NEURAL NETWORK MODELING OF THE LITHIUM/THIONYL CHLORIDE BATTERY SYSTEM

Christian C. O'Gorman, David Ingersoll, Rudolph G. Jungst, and Thomas L. Paez

Sandia National Laboratories
Albuquerque, NM 87185-0557

Abstract

Battery systems have traditionally relied on extensive build and test procedures for product realization. Analytical models have been developed to diminish this reliance, but have only been partially successful in consistently predicting the performance of battery systems. The complex set of interacting physical and chemical processes within battery systems has made the development of analytical models a significant challenge. Advanced simulation tools are needed to more accurately model battery systems which will reduce the time and cost required for product realization. Sandia has initiated an advanced model-based design strategy to battery systems, beginning with the performance of lithium/thionyl chloride cells. As an alternative approach, we have begun development of cell performance modeling using non-phenomenological models for battery systems based on artificial neural networks (ANNs). ANNs are inductive models for simulating input/output mappings with certain advantages over phenomenological models, particularly for complex systems. Among these advantages is the ability to avoid making measurements of hard to determine physical parameters or having to understand cell processes sufficiently to write mathematical functions describing their behavior. For example, ANN models are also being studied for simulating complex physical processes within the Li/SOCl2 cell, such as the time and temperature dependence of the anode interfacial resistance. ANNs have been shown to provide a very robust and computationally efficient simulation tool for predicting voltage and capacity output for Li/SOCl2 cells under a variety of operating conditions. The ANN modeling approach should be applicable to a wide variety of battery chemistries, including rechargeable systems.

INTRODUCTION

Historically, battery development has relied heavily on a very pragmatic "build and test" approach to demonstrate that a particular design can meet the requirements of a given application. Due to the time-intensive nature of the testing that is needed, this point design approach is expensive and typically has difficulty responding to changing or divergent sets of requirements. Performance and manufacturing models of power sources have aimed to alleviate these problems and provide enhanced capability for evaluating product performance, reliability, and life. Analytical models, however, have only been partially successful. The complex set of interacting physical and chemical processes within battery systems has made the development of such analytical models a significant challenge. The availability of advanced design tools is needed to more accurately evaluate and predict product performance. In order to accomplish this, an infrastructure of proven models, tools, and processes will be necessary. In an effort to provide such advanced simulation tools, Sandia has initiated a model-based design strategy for battery development, beginning with the performance of lithium/thionyl chloride cells. As an alternative approach, we have begun development of cell performance modeling using non-phenomenological models for battery systems based on artificial neural networks (ANNs).

Modeling of power source behavior with ANNs has not been previously demonstrated, but this approach clearly has benefits, for example enhanced computational efficiency. Artificial neural networks are inductive models for simulating input/output mappings with certain advantages over phenomenological models, particularly for complex systems. Among these advantages is the ability to avoid making measurements of hard to determine physical parameters or having to understand cell processes sufficiently to write mathematical functions describing their behavior. An ANN "learns" system behavior based on training by providing examples of measured output for different sets of input conditions. In the power source case, the inputs can include many factors, such as the chemistry, state-of-charge, load profile, size, temperature, and possibly the previous use and environmental history, while the outputs may be voltage, current, or deliverable capacity. When modeling battery systems with ANNs, time consuming and difficult

DISCLAIMER

measurements of the fundamental cell parameters are not necessary since only electrical performance data under the conditions of interest are used for training. A brief description of ANNs will be presented followed by a number of examples illustrating the use of ANNs in modeling battery systems.

ARTIFICIAL NEURAL NETWORKS (ANNs)

The ANN is an inductive, or data-based model for the simulation of input/output mappings. The ANN can be used in numerous frameworks to simulate many types of system behavior including physical, financial, and, as will be shown here, electrochemical systems. ANNs require training data to learn patterns of input/output behavior and, once trained, can be used to simulate system behavior within that training space. They do this by interpolating specified inputs among the training inputs to yield outputs that are interpolations of training outputs. The reason for using ANNs to simulate system behavior is they provide accurate approximations of system behavior and are typically much more computationally efficient than phenomenological models. This efficiency is very important in situations where multiple response or prediction computations are required. Some examples of computationally intensive applications are: (1) optimization of system performance, (2) system identification, (3) system design, and (4) Monte Carlo analysis of probabilistic system response.

