

Model-based energy analysis of battery powered systems

Marijn Remco Jongerden

Promotiecommissie:

Voorzitter: Prof. dr. ir. Anton J. Mouthaan
Promotoren: Prof. dr. ir. Boudewijn R.H.M. Haverkort
Prof. dr. ir. Joost-Pieter Katoen

Commissieleden:

Prof. dr. Maria Fox University of Strathclyde, Glasgow
Prof. dr. ing. Markus Siegle Universität der Bundeswehr München
Prof. dr. ir. Twan Basten Technische Universiteit Eindhoven
Prof. dr. Frits W. Vaandrager Radboud Universiteit Nijmegen
Prof. dr. ir. Gerard J.M. Smit Universiteit Twente
Dr. ir. Maurits de Graaf Universiteit Twente,
Thales Nederland B.V., Huizen

CTIT

CTIT Ph.D.-thesis Series No. 10-183

Centre for Telematics and Information Technology
University of Twente
P.O. Box 217, 7500 AE Enschede
The Netherlands

ISSN 1381-3617

ISBN 978-90-365-3114-6

Publisher: Wöhrmann Print Service.

Cover design: Marijn Jongerden.

MODEL-BASED ENERGY ANALYSIS OF BATTERY POWERED SYSTEMS

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
prof. dr. H. Brinksma,
volgens besluit van het College voor Promoties,
in het openbaar te verdedigen
op vrijdag 10 december 2010 om 15.00 uur

door

Marijn Remco Jongerden

geboren op 24 april 1976
te Amsterdam

Dit proefschrift is goedgekeurd door de promotoren,
Prof. dr. ir. Boudewijn R.H.M. Haverkort en Prof. dr. ir. Joost-Pieter Katoen.

Abstract

The use of mobile devices is often limited by the lifetime of the included batteries. This lifetime naturally depends on the battery’s capacity and on the rate at which the battery is discharged. However, it also depends on the usage pattern, *i.e.*, the workload, of the battery. When a battery is continuously discharged, a high current will cause it to provide less energy until the end of its lifetime than a lower current. This effect is termed the rate-capacity effect. On the other hand, during periods of low or no discharge current, the battery can recover to a certain extent. This effect is termed the recovery effect. In order to investigate the influence of the device workload on the battery lifetime a battery model is needed that includes the above described effects.

Many different battery models have been developed for different application areas. We make a comparison of the main approaches that have been taken. Analytical models appear to be the best suited to use in combination with a device workload model, in particular, the so-called kinetic battery model. This model is combined with a continuous-time Markov chain, which models the workload of a battery powered device in a stochastic manner.

For this model, we have developed algorithms to compute both the distribution and expected value of the battery lifetime and the charge delivered by the battery. These algorithms are used to make comparisons between different workloads, and can be used to analyse their impact on the system lifetime.

In a system where *multiple* batteries can be connected, battery *scheduling* can be used to “spread” the workload over the individual batteries. Two approaches have been taken to find the optimal schedule for a given load. In the first approach scheduling decisions are only taken when a change in the workload occurs. The kinetic battery model is incorporated into a priced-timed automata model, and we use the model checking tool **Uppaal Cora** to find schedules that lead to the longest system lifetime.

The second approach is an analytical one, in which scheduling decisions can be made at any point in time, that is, independently of workload changes. The analysis of the equations of the kinetic battery model provides an upper bound for the battery lifetime. This upper bound can be approached with any type

Abstract

of scheduler, as long as one can switch fast enough. Both the approaches show that battery scheduling can potentially provide a considerable improvement of the system lifetime. The actual improvement mainly depends on the ratio between the battery capacity and the average discharge current.

Contents

Abstract	v
1 Introduction	1
2 Battery modeling	7
2.1 Battery basics	7
2.2 Electrochemical models	10
2.3 Electrical-circuit models	10
2.4 Analytical models	12
2.4.1 Peukert’s law	12
2.4.2 Kinetic Battery Model	13
2.4.3 Rakhmatov and Vrudhula’s diffusion model	16
2.5 Stochastic models	18
2.5.1 Chiasserini and Rao	18
2.5.2 Stochastic modified KiBaM	20
2.6 Evaluation	22
3 Comparing the KiBaM and the diffusion model	25
3.1 KiBaM coordinate transformation	25
3.2 Discretized diffusion model	27
3.3 Comparing the analytical models	28
3.3.1 Continuous discharge	29
3.3.2 Frequency response	29
3.3.3 Computing lifetimes	32
3.4 Limitations of analytical battery models	36
3.5 Conclusions	37
3.A Appendix	38
4 Computing battery lifetime distributions	39
4.1 Markov reward models	39
4.1.1 Homogeneous Markov reward models	39
4.1.2 Inhomogeneous Markov reward models	40

CONTENTS

4.2	Computing distributions and expected values	43
4.2.1	Discretization of the state space	43
4.2.2	Transitions in the new generator	44
4.2.3	Battery lifetime	47
4.2.4	Delivered charge	47
4.2.5	Complexity	49
4.3	Stochastic workload models	50
4.4	Results	51
4.4.1	Modeled battery	52
4.4.2	On/Off model (SWL1)	53
4.4.3	Simple & burst model (SWL2 & SWL3)	57
4.5	Conclusion	61
5	Computing best battery schedules using priced timed automata	63
5.1	Battery scheduling	63
5.2	Priced timed automata	65
5.2.1	Networks of timed automata	65
5.2.2	Schedule generation using PTA	68
5.3	A timed automata model for KiBaM	68
5.3.1	Discretization of the KiBaM	69
5.3.2	Towards modeling	70
5.3.3	Basic battery model	72
5.3.4	Battery scheduling	72
5.3.5	Complexity	75
5.4	Validation of the TA-KiBaM	77
5.5	Scheduling results	78
5.6	Towards random loads	82
5.7	Conclusions	85
6	An analytic and simulation approach to battery scheduling	87
6.1	Nonuniform scheduler generation	87
6.2	Greedy scheduler	92
6.2.1	Introduction	92
6.2.2	Constant load	92
6.2.3	Greedy scheduler results	93
6.3	Parallel discharge	95
6.4	Random load scheduling	96
6.4.1	Simulation set-up	96
6.4.2	Round robin frequency dependence	97
6.4.3	Random times	99
6.4.4	Random currents	100
6.4.5	Full random load	102

6.5 Discussion	104
7 Concluding remarks	105
A A first investigation of the practical efficiency of battery scheduling	109
A.1 Experimental setup	109
A.2 The batteries	110
A.3 Discharge measurements	112
A.4 Results	112
A.5 Discussion	117
Bibliography	119
Samenvatting	125
Dankwoord	129

Chapter 1

Introduction

With the proliferation of cheap wireless access technologies, such as wireless LAN, Bluetooth as well as GSM, the number of wireless devices an average citizen is using has been steadily increasing over the last decade. For example, between 2004 and 2008 the number of mobile phones within the Netherlands has grown from 16 million to 19.7 million, and the percentage of households owning a laptop grew from 27% to 62% [11]. Such devices not only add to the flexibility with which we can live our lives and do our work, but also add to our reachability and our security. Next to these personal wireless devices, an ever growing number of wireless devices is used for surveillance purposes, most notably in sensor-type networks. A common issue to be dealt with in the design of all of these devices is power consumption. Since all of these devices use batteries of some sort, mostly rechargeable, achieving low power consumption for wireless devices has become a key design issue. This fact is witnessed by many recent publications on this topic, for example the special issue of *IEEE Computer* (November 2005) that has been devoted to it [34]. Also, conferences dedicated to energy efficiency are becoming commonplace now, such as the e-Energy conference [21] or the ICGreen conference [33].

Low-power design is a very broad area in itself, with so-called “battery-driven system design” a special branch of it, that becomes, due to the reasons mentioned, more and more important. A key issue to be addressed is to find the right tradeoff between battery usage and required performance: how can we design a (wireless) system such that with a given battery, good performance (throughput, reachability, and so on) is obtained, for a long-enough period? Stated differently, how should the processes in the wireless device be organized such that the battery lifetime (which determines the system lifetime) will be as high as possible? In this thesis, the battery lifetime is the time of one discharge period of the battery, from full to empty.

Influence of workload on battery lifetime

The impact of the workload on the battery lifetime is not straightforward. The workload of many devices can not be easily defined, because of random influences, such as human behavior. However, an overall workload pattern can often be modeled using a stochastic workload model. For example, in [60] Simunic *et al.* use semi-Markov decision processes to model a portable device. The model is used to compute dynamic power management strategies, *i.e.*, when to put the device in a power-saving sleep mode. Here a trade off has to be made between lowering the overall power consumption, which extends the battery lifetime, and the perceived performance of the device, since waking up from the sleep mode will cost extra time. Similar research is done by Chen *et al.* in [12], where the workload is modeled by a continuous-time Markov chain to investigate the system power and latency characteristics of dynamic power management. In this model also the power consumption induced by the mode switching is included.

In the mentioned studies, the focus is on lowering the overall power consumption to improve the battery lifetime, without influencing the overall system performance. The battery is considered to be an ideal power source, which will always supply its full capacity. This is not realistic. In a real battery two non-linear properties play an important role. The first is the *rate-capacity effect*. When a battery is continuously discharged, a high discharge current will cause it to provide less energy until it is emptied than a lower current. On the other hand, during periods of low or no discharge current, the battery can recover some of its “lost” capacity. This is termed the *recovery effect*. These effects result in the fact that the battery lifetime is not only determined by the average load, but also by the way the load is distributed over time.

In this thesis, we investigate the impact of the workload on the battery lifetime. We combine stochastic workload models, described by continuous-time Markov chains, with a battery model. For this combined model we supply algorithms for the computation of both the distribution and expected value of the battery lifetime and amount of charge delivered by the battery.

Before we can do this, we need a battery model that correctly models the two non-linear effects described above. In the literature, many different battery models can be found. Highly detailed electro-chemical models [19, 20] have been developed for the use in battery design. In electrical engineering, electrical circuit models [25, 26], which describe the electrical properties of the battery, are used. Next to these specialized models, also various high-level analytic and stochastic battery models are available. We have investigated what is the model that can best be used in combination with a workload model. Here, the best model is a model that does include the important non-linear battery properties to yield accurate computations, but has a relatively low computational complexity, thus keeping the composed model still manageable. The model that adheres to these

conditions is the kinetic battery model [42, 43, 44].

Battery scheduling

The extent to which one can change a workload is often limited by the performance a user expects. A user does not want to wait to use his phone just because it may be beneficial for the battery in the future. However, in systems where more than one battery can be connected there is more freedom to influence the workload for each of the batteries without any impact on the overall system performance as perceived by the user. By switching between the batteries the system is always powered, and at the same time one can change the workload of each of the batteries.

Besides in devices powered by multiple batteries, battery scheduling may also be beneficial in sensor networks. Although each sensor, in general, is powered by only one battery, the entire network is powered by many. Often there are several routes from a sensor node to the data sink to send the collected data through the network. To keep all the sensors powered as long as possible, battery-aware routing has to be done, *i.e.*, the decision on which sensor has to forward the data has to be based on the status of the sensor's batteries. Switching from one route to the other will give the batteries time to recover and thus results in a longer lifetime to the sensor network as a whole. In this way, the routing problem can be regarded as a battery scheduling problem.

Some work on system lifetime improvement by battery scheduling has already been done, for example in [7, 15]. In these studies, some simple scheduling schemes, like round robin or best-of-two scheduling, are used. In the former scheme the used battery is switched at regular intervals regardless of the status of the battery, in the latter scheme the battery that is best, *i.e.*, with the most charge left, is picked. The results show an increase of system lifetime when battery scheduling is applied. However, it is still unclear what is the *best* way to schedule the batteries, and what is the maximum lifetime gain that one can achieve.

We take two approaches to find the best battery schedule. In the first approach we use priced-timed automata [5] to model a system with two batteries. At fixed moments in time the battery scheduler chooses which battery is to be used. The Uppaal CORA model checker [65] is used to find the optimal battery schedule. Due to the computational complexity of the model, it is not practically feasible to compute schedules for systems with more than two batteries.

The second analytical approach does not have this limitation. In this approach, we relax the conditions on the moments of scheduling; at any moment in time the scheduler is allowed to switch to another battery. Under this condition one can obtain the maximum possible system lifetime, given the equations of the kinetic battery model.

Outline of the thesis

Chapter 2 gives an introduction to the most important physical properties of batteries. Furthermore, an overview of the battery models available in the literature is given. Electro-chemical models describe the chemical processes within the battery in detail, whereas electrical circuit battery models focus on the electrical properties of the battery. Next to these specialized models also high-level analytic and stochastic models are available, that focus on the main battery properties needed to predict the battery lifetime. The different modeling approaches are compared for their suitability to be used in combination with workload models. The simple analytical models, with their compact representation, are best for this purpose.

In **Chapter 3**, two analytical battery models, the kinetic battery model and the diffusion model, are compared in more detail. It is shown for the first time that the two are actually closely related, the former being a first-order approximation of the latter. Also, it is shown that fitting the parameters of the kinetic battery model to the diffusion model results in accurate battery lifetime computations.

In **Chapter 4**, we combine the kinetic battery model with a stochastic workload, modeled by a continuous-time Markov chain. We develop new algorithms to compute the cumulative distribution and mean value of both the battery lifetime and the charge delivered by the battery. The approach is applied to a simple and burst workload. This analysis shows the impact of the workload on the battery lifetime.

In **Chapter 5**, the kinetic battery model is incorporated into a priced-timed automata model. Two batteries are modeled to investigate the impact of battery scheduling on the system lifetime. With the UPPAAL-CORA tool, the model is used to find the best strategies to balance the load over two batteries. This has not been done before. The system lifetime from the priced-timed automata model is compared with some straightforward scheduling schemes.

Chapter 6 provides a new analytic approach to the battery scheduling problem. By loosening the conditions on the scheduling moments an analytic solution can be derived directly from the equations that describe the kinetic battery model. The results show that the optimal lifetime can easily be approached by any scheduling algorithm, as long as the switching between the batteries is fast enough.

In **Chapter 7**, we summarize the content of this thesis, and give some ideas for future work.

Origin of the Chapters

- Chapter 2 and 3 are based on [37] and [38], and the journal paper [39].

-
- [37] M. R. Jongerden and B. R. Haverkort. Battery modeling. Technical Report TR-CTIT-08-01, Centre for Telematics and Information Technology, University of Twente, 2008
 - [38] M. R. Jongerden and B. R. Haverkort. Which battery model to use? In *Proceedings of the 24th UK Performance Engineering Workshop (UKPEW)*, Technical Report Series of the Department of Computing, Imperial College London, pages 76–88, 2008
 - [39] M. R. Jongerden and B. R. Haverkort. Which battery model to use? *IET Software*, 3(6):445–457, December 2009
 - Chapter 4 is an extension of [18]. The extension on the computation of the distribution and expected value of the delivered charge and the expected value of the battery lifetime has been submitted as an extended abstract to the International Conference on Operations Research 2010 in Munich [35].
 - [18] L. Cloth, B. R. Haverkort, and M. R. Jongerden. Computing battery lifetime distributions. In *Proceedings of the 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN '07)*, pages 780–789. IEEE Computer Society Press, 2007
 - [35] M. Jongerden and B. Haverkort. Computing lifetimes for battery-powered devices. *extended abstract accepted for post-conference proceedings of the International Conference on Operations Research, Munich, 2010*
 - Chapters 5 and 6 are extensions of [40] and [36]. The analytical approach to the scheduling problem presented in Chapter 6 was worked on in collaboration with Alexandru Mereacre.
 - [40] M. R. Jongerden, B. R. Haverkort, H. C. Bohnenkamp, and J.-P. Katoen. Maximizing system lifetime by battery scheduling. In *Proceedings of the 39th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2009)*, pages 63–72. IEEE Computer Society Press, 2009
 - [36] M. Jongerden, A. Mereacre, H. Bohnenkamp, B. Haverkort, and J.-P. Katoen. Computing optimal schedules for battery usage in embedded systems. *IEEE Transactions on Industrial Informatics*, 6(3):276–286, August 2010
 - In Appendix A the battery scheduling measurements done by Damien Miliche at Thales Nederland B.V. are presented. This appendix is based on [49].
 - [49] D. Miliche, M. de Graaf, G. Hoekstra, M. Jongerden, and B. Haverkort. A first experimental investigation of the practical efficiency of

battery scheduling. In *Workshop Proceedings of the 23th International Conference on Architecture of Computing Systems (ARCS '10)*, pages 241–246, 2010

Acknowledgements

The work presented in this thesis was supported by the ITEA2 GEODES Project 07013 (grant: PNEI081011), and by the Centre for Telematics and Information Technology (CTIT) of the University of Twente.

Chapter 2

Battery modeling

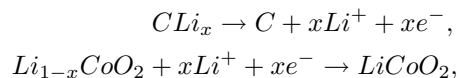
Over the years, many different types of battery models have been developed for different application areas. In this chapter we give an overview of the most important approaches that have been taken. First, we give a short introduction into batteries in Section 2.1, in which we describe the main properties of the battery we want to be modeled. In Sections 2.2 through 2.5 the different battery models are described. We conclude this chapter in Section 2.6, evaluating which of the models is suited to be used in battery performance modeling.

2.1 Battery basics

A battery consists of one or more electrochemical cells. Although strictly speaking a battery consists of multiple cells, a battery is also used to refer to a single cell. In these cells, chemically stored energy is converted into electrical energy through an electrochemical reaction. Figure 2.1 shows a schematic picture of an electrochemical cell. A cell consists of an anode, a cathode and the electrolyte, which separates the two electrodes. During the discharge, an oxidation reaction at the anode takes place. In this reaction a reductant (R_1) donates m electrons (e^-), which are released into the (connected) circuit. At the cathode a reduction reaction takes place. In this reaction, n electrons are accepted by an oxidant (O_2):

$$\begin{cases} R_1 \rightarrow O_1 + me^-, & \text{at the anode,} \\ O_2 + ne^- \rightarrow R_2, & \text{at the cathode.} \end{cases}$$

As an example of a chemical reaction, this is what happens in the widely-used Lithium-ion batteries [51]:



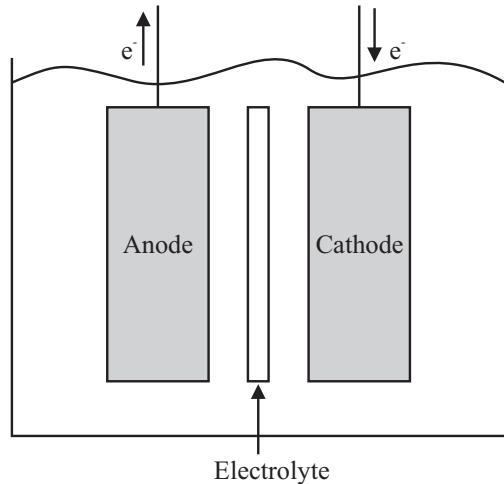


Figure 2.1 Schematic picture of an electrochemical cell.

where $0 < x \leq 1$. These are the reactions for discharging the battery. For charging the battery the arrows in the reaction equations are directed to the left.

Modeling the behavior of batteries is complex, because of non-linear effects during discharge. In the ideal case, the voltage stays constant during discharge, with an instantaneous drop to zero when the battery is empty. The ideal capacity would be constant for all discharge currents, and all energy stored in the battery would be used. However, for a real battery the voltage slowly drops during discharge and the effective capacity is lower for high discharge currents, as illustrated in Figure 2.2. This effect is termed the *rate capacity effect*. Besides this, there is the so-called *recovery effect*: during periods of no or very low discharge, the battery can recover the capacity “lost” during periods of high discharge to a certain extent, as illustrated in Figure 2.3. In this way the effective capacity is increased and the battery lifetime is lengthened. For all types of batteries these effects occur. However, the extent to which they are exhibited depends on the battery type.

The above mentioned effects are mainly caused by the slow diffusion of reactants in the battery. For example, in the Lithium-ion battery, described above, the Li^+ ions made at the anode have to diffuse to the cathode when a current is drawn from the battery. When the current is too high, the internal diffusion cannot keep up with the rate the ions react at the cathode. As a result, the positive charge at the cathode drops and rises at the anode. This causes a drop in the output voltage of the battery. However, when the battery is less loaded for a while, the ions have enough time to diffuse again and charge recovery takes place.

For constant loads, we can easily calculate the ideal battery lifetime (L) by dividing its capacity (C), usually given in mAh or As, by the discharge current

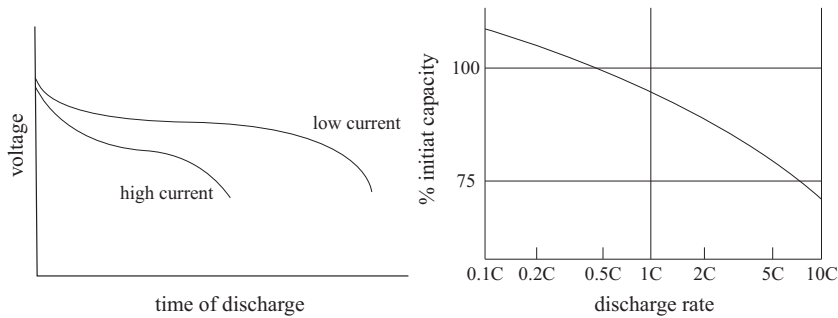


Figure 2.2 Rate capacity effect: The left figure shows the evolution of the voltage over time for a low and high discharge current. The voltage drops faster for high discharge currents. The right figure shows the capacity as a function of the discharge rate. The discharge rate is given in terms of C rating, a C rating of $2C$ means that the battery is discharged in $\frac{1}{2}$ hour. The measured capacities are given relatively to the capacity at the 2 hour discharge rate, $0.5 C$. The figure shows that the effective capacity drops for high discharge rates [45].

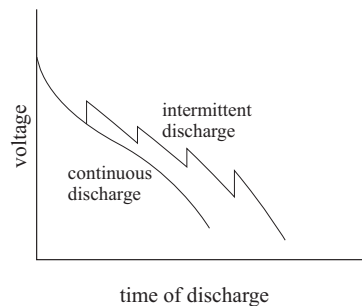


Figure 2.3 Recovery effect: for intermittent discharges the battery can recover during idle periods. In this plot the off-time is not shown, which leads to the vertical jumps in the plot. In this way, one can clearly see the extension of the battery lifetime [45].

(I):

$$L = C/I. \quad (2.1)$$

However, due to the rate capacity and the recovery effects this relation does not hold for real batteries. Many models have been developed to predict real battery lifetimes under a given load. In the following sections several of these models will be discussed.

2.2 Electrochemical models

Electrochemical models are based on the chemical processes that take place in the battery. The models describe these battery processes in great detail. This makes these models the most accurate battery models. However, the highly detailed description makes the models complex and difficult to configure.

Doyle, Fuller and Newman developed an electrochemical model for lithium and lithium-ion cells [20, 23, 24]. This model consists of six coupled, non-linear differential equations. Solving these equations gives the voltage and current as functions of time, and the potentials in the electrolyte and electrode phases, ion concentration, reaction rate and current density in the electrolyte as functions of time and position in the cell. Similar models have been developed for NiCd [19] and alkaline batteries [52].

`Dualfoil` is a Fortran program that uses the model of Doyle *et al.* to simulate lithium-ion batteries. The program is freely available on the internet [22]. It computes how all the battery properties change over time for the load profile set by the user. From the output data, it is possible to obtain the battery lifetime. Besides the load profile, the user has to set over 50 battery related parameters, e.g., the thickness of the electrodes, the initial ion concentration in the electrolyte and the overall heat capacity. To be able to set all these parameters one needs a very detailed knowledge of the battery that is to be modeled. On the other hand, the accuracy of the model is very high. `Dualfoil` is often used as a comparison against other models, instead of using experimental results to check the accuracy.

2.3 Electrical-circuit models

In electrical-circuit models, the electrical properties of the battery are modeled using PSpice circuits [64] consisting of voltage sources, lookup tables and linear passive elements, such as resistors and capacitors. The first electrical-circuit models were proposed by Hageman [26]. He used simple PSpice circuits to simulate nickel-cadmium, lead-acid and alkaline batteries. Gold [25] proposed a similar model for lithium-ion batteries. The core of the models for the different types of batteries is the same:

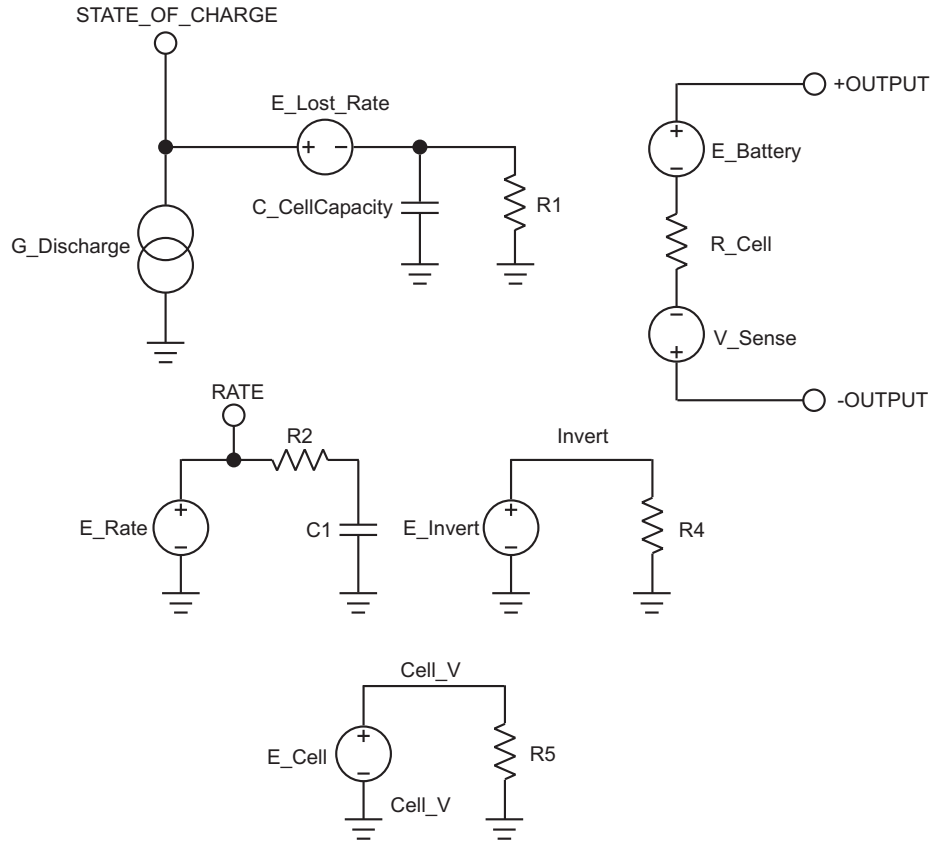


Figure 2.4 Basic functional schematic covering all the modeled cell types. This basic schematic requires minor changes to complete the models for each specific cell [26].

- a capacitor represents the capacity of the battery,
- a discharge-rate normaliser determines the lost capacity at high discharge currents,
- a circuit to discharge the capacity of the battery,
- a voltage versus state-of-charge lookup table,
- a resistor representing the battery's resistance.

Figure 2.4 shows the basic circuits used to model an arbitrary cell. A PSpice program describes the interaction between the different circuits. Minor changes have to be made to complete the model for a specific cell type. Although the models are simpler than the electrochemical models and therefore computationally

less expensive, it still takes quite some effort to configure them. Especially the lookup tables used in the model require much experimental data on the battery’s behavior. Furthermore, the models are less accurate in predicting battery lifetime, having errors up to approximately 12% [25].

2.4 Analytical models

Analytical models describe the battery at a higher level of abstraction than the electrochemical and electrical circuit models. The major properties of the battery are modeled using only a few equations. This makes this type of model much easier to use than the electrochemical and electrical circuit models.

2.4.1 Peukert’s law

The simplest model for predicting battery lifetimes that takes into account part of the non-linear properties of the battery is Peukert’s law [53]. It captures the non-linear relationship between the lifetime of the battery and the rate of discharge, but without modeling the recovery effect. According to Peukert’s law, the battery lifetime (L) can be approximated as:

$$L = \frac{a}{I^b}, \tag{2.2}$$

where I is the discharge current, and a and b are constants which are obtained from experiments. Ideally, a would be equal to the battery capacity and b would be equal to 1. However, in practice a has a value close to the battery’s capacity, and b is a number greater than one. For most batteries the value of b lies between 1.2 and 1.7 [45].

The results obtained by applying Peukert’s law for predicting battery lifetimes are reasonably good for constant continuous loads. But the model does not deal well with variable or interrupted loads. In [53], Rakhmatov and Vrudhula give an extended version of Peukert’s law for non-constant loads: in Equation (2.2), I is replaced by the average current up to $t = L$. For a piecewise constant discharge profile, with t_k the points in time of current change, as shown in Figure 2.5, this yields:

$$L = \frac{a}{\left[\frac{\sum_{k=1}^n I_k(t_k - t_{k-1})}{L} \right]^b}. \tag{2.3}$$

This equation is not as simple as it looks. It is impossible to easily isolate L in the equation, since L turns up inside the n -term sum as well ($t_n = L$). For $n = 1$, Equation (2.3) reduces to (2.2). Although the extended Peukert’s law can handle non-constant discharge profiles, it is still too simple. Only the average discharge current is taken into account, and the recovery effect is not taken into account.

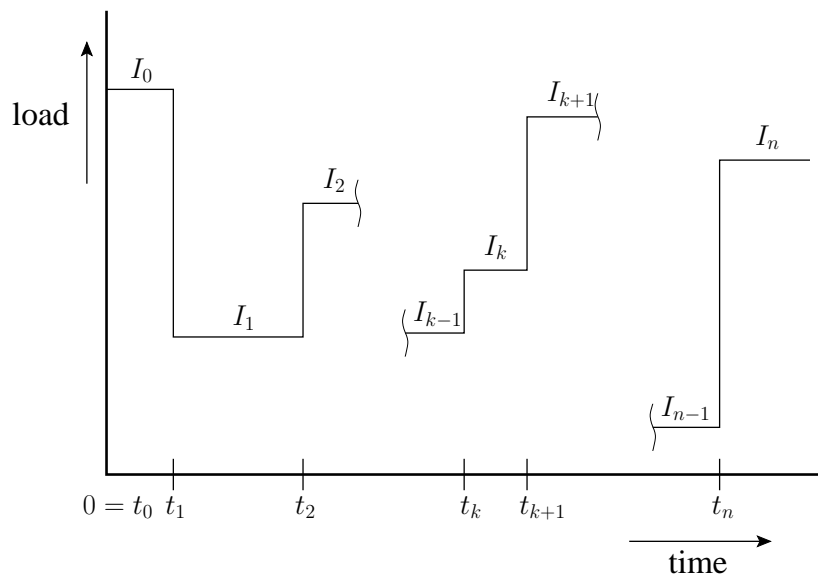


Figure 2.5 An example of a piecewise constant discharge profile.

2.4.2 Kinetic Battery Model

A second analytical model is the Kinetic Battery Model (KiBaM) of Manwell and McGowan [42, 43, 44]. The KiBaM is a very intuitive battery model. It is called kinetic because it uses a chemical kinetics process as its basis. In the model, the battery charge is distributed over two wells: the *available-charge* well and the *bound-charge* well (cf. Figure 2.6). A fraction c of the total capacity is put in the available-charge well (denoted $y_1(t)$), and a fraction $1 - c$ in the bound-charge well (denoted $y_2(t)$). The available-charge well supplies electrons directly to the load ($i(t)$), whereas the bound-charge well supplies electrons only to the available-charge well. The charge flows from the bound-charge well to the available-charge well through a “valve” with fixed conductance, k . The parameter k has the dimension 1/time and limits the rate at which the charge can flow between the two charge wells. Next to this parameter, the rate at which charge flows between the wells depends on the height difference between the two wells. The heights of the two wells are given by: $h_1(t) = y_1(t)/c$ and $h_2(t) = y_2(t)/(1 - c)$. The change of the charge in both wells is given by the following system of differential equations:

$$\begin{cases} \frac{dy_1}{dt} = -i(t) + k(h_2 - h_1), \\ \frac{dy_2}{dt} = -k(h_2 - h_1), \end{cases} \quad (2.4)$$

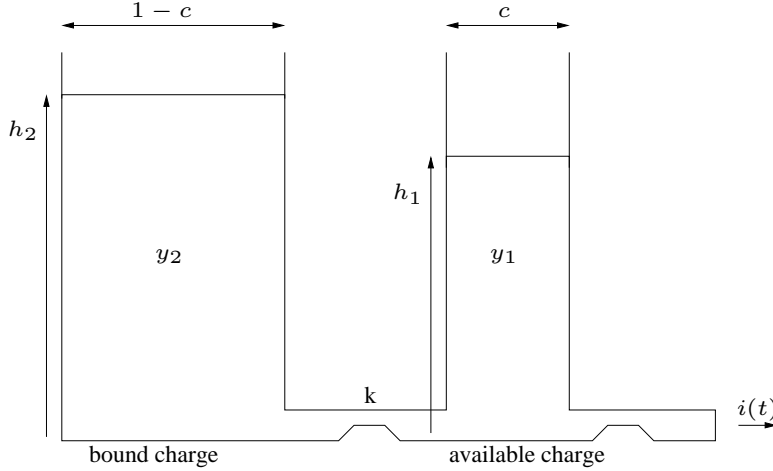


Figure 2.6 Two-well-model of the Kinetic Battery Model.

with initial conditions $y_1(0) = c \cdot C$ and $y_2(0) = (1 - c) \cdot C$, where C is the total battery capacity. The battery is considered empty when it is observed that there is no charge left in the available-charge well.

When a load is applied to the battery, the available charge reduces, and the height difference between the two wells grows. When the load is removed, charge flows from the bound-charge well to the available-charge well until h_1 and h_2 are equal. So, during an idle period, more charge becomes available and the battery lasts longer than when the load is applied continuously. In this way the recovery effect is taken into account. Also, the rate capacity effect is covered, since for a higher discharge current the available-charge well will be drained faster, hence, less time will be available for the bound charge to flow to the available charge. Therefore, more charge will remain unused, and the effective capacity is lower.

The differential equations (2.4) can be solved for the case of a constant discharge current ($i(t) = I$) using Laplace transforms, which yields:

$$\begin{cases} y_1(t) &= -cIt + cC - \frac{I(1-c)}{k'} (1 - e^{-k't}), \\ y_2(t) &= -(1-c)It + (1-c)C + \frac{I(1-c)}{k'} (1 - e^{-k't}), \end{cases} \quad (2.5)$$

where k' is defined as $k' = k/(c(1-c))$. From these equations one can obtain the battery lifetime (L). From $y_1(t) = 0$ follows:

$$L = \frac{C}{I} + \frac{1}{k'} \left(1 - \frac{1}{c} + W \left(\frac{1-c}{c} e^{-\frac{c k'}{I} + \frac{1-c}{c}} \right) \right), \quad (2.6)$$

where W is the Lambert W function. The Lambert W function is the inverse function of $f(x) = xe^x$ [69].

Equation (2.5) describes the evolution of the available and bound charge for the special case of a constant load starting with a full battery. The equations can be generalized to be able to compute the evolution of the available and bound charge in the case of a piecewise constant discharge profile. For a piecewise constant load, the differential equations (2.4) can be solved iteratively for each constant piece, by using the values of the available and bound charge at the end of one piece as initial condition for the next. For the general initial conditions $y_1(0) = \bar{y}_1$ and $y_2(0) = \bar{y}_2$ the evolution of the available and bound charge is given by:

$$\begin{cases} y_1(\bar{y}_1, \bar{y}_2, I, t) &= -cIt + c(\bar{y}_1 + \bar{y}_2) + ((1-c)\bar{y}_1 - c \cdot \bar{y}_2)e^{-k't} \\ &\quad - \frac{(1-c)I}{k'}(1 - e^{-k't}), \\ y_2(\bar{y}_1, \bar{y}_2, I, t) &= -(1-c)It + (1-c)(\bar{y}_1 + \bar{y}_2) + (c\bar{y}_2 - (1-c)\bar{y}_1)e^{-k't} \\ &\quad + \frac{(1-c)I}{k'}(1 - e^{-k't}). \end{cases} \quad (2.7)$$

It is straightforward to verify that Equation (2.7) reduces to (2.5) in the case that $y_1(0) = cC$ and $y_2(0) = (1 - c)C$.

