Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institut für
Technische Informatik und
Kommunikationsnetze

NiMH Battery State-of-Charge Approximation

SEMESTER THESIS

Mathis Steichen

Advisors: Bernhard Buchli
Felix Sutton

Professor: Prof. Dr. Lothar Thiele

Computer Engineering and Networks Laboratory
Department of Information Technology and Electrical Engineering
ETH Zürich

June 26, 2013

Acknowledgement

I would like to thank Prof. Dr. Lothar Thiele and his research group for giving me the opportunity to conduct this semester thesis.

Furthermore I would like to express my deepest gratitude to Bernhard Buchli and Felix Sutton for providing me with this interesting thesis, for the wonderful aid that they gave me and for their interest in this work.

Finally I would like to thank my family for their everlasting support.

Abstract

The goal of this thesis is to provide state-of-charge information for a nickel-metal hydride battery during the discharge process. Furthermore the resource constraints of a commonly used embedded system are considered. The reasons for using such a battery are as follows.
The, for this purpose, most important advantages of a nickel-metal hydride battery are their comparably high capacity, large energy density and long cycle life [8]. The disadvantages, as stated in [8], are a reduced performance at higher currents, a high self-discharge rate and the reduction of the capacity due to the memory effect.

Recent research [4] has provided a state-of-charge algorithm that on one hand does not rely on complex hardware and on the other hand is lightweight in terms of computational resources. Moreover it has been shown that this algorithm yields precise results in combination with lead acid batteries [4].
As shown in this thesis the mentioned algorithm yields imprecise results for the discharge process of a nickel-metal hydride battery if the computational complexity is to be kept low. The mentioned algorithm is therefore modified to consider the discharge process of the named battery and the resource constraints.
The performed adaptation consists in the introduction of a threshold voltage that divides the state-of-charge into two segments that are then approximated in a way similar to the one described in [4]. For this purpose execution times of polynomial evaluations and approximations of the depth-of-discharge are weighted against each other. Thereby a satisfying approximation of the state-of-charge without substantially increasing the resource demand of the modified algorithm is obtained. This results in the use of cubic polynomials to approximate the two mentioned segments of the state-of-charge.

Finally two different discharge processes that employ either a constant or a time-dependent load are conducted and discussed. A method named coulomb counting is used to evaluate the obtained state-of-charge approximations. For verification purposes this step is conducted for the mentioned nickel-metal hydride battery and for a lead acid battery which has been considered in [4]. Thereby results that the modified algorithm is capable of approximating the state-of-charge of a nickel-metal hydride battery with a reasonable accuracy. Still the resulting state-of-charge shows a non-linear behavior that is not present in the state-of-charge of the lead acid battery. These variations are although expected and result from the decisions made in order to keep the modified algorithm's resource requirements low.

Contents

1: Introduction *1*
1.1 Motivation . . . 1
1.2 Assumptions . . . 2
1.3 Contribution . . . 2
1.4 Outline . . . 3

2: Background *5*
2.1 State-of-Charge Approximation . . . 5
2.2 Differences to Existing Work . . . 6

3: Hardware Design and Implementation *9*
3.1 Functional Requirements . . . 10
3.2 Architecture . . . 10
3.2.1 TinyNode184 . . . 11
3.2.2 Voltage Measurement . . . 12
3.2.3 Voltage References . . . 13
3.2.4 Current Measurement . . . 14
3.2.5 Time-Dependent Load . . . 15
3.2.6 Communication . . . 16

4: State-of-Charge Algorithm and Implementation *17*
4.1 Overview of the state-of-charge Algorithm . . . 17
4.2 Resource Constrained Parameter Extraction and Optimization . . . 19
4.2.1 Discharge Profiles . . . 20
4.2.2 Data Approximation: First Approach . . . 22
4.2.3 Data Approximation: Approach 2 . . . 27

5: Evaluation Method *37*
5.1 System Setup . . . 37
5.1.1 Finite State Machine for Current Determination . . . 37
5.1.2 Parameter Sets . . . 38
5.1.3 Results and Discussion . . . 40

6: Conclusion and Further Work *47*
6.1 Present Implementation 47
6.2 Performance Evaluation of the adapted State-of-Charge Algorithm . . 48
6.3 Battery Differences 49

A: Lead Acid Battery *51*

B: Hardware Schematic *55*

C: Smoothing Capacitance *57*

D: Further Plots *59*

Tables

3-1 Resistor and capacitor values for the voltage divider as well as the voltage conversion value (CV) and the filter time constant. $I_{VD,DC}$ is the current drawn by the voltage divider for a DC voltage of 4.2 V for the NiMH battery and 4.4 V for the lead acid battery. 13

3-2 Resistive values for the resistive loads for both batteries. 15

3-3 Representation of the four different loads used in combination with the NiMH battery. The table also contains the hardware employed to generate those and the current they draw. Those currents have been measured at approximately 3.8 V with consideration of the effect of the measurement device on the system. 16

4-1 Execution times for quadratic and cubic polynomial evaluation on the MSP430. 20

4-2 Remaining capacity of the NiMH battery for different constant loads at 3.6 V. The initial battery capacity is given as 2200 mAh. 21

4-3 Comparison of coefficients of cubic approximations ($\widehat{DOD}(V_{bat}) = a_3 \cdot V_{bat}^3 + a_2 \cdot V_{bat}^2 + a_1 \cdot V_{bat} + a_0$) of the discharge profile (for a load of 60 mA) under different modifications. 24

4-4 Maximal and minimal values for the $\widehat{SOC}$ and $\widehat{DOD}$. These values result from the depth-of-discharge scaling and are used in combination with the first approach. 24

4-5 Root mean square errors for different approximation polynomials using floating point and integer data types. The maximum depth-of-discharge is used as a reference for comparison purposes. 25

4-6 Cubic fitting of the depth-of-discharge with the coefficients a_k and the quadratic approximation of a_k with $b_{l,k}$. The $\widehat{DOD}$ is given as a function of $V_{bat} - V_{cutoff}$ for all five loads. 26

4-7 Polynomial approximation of the threshold voltage V_{thresh} such that $V_{thresh} \simeq c_2 \cdot I_{bat}^2 + c_1 \cdot I_{bat} + c_0$. 30

4-8 Root-mean-square errors for the upper and the lower part of the polynomial approximation. The polynomials are evaluated using floating point numbers and are approximated using weighted polynomial least squares fitting. Again the maximal depth-of-discharge is used as a reference. 34

4-9 Coefficients of the upper and lower polynomials for all five loads. These are determined with use of the values given in Table 4-10. The $\widehat{DOD}$ is given as a function of $V_{bat} - V_{cutoff}$. 35

4-10 Maximal and minimal values for the $\widehat{SOC}$ and $\widehat{DOD}$. These values are used in combination with the second approach. 35

5-1 Parameters used with the state-of-charge algorithm for the NiMH battery. The coefficients $b_{l,k,upper}$ are used to compute a_k for the upper part of the depth-of-discharge. Accordingly $b_{l,k,lower}$ is used to obtain the coefficients a_k for the lower part of the depth-of-discharge. The parameters c_i are used to compute the threshold voltage V_{thresh}. 39

5-2 Parameters used with the state-of-charge algorithm for the NiMH battery. 40

5-3 Threshold voltages corresponding to the loads used to obtain Figure 5-7. The instant $t_{V_{thresh},first}$ denotes the first time instant at which the threshold voltage has been reached. Accordingly $t_{V_{thresh},last}$ denotes the last time instant at which the threshold voltage is overstepped. 46

A-1 Representation of the four different loads used in combination with the lead acid battery. The table also contains the hardware employed to generate those and the current they draw. Those currents have been measured at approximately 4 V with consideration of the effect of the measurement device on the system. 54

Figures

1-1 Considered embedded system coupled to a battery and an energy harvesting system. 1

2-1 The calculation of the state-of-charge is performed entirely in software. The only required physical value is the battery voltage V_{bat}. An overview of TinyOS is given in [2]. 6

3-1 Photograph of the hardware setup. 9
3-2 Block diagram containing all hardware components and the connections in-between them. Additionally the sections describing those components are given next to the respective element. 11
3-3 Photograph of a TinyNode184 and an attached extension board. 12
3-4 Voltage divider with an additional smoothing capacitance. 12
3-5 Upper and lower voltage references with output capacitances. 13
3-6 Current measurement device. This component determines the voltage at the terminals of the R_{SENSE} resistor and amplifies it by ten. 14
3-7 Time-dependent load implemented with resistors and a field effect transistor (FET) directly controlled by the MSP430. 15
3-8 Serial communication. 16

4-1 State-of-Charge parameter extraction and optimization. 18
4-2 Parameter extraction steps. 19
4-3 Discharge profiles of the NiMH battery at currents of 60 mA, 50 mA, 40 mA, 30 mA and 20 mA respectively. 21
4-4 Depth-of-discharge of the NiMH battery with a load of 60 mA. 27
4-5 Visualization of thresholds. 28
4-6 Derivatives of discharge profiles. 29
4-7 Dependence of thresholds from the current. 30
4-8 Depth-of-discharge and different approximations for the 60 mA load. . 31
4-9 Coefficients for the computation of the terms of the upper polynomial. 32
4-10 Coefficients for the computation of the terms of the lower polynomial. 33

5-1 Current determining state machine. 38

5-2 LA battery powered state-of-charge calculation with a constant load of approximately 57.7 mA. 41
5-3 LA battery voltage corresponding to the state-of-charge approximation depicted in Figure 5-2. 42
5-4 LA battery powered state-of-charge calculation with a variable load. . 43
5-5 NiMH battery powered state-of-charge calculation with a constant load of approximately 44.5 mA. 44
5-6 NiMH battery voltage corresponding to the state-of-charge approximation depicted in Figure 5-5. 45
5-7 NiMH battery powered state-of-charge calculation with a variable load. 46

A-1 Discharge profiles of the lead acid battery at currents of 60 mA, 40 mA and 20 mA respectively. 51
A-2 Approximation of the lead acid discharge profile at 60 mA. 52
A-3 Approximation of the lead acid discharge profile at 40 mA. 53
A-4 Approximation of the lead acid discharge profile at 20 mA. 53

B-1 Complete schematic of the hardware setup. 55

D-1 Depth-of-Discharge of the NiMH battery with a load of 50 mA. 59
D-2 Depth-of-Discharge of the NiMH battery with a load of 40 mA. 60
D-3 Depth-of-Discharge of the NiMH battery with a load of 30 mA. 60
D-4 Depth-of-Discharge of the NiMH battery with a load of 20 mA. 61
D-5 Depth-of-Discharge and different approximations for the 50 mA load. . 62
D-6 Depth-of-Discharge and different approximations for the 40 mA load. . 62
D-7 Depth-of-Discharge and different approximations for the 30 mA load. . 63
D-8 Depth-of-Discharge and different approximations for the 20 mA load. . 63

1

Introduction

1.1 Motivation

Embedded systems are used all over the world to process all kinds of information. Some of them are used as mobile devices or deployed at locations that cannot be reached often or easily due to site issues. Many of these systems are therefore not connected to a continuous power source. In this case a possibility to nevertheless operate such an entity is to use a battery. As shown in Figure 1-1 an energy harvesting system may also be added to mitigate the efforts due to battery replacement. Even with a battery and an energy harvesting system it is not guaranteed that the system will be operating at all times or until replacement of the battery is possible. Especially high energy consumers (for example the system presented in [5]) are at risk of depleting the battery when the harvested power does not meet the system requirements. This is the case if the sources (e.g: solar, wind) from which energy is harvested do not deliver enough power.
In order to increase the overall system's availability and usability it is mandatory

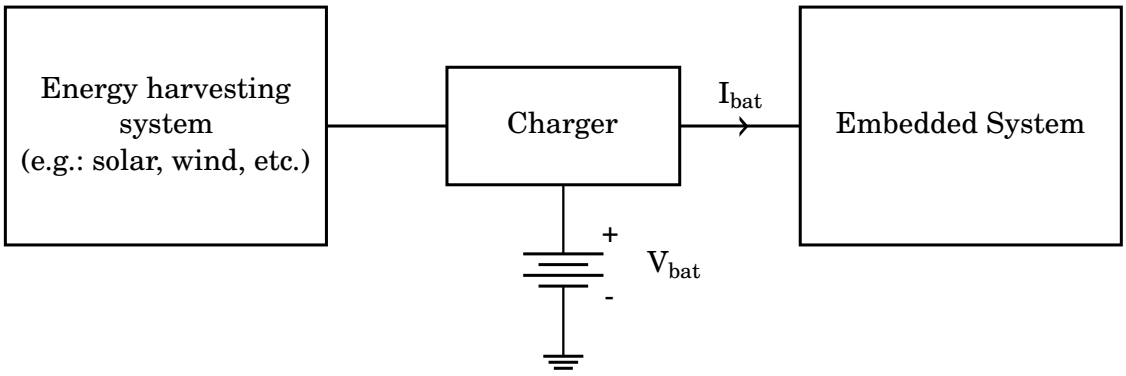

Figure 1-1: Considered embedded system coupled to a battery and an energy harvesting system.

to consider the available energy and power when making runtime decisions on the system's behavior and energy consumption. This is achievable without the need for any specialised or expensive hardware for example with the algorithm published in [4].

Furthermore, since cost and computational power of an embedded system and the modifiability of an energy harvesting system are most often limited, it is undesirable to rely on special purpose hardware or complex algorithms.

Finally the investigated battery for this thesis is a nickel-metal hydride (NiMH) battery because of the following reasons. As is written in [8] the NiMH battery is more environmentally friendly, has a higher capacity and energy density than comparable batteries. Moreover it requires less maintenance, has a long cycle life, needs less protective circuitry (for example than lithium-ion batteries) and the reduction of its capacity due to higher temperatures is not as severe as with other battery types.
Drawbacks of this battery type are the reduced performance at higher currents, the high self-discharge rate and the degradation of the capacity due to the memory effect.
Its advantages nevertheless outweigh its disadvantages, since the expected currents are comparably small and the battery is not supposed to be stored for prolonged periods of time. Furthermore, according to [8], the memory effect can to a great extent be counteracted by full discharge-charge cycles.

To recapitulate this thesis consists of the modification of a published state-of-charge algorithm [4] to account for the discharge profile of a NiMH battery and its implementation on an embedded system with limited resources.

1.2 Assumptions

In this thesis the following assumptions, based on the motivation are made:

- the needed battery information to determine the state-of-charge can be measured
- and the determination of the state-of-charge during the discharging of the battery is independent of the energy harvesting system.

1.3 Contribution

This thesis' contributions are as follows.

- Adaptation of the state-of-charge algorithm [4] to NiMH batteries.
- Implementation and verification of the modified algorithm on a resource constrained embedded system.
- Comparison of the battery types with respect to their discharge profiles and evaluation of the modified algorithm's performance.

1.4 Outline

This thesis is arranged as is stated in the following.

Chapter 2 presents the basis for the conducted work. Different possibilities to determine the state-of-charge are described. An explanation for choosing the algorithm published in [4] is also given here.

Chapter 3 describes the hardware setup. Different components are required to execute the chosen state-of-charge algorithm. The functionality implemented by those components and the choices made, as well as their justifications are contained in this chapter.

Chapter 4 precisely presents the state-of-charge algorithm [4]. Furthermore the need for modifying it is explained. Finally an adaptation that considers the discharge profiles of the used NiMH battery is elaborated. This chapter is derived under consideration of the employed, resource constrained system.

Chapter 5 evaluates the implementation of the modified state-of-charge algorithm on a resource constrained embedded system. The state-of-charge approximation conducted with a lead acid (LA) battery is used as a reference.

Finally Chapter 6 contains a summary of this semester thesis, an outlook that suggests further considerations and a comparison of the two batteries.

2

Background

2.1 State-of-Charge Approximation

Different approaches to obtain state-of-charge information exist and their advantages and drawbacks will be discussed in this section. Moreover the reason for choosing the algorithm presented in [4] is derived. The following sections are based on [4].

Dedicated Hardware. One common approach is to use dedicated hardware to determine the charge drawn from and the charge delivered to the battery. This so-called coulomb counting is done by integrating currents flowing into and out of the battery. The remaining charges are then determined by subtracting the result of the integration from the expected capacity. This approach has two disadvantages. First this method does not consider battery imperfections, as for example the dependence of the capacity from temperature and from the intensity of the drawn current. Second the determination of the state-of-charge is strongly dependent on the measurement device's precision which may lead to costly purchases if a certain accuracy needs to be reached.
[4] also mentions a technique called book-keeping that considers battery inefficiencies. A drawback of this method is the increased complexity of the implementation.

Software Battery Model. Another method is to construct battery models for possible load scenarios, thereby being able to mathematically model most of the physical properties. An advantage is that this approach allows the state-of-charge calculation to be executed in software so that only a few additional hardware components are needed. Drawbacks are the complexity of this method due to the tremendous amount of configuration possibilities and the computational power required to perform such calculations alongside the system's other functions.

Since the considered embedded systems are independent of the energy harvesting system and have limited resources, both of the previously described methods are not suited for this thesis.

State-of-Charge Algorithm Presented in [4]. This approach yields accurate results for lead acid batteries and is especially intended for wireless sensor networks

and therefore claims to be lightweight on computational resources and independent on complex hardware dedicated to accurately determine the state-of-charge, since this is done completely in software as is shown in Figure 2-1. A more precise description of the function of the state-of-charge algorithm is given in Section 4.1.

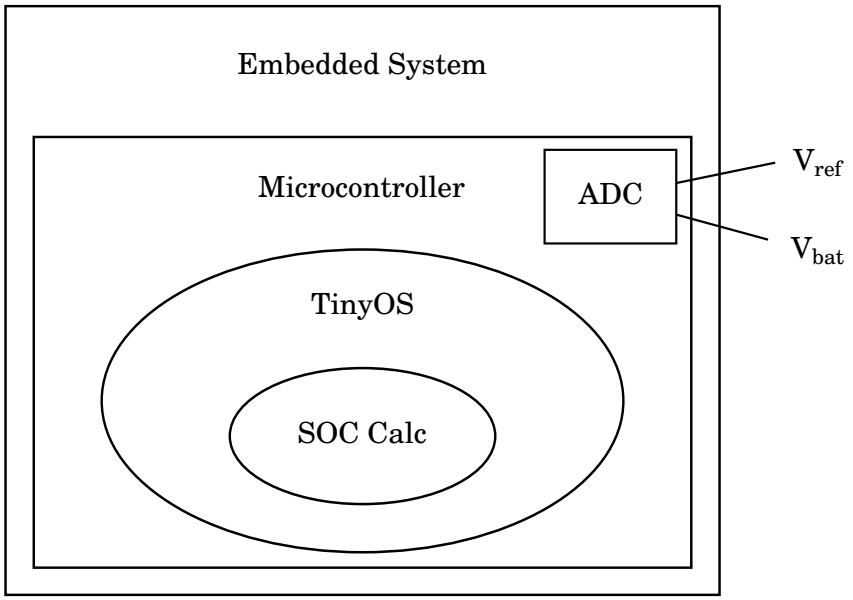

Figure 2-1: The calculation of the state-of-charge is performed entirely in software. The only required physical value is the battery voltage V_{bat}. An overview of TinyOS is given in [2].

2.2 Differences to Existing Work

The state-of-charge algorithm employed in this thesis has recently been published in [4]. It has been shown that the algorithm yields relatively accurate results using lead acid batteries. However it is yet unknown how the algorithm operates with other types of batteries. Especially since the chemical reactions taking place are different.
Hence this thesis considers a NiMH battery. As is written in Chapter 4 the algorithm previously described has to be modified in order to accurately describe the state-of-charge for this type of battery.
To verify the correctness of the algorithm and be able to interpret the results, a lead acid battery is used. The algorithm execution for such a battery is used as a reference. This allows a comparison, provided in Chapter 6, of the batteries' discharge profiles and the behavior of their state-of-charge approximations that are illustrated in Chapter 5. For this purpose the battery's current and voltage are measured with the hardware setup described in Chapter 3. Thereby state-of-charge approximations that are delivered by a battery discharge while executing the modified algorithm are obtained.

Additionally the use of limited computational resources affects the maximal allowable duration of execution of the algorithm and the precision of the results. This poses an additional challenge and therefore different possibilities to compute this state-of-charge algorithm are presented in Chapter 4. These different possibilities are evaluated with respect to measured execution times and the precision of the approximation.

3

Hardware Design and Implementation

This chapter presents the functionality that is implemented by the hardware setup depicted in Figure 3-1. Furthermore the individual components and the function provided by these elements are described.

