MACHINE LEARNING APPLICATION FOR ACTIVE EXPLORATION OF WELD SEQUENCE SCENARIOS

Mahyar Asadi¹, Mohammad Mohseni, Majid Tanbakuei Kashani, Michael Fernandez, Mathew Smith

Applus (SKC Engineering), 19165 94th Avenue, Surrey, BC, Canada, V4N 3S4

Keywords: Weld Distortion Control, Weld Sequence Design, Machine Learning for Welding, Digital Weld Engineering, Weld Modeling and Simulation, Data-Driven Modeling

Abstract
Distortion is a common problem in welded structures, and the process of finding an effective weld sequence to mitigate the distortion is a challenging task given a large number of possible combinations. Despite efficient simulation tools and powerful supercomputers, yet simulation tools have been limited by CPU time to optimize a welding sequence and therefore not mature for practical designs. To this end, we constructed and integrated machine learning (ML) algorithms with the simulation capability. These ML models were then trained to increase the fidelity by a wisely chosen training-set of simulation to construct a meta-model for active exploration of various weld sequence scenarios in real time. As opposed to existing ML algorithms that require an extensive data set to train, our algorithm picks relatively small training-set to construct a meta-model. We present an example of our algorithm implemented in a real welded structure project.

Introduction
Today’s structures are more complex and demanding much tighter fabrication tolerances than our routine practice. Welding plays a challenging role in meeting those tolerances in particular when it comes to distortion. Welding sequence and intermittent welding design, which determines the best welding pattern in multi-pass welds, are familiar techniques to control the distortion when dealing with multi-pass welded structures. Finding the best solution for such a design is usually intuitive and based on the similarity of previously welded structures because this is not feasible through shop trials. An alternative is welding simulation tools that model several sequences to select the one with minimal distortion.

Excellent simulation software is now available to capture and couple thermal, microstructure and stress effects of welds based on 3D transient temperature and thermal stress-strain analysis [1]. Despite powerful supercomputers, yet welding simulation tools are limited by computational time, and therefore they are not mature for practical designs. For example, having “n” welds requires choosing from $2^n$ possible scenarios or combinations of the welds where n! counts for permutations and $2^n$ counts for change in the direction of welding, i.e., several million for typical weld consisting of 10 weld passes or more.

More affordable approaches have been developed to generate a sufficient and reliable level of understanding of the behavior of structures in order to find an optimal sequence with a few numbers of simulation. One approach is to use a fast but low-fidelity model that captures the most dominant physics of the problem, for example, depositing each weld pass at once [2]. Although

¹ Corresponding author and presenter; Mahyar.Asadi@applusrtd.com
such a code or algorithm loses accuracy, it provides a useful approximation of relative behavior for judgment between weld sequencing scenarios. In many design cases, a designer can decide based on this rough approximation of the behavior. Better fidelity can be obtained through emerging machine learning (ML) algorithms ranging from the regression, classification, neural network, deep learning, support vector machine, and so forth where the machine can learn about the behavior of a system within multiple level non-linearity.

**Digital Weld Sequence Engineering**

We previously [3] developed a combinatorial optimization algorithm for weld sequence design based on joint rigidity where the methodology can find the best sequence with a minimal number of welding simulations. The quickest joint rigidity method uses “n” simulations to find a sequence for minimal distortion where “n” is the number of weld passes. The advanced progressive joint rigidity method can select a better sequence with \( n(n+1)/2 - 1 \) analysis. This methodology uses significantly small sub-space of the total combinatorial possibility of welding “n” passes when compares to \( 2^n n! \) possible scenarios. The main limitation of our joint rigidity method was the progressive nature of the methodology that prevents it from parallel computing and therefore CPU time can increase for a large number of weld passes.

Later, we presented another methodology [4] to construct a meta-model of distortion based on surrogate algorithm [5] in combinatorial space. This algorithm was well suited for parallel computing and can find the best sequence by running “4n” independent simulations in parallel and within the time frame of a single simulation run. The algorithm offers a fast and reasonable convergence to the optimal solution. Nevertheless, even fast meta-models are not sufficient for exploring a large number of weld passes, unless coupled with a valid search algorithm. A search algorithm was used from [6] based on a genetic algorithm (GA) to find an optimal welding sequence directly. However, this GA was directly coupled with the expensive simulation perdition and therefore was limited to simple analysis. Some new evolutionary operators were introduced in [7] and [8] that are more suitable for weld sequence design. As a method used in [9], we coupled our meta-model with the search algorithm to explore the sequence design space. This algorithm works well on a structure with a limited number of weld passes, for example, less than ten passes. However, the fidelity of prediction fades for a more significant number of weld passes.

