Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Power Sources 145 (2005) 383–391
JOURNAL OF POWER SOURCES
www.elsevier.com/locate/jpowsour

Rapid (practical) methodology for creation of fuel cell systems models with scalable complexity

C. Nitsche [a,*], S. Schroedl [b], W. Weiss [a], E. Pucher [c]

[a] *DaimlerChrysler Research and Technology at the California Fuel Cell Partnership, 3300 Industrial Blvd. Suite 800, West Sacramento, CA 95691, USA*
[b] *DaimlerChrysler Research and Technology, 1510 Page Mill Road, Palo Alto, CA 94304, USA*
[c] *Vienna University of Technology, Karlsplatz 13 A-1040, Vienna, Austria*

Accepted 1 February 2005
Available online 31 May 2005

Abstract

In order to study various aspects of fuel cell systems, like a fuel cell propulsion system for transportation, several challenges arise: in actual real-world operation, as opposed to benchmark tests, the system is subject to a variety of non-stationary and environmental nuisance factors that are hard to monitor and control; investigating the system's behavior at the limits of its ranges while avoiding any adverse effects; due to sensor capabilities and costs, not every relevant variable can be monitored with sufficiently high temporal resolution.

For these reasons, simulation tools are playing a crucial role in the analysis of these system aspects. The first step is therefore to create a mathematical representation of the system (a model) which can then be embedded into a simulation environment. To this end, a methodology is needed for the rapid creation of the mathematical representation of a system which is capable of overcoming the hurdles of dynamic and transient variables.

Usually, knowledge-based modeling a system this complex takes several years to accomplish and still does not take nuisance factors into account. In contrast, the approach presented here can be finished within a fraction of that time. We propose to employ black-box adaptive modeling; the key issue in here, selecting an appropriate set of input features, can be solved by either applying iterative wrapper methods, or by making use of the automatic relevance detection technique that has been developed earlier within the framework of Bayesian neural networks. These procedures allow to easily scale the complexity of models in order to accommodate different constraints in terms of modeling effort, sensor availability and cost, and required model accuracy. Our approach can as well be used for the development of diagnostic models for on- and off-board diagnostics.

Keywords: Fuel cell powertrain; Fuel cell modeling; Feature selection; Automatic relevance detection; Artificial neural networks

1. Introduction

The fuel cell vehicles industry is now approaching the transition from a vehicle prototype stage to commercialization. Therefore, the modeling efforts can now be based on data recorded from existing vehicles. These data contain valuable information for a post analysis of driving operations, and allows for the accumulation of knowledge about these systems to be used in the development of the next generation of fuel cell vehicles.

Several fuel cell vehicle manufacturers are currently approaching the market with small fuel cell vehicle fleets. One of the purposes of these fleets is to give feedback from real world operations indicating vehicle performance and component lifetime.

The modeling method presented here will enable engineers to address many of the real world interests arising from the deployment of these vehicle fleets. Perhaps the first aim is to monitor the degradation of these systems over the course

* Corresponding author. Tel.: +1 916 375 0377; fax: +1 916 375 0378.
E-mail address: cn74@daimlerchrysler.com (C. Nitsche).

0378-7753/$ – see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.jpowsour.2005.02.055

of their lifespan. Secondly, any unexpected influences on fuel cell performance and condition, like air pollution or extreme climate condition, can be investigated. Additionally, it can be closely monitored if the powertrain's operating strategies are well tuned, or if they need adjustment to internal and external influences.

In order to do such an analysis, rapid modeling concepts need to be employed that provide powerful tools for the analysis of the huge amounts of time series data recorded while driving the vehicles. Therefore, our method was developed, within the MATLAB software environment, to cover the demands for the analysis of fuel cell powertrain data.

2. Modeling approach

There are several ways to go about creating a mathematical representation of a physical system. The classical way is to find mathematical formulas to describe every relevant aspect of the system. The advantage of this method is that it offers a deep insight into a system by providing physical and causal relationships. Secondly, if the physical system has yet not been built, it is the only possible way, since the black-box adaptive modeling described below cannot be applied without empirical measurements. However, this knowledge-based approach is very labor-intensive; even if a detailed formalization has been found, it is hard to optimize the generally large number of parameters; and it is difficult to incorporate external nuisance factors into it.

In contrast to this knowledge-intensive approach, we can restrict our attention to the relation between input and output variables of individual system components, or the entire system itself. In principle, any representation formalism for multi-dimensional functions, such as B-splines, classification and regression trees, dynamic Bayes networks (including Hidden Markov models as a special case), or artificial neural networks (ANNs) can determine these relationships. When empirical data is available, this 'black-box' modeling approach can be implemented with much less time and effort than an explicit model; while it can accurately simulate and predict the system behavior, its drawback lies in the lack of causal interpretation available to a human domain expert.

The modeling method described here belongs to the latter class. Its advantage lies in its automated *rapid model creation* with a *scalable complexity*. The model complexity can be adjusted according to its purpose. For example, if the model is designed for an onboard diagnostics device, then it needs to be very accurate (to be able to compare actual to simulated signals), compact (to enable an implementation in a limited CPU-performance computer platform) and sufficiently fast (in order to run as a real-time system). On the other hand, if the model is intended for a simulation environment on a fast processing computer, then it can be scaled up to using several hundreds of input signals as well as output signals.

