Low-Cost Programmable Battery Dischargers and Application in Battery Model Identification

Karsten Propp, Abbas Fotouhi and Daniel J. Auger

School of Aerospace, Transport and Manufacturing
Cranfield University
College Road, Cranfield, Beds., MK43 0AL, UK
Email: d.j.auger@cranfield.ac.uk

Abstract—This paper describes a study where a low-cost programmable battery discharger was built from basic electronic components, the popular MATLAB programming environment, and an low-cost Arduino microcontroller board. After its components and their function are explained in detail, a case study is performed to evaluate the discharger's performance. The setup is principally suitable for any type of battery cell or small packs. Here a 7.2 V NiMH battery pack including six cells is used. Consecutive discharge current pulses are applied and the terminal voltage is measured as the output. With the measured data, battery model identification is performed using a simple equivalent circuit model containing the open circuit voltage and the internal resistance. The identification results are then tested by repeating similar tests. Consistent results demonstrate accuracy of the identified battery parameters, which also confirms the quality of the measurement. Furthermore, it is demonstrated that the identification method is fast enough to be used in real-time applications.

Keywords—NiMH battery; model identification; Arduino; current sink.

I. INTRODUCTION

For the development of advanced electric vehicles, energy storage technology plays a critical role. To improve the battery performance, researches not only focus on the cell chemistry and new battery technologies, but also on ways of optimal battery use. To get optimal performance and lifetime of a battery pack, it is very useful to know its internal states precisely. For this purpose a wide range of battery models have been developed, simulating different physical phenomena with varying demands in precision and insight into the internal processes [1].

Since the computational power of a typical electronic control unit (ECU) or battery management system (BMS) is limited, simple low-complexity battery models are often needed. Examples of such simplified models are equivalent electrical circuits, which reproduce the transient behaviour of a battery with electrical components like resistors, capacitors and a voltage source [2]. Though these models are not able to give insight into the batteries internal chemical processes, they can help to estimate its internal states, such as state of charge (SoC) and state of health (SoH), with relatively low computational effort and simple measurements of current and terminal voltage. Usually these models use current as an input, representing the different torque requirements in a drive cycle, and the terminal battery voltage as an output. Estimation of the internal

state is possible due to the varying battery output voltage behaviour over the charge/discharge range and age, determinable through changing values for resistances, capacities and voltage source.

To compare the models with real batteries, but also to parameterize them and test estimation algorithms, discharge and charge tests are performed. As with the models, common inputs and outputs for these tests are load current and battery terminal voltage respectively. Typically, experimental layouts contain a physical charge/discharge facility, a data acquisition system and a host computer setting current demands and logging the measurements of voltages, currents, Ampere hours and Watt hours. In principle, this layout is independent of the cell chemistry or battery pack size. Examples for lead-acid [3], $LiFePo_4$ [4] $LiMnO_2$ [5], and LiPB [6][7] and otherwise unspecified Li-Ion batteries [8][9] exist in the literature. The proposed experimental layouts have one more similarity: The layout focusses on accuracy and depends on complex and expensive components. This makes good sense for a large-scale commercial operation, but it is less well suited to quick turnaround bench-top experimentation, and the costs may restrict access.

In this study, a programmable battery discharger is built and tested that focusses on simplicity and low cost. The proposed hardware gives an opportunity to supplement established laboratory equipment with an easy-to-build battery discharger. The proposed setup is suitable for scaled-down battery tests such as single cell or small battery pack tests. The main components are the widely-available MATLAB programming environment, an Arduino Uno low-cost processor board, and a simple current sink constructed from readily-available electronics components; these will be explained in detail. The equipment is demonstrated in use on a 7.2 V NiMH battery pack including six cells.

To further demonstrate the equipment's usefulness, an example of algorithmic work is presented. Battery model identification is performed to parameterize an equivalent circuit network (ECN) model for a NiMH battery pack. Equivalent circuit network modelling is one of the most common battery modelling approaches especially for real-time application. Having manageable complexity – much less than a full-blown electrochemical model – ECN mode models have been used in a wide range of applications and various types of batteries [10][11]. The simplest form of an ECN battery model is a

source-resistance model [12] which is used here because of its simplicity and speed for real-time applications. After that, the identification results are analysed and a simple look-up table model is presented for the NiMH battery pack.

