Journal of Power Sources 187 (2009) 422–430

Contents lists available at ScienceDirect

Journal of Power Sources

journal homepage: www.elsevier.com/locate/jpowsour

Real-time optimization of net power in a fuel cell system

Judith O'Rourke [a,*], Murat Arcak [b], Manikandan Ramani [c]

[a] Department of Electrical Computer and Systems Engineering, 110 8th Street, Rensselaer Polytechnic Institute, Troy, NY 12180, USA
[b] Department of Electrical Engineering and Computer Science, Cory Hall, University of California, Berkeley, CA 94720, USA
[c] Plug Power Inc., 968 Albany-Shaker Road, Latham, NY 12110, USA

ARTICLE INFO

Article history:
Received 18 October 2008
Received in revised form 7 November 2008
Accepted 10 November 2008
Available online 24 November 2008

Keywords:
Fuel cell
Real-time optimization
Cathode air flow rate control

ABSTRACT

This paper proposes the use of extremum seeking algorithms to control the air flow into the cathode of a fuel cell system. Through experimental testing of an eight-cell, hydrogen-fueled polymer electrolyte stack, it shows that extremum seeking is a viable method to control the air flow rate. The algorithms do not rely on knowledge of system modeling parameters, and adapt to changes in those parameters that occur due to disturbances and degradation. An additional penalty on the ohmic resistance is proposed for the objective function to increase the life of the cell.

© 2008 Elsevier B.V. All rights reserved.

1. Introduction

The control and management of air flow, which provides oxygen to a polymer electrolyte fuel cell, has been the focus of many studies. The literature includes dynamic fuel cell models that incorporate feedforward and feedback air control schemes [1,2], a nonlinear reference load governor [3] to avoid oxygen starvation, and optimization methods to maximize the voltage by controlling the inlet air pressure and stoichiometry taking into account the relative air humidity [4]. Reference [5] uses a multivariable model-based control structure to control cathode pressure and oxygen flow rate. Reference [6] proposes a predictive control strategy using dynamic matrix control (DMC), which models the process using step responses. Reference [7] regulates the air and hydrogen flow rates using a multivariable transfer function matrix model. These control strategies stabilize a predetermined air stoichiometry which may or may not be the optimal operating condition. Another disadvantage of these methods is that they require knowledge of system parameters, which must be either measured offline or estimated. However, in practice, parameters vary over time because of performance degradation, clogging of air filters, and contamination of gas diffusion layers.

This paper proposes an air flow control strategy that optimizes net power in real-time without relying on system parameters. The proposed *extremum seeking* algorithm manipulates the air flow rate into the cathode and measures the net output power, which is the stack power minus the power consumed by the air blower/compressor. In a low temperature PEMFC system, as the air flow rate is increased the net power increases up to a point where parasitic losses (blower/compressor power) cause a decline. This defines a function between air flow rate and net power that exhibits a peak. To locate the peak of this air flow rate-net power curve, our algorithm starts the air flow at a nominal rate, makes a small step change and measures the net power. The next step change to the air flow rate is based on the net power gradient, which is estimated using finite difference methods without relying on the knowledge of the air flow-net power curve. The algorithm continues adjusting the air flow rate until the estimated gradient is below a threshold. Extremum seeking in various forms has long been used by control engineers in numerous applications [8–11].

After successfully demonstrating the basic extremum seeking algorithms, we proceed to propose several variants: The first variant includes a regulating valve for the air outlet manifold as a second control input as suggested in [6]. The second variant interrupts the algorithm when significant changes in the load current occur, and resets the airflow to avoid oxygen starvation and to reduce convergence time.

During the optimization as air flow step changes are made, the net power initially has an inverse response. The blower or compressor reacts much faster than the stack voltage which causes an initial drop or spike in the net power. Since this behavior is a concern for the power conditioning modules, we propose a third variant of our design in which a step change is replaced with a slower ramping of the air flow rate, thus reducing the magnitude of the net power inverse response.

A disadvantage of existing air control schemes is that they consider only the short-term effect of the air flow on the stack power.

* Corresponding author. Tel.: +1 518 861 5016; fax: +1 518 276 6261.
E-mail addresses: orourkej@strose.edu (J. O'Rourke), arcak@eecs.berkeley.edu (M. Arcak), Manikandan_Ramani@plugpower.com (M. Ramani).

0378-7753/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.jpowsour.2008.11.060

In the long term, however, increasing the air flow to maximize the net power may compromise the durability of the cell by causing drying of the membrane, which then leads to hot spots and faster deterioration. To trade short-term net power with durability, in the fourth variant of our design we modify the objective function by including a penalty on high resistance values which indicate membrane drying.

The rest of the paper is organized as follows: Section 2 explains the extremum seeking algorithms used and demonstrates their performance with a fuel cell simulation model developed by Stefanopoulou and coworkers [1,2,12]. Section 3 presents experimental results performed at Plug Power. Due to limited availability of the testing facilities, some of the variants of our algorithms were investigated using only simulations. Section 4 presents simulation results using two manipulated inputs. Section 5 appends the extremum seeking algorithms to account for changes in the current load. Section 6 alleviates the inverse response of the net power by replacing step changes with tapered input profiles. Section 7 demonstrates the effect of air flow on resistance, derives a penalty function for our objective function, and then presents simulation results. Section 8 gives the conclusions.

2. Extremum seeking algorithm

The objective of the extremum seeking algorithm is to manipulate the air flow rate into the cathode to maximize net power. The net power is defined as the power produced by the fuel cell stack minus the power consumed by the air blower or compressor. The algorithms developed in this section consist of a single measured plant parameter, and a single manipulated input. All other operating parameters are considered fixed during the optimization. In the next step, the air flow rate is determined by the gradient of the net power, which is estimated by finite difference methods [13].

2.1. Coarse steepest ascent method

The 'coarse' steepest ascent optimization algorithm allows the step size to vary in accordance with an estimate of the gradient [13]. An initial step increase to the manipulated input u is made, the system is run until it reaches steady state, and the output y is measured. The next step change to the compressor is calculated using Eq. (1), where G is a tuning gain factor:

$$u(k+1) - u(k) = G\frac{y(k) - y(k-1)}{u(k) - u(k-1)} \quad (1)$$

The algorithm stops when the change in net power is sufficiently close to zero, as determined by a user-defined deadband.