3. Application to fuel cell vehicles

A fuel cell engine (Fig. 1) [1,2] consists, in our case, of a PEM fuel cell stack, an air feed using an air compressor, a hydrogen feed from the hydrogen tanks and an electric motor which uses the electricity generated by the fuel cell (by combining hydrogen and oxygen from the air) to propel the vehicle.

The main physical variables to run a fuel cell powertrain are the air and hydrogen flow through the stack, the temperatures, pressures and humidity of these gases, the output current and voltage of the stack and the temperature of the medium in the stack cooling loop.

In a state of the art fuel cell vehicle, the number of relevant signals and parameters easily reaches a count of several hundreds. Our method discussed here is a powerful tool that it is capable of filtering through these hundreds of signals; extracting and analyzing only the signals pertinent to the desired model and output.

We used this approach to model several physical signals like the output voltage of the fuel cell stack to see how various signals influence it in a dynamic and steady-state manner. Since the fuel cell powertrain operation is highly dynamic, the models of the powertrain have to account for this transient behavior.

Another hurdle for the creation of accurate models is the fact that these vehicles are not operated under predefined load cycles and constant environmental conditions, but are driven on the road under varying conditions. This fact is a big obstacle for 'classical' modeling but can be covered by our approach.

The time series data we use for our analysis gets recorded from the vehicles controller area network (CAN)-Bus communication network. The CAN-Bus itself gets fed with several signals by the electronic control unit (ECU) which gets the signals from the various sensors the vehicle is equipped with.

The importance of time dependency is obvious in any physical system, and in particular for a fuel cell system which is supplied by an air compressor and other components that have a measurable response time.

In order to model temporal behavior, one natural candidate for model representation would be Dynamic Bayes Networks [3]. However, in existing software packages the

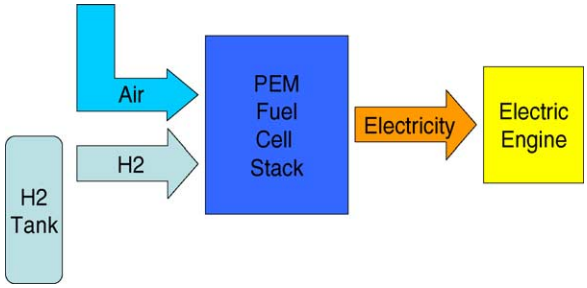

Fig. 1. Block diagram of fuel cell engine.

ways for expressing non-linear functions are rather restricted to keep computability feasible and fast. Hence, we opted for using artificial neural networks (ANNs) [4]. Time dependency is realized by the possibility of feeding it not only signal values observed at the current sampling interval, but also those from a fixed size, small historical window (maximum time delay between cause and effect). These past signals appear to the ANN just the same as the current ones, i.e., time dependency is incorporated in a flat, implicit way.

To account for the time response characteristics, virtual signals are created by shifting all input signals to the right on the timescale by x times its sampling interval, where x is 1, 2, 3, 4, 5, 10, 15, 20. Based on a sampling rate of 10 Hz of the time series, this range allows the ANN to see input signals up to 2 s before the current time stamp.

The time series data is usually recorded while driving the fuel cell vehicles on the road or while conducting defined drive cycles on a dynamometer. In general, it makes sense that this time series covers the entire dynamic and power range spectrum. For alternate analyses, this series can also utilize data from a stationary fuel cell system.

4. Modeling method using automatic relevance detection or wrapper method

In this section, we discuss a method to filter out, from a (possibly very long) list of possible input features, those ones that are most relevant for predicting the output behavior of the system. This step, which is usually called the *Feature Selection problem* in the machine learning literature [5], is a key ingredient of system identification [6]. While using more independent input signals can provide valuable information to make the prediction more accurate, using the wrong ones can also confuse a modeling algorithm and affect the model performance.

Our method consists of several parts. It is fully automated (in the MATLAB software environment) and needs the user only to specify the available input signals, the desired output signals, the maximum delay time from changes in the input values to changes in the output values, and optionally a desired model accuracy, i.e., an acceptable error threshold at which the model identification process can be terminated.

Our method utilizes either the 'automatic relevance detection' (ARD) [3] method or the 'Greedy Wrapper' method [5] to perform this relevance analysis. The result of this analysis is a relevance table, which is a list of a subset of the input features that are relevant for the prediction task at hand, in a model scaled to particular complexity constraints.

4.1. Automatic relevance detection (ARD)

In the traditional view, an artificial neural network is a non-linear mapping from an input x to an output y, depending on a set of parameters w. These networks can be trained using pairs (x,t) of input and target outputs; the error E_D is measured as the squared difference between t and the actual output of the net (summed over all training examples). By increasing the complexity of the network, we can easily get arbitrarily close to the training data; however, we are also fitting the random errors in this particular selection of examples, so that the predictive capabilities of the ANN for new instances can actually degrade. Therefore, it is common to include a regularizer in the error function, such as weight decay, to penalize model complexity. The combined objective function to be minimized becomes $M(w) = \alpha E_W + \beta E_D$, for appropriately chosen hyperparameters α and β, where E_W is the sum of squares of all weights w.