The objective of any mapping ANN can be expressed mathematically using the following input/output notation and terminology. Let $\{x\}$ represent a vector of inputs presented to a system. The system to be simulated operates on the data in $\{x\}$, yielding an output vector $\{z\}$. There is a functional relationship, $\{z\} = g(\{x\})$, for $\{z\}$ and $\{x\}$ where the function g(.) is assumed to be deterministic but unknown. An ANN is constructed so that it can also be used to operate on the inputs $\{x\}$ to yield $\{y\}=h(\{x\})$. The function $h(.)$ is deterministic, has a pre-established framework, and parameters $\{p\}$. The function $h(.)$ will be an approximation to the system function g(.). Given a sequence of input/output exemplars $\{x_j\}$ and $\{z_j\}$, j=1,...,R, we seek to adjust the parameters $\{p\}$ of the ANN to minimize the error between the actual system output $\{z\}$ and the ANN output $\{y\}$, when presented with the same input $\{x\}$. This is accomplished through a training process that involves error minimization through the variation of the parameters $\{p\}$. The error minimization can be performed using a local search (for example, gradient search, least mean square, Newton's method) or global search (for example, simulated annealing, genetic algorithm). Once training is complete, it is hoped that, given an input $\{x\}$ different from those used during training, an ANN will yield outputs $\{y\}$, that are accurate approximations of the outputs $\{z\}$ produced by the system being modeled. The ANN serves as an interpolator of output vectors among the output exemplars as a function of the position of the input vector among the input exemplars. A diagram of this process is shown in Figure 1.

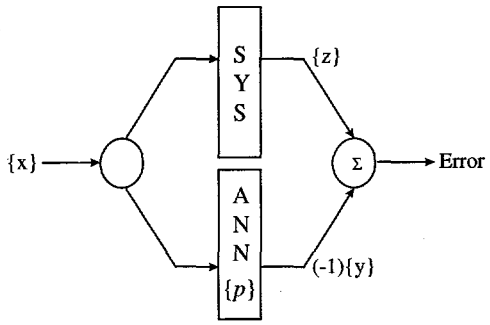

Figure 1. Diagram showing error of ANN model simulating a physical system.

Our ultimate objective is to simulate complex electrochemical systems for both optimization of system performance and system design. Artificial neural networks are able to efficiently accomplish this simulation without extensive identification of physical parameters such as cell impedance or diffusion characteristics that are required in parametric models. There are a number of ANN architectures available to facilitate our electrochemical simulation. The structure, rules of operation, and environment for general ANNs have been discussed in detail, for example, in Rumelhart and McClelland [1] and Vanluchene and Sun [2]. Although there are many types of ANNs to accomplish our task, the two used to model battery behavior are the connectionist normalized linear spline (CNLS) network and the feedforward back propagation network (BPN).

CNLS NETWORK

The CNLS network is an extension of the generalized radial basis function neural network (Moody and Darken, [3]), and is described in detail in Jones, et. al., [4]. The CNLS net is designed to approximate a functional mapping by superimposing the effects of basis functions that approximate the mapping in local regions of the input space. Because the CNLS net is a local approximation network, it is inefficient for this network to be used in applications involving systems with high dimensional input spaces.

The CNLS net is motivated by the following identity. Let $z = g(\{x\})$ denote a mapping from the input vector $\{x\}$ to the output (scalar, in this case) z. Let us define the sequence of radial basis functions $u(\{x\},[\beta],\{c_j\})$, j=1,...,N, (the multivariate normal density functions to be specified later) as scalar functions of the vector $\{x\}$, with parameters $\{c_j\}$, j=1,...,N, and $[\beta]$. The functions $u(\{x\},[\beta],\{c_j\})$, j=1,...,N, are nonzero for finite ||x||, and will be described later. The following is an identity:

$$g(\{x\})\sum_{j=1}^{N} u(\{x\},[\beta],\{c_j\}) \equiv \sum_{j=1}^{N} g(\{x\})\, u(\{x\},[\beta],\{c_j\}) \tag{1}$$