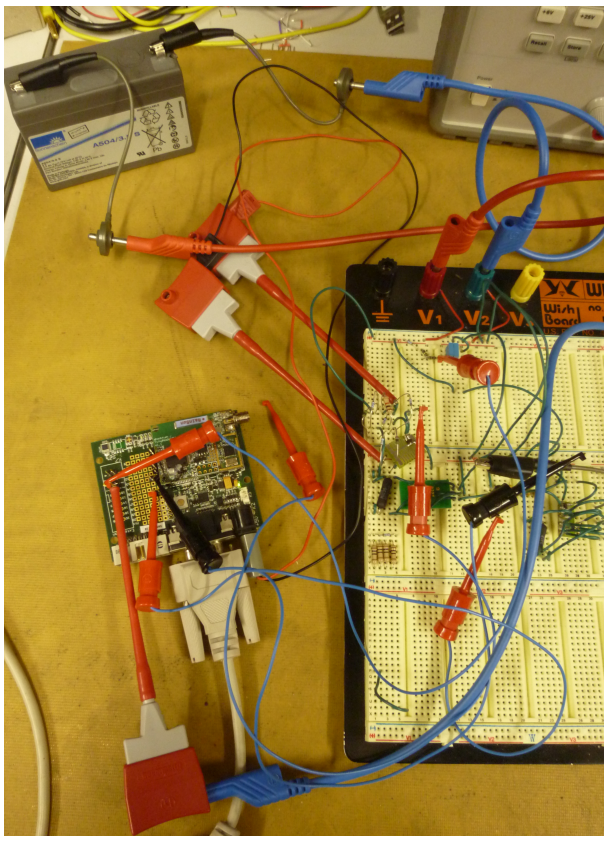

Figure 3-1: Photograph of the hardware setup.

3.1 Functional Requirements

The state-of-charge determination is performed entirely in software. Nevertheless some hardware components are required to obtain battery data to approximate it with the algorithm detailed in Chapter 4. Before describing the used components, the functionality that has to be implemented is presented.

Determination of the battery voltage. For this algorithm the battery voltage is the only indispensable value for computing the state-of-charge. Furthermore the precision with which the battery voltage is obtained determines the accuracy of the resulting approximation.

Measuring the system current. The state-of-charge is commonly computed by terms of coulomb-counting. Therefore the current is measured and integrated to evaluate the results obtained with the herein described algorithm. To avoid substantial deviations from the actually drawn charges, the measurement errors have to be as small as possible. Furthermore the current has to be measured over a wide range of values. This is given by the current demands of the considered embedded system which may range from a fey milliamperes, when the system is in a low power mode, to several tens of milliamperes, when the system performs heavy calculations in addition to the auxiliary hardware's current demand.

Evaluating the algorithm in a real case scenario. To be able to evaluate the herein modified state-of-charge algorithm under real circumstances a time-dependent load is implemented. This allows evaluation of the algorithm's performance under dynamically varying loads.

3.2 Architecture

Figure 3-2 illustrates the block diagram of the whole battery powered system. The hardware contains the digital part of the system, the voltage and current measurement devices, the voltage references and the load. These are discussed in the following.

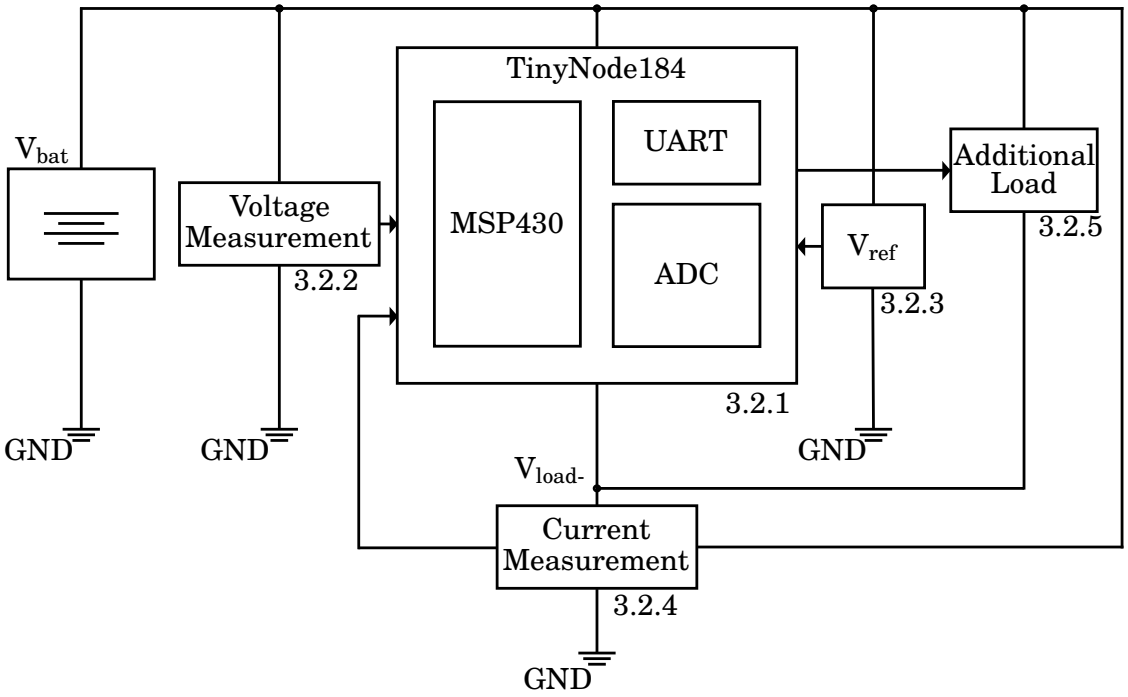

Figure 3-2: Block diagram containing all hardware components and the connections in-between them. Additionally the sections describing those components are given next to the respective element.

3.2.1 TinyNode184

The TinyNode184 [12] is an ultra-low power wireless sensor platform. The TinyNode184 is controlled by a Texas Instruments MSP430 [13] ultra-low-power microcontroller. The MSP430 consists of a 16-bit RISC architecture with a clock frequency of 16 MHz. It also contains a 12-bit Analog-to-Digital converter (ADC) with two external voltage reference inputs and several general purpose input/output (GPIO) pins.

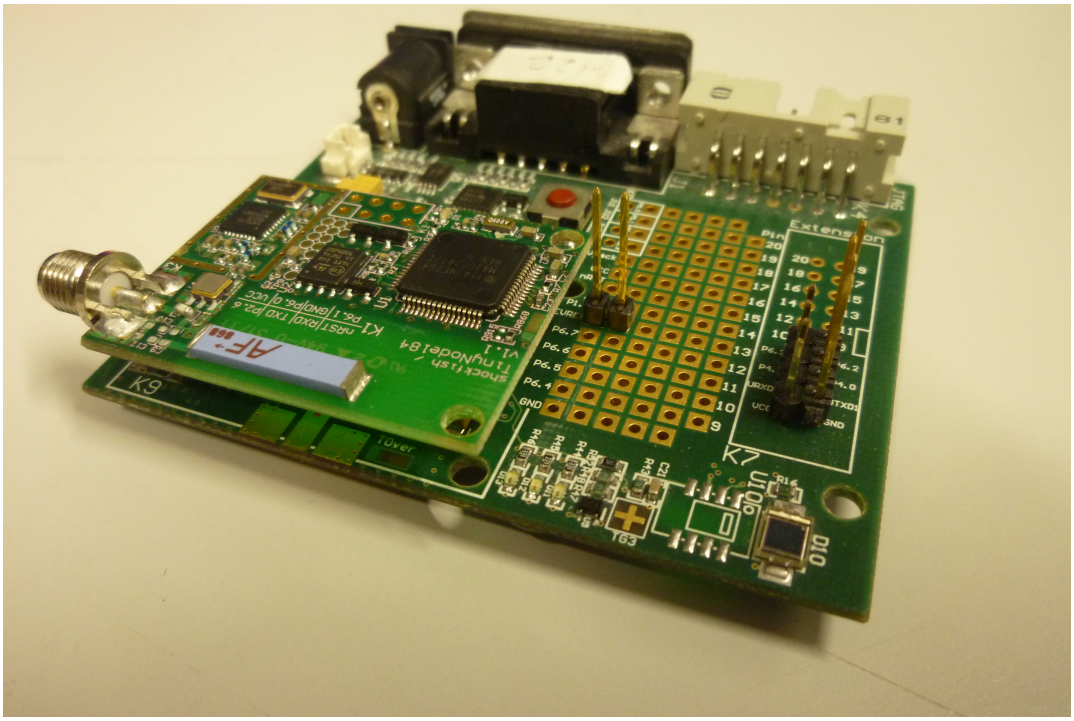

Figure 3-3: Photograph of a TinyNode184 and an attached extension board.

The TinyNode184 can be combined with an extension board that adds extra functionality to the system. The extension board includes a serial port which is used to communicate between the TinyNode184 and an attached computer. The TinyNode184 connected to such an extension board is shown in Figure 3-3.

3.2.2 Voltage Measurement

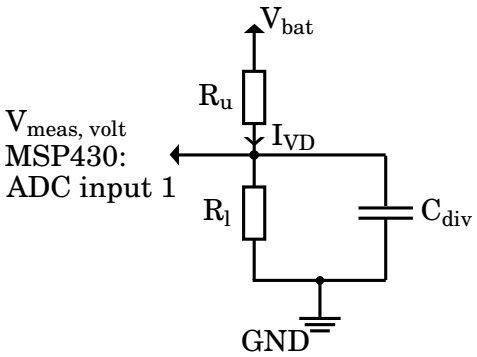

Figure 3-4: Voltage divider with an additional smoothing capacitance.

A voltage divider as depicted in Figure 3-4 maps the battery voltage V_{bat} to a voltage range suitable for the ADC. The resistors R_u and R_l are chosen so as to only allow a very small current to flow, and therefore must have resistive values in the

MΩ range. For both batteries the voltage dividers are chosen so that a high measurement resolution is attained. Therefore the highest occurring battery voltage is mapped to 2.5 V, which is the comparative value provided by the upper voltage reference. For the NiMH battery (Emmerich ready to use battery pack: 2200 mAh, 3.6 V, manufacturer part number: 255066) the maximum voltage is 4.2 V. Since the lead acid battery [3] has a maximal voltage of 4.4 V two different voltage dividers are implemented. Equation 3.1 then describes the relationship between the digital representation and the physical value of the battery voltage. The voltages $V_{ref,upper}$ and $V_{ref,lower}$ are provided by the voltage references and are described in Section 3.2.3.

$$V_{bat,ADC} = \frac{(\frac{V_{bat}}{CV} - V_{ref,lower})}{V_{ref,upper} - V_{ref,lower}} \cdot 4095 \tag{3.1}$$

The capacitor adds smoothing to the measured signal. Since voltages and currents are measured once every second (as discussed in Section 5) a filtering time constant of 500 ms which is derived in Appendix C, has been chosen. Again this time constant is the same for both batteries to compare the results presented in Chapter 5.
The used resistive and capacitive values are presented in Table 3-1.

	R_l (kΩ)	R_u (MΩ)	C_{div} (nF)	$CV = \frac{V_{bat}}{V_{meas}}$	τ (ms)	$I_{VD,DC}$ (μA)
LA	1.5	1.14	200	1.76	500	1.67
NiMH	1.5	1.02	220	1.68	500	1.67

Table 3-1: Resistor and capacitor values for the voltage divider as well as the voltage conversion value (CV) and the filter time constant. $I_{VD,DC}$ is the current drawn by the voltage divider for a DC voltage of 4.2 V for the NiMH battery and 4.4 V for the lead acid battery.

3.2.3 Voltage References

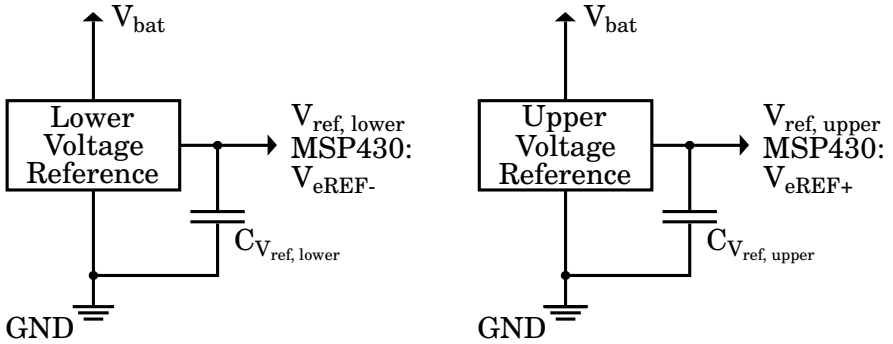

Figure 3-5: Upper and lower voltage references with output capacitances.

Two voltage references with values $V_{ref,lower} = 1.5$ V and $V_{ref,upper} = 2.5$ V are used to obtain precise conversions with a high resolution. Due to the overall system's performance dependence on the results of the ADC, it is important that the

voltage references are steady. As is shown in Figure 3-5, this is also improved by two capacitances with values of $9.4\ \mu$F for $C_{vref,lower}$ and $2\ \mu$F for $C_{vref,upper}$ that are chosen in accordance with the datasheets [7, 9].
With these references the range of readable voltages is $2.5\ \text{V} - 1.5\ \text{V} = 1\ \text{V}$. This results in a maximal achievable resolution of $\frac{1\ \text{V}}{4095} = 0.244$ mV at the ADC's input.

3.2.4 Current Measurement

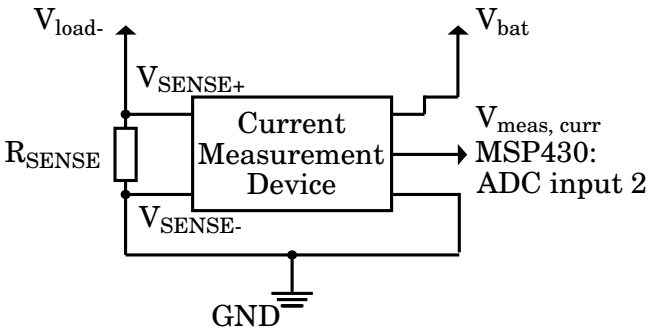

Figure 3-6: Current measurement device. This component determines the voltage at the terminals of the R_{SENSE} resistor and amplifies it by ten.

The current measurement device shown in Figure 3-6 has been selected so as to present a high precision over the current range of interest. This device measures the voltage across a sense resistor R_{SENSE} and amplifies it by ten. The output voltage $V_{meas,curr}$ can then be converted by the ADC. The value of R_{sense}, which is **2.5 Ω**, in combination with the datasheet [14] of the measurement device gives **0** mA and **120** mA as theoretical achievable boundaries for current measurement. These values are chosen to fit the demands of the system under consideration. The digital representation of the current that is drawn by the system is given in Equation 3.2. As is shown in this equation only the upper voltage reference is used to convert the current.

$$I_{bat,ADC} = 4095 \cdot \frac{10 \cdot I_{bat} \cdot R_{sense}}{V_{ref,upper}} \tag{3.2}$$

The current measurement device is placed in the current's return path in order to respect the maximal applicable voltage to the V_{SENSE-} and V_{SENSE+} pins which is defined in the datasheet.
This current sensor, although precise, has the following disadvantage. The use of a resistor to determine the current leads to a voltage drop at its terminals. This influences the voltage applied to other devices nearby (connected in series or in parallel to the resistor) and that is the reason for which the current that flows through the voltage divider cannot be measured.
Additionally the current drawn by the voltage references is not allowed to be measured since this would lead to a noticeable and load dependent shift in potential between the grounds of the voltage references and of the voltage divider.

3.2.5 Time-Dependent Load

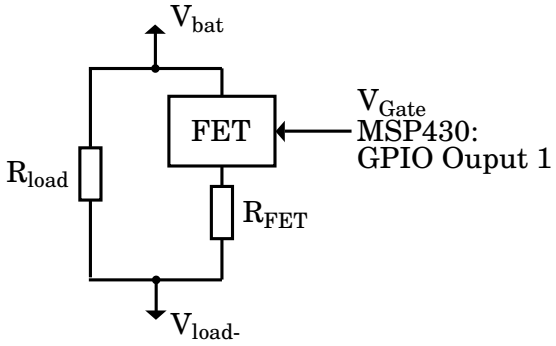

Figure 3-7: Time-dependent load implemented with resistors and a field effect transistor (FET) directly controlled by the MSP430.

To implement a time variation in the current drawn from the battery, three different loads are used. The first two are represented in Figure 3-7 while the third is part of the TinyNode184 and therefore discussed further below.
First a **resistor** that permanently raises the drawn current is used. Hereby the system current is brought into a region where discharge profiles for different loads are known.
Second a **resistor that is enabled and disabled** by the TinyNode184 is added in order to allow the load to be modified during a discharge. This load is implemented by a transistor [6] serially connected to the R_{FET} resistor. The transistor obtains its gate signal directly from the TinyNode184 and does not need a supplementary gate driver.

Load type	Resistive value (NiMH) (Ω)	Resistive value (LA) (Ω)
Resistor R_{load}	185	130
Switched resistor R_{FET}	220	220

Table 3-2: Resistive values for the resistive loads for both batteries.

At last the TinyNode184 itself is used to obtain further load variation possibilities. The TinyNode184 and its extension board contain a total of three LEDs. These draw a current of 7.6 mA if all of them are turned on. The TinyNode184 on its own draws a current of 2.5 mA. Therefore a total of four loads that are represented in Table 3-3 can be generated.

Load designation	Resistor R_{load}	Switched resistor R_{FET}	LEDs	TinyNode 184	Drawn current (mA)
LC	yes	no	no	yes	22.9
LMC	yes	no	yes	yes	30.7
HMC	yes	yes	no	yes	38.6
HC	yes	yes	yes	yes	46.0

Table 3-3: Representation of the four different loads used in combination with the NiMH battery. The table also contains the hardware employed to generate those and the current they draw. Those currents have been measured at approximately 3.8 V with consideration of the effect of the measurement device on the system.

3.2.6 Communication

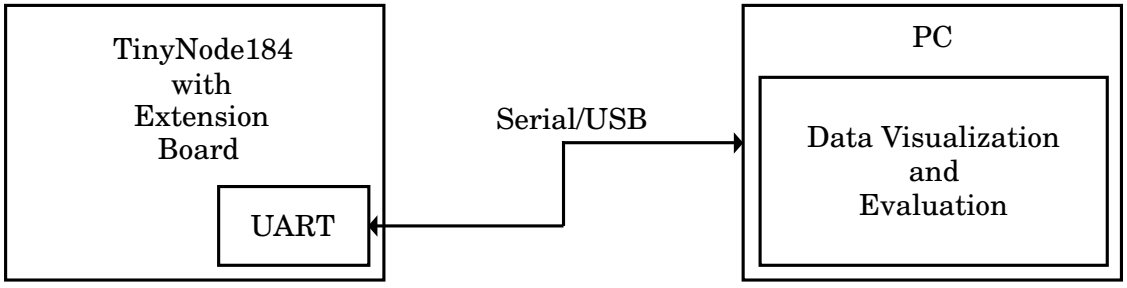

Figure 3-8: Serial communication.

As illustrated in Figure 3-8 information is transferred to a computer via the serial communication port. This allows visualization and further processing of the data. The communicated values are given in the following.

Measured Battery Voltage. Since the state-of-charge is computed as a function of the battery voltage, this value is required to be able to evaluate the approximation of the state-of-charge and its peculiarities.

Estimated Battery Current. As is described in Section 4 the state-of-charge algorithm is independent of the current if certain requirements are met. This is the case in this thesis. Therefore the currents used to determine the state-of-charge are only existent in software. By transferring them to a computer, in addition to the measured battery voltages, the obtained state-of-charge curve can be recomputed and is thereby analyzed.

Measured Battery Current. As stated in the functional requirements that introduced this chapter, the current is measured to compute a reference to the herein approximated state-of-charge.

Computed State-of-Charge. The state-of-charge is the last value that is transferred to a computer. This allows visualization, comparison and evaluation of the state-of-charge.

4

State-of-Charge Algorithm and Implementation

This chapter details the state-of-charge algorithm based on [4]. Afterwards a method of approximating the state-of-charge is described. Finally this method is adapted to account for the discharge profiles of the NiMH battery.

4.1 Overview of the state-of-charge Algorithm

The state-of-charge algorithm employed in this thesis consists of a state machine that accounts for the different electrochemical states that the battery can be in. Thus the state-of-charge is accurately computed during discharging and charging of the battery.
The **charging process** of the lead acid battery consists of three phases that are modelled in the mentioned state machine. These charging states are bulk charging, absorption charging and float charging and are used to determine the state-of-charge by only relying on the voltage of the battery. Since the charging process of the used NiMH battery is unknown and its determination would go beyond the scope of this thesis the charging process of both batteries is not considered in the remaining of this text. Further information on the state-of-charge approximation during charging of a lead acid battery is provided in [4].
During the **discharging** of the battery the algorithm computes the state-of-charge based on two values. First the **battery voltage** needs to be known. If the corresponding discharge profile is known and the load is constant, the depth-of-discharge is given by Equation 4.1 where only the coefficients a_k have to be determined a priori.

$$\widehat{DOD}(V_{bat}, a_0(RL), ..., a_n(RL)) = \Sigma_{k=1}^{n} a_k(RL) \cdot V_{bat}^k \tag{4.1}$$

The relation between the state-of-charge and the depth-of-discharge is given in Equation 4.2 where $\widehat{SOC}_{max}$ is the maximal state-of-charge and denotes a fully

charged battery. Accordingly $\widehat{SOC}_{min}$ designates a completely discharged battery.

$$\widehat{SOC}(V_{bat}, a_0(RL), ..., a_n(RL)) = \widehat{SOC}_{max} - \widehat{DOD}(V_{bat}, a_0(RL), ..., a_n(RL)) \quad (4.2)$$

Second the **battery current** needs to be known to account for variable loads. In this case, and as shown in Equation 4.3, the coefficients a_k are calculated at runtime depending on the relative load $RL = \frac{I_{bat}}{C_{bat}}$. C_{bat} is the battery's nominal capacity. By interpolating a_k with coefficients $b_{l,k}$, the algorithm can be executed with loads different to those used during the generation of the discharge profiles.

$$a_k(RL(I_{bat})) = \sum_{l=0}^{m} b_{l,k} \cdot RL(I_{bat})^l \quad (4.3)$$

Thus the state-of-charge of the battery during discharging is determined by using its voltage and current. By assuming that the current drawn from the load is not heavily dependent on the battery voltage and that all load states are known prior to implementing the algorithm, the computation of the state-of-charge becomes independent of the actual I_{bat} and does therefore not need to be measured. The different load currents are then determined preliminary and fed into the algorithm.