As an alternative to this conventional setting, MacKay [3] developed the framework of *Bayesian Neural Networks*, which can be exploited in our context. Instead of focusing on a single best-fit ANN, there is a continuum of models in the weight space, and each possible network has an associated probability composed of it's a priori probability, and the likelihood that the observed training data would have been produced by it, given Gaussian noise on the output. In fact, if we assume a prior distribution of the weights that is Gaussian with mean zero, the log likelihood of a parameter vector w is proportional to $M(w)$. this interpretation, α turns out to be the inverse of the variance of the weight prior, and β the variance of the noise. By making the simplifying assumption that the distribution of the hyperparameters, like that of the weights, is sharply peaked around a single maximum, their most probable value can be estimated from the data. For more complex ANNs, we do not have to suppose that the prior distribution is identical for all weights; we can allow different variances for different classes of them, even to each individual one. In this generalization, we can estimate a separate α for each weight. A low value means a high standard deviation, i.e., the choice of the weight influences the output of the net less significantly. In a conventional ANN, due to random correlations irrelevant input features will still have non-zero weight, and will hurt the network's performance. Automatic relevance detection, on the other hand, interleaves weight optimization and hyperparameter estimation; therefore, it is more robust to spurious features, their α-value will be decreased so as to softly switch them off.

Our method starts the ARD (Fig. 2) once to determine the α-parameter for each input feature. This result is used to statically sort the features. The algorithm then iteratively adds features from this ordered list and retrains an ANN; the termination criterion can be determined to scale the resulting model complexity. For example, the user can provide an error threshold, below which the model is deemed acceptable. The error is determined by *10-fold cross-validation*: i.e., out of the time series data 90% of it is always used for training the system and the other 10% is used to feed the ANN in order to validate its predictability. This is done until all 10% portions

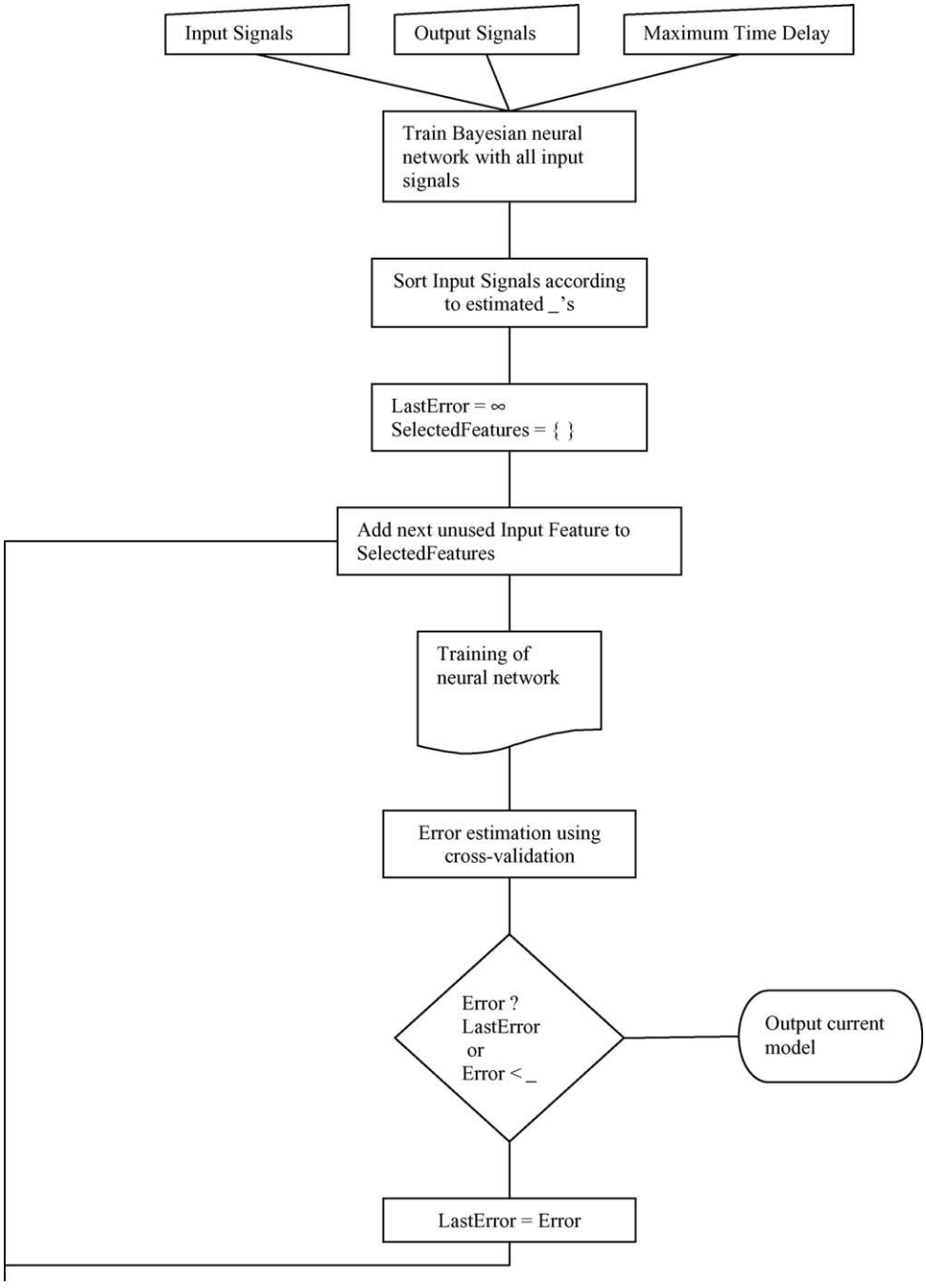

Fig. 2. Feature selection using automatic relevance detection.

of the time series data is used for validation. Then a mean error for these runs is calculated.