Next to the charge in the battery, the KiBaM models the voltage during discharge. The battery is modeled as a voltage source in series with an internal resistance. The level of the voltage varies with the depth of discharge. The voltage is given by:

$$V = E - IR_0, \quad (2.8)$$

where I is the discharge current and R_0 is the internal resistance. E is the internal voltage, which is given by:

$$E = E_0 + AX + \frac{BX}{D - X}, \quad (2.9)$$

where E_0 is the internal battery voltage of the fully charged battery, A is a parameter reflecting the initial linear variation of the internal battery voltage with the state of charge, B and D are parameters reflecting the decrease of the battery voltage when the battery is progressively discharged, and X is the normalized charge removed from the battery. These parameters can be obtained from discharge data. At least 3 sets of constant discharge data are needed for the non-linear least square curve fitting, which is described in detail in [43].

The KiBaM was developed to model large lead-acid storage batteries, with a capacity of approximately 200 Ah. These batteries have a flat discharge profile, which is well captured by (2.8) and (2.9). These equations do not hold for the modern batteries used in mobile devices, like Li-ion batteries, which have a sloped discharge profile. However, if one is only interested in the battery lifetime, and not so much in its actual voltage during discharge, one can still use the two-well model of the KiBaM, because the it describes both the rate capacity and the recovery effect. In Chapter 3, we compare battery lifetimes according to the KiBaM with the `Dualfoil` program. The results show a close correspondence between the two

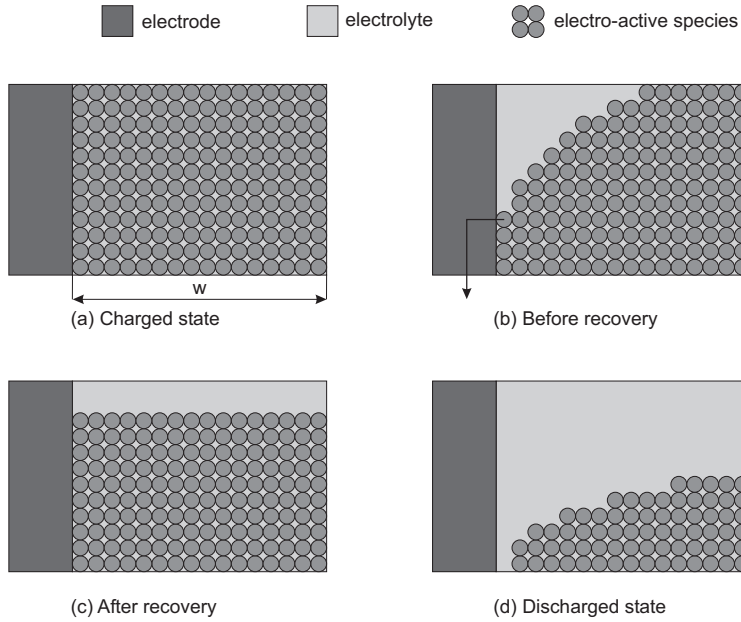


Figure 2.7 Physical picture of the model by Rakhmatov and Vrudhula [53, 54, 55].

models for the modeled Li-ion battery. For other battery types, one still may need to adapt the term of the flow charge between the two wells in (2.4), which for example is done in [57] for Nickel-metal hydride (Ni-MH) batteries. This model is described in detail in Section 2.5.2.

2.4.3 Rakhmatov and Vrudhula’s diffusion model

A third analytical model was developed in 2001 by Rakhmatov and Vrudhula [53, 54, 55]. This model is based on the diffusion of the ions in the electrolyte. The model describes the evolution of the concentration of the electro-active species in the electrolyte to predict the battery lifetime under a given load. In the model the processes at both electrodes are assumed to be identical, thus the battery is assumed symmetric with respect to the electrodes and only one of the electrodes is considered.

Figure 2.7 shows a simplified view of the battery operation according to the diffusion model. At first, for the full battery, the concentration of the electro-active species is constant over the full width (w) of the electrolyte (Figure 2.7(a)). When a load is applied to the battery, the electrochemical reaction results in a reduction of the concentration of the species near the electrode. Thus, a gradient is created across the electrolyte (Figure 2.7(b)). This gradient causes the species

to diffuse towards the electrode. Now, when the load is switched off, the concentration of the species at the electrode will increase again (recover) due to the diffusion, and eventually the species will be evenly distributed over the electrolyte again. The concentration, however, will be lower than for the full battery (Figure 2.7(c)). Finally, when the concentration at the electrode drops below a certain value (C_{cutoff}), the chemical reaction can no longer be maintained and the battery is considered to be empty (Figure 2.7(d)).

The concentration of the electro-active species at time t and distance $x \in [0, w]$ is denoted by $C(x, t)$. For the full battery the concentration is constant over the width w of the electrolyte: $C(x, 0) = C^*$, $x \in [0, w]$. The battery is considered empty when $C(0, t)$ drops below the cutoff level C_{cutoff} . The evolution of the concentration is described by Fick's laws [53]:

$$\begin{cases} -J(x, t) &= D \frac{\partial C(x, t)}{\partial x}, \\ \frac{\partial C(x, t)}{\partial t} &= D \frac{\partial^2 C(x, t)}{\partial x^2}, \end{cases} \quad (2.10)$$

where $J(x, t)$ is the flux of the electro-active species at time t and distance x from the electrode, and D is the diffusion constant. The flux at the electrode surface ($x = 0$) is proportional to the current ($i(t)$). The flux on the other side of the diffusion region ($x = w$) equals zero. This leads to the following boundary conditions:

$$\begin{cases} D \frac{\partial C(x, t)}{\partial x} \Big|_{x=0} &= \frac{i(t)}{\nu F S}, \\ D \frac{\partial C(x, t)}{\partial x} \Big|_{x=w} &= 0, \end{cases} \quad (2.11)$$

where S is the area of the electrode surface, F is Faraday's constant, and ν is the number of electrons involved in the electrochemical reaction at the electrode surface.

It is possible to obtain an analytical solution for the set of partial differential equations (2.10) together with the initial condition and the boundary conditions (2.11) using Laplace transforms. From that solution one can obtain an expression for the apparent charge lost from the battery ($\sigma(t)$) [56]:

$$\sigma(t) = \underbrace{\int_0^t i(\tau) d\tau}_{l(t)} + \underbrace{\int_0^t i(\tau) \left(2 \sum_{m=1}^{\infty} e^{-\beta^2 m^2 (t-\tau)} \right) d\tau}_{u(t)}, \quad (2.12)$$

where $\beta = \pi\sqrt{D}/w$. The apparent charge lost can be separated in two parts: the charge lost to the load ($l(t)$) and the unavailable charge ($u(t)$). The first is the charge used by the device. The second is charge which remains unused in the

battery, depicted in Figure 2.7(d). The battery is empty when the apparent charge lost is equal to the battery’s capacity.

For a constant current I , (2.12) can easily be solved. For $l(t)$ one obtains: $l(t) = It$. For the unavailable charge one can interchange the integral and the summation, which leads to:

$$u(t) = 2I \sum_{m=1}^{\infty} \frac{1 - e^{-\beta^2 m^2 t}}{\beta^2 m^2}. \quad (2.13)$$

During idle periods, the unavailable charge will decrease and will be available again for the load. One can compute the function that describes the evolution of the unavailable charge during an idle period after a load I that lasted for a period of length t_l :

$$u(t_i) = 2I \sum_{m=1}^{\infty} \frac{e^{-\beta^2 m^2 t_i} (1 - e^{-\beta^2 m^2 t_l})}{\beta^2 m^2}, \quad (2.14)$$

where t_i is the idle time.

In [53, 54, 55] the authors compare their diffusion model with the `Dualfoil` battery simulation program. The results of the `Dualfoil` simulations are used as reference values, since these simulations are very precise. For constant continuous loads, the model predicts lifetimes with an average error of 3%, and a maximum error of 6% compared to those obtained using the `Dualfoil` program. For interrupted and variable loads in the experiments, the diffusion model does even better, with a 2.7% maximum error and an average error of less than 1%.

2.5 Stochastic models

Stochastic models aim to describe the battery in an abstract manner, like the analytical models. However, the discharging and the recovery effect are described as stochastic processes.

2.5.1 Chiasserini and Rao

The first stochastic battery models were developed by Chiasserini and Rao. Between 1999 and 2001, Chiasserini and Rao published a series of papers on battery modeling based on discrete-time Markov chains [13, 14, 15, 16]. In [14], two models of a battery of a mobile communication device for transmitting packets are described. In the first and simplest model, the battery is described by a discrete time Markov chain with $N + 1$ states, numbered from 0 to N (cf. Figure 2.8). The state number corresponds to the number of charge units available in the battery. One charge unit corresponds to the amount of energy required to transmit a single packet. N is the number of charge units directly available based on continuous

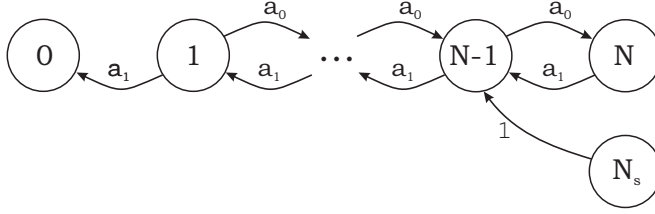


Figure 2.8 The basic Markov chain battery model by Chiasserini and Rao [14].

use. In this simple model, every time step either a charge unit is consumed with probability $a_1 = q$ or recovery of one unit of charge takes place with probability $a_0 = 1 - q$. The battery is considered empty when the absorbing state 0 is reached or when a maximum of T charge units have been consumed. The number of T charge units is equal to the theoretical capacity of the battery ($T > N$).

For this simple model it is possible to give analytical expressions for the properties of interest. The main properties investigated are the expected number of transmitted packets (\bar{m}_p) and the gain (G) obtained from a pulsed discharge relative to a constant discharge, defined as: $G = \bar{m}_p/N$. Clearly, pulsed discharge will lead to a gain that exceeds 1, due to the possibility to recover some charge units. However, this model is too simple. The rate of recovery is not constant during discharge, and in most systems the discharge current changes over time.

In the models in [13, 15, 16], several extensions are made to solve these problems. To improve the model, the recovery probability is made state-dependent. When less charge units are available, the probability to recover a charge unit will become smaller. Next to the state dependence of the recovery, there is a phase dependence. The phase number (f) is a function of the number of charge units that has been consumed. When more charge units have been consumed, the phase number increases and this causes the probability of recovery to decrease. Also, it is possible to consume more than one charge unit in any one time step, with a maximum of M charge units ($M \leq N$). In this way a more bursty consumption of energy can be modeled. Another aspect that has been added to the model, is the non-zero probability of staying in the same state. This means no energy consumption or recovery takes place during a time step.

In Figure 2.9, the state transition diagram of the model with all the extensions is shown. With probability q_i , i charge units are requested in one time slot. During the idle periods in state j , the battery either recovers one charge unit with probability $p_j(f)$, or stays in the same state with probability $r_j(f)$. The recovery probability in state j and phase f is defined as [15]:

$$p_j(f) = q_0 e^{(N-j)g_N - g_C(f)}, \quad (2.15)$$

where g_N and $g_C(f)$ depend on the recovery behavior of the battery. One can

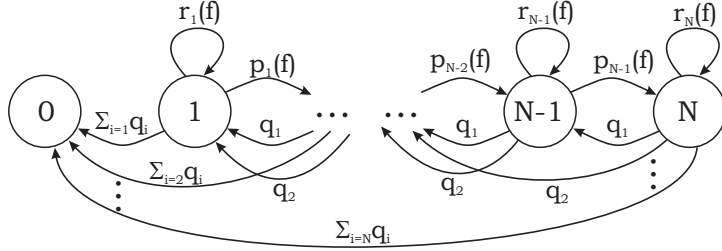


Figure 2.9 The extended Markov chain battery model by Chiasserini and Rao [15].

model different loads by setting the transition probabilities appropriately.

In [15], the final version of the model is used to model a lithium-ion battery. To model the battery, N is set to $\sim 2 \times 10^6$, and 2 phases are used. This results in a Markov chain with approximately 4×10^6 states. The model is analyzed by simulation, and the results are compared with `Dualfoil` (cf. Section 2.2) [15]. With both models, the gain obtained from pulsed discharge compared to constant discharge is calculated for different discharge currents. The results of the stochastic model have a maximum deviation of 4% from the electro-chemical model, with an average deviation of 1%. These results show that the stochastic model gives a good qualitative description of battery behavior under pulsed discharge. However, it is unclear how well the model performs quantitatively, since only results of the gain and no numbers for the computed lifetimes are given.

2.5.2 Stochastic modified KiBaM

Rao et al. [57] proposed a stochastic battery model in 2005, based on the analytical Kinetic Battery Model (KiBaM) proposed by Manwell and McGowan. The stochastic KiBaM is used to model a Ni-MH battery, instead of a lead-acid battery for which KiBaM originally was developed. To be able to model this different type of battery, a couple of modifications have been made to the model. First, in the term corresponding to the flow of charge from the bound-charge well to the available-charge well an extra factor h_2 is added, changing (2.4) into:

$$\begin{cases} \frac{dy_1}{dt} = -i(t) + k_s h_2 (h_2 - h_1), \\ \frac{dy_2}{dt} = -k_s h_2 (h_2 - h_1). \end{cases} \quad (2.16)$$

This causes the recovery to be slower when less charge is left in the battery. The second modification is that in the stochastic model the possibility of no recovery during idle periods is added.

The battery behavior is represented by a discrete-time Markov process. The states of the Markov chain are of the form (i, j, t) . The parameters i and j are the

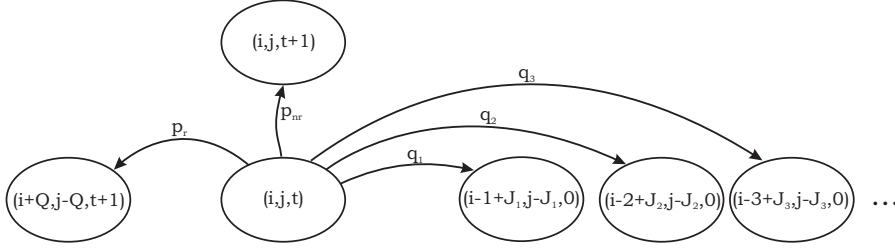


Figure 2.10 Part of the state transition diagram of the stochastic KiBaM [57].

discretized levels of the available-charge well and bound-charge well respectively, and t is the discretized length of the current idle slot; this is the number of time steps taken since the last time some current was drawn from the battery.

Figure 2.10 shows a part of the state transition diagram. The transitions are summarized as follows:

$$(i, j, t) \longrightarrow \begin{cases} (i + Q, j - Q, t + 1), \\ (i, j, t + 1), \\ (i - I + J_I, j - J_I, 0), \end{cases} \quad (2.17)$$

where $0 \leq i \leq M$, $0 \leq j \leq N$ and $t \geq 0$. M and N correspond to the discretized levels of the full available and bound-charge well, respectively. The first two transitions in Equation (2.17) correspond to the time steps in which the current is zero. With probability p_r , the battery recovers Q charge units, and with probability p_{nr} no recovery occurs. Both p_r and p_{nr} depend on the length of the idle timeslot (t). The third transition corresponds to the time steps in which a current is drawn from the battery. With probability q_I , I charge units are drawn from the available-charge well, and at the same time J_I charge units are transferred from the bound to the available-charge well.

The probabilities q_I are defined by the load profile. Since the q_I are equal for all states, it is impossible to control in what sequence the currents are drawn from the battery in this model, and thus to fully model a real usage pattern.

In the model of the Ni-MH battery the charge in the available and bound-charge well is discretized in $27 \cdot 10^7$ and $45 \cdot 10^7$ charge units respectively. This results in a Markov chain too big to handle as a whole, and no analytical solution to the model can be given. To obtain battery lifetimes several runs of discharging the battery are simulated with the model.

In [57], Rao et al. compare the calculated battery lifetimes with some experimental results. In a simple experimental setup, different periodic loads are applied to an AAA Ni-MH battery, and its lifetime is measured. In the first set of experiments the frequency of the applied load is varied, keeping the ratio of on and off time constant at one. In these experiments, the battery lifetime increases as the

frequency decreases. In a second set of experiments, the ratio between on and off time is varied by keeping the on-time fixed to 2 seconds and increasing the off-time from 0 to 3.5 seconds. As expected, the lifetime and the delivered charge increase when the off-time increases, since the battery has more time to recover.

The results of the simulations show that the model is quite accurate for predicting battery lifetime and charge drawn from the battery. since a maximum error of 2.65% for the simulations, with regard to the experimental values, was found.

2.6 Evaluation

We want to use a battery model to combine it with a workload model. With this combination it will be possible to model the energy consumption of battery powered devices, and predict battery lifetimes for different usage patterns. For this purpose, we need a “simple” battery model that still gives a good description of the most important non-linear effects, i.e., the rate capacity effect and recovery effect.

Table 2.1 gives an overview of the different battery models and their relevant properties. Most battery models are not well suited to be combined with a workload model. Although the electro-chemical model is the most accurate model and the `Dualfoil` program is often used as “reality” to check the performance of other battery models, the model is too complex for our needs. A very detailed knowledge of the battery is necessary to be able to set all the parameters of the model. Furthermore, the computational complexity of solving the six coupled partial differential equations is very high, which makes the execution of the program slow. Like the electro-chemical model, the electrical circuit models are too complex. The modeling of the battery’s electrical properties is too detailed for what we want from the battery model. Peukert’s formula, on the other hand, is too simple. It could be easily integrated with a workload model. However, it does not take the recovery effect into account. Therefore, it will underestimate battery lifetimes for usage patterns with idle periods. The stochastic model by Chiasserini is also too limited. The model is designed for pulsed discharge of the battery, and it focuses on the recovery effect only.

The KiBaM and the diffusion model by Rakhmatov et al. do take into account both the rate capacity effect and the recovery effect. Both models use a system of two differential equations to describe the battery and with both models one can compute the battery lifetime for an arbitrary piecewise constant load profile. The compact description of the relevant battery processes combined with the availability of an analytical solution makes these models well suited for our purpose. In the next chapter a more detailed comparison between these two analytical models is made. We will see that, although the two models may seem different, the KiBaM

	battery type	rate capacity effect	recovery effect	number parameters	accuracy
Dualfoil [20, 24, 23]	Li-ion	+	+	> 50	very high
Electrical circuit [26]	Ni-Cd, alkaline Lead-acid	+	+	15-30	medium
Peukert's law [53]	all	+	-	2	medium, 10% error
Diffusion model [53]	Li-ion	+	+	2	high, 5% error
KiBaM [42]	Lead-acid	+	+	2	high
Chiasserini [14, 13, 16, 15]	Li-ion	-	+	2	high, 1% error
Stochastic KiBaM [57]	Ni-MH	+	+	2	high, 2% error

Table 2.1 Battery models overview. The errors of Peukert's law, the diffusion model and the stochastic model of Chiasserini are relative to the results of the `Dualfoil` program.

Battery modeling

is actually a first order approximation of the diffusion model.

Chapter 3

Comparing the KiBaM and the diffusion model

In this chapter we take a closer look at the two analytical models which are suitable to use for battery performance modeling, the Kinetic Battery Model and the diffusion model. When one compares the KiBaM and the diffusion model, one sees some similarities. In both models, the charge in the battery has to flow “to one side” to be available for use, and part of it will stay behind in the battery when the battery is empty. The analysis in Sections 3.1 and 3.2 shows that the KiBaM is actually a first order approximation of the diffusion model. A further comparison between the two models is made in Section 3.3. In Section 3.4 we address the limitations of the analytical battery models. Section 3.5 concludes this chapter.

3.1 KiBaM coordinate transformation

To enable the comparison between the KiBaM and the diffusion model we first apply a coordinate transformation to the differential equations of the KiBaM. We recall the differential equations which describe the dynamics of the charge in the two wells:

$$\begin{cases} \frac{dy_1}{dt} = -i(t) + k(h_2 - h_1), \\ \frac{dy_2}{dt} = -k(h_2 - h_1), \end{cases} \quad (3.1)$$

Although these differential equations nicely describe the discharge process of the battery, and an analytical solution can be obtained for constant discharge currents, the equations can be simplified when a coordinate transformation is applied. In this way even more insight can be obtained in the way the model behaves. It also allows to easily obtain an expression for the apparent charge lost, like in Equation (2.12).

From the differential equations of the KiBaM, cf (2.4), one can see that the height difference between the two wells ($h_2 - h_1$) plays a major role in the model. This is one of the coordinates after the transformation, the other is the total charge in the battery. So, the transformation changes the coordinates from y_1 and y_2 to $\delta = h_2 - h_1$ and $\gamma = y_1 + y_2$. This transformation changes the differential equations to:

$$\begin{cases} \frac{d\gamma}{dt} = -i(t), \\ \frac{d\delta}{dt} = \frac{i(t)}{c} - k'\delta, \end{cases} \quad (3.2)$$

where $k' = k/(c(1 - c))$. The initial conditions after the transformation are $\delta(0) = 0$ and $\gamma(0) = C$. In the new coordinate system the condition for the battery to be empty is:

$$\gamma(t) = (1 - c)\delta(t). \quad (3.3)$$

The differential equations (3.2) are independent of each other and are straightforwardly solved for constant discharge currents, $i(t) = I$:

$$\begin{cases} \gamma(t) = C - It, \\ \delta(t) = \frac{I}{c} \cdot \frac{1 - e^{-k't}}{k'}. \end{cases} \quad (3.4)$$

During idle periods, the height difference will decrease due to the flow of charge from the bound-charge well to the available-charge well. One can compute the function that describes the evolution of the height difference during an idle period after a load I that lasted for a period of length t_i :

$$\delta(t_i) = \frac{I}{c} \cdot \frac{e^{-k't_i}(1 - e^{-k't_i})}{k'}, \quad (3.5)$$

where t_i is the length of the idle period considered.

It is possible to use the solution of the transformed KiBaM to divide the charge lost from the battery into a part lost to the load ($l(t)$) and a part that is unavailable ($u(t)$), as is done in the diffusion model, cf. Section 2.4.3. The unavailable charge in the KiBaM is the height difference times $1 - c$. For constant current discharge this yields:

$$l(t) = C - \gamma(t) = It, \quad (3.6)$$

$$u(t) = (1 - c) \cdot \delta(t) = \frac{(1 - c)I}{c} \frac{1 - e^{-k't}}{k'}. \quad (3.7)$$

The evolution of the unavailable charge during an idle period after a load I that lasted for t_i is now given by:

$$u(t_i) = \frac{(1 - c)}{c} \cdot \frac{e^{-k't_i}(1 - e^{-k't_i})}{k'}. \quad (3.8)$$

When one compares (3.7) and (3.8) with the first-order expansion of (2.13) and (2.14), one sees that the two have the same form. This provides a first indication the two models are related.

3.2 Discretized diffusion model

To show that the two models indeed are closely related, we discretize the diffusion model in n steps and show that this results in an n -well KiBaM. To do this, one first needs to normalize the width w (cf. Figure 2.7) of the battery, $x' = x/w$. Now, x' is a dimensionless space coordinate, and takes a value between 0 and 1. This changes the differential equations (2.10) into:

$$\begin{cases} -J(x', t) &= \frac{D}{w} \frac{\partial C(x', t)}{\partial x'}, \\ \frac{\partial C(x', t)}{\partial t} &= \frac{D}{w^2} \frac{\partial^2 C(x', t)}{\partial x'^2}, \end{cases} \quad (3.9)$$

and the boundary conditions into:

$$\begin{cases} \left. \frac{D}{w} \frac{\partial C(x', t)}{\partial x'} \right|_{x'=0} &= \frac{i(t)}{\nu FS}, \\ \left. \frac{D}{w} \frac{\partial C(x', t)}{\partial x'} \right|_{x'=1} &= 0. \end{cases} \quad (3.10)$$

The next step is to transform the ion concentration ($C(x', t)$ in mol/m²) into charge ($h(x', t)$ in As). Every ion yields ν electrons in the chemical reactions. The electric charge per mole of electrons is given by Faraday's constant $F \approx 9.45 \cdot 10^4$ As/mol. This then yields, $h(x', t) = C(x', t)\nu FS$. Substituting this in Equations (3.9) and (3.10) respectively yields

$$\begin{cases} -J(x', t)\nu FS &= \frac{D}{w} \frac{\partial h(x', t)}{\partial x'}, \\ \frac{\partial h(x', t)}{\partial t} &= \frac{D}{w^2} \frac{\partial^2 h(x', t)}{\partial x'^2}, \end{cases} \quad (3.11)$$

for the differential equations, and

$$\begin{cases} \left. \frac{D}{w} \frac{\partial h(x', t)}{\partial x'} \right|_{x'=0} &= i(t), \\ \left. \frac{D}{w} \frac{\partial h(x', t)}{\partial x'} \right|_{x'=1} &= 0, \end{cases} \quad (3.12)$$

for the boundary conditions. Finally, the spatial coordinate x' is discretized. Figure 3.1 gives a schematic overview of the discretized model. The electrolyte is divided in n parts of size $\alpha = 1/n$. The level of the charge in part i is denoted by h_i . We apply the finite difference method for second-order derivatives, as follows,

$$\frac{\partial^2 h}{\partial x^2} = \frac{h(x + \alpha) - 2h(x) + h(x - \alpha)}{\alpha^2}, \quad (3.13)$$

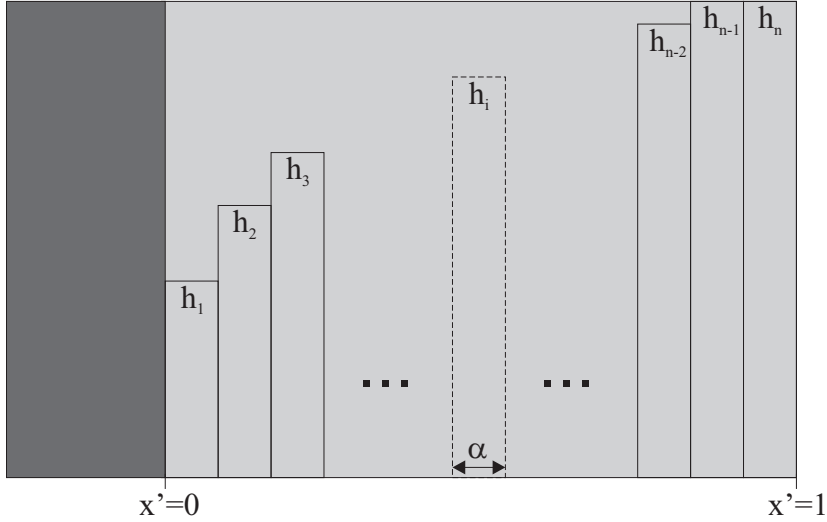


Figure 3.1 Discretization of the diffusion model.

where α is the step-size of the discretization. For $h(0, t)$ we write $h_1(t)$, and for $h(1, t)$ we write $h_n(t)$. This turns the differential equations together with the boundary conditions into a system of n coupled differential equations:

$$\left\{ \begin{array}{l} \frac{\partial h_1(t)}{\partial t} = \frac{1}{\alpha} \left(\frac{D}{\alpha w^2} (h_2 - h_1) - i(t) \right), \\ \frac{\partial h_2(t)}{\partial t} = \frac{1}{\alpha} \left(\frac{D}{\alpha w^2} ((h_3 - h_2) - (h_2 - h_1)) \right), \\ \vdots \\ \frac{\partial h_{n-1}(t)}{\partial t} = \frac{1}{\alpha} \left(\frac{D}{\alpha w^2} ((h_n - h_{n-1}) - (h_{n-1} - h_{n-2})) \right), \\ \frac{\partial h_n(t)}{\partial t} = \frac{1}{\alpha} \left(-\frac{D}{\alpha w^2} (h_n - h_{n-1}) \right). \end{array} \right. \quad (3.14)$$

These equations are exactly the equations one would get when the KiBaM is extended to n equally sized wells. The variable $h_i(t)$ gives the height of well number i at time t . When $n = 2$ the model is reduced to the KiBaM with two wells, each containing half of the total charge, that is, $c = 0.5$, and $k = 2D/w^2$.

3.3 Comparing the analytical models

In Section 3.2, we have shown that the diffusion model is a continuous version of the KiBaM. In this section we make a further comparison of the two models. In Section 3.3.1 and Section 3.3.2 a further theoretical comparison is made. In Section 3.3.3 a practical comparison is made by using both models to compute battery lifetimes for various loads.

3.3.1 Continuous discharge

When we compare (3.7) and (3.8) with (2.13) and (2.14), we see that by setting $c = \frac{1}{3}$ and $k' = \beta^2$ in the KiBaM we exactly obtain the first term of the infinite sum of the diffusion model. This is of course, a bad approximation of the infinite sum. Note that, for t going to infinity the unavailable charge for continuous discharge in the diffusion model will reduce to:

$$\lim_{t \rightarrow \infty} u^{\text{diff}}(t) = \frac{2I}{\beta^2} \frac{\pi^2}{6} \approx \frac{2I}{\beta^2} \cdot 1.64, \quad (3.15)$$

while for the KiBaM the limit is:

$$\lim_{t \rightarrow \infty} u^{\text{KiBaM}}(t) = \frac{1-c}{c} \frac{I}{k'}. \quad (3.16)$$

So, if the KiBaM is used to approximate the diffusion model with $c = \frac{1}{3}$ and $k' = \beta^2$ an error of approximately 64% is made.

One can obtain a much better approximation, when the parameters c and k' are used to fit the KiBaM equation of $u(t)$ to the equation of the diffusion model. Figure 3.2(a) shows the result of a least squares fitting procedure for the case that $I = 1$ A. When $\beta = 0.273 \text{ min}^{-\frac{1}{2}}$, the fit results in $c = 0.166$ and $k' = 0.122 \text{ min}^{-1}$. In Figure 3.2(b) the relative difference between the two curves is shown. This difference is independent of the discharge current. The relative difference is very large, up to 100%, for times smaller than 10 minutes. When the battery lifetime is within this region, i.e., when the battery is discharged with a very high current, the results for battery lifetime computations will give a big difference.

3.3.2 Frequency response

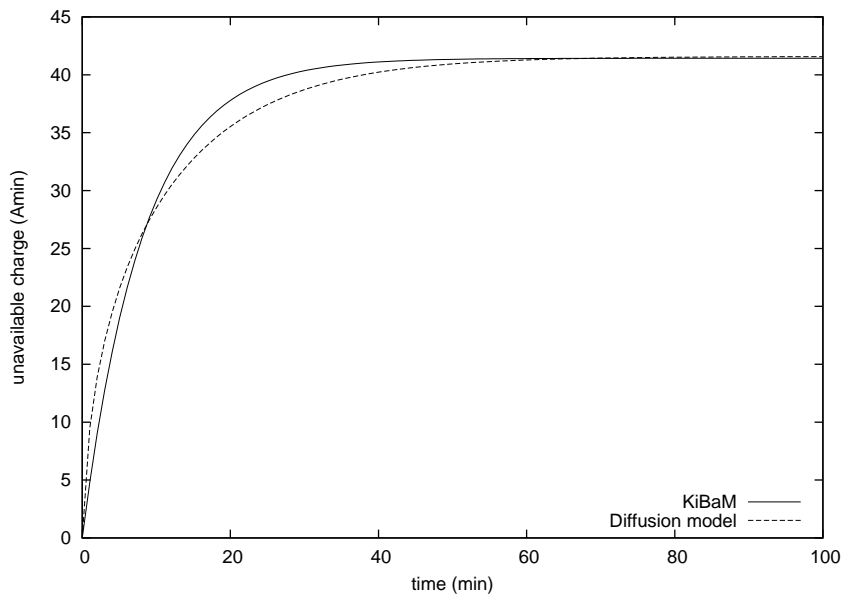
Following the method described in [56], we now perform an analysis of the frequency response of both the KiBaM and the diffusion model. In this method, the battery model is represented by the linear time-invariant (LTI) system shown in Figure 3.3. For both battery models, $h_l(t)$ is the unit step function. For the diffusion model $h_u(t)$ is given by

$$h_u^{\text{diff}}(t) = 2 \sum_{m=1}^{\infty} e^{-\beta^2 m^2 t}, \quad (3.17)$$

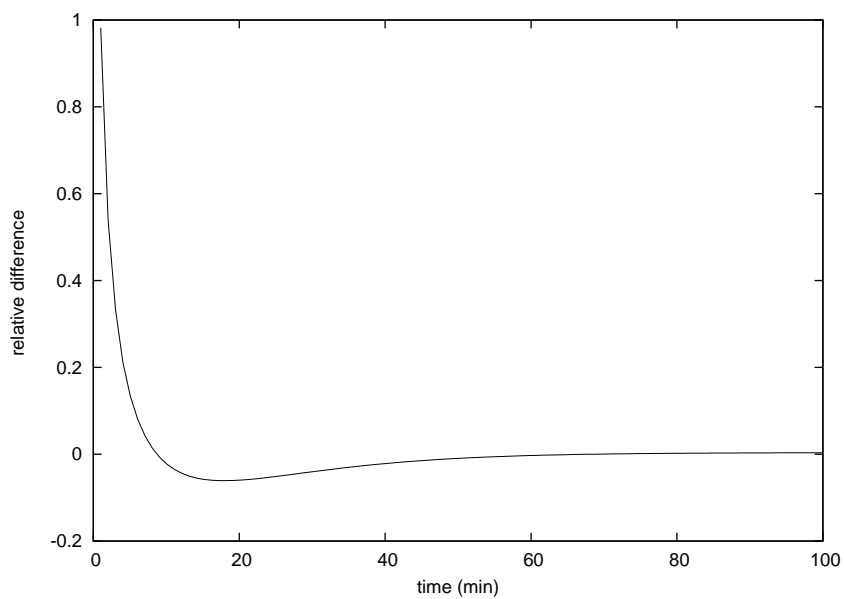
and for the KiBaM it is

$$h_u^{\text{KiBaM}}(t) = \frac{1-c}{c} e^{-k't}. \quad (3.18)$$

The component $h_l(t)$ expresses the actual charge lost and does not depend on the battery parameters. For both models only $h_u(t)$ depends on the battery parameters. Therefore, to characterize the frequency response of the battery it is



(a) fit



(b) relative difference

Figure 3.2 Fit of the KiBaM to the diffusion model. The evolution of the unavailability charge in both the diffusion model and the fitted KiBaM is given in (a). In (b) the relative difference between the two curves, $(u^{\text{diff}}(t) - u^{\text{KiBaM}}(t))/u^{\text{diff}}(t)$, is given.

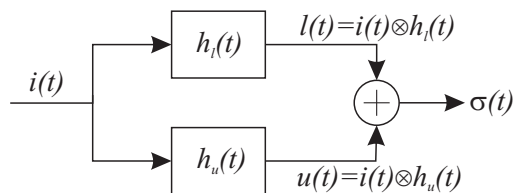


Figure 3.3 Linear time-invariant system of the battery model [56].

sufficient to find the Fourier transform of $h_u(t)$. The Fourier transform $H_u(f)$ is given by:

$$H_u(f) = \int_{-\infty}^{\infty} h_u(t) e^{-2\pi j f t} dt, \quad (3.19)$$

where $j = \sqrt{-1}$. When we apply the Fourier transform to $h_u^{\text{diff}}(t)$ and $h_u^{\text{KiBaM}}(t)$ we obtain:

$$H_u^{\text{diff}}(f) = 2 \sum_{m=1}^{\infty} \frac{1}{\beta^2 m^2 + 2\pi j f} \quad (3.20)$$

for the diffusion model, and

$$H_u^{\text{KiBaM}}(f) = \frac{1-c}{c} \frac{1}{k' + 2\pi j f} \quad (3.21)$$

for the KiBaM. The direct current response, $f = 0$, for the diffusion model can be reduced to $H_u^{\text{diff}}(0) = \pi^2/3\beta^2$.

Figure 3.4 shows the frequency response for both the diffusion model and the KiBaM. The same parameters as in the previous section have been used, $\beta = 0.273 \text{ min}^{-\frac{1}{2}}$, $c = 0.166$ and $k' = 0.122 \text{ min}^{-1}$. The figure shows that the diffusion model has a higher frequency response for high frequencies. This is due to the higher-order terms that are included in the diffusion model and not in the KiBaM. However, both models are highly insensitive to high frequency current switching, in the ideal case the frequency response is zero for all frequencies. The insensitivity implies that currents varying faster than 0.01 Hz can be replaced with an average current without giving significant errors in the battery lifetime computations. Therefore in both models, it is not useful to schedule tasks at small time scales, smaller than minutes, in order to benefit from the recovery effect, since the average current will stay the same. Ordering tasks at processor level will not have any effect on the battery lifetime. However, scheduling on a larger time scale, minutes or longer, can be beneficial.

The frequency response is mainly determined by the size of the recovery parameter (k' or β). When this parameter is increased, the recovery will be faster and the battery behavior will be closer to that of an ideal battery. So, an increase of this parameter results in a higher frequency response, hence, to a higher sensitivity to fine-grained scheduling.

