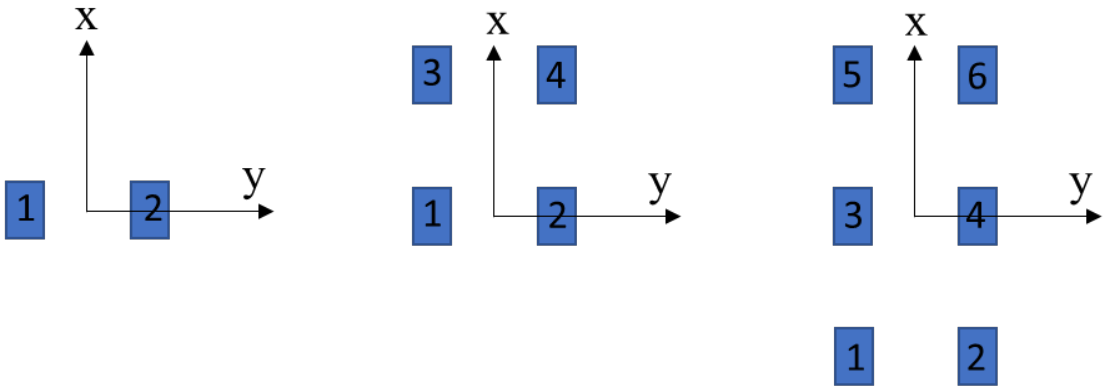
National Cooperative Highway Research Program. (2003). *Guide for mechanistic-empirical design of new and rehabilitated pavement structures, Appendix KK: transverse cracking of JPCP.* https://onlinepubs.trb.org/onlinepubs/archive/mepdg/2appendices_kk.pdf

Rezaei-Tarahomi, A., Kaya, O., Ceylan, H., Gopalakrishnan, K., Kim, S., & Brill, D. R. (2010). ANNFAA: artificial neural network-based tool for the analysis of Federal Aviation Administration's rigid pavement systems. *International Journal of Pavement Engineering, 23*(2), 400-413. https://doi.org/10.1080/10298436.2020.1748627

Tuleubekov, K. (2016). *Replacement of FAARFIELD tandem factors with cumulative damage factor methodology* (DOT/FAA/TC-16/46). https://rosap.ntl.bts.gov/view/dot/57693

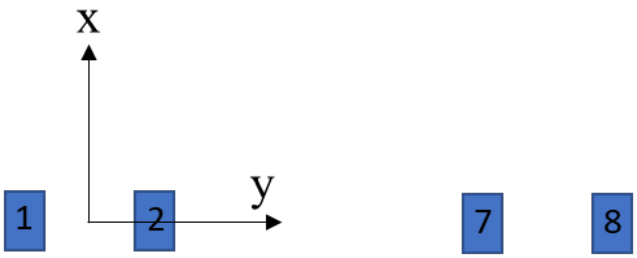
Yao, Z., Gholami, A., Shen, S., Mustafa, M., Keutzer, K., & Mahoney, M. (2020). ADAHESSIAN: An adaptive second order optimization for machine learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12), 10665-10673. https://doi.org/10.1609/aaai.v35i12.17275