After this introduction to the employed state-of-charge algorithm the steps to take to implement it are depicted in Figure 4-1. They are described in the following.

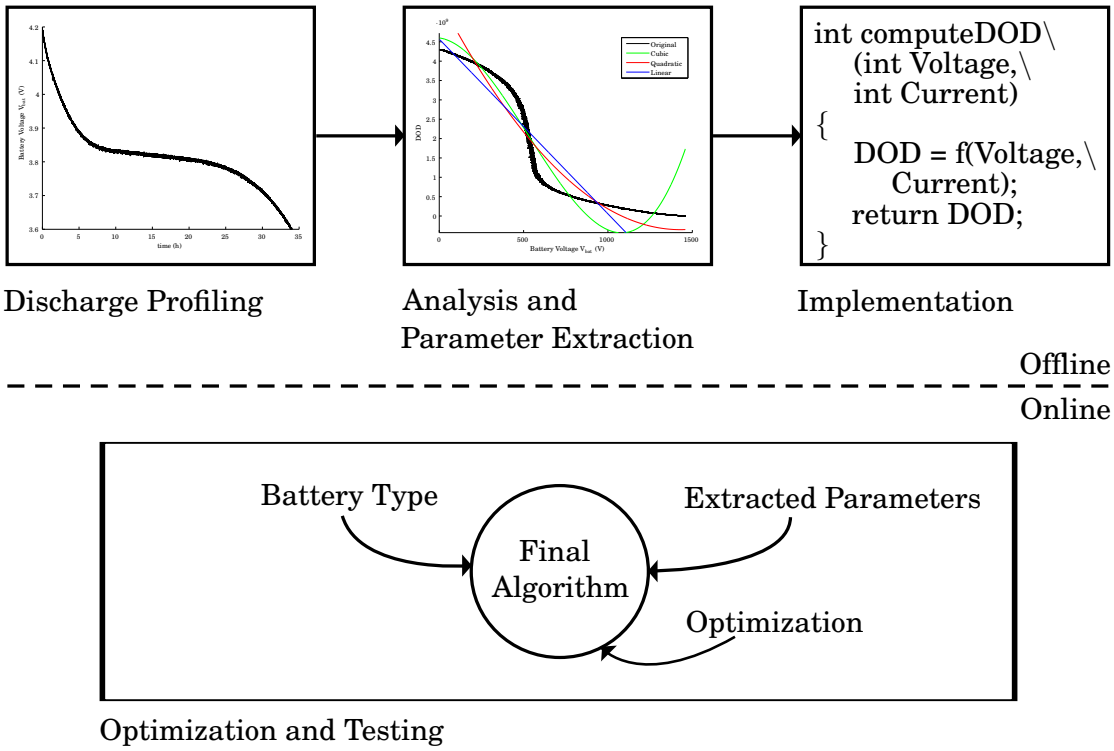

Figure 4-1: State-of-Charge parameter extraction and optimization.

Discharge Profiling As previously mentioned, discharge profiles with constant battery currents I_{bat} have to be known. The previously described approximation may become invalid for battery currents that are very different from currents used to determine the discharge profiles. Therefore the discharge profiles are conducted for currents that are expected to occur during the operation of the system.

Analysis and Parameter Extraction The aforementioned battery discharge profiles are used to determine the coefficients a_k. These are then interpolated which results in the coefficients $b_{l,k}$. Both sets of coefficients are approximated by polynomials with the use of the least squares method [10]. The polynomials shall on one hand deliver an accurate approximation of the measured data points and on the other hand be computable with a reasonable effort. Therefore the order of the polynomials shall be carefully chosen as it strongly influences the overall state-of-charge approximation. This procedure is shown in Figure 4-2.

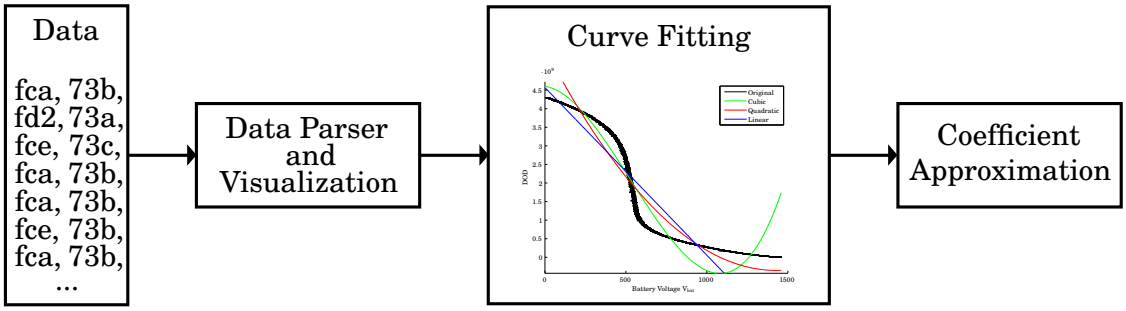

Figure 4-2: Parameter extraction steps.

Algorithm Implementation The previously described steps yield information that is then used to implement the state-of-charge algorithm on the TinyNode184. Additionally an application is implemented to test the modified algorithm.

Verification and Testing Finally the algorithm's performance is analyzed and compared to expected results obtained through coulomb counting. These expected results are the reference to which the state-of-charge determined by the modified algorithm is compared to.

The first two steps are conducted in the remaining of this chapter for two different approaches. The first approach will in general follow the steps performed in [4]. The second one improves on the first one to optimize it for the used NiMH battery.

4.2 Resource Constrained Parameter Extraction and Optimization

The data type of the extracted parameters and the orders of the approximating polynomials define the algorithm's execution time and the accuracy of the obtained results. Therefore the times taken to evaluate polynomials of second and third order,

for different data types, are compared. This information alongside the plots of the approximations are used to choose the order of the approximating polynomials in the following.

Data type	Quadratic execution time (μs)	Cubic execution time (μs)
unsigned 16-bit integers	30	44
unsigned 32-bit integers	37	72
unsigned 64-bit integers	98	198
floating point units	441	664

Table 4-1: Execution times for quadratic and cubic polynomial evaluation on the MSP430.

Table 4-1 provides the time taken by the MSP430 to perform a quadratic or a cubic polynomial evaluation using Horner's method that is explained in [1]. The MSP430 natively operates on 16-bit values and does not contain a floating point unit. Therefore 32-bit, 64-bit and floating point operations are not supported by the hardware and have to be implemented in software. This results in longer execution times and has to be considered while choosing which data type to use and the order of the polynomial approximation. As is described in Section 4.2.2 using a 32-bit data type yields more precise fitting (due to shifting of the abscissa and rescaling of the depth-of-discharge) yet intermediate results of the polynomial evaluation will exceed the capacity of a 32-bit integer value.
To inhibit optimization by the nesC compiler during the determination of the durations given in Table 4-1, the polynomials are evaluated at different values. Similar execution times are thus expected for a polynomial evaluation in the state-of-charge algorithm.

In conclusion, using integer values is more favorable than performing floating point operations. On the contrary, as results from Section 4.2.2, a high polynomial order is necessary in order to approximate the whole depth-of-discharge with only one polynomial. This leads to prolonged execution durations. Therefore a second approach is elaborated in Section 4.2.3. Its advantage is that the order of the polynomials remains small while more precisely approximating the depth-of-discharge at the cost of an additional branch instruction. The disadvantage of this approach is the small magnitude of the coefficients a_k that are as a consequence represented as floating point numbers in the algorithm implementation.

To summarize, the implementation of the state-of-charge algorithm is done using floating point numbers in combination with the approach described in Section 4.2.3. The first method (4.2.2) uses 64-bit operations to compute, and 32-bit data types to store the resulting state-of-charge. It shows the need for a modification of the algorithm published in [4], which is presented in Section 4.2.3.

4.2.1 Discharge Profiles

Figure 4-3 illustrates the measured voltage profiles for the NiMH battery. The extraction of parameters in the following is based on these measurements. Since the

NiMH battery's behavior has not yet been analyzed for use with the algorithm presented in [4] a total of five equidistant discharge profiles are considered. This enables verification of the polynomials' orders with which the coefficients a_k are interpolated.

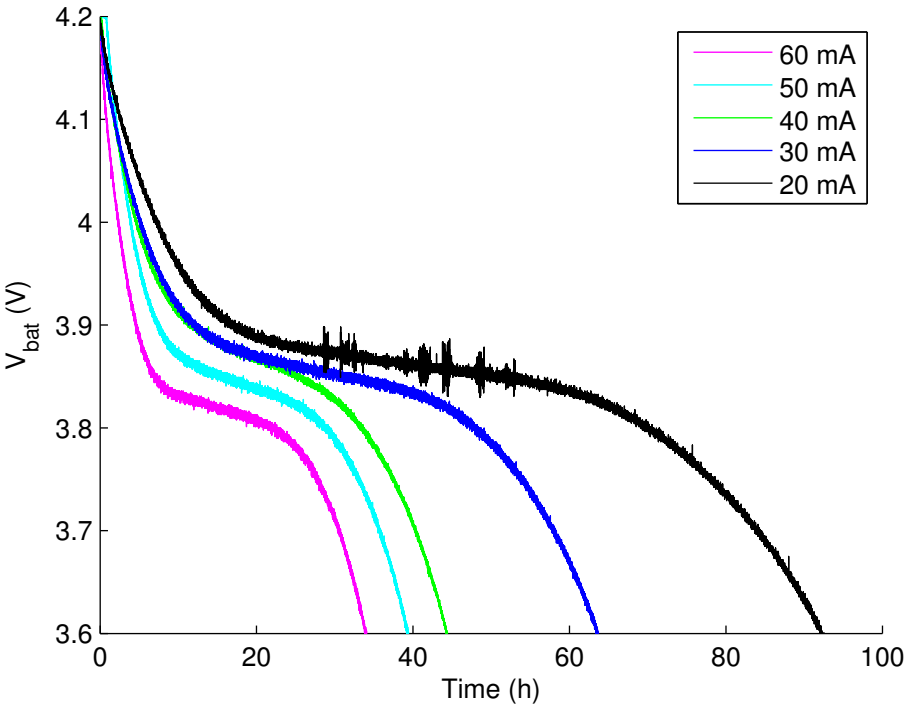

Figure 4-3: Discharge profiles of the NiMH battery at currents of 60 mA, 50 mA, 40 mA, 30 mA and 20 mA respectively.

The cutoff voltage that defines the voltage at which the battery is considered empty is set to 3.6 V. The slope of the voltage starts to decrease at nearly 3.8 V and is decreasing even further before reaching 3.6 V. After this voltage has been reached the battery is unable to keep its voltage high enough for the application to continue operating for a prolonged period of time. Additionally the battery spent most of its charges at that voltage. It is therefore reasonable to define 3.6 V as the cutoff voltage (V_{cutoff}). Furthermore the biggest part of the charges is delivered at the plateau at which the slope is maximal. The remaining battery capacities, for an initial value of 2200 mAh, are represented in Table 4-2. It is therefore expected to obtain a $\widehat{SOC}$ of zero percent at approximately ten percent remaining battery capacity.

Load currents (mA)		60	50	40	30	20	Average
Remaining battery capacity	(mA)	171	214	347	255	193	236
	(%)	7.8	9.7	15.8	11.6	8.8	10.7

Table 4-2: Remaining capacity of the NiMH battery for different constant loads at 3.6 V. The initial battery capacity is given as 2200 mAh.

The remaining of this chapter considers the mentioned approaches and is one of the major contributions of this thesis.

4.2.2 Data Approximation: First Approach

The obtained discharge profiles represent the battery voltage over time. At the beginning of the discharge process the battery contains its maximal capacity. Therefore this time and voltage tuple $(t_0, V_{initial})$ is defined to represent 100 percent state-of-charge. The state-of-charge as a function of this tuple is given in Equation 4.4. Accordingly Equation 4.5 represents the depth-of-discharge for the same tuple. Both equations are represented with the previously defined values $\widehat{SOC}_{max}$ and $\widehat{SOC}_{min}$.

$$\widehat{SOC}(t_0, V_{initial}) = \widehat{SOC}_{max} \tag{4.4}$$

$$\widehat{DOD}(t_0, V_{initial}) = \widehat{SOC}_{min} \tag{4.5}$$

The instant at which the cutoff voltage is reached together with that voltage (t_1, V_{cutoff}) is set to represent the point at which the battery is fully discharged. Therefore $\widehat{SOC}\,(t_1, V_{cutoff})$ is set to represent zero percent state-of-charge. This results in Equations 4.6 and 4.7.

$$\widehat{SOC}(t_1, V_{cutoff}) = \widehat{SOC}_{min} \tag{4.6}$$

$$\widehat{DOD}(t_1, V_{cutoff}) = \widehat{SOC}_{max} \tag{4.7}$$

In between these two defined points the state-of-charge decreases linearly. The depth-of-discharge behaves according to Equation 4.2 and therefore linearly increases. The non-linearity in the battery voltage (that is represented in Figure 4-3) is thereby removed from the approximated state-of-charge that the algorithm provides.

It is now possible to exchange the linearly increasing time and the linearly increasing depth-of-discharge. Thereby all discharge profiles become independent of time and show the dependence of the battery voltage from the depth-of-discharge. By switching the two axes the depth-of-discharge is obtained as a function of the battery voltage.

The state-of-charge is now a function of the battery voltage. As previously seen it is beneficial to use integer numbers. At this point the coefficients that are obtained through curve fitting are orders of magnitude smaller than one. Additionally errors become unacceptably large when using integer numbers. This is mainly due to the reduction of the order of the approximating polynomial, because the higher order coefficients are vanishingly small. A cubic polynomial then becomes quadratic or linear and a close fitting of the polynomial to the data is no longer possible.

Therefore, to still be able to use integer operations, the coefficients are in the following modified by transformations applied directly to the data.

Shifting of the Abscissa. Even though the voltage references discussed in Chapter 3 increase the resolution and decrease the amount of unused measurable voltages, a part of those voltages remain unused and this is used herein. The intent behind this transformation is to increase the coefficients a_k by approximating the same data over a voltage that is smaller in magnitude. Thereby the coefficients might compensate the much smaller input voltage.
The shifting of the abscissa is therefore performed as follows. The abscissa is shifted by the cutoff voltage and in direction of smaller numbers. The cutoff voltage is therefore pinned to the origin. The depth-of-discharge accordingly reaches its maximal value at this point.
The new polynomial is represented in Equation 4.8 where $a_{k,shifted}$ are the coefficients obtained by fitting a polynomial to the shifted data.

$$\begin{aligned}\widehat{DOD}(V_{bat}, a_0(RL), ..., a_n(RL)) &= \sum_{k=1}^{n} a_k(RL) \cdot V_{bat}^k \\ &= \sum_{k=1}^{n} a_{k,shifted}(RL) \cdot (V_{bat} - V_{cutoff})^k\end{aligned} \tag{4.8}$$

Depth-of-Discharge Scaling. In order to enlarge the coefficients that are too small to be contained in an integer data type the whole polynomial and thereby every single a_k is multiplied with a constant number M. The resulting $\widehat{DOD}$ is also enlarged by M as is shown in Equation 4.9. This number is chosen such that the $\widehat{DOD}$ can be contained in an unsigned 32-bit integer and is therefore given as $M = \frac{4294967295}{4095} \simeq 1.05 \cdot 10^6$. The number 4095 is the maximal ADC output. The thereby obtained coefficients are denoted $a_{k,scale}$ and are computed with $a_{k,scale} = M \cdot a_k$ Since the state-of-charge is most important as a relation to the battery's capacity, the scaling of the depth-of-charge is valid and does not adversely affect the computed result.

$$\begin{aligned}& \widehat{DOD}(V_{bat}, a_0(RL), ..., a_n(RL)) &&= \sum_{k=1}^{n} a_k(RL) \cdot V_{bat}^k \\ \iff\quad & M \cdot \widehat{DOD}(V_{bat}, a_0(RL), ..., a_n(RL)) &&= M \cdot \sum_{k=1}^{n} a_k(RL) \cdot V_{bat}^k \\ \iff\quad & M \cdot \widehat{DOD}(V_{bat}, a_0(RL), ..., a_n(RL)) &&= \sum_{k=1}^{n} M \cdot a_k(RL) \cdot V_{bat}^k \\ \iff\quad & M \cdot \widehat{DOD}(V_{bat}, a_0(RL), ..., a_n(RL)) &&= \sum_{k=1}^{n} a_{k,scale}(RL) \cdot V_{bat}^k\end{aligned} \tag{4.9}$$

The effects that the shifting of the abscissa and the scaling of the depth-of-charge have on the approximating polynomial's coefficients are presented in Table 4-3.
The scaling of the depth-of-discharge is in this case the most useful transformation. As intended the coefficients are multiplied by M and this number is large enough to allow the use of integer data types without reducing the order of the approximating polynomial.

The shifting of the abscissa does alter the coefficients, but its effect has no implication on the remainder of this thesis. Nevertheless an advantage of this transformation is that the magnitude of the coefficient a_0 is influenced as is described in the

following.
For voltages that are smaller than the cutoff voltage the polynomial used to approximate the data does not need to fulfill any requirements. Therefore, depending on the exact approximation, the polynomial may for these voltage values behave in any possible way. This may then lead to a big coefficient a_0. For example if the polynomial's slope is negative in-between the origin and the cutoff voltage, a_0 might be considerably bigger than the maximal state-of-charge. By shifting the abscissa by the cutoff voltage the polynomial has to fit the data that is close to the origin. Since the depth-of-discharge is defined to be maximal at the cutoff voltage, this transformation forces the coefficient a_0 to be close to $\widehat{SOC}_{max}$.
To conclude, this transformation is useful if the coefficient a_0 is too large to be contained in the chosen data type.

Modification	a_3	a_2	a_1	a_0
No modification	$7.12 \cdot 10^{-6}$	$-6.76 \cdot 10^{-2}$	$2.07 \cdot 10^{2}$	$-2.03 \cdot 10^{5}$
Shifting of the abscissa	$7.12 \cdot 10^{-6}$	$-1.13 \cdot 10^{-2}$	$-5.57 \cdot 10^{-1}$	$4.39 \cdot 10^{3}$
Rescaling of the $\widehat{DOD}$	7.47	$-7.09 \cdot 10^{4}$	$2.17 \cdot 10^{8}$	$-2.13 \cdot 10^{11}$
Shifting of the abscissa and rescaling of the $\widehat{DOD}$	7.47	$-1.18 \cdot 10^{4}$	$-5.85 \cdot 10^{5}$	$4.60 \cdot 10^{9}$

Table 4-3: Comparison of coefficients of cubic approximations ($\widehat{DOD}(V_{bat}) = a_3 \cdot V_{bat}^3 + a_2 \cdot V_{bat}^2 + a_1 \cdot V_{bat} + a_0$) of the discharge profile (for a load of $60\,mA$) under different modifications.

In order to be able to compare different polynomials and their fitting to the data the root-mean-square error is computed in the following. For an easier comparison of different approximations the root-mean-square error is divided by the maximal depth-of-discharge. This value is given in Table 4-4 for this approach.

Value	Maximum	Minimum
$\widehat{SOC}$	4294967295	0
$\widehat{DOD}$	4294967295	0

Table 4-4: Maximal and minimal values for the $\widehat{SOC}$ and $\widehat{DOD}$. These values result from the depth-of-discharge scaling and are used in combination with the first approach.

Table 4-5 represents the root-mean-square errors that are produced by polynomials of different orders and data types. It can be seen that the errors only slightly increase when using integer data types. As previously seen this allows the state-of-charge to be determined faster by a factor of three. Additionally this table shows that the error produced by the cubic approximation is smaller by an order of magnitude than the errors that are obtained with the linear or the quadratic curve fitting.