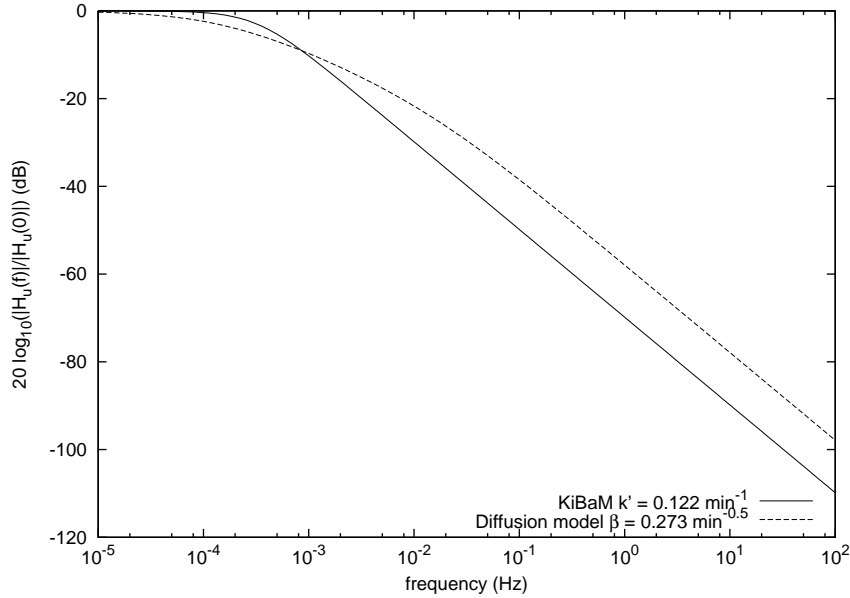


Figure 3.4 Frequency response for KiBaM and diffusion model.

3.3.3 Computing lifetimes

Next to the theoretical analysis of the two models, both models were used to compute battery lifetimes for various load profiles.

In [55], Rakhmatov et al. give the battery lifetimes for load profiles of a Compaq Itsy pocket computer, computed both with their diffusion model and the electro-chemical model *Dualfoil* [22] (cf. Section 2.2). To these results, the lifetimes according to the KiBaM model have been added in Table 3.1 for constant loads and Table 3.2 for variable-load profiles. Details of the variable-load profiles are given in Table 3.3 (in Appendix 3.A).

The lifetimes computed using the KiBaM and diffusion model match very well. The results for continuous discharge only deviate at high discharge currents, as expected from the analysis of the equations, but the difference still is less than 7%. Also, for the variable loads the difference is largest for short battery lifetimes, with a maximum of 5.4% for Case C21.

Figure 3.5 shows a plot of the lifetimes computed with both models versus the lifetimes computed with the electro-chemical simulation program *Dualfoil*. In comparison with *Dualfoil* both models overestimate the battery lifetime for the low continuous loads (long lifetimes), with errors growing upto 10%. The results of the variable loads are even better, with a maximum error of 5%.

Besides the results of the two models, also the lifetimes according to Peukert's

3.3 Comparing the analytical models

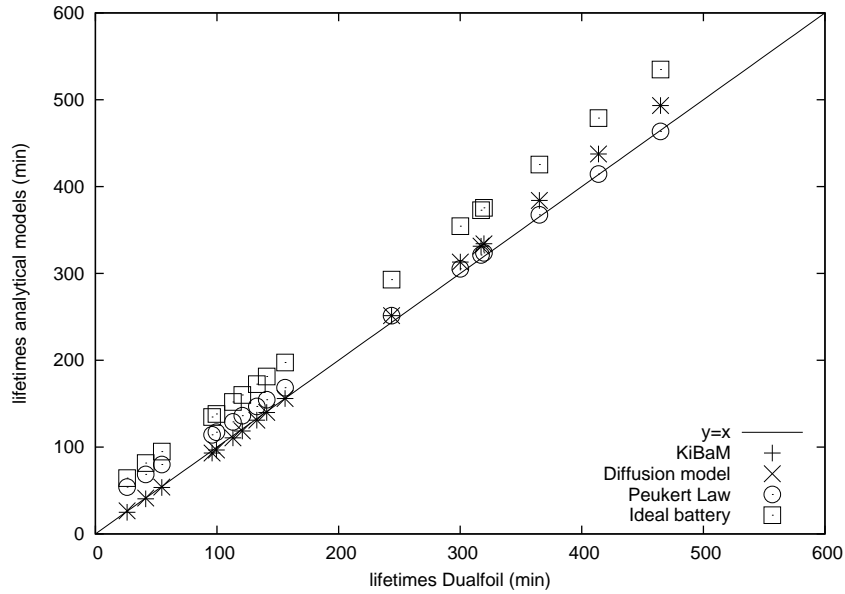
Test	Name	I_{ave} (mA)	Dualfoil (min)	Diffusion t (min)	KiBaM (min)	Peukert (min)	Ideal (min)
T1	MPEG	222.7	140.9	139.9	139.9	154.5	181.3
T2	Dictation	204.5	156.0	156.0	156.0	168.4	197.4
T3	Talk1	108.3	317.2	331.4	331.4	321.3	372.8
T4	Talk2	107.5	319.5	334.1	334.1	323.7	375.6
T5	Talk3	94.9	365.1	384.0	384.0	367.5	425.4
T6	WAV1	84.3	413.7	437.5	437.5	414.4	478.9
T7	WAV2	75.5	464.8	493.3	493.3	463.6	534.8
T8	Idel1	28.0	1278	1400	1401	1270	1442
T9	Idle2	19.5	1852	2029	2029	1835	2071
T10	SleepDC	3.0	12285	13417	13417	12288	13458
T11	IAT	628.0	26	26.6	24.9	53.9	64.3
T12	IAR	494.7	41.3	41.4	40.5	68.6	81.6
T13	IST	425.6	54.6	53.9	53.5	80.0	94.9
T14	ISR	292.3	99.5	96.7	96.7	117.2	138.1
T15	IAD	265.6	113.1	110.6	110.6	129.1	152.0
T16	MSD	252.3	120.8	118.6	118.6	136.1	160.0
T17	DSD	234.1	132.7	131.0	131.0	146.8	172.5
T18	TSD	137.9	243.6	251.3	251.3	251.4	292.8
T19	WSD	113.9	300.1	313.0	313.0	305.3	354.5
T20	ISD	57.6	616.3	659.5	659.5	610.3	701.0
T21	SSD	32.5	1101	1201	1201	1092	1242
T22	Boot	300.0	96.0	93.2	93.1	114.1	134.6

Table 3.1 Battery lifetimes for continuous current discharge computed with Dualfoil, the diffusion model, KiBaM, and the formula's of Peukert's law and the ideal battery. The numbers for Dualfoil, the diffusion model and Peukert's law have been taken from [55].

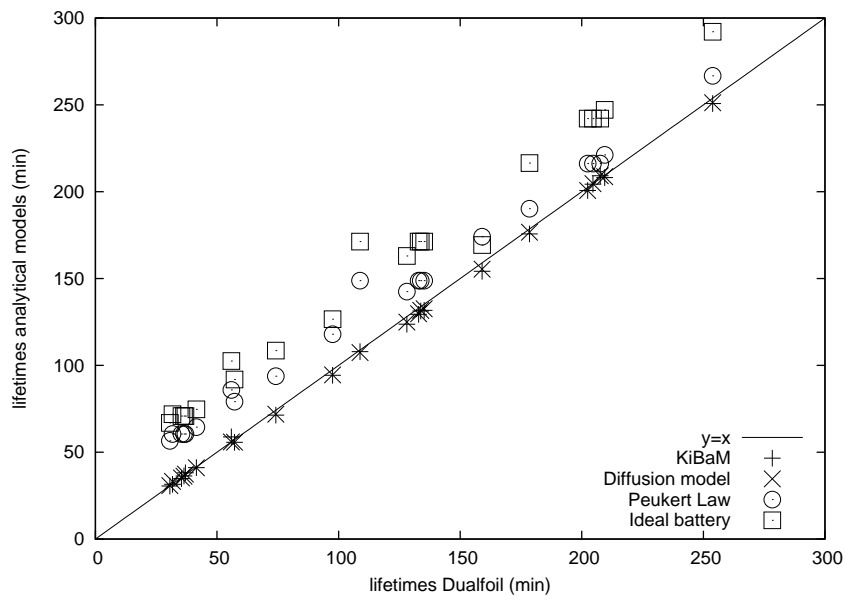
Case	Dualfoil (min)	Diffusion (min)	KiBaM (min)	Peukert (min)	Ideal (min)
C1	36.4	36.2	36.3	60.5	70.8
C2	57.2	55.8	55.7	79.1	91.9
C3	74.2	71.9	71.4	93.8	108.5
C4	128.1	124.9	123.6	142.5	163.0
C5	178.5	176.7	175.7	190.2	216.5
C6	41.5	41.0	41.1	64.4	74.7
C7	30.6	30.8	30.5	56.5	66.9
C8	37.0	37.4	38.1	60.5	70.8
C9	35.4	35.2	34.8	60.5	70.8
C10	135.2	132.6	131.7	148.8	171.3
C11	108.8	107.4	107.9	148.8	171.3
C12	159.0	155.4	154.1	174.1	169.3
C13	133.8	131.7	131.3	148.8	171.3
C14	132.9	129.7	129.4	148.8	171.3
C15	207.6	209.2	209.2	216.2	242.1
C16	202.4	200.7	200.7	216.2	242.1
C17	253.8	251.2	250.8	266.7	292.1
C18	204.6	204.6	204.3	216.2	242.1
C19	209.4	208.7	208.2	221.2	247.1
C20	31.7	33.2	31.5	60.5	71.9
C21	55.9	55.9	58.8	85.9	102.5
C22	97.5	94.5	94.3	117.9	126.6

Table 3.2 Battery lifetimes for variable-load profiles (cf. Appendix 3.A) computed with **Dualfoil**, the diffusion model, KiBaM, and the formula's of Peukert's law and the ideal battery. The numbers for **Dualfoil**, the diffusion model and Peukert's law have been taken from [55].

3.3 Comparing the analytical models



(a) constant load



(b) variable load

Figure 3.5 Computed lifetimes according to the `Dualfoil` simulation program versus the diffusion model and the KiBaM for constant loads (a) and variable loads (b). Next to the two analytical models, the lifetimes according to the formulas of the ideal battery (2.1) and Peukert's law (2.3) are shown.

law and the ideal-battery model are given. The ideal-battery model always predicts longer lifetimes, up to twice as long for high loads, since it does not take into account any loss of capacity due to the rate capacity effect. Also, Peukert's law overestimates the battery lifetimes for most cases. Only for the low continuous loads it gives better predictions than the KiBaM and diffusion model.

3.4 Limitations of analytical battery models

In the previous section we have seen that both models give nearly the same results. In this section, all further results are obtained with the KiBaM, but the conclusions also apply to the diffusion model.

For the KiBaM the effect of a varying load on the charge delivered by the battery was analyzed in more detail. A square wave, switching between on (1 A) and off (0 A), was used as load. In Figure 3.6 the charge delivered is shown as a function of the frequency of the periodic load. For low frequencies the delivered charge is constant, because the battery is emptied during the first on-period. Therefore, the charge delivered is equal to the case of continuous discharge at 1 A. When the frequency is increased, one sees a sudden discontinuous increase of the charge delivered by the battery. At the point of this jump, the battery is nearly empty at the end of the first on-period, and it has an off-period to recover some of its capacity. The recovered charge can be used in the next on-period, resulting in a considerable increase of the delivered charge. After this increase, the delivered charge slowly decreases when the frequency is further increased. The explanation of this decrease is twofold. First, the off-period is shorter and therefore there is less time for recovery. Second, the first on-period is shorter and less charge is delivered to the load during this time.

Further increase of the frequency results in a discontinuous increase of the charge delivered each time the battery can recover during an extra off-period, followed again by a slow decrease. The increase gets smaller for higher frequencies since the extra recovery-time decreases. When the frequency is larger than 10^{-2} Hz, the charge delivered is constant again. This is due to the short extra off-time, and the low frequency response at these high frequencies (cf. Section 3.3.2).

For the chosen load and set of battery parameters the charge delivered is highest for a frequency around 10^{-4} Hz. However, the position of the peaks depends highly on the battery parameters and the level of the on-current, and a slight variation might result in a big change in the charge delivered by the battery. In practice, the battery parameters vary even between batteries of the same size and type. Therefore, it does not make sense to do battery lifetime predictions using single traces of a load profile. The used trace could result in a high performance of the battery with one set of the parameters, and a low performance with a slightly different set of parameters.

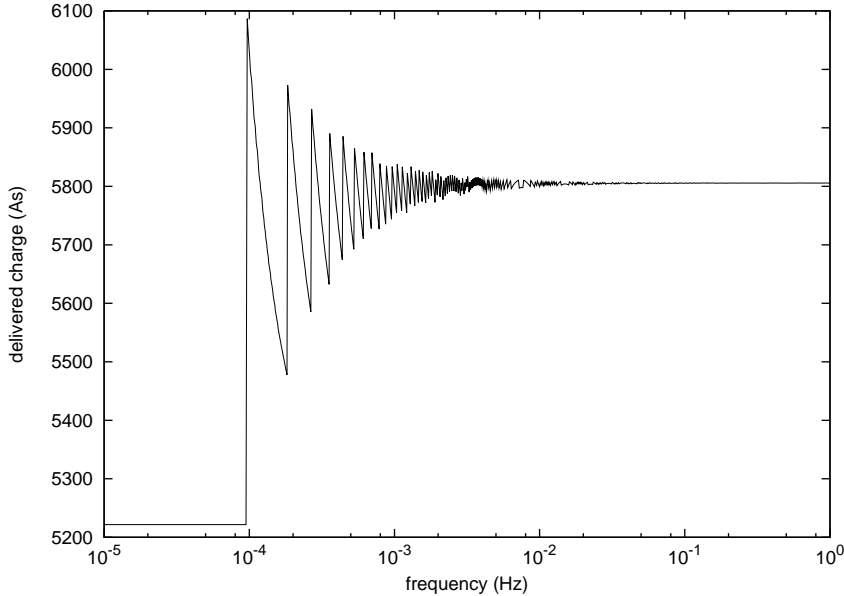


Figure 3.6 Charge delivered by the battery as a function of the frequency for a square wave load. The charge delivered is computed using the KiBaM, with the parameters $c = 0.625$, $k = 4.5 \cdot 10^{-5} \text{ s}^{-1}$ and the capacity of 7200 As.

3.5 Conclusions

Of the battery models available in literature, the analytic models are best suited to be integrated with a workload model. The analysis of the KiBaM and diffusion model shows that the KiBaM is actually a first-order approximation of the diffusion model. The parameters of the KiBaM can be adapted to make a better approximation of the diffusion model. The performed experiments with both models show that this approximation is very good for most practical loads. Therefore, it is better to use the simpler KiBaM model. In the rest of this thesis we will use the KiBaM for modeling the batteries.

However, one has to be careful using this type of model when drawing conclusions from only a few workloads. A slight change in the battery parameters can change the battery lifetime dramatically especially when the load switching frequencies are low. A good way to avoid this problem is to make use of stochastic workload models. With these models one can capture the full range of different possible workloads. This results in a battery lifetime distribution, which tells us the probability of the battery being empty at time t given the type of workload. Comparing these probabilities one can find the best way to use the battery. Slight changes in the battery parameters, now, will not affect the results dramatically.

In the next chapters we present two approaches for extending the system lifetime by smart battery usage. In Chapter 4, we model systems with one battery. We combine the KiBaM with a Markov model that represents the system workload. By adapting the workload one can take advantage of the recovery effect, and extend the battery lifetime. In the Chapters 5 and 6, systems with multiple batteries are considered. By switching between the batteries one can give all batteries more time to recover and in this way extend the system lifetime. Different approaches have been taken to find the best way to switch between the batteries.

3.A Appendix

Case	Description	Timing (min)
C1	IAT-off-IAT	(0, 19.5, 26.0)
C2	IAR-off-IAR	(0, 31.0, 41.3)
C3	IST-off-IST	(0, 41.0, 54.6)
C4	ISR-off-ISR	(0, 74.6, 99.5)
C5	MPEG-off-MPEG	(0, 105.7, 140.9)
C6	IAT-off-IAT	(0, 19.5, 29.9)
C7	IAT-off-IAT	(0, 19.5, 22.1)
C8	IAT-off-IAT	(0, 23.4, 29.9)
C9	IAT-off-IAT	(0, 15.6, 22.1)
C10	Boot-IAT-IAR-MSD-DSD-TSD-WSD-IAD	(0, 0.5, 5.5, 10.5, 35.5, 60.5, 85.5, 110.5)
C11	Boot-WSD-TSD-DSD-MSD-IAR-IAT-IAD	(0, 0.5, 25.5, 50.5, 75.5, 100.5, 105.5, 110.5)
C12	Boot-WSD-TSD-DSD-MSD-IAR-off-... Boot-IAT-IAD	(0, 0.5, 25.5, 50.5, 75.5, 100.5, 105.5, ... 130.5, 131.0, 136.0)
C13	Boot-[IAT-IAR-MSD-DSD-TSD-WSD] ⁵ -IAD	(0, [0.5, 1.5, 2.5, 7.5, 12.5, 17.5] _{22.5} ⁵ , 110.5)
C14	Boot-[WSD-TSD-DSD-MSD-IAR-IAT] ⁵ -IAD	(0, [0.5, 5.5, 10.5, 15.5, 20.5, 21.5] _{22.5} ⁵ , 110.5)
C15	MPEG-Dictation-Talk1-WaV1-MPEG	(0, 50.0, 100.0, 150.0, 200.0)
C16	WAV1-Talk1-Dictation-MPEG-MPEG	(0, 50.0, 100.0, 150.0, 200.0)
C17	WAV1-Talk1-Dictation-off-MPEG-MPEG	(0, 50.0, 100.0, 150.0, 200.0, 250.0)
C18	[WAV1-Talk1-Dictation-MPEG] ¹⁰ -MPEG	([0, 5.0, 10.0, 15.0] _{20.0} ¹⁰ , 200)
C19	[WAV2-Talk3-Dictation-MPEG] ¹⁰ -MPEG	([0, 5.0, 10.0, 15.0] _{20.0} ¹⁰ , 200)
C20	[IAR-IAT] [∞]	([0, 1.0] _{2.0} [∞])
C21	[IAR-IAT-ISD] [∞]	([0, 1.0, 2.0] _{3.0} [∞])
C22	5.0 + (5.0 per min)	(0, 1.0, 2.0, ...)

Table 3.3 The simulated variable-load profiles [55].

Chapter 4

Computing battery lifetime distributions

In this chapter we combine the Kinetic Battery Model with a stochastic workload model. The system workload is modeled using a Markov model. The two charge wells of the KiBaM are integrated into the Markov model as accumulated rewards. This leads to an inhomogeneous Markov reward model, since the reward rates depend on the level of the accumulated rewards, as we will see below. We will show how to compute the distribution and expected value of both the battery lifetime and the charge delivered by the battery.

In Section 4.1 we briefly introduce the Markov reward models, starting with the homogeneous case (Section 4.1.1) and extending the theory to the inhomogeneous case (Section 4.1.2). Section 4.2 presents the approach to computing the distribution and expected value of the battery lifetime and the delivered charge. In Section 4.3 the modeled workloads are presented, and in Section 4.4 the results are given for these workloads. We end with a conclusion in Section 4.5.

4.1 Markov reward models

4.1.1 Homogeneous Markov reward models

A (homogeneous) Markov reward model (MRM) consists of a finite state space $S = \{1, \dots, N\}$, the transition rate matrix $\mathbf{Q} \in \mathbb{R}^{N \times N}$ and a reward vector $\underline{r} \in \mathbb{R}^N$.

The matrix \mathbf{Q} is an infinitesimal generator matrix, *i.e.*, with entries $q_{i,j} \geq 0$, $j \neq i$, and $q_{i,i} = -\sum_{j \in S, j \neq i} q_{i,j}$. The diagonal entry $q_{i,i}$, which is often denoted as $-q_i$ with $q_i = \sum_{j \in S, j \neq i} q_{i,j}$, describes the rate at which state i is left. This rate is to be interpreted as the rate of a negative exponential distribution, *i.e.*, the probability that state i is left within s seconds is given as $1 - e^{-q_i \cdot s}$. The next

state then is j with probability $q_{i,j}/q_i$. The initial distribution of states at time $t = 0$ is denoted as $\underline{\alpha}$. The generator matrix \mathbf{Q} together with $\underline{\alpha}$ determines the Continuous Time Markov Chain (CTMC) $X(t)$.

When in state i , reward is accumulated with rate r_i which might be positive or negative as defined in [32]. The total reward accumulated when residing in state i from time t_1 until time $t_2 \geq t_1$ is denoted $y_i(t_1, t_2)$ and equals

$$y_i(t_1, t_2) = r_i \cdot (t_2 - t_1). \quad (4.1)$$

Given the state process $X(t)$, the accumulated reward at time t , $Y(t)$, is defined as

$$Y(t) = \int_0^t r_{X(s)} ds. \quad (4.2)$$

The distribution of $Y(t)$, the so-called performability distribution [47, 48], equals

$$F^Y(t, y) = \Pr \{Y(t) \leq y\}. \quad (4.3)$$

The corresponding density (with respect to y) equals

$$\begin{aligned} f^Y(t, y) &= \frac{\partial F^Y(t, y)}{\partial y} \\ &= \lim_{h \downarrow 0} \frac{1}{h} \Pr \{y \leq Y(t) \leq y + h\}. \end{aligned} \quad (4.4)$$

An MRM can have more than one reward structure. State i is then equipped with reward rates $r_{i,1}$ through $r_{i,K}$, i.e., we have a reward matrix $\mathbf{R}(t, \underline{y}) \in \mathbb{R}^{N \times K}$ for $\underline{y} \in \mathbb{R}^K$. The accumulated reward is then a vector of random variables $\underline{Y}(t) = (Y_1(t), \dots, Y_K(t))$ and its distribution is defined as

$$F^{\underline{Y}}(t, (y_1, \dots, y_K)) = \Pr \{Y_1(t) \leq y_1, \dots, Y_K(t) \leq y_K\}. \quad (4.5)$$

4.1.2 Inhomogeneous Markov reward models

In the inhomogeneous case, the transition rate matrix \mathbf{Q} and the reward vector \underline{r} can depend on the time t (time-inhomogeneous) and on the accumulated reward y (reward-inhomogeneous). We then have $\mathbf{Q}(t, y)$ and $\underline{r}(t, y)$, where y is the current level of accumulated reward. The reward accumulated between time t_1 and $t_2 \geq t_1$ when residing continuously in state i is described by the following differential equation with initial value $y_i(t_1, t_1) = 0$:

$$\frac{dy_i(t_1, t_2)}{dt_2} = r_i(t_2, y_i(t_1, t_2)). \quad (4.6)$$

This equation describes the rate of change at the end of the interval $[t_1, t_2]$ and so the reward rate depends on t_2 . The accumulated reward until time t in this case is defined as

$$Y(t) = \int_0^t r_{X(s)}(s, Y(s)) ds. \quad (4.7)$$

To integrate the KiBaM into an MRM, we need an MRM that is time-homogeneous but reward-inhomogeneous and has two types of rewards, one for the available charge and one for the bound charge. We therefore denote the generator matrix as $\mathbf{Q}(y_1, y_2)$ and the reward rates as $\mathbf{R}(y_1, y_2) \in \mathbb{R}^{N \times 2}$. The reward accumulated in a state i between time t_1 and time t_2 is described by the following differential equations with initial values $y_{i,1}(t_1, t_1) = y_{i,2}(t_1, t_1) = 0$:

$$\begin{cases} \frac{dy_{i,1}(t_1, t_2)}{dt_2} = r_{i,1}(y_{i,1}(t_1, t_2), y_{i,2}(t_1, t_2)), \\ \frac{dy_{i,2}(t_1, t_2)}{dt_2} = r_{i,2}(y_{i,1}(t_1, t_2), y_{i,2}(t_1, t_2)). \end{cases} \quad (4.8)$$

The accumulated reward is then defined as

$$\underline{Y}(t) = (Y_1(t), Y_2(t)) \quad (4.9)$$

$$= \int_0^t \underline{r}_{X(s)}(\underline{Y}(s)) ds \quad (4.10)$$

$$= \int_0^t (r_{X(s),1}(Y_1(s), Y_2(s)), r_{X(s),2}(Y_1(s), Y_2(s))) ds, \quad (4.11)$$

and its distribution equals

$$F^{(Y_1, Y_2)}(t, y_1, y_2) = \Pr \{Y_1(t) \leq y_1, Y_2(t) \leq y_2\}. \quad (4.12)$$

We assume that the accumulated rewards are nonnegative and are bounded by a minimum $\underline{l} = (l_1, l_2)$ and a maximum $\underline{u} = (u_1, u_2)$. This is reasonable when considering batteries because their charge is always between 0 and a predefined capacity C . We then have

$$\begin{aligned} f^{(Y_1, Y_2)}(t, y_1, y_2) = 0, & \quad \text{for } y_1 < l_1 \text{ or } y_2 < l_2 \\ & \quad \text{or } y_1 > u_1 \text{ or } y_2 > u_2. \end{aligned} \quad (4.13)$$

In the following we often consider the joint distribution of state and accumulated rewards, that is,

$$F_i(t, y_1, y_2) = \Pr \{X(t) = i, Y_1(t) \leq y_1, Y_2(t) \leq y_2\}, \quad (4.14)$$

with density $f_i(t, y_1, y_2)$. The distribution of the accumulated rewards can then be calculated using

$$F^{(Y_1, Y_2)}(t, y_1, y_2) = \sum_{i \in S} F_i(t, y_1, y_2). \quad (4.15)$$

Figure 4.1 shows a schematic representation of the Markov Reward KiBaM (MRKiBaM). The CTMC states $\{1, \dots, N\}$ of the MRKiBaM reflect the different operating modes of the device. Along with the CTMC states, the model has two

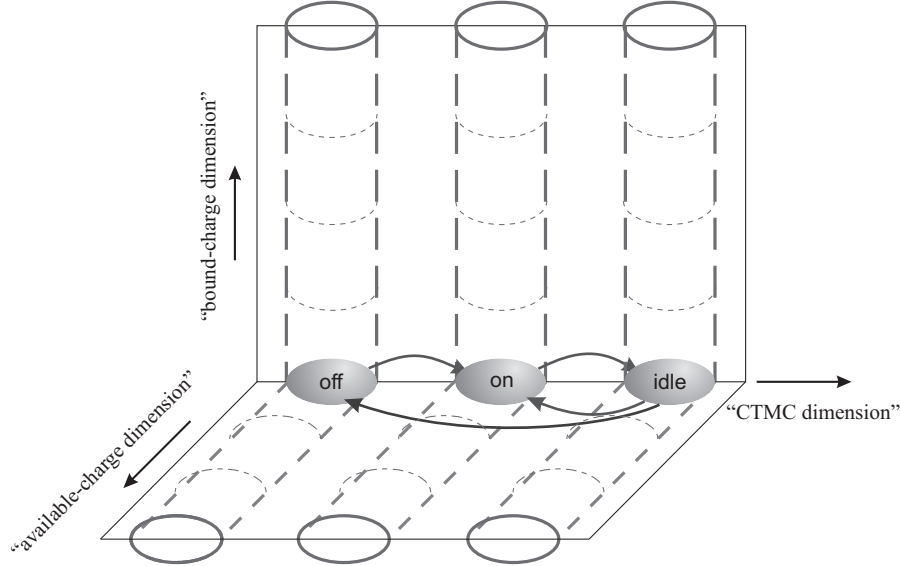


Figure 4.1 Schematic representation of the Markov Reward KiBaM.

accumulated rewards. The first accumulated reward $Y_1(t)$ represents the available-charge well, the second accumulated reward $Y_2(t)$ represents the bound-charge well. The corresponding rates are derived from the KiBaM differential equations (2.4), using the constants k and c and the equations $h_1 = y_1/c$ and $h_2 = y_2/(1-c)$. Let I_i be the current drawn from the battery in state $i \in S$. The first reward rate then is

$$r_{i,1}(y_1, y_2) = \begin{cases} -I_i + k \cdot (h_2 - h_1), & h_2 > h_1 > 0, \\ 0, & \text{otherwise,} \end{cases} \quad (4.16)$$

and the second reward rate is

$$r_{i,2}(y_1, y_2) = \begin{cases} -k \cdot (h_2 - h_1), & h_2 > h_1 > 0, \\ 0, & \text{otherwise.} \end{cases} \quad (4.17)$$

The interesting question for battery-powered devices is “When does the battery get empty?” In our model, the battery is empty at time t if the available-charge well $Y_1(t)$ is empty. Since the accumulated rewards $Y_1(t)$ and $Y_2(t)$ are random variables, we can only indicate the *probability* that the battery is empty at time t :

$$\Pr \{\text{battery empty at time } t\} = \Pr \{Y_1(t) = 0\} \quad (4.18)$$

The *lifetime* L of a battery is the instant the battery is perceived empty for the first time, *i.e.*,

$$L = \min\{t \mid Y_1(t) = 0\}. \quad (4.19)$$

There is some other work available that addresses performability-like measures in an inhomogeneous context. In the 1990's some work has been published on the computation of transient state probabilities for inhomogeneous Markovian models without rewards [67, 58, 68]. A more recent paper [62] characterizes the performability distribution in inhomogeneous MRMs through a coupled system of partial differential equations that is solved through discretization, and used to derive systems of ordinary differential equations to determine moments of accumulated reward.

In what follows we approximate the joint distribution of state process and accumulated reward by the transient solution of a derived homogeneous CTMC, that is, by a phase-type distribution. The approximation is applicable if the generator matrix and the reward rates depend on the current accumulated reward and not on the current time. This is exactly the case with our battery model and we therefore restrict the presentation to a two dimensional reward structure, even though the approach applies for three or more reward types equally well.

4.2 Computing distributions and expected values

In this section we present a numerical algorithm for the computation of the distribution of the accumulated reward in an inhomogeneous Markov reward model. It uses a Markovian approximation, in which the computation is reduced to the transient solution of a homogeneous CTMC via uniformization. The underlying idea already appeared in [10] and is also used in [30] and [31] (steady-state solution). The algorithm is described for homogeneous MRMs with positive reward rates in the Continuous Stochastic Reward Logic (CSRL) context in [28, 29], and extended to reward-inhomogeneous models with positive reward rates in [17].

4.2.1 Discretization of the state space

The joint distribution of state and accumulated reward (4.14) can be rewritten by summing over evenly-sized subintervals of the reward intervals $[l_1, y_1]$ and $[l_2, y_2]$:

$$F_i(t, y_1, y_2) = \sum_{j_1=\frac{l_1}{\Delta}}^{\frac{y_1}{\Delta}-1} \sum_{j_2=\frac{l_2}{\Delta}}^{\frac{y_2}{\Delta}-1} \Pr \left\{ \begin{array}{l} X_t = i, \\ Y_1(t) \in (j_1\Delta, (j_1+1)\Delta], \\ Y_2(t) \in (j_2\Delta, (j_2+1)\Delta] \end{array} \right\}. \quad (4.20)$$

Here, Δ is the step-size at which the state space is discretized.

We want to approximate the terms

$$\Pr \{X_t = i, Y_1(t) \in [j_1\Delta, (j_1+1)\Delta], Y_2(t) \in [j_2\Delta, (j_2+1)\Delta]\} \quad (4.21)$$

in such a way that the computation is done for a homogeneous CTMC without rewards. This is accomplished as follows. An MRM modeling a battery can be

seen as having an infinite and uncountable state space $S \times [l_1, u_1] \times [l_2, u_2]$, where state (s, y_1, y_2) indicates that the “CTMC part” of the MRM is in state s and the accumulated reward of the first type is y_1 and of the second type is y_2 . For our approximation we break down the uncountable state space to a finite, countable one. Let

$$S^* = S \times \left\{ \frac{l_1}{\Delta}, \dots, \frac{u_1}{\Delta} \right\} \times \left\{ \frac{l_2}{\Delta}, \dots, \frac{u_2}{\Delta} \right\} \quad (4.22)$$

be the state space of the new CTMC. A state (s, j_1, j_2) then indicates that the MRM is in state s and has accumulated rewards in the intervals $(j_1\Delta, (j_1 + 1)\Delta]$ and $(j_2\Delta, (j_2 + 1)\Delta]$, respectively (for $j_1 = 0$ or $j_2 = 0$ these intervals are left-closed). In the special case where $c = 1$ ($y_2 = 0$) only the first accumulated reward y_1 has to be discretized.

The initial distribution $\underline{\alpha}^*$ depends on the original initial distribution α and the initial values for the accumulated rewards a_1 and a_2 :

$$\alpha_{(i, j_1, j_2)}^* = \begin{cases} \alpha_i, & a_1 \in (j_1\Delta, (j_1 + 1)\Delta] \text{ and} \\ & a_2 \in (j_2\Delta, (j_2 + 1)\Delta], \\ 0, & \text{otherwise.} \end{cases} \quad (4.23)$$

The distribution of the accumulated rewards is then approximated as

$$F^{(Y_1, Y_2)}(t, y_1, y_2) \approx \sum_{i \in S} \sum_{j_1 = \frac{l_1}{\Delta}}^{\frac{y_1 - 1}{\Delta}} \sum_{j_2 = \frac{l_2}{\Delta}}^{\frac{y_2 - 1}{\Delta}} \pi_{(i, j_1, j_2)}(t), \quad (4.24)$$

where $\pi_{(i, j_1, j_2)}(t)$ is the transient probability of residing in state (i, j_1, j_2) at time t in the derived CTMC.

For battery models, the probability that the battery is already empty at time t , cf. (4.18), is approximated as:

$$\Pr \{ \text{battery empty at time } t \} \approx \sum_{i \in S} \sum_{j_2 = \frac{l_2}{\Delta}}^{\frac{u_2}{\Delta}} \pi_{(i, 0, j_2)}(t). \quad (4.25)$$

Recall that the battery is empty when the available-charge well is empty, that is when $j_1 = 0$.

4.2.2 Transitions in the new generator

In the following we restrict the presentation to the solution of the MRKiBaM. However, the approach is easily applicable to general inhomogeneous MRMs with multiple rewards.

Two types of transitions are possible in the new CTMC with generator \mathbf{Q}^* : transitions taken from the original CTMC and transitions between different reward levels (for each of the two reward types). An entry in the new generator matrix

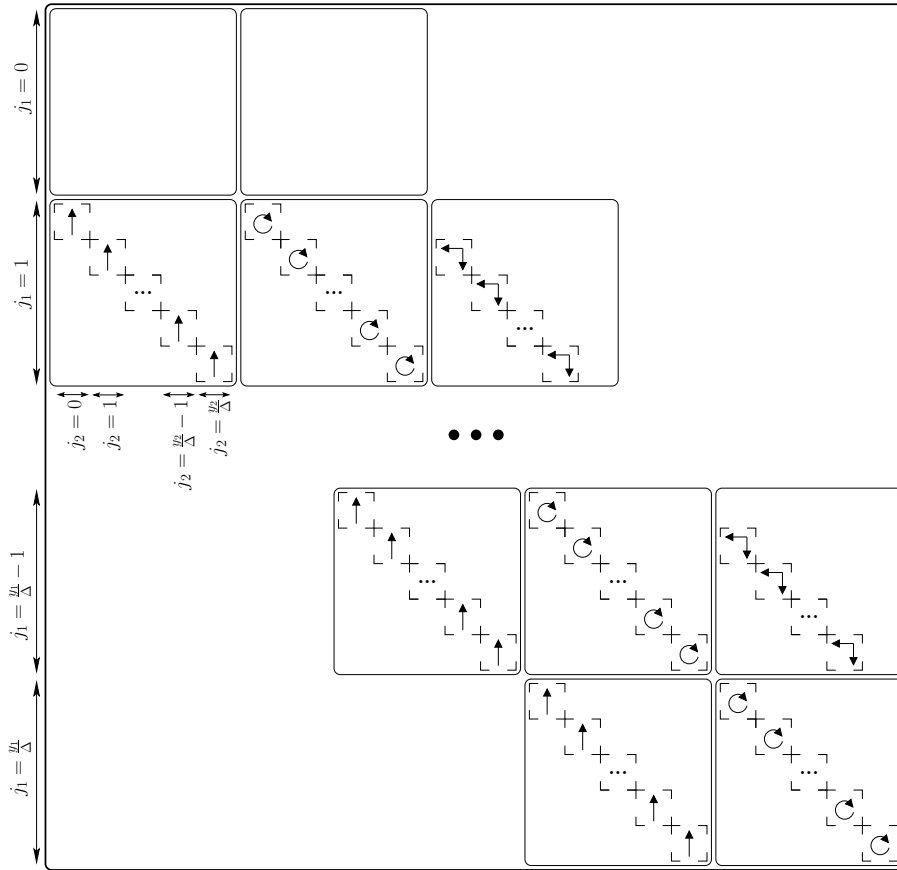


Figure 4.2 Structure of the new generator matrix Q^* .