To examine the performance of our algorithms we use an existing dynamic fuel cell Simulink system model developed in [1,2,12]. The fuel cell system Simulink model incorporates auxiliary components, specifically an Allied Signal compressor, manifolds, an air cooler and a humidifier. A characteristic map is used to model the air compressor. The fuel cell stack model is comprised of four submodels: stack voltage, cathode flow, anode flow and membrane hydration. The stack voltage is a function of the cell temperature, air pressure, oxygen and hydrogen partial pressures, and membrane humidity. The number of cells is set to 381 with active cell area of 280 cm^2. The air is supplied by a compressor and the flow rate is controlled by the voltage applied to the compressor. The compressor voltage is our manipulated input. The output is the calculated net power, which is the stack voltage multiplied by the current load minus the power consumed by the compressor. Prior to running our optimization routines, several different air flow rates and their corresponding net powers are plotted in Fig. 1. The simulations show that at the given operating conditions the net power reaches a maximum of 27,922 W at an air stoichiometry of 2.5. The cathode

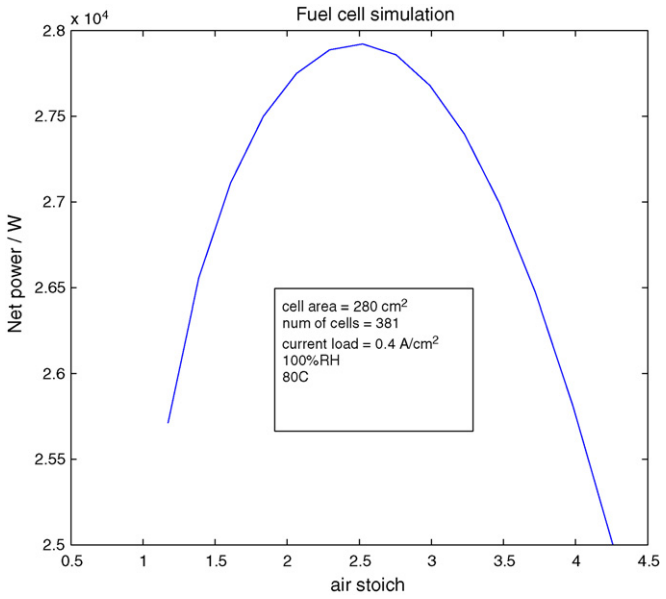

Fig. 1. Net power as a function of cathode air flow rate shows a peak net power of 27,922 W at an air stoichiometry of 2.5.

pressure increases as the air flow rate increases, causing stack voltage to increase. However at air flow rates above a stoichiometry of 2.5 the net power decreases because the power consumed by the compressor is increasing more than the gain in stack power.

Fig. 2 displays the results of the coarse steepest ascent algorithm with a gain $G = 0.5$ and a deadband of 1.0 W. In each step, we let the system run for two seconds to reach steady state. It takes 4 steps thus 8 s, starting from a cathode stoichiometry of 1.8 to reach the maximum net power at a cathode stoichiometry of 2.5. The spikes in net power at each step change in air flow rate are caused by the immediate power consumption of the compressor. The stack power takes longer to reach its new value.

2.2. Refined steepest ascent method

The coarse steepest ascent algorithm used a forward difference formula to estimate the gradient. To improve the accuracy of the

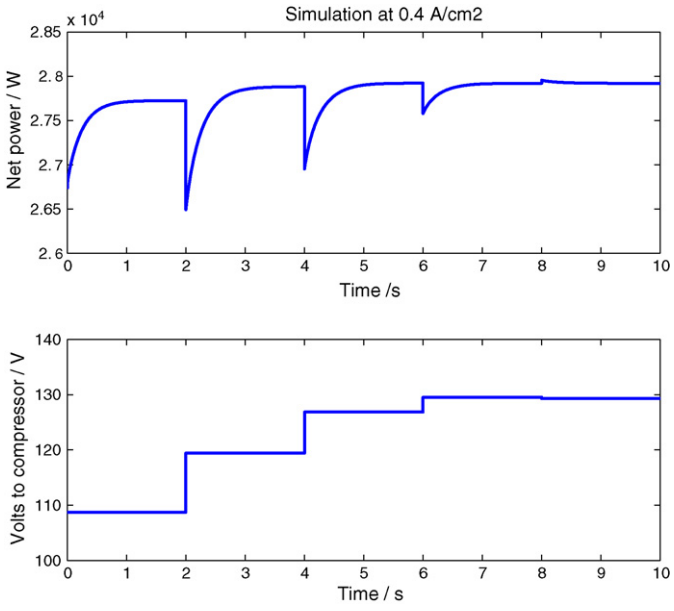

Fig. 2. Coarse steepest ascent algorithm with gain = 0.5 and deadband = 1.0 W, converges in 10 s.

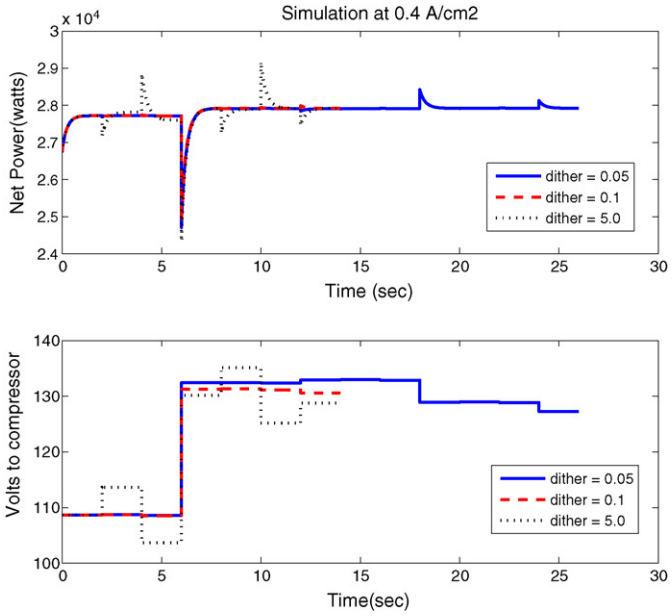

Fig. 3. Refined steepest ascent algorithm with various dither values. A large range of dither sizes yield comparable performance.

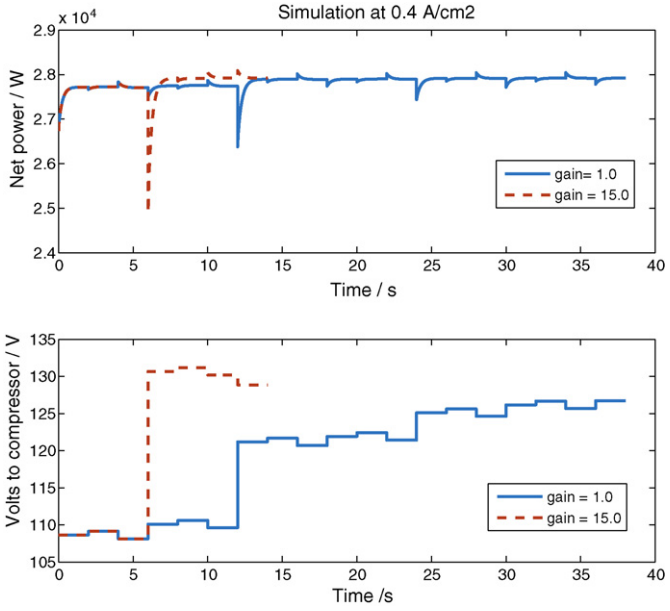

Fig. 4. Newton-type algorithm with gains of 1 and 15. A higher gain is needed to reduce convergence time.

gradient estimation, we now use a central difference formula and introduce an independent dither parameter h. Each step now consists of two dither periods where the input is $u^+(k) = u(k) + h$ and $u^-(k) = u(k) - h$. The resulting net power values $y^+(k)$ and $y^-(k)$ are recorded, and the step change for u is determined from:

$$u(k+1) - u(k) = G\frac{y^+(k) - y^-(k)}{2h} \tag{2}$$

The routine continues until the difference in net power is within the deadband.

Simulation results are shown in Fig. 3 for a gain of 1.0, deadband of 1.0 W and three different dithers; 0.05, 0.10 and 5.0 V. The refined steepest algorithm performs the same for dithers in the range of 0.10–5.0 V, converging to the maximum net power within 14 s. At a low dither of 0.01 volts, the algorithm reaches the maximum net power but convergence takes longer.