Approximation type	Discharge current (mA)	Floating point	Integer
Linear	60	$1.35 \cdot 10^{-1}$	$1.35 \cdot 10^{-1}$
	50	$1.26 \cdot 10^{-1}$	$1.26 \cdot 10^{-1}$
	40	$9.86 \cdot 10^{-2}$	$9.86 \cdot 10^{-2}$
	30	$1.14 \cdot 10^{-1}$	$1.14 \cdot 10^{-1}$
	20	$1.17 \cdot 10^{-1}$	$1.17 \cdot 10^{-1}$
Quadratic	60	$1.14 \cdot 10^{-1}$	$1.14 \cdot 10^{-1}$
	50	$1.05 \cdot 10^{-1}$	$1.05 \cdot 10^{-1}$
	40	$9.26 \cdot 10^{-2}$	$9.26 \cdot 10^{-2}$
	30	$1.07 \cdot 10^{-1}$	$1.07 \cdot 10^{-1}$
	20	$1.12 \cdot 10^{-1}$	$1.12 \cdot 10^{-1}$
Cubic	60	$9.52 \cdot 10^{-2}$	$1.10 \cdot 10^{-1}$
	50	$8.07 \cdot 10^{-2}$	$9.06 \cdot 10^{-2}$
	40	$5.71 \cdot 10^{-2}$	$8.55 \cdot 10^{-2}$
	30	$7.27 \cdot 10^{-2}$	$9.71 \cdot 10^{-2}$
	20	$7.68 \cdot 10^{-2}$	$8.93 \cdot 10^{-2}$

Table 4-5: Root mean square errors for different approximation polynomials using floating point and integer data types. The maximum depth-of-discharge is used as a reference for comparison purposes.

The set of coefficients used to implement the state-of-charge algorithm are given in Table 4-6. The depth-of-discharge is approximated for all five loads. The table shows that the coefficients with which the a_k are approximated are too small to be represented as integer numbers. Therefore it is not possible to compute the algorithm in a precise way when restricting the used data type to integers.

Used data type	Approximated value	Order of coefficient			
		cubic	quadratic	linear	constant
Float	$\widehat{DOD}_{60mA}$	7.47	$-1.18 \cdot 10^4$	$-5.85 \cdot 10^5$	$4.60 \cdot 10^9$
	$\widehat{DOD}_{50mA}$	6.24	$-1.13 \cdot 10^4$	$5.39 \cdot 10^5$	$4.43 \cdot 10^9$
	$\widehat{DOD}_{40mA}$	7.41	$-1.44 \cdot 10^4$	$3.09 \cdot 10^6$	$4.07 \cdot 10^9$
	$\widehat{DOD}_{30mA}$	9.51	$-1.79 \cdot 10^4$	$4.19 \cdot 10^6$	$4.02 \cdot 10^9$
	$\widehat{DOD}_{20mA}$	9.66	$-1.84 \cdot 10^4$	$4.68 \cdot 10^6$	$3.95 \cdot 10^9$
	$a_3(I_{bat})$		$1.56 \cdot 10^{-6}$	$-6.98 \cdot 10^{-3}$	$1.48 \cdot 10^1$
	$a_2(I_{bat})$		$-1.07 \cdot 10^{-3}$	8.34	$-2.52 \cdot 10^4$
	$a_1(I_{bat})$		-1.16	$3.48 \cdot 10^2$	$5.32 \cdot 10^6$
	$a_0(I_{bat})$		$2.18 \cdot 10^2$	$-2.98 \cdot 10^5$	$4.04 \cdot 10^9$
Integer	$\widehat{DOD}_{60mA}$	7	$-1.18 \cdot 10^4$	$-5.85 \cdot 10^5$	$4.60 \cdot 10^9$
	$\widehat{DOD}_{50mA}$	6	$-1.13 \cdot 10^4$	$5.39 \cdot 10^5$	$4.43 \cdot 10^9$
	$\widehat{DOD}_{40mA}$	7	$-1.44 \cdot 10^4$	$3.09 \cdot 10^6$	$4.07 \cdot 10^9$
	$\widehat{DOD}_{30mA}$	10	$-1.79 \cdot 10^4$	$4.19 \cdot 10^6$	$4.02 \cdot 10^9$
	$\widehat{DOD}_{20mA}$	10	$-1.84 \cdot 10^4$	$4.68 \cdot 10^6$	$3.95 \cdot 10^9$
	$a_3(I_{bat})$		0	0	$1.60 \cdot 1$
	$a_2(I_{bat})$		0	8	$-2.52 \cdot 10^4$
	$a_1(I_{bat})$		-1	$3.48 \cdot 10^2$	$5.32 \cdot 10^6$
	$a_0(I_{bat})$		$2.18 \cdot 10^2$	$-2.98 \cdot 10^5$	$4.04 \cdot 10^9$

Table 4-6: Cubic fitting of the depth-of-discharge with the coefficients a_k and the quadratic approximation of a_k with $b_{l,k}$. The $\widehat{DOD}$ is given as a function of $V_{bat} - V_{cutoff}$ for all five loads.

Figure 4-4 illustrates the depth-of-discharge and its linear, quadratic and cubic approximations for a load of 60 mA. This figure shows that the depth-of-discharge cannot be approximated with a polynomial of order three or less. For example the approximation yields a depth-of-discharge of approximately 40 percent for a fully charged battery.

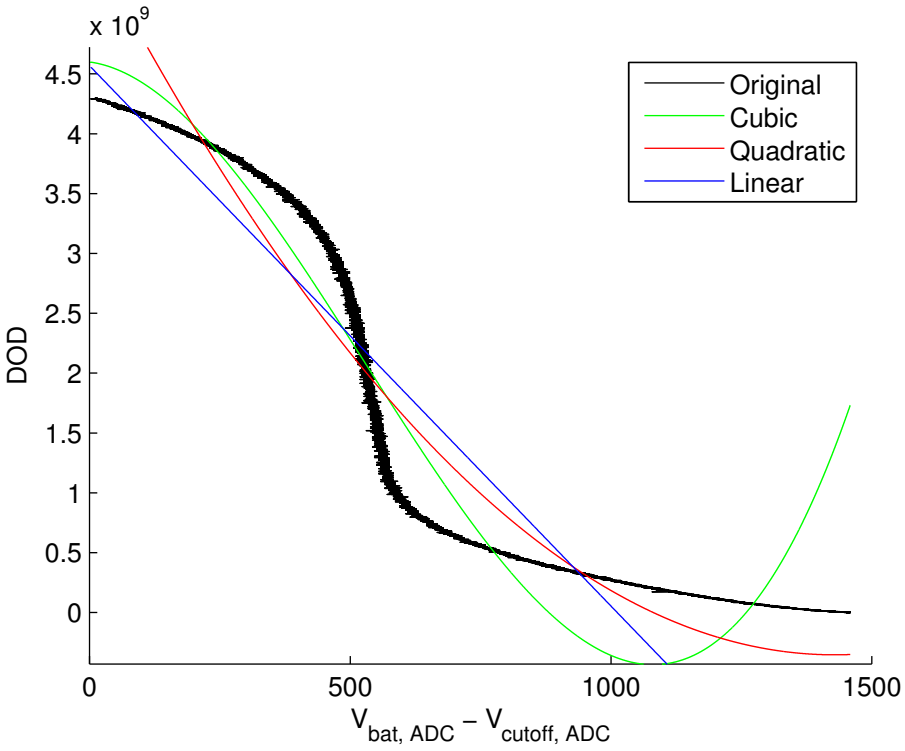

Figure 4-4: Depth-of-discharge of the NiMH battery with a load of 60 mA.

To conclude this approach shows that the use of integer numbers to approximate the depth-of-discharge only negligibly increases the produced error. Unfortunately the depth-of-discharge of the NiMH battery cannot be approximated with a polynomial of order three or less, because of the bad fitting of the polynomial to the data. Therefore a polynomial of higher order can be used to nevertheless fit one polynomial to the whole depth-of-discharge. Another possibility to approximate the depth-of-discharge is to split the data in several segments that can then be approximated with a polynomial of order three. This method is described and used in the following.

4.2.3 Data Approximation: Approach 2

To avoid having to increase the order of the approximating polynomial the discharge profile is split into two parts that are then both approximated by two cubic polynomials. An upper part computes the depth-of-discharge in the case that the battery voltage is higher than a certain value, which is subsequently referred to as threshold voltage (V_{thresh}), and is determined in the following. A lower part computes the depth-of-discharge if the battery voltage is smaller than the mentioned threshold voltage. A drawback of this method is that the differences of these two parts and of there derivatives is noticeable where the threshold voltage is reached.

All values determined to this point are parameters of the battery under consideration. A good way to define the threshold voltage is therefore to continue in this manner and determine it as a parameter of the discharge profile. It is then also a

parameter of the battery. Furthermore this approach may then be applied to other NiMH batteries without having to redefine an arbitrarily chosen value.
The voltage at which the maximum derivative occurs is such a battery parameter. This choice of threshold voltage is illustrated in Figure 4-5 with respect to time.

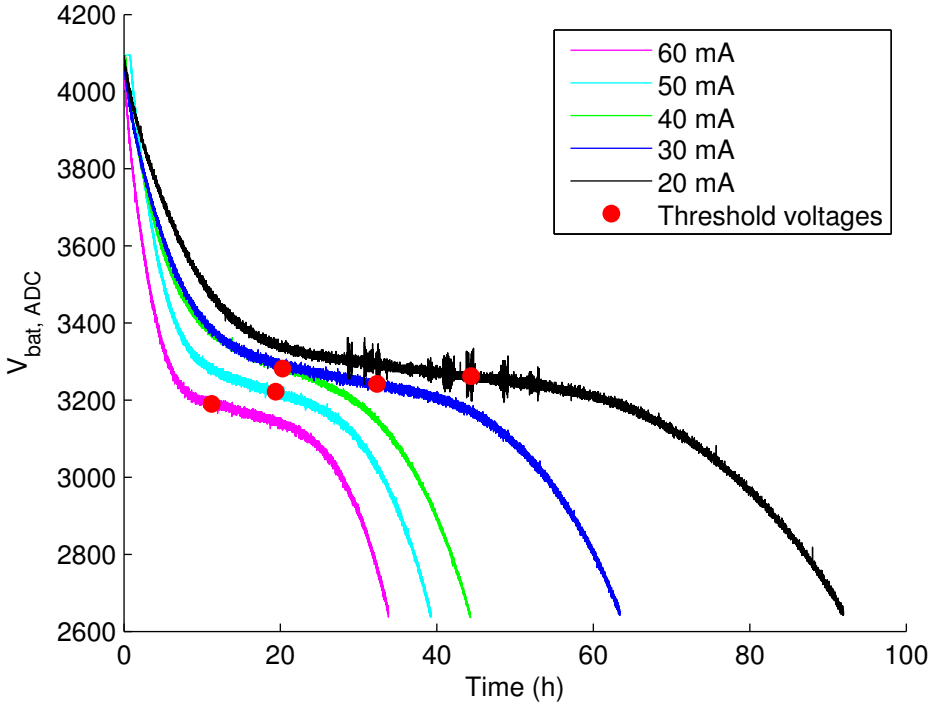

Figure 4-5: Visualization of thresholds.

The derivatives of the discharge profiles are depicted in Figure 4-6. The derivative's maximum is located somewhere in the aforementioned plateau. An advantage is that both parts approximate the depth-of-discharge for similar durations. Additionally the switching between the two parts does not influence the depth-of-discharge at the beginning or the end of the discharging process. These are the regions where information provided by the state-of-charge algorithm is the most interesting. A disadvantage of this choosing of V_{thresh} is the prolonged time that the battery's voltage stays close to the threshold voltage. This is due to the smaller magnitude of the slope and leads, in combination with noise on the ADC output, to more switches between the two parts of the depth-of-discharge approximation.

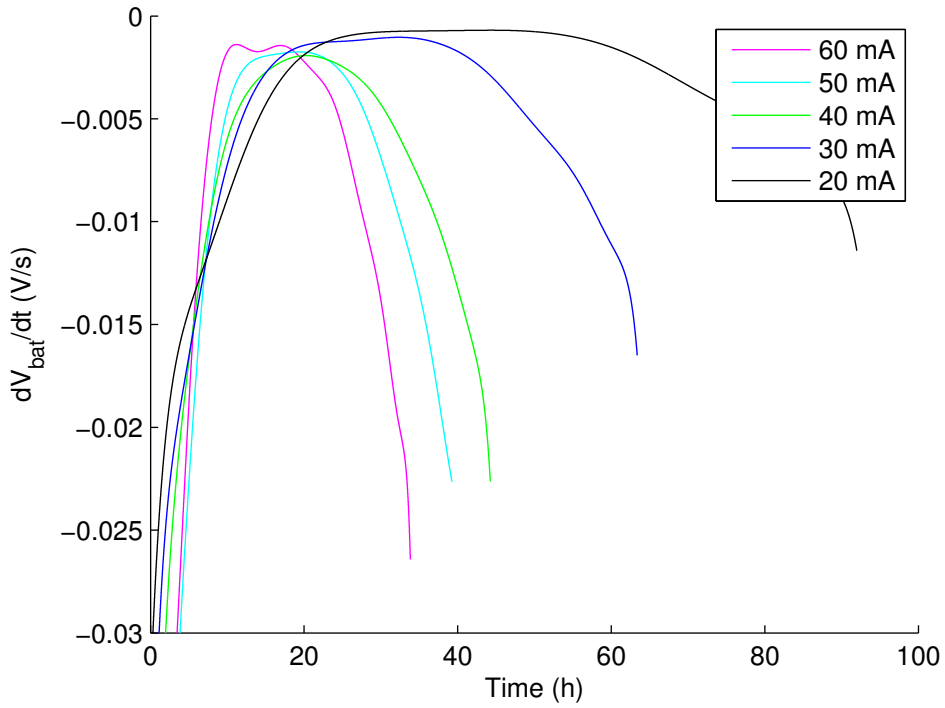

Figure 4-6: Derivatives of discharge profiles.

Different possibilities to determine these threshold voltages during the system's runtime are presented in the following.

Finite State Machine If the currents that are drawn from the battery are known prior to the system's execution the corresponding threshold voltages are determined offline. It is then possible to feed them into the state-of-charge algorithm with a finite state machine. This reduces the amount of operations performed to obtain the threshold voltage in comparison to the two next possibilities. This is especially the case if a finite state machine is already used to feed state currents into the algorithm.

Lookup Table In order to account for a higher variety of currents and if the currents drawn from the battery are unknown prior to the execution of the embedded system a lookup table can be used to approximate the threshold voltage. This possibility's advantage is its simplicity. Different threshold voltages are determined through extraction of parameters from discharge profiles.

Polynomial Approximation This is the most general possibility and is therefore used in the remaining of this thesis. The gathered data are used to approximate the threshold voltages with a polynomial that is then evaluated during execution of the algorithm. Based on this the program decides which part of the depth-of-discharge approximation to use.
Since threshold voltages have not yet been used in combination with the considered state-of-charge algorithm, it is not possible to know how these behave for different currents. Therefore the order by which the threshold voltages are best approximated is yet unknown.

Figure 4-7 shows the dependence of the threshold voltages from the current that is drawn by the battery. The threshold voltages are approximated by a polynomial of second order which is also illustrated in the named figure. This approximation is reasonable for the considered current range. Further investigations have to be conducted to approximate the threshold voltages for currents that are bigger than 60 mA or smaller than 20 mA.

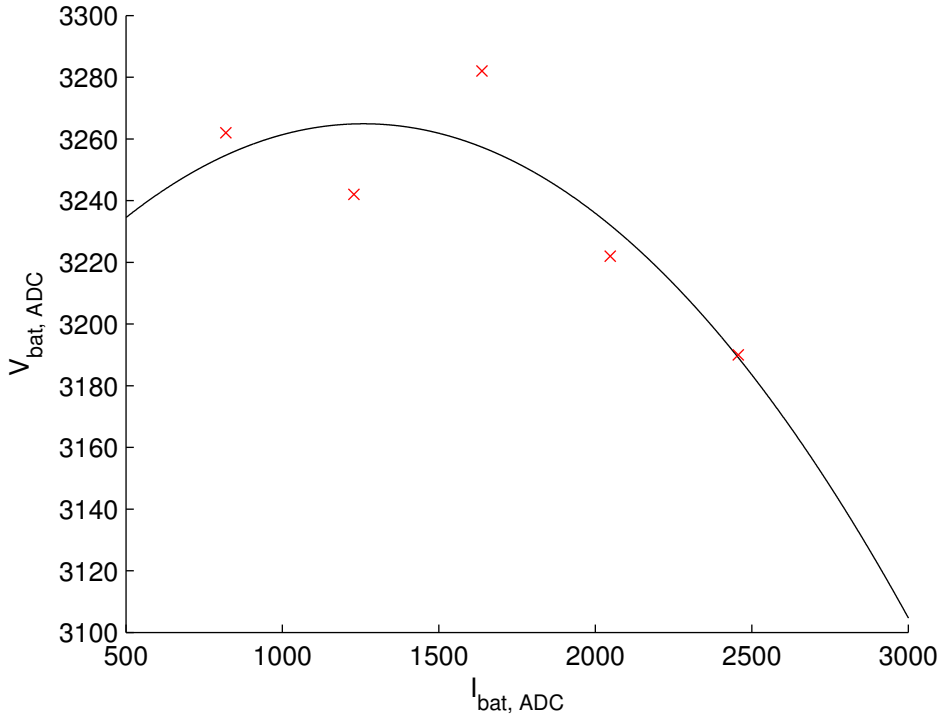

Figure 4-7: Dependence of thresholds from the current.

The voltage at which the discharge profile is split into two parts is now known and approximated by the polynomial with coefficients c_i given in Table 4-7. Due to the dependence of the depth-of-discharge from the battery voltage the consideration of the threshold voltage can easily be added to the considered algorithm. The remainder of this chapter considers this modification of the state-of-charge algorithm.

Coefficient	c_2	c_1	c_0
Value	$-5.28 \cdot 10^{-5}$	$1.33 \cdot 10^{-1}$	$3.18 \cdot 10^{3}$

Table 4-7: Polynomial approximation of the threshold voltage V_{thresh} such that $V_{thresh} \simeq c_2 \cdot I_{bat}^2 + c_1 \cdot I_{bat} + c_0$.

The threshold voltage has been defined and approximated by a polynomial to allow its computation for different battery currents. Therefore the next step to take is to approximate the two parts that determine the depth-of-discharge above and below the threshold voltage. Its result is depicted in Figure 4-8. The different approximations of the depth-of-discharge are explained further below.

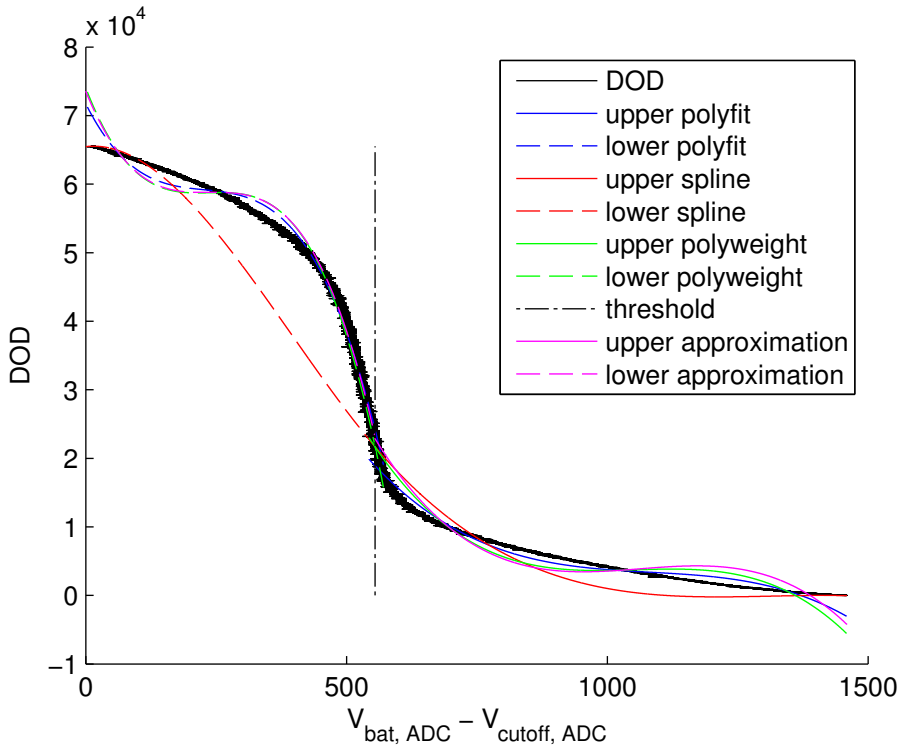

Figure 4-8: Depth-of-discharge and different approximations for the 60 mA load.

Polynomial Fitting First the same method that has already been used in the first approach is employed to approximate the two parts of the depth-of-discharge separately. The data is again approximated using the method of the least squares. While this minimizes the resulting sum of the squares of the errors, it does not consider a discontinuity that appears where the two polynomials meet.

The intention of the two following methods is to reduce or remove that difference of the two polynomials at the threshold voltage.

Splines Splines are a multitude of polynomials that are used to approximate an interval of a more complex function. Their advantage is the continuity at the meeting points of two neighboring polynomials. A major drawback is that the computation of the splines is only based on the function at the points where they meet and the equality of their derivatives at that point. The splines in consequence do only consider the data at which two polynomials meet. Applied to the task at hand this means that only three data points are considered during the calculation of the splines. These are the depth-of-discharges at the cutoff voltage, at the threshold voltage and at the voltage at which the battery is fully charged $V_{initial}$.
At the initial voltage and at the cutoff voltage the derivatives of the spline has been set to zero. Thereby the spline is uniquely defined and the result is shown in Figure 4-8.