\mathbf{Q}^* is defined depending on the type of transition it represents. Figure 4.2 shows the structure of the generator matrix \mathbf{Q}^* . Each small block corresponds to a fixed j_1 and j_2 and has dimension $N \times N$, each of the big blocks corresponds to one value of j_1 .

- *Transitions from the original generator.* If the original CTMC part of two states (i, j_1, j_2) and (i', j_1, j_2) are different ($i \neq i'$) but the reward levels are identical, the entry is taken from the original generator. Since it is a reward-inhomogeneous MRM, the current reward level $(j_1\Delta, j_2\Delta)$ must be taken into account, that is,

$$Q_{(i,j_1,j_2),(i',j_1,j_2)}^* = Q_{i,i'}(j_1\Delta, j_2\Delta). \quad (4.26)$$

In Figure 4.2 these entries are found in the blocks $\lceil \mathcal{C} \rceil$.

- *Transitions indicating the consumption of energy.* If the original CTMC states, that is the first component in the states (i, j_1, j_2) , are identical, the levels of the first accumulated reward are different and the levels of the second accumulated reward are again identical, the entry indicates a change in the first accumulated reward, the available charge well. Such a change can only happen between neighboring levels, hence, between j_1 and $j_1 - 1$ (entries in blocks $\lceil \uparrow \rceil$).

$$Q_{(i,j_1,j_2),(i,j_1-1,j_2)}^* = \frac{I_i}{\Delta}, j_1 > 0 \quad (4.27)$$

- *Transitions indicating the transfer from the bound-charge well to the available charge well.* When charge is transferred between the two wells the level of the first reward has to increase while simultaneously the level of the second reward decreases. This corresponds to a transition between state (i, j_1, j_2) and $(i, j_1 + 1, j_2 - 1)$ for $j_1 < u_1/\Delta$, $j_2 > 0$ and $h_2 \geq h_1$:

$$Q_{(i,j_1,j_2),(i,j_1+1,j_2-1)}^* = \frac{k(h_2 - h_1)}{\Delta} = k \left(\frac{j_2}{1-c} - \frac{j_1}{c} \right), \quad (4.28)$$

where $h_1 = (j_1\Delta)/c$ and $h_2 = (j_2\Delta)/(1-c)$. These entries can be found in the blocks $\lceil \uparrow \downarrow \rceil$.

The entries in the first row of big blocks in Figure 4.1 correspond to $j_1 = 0$, which means that the battery is empty. These states are made absorbing, because the lifetime of a battery is defined to be the first time at which it gets empty, so we do not allow recovery in this case. However, the recovery transitions could easily be included. All other off-diagonal entries of \mathbf{Q}^* are zero, the diagonal entries are defined as the negative row sums.

4.2.3 Battery lifetime

The pure CTMC approximation allows us to compute the battery lifetime distribution for a given workload model. This CTMC approximation is, in fact, a phase-type distribution for a given workload model. Its absorbing states are the ones where the battery is perceived empty, that is, where the available charge Y_1 reaches zero. Such state is of the form $(i, 0, j_2)$, where i is an original MRM state, and j_2 is the discretized level of the charge in the bound-charge well.

The generator matrix \mathbf{Q}^* can be arranged in such a way that

$$\mathbf{Q}^* = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{T}^0 & \mathbf{T} \end{pmatrix}, \quad (4.29)$$

where \mathbf{T} contains the rates of transitions between non-absorbing states, and \mathbf{T}^0 is the matrix with the rates from each non-absorbing state to the absorbing states, which indicate that the battery is empty. If we merge all absorbing states into one, the generator matrix reduces to:

$$\mathbf{Q}^* = \begin{pmatrix} 0 & \underline{\mathbf{0}} \\ \underline{\mathbf{T}}^0 & \mathbf{T} \end{pmatrix}, \quad (4.30)$$

where $\underline{\mathbf{T}}^0$ is a column vector with the cumulative rates to the absorbing state. The represented phase-type distribution is an approximate distribution for the random variable describing the time it takes for the battery to be emptied. This distribution can efficiently be computed using uniformization [27].

The expected value of a random variable L having a phase-type distribution is described by [50]:

$$\mathbb{E}[L] = -\underline{\alpha} \mathbf{T}^{-1} \underline{\mathbf{1}}, \quad (4.31)$$

where $\underline{\mathbf{1}}$ is a column vector of appropriate size with each element equal to one. Thus, if we solve the system of linear equations

$$\underline{\mathbf{x}} \mathbf{T} = -\underline{\alpha}, \quad (4.32)$$

we have

$$\mathbb{E}[L] = \sum_i x_i. \quad (4.33)$$

Using this approach we can approximate the expected battery lifetime for a given workload.

4.2.4 Delivered charge

The amount of charge that is actually delivered by the battery depends on the workload. When we look at the possible absorbing states of the approximating CTMC, we see that if the CTMC ends up in a state $s_a = (i, 0, j_2)$ it means that the

delivered charge is approximately $C - j_2\Delta$. We can thus compute approximations to the distribution and expected value of the delivered charge.

Since for these computations the time until absorption is not important, it suffices to consider the embedded discrete-time Markov chain with probability matrix \mathbf{P}^* , where

$$P_{s,s'}^* = \begin{cases} 1, & \text{if } s = s' \text{ and } Q_{s,s}^* = 0, \\ \frac{Q_{s,s'}}{Q_{s,s}^*} & \text{if } s \neq s' \text{ and } Q_{s,s}^* \neq 0, \\ 0, & \text{elsewhere.} \end{cases} \quad (4.34)$$

Following the notation introduced for phase-type distributions we can arrange \mathbf{P}^* such that

$$\mathbf{P}^* = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{R}^0 & \mathbf{R} \end{pmatrix}. \quad (4.35)$$

The probability A_{s,s_a} to end in a certain absorbing state s_a , having started in state s is determined by the following system of linear equations:

$$A_{s,s_a} = \begin{cases} 1, & \text{if } s = s_a, \\ \sum_z P_{s,z}^* A_{z,s_a}, & \text{otherwise.} \end{cases} \quad (4.36)$$

If \mathbf{B} is the matrix consisting of the A_{s,s_a} where s is a transient state, the system of linear equations can be written as:

$$\mathbf{R}\mathbf{B} + \mathbf{R}^0\mathbf{I} = \mathbf{B} \quad \text{or} \quad (\mathbf{I} - \mathbf{R})\mathbf{B} = \mathbf{R}^0. \quad (4.37)$$

This system can be solved for one column of \mathbf{R}^0 at a time using standard solution algorithms. The complete matrix \mathbf{A} is obtained by extending the above computed matrix \mathbf{B} to include also the absorbing states as initial states:

$$\mathbf{A} = \begin{pmatrix} \mathbf{I} \\ \mathbf{B} \end{pmatrix}. \quad (4.38)$$

Multiplying the initial distribution vector $\underline{\alpha}$ with \mathbf{A} gives the probability distribution \underline{a} to end up in the different absorbing states,

$$\underline{a} = \underline{\alpha}\mathbf{A}. \quad (4.39)$$

One element $a_{(i,0,j_2)}$ denotes the probability that the battery gets empty with a residual charge of $j_2\Delta$, and thus having delivered a charge of $C - j_2\Delta$. In this way one can obtain the distribution of the delivered charge. The expected delivered charge is given by:

$$\mathbb{E}[dC] = C - \sum_{\substack{(i,0,j_2) \\ \text{is absorbing}}} j_2\Delta a_{(i,0,j_2)}. \quad (4.40)$$

space	time	
	lifetime distribution	charge distribution expected lifetime and charge
$\mathcal{O}\left(N^2 \cdot \frac{y_1}{\Delta} \cdot \frac{y_2}{\Delta}\right)$	$\mathcal{O}\left(N^2 \cdot qt \cdot \frac{y_1}{\Delta} \cdot \frac{y_2}{\Delta}\right)$	$\mathcal{O}\left(N^2 \cdot \frac{y_1}{\Delta} \cdot \frac{y_2}{\Delta}\right)$

Table 4.1 Complexities for the computation of the distribution and expected value of the battery lifetime and the delivered charge.

4.2.5 Complexity

Table 4.1 shows the time complexity of the algorithms used to compute the distribution and expected value of the battery lifetime and delivered charge. The space complexity is determined by the size of the generator matrix, which is the same for all algorithms. We only need to store the nonzero elements. Figure 4.2 shows that only the small blocks contain nonzero elements. These blocks have the size of the generator matrix of the original workload. Since this matrix may be dense, the space complexity is quadratic in the number of states of the workload (N). The big blocks contain $\frac{y_2}{\Delta} \times \frac{y_2}{\Delta}$ small blocks and the total matrix contains $\frac{y_1}{\Delta} \times \frac{y_1}{\Delta}$ of big blocks. Since only three diagonals of the big blocks, and per big block only one diagonal of small blocks, contain nonzero elements, the space complexity is linear in $\frac{y_1}{\Delta}$ and $\frac{y_2}{\Delta}$.

Regarding the run time of the uniformization algorithm used to compute the lifetime distribution, the algorithm is linear in the number of nonzero elements of the generator matrix [61]. For the MRKiBaM, the step size Δ enters as Δ^{-2} . However, the step size is also coded into the generator matrix of the new CTMC by multiplying the reward rates with $\frac{1}{\Delta}$ (see the definition of \mathbf{Q}^*). The transient solution of the new CTMC has a time complexity linear in the *uniformisation constant* q and the time t for which the transient probabilities are computed. For small Δ , this uniformization constant becomes linear in $\frac{1}{\Delta}$, hence, we obtain an overall time complexity in Δ^{-3} .

The algorithms for computing the expected lifetime and the distribution and expected value of the delivered charge have the same complexity. In all these algorithms a set of linear equations needs to be solved. The time complexity for this is at most cubic in the number of states in the generator matrix, when an algorithm like Gaussian elimination is used [61]. However, this does not take into account the sparse structure of the generator matrix. When we do take this into account, we see that the complexity is improved to $\mathcal{O}\left(N^2 \cdot \frac{y_1}{\Delta} \cdot \frac{y_2}{\Delta}\right)$, which is only quadratic in $\frac{1}{\Delta}$.

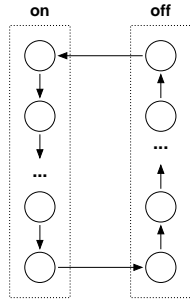


Figure 4.3 Simple on/off model (SWL1).

4.3 Stochastic workload models

In the following we consider three stochastic workload models.

Stochastic workload 1 (SWL1). First we concentrate on simple on/off models like the ones used in [57] with the only difference that those were not stochastic. For a given frequency f , the workload toggles between the off-state ($I = 0$ A) and the on-state ($I = 0.96$ A). We model the on/off times as Erlang- K distributions such that with increasing K they become close to deterministic.

Figure 4.3 shows the state-transition diagram for this simple model. For frequency f , all transitions have rate

$$\lambda = 2 \cdot f \cdot K.$$

The expected on and off times, respectively, are then $K/(2fK) = 1/(2f)$ which leads exactly to a frequency f .

We furthermore consider two workload models of a small battery-powered device.

Stochastic workload 2 (SWL2). The first, simple one consists of three states as depicted in Figure 4.4. Initially, the model is in **idle** state. With rate $\lambda = 2$ per hour there is the necessity to send data over the wireless interface. If such data is present, the model moves into the **send** state. The sending of data is complete in 10 minutes on average (resulting in a sending rate of $\mu = 6$ per hour). From the idle state the device can also move into a power-saving **sleep** state, this is done – on average – once per hour ($\tau = 1$). The power-consumption rate is low when idling ($I_0 = 8$ mA), it is high when sending data ($I_1 = 200$ mA) and negligible in the sleep state ($I_2 = 0$ mA).

Stochastic workload 3 (SWL3). To extend the overall battery lifetime it seems to be beneficial to have short periods of high sending activity (bursts) and long periods without sending activity, since the longer idle periods will give the battery more time to recover from the high loads while sending.

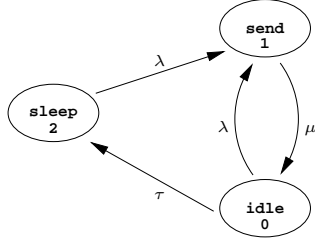


Figure 4.4 State transition diagram for the simple model (SWL2).

label	rate (h ⁻¹)
λ	2
μ	6
τ	1
<code>switch_on</code>	1
<code>switch_off</code>	6
λ_{burst}	182

Table 4.2 Transition rates of the simple and burst model.

In the modeled wireless device this could be achieved by accumulating the data to be transmitted and then send all in direct sequence instead of transmitting smaller amounts of data more frequently. This can be modeled by buffering the flow of arriving data. When the flow is active, data arrives with a very high rate. If the flow is inactive, the device can safely go to sleep. Figure 4.5 shows a state-transition diagram for such a burst model. It has the same sending rate μ and timeout rate τ as the simple model. Bursts start with rate `switch_on`=1 per hour and stop with rate `switch_off`=6 per hour. To make any results of the latter two models comparable, we have chosen $\lambda_{\text{burst}} = 182$ per hour such that the steady-state probability to be in `off - send` or `on - send` in the burst model is the same as the probability to be in `send` in the simple model. As could be expected, the steady-state probability to be in `sleep` is higher in the burst model than in the simple model.

4.4 Results

In this section we evaluate the distribution and expected value of the battery lifetime and delivered charge for the systems described in Section 4.3. We validate our Markovian approximation algorithm by comparison with detailed simulations

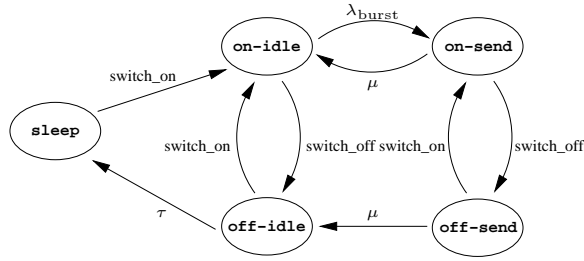


Figure 4.5 State transition diagram for the burst model (SWL3).

Frequency (Hz)	Experimental lifetime [57] (min)	KiBaM lifetime (min)	Modified KiBaM lifetime	
			stochastic [57] (min)	numerical (min)
Continuous	90	91	90	89
1	193	203	193	193
0.2	230	203	226	193

Table 4.3 Experimental and computed battery lifetimes. The computations are done with the KiBaM and the modified KiBaM [57].

using the analytical KiBaM.

4.4.1 Modeled battery

The battery we model in this section is a Ni-MH battery, which was also modeled by Rao *et al.* in [57]. The battery has a capacity of 7200 As. The battery parameters c and k are computed from the experimental data given in [57]. The parameter c can be calculated from the capacity delivered under very large and very small loads. At very large loads the battery lifetime is short, and there is no time for the charge to move from the bound-charge well to the available-charge well. The capacity delivered equals the amount of charge in the available-charge well. At very small loads, however, all the charge from both the bound and available-charge well is delivered. The quotient of these two numbers is exactly c ; from [57] we take $c = 0.625$. We set the parameter k in such a way that the calculated lifetime for a continuous load of 0.96 A corresponds to the experimental value given in [57]. This results in $k = 4.5 \cdot 10^{-5} \text{ s}^{-1}$.

In Table 4.3, second and third column 2, we present the battery lifetimes according to the KiBaM and some experimental results given in [57]. We see that for KiBaM the lifetime is constant for both frequencies. However, the experimental results show a longer lifetime for the slower frequency. To overcome this problem, Rao *et al.* have developed a modified KiBaM [57]. In the modified model the recovery rate has an additional dependence on the height of the bound-charge well,

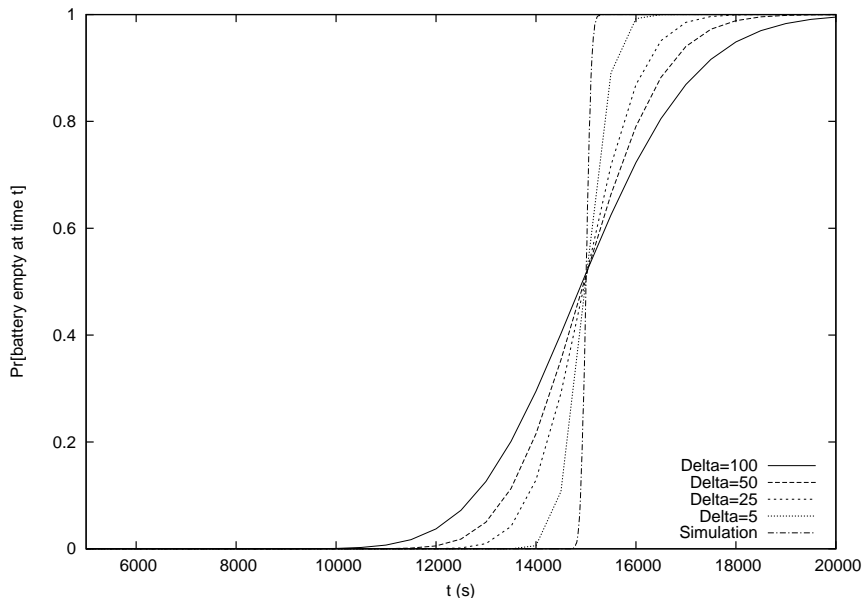


Figure 4.6 Battery lifetime distribution for the on/off-model ($f = 1$ Hz, $K = 1$, $C = 7200$ As, $c = 1$, $k = 0$ s $^{-1}$).

making the recovery slower when less charge is left in the battery. With a stochastic simulation of this model they obtain very good results for the battery lifetimes, see the fourth column in Table 4.3. However, we numerically evaluated the modified KiBaM with a deterministic workload and saw that the lifetime still does not depend on the frequency, as indicated in the last column in Table 4.3. Personal correspondence with the authors of [57] did not shed light on the discrepancy.

4.4.2 On/Off model (SWL1)

We start with a degenerate case of the KiBaM, where the bound-charge well is empty from the beginning and the complete charge is in the available-charge well. There is no transfer of charge between the two wells. We choose the simplest Erlang model (see Figure 4.3) for frequency $f = 1$ Hz with $K = 1$, that is, on- and off-times follow a negative exponential distribution with rate $\lambda = 2$ s $^{-1}$. The battery capacity is $C = 2000$ mAh = 7200 As; the KiBaM constants are $c = 1$ and $k = 0$ s $^{-1}$.

Figure 4.6 shows the resulting lifetime distribution calculated by simulation and using the approximation algorithm using different step-sizes Δ .

The simulation results are obtained by 10000 independent runs. They suggest that the battery lifetime is close to deterministic with a mean of about 15000 sec-

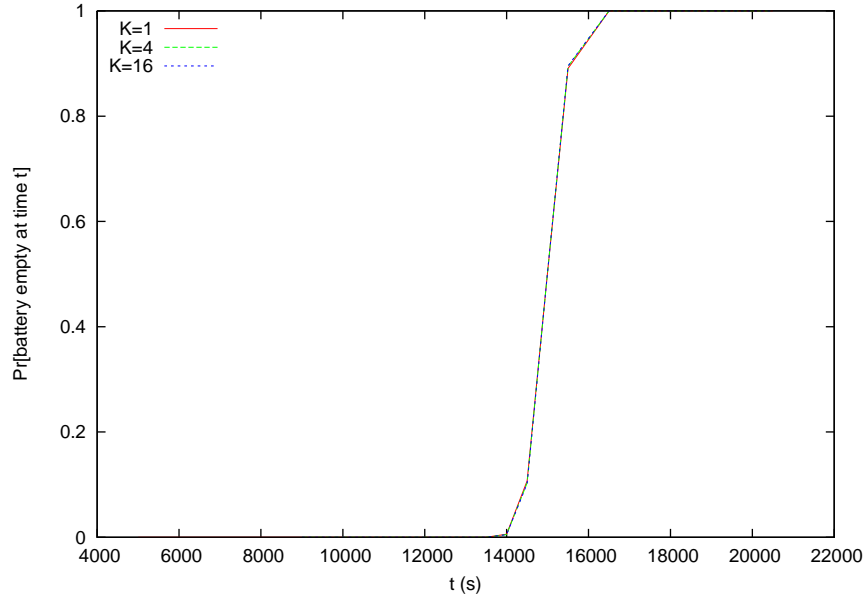


Figure 4.7 Battery lifetime distribution for the on/off-model for varying values of K ($f = 1$ Hz, $C = 7200$ As, $c = 1$, $k = 0$ s $^{-1}$, $\Delta = 5$ As).

onds. This is reasonable since the overall time spent in the on-state in one of the runs has approximately an Erlang $_{15000}(2$ s $^{-1})$ distribution, which is a good approximation of a deterministic distribution with mean 7500 seconds. In 7500 seconds the consumed energy is 7500 s \cdot 0.96 A = 7200 As = C . For pure deterministic on- and off-times, the analytical KiBaM also yields a lifetime of 15000 seconds.

For decreasing step-size Δ the curves from the approximation algorithm approach the simulation curve. This is an indication for the correct operation of the algorithm. However, even for $\Delta = 5$ As, the approximation is not really a good one, since it is in general difficult to closely approximate an almost deterministic value through a phase-type distribution.

We also evaluated the battery lifetime of the on/off-model for better approximations to the deterministic on- and off-times, that is, for $K > 1$ in the Erlang model. The results are given in Figure 4.7. The three curves for $K = 1$, $K = 4$ and $K = 16$ completely overlap. The added phases in the on/off model do not yield any significant improvement. This is due to the fact that the approximation is not good enough to capture the relatively small differences. Therefore, all the remaining results for the on/off model are obtained only for $K = 1$.

Figure 4.8 shows the lifetime distribution of the on/off model for $c = 0.625$, that is, initially 62.5% of the charge is in the available-charge well and 37.5% is in the bound-charge well. The constant for the flow between the two wells is

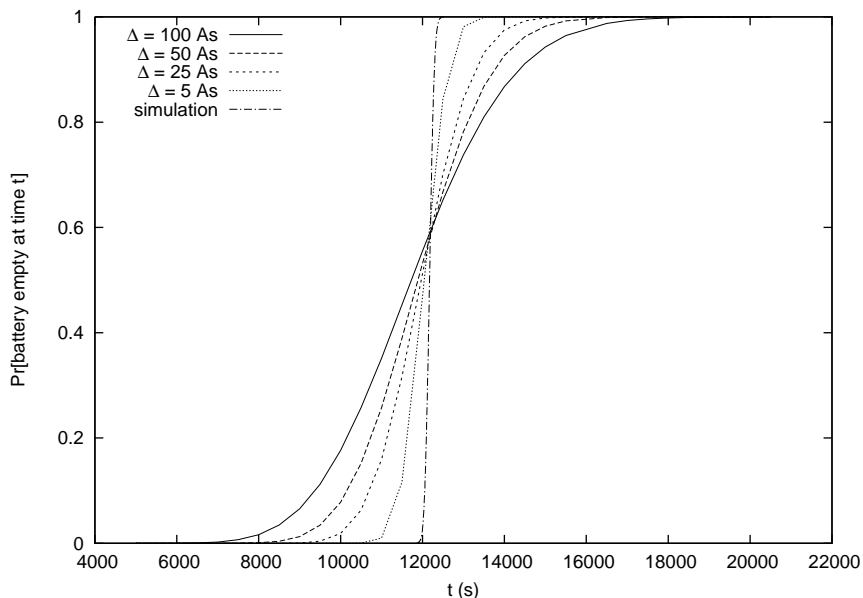


Figure 4.8 Battery lifetime distribution for the on/off model ($f = 1$ Hz, $K = 1$, $C = 7200$ As, $c = 0.625$, $k = 4.5 \cdot 10^{-5} \text{ s}^{-1}$).

$k = 4.5 \cdot 10^{-5} \text{ s}^{-1}$.

The curves for the approximation algorithm are quite far away from the one obtained by simulation. Unfortunately it is not feasible to consider a substantially smaller Δ for this example. The CTMC for $\Delta = 5$ As has about $9.7 \cdot 10^5$ states and the generator matrix Q^* has about $1.7 \cdot 10^6$ non-zero entries. For $t = 10000$ s, uniformization requires more than $2.3 \cdot 10^4$ iterations, each with $1.7 \cdot 10^6$ multiplications. For $t = 20000$ s, more than $4.6 \cdot 10^4$ iterations are needed.

Although the distribution is hard to approximate, the expected battery lifetime can be well approximated by the MRKiBaM. The results of the computations of the expected lifetime for different values of Δ are given in Table 4.4. The simulations result in an expected lifetime of 12175 s, whereas the expected lifetime computed with the MRKiBaM for $\Delta = 5$ As is 12040 s. This is a difference of only 1.1%. Even for $\Delta = 100$ As the difference is only 3.0%, with an expected value of 11815 s for the battery lifetime.

In Figure 4.9 we compare the lifetime distribution of the two cases already described with a third scenario, where the initial capacity of the battery is only $4500 \text{ As} = 0.625 \cdot 7200 \text{ As}$ and completely in the available-charge well. In the first case ($C = 7200$ As, $c = 1$) the battery lasts generally longer than in the second case ($C = 7200$ As, $c = 0.625$), because all charge is available. In the third case ($C = 4500$ As, $c = 1$), the battery lifetime is in general shorter, because there is

Δ (As)	100	50	25	5
$\mathbb{E}[L]$ s	11815	11930	11990	12040
error (%)	3.0	2.0	1.5	1.1

Table 4.4 The expected lifetime for the on/off-model according to the MRKiBaM for different values of Δ and the error compared to the simulation result.

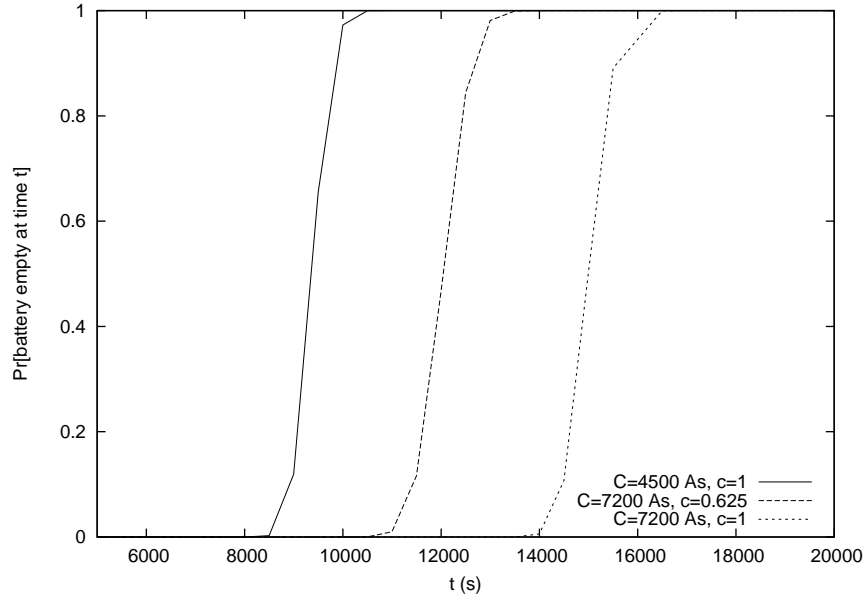


Figure 4.9 On/off model with different initial capacities ($\Delta = 5$ As).

no bound-charge to be transferred to the available-charge well.

Figure 4.10 shows the distribution of the delivered charge of the on/off model with $K = 1$ and $c = 0.625$. For the delivered charge distribution the case with $c = 1$ is not interesting, since in this case always all the charge will be drawn from the battery. The curves computed with the Markovian approximation approach are step functions, since in the approach we discretize the charge levels of the available and bound charge wells. The simulation results are, again, based on 10000 independent runs. Like with the battery lifetime, the simulations show that the delivered charge is close to deterministic. The mean of the delivered charge of the simulations is 5845 As, with a standard deviation of 7.9 As. We see, again, that the Markovian approximation does not approach the simulations well. Even for $\Delta = 5$ As the approximation is not good.

Like the expectation of the battery lifetime, the expectation of the delivered charge computed with the Markovian approximation does approach the simulation

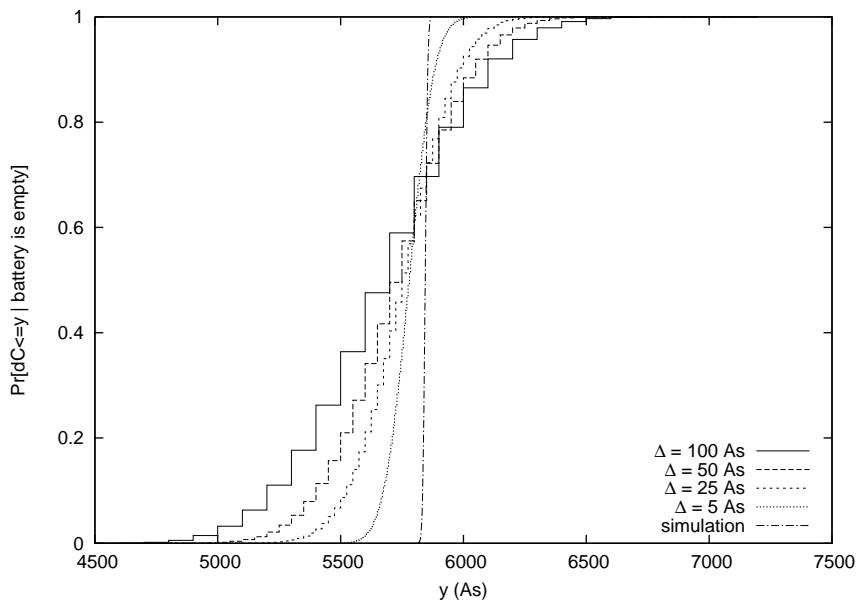


Figure 4.10 Distribution of the delivered charge for the on/off-model ($f = 1$ Hz, $K = 1$, $C = 7200$ As, $c = 0.625$, $k = 4.5 \cdot 10^{-5} \text{ s}^{-1}$).

Δ (As)	100	50	25	5
$\mathbb{E}[dC]$ As	5672	5726	5755	5779
error (%)	3.0	2.0	1.5	1.1

Table 4.5 The expected delivered charge for the on/off-model according to the MRKiBaM for different values of Δ and the error compared to the simulation result.

results very well. The results are given in Table 4.5. For $\Delta = 5$ As the expected delivered charge is 5779 As, which is only 1.1% less than the mean value of the simulations, which is 5845 As.

4.4.3 Simple & burst model (SWL2 & SWL3)

We now evaluate and compare the results for the simple and the burst model. For the computations of the simple and burst model the battery capacity has been set to 800 mAh. This is the capacity of batteries that are often used in mobile phones. The other battery parameters, c and k are kept the same.

For the simple model, it is possible to compute good approximations using the Markovian approximation algorithm. Figure 4.11 shows the lifetime distributions computed for different values of Δ . We see that for decreasing Δ the distributions

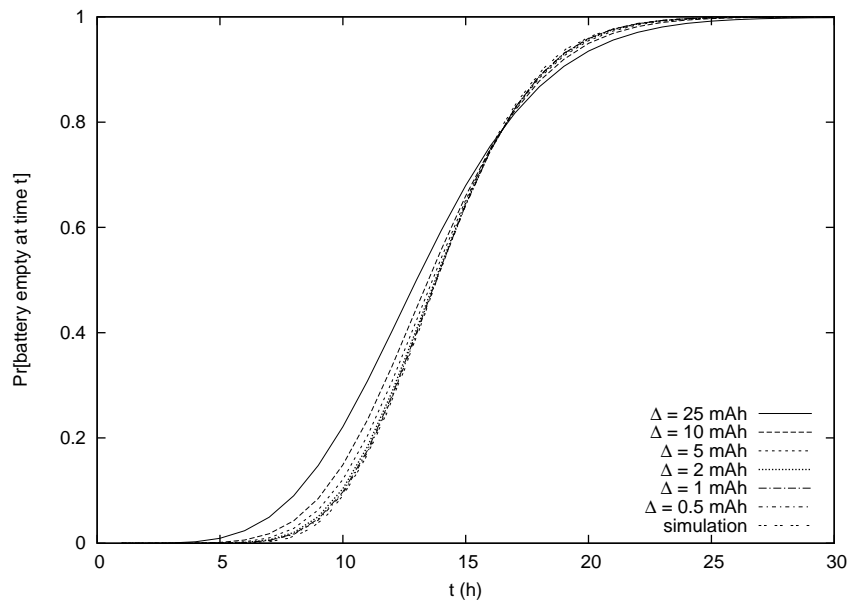


Figure 4.11 Battery lifetime distribution for the simple model.

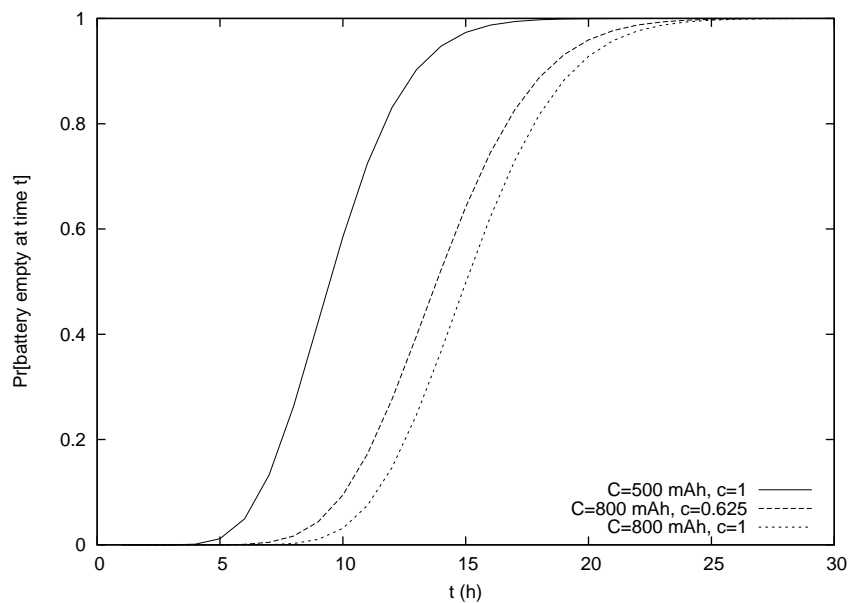


Figure 4.12 Battery lifetime distribution for the simple model with different initial capacities ($\Delta = 0.5$ mAh).

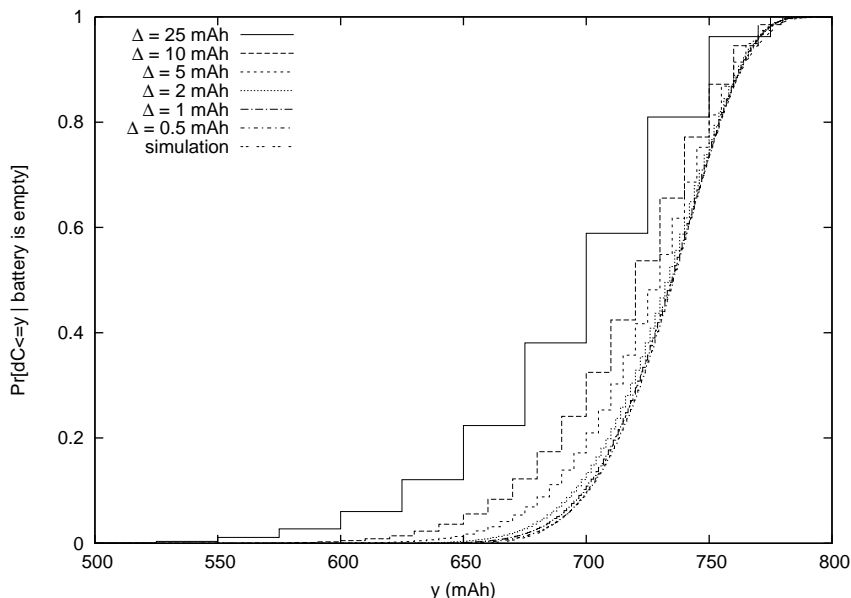


Figure 4.13 Distribution of the delivered charge for the simple model.

converge towards the distribution obtained by simulation. For $\Delta = 1$ mAh a good approximation is obtained.

The difference in accuracy with the on/off model could be caused by the difference in the rates of the workload change. In the simple model for $\Delta = 1$ mAh the change of workload state is much slower than the change in the available and bound charge. In contrast, even for the smallest discretization step of $\Delta = 5$ As, in the on/off model the workload change is faster than the change in available and bound charge. The change in available and bound charge could be made faster by a further decrease of Δ . However, this leads to a state space which can not be analyzed anymore.