4.2. Greedy Wrapper search

An alternative to automatic relevance detection is to use a Greedy Wrapper method (Fig. 3) to determine the best suitable input signals for the desired output signal. The algorithm starts with an empty list of input features. Then, in each iteration a number of modification operators are applied in an attempt to improve the ANN's performance. Possible modification operators include adding or discarding an input feature, or changing the number of hidden neurons. The method evaluates each individual modified neural net using cross validation and chooses the one with minimum error. The process continues until no operator can achieve any improvement over the current best model. Alternatively, and similarly as in the ARD algorithm, we can also terminate if the error falls below a user-supplied error threshold.

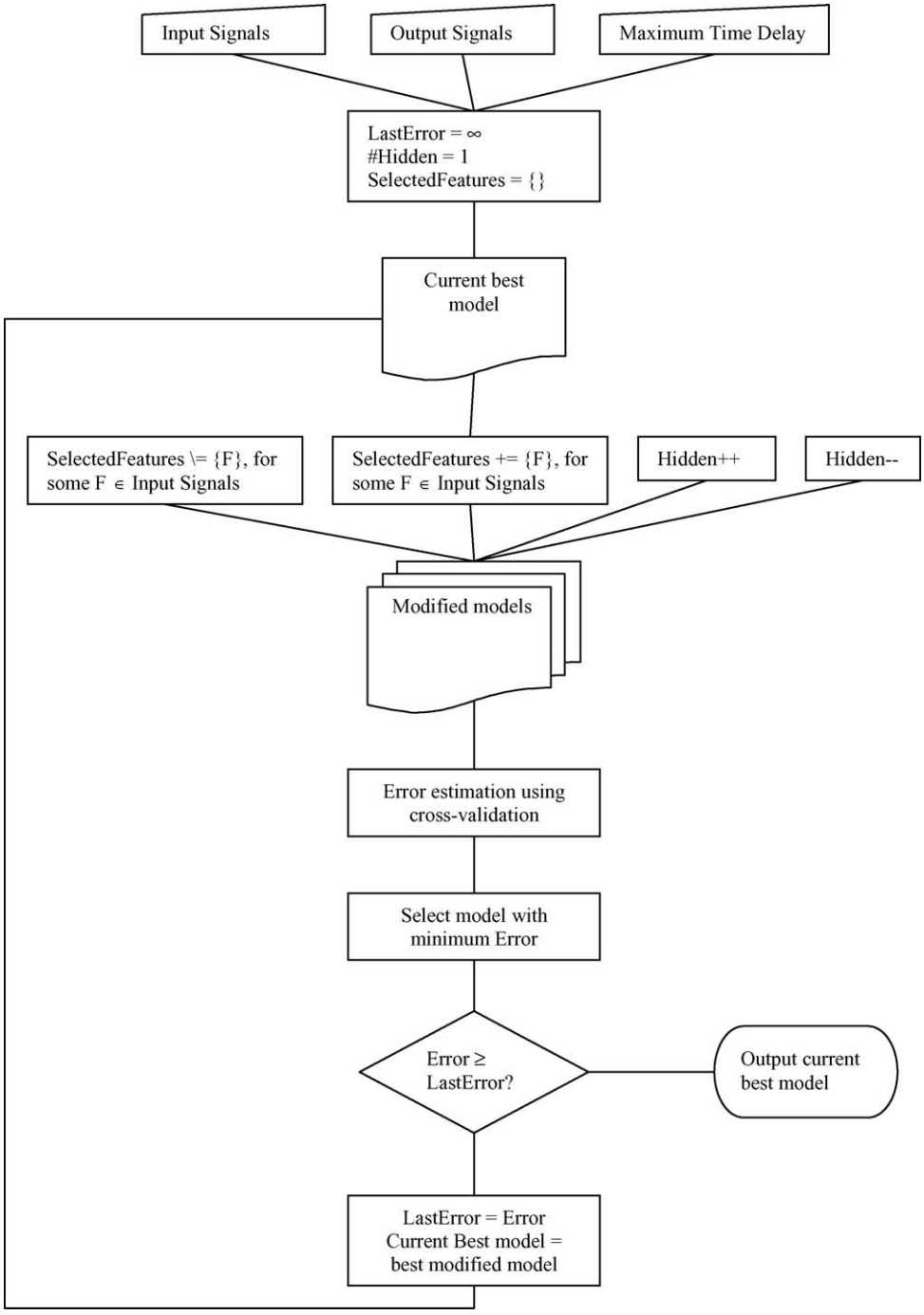

Fig. 3. Feature selection using Greedy Wrapper approach.