This can be written because $g(\{x\})$ is not a function of the index j. Now approximate $g(\{x\})$ in the sum on the right hand side with the first two terms in its Taylor series, expanded about $\{c_j\}$. This is

$$g(\{x\}) \cong g_j + \{d_j\}^T (\{x\} - \{c_j\}), \tag{2}$$

where $g_j=g(\{c_j\})$ is a scalar constant, and the i^{th} element of the vector $\{d_j\}$ is $\partial g/\partial x_i$ evaluated at $\{x\}=\{c_j\}$. The g_j and $\{d_j\}$ are the constant and linear terms, respectively, of the Taylor series approximation. Move the sum on the left hand side of Eq. (1) to the right hand side, and use Eq. (2) in Eq. (1). The result is the approximation

$$z \cong y = h(\{x\}) = \frac{\sum_{j=1}^{N}\left[g_j + \{d_j\}^T (\{x\} - \{c_j\})\right] u(\{x\},[\beta],\{c_j\})}{\sum_{j=1}^{N} u(\{x\},[\beta],\{c_j\})} \tag{3}$$

The operations on the right hand side, above, express the framework for the CNLS net approximation to $z=g(\{x\})$. The scalar y is the output of the CNLS net, and we seek parameters for the network that will make y a good approximation to z, the output of the actual system. We now define the form of the radial basis function. It is typically taken to have the form of the multivariate normal probability density function, namely,

$$u(\{x\},[\beta],\{c_j\}) = \exp\left(-(\{x\} - \{c_j\})^T [\beta](\{x\} - \{c_j\})\right) \tag{4}$$

This is a "bell-shaped curve" defined on the n-dimensional real space. The center of the curve is defined by $\{c_j\}$, and the "width" of the curve in each of the directions of the elements of $\{x\}$ is an element in the diagonal matrix $[\beta]$. In practical applications, the parameters of the radial basis functions, $[\beta]$ and $\{c_j\}$, and the linear parameters, g_j and $\{d_j\}$, can be adjusted to make the surface approximate any desired shape. Clearly, the number of radial basis functions used in the approximation affects the accuracy that is achievable in the approximation.

Training of the CNLS net involves the adjustment of all the parameters to optimize an input/output approximation. This training is accomplished using an approach that takes advantage of the combined

linear plus nonlinear dependence on the parameters. Training of any network utilizes training exemplars (examples of correct behavior for the system being modeled) to optimize the simulation of the system. The CNLS net is trained by presenting exemplars to the net, one set at a time. To start, the centers $\{c_j\}$ are chosen from among the input exemplars, $\{x_j\}$, j=1,...,R and distributed among the input space using a self-organizing map to best represent the input exemplars. Next, the β values are modified based on a quadratic approximation of the error surface, that minimizes the error with respect to the center locations. Finally, a series of training loops is executed to modify the linear terms using the least mean square (LMS) algorithm, described, for example, in Widrow and Stearns [5]. The training modifies the linear terms to minimize the differences between the CNLS net output and the output exemplar z_j.

BACKPROPAGATION NETWORK

The BPN is the most widely used ANN and it is described in detail in many texts and papers, for example Freeman and Skapura [6], and Haykin [7]. The BPN is very general in the sense that it can approximate mappings over a wide range of input dimension. It has been shown that, given sufficient training data, a BPN with at least one hidden layer and sufficient neurons can approximate a mapping to arbitrary accuracy (Hornik, Stinchcombe, and White [8]). The BPN is a multilayered perceptron network that uses a special type of training algorithm. Figure 2 shows a schematic of the BPN with an input layer, arbitrary number of hidden layers, and an output layer. In this case, all the output signals produced in a particular layer are distributed to every neuron in the following layer causing the network to be "fully connected", which is usually the way BPNs are implemented.

