Proceedings of IDETC/CIE 2005
ASME 2005 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
September 24-28, 2005, Long Beach, California, USA

DETC2005-85257

A Virtual Model for Aluminum Hot Forging Using An Artificial Neural Network Material Model within Finite Element Analysis

B. Scott Kessler and A. Sherif El-Gizawy
University of Missouri-Columbia
E2411 Engineering Building East
Columbia, MO 65211
Phone: (573) 882-9569, Fax: (573) 884-5090

ABSTRACT

The accuracy of a finite element model for design and analysis of a metal forging operation is limited by the incorporated material model's ability to predict deformation behavior over a wide range of operating conditions. Current rheological models prove deficient in several respects due to the difficulty in establishing complicated relations between many parameters. More recently, artificial neural networks (ANN) have been suggested as an effective means to overcome these difficulties.

In the present work, a previously developed ANN with the ability to determine flow stresses based on strain, strain rate, and temperature is incorporated with finite element code. Utilizing this linked approach, a preliminary model for forging an aluminum wheel is developed. This novel method, along with a conventional approach, is then measured against the forging process as it is currently performed in actual production.

INTRODUCTION

Finite element modeling of manufacturing processes has been gaining wider acceptance over the last several years. Modeling prior to the start of actual production can save considerable time, effort, and money. While modeling may provide these benefits, it must be kept in mind that finite element software can only provide accurate simulations of a "real" process if appropriate material models are utilized.

In this paper, a novel material model is presented and compared to conventional models. For lack of an exact mathematical model, an intelligent algorithm, the ANN, will be used to map relationships between the hot forging parameters and the flow stress of the material. The ANN learns the patterns and offers robust and adaptive processing capabilities by implementing learning and self-organization rules.

In the present work, an ANN is generated and trained based on physical testing of 6061 aluminum as found in published literature [1]. This trained network is then used as the material model, which when linked with a commercial finite element code, provides a model capable of more accurately reflecting actual experience. Much of this results from a robust ANN's ability to predict outputs between, and to some degree, outside the bounds established by a training set. For this application, values of strain, strain rate, and temperature not matching the family of curves used for training can be submitted to the network, and intermediate values of flow stress found.

The conventional modeling approach requires experimental data curves to be fit to some form of hardening law and intermediate values interpolated by some means. While this may not be particularly difficult, the curve fitting process itself can be exceptionally tedious and in many cases does not produce accurate fits of the data. The ANN is much simpler to implement. Set up the network, train it, submit input values, and output is generated.

To these ends, initially, a review of conventional material models and their limitations will be presented. The various factors leading to difficulties in addressing real problems will be summarized. The development and use of artificial neural networks will be covered with the specific aim of developing an unconventional material model for linking with finite element code.

Conventional and ANN-based material models are then developed for 6061 aluminum using published data. The

training of the ANNs is accomplished using MATLAB's Neural Network Toolbox. The conventional model is used directly with the commercial finite element code, ABAQUS. The ANN-based model requires the generation of Fortran code that is linked by means of a subroutine within ABAQUS. The model predictions, both conventional and ANN based, are then directly compared to the actual forging operation as it is now carried out.

BACKGROUND

Conventional Material Modeling

Forging operations require material deformation well beyond the elastic limit, and as such require elasto-plastic models. For the most part, the models differ in their approach to describing the plastic portions of deformation (i.e. hardening behavior). Typically, finite element codes treat the elastic and plastic portions of strain separately as shown in Eq. (1).

$$\varepsilon = \varepsilon_e + \varepsilon_p = \frac{\sigma}{E} + \varepsilon_p \qquad \text{(EQ 1)}$$

Several significant difficulties arise when attempting to model material behavior beyond yield. As shown in Fig. 1, real materials may exhibit difficult to characterize behavior once plastic deformation begins to take place. While many materials display strain flattening under a narrow set of conditions, other behaviors result when outside that range. Strain softening and/or oscillating flow stresses may occur due to dynamic recrystallation, recovery, or other somewhat poorly understood phenomenon.

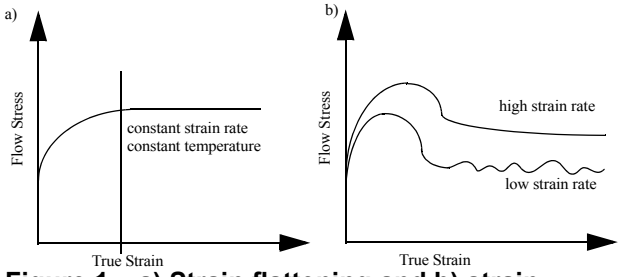


Figure 1. a) Strain flattening and b) strain softening behavior [2].

Typically, forging processes assume that flow stresses, σ_f, follow some form of the power law. Again treating elastic and plastic strains separately, the Ramberg-Osgood relation (Eq. (2)) incorporates the power law treatment for plastic portion of deformation. The reference stress, K, and strain hardening exponent, n, are determined through curve fitting a stress/strain (σ_f/ε) diagram obtained from compression testing the material to be modeled.

$$\varepsilon = \frac{\sigma}{E} + \left(\frac{\sigma_f}{K}\right)^{\frac{1}{n}} \qquad \text{(EQ 2)}$$

If the material is being hot worked (i.e., many forging processes) the strain rate is substituted for strain as given by Eq. (3).

$$\dot{\varepsilon} = \left(\frac{\sigma_f}{C}\right)^{\frac{1}{m}} \qquad \text{(EQ 3)}$$

Severe difficulties arise in fitting the above equations to actual test data. For each of the above cases, compression tests are performed and the constants determined through curve fitting. Other more complicated equations, containing additional constants have been developed [3][4][5]. While they may, in theory, more properly reflect behavior, they require more tedious curve fitting techniques with the requisite possibility for error. In any case, constant strain rate testing must be employed using several different strain rates and temperatures [6][7]

Neural Networks

Over the last decade, several artificial intelligence tools such as artificial neural networks (ANN), fuzzy logic, and genetic algorithms (GA) have been introduced and applied in the field of manufacturing process engineering [9][10]. They provide for more accurate models than the available analytical ones. More recently, artificial neural networks (ANN) have been proposed to describe the material flow stress under the considered processing conditions [11][12][13][14].

The general idea behind artificial neural networks is to emulate the signal processing scheme used by nature. Several dendrites accept input that a given neuron processes before exiting at the axon. The axon then in-turn transmits a signal to another neuron's dendrites. In this way, information is processed or modified appropriately as it passes through the nervous system. Artificial neural networks have performance characteristics similar to biological neural networks and are based on the following assumptions [15]:

- Information processing occurs at many simple elements called neurons.
- Signals are passed between neurons over connection links.
- Each connection link has an associated weight, which, in a typical neural net, multiplies the signal transmitted.
- Each neuron applies an activation function (usually nonlinear) to its net input (sum of weighted input signals) to determine its output signal.

A schematic of a simple multilayer artificial neural network is shown in Fig. 2. Each of the inputs is connected to each of the first hidden layer neurons and each of the first hidden layer neurons connects to each of the second hidden layer neurons. Finally, the second hidden layer combines to form a single output.

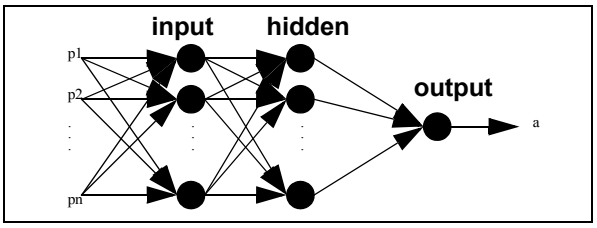
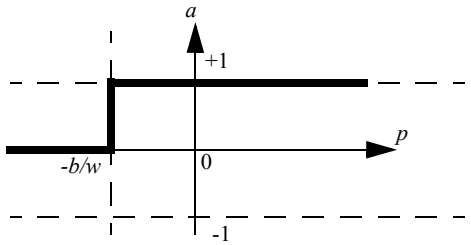


Figure 2. Schematic of a simple artificial neural network architecture.

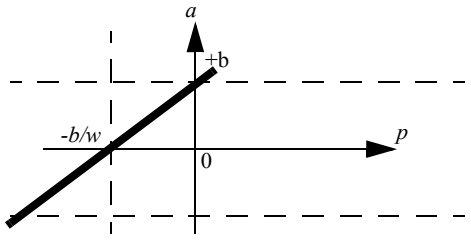
The ability of a network to mirror behavior results from the number of neurons, the number of layers, their interconnectedness, and by the transfer or activation functions chosen. Transfer functions vary from simple McCulloch-Pitts neurons [16] as given by Eq. (4) and Fig. 3 to those provided by Fig. 4, Fig. 5, and Eq. (5).

$$a = hardlim(n) = \begin{pmatrix} 1 \text{ if } n \geq 0 \\ 0 \text{ otherwise} \end{pmatrix} \qquad \text{(EQ 4)}$$



a = hardlim(wp+b)

Figure 3. *Hardlim* transfer function within MATLAB, where **W** is the weight(s), **p** the input(s), and **b** the bias.

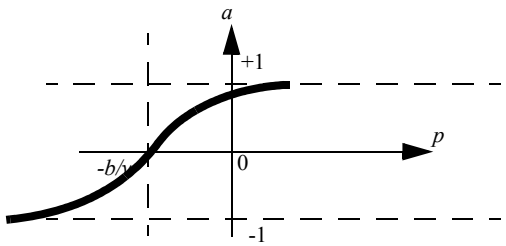


a = purelin(wp+b)

Figure 4. Linear transfer function, single-input *purelin* neuron.

Another commonly employed transfer function, tan-sigmoid, is provided by Eq. (5) and is shown graphically in Fig. 5.

$$a = \frac{2}{1 + e^{-2n}} - 1 \qquad \text{(EQ 5)}$$



***a** = tansig(**wp**+**b**)*

Figure 5. Tan-sigmoid transfer function, single-input *tansig* neuron.

As an example, the material model required for forging requires the determination of flow stress as it depends on strain, strain rate, and temperature; Eq. (6) shows the matrix form for the hidden layer with strain, strain rate, and temperature as inputs.

$$\text{tansig}\left(\begin{bmatrix} w1_{11} & w1_{12} & w1_{13} \\ w1_{21} & w1_{22} & w1_{23} \\ \cdots & \cdots & \cdots \\ w1_{s1} & w1_{s2} & w1_{s3} \end{bmatrix} \begin{bmatrix} \varepsilon \\ \dot{\varepsilon} \\ T \end{bmatrix} + \begin{bmatrix} b1_1 \\ b1_2 \\ \cdots \\ b1_s \end{bmatrix}\right) = \begin{bmatrix} a_1 \\ a_2 \\ \cdots \\ a_s \end{bmatrix} \qquad \text{(EQ 6)}$$

The subscript *s* refers to the particular neuron (i.e., *s* equals the number of neurons per input value in a layer). The hidden layer values, *a*, are then fed into the output layer as shown in Eq. (7) which results in a single value for the flow stress. The dot product in this case functions the same as a linear transfer function.

$$\sigma = \begin{bmatrix} w2_1 & w2_2 & \cdots & w2_s \end{bmatrix} \bullet \begin{bmatrix} a_1 \\ a_2 \\ \cdots \\ a_s \end{bmatrix} + b2 \qquad \text{(EQ 7)}$$

Feedforward Backpropagation

Feedforward backpropagation (FBP) networks are commonly utilized for function approximation. The presentation above describes the feedforward portion of the network. Backpropagation refers to the particular method of adjusting or correcting the weights and biases to produce a network output consistent with the training set. This is accomplished by presenting a training set consisting of inputs that result in a known output. The weights and biases are initialized, the inputs presented, and an output determined. The output is compared against the target and the error determined. This error is used to adjust the weights and biases starting with the last layer and working backwards through the network. This procedure is repeated until an acceptable error is achieved. Several backpropagation schemes are available. The Levenberg-

Marquardt (LM) algorithm, a variation of Newton's methods is typically efficient, from a computational perspective, and has proven to provide reasonable results [17].

Modeling Difficulties

If the number of neurons is excessive, or training carried out for too many epochs, the network may produce wild swings developing a greater number of inflections than the data, producing overfitting (see Fig. 6).

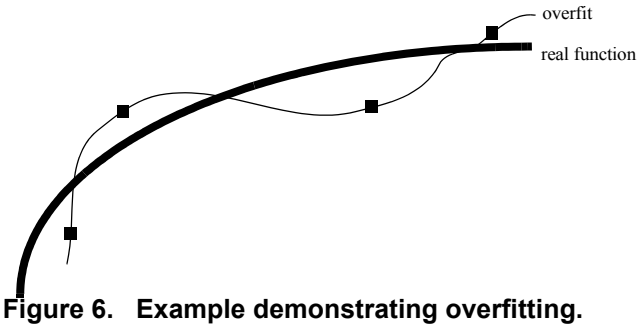


Figure 6. Example demonstrating overfitting.

Convergence, another issue, can result when training produces a network that may have found a local minimum, but not a global minimum. The learning rate specified, and many times, the initial conditions (i.e., starting values of the weights and biases) can produce a non-ideal solution as multi-layer networks may have many local minima.

If prior knowledge about possible relations between particular input variables and expected output exists, it is often beneficial to include those relations through modification of the inputs. In the case of flow stresses, it is known that the log of the stress may relate linearly to the log of the strain rate. So, it may make sense to provide the network with log strains or strain rate in addition to, or instead of strains and strain rates.

Additionally scaling or normalizing the input variables and/or training values may enable the transfer functions to better handle the data, though with the added complication of re-processing the output from the trained network.

Bayesian Regularization

Bayesian regularization (BR) addresses overfitting, or improving generalization, by attempting to reduce the model complexity to the minimum necessary for reasonable performance. Bayesian methods automatically incorporate the principle of "Occam's razor" which states that the simplest model that adequately fits the data should be preferred over more complex representations [18][19][20].

Bayesian regularization adds a term to the performance function, or squared errors, used when comparing targets to ANN output. If E_D represents the squared error and E_W the sum of squares of the weights, then a modified performance index can be developed Eq. (8).

$$F = \beta E_D + \alpha E_W \qquad \text{(EQ 8)}$$

where α and β are objective function parameters. As β grows larger and α grows smaller, then network errors are forced to be smaller. If the reverse is true, training attempts to minimize the squared weights. Decreasing the values of the weights aids in smoothing the network response and should improve generalization. The basic effect produced by BR, is to decrease the number of weights and biases that actively transform the input.

6061 ALUMINUM MATERIAL MODEL DEVELOPMENT

The material models that follow, conventional and ANN, are both developed from published literature [1]. The flow stress curves shown in Fig. 7 and Fig. 8, provide data for temperatures of 300 and 550ºC. The curves were digitized to provide numerical input for the ANN training. In addition, the same literature provides tabular flow stress data for several strains and strain rates at 450ºC as shown in Table 1 (the forging process to be modeled is assumed to be isothermal and at 450ºC).

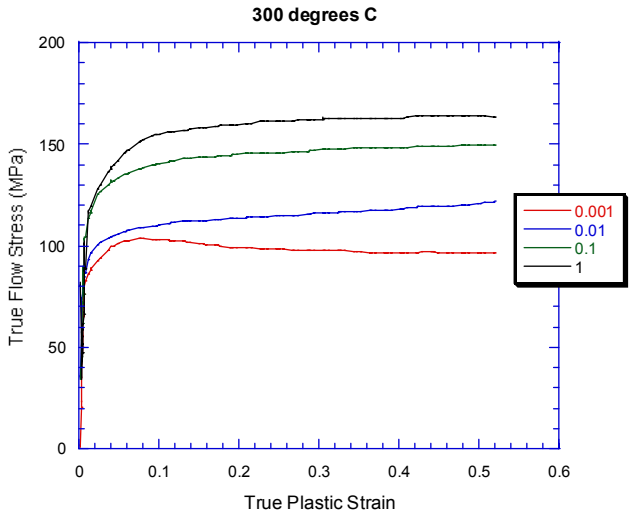


Figure 7. 6061 aluminum flow stress as a function of strain for 300ºC.

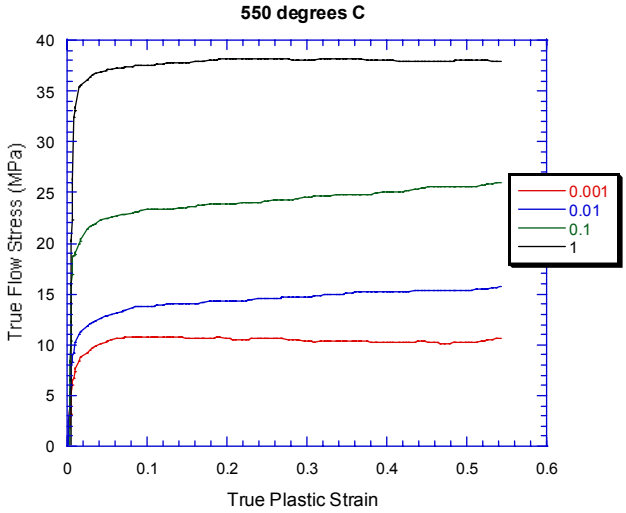


Figure 8. 6061 aluminum flow stress as a function of strain for 550ºC.

Table 1: Flow stress (MPa) at 450ºC.

Rate/Strain	0.1	0.2	0.3	0.4	0.5
0.001	19.6	19.5	19.8	19.9	20.4
0.01	30.2	30.9	31.5	32.1	32.6
0.1	42.3	43.0	44.4	45.1	45.6
1.0	60.5	63.0	64.5	64.4	64.3

Conventional Material Model

ABAQUS provides two applicable conventional material models; one using the power law approach and another by directly inputting tabular values for strain, strain rate, and flow stress as provided above. ABAQUS determines strain rate dependence using the overstress power law as given by Eq. (9), a slight variation on Eq. (3).

$$\dot{\varepsilon}_{plastic} = D\left(\frac{\sigma_{flow}}{\sigma_{\text{static yield}}} - 1\right)^{p} \text{ for } (\sigma_{flow} \geq \sigma_{\text{static yield}}) \qquad \text{(EQ 9)}$$

As noted previously, curve fits must be generated for a power law equation implementation. In practice, when actual test data is available, tabular input is the preferred approach and as such is the path taken here. With either the overstress power law or the tabular based model, ABAQUS requires estimation of static yield stress and elastic modulus, in this case at 450ºC. Values of 15 MPa, and 54 GPa were used for static yield and modulus, respectively.

Neural Network Development

While many network architectures may produce acceptable results, the following demonstrates a reasonable approach [21]. Using the flow stress curves from Fig. 7 and Fig. 8, an initial training set comprised of 1008 total data points, split evenly for 300ºC and 550ºC, with equal numbers for points for each strain rate (i.e., 126 points for each separate curve, four curves per temperature) was extracted using digitizing software. In addition, a set of tabular training data consisting of flow stresses at strains of 0.1, 0.2, 0.3, 0.4, and 0.5 for strain rate of 0.001, 0.01, 0.1, and 1 for temperatures of 300, 350, 400, 450, 500, and 550ºC were also supplied from the same published source. The input data was modified prior to network training by using normalized values of the natural log of the strain, strain rate, and flow stress along with the reciprocal of temperature.

A neural network comprised of 15 input neurons followed by a second hidden layer of 3 neurons was trained with Bayesian Regularization. Figure 9 is an automatically produced aid for judging network training progression. The upper portion of the figure indicates the gradually decreased sum of squared errors. The middle section notes the squared value of the weights as the algorithm attempts to decrease this value. The bottom tracks the effective number of parameters (i.e., weights and biases). In this case the training was stopped after 4575 epochs due to the stabilization of the each of these curves.

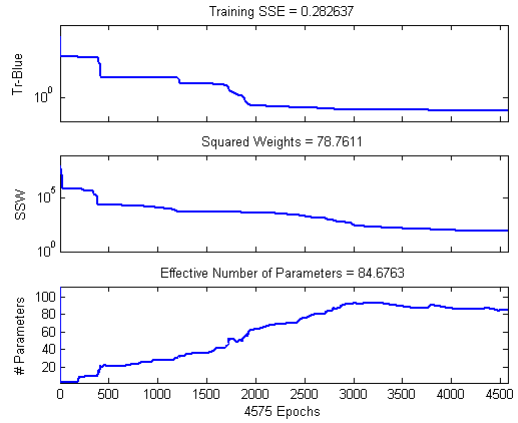


Figure 9. Progression of training.

As shown in Fig. 10 and Fig. 11, the network has the ability to almost perfectly match the targets when queried using the training input values. It also produces very accurate results when values of strain varying from 0 to 0.5 at intervals of 0.001 are presented for strain rates of 0.001, 0.01, 0.1, 1.0. This good performance is also demonstrated by the linear regression analysis shown in Fig. 12. When intermediate values of strain rate (0.005, 0.05, 0.5, and 5) are submitted, the curves produced appear to reflect that which would be anticipated based on

experience (see Fig. 13 and Fig. 14), though it should be noted that at a strain rate of 5 at 550ºC the ANN output does not appear reasonable. The final trial, establishes the network output for a temperature of 450ºC at 0.001, 0.01, 0.1, and 1.0 values of strain rate (see Fig. 15).

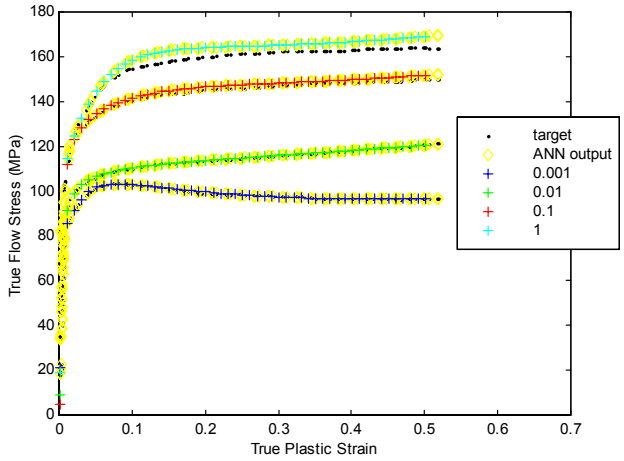


Figure 10. 15-3 BR neuron network output using data set C for 300ºC.

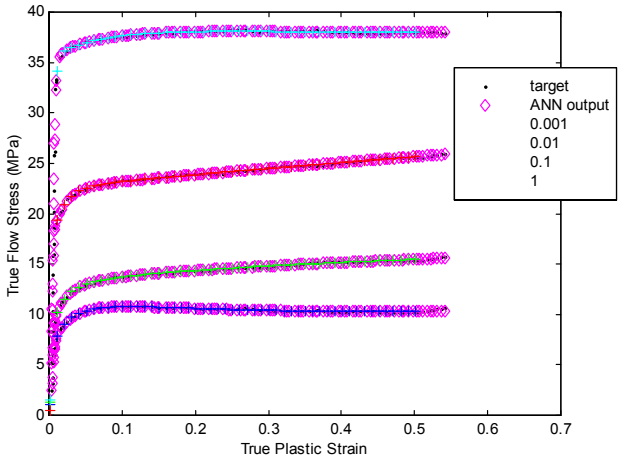


Figure 11. 15-3 BR neuron network output using data set C for 550ºC

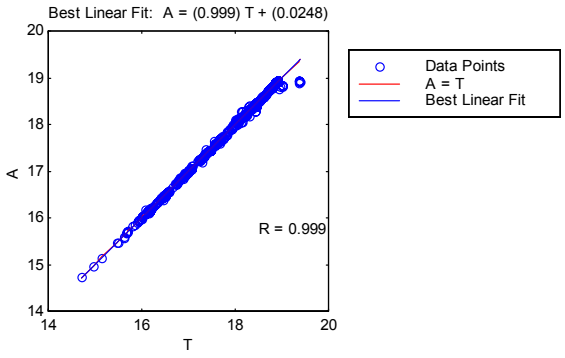


Figure 12. Linear regression for the 15-3 BR neuron network.

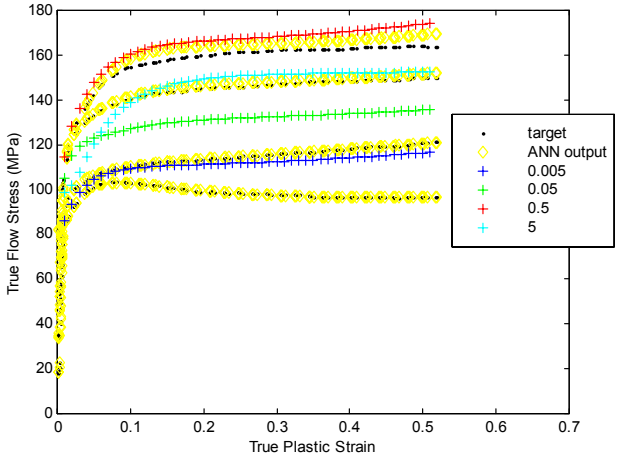


Figure 13. 15-3 BR neuron network output using data set C for 300ºC at intermediate strain rates.

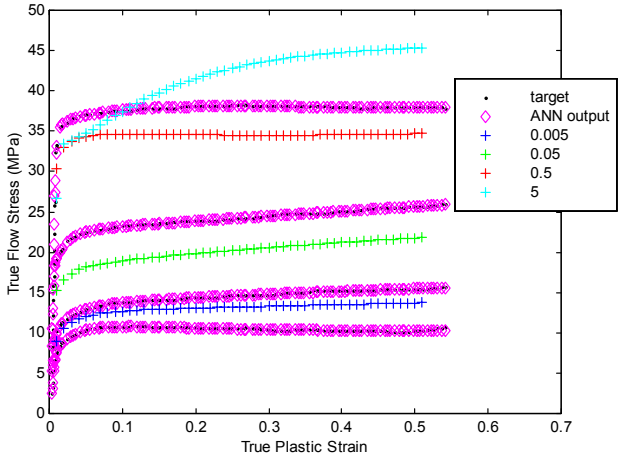


Figure 14. 15-3 BR neuron network output using data set C for 550ºC at intermediate strain rates.

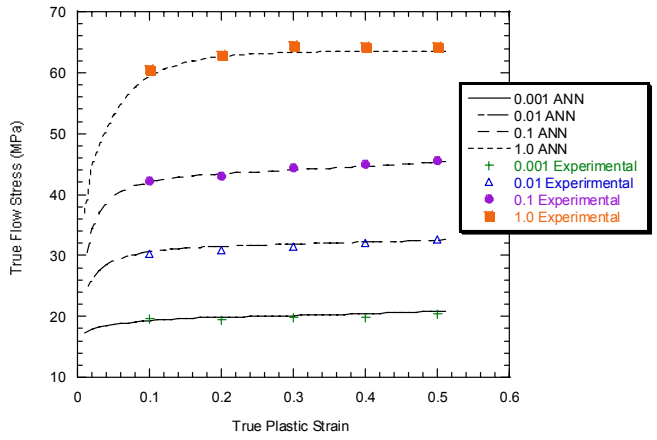


Figure 15. Neural network output compared to tabular experimental data at 450ºC.

EXPERIMENTAL DATA RETREIVED FROM WHEEL FORGING PROCESS

The ultimate aim of this research is to produce a virtual model that accurately reflects real world behavior. Verification of the model is established by measuring pertinent parameters during the actual forging of an aftermarket automotive wheel. The forging operation is basically a three step process. A billet at approximately 450ºC (842ºF), measuring 203.2 mm (8 in) in diameter by 467.36 mm (18.400 in) long is stood up in the center of the bottom die in a hydraulic press. The top die is run downward until the maximum press capacity is achieved (approximately 34.47 MPa or 5000 psi). The crosshead is then raised and the part ejected so a graphite lubricant at 74ºC (165ºF) can be applied. The part is reseated and the top die closed again with maximum pressure applied. The force is then released momentarily and the maximum applied again.The first application of force utilizes two side cylinders, each of 330.2 mm (13 inches) diameter. The later two operations add the center ram having a 889 mm (35 inch) bore. Figure 16 shows the billet in crossection after completion of the forging operations.

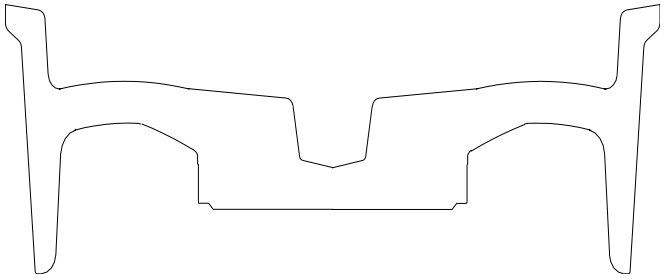


Figure 16. Crossection of wheel after forging operations.

A National Instruments DAQCard-AI-16XE-50 multifunction I/O card in conjunction with Labview software was used for data acquisition. The forging press crosshead displacement was measured with an LVDT (G.L. Collins Corp., LMT-689VO1 with a Schaevitz ATA-101 amplifier) and the hydraulic pressure acquired with a pressure transducer (Sensometrics SP97KFS). Billet surface temperature measurements were also made at the beginning and completion of each operation. Readings were taken for ten consecutive wheel forgings. Figure 17 shows how the displacement of the crosshead and hydraulic pressure varied through time for a representative trial. Figure 18 provides the force versus displacement curve by converting hydraulic pressure to force based on the ram crossection.

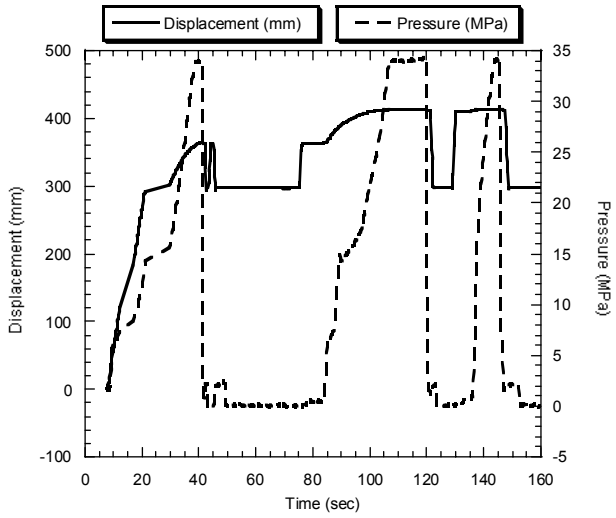


Figure 17. Forging operation crosshead displacement and hydraulic pressure.

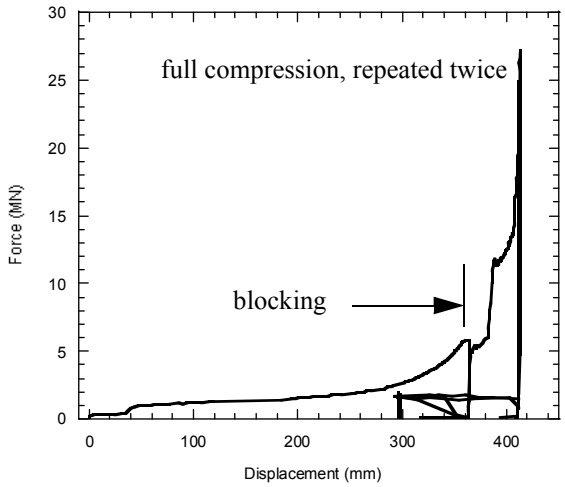


Figure 18. Forging force versus displacement curve.

ANN AND CONVENTIONAL FEA COMPARISON

The finite element model, whether conventional or ANN based, assumed the forging process to be isothermal (450ºC) using rigid upper and lower dies. The lower die was fixed and the upper die moved downward using a velocity curve derived from actual production. An axisymmetric model was developed, consisting of CAX4R elements (4-node bilinear, reduced integration, hourglass control). The initial mesh employs square elements having 2.5 mm sides. In an effort to control the severe mesh distortions encountered, Abaqus' explicit solver possessing adaptive meshing capabilities was utilized. Adaptive meshing is carried out every 5 increments with up to 3 sweeps.

For implementation of the neural network material model, the weights and biases from the 15-3 neural network developed above were input into a FORTRAN program that produces the feedforward portion of the ANN. ABAQUS, through its VUMAT capability calls the program, which includes definitions for the yield surface (based on Von Mises stresses), the flow rule, and the evolution law (i.e. hardening behavior). The conventional model determines flow stresses based on regularization of Table 1 above. Previous work by the author has shown the tabular approach to be superior to the overstress power law for simple compression of aluminum billets [21].

Figure 19 provides a comparison of experimental, conventional FEA, and ANN based FEA force-displacement curves. Figure 20 shows the initial blocking operation. During the initial loading, the ANN based model predicts lower forces which more closely matches experimental values. The conventional model also appears more unstable and eventually the model collapses due to excessive element distortion. The ANN model, while not run to completion at this point, still trends upward with curvature similar to that measured.

Figure 21shows the second portion of the forging operation after mesh refinement which allows the model to run to completion. The deformed shape was extracted from the model and remeshed with much smaller elements. Fortunately, the process is carried out near aluminum's solution temperature which prevents significant residual stresses from developing. The ANN-based model much more closely mimics the experimental results until near completion of the process. Both models predict much larger forces that actually achieved, likely due to problems with the resulting elements sizes and shapes. This particular forging problem tests the ability of the FEA code to produce a reasonable solution due to the excessive amounts of plastic deformation.

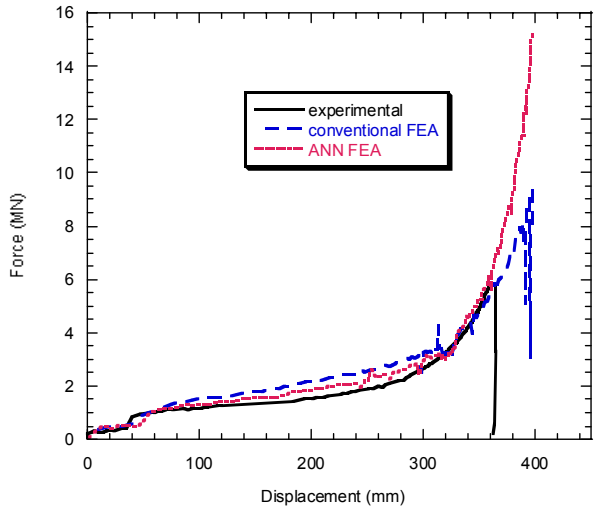


Figure 19. Comparison of measured forces to conventional and ANN based FEA predictions.

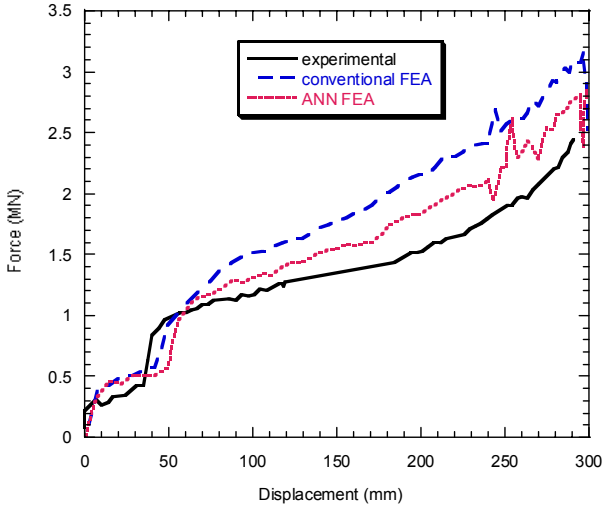


Figure 20. Blocking operation.

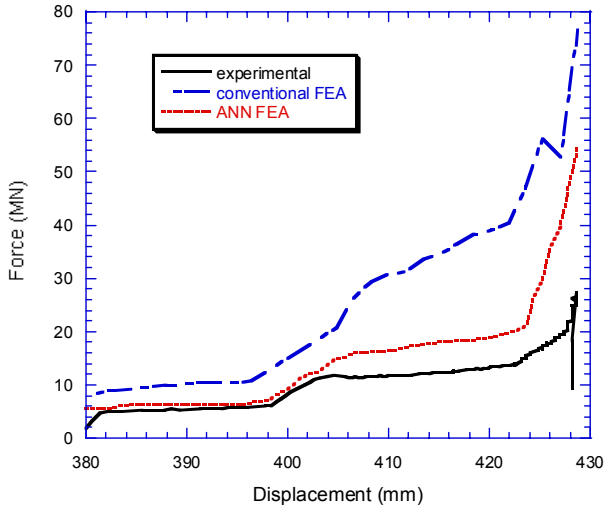


Figure 21. Second forging operation.

The figures clearly demonstrate that the ANN material model possesses a superior ability to mirror experimental results.The tabular based model starts out providing a reasonable approximation, but fails if continued beyond blocking. It should also be noted that values of static yield stress and elastic modulus had to be estimated for input into the model. The ANN model does not require yield estimates as it has the ability to supply flow stress values for the entire range of strain and strain rates.

CONCLUSIONS

The above work demonstrates the ability of an ANN material model, when implemented within a commercial finite element code, to produce virtual models more closely matching experimental experience as compared to conventional material modeling methods. The ANN model, while not quite as simple as the tabular approach, is still fairly straight-forward to implement now that the appropriate code has been developed. Further, the conventional curve fitting approaches, utilizing constitutive equations, require compression tests be carried out at constant strain rates. While not demonstrated in this paper, an ANN would have the ability to be trained using typical compression tests performed under constant velocity or even with varying velocities achieved during actual forging. This capability would enable tests to be carried out on equipment available in a manufacturing setting rather than under strict laboratory conditions.

REFERENCES

[1] Prassad, Y.V.R.K., Sasidhara, S., Gegel, H.L., and Malas., J.C., 1997, *Hot Working Guide: A Compendium of Processing Maps*, ASM International, p 104-105.

[2] Lange, Kurt, 1985, *Handbook of Metal Forming,"* McGraw-Hill, New York, p 3.21.

[3] Sellars, C.M. and Tegart, W.J.McG., 1963, *Mem. Sci. Rev. Metall.,* **63**, p 86.

[4] Laasroui, A., and Jonas, J.J., 1966, "Prediction of Steel Flow Stresses at High Temperatures and Strain Rates," *Metall. Trans. A,* **22**, p 1545-1558.

[5] Brown, S.B., Kim, K.H., and Anand, L., 1989, "An Internal Variable Constitutive Model for Hot Working of Metals," *Int. J. Plast.,* **5**, p 95-130.

[6] "Standard Test Methods of Compression Testing Metallic Materials at Room Temperature," ASTM E9-89a, *Annual Book of ASTM Standards.*

[7] "Standard Practice for Compression Tests of Metallic Materials at Elevated Temperatures with Conventional or Rapid Heating Rates and Strain Rates," ASTM E209-00, *Annual Book of ASTM Standards.*

[8] Zhao, Dan, 2000, "Testing for Deformation Modeling," *ASM Handbook,* ASM International, **8**, p 798-809.

[9] Rao, K.P., and Prasad, Y., 1990, "Neural Network Approach to Flow Stress Evaluation in Hot Deformation," *Journal of Materials Processing Technology,* **53**, p 552-566.

[10] Hodgson, P.D., Kong, L.X., and Davies, C., 1999, "The Prediction of the Hot Strength in Steels with an Integrated Phenomenological and Artificial Neural Network Model," *Journal of Materials Processing Technology,* **87**, p 131-138.

[11] Enemuoh, E.U., and El-Gizawy, A. Sherif, 2002, "A Robust Neural Network Model for Online Prediction of Process-Induced Damage in Drilling Epoxy Composites," Proceeding 3rd CIRP Int. Sem. On Intelligent Computation in Mfg. Eng., Naples, Italy, p 183-188.

[12] El-Gizawy, A.S.,Hwang, J.Y, and Brewer, D.H., 1990, "A Strategy for Integrating Product and Process Design of Aerospace Components," *Manufacturing Review*, ASME, **3**, p 178-186.

[13] Enemuoh, E.U., and El-Gizawy, A. Sherif, 2002, "A Robust Neural Network Model for Online Prediction of Process-Induced Damage in Drilling Epoxy Composites," Proceeding 3rd CIRP Int. Sem. On Intelligent Computation in Mfg. Eng., Naples, Italy, p 183-188.

[14] Bariani, P.F., Bruschi, S., and Negro, T.Dal, 2002, "Neural Networks to Describe Superalloys Rheological Behavior Under Hot Working Conditions," Proceeding 3rd CIRP Int. Sem. On Intelligent Computation in Mfg. Eng., Naples, Italy, p 585-588.

[15] Fausett, L., 1994, *Fundamentals of Neural Networks, Architectures, Algorithms, and Applications*, Prentice-Hall,

Englwood Cliffs, NJ, p xiii.

[16] McCulloch, W. and Pitts, W., 1943, “A logical calculus of the ideas immanent in nervous activity,” *Bulletin of Mathematical Biophysics*, **5**, p 115-133.

[17] Hagan, Martin T., Demuth, Howard B., and Beale, Mark, 1996, *Neural Network Design,* PWS Publishing, p 12-19.

[18] Jeffreys, H., 1939, *Theory of Probability*, Oxford University Press, Oxford.

[19] Gull, S.F., 1988, “Bayesian inductive inference and maximum entropy,” *Maximum Entropy and Bayesian Methods in Science and Engineering, Vol. 1: Foundations*, G.J.Erickson and C.R. Smith, eds., Kluwer, Dordrecht, p 53-74.

[20] MacKay, David J.C., 1992, “Bayesian Interpolation,” *Neural Computation*, Massachusetts Institute of Technology, **4**, p. 415-447.

[21] Kessler, B. Scott, El-Gizawy, Sherif, and Smith, Douglas E.,2005, “Incorporating Neural Network Material Models within Finite Element Analysis for Rheological Behavior Prediction,” Proceedings of 2005 ASME PVP Conference, Denver, Colorado.