II. TEST BENCH HARDWARE

Our chosen layout includes a current sink for physical discharge and a microcontroller used as data acquisition device to communicate with MATLAB. The demanded currents and the voltage measurements are processed by a MATLAB function sending, receiving data and storing them in the form of column vectors for time, current and voltages.

A. Current Sink

The current sink controls the discharge current of the battery independently of its terminal voltage by automatically adjusting the effective resistive load [13]. The circuit consists of a power resistor (as a shunt) and a MOSFET transistor (IRF630) acting as variable resistor, connected in series between the battery poles. The current is controlled through the output voltage V_{out} of the operational amplifier, changing the resistance of the MOSFET in that way that V_1 is equal to V_2. Therefore, the control voltage V_1 also is equal to the current in the circuit due to the used resistor of one ohm (Fig.1). In other words, a control voltage of 0-5V will generate equivalent currents of 0-5A (see [14] for more details). As supply for the operational amplifier, a computer power supply is used because of its stable +/-12V output voltages. Two standard (LT1001) operational amplifiers are used in parallel, to reduce noise effects with higher currents.

B. Arduino Uno

Communication between host computer and current sink is realized via an Arduino Uno microcontroller [15]. The Arduino is an open source board for microcontrollers with a wide range of appliations and easy-to-use MATLAB/Simulink connectivity. Voltages up to 5 V can be measured and created through the A/D converter inputs and the PWM outputs. For the proposed current sink, the PWM signal (8 bit) generates the control voltage V_1, which sets the current, and the A/D converter (10 bit) measures the battery terminal voltage and the voltage across the one ohm resistor. As mentioned before, the potential drop over the resistor represents the current. (This means we can use the actual current rather than its theoretical value.) The voltages generated and measured by the Arduino itself are limited to 5 V which therfore limit the current to the same value when no more components are used.
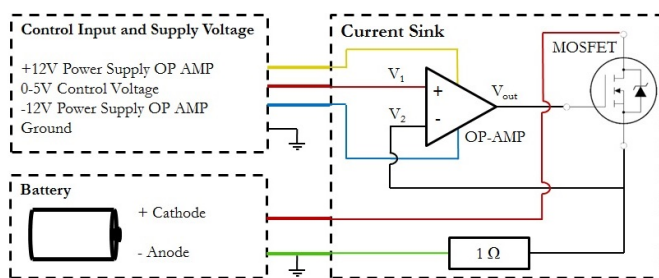

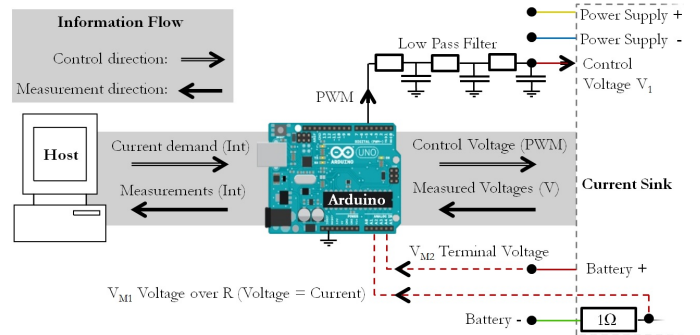Fig. 1: Circuit of the physical current sink

Fig. 2: Communication between host computer, Arduino and the current sink

C. MATLAB/Simulink

To connect the microcontroller board to MATLAB the Arduino-IO Package is used [16], generating a server on the microcontroller. When MATLAB sends commands to the Arduino via the serial USB connection, signals and measurements can be generated or read and directly used in the MATLAB workspace. A practical example for the communication with Simulink is given here [17].

III. TEST BENCH SOFTWARE

The software manages the interaction of the three components. The information flow, shown in Fig. 2, contains the current demand, which is transferred to the control voltage V_1, and the measurements V_{M1}, V_{M2}, which are converted to appropriately-scaled integers and send to MATLAB.