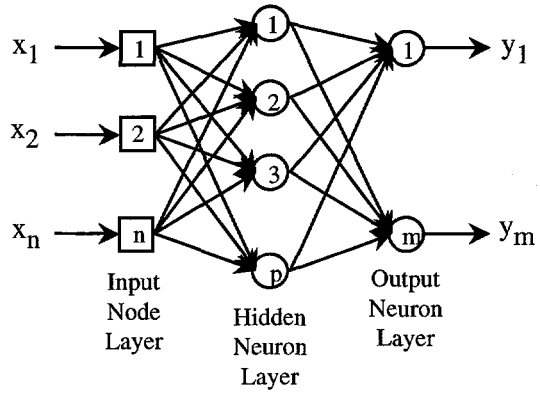

Figure 2. Diagram of a multi-layered perceptron (Backpropagation) network.

The operations performed in each neuron of a BPN are shown mathematically below in Eq. (5). Here, y_k represents the output of neuron k, while y_j and w_{kj} represent the output from neuron j in the previous layer and the corresponding weighting factor, respectively. Note, if y_k represents the output of a neuron in the input layer, all y_j values would be replaced by inputs, x_j. The quantity θ_k is a bias or threshold to allow for offset in the neuron activation. The summation of the neuron inputs and the threshold are operated upon by a neuron activation function, f, which yields the neuron output.

$$y_k = f\left[\sum_{j=1}^{N}(y_j)(w_{kj}) - \theta_k\right]. \tag{5}$$

Note the summation portion of the equation resembles an adaptive filter, as described in Widrow and Stearns [5]. Several types of functions can be used for the activation function. Some examples include linear functions, step functions, and sigmoidal functions. Sigmoidal functions are useful in continuous input/continuous output applications. A sigmoidal function was used in this study and the mathematical form of the function is:

$$f(x) = \frac{1}{1+e^{-x}}, \quad -\infty < x < \infty. \tag{6}$$

When an input is presented to the BPN, it can be fed forward through the entire network to obtain the output. This output can be compared to the output exemplar, and the parameters of the ANN can be adjusted to diminish the difference between the ANN output and the training output. As implied by its name, the training of the BPN involves "back-propagating" the error from the output layer to the input layer. A common training procedure known as the generalized delta rule uses the back-propagated error to optimize the weights of the network. For more detail on the training of BPNs, see Freeman and Skapura [6] and Rumelhart, McClelland, et.al., [1].

LiSOCl2 CELL SIMULATIONS

As mentioned previously, many attempts have been made in different frameworks to model electrochemical systems. Because these systems are so complex, the ability to develop an accurate parametric model that can be used for both optimization of performance and system design has been limited; therefore, time-consuming build and test procedures remain the norm. Therefore, using ANNs to efficiently simulate and predict the behavior of these systems could prove to be extremely cost and time effective while improving system performance. The electrochemical system used for these studies of ANN modeling is the lithium/thionyl chloride battery. The challenge in this investigation is to use relatively simple but specialized experimental results to build a model that accurately and efficiently reflects real battery behavior under complex use profiles. Simulations of this battery system progressed from simple constant load discharges to more complex and practical simulations including temperature variation and load pulses. Results of a sequence of model developments are presented in the following discussion.

Initial feasibility studies began by simply trying to replicate a discharge curve of the lithium/thionyl chloride battery with the CNLS network. Figure 3 shows a simulation of the battery voltage under a constant discharge load of 50 ohms at 25°C. As can be seen in the figure, the simulation approximates the experimental data very well. Because one of the major interests in this feasibility study is to determine if ANNs are able to predict the amount of capacity delivered by a battery, it is important that the nets be able to simulate the discharge curve near the "knee", i.e., where the voltage begins to drop. In this case, that is near 200 hours of discharge. Although the exemplars used to train the CNLS network are largely concentrated around 3.5 volts (0 hours - 200 hours), the network was still able to capture the dynamics of the discharge at the knee. This is due to the local approximation capability of the radial basis functions used in the CNLS network.