5. Creation of fuel cell system models

5.1. Results from the Greedy Wrapper and ARD methods

We now want to look deeper into the details of using this method. For example, we want to study how the various operating parameters and the amount of current drawn from the fuel cell stack influence the fuel cell stack voltage. The goal is therefore to create an accurate model of the fuel cell stack.

We define the flow, temperature and pressure of the air flowing into the stack, the temperature and pressure of the hydrogen gas inside the stack, the temperature of the stack cooling medium and the electrical current which is drawn from the stack as the input signals. The fuel cell stack voltage gets assigned as the output signal (Table 1).

Table 2 shows the error improvement table resulting from the Greedy Wrapper search method (based on data recorded while driving a standard drive cycle), by utilizing the input

Table 1
Input and output signals for fuel cell system model

Signal	Unit
Input	
Electrical current: current drawn from fuel cell stack	A
Pressures: air pressure stack inlet, (cathode), hydrogen pressure stack inlet (anode)	bar
Temperatures: cooling water stack outlet, air temperature stack outlet, hydrogen temperature stack inlet	K
Flow: air flow through stack	$kg\,h^{-1}$
Output	
Voltage: voltage of fuel cell stack at electrical outlet	V

and output signals from Table 1 including the time-shifted virtual input signals.

The input signals are sorted in the order of the degree of their influence for the smallest possible average error from the top of the table to the bottom. The 'Average Error' column describes the average error (in volts) of the simulated to the actual stack voltage (using 10-fold cross-validation). The average error is determined by building a model using only the most important signal from this list, then adding the next important one, and so on. As can be seen from this table, by adding more and more input signals the error keeps decreasing.

Table 3 shows the relevance table resulting from the automatic relevance detection method by utilizing the same data as well as the same input and output signals as for the Greedy Wrapper method.

The average error was determined by adding one signal at a time to the modeling procedure in the order of increasing relevance factors (the lower the relevance factor, the higher its signal's relevance). As can be seen clearly from Tables 2 and 3, both methods lead to about the same optimum average error (model accuracy). The difference is that the Greedy Wrapper method offers a selection of input signals that leads to the smallest error. On the other hand, the ARD method gives a relevance factor for each input signal. Therefore by adding more and more signals down the list from Table 3 does not necessarily lead to an ever decreasing error.

Table 2
Improvement of error using Greedy Wrapper method

Average error (volt)	Signal description
7.26	Fuel cell stack current
1.90	Hydrogen pressure fuel cell stack (anode)
1.18	Fuel cell stack current ($t-0.1$ s)
1.08	Airflow trough fuel cell stack ($t-0.1$ s)
1.05	Hydrogen pressure fuel cell stack (anode) ($t-0.4$ s)
1.00	Air pressure fuel cell stack (cathode)
0.96	Air temperature stack outlet
0.91	Cooling water temperature fuel cell stack outlet ($t-0.4$ s)
0.91	Air pressure fuel cell stack (cathode) ($t-0.4$ s)
0.87	Fuel cell stack current ($t-0.4$ s)
0.83	Cooling water temperature fuel cell stack outlet ($t-0.1$ s)

Table 3
Relevance table by ARD method

Average error (volt)	Relevance factor (α)	Signal description
7.89	1.33	Fuel cell stack current
7.32	1.34	Fuel cell stack current ($t-0.1$ s)
7.12	2.86	Fuel cell stack current ($t-0.4$ s)
3.11	7.40	Airflow trough fuel cell stack
3.32	8.01	Airflow trough fuel cell stack ($t-0.1$ s)
1.75	8.77	Cooling water temperature fuel cell stack outlet ($t-0.4$ s)
1.18	10.43	Hydrogen pressure fuel cell stack (anode)
1.17	11.21	Airflow trough fuel cell stack ($t-0.4$ s)
0.92	11.87	Air temperature stack outlet
0.94	12.07	Air pressure fuel cell stack (cathode) ($t-0.1$ s)
0.99	12.73	Cooling water temperature fuel cell stack outlet($t-0.1$ s)
1.19	14.53	Air temperature stack outlet ($t-0.4$ s)
0.83	15.47	Hydrogen pressure fuel cell stack (anode) ($t-0.1$ s)
0.90	15.92	Air pressure fuel cell stack (cathode)
0.94	19.50	Air pressure fuel cell stack (cathode) ($t-0.4$ s)
0.99	19.98	Hydrogen pressure fuel cell stack (anode) ($t-0.4$ s)
1.08	20.82	Air temperature stack outlet ($t-0.1$ s)
1.09	41.17	Cooling water temperature fuel cell stack outlet

Fig. 4 shows the overall error for the hidden neuron number optimizer (utilizing cross validation) [5] for the 11 input features chosen by the Greedy Wrapper method (Table 2). The Greedy Wrapper method employs this optimizer for every new combination of input signals. As can be seen clearly in this figure, for the final selection of 11 features the optimizer found an overall minimum at 13 hidden neurons. The local minimum at eight hidden neurons offers a reasonable tradeoff between the average error and the number of hidden neurons (representing the model complexity).