APPENDIX A—CALCULATION OF TIRE COORDINATES

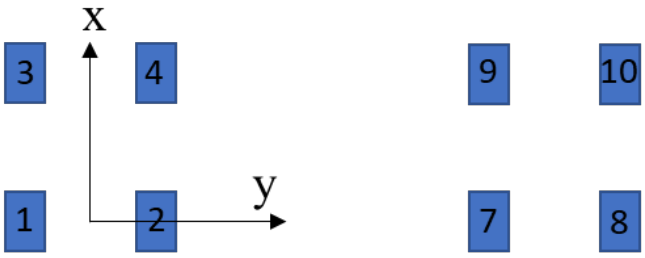


(a) D Gear (b) 2D Gear (c) 3D Gear

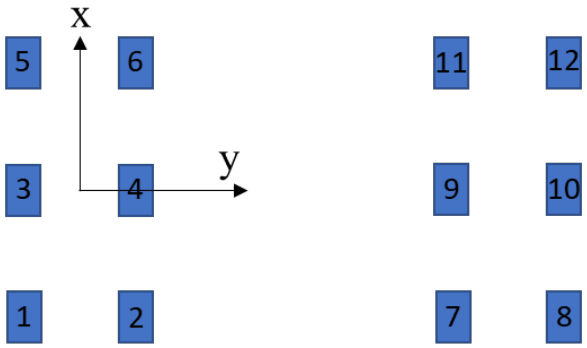
Figure A-1. Local Axis of Individual Gear Types



(a) D Full Landing Gear



(b) 2D Full Landing Gear



(c) 3D Full Landing Gear

Figure A-2. Local Axis of Full Landing Gear Types

Table A-1. Calculation of Tire Coordinates for Each Individual Gear Type

Gear Type	Tire x-Coordinate (<X> tags)						Tire y-Coordinate (<Y> tags)					
	Tire 1	Tire 2	Tire 3	Tire 4	Tire 5	Tire 6	Tire 1	Tire 2	Tire 3	Tire 4	Tire 5	Tire 6
D	0	0	-	-	-	-	$\frac{-DS}{2}$	$\frac{DS}{2}$	-	-	-	-
2D	0	0	TS	TS	-	-	$\frac{-DS}{2}$	$\frac{DS}{2}$	$\frac{-DS}{2}$	$\frac{DS}{2}$	-	-
3D	-TS	-TS	0	0	TS	TS	$\frac{-DS}{2}$	$\frac{DS}{2}$	$\frac{-DS}{2}$	$\frac{DS}{2}$	$\frac{-DS}{2}$	$\frac{DS}{2}$

Note: DS and TS indicate Dual Spacing and Tandem Spacing. All units are in inch.

Table A-2. Calculation of Tire x-Coordinates for Each Full Landing Gear Type

Gear Type	Tire x-Coordinate (<X> tags)											
	Tire 1	Tire 2	Tire 3	Tire 4	Tire 5	Tire 6	Tire 7	Tire 8	Tire 9	Tire 10	Tire 11	Tire 12
D	0	0	-	-	-	-	0	0	-	-	-	-
2D	0	0	TS	TS	-	-	0	0	TS	TS	-	-
3D	-TS	-TS	0	0	TS	TS	-TS	-TS	0	0	TS	TS

Note: TS indicates Tandem Spacing. All units are in inch.

Table A-3. Calculation of Tire y-Coordinates for Each Full Landing Gear Type

Gear Type	Tire y-Coordinate (<Y> tags)											
	Tire 1	Tire 2	Tire 3	Tire 4	Tire 5	Tire 6	Tire 7	Tire 8	Tire 9	Tire 10	Tire 11	Tire 12
D	$\frac{-DS}{2}$	$\frac{DS}{2}$	-	-	-	-	$\frac{-\mathrm{DS}}{2}+\mathrm{Tr}$	$\frac{\mathrm{DS}}{2}+\mathrm{Tr}$	-	-	-	-
2D	$\frac{-DS}{2}$	$\frac{\mathrm{DS}}{2}$	$\frac{-\mathrm{DS}}{2}$	$\frac{\mathrm{DS}}{2}$	0	0	$\frac{-\mathrm{DS}}{2}+\mathrm{Tr}$	$\frac{\mathrm{DS}}{2}+\mathrm{Tr}$	$\frac{-\mathrm{DS}}{2}+\mathrm{Tr}$	$\frac{\mathrm{DS}}{2}+\mathrm{Tr}$	0	0
3D	$\frac{-DS}{2}$	$\frac{\mathrm{DS}}{2}$	$\frac{-\mathrm{DS}}{2}$	$\frac{\mathrm{DS}}{2}$	$\frac{-\mathrm{DS}}{2}$	$\frac{\mathrm{DS}}{2}$	$\frac{-\mathrm{DS}}{2}+\mathrm{Tr}$	$\frac{\mathrm{DS}}{2}+\mathrm{Tr}$	$\frac{-\mathrm{DS}}{2}+\mathrm{Tr}$	$\frac{\mathrm{DS}}{2}+\mathrm{Tr}$	$\frac{-\mathrm{DS}}{2}+\mathrm{Tr}$	$\frac{\mathrm{DS}}{2}+\mathrm{Tr}$