A. MATLAB Software

In the programming environment, a current demand is generated as a row vector of integers from 0 to 255 (0 to 5 A), each element representing a single time step. A simple and flexible way to create this vector is to use a Simulink model, shown in Fig. 3. Since the signal needs to be smoothed for the current sink, a low pass filter is used between the PWM output and control voltage input.

The Arduino-IO Package allows the interaction with the microcontroller directly with MATLAB code. During measurement, a MATLAB function sends the current demand, receives the voltage measurements and stores the values in a column vector once per second. This loop will continue until a time limit has been reached or other conditions like minimal battery terminal voltages are fulfilled. The main procedure of the loop is given by:
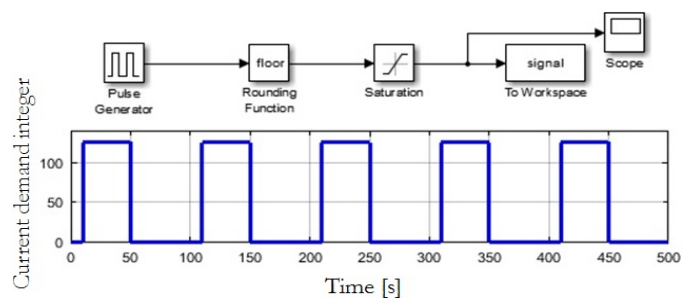

Fig. 3: Simulink model for current demand vector. Sample time signal = 1s

- Read the *i*th value from the Simulink current demand vector and send the corresponding integer to the Arduino microcontroller board via USB.

- Ask for the measurement of the voltage over the power resistor, calculate the current and store the value.

- Ask for the measurement of the battery terminal voltage, calculate the voltage and store the value.

- If the battery terminal voltage is too low: STOP

It is simple to add more conditions or calculations to the loop. Here however, only the very basic principle is shown.

B. Arduino Microcontroller Board

The program on the microcontroller is provided by the MATLAB Arduino-IO Package and automatically connects to the host computer. It can be executed, and when it does a loop constantly checks the USB port for commands from MATLAB [18]. When the mentioned loop on the host computer is running, the following tasks will be executed:

- Get an integer from USB port and create the appropriate PWM signal for the control voltage.

- Read the voltage from A/D port one, converts it into an integer, and send this integer (which represents the voltage over the power resistor) to MATLAB.

- Reads the voltage from A/D port two, convert it into an integer, and send this integer (which represents battery terminal voltage) to MATLAB.

C. Current Sink

Since there is no software added in the current sink, only the main principle of the current sink is described here. The operational amplifier receives the smoothed control voltage V_1 and sets the current to the same value.

Summarised, the processes are:

- Receiving the control voltage from Arduino board

- Equalizing V_1 and V_2 to set the right current.

IV. EXPERIMENTS

As a case study, a six-cell pack of NiMH batteries was tested using the proposed test bench. The battery pack was selected due to its simple and save handling as well as its convenient output voltage. Specifications of the battery pack are listed in Table I. The experiment was conducted by applying consecutive discharge current pulses to the battery and measuring the terminal voltage as the output. Since the maximal terminal voltage is outside the measurement range of 5 V, a voltage divider is applied. The whole test bench is shown in Fig. 4.

TABLE I: NiMH battery pack specifications

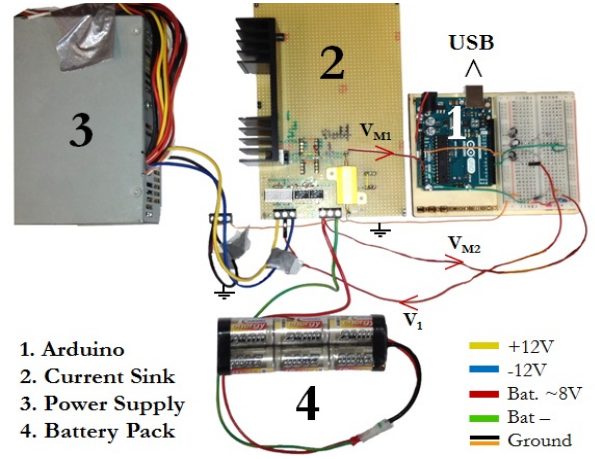| Parameter	Value
Rated capacity per cell	2400 mAh
No. of cells	6
Rated voltage	7.2 V
Full-Charged voltage	8.5 V
Cut-off voltage	6 V

Fig. 4: Battery discharge test bench + NiMH battery pack

The measurement vector includes voltage, current and time values so that other parameters of interest, such as ampere-hours, can be calculated out of the data quickly. Furthermore, modifications like more precise sensors or added temperature sensors are possible as well.