Weighted Polynomial Fitting Another possibility to force the polynomials to meet at the threshold voltage is to use weighting [11]. The polynomials are fitted to the data in a least square way. The difference to the previous polynomial fitting is the

priority that is given to chosen data points over others. By choosing a very high weight of the data point at the threshold voltage, both polynomials are forced to pass through that point. Thereby the difference that appeared at the threshold voltage and between the two polynomials is removed. The consequence is the reduction of the overall precision of the fitting of the polynomials to the data points. As is illustrated in Figure 4-8 this loss in precision is very small and can therefore be considered to be negligible. This is therefore the method used to obtain the coefficients that are employed in the implementation of the state-of-charge algorithm.

The approximation of the coefficients, as is shown in Figure 4-8, reintroduces a difference of the two polynomials at the threshold voltage. That difference has therefore only been reduced and not completely removed.

Figures 4-9 and 4-10 depict the coefficients and their approximation as a function of the battery current for the upper and the lower polynomial. These figures also validate the quadratic approximation of a_k by $b_{l,k}$.

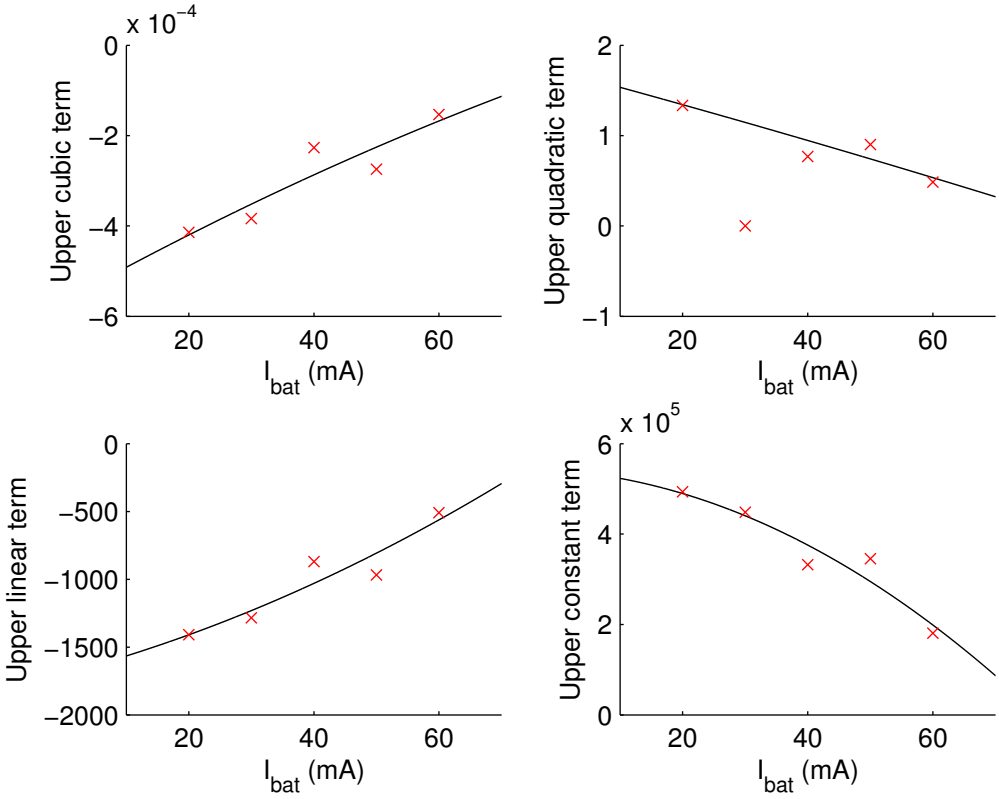

Figure 4-9: Coefficients for the computation of the terms of the upper polynomial.

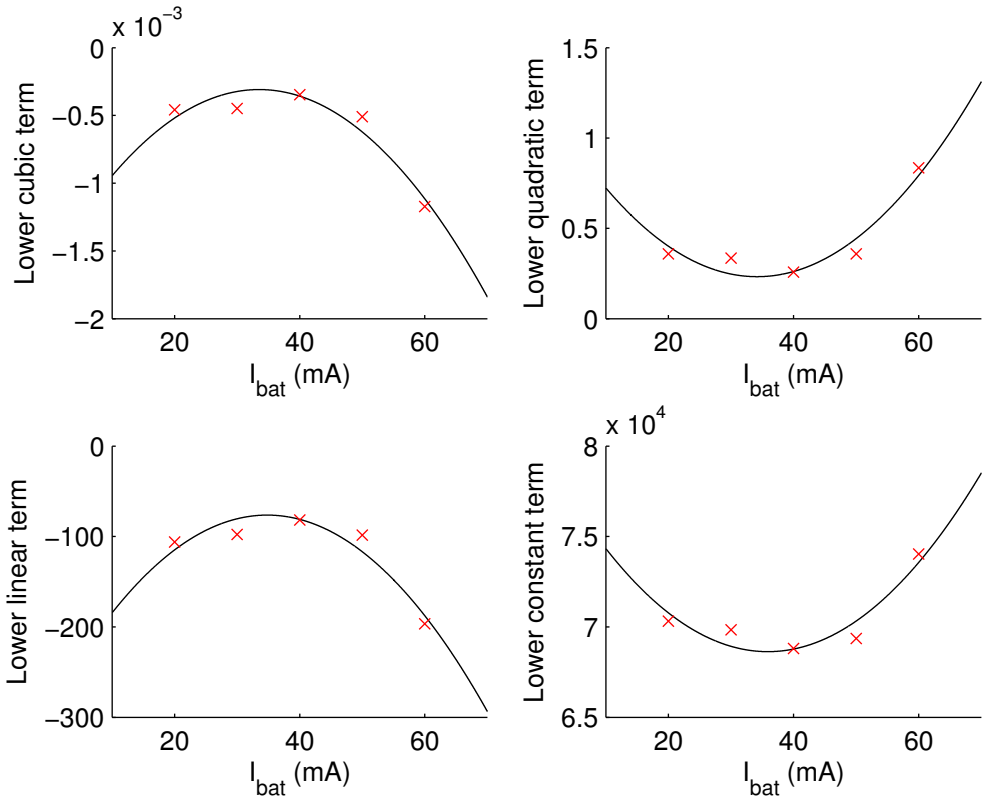

Figure 4-10: Coefficients for the computation of the terms of the lower polynomial.

The root-mean-square errors are represented in Table 4-8. This approach significantly decreases them. As mentioned previously the error obtained through polynomial fitting is slightly smaller than the error that is obtained with the chosen approximation method. This difference is negligible and the advantage gained by that method is more important.

Polynomial part	Discharge current (mA)	Polynomial fitting	Splines	Weighted polynomial fitting
Upper part	60	$1.69 \cdot 10^{-2}$	$3.78 \cdot 10^{-2}$	$2.78 \cdot 10^{-2}$
	50	$2.70 \cdot 10^{-2}$	$1.04 \cdot 10^{-1}$	$3.95 \cdot 10^{-2}$
	40	$1.49 \cdot 10^{-2}$	$7.64 \cdot 10^{-2}$	$1.95 \cdot 10^{-2}$
	30	$2.50 \cdot 10^{-2}$	$1.24 \cdot 10^{-1}$	$3.41 \cdot 10^{-2}$
	20	$2.96 \cdot 10^{-2}$	$1.23 \cdot 10^{-1}$	$3.90 \cdot 10^{-2}$
Lower part	60	$2.09 \cdot 10^{-2}$	$1.57 \cdot 10^{-1}$	$2.69 \cdot 10^{-2}$
	50	$1.30 \cdot 10^{-2}$	$8.14 \cdot 10^{-2}$	$1.60 \cdot 10^{-2}$
	40	$9.77 \cdot 10^{-3}$	$7.91 \cdot 10^{-2}$	$1.42 \cdot 10^{-2}$
	30	$1.28 \cdot 10^{-2}$	$7.28 \cdot 10^{-2}$	$1.71 \cdot 10^{-2}$
	20	$1.51 \cdot 10^{-2}$	$8.19 \cdot 10^{-2}$	$2.12 \cdot 10^{-2}$

Table 4-8: Root-mean-square errors for the upper and the lower part of the polynomial approximation. The polynomials are evaluated using floating point numbers and are approximated using weighted polynomial least squares fitting. Again the maximal depth-of-discharge is used as a reference.

Table 4-9 represents the coefficients a_k and $b_{l,k}$ obtained with this second approach. The coefficients $b_{l,k}$ are so small that it is not possible to enlarge them enough to be able to use integer data types to store them without losing too much precision. Therefore the algorithm is implemented with floating point numbers. With these the execution time is longer but the result is much more precise. Still as can be seen from Table 4-1 the complete algorithm does not take a big amount of time and leaves a lot of computational power for the embedded system's other functions.

Polynomial part	Approximated value	Order of coefficients			
		cubic	quadratic	linear	constant
Upper part	$\widehat{DOD}_{60mA}$	$-1.53 \cdot 10^{-4}$	$4.85 \cdot 10^{-1}$	$-5.09 \cdot 10^{2}$	$1.81 \cdot 10^{5}$
	$\widehat{DOD}_{50mA}$	$-2.75 \cdot 10^{-4}$	$9.01 \cdot 10^{-1}$	$-9.68 \cdot 10^{2}$	$3.46 \cdot 10^{5}$
	$\widehat{DOD}_{40mA}$	$-2.26 \cdot 10^{-4}$	$7.69 \cdot 10^{-1}$	$-8.70 \cdot 10^{2}$	$3.32 \cdot 10^{5}$
	$\widehat{DOD}_{30mA}$	$-3.84 \cdot 10^{-4}$	1.22	$-1.29 \cdot 10^{3}$	$4.48 \cdot 10^{5}$
	$\widehat{DOD}_{20mA}$	$-4.14 \cdot 10^{-4}$	1.33	$-1.41 \cdot 10^{3}$	$4.93 \cdot 10^{5}$
	a_3		$-1.02 \cdot 10^{-11}$	$1.87 \cdot 10^{-7}$	$-5.67 \cdot 10^{-4}$
	a_2		$-1.14 \cdot 10^{-8}$	$-4.56 \cdot 10^{-4}$	1.72
	a_1		$6.64 \cdot 10^{-5}$	$3.00 \cdot 10^{-1}$	$-1.70 \cdot 10^{3}$
	a_0		$-4.68 \cdot 10^{-2}$	$-2.42 \cdot 10^{1}$	$5.41 \cdot 10^{5}$
Lower part	$\widehat{DOD}_{60mA}$	$-1.17 \cdot 10^{-3}$	$8.35 \cdot 10^{-1}$	$-1.97 \cdot 10^{2}$	$7.40 \cdot 10^{4}$
	$\widehat{DOD}_{50mA}$	$-5.09 \cdot 10^{-4}$	$3.59 \cdot 10^{-1}$	$-9.84 \cdot 10^{1}$	$6.94 \cdot 10^{4}$
	$\widehat{DOD}_{40mA}$	$-3.47 \cdot 10^{-4}$	$2.58 \cdot 10^{-1}$	$-8.19 \cdot 10^{1}$	$6.88 \cdot 10^{4}$
	$\widehat{DOD}_{30mA}$	$-4.49 \cdot 10^{-4}$	$3.35 \cdot 10^{-1}$	$-9.77 \cdot 10^{1}$	$6.98 \cdot 10^{4}$
	$\widehat{DOD}_{20mA}$	$-4.58 \cdot 10^{-4}$	$3.58 \cdot 10^{-1}$	$-1.06 \cdot 10^{2}$	$7.03 \cdot 10^{4}$
	a_3		$-6.86 \cdot 10^{-10}$	$1.88 \cdot 10^{-6}$	$-1.60 \cdot 10^{-3}$
	a_2		$5.01 \cdot 10^{-7}$	$-1.40 \cdot 10^{-3}$	1.22
	a_1		$-1.05 \cdot 10^{-4}$	$2.99 \cdot 10^{-1}$	$-2.89 \cdot 10^{2}$
	a_0		$5.07 \cdot 10^{-3}$	$-1.49 \cdot 10^{1}$	$7.96 \cdot 10^{4}$

Table 4-9: Coefficients of the upper and lower polynomials for all five loads. These are determined with use of the values given in Table 4-10. The $\widehat{DOD}$ is given as a function of $V_{bat} - V_{cutoff}$.

Since floats are now used to compute the state-of-charge the maximal state-of-charge and depth-of-discharge are given in Table 4-10. Thereby the state-of-charge is a 16-bit value which is the amount of bits that the MSP430 natively operates on. Again since the state-of-charge is only interesting with respect to the $\widehat{SOC}_{max}$ no information is lost in reducing the $\widehat{SOC}$ to the given values.

Value	Maximum	Minimum
$\widehat{SOC}$	65535	0
$\widehat{DOD}$	65535	0

Table 4-10: Maximal and minimal values for the $\widehat{SOC}$ and $\widehat{DOD}$. These values are used in combination with the second approach.

To conclude, this chapter has shown the choices that have been made during this thesis in order to obtain a state-of-charge algorithm that provides accurate results while considering the resource constrained system that it is designed for. Therefore the state-of-charge algorithm is implemented, tested and evaluated in the following with the values provided during the description of this second approach.

5

Evaluation Method

This chapter describes the method that is used to evaluate the previously described state-of-charge algorithm and the modification applied to it. Furthermore the parameters that are employed to obtain the results are presented herein. Finally the named results are illustrated and discussed.

5.1 System Setup

The modified state-of-charge algorithm is in the following evaluated for two different types of loads. First, as a best case scenario, the algorithm is executed with a constant load. Thus the battery does not need to react to current variations and the $\widehat{SOC}$ is in consequence not perturbed by load transitions.
Second a more practical scenario with variable loads is evaluated. This emulates a realistic scenario for an embedded system. These variations are then due to the powering on or off of peripherals, the switching between different power modes of the processor and the possibly time-dependent current demanded by additional hardware components. Therefore a total of four loads as described in Chapter 3 are chosen to emulate such conditions.
The currents that result from these loads need to be known by the state-of-charge algorithm in order to compute the $\widehat{SOC}$ for different physical conditions.

5.1.1 Finite State Machine for Current Determination

As described in Chapter 4 the state-of-charge algorithm does not need the battery current to be measured. Nevertheless the coefficients a_k are computed as a function of the current that is flowing out of the battery. A state machine is implemented in order to communicate the instantaneous current drain to the state-of-charge algorithm. Additionally this state machine determines the state of the variable load. Figure 5-1 shows the state transitions and the currents that are drawn by the load during the execution of the different states. The order of the execution of the states is also shown in this figure.

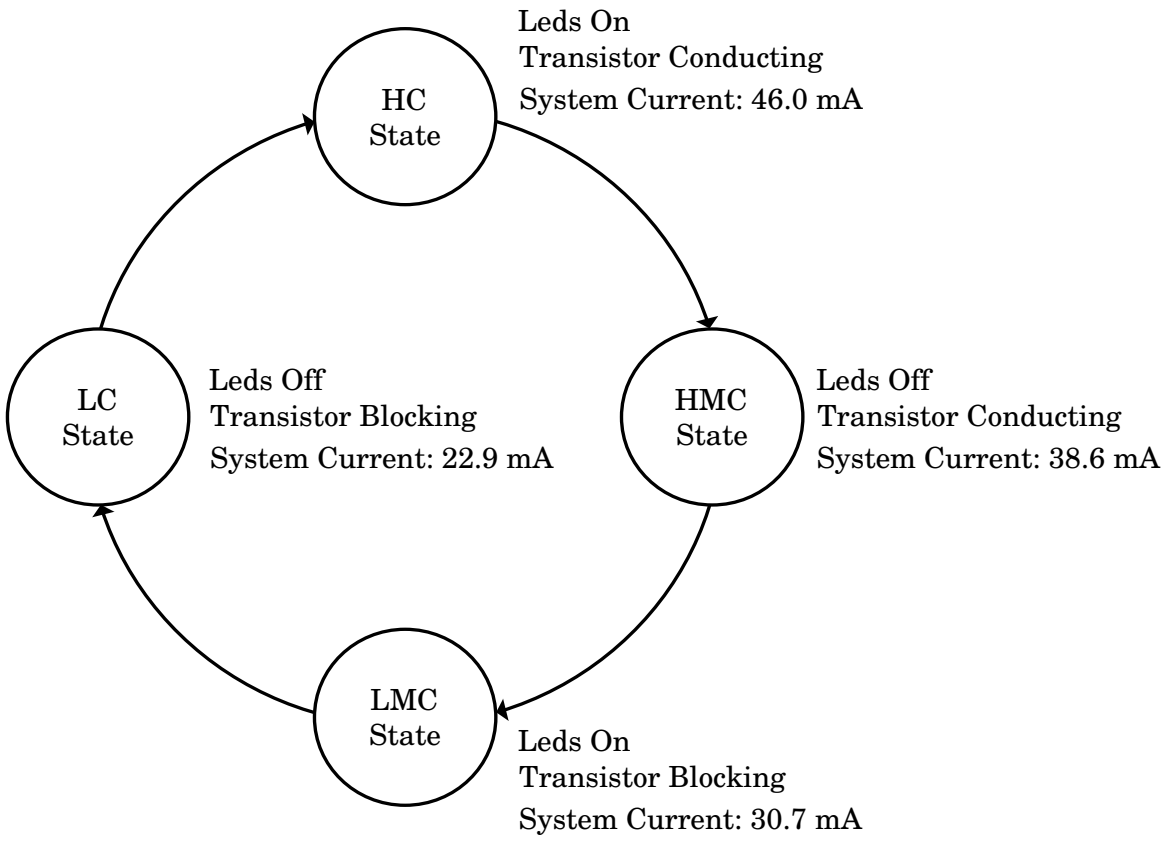

Figure 5-1: Current determining state machine.

After a state transition the battery needs some time to adapt to the new load. Therefore a time of approximately one hour is chosen for the duration of every state. A timer located on the MSP430 is used to regularly update the state. When this timer expires a predefined number of times the finite state machine is executed and the current state is terminated before the next one is initialized.

The state-of-charge algorithm is implemented as a TinyOS component and expects to obtain battery currents and voltages from the top application. This last one is responsible for determining those values and executes the previously elaborated state machine.

5.1.2 Parameter Sets

The modified state-of-charge algorithm is implemented in such a way that all necessary battery values can be given to the algorithm as parameters. Therefore the same implementation can be used in combination with different batteries, provided that the battery defining values are adapted.

The exact parameters that are used in combination with the algorithm and the finite state machine are presented in Table 5-1. The values in this table are the exact numbers used for the calculation of the state-of-charge. The numbers given in

Chapter 4 are the same values with the difference that they are only represented with three digits.

Parameter	Value
$b_{2,3,upper}$	$-1.0198 \cdot 10^{-11}$
$b_{1,3,upper}$	$1.8748 \cdot 10^{-7}$
$b_{0,3,upper}$	$-5.6682 \cdot 10^{-4}$
$b_{2,2,upper}$	$-1.14 \cdot 10^{-8}$
$b_{1,2,upper}$	$-4.5587 \cdot 10^{-4}$
$b_{0,2,upper}$	1.7237
$b_{2,1,upper}$	$6.6433 \cdot 10^{-5}$
$b_{1,1,upper}$	$2.9969 \cdot 10^{-1}$
$b_{0,1,upper}$	$-1.6998429 \cdot 10^{3}$
$b_{2,0,upper}$	$-4.6806 \cdot 10^{-2}$
$b_{1,0,upper}$	$-2.42371 \cdot 10^{1}$
$b_{0,0,upper}$	$5.4103185 \cdot 10^{5}$
$b_{2,3,lower}$	$-6.8618 \cdot 10^{-10}$
$b_{1,3,lower}$	$1.8845 \cdot 10^{-6}$
$b_{0,3,lower}$	$-1.6028 \cdot 10^{-3}$
$b_{2,2,lower}$	$5.0133 \cdot 10^{-7}$
$b_{1,2,lower}$	$-1.4036 \cdot 10^{-3}$
$b_{0,2,lower}$	1.215
$b_{2,1,lower}$	$-1.0468 \cdot 10^{-4}$
$b_{1,1,lower}$	$2.9857 \cdot 10^{-1}$
$b_{0,1,lower}$	$-2.892434 \cdot 10^{2}$
$b_{2,0,lower}$	$5.0672 \cdot 10^{-3}$
$b_{1,0,lower}$	$-1.49039 \cdot 10^{1}$
$b_{0,0,lower}$	$7.95897635 \cdot 10^{4}$
c_2	$-5.2818 \cdot 10^{-5}$
c_1	$1.3298 \cdot 10^{-1}$
c_0	$3.1812 \cdot 10^{3}$
$V_{cutoff,ADC}$	2635

Table 5-1: Parameters used with the state-of-charge algorithm for the NiMH battery. The coefficients $b_{l,k,upper}$ are used to compute a_k for the upper part of the depth-of-discharge. Accordingly $b_{l,k,lower}$ is used to obtain the coefficients a_k for the lower part of the depth-of-discharge. The parameters c_i are used to compute the threshold voltage V_{thresh}.