DS = Dual spacing
Tr = Track spacing
Note all units are in inches.

APPENDIX B—CALCULATION OF GEAR Y-OFFSET

For D Gear:

$$y - offset = \frac{-Slab\ Length \times 12}{2} + \frac{Tire\ Length}{2}$$

For 2D Gear:

$$y - offset = \frac{-Slab\ Length \times 12}{2} - \frac{Tandem\ Spacing}{2} + \frac{Tire\ Length}{2}$$

For 3D Gear (longitudinal edge):

$$y - offset = \frac{-Slab\ Length \times 12}{2} - \frac{Tandem\ Spacing}{2} + \frac{Tire\ Length}{2}$$

For 3D Gear (transverse edge):

$$y - offset = \frac{-Slab\ Length \times 12}{2} + \frac{Tire\ Length}{2}$$

where the tire length is calculated as:

$$Tire\ Length = 0.8712 \times \sqrt{\frac{Gear\ Weight}{n \times 0.5227 \times Tire\ Pressure}}$$

where *n* is the number of wheels on the gear.

APPENDIX C—SAMPLE FEAFAA INPUT XML SCHEMA

Following is the XML schema generated for "Case2D_LT_7072.FEA":

```
<?xml version="1.0" encoding="utf-8"?>
<FEAFAAJobInfo>
 <AirplaneSelectionTab>
 <AirplaneGroup>6.00</AirplaneGroup>
 <AirplaneIndex>0.00</AirplaneIndex>
 <GrossWeight>157164</GrossWeight>
 <PcntOnMainGears>100</PcntOnMainGears>
 <NMainGears>1</NMainGears>
 <NWheels>4</NWheels>
 <TirePressure>178</TirePressure>
 <WheelCoordinates>
 <X>0</X>
 <Y>-21.9</Y>
 <X>0</X>
 <Y>21.9</Y>
 <X>61.6</X>
 <Y>-21.9</Y>
 <X>61.6</X>
 <Y>21.9</Y>
 </WheelCoordinates>
 <PCARectangle>True</PCARectangle>
 </AirplaneSelectionTab>
 <PavementStructureTab>
 <PCCOverlay>False</PCCOverlay>
 <NumberOfLayers>4</NumberOfLayers>
 <LayerData>
 <LayerTypes>PCC Slab</LayerTypes>
 <EModulus>4990012</EModulus>
 <PoissonsRatio>0.15</PoissonsRatio>
 <Thickness>17</Thickness>
 <LayerTypes>Subbase 1</LayerTypes>
 <EModulus>507594</EModulus>
 <PoissonsRatio>0.20</PoissonsRatio>
 <Thickness>8</Thickness>
 <LayerTypes>Subbase 2</LayerTypes>
 <EModulus>79339</EModulus>
 <PoissonsRatio>0.35</PoissonsRatio>
 <Thickness>25</Thickness>
 <LayerTypes>Subgrade</LayerTypes>
 <EModulus>24939</EModulus>
 <PoissonsRatio>0.40</PoissonsRatio>
 <Thickness>Infinite</Thickness>
 </LayerData>
 <SlabMeshGroupBox>
 <XDimension>16</XDimension>
 <YDimension>18</YDimension>
