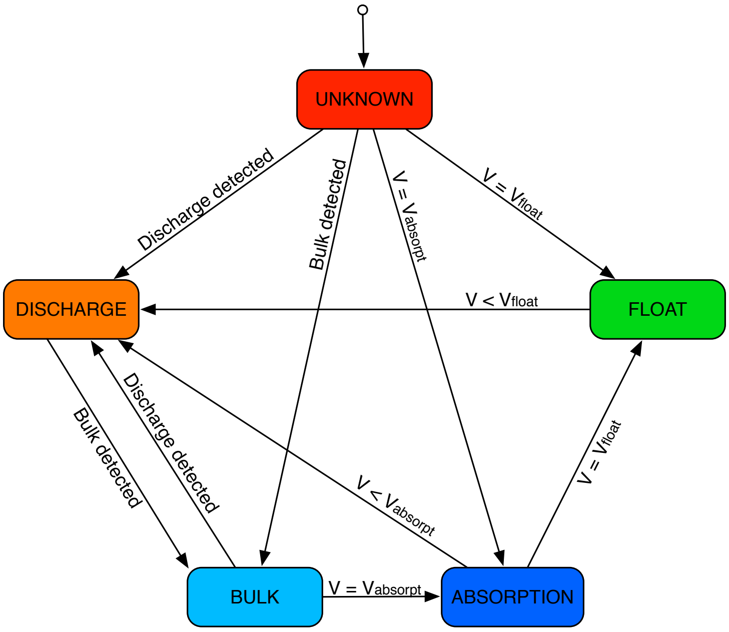
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institut für
Technische Informatik und
Kommunikationsnetze

**Daniel Aschwanden**
asdaniel@ee.ethz.ch
07-907-769

# Battery State of Charge Modeling

## A Lightweight Trace-Based Approximation Approach



Semester Thesis SA-2011-30
September 2011 – March 2012

**Advisor:**
Bernhard Buchli
bbuchli@ethz.ch

**Professor:**
Prof. Dr. Lothar Thiele
thiele@tik.ee.ethz.ch

Computer Engineering Group – TEC
Computer Engineering and Networks Laboratory – TIK
Department of Information Technology and Electrical Engineering – ITET
Swiss Federal Institute of Technology – ETH

## Abstract

Battery state of charge models are either complex and computationally intensive or require special purpose hardware components, such as a coulomb counter, available to the system. This drives up production cost of implementations, especially in the case of low-cost WSN devices. Therefore, this thesis proposes a lightweight battery model which relies fully on closed-loop voltage and drain current measurements. The model's only requirement is the presence of an off-the-shelf charge controller with a well defined cut-off voltage which allows the definition of the 0% state of charge in terms of voltage. In addition, the behavior of the battery under different load situations is learnt from traces and transformed into an approximative model. In addition, the model considers temperature and aging effects implicitly by their effects on the battery voltage. To sum up, this lightweight battery state of charge model requires no hardware modification if voltage and current sensors are available and provides a low operational overhead which favors wireless sensor network deployments. Although, the initial effort for generating the traces and the determination of the model coefficients is not negligible.

**Keywords**

Battery Modeling, State of Charge

## Acknowledgments

During this semester thesis several people supported me. At this place, I would like to thank them.

At first, I am deeply grateful to Ben Buchli for his support, his expertise and his patience in explaining me some basics. During the entire semester he always provided good remarks and excellent advisory – not only in technical details. I really enjoyed the discussions and the iterative design process we had.

Furthermore, I would like to thank Tonio Gsell for excellent advisory regarding the GSN implementation and the backlog framework used in the core stations.

Finally, I would like to express my sincere gratitude to Prof. Dr. Lothar Thiele for providing the opportunity to write this thesis in his research group. I enjoyed the support of the entire Computer Engineering Group (TEC) in various aspects.

Daniel Aschwanden

# Contents

# CHAPTER 1

# Introduction

Wireless Sensor Networks (WSN) are used for a wide variety of applications. In most of the use cases, energy consumption is a crucial design requirement. This is particularly true if a WSN operates in remote areas and thus must rely on batteries due to unavailable power infrastructure. Moreover, these sensor nodes must obey a very strict power budget to successfully fulfill their duties while still providing an acceptable lifetime.

To increase the lifetime, a sensor node may gain energy out of natural resources such as wind and solar. These energy-harvesting sensor nodes try to operate in a energy-neutral state by not using more energy than provided by the energy-harvesting system in the long term. To prevent the death of the sensor node by running out of battery energy, the node may decide to run or not to run energy-hungry tasks while maintaining a well defined baseline operation. In order to achieve such a state, the sensor node must keep track of the energy available in the battery, commonly referred as the *State of Charge* (SoC).

## 1.1. Motivation

Battery state of charge determination is either based on complex, computationally intentsive models or requires special purpose hardware components, such as a coulomb counter, available to the system. This drives up the production costs of an implementation, especially in case of a low-cost WSN device. Therefore, this thesis proposes a lightweight battery model which relies fully on closed-loop voltage and drain current measurements. The behavior of the battery under different load situations is learnt from traces. Although this model requires a considerable effort for generating these traces and extracting the behavior from them, it implies only a minimal operational overhead.

## 1.2. Contribution

The semester thesis contributions are the following:

- modeling of a state of charge approximation of a lead-acid battery based on closed-loop voltage and current measurements.
- design and implementation of a feasible battery monitoring module for the *Permasense/X-Sense* [PermaSense Consortium, 2012] *Core Station* [Buchli et al., 2011], acting as a basis for an energy-dependent task scheduler.

## 1.3. Related Work

Many different battery models are described in the literature. Their approaches vary from very extensive electro-chemical models over analytical models to high-level stochastic models. Electro-chemical models as proposed by Doyle et al. [1993] describe the chemical processes of a battery in a very detailed way which is too complex for WSN devices to compute. Therefore, analytical methods were proposed to abstract from the chemical process and model the battery on a higher level. The simplest one is Peukert's Law [Rakhmatov and Vrudhula, 2001] which describes the non-linearities of a battery by two coefficients. Nevertheless, Peukert's law is limited to constant load scenarios and thus not performing well in WSN applications. For this reason, Rakhmatov and Vrudhula [2001] proposed another analytical model which models the diffusion process in detail. The performance of this model is superior to Peukert's law. In spite of that, it is still based on heavy computational operations and is not suitable for a WSN device. Manwell and McGowan [1993] proposed a battery model called KiBaM which is intuitive and is based on the chemical kinetics process [Jongerden and Haverkort, 2008]. The KiBaM model was slightly modified by Rao et al. [2005] to fit the needs of embedded systems setups.

## 1.4. Outline

Chapter 2 defines the specifications of the system architecture and the functional specifications which have to be considered in the implementation phase. In chapter 3 the battery model is covered in detail. This includes a brief description of the battery processes and a corresponding finite-state machine for tracking these processes. Furthermore, the trace generation and their transformation into the battery state of charge model is specified. Chapter 4 covers the proof of concept implementation of the proposed model. The setup of the evaluation and its results are discussed in chapter 5. Finally, chapter 6 presents a short conclusion.

# CHAPTER 2

# System Requirements

## 2.1. System Architecture

When an energy harvester operates with a battery, the WSN mote should be aware of the energy stored in the battery for efficiently schedule its tasks and to enlarge its lifetime. To keep track of the battery's state of charge various battery models are proposed as outlined in section 1.3. However, these models often rely on special sensors or become too complex in case a charge controller is enforcing a cut-off voltage to prevent deep-discharge cycles. This thesis proposes a lightweight approach which exploits the presence of the charge controller for defining the state of charge in terms of voltage and drain current. Figure 2.1 illustrates the setup which is briefly discussed in the following.

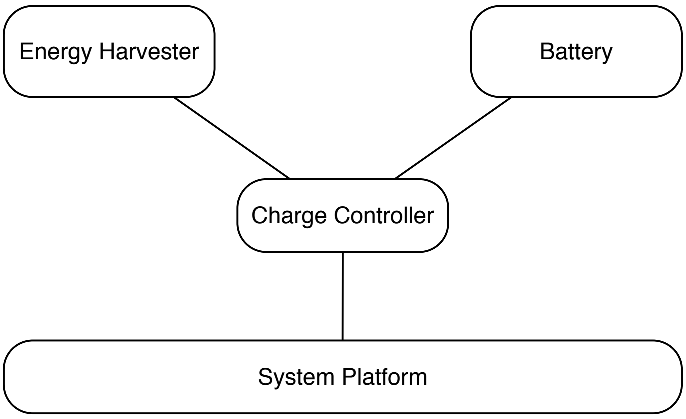


Figure 2.1.: Block diagram of system architecture

### 2.1.1. System Platform

The battery model does not depend on any special purpose hardware for energy monitoring such as a coulomb counter. However, low-cost voltage and current sensors have to be available on the system. Such low-cost off-the-shelf sensors are present on many system platforms and thus not considered as special purpose hardware for energy monitoring. Therefore, no hardware modifications are required on existing platforms where voltage and current sensors are available.

### 2.1.2. Battery

There exists a large number of different batteries suitable for WSN devices. On the one hand, they vary in the chemistry (e.g. Lead-Acid, Nickel-Cadmium, Lithium-Polymer, Lithium-Ion, Lithium-Phosphate, etc. ). On the other hand, they vary in the nominal capacity they provide. The proposed battery model intents to model high-capacity (30Ah+) 12V lead-acid batteries. As this is not a restriction per design, it may also be adopted for other types of batteries operating at different nominal voltages and other types of chemistry.

### 2.1.3. Charge Controller

Batteries operated with energy harvesting systems such as photovoltaic cells or micro wind turbines often employ an off-the-shelf charge controller for cost reasons. The charge controller protects the battery from over-charging and over-voltage and disconnects the load from the battery when the battery voltage drops below a voltage threshold, called cut-off voltage. This is to protect the battery from deep discharge cycles and thus essentially extend the battery's lifetime.

## 2.2. Functional Specifications

A proof of concept of a battery monitoring system with the proposed state of charge battery model is implemented. This implementation covers the following specification:

**Autonomous Measurements and Computation:** The battery monitoring system performs autonomous measurements and automatically updates the information of the battery state, the current remaining time and the state of charge.

**Polling Interface:** The implementation provides an interface for polling information of the current state of charge and the remaining time at the current and at a specified load. The goal is that the information of the battery monitoring system can be extended to allow for energy dependent scheduling.

**Modular Design:** The design of the monitoring system is modular with regard to different battery types. The battery dependent implementation is separated from the functional behavior of the system. This enables the system to adapt the functionality to different batteries within a reasonable time or to refine the battery model independent of the interface and measurement implementation.

## 2.3. Host Platform Requirements

**Charge Controller** is the essential hardware component which has to be present in the setup. Furthermore, the cut-off voltage of the charge controller must be well-known and independent of temperature. In addition, the internal resistance of the charge controller must be known.

**Voltage and Current Sensors** must be present on the system for measuring the system's input voltage and its drain current.

**Cable Resistance** must be known because the systems's input voltage is not equal to the battery's voltage in general. Therefore, the voltage drop of the cable must be considered for an accurate modeling of the battery.

# CHAPTER 3

# Model

The battery model discussed in this chapter can be classified as a trace-based state of charge approximation model. The model can achieve battery state of charge estimations with measurements of low-cost voltage and current sensors combined with a finite-state machine. The battery model is based on battery characterization using discharge and charge traces.

## 3.1. Battery Charge and Discharge Classification

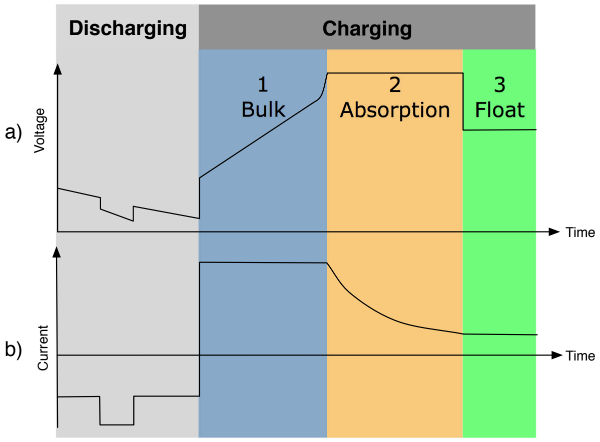


Figure 3.1.: Qualitative illustration of the charging and discharging process

The behavior of a battery can be classified into two main processes: charging and discharging. The charging process can be further divided into three sub-phases. The charging and discharging processes are illustrated in figure 3.1 and briefly discussed in the following.

**Discharging.** During *discharging* the battery is drained with a certain current which defines the battery discharge rate. The voltage of the battery is monotonically decreasing if the battery is drained with a constant current. If there is a sudden increase of the load in terms of power, the drain current increases and the battery voltage drops as illustrated in figure 3.1a). In contrast, if there is a sudden decrease of the load, the drain current will decrease and the battery voltage will increase accordingly.

