Analytical Solution for Long Battery Lifetime Prediction in Nonadaptive Systems

Dmitry Ivanov[1,2], Kim G. Larsen[2], Sibylle Schupp[1], and Jiří Srba[2]

[1] Hamburg University of Technology, Germany
{d.ivanov,schupp}@tuhh.de
[2] Department of Computer Science, Aalborg University, Denmark
{kgl,srba}@cs.aau.dk

Abstract. Uppaal SMC is a state-of-the-art tool for modelling and statistical analysis of hybrid systems, allowing the user to directly model the expected battery consumption in battery-operated devices. The tool employs a numerical approach for solving differential equations describing the continuous evolution of a hybrid system, however, the addition of a battery model significantly slows down the simulation and decreases the precision of the analysis. Moreover, Uppaal SMC is not optimized for obtaining simulations with durations of realistic battery lifetimes. We propose an analytical approach to address the performance and precision issues of battery modelling, and a trace extrapolation technique for extending the prediction horizon of Uppaal SMC. Our approach shows a performance gain of up to 80% on two industrial wireless sensor protocol models, while improving the precision with up to 55%. As a proof of concept, we develop a tool prototype where we apply our extrapolation technique for predicting battery lifetimes and show that the expected battery lifetime for several months of device operation can be computed within a reasonable computation time.

1 Introduction

Battery lifetime is one of the main concerns in the usability evaluation of modern portable devices. Modelling of battery consumption can help to evaluate the expected power consumption prior to the actual construction and deployment of the system. Uppaal SMC [4] is a state-of-the-art tool that allows one to model and evaluate a behaviour of (stochastic) hybrid systems via a statistical model checking approach. Among others, it can be applied to modelling of battery-operated devices as abstract hybrid automata. We take a closer look at the possibilities of battery modelling in Uppaal SMC, focusing primarily on estimating the battery lifetime for nonadaptive systems, i.e., systems whose behaviour does not depend on the actual state of the battery.

A straightforward approach to modelling energy consumption is to annotate an existing Uppaal SMC model with the information in what locations energy is consumed and by what rate, and to extend the model with a hybrid automaton representing the battery. However, this method has several practical drawbacks and limitations. First, hybrid modelling in Uppaal SMC is performed numerically with the Euler method [22] and this causes a large performance overhead and precision loss, in particular when we want to reason about long simulation runs lasting for months. Second, Uppaal SMC is not optimized for generating long lasting simulations that are necessary for estimating typical battery lifetimes. In applications like adhoc wireless networks where sensors and other network components are battery-operated, an average battery life is expected to last for several months, whereas simulating just a few hours of such a network operation in Uppaal SMC takes unaffordable computation time and memory.

We propose an analytical approach for computing the current battery charge. As we restrict ourselves to nonadaptive systems, we can first simulate the system's behaviour (without the battery model) with Uppaal SMC and then annotate the produced simulation trace with battery consumption, instead of adding a hybrid automaton of the battery to the Uppaal SMC model. We also replace the numerical method used in the tool by computing a precise analytical solution to the battery model equations based on the kinetic battery model (KiBaM) [18]. We demonstrate on two industrial-strength case studies that our approach considerably speeds up the simulations and improves the precision of battery prediction. By analytically solving the equations for polynomials of arbitrary degree, we enable an accurate modelling of the electrical current as an input to the battery model. Our experiments show that this is achievable within acceptable time and memory usage and significantly faster than a direct modelling in Uppaal SMC. Moreover, we propose a trace extension technique for the traces generated by Uppaal SMC in order to estimate battery lifetimes for several months of device operation. We apply this technique to two case studies that model two different wireless network protocols and show that our approach allows us to obtain simulation results within an acceptable time horizon, while at the same time extending the prediction precision.

Related work. The research in battery modelling and lifetime optimization has several directions. Some of the existing battery models were extended to capture environmental parameters and usage patterns more precisely. Jongerden et al. [15] observed that KiBaM parameters change over time for rechargeable batteries. Their experiments show that the modelling error grows with the amount of the recharge cycles of the battery. Rodrigues et al. [21] proposed a temperature-dependent KiBaM (T-KiBaM). They applied the Arrhenius law to the parameter k of the KiBaM making it dependent on the temperature. Thus, T-KiBaM represents the intensity of the chemical processes inside the battery induced by the temperature. Various approaches exist for applying battery models to battery lifetime maximization problems. Model checking is one of such approaches. Bisgaard et al. [1] used Uppaal CORA for finding an optimal schedule for a satellite. However, they used a naive battery model instead of KiBaM because the model does not match the priced timed automata (PTA) formalism used by Uppaal CORA. Jongerden et al. [13] proposed a discretized version of KiBaM for finding an optimal schedule with Uppaal CORA. The discretized model can be expressed as a PTA and thus passed to Uppaal CORA.

They experimentially compared the accuracy of the discretized version to the original model. The experiments showed a deviation up to 1%. However, the comparison was done for parameter sets corresponding to batteries with very small capacities (92 mAh and 183 mAh). Moreover, the experiments were done for scenarios with very short battery lifetimes (less than 100 min). Hence, it is an open question whether this experimental setup is relevant for evaluating the modelling accuracy for much longer battery lifetimes.

Wognsen et al. [24] introduced a wear score function based on dynamic evolution of state-of-charge. Using the recent branch UPPAAL Stratego the score function was used to optimize the life time of a battery powered nano-satellite of the company GOMSpace. David et al. [5] applied Uppaal SMC to measure the power consumption of the LMAC protocol. No battery model was considered in this work