In this paper, we introduced another methodology that constructs a deep learning artificial neural network for data-driven prediction. When comparing our deep learning here to our surrogate methodology in [4], deep learning requires larger training-set than surrogate but converges to significantly better fidelity at a more significant number of weld passes (e.g., higher than ten). The resulting optimal sequence significantly reduced the final distortion, and we showed the result on a panel structure with eleven weld passes. A summary of this methodology is presented here. Experimental validation is not part of this paper; however, simple experimental tests were performed in the background for validation of distortion and thermal prediction.

**Panel Structure**

Panel fabrication is part of many engineering structures, and welding is the sole fabrication method to erect such structures. In this paper, a panel, without the loss of generality, is selected to present our methodology and find the best welding sequence pattern for minimal distortion on the panel plate. Figure 1 illustrates the panel structure with eleven weld passes that connect a 658x360x19 mm panel plate to multiple stiffeners with varied dimensions and thickness from 42 to 50 mm as shown in this figure. There is no symmetry in the configuration of stiffeners, and the stiffeners are tack-welded on both ends before welding starts. An optimal clamping pattern was designed as
a separate task where a computational weld mechanics (CWM) tool was used to evaluate several clamping scenarios and to iterate toward the optimal clamping shown. The detail for the optimization of the clamping pattern is not in the scope of this paper. The panel material is Aluminum 6061 T6, and material properties were temperature dependent on the analysis. By convention, Figure 1 also shows the name designation for each weld pass, and changing capital letters to little letters means changing in the direction, as well, as the objective function that was characterizing the distortion using Eq. 1. Our task was to determine the welding sequence out of (A/a, B/b, C/c, D/d, E/e, F/f, G/g, H/h, I/i, J/j, K/k) welds.

Figure 1 Panel structure for JRM implementation (left), FEA nodes used for the objective function Eq.1 (top right), and tag and direction convention for weld passes in the panel.

\[
\left( \frac{1}{J} \sum_{j} \left( \sqrt{dx^2 + dy^2 + dz^2} \right) \right) \left( \max \frac{dx}{dy} \right) - \min \left( \frac{dx}{dz} \right) 
\]

(1)

**Computational Setup and Analysis of Weld**

A full 3D model of the panel was created using Abaqus Welding Interface (AWI) and in-house subroutines. The AWI uses the fusion line defined by the user and assigns a melting temperature. We used the Dirichlet temperature; however, AWI also offers a flux-based model based on Goldak’s Double Ellipsoid [1]. The user controlled the weld sequence through an in-house subroutine. This subroutine allowed for the automation of each weld pass in sequence. The welding time was automatically calculated from the pass length and the torch speed. Weld passes were deposited in five chunks to save CPU time while capturing the effect of welding direction. A series of cooling down steps were added after the welding was finished. Figure 2 shows snapshots of welding thermal results.

In this analysis, the initial temperature was 21 °C. A convection boundary condition generated a boundary flux on all external surfaces. The temperature-dependent convection coefficients \((w/m^2 °C)\) is computed from Eq. 2 [10] where \(T\) is the temperature in °C.

\[
h_c = 7.2 - \frac{355000}{(T+273)^2} + 0.001 (T + 273)
\]

(2)

The stress analysis was quasi-static because inertial or dynamic forces are sufficiently small to neglect. Therefore, at each instant of time, the domain is in static equilibrium. However, the
temperature is time dependent and therefore the thermal strain due to thermal expansion is time-dependent. The initial state was assumed to be stress-free. The boundary conditions were identical to the clamping defined and shown in Figure 1. The system is solved using a time marching scheme with time step lengths used for thermal analysis. The stress analysis followed immediately after the thermal analysis. Figure 3 shows the plate displacement after a given sequence.

![Figure 2](image1.png) A snapshot of welding thermal analysis for the start (left) and end (right) of a pass.

![Figure 3](image2.png) Plate displacement after completing a sequence.

**Machine-Learning for Engineering**

These days, artificial intelligence (AI) is so pervasive including in engineering such that we use it without knowing it. High-performance computing (HPC) enables engineers to perform data-driven engineering where a computer can act without being explicitly programmed. Data is still at the heart of the matter. The current data-driven algorithms like machine learning (ML) takes a large initial data set to construct a model and make a prediction, i.e., tens of thousands. In most engineering applications, there is no initial large data set to draw on.

The challenge in the training set is to select the minimum number of possible welding sequences that can adequately represent the entire population of welding sequences. We took the idea of the sampling technique from [5] and scripted code to generate the training set automatically. In brief, our sampling technique sequentially selects among randomly generated sequences until fully satisfies two constraints which are called type 1 denoted R’ and type 2 denoted R” matrix in [5]. R’ matrix assures that all weld passes and directions occur at least once in every position. The second constraint, R” matrix, assures every pair occurs at least once somewhere in sequence. There is no single solution for a training set, but an optimal R’ and R” matrix can lower the number of sequences in the training set while remain unbiased on sequence selections. The theoretical minimum number of sequences in a training set is “4n” where “n” is the number of weld passes, however, the training set can become practically larger than “4n” for larger “n”.