2.3. Newton-type method

Next a Newton-type algorithm is investigated. The Newton algorithm includes using finite differences to estimate the Hessian matrix. An estimate of the second derivative is included in the step adjustment to the input:

$$u(k+1) - u(k) = -G\frac{y'}{y''}$$

$$y' = \frac{y^+(k) - y^-(k)}{2h}$$

$$y'' = \frac{\frac{y^+(k) - y(k)}{h} - \frac{y(k) - y^-(k)}{h}}{h} \tag{3}$$

Simulation results are shown in Fig. 4, with a dither of 0.50 V, a deadband of 1.0 W and gains of 1.0 and 15. The gain of 1.0 takes 38 s to converge to an air stoichiometry of 2.45 and a net power of 27,921 W. A gain of 15 results in convergence within 14 s. A higher gain is needed for the Newton-type algorithm. The Newton-type algorithm finds the optimal with two-step changes to the air flow rate. However the time it takes to make these step changes is longer than the coarse steepest ascent algorithm because the Newton method requires an estimate for the second derivative. Simulation results do not indicate any performance improvement with Newton method.

3. Experimental

Testing was performed in a laboratory at Plug Power Inc. in Latham, NY. The experiments involve an eight-cell stack with a 3 M commercially available membrane electrode assembly. The membrane thickness is 1.1 mils (0.002794 cm) and the active area of the cell is 262 cm^2. The reactant flows to the stack are hydrogen and air. The anode and cathode gas streams are 100% humidified with external humidifiers. The coolant, air and hydrogen into the stack are heated with external heaters and the temperature of the gas

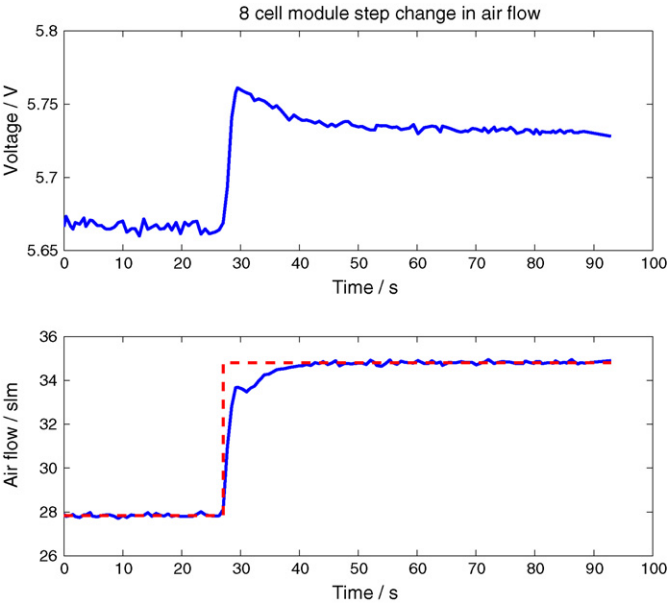

Fig. 5. Experimental results for an open loop set change in air flow rate. Stack voltage reaches steady state after 23 s.

into the cathode and anode are controlled to the same temperature as the coolant inlet temperature of 55 °C.

National Instrument's SCXI and LabVIEW provide data logging, user interface, and signal conditioning for the pressure, temperature, voltage, current and flow rate sensors. A HC Power Inc. load bank supplies the current load to the stack. The air to the stack is supplied by shop air and does not use an individual blower. To estimate the power loss that a blower would introduce, the fluid power formula is used (the product of the air inlet measured pressure and commanded flow rate).

To evaluate the performance of the extremum seeking algorithms, it is necessary to understand the behavior of net power as a function of air flow rates. A step change is made to the air flow setpoint and the air flow and the stack voltage are measured. All other parameters are held constant. The hydrogen flow rate is fixed at a stoichiometry of 1.9, current density is set to 0.4 A cm^{-2}, temperature of the cell and inlet reactants are controlled to 55 °C and air and hydrogen are 100% externally humidified. Fig. 5 displays the response to the step change. Since the air is not supplied by a blower or a compressor there is no large spike in consumption of power when the air flow setpoint is increased. The mass flow air control valve allows the air to reach setpoint in 13 s without any overshoot. The stack voltage rises slightly above its steady state point when the air flow rate is initially increased and then drops to its steady state after 23 s. This experiment indicates that we need to allow a significant settling time before taking measurements. 30 s is used throughout the lab experiments.

Fig. 6 displays the net power of the eight-cell stack as a function of air flow rate for both current loads of 0.4 and 0.7 A cm^{-2}. The net power measurements have a considerable amount of scatter at each flow rate which is caused by the fluctuation in stack voltage. This scatter adds to the difficulty of the extremum seeking algorithm finding the maximum.

First the coarse steepest ascent algorithm is implemented. The gain is 10, the deadband is 0.25 W and the initial change in air flow rate is 1.0 slm. All other parameters are kept the same as in the prior step test. Fig. 7 displays the results. The optimization begins at an air flow of 4.0 stoichiometry and after 240 s (7-step changes to the air flow) the algorithm reaches a net power of 600 W at an air flow of 2.6 stoichiometry. The net power is slightly higher than shown in Fig. 6 (0.4 A cm^{-2}). This supports our argument that a fuel cell stack has an uncertain power map and there is a need