As an example, Fig. 5 shows a 1000-second segment of an experiment. By looking closely, the data is a bit noisy for high precision goals. However, our results demonstrate that it is good enough for battery model identification. In addition, the lower accuracy of the proposed test bench is compensated by its low costs, its safe performance due to low voltages and easy handling without the need for additional software.

V. BATTERY MODEL IDENTIFICATION

A very simple 'source plus internal resistance' battery model [12] is used for system identification. The model includes an ideal voltage source V_{OC} and a resistance R_0 as depicted in Fig. 6 where V_t is battery terminal voltage and I_L is load current. The reason this model was chosen is its simplicity and speed which mean it can be used in real-time applications. Using an efficient and fast model, battery model identification and SoC estimation can be performed in real-time.
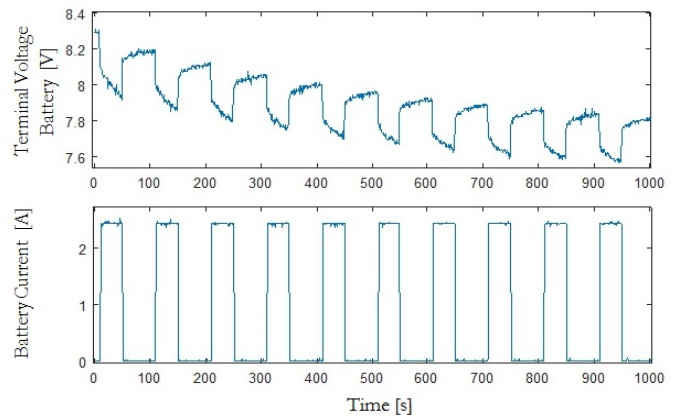

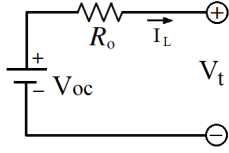Fig. 5: 1000 s extract of discharge test

Fig. 6: Cell model for internal resistance identification

The Prediction-Error Minimization (PEM) algorithm [19] is utilized as the identification algorithm. PEM algorithm is fast enough to be used in real-time battery management system (BMS) for battery model identification. In the identification procedure, the model parameter vector θ is determined so that the prediction error ε is minimized. The error is defined as follows:

$$\varepsilon(t_k, \theta) = y(t_k) - \hat{y}(t_k | t_{k-1}; \theta) \tag{1}$$

Where $y(t_k)$ is the real output at time k and $\hat{y}(t_k|t_{k-1};\theta)$ is predicted value of the output at time k using the parameters θ. The prediction error depends on the parameter vector, so an iterative minimization procedure has to be applied. Consequently a scalar fitness function is minimized as follows:

$$E_N(\theta) = \det\left(\frac{1}{N} \sum_{k=1}^{N} \varepsilon(t_k, \theta)\, \varepsilon^T(t_k, \theta) \right) \tag{2}$$

In this study, the parameters vector contains cell's internal resistance and OCV. The parameters are optimized so that the least difference between measured terminal voltage V_t and model's output is achieved.

$$\theta = [R_O, V_{OC}] \tag{3}$$

$$\varepsilon(t_k, \theta) = V_t(t_k) - \hat{V}_t(t_k | t_{k-1}; \theta) \tag{4}$$

The battery's parameters vary with the state-of-charge (which is constantly changing as current is drawn), so the model's parameters need to be updated in regular time intervals or SoC intervals. Consequently, a short history of battery charge/discharge inside the identification window is used in the calculations. Because of the application of this study in EV energy storage system, SoC identification interval is preferred. Here an identification window of 3% SoC is used which means that the battery model's parameters are obtained every 3 % change in charge level. Considering the rated capacity of 2.4Ah for the whole pack, the identification is repeated every 72 mAh capacity use during the test.