**Bulk.** The first battery charging sub-phase is commonly called the *bulk* phase and is shown in phase 1 of figure 3.1a) and b). In an ideal charging case with a stable charging source, the *bulk* phase can be characterized by a monotonically increasing battery voltage (phase 1 of figure 3.1a) which depends on the charging current (phase 1 of figure 3.1b).

**Absorption.** The battery reaches the *absorption* phase as soon as the voltage has reached the *absorption voltage* level. In this phase, the charging current is declining exponentially (phase 2 in 3.1b). The value of the *absorption voltage* depends on the temperature and the battery type.

**Float.** The *float* phase is reached when the battery is fully charged. The battery voltage drops from the *absorption voltage* to the *float voltage*. During this phase the battery charging current remains on a constant level which is usually about 2-10% of the capacity depending on the battery type. The *float* phase is illustrated in phase 3 of figure 3.1. If the battery is used too heavily while being charged, the float phase is not reached.

## 3.2. Finite-State Machine

For observing the behavior of a battery over time, it is appropriate to define a Finite-State Machine (FSM) to track each of the specified phases of the charging and discharging processes. Figure 3.2 shows an overview of the proposed FSM including its states and transitions. The individual states are briefly explained in the following.

**UNKNOWN** is the starting state. While being in the *UNKNOWN* state, the system gathers information to determine in which state the battery is. It monitors the evolution of the voltage and current for detecting the conditions of the transitions to the other states. If it detects a discharging behavior, the FSM changes over to the *DISCHARGE* state. The discharging behavior is detected by a falling voltage tendency combined with a constant or decreasing current tendency.

Likewise, if the system detects a bulk behavior, it transitions into the *BULK* state. The bulk behavior is detected by an increasing voltage tendency combined with a constant or increasing current tendency.

The FSM changes over to the *ABSORPTION* state if the voltage reaches the absorption voltage level $V_{absorption}$. If the voltage remains stable at the float voltage level $V_{float}$, the system transitions to the *FLOAT* state.

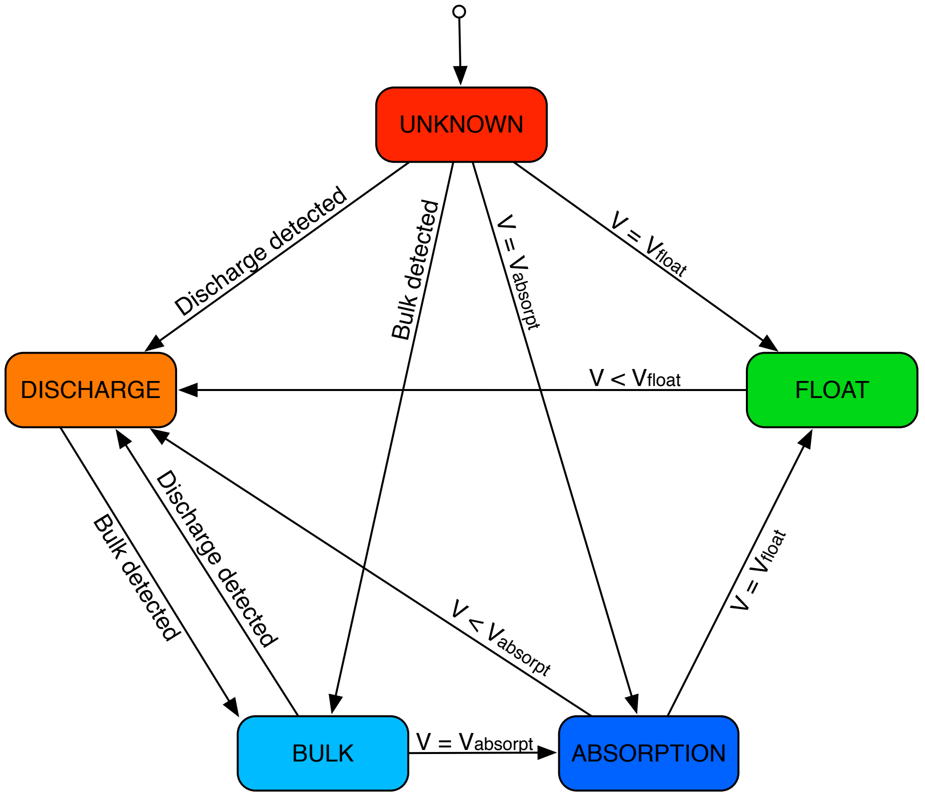


Figure 3.2.: Overview of the finite-state machine with its states and abstract transitions

**DISCHARGE** represents the discharging process of the battery. The battery remains in this state as long as the battery is discharging. In the *DISCHARGE* state the system monitors the voltage and the drain current for detecting a *bulk* behavior that is characterized by a monotonically increasing voltage. As soon as a bulk behavior is detected, the system transitions into the *BULK* state.

**BULK** is the first state of the three-step charging process introduced in section 3.1. As soon as the voltage reaches the absorption threshold voltage $V_{absorption}$, the FSM changes over to the *ABSORPTION* state. Otherwise, when the battery is discharging which is indicated by a falling voltage tendency or even a sudden voltage drop, the system changes over to the *DISCHARGE* state.

**ABSORPTION** can only be reached from the *BULK* or *UNKNOWN* state. This is because a charging process always follows the path from *BULK* to *ABSORPTION*. The typical characteristic of the *absorption* phase is that the battery voltage does not increase anymore as in the *bulk* phase. When the battery voltage drops to the float voltage level $V_{float}$ the system changes over to the *FLOAT* state. When the battery voltage falls below absorption threshold voltage $V_{absorption}$ and monotonically decreases, the battery is not charged anymore. In this case the FSM transitions to the *DISCHARGE* state.

**FLOAT** is the final state of the charging process. In this state, the battery is fully charged and is kept at $100\%$ with a small charging current. If the voltage drops below the float voltage level $V_{float}$, the battery is discharging and thus the system transitions into the *DISCHARGE* state.

## 3.3. State of Charge Approximation

To capture the behavior of the battery several traces of the charging and discharging processes are recored. A single trace consists of voltage and drain current measurements of a charging or discharging process. In the discharging case several traces at different load levels are recorded. Likewise, a few traces with different charging currents are recorded for the charging process. Afterwards, these traces are used to generate an approximative model of the state of charge depending on the load and voltage. At first, the voltage curve of each load level is approximated. The resulting coefficients of each voltage curve are a function of the drain current which is approximated in a second step. This section describes the idea of this trace-based state of charge approximation in detail.

### 3.3.1. Battery Measurements

Figure 3.3 shows an off-the-shelf setup with a photodiode representing the charging source, a charge controller, a battery and a system platform. As figure 3.3 shows the voltage sensors of the system platform can only measure the system input voltage $V_{sys}$. However, for modeling the behavior of the battery the battery voltage $V_{bat}$ is relevant. Therefore, the voltage drop over the cable and the charge controller $V_{drop}$ must be considered. Equation 3.1 shows the relation of $V_{bat}$ to $V_{sys}$. The voltage drop $V_{drop}$ is dependent on the resistance of the charge controller and the cable resistance as equation 3.2 shows. Furthermore, the resistance of the cable $R_{cable}$ depends on the specific resistance $\rho$, the length $l_{cable}$ and the cross area $A$ of the cable (see equation 3.3).

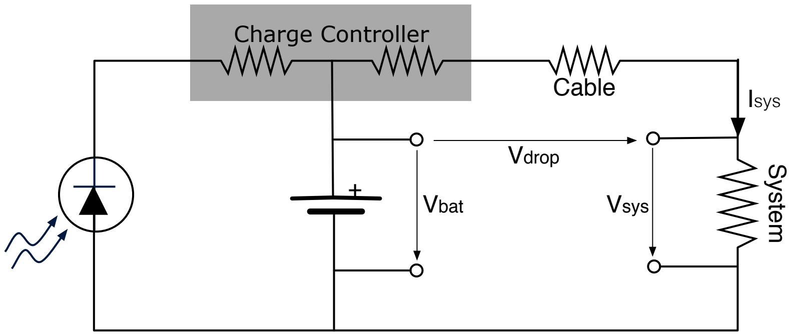


Figure 3.3.: Simplified model of the voltages and currents of the system platform

$$V_{bat} = V_{sys} + V_{drop} \tag{3.1}$$

$$V_{drop} = (R_{charge\,controller} + R_{cable}) \cdot I_{sys} \tag{3.2}$$

$$R_{cable} = \rho \cdot \frac{l_{cable}}{A} \tag{3.3}$$

### 3.3.2. Trace Generation

For approximating the discharging and charging behavior of a battery, discharge traces at different discharging currents and charge traces with different charging currents are required. A trace consists of fixed interval measurements of the system input voltage or of battery voltage and system drain current. A qualitatively good trace is achieved by measuring the battery voltage directly on the battery. This excludes a potential error of correcting the voltage drop over the cable.

In the discharging case, the discharging currents $I_{sys}$ have to be chosen in such a way, that the traces are covering the entire operational range with respect to their load. Accordingly, the charge traces have to be generated with charging currents covering the entire range of possible charging currents experienced in operation. Furthermore, for modeling the charging behavior the exact charging current have to be recorded besides the system's drain current and the system's input voltage.

### 3.3.3. Discharge Approximation

The load on the battery imposed by the drain current $I_{sys}$ is relative to the nominal capacity of the battery $C_{bat}$. For example, a high current on a high-capacity battery imposes a similar load as a lower current on a low-capacity battery. Therefore, the behavior of both batteries are similar. For that reason, the *relative load* $RL$ is introduced. It normalizes the load of the discharging current with respect to the battery's nominal capacity. The *relative load* is defined by equation 3.4.

$$RL_{sys} = \frac{I_{sys}}{C_{bat}} \tag{3.4}$$

The discharge approximation maps the measured discharge trace for each *relative load* to an according depth of discharge[1] curve as shown in figure 3.5. The depth of discharge curve of each trace can be approximated as a function of battery voltage $V_{bat}$ and the *relative load* $RL$. In the following, this mapping process and the depth of approximation is explained in detail.

The discharge traces in figure 3.4 show the voltage curves over time for two different discharging currents denoted as high load and low load. Clearly, the battery drains faster at higher load. As it is to be expected, when the load is higher, the battery voltage is lower. If the voltage is equal to the cut-off voltage of the charge controller, the load is disconnected by the charge controller.

The start of each trace is represented by point A in figure 3.4. The battery is considered fully charged and therefore, the depth of discharge is equal to $0\%$ *DoD* at that point. Similarly, the time when the battery voltage equals the cut-off voltage of the charge controller is defined as $t_{cutoff}$ (points B and C in figure 3.5). For these points the depth of discharge is defined $100\%$ *DoD*, since these correspond to the voltage when the charge controller disconnects the load. In between, the depth of discharge is assigned linearly in time. This process can be considered as changing the time domain of the trace to a depth of dis-

[1] Depth of Discharge (*DoD*) is the complement of the state of charge (*SoC*), i.e. $SoC = 100\% - DoD$

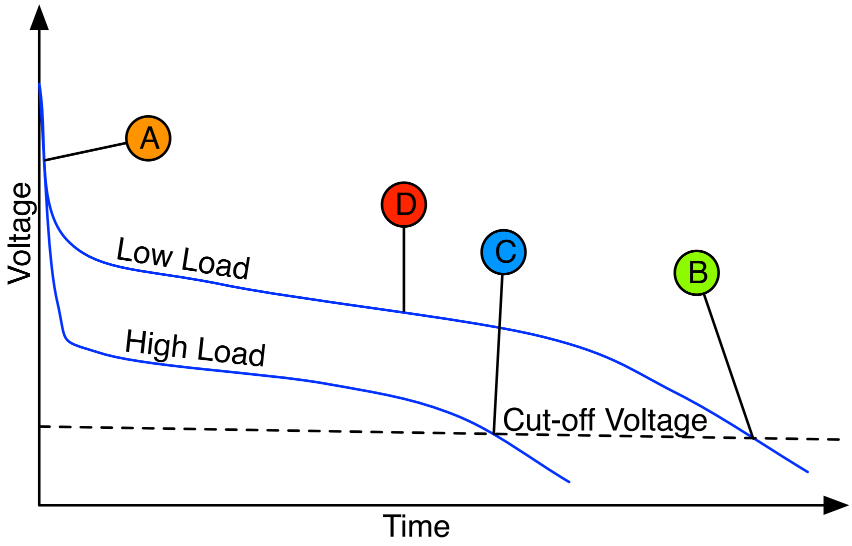


Figure 3.4.: Qualitative illustration of discharge traces at two different load levels with the charge controller specific cut-off voltage indicated by points C and B