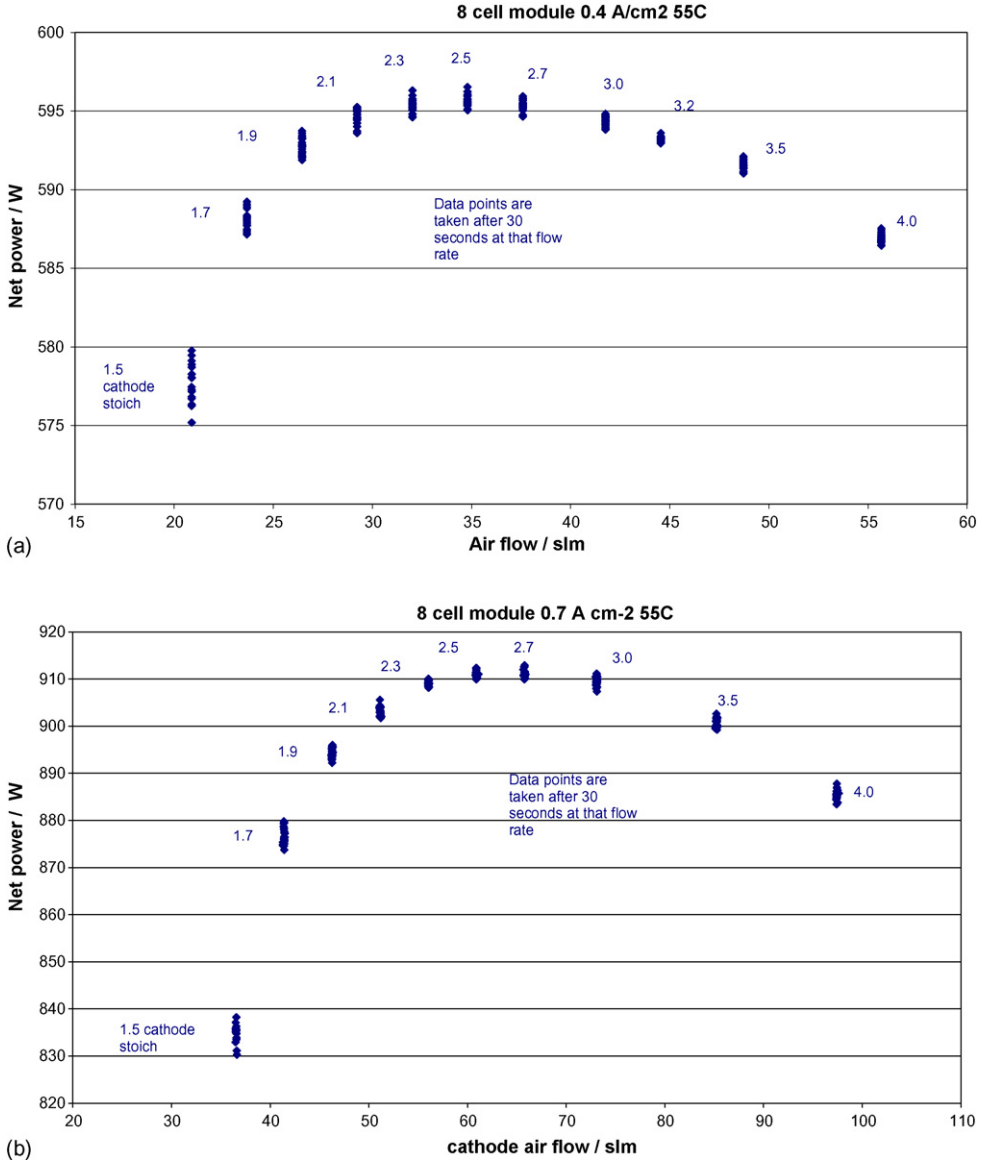

Fig. 6. Experimental data net power as a function of air flow rates. Large scatter in measurements at fixed air flow rate.

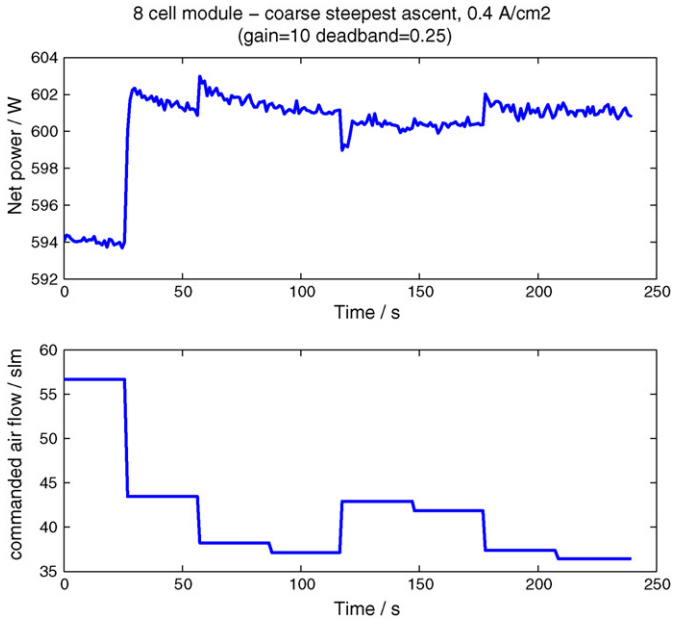

Fig. 7. Experimental results of the coarse steepest ascent algorithm. It takes 7-step changes to air flow rate to reach maximum net power.

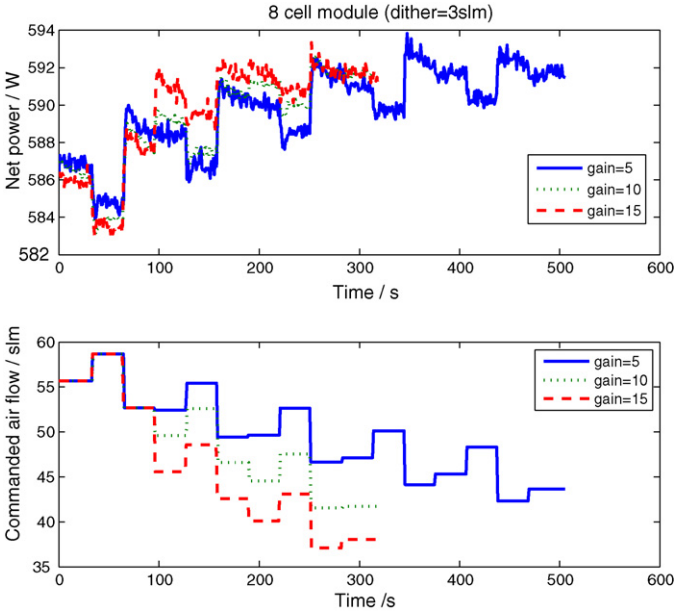

Fig. 9. Experimental results for refined steepest ascent algorithm at different gains. Higher gain yields faster convergence.

for an extremum seeking algorithm to find the optimal air flow rate.

The test is repeated by starting at a lower air flow rate of 1.7 stoichiometry. Fig. 8 displays the results. The algorithm took 337 s to reach a net power of 599 W at an air stoichiometry of 3.00. The current load on the stack is increased to 0.7 A cm^{-2} and the algorithm is tested again. After 731 s the algorithm finds a maximum net power of 910 W at 2.8 air stoichiometry. The test is immediately repeated to see if the algorithm will find the same optimal air flow rate. The algorithm converges at a slightly higher air flow rate of 2.9 stoichiometry with a net power of 906. If we test lower gains of 2 and 3 the algorithm converges in half the time but results in lower net power (892 and 889 W) at much higher air stoichiometry of 3.6 and 3.5. Fig. 6 demonstrates that at a fixed air flow rate,

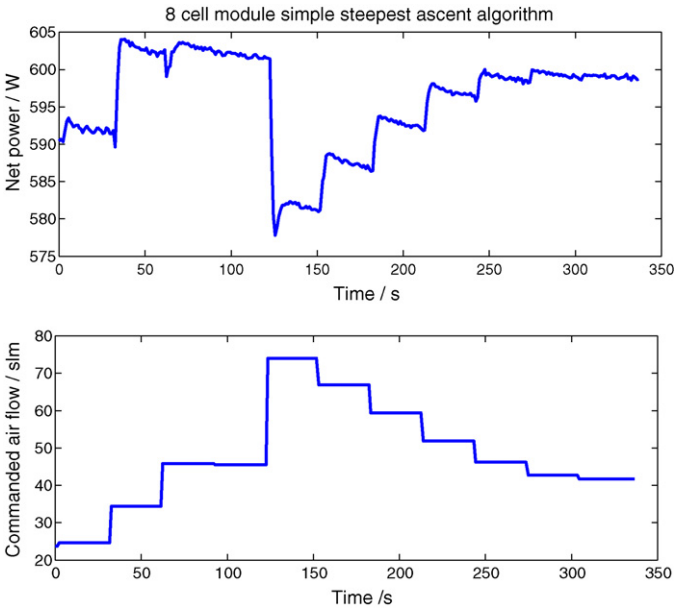

Fig. 8. Experimental results of the coarse steepest ascent algorithm starting at low air flow rate.

the net power has a range of 8 W, because of this large scatter (due to the behavior of the voltage measurements) the optimal air flow rate could be between 2.3 and 3.0 stoichiometry. When we reduced the gain to 2 or 3 the coarse steepest ascent did not converge to the global maximum.