The identified values of parameters V_{OC} and R_0 during three separate experiments are presented in Fig. 7 and Fig. 8 respectively. The experiment is repeated in order to investigate the consistency of the results.

Table II contains the model parameters at different SoC which can be used as a look-up table battery model.

In addition, the proposed measurement and identification techniques are usable for real-time battery SoC estimation due to their simplicity and speed. Fig. 9 demonstrates the identification time during the experiments. Using a CPU at 2.10 GHz, the average identification time is around half a second and the maximum time is not more than 0.8 seconds.

VI. DISCUSSION

As shown above, the measurements follow a stable pattern. In addition to the general agreement of the measurements with previous studies on NiMH battery parameterisation [20], the results do appear useful, providing a consistent way to estimate model parameters.

The measurements do appear slightly noisy. However, even if the accuracy of the taken measurements is not directly comparable with professional laboratory equipment and the measurement noise is more pronounced, the test bench that has been presented has the potential to supplement existing equipment due to its low cost, safe handling and flexible structure. Furthermore, noisy measurements are a common issue in the field of battery state estimation and can be handled well through statistical algorithms like the Kalman Filter. Any practical application is likely to suffer from noise, so it is pragmatic to develop algorithms that can cope with it. The specific layout of the current sink, directly interacting with the MATLAB environment, can also help to simplify the implementation of these algorithms.
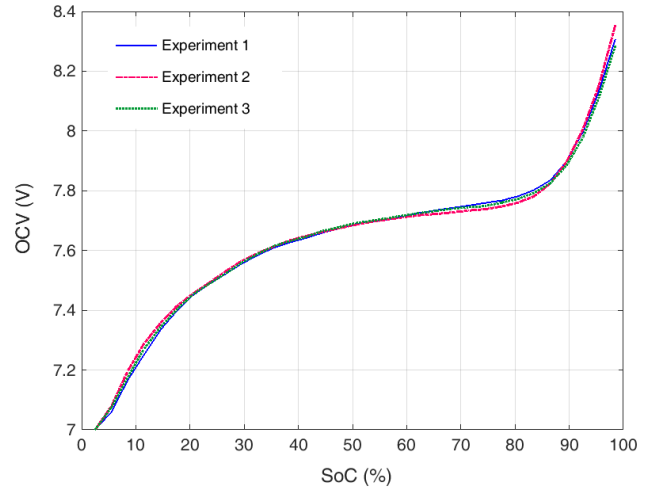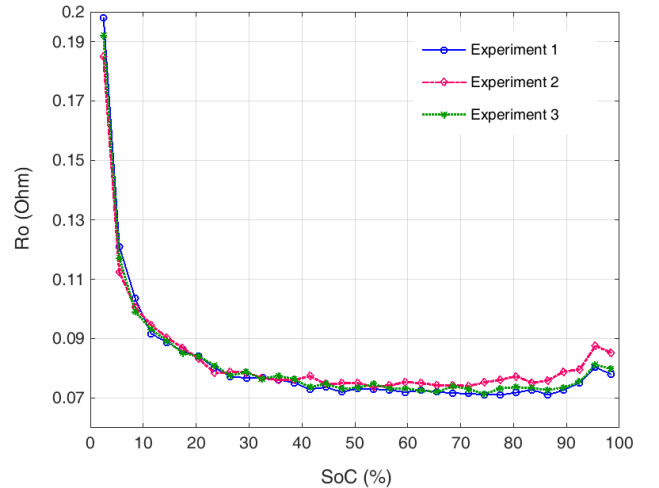

Fig. 7: NiMH battery OCV curve vs. SoC

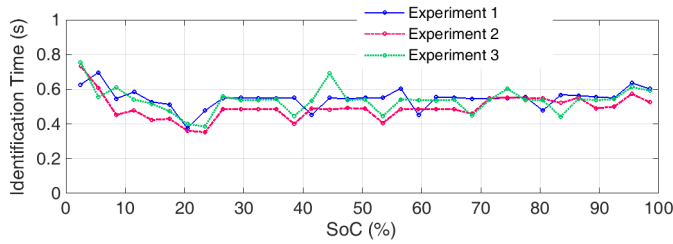Fig. 8: NiMH battery internal resistance curve vs. SoC