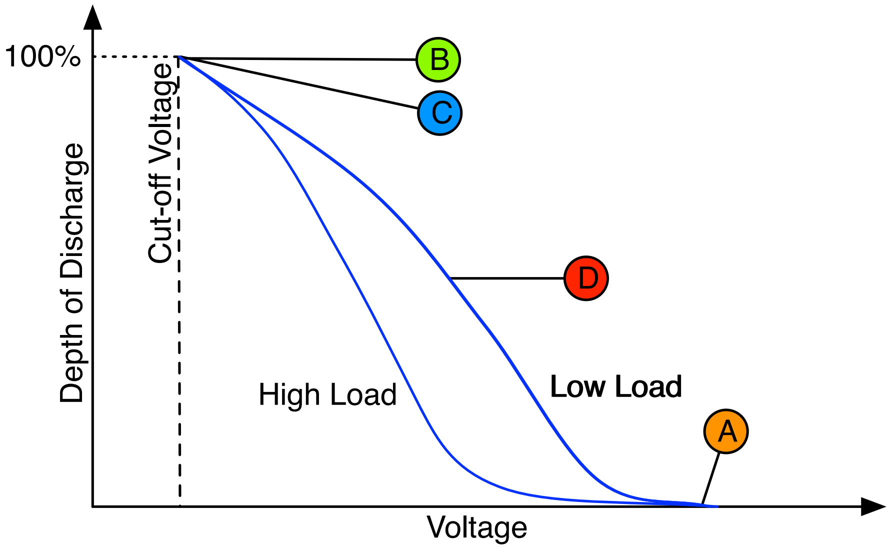


Figure 3.5.: Depth of discharge curves achieved by domain transformation and inversion of the discharge traces shown in figure 3.4

charge domain which is defined for $[0,1]$. In order to change the time domain, the depth of discharge domain transformation is defined as in equation 3.5.

$$DoD(t) = \frac{t}{t_{cutoff}} \tag{3.5}$$

After applying this domain transformation, each trace represents a voltage curve as a function of depth of discharge. However, for the discharge approximation, the depth of discharge must be a function of voltage. Therefore, the traces are inverted and the resulting traces are shown in figure 3.5.

Subsequently, the depth of discharge approximation define a well-matching function for the depth of discharge curves shown in 3.5. To this end, the depth of discharge approximation $\widehat{DoD}$ can be defined as a polynomial approximation of order $n$ for each depth of discharge trace as shown in equation 3.6. The choice of the approximation order $n$ is a trade-off between computation complexity and the fitness of the traces.

$$\widehat{DoD} = a_n \cdot V_{bat}^n + a_{n-1} \cdot V_{bat}^{n-1} + \cdots + a_1 \cdot V_{bat} + a_0 \tag{3.6}$$

Since the coefficients $a_i$ for all $i \in \{0,\dots,n\}$ are different for each trace, they can be viewed as dependent of the relative load of each trace. This implication leads to equation 3.7.

$$\widehat{DoD} = a_n(RL_{sys}) \cdot V_{bat}^n + a_{n-1}(RL_{sys}) \cdot V_{bat}^{n-1} + \cdots + a_1(RL_{sys}) \cdot V_{bat} + a_0(RL_{sys}) \tag{3.7}$$

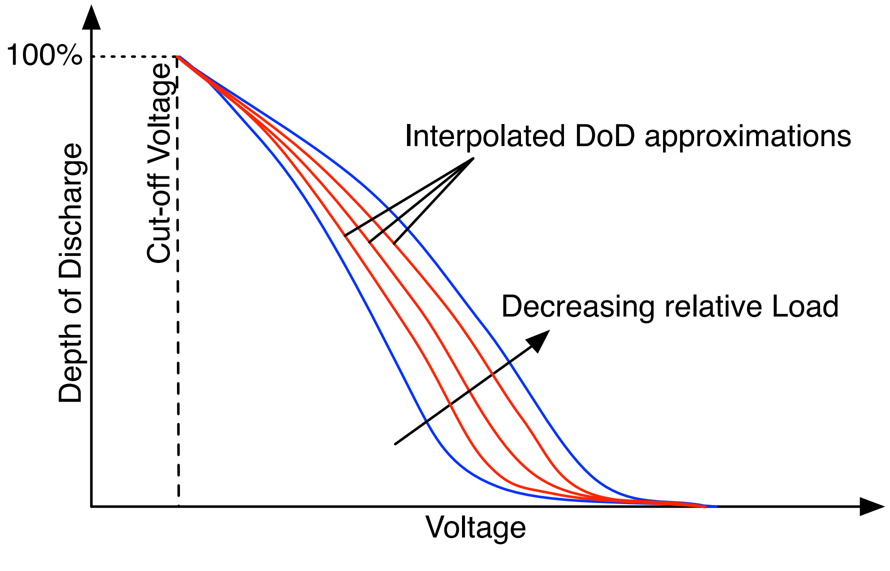


Figure 3.6.: Interpolated approximations by relative load

Furthermore, the resulting coefficients $a_i$ of all traces can be viewed as discrete data points of the *relative load* $RL_{sys}$. These data points can be expressed by a polynomial approximation of order $m$ for interpolating the coefficients for different *relative loads* other than the measured traces. This polynomial approximation of the coefficients $a_i$ is defined in equation 3.8. A qualitative illustration of the interpolated polynomial *DoD* approximation is shown in figure 3.6. The blue curves represent the approximations of the traces, while the red curves are approximations with interpolated coefficients for relative loads other than the measured traces. The approximation order $m$ must be chosen in such a way that the resulting interpolation fits all coefficients the best.

$$a_i(RL_{sys}) = b_{i,m} \cdot RL_{sys}^{m} + b_{i,m-1} \cdot RL_{sys}^{m-1} + \cdots + b_{m,1} \cdot RL_{sys} + b_{m,0} \quad \forall i \in \{0,\ldots,n\} \tag{3.8}$$

### 3.3.4. Charge Approximation

In case of the battery charging process, the state of charge approximation is also done by measuring the system voltage and the system drain current as in the discharge case. A further complication is that the charging current $I_{charge}$ cannot be measured directly, because the system can only measure the system drain current $I_{sys}$ as figure 3.7 illustrates. The charging source is represented by a photodiode. The charging current $I_{charge}$ is split up by the charge controller into $I_{bat}$ and $I_{sys}$. Depending on the amount of the charging current, the resulting net current flowing into the battery $I_{bat}$ may be positive or negative.

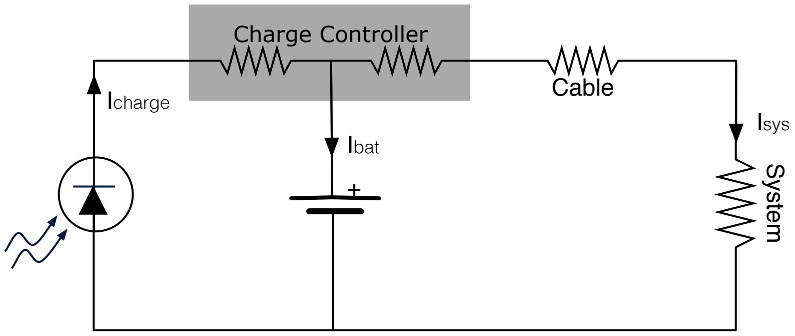


Figure 3.7.: Overview of the currents. The photodiode represents a charging source.

$$I_{sys} > I_{charge} \Longleftrightarrow I_{bat} < 0 \tag{3.9}$$

$$I_{sys} < I_{charge} \Longleftrightarrow I_{bat} > 0 \tag{3.10}$$

If $I_{bat}$ is negative, the battery is discharging because the system current $I_{sys}$ is higher than the charging current $I_{charge}$ as in equation 3.9. In this case, the model will calculate the state of charge as in a normal discharging case. However, the model does not know that the battery is charged and the current flowing out of the battery is effectively lower than the system drain current. Therefore, the voltage of the battery will remain more stable than in the discharging case where all current is supplied by the battery alone. For this reason, the state of charge approximation will remain more stable as well. In case of a positive $I_{bat}$ as shown in equation 3.10, there is a net current flowing into the battery and is thus raising its charge.

The *relative charge RC* can be defined similar to the *relative load* by equation 3.11. A charging current's contribution to the state of charge is relative to the nominal capacity of a battery. If a battery is charged for a certain amount of time with a charging current $I_{charge}$, the increase of the state of charge of the battery is relative to the nominal capacity $C_{bat}$.

$$RC_{bat} = \frac{(I_{charge} - I_{sys})}{C_{bat}} \tag{3.11}$$

The availability of state of charge information during the charging process is not mandatory because the new state of charge can be determined precisely as soon as the battery is discharging again. If the implementation of the battery model is targeting a low-resource architecture, the charge approximation can be neglected and the last state of charge of the discharging state can be assumed. This represents a worst-case state of charge during the entire charging process. However, the approximation of the charging process's state of charge is a nice improvement and is quite useful for energy dependent task scheduling.

### Bulk Approximation

In the ideal charging case, the bulk charging current flowing into the battery $I_{bat_{bulk}}$ is constant and the voltage is increasing monotonically. The *relative charge* $RC_{bat_{bulk}}$ can be used to determine the current state of charge in bulk in a conservative way[2]. There is for example a bulk phase for 3 hours and a *relative charge* of $\frac{3A}{30Ah} = 0.10h^{-1} = 10\%h^{-1}$. This value can be interpreted as an increase in state of charge of 10% per hour, resulting in a 30% increase during the entire bulk period. Nevertheless, this increase in state of charge is relative to the state of charge of the previous state, i.e. *DISCHARGING*. Hence, the calculation of the state of charge during bulk must rely on the last state of charge before entering the bulk state. Furthermore, the battery model must remember the time of entering the state bulk for calculating the time in bulk $t_{bulk}$.

$$SoC = SoC_{prevState} + RC_{bat_{bulk}} \cdot t_{bulk} \tag{3.12}$$

Since it is assumed that the bulk charging current cannot be measured directly by the system, the exact *relative charge* $RC_{bat_{bulk}}$ is not known. Therefore, it has to be approximated indirectly by the slope of the voltage: $\frac{\delta V}{\delta t}$. This can be done because of the fact that the voltage is increasing slower if the *relative charge* is lower. Figure 3.8 shows two qualitative voltage traces of the bulk charging process of the same battery for a low battery net charging current $I_{bat,1}$ and a higher net charging current $I_{bat,2}$. The bulk charging process with the higher current, and thus with the higher *relative charge* $RC_{bat,2}$, reaches the absorption voltage threshold faster. The voltage slope can be easily approximated by a linear regression. Since the *relative charge* of the traces are known, the corresponding voltage slopes of each trace are used to define the coefficients $c_0$ and $c_1$ of the first order approximation of the *relative charge* $\widehat{RC}_{bat_{bulk}}$ shown in equation 3.13.

$$\widehat{RC}_{bat_{bulk}} = c_1 \cdot \frac{\delta V}{\delta t} + c_0 \tag{3.13}$$

[2] This is a conservative interpretation because the current is relative to the nominal capacity of the battery and not to the effective capacity. The effective capacity is smaller than the nominal capacity due to the charge controller which prevents deep discharge cycles.

Consequently, for determing the state of charge, equation 3.12 has to be slightly modified to use the approximated *relative charge* $\widehat{RC}_{bat_{bulk}}$ instead of $RC_{bat_{bulk}}$ as equation 3.14 shows.

$$SoC = SoC_{prevState} + \widehat{RC}_{bat_{bulk}} \cdot t_{bulk} \tag{3.14}$$

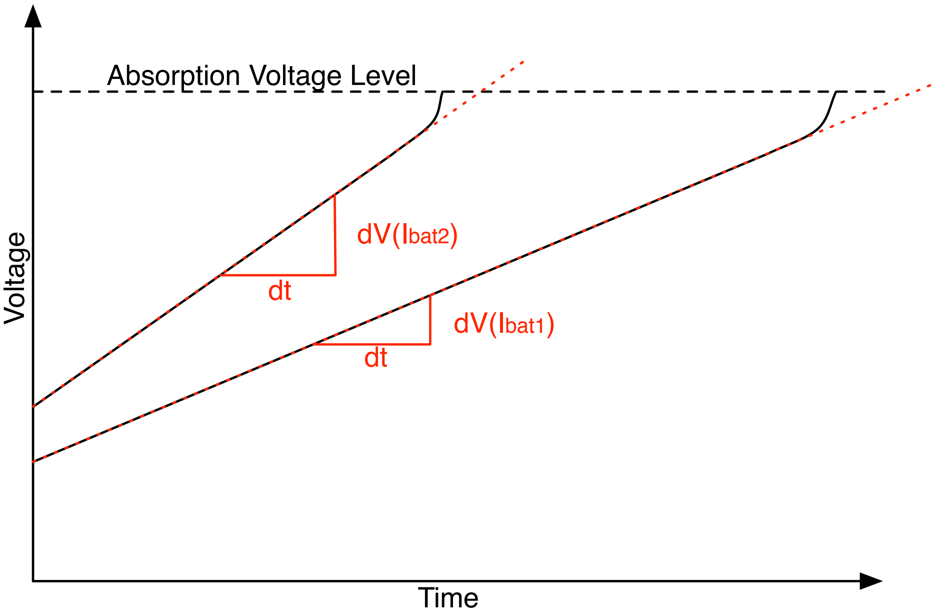


Figure 3.8.: Linear approximation of the voltage slope of two different net charging currents

### Absorption Approximation

As explained in section 3.1, the absorption phase is characterized by a constant battery voltage and an exponentially decreasing charging current. The absorption charging current starts to decrease exponentially from the bulk charging current level which is $I_{bat_{bulk}}$. For determining the state of charge in absorption the *relative charge* is used as in the bulk case. However, the charging current cannot be approximated by the battery voltage due to the constant battery voltage in absorption. Hence, there is no approximation based on voltage or current measurements. The only possibility to approximate the *relative charge* is to consider the fact that in the ideal case the current decays exponentially from the bulk charging current. This assumes that the current source is not influenced externally and changes its behavior. This is an ideal case and especially for energy harvester such as photovoltaic cells or micro wind turbines this may not be true.