In Figure 4.12 the effect of using the KiBaM model, instead of an ideal battery, is shown. The figure compares the battery lifetime distribution according to the KiBaM with two distributions using an ideal battery. In the first case the ideal battery has a capacity limited to the available charge of the KiBaM, *i.e.*, 500 mAh. In the second case the ideal battery has a capacity that also includes the bound charge, *i.e.*, 800 mAh. From Figure 4.12 one can see that if only 62.5% of the capacity becomes available at all (leftmost curve) the battery is almost certainly empty (with probability greater than 99%) after about 17 hours. If the rest of the charge is initially in the bound-charge well, the battery gets almost surely empty after about 23 hours, if all capacity is readily available (rightmost curve), after about 25 hours. Hence, for this workload model it is in general not possible to

make use of the total capacity of 800 mAh, if it is distributed between the bound-charge well and the available-charge well. However, a large fraction of the total capacity becomes available, which is shown by the fact that the middle curve is closer to the right curve than to the left curve.

In Figure 4.13 the distribution of the delivered charge for the simple model is given for different values of Δ and simulations. The distributions obtained with Markovian approximation approach the simulation from the left. With the coarse discretization, $\Delta = 25$ mAh, the delivered charge is severely underestimated. For $\Delta = 1$ mAh a good approximation is obtained.

In Figure 4.14 we finally compare the results of the simple and the burst model. The burst model condenses the send activity and consequently spends more time in sleep mode. This lets the battery last longer, that is, its lifetime distribution curve lies for the majority to the right of the one for the simple model. For example, after 20 hours the battery is empty with a probability of about 95% when using the simple model while it is empty with probability only about 89% in case of the burst model. The expected lifetime of the burst model is 14.46 h. This is 2.8% longer than the expected lifetime of the simple model, which is 14.06 h.

However, Figure 4.15 shows that with the burst model the battery delivers less charge than with the simple model, the distribution of the burst model lies completely to the left of the distribution of the simple model. The expected

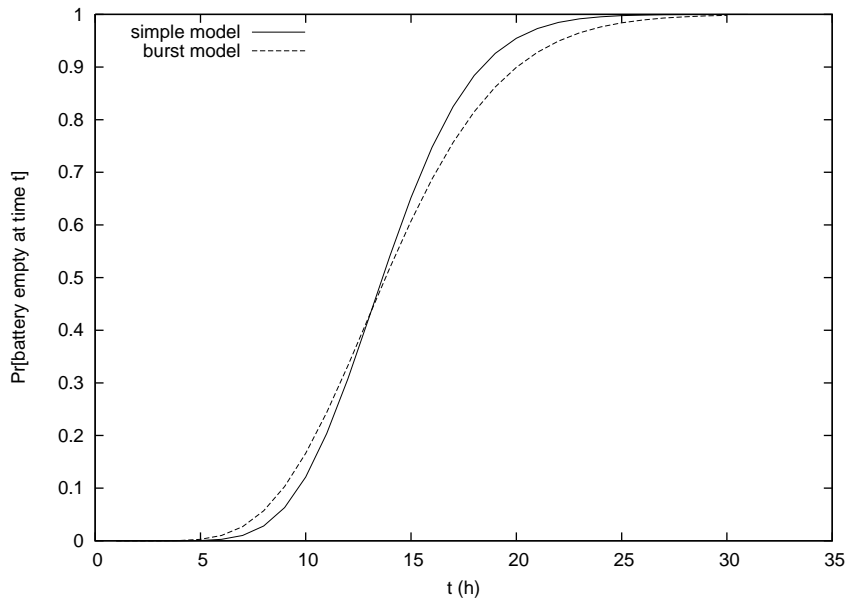


Figure 4.14 Battery lifetime distribution for the simple and the burst model ($C = 800$ mAh, $c = 0.625$, $\Delta = 5$ mAh).

delivered charge in the burst model is 0.9% lower than the expected delivered charge in the simple model, 733.7 mAh and 740.3 mAh respectively. This is highly counterintuitive. One would expect the battery to deliver more charge in the burst model, while the burst model leads to longer battery lifetimes.

These results imply that the gain of more idle time in the burst model, and therefore a lower average discharge current, is not fully transferred into an extension of the battery lifetime. Although in the burst model the battery has more time to recover, the bursts of on periods result in less charge to be drawn from the battery.

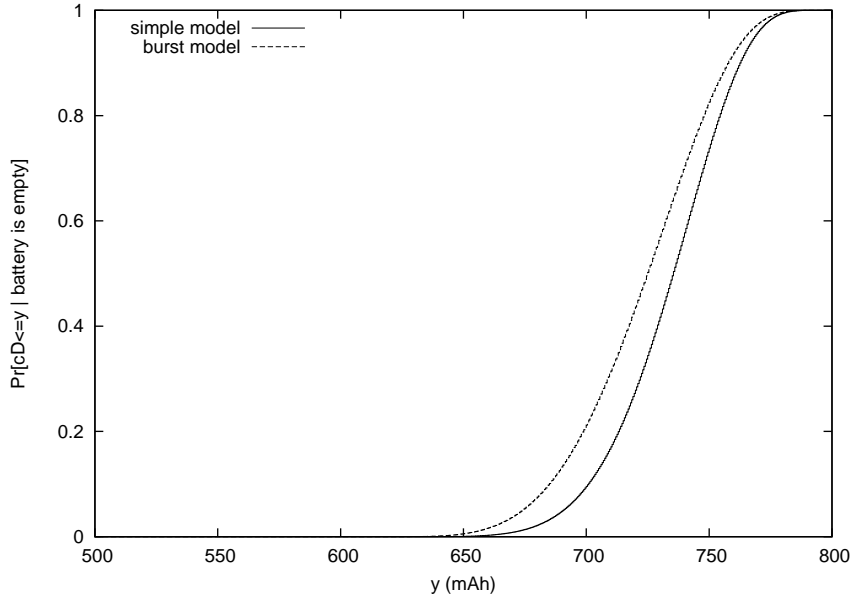


Figure 4.15 Distribution of the delivered charge for the simple and the burst model ($C = 800$ mAh, $c = 0.625$, $\Delta = 0.5$ mAh).

4.5 Conclusion

In this chapter we combined the analytical KiBaM with a Markov model, that represents the stochastic workload, into the MRKiBaM, a reward-inhomogeneous Markov reward model for batteries. With this model we can assess the distribution and expected value of both the battery lifetime and the charge delivered by the battery for a stochastic workload model. For the actual computation we provide an efficient approximation algorithm where the accumulated rewards are discretized. The computation of the battery lifetime distribution then boils down to the transient solution of a CTMC.

With an implementation of this algorithm we have evaluated the lifetime of some small workload models, thereby also comparing to simulation results. Trying to approximate the almost deterministic lifetime for the on/off model results in a poor accuracy. In contrast, for the simple and burst mode of a wireless device, the algorithm gives good results. Also, the computations of the distribution of the delivered charge result have higher accuracy for the simple and burst workload models than for the on/off model. However, the computations of the expected value of the battery lifetime and the delivered charge give good results even for the on/off model.

In the comparison between the simple and the burst model we see the importance of including the battery properties into the model for predicting the battery lifetimes. The burst model with its longer idle time and lower average current did, indeed, give an improvement in battery lifetime. However, the delivered charge for the burst model was lower than for the simple model. This means the lifetime improvement was not as long as one would expect when a fixed battery capacity was assumed. The reason behind this remains not completely understood.

Chapter 5

Computing best battery schedules using priced timed automata

Some mobile devices, for example laptop computers, have the option to connect a second battery in order to improve the system lifetime. Most of these systems, however, use the batteries in sequential order, that is, the second one is only used when the first one is empty. Whereas this is clearly an effective way to prolong the systems lifetime, it is not the most efficient one. By regularly switching between the batteries one will give both batteries more time to recover, and in this way possibly extend the lifetime even more than twice the lifetime of a single battery.

In this chapter we give an approach to find the best battery schedule by using priced timed automata (PTA). Section 5.1 introduces the idea behind battery scheduling, and gives a short overview of the related work performed in this area. In Section 5.2, we give an introduction to PTA. Section 5.3 shows how the KiBaM can be modeled using PTA. In Section 5.4, this model is validated to the original KiBaM. In Section 5.5, the PTA model is used to compute the battery schedule that results in the longest lifetime for a set of test loads. We compare these results with the lifetimes obtained by some simple scheduling strategies. In Section 5.6 the system lifetime is computed for randomly generated loads. We end this chapter with some conclusions in Section 5.7.

5.1 Battery scheduling

Influencing the usage pattern of a battery is hard in a single battery system. However, some devices allow the connection of multiple batteries. In such systems one

can, instead of using the two batteries sequentially, switch between the batteries. By applying battery scheduling, one can then easily influence the usage pattern of the batteries.

Battery scheduling may also be beneficial in sensor networks. Although each sensor, in general, is powered by only one battery, the entire network is powered by many. Often there are several routes from a sensor node to the data sink to send the collected data through the network. To keep all the sensors powered as long as possible, battery-aware routing has to be done, *i.e.*, the decision on which sensor has to forward the data has to be based on the status of the sensor's batteries. Switching from one route to the other will give the batteries time to recover and thus give a longer lifetime to the sensor network as a whole. In this way, the routing problem is turned into a battery scheduling problem.

Already, scheduling of batteries has attracted quite some attention in the literature. The most important scheduling schemes that are studied are:

- Sequential scheduling: another battery is only picked when the previous one is empty.
- Round robin scheduling: at fixed moments in time another battery is used. The batteries are used in a fixed order.
- Pick-best scheduling: at fixed moments in time the status of all batteries is checked and the best battery is used. What is the best battery can be determined in several ways, for example the battery with the highest voltage, or the battery that has been used for the shortest period of time.

We consider here the main approaches.

In [15], Chiasserini and Rao use a discrete-time Markov battery model, cf. Section 2.5.1, to compare three different scheduling schemes in a multiple battery system. The complexity of the used model limits the battery capacity to very small batteries. The three schedulers that are considered are the round robin, best-of-two scheduler and a random scheduler. The schedulers are compared for different job arrival rates. The results show that the best-of-two scheduler outperforms the other two. However, the model does not allow for any optimization with respect to battery schedule.

Also in [7], the analysis is limited to a set of simple schedulers. Benini et al. [7] consider sequential scheduling, round robin scheduling and various types of best-of-two scheduling, where either the output voltage or the time that a battery has been unused determines which battery is to be scheduled. The different scheduling schemes are applied to several battery configurations containing up to four batteries. The loads that have been used are simple continuous and intermittent loads and two real-life example load profiles. Which scheduler performs best depends on the applied load. However, it is shown that for round robin scheduling the system lifetime increases when the switching frequency is increased.

A completely different battery scheduling approach is taken in [70]. Instead of using two identical batteries, Wu et al. use two batteries with different discharge characteristics. The first battery has a high capacity and performs well at low discharge currents, whereas the second has a lower capacity but performs better at high discharge currents. The scheduling decision is taken with respect to the level of the discharge current. In this way both batteries are used only for currents where they perform the best. It is shown, both analytically and by simulations, that this way of scheduling may lead to a 30% lifetime improvement compared to using only one type of battery.

In all this work the battery scheduling is limited to *deterministic* scheduling schemes. All show that battery scheduling gives longer system lifetime than when the batteries are used sequentially. However, the improvement varies a lot between the different modeling approaches. Where Benini et al. predict an average improvement of approximately 11% for a two battery system, Chiasserini shows improvements of more than 100%. Also, they do not indicate whether longer lifetime could be possible by using even smarter scheduling. In [59], Sarkar and Adamou propose an algorithm for computing an optimal scheduling scheme based on the stochastic battery model of Chiasserini and Rao. To do this they translate the problem to a stochastic shortest path problem. The optimal solution can only be computed for small batteries. However, they do show that for the modeled pulsed load best-of-two scheduling performs close to optimal.

In this and the next chapter we take two new approaches to finding the optimal battery schedule. The first is by using PTA, which are proven to be well fitted for finding optimal schedules [6]. In this approach the scheduling moments and workload are set in advance and the PTA model is used to compute which schedule will lead to the longest lifetime. In Chapter 6 an analytic approach to the scheduling problem is taken. In this approach we loosen the restriction on the moments the scheduling decision has to be taken; at any point in time the scheduler may switch the battery that is used.

5.2 Priced timed automata

In this section we informally describe *networks of timed automata* (NTA), used as input to Uppaal Cora [65]. Uppaal Cora is a branch of Uppaal [66] for Cost Optimal Reachability Analysis. As we only use the PTA as a modeling tool, we introduce the important ingredients by means of an example. A extensive formal description of priced timed automata can be found in [5].

5.2.1 Networks of timed automata

The basic ingredients of NTA are *locations*, *switches*, *clocks*, *guards*, *invariants*, *assignments*, and *channels*. An NTA is composed of a collection of timed automata,

which run in parallel and communicate via the channels.

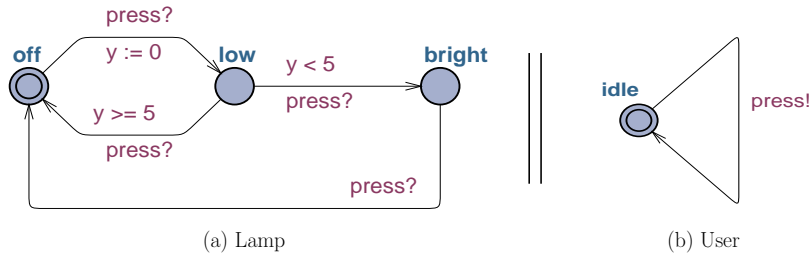


Figure 5.1 Model of lamp.

Figure 5.1 (example taken from [4]) depicts a very simple NTA, comprising only two components. The network models a lamp, which can be turned on and off by a user pressing a switch. In Figure 5.1(a) the behavior of the lamp is modeled. We see three locations *off*, *low*, and *bright*, which denote the three states the lamp can be in: in the location *off* the lamp is off, in location *low*, it is on, giving a low light, and in location *bright*, it shines brightly. Location *off* is the starting location. The switch from *off* to *low* (abbreviated *off* → *low*) is labeled with a receive operation (signified with the question mark) on channel *press*. This switch can only be taken if the timed automaton in Figure 5.1(b) (a model of the user) executes the switch labeled with a send operation (signified with the exclamation mark) on channel *press*, *i.e.*, both timed automata synchronize on channel *press*.

While *off* → *low* is executed, clock variable *y* is reset to 0 by the assignment $y := 0$. Clocks are real-valued variables which are used to measure time: clock values increase linearly with rate 1 as time progresses. Clocks are used to express enabling- and urgency-conditions depending on time. For example, the switch *low* → *off* is labeled with a *guard* $y \geq 5$, which allows this switch to be taken only if clock *y* has a value greater or equal to 5. In that case, if the user presses the button again, the lamp goes off. The switch *low* → *bright*, on the other hand, is guarded by expression $y < 5$, *i.e.*, the negation of the previous guard. Thus, if the user presses the button a second time *within* 5 time units, the lamp switches to brighter light. From location *bright*, another button press will switch the lamp off again, unconditionally.

Invariants are used to express urgency conditions, *i.e.*, unlike guards, which express when something *can* happen, they express when something *must* happen, without delay. In Figure 5.2 we see a slightly modified version of the lamp automaton. The lamp is now switching off automatically, without user intervention. Switches *low* → *off* or *bright* → *off* are thus now unsynchronized. Instead, locations *low* and *bright* are now labeled with the invariant $y \leq 10$ each, which expresses that both locations can only be entered and occupied as long as $y \leq 10$. This means that the respective locations *must* be left 10 time units after the lamp was

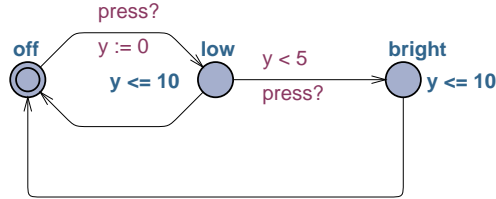


Figure 5.2 Model of automatic lamp.

switched on *at the latest*, which in both cases means that the lamp goes either via $\text{low} \rightarrow \text{off}$ or $\text{bright} \rightarrow \text{off}$ to location off .

Using a synchronizing channel to model the communication for the automatic lamp is unrealistic in the sense that, if the lamp is in location bright , then the user can not press the button (because the send operation press! has no corresponding receive operation). A channel may be declared a *broadcast channel*, which means that a send operation has to be synchronized with those processes only which are ready to execute a receive operation on that channel. If no process is ready to do so, the send operation can be executed anyway. In our example, if press is declared a broadcast channel, the press! of the user can be executed even if the lamp is in location bright . We will make use of this *1-to-many synchronization* feature.

Switching on a lamp and letting it burn uses energy, which causes costs. Up-paal Cora provides the possibility to keep track of costs accumulated during the operation of the modeled system. For this purpose, there is a special variable cost , which can be increased explicitly during switching by an update, or implicitly by specifying a rate. In Figure 5.3, switch $\text{off} \rightarrow \text{low}$ is labeled with update $\text{cost} += 50$, indicating that it takes 50 energy units to switch on the lamp. In locations low and bright we have the extra “invariants” $\text{cost}' == 10$ and $\text{cost}' == 20$, respectively, which indicate that the energy consumption is 10 and 20 units per time unit in the respective locations. When staying in these locations, cost is implicitly increasing with time, with rate 10 and 20, respectively.

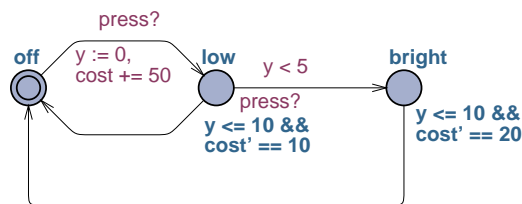


Figure 5.3 Model of automatic lamp with costs.

In order to express the prioritized execution of certain transitions, locations

can be marked *Committed*. If one component is in a committed location, then the next switch to take must emanate from a committed location, and it must be taken without any delay.

It is possible to use simple data types in NTA like arrays and structures, based on integers, which can be declared either local to a single, or global to all automata. These data types can be manipulated in assignments, which are executed while switches are executed, and can be referenced in guards and—with some restrictions—also invariants. In our NTA battery model we will make extensive use of this feature.

5.2.2 Schedule generation using PTA

Uppaal Cora is a model checker for PTA, *i.e.*, a tool to check whether the modeled PTA has certain properties which are expressed as logic formulae in a fragment of the *timed computation tree logic* (TCTL) [1]. An important problem to solve for PTA to make model checking feasible is *minimum-cost reachability*, *i.e.*, given a PTA and a target state (where a *state* is a combination of a location and additional information about the clocks), determine the minimal cost of all paths leading from the initial location to the target state [5], while meeting a certain time bound. In [2, 5] it has been shown independently that this problem is effectively computable.

This very important result allows us to use PTA for schedule generation as follows. PTA models can be nondeterministic. A model checker like Uppaal Cora can find paths—starting in the initial location—through the state space of the network of timed automata to target states and compute the minimal cost to do so. These paths resolve nondeterministic choices on the way to the target. The idea of schedule generation with model checkers is to model the system to be scheduled, which is in our case a combination of resources and load, but to leave the scheduling decisions open, *i.e.*, nondeterministic. If a certain scheduling objective can be formulated as a state property of this system, then the model checker can be employed to find such a state and the path leading to it; *the determined path is a schedule*. The scheduling objective can depend on timing properties (“look for the fastest schedule”), or, in the case of Uppaal Cora, on the cost variable: the optimization criterion is the minimality of the accumulated cost, *i.e.*, Uppaal Cora tries to find the path with the lowest cost leading to the target state. This is what we will use to generate battery schedules.

5.3 A timed automata model for KiBaM

In this section we introduce the network of timed automata used to model the KiBaM. We denote this model by TA-KiBaM.

5.3.1 Discretization of the KiBaM

To be able to use the KiBaM in the timed automata setting of Uppaal Cora, one needs to discretize the model, in particular the total charge, and height difference in the KiBaM, since Uppaal Cora only uses integer variables. The battery behavior, discharge and recovery, is deterministic for a given load. Therefore, we also discretize the time in order to be able to define the points in time when charge is drawn from the battery or the battery recovers some charge. From the equations of the transformed KiBaM (cf.(3.2)):

$$\begin{cases} \frac{d\gamma}{dt} = -i(t), \\ \frac{d\delta}{dt} = \frac{i(t)}{c} - k'\delta, \end{cases} \quad (5.1)$$

it follows that the total charge only changes when a current is drawn from the battery. The height difference changes by two processes: it increases when a current is drawn, and decreases when charge flows from the bound-charge well to the available-charge well. In the discretization of the model we separate these two processes.

We discretize time in steps of size T . Within a time step the discharge current is assumed constant. For a constant current (I), the total charge will decrease linearly. The total charge is discretized in N parts of size $\Gamma = C/N$. It will take $\Gamma/(I \cdot T)$ time units to decrease the total charge with one charge unit at a discharge rate of I .

At the same time, the discharge with current I will increase the height difference with Γ/c . This will be the step size of the discretization of the height difference. Once some charge is drawn from the available-charge well, charge will start to flow between the bound and available-charge well. This is a non-linear process, described by the second part of the second equation in (5.1):

$$\frac{d\delta}{dt} = -k'\delta. \quad (5.2)$$

The solution to this differential equation is given by:

$$\delta(t) = \delta(t_0)e^{-k't}. \quad (5.3)$$

If at time point t_0 we have $\delta(t_0) = m \cdot \Gamma/c$, then the time t needed to decrease the height difference with one unit to $\delta(t) = (m - 1) \cdot \Gamma/c$, is

$$t = -\frac{1}{k'} \cdot \ln \left(\frac{m-1}{m} \right). \quad (5.4)$$

To obtain the number of time steps of size T it takes to decrease the height difference by one unit, we divide this time by T and round to the nearest integer.

5.3.2 Towards modeling

In the TA-KiBaM, the relevant information of the battery state is kept in arrays of integers. We assume that every battery in the model has a local variable id , set with a value different to that of all other batteries.

We keep track of the *total remaining charge* and the *height difference of the two wells in numbers of charge units*. We introduce two arrays, n_gamma and m_delta , of the size equal to the number of batteries. $n_gamma[id]$ is the total charge left in battery id , $m_delta[id]$ the height difference. Thus, initially $n_gamma[id] = N$ and $m_delta[id] = 0$.

The recovery characteristics of a battery is described by the array $recov_times$. The contents of this array is pre-computed using (5.4). If the height difference of the battery is $m_delta[id]$, then $recov_times[m_delta[id]]$ is the time it takes to decrease the height difference by one charge unit. The array $recov_times$ is independent of the load put on the battery. The load is described by three arrays of equal length. Figure 5.4 gives an example of a load and the corresponding arrays. The size of T is set to 1 s and the Γ is set to 1 As.

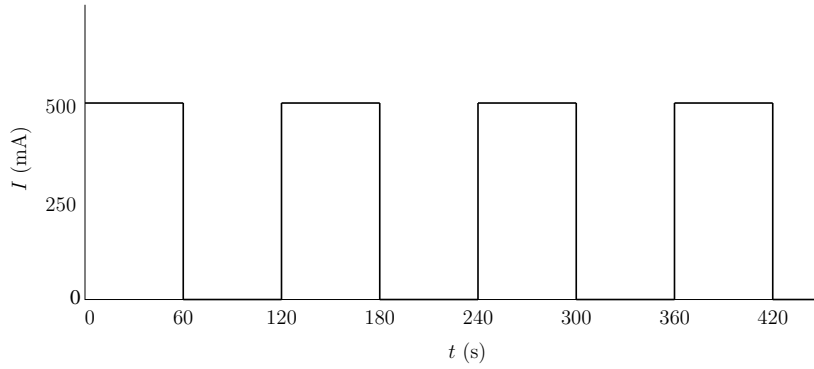
- In array $load_time$, the times are stored when the load changes. The times are defined absolutely, counted from system start, *i.e.*, $load_time$ contains a strictly increasing sequence of numbers. This array defines certain *epochs* of the battery usage period, where $load_time[j]$ is the time where epoch j ends and epoch $j + 1$ starts.
- The array cur_times has the same size as $load_time$ and gives the number of time units it takes to draw some charge units from the battery during the epochs.
- Array cur has the same size as $load_time$ and together with the array cur_times it defines the current drawn from the battery during an epoch. The array gives the number of charge units drawn from the battery during one of the periods defined in cur_times . The current (I_j) drawn from the battery during epoch j is given by the equation:

$$I_j = \frac{cur[j]\Gamma}{cur_times[j]T}. \quad (5.5)$$

In the example, during the periods with positive current, every two seconds one charge unit of 1 As is drawn from the battery.

The three arrays are created using an external program, and imported into the TA-KiBaM. The described variables are listed for reference in Table 5.1.

The criterion of when a battery is empty is given by $\gamma(t) = (1 - c)\delta(t)$, see Equation (3.3). The fact that we use charge units as the central measure makes it necessary to rephrase this criterion to be usable in the NTA model. If n is the total



load_time = [60, 120, 180, 240, 300, 360, 420, ...]
 cur_times = [2, 0, 2, 0, 2, 0, 2, 0, 2, 0, ...]
 cur = [1, 0, 1, 0, 1, 0, 1, 0, ...]

Figure 5.4 Example load and the arrays describing it.

Var	Type	Description
id	int	unique number for each battery
n_gamma	array	the current total charge for each battery, measured in charge units, initially N (cf. Section 5.3.1)
m_delta	array	the current height difference for each battery, measured in charge units, initially zero
load_time	array	times when epoch ends (precomputed)
cur_times	array	times it takes to discharge one charge unit (precomputed)
cur	array	number of charge units consumed within cur_times (precomputed)
j	int	current epoch, and thus index to arrays load_time, cur_times, and cur
recov_time	array	times it takes to recover one charge unit (precomputed)

Table 5.1 Important variables.

number of charge units left in the battery and m the height difference in number of charge units at time t , then $\gamma(t) = n\Gamma$, $\delta(t) = m\Gamma/c$, hence $\gamma(t) = (1 - c)\delta(t)$ can be transformed to $cn = (1 - c)m$. We change the equality sign to a less-than-or-equal-to sign, to account for errors due to the discretization:

$$cn \leq (1 - c)m. \tag{5.6}$$

5.3.3 Basic battery model

The discharge and recovery behavior of the discretized KiBaM is modeled by two timed automata, which are depicted in Figure 5.5. Figure 5.5(a) shows the *total charge* automaton, Figure 5.5(b) the *height difference* automaton for battery `id`. The total charge automaton keeps track of the level of the total charge in the battery with number `id` (`n_gamma[id]`). The automaton starts in location `idle`. When the battery is used, signaled on channel `go_on`, the automaton changes state to location `on` and the clock `c_disch` is reset. The global variable `j` (modified in another component) is the current epoch, and is used as index to the arrays `load_time`, `cur_times`, and `cur`. After `cur_times[j]` time units, a number of `cur[j]` charge units is subtracted from `n_gamma[id]` and added to `m_delta[id]`. Both `cur_times[j]` and `cur[j]` depend on the current which is discharging the battery.

The battery can be used until it is empty. In the total charge automaton this is checked using the guard $(1000 - c) * m_delta \geq c * n_gamma$, which is exactly condition (5.6). When this inequality holds the battery is empty, and the total charge automaton changes state to location `empty`. While all variables in Uppaal Cora can only take integer values, the parameter c , which in the KiBaM lies between 0 and 1, is multiplied with 1000. This also is the reason the factor 1000 appears in the guard.

The height difference automaton keeps track of the height difference between the two wells, `m_delta[id]`. Initially, `m_delta[id]` is zero. For every charge unit drawn from the battery, `m_delta` is increased with one unit. When `m_delta[id] > 1`, recovery starts and `m_delta[id]` is decreased by the automaton. The time it takes to recover one unit depends on `m_delta[id]`. The times are precomputed, using (5.4) and stored in the array `recov_times`. Thus, after `recov_times[m_delta[id]]` time units, `m_delta[id]` is decreased by one unit. The clock `c_recov` is used to check whether the appropriate time has passed to recover a unit.

5.3.4 Battery scheduling

In the load automaton (Figure 5.6(a)) the array `load_time` is used to determine the start and end times of the different epochs in the load. The array `cur` is used to determine whether an epoch contains a job (`cur[j] > 0`) or an idle period (`cur[j] = 0`). When the epoch contains a job the load automaton will synchronize with the scheduler automaton at the start of the epoch through the `new_job` channel

5.3 A timed automata model for KiBaM

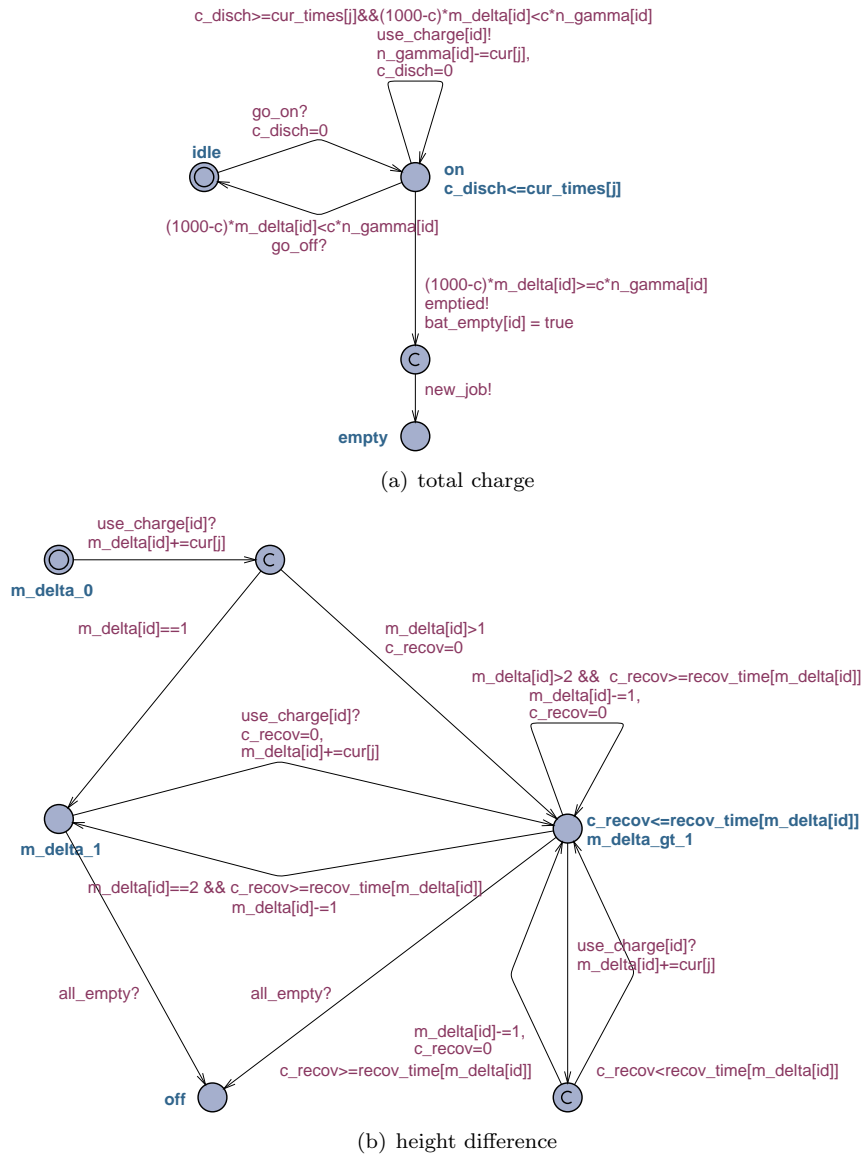


Figure 5.5 Timed automata modeling the KiBaM.

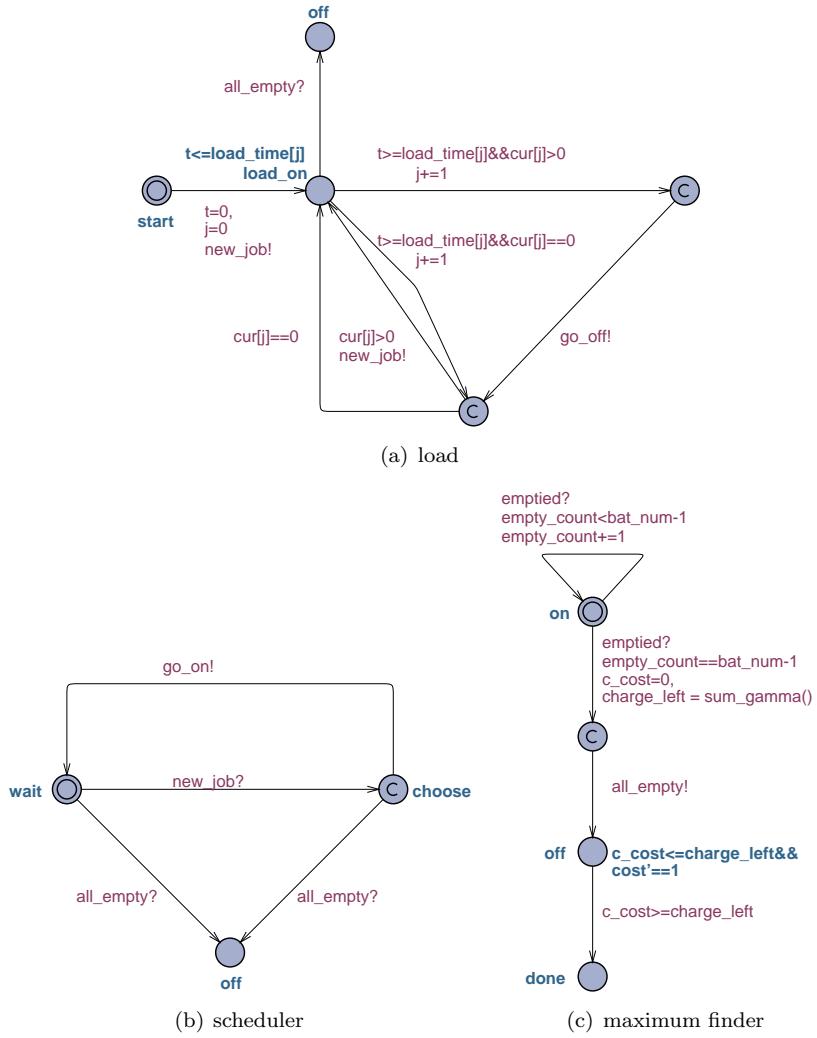


Figure 5.6 Timed automata for battery scheduling.

to schedule the next battery. At the end of the job the load automaton synchronizes through the `go_off` channel with the assigned battery to switch it off. These two synchronizations are not needed for idle periods, since no battery is used. When the load automaton synchronizes with the scheduler automaton (Figure 5.6(b)) through the `new_job` channel, the scheduler will nondeterministically synchronize through the `go_on` channel with one of the batteries. When a battery is emptied during a job, the scheduler chooses another battery to continue the job at the point the other stopped. Although the battery can still recover some charge, we assume it can not be used anymore once observed empty. The optimal schedule, *i.e.*, the one that yields the longest system lifetime, is the schedule that takes the longest for both batteries to reach the location `empty`. We want the model checker to find the longest path to this state. Unfortunately, Uppaal Cora is not able to find the longest path leading to a target state. It is thus necessary to translate the question for the longest path that leads to the maximum lifetime into a question for a minimum cost trajectory. This has been done by adding the *maximum finder* automaton (Figure 5.6(c)). When a battery is empty, the total charge automaton will signal the maximum finder automaton on channel `emptied`. The maximum finder automaton counts the number of empty batteries. When all batteries are empty the maximum finder automaton broadcasts on the channel `all_empty` to stop all processes in the other automata. Now, the maximum finder automaton converts the charge remaining in the bound-charge well of the batteries into a cost. The path that will lead to the longest system lifetime will have used the most charge from the batteries and therefore will have the smallest amount of charge remaining in the bound-charge wells. In Table 5.2 all used synchronization channels are listed to given an overview of the interactions between the different automata.

We use thus Uppaal Cora to check the simple TCTL property

$$A[] \text{ not max.done.} \quad (5.7)$$

This property is not satisfied, and Uppaal Cora returns, with appropriately chosen options, a path as a counterexample which minimizes the cost and maximizes the system lifetime.

5.3.5 Complexity

The *complexity* of finding the optimal schedule clearly depends exponentially on the number of scheduling decisions that have to be made, where the number of batteries (B) is the base. At every scheduling point one can choose between all B batteries. The number of scheduling points depends on the battery's capacity and the load applied.