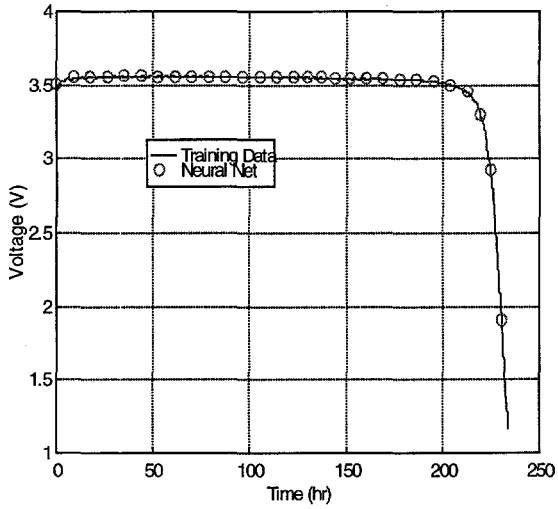

Figure 3. CNLS simulation of battery voltage data (50 ohm discharge load, 25°C)

Our next step in increasing the complexity of ANN simulations was to model a system with different loads. The results are shown in Figure 4. The results indicate that the CNLS network can accurately simulate battery discharge curves with this additional dimension to the input space. Again, the accuracy with which this network predicts the discharge voltage is largely due to the local approximation capability of the CNLS network.

The CNLS network uses a self-organizing layer on input exemplars to determine the center locations for its radial basis functions. This contributes to the accuracy for local approximations of the network. The advantages of the local approximation, however, also imply limitations on the curve-to-curve interpolative capabilities of the CNLS net, which are required in this application. Therefore, the BPN was used in the remainder of the battery simulation applications.

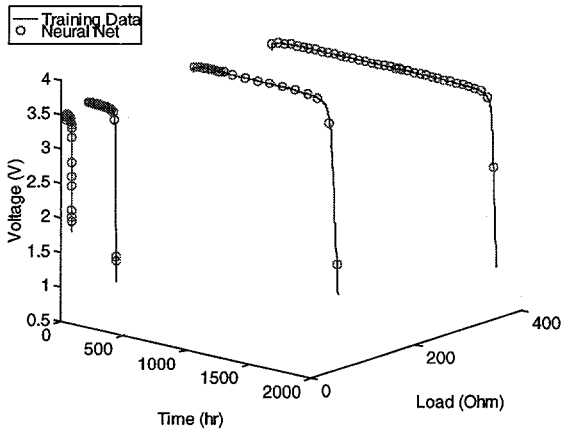

Figure 4. Non-recurrent CNLS simulation with load as an input (T=25°C).

The circles in Figure 5 show the simulation of a constant load discharge at 300 ohms. The BPN was trained with 25°C constant load discharge data at 50, 250, and 400 ohms (solid unmarked curves) and an interpolated time series at 300 ohms is presented. The BPN appears to provide very good interpolation characteristics for loads between 50 and 400 ohms.

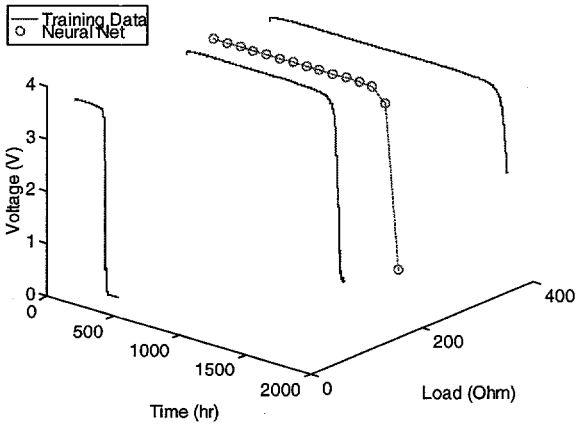

Figure 5. BPN interpolation of a constant load discharge at 300 ohms (T=25°C).

Our next objective was to extend our predictions to pulsed loads and variable temperature. As little pulsed data were available at this point, isothermal and constant load discharge data were used to train the BPN in order to determine whether their use could be extended to pulsed load or pulsed temperature conditions. Specifically, the constant load discharge data available for training are outlined in Table I.

Table I. Constant load, constant temperature data available for ANN simulations.