In consequence, the *relative charge* in the absorption phase $RC_{bat_{absorption}}$ decays exponentially starting from $RC_{bat_{bulk}}$ as equation 3.15 shows. The parameter $\lambda$ is the decay constant and is dependent on the *relative charge* of the bulk phase because $RC_{bat_{bulk}}$ defines the initial level of the charging current in absorption. The higher this level the quicker the absorption charging current decreases due to electrochemical effects.

$$RC_{bat_{absorption}}(t) = RC_{bat_{bulk}} \cdot e^{(\lambda(RC_{bat_{bulk}}) \cdot t_{absorption})} \tag{3.15}$$

For approximating $RC_{bat_{absorption}}$ only the decay constant $\lambda$ and $RC_{bat_{bulk}}$ have to be known. Since $RC_{bat_{bulk}}$ is already approximated by the bulk approximation, it can be replaced by $\widehat{RC}_{bat_{bulk}}$. The decay constant $\lambda$ of each trace is found by an exponential approximation of the corresponding net charging current curve. Then, as equation 3.16 shows $\lambda$ can be approximated by a linear interpolation of the relative charge $\widehat{RC}_{bat_{bulk}}$ of bulk. The coefficients $d_1$ and $d_0$ are found by a linear regression of $\lambda$ and $RC_{bat_{bulk}}$ for each trace.

$$\widehat{\lambda}(\widehat{RC}_{bat_{bulk}}) = d_1 \cdot \widehat{RC}_{bat_{bulk}} + d_0 \tag{3.16}$$

$$\widehat{RC}_{bat_{absorption}}(t) = \widehat{RC}_{bat_{bulk}} \cdot e^{(\widehat{\lambda}(\widehat{RC}_{bat_{bulk}}) \cdot t_{absorption})} \tag{3.17}$$

For determining the state of charge, a similar method as in the bulk approximation is applied. However, the *relative charge* is decreasing over time and therefore, the state of charge must be updated differentially for each discrete approximation step $\Delta t$ as shown in equation 3.18.

$$SoC(t) = SoC(t + \Delta t) + \widehat{RC}_{bat_{absorption}}(t) \cdot \Delta t \tag{3.18}$$

## 3.3.5. Aging and Temperature Effect

### Battery Aging

Each discharging and charging process of a battery implies some irreversible chemical reactions which are gradually decreasing the actual capacity of the battery over time. Since available capacity and voltage are directly related, the lower capacity will lead to a lower battery voltage. Furthermore, the lower voltage combined with the slightly higher drain current for delivering the same power to the load yields a lower state of charge. Therefore, the proposed battery model considers the aging effect indirectly via the effect of the aging on voltage and drain current.

### Temperature Effect

As Rao et al. [2003] point out, temperature effect has a strong impact on the battery's discharging and charging behavior. Low temperatures reduce the electrochemical activity and increase the internal resistance which results in a reduced capacity. The proposed state of charge model takes the temperature effect indirectly into account. Since the internal resistance and the reduced chemical activity leads to a lower battery voltage and a higher drain current, the state of charge approximation yields a lower state of charge than at room temperature. It is important to state that the temperature does not affect the 0% state of charge because the charge controller disconnects the load on a temperature independent voltage level.

The temperature affects the absorption and float voltage levels which have to be considered by the FSM to correctly determine the battery state as explained in section 3.2.

# CHAPTER 4

# Implementation

## 4.1. Software Implementation

### 4.1.1. Framework Overview

As a proof of concept, the proposed battery model is implemented for an existing software framework called *Backlog* which runs on top of a linux operating system optimized for embedded systems [Buchli et al., 2011]. As figure 4.1 shows, *Backlog* can be functionally separated into two scopes: core functionality and plug-in functionality. The core of *Backlog* provides basic maintenance functionality, hardware abstraction, a connection handler for the Global Sensor Network middleware (*GSN*) [Aberer et al., 2006] and a plugin-API. This plugin functionality allows to easily register plugins for periodic activities invoked by the *Backlog* core. The plug-ins perform for example measurements and prepare messages to other sensor nodes or the *GSN* backend.

### 4.1.2. Power Monitor Module

The Power Monitor Module implements the proposed battery model as a proof of concept implementation specific to the needs of the *Permasense / X-Sense Project* [PermaSense Consortium, 2012]. The Power Monitor Module is implemented as a separate thread of the *Backlog* Core and provides an interface to core libraries and plugins. The functionality of the Power Monitor Module is divided into two classes: The *Power Monitor* and the *Battery Class*. The *Power Monitor* is responsible for all measurement tasks and the polling interface. The *Battery Class* implements the battery model discussed in chapter 3 with its approximation and the corresponding finite-state machine.

Besides the state of charge approximation, the *Battery Class* estimates the remaining time

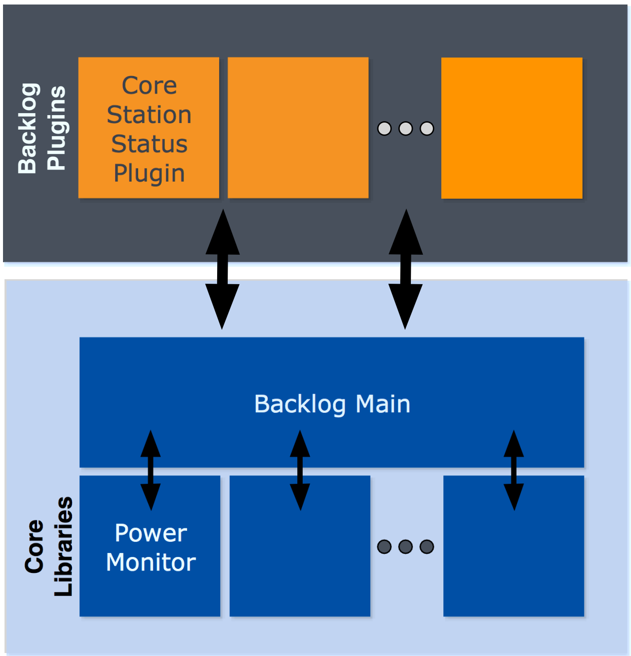


Figure 4.1.: Backlog Software Framework

based on the state of charge approximation and with respect to a short-term average load. This average load is determined by a weighted moving average with a configurable history size. Moreover, it provides an interface for questioning the remaining time of a specific load which is useful for scheduling purposes.

The absorption and float voltage threshold detection considers the current temperature and its effect on the float voltage level and absorption voltage level. Thus, these voltage levels are defined with respect to temperature. Moreover, the state is not changed unless the voltage drops below a configurable offset. This prevents unnecessary state changes in unstable charging situations. In addition, a weighted moving average is applied on the voltage measurements to smooth the voltage curve. This further reduces the noise on the voltage measurements and guard against unnecessary state changes at the cost of a reduced reaction time of real state changes.

### 4.1.3. Core Station Status Plugin

The core station status plugin is extended by the information provided by the Power Monitor Module. This plugin is responsible for sending status information such as voltages, currents, temperature and errors to the GSN backend. By extending the existing information with the battery state of charge, battery voltage and remaining time, the behavior and state of the battery can also be viewed from the backend. This is particularly useful for example for determining unhealthy batteries and schedule them for replacement. The core station status plugin uses the polling interface of the Power Monitor Module.

## 4.2. Host Platform

For evaluating the proposed battery model the proof of concept implementation for the PermaSense / X-Sense core station is used. The hardware of this core station is explained in detail by Buchli et al. [2011].

The battery used to power this system platform is the lead-acid battery *Lifeline-GPL U1M* with a nominal capacity of $34Ah$. Further technical details of the battery may be found in the technical manual of the manufacturer Lifeline Batteries Inc [2012].

The charge controller SunSaver-6 (SS-6L-12V) is an off-the-shelf component produced by Morningstar. This charge controller enforces a lower voltage bound of 11.5V. Morningstar Corporation [2012] provides further technical details.

A solar cell is used as charging source in operation. However, the kind of energy harvester is irrelevant as long as it provides enough energy for charging the battery.

# CHAPTER 5

# Evaluation

## 5.1. Model Coefficients

In the following, the approximation and interpolation coefficients of the *Lifeline-GPL U1M* battery are discussed. These coefficients correspond to the notation used in chapter 3.

### Discharge Approximation

Figure 5.1 shows three traces of the discharge process of the *Lifeline-GPL U1M* battery. In this case, the depth of discharge is approximated by a quadratic fit ($n = 2$) of the battery voltage. Table 5.1 shows the coefficients of the trace approximation shown by equation 3.7. The value of $R^2$ of this approximation is an indicator for the goodness of the fit. The residuals in figure 5.1 yields that the quadratic discharge approximations are almost in the entire voltage band within an accuracy of $\pm 2\%$. The voltages are relative to the cut-off voltage of 11500mV. As figure 5.1 shows, only at the start of the traces is the accuracy inferior than $\pm 2\%$. This can be explained by the high dynamic of the transition from charging to discharging. For capturing this dynamic more precisely, at least a cubic approximation must be applied. However, the quadratic approximation fits the need of this model well enough, because an error at a high state of charge can be accepted.

$$\widehat{DoD} = a_n(RL_{sys}) \cdot V_{bat}^n + a_{n-1}(RL_{sys}) \cdot V_{bat}^{n-1} + \cdots + a_1(RL_{sys}) \cdot V_{bat} + a_0(RL_{sys}) \quad (3.7)$$

Since the model relies on three traces at different loads the interpolation can only be done

| Trace | Relative Load $RL$ | Coefficient | Value | $R^2$ |
|---|---|---|---|---|
| 1 | 0.004710 | $a_0$ | 101.6 | 0.999537 |
| | | $a_1$ | -0.05356 | |
| | | $a_2$ | -8.836e-06 | |
| 2 | 0.011942 | $a_0$ | 100.7 | 0.999657 |
| | | $a_1$ | -0.04573 | |
| | | $a_2$ | -1.537e-05 | |
| 3 | 0.016381 | $a_0$ | 101.2 | 0.999977 |
| | | $a_1$ | -0.05152 | |
| | | $a_2$ | -1.516e-05 | |

Table 5.1.: The coefficients of equation 3.7 for each of the three discharge traces

quadratic ($m = 2$). The coefficients for the load interpolation from equation 3.8 are shown in table 5.2. For increasing the accuracy of the model, the number of traces has to be increased.

$$a_i(RL_{sys}) = b_{i,m} \cdot RL_{sys}^{m} + b_{i,m-1} \cdot RL_{sys}^{m-1} + \cdots + b_{m,1} \cdot RL_{sys} + b_{m,0} \quad \forall i \in \{0, \ldots, n\} \tag{3.8}$$

| Coefficient | Value | Coefficient | Value | Coefficient | Value |
|---|---|---|---|---|---|
| $b_{0,0}$ | 100 | $b_{1,0}$ | -0.07018 | $b_{2,0}$ | -8.321e-9 |
| $b_{0,1}$ | 0 | $b_{1,1}$ | 4.492 | $b_{2,1}$ | -0.002257 |
| $b_{0,2}$ | 0 | $b_{1,2}$ | -204.7 | $b_{2,2}$ | 0.08133 |