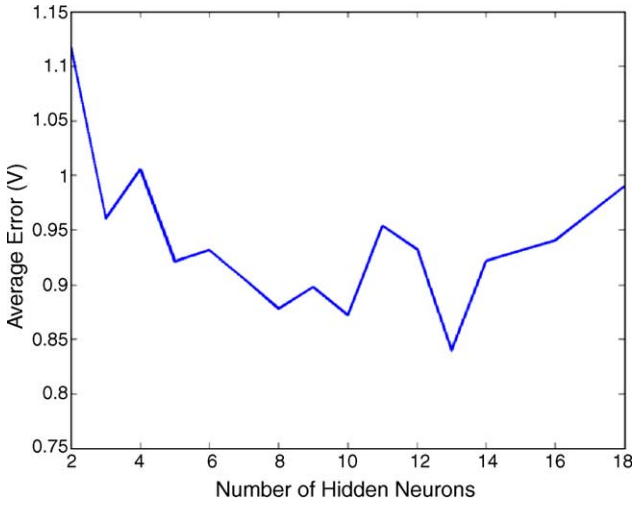

Fig. 4. Hidden neuron number optimisation.

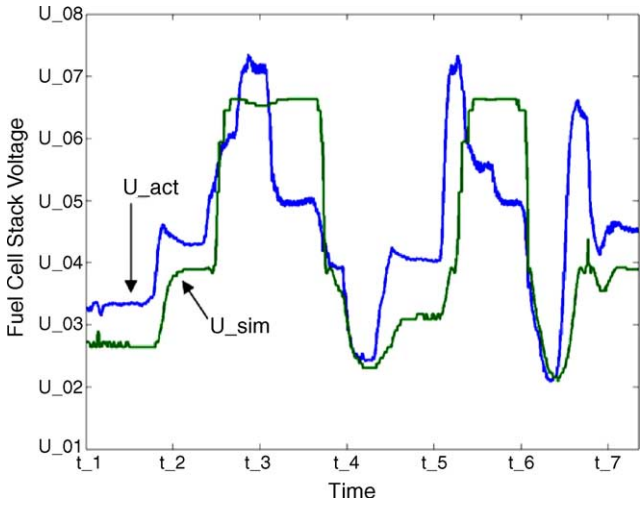

Fig. 5. Modeling with two features.

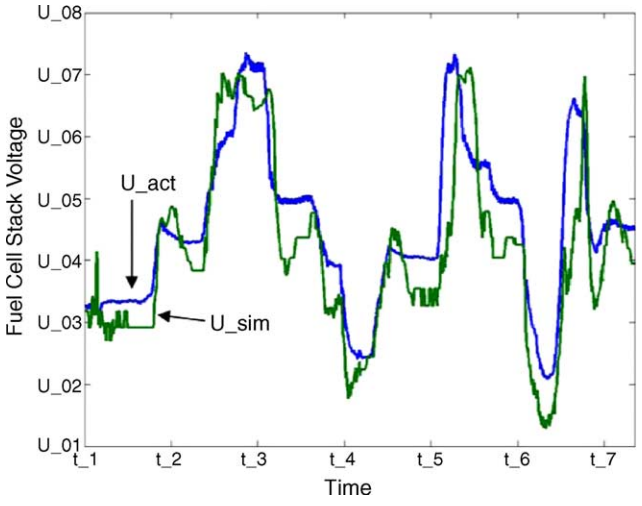

Fig. 6. Modeling with four features.

In comparison, the ARD method is much faster as the Greedy Wrapper method, since not for every input signal a trial has to be run and cross validation is not required. However cross validation might be a better estimator of the predictive potential. Another disadvantage of ARD is the fact that when fewer input signals are used that the order of signals in the relevance table might change, because the relevance values are only approximations.

5.2. Example for model refinement

The following figures are good examples to show the improvement in model accuracy by adding more and more relevant signals (features) to the ANN. The U_act represents the actual data recorded while driving the car; U_sim represents the simulated curve calculated by our model. The closer these curves are to each other, the better the model accuracy.

In Fig. 5 the two most relevant signals (from Table 2) were used to model the fuel cell stack voltage. As expected the accuracy is very poor.

In the next step the next two features (from Table 2) below the two initial features have been added to the modeling procedure. Here you can already see improvement to the model accuracy as shown in Fig. 6.

Further, we added another three features from Table 2 and reach the accuracy shown in Fig. 7.

Finally we added the last four features from Table 2 to a total of 11 features. As can be seen the accuracy of the actual real value and the simulated value match each other very well (Fig. 8).