Because the state-of-charge algorithm has not yet been used in combination with a NiMH battery, a lead acid battery is employed for verification of its correctness. Similar steps to obtain the coefficients $b_{l,k}$ as presented in Chapter 4 are performed for the lead acid battery. These are described in Appendix A. The parameters used for this battery are stated in Table 5-2. Since the lead acid battery is reasonably accurately approximated with one polynomial (as shown in [4]) the coefficients $b_{l,k,upper}$ and $b_{l,k,lower}$ are equal. Therefore the threshold voltage is chosen arbitrarily and defined to be 3208 which represents 4 V.

Parameter	Value
$b_{2,3,upper}$	$-4.4177 \cdot 10^{-11}$
$b_{1,3,upper}$	$1.1593 \cdot 10^{-7}$
$b_{0,3,upper}$	$-2.4821 \cdot 10^{-5}$
$b_{2,2,upper}$	$5.0112 \cdot 10^{-8}$
$b_{1,2,upper}$	$-1.3034 \cdot 10^{-4}$
$b_{0,2,upper}$	$7.8329 \cdot 10^{-3}$
$b_{2,1,upper}$	$-1.14401 \cdot 10^{-5}$
$b_{1,1,upper}$	$3.39792 \cdot 10^{-2}$
$b_{0,1,upper}$	$-6.28364 \cdot 10^{1}$
$b_{2,0,upper}$	$2.52620443 \cdot 10^{-3}$
$b_{1,0,upper}$	-6.06760282
$b_{0,0,upper}$	$6.91714082 \cdot 10^{4}$
$b_{2,3,lower}$	$-4.4177 \cdot 10^{-11}$
$b_{1,3,lower}$	$1.1593 \cdot 10^{-7}$
$b_{0,3,lower}$	$-2.4821 \cdot 10^{-5}$
$b_{2,2,lower}$	$5.0112 \cdot 10^{-8}$
$b_{1,2,lower}$	$-1.3034 \cdot 10^{-4}$
$b_{0,2,lower}$	$7.8329 \cdot 10^{-3}$
$b_{2,1,lower}$	$-1.14401 \cdot 10^{-5}$
$b_{1,1,lower}$	$3.39792 \cdot 10^{-2}$
$b_{0,1,lower}$	$-6.28364 \cdot 10^{1}$
$b_{2,0,lower}$	$2.52620443 \cdot 10^{-3}$
$b_{1,0,lower}$	-6.06760282
$b_{0,0,lower}$	$6.91714082 \cdot 10^{4}$
c_2	0
c_1	0
c_0	$3.208 \cdot 10^{3}$
$V_{cutoff,ADC}$	2740

Table 5-2: Parameters used with the state-of-charge algorithm for the NiMH battery.

5.1.3 Results and Discussion

After the system used to compute the state-of-charge and the corresponding hardware components have been described, a full battery discharge with determination of the $\widehat{SOC}$ is conducted. These discharges are presented and discussed in the following. As stated in Chapter 4, the state-of-charge is most interesting with respect to the $\widehat{SOC}_{max}$. Therefore the state-of-charge approximations that are illustrated in the following are represented in percent of $\widehat{SOC}_{max}$.

The state-of-charge computed during a constant discharge of the lead acid battery is illustrated in Figure 5-2. The current drawn from the battery is approximately 57.7 mA.

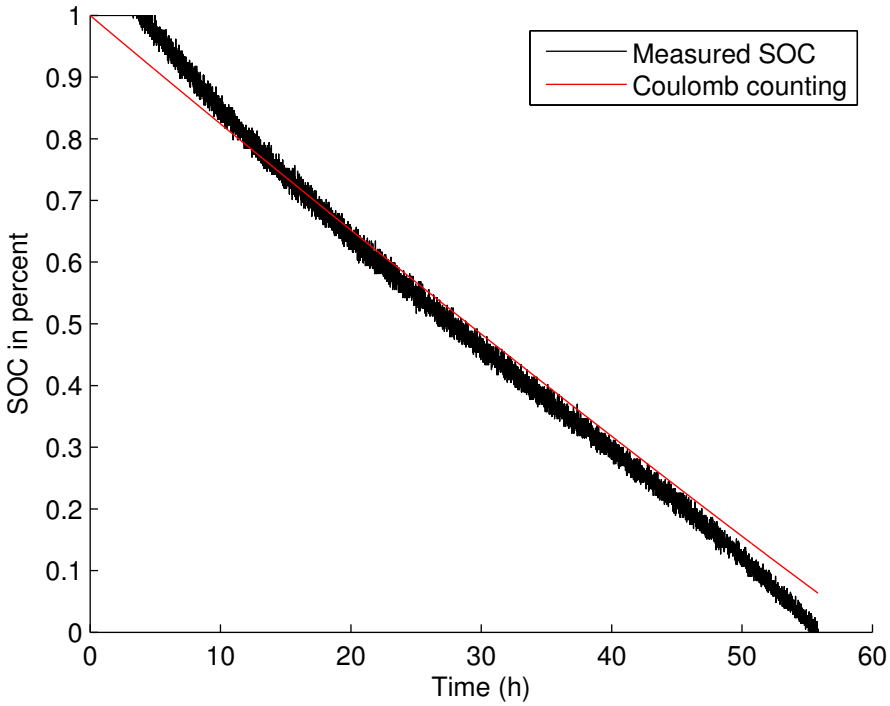

Figure 5-2: LA battery powered state-of-charge calculation with a constant load of approximately 57.7 mA.

As is expected, the state-of-charge is close to a linear function. At the beginning of the discharge the algorithm computations result in a $\widehat{SOC}$ of 100 percent. This behavior is also present in the approximations of the depth-of-discharge in the Figures A-2, A-3 and A-4. The battery is fully charged hence the depth-of-discharge equals zero percent. Nevertheless a part of the approximation considers bigger voltages and for such values results in a negative depth-of-discharge. Results exceeding the state-of-charge limits that in conclusion are bigger than $\widehat{SOC}_{max}$ or smaller than $\widehat{SOC}_{min}$, are limited to the closest boundary. This is also the explanation for the state-of-charge staying at its maximal value at the beginning of the discharge. The voltage corresponding to the discharging of the battery is given in Figure 5-3. Since the chosen load generates a current that is close to the current used to obtain Figure A-2, this figure is used as a reference. As the previously given explanation suggests the battery voltage at the beginning of the discharge is clearly above the value at which the depth-of-discharge becomes zero.

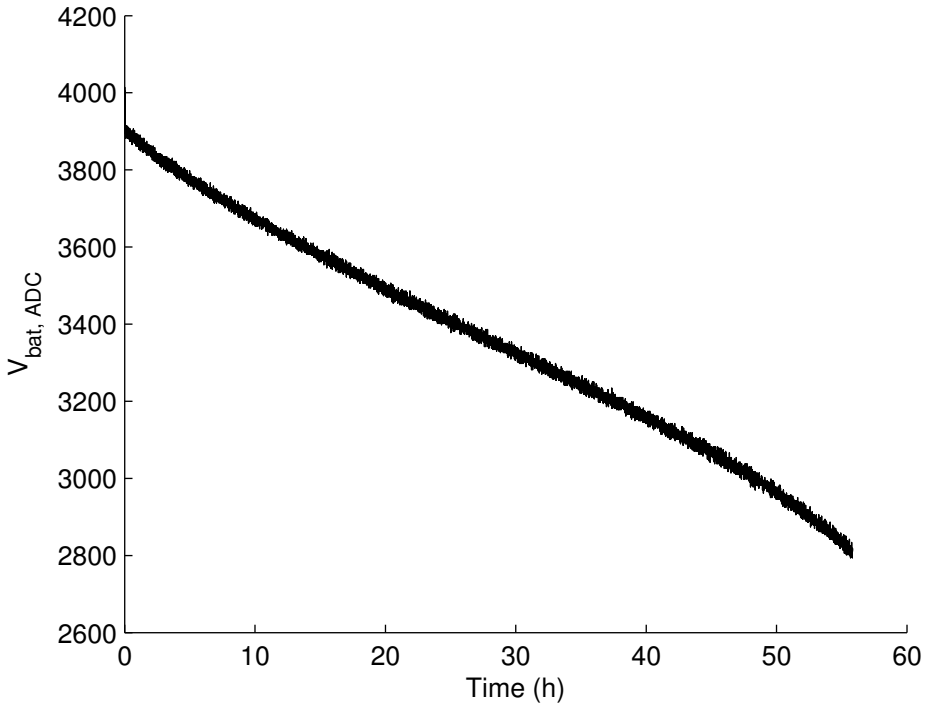

Figure 5-3: LA battery voltage corresponding to the state-of-charge approximation depicted in Figure 5-2.

The state-of-charge for a variable load and in combination with a lead acid battery is illustrated in Figure 5-4. A similar behavior as discussed with a constant load can be observed here. Again the approximation follows a linear function and a deviation of approximately ten percent from the state-of-charge obtained with coulomb counting is observable.
Nevertheless a difference from the discharge with the constant load is the reflection of the changes of the load in the computed state-of-charge. This is although expected since four different polynomials approximate the depth-of-discharge in this case.

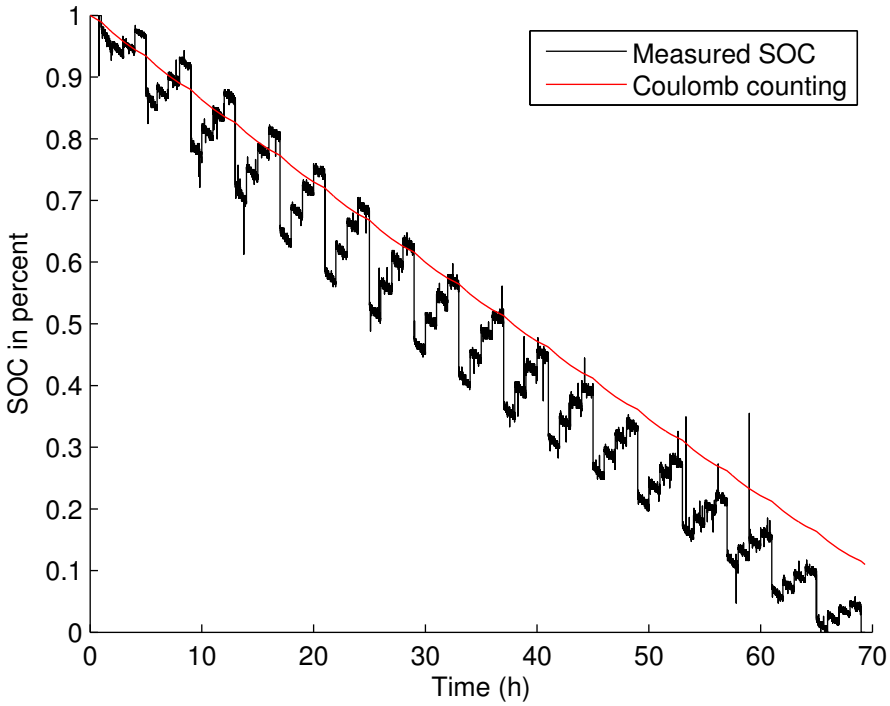

Figure 5-4: LA battery powered state-of-charge calculation with a variable load.

In the following the NiMH battery is considered. The previously employed lead acid battery has shown, together with the corresponding state-of-charge approximation that the algorithm operates as expected.
Again, as a best case scenario, a constant load is used first. Afterwards the state-of-charge is computed for the discharging of the battery through a variable load.

The constant discharge is represented in Figure 5-5 for a current of approximately 44.5 mA.

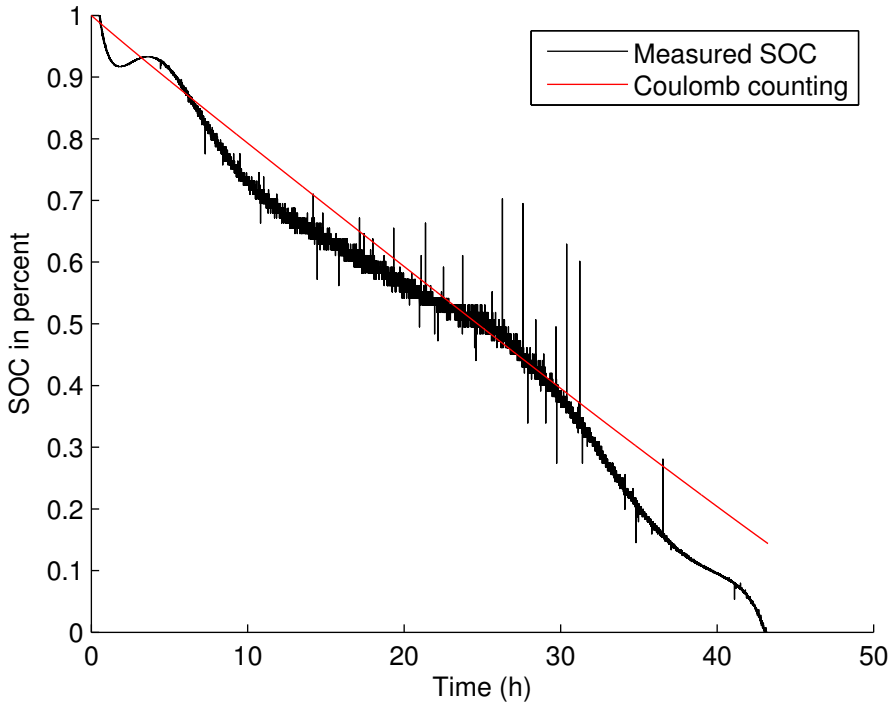

Figure 5-5: NiMH battery powered state-of-charge calculation with a constant load of approximately 44.5 mA.

Several characteristics present in the state-of-charge approximation are discussed in the following. These are the sudden changes of the state-of-charge that are located between 25 h and 35 h, the variation from the expected straight line and finally the effect of the threshold on the $\widehat{SOC}$.

The battery voltage that is used to determine the state-of-charge is illustrated in Figure 5-6. It shows that the mentioned **sudden changes of the state-of-charge** are also present in the voltage measurements. Furthermore, because of the curve of the discharge profiles a small increase in voltage near the plateau (or a small error in the measured voltage) yields a much bigger deviation from the state-of-charge. This is a consequence of the mentioned voltage plateau.
For example the voltage peak occurring at approximately 27 h is equal to a voltage that is present after 10 h. Since the depth-of-discharge linearly increases with time, it is clear that the state-of-charge algorithm considers the voltage peak present as 27 h to belong to a battery that contains more charges. Therefore measurement errors occurring close to the voltage plateau have a considerable impact on the state-of-charge approximation.

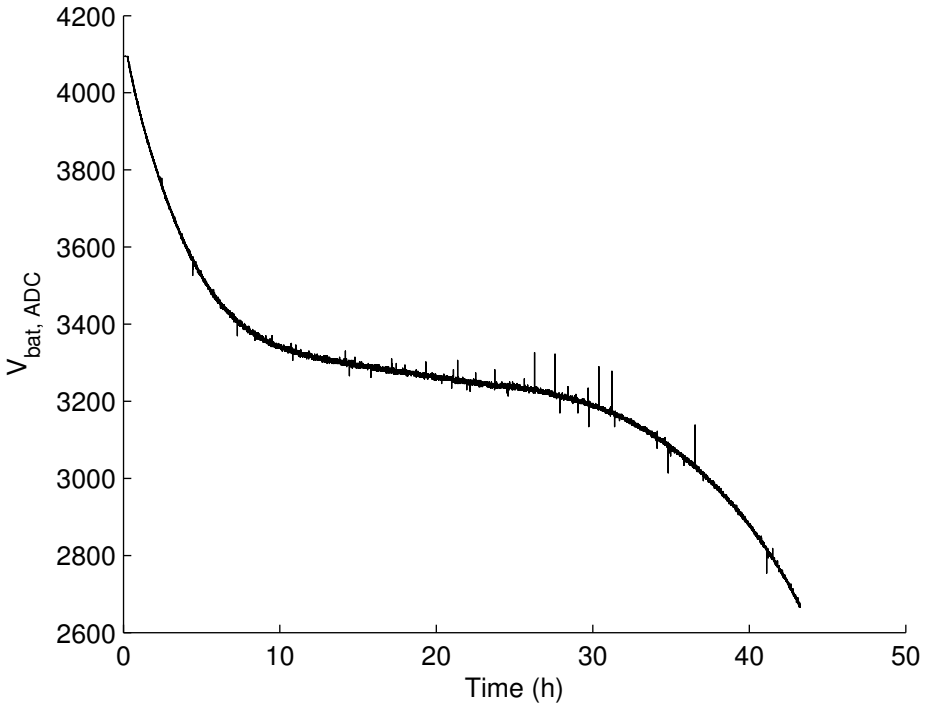

Figure 5-6: NiMH battery voltage corresponding to the state-of-charge approximation depicted in Figure 5-5.

The **variation of the state-of-charge with respect to an ideal straight line** is due to the approximation by polynomials of third order. Figures 4-8, D-5, D-6, D-7 and D-8 show that a certain variation from a straight line is expected since the approximation also over- and undershoots the depth-of-discharge. A total of three overshoots and three undershoots are expected. These are also present in the computed state-of-charge and are therefore a result of the approximation method.

At last, for the constant discharge, the threshold voltage is discussed. It is for the given current located at approximately 3248 which is equal to 3.85 V. The threshold is also only slightly noticeable in Figure 5-5. This shows the closeness of the two polynomials at V_{thresh}.

Figure 5-7 illustrates the approximation of the state-of-charge for a variable load. The observations made for the constant discharge can also be made here. The state-of-charge shows some variations that are due to the cubic approximation of the depth-of-discharge. Furthermore the peaks that are present at 30 h and 40 h are a consequence of the measured voltage. Finally the switches between the two segments are only slightly noticeable.

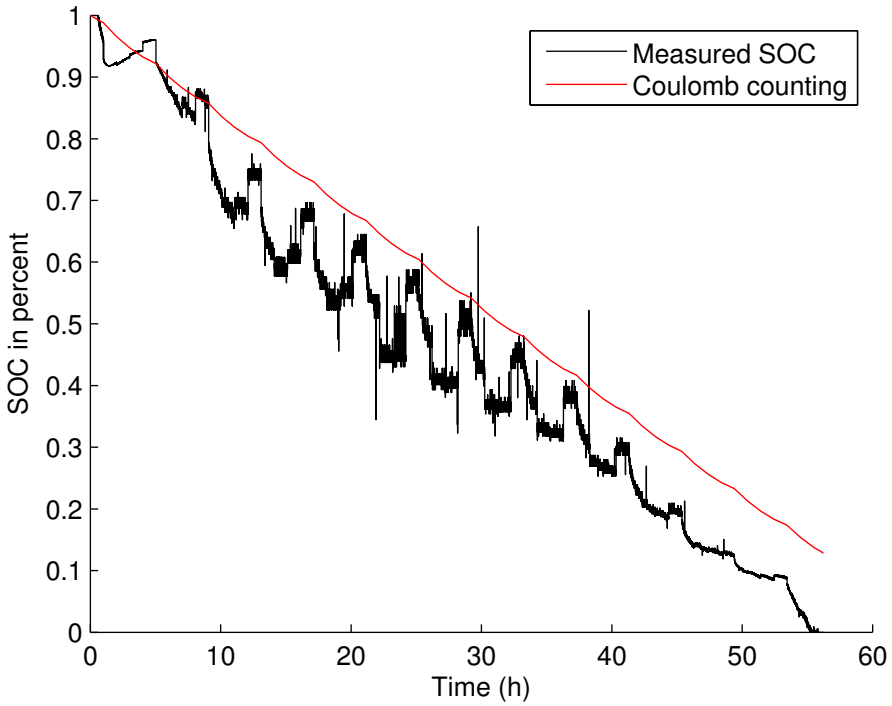

Figure 5-7: NiMH battery powered state-of-charge calculation with a variable load.

The value of the threshold voltages corresponding to the different states of the variable load are presented in Table 5-3. Due to the varying current the threshold voltages are overstepped multiple times before the state-of-charge is finally computed only with the lower polynomial. The region in which this occurs is between 17.5 h and 33.3 h. At last Figure 5-7 shows that the state-of-charge is perturbed by the varying load but remains close to the expected state-of-charge.

Load (mA)	Threshold voltage (internal representation)	Threshold voltage (V)	$t_{V_{thresh},first}$ (h)	$t_{V_{thresh},last}$ (h)
46.0	3244	3.85	21.5	33.3
38.6	3259	3.86	17.7	29.8
30.7	3265	3.86	17.5	29.8
22.9	3259	3.86	17.7	29.8

Table 5-3: Threshold voltages corresponding to the loads used to obtain Figure 5-7. The instant $t_{V_{thresh},first}$ denotes the first time instant at which the threshold voltage has been reached. Accordingly $t_{V_{thresh},last}$ denotes the last time instant at which the threshold voltage is overstepped.

6

Conclusion and Further Work

In Chapter 4 this thesis presented an adaptation of the algorithm published in [4] to operate with a NiMH battery while maintaining the capability to compute the state-of-charge for lead acid batteries.
Furthermore the algorithm has been evaluated for a constant and a time-dependent load in Chapter 5. In addition to the observation of the expected behavior of the state-of-charge approximation (due to the cubic approximation of the depth-of-discharge), the crossing of the threshold voltage is nearly unnoticeable. Therefore the threshold voltage with all of its consequences (splitting of the depth-of-discharge; additional polynomial evaluation during system runtime) allows a reasonably accurate approximation of the state-of-charge for a NiMH battery with third order polynomials.