Table 5.2.: The coefficients of equation 3.8 used for the curve interpolation

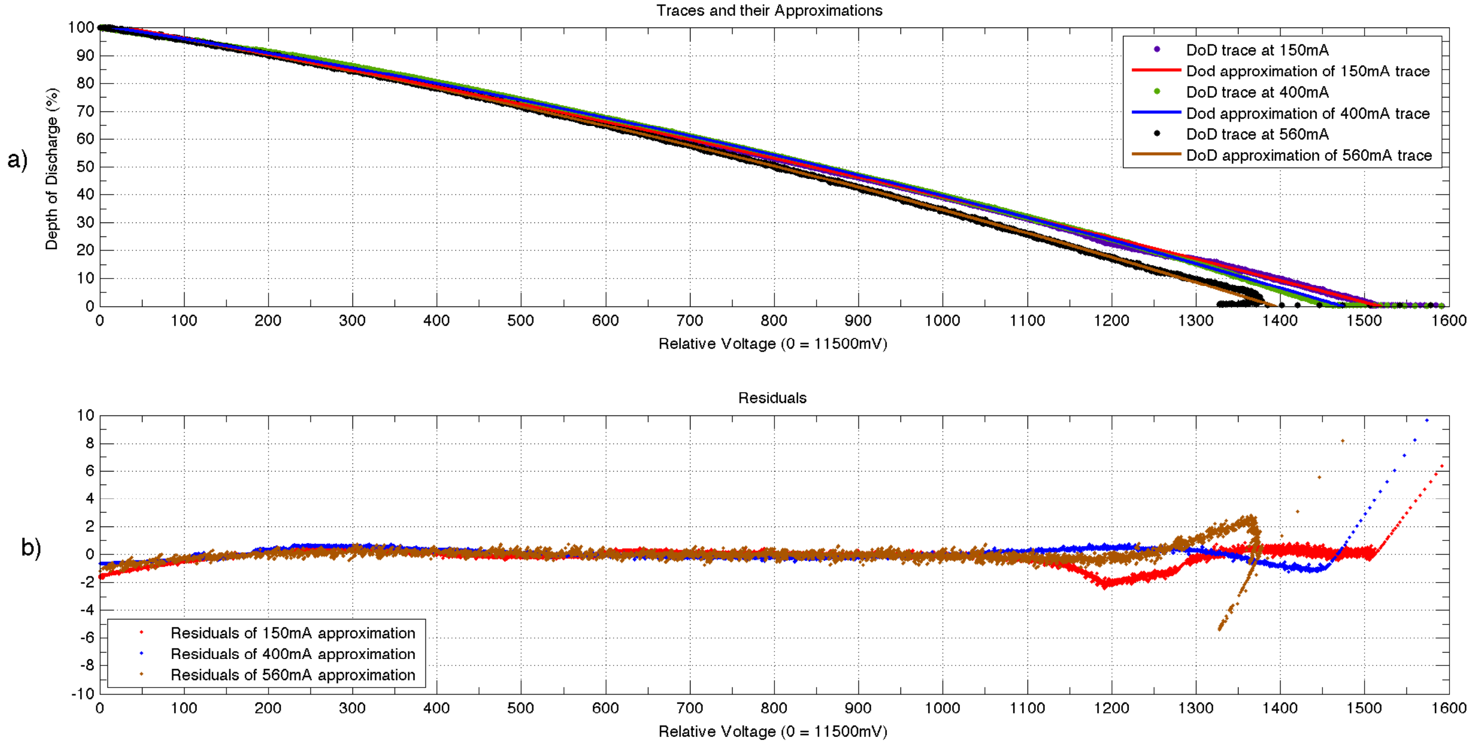


Figure 5.1.: a) Discharge traces of Lifeline-GPL U1 and their approximations. b) shows the resulting residuals of the approximations.

### Bulk Approximation

The bulk traces are shown in figure 5.2 and the details of the traces are listed in table 5.3. The violet trace is denoted as trace 1 and the green trace as trace 2. The coefficients $c_0$ and $c_1$ of the approximation of the relative charge, $\widehat{RC}_{bat_{bulk}}$, are shown in table 5.4.

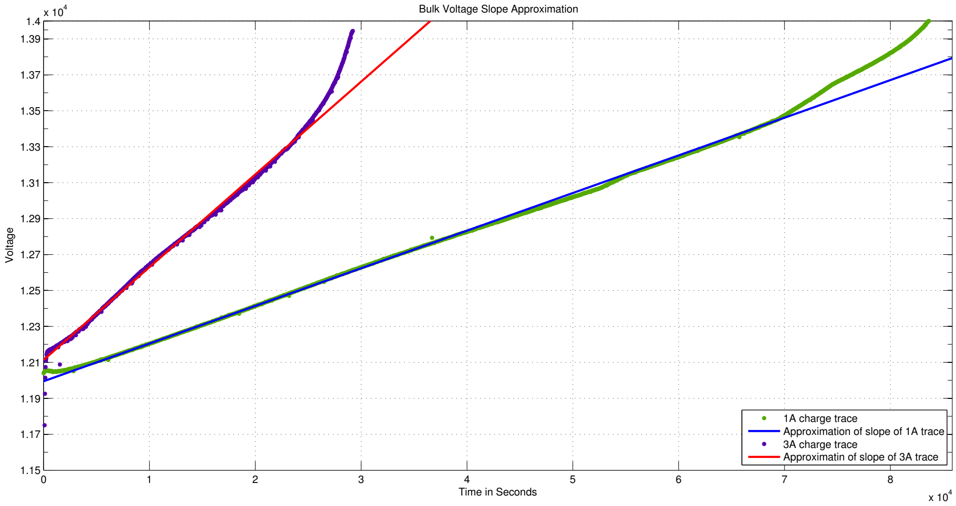


Figure 5.2.: Linear approximation of the voltage slope of two different charging currents

| Trace | Charge Current | Relative Charge $RC$ | Voltage Slope $\delta V/\delta t$ | $R^2$ |
|---|---|---|---|---|
| 1 | 2.4639 A | 0.072468 | 0.051614 mV/s | 0.999900 |
| 2 | 1.1125 A | 0.032720 | 0.020945 mV/s | 0.999925 |

Table 5.3.: The voltage slope approximation of the two charge traces.

$$\widehat{RC}_{bat_{bulk}} = c_1 \cdot \frac{\delta V}{\delta t} + c_0 \qquad (3.13)$$

| Coefficient | Value |
|---|---|
| $c_0$ | 0.005573 |
| $c_1$ | 1.296 |

Table 5.4.: The coefficients for the bulk approximation

### Absorption Approximation

The coefficients for the decay constant $\lambda$ approximation of equation 3.16 are shown in table 5.5.

| Coefficient | Value |
|---|---|
| $d_0$ | -0.003891 |
| $d_1$ | -0.1091 |

Table 5.5.: The coefficients for absorption approximation

$$\widehat{\lambda}(\widehat{RC}_{bat_{bulk}}) = d_1 \cdot \widehat{RC}_{bat_{bulk}} + d_0 \tag{3.16}$$

### Float Approximation

The state of charge of the float phase has not to be approximated since the battery if fully charged when it reaches the float phase. However, the float voltage is temperature dependent. The data sheet of the battery provides detailed information about the temperature effect on the voltage. The approximation of $V_{float}$ with respect to temperature $T$ is shown in equation 5.1

$$V_{float}(T) = \max\{13000, 0.239 \cdot T^2 - 35.94 \cdot T + 14040\} \tag{5.1}$$

## 5.2. Evaluation Setup

For evaluating the model, a discharge process is recorded with an external coulomb counter [Maxim Integrated Products, 1998] and with the proof of concept implementation of the battery model. The coulomb counter is placed between the charge controller and the battery as figure 5.3 shows. The coulomb counter measures the charge flowing through the counter. These measurements are then transformed into the state of charge. At the beginning of the discharge process the state of charge is defined as 100% and when the charge controller disconnects the load the state of charge is defined as 0%. This allows the comparison of the actual state of charge inferred by the coulomb counter and the approximated state of charge of the battery model.

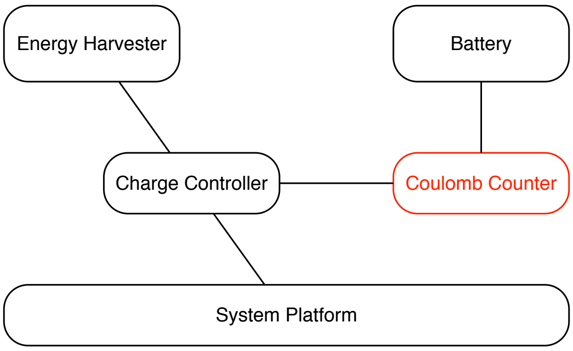


Figure 5.3.: Block diagram of experimental setup

## 5.3. Experimental Results

Figures 5.4 a) and b) show the results of the experiment. The dashed curve of figure 5.4a) represents the actual depth of discharge inferred by a coulomb counter. The red curve is the depth of discharge of the model computed by the proof of concept implementation. Since the coulomb counter is placed between the battery and the charge controller, it introduces an additional resistance which was not considered in the cable resistance. Therefore, the battery voltage is not computed accurately because the voltage drop over the cable, the charge controller and the coulomb counter was not determined correctly. This introduced an additional error of around $2-3\%$ as shown in table 5.6. The green curve is the state of charge approximation inferred by corrected voltage measurements.

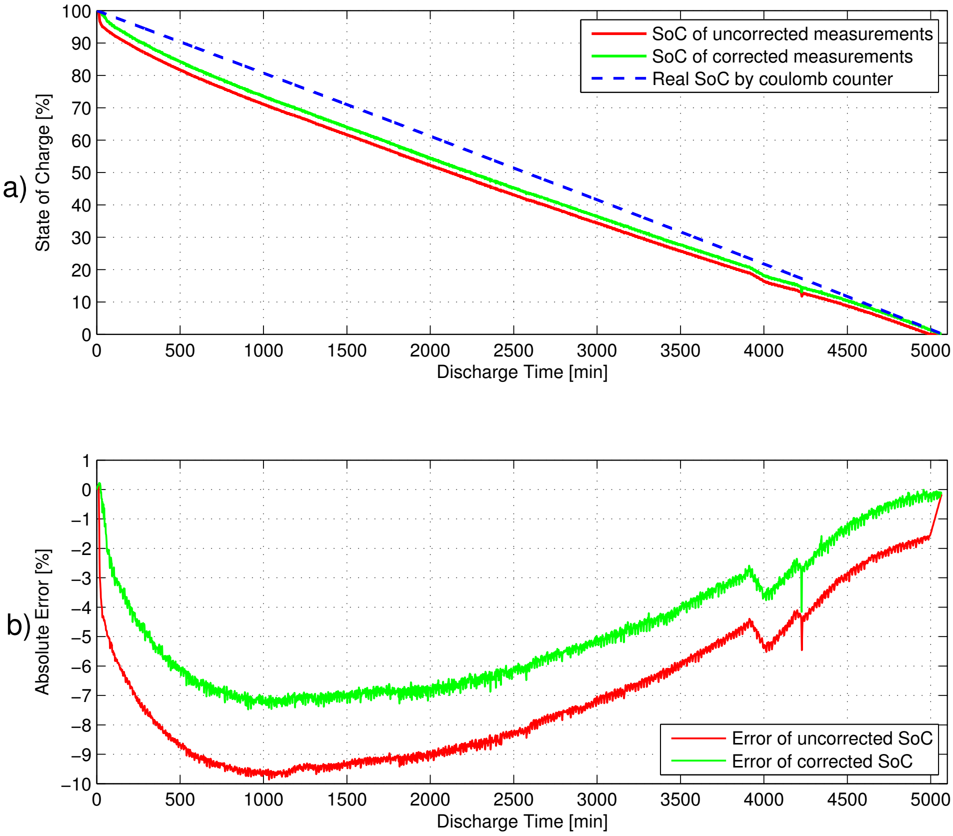


Figure 5.4.: a) Approximative and real state of charge over time, and b) the error of the approximation

Figure 5.4b) shows the actual error of the depth of discharge approximation to the actual depth of discharge deduced by the coulomb counter. The actual error curves of both approximations are decreasing with lower state of charge because the cut-off voltage is getting closer and state of charge curves of different loads are converging to each other. This implies that the closer the voltage gets to the cut-off voltage the smaller is effect of the load on the state of charge approximation. However, if the voltage drop is considered too small, as in the measurement of the red curve, the 0% state of charge is assumed before the load is actually disconnected. Therefore, the error is not reaching zero at the end. On the other hand, the error of the corrected state of charge is almost going to zero at the end. The approximation of the state of charge is in the entire range negative. This means that the actual state of charge is always underestimated.

| Corrected | Temperature | Relative Load $RL$ | Average Error | Max Error | Min Error |
|---|---|---|---|---|---|
| yes | 26° C | 0.0103 | 4.720% | 7.257% | 0.000% |
| no | 26° C | 0.0103 | 6.798% | 9.681% | 0.003% |

Table 5.6.: Error of the corrected and uncorrected SoC of the evaluation discharge process