Fig. 9: Battery model identification time during the experiments

As a part of the REVB project our research focusses on the development of state estimation algorithms for lithium sulphur batteries, capable of managing the harsh automotive environment. After adding a charge functionality to simulate vehicle drive cycles and optimizing the measurement accuracy, the test bench should be used to test algorithms for online parameter estimation, state of charge or state of health estimation under the influence of noise, before the validation with professional hardware.

TABLE II: NiMH battery model parameters at different SoC

SoC (%)	OCV (V)	Ro (milliohm)
98	8.31	77.95
95	8.13	80.31
92	7.99	75.16
89	7.90	72.65
86	7.83	71.11
83	7.80	72.80
80	7.78	71.84
77	7.77	71.06
74	7.76	71.16
71	7.75	71.48
68	7.74	71.69
65	7.74	72.07
62	7.73	72.61
59	7.71	71.95
56	7.70	72.75
53	7.70	73.03
50	7.69	73.16
47	7.67	72.03
44	7.66	73.66
41	7.64	72.94
38	7.63	75.13
35	7.61	76.24
32	7.58	76.86
29	7.56	76.73
26	7.52	77.28
23	7.49	80.20
20	7.45	84.02
17	7.40	85.87
14	7.33	88.69
11	7.25	91.67
8	7.17	103.51
5	7.06	121.05
2	7.00	198.13

VII. Conclusions

This paper has described a novel battery test bench based on low cost hardware and basic electronic components. The test bench uses the MATLAB programming environment and an Arduino Uno microcontroller to set the current demand and receive actual voltage and current measurements; the current demand is converted into a physical current using a simple custom-built current sink.

The usefulness of this approach has been shown with a parameter estimation study carried out on a six cell NiMH battery pack. Due to the stable measurements and comparable results in the literature, the test bench has enough potential to be used for estimation techniques for other cell chemistries.

Acknowledgment

References

[1] V. Ramadesigan, P. W. C. Northrop, S. De, S. Santhanagopalan, R. D. Braatz and V. R. Subramaniana, "Modeling and Simulation of Lithium-Ion Batteries from a Systems Engineering Perspective," *J. Electrochem. Soc.*159, no. 3 (2012): R31–45. doi: 10.1149/2.018203jes.

[2] X. Hu, S. Li and H. Peng, "A Comparative Study of Equivalent Circuit Models for Li-Ion Batteries," *J. Power Sources* 198 (2012): 359–67. doi: 10.1016/j.jpowsour.2011.10.013.

[3] B. S. Bhangu, P. Bentley, D. A. Stone and C. M. Bingham, "Nonlinear Observers for Predicting State-of-Charge and State-of-Health of Lead-Acid Batteries for Hybrid-Electric Vehicles," *IEEE Trans. Veh. Technol.* 54, no. 3 (2005): 783–94. doi: 10.1109/TVT.2004.842461.

[4] H. He, R. Xiong and H.Guo, "Online Estimation of Model Parameters and State-of-Charge of LiFePO4 Batteries," *Applied Energy*, 89, no.1 (2012): 413–20. doi: 10.1016/j.apenergy.2011.08.005

[5] R. Xiong, H. He, F. Sun and K. Zhao, "Evaluation on State of Charge Estimation of Batteries With Adaptive Extended Kalman Filter by Experiment Approach," *IEEE Trans. Veh. Technol.* 62, no. 1 (2013): 108–17. doi: 10.1109/TVT.2012.2222684.

[6] Z. Chen, Y. Fu and C. C. Mi, "State of Charge Estimation of Lithium-Ion Batteries in Electric Drive Vehicles Using Extended Kalman Filtering," *IEEE Trans. Veh. Technol.* 2013, Vol. 62, no. 3 (2013): 1020–30. doi: 10.1109/TVT.2012.2235474.