6.1 Present Implementation

Beginning with different discharge profiles this thesis presented a method of modifying the state-of-charge algorithm to consider the discharging process of NiMH batteries. Since the depth-of-discharge approximating polynomial was not precise enough, a threshold voltage V_{thresh} has been introduced. With this parameter of the considered battery the order of the polynomial was kept low and therefore the algorithm's execution time did not substantially increase. This then lead to using floating point numbers which also allow a higher precision.

An opportunity to improve the herein described state-of-charge algorithm is to also consider the charging process of the NiMH battery. Thereby the utility and the availability of the system can be further increased.
The exact charging process has to be analyzed and divided into different battery states that allow its description. Afterwards the algorithm can be extended with this new functionality and for example a field test can be conducted.

6.2 Performance Evaluation of the adapted State-of-Charge Algorithm

The modification applied to the algorithm published in [4] contribute in the ways stated in Chapter 1.
First the algorithm successfully approximates the state-of-charge of a NiMH battery for a constant and time-dependent load. Second the algorithm, as stated in the following only needs a small amount of time to compute the state-of-charge on a resource constrained platform. By using the values given in Table 4-1 an approximate execution time of $(4+1) \cdot 441 \cdot 10^{-6}s + 1 \cdot 664 \cdot 10^{-6}s = 2.869ms \simeq 3ms$ can be calculated. If the herein modified state-of-charge algorithm is executed once a second, $\frac{1 \cdot s - 3 \cdot 10^{-3}s}{1 \cdot s} = 99.7\%$ of the system's runtime can still be used to perform other functions. As can be perceived in the discharge profiles (Figure 4-3), the state-of-charge develops over hours and does therefore not need to be calculated that often.

Nevertheless the work done for this thesis can be continued as stated hereafter. Although three possibilities are given in the following, they all consider the precision of the results obtained in Chapter 5.

- First a polynomial of higher order than three can be used to approximate the depth-of-discharge. While this yields a closer approximation of the state-of-charge, the increase in precision needs to be weighted against the increase in the duration of the computation.
- Second if the exact loads are known an approximation with splines can be considered. The splines considered in this thesis only divided the depth-of-discharge in two parts. With a higher amount of splines the approximation can be made arbitrarily close to the measured discharge. Because of the meeting points that might shift if the splines are approximated, it seems best to determine the discharge profiles for all the loads that appear during the system runtime. These are then approximated with splines and then implemented with a lookup table. Thereby a highly precise approximation results with smooth crossings between segments of a spline.
  This possibility reduces the duration of the determination of the state-of-charge. Its impact on the system memory depends on the amount of discharge profiles represented in the lookup table.
- Third as a similar approach to the previous one is to use lookup tables that can again achieve very accurate results. When using enough data points (coefficients a_k for many different loads) the interpolation between two of them can be performed during system runtime and with a linear polynomial. Thereby the algorithm becomes faster in its execution but has a much larger impact on system memory.

Furthermore it is yet unknown how the threshold voltage and the coefficients a_k behave for currents that are smaller than 20 mA or bigger than 60 mA.
In order to increase the current range for which the algorithm reliably determines the state-of-charge, more discharge profiles can be conducted. Thereby it is possible to obtain a deeper understanding of the threshold voltage and to validate or to reject the choosing of a second order polynomial to approximate V_{thresh}.

6.3 Battery Differences

Finally a short description of the observed differences between the two batteries is given.

- First the behavior at the beginning of the discharge profile is very different. As can be taken from Figure A-1 the lead acid battery reacts to the magnitude of the drawn current by a voltage drop at its terminals and a thereon depending slope. On the opposite the NiMH battery, as is depicted in Figure 4-3, has a battery voltage of approximately 4.2 V regardless of the drawn current.
- Second the lead acid battery has a more or less constant voltage slope until nearly all of its charges have been spent. The voltage of the NiMH battery falls very steeply at the beginning and then reaches a plateau where it is close to constant. Then after the battery's capacity has been reduced by a certain amount of charges, the voltage V_{bat} again falls off quicker. At this point the battery still contains over a third of its charges.

A

Lead Acid Battery

Similar steps as are performed for the NiMH battery in this thesis are in the following conducted for the lead acid battery. Therefore the discharge profiles are given Figure A-1.

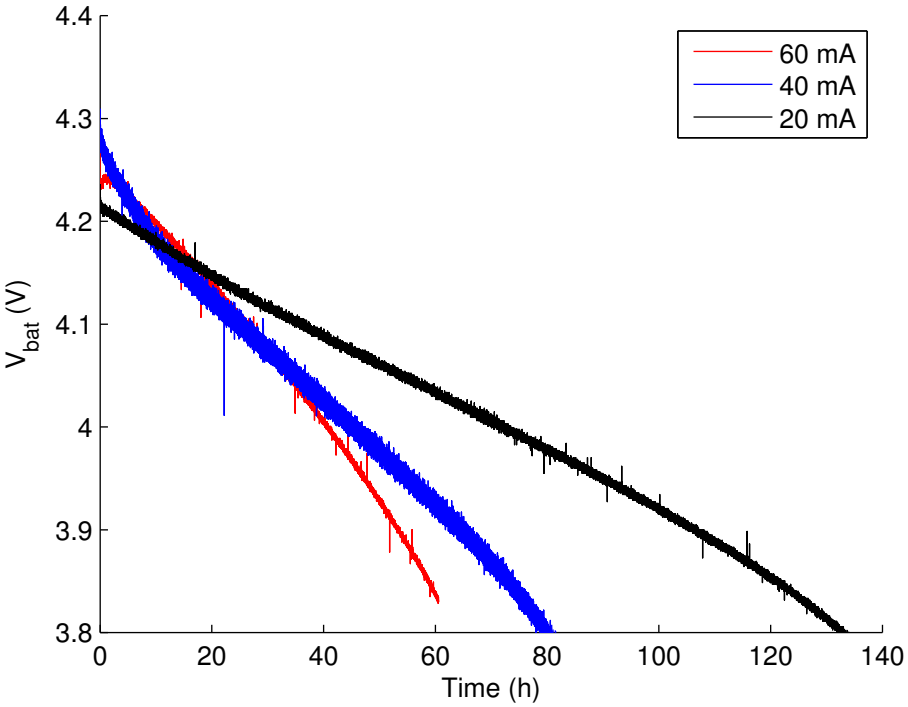

Figure A-1: Discharge profiles of the lead acid battery at currents of 60 mA, 40 mA and 20 mA respectively.

Again the discharge profiles are approximated with linear, quadratic and cubic polynomials. The approximation and the respective depth-of-discharge are given in Figures A-2, A-3, and A-4. Since the algorithm published in [4] has been elaborated for

a lead acid battery, its method is directly applied herein.

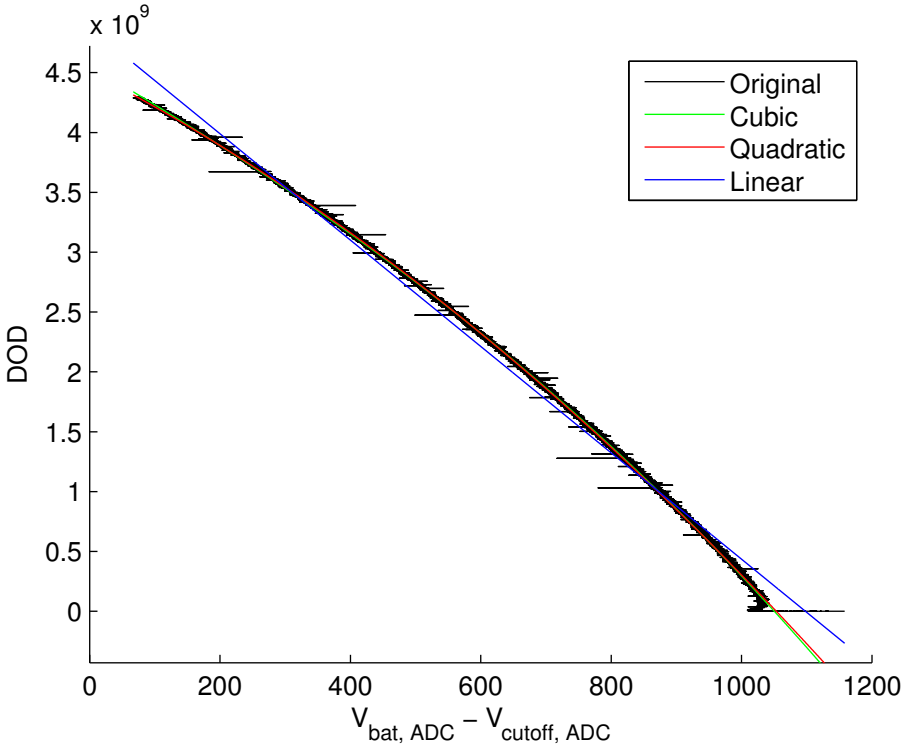

Figure A-2: Approximation of the lead acid discharge profile at 60 mA.

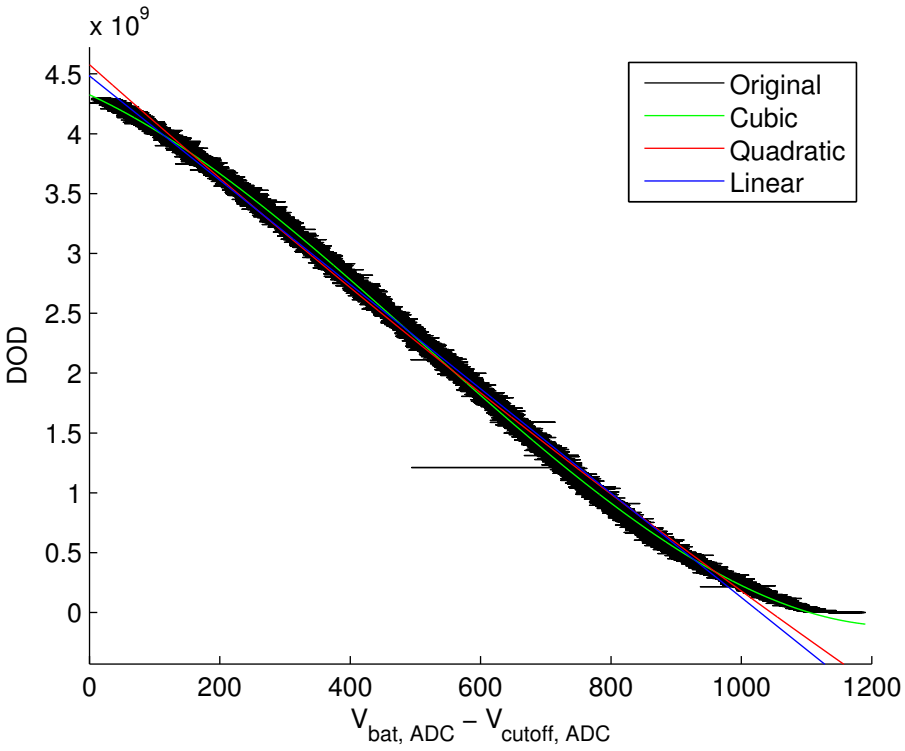

Figure A-3: Approximation of the lead acid discharge profile at 40 mA.

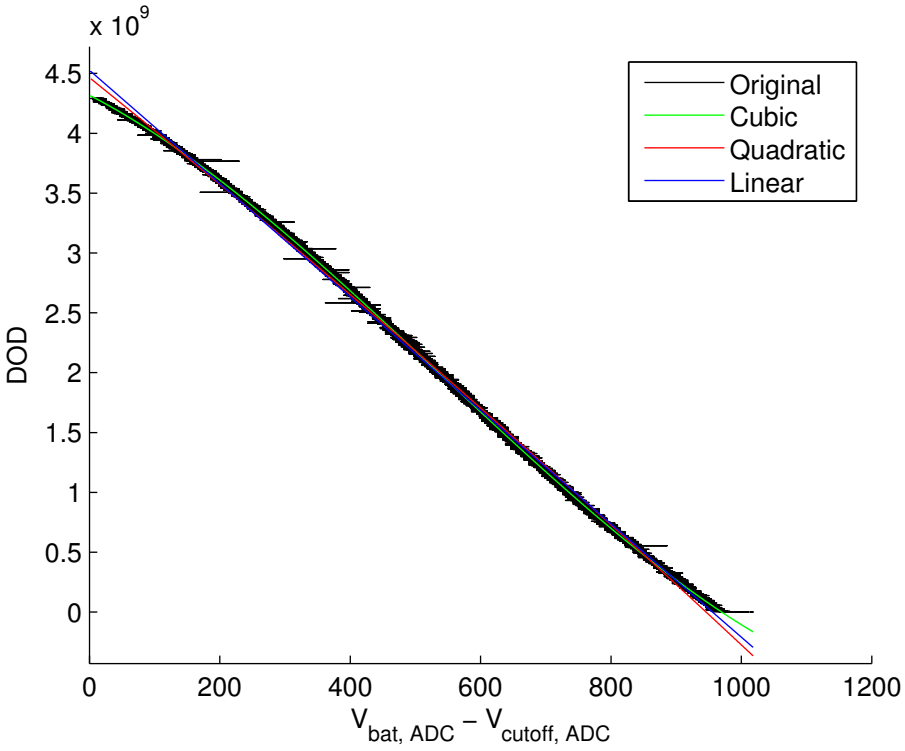

Figure A-4: Approximation of the lead acid discharge profile at 20 mA.

As is shown in Chapter 5 the performance of the state-of-charge algorithm with a lead acid battery is used as a reference for its operation with a NiMH battery. Therefore both batteries have been used in combination with a constant and a time-dependent load. To shorten the duration of the discharging of the battery the loads have been lessened for the lead acid battery in order to draw a bigger current. Those are represented in table A-1 for the variable load. The resistive value used for the constant load is represented in Table 3-2.

Load designation	Resistor	Switched resistor	LEDs	TinyNode184	Drawn current (mA)
LC	yes	no	no	yes	32.4
LMC	yes	no	yes	yes	40.1
HMC	yes	yes	no	yes	48.5
HC	yes	yes	yes	yes	55.9

Table A-1: Representation of the four different loads used in combination with the lead acid battery. The table also contains the hardware employed to generate those and the current they draw. Those currents have been measured at approximately 4 *V with consideration of the effect of the measurement device on the system.*

B

Hardware Schematic

The complete schematic of the hardware setup is depicted in Figure B-1.

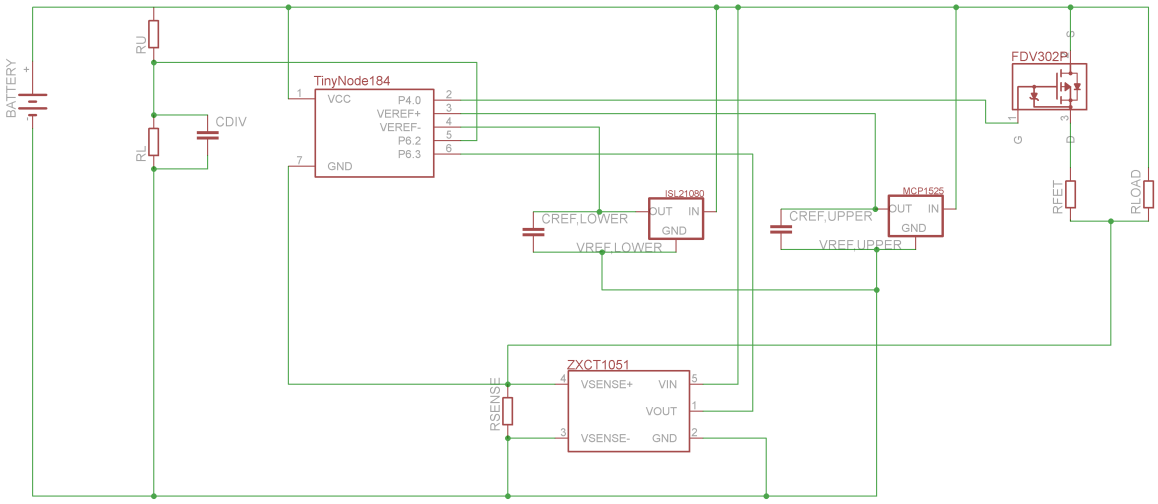

Figure B-1: Complete schematic of the hardware setup.

C

Smoothing Capacitance

In this Appendix the time constant used in combination with the voltage dividers is derived.

The battery voltage is the interesting value. Therefore if the battery voltage is applied to the voltage divider at $t_0 = 0$ s, the time constant of 500 ms is chosen so as to indicate the instant at which the output voltage reaches approximately $p = 98\%$ of its final value.

The voltage divider is depicted in Figure 3-4. The following definitions are used to compute the time constant.

V_u	is the voltage at the terminals of the upper resistor R_u.
V_l	is the voltage drop that appears across the lower resistor R_l.
V_C	is the voltage at the capacitance C_{div}.
I_u	is the current that flows through the upper resistor R_u.
I_l	is the current that results from the voltage drop V_l.
I_C	is the current that charges the capacitance C_{div}.

Equations C.1, C.2, C.3, C.4 and C.5 hold for this voltage divider.

$$I_u = I_l + I_C \tag{C.1}$$

$$\begin{aligned} V_{bat}(t) &= V_u(t) + V_l(t) \\ \iff V_{bat}(t) &= V_u(t) + V_C(t) \end{aligned} \tag{C.2}$$

$$I_u(t) = \frac{V_u(t)}{R_u} \tag{C.3}$$

$$I_l(t) = \frac{V_l(t)}{R_l} \tag{C.4}$$

$$C_{div} \cdot \frac{dV_C\left(t\right)}{dt} = I_C\left(t\right) \tag{C.5}$$

With these relations between the current and voltages in the voltage divider, Equation C.6 is derived.

$$\begin{aligned} & \tfrac{V_{bat}(t)-V_C(t)}{R_u} && = \tfrac{V_C(t)}{R_l} + C_{div} \cdot \tfrac{dV_C(t)}{dt} \\ \iff \quad & \tfrac{V_{bat}(t)}{C_{div} \cdot R_u} && = \tfrac{R_u+R_l}{C_{div} \cdot R_u \cdot R_l} \cdot V_C\left(t\right) + \tfrac{dV_C(t)}{dt} \end{aligned} \tag{C.6}$$

To solve the obtained differential equation, it is transformed into the Laplace domain and further manipulated in Equation C.7.
As mentioned previously the battery voltage is applied to the voltage divider at $t_0 = 0$ s and is considered constant. This simplification is valid because of the comparably slow variation of the battery voltage.

$$\begin{aligned} & \tfrac{V_{bat}}{s \cdot C_{div} \cdot R_u} && = \left(\tfrac{R_u+R_l}{C_{div} \cdot R_u \cdot R_l} + s\right) \cdot V_C\left(s\right) \\ \iff \quad & V_C\left(s\right) && = \tfrac{1}{s \cdot C_{div} \cdot R_u} \cdot \tfrac{1}{\frac{R_u+R_l}{C_{div} \cdot R_u \cdot R_l}+s} \cdot V_{bat} \\ \iff \quad & V_C\left(s\right) && = \tfrac{\frac{R_u+R_l}{C_{div} \cdot R_u \cdot R_l}}{s\left(s+\frac{R_u+R_l}{C_{div} \cdot R_u \cdot R_l}\right)} \cdot \tfrac{R_l}{R_u+R_l} \cdot V_{bat} \end{aligned} \tag{C.7}$$

By back transformation into the time domain, the result of the differential equation is obtained as is stated in Equation C.8.

$$V_C\left(t\right) = \left(1 - e^{-\frac{R_u+R_l}{C_{div} \cdot R_u \cdot R_l} \cdot t}\right) \frac{R_l}{R_u + R_l} \cdot V_{bat} \tag{C.8}$$

With the values chosen in this section and in Chapter 3, the value of the capacitance C_{div} is computed in Equation C.9.

$$\begin{aligned} & && V_C\left(\tau\right) && = p \cdot \tfrac{R_l}{R_u+R_l} \cdot V_{bat} \\ \iff \quad & \left(1 - e^{-\frac{R_u+R_l}{C_{div} \cdot R_u \cdot R_l} \cdot \tau}\right) \cdot \tfrac{R_l}{R_u+R_l} \cdot V_{bat} && = p \cdot \tfrac{R_l}{R_u+R_l} \cdot V_{bat} \\ \iff \quad & \left(1 - e^{-\frac{R_u+R_l}{C_{div} \cdot R_u \cdot R_l} \cdot \tau}\right) && = p \\ \iff \quad & e^{-\frac{R_u+R_l}{C_{div} \cdot R_u \cdot R_l} \cdot \tau} && = 1 - p \\ \iff \quad & -\tfrac{R_u+R_l}{C_{div} \cdot R_u \cdot R_l} \cdot \tau && = \ln\left(1-p\right) \\ \iff \quad & \tfrac{R_u+R_l}{C_{div} \cdot R_u \cdot R_l} \cdot \tau && = \ln\left(\tfrac{1}{1-p}\right) \\ \iff \quad & C_{div} && = \ln(\tfrac{1}{1-p})^{-1} \tfrac{R_u+R_l}{R_u \cdot R_l} \cdot \tau \end{aligned} \tag{C.9}$$

Finally, values approximately equal to those given in Table 3-1 result.