1005
The critical implementation in the algorithm that directs it toward a minimum number of sequences is a pairing check. For this, every randomly-generated sequence is checked to see how many new pairs in the $R''$ will be satisfied by adding this sequence to the set of sequences. Inherently, this approach minimizes the repeat if individual pairs, i.e., fewer sequences are required to assure every pair occurs at least once. Our optimal training set had 61 weld sequences that were used for ML training.

**Artificial Neural Network (ANN) Construction**

The limited training data issue was addressed with the application of Theory-guided machine learning (TGML) [11]. This TGML was implemented through an informed selection of initial training dataset as well as the physics-guided feature vector. The initial training dataset was based on a previously explained informed training-set selection. Further, a physics-based feature selection approach was used to include critical physical aspects of a sequential welding process. Four main components of the feature vector were:

- Weld Position
- Weld Orientation
- Weld Vicinity
- Distortion Resemblance

Development and training of a Dense Neural Network (DNN) in the present work, take advantage of the high-level API, Keras, running on top of Tensorflow library. The training dataset consists of 61 selective samples, and test-set includes 20 samples. Although the DNN requires the application of all 61 samples during training, the order of these samples in the input matrix is randomized to avoid bias. Feature space ($x$) is standardized ($x'$) using the built-in function form Scikit Learn. Model training uses Rectified linear unit (ReLU) [12] activation function and an L2-regularized Adam optimizer [13]. A linear activation function was used for the output layer. The neural network trained by limited training data is prone to the noises, hence overfitting. Dropout method [14] is used to address this overfitting issue. The loss function was set to root-mean-square-error (RMSE).

In addition to RMSE, correlation coefficient ($r$-value) and the slope of the regression line evaluate the developed model’s predictability for each data point in the test set. Finally, model hyperparameters are selected using a Bayesian optimization [15] tool from SciKit Optimize library. These hyperparameters comprise; the number of hidden layers, number of neurons in a hidden layer, dropout rate, and regularization coefficient.

**Panel Distortion Control**

We constructed a series of independent ML models where each represents the deposited weld, except for the first weld that has an identical FEA solution. An Artificial Neural Network (ANN) defines the history of weld sequence deposition and makes a prediction when informed with four main components of the feature vector explained earlier. The ANN hyper-parameters (here, “dropout”, Number of hidden layers”, “number of neurons in every hidden layer”, and “learning rate”) were automatically optimized by a Bayesian Optimization tool (SciKit Optimize). Figure 4 compares ML prediction and FEA prediction of distortion on the panel after depositing each weld pass for a given weld sequence. ML prediction performs better in terms of accuracy for initial depositions than the later ones when compared with FEA; however, the CPU time for ML prediction was instantaneous on a modest processor such as in I-Pad vs. FEA with several hours of CPU time on an HPC server.
Figure 4 Comparing ML prediction with FEA after each pass of sequence fGHaJDcEKbi
**Conclusion**

Weld sequence engineering is now practically feasible for our complex structures. This practice requires the skill of digital engineering of welding together with different search methodologies for defining a proper weld sequence. A designer can choose between different methods based on available time and budget to achieve a proper sequence. A better sequence takes longer engineering time and further budget.

Quick JRM [14] can find a weld sequence with “n” parallel computations and CPU time of a single weld simulation where “n” is the number of weld passes. This method is the fastest and least expensive method for a very tight schedule and budget.

In most projects, the schedule and budget can accommodate the Progressive JRM [24] where the designer can find a much better weld sequence at the cost of longer CPU time. The Progressive JRM needs \( \frac{n(n+1)}{2} - 1 \) simulations which this is significantly small sub-space of the total combinatorial possibility of welding “n” passes when compares to \( 2^nn! \) possible scenarios.

Machine learning algorithms such as surrogate model [15] and ANN (this paper) are suitable for design with tight tolerance where the optimization requires considering the non-linear effect of weld pass pairs, trios, quadros, and so on. The surrogate model needs minimum “4n” to “6n” parallel simulations to construct the surrogate model, and usually adds extra iterations of "n" parallel simulations to find a good solution for larger “n.” ANN needs 80 to 100 simulations to form minimal fidelity and additional iterations for improving the fidelity. Therefore the surrogate is more effective for small “n” (less than 10), and ANN works better or large “n” (higher than 10).

For repetitive work and mass production that continuous improvement is desirable, the evolutionary algorithms such as GA [15] can be constructive in particular when coupled with fast prediction tools such as ML algorithms. The evolutionary algorithms can continuously search the total design space of \( 2^nn! \) possible scenarios for a better solution.

Automation and integration of these methodologies into welding simulation packages are a crucial factor for practical applications. Nevertheless, the skill of digital engineering is paramount.

**References**