To improve robustness and reduce the effects of the voltage measurement scatter, the refined steepest ascent algorithm is used and a voltage averaging filter is introduced. After the system runs for 30 s the voltage is measured for three consecutive time steps and the average of these is used in the algorithm.

Fig. 9 displays the results of the steepest ascent algorithm using the central difference formula for a load of 0.4 A cm^{-2}, dither of 3 slm, deadband of 0.75, starting stoichiometry of 4.0 and three different gains. At gains of 5, 10 and 15 the algorithm converges in 318, 318 and 505 s respectively. The air flows converge to values between 2.7 and 3.1 stoichiometry and the corresponding net powers range from 592 to 597. At a gain of 10 and 15 three iterations are needed to reach its optimal air flow rate. Though it takes only three iterations, the added time to dither and filter makes the convergence time slightly longer than the simple algorithm. The advantage of using the filter and the central difference formula is improved repeatability.

The next test is performed at a gain of 10 and starting the optimization at a low air flow rate of 1.7 stoichiometry. Fig. 10 shows the comparison between starting at a low and high air flow rate. The algorithm reaches an optimal air flow rate of 40 slm (2.87 stoichiometry) with a net power of 596 W if optimization starts at a low air flow rate. It reaches an optimal of 42 slm (2.99 stoichiometry) with a net power of 591 W at a high starting air flow rate.

The next test allows the algorithm to find the optimal air flow rate at different operating points. Fig. 11 displays the results. The test starts at a current load of 0.7 A cm^{-2} (183 A) and a net power of 881 W. After 330 s the algorithm adjusts the air flow and finds the optimum at 899 W. The current load is decreased to 0.50 A cm^{-2} (133 A) and the net power starts at 700 W. The algorithm is activated and results in a net power of 717 W. The load is increased to 0.55 A cm^{-2} and the net power is 765 W. The algorithm runs and manipulates air flow so the net power is increased to 769 W. The results show that the algorithm does find the maximum net power at each operating point. It demonstrates that if the air flow is in

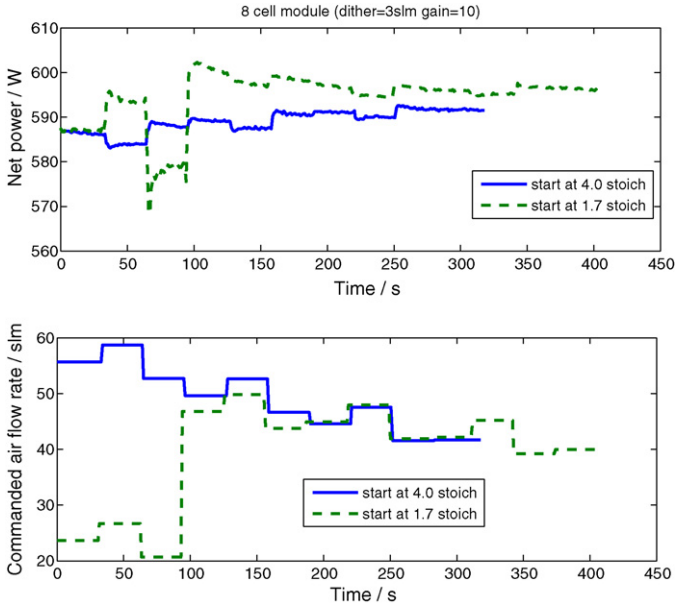

Fig. 10. Experimental results for refined steepest ascent algorithm at different starting flow rates.

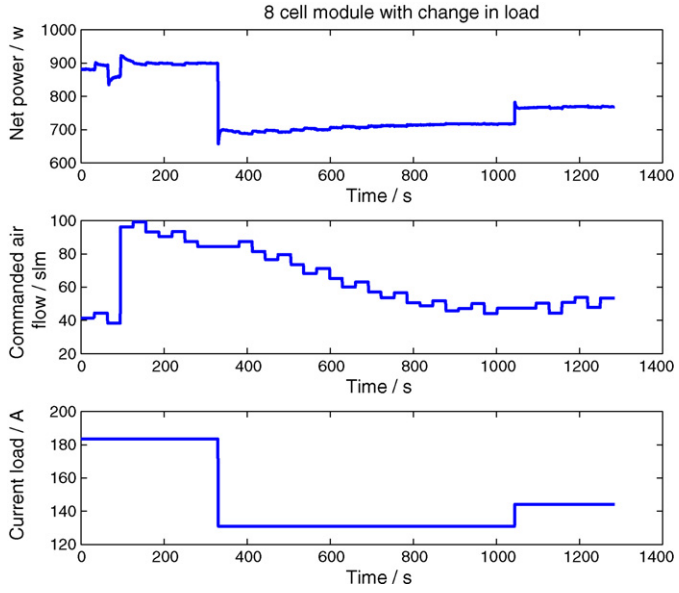

Fig. 11. Experimental results for refined steepest ascent algorithm at varying current loads.

the range of the optimal point, as in the case of going from a load of 0.5–0.55 A cm^{-2}, the algorithm slightly manipulates the air flow and increases the output by more than 4 W.

4. Two manipulated inputs

Our extremum seeking algorithms are now extended from a single input-single output model to a two input-single output system. As proposed in [6], we take the second input to be a regulating valve that controls the cross sectional area of the air outlet manifold. Changes in the outlet area change the air flow rate and influence performance. Increasing the cross sectional area of the outlet manifold allows a greater air flow rate, thus increasing the compressor power consumption. However, it also increases the oxygen partial pressure, which increases the stack power. At larger cross sectional areas, increasing the area causes a slight drop in oxygen partial pressure minimally decreasing stack power.