D

Further Plots

This appendix contains the plots of the approximations of the depth-of-discharge for the NiMH battery for the remaining loads. Due to their similarity to Figure 4-4 they are not depicted in the respective Chapter.

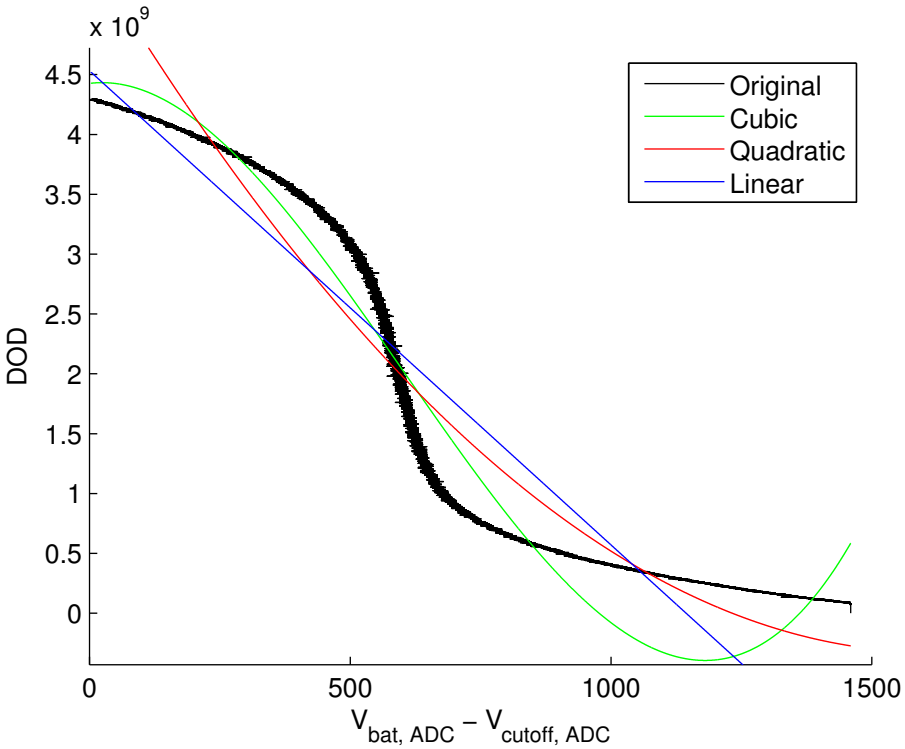

Figure D-1: Depth-of-Discharge of the NiMH battery with a load of 50 mA.

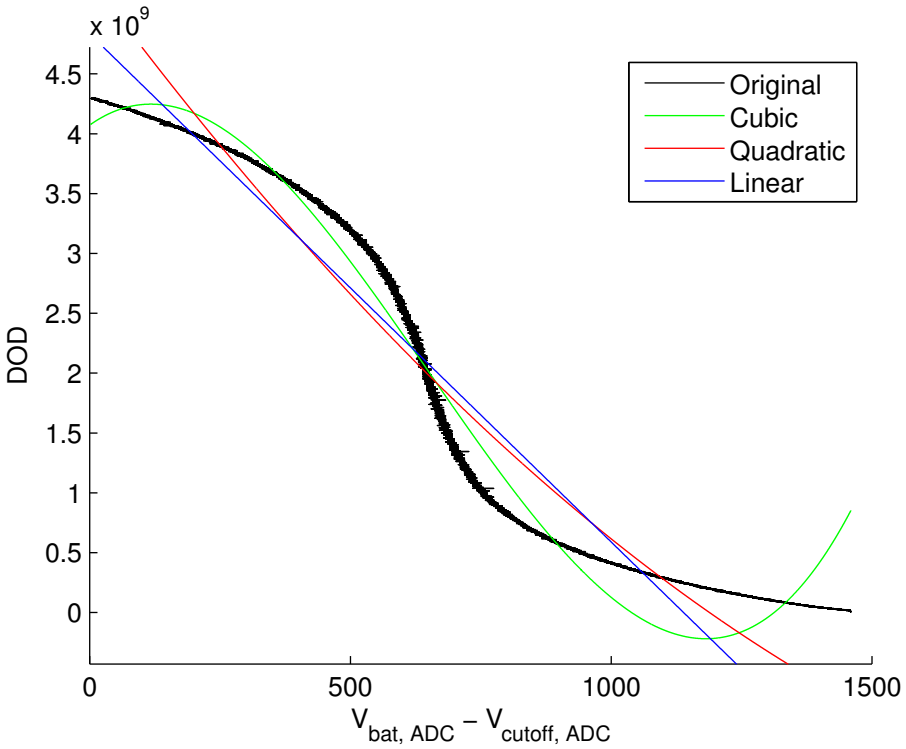

Figure D-2: Depth-of-Discharge of the NiMH battery with a load of 40 mA.

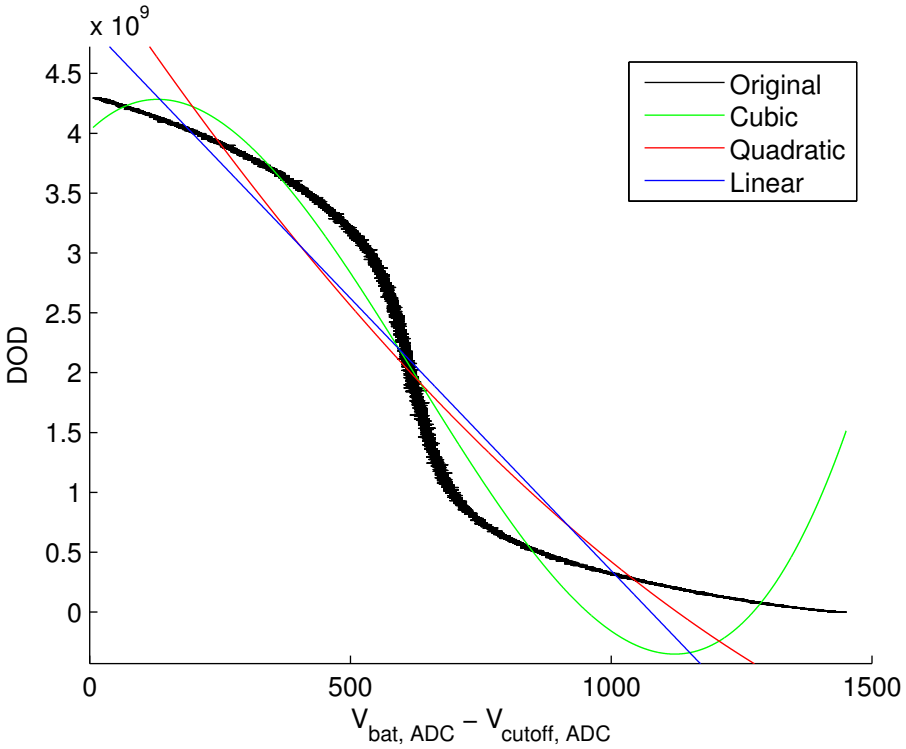

Figure D-3: Depth-of-Discharge of the NiMH battery with a load of 30 mA.

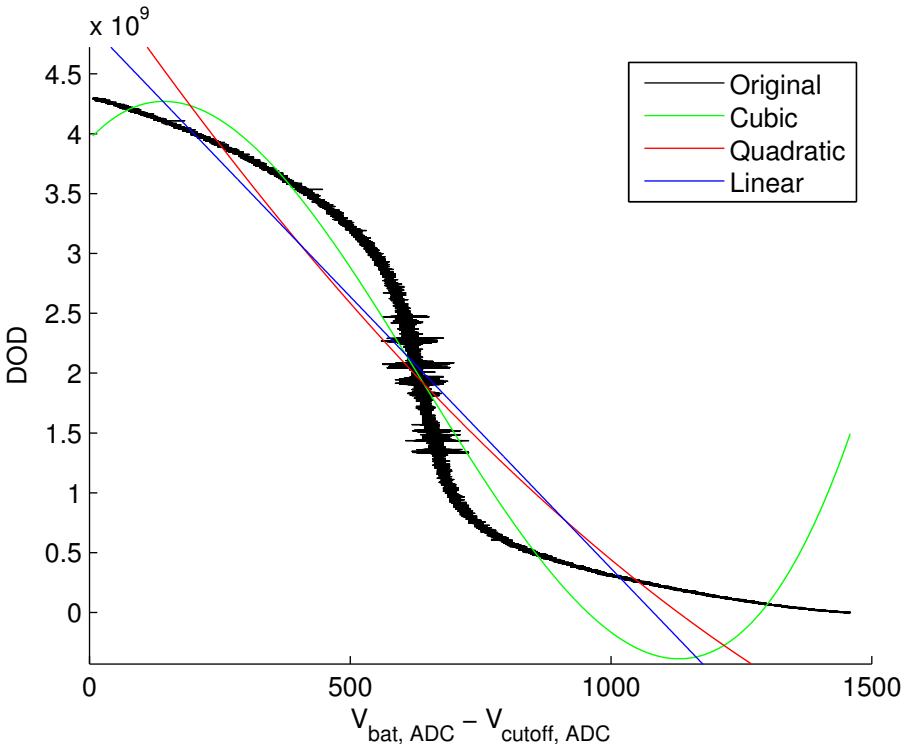

Figure D-4: Depth-of-Discharge of the NiMH battery with a load of 20 mA.

Figures D-5, D-6, D-7 and D-8 depict the respective depth-of-discharge curves and their different approximations that are obtained with the second approach presented in this thesis.

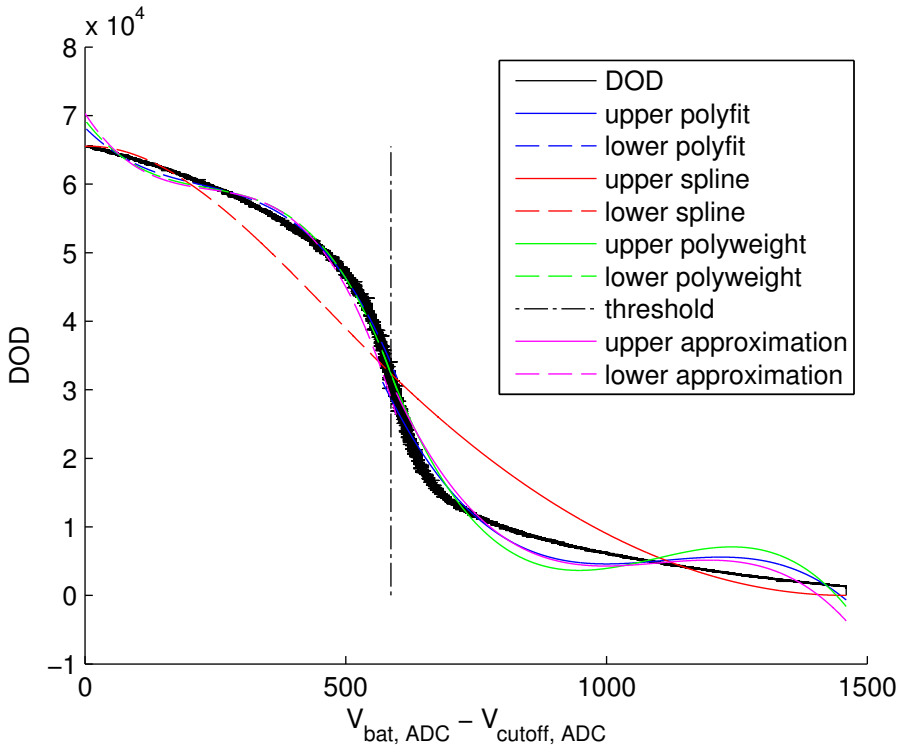

Figure D-5: Depth-of-Discharge and different approximations for the 50 mA load.

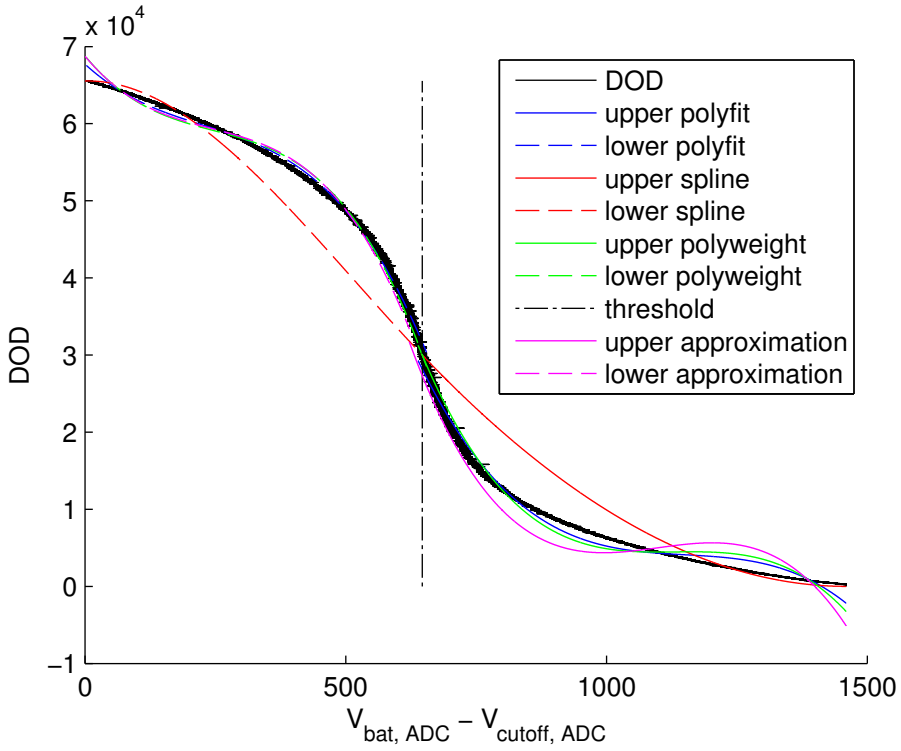

Figure D-6: Depth-of-Discharge and different approximations for the 40 mA load.

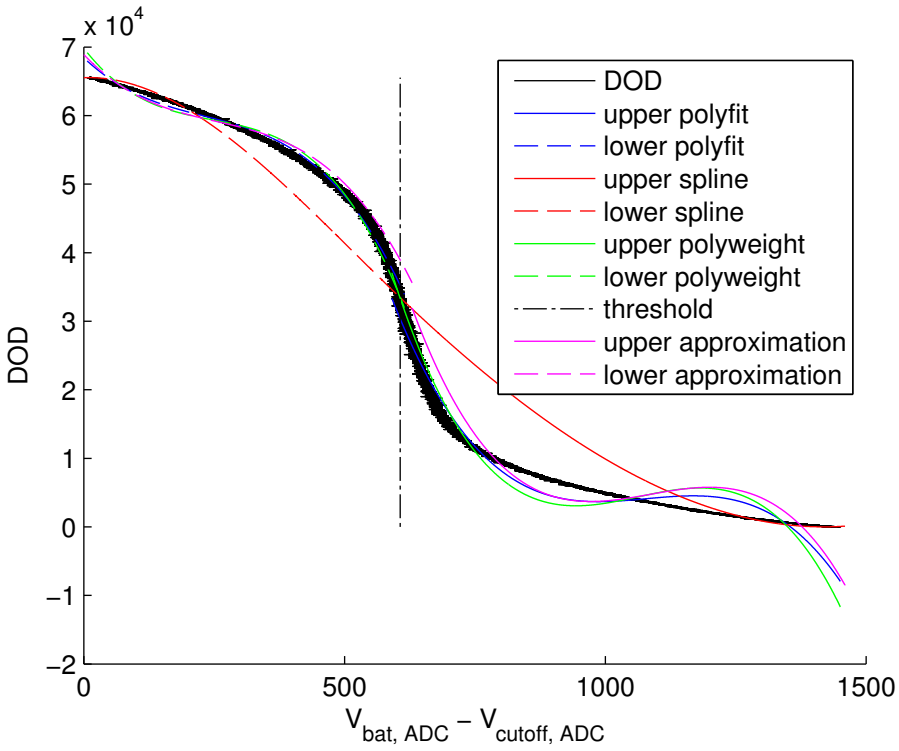

Figure D-7: Depth-of-Discharge and different approximations for the 30 mA load.

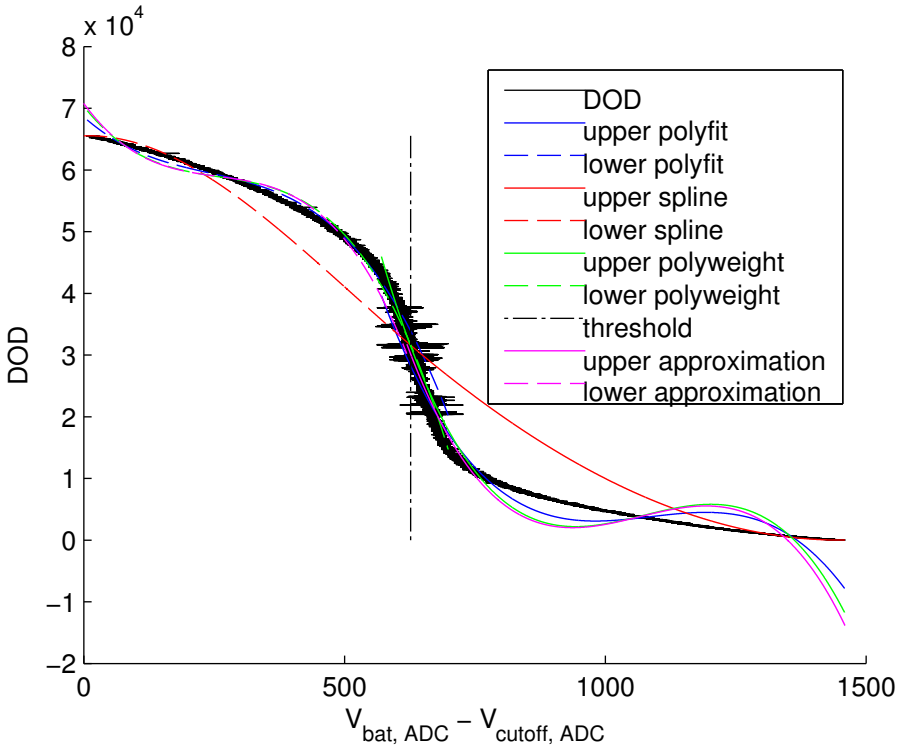

Figure D-8: Depth-of-Discharge and different approximations for the 20 mA load.

Bibliography

[1] Horner's method, URL: en.wikipedia.org/wiki/Horner's_method, June 2013.

[2] TinyOS Overview, URL: http://docs.tinyos.net/tinywiki/index.php/TinyOS_Overview, June 2013.

[3] Accumulatorenfabrik Sonnenschein GmbH, Eichborndamm 129-139, D-13403 Berlin. Germany. *Sonnenschein A504/3,5 S Datenblatt*, 2008.

[4] Bernhard Buchli, Daniel Aschwanden, and Jan Beutel. Battery State-of-Charge Approximation for Energy Harvesting Embedded Systems. In *Lecture Notes on Computer Science. Proc. of 10th European Conference on Wireless Sensor Networks (EWSN 2013)*, pages 179–196, Ghent, Belgium, February 2013. Springer.

[5] Bernhard Buchli, Felix Sutton, and Jan Beutel. GPS-equipped Wireless Sensor Network Node for High-accuracy Positioning Applications. In *Lecture Notes on Computer Science 7158. Proc. of 9th European Conference on Wireless Sensor Networks (EWSN 2012)*, pages 179–195, Trento, Italy, February 2012. Springer.

[6] Fairchild Semiconductor Corporation, 3030 Orchard Parkway, San Jose, CA 95134, U.S.A. *FDV302P, Digital FET, P-Channel*, October 1997.

[7] Intersil, 1001 Murphy Ranch Road, Milpitas, CA 95035. *300nA Nanopower Voltage References*, May 2010.

[8] David Linden and Thomas B. Reddy, editors. *Handbook of batteries*. McGraw-Hill, 3 edition, 2001.

[9] Microchip Technology Inc., 2355 West Chandler Blvd., Chandler, Arizona, USA 85224-6199. *2.5V and 4.096 Voltage References*, February 2005.

[10] Kaspar Nipp and Daniel Stoffer. *Lineare Algebra, Eine Einführung für Ingenieure unter besonderer Berücksichtigung numerischer Aspekte*. vdf, Hochschul.-Verl. an der ETH Zürich, 5 edition, 2002.

[11] Salman Rogers. Polyfitweighted, URL: http://www.mathworks.ch/matlabcentral/fileexchange/13 polyfitweighted.

[12] Shockfish SA, Rue des Terreaux 10, 1003 Lausanne, Switzerland. *User's Manual*, 1.1 edition, October 2010.

[13] Texas Instruments, Post Office Box 655303, Dallas, Texas 75265. *MSP430 Datasheet*, November 2012.

[14] Zetex Semiconductors plc, Zetex Technology Park, Chadderton, Oldham, OL9 9LL, United Kingdom. *ZXCT1051, Precision wide input range current monitor*, November 2006.