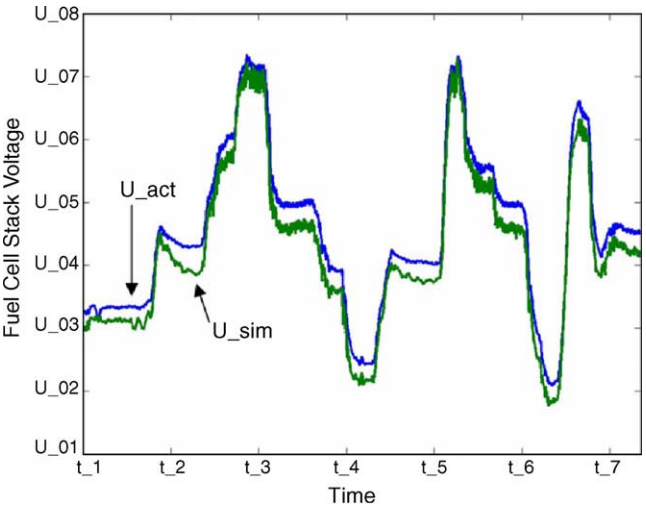

Fig. 7. Modeling with seven features.

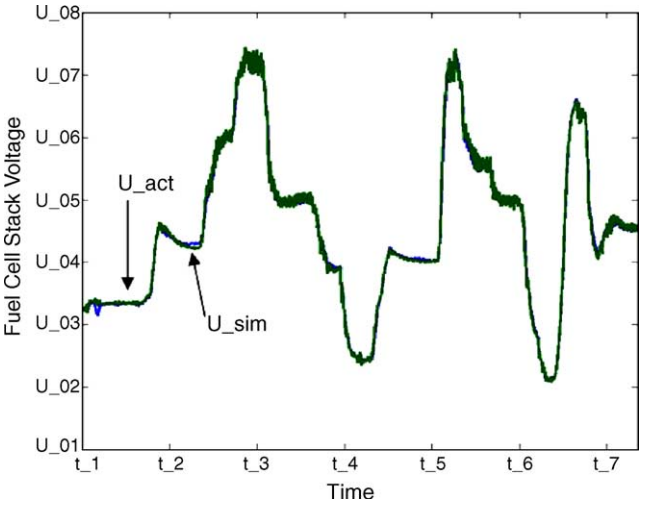

Fig. 8. Modeling with 11 features.

6. Example applications

6.1. Extraction of steady-state characteristics from dynamic data

As shown in [7] this method can be used to extract system characteristics from the fuel cell powertrain that is op-

erated under real world conditions on the streets. Therefore, the model can be trained and built in real-time (e.g. as part of an onboard diagnostics system), using two modes, one called training mode, the other called diagnostics mode.

In online training, the model gets adjusted to properly (within predefined limits) represent the transient behavior of the powertrain. At certain intervals, the system switches to diagnostics mode. That means that the model gets detached from the online signals and is fed a predefined input simulating a controlled laboratory test of the same powertrain. Since the polarization curve of the fuel cell is a good indicator of the health of the system, in particular of the degradation over time, this method can be used to derive the polarization curve, in a steady-state manner by simulating a slowly ramping up (i.e. gas flows and pressures) the fuel cell system as well as increasing the current which is drawn from the system. In Fig. 9 you see such a result.

Additionally, one can utilize this simulated polarization curve to derive the characteristic constants for the polarization curve formulas such as the activation losses, mass transfer losses and proton conductivity [1].

6.2. Simulation of varying operating conditions

In order to study various aspects of fuel cell systems, like a fuel cell propulsion system for transportation, the problem arises that some aspects cannot be studied on the system itself, either to avoid damage, or because the system is not accessible for such studies. However, such an analysis can be safely conducted in a simulation environment. The first step is, again, to create a mathematical representation of your system (a model) which can then be embedded into a simulation environment.

Therefore, building on the framework presented in Section 6.1 the same approach can be used. The created powertrain model can be fed with certain operating conditions and in order to study the effects of i.e. a higher operating pressure

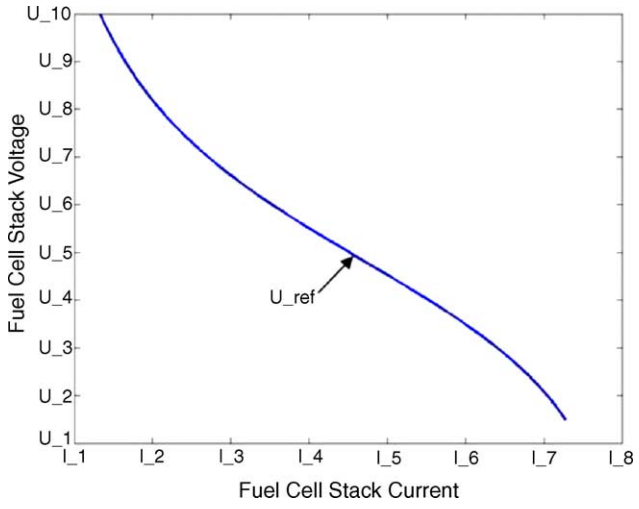

Fig. 9. Steady-state polarization curve derived from dynamic fuel cell system model.