### 5.3.1. Temperature Effect

Besides the evaluation of the model based on the experiment at 26° celsius, another discharge process at the same load is recorded at the temperature -6° celsius. The results of this experiment are shown in figures 5.5a) and b). The results of the discharge experiment at 26° celsius are shown as well in figure 5.5a) and b) for comparison. The coulomb counter inferred a 20 percent reduction in capacity at -6° celsius with respect to the 26° celsius case. As outlined in section 3.3.5 the temperature effect is indirectly considered through the effect of the temperature on the battery voltage. Table 5.7 shows the details of the error analysis of the low temperature compared with the room temperature scenario.

| Temperature | Relative Load $RL$ | Average Error | Max Error | Min Error |
|---|---|---|---|---|
| 26℃ | 0.0103 | 4.720% | 7.257% | 0.000% |
| -6℃ | 0.0101 | 1.223% | 3.113% | 0.003% |

Table 5.7.: Details of SoC determination at different temperatures

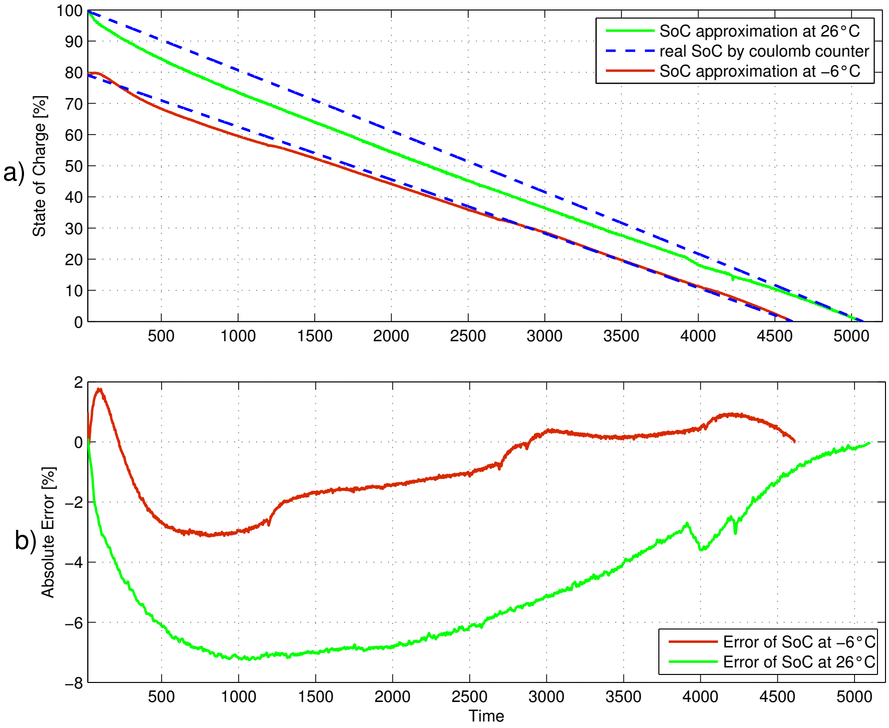


Figure 5.5.: a) Approximative and real state of charge over time at the same load but at different temperatures, and b) the error of the SoC approximations

# CHAPTER 6

# Conclusion

The proposed state of charge battery model relies fully on closed-loop voltage and drain current measurements and does not require any special purpose hardware for energy monitoring. Therefore, if the target platform provides voltage and current sensors, no changes on existing platforms are required.

The recording of battery charge and discharge traces and the approximation of the state of charge with drain current and voltage implies some initial effort. However, if the traces are recorded and the state of charge model is fully defined, this model implies a low operational overhead with a reasonable accuracy. Furthermore, the accuracy may be increased by refining the approximation model with additional traces. The accuracy of the model heavily depends on the quality of the traces and the number of the traces at different loads.

The main drawback of the proposed model is that the approximation model must be defined for each battery type. Furthermore, the approximation accuracy of the battery state of charge heavily depends on the voltage drop of the cable from the battery to the measurement sensor that can be also temperature dependent. If the resistance of the cable and of the charge controller cannot be exactly determined an additional non-negligible error is introduced.

To conclude, the proposed battery model implies some initial effort for generating the traces of a battery type and determine the coefficients of the model. Nevertheless, this lightweight battery state of charge modeling approach provides a reasonable accuracy and it imposes only a minimal operational overhead with no hardware modifications required.

# APPENDIX A

# Appendix

## A.1. Matlab Model Generation Script

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Discharge Trace Approximation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    clear;
    load ../../data/data_discharge_150mA_2.mat;
    load ../../data/data_discharge_400mA.mat;
    load ../../data/data_discharge_560mA.mat;


    % Filter the data and normalize it..

    [I_AVG_150mA,V_AVG_150mA]=Normalize_Data(I_V12DC_IN_150mA, V12DC_IN_150mA);
    [I_AVG_400mA,V_AVG_400mA]=Normalize_Data(I_V12DC_IN_400mA+I_V12DC_EXT_400mA, V12DC_IN_400mA)↩
        ;
    [I_AVG_560mA,V_AVG_560mA]=Normalize_Data(I_V12DC_IN_560mA+I_V12DC_EXT_560mA, V12DC_IN_560mA)↩
        ;

    % Define the measurement cable resistance and the battery capacity
    R_cable = 0.26;
    Capacity = 34;

    % Calculate the battery Voltage, add the Cable drop..
    V_BAT_150mA = V_AVG_150mA + R_cable*I_AVG_150mA/1000;
    V_BAT_400mA = V_AVG_400mA + R_cable*I_AVG_400mA/1000;
    V_BAT_560mA = V_AVG_560mA + R_cable*I_AVG_560mA/1000;

    % Shift the Voltage to relative base (0 = 11500 mV), the min of the traces
    % are the 11500mV threshold!
    V_SHIFTED_150mA = V_BAT_150mA - min(V_BAT_150mA);
    V_SHIFTED_400mA = V_BAT_400mA - 11500;
    V_SHIFTED_560mA = V_BAT_560mA - min(V_BAT_560mA);
    % we have to linearly extend the 400mA trace, since the trace does not
    % cover the last 108mV... (this part is quite linear!)
    Extension = V_SHIFTED_400mA(8000:1:end);
    X = [8000:1:8000+size(Extension,1) -1]';
    P = polyfit(X,Extension,1);
    X_0 = -(P(2)/P(1));
    X = [size(V_SHIFTED_400mA)+1:1:X_0]';
    V_SHIFTED_400mA = [V_SHIFTED_400mA; P(1)*X+P(2)];

    % The current must be in terms of capacity...
    I_RATED_150mA = I_AVG_150mA/(Capacity*1e6);
    I_RATED_400mA = I_AVG_400mA/(Capacity*1e6);
    I_RATED_560mA = I_AVG_560mA/(Capacity*1e6);

    % Create the percentage values of the Depth of Discharge
    DoD_150mA = linspace(0,100,size(V_SHIFTED_150mA,1));
    DoD_400mA = linspace(0,100,size(V_SHIFTED_400mA,1));
    DoD_560mA = linspace(0,100,size(V_SHIFTED_560mA,1));

    createFittedCurves(V_SHIFTED_150mA,DoD_150mA,V_SHIFTED_400mA,DoD_400mA,V_SHIFTED_560mA,DoD_560mA);

    a=[fittedmodel_150mA.p1,fittedmodel_400mA.p1,fittedmodel_560mA.p1];
    b=[fittedmodel_150mA.p2,fittedmodel_400mA.p2,fittedmodel_560mA.p2];
    c=[fittedmodel_150mA.p3,fittedmodel_400mA.p3,fittedmodel_560mA.p3];
    I_RATED_150mA_AVG = mean(I_RATED_150mA);
    I_RATED_400mA_AVG = mean(I_RATED_400mA);
    I_RATED_560mA_AVG = mean(I_RATED_560mA);
    x = [I_RATED_150mA_AVG, I_RATED_400mA_AVG, I_RATED_560mA_AVG];
```

```matlab

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Helper Function for Discharge Trace Approximation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 function createFittedCurves(V_SHIFTED_150mA,DoD_150mA,V_SHIFTED_400mA,DoD_400mA,V_SHIFTED_560mA,DoD_560mA)
%CREATEFIT Create plot of data sets and fits
%   CREATEFIT(V_SHIFTED_150MA,DOD_150MA,V_SHIFTED_400MA,DOD_400MA,V_SHIFTED_560MA,DOD_560MA)
%   Creates a plot, similar to the plot in the main Curve Fitting Tool,
%   using the data that you provide as input.  You can
%   use this function with the same data you used with CFTOOL
%   or with different data.  You may want to edit the function to
%   customize the code and this help message.
%
%   Number of data sets:  3
%   Number of fits:  3

% Data from data set "DoD_150mA vs. V_SHIFTED_150mA":
%     X = V_SHIFTED_150mA:
%     Y = DoD_150mA:
%     Unweighted

% Data from data set "DoD_400mA vs. V_SHIFTED_400mA":
%     X = V_SHIFTED_400mA:
%     Y = DoD_400mA:
%     Unweighted

% Data from data set "DoD_560mA vs. V_SHIFTED_560mA":
%     X = V_SHIFTED_560mA:
%     Y = DoD_560mA:
%     Unweighted

% Auto-generated by MATLAB on 03-Feb-2012 16:03:50

% Set up figure to receive data sets and fits
 f_ = clf;
```

```
figure(f_);
set(f_,'Units','Pixels','Position',[1 120 1389 755]);
% Line handles and text for the legend.
legh_ = [];
legt_ = {};
% Limits of the x-axis.
xlim_ = [Inf -Inf];
% Axes for the plot.
ax_ = axes;
set(ax_,'Units','normalized','OuterPosition',[0 0 1 1]);
set(ax_,'Box','on');
grid(ax_,'on');
axes(ax_);
hold on;

% --- Plot data that was originally in data set "DoD_150mA vs. V_SHIFTED_150mA"
V_SHIFTED_150mA = V_SHIFTED_150mA(:);
DoD_150mA = DoD_150mA(:);
h_ = line(V_SHIFTED_150mA,DoD_150mA,'Parent',ax_,'Color',[0.333333 0 0.666667],...
    'LineStyle','none', 'LineWidth',1,...
    'Marker','.', 'MarkerSize',12);
xlim_(1) = min(xlim_(1),min(V_SHIFTED_150mA));
xlim_(2) = max(xlim_(2),max(V_SHIFTED_150mA));
legh_(end+1) = h_;
legt_{end+1} = 'DoD_150mA vs. V_SHIFTED_150mA';

% --- Plot data that was originally in data set "DoD_400mA vs. V_SHIFTED_400mA"
V_SHIFTED_400mA = V_SHIFTED_400mA(:);
DoD_400mA = DoD_400mA(:);
h_ = line(V_SHIFTED_400mA,DoD_400mA,'Parent',ax_,'Color',[0.333333 0.666667 0],...
    'LineStyle','none', 'LineWidth',1,...
    'Marker','.', 'MarkerSize',12);
xlim_(1) = min(xlim_(1),min(V_SHIFTED_400mA));
xlim_(2) = max(xlim_(2),max(V_SHIFTED_400mA));
legh_(end+1) = h_;
legt_{end+1} = 'DoD_400mA vs. V_SHIFTED_400mA';

% --- Plot data that was originally in data set "DoD_560mA vs. V_SHIFTED_560mA"
V_SHIFTED_560mA = V_SHIFTED_560mA(:);
DoD_560mA = DoD_560mA(:);
h_ = line(V_SHIFTED_560mA,DoD_560mA,'Parent',ax_,'Color',[0 0 0],...
    'LineStyle','none', 'LineWidth',1,...
    'Marker','.', 'MarkerSize',12);
xlim_(1) = min(xlim_(1),min(V_SHIFTED_560mA));
xlim_(2) = max(xlim_(2),max(V_SHIFTED_560mA));
legh_(end+1) = h_;
legt_{end+1} = 'DoD_560mA vs. V_SHIFTED_560mA';

% Nudge axis limits beyond data limits
if all(isfinite(xlim_))
    xlim_ = xlim_ + [-1 1] * 0.01 * diff(xlim_);
    set(ax_,'XLim',xlim_)
else
    set(ax_, 'XLim',[-26.451534999999921638, 2671.6050349998986349]);
end

% --- Create fit "150mA_fit"

% Apply exclusion rule "0_100&0_1600"
ex_ = (V_SHIFTED_150mA < 0 | V_SHIFTED_150mA >= 1600) | (DoD_150mA < 0 | DoD_150mA >= 100);
fo_ = fitoptions('method','LinearLeastSquares','Robust','LAR');
ok_ = isfinite(V_SHIFTED_150mA) & isfinite(DoD_150mA);
if ~all( ok_ )
    warning( 'GenerateMFile:IgnoringNansAndInfs',...
        'Ignoring NaNs and Infs in data.' );
end
set(fo_,'Exclude',ex_(ok_));
```

```
ft_ = fittype('poly2');