Between the scheduling points the model is fully deterministic. The number of states in between two scheduling points will depend on the granularity of the

channel	sending automata	receiving automata	action
new_job	load, total charge	scheduler	call the scheduler to make a scheduling decision when a new job starts or a battery is empty
go_on	scheduler	total charge	switch the battery chosen by the scheduler to on
use_charge	total charge	height difference	increase the height difference with one unit for every charge unit drawn from the battery
emptied	total charge	maximum finder	add one to the number of batteries that are empty
all_empty	maximum finder	height difference, load, scheduler	stop all battery processes when all batteries are empty

Table 5.2 Overview of the synchronization channels used in the TA-KiBaM.

test load	lifetime	lifetime	difference
	KiBaM (min)	TA-KiBaM (min)	%
CL_250	4.53	4.56	0.7
CL_500	2.02	2.04	1.0
CL_alt	2.58	2.60	0.8
ILs_250	10.80	10.84	0.4
ILs_500	4.30	4.32	0.5
ILs_alt	4.80	4.82	0.4
ILs_r1	4.72	4.74	0.4
ILs_r2	4.72	4.74	0.4
ILl_250	21.86	21.88	0.1
ILl_500	6.53	6.56	0.5

Table 5.3 Battery lifetimes of battery B_1 for the different loads computed with both the analytical KiBaM and the discretized timed automata KiBaM.

discretization. The maximum number of state changes that can be made due to discharging is $N = C/\Gamma$, when all charge units are drained one at a time. The maximum number of state changes due to recovery is not so easy to define. However, it will be proportional to $1/\Delta$. Since in the model, $\Delta = \Gamma/c$ the maximum number of states will be proportional to $(1/\Gamma)^2$. The discretization of time will not influence the maximum number of states, when the discretization of the charge is not changed. The discretization of time is only used to define the guards and the invariants in the model. Introducing smaller time steps will only increase the numbers in the guards and invariants.

5.4 Validation of the TA-KiBaM

Before computing schedules with the TA-KiBaM, we need to check whether it models the KiBaM correctly. To validate the TA-KiBaM, we compare the battery lifetimes computed with this model and with the original KiBaM. We consider a single-battery case, and use two different battery types, one with capacity 5.5 Amin (Ampere-minute) (B_1) and one with capacity 11 Amin (B_2). The battery parameters are the same for both batteries: $c = 0.166$ and $k' = 0.122 \text{ min}^{-1}$ [37], corresponding to the lithium-ion battery used in the Itsy pocket computer, which was also simulated by Rakhmatov et al. [54, 55]. The time step size is set to 0.01 min. The total charge is discretized in steps of size 0.01 Amin. This leads to the discretization step size of the height difference of $0.01/c = 0.06$ Amin .

The Itsy pocket computer operates with currents up to 700 mA. In the tests we used two types of jobs, a low current job (250 mA) and a high current job

(500 mA). With these jobs, ten different test loads have been created:

- three continuous loads (CL) with no idle periods between the jobs: one load with only low current jobs (CL_250), one with only high current jobs (CL_500), and one alternating between a low current job and a high current job (CL_alt).
- five intermittent loads with short idle periods of one minute between the jobs (ILs) : one with only low current jobs (ILs_250), one with only high current jobs (ILs_500), one alternating between a low current job and a high current job (ILs_alt), and two where the job is randomly chosen (ILs_r1 and ILs_r2)
- two intermittent loads with long idle periods of two minutes between the jobs (ILℓ): one with only low current jobs (ILℓ_250) and one with only high current jobs (ILℓ_500).

The results of the tests for the two battery types are given in Table 5.3 and 5.4. For most loads the lifetime in the timed automaton battery model is only between 0.02 and 0.04 min longer for the analytical KiBaM. For the loads CL_250 and CL_alt TA-KiBaM gives a bigger difference for battery B_2 . This bigger difference in computed lifetimes is due to the discretization of the height difference and the time needed to recover one height unit. When a battery is discharged, the height difference can grow up to the point that the time to increase the height difference with one unit equals the time to decrease the height difference with one unit, *i.e.*, $cur_times[x] = recov_time[y]$. The $recov_time$ is computed according to (5.4) and rounded to the nearest number of time steps. The rounding causes the height difference to grow less than it does in the analytical version of the model. The amount of charge that is unavailable when the battery is empty is proportional to the height difference, so the smaller height difference will give a longer lifetime. The moment the height difference does not increase any more is only reached with the loads CL_250 and CL_alt applied to B_2 . In all other cases, either the battery lifetime is too short for the height difference to reach this point, or the load has idle periods in which the height difference will decrease. Note that even for the loads with the bigger difference, the relative difference is still only 1%.

Although the discretized model sometimes gives small differences in lifetime computation compared to the analytical model, we regard it perfectly usable for attacking the scheduling problem.

5.5 Scheduling results

We use the multiple battery priced timed automaton model to find the optimal way to schedule two batteries on the same test loads as used in Section 5.4. We use batteries of type B_1 .

test load	lifetime	lifetime	difference
	KiBaM (min)	TA-KiBaM (min)	%
CL_250	12.16	12.28	1.0
CL_500	4.53	4.54	0.2
CL_alt	6.45	6.52	1.1
ILs_250	44.78	44.80	0.04
ILs_500	10.80	10.84	0.4
ILs_alt	16.93	16.94	0.1
ILs_r1	22.71	22.74	0.1
ILs_r2	14.81	14.84	0.2
ILl_250	84.90	84.92	0.02
ILl_500	21.86	21.88	0.1

Table 5.4 Battery lifetimes of battery B_2 for the different loads computed with both the analytical KiBaM and the discretized timed automata KiBaM.

Next to computing the maximum lifetime, we used the TA-KiBaM to compute the lifetime using three deterministic scheduling schemes. Hence, we compare four schedules:

- Sequential schedule. The batteries are used sequentially, *i.e.*, the second battery is only chosen when the first one is empty.
- Round robin schedule. For every new job a new battery is chosen. The batteries are chosen in a fixed order.
- Best-of-two schedule. At the start of a job, the status of the batteries is checked. The battery with the most charge in the available-charge well is chosen to supply the charge for the job.
- TA-KiBaM schedule. The schedule computed using Uppaal Cora.

The computed lifetimes are given in Table 5.5, along with the relative difference to the lifetime using the round robin scheduling. For the test loads, one can see the order in performance of the different scheduling schemes. One can easily show, using the Uppaal Cora model, that the sequential scheduling is actually the worst possible way to schedule the batteries. For the test loads the round robin and best-of-two scheme differ only in the cases of the alternating jobs. These cases are clearly very bad for the round robin scheme, since the heavy load is always put onto the same battery. This battery will be emptied very fast, and then only one battery is left to handle the remaining load, leaving this battery with less idle time to recover. The best-of-two scheme balances the load better over the two

test load	sequential		round robin	best-of-two		TA-KiBaM	
	lifetime (min)	difference %	lifetime (min)	lifetime (min)	difference %	lifetime (min)	difference %
CL_250	9.12	-21.4	11.6	11.6	0	12.04	3.79
CL_500	4.10	-9.5	4.53	4.53	0	4.58	1.1
CL_alt	5.48	-10.2	6.10	6.12	0.3	6.48	6.2
ILs_250	22.80	-41.5	38.96	38.96	0	40.80	4.7
ILs_500	8.60	-17.9	10.48	10.48	0	10.48	0
ILs_alt	12.38	-3.4	12.82	16.30	27.2	16.91	31.9
ILs_r1	12.80	-21.28	16.26	16.26	0	20.52	26.2
ILs_r2	12.24	-15.59	14.50	14.50	0	14.54	0.3
ILℓ_250	45.84	-39.7	76.00	76.00	0	78.96	3.9
ILℓ_500	12.94	-18.9	15.96	15.96	0	18.68	17.0

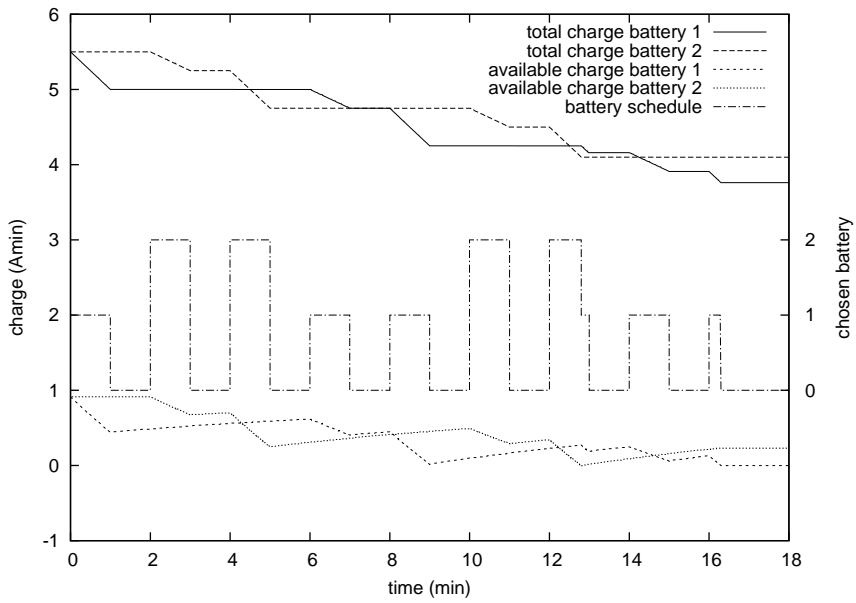
Table 5.5 System lifetime computed for all test loads according to the four scheduling schemes. Next to the values of the lifetimes, the difference relative to the round robin scheme are given.

batteries, which leads to a longer lifetime, especially in the `ILs_alt` case. In the other cases the best-of-two scheme behaves exactly like the round robin scheme. Although the round robin and best-of-two schedulers perform close to optimal in most cases, for some loads the schedules are far from optimal. The optimal scheduler yields lifetime improvements up to 32%.

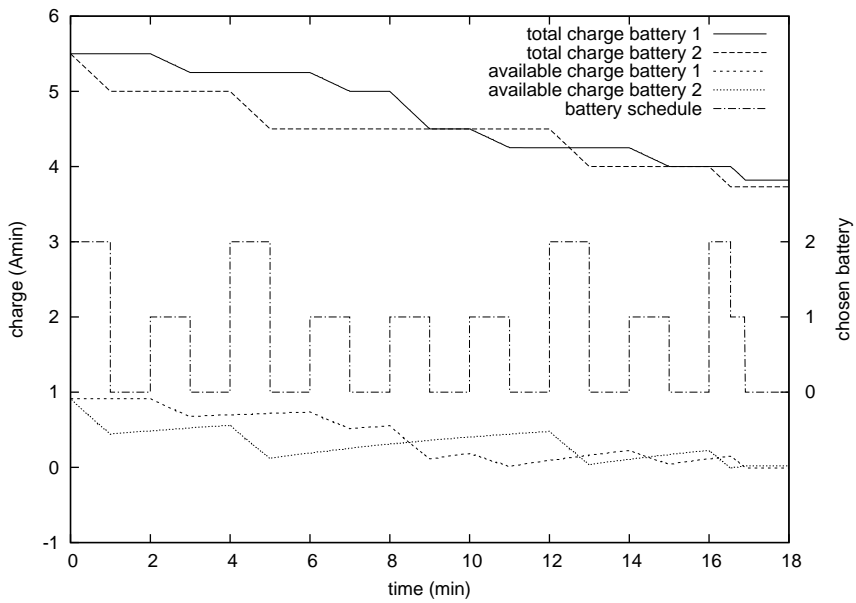
Besides the system lifetimes, the Uppaal Cora evaluation of the timed automata battery model also provides the actual schedules which lead to these lifetimes, as well as the evolution of the charge in the battery. Figure 5.7 shows the evolution of the total and available charge in the two batteries (left y -axis) for both the best-of-two scheduler (Figure 5.7(a)) and the optimal scheduler (Figure 5.7(b)), in the `ILs_alt` case. In the figure, also the two schedules (right y -axis) are shown. When a battery is chosen, one can see the total and available charge decrease due to the load. The slope of the curves is proportional to the discharge current. When a battery is not used, one can see the available charge rise again. This is due to the recovery effect. Note that, when the batteries are empty, still a relatively large amount of charge remains in the battery (approximately $3.9 A_{min}$).

Due to the complexity of finding the optimal schedule, it is possible to model only a limited total battery capacity. The used discharge currents are relatively high for the battery’s capacity, and will drain the available-charge well fast. Therefore, there will be little time for the bound charge to become available, and a large fraction will remain unused. When the battery capacity is increased this fraction will be smaller. Using the deterministic scheduling scheme, we can compute the results for larger capacities. For example, with a ten times larger capacity, the fraction of charge left behind in the batteries will be less than 10% in the case of best-of-two scheduling.

When we look at the two schedules in Figure 5.7, we see that the best-of-two



(a) best-of-two



(b) optimal

Figure 5.7 The schedules and the total and available charge in the batteries for the best-of-two (a) and the optimal (b) schedule for the ILs_{alt} load.

scheduler	mean lifetime (min)	variance (min ²)
sequential	19.72	2.39
round-robin	30.31	10.06
best-of-two	31.11	9.50
TA-KiBaM	33.52	8.73

Table 5.6 Mean and variance of the lifetimes obtained with the different schedulers for the loads with random on-times.

schedule acts like a round robin scheduler that switches batteries after the high current jobs. The optimal schedule seems to behave randomly, no direct relation between the state of the batteries and the schedule can be made. The optimal schedule does depend strongly on the size of the batteries and their parameters, as well as on the load that is applied.

5.6 Towards random loads

The test loads presented in the previous section are very regular. The discharge current switches after fixed time periods. However, most realistic loads are not regular and have discharge and idle periods of random length. In this section we introduce some randomness into the loads.

As a first step towards more realistic loads we introduce discharge periods of random length. The load profile we use is still an on-off load, where the discharge current switches between 250 mA and 0 mA. All off-periods last one minute. However, the lengths of the on-periods are chosen from a uniform distribution on $[\frac{1}{2}, \frac{3}{2}]$ min. The scheduling decision is made at the start of each on-period. To see how the different scheduling schemes perform for this random load, 500 load traces have been generated. The TA-KiBaM scheduler is used to compute the optimal schedule for a system with two batteries separately for every trace. Also, for each of these traces the system lifetime under the three deterministic scheduling schemes has been computed. The battery capacity in these computations was set to 5 Amin for each battery. Since the loads are random, the result is now an empirical distribution. The empirical distributions are shown in Figure 5.8. The mean and variance of the distributions are given in Table 5.7. Like with the test loads, the sequential scheduler results in much shorter lifetimes than the other schedulers, and is far from optimal. On average the TA-KiBaM scheduler outperforms the sequential scheduling scheme by 70%. The results of the round robin scheduler and the best-of-two scheduler lie close to each other, although the latter performs slightly better. The schedules of TA-KiBaM outperform round robin and best-of-two by 10% and 8%, respectively.

In a second set of random loads, we keep the duration of the discharge periods

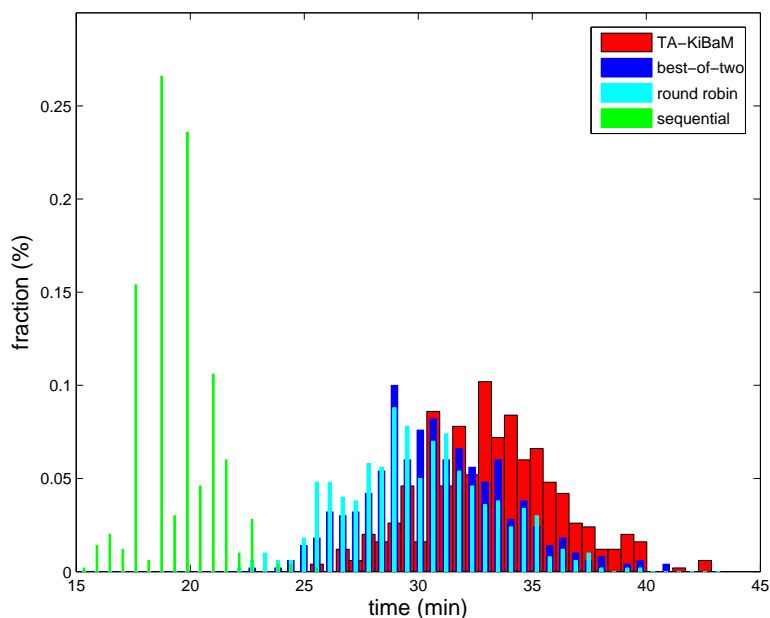


Figure 5.8 Empirical lifetime distributions of the four scheduling schemes for 500 random loads with random on-times.

constant, but we randomly choose the discharge current. As in the test loads, the periods of constant current last one minute. Every minute we uniformly choose the discharge current from the set 0, 100, 200, 300, 400 500 mA. Again, 500 load traces have been generated, and for each of these traces the system lifetime under the different schedulers has been computed.

The results of these computations are give in Figure 5.9 and Table 5.7. For this set of loads the system lifetimes under the different schedulers lie much closer to each other. Still the same order of performance of the schedulers can be observed. The average load in this set of load traces is twice as high as for the loads with random on-times. This results in shorter system lifetimes, and thus in less scheduling moments. Therefore the schedulers have less opportunities to balance the load over the batteries and, therefore, have less opportunities to improve the system lifetime.

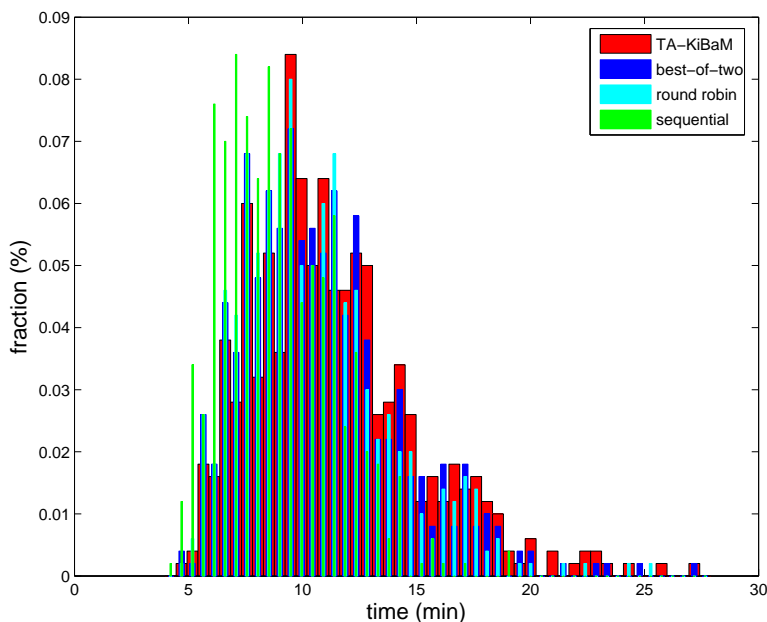


Figure 5.9 Empirical lifetime distributions of the four scheduling schemes for 500 random loads with random discharge current.

scheduler	mean lifetime (min)	variance (min ²)
sequential	9.17	6.11
round-robin	10.08	10.32
best-of-two	11.02	11.35
TA-KiBaM	11.61	13.37

Table 5.7 Mean and variance of the lifetimes obtained with the different schedulers for the loads with random discharge currents.

5.7 Conclusions

We have shown a new approach to maximize the system lifetime through battery scheduling. The priced timed automaton battery model allows us to compute the optimal way to schedule multiple batteries for a given load. The optimal schedule and lifetime can easily be compared to straightforward scheduling schemes, like round robin. For most of the tested loads the round robin and best-of-two schedulers perform close to optimal. However, the optimal schedules determined by model checking reveal that there still is room for improvement.

The random loads are analyzed using a large set of randomly generated traces, and computing the optimal schedule for each trace separately. This results in a large set of schedules, which needs to be analyzed in order to find a general scheduling strategy for the random workload. So far, this has not led to any clear insights. To be able to include the actual random workload into the priced timed automata model and generate an optimal scheduling scheme for the workload at once, one needs probabilistic priced timed automata. For this, algorithms and a tool are in development [8, 9].

The optimal scheduler is not something one can use in real life systems, since complete knowledge of the future load is needed. Also, the computational complexity of the approach limits the number of scheduling points. Therefore, the computation of optimal schedules is limited to batteries with a small capacity. Also, computing schedules for systems with more than two batteries is not practically feasible. In the next chapter we will take another, analytic approach to the battery scheduling problem, to overcome the limitations of the priced-timed automata model.

Chapter 6

An analytic and simulation approach to battery scheduling

The computations of optimal schedules with the linear priced timed automata approach described in the previous chapter are limited to small batteries and a maximum of two batteries. In this chapter we will take another, analytic approach to the scheduling problem that also gives a solution for large batteries and arbitrarily many batteries.

Section 6.1 presents the analytic approach to the scheduling problem; it derives a formula to compute the maximum possible lifetime gain one can obtain theoretically by battery scheduling. In Section 6.2 a new scheduler, named the greedy scheduler, is used to approach the maximum possible lifetime. In Section 6.3, we compare the maximum lifetime gain with the one obtained by parallel discharge. In Section 6.4, we apply different battery schedulers to random loads for a system of two batteries with realistic capacity. Section 6.5 concludes this chapter.

6.1 Nonuniform scheduler generation

The linear priced timed automata approach described in Chapter 5 was based on the assumption that a scheduling decision must be made at predefined points in time. Even with this restriction, schedule generation for realistic battery capacities is not feasible because the time needed to assess all possible schedules grows exponentially when the number of scheduling points is increased. Due to the computational complexity, also systems with more than two batteries are infeasible to analyse.

In this chapter we investigate a slightly more general scheduling problem, which differs in two points. First, the question is not only to choose the optimal among available batteries, but also *when* to do this. Second, a battery can be reused after its available charge well has been completely drained and some recovery period afterwards. The chosen approach to tackle this scheduling problem is *analytic*. This approach does supply a solution for both realistic battery capacity and systems with more than two batteries.

We consider a system with M identical batteries with KiBaM parameters c , k , and C . For this system we define a switching mechanism consisting of a collection of M functions $[u^{(1)}, \dots, u^{(M)}]$ with $u^{(j)}(t) : \mathbb{R}_{\geq 0} \rightarrow \{0, 1\}$, $j \in \{1, \dots, M\}$. $u^{(j)}(t) = 1$ iff battery j is selected at time t . Note that, since always a battery is selected at any time t , $\sum_{j=1}^M u^{(j)}(t) = 1$, for $t \geq 0$.

When a battery is selected, a charge is drawn from it with current $i(t)$. Given the switching mechanism $[u^{(1)}, \dots, u^{(M)}]$, the available and bound charges of all M batteries can be expressed using the system of ODEs in Equation (2.4):

$$\begin{cases} \frac{dy_1^{(j)}(t)}{dt} = -i(t)u^{(j)}(t) + k(h_2^{(j)}(t) - h_1^{(j)}(t)), \\ \frac{dy_2^{(j)}(t)}{dt} = -k(h_2^{(j)}(t) - h_1^{(j)}(t)), \end{cases} \quad j = 1, \dots, M \quad (6.1)$$

where $y_1^{(j)}$ is the available charge and $y_2^{(j)}$ is the bound charge in battery j . Here we take again the initial conditions $y_1^{(j)}(0) = c \cdot C$ and $y_2^{(j)}(0) = (1-c) \cdot C$, where C is the total initial battery charge. Similar as before, $h_1^{(j)}(t) = \frac{y_1^{(j)}(t)}{c}$ and $h_2^{(j)}(t) = \frac{y_2^{(j)}(t)}{1-c}$.

The available charge of all the batteries may never be smaller than 0. This gives the following set of set of boundary conditions:

$$y_1^{(j)}(t) \geq 0, \quad \forall t \geq 0, j = 1, \dots, M. \quad (6.2)$$

Example 1 Figure 6.1 depicts the available charge of two batteries, where the switching functions $u^{(1)}(t) := u(t)$ and $u^{(2)}(t) := 1 - u(t)$ are defined in terms of the piecewise constant control function u (upperpart of Figure 6.1. Note that in the interval of time $[t_0, t_1)$, the first battery is selected and therefore its available charge decreases, while the available charge of the second battery remains constant. When in the interval of time $[t_1, t_2)$ the second battery is selected, the available charge $y_1^{(2)}(t)$ decreases, while $y_1^{(1)}(t)$ increases due to the recovery effect.

For a system with M identical batteries with parameters k , c and capacity C , and a switching mechanism $[u^{(1)}, \dots, u^{(M)}]$, the total available charge is $\hat{y}_1(t) = \sum_{j=1}^M y_1^{(j)}(t)$ and total bound charge $\hat{y}_2(t) = \sum_{j=1}^M y_2^{(j)}(t)$ at time t is

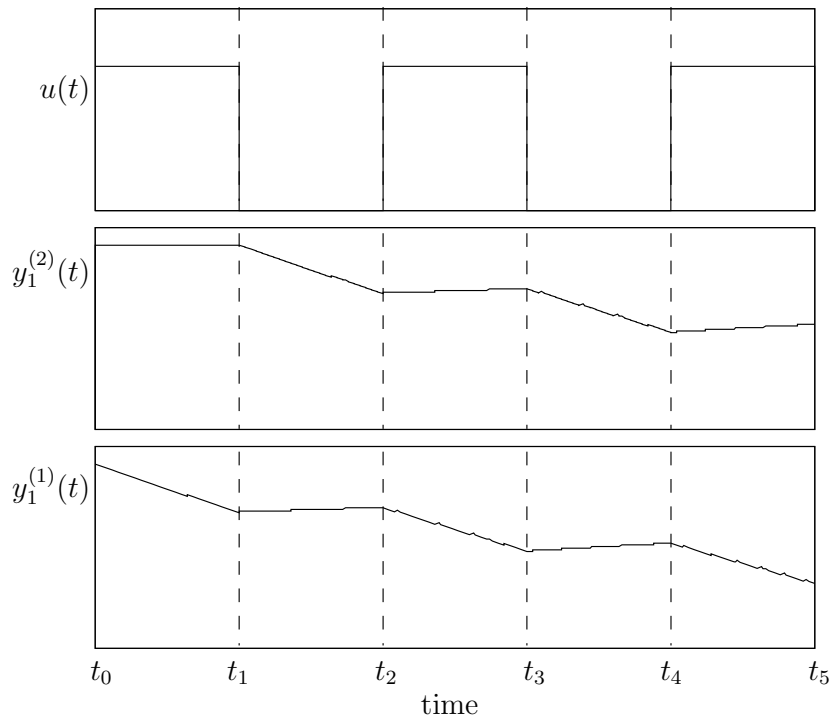


Figure 6.1 Control function $u(t)$ and the available charges $y_1^{(1)}(t)$ and $y_1^{(2)}(t)$.

given by the following system of ODEs:

$$\begin{cases} \frac{d\hat{y}_1(t)}{dt} = -i(t) + k(\hat{h}_2(t) - \hat{h}_1(t)), \\ \frac{d\hat{y}_2(t)}{dt} = -k(\hat{h}_2(t) - \hat{h}_1(t)), \end{cases} \quad (6.3)$$

where $\hat{h}_1(t) = \frac{\hat{y}_1(t)}{c}$, $\hat{h}_2(t) = \frac{\hat{y}_2(t)}{1-c}$ and the initial conditions are $\hat{y}_1(0) = M \cdot c \cdot C$ and $\hat{y}_2(0) = M \cdot (1-c) \cdot C$. This system of ODEs is obtained by adding up the Equations (6.1), and using the fact that $\sum_{j=1}^M u^{(j)}(t) = 1$ for all $t \geq 0$.

Equation (6.3) shows that the total remaining available charge of M batteries at any time t is equal to the total charge of a single battery \hat{B} with initial capacity $M \cdot C$, and parameters k and c . More remarkable is the fact that the total remaining available charge of all M batteries does not depend on the control functions $[u^{(1)}, \dots, u^{(M)}]$. This does *not* imply that the control functions do not influence the system lifetime. For all the separate batteries the boundary conditions in Equation (6.2) still need to hold. However, when we relax the boundary conditions such that only the total available charge may not be smaller than 0, $\hat{y}_1 \geq 0$, we can use the system of ODEs in Equation (6.3) to compute an upper bound for the system lifetime.

For a constant load function $i(t) = L$ with $L > 0$, the solution to Equation (6.3) can be derived as

$$\begin{cases} \hat{y}_1(t) &= -cLt + cMC - \frac{L(1-c)}{k'} (1 - e^{-k't}), \\ \hat{y}_2(t) &= -(1-c)Lt + (1-c)MC + \frac{L(1-c)}{k'} (1 - e^{-k't}), \end{cases} \quad (6.4)$$

where $k' = k/(c(1-c))$. The maximum total lifetime for a system with M batteries, $t_{f,M}$, can be expressed analytically as follows:

$$t_{f,M} = \frac{MC}{L} - \frac{1}{k'} \left(\frac{1-c}{c} - W \left(\frac{1-c}{c} e^{-\frac{MCk'}{L} + \frac{1-c}{c}} \right) \right) \quad (6.5)$$

where W is the Lambert W function. The Lambert W function is the inverse function of $f(x) = xe^x$ [69].

Using Equation (6.5) we can compute the maximum possible gain one can obtain by applying battery scheduling in the case of a constant discharge current. For a single battery discharged with a constant current the lifetime t_s is given by:

$$t_s = \frac{C}{L} - \frac{1}{k'} \left(\frac{1-c}{c} - W \left(\frac{1-c}{c} e^{-\frac{ck'}{L} + \frac{1-c}{c}} \right) \right), \quad (6.6)$$

The system lifetime when using M batteries sequentially will be Mt_s , hence the maximum possible gain with M batteries G_M is given:

$$G_M = \frac{t_{f,M}}{Mt_s}. \quad (6.7)$$

In Figure 6.2 the gain for a system with 2 batteries (G_2) is given as a function of the discharge current. The batteries that have been used in this computation are similar to those used in Chapter 5, *i.e.*, $c = 0.166$ and $k = 2.03 \cdot 10^{-4} \text{ s}^{-1}$. However, here the capacity is increased to a realistic value, $C = 2400 \text{ As}$. The discharge current has been varied between 0.1 A and 10 A. For this system of batteries the highest gain is obtained at a discharge current of approximately 0.85 A, where the gain is more than 1.9. The peak can be explained as follows. When the discharge current gets too high, the available-charge well will be depleted too fast and the slow recovery process will hardly increase the usable capacity, even when scheduling is applied. At low discharge currents the loss of capacity due to the rate capacity effect is low, *i.e.*, the flow of charge from the bound to the available charge well can keep up with the demand, and little charge will be left behind in the bound-charge well. Therefore, the gain of allowing batteries to recover by the scheduling is limited. However, at a discharge current of 0.1 A the gain still is approximately 1.05, and a 5% lifetime extension is still a considerable improvement.

When we look at Equation (6.5) and Equation (6.6) we see that the discharge current L always appears in direct relation with the battery capacity C , in the

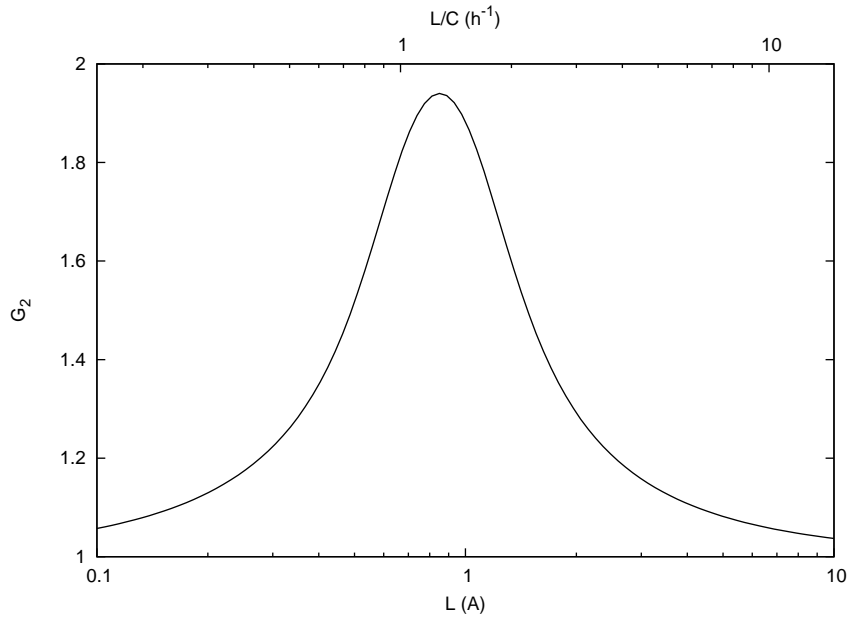


Figure 6.2 The maximum lifetime gain for a system with two batteries as a function of the constant discharge current. Note that the current is plotted in a logarithmic scale. The top x-axis gives the current normalized to the capacity of one battery.

form $\frac{C}{L}$. This implies that when the battery capacity is halved, and the other battery parameters stay the same, the discharge current needs to be halved as well to obtain the same lifetime gain. Using the top x-axis Figure 6.2 shows how the maximum lifetime gain depends on the current normalized to the capacity of one battery (L/C).

6.2 Greedy scheduler

6.2.1 Introduction

As stated before, Equation (6.3) shows that the sum of the available charge in all the batteries does not depend on the switching mechanism. However, one still needs to take into account that for all batteries, $y_1^{(j)}(t) \geq 0$ must hold at all time. In the previous section we used the condition $\hat{y}_1(t_f) = 0$ to compute the maximum lifetime t_f . This condition implies that for each individual battery j , $y_1^{(j)}(t_f) = 0$, *i.e.*, the available-charge well of the M batteries must all be empty at this precise point in time. However, for all switching mechanisms at any time t , only one battery is used. While this battery is used, the others may recover. When for the used battery the available charge well is emptied, one can switch to one of the other batteries that just recovered some charge and use this until it is empty. This leaves some time for the first battery to recover again. Since the recovery process is slow, and becomes even slower when also the bound-charge well is emptied, the time points at which one has to switch get closer to each other as the time approaches t_f , until the point that the switching will have to be infinitely fast.

This actually means that, when one allows for the once-emptied batteries to be reused after they recovered some charge, the switching mechanism does not impact the system lifetime as long as fast switching is possible when the maximum possible lifetime is reached.

6.2.2 Constant load

To investigate how fast the switching needs to be to get close to the maximum lifetime we define a new simple scheduler, termed the greedy scheduler. This scheduler only switches to the next battery when the available-charge well of the currently used battery is emptied. When all the batteries have been used once, the first, which had the most time to recover, is picked again.

We have applied this greedy scheduler to two constant loads, of 0.5 A and 1 A. Like before, we consider a system with two batteries with a capacity of 2400 As each, and the KiBaM parameters $c = 0.166$ and $k = 2.03 \cdot 10^{-4} \text{ s}^{-1}$.

Figure 6.3 shows the dependence of the system lifetime as a function of the number of times the greedy scheduler switches between the batteries. We clearly

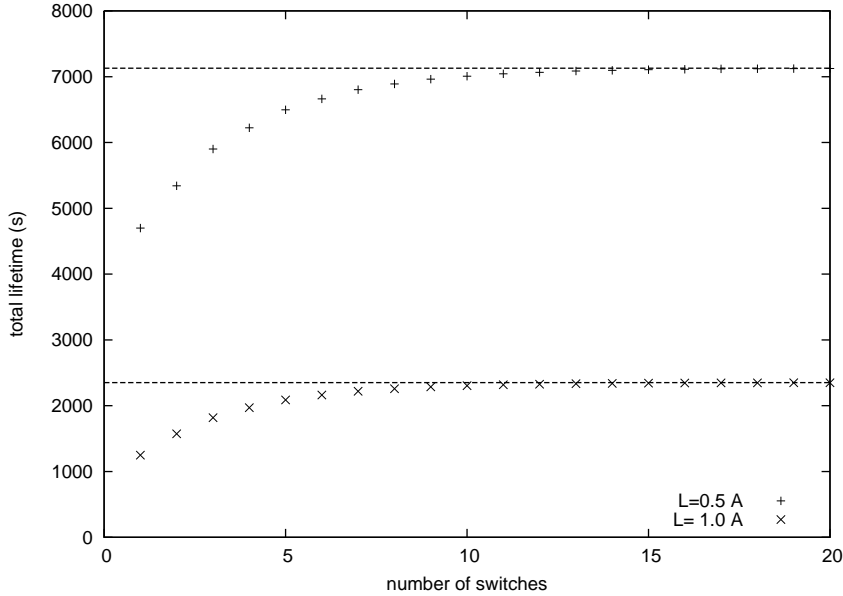


Figure 6.3 Dependency of the total lifetime regarding the number of switches using the greedy scheduler for two constant load functions and two batteries. The maximum possible lifetimes for the loads are indicated with the dashed lines (top: 0.5 A case, bottom: 1 A case).

see that the total lifetime asymptotically approaches a constant value, which is actually the optimal lifetime t_f , indicated by the dashed lines.

To see how fast the lifetime of the greedy scheduler approaches t_f , Figure 6.4 shows the difference between the two lifetimes as a function of the number of switches. We see that the difference decreases exponentially when the number of switches is increased. For both loads the difference is less than one second for only 25 times switching between the batteries. This is well below 0.1% of the total lifetime.