 <NumberOfElementsSlab>30</NumberOfElementsSlab>
 <NumberOfSlabs>9</NumberOfSlabs>
 </SlabMeshGroupBox>
 <FoundationMeshGroupBox>
 <NumberOfElementsFoundation>30</NumberOfElementsFoundation>
 </FoundationMeshGroupBox>
 <LoadingGroupBox>
 <LoadingType>Static Load</LoadingType>
 <Angle>0</Angle>
 <PositonGear>
 <GearXoffset>-21</GearXoffset>
 <GearYoffset>-129.8485</GearYoffset>
 </PositonGear>
 </LoadingGroupBox>
 <SlabTemperatureGroupBox>
 <SlabTemperatureCheckBox>True</SlabTemperatureCheckBox>
 <LETG>-1.82</LETG>
 <TermalCoefficient>5.6e-06</TermalCoefficient>
 <SlabCurlingShape>Spherical</SlabCurlingShape>
```

```
 </SlabTemperatureGroupBox>
 </PavementStructureTab>
 <JointModelingTab>
 <XDirectionDowelBarDataGroupBox>
 <XDirectionDowelBarDataCheckBox>True</XDirectionDowelBarDataCheckBox>
 <XDowelBarDiameter>1.5</XDowelBarDiameter>
 <XDowelBarSpacing>18</XDowelBarSpacing>
 <XJointOpening>0.375</XJointOpening>
 <MethodOfDowerBarPlacement>
 <XBarPlacedInFreshedConcrete>True</XBarPlacedInFreshedConcrete>
 </MethodOfDowerBarPlacement>
 <XEquivalentJointStiffness>236854</XEquivalentJointStiffness>
 </XDirectionDowelBarDataGroupBox>
 <XDirectionSpringConstraintGroupBox>
 <XDirectionSpringConstraintCheckBox>True</XDirectionSpringConstraintCheckBox>
 <XEquivalentBoundaryStiffness>1000</XEquivalentBoundaryStiffness>
 </XDirectionSpringConstraintGroupBox>
 <YDirectionDowelBarDataGroupBox>
 <YDirectionDowelBarDataCheckBox>True</YDirectionDowelBarDataCheckBox>
 <YDowelBarDiameter>1.5</YDowelBarDiameter>
 <YDowelBarSpacing>18</YDowelBarSpacing>
 <YJointOpening>0.375</YJointOpening>
 <MethodOfDowerBarPlacement>
 <YBarPlacedInFreshedConcrete>True</YBarPlacedInFreshedConcrete>
 </MethodOfDowerBarPlacement>
 <YEquivalentJointStiffness>236854</YEquivalentJointStiffness>
 </YDirectionDowelBarDataGroupBox>
 <YDirectionSpringConstraintGroupBox>
 <YDirectionSpringConstraintCheckBox>True</YDirectionSpringConstraintCheckBox>
 <YEquivalentBoundaryStiffness>1000</YEquivalentBoundaryStiffness>
 </YDirectionSpringConstraintGroupBox>
 </JointModelingTab>
</FEAFAAJobInfo>
```