% Fit this model using new data
if sum(~ex_(ok_))<2
    % Too many points excluded.
    error( 'GenerateMFile:NotEnoughDataAfterExclusionRule',...
        'Not enough data left to fit ''%s'' after applying exclusion rule ''%s''.',...
        '150mA_fit', '0_100&0_1600' );
else
    cf_ = fit(V_SHIFTED_150mA(ok_),DoD_150mA(ok_),ft_,fo_);
end
% Alternatively uncomment the following lines to use coefficients from the
% original fit. You can use this choice to plot the original fit against new
% data.
%    cv_ = { -8.8362984306375879793e-06, -0.053563780522033831166, 101.55394381263083403};
%    cf_ = cfit(ft_,cv_{:});

% Plot this fit
h_ = plot(cf_,'fit',0.95);
set(h_(1),'Color',[1 0 0],...
    'LineStyle','-', 'LineWidth',2,...
    'Marker','none', 'MarkerSize',6);
% Turn off legend created by plot method.
legend off;
% Store line handle and fit name for legend.
legh_(end+1) = h_(1);
legt_{end+1} = '150mA_fit';

% --- Create fit "400mA_fit"

% Apply exclusion rule "0_100&0_1600"
ex_ = (V_SHIFTED_400mA < 0 | V_SHIFTED_400mA >= 1600) | (DoD_400mA < 0 | DoD_400mA >= 100);
fo_ = fitoptions('method','LinearLeastSquares','Robust','LAR');
ok_ = isfinite(V_SHIFTED_400mA) & isfinite(DoD_400mA);
if ~all( ok_ )
    warning( 'GenerateMFile:IgnoringNansAndInfs',...
        'Ignoring NaNs and Infs in data.' );
end
set(fo_,'Exclude',ex_(ok_));
ft_ = fittype('poly2');

% Fit this model using new data
if sum(~ex_(ok_))<2
    % Too many points excluded.
    error( 'GenerateMFile:NotEnoughDataAfterExclusionRule',...
        'Not enough data left to fit ''%s'' after applying exclusion rule ''%s''.',...
        '400mA_fit', '0_100&0_1600' );
else
    cf_ = fit(V_SHIFTED_400mA(ok_),DoD_400mA(ok_),ft_,fo_);
end
% Alternatively uncomment the following lines to use coefficients from the
% original fit. You can use this choice to plot the original fit against new
% data.
%    cv_ = { -1.5367029770583737620e-05, -0.045725818152965745644, 100.65650851932394971};
%    cf_ = cfit(ft_,cv_{:});

% Plot this fit
h_ = plot(cf_,'fit',0.95);
set(h_(1),'Color',[0 0 1],...
    'LineStyle','-', 'LineWidth',2,...
    'Marker','none', 'MarkerSize',6);
% Turn off legend created by plot method.
legend off;
% Store line handle and fit name for legend.
legh_(end+1) = h_(1);
legt_{end+1} = '400mA_fit';

```

```
% --- Create fit "560mA_fit"

% Apply exclusion rule "0_100&0_1600"
ex_ = (V_SHIFTED_560mA < 0 | V_SHIFTED_560mA >= 1600) | (DoD_560mA < 0 | DoD_560mA >= 100);
fo_ = fitoptions('method','LinearLeastSquares','Robust','LAR');
ok_ = isfinite(V_SHIFTED_560mA) & isfinite(DoD_560mA);
if ~all( ok_ )
    warning( 'GenerateMFile:IgnoringNansAndInfs' ,...
        'Ignoring NaNs and Infs in data.' );
end
set(fo_,'Exclude',ex_(ok_));
ft_ = fittype('poly2');

% Fit this model using new data
if sum(~ex_(ok_))<2
    % Too many points excluded.
    error( 'GenerateMFile:NotEnoughDataAfterExclusionRule' ,...
        'Not enough data left to fit ''%s'' after applying exclusion rule ''%s''.' ,...
        '560mA_fit', '0_100&0_1600' );
else
    cf_ = fit(V_SHIFTED_560mA(ok_),DoD_560mA(ok_),ft_,fo_);
end
% Alternatively uncomment the following lines to use coefficients from the
% original fit. You can use this choice to plot the original fit against new
% data.
%    cv_ = { -1.5162123533972063255e-05, -0.051518259237075779422, 101.17883146215153545};
%    cf_ = cfit(ft_,cv_{:});

% Plot this fit
h_ = plot(cf_,'fit',0.95);
set(h_(1),'Color',[0.666667 0.333333 0],...
    'LineStyle','-', 'LineWidth',2,...
    'Marker','none', 'MarkerSize',6);
% Turn off legend created by plot method.
legend off;
% Store line handle and fit name for legend.
legh_(end+1) = h_(1);
legt_{end+1} = '560mA_fit';

% --- Finished fitting and plotting data. Clean up.
hold off;
% Display legend
leginfo_ = {'Orientation', 'vertical', 'Location', 'NorthEast'};
h_ = legend(ax_,legh_,legt_,leginfo_{:});
set(h_,'Interpreter','none');
% Remove labels from x- and y-axes.
xlabel(ax_,'');
ylabel(ax_,'');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Charge Trace Approximation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% CoreStation Voltage Measurements of 1A charging
V_BULK_1A = V_BAT_1A(end:-1:1);
V_BULK_1A = V_BULK_1A(1:1:2700);
time_bulk_1A = generation_time_1A(end:-1:1);
time_bulk_1A = time_bulk_1A(1:1:2700);
time_bulk_1A = time_bulk_1A -time_bulk_1A(1);
time_bulk_1A = time_bulk_1A/1000;

% CoreStation Voltage Measurements of 3A charging
V_BULK_3A = V_BAT_3A(end:-1:1);
V_BULK_3A = V_BULK_3A(1:1:930);
```

```matlab
time_bulk_3A = generation_time_3A(end:-1:1);
time_bulk_3A = time_bulk_3A(1:1:930);
time_bulk_3A = time_bulk_3A-time_bulk_3A(1);
time_bulk_3A = time_bulk_3A/1000;

% Calculate the load of the Corestation and calucalte the netto charge
% current of the battery!
I_V12DC_BULKLOAD_1A = I_V12DC_TOT_1A(end:-1:1);
I_V12DC_BULKLOAD_1A = I_V12DC_BULKLOAD_1A(1:1:2700);
I_V12DC_BULKLOAD_3A = I_V12DC_TOT_3A(end:-1:1);
I_V12DC_BULKLOAD_3A = I_V12DC_BULKLOAD_3A(1:1:930);
I_BAT_NETTO_1A = CurrAvg10_1A*1000 - mean(I_V12DC_BULKLOAD_1A);
I_BAT_NETTO_3A = CurrAvg_3A*1000 - mean(I_V12DC_BULKLOAD_3A)/1000;
I_BAT_NETTO_3A = I_BAT_NETTO_3A(35:1:end); % there is a weird datapoint at index 34! (everything before 34 is 0)
% only consider the BULK phase...
I_BAT_NETTO_1A_BULK = mean(I_BAT_NETTO_1A(1:1:853232));
I_BAT_NETTO_1A_BULK_RATED = I_BAT_NETTO_1A_BULK/34000;
I_BAT_NETTO_3A_BULK = mean(I_BAT_NETTO_3A(1:1:290356));
I_BAT_NETTO_3A_BULK_RATED = I_BAT_NETTO_3A_BULK/34000;

% CURVE FITTING

% Regression of rated BULK CURRENT by slope..
I_BULK_CURRENT = [I_BAT_NETTO_1A_BULK,I_BAT_NETTO_3A_BULK]'/34000;
VOLTAGE_SLOPE = [BULK_1A_FIT.p1, BULK_3A_FIT.p1]';


% ABSORPTION FITTING...
I_ABSLOAD_1A = I_V12DC_TOT_1A(end:-1:1);
I_ABSLOAD_1A = I_ABSLOAD_1A(2701:1:end);
I_BAT_NETTO_1A_ABS = CurrAvg10_1A*1000;
I_BAT_NETTO_1A_ABS = I_BAT_NETTO_1A_ABS(853232:1:end);
I_BAT_NETTO_1A_ABS_RATED = I_BAT_NETTO_1A_ABS/34000;
I_BAT_NETTO_1A_ABS_RATED = downsample(I_BAT_NETTO_1A_ABS_RATED, 200);
I_BAT_NETTO_1A_ABS_RATED_LN = log(I_BAT_NETTO_1A_ABS_RATED);


I_ABSLOAD_3A = I_V12DC_TOT_3A(end:-1:1);
I_ABSLOAD_3A = I_ABSLOAD_3A(931:1:end);
I_BAT_NETTO_3A_ABS = CurrAvg_3A*1000;
I_BAT_NETTO_3A_ABS = I_BAT_NETTO_3A_ABS(290356:1:end);
I_BAT_NETTO_3A_ABS_RATED = I_BAT_NETTO_3A_ABS/34000;
I_BAT_NETTO_3A_ABS_RATED = downsample(I_BAT_NETTO_3A_ABS_RATED, 200);
I_BAT_NETTO_3A_ABS_RATED_LN = log(I_BAT_NETTO_3A_ABS_RATED);


% Select the linear part of the LN currents
I_BAT_NETTO_1A_ABS_RATED_LN_LIN = I_BAT_NETTO_1A_ABS_RATED_LN(1:1:1000);
I_BAT_NETTO_3A_ABS_RATED_LN_LIN = I_BAT_NETTO_3A_ABS_RATED_LN(1:1:500);

% create timevector in terms of seconds (sample interval is 104ms)
pa_time_1A = [0:104:104*size(I_BAT_NETTO_1A_ABS_RATED_LN_LIN,1) -1]/1000';
pa_time_3A = [0:104:104*size(I_BAT_NETTO_3A_ABS_RATED_LN_LIN,1) -1]/1000';