Figure 6.5 gives the lifetime that is added for every extra switching point. From this figure one can obtain an indication of how fast the scheduler needs to be in order to retrieve the extra charge from the batteries when the batteries are nearly empty. Switching at 1 Hertz will be sufficient to get within two seconds of the maximum possible lifetime.

6.2.3 Greedy scheduler results

The system lifetime for the test loads introduced in Chapter 5 has also been computed using the greedy scheduler. The results are given in Table 6.1. For most of the test loads the results are close to the lifetimes obtained by the TA-

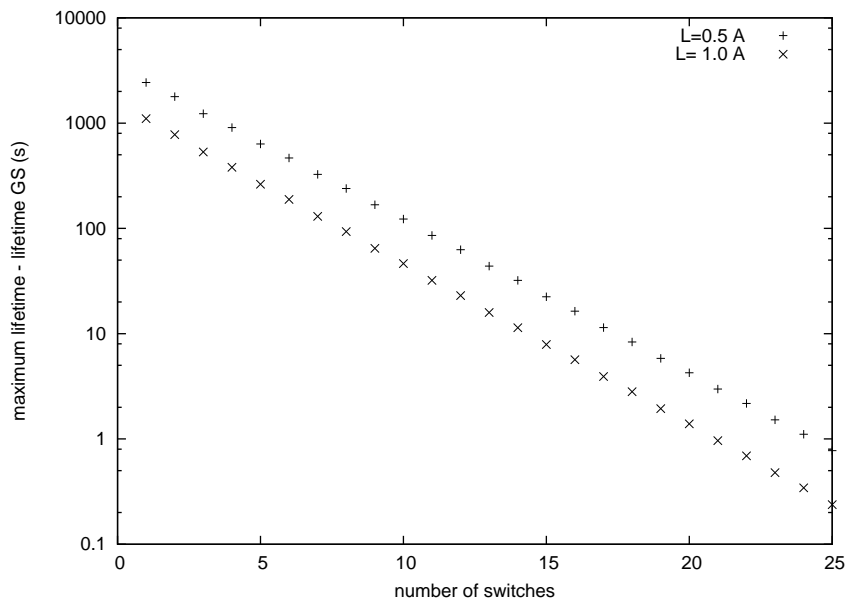


Figure 6.4 Difference between the maximum possible lifetime and the lifetime obtained with the greedy scheduler as a function of the number of switches.

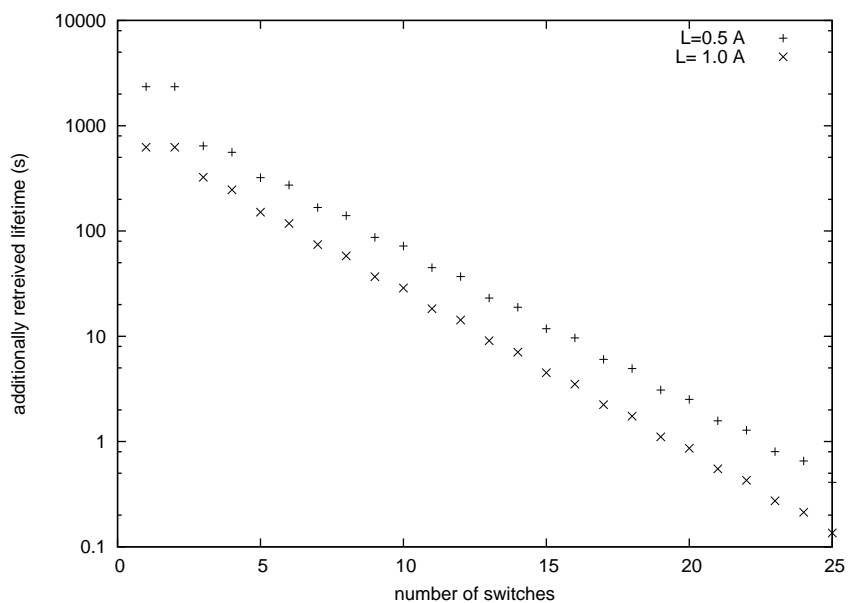


Figure 6.5 The additionally retrieved lifetime of the switching points using the greedy scheduler for two constant load functions and two batteries.

test load	GS algorithm lifetime (min)	TA-KiBaM lifetime (min)
CL_250	12.16	12.04
CL_500	4.53	4.58
CL_alt	6.45	6.48
ILs_250	44.77	40.80
ILs_500	10.80	10.48
ILs_alt	16.93	16.91
ILl_250	84.90	78.96
ILl_500	21.86	18.68

Table 6.1 System lifetime for the test loads using the greedy schedules and the TA-KiBaM schedules. For the highlighted loads the greedy schedule leads to a significantly longer lifetime.

KiBaM scheduler. However, for three of the loads, highlighted in gray in the table, the schedules obtained from the greedy scheduler result in a significantly longer lifetime. This difference is due to the fact that the TA-KiBaM scheduler can only switch batteries at fixed predefined points in time, whereas in the greedy schedule batteries may be switched at any time. The limitation in switching options may result in one of the batteries being emptied sooner, and thus giving a shorter system lifetime.

For the two loads CL_500 and CL_alt the lifetime for the TA-KiBaM seems to be slightly longer than for the greedy scheduler. This is caused by the small overestimation of the system lifetime by the TA-KiBaM, as was observed in Section 5.4. Taking this overestimation into account, the two schedulers result in approximately the same lifetime.

Besides that the schedules of the greedy scheduler often lead to longer lifetimes, the greedy scheduler has one major benefit compared to the TA-KiBaM. Where the TA-KiBaM cannot handle batteries with large capacity due to the exponential growth of the state space, the greedy scheduler can easily cope with larger batteries.

6.3 Parallel discharge

When we compare Equation (6.5) with the solution for the lifetime of a single battery, Equation (6.6), we see that the maximum lifetime of the M battery system is the same as the lifetime of a single battery that is discharged with an L/M current. This implies that discharging the M batteries in parallel, where the discharge current of every battery is equal to L/M , leads to the maximum possible

lifetime of the system.

However, the possibility of connecting batteries directly in parallel is under debate. Where [3] claims lithium batteries are well suited to connect in parallel, [41] says one should not do this. One problem of connecting batteries in parallel is that even for two batteries of the same type a difference in potential can occur. When this happens a current will flow between the batteries, resulting in a loss of capacity and possibly damage to the batteries. Using batteries in parallel requires extra electronic circuitry, which consumes some power and decreases efficiency.

Also, in some situations, like the routing problem described in the Section 5.1, parallel usage is simply impossible. Using a simple scheduling scheme, like greedy scheduling or round robin scheduling, one can circumvent the problems of parallel usage, and still obtain an improvement in system lifetime.

6.4 Random load scheduling

So far, we have applied our scheduling algorithms to simple regular loads. However, realistic loads are not that regular. In the previous chapter we presented a first step towards random loads with the priced timed automata model. However, the priced timed automata approach is limited to small batteries. The analytic approach presented in this chapter allows us to increase the battery capacity to a realistic size. In this section we introduce randomness into the loads in three steps. The first two are the same as presented in Section 5.6, these are used for comparison reasons, to see the effect of the capacity increase on the lifetime gain. In the third step, a fully random load, modeled by a continuous-time Markov chain, is used.

The approach presented in Chapter 4 of combining the KiBaM with a stochastic workload into the MRKiBaM is not feasible for the battery scheduling problem. The state space will grow exponentially with the number of batteries. Together with the addition of the scheduler, this would make the model too complex to deal with. Therefore, in this section we use simulations to obtain the scheduling results for random loads.

6.4.1 Simulation set-up

Like in the previous chapter, the batteries we model are the lithium-ion batteries that are used in the Itsy pocket computer. However, now the modeled battery has a capacity of 2400 As, which is the size used in the pocket computer. The KiBaM parameters for this battery are the same as in the previous chapter: $c = 0.166$ and $k = 2.03 \cdot 10^{-4} \text{ s}^{-1}$. In the analysis we use four basic scheduling schemes:

- *sequential*:, a next battery is chosen when the current one is empty.

- *load-round-robin*:, the batteries are chosen in a fixed order, a switch between batteries takes place at the moment the discharge current is changed to another positive current.
- *best-of-two*:, at the moment the load changes the battery with the most charge in the available-charge well is chosen.
- *time-round-robin*:, the batteries are chosen in a fixed order, a switch between batteries takes place after a fixed amount of time has passed.

In Chapter 5 we used three of these schedulers in the setting of priced timed automata: sequential, load-round-robin and best-of-two. These schedulers are used here to see what the effect is of the bigger battery capacity on the lifetime gain.

Before, we showed that when there are no restrictions on the moments of switching between batteries the actual scheduling scheme is not important, as long as one can switch fast enough when the batteries are nearly empty. Therefore, we use a fast switching round robin scheduler, that switches at fixed points in time. This scheduler is much easier to implement in practice than the greedy scheduler presented in Section 6.2, since for the round robin scheduler one does not need to monitor the battery status to be able to switch in time. Of course, the round robin scheduler will have to switch more often and at a much higher rate than the greedy scheduler, which will cost some extra energy. However, the energy needed to switch between batteries will be negligible compared to the actual load. In [46], Matsuura presents a low-power pulse generator which operates with a discharge current of $0.15 \mu\text{A}$ at a voltage of 1.5 V. This current is at least a factor 1000 less than the discharge current the device operates at, which is in the order of mA. Therefore, the cost of switching using the fast round-robin scheduler can be neglected without introducing any significant error to the computed system lifetime.

6.4.2 Round robin frequency dependence

In order to find what switching frequency is efficient to approach the maximum lifetime, we investigate how the gain in lifetime depends on the switching frequency in case of round robin scheduling. The system of two batteries is discharged with a constant current of 1 A. We compare the system lifetime obtained with the round robin scheduler with that of sequential battery usage.

In Figure 6.6, we show the ratio of the system lifetime using round robin scheduling to the lifetime with sequential scheduling as a function of the round robin switching frequency. We see that the gain in lifetime of using the scheduler grows to a level of 1.89 when the switching frequency is increased. This level is the gain one would get with parallel discharge, which can be seen as switching with

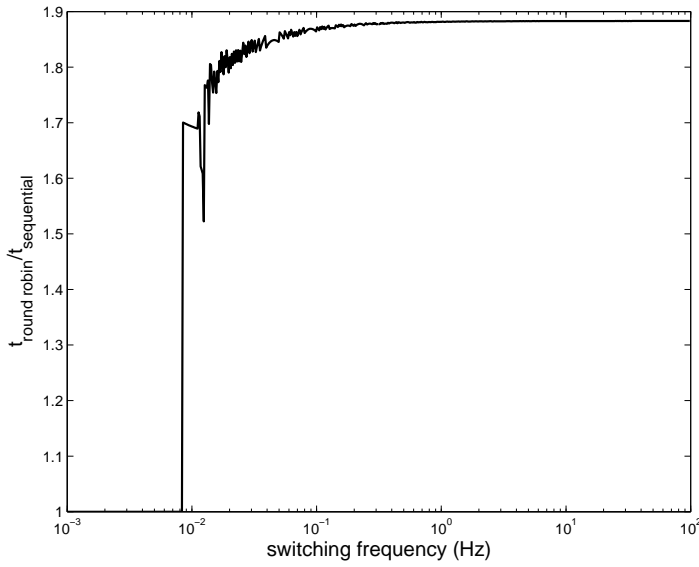


Figure 6.6 Gain of using a round robin scheduler compared to sequential usage as a function of the switching frequency.

infinite frequency. The figure shows that already for a switching frequency of 1 Hz the gain is close to optimal, so switching at higher frequencies is not necessary.

On the side of the low frequencies, smaller than 0.1 Hz, the graph fluctuates with clear downward tendency, that is, a small increase of the switching frequency may result in a considerable change in lifetime. This can be explained as follows. At very low frequency, lower than 0.008 Hz, the batteries are emptied in one period and the round robin scheduler results in sequential usage. When the frequency is increased the point will be reached where the first battery will not be emptied completely before the switch takes place. While the second battery is used the first can recover. Due to this recovery time the battery can be used longer, and the system lifetime is increased. This results in the first jump in the graph. Every time the batteries can recover for one period more a next jump in the graph occurs. The size of the jumps decreases as the frequency increases, since the extra recovery time will be shorter at higher frequencies. Between the jumps the system battery lifetime decreases, since the extra recovery time decreases as the switching frequency is increased. Thus, the ratio between on and off time will decrease until the next jump occurs.

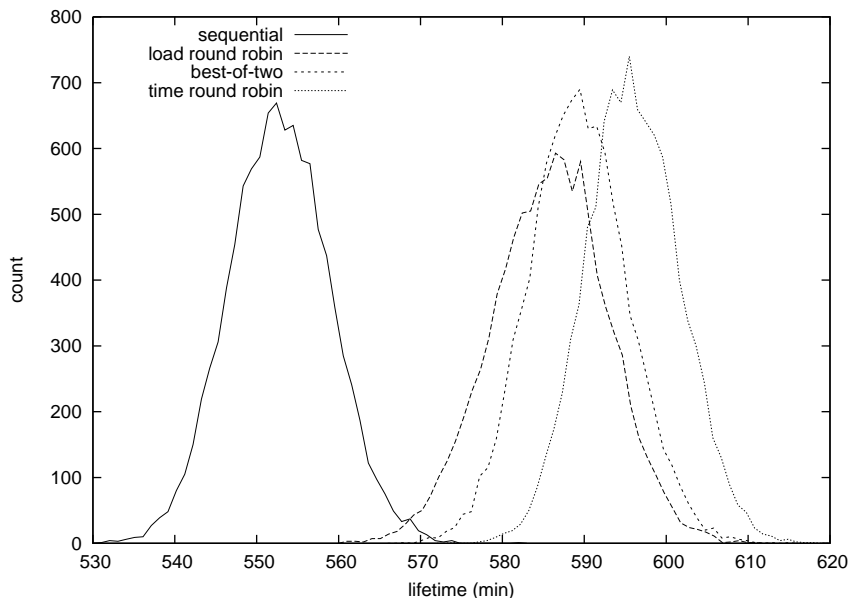


Figure 6.7 Empirical lifetime distributions generated with 10000 on-off loads with random on-times.

6.4.3 Random times

As first random load, we take an on-off load with 250 mA on-current. The off periods last 1 minute, and the on periods are uniformly distributed over the interval $[\frac{1}{2}, \frac{3}{2}]$ minute. This load has also been used in Section 5.6, but there the modelled batteries had a capacity that was approximately 8 times smaller than the real capacity, which is used here.

We compute the system lifetime for 10000 randomly generated loads using the four schedulers mentioned in Section 6.4.1. In Figure 6.7 the empirical lifetime distributions according to the different schedulers is given. For clarity the histograms are plotted using lines. In Table 6.2 the mean and variance of the computed life-

scheduler	mean lifetime (min)	variance (min^2)
sequential	552.87	39.36
load-round-robin	585.90	50.39
best-of-two	589.33	37.44
time-round-robin	596.01	33.38

Table 6.2 Mean and variance of the lifetimes obtained with the different schedulers for the loads with random on-times.

scheduler	mean lifetime (min)	variance (min ²)
sequential	229.55	237.98
load-round-robin	266.12	206.73
best-of-two	270.10	195.44
time-round-robin	274.84	197.20

Table 6.3 Mean and variance of the lifetimes obtained with the different schedulers for the loads with random discharge currents.

times are given for the different schedulers. As can be observed, clear system lifetime improvement is obtained when battery scheduling is applied. On average the load-round-robin and best-of-two scheduler outperform sequential usage by 6% and 6.6% respectively. Also, the two schedulers perform only slightly worse than the time-round-robin scheduler.

When we compare these results with those in Chapter 5, in which a gain of 65% was observed, we see that the relative gain in lifetime obtained by battery scheduling is much less than for smaller batteries. This is related to the result in Section 6.1, where the maximum possible gain is given as a function of the discharge current, as follows. The mean of the discharge current of the used loads is 125 mA. This gives a ratio between the load and the battery capacity of $0.125 \text{ A}/0.666 \text{ Ah} = 0.1875 \text{ h}^{-1}$ for the real size battery. For the smaller battery used in Chapter 5 the ratio is $0.125 \text{ A}/0.0916 \text{ Ah} = 1.36 \text{ h}^{-1}$. Using the top x-axis in Figure 6.2 one sees that the ratio of 0.1875 h^{-1} allows for a gain of less than 10%, whereas the ratio of 1.36 h^{-1} is close to the peak value of a maximum possible gain of 90%.

6.4.4 Random currents

The second set of random loads is also used in Chapter 5. In this set of random loads every minute we uniformly choose the discharge current from the set $\{0, 100, 200, 300, 400, 500\}$ mA. The current will stay constant for one minute until the next current is picked. We use the same schedulers as in the previous section. The load-round-robin and best-of-two scheduler now make a scheduling decision every minute, when the new current is picked.

Again, 10000 loads were generated. The lifetime distributions for these loads are given in Figure 6.8, and the numbers for the mean and variance of the simulations are given in Table 6.3. The trend is similar to the previous random load. The best-of-two scheduler performs slightly better than the load-round-robin scheduler, and both perform close to the time-round-robin scheduler. The average improvements relative to the sequential scheduler are 16% and 18% for the load-round-robin and best-of-two, respectively. This is much better than for the loads with

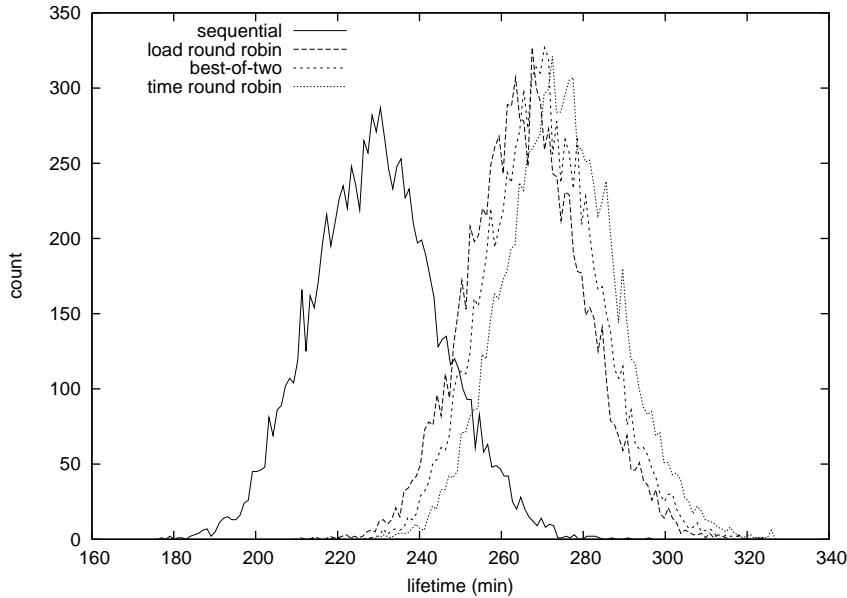


Figure 6.8 Empirical lifetime distributions generated with 10000 loads with random discharge current.

random on times due to the higher average discharge current. For the loads with random currents the average discharge current is 250 mA. In Figure 6.2 we can see that for a discharge current of 250 mA the maximum lifetime gain is just under 20% when the system is discharged with a continuous current of 250 mA. On the other hand, the maximum gain for the random on-times, which have an average discharge current of 125 mA, is approximately 10%.

Due to the higher variance in discharge current, the variance in lifetime is larger for this load, as visible through the “wider” graphs, and the numbers for the variance in Table 6.2 and 6.3.

When we compare the results with those of Chapter 5, we see that the difference in lifetime gain is not as large as with the previous set of random loads. The TA-KiBaM schedule resulted in a system lifetime that was 26% longer than the sequential schedule. For the smaller batteries used in Chapter 5, the ratio between the discharge current and the battery capacity is $0.250 \text{ A}/0.0916 \text{ Ah} = 2.73 \text{ h}^{-1}$. This ratio leads, according to Figure 6.2, to a maximum possible gain of approximately 26%, which is obtained by the TA-KiBaM scheduler.

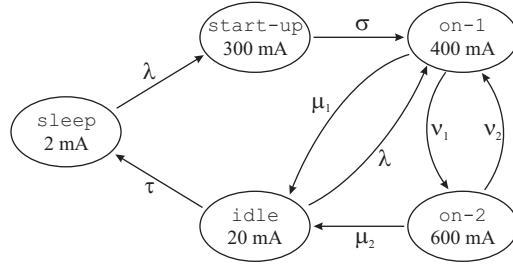


Figure 6.9 State transition diagram of the workload model.

transition	λ	σ	μ_1	ν_1	μ_2	ν_2	τ
rate (min^{-1})	$\frac{1}{5}$	2	$\frac{1}{14}$	$\frac{1}{14}$	$\frac{1}{25}$	$\frac{4}{25}$	$\frac{1}{2}$

Table 6.4 Transition rates of the Markov model.

6.4.5 Full random load

The final step in introducing randomness into the loads is having both random discharge times and random currents. This is done by using a Markov model that represents a simple workload of a device. The state transition diagram of this Markov model is given in Figure 6.9. The device has 5 different states: **sleep**, **start-up**, **on-1**, **on-2** and **idle**. In the **sleep** state the device draws a 2 mA current from the battery. From the **sleep** state the device first has to start-up before it can go to the **on-1** state. The start-up takes 30 seconds on average, and during start-up the discharge rate is 300 mA. From the **on-1** state a transition is made either to the **idle** state, or to the **on-2** state, both with probability $\frac{1}{2}$. In the **on-1** and **on-2** state the discharge current is 400 and 600 mA, respectively. The average residence time in the **on-1** and **on-2** state is 7 and 6 minutes, respectively. From the **on-2** with probability $\frac{4}{5}$ it will go back to **on-1**, and with probability $\frac{1}{5}$ go to **idle**. In the **idle** state the current is 20 mA, and the average time it takes to go back to **sleep** is 2 minutes. The used discharge currents are based on the average discharge currents for different modes of the Itsy pocket computer [55]. An overview of the transition rates is given in Table 6.4.

Again, we use the sequential, load-round-robin, best-of-two scheduler and time-round-robin to compute the system lifetime for 10000 randomly generated loads. The scheduling choices are made at the state changes.

In Figure 6.10 the empirical lifetime distributions for the Markov workload model are given. In Table 6.5 the mean and variance of the lifetimes are given. Again, we see the same order in performance of the four schedulers. However, the difference between the time-round-robin scheduler, and the best-of-two and load-round-robin scheduler is a lot larger. Even though the average improvement

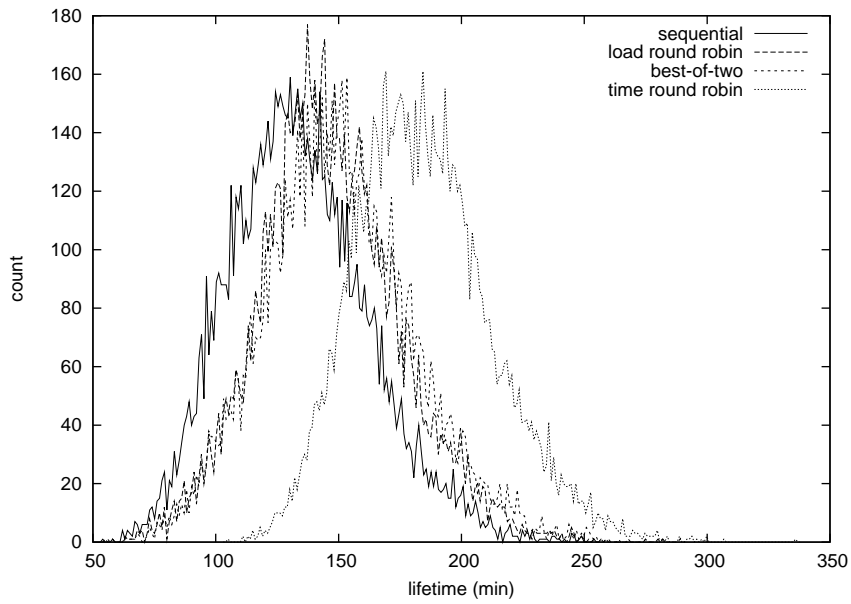


Figure 6.10 Empirical lifetime distributions generated with 10000 Markov model loads.

scheduler	mean lifetime (min)	variance (min ²)
sequential	133.72	794.12
load-round-robin	145.99	803.83
best-of-two	149.11	892.97
time-round-robin	183.61	775.06

Table 6.5 Mean and variance of the lifetimes obtained with the different schedulers for the Markov model loads.

compared to sequential discharge of the load-round-robin and best-of-two scheduler is 10.3% and 12.6%, respectively, the time-round-robin scheduler leads to an even longer lifetime. The time-round-robin scheduler outperforms sequential scheduling with 40.6% for this workload. The large difference between the time-round-robin scheduling and the round robin and best-of-two schedulers is caused by the longer average time between scheduling moments.

6.5 Discussion

In this chapter we show an analytic approach to the battery scheduling problem. This approach shows that when the moments of scheduling can be chosen freely, and the empty batteries that had time to recover some charge can still be used, the actual scheduling mechanism does not matter for the system lifetime, as long as one can switch fast enough between the batteries.

The analytic approach gives an easy way to compute the maximum possible lifetime gain one may obtain by battery scheduling in the case of continuous currents. This result can be used to get an indication for the potential lifetime gain for any load profile based on the average discharge current. This maximum gain clearly depends on the level of the discharge current. The optimal current, *i.e.*, where scheduling will lead to the largest lifetime improvements, depends, next to the battery parameters, on the ratio between the discharge current and battery capacity.

The benefit of the analytic approach over the priced timed automata model of Chapter 5 is that it can also be used for systems with larger and more batteries, where the priced timed automata approach falls short for scalability reasons. However, it is important to note that the analytical results obtained in this chapter only hold for the case that the batteries have the same KiBaM parameters k and c . In the case that batteries with different parameters are considered, as is done for example in [70], the addition of the differential equations to obtain Equation (6.3) is not possible. The priced-timed automata, on the other hand, can easily be adapted to model combinations of different types of batteries.

Finally, the simulation results show that also for more complex random loads, battery scheduling helps to improve the system lifetime considerably. The gain in lifetime compared to sequential discharge of the batteries for the different schedulers varies with the type of load. The average maximum lifetime gain can be predicted well by computing the maximum possible lifetime gain for a continuous discharge current using the average current of the random load.

Chapter 7

Concluding remarks

For the wireless devices used in modern society, a long lasting battery is essential. The nonlinear properties of batteries, the rate-capacity and recovery effect, suggest that battery lifetime can not only be extended by lowering the average discharge current, but also by changing how the load is distributed over time. In this thesis we investigated the influence of the workload on the battery lifetime by combining a battery model with different workload models in various ways.

The literature provides many different approaches to model batteries for doing lifetime predictions. We have provided an overview of the most important approaches that have been taken. Most of these models have been developed to be used in a specific application area. For example, the highly detailed electrochemical models are used in battery development, and the electrical circuit models are used in electrical engineering. These specialized models focus on the properties of interest of their users. This makes them overly complex, and hard to combine with workload models.

Other models focus on only one of the non-linear properties. On the one hand, Peukert's law only models the rate-capacity effect. For this model only the average load has an impact on the battery lifetime. On the other hand, the stochastic model by Chiasserini focuses on the recovery effect, ignoring the rate-capacity effect. These models are too limited, since they fail to take into account an essential part of the battery behavior.

The analytical diffusion model and kinetic battery model are best suited to be used in combination with workload models. These models focus on the externally visible battery properties, rate-capacity and recovery effect, and are based on only two differential equations, which still makes the combined model manageable.

We have shown that the two analytical models are closely related, the kinetic battery model being a first order approximation of the diffusion model. The detailed analysis of the analytic battery models in Chapter 3 shows that the frequency response of the battery is very low for frequencies higher than 10^{-2} Hz. There-

fore, rearranging the load on a sub-minute time scale will not impact the lifetime. The processes in the battery are too slow to keep up with the changing current. This means that the battery will only “see” the average current. Only when the average current is lowered by the sub-minute load change the battery lifetime will be extended. However, by rearranging the load on a coarser time scale one may take advantage of the recovery effect and improve the lifetime even if the average load does not change.

The comparison between the two analytical models shows that these are closely related. The kinetic battery model is a first-order approximation of the diffusion model. By properly choosing the parameters of the kinetic battery model a good approximation can be made of the diffusion model, which leads to similar computed battery lifetimes for realistic loads. Therefore, we chose to use the simpler kinetic battery model to model the battery.

We have used the kinetic battery model in combination with other models to investigate the impact of the workload on the battery lifetime. In Chapter 4 we have combined a stochastic workload, modeled by a continuous-time Markov chain, with the battery model. This results in an inhomogeneous Markov reward model with two rewards, in which the reward rates depend on the levels of the accumulated rewards. For this model we have developed new algorithms to compute the distribution and the expected value of both the battery lifetime and the charge delivered by the battery.

The accuracy of the algorithms for computing the distributions varies with the applied load. It is rather poor in the case that the battery lifetime is nearly deterministic. In contrast, the algorithms give good results in the case that the lifetime has a high variance. More work is necessary to investigate the exact conditions under which the used approximation leads to accurate results. Indeed, we see that the distributions and expected value of the battery lifetime is influenced by the workload, and the battery lifetime does not depend only on the average current. However, we also have seen some contradicting results, where the workload that leads to the longer average lifetime also leads to less charge delivered by the battery.

For systems with one battery, the workload pattern is not so easily influenced, especially when the workload is highly dependent on the users behavior. The change of workload should not influence the system performance perceived by the user, which limits the options to delay tasks in order to improve battery lifetime. In a system which is powered by more batteries one can freely change the workloads of the separate batteries by using battery scheduling.

We have taken two approaches to find the optimal battery schedule in a system powered by two batteries. In the first, the kinetic battery model is translated to a priced timed automata model. The optimal schedules are obtained through model checking techniques. In this approach, the moments at which a scheduling decision is made need to be predefined. Due to the computational complexity, the

optimal schedules can only be obtained for a small number of scheduling points, and therefore only for small batteries. Still, the results indicate that battery scheduling (potentially) leads to big lifetime improvements.

In the second, analytic approach no limitations to the scheduling moments are given, *i.e.*, the battery scheduler may switch between the batteries at any point in time. The question now is to determine the best moments to switch between the batteries. The analysis shows that the scheduling scheme does not impact the system lifetime as long as the scheduler can switch at a high rate between the batteries. The optimal lifetime is actually obtained when the batteries are discharged in parallel. This result implies that the rate-capacity effect is more important than the recovery effect, because with parallel discharge neither battery will have any additional idle time during the discharge. Although parallel usage of the batteries may not be possible for all types of batteries, one should not connect the batteries in parallel directly, one can approach parallel usage by switching at a fast rate between the batteries. As stated before, when doing so, the battery cannot keep up with the fast switching and will act as if it sees the average current, which is the same as for parallel discharge.

We have applied the battery scheduling to stochastic workloads by means of simulation. The actual lifetime gain depends on the type of workload. However, one can predict the average maximum lifetime gain by computing the maximum possible gain for a continuous current at the level of the average discharge current of the random load.

Further work

The results presented in this thesis are based on the kinetic battery model. Although this model has been validated with the highly detailed `Dualfoil` model in Chapter 5, experimental validation of the results is an important issue. In Appendix A results of a first investigation of the practical efficiency of battery scheduling are given. The results so far are inconclusive, no clear gain in system lifetime has been observed. The high variance in battery capacity that is introduced in the experiments makes more measurements necessary.

One of the possible application areas of the battery scheduling is battery-aware routing in wireless sensor networks, as presented in Section 5.1. Although the results in Chapter 6 indicate that high frequency switching leads to an optimal system lifetime, this is not a good option in wireless sensor networks. Fast switching between different routes in the network will lead to high overhead and extra loss of battery capacity. In this case slow switching is better. This still leads to longer system lifetimes. In this case the recovery effect will play a more important role. Due to the slow switching the batteries may be depleted further and timely recovery is essential. Here, further work is necessary, where the cost of switching

Concluding remarks

between the batteries is taken into account as well.

Also, it would be interesting to extend the battery model to allow for recharging of the batteries. This extension can help in answering questions like: “At what frequency, and how long does one need to recharge in order for the battery never to be empty?”, or reversely “What size should the battery be if the recharge periods are given?”. Especially the latter question is interesting for sensor nodes that are recharged using solar cells. These batteries can recharge during the day, but need to have enough capacity to survive during the night.

Appendix A

A first investigation of the practical efficiency of battery scheduling

Next to the analytical studies and simulations presented in this thesis, some explorative experiments were performed to investigate whether the theoretical gain also is practically achievable. These experiments were performed by Damien Miliche at Thales Nederland B.V., Huizen. This appendix is based on [49]:

- D. Miliche, M. de Graaf, G. Hoekstra, M. Jongerden, and B. Haverkort. A first experimental investigation of the practical efficiency of battery scheduling. In *Workshop Proceedings of the 23th International Conference on Architecture of Computing Systems (ARCS '10)*, pages 241–246, 2010.

A.1 Experimental setup

The most important contribution of battery scheduling is supposed to be the extension of the lifetime, which is related to the capacity (in Ah), or the amount of charge, that the battery can deliver before getting empty. So, in order to measure the effect of battery scheduling it is necessary to measure the provided capacity. To do so, different devices are available, including the Amp-hour meter, the Watt-hour meter, or the Coulomb counter. For practical reasons, however, an indirect way has been chosen: the capacity is computed indirectly from the measurement of the current signal. Actually, the capacity is nothing else than the integration over time of the current.

Figure A.1 shows a schematic picture of the used setup. The actual setup is illustrated in Figure A.2. The setup consists of:

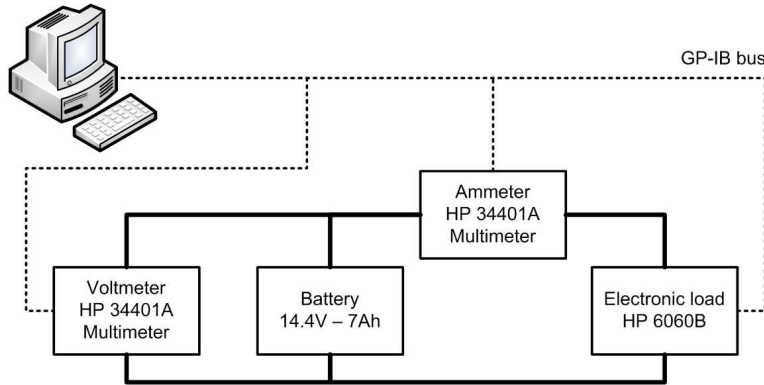


Figure A.1 Schematic overview of the measurement setup [49].

- an electronic load (model HP6060B)
- two multimeters (model HP34401A)
- a battery, one of the types described in Section A.2
- a desktop computer

The electronic load is used to create the load for discharging the battery in a well-controlled manner. During the discharge, both the discharge current and the output voltage of the battery are monitored with the two multimeters, one used as ampere meter and the other as voltmeter. The voltmeter is used to ensure that the battery voltage does not drop too low, which could damage the battery. The battery is considered empty when its cut-off voltage is reached.

The multimeters and the electronic load are connected to the computer via a GP-IB bus. This allows us to obtain and log the measured values, and also to automatically configure the devices. A simple C program is used to monitor and log the discharge process, looping until the battery is discharged, *i.e.*, when the cut-off voltage of the battery is reached.

A.2 The batteries

In the experiments two types of batteries have been used. The first type is based on the Li-ion technology. The batteries have a nominal capacity of 7 Ah, and a nominal voltage of 14.4 V. The battery is a so-called smart battery, *i.e.*, some electronics have been added to the battery pack to control the charging and discharging, so-called cell balancing, and to provide an I²C-like interface (SMBus) for the system. With Li-ion batteries, attention has normally to be paid to the

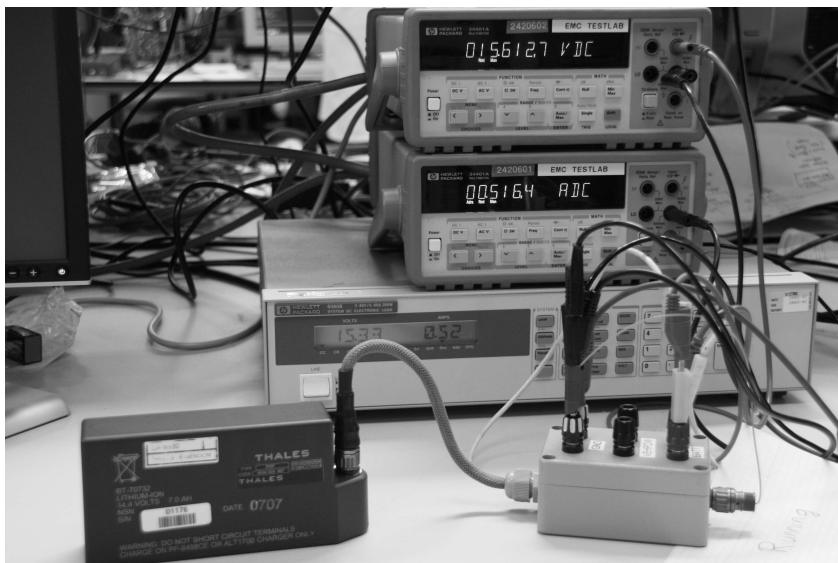


Figure A.2 Measurement setup [49].

charge and the discharge process in order not to damage the battery. For the used battery this is managed by the embedded electronics and an appropriate charger.