Following is the XML schema generated for “Belly3D_T_8740.FEA”:

```
<?xml version="1.0" encoding="utf-8"?>
<FEAFAAJobInfo>
 <AirplaneSelectionTab>
 <AirplaneGroup>6.00</AirplaneGroup>
 <AirplaneIndex>0.00</AirplaneIndex>
 <GrossWeight>675017</GrossWeight>
 <PcntOnMainGears>100</PcntOnMainGears>
 <NMainGears>1</NMainGears>
 <NWheels>12</NWheels>
 <TirePressure>214</TirePressure>
 <WheelCoordinates>
 <X>-67</X>
 <Y>-29.5</Y>
 <X>-67</X>
 <Y>29.5</Y>
 <X>0</X>
 <Y>-29.5</Y>
 <X>0</X>
 <Y>29.5</Y>
 <X>67</X>
 <Y>-29.5</Y>
 <X>67</X>
 <Y>29.5</Y>
 <X>-67</X>
 <Y>161.5</Y>
 <X>-67</X>
 <Y>220.5</Y>
 <X>0</X>
 <Y>161.5</Y>
 <X>0</X>
 <Y>220.5</Y>
 <X>67</X>
 <Y>161.5</Y>
 <X>67</X>
 <Y>220.5</Y>
 </WheelCoordinates>
 <PCARectangle>True</PCARectangle>
 </AirplaneSelectionTab>
 <PavementStructureTab>
 <PCCOverlay>False</PCCOverlay>
 <NumberOfLayers>4</NumberOfLayers>
 <LayerData>
 <LayerTypes>PCC Slab</LayerTypes>
 <EModulus>5677051</EModulus>
 <PoissonsRatio>0.15</PoissonsRatio>
 <Thickness>15</Thickness>
 <LayerTypes>Subbase 1</LayerTypes>
 <EModulus>436720</EModulus>
 <PoissonsRatio>0.20</PoissonsRatio>
 <Thickness>6</Thickness>
 <LayerTypes>Subbase 2</LayerTypes>
 <EModulus>56622</EModulus>
 <PoissonsRatio>0.35</PoissonsRatio>
 <Thickness>11</Thickness>
 <LayerTypes>Subgrade</LayerTypes>
 <EModulus>18519</EModulus>
 <PoissonsRatio>0.40</PoissonsRatio>
 <Thickness>Infinite</Thickness>
 </LayerData>
 <SlabMeshGroupBox>
 <XDimension>13</XDimension>
 <YDimension>15</YDimension>
 <NumberOfElementsSlab>30</NumberOfElementsSlab>
 <NumberOfSlabs>9</NumberOfSlabs>
 </SlabMeshGroupBox>
 <FoundationMeshGroupBox>
 <NumberOfElementsFoundation>30</NumberOfElementsFoundation>
 </FoundationMeshGroupBox>
```

```
 <LoadingGroupBox>
 <LoadingType>Static Load</LoadingType>
 <Angle>0</Angle>
 <PositonGear>
 <GearXoffset>53</GearXoffset>
 <GearYoffset>-80.2316</GearYoffset>
 </PositonGear>
 </LoadingGroupBox>
 <SlabTemperatureGroupBox>
 <SlabTemperatureCheckBox>True</SlabTemperatureCheckBox>
 <LETG>-1.78</LETG>
 <TermalCoefficient>4.4e-06</TermalCoefficient>
 <SlabCurlingShape>Spherical</SlabCurlingShape>
 </SlabTemperatureGroupBox>
 </PavementStructureTab>
 <JointModelingTab>
 <XDirectionDowelBarDataGroupBox>
 <XDirectionDowelBarDataCheckBox>True</XDirectionDowelBarDataCheckBox>
 <XDowelBarDiameter>1.25</XDowelBarDiameter>
 <XDowelBarSpacing>15</XDowelBarSpacing>
 <XJointOpening>0.375</XJointOpening>
 <MethodOfDowerBarPlacement>
 <XBarPlacedInFreshedConcrete>True</XBarPlacedInFreshedConcrete>
 </MethodOfDowerBarPlacement>
 <XEquivalentJointStiffness>142256</XEquivalentJointStiffness>
 </XDirectionDowelBarDataGroupBox>
 <XDirectionSpringConstraintGroupBox>
 <XDirectionSpringConstraintCheckBox>True</XDirectionSpringConstraintCheckBox>
 <XEquivalentBoundaryStiffness>1000</XEquivalentBoundaryStiffness>
 </XDirectionSpringConstraintGroupBox>
 <YDirectionDowelBarDataGroupBox>
 <YDirectionDowelBarDataCheckBox>True</YDirectionDowelBarDataCheckBox>
 <YDowelBarDiameter>1.25</YDowelBarDiameter>
 <YDowelBarSpacing>15</YDowelBarSpacing>
 <YJointOpening>0.375</YJointOpening>
 <MethodOfDowerBarPlacement>
 <YBarPlacedInFreshedConcrete>True</YBarPlacedInFreshedConcrete>
 </MethodOfDowerBarPlacement>
 <YEquivalentJointStiffness>142256</YEquivalentJointStiffness>
 </YDirectionDowelBarDataGroupBox>
 <YDirectionSpringConstraintGroupBox>
 <YDirectionSpringConstraintCheckBox>True</YDirectionSpringConstraintCheckBox>
 <YEquivalentBoundaryStiffness>1000</YEquivalentBoundaryStiffness>
 </YDirectionSpringConstraintGroupBox>
 </JointModelingTab>
</FEAFAAJobInfo>
```