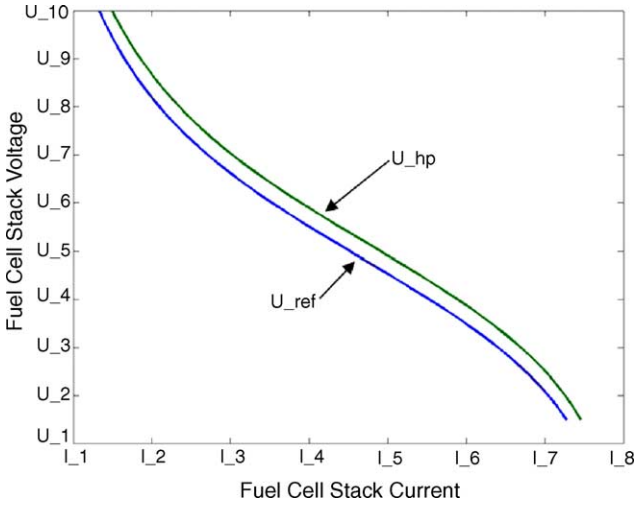

Fig. 10. Upward shift of the polarization curve due to simulating higher anodic/cathodic pressures.

on the anode and cathode sides, these new conditions can be studied using a simulation.

In the experiment shown below (Fig. 10), we simulated higher pressures inside the stack, and this resulted in the polarization curve getting shifted towards higher voltage (from U_ref to U_hp). We expected this behavior and as explained on page 104/105 in [1], it is mainly caused by increasing the catalyst site occupancy which leads to a reduction in the cathode activation voltage.

This method allows the engineer to study the effect of how much you have to increase the operating pressures to have a higher system power. At the same time you lose more energy for producing these higher pressures. This fact leads in general to an optimization problem, which can be solved using this method iteratively with a parasitic energy consumption comparison.

6.3. Modeling of complete system and system components

Our approach can easily be utilized to model not only the fuel cell powertrain as a whole but also its subsystems, like the air compressor and electric motor.

For the compressor, typical performance charts can be derived using this method. This is very useful to monitor degradation of the compressor, as well as to monitor its behavior under varying external conditions like elevation, ambient temperature and pressure, and humidity. For example, if the ambient air density changes, i.e. while driving up a mountain road; the compressor has to work harder to get the same amount of oxygen into the fuel cell system in order that the same output power can be delivered.

6.4. Optimization of sensor placement

In the past, fuel cell vehicles were more in a prototype stage than being a mass producible vehicle. Most of the cur-

rent fuel cell vehicle manufacturers are now entering the stage of going into small-scale mass production for their fuel cell vehicles. Prototype fuel cell vehicles, in general, employ several sensors and complex control systems. In order to reduce costs and the complexity of these vehicles on the way to commercialization; it is beneficial that the number of sensors used is minimized and the control systems are downsized.

Our approach enables engineers to simplify the vehicular control system through two strategies. First, this method determines the relevancy of the individual sensor signals among the array of installed vehicular sensors. Engineers can use this relevance analysis to eliminate sensors, which are not providing critical data to the control system. Second, this method is capable of modeling sensor signals based on other sensor data, therefore, creating a 'virtual' sensor. If an actual sensor signal can be accurately modeled by using other actual sensor signals, then this sensor is obsolete and can be removed.

7. Conclusion

The method of creating accurate mathematical representations (models) of an existing physical system (i.e. fuel cell powertrain) using the methods of feature selection and artificial neural networks is a powerful tool for the accelerated development of enhanced systems.

As explained in detail, the first part of this method, feature selection, derives a relevance table for predefined input signals to predefined output signals through either of the strategies (ARD/Greedy) discussed. The second part of this method builds the desired model using artificial neural network techniques. These techniques utilize the applicable input signals derived from the first step and found in the relevance table.

These models can be utilized for a variety of tasks including the simulation of various operating conditions, assessment of fuel cell powertrain status (such as fuel cell system degradation) and the optimization of sensor placement and quantities.

We presented different examples for what this method can be used. First, we show how steady-state characteristics of a fuel cell system like the polarization curve can be derived from dynamic test data. Further, we simulate higher fuel cell stack anode/cathode operating pressures and how they influence the characteristics of an existing fuel cell system.

One major drawback of artificial neural networks is the lack of physical insight into the systems they represent. However, the method presented here works around this drawback by extracting system characteristics by feeding the artificial neural network with predefined discrete time series data. This produces characteristic curves displaying the causal relationships between the input and output signals. Combining these relationships with an understanding of fundamental scientific principles governing the system gives one insight into the dynamics and physical dependencies of the system.

The method presented here is a powerful tool for the accelerated creation of models of existing systems, whereas these models are adjusted in their complexity according to their purpose.

References

[1] J. Larminie, A. Dicks, Fuel Cell Systems Explained, 2nd ed., Wiley, 2003.
[2] G. Hoogers, Fuel Cell Technology Handbook, 1st ed., CRC Press, 2002.
[3] D. MacKay, Probable networks and plausible predictions – a review of practical Bayesian methods for supervised neural networks, 1995. (http://www.inference.phy.cam.ac.uk/mackay/BayesNets.html).
[4] I. Nabney, Netlab: Algorithms for Pattern Recognition, Advances in Pattern Recognition, Springer, 2002, ISBN 1-85233-440-1.
[5] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Machine Learn. Res. 3 (2003) 1157–1182.
[6] R. Isermann, Identifikation dynamischer Systeme, 2nd ed., Springer, 2002.
[7] C. Nitsche, Onboard diagnostics concept for fuel cell vehicles using adaptive modelling, in: Proceedings of the 2004 IEEE Intelligent Vehicles Symposium, 2004.