The embedded electronics are used to monitor the state-of-charge of the battery. When the battery is detected as empty, *i.e.*, when the cut-off voltage of 12 V is reached, the battery cells are actually disconnected from the battery pack output. For the charging process, a smart charger is used, which communicates with the electronics of the battery pack, and stops charging when the cells are determined as full.

These batteries are used in the Thales personal communication system, called CIM (for Communication and Information Module) [63]. This system aims to centralize the data for a radio, a GPS, a display, and other devices constituting the useful equipment for any kind of urgent or dangerous interventions done by for example police, fire fighters or army. The system is powered by two independent batteries, which so far are discharged in a sequential manner, *i.e.*, the second is only used when the first is completely discharged. The motivation for using the batteries sequentially is mainly operational. It should be avoided that the personal communication system runs out of power with no backup battery available.

The second type of battery used for the experiments is a NiCd battery as is commonly used in remote control cars. These batteries are characterized by a nominal capacity of 1.6 Ah under a nominal voltage of 7.2 V.

The NiCd batteries do not have any kind of smart management, neither within the battery pack, nor within the charger. All the management has to be done

manually. Moreover this kind of battery can suffer from a memory effect, if the charge-discharge cycles are not done completely. Indeed, due to crystallization of the electrodes, this effect causes the battery to deliver only the capacity used during the preceding charge/discharge cycles.

We define the battery empty when its voltage, measured by the voltmeter, drops below the cut-off voltage of 6 V. The full-charged state is considered to be reached when the battery is heating (due to chemical reaction at the end of the charging process), and when the voltage stabilizes around 8.8 V.

A.3 Discharge measurements

Three measurement series have been performed with both battery types.

The first series of measurements consists of discharging the battery with a constant current. During the experiment, the voltage and the current are sampled with fixed intervals. In the end, we obtain the discharge time and can compute the provided capacity by numerical integration of the discharge current over the time. This first series aims to give us a reference point of the delivered charge without scheduling, to which we can compare the results of the other two series.

In the second and third series of measurements, we mimic the scheduling using a single battery. The purpose is to observe the impact of scheduling, and not to implement a real scheduling system. The real implementation can indeed be relatively complex, requiring a circuit to manage the switching between the batteries, and an algorithm to determine the switching sequence. Instead, we just look at the behavior of one battery in a two-battery system, where a battery is used half of the time. The scheduling algorithm we use is a fast switching round robin schedule. Two different frequencies are used: 0.1 Hz and 1 Hz, thus switching every 5 s and 0.5 s. According to the theoretical results, one expects a significant extension of lifetime compared to sequential scheduling for both frequencies. The alternating battery selection is simulated by turning on and off the electronic load. In the same way as with the first series, the provided capacity and the discharge time are computed.

A.4 Results

For both the Li-ion and the NiCd battery type, two batteries have been used. The described measurement series have been performed once with every battery. In the first series of measurements for the Li-ion batteries, the battery was discharged at a constant current of 250 mA, 500 mA, 750 mA, 1 A, 1.5 A, 2 A and 2.5 A. While the experiments are time-consuming, the other measurement series are performed with a subset of these currents. The discharge currents used in the other measurement series, both for the Li-ion and NiCd batteries, were 500 mA, 1 A or 2 A.

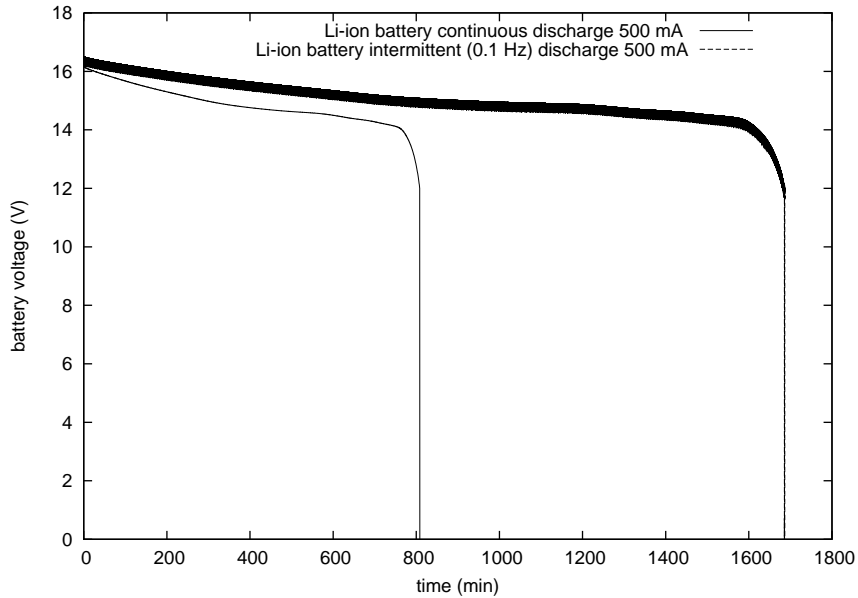


Figure A.3 Evolution of the battery voltage during the discharge.

Figure A.3 shows the measured battery voltage of a Li-ion battery as a function of the time for a continuous and an intermittent load with 500 mA discharge current. The intermittent load has a frequency of 0.1 Hz. In the figure one can see that, in this example, the intermittent discharge leads to a lifetime that is more than twice as long as the lifetime for the continuous discharge.

The figure also shows how the battery voltage slowly drops during the discharge until the cut-off voltage of 12 V is reached. At this point the embedded electronics disconnect the cells from the output, which makes the voltage drop to zero. The voltage of the battery decreases at a lower rate when the battery is discharged with an intermittent load. When we zoom into the curve, we can actually see that there is a small increase of the voltage during the idle periods. This is shown in Figure A.4. Here we can see the recovery effect, *i.e.*, during the short idle periods the output voltage recovers slightly.

From the first series of measurements also the rate capacity effect can be shown. Figure A.5 and Figure A.6 show the delivered charge as a function of the discharge current for the NiCd and Li-ion batteries, respectively. For both of the NiCd batteries a clear drop in delivered charge is observed when the discharge current is increased (cf. Figure A.5). The delivered capacity at a current of 2 A is approximately 18% lower than the delivered charge at 0.5 A.

The rate capacity effect is less visible for the Li-ion batteries (cf. Figure A.6). For one of the batteries, Li-ion 2, the delivered charge drops with approximately

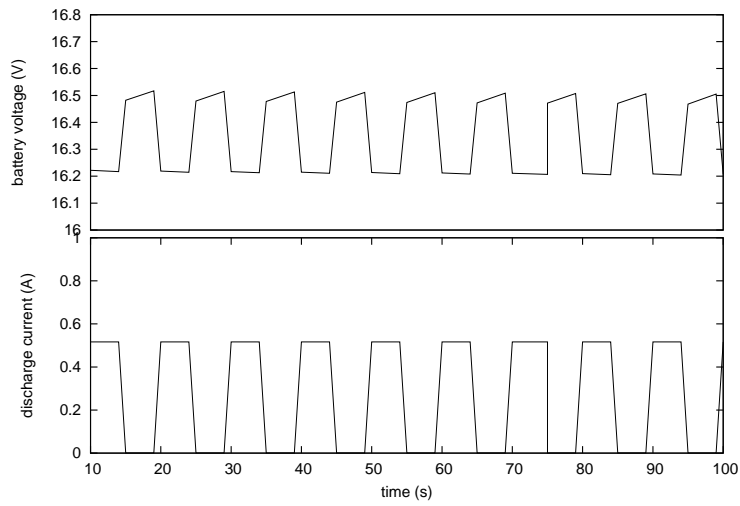


Figure A.4 Top: zoom in of the evolution of the battery voltage for the intermittent discharge. Bottom: the discharge current as a function of the time.

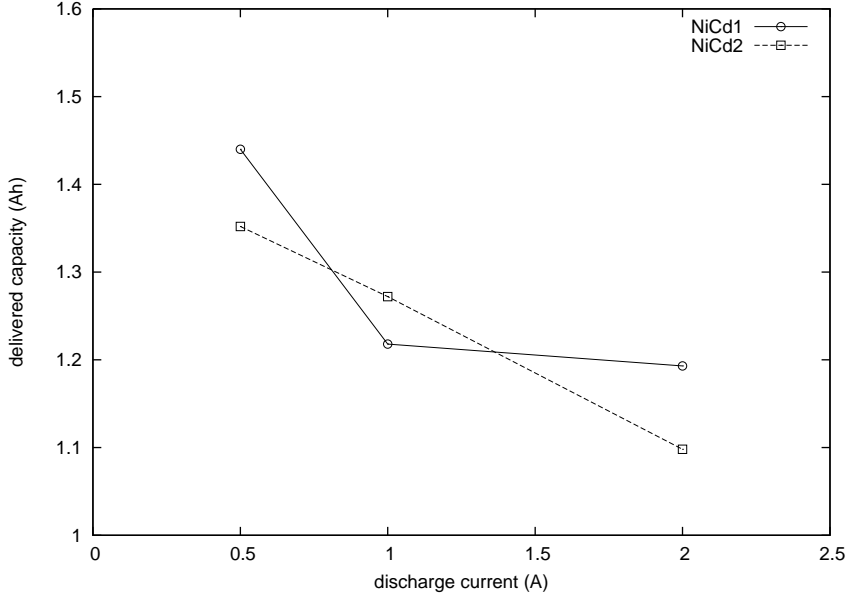


Figure A.5 Rate capacity effect for the NiCd batteries, as in [49].

10% when the discharge current is increased from 1 A to 2.5 A. However, for the other Li-ion battery the delivered charge is lower only for the 2 A discharge current. For all the other discharge currents the delivered charge is approximately 7 Ah.

A first explanation of the difference between the results of the two battery types can be the difference in nominal capacity between the two battery types. The increase of the discharge current will have a larger impact on the NiCd batteries which have a lower capacity. Another explanation can be found in the fact that the Li-ion batteries are enriched with some electronics. The cell-balancing performed by the electronics controls the discharge of the separate battery cells, and may result in less loss of capacity when the discharge current is increased.

Table A.1 shows the relative lifetime extension for both types of battery when using scheduling. Both scheduling frequencies of 0.1 Hz and 1 Hz have been used for the NiCd battery and for the Li-Ion battery. One can see, for the NiCd batteries, that the gain varies between -8.2% and +7.3% with the 1 Hz scheduling, and between -5.3% and +5.3% with the 0.1 Hz scheduling. For the Li-Ion batteries, the gain varies between -13.9% and +9% and between -15.9% and +21% respectively.

The results are not as expected. In the given set of experiments some anomalies were observed. Scheduling does not always improve the lifetime, and there is no clear relation with the discharge current. One explanation might be found in the fact that it is difficult to do the measurements under exactly the same conditions.

battery type	discharge current (A)	continuous discharge	intermittent discharge 1 Hz		intermittent discharge 0.1 Hz	
		delivered charge (mAh)	delivered charge (mAh)	$\Delta\%$	delivered charge (mAh)	$\Delta\%$
NiCd 1	0.5	1440	1470	2.1	1457	1.2
	1	1218	1261	3.5	1283	5.3
	2	1193	1095	-8.2	1164	-2.4
NiCd 2	0.5	1352	1451	7.3	1420	5.0
	1	1272	1222	-3.9	1205	-5.3
	2	1098	1114	1.5	1086	-1.1
Li-ion 1	0.5	6957	7080	1.8	7184	3.3
	1	6874	6418	-6.6	6649	-3.3
	2	6005	6546	9.0	7265	21.0
Li-ion 2	0.5	6898	5942	-13.9	6547	-5.1
	1	6922	6003	-13.3	5824	-15.9
	2	6335	6868	8.4	5927	-6.4

Table A.1 Delivered charge measured for two NiCd batteries and two Li-ion batteries with continuous and intermittent discharge [49].

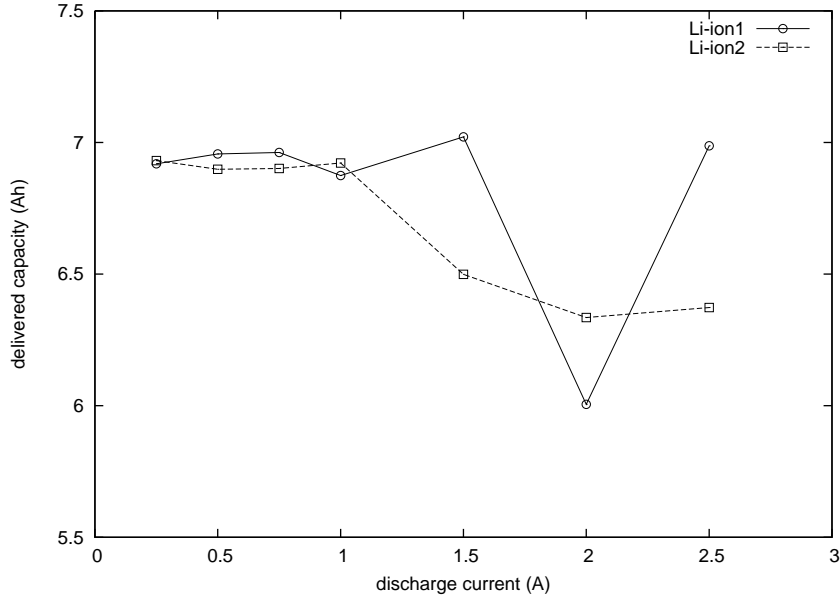


Figure A.6 Rate capacity effect for the Li-ion batteries, as in [49].

The state of the battery before each discharge is hardly ever precisely the same, as the battery ages and thus its parameters vary. Nevertheless, more measurements should be performed in order to obtain better statistics in the results, canceling the measurement errors and the parameter variability.

A.5 Discussion

Although the simulations show that, also with real sized batteries, battery scheduling does lead to a considerable gain in lifetime, this is not seen so clearly in the experiments. For some of the experiments the lifetime with scheduling is longer than without. However, no structural gain can be observed. The most important reason for these results is, most probably, the uncertainty in the state-of-charge of the full and empty batteries. One can not tell whether the battery is recharged to the same level before every experiment, and discharged to the same depth. To remove this uncertainty many more experiments have to be done. However, this will also bring an extra uncertainty into the experiments, since a battery may slowly degrade after many discharge-charge cycles. Also, variation in capacity between different batteries may be large.

Bibliography

- [1] R. Alur, C. Courcoubetis, and D. Dill. Model-checking in dense real-time. *Information and Computation*, 104(1):2–34, May 1993.
- [2] R. Alur, S. L. Torre, and G. J. Pappas. Optimal paths in weighted timed automata. In *Proceedings of Hybrid Systems: Computation and Control (HSCC '01)*, volume 2034 of *LNCS*, pages 49–62. Springer, 2001.
- [3] Battery University.com. <http://www.batteryuniversity.com/partone-24.htm>, April 2010.
- [4] G. Behrmann, A. David, and K. Larsen. A tutorial on uppaal. In *Formal Methods for the Design of Real-Time Systems*, volume 3185 of *LNCS*, pages 200–236. Springer, 2004.
- [5] G. Behrmann, A. Fehnker, T. Hune, K. Larsen, P. Pettersson, J. Romijn, and F. Vaandrager. Minimum-cost reachability for priced time automata. In *Proceedings of Hybrid Systems: Computation and Control (HSCC '01)*, volume 2034 of *LNCS*, pages 147–161. Springer, 2001.
- [6] G. Behrmann, K. G. Larsen, and J. I. Rasmussen. Optimal scheduling using priced timed automata. *ACM SIGMETRICS Performance Evaluation Review*, 32(4):34–40, March 2005.
- [7] L. Benini, G. Castelli, A. Macii, E. Macii, M. Poncino, and R. Scarsi. Extending lifetime of portable systems by battery scheduling. In *Proceedings of Design, Automation and Test in Europe (DATE '01)*, pages 197–203. IEEE Computer Society Press, 2001.
- [8] J. Berendsen, D. N. Jansen, and J.-P. Katoen. Probably on time and within budget – on reachability in priced probabilistic timed automata. In *Proceedings of the International Conference on Quantitative Evaluation of Systems (QEST '06)*, pages 311–322. IEEE Computer Society Press, 2006.

BIBLIOGRAPHY

- [9] J. Berendsen, D. N. Jansen, and F. Vaandrager. Fortuna: Model checking priced probabilistic timed automata. In *Proceedings of International Conference on the Quantitative Evaluation of SysTems (QEST '10)*, pages 273–281. IEEE Computer Society Press, 2010.
- [10] A. Bobbio and L. Roberti. Distribution of the minimal completion time of parallel tasks in multi-reward semi-Markov models. *Performance Evaluation*, 14(3-4):239–256, February 1992.
- [11] CBS. *De digitale economie 2009*. 2009.
- [12] Y. Chen, F. Xia, D. Shang, and A. Yakovlev. Fine-grain stochastic modelling of dynamic power management policies and analysis of their power - latency tradeoffs. *IET Software*, 3(6):458–469, December 2009.
- [13] C. Chiasserini and R. Rao. A model for battery pulsed discharge with recovery effect. In *Proceedings of Wireless Communications and Networking Conference (WCNC)*, pages 636 – 639. IEEE Press, 1999.
- [14] C. Chiasserini and R. Rao. Pulsed battery discharge in communication devices. In *Proceedings of the 5th International Conference on Mobile Computing and Networking (MOBICOM)*, pages 88 – 95, 1999.
- [15] C. Chiasserini and R. Rao. Energy efficient battery management. *IEEE Journal on Selected Areas in Communications*, 19(7):1235 – 1245, July 2001.
- [16] C. Chiasserini and R. Rao. Improving battery performance by using traffic shaping techniques. *IEEE Journal on Selected Areas in Communications*, 19(7):1385 – 1394, 2001.
- [17] L. Cloth. *Model Checking Algorithms for Markov Reward Models*. PhD thesis, University of Twente, 2006.
- [18] L. Cloth, B. R. Haverkort, and M. R. Jongerden. Computing battery lifetime distributions. In *Proceedings of the 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN '07)*, pages 780–789. IEEE Computer Society Press, 2007.
- [19] P. De Vidts and R. E. White. Mathematical modeling of a nickel-cadmium cell: Proton diffusion in the nickel electrode. *Journal of the Electrochemical Society*, 142(5):1509–1519, May 1995.
- [20] M. Doyle, T. F. Fuller, and J. Newman. Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell. *Journal of the Electrochemical Society*, 140(6):1526 – 1533, June 1993.
- [21] e-Energy 2010 webpage. <http://www.e-energy.uni-passau.de/>, August 2010.

-
- [22] FORTRAN Programs for the Simulation of Electrochemical Systems. <http://www.cchem.berkeley.edu/jsngrp/fortran.html>, June 2010.
- [23] T. F. Fuller, M. Doyle, and J. Newman. Relaxation phenomena in lithium-ion-insertion cells. *Journal of the Electrochemical Society*, 141(4):982 – 990, April 1994.
- [24] T. F. Fuller, M. Doyle, and J. Newman. Simulation and optimization of the dual lithium ion insertion cell. *Journal of the Electrochemical Society*, 141(1):1 – 10, January 1994.
- [25] S. Gold. A PSPICE macromodel for lithium-ion batteries. In *Proceedings of the 12th Annual Battery Conference on Applications and Advances*, pages 215–222, 1997.
- [26] S. C. Hageman. Simple PSpice models let you simulate common battery types. *Electronic Design News*, 38(22):117 – 129, 1993.
- [27] B. R. Haverkort. *Performance of Computer Communication Systems. A Model-Based Approach*. John Wiley & Sons, 1998.
- [28] B. R. Haverkort, L. Cloth, H. Hermanns, J.-P. Katoen, and C. Baier. Model checking performability properties. In *Proceedings of the International Conference on Dependable Systems and Networks (DSN'02)*, pages 102–112. IEEE Press, 2002.
- [29] B. R. Haverkort, H. Hermanns, J.-P. Katoen, and C. Baier. Model checking CSRL-specified performability properties. In *Proceedings of the 5th International Workshop on Performability Modeling of Computer and Communications Systems (PMCCS'01)*, pages 105–109, 2001.
- [30] G. Horton, V. G. Kulkarni, D. M. Nicol, and K. S. Trivedi. Fluid stochastic Petri nets: Theory, applications, and solution techniques. *European Journal of Operational Research*, 105(1):184–201, February 1998.
- [31] A. Horvath and M. Gribaudo. Matrix geometric solution of fluid stochastic Petri nets. In *Proceedings of the 4th International Conference on Matrix Analytic Methods in Stochastic Models*. World Scientific, 2002.
- [32] R. Howard. *Dynamic Probabilistic Systems; Volume II: Semi-Markov and decision processes*. John Wiley & Sons, 1971.
- [33] ICGreen 2010 webpage. <http://www.icgreen.innov.org/>, August 2010.
- [34] *IEEE Computer*, volume 38. IEEE Press, 2005.

BIBLIOGRAPHY

- [35] M. Jongerden and B. Haverkort. Computing lifetimes for battery-powered devices. *extended abstract accepted for post-conference proceedings of the International Conference on Operations Research, Munich*, 2010.
- [36] M. Jongerden, A. Mereacre, H. Bohnenkamp, B. Haverkort, and J.-P. Katoen. Computing optimal schedules for battery usage in embedded systems. *IEEE Transactions on Industrial Informatics*, 6(3):276–286, August 2010.
- [37] M. R. Jongerden and B. R. Haverkort. Battery modeling. Technical Report TR-CTIT-08-01, Centre for Telematics and Information Technology, University of Twente, 2008.
- [38] M. R. Jongerden and B. R. Haverkort. Which battery model to use? In *Proceedings of the 24th UK Performance Engineering Workshop (UKPEW)*, Technical Report Series of the Department of Computing, Imperial College London, pages 76–88, 2008.
- [39] M. R. Jongerden and B. R. Haverkort. Which battery model to use? *IET Software*, 3(6):445–457, December 2009.
- [40] M. R. Jongerden, B. R. Haverkort, H. C. Bohnenkamp, and J.-P. Katoen. Maximizing system lifetime by battery scheduling. In *Proceedings of the 39th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2009)*, pages 63–72. IEEE Computer Society Press, 2009.
- [41] Low Cost Batteries.com website. <http://www.lowcostbatteries.com/articles.asp?id=107>, April 2010.
- [42] J. Manwell and J. McGowan. Lead acid battery storage model for hybrid energy systems. *Solar Energy*, 50(5):399–405, 1993.
- [43] J. Manwell and J. McGowan. Extension of the kinetic battery model for wind/hybrid power systems. In *Proceedings of the 5th European Wind Energy Association Conference (EWEC '94)*, pages 284–289, 1994.
- [44] J. Manwell, J. McGowan, E. Baring-Gould, W. Stein, and A. Leotta. Evaluation of battery models for wind/hybrid power system simulation. In *Proceedings of the 5th European Wind Energy Association Conference (EWEC '94)*, pages 1182–1187, 1994.
- [45] T. L. Martin. *Balancing Batteries, Power, and Performance: System Issues in CPU Speed-Setting for Mobile Computing*. PhD thesis, Carnegie Mellon University, 1999.
- [46] Y. Matsuura. Low-power consumption reference pulse generator. *United States Patent 4,618,837*, 1986.

-
- [47] J. F. Meyer. On evaluating the performability of degradable computing systems. *IEEE Transactions on Computers*, 29(8):720–731, 1980.
- [48] J. F. Meyer. Performability: a retrospective and some pointers to the future. *Performance Evaluation*, 14(3):139–156, 1992.
- [49] D. Miliche, M. de Graaf, G. Hoekstra, M. Jongerden, and B. Haverkort. A first experimental investigation of the practical efficiency of battery scheduling. In *Workshop Proceedings of the 23th International Conference on Architecture of Computing Systems (ARCS '10)*, pages 241–246, 2010.
- [50] M. Neuts. *Matrix-Geometric Solutions in Stochastic Models. An Algorithmic Approach*. Dover Publications, 1994.
- [51] Panasonic: Overview of Lithium Ion Batteries. http://www.panasonic.com/industrial/includes/pdf/panasonic_liion_overview.pdf, June 2010.
- [52] E. Podlaha and H. Cheh. Modeling of cylindrical alkaline cells. *Journal of the Electrochemical Society*, 141(1):15–27, January 1994.
- [53] D. Rakhmatov and S. Vrudhula. An analytical high-level battery model for use in energy management of portable electronic systems. In *Proceedings of the International Conference on Computer Aided Design (ICCAD'01)*, pages 488–493, 2001.
- [54] D. Rakhmatov, S. Vrudhula, and D. A. Wallach. Battery lifetime predictions for energy-aware computing. In *Proceedings of the 2002 International Symposium on Low Power Electronics and Design (ISLPED '02)*, pages 154–159, 2002.
- [55] D. Rakhmatov, S. Vrudhula, and D. A. Wallach. A model for battery lifetime analysis for organizing applications on a pocket computer. *IEEE Transactions on VLSI Systems*, 11(6):1019–1030, 2003.
- [56] R. Rao, S. B. K. Vrudhula, and N. Chang. Battery optimization vs energy optimization: which to choose and when? In *Proceedings of the International Conference on Computer Aided Design (ICCAD'05)*, pages 439–445. IEEE Computer Society, 2005.
- [57] V. Rao, G. Singhal, A. Kumar, and N. Navet. 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. IEEE Computer Society, 2005.
- [58] A. Rindos, S. Woollet, I. Viniotis, and K. Trivedi. Exact methods for the transient analysis of nonhomogeneous continuous time Markov chains. In

BIBLIOGRAPHY

- Proceedings of the 2nd International Workshop on the Numerical Solution of Markov Chains*, pages 121–133, 1995.
- [59] S. Sarkar and M. Adamou. A framework for optimal battery management for wireless nodes. *IEEE Journal on Selected Areas in Communications*, 21(2):179–188, 2003.
- [60] T. Simunic, L. Benini, P. Glynn, and G. De Micheli. Dynamic power management for portable systems. In *Proceedings of the 6th annual international conference on Mobile computing and networking (MOBICOM '00)*, pages 11–19. ACM, 2000.
- [61] W. J. Stewart. *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press, 1994.
- [62] M. Telek, A. Horváth, and G. Horváth. Analysis of inhomogeneous Markov reward models. *Linear Algebra and its Applications*, 386:383–405, 2004.
- [63] Thales soldier systems. http://www.thalesgroup.com/Portfolio/Defence/Soldier_Systems/?pid=1568, September 2010.
- [64] The Spice Page. <http://bwrc.eecs.berkeley.edu/classes/icbook/spice/>, June 2010.
- [65] UPPAAL Cora webpage. www.cs.aau.dk/~behrmann/cora/index.html, July 2010.
- [66] UPPAAL webpage. www.uppaal.com, July 2010.
- [67] N. M. van Dijk. Uniformisation for nonhomogeneous Markov chains. *Operations Research Letters*, 12:283–291, November 1992.
- [68] A. van Moorsel and K. Wolter. Numerical solution of non-homogeneous Markov processes through uniformisation. In *Proceedings of the 12th European Simulation Multiconference*, pages 710–717, 1998.
- [69] Wolfram Mathworld Lambert-W Function. <http://mathworld.wolfram.com/lambertw-function.html>, April 2010.
- [70] Q. Wu, Q. Qiu, and M. Pedram. An interleaved dual-battery power supply for battery-operated electronics. In *Proceedings of the Conference on Asia South Pacific Design Automation (ASP-DAC '00)*, pages 387–390. ACM, 2000.

Samenvatting

Tegenwoordig bezit vrijwel iedereen meerdere apparaten die op batterijen werken, zoals een mobiele telefoon, PDA of laptop. Volgens het CBS is het aantal mobiele telefoons in Nederland tussen 2004 en 2008 gegroeid van 16 miljoen naar 19.7 miljoen, en het percentage huishoudens dat een laptop bezit is gegroeid van 27% naar 62%.

Een belangrijk aspect met betrekking tot de gebruikersvriendelijkheid van al deze apparaten is dat de hoeveelheid energie in de batterijen van de apparaten beperkt is. Hoe lang je een apparaat kan gebruiken wordt bepaald door de levensduur van de batterij. In dit proefschrift is de levensduur van de batterij de tijd die het kost tot een volle batterij niet voldoende energie meer kan leveren om het apparaat te laten werken; hierbij wordt de batterij tussendoor niet opgeladen. Voor een ideale batterij hangt de levensduur alleen af van de capaciteit van de batterij en het niveau van de ontladstroom waarmee de batterij belast wordt. De levensduur zal langer zijn wanneer de capaciteit groter is, of de ontladstroom lager is. Een ideale batterij zal onafhankelijk van de ontladstroom altijd al zijn energie kunnen leveren aan het apparaat. In werkelijkheid is dit echter niet het geval. Wanneer een echte batterij met een constante stroom ontladen wordt, zal deze bij een hoge stroom minder energie kunnen leveren tot hij leeg is dan bij een lage ontladstroom. Dit effect wordt het *rate-capacity* effect genoemd. Dit effect kan (deels) ongedaan gemaakt worden door na een periode van hoge belasting de batterij een tijdje niet te belasten. Tijdens zo'n rustperiode kan de batterij herstellen en kan de energie die verloren leek gegaan toch nog door de batterij geleverd worden. Dit effect wordt het *recovery* effect genoemd. Deze twee effecten zorgen ervoor dat de levensduur van de batterij niet alleen door het gemiddelde belastingsniveau wordt bepaald, maar ook door het belastingsprofiel. De vraag is nu, hoe het belastingsprofiel moet worden aangepast zodat de levensduur van de batterij, en dus het gebruiksgemak van het apparaat, het langst is?

Voor het beantwoorden van deze vraag maken we gebruik van wiskundige modellen. We hebben batterijmodellen nodig die de belangrijkste processen in de batterij beschrijven en werklustmodellen die het belastingsprofiel van een apparaat beschrijven. In de literatuur zijn veel verschillende batterijmodellen beschikbaar.

In **Hoofdstuk 2** vergelijken we de belangrijkste batterijmodellen met elkaar, en onderzoeken we welke modellen geschikt zijn om te combineren met een werklastermodel. Een geschikt model moet de twee belangrijke effecten die hierboven beschreven zijn goed beschrijven, en tegelijkertijd niet te groot zijn, zodat het gecombineerde batterij-werklastermodel niet te complex wordt en de berekeningen die hiermee gedaan moeten worden niet te lang duren. De batterijmodellen die aan deze eisen voldoen zijn de zogenaamde analytische modellen, het *kinetic battery* model en het *diffusion* model.

In **Hoofdstuk 3** vergelijken we deze twee modellen in meer detail met elkaar. Hieruit blijkt dat deze twee modellen sterk aan elkaar gerelateerd zijn. Het *kinetic battery* model is een eerste orde benadering van het *diffusion* model. Wanneer we de parameters van het *kinetic battery* model goed kiezen, levert deze eerste orde benadering dezelfde resultaten bij de berekeningen van de batterijlevensduur op. We hebben daarom het *kinetic battery* model gekozen om de verdere berekeningen in dit proefschrift mee te doen.

In **Hoofdstuk 4** combineren we het *kinetic battery* model met een werklastermodel. Voor veel apparaten is het lastig om de exacte werklaster te voorspellen omdat deze wordt bepaald door toevallige invloeden van buitenaf, zoals het gedrag van de gebruiker. Om de random patronen in de werklaster te modelleren maken we gebruik van een Markov model. De combinatie van het batterijmodel met het werklastermodel is een zogenaamd *inhomogeen Markov reward* model. Voor dit model presenteren we nieuwe algoritmes voor het berekenen van de verdeling en de verwachtingswaarde van de levensduur van de batterij en de hoeveelheid lading die de batterij levert voor een gegeven werklaster. Hiermee kunnen we de invloed van werklaster op de levensduur van de batterij bepalen.

In een apparaat dat door slecht één batterij van energie wordt voorzien is het vaak lastig om de werklaster van de batterij te veranderen. Een gebruiker zal, bijvoorbeeld, niet willen wachten met het bellen met zijn mobiele telefoon omdat dit in de toekomst misschien beter is voor de batterij. Echter, in een apparaat waar meerdere batterijen gebruikt kunnen worden, kan men door te schakelen tussen de batterijen de werklaster over de batterijen verdelen, en zo de werklaster van iedere batterij op zich beïnvloeden. In **Hoofdstuk 5** en **Hoofdstuk 6** bepalen we op twee manieren wat de beste strategie is om de werklaster over meerdere batterijen te verdelen.

In **Hoofdstuk 5** beschrijven we het *kinetic battery* model met *priced timed automata*. In dit nieuwe model kan er alleen geschakeld worden tussen de batterijen op het moment dat er een verandering plaats vindt in de werklaster. Met behulp van *model checking* technieken wordt voor een gegeven belasting het optimale schema voor het verdelen van de werklaster gevonden. De berekeningen laten zien dat door het verdelen van de werklaster over de batterijen de totale levensduur van de batterijen langer wordt. Het probleem van dit model is echter, dat de berekeningen alleen gedaan kunnen worden voor batterijen met een kleine capaciteit, voor grotere bat-

terijen worden de berekeningen te zwaar.

De oplossingstechniek die in **Hoofdstuk 6** wordt gebruikt heeft dit probleem niet. In dit hoofdstuk laten we de restricties voor wanneer er geschakeld mag worden tussen de batterijen vallen. Uit de analyse van de vergelijkingen van het *kinetic battery* model voor het totale systeem kan nu een bovengrens worden bepaald voor de levensduur van de batterijen. Het blijkt dat deze bovengrens ook dicht benaderd kan worden als men toe laat dat er snel geschakeld wordt tussen de batterijen. De winst in levensduur die men behaalt wanneer het schakelen tussen batterijen wordt toegepast hangt voornamelijk af van de verhouding tussen de capaciteit van de batterij en de gemiddelde belasting.

Dankwoord

Na het lezen van al het voorgaande, zal je begrijpen dat ik dit proefschrift niet heb kunnen schrijven zonder de hulp en inspiratie van anderen. Hierbij wil ik dan ook de mensen bedanken die direct of indirect hebben bijgedragen aan de totstandkoming van dit boekje.

Om te beginnen wil ik mijn promotoren, Boudewijn en Joost-Pieter, bedanken. Boudewijn, regelmatig kwam je in onze wekelijkse meetings met nieuwe ideeën en oplossingen en kon ik hierna met hernieuwde energie aan de slag. Wat ook niet onvermeld mag blijven is de snelheid waarmee je de manuscripten voor de papers en hoofdstukken van feedback voorzag, meestal kreeg ik de teksten binnen 24 uur voorzien van commentaar weer terug.

Joost-Pieter, de bezoeken aan je groep in Aken hebben een vruchtbare samenwerking opgeleverd. De hoofdstukken 5 en 6 zijn hier het resultaat van. Henrik and Alexandru, I have enjoyed working with you on the problems of battery scheduling, and I want to thank you for your contributions to this work.

Verder wil ik alle leden van DACS bedanken voor de prettige werksfeer binnen de groep. Speciale dank gaat uit naar mijn kamergenotes. De laatste vier jaar heb ik mijn kamer mogen delen met drie dames, iets wat uitzonderlijk is binnen de faculteit EWI. Anna, Desi and Yimeng, thanks for the often needed distractions and “gezelligheid”, both in and outside the office. As we are now all moving on to the next phase in our lives, I hope our friendship will last for many more years.

Wanneer je vanuit het westen des lands naar Enschede verhuist zijn er een hoop mensen die denken dat je gaat emigreren. Het scheelt natuurlijk ook maar 10 km. Ik wil dan ook alle familie en vrienden bedanken die gedurende de afgelopen jaren de reis naar het verre oosten hebben gemaakt en menig weekend hebben opgefleurd met een bezoek.

Karen, hier is mijn proefschrift dan. Jij hebt alle hoogte- en dieptepunten meegemaakt. Het afgelopen jaar was buitengewoon druk, maar bevatte al vele mooie momenten, en het is nog niet afgelopen. Na het voltooien van mijn proefschrift, wacht ons dit jaar nog een nieuw avontuur, met z’n drieën.