Load (ohms)	Temperatures (°C)
50	-40, 25, 49
250	-40, 25, 49
400	-40, 25, 49

The hope is that these data could be used to train a BPN which could be used to simulate pulsed loads between 50 and 400 ohms, and temperature variation between -40°C and 49°C, or a combination of both. A load pulse could be simulated if the rule for an isothermal transition from one constant load discharge curve to another could be determined. Figure 6 shows a simulation developed by training the BPN with

constant load discharge data at 50 and 250 ohms at 25°C. Once trained, the network was used to simulate a pulsed load condition between these two loads with 3-Amp hour capacity dwells at each load. Note the independent temporal variable has been converted from time to current capacity used; this introduced a more convenient normalization of the temporal variable. The pulsed load simulation was performed in an attempt to identify how the deliverable capacity might vary depending on the pulsed load conditions. Because the measured capacity of these battery cells is known to change with load, a critical question is how to map the battery capacity "state" from one load to the other during a pulse. Specifically, the transition could be based on the "fraction of capacity delivered", the "absolute capacity used", or some other rule. One might expect the fraction of capacity delivered to yield a more accurate a better transition between the two loads, but in fact absolute capacity used seemed to perform a better mapping in this case. Therefore, this is the transition rule used in the sequel. Further investigation of this mapping rule needs to be done to determine the best forms of transition for all situations.

As mentioned above, the simulation shown in Figure 6 is based on the network being trained with constant load discharge data. The experimental pulse displayed in Figure 6 is based on data that were gathered later following the ANN simulations. A slight difference is noticeable between the experimental voltage curve and the simulated curve at 50 ohms. The constant load discharge data at 50 ohms that were used to train the BPN had a slightly higher voltage level than the actual pulsed data at that load, which is reflected in this simulation. Nevertheless, the ANN simulation is clearly predictive.

The simulated pulse also seems to approach failure (voltage < 2V) slightly before the experimental pulse, causing a more rapid drop-off at the knee. Currently, other batteries are being tested with varied pulsed load profiles in attempts to fully understand the transition between load curves during pulse discharge and to determine if observed differences between actual and predicted behavior are a result of cell-to-cell variability. To develop rules for transition between temperatures, tests are being run on batteries in which only the temperature is varied under constant load. In this case, it may be more likely that the "fraction of capacity delivered" rule will be followed for transition as there are greater fluctuations in measured capacity when moving from one temperature to another (especially to low temperatures) than occur over the above range of loads.

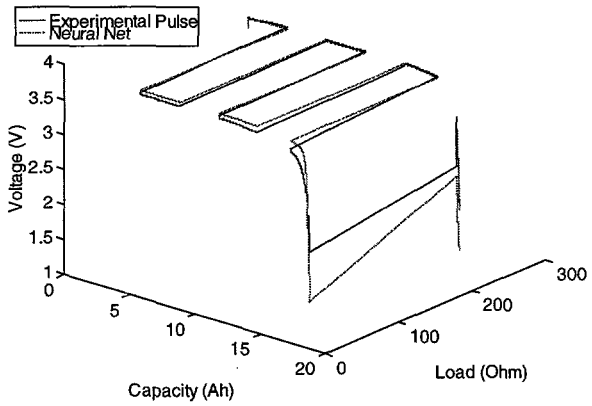

Figure 6. Isothermal pulsed load simulation between 50 and 250 ohms at 25°C.

Once further tests are performed and the separate rules for transition between loads and temperatures are more completely understood, we intend to simulate a profile in which both the load and temperature are pulsed. We currently have experimental data taken under this type of condition. Specifically, the temperature cycle consists of a cold, an ambient, and a hot level while the load is pulsed between a very low current background and a moderate on-pulse level. Constant load discharge curves are still being developed for the load range needed to correspond to this test. Because constant load discharge data were available at three different temperatures, an arbitrary simulation was done to predict the battery voltage using a BPN network trained with capacity, load, and temperature inputs. Figure 7 shows this voltage simulation during load and temperature pulses (the three solid lines at each load represent the three temperatures used in training).

The important characteristics to note in this pulse profile are the ability of the network to predict the change in voltage from one temperature to another and that more capacity can be delivered when moving

from a low temperature to a high temperature. An example of the network predicting a reduction in voltage moving from a higher to a lower temperature is shown below near 4 Amp hours of capacity removed. Here, the network predicts a reduction of about 0.5 volts as the temperature is changed from 25°C to -40°C. Near 12 Amp hours removed, the simulation suggests no voltage output from the cell as the temperature is lowered from 25°C to -40°C, but shows capacity still exists when the temperature is returned to ambient. These types of responses shown in the simulation are clearly representative of the battery behavior seen in laboratory experiments and further analysis with ANNs will increase our understanding of discharge behavior under both constant and pulsed conditions.