% CURVE FITTING
LAMBDA = [ABS_1A_LAMBDA.p1 ABS_3A_LAMBDA.p1]
CUR_BULK = [I_BAT_NETTO_1A_BULK_RATED, I_BAT_NETTO_3A_BULK_RATED]
```

## A.2. Thesis Description

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Computer Engineering and Networks Lab
(TIK)

Fall Semester 2011

SEMESTERTHESIS

for
Daniel Aschwanden

Supervisors:
Bernhard Buchli, bbuchli@ethz.ch

Professor:
Prof. Dr. Lothar Thiele, thiele@tik.ee.ethz.ch

Project Start: 27. September, 2011
Project End: 27. December, 2011

# Battery and Energy Harvesting Modeling

### 1 Introduction

For this semester thesis you will be part of an exciting real world project, the X-Sense project [7]. This joint-project between geosciences and engineering has the goal of developing a Differential-GPS (DGPS) system that can monitor alpine slope movements at sub-centimeter accuracy over extended periods.

### 2 X-Sense Background

The successfully deployed WSN infrastructure [5, 7, 1], which collects measurements of numerous physical parameters, has been enhanced with GPS-equipped sensor nodes (see Figure 1) to provide relative position and movement information of the observed agent. To achieve high-accuracy positioning at reasonable cost, the system leverages low-cost GPS receivers in combination with offline post-processing of the collected data [4].

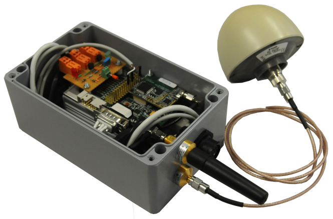


Figure 1: GPS-equipped Sensor Node Prototype.

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Computer Engineering and Networks Lab
(TIK)

## 3 Project Aim

Due to operation in remote areas, wireless sensor nodes are generally battery operated devices. However, due to size and cost restrictions, batteries cannot be arbitrarily large to satisfy the energy requirement. Furthermore, inaccessible deployment sites complicate the replacement of batteries significantly. Therefore sensor nodes must adhere to a very stringent power budget to achieve acceptable lifetimes of months to years. Unfortunately, GPS receivers are very power-hungry devices, and the differential processing approach employed further exacerbates the energy requirement by depending on measurement periods in the order of multiple consecutive hours per day.

Contrary to common assumption in many WSNs, the radio module of the nodes is not necessarily the main energy consumer. Certain applications may rely on sensors that have power requirements exceeding that of common radio modules. Coupled with possibly long acquisition times, the sensing operation, rather than the data communication, can become the main energy consumer. For this reason, energy harvesting techniques, such as photo-voltaic energy harvesting, are often used to recharge the batteries, permitting continuous operation of the sensor network. However, the amount of energy that a PV-system can harvest depends on deployment site, particular location and orientation of the solar panel, and the solar radiation. While the radiation at the selected deployment site can be computed a priori, actual values will vary due to non-deterministic weather patterns and site specific factors, resulting in varying energy input.

To schedule energy-hungry sensors as efficiently as possible and maintaining extended operation, knowledge about the available energy as well as expected future energy input is required. Unfortunately, exactly determining a batterys fill level is only possible with no load connected, and, for run-time measurements, requires hardware support. To provide a light-weight solution to this problem, the state of charge can be approximated by monitoring the battery voltage and measurement of drain current under load. Due to the inaccuracies of this method, state information must be maintained in software to adjust for non-idealities and improve the approximation accuracy.

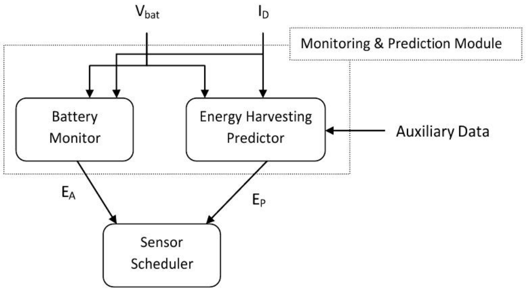
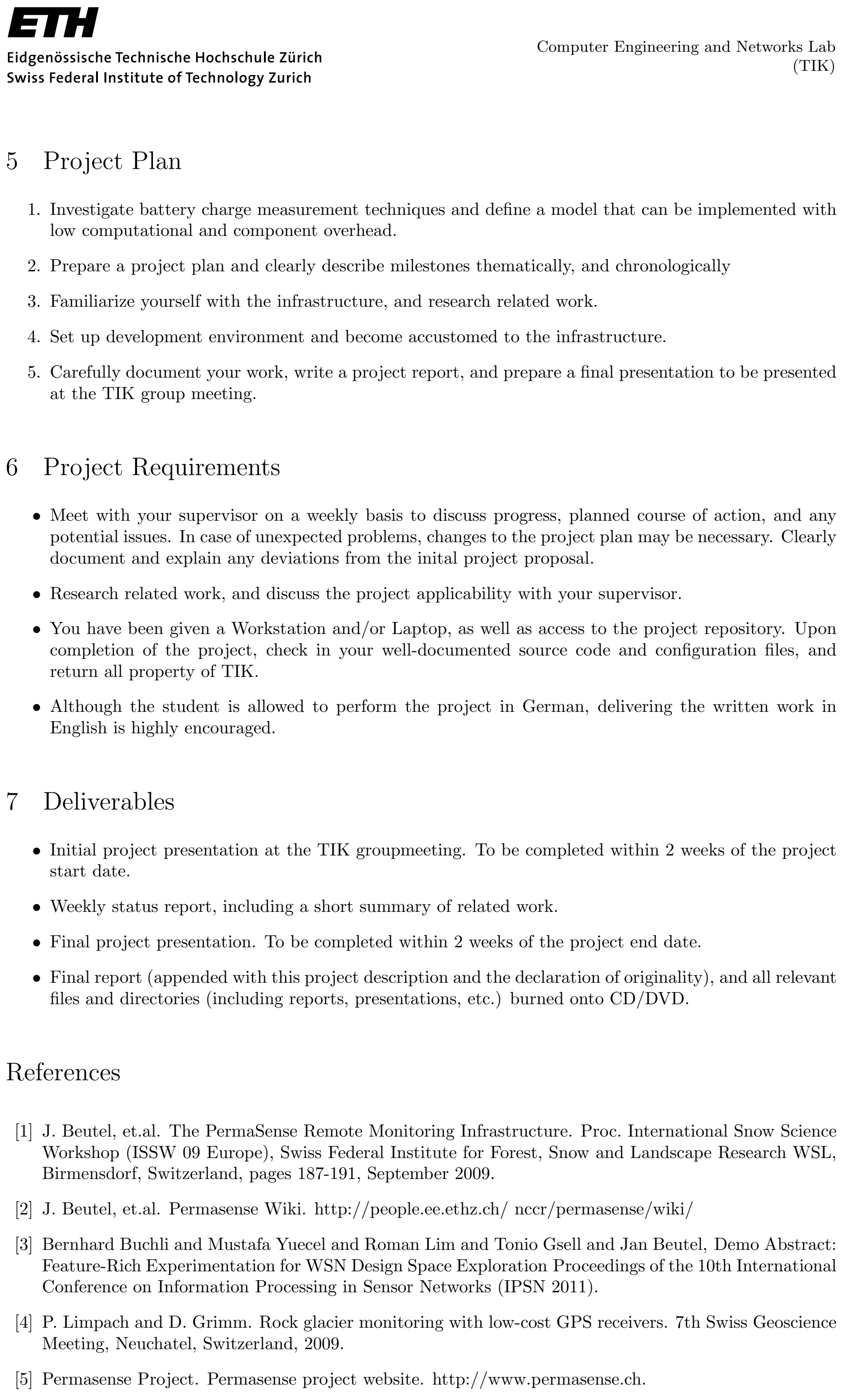


Figure 2: Battery Monitoring and Harvesting Prediction.

## 4 Project Goals

As part of this project, you will devise and implement a model for tracking energy input and estimating available energy stored in the battery using the light-weight approach introduced above. Then you will implement the model for the backlog application running on the CoreStation [2, 3] and evaluate its performance and accuracy (compared to a coulomb counter). As a second step, you will implement functionality that allows runtime adaption of individual components' duty-cycles based on the available energy stored in the battery. Time permitting, you will implement a simple energy input prediction model and simulate and evaluate its performance compared to optimal scheduling (full knwoledge of future input), as well as static scheduling.



ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Computer Engineering and Networks Lab
(TIK)

## 5 Project Plan

1. Investigate battery charge measurement techniques and define a model that can be implemented with low computational and component overhead.
2. Prepare a project plan and clearly describe milestones thematically, and chronologically
3. Familiarize yourself with the infrastructure, and research related work.
4. Set up development environment and become accustomed to the infrastructure.
5. Carefully document your work, write a project report, and prepare a final presentation to be presented at the TIK group meeting.

## 6 Project Requirements

- Meet with your supervisor on a weekly basis to discuss progress, planned course of action, and any potential issues. In case of unexpected problems, changes to the project plan may be necessary. Clearly document and explain any deviations from the inital project proposal.
- Research related work, and discuss the project applicability with your supervisor.
- You have been given a Workstation and/or Laptop, as well as access to the project repository. Upon completion of the project, check in your well-documented source code and configuration files, and return all property of TIK.
- Although the student is allowed to perform the project in German, delivering the written work in English is highly encouraged.

## 7 Deliverables

- Initial project presentation at the TIK groupmeeting. To be completed within 2 weeks of the project start date.
- Weekly status report, including a short summary of related work.
- Final project presentation. To be completed within 2 weeks of the project end date.
- Final report (appended with this project description and the declaration of originality), and all relevant files and directories (including reports, presentations, etc.) burned onto CD/DVD.

## References

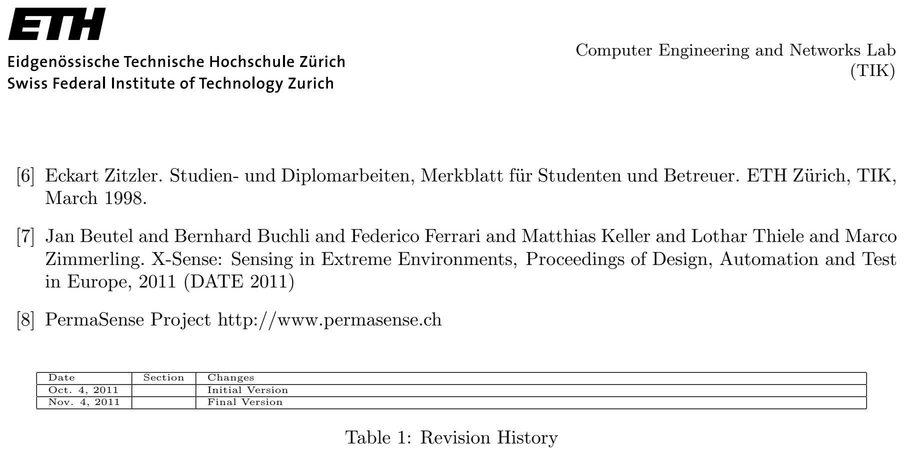
[1] J. Beutel, et.al. The PermaSense Remote Monitoring Infrastructure. Proc. International Snow Science Workshop (ISSW 09 Europe), Swiss Federal Institute for Forest, Snow and Landscape Research WSL, Birmensdorf, Switzerland, pages 187-191, September 2009.

[2] J. Beutel, et.al. Permasense Wiki. http://people.ee.ethz.ch/ nccr/permasense/wiki/

[3] Bernhard Buchli and Mustafa Yuecel and Roman Lim and Tonio Gsell and Jan Beutel, Demo Abstract: Feature-Rich Experimentation for WSN Design Space Exploration Proceedings of the 10th International Conference on Information Processing in Sensor Networks (IPSN 2011).

[4] P. Limpach and D. Grimm. Rock glacier monitoring with low-cost GPS receivers. 7th Swiss Geoscience Meeting, Neuchatel, Switzerland, 2009.

[5] Permasense Project. Permasense project website. http://www.permasense.ch.



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Computer Engineering and Networks Lab
(TIK)

[6] Eckart Zitzler. Studien- und Diplomarbeiten, Merkblatt für Studenten und Betreuer. ETH Zürich, TIK, March 1998.

[7] Jan Beutel and Bernhard Buchli and Federico Ferrari and Matthias Keller and Lothar Thiele and Marco Zimmerling. X-Sense: Sensing in Extreme Environments, Proceedings of Design, Automation and Test in Europe, 2011 (DATE 2011)

[8] PermaSense Project http://www.permasense.ch

| Date | Section | Changes |
|---|---|---|
| Oct. 4, 2011 | | Initial Version |
| Nov. 4, 2011 | | Final Version |

Table 1: Revision History

# List of Figures

# List of Tables

# Bibliography

Aberer, K., Hauswirth, M., and Salehi, A. (2006). A middleware for fast and flexible sensor network deployment. In *Proceedings of the 32nd international conference on Very large data bases*, VLDB '06, pages 1199–1202. VLDB Endowment.

Buchli, B., Yuecel, M., Lim, R., Gsell, T., and Beutel, J. (2011). Demo abstract: Feature-rich experimentation for wsn design space exploration. In *Proceedings of the 10th International Conference on Information Processing in Sensor Networks (IPSN 2011)*, pages 115–116, Chicago, IL, USA. ACM/IEEE.

Doyle, M., Fuller, T. F., and Newman, J. (1993). Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell. *Journal of the Electrochemical Society*, 140(6):1526.

Jongerden, M. and Haverkort, B. (2008). Battery modeling.

Lifeline Batteries Inc (2012). Lifeline Technical Manual. `http://www.lifelinebatteries.com/manual.pdf` [Accessed: March 28, 2012].

Manwell, J. F. and McGowan, J. G. (1993). Lead acid battery storage model for hybrid energy systems. *Solar Energy*, page 399.

Maxim Integrated Products (1998). Evaluation Kit for the MAX1660. `http://datasheets.maxim-ic.com/en/ds/MAX1660EVKIT.pdf` [Accessed: March 28, 2012].

Morningstar Corporation (2012). SunSaver 6 (Gen 3) Installation and Operation Manual. `http://www.morningstarcorp.com/en/support/library/SS3.IOM.Operators_Manual.01.EN.pdf` [Accessed: March 28, 2012].

PermaSense Consortium (2012). PermaSense. `http://www.permasense.ch`.

Rakhmatov, D. N. and Vrudhula, S. B. K. (2001). An analytical high-level battery model for use in energy management of portable electronic systems. In *Proceedings of the 2001*

*IEEE/ACM international conference on Computer-aided design*, ICCAD '01, pages 488–493, Piscataway, NJ, USA. IEEE Press.

Rao, R., Vrudhula, S., and Rakhmatov, D. N. (2003). Battery modeling for energy-aware system design. *Computer*, 36:77–87.

Rao, V., Singhal, G., Kumar, A., and Navet, N. (2005). Battery model for embedded systems. In *Proceedings of the 18th International Conference on VLSI Design held jointly with 4th International Conference on Embedded Systems Design*, VLSID '05, pages 105–110, Washington, DC, USA. IEEE Computer Society.

# Eigenständigkeitserklärung

Ich erkläre hiermit, dass es sich bei der von mir eingereichten schriftlichen Arbeit mit dem Titel **"Battery State of Charge Modeling"** um eine von mir selbständig und in eigenen Worten verfasste Originalarbeit handelt.

**Verfasser:**
Daniel Aschwanden

**Betreuer:**
Ben Buchli

Mit meiner Unterschrift bestätige ich, dass ich über fachübliche Zitierregeln unterrichtet worden bin und das Merkblatt (`http://www.ethz.ch/students/exams/plagiarism_s_de.pdf`) gelesen und verstanden habe. Die im betroffenen Fachgebiet üblichen Zitiervorschriften sind eingehalten worden. Eine Überprüfung der Arbeit auf Plagiate mithilfe elektronischer Hilfsmittel darf vorgenommen werden.

Zürich, March 28, 2012 Dani Aschwanden