[7] M. Chen and G. A. Rincon-Mora, "Accurate Electrical Battery Model Capable of Predicting Runtime and I-V Performance" *IEEE Trans. Energy Conv.* 21, no. 2 (2006): 504–11. doi: 10.1109/TEC.2006.874229.

[8] J. Kim, S. Lee and B. Cho, "Determination of State of Charge Based on Extended Kalman Filter Using Per-Unit System and Time Constant Principle," in *Proc. 31st Int. Telecommunications Energy Conf.: INTELEC 2009, Incheon, Oct. 2009.* 1–6. doi: 10.1109/INTLEC.2009.5352049.

[9] H. He, R. Xiong, X. Zhang, F. Sun, and J. X. Fan, "State-of-Charge Estimation of the Lithium-Ion Battery Using an Adaptive Extended Kalman Filter Based on an Improved Thevenin Model," *IEEE Trans. Veh. Technol.* 60, no. 4 (2011): 1461–9. doi: 10.1109/TVT.2011.2132812.

[10] T. Marc, B. Oliver and U. S. Dirk, "Development of a Voltage-Behavior Model for NiMH Batteries Using an Impedance-Based

Modeling Concept," *J. Power Sources* 175, no. 1 (2008): 635–43. doi: 10.1016/j.jpowsour.2007.08.039.

[11] H. He, R. Xiong and J. Fan, "Evaluation of Lithium-Ion Battery Equivalent Circuit Models for State of Charge Estimation by an Experimental Approach," *Energies* 4, no. 4 (2011): 582–98. doi: 10.3390/en4040582.

[12] V. H. Johnson, "Battery Performance Models in ADVISOR," *J. Power Sources* 110, no. 2 (2002): 321–9. doi: 10.1016/S0378-7753(02)00194-5

[13] E. Holland, "Everything Including the Kitchen (Current) Sink!" in *Rants From the Embedded Hardware Guy* (blog), July 23, 2011, http://embeddederic.blogspot.co.uk/2011/07/everything-including-kitchen-current.html. [Accessed: Apr. 30, 2015.]

[14] D. L. Jones, "DIY Constant Current Dummy Load for Power Supply and Battery Testing," in *Electronics Engineering Video Blog*, Aug. 1, 2010. http://www.eevblog.com/2010/08/01/eevblog-102-diy-constant-current-dummy-load-for-power-supply-and-battery-testing. [Accessed: Apr. 30, 2015.]

[15] Arduino website, http://www.arduino.cc. [Accessed: Apr. 17, 2015.]

[16] MathWorks, "MATLAB Support for Arduino (aka ArduinoIO Package)" (File Exchange post), July 28, 2011, updated Sept. 5, 2014, http://www.mathworks.com/matlabcentral/fileexchange/32374-matlab-support-for-arduino--aka-arduinoio-package-. [Accessed: May 6, 2015.]

[17] R. Barber, M. De la Horra Köllmer and J. Crespo, "Control Practices using Simulink with Arduino as Low Cost Hardware," in *Proc. 10th IFAC Symposium Advances in Control Education, Sheffield, Aug. 2013*. 250–5. doi: 10.3182/20130828-3-UK-2039.00061.

[18] G. Campa, "Arduino IO Package: Slides and Examples," (MATLAB File Exchange post), June 6, 2010, updated Apr. 9, 2014, http://www.mathworks.com/matlabcentral/fileexchange/27843-arduino-io-package--slides-and-examples. [Accessed: May 6, 2015.]

[19] L. J. Ljung, *System Identification: Theory for the User*. 2nd ed. New York: Prentice Hall, 1999.

[20] R. Bornatico, A. Storti, L. Mandrioli, A. Zappavigna, Y. Guezennec and G. Rizzoni. "NIMH Battery Characterization and State of Charge Estimation for HEV Applications," *Proc. ASME Int. Mechanical Engineering Congress and Exposition, Seattle, 2007*. 205–14. doi: 10.1115/IMECE2007-42484.

2015-09-25

Low-cost programmable battery dischargers and application in battery model identification

Propp, Karsten

IInstitute of Electrical and Electronics Engineers