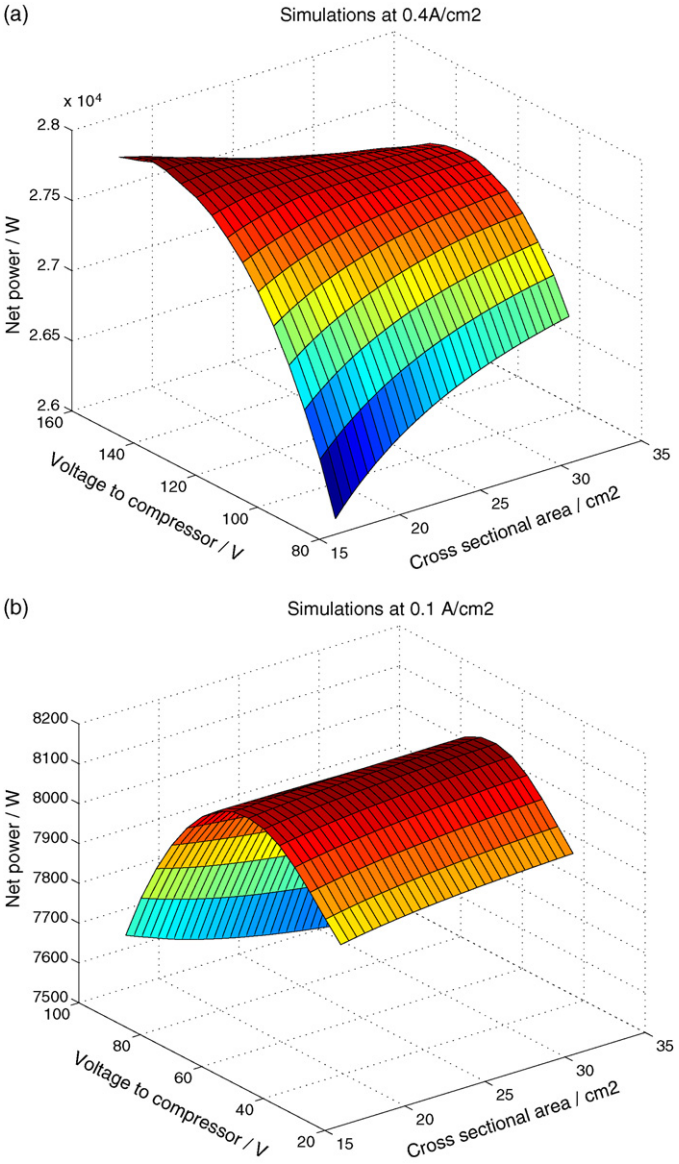

Fig. 12. Simulations for various air flow rates and air outlet manifold cross sectional area.

Manipulating the cross sectional area of the air outlet manifold in addition to the voltage to the compressor results in a maximum net power greater than when manipulating one input. Fig. 12 displays simulations results of the net power at various combinations of voltage to the air compressor and cross sectional areas for a current load of 0.4 and 0.1 A cm^{-2}. The area near the maximum net power is flat, indicating that various configurations result in relatively similar net power.

We modify our extremum seeking algorithm to manipulate the two inputs concurrently: Each step now consists of four dither periods where, in the first two, voltage to the compressor is $u_1^+(k) = u_1(k) + h_1$ and $u_1^-(k) = u_1(k) - h_1$, with corresponding net power values $y_1^+(k)$ and $y_1^-(k)$. In the third and fourth dither periods $u_1(k)$ is kept constant and the air outlet cross sectional area is modified to $u_2^+(k) = u_2(k) + h_2$ and $u_2^-(k) = u_2(k) - h_2$, with corresponding net power values $y_2^+(k)$ and $y_2^-(k)$. The update law is

$$U(k+1) = U(k) = [Y^+(k) - Y^-(k)] \begin{bmatrix} \frac{G_1}{2h_1} & 0 \\ 0 & \frac{G_2}{2h_2} \end{bmatrix} \quad (4)$$

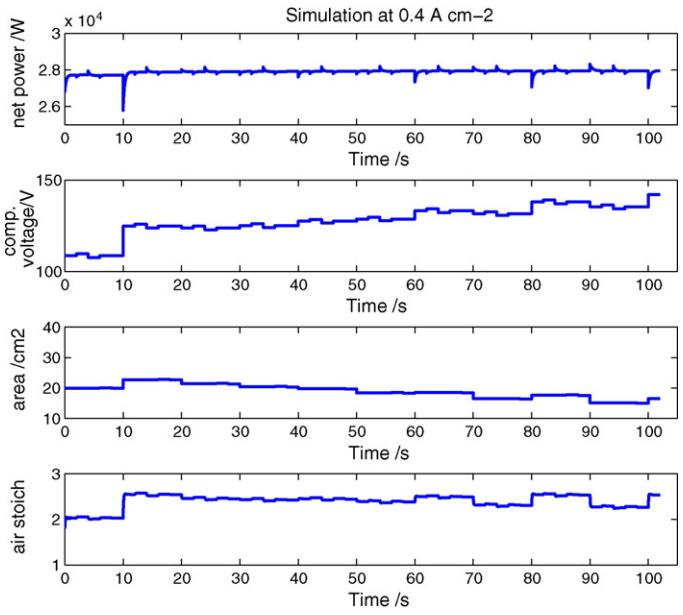

Fig. 13. Extremum seeking that manipulates two inputs concurrently.

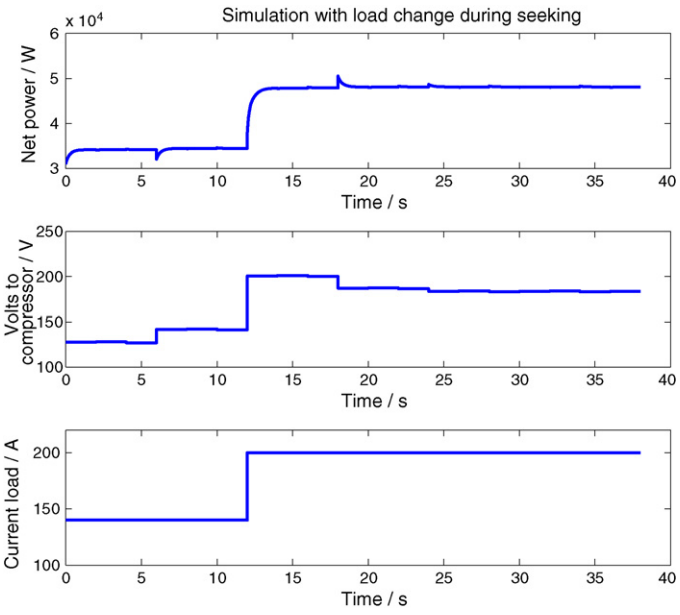

Fig. 14. Simulation with load change during optimization.

where G_1 and G_2 are the tuning gains, and

$\boldsymbol{U}(k) = [u_1(k),\ u_2(k)]$
$\boldsymbol{Y}(k) = [y_1(k),\ y_2(k)]$
$\boldsymbol{Y}^+(k) = [y_1^+(k),\ y_2^+(k)]$
$\boldsymbol{Y}^-(k) = [y_1^-(k),\ y_2^-(k)]$

Simulations are performed at a current load of 0.4 A cm^{-2}. It takes 102 s to converge to a net power of 27,951 W, as shown in Fig. 13. Including a regulating valve to adjust the cross sectional area of the air outlet manifold slightly increases the maximum net power attained, but convergence time is increased.

5. Adapting to changes in the current load

So far we have presented optimization methods at a fixed current load. In this section we augment our single input extremum seeking algorithms to incorporate changes in the current load during and after optimization. Using the fuel cell simulation model, we propose to interrupt the optimization routine when there is a significant current load change. If the change in current load is greater than a predetermined threshold, we scale the air flow rate by an amount proportional to the change in current load and continue the optimization algorithm with this reset value as the initial condition:

$$u(k+1) = u(k)\left(\frac{I(k)}{I(k-1)}\right) \tag{5}$$

$u(k)$ = voltage to the compressor (V); $I(k)$ = current load (A).

The particular form of the reset rule (5) is because, if current load changes, then the oxygen usage changes proportionally.

In simulations we chose a threshold of 3% and interrupted the algorithm whenever the current load changed by more than 3%. Fig. 14 displays simulations results where the current load changes from 140 to 200 A during an optimization routine.

6. Tapering the step changes in the input

Both experimental and simulation data show that the net power has an initial inverse response to a step increase in the air flow rate. This is because, when the voltage to the compressor is increased, the compressor reacts faster than the stack, thus causing an initial drop in net power until the stack voltage increases. To reduce this inverse response, we employ ramp inputs rather than step changes in the air flow rate. Fig. 15 shows the stack voltage and net power responses after an open loop step change in the voltage (110–120 V) to the compressor and ramp inputs with slopes of 10, 20 and 40. We note that the magnitude of the net power inverse response indeed decreases as the slope of the ramp decreases, but settling time increases.