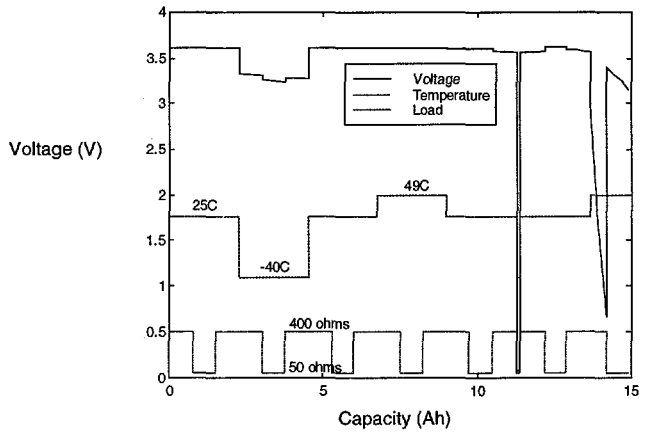

Figure 7. ANN simulation of arbitrary load and temperature pulse.

As mentioned above, using ANNs to model battery systems offer a variety of advantages over phenomenological models. Computational efficiency and focus on overall cell performance rather than specific processes are among the most significant. Analytical models, however, are able to provide information regarding "local dynamics" that are more difficult for ANNs to capture when being trained over a large input space. An analytical model has been developed for the Li/SOCl2 cell, but is computationally very intensive and can take several hours to run. In an attempt to take advantage of the strengths of both analytical and ANN models, preliminary steps have been taken to develop a "hybrid" model. Figure 8 details the results of our preliminary study using both the analytical and ANN models to simulate the lithium/thionyl chloride battery system.

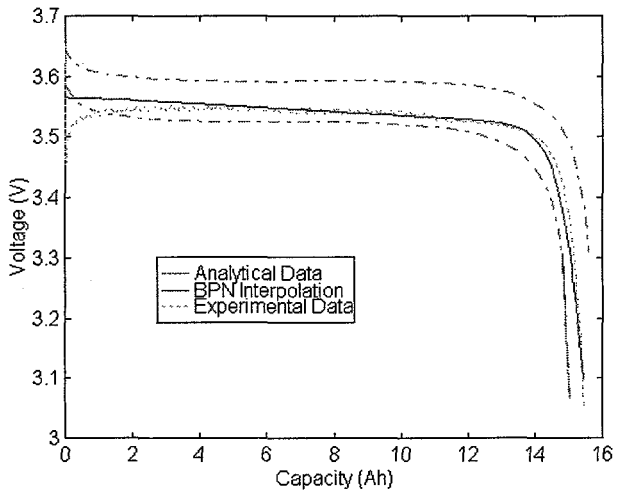

Figure 8. ANN comparison to experimental data when trained with analytical data (T=25°C).

A neural network was used to model cell impedance for input to the analytical model, because developing a functional relationship for this parameter would be very difficult. Using the ANN model of the cell impedance, the analytical model was used to generate to constant load discharge curves at 30 and 80 ohms (bounding discharge curves in Figure 8). These analytical data were used to train the BPN within the 30-80 ohm input space at 25°C. The trained BPN was then used to simulate a discharge at 50 ohms

(solid line between bounding curves). An experimental curve at 50 ohms has been plotted in Figure 8 (dotted line between bounding curves) showing the accuracy to which the hybrid model was able to simulate the discharge of a Li/SOCl2 cell. Based on the results of this preliminary test, efforts are being made to further test the applicability of this, and other, hybrid models in their application to both optimization of performance and system design.

MODELING UNCERTAINTY

There are two classes of uncertainty issues that affect the specification and use of ANNs. First, because the ANN is a non-phenomenological model of system behavior, the map that is learned by an ANN cannot precisely replicate the map that is the source of its input/output exemplars. That is, the ANN is an uncertain representation of the source map. The training techniques used to identify the parameters of ANNs are designed to minimize the error with this origin. Beyond this, it is practically always a fact that the input/output exemplars presented to an ANN during training contain measurement noise. This precludes the possibility that the exemplars exactly represent system behavior. This problem is mitigated by the fact that training procedures for ANNs typically yield models that average through the measurement noise yielding an average model of system input/output behavior.