APPENDIX D—.NET LIBRARY CLASSES

D.1 PAVEMENTDATA

PavementData is a class to organize the pavement input data for the ML model. Each PavementData object describes a single pavement structure.

D.1.1 CONSTRUCTOR

PavementData()	Initializes a pavement data object

D.1.2 PROPERTIES

Property	Type	Range	Description
PCCModulus	Single	4E6 to 6E6	Modulus of PCC layer, psi
BaseModulus	Single	100,000 to 800,000	Modulus of base layer, psi
SubbaseModulus	Single	20,000 to 90,000	Modulus of subbase layer, psi
SubgradeModulus	Single	4,500 to 30,000	Modulus of subgrade layer, psi
PCCThickness	Single	8 to 24	Thickness of PCC layer, inches
BaseThickness	Single	5 to 10	Thickness of base layer, inches
SubbaseThickness	Single	6 to 30	Thickness of subbase layer, inches
SlabWidth	Single	12 to 25	Width of slab, feet
SlabLength	Single	12 to 25	Length of slab, feet
ELTG	Single	-2 to 0.5	Thermal gradient, degrees °F/inch
CTE	Single	4E-6 to 6E-6	Coefficient of thermal expansion for PCC, inch/degree °F
EJS	Single	110,000 to 315,000	Equivalent joint stiffness, psi
xLocations	Array of Singles	0 to 300 (Must be between zero and SlabWidth ×12)	Locations along transverse joint at which to calculate stress, inches. They are at (xLocations, -SlabLength×12/2). See Figure D-1.
yLocations	Array of Singles	-150 to 150 (Must be between ±SlabLength×12/2)	Locations along longitudinal joint at which to calculate stress, inches. They are at (0, yLocations). See Figure D-2.

Interval (inches)	xLocations (inches)
If SlabLength =< 22 ft $\text{Interval} = \frac{\text{SlabWidth} \times 12}{30}$	xLocations(0) = 0 xLocations(1) = xLocations(0)+ Interval . . xLocations(30) = SlabWidth × 12 xLocations(31) to xLocations(35) are zero.
If SlabLength > 22 ft $\text{Interval} = \frac{\text{SlabWidth} \times 12}{35}$	xLocations(0) = 0 xLocations(1) = xLocations(0)+ Interval . . xLocations(35) = SlabWidth × 12

Figure D-1. Calculation of xLocations

Interval (inches)	yLocations (inches)
If SlabLength =< 22 ft $\text{Interval} = \frac{\text{SlabLength} \times 12}{30}$	yLocations(0) = - SlabLength × 12/2 yLocations(1) = yLocations(0)+ Interval . . yLocations(30) = SlabLength × 12/2 yLocations(31) to yLocations(35) are zero.
If SlabLength > 22 ft $\text{Interval} = \frac{\text{SlabLength} \times 12}{35}$	yLocations(0) = - SlabLength × 12/2 yLocations(1) = yLocations(0)+ Interval . . yLocations(35) = SlabLength × 12/2

Figure D-2. Calculation of yLocations

D.2 AIRCRAFTDATA

AircraftData is a class to organize the aircraft load input data for the ML model. Each AircraftModel object describes a single aircraft loading.

D.2.1 Constructor

AircraftData()	Initializes an aircraft data object

D.2.2 PROPERTIES

Property	Type	Range	Description
GearType	Single	1.0=dual (D) 2.0=dual tandem (2D) 3.0=dual tridem (3D)	Description of gear type
GearWeight	Single	50,000 to 380,000 (varies by gear type)	Weight on all wheels of one gear strut, pounds
TirePressure	Single	150 to 250 (varies by gear type)	Tire pressure, psi
Dual	Single	28 to 64 (varies by gear type)	Center-to-center dual spacing, inches
Tandem	Single	41 to 78 (varies by gear type)	Center-to-center tandem spacing, inches
GearXOffset	Single	See Figure D3.	Distance from longitudinal joint to center of gear, inches
BellySpacing	Single	130 to 300	Track Spacing, distance between gear centers, inches

Parameter	Left Boundary	Right Boundary
GearXOffset for Individual Gear (in)	$-\frac{\text{SlabWidth} \times 12}{4}$	$\frac{\text{SlabWidth} \times 12}{2}$
GearXOffset for Full Landing Gear (in)	$\frac{\text{BellySpacing}}{2} - \frac{\text{SlabWidth} \times 12}{4}$	$(\text{Slab Width} \times 12) - \frac{\text{BellySpacing}}{2}$

Figure D-3. Gear x-offset Range

D.3 TOPDOWNSTRESSMODEL

Predict the transverse and longitudinal stresses at the top of a PCC pavement along the edges of a pavement slab using an ML model. An aircraft list (list of AircraftData objects) and pavement structure are the arguments. The InferStresses method returns a list of InferenceResult objects, one for each AircraftData object.

D.3.1 CONSTRUCTOR

TopDownStressModel ()	Initializes surrogate model object

D.3.2 PROPERTIES

None.

D.3.3 METHODS

Method	Argument	Returns	Description
InferStresses()	[List of AircraftData], PavementData	List of InferenceResult	Takes aircraft/pavement configuration and returns stresses as a list of InferenceResults, one list element for each item in the list of AircraftData

D.4 INFERENCERESULT

Results of the top-down cracking stress prediction. It provides the array of stresses along the transverse and longitudinal joints, the peak stresses in each array, and an estimate of the error in the prediction.

D.4.1 CONSTRUCTOR

InferenceResult()	Initializes an inference data object

D.4.2 PROPERTIES

Property	Type	Description
TransStresses	List of Singles	Predicted stresses along transverse joint, psi. Array length matches length of xLocations
LongStresses	List of Singles	Predicted stresses along longitudinal joint, psi. Array length matches length of yLocations
TransPeakStress	Single	Max of TransStresses, psi
LongPeakStress	Single	Max of LongStresses, psi
TransSigma	Single	Uncertainty in TransPeakStress, psi
LongSigma	Single	Uncertainty in LongPeakStress, psi

D.5 DEPLOY THE ML DLL IN AN APPLICATION

The developed ML DLL is provided in the form of an archive file (called *FAALibrary-Final.zip*) containing several subfolders. Deploying the model to a new application (e.g., FAARFIELD) in VB.NET is a simple procedure. To add the DLL to a project, execute the following steps:

1. Unzip *FAALibrary-Final.zip*.
2. Copy the *SurrogateLib*, *TorchWrapper*, and *x64* folders to the application source code directory (e.g., copy the folders to the FAARFIELD source code folder).
3. In the application, add the projects *SurrogateLib* and *TorchWrapper*.
4. Add the DLL as reference to the startup project (e.g., FF2 on FAARFIELD).
5. Ensure that the DLL projects and the startup project on the application are set to x64 CPU on the project properties.