We modified our refined steepest algorithm to ramp to the desired new compressor input voltage within 0.25 s. Fig. 16 shows a comparison of the ramp and step input profiles during optimization. Both have similar behavior with the exception that the magnitude of the largest change in input voltage results in an 11% inverse magnitude response for the step input and a 6% inverse response for the ramp input profile.

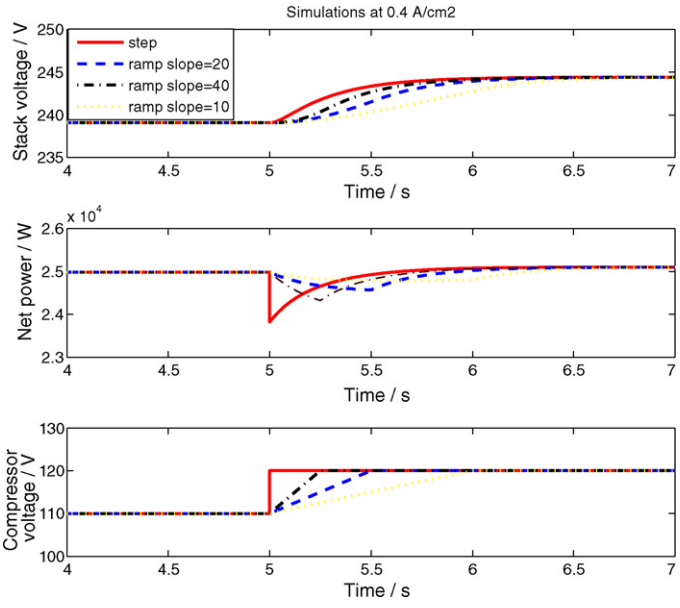

Fig. 15. Open loop simulations with various input profiles.

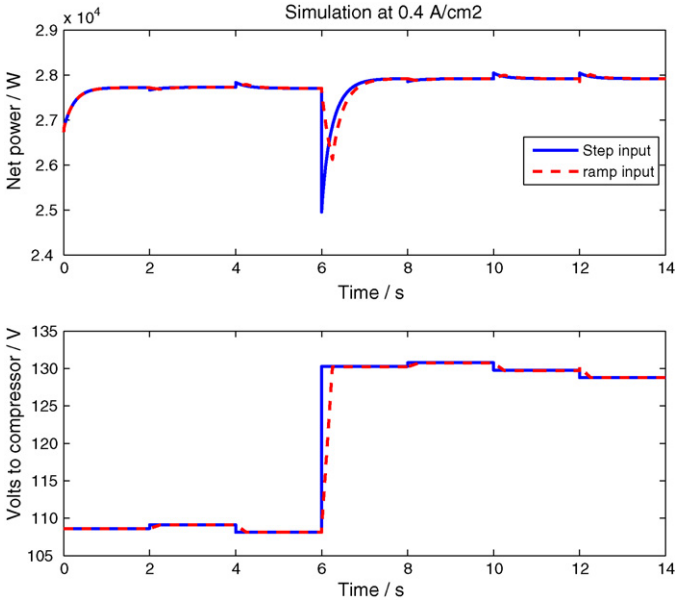

Fig. 16. Refined extremum seeking with step input changes vs. ramp input changes.

7. Introducing penalty on the resistance

We now take into account the effect on the life of the cell when we increase the air flow. Running at higher air flow rates and not externally humidifying the air will cause the ohmic resistance to increase. Higher ohmic resistance means faster deterioration of the cell. Thus, there exists a tradeoff between the immediate increase in net power at a higher air flow rate and the lifetime of the cell.

We first demonstrate that if the air is not 100% humidified, as the air flow rate increases so does the ohmic resistance. Fig. 17 displays the test data for three different air humidifications. At a current load of 0.4 A cm^{-2} the air flow rate is held constant for five minutes while the impedance is continuously measured at a frequency of 3700 Hz. The experiment with no external humidification shows that as the air is increased at each step, the ohmic resistance increases. There is a 25% increase in ohmic resistance from an air stoichiometry of 1.5–3.0. The ohmic resistance increases from 0.109 to 0.136 ohm cm^2. The 100% humidification shows no change in resistance and the 50% humidification shows a 2% increase in resistance over the whole range of air flows. The test is repeated for operating points of 0.7 and 0.1 A cm^{-2}. The results for the higher flow conditions (0.7 A cm^{-2}) are very similar to those in Fig. 17. However, for the low air flow conditions (0.1 A cm^{-2}), the overall change in ohmic resistance as non-humidified air flow increases is 46%. Ohmic resistance is a function of membrane thickness and conductivity [14]. The conductivity is a function of membrane water content and temperature. The temperature is fixed and at 100% humidification the membrane water content remains fixed, therefore resistance does not vary with air flow rates.

Many fuel cell systems do not run at 100% humidified inlet air. Systems may use internal humidification, or if using an external humidifier, choose not to run at 100% because of concerns about flooding. This leads to the conclusion that the air flow rate has an effect on the membrane water content and membrane water content plays a key role in the length of life.

To create a relationship between resistance and life of a cell we used the experimental results of the number of hours of cell life at different relative humidity conditions given in [15]. We then correlate this with ohmic resistance values at the same relative humidity conditions. Using the Simulink fuel cell model we ran three different operating conditions: 100% relative humidity anode with 70% cathode, 100% anode with 0% cathode, and 0% anode and 0% cathode. The ohmic resistances were 0.0377, 0.0712 and 0.1328 ohm cm^2 respectively. Simulations are done at a current load of 0.54 A cm^{-2}, an air flow rate of 2.5 stoichiometry and a fuel cell temperature of 80 °C. A mathematical equation is derived to correlate cell life (hours) as a function of ohmic resistance (ohm cm^2). A power law is determined to best correlate the data:

$$\text{cell\_life} = 4.527R^{-2.02}. \tag{6}$$

Using the Simulink fuel cell model with a fuel cell temperature of 80 °C and both the anode and cathode 100% humidified, the ohmic resistance for all current loads is approximately $R_0 = 0.037$. Any resistance larger than R_0 creates a greater loss to the life of the cell and is incorporated as a penalty function in our optimization algorithm. Thus, when $R > R_0$, we modify the objective function to be

$$\text{netpower} - \lambda\{4.527(R_0^{-2.02} - R^{-2.02})\} \tag{7}$$

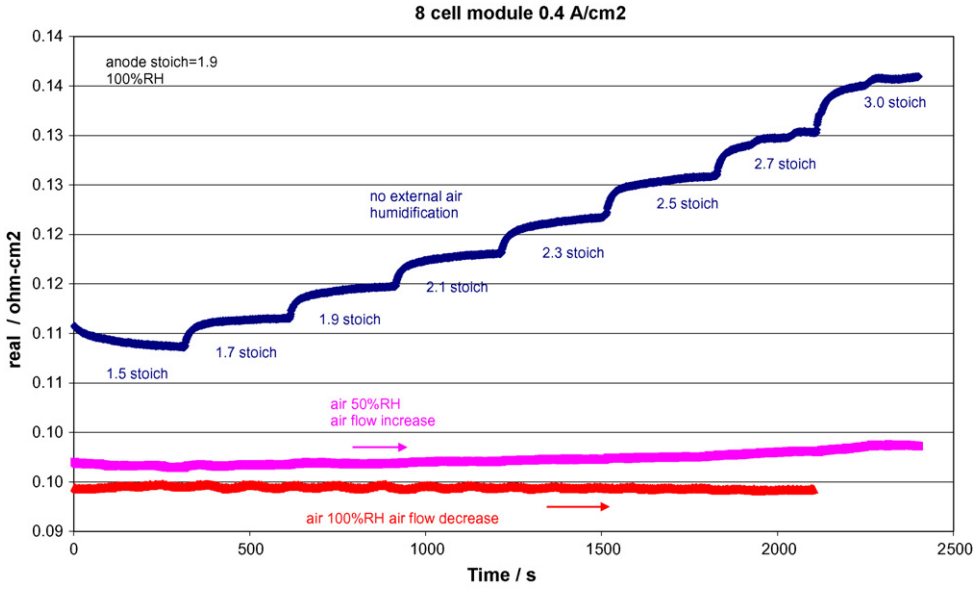

Fig. 17. Ohmic resistance as a function of air flow rate. In the absence of humidification as air flow rate increases ohmic resistance increases.

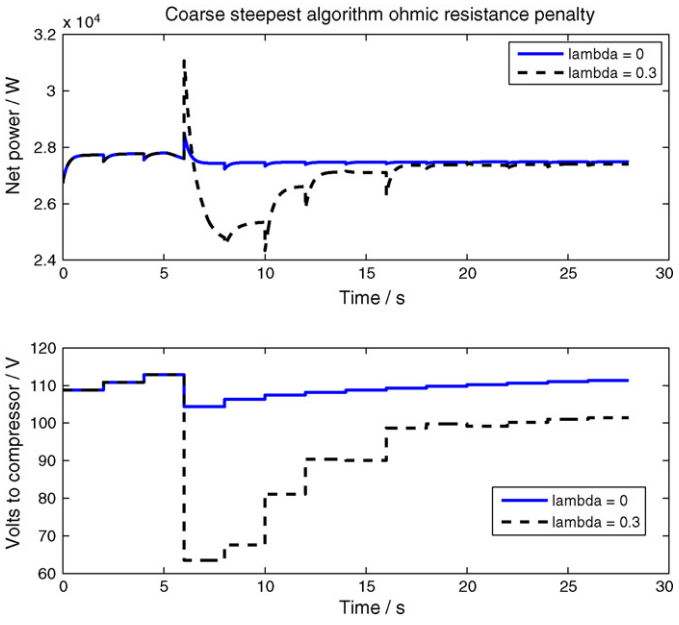

Fig. 18. Coarse steepest ascent algorithm with penalty function (5). An increase in the life of the cell is expected with lower air flow rates.

The parameter λ in Eq. (7) has units of W h^{-1} and is a variable that is to be selected by the user. It describes how many watts of immediate net power the user is willing to sacrifice for a unit increase in the lifetime of the cell.

Fig. 18 compares the simulations results for the coarse steepest descent algorithm with the cathode air at 0% relative humidity and λ at 0 and 0.3. With inlet air being at 0% relative humidity and no penalty to net power for higher resistance ($\lambda = 0$), the algorithm results in the air stoichiometry of 2.2 and a net power of 27,480 W. If we include a penalty with $\lambda = 0.3$, the optimal air flow becomes 1.9 and net power is reduced to 27,402, which means we give up 78 W.

8. Conclusions

This paper proposed extremum seeking algorithms that can be used to enhance performance of the fuel cell system. Our algorithms manipulate air flow rate into the cathode to maximize the net power. The algorithms adapt to changes in operating parameters, disturbances and system degradation. We demonstrate and compare performances of a coarse, refined, and Newton-type algorithm. Simulation results showed that all three algorithms had comparable accuracy. However, the speed of convergence was much faster for the coarse steepest ascent. The refined and Newton-type algorithms use the central difference formula to estimate the gradient, and, thus require additional time to dither the air flow. The simple and refined algorithms were tested using an 8 cell stack module. Experimental data showed noisy stack voltage measurements (Fig. 6) which affected the accuracy of the extremum seeking algorithms. This problem was alleviated by adding an averaging filter to the voltage measurement. Several modifications of the algorithms were then proposed to include a second input variable (regulating valve opening for the air outlet manifold), to reset the initial conditions of the algorithm in response to large changes in the current load, and to taper the step changes employed in the algorithm so that negative spikes in the power are mitigated. Finally, we proposed a penalty to the objective function that accounts for high air flow rates that may degrade the membrane at a faster rate. The designer decides how many watts of immediate net power she is willing to sacrifice for a unit increase in the lifetime of the cell.

Acknowledgements

The work of the first and second authors is supported in part by the National Science Foundation under Grant No. ECCS 0238268. The authors would like to thank Ms. Reeney Umstead and Tomas Gibney at Plug Power Inc. for their assistance in setting up and testing of the modules. Many thanks are given to Matthew Kuure-Kinsey for his guidance and ideas regarding the operation of fuel cell systems.

References

[1] J.T. Pukrushpan, Modeling and control of fuel cell systems and fuel processors, Dissertation, University of Michigan, 2003.
[2] J.T. Pukrushpan, A.G. Stefanopoulou, H. Peng, IEEE Control Systems Magazine (April 2004) 30–46.
[3] J. Sun, I. S Kolmanovsky, Proceedings of the American Control Conference, vol. 4, June 2004, pp. 3075–3081.
[4] B. Blunier, A. Miraoui, Vehicle Power and Propulsion, 2005 IEEE Conference (2005) 273–279.
[5] M. Danzer, J. Wilhelm, H. Aschemann, E.P. Hofer, Journal of Power Sources 176 (2008) 515–522.
[6] D. Feroldi, M. Serra, J. Riera, Journal of Power Sources 169 (2007) 205–212.
[7] F.C. Wang, H.T. Chen, Y.P. Yang, J.Y. Yen, Journal of Power Sources 177 (2008) 393–403.
[8] P.E. Wellstead, M.B. Zarrop, Self-Tuning Systems Control and Signal Processing, Wiley, England, 1991.
[9] V. Eveleigh, Adaptive Control and Optimization Techniques, McGraw-Hill Inc., New York, 1967.
[10] K. Ariyur, M. Krsti, Real Time Optimization by Extremum Seeking Control, Wiley, 2003.
[11] D. Popovi, M. Jankovi, S. Magner, A.R. Teel, IEEE Transactions on Control Systems Technology 14 (May (3)) (2006).
[12] A.Y. Karnik, A.G. Stefanopoulou, J. Sun, Journal of Power Sources 164 (2007) 590–605.
[13] J.E. Dennis Jr., R. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice Hall, Inc., Englewood Cliffs, NJ, 1983.
[14] T.E. Springer, T.A. Zawodzinski, S. Gottesfeld, Journal of Electrochemical Society 138 (8) (1991) 2334–2342.
[15] S.D. Knights, K.M. Colbow, J. St-Pierre, D.P. Wilkinson, Journal of Power Sources 127 (2004) 127–134.