There is a second uncertainty issue regarding the use of ANNs. Under certain circumstances the inputs to an ANN may be random variables or random processes. In this case, the inputs map to random outputs, as they would with any deterministic map. The ANN can be used in the same way that a phenomenological model is used to establish the probability distribution of one or more random outputs given information on random inputs. In fact, because of its relative accuracy and computational efficiency, ANNs are sometimes used as substitutes for phenomenological models where numerous model runs are required.

CONCLUSIONS

Initial modeling efforts of electrochemical battery systems with artificial neural networks have proven to be very successful. Our feasibility study for this form of non-phenomenological modeling began by looking at single load constant discharge data. The CNLS network was able to accurately simulate these discharge curves and the local approximation capability of this network allowed excellent representation of the area surrounding the "knee" of the discharge curve. Although the CNLS network performed very well in the input space where training exemplars were located, the interpolation characteristics between load values were not expected to be as good. Efforts to improve the interpolation capabilities of the CNLS net are currently under way.

The parameters of the BPN do not depend in the same way on the training exemplars like the centers of the radial basis functions do in the CNLS network. Therefore, the BPN was used to investigate interpolation characteristics of battery data. Specifically, a 300 ohm constant load discharge simulation was performed with the BPN and the results compared very well to what is anticipated in the laboratory. The BPN has also been used to simulate pulsed load conditions at constant temperature. The BPN was trained with constant load discharge data as no pulsed data were available. A pulsed load experiment matching the profile of the simulation has now been completed and the ANN prediction compares well to these experimental results. The ability of the BPN network to simulate battery pulsed discharge data when trained with the constant load exemplars is a significant finding.

Current efforts involve completing a similar experiment with variable temperature under constant load conditions. Once this information is available, simulations in which both the temperature and load are arbitrarily changed can be performed. With additional experimental pulse data being generated, ANN architecture and training can be optimized to further increase the accuracy of the battery simulations. This will involve ANN training using experimental battery data where temperature and load are varied simultaneously. It is not clear that any simple rule or combination of rules will suffice to generate accurate ANN simulations of real battery behavior. Additional tools like genetic algorithms and/or genetic programming may be used to establish more accurate transition rules.

These initial efforts on battery modeling have proven to be very effective, and more complex simulations of battery behavior will be performed. With advanced study of ANN modeling, and further development of the parametric model, additional simulations can be performed using the hybrid model to help efficiently design and optimize robust battery systems.

ACKNOWLEDGEMENTS

Sandia National Laboratories is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

REFERENCES

[1] Rumelhart, D. E., McClelland, J. L., et.al., (1986), *Parallel Distributed Processing, Explorations in the Microstructure of Cognition, Volume 1: Foundations*, MIT Press, Cambridge, Massachusetts.

[2] Vanluchene, R. D., Sun, R., (1990), "Neural Networks in Structural Engineering," *Microcomputers in Civil Engineering*, 5, n. 3, 207-215.

[3] Moody, J., Darken, C., (1989), "Fast Learning Networks of Locally-Tuned Processing Units," *Neural Computation*, V. 1, 281-294.

[4] Jones, R. D., et. al., (1990), "Nonlinear Adaptive Networks: A Little Theory, A Few Applications," *Cognitive Modeling in System Control*, The Santa Fe Institute.

[5] Widrow, B., Stearns, S. D., (1985), *Adaptive Signal Processing*, Prentice-Hall, Upper Saddle River, New Jersey.

[6] Freeman, J. A., Skapura, D. M., (1991), *Neural Networks, Algorithms, Applications, and Programming Techniques*, Addison-Wesley Publishing Company, Reading Massachusetts.

[7] Haykin, S., (1994), *Neural Networks, A Comprehensive Foundation*, Prentice Hall, Upper Saddle River, New Jersey.

[8] Hornik, K., Stinchcombe, M., White, H., (1989), "Multilayer Feedforward Networks are Universal Approximators," *Neural Networks*, V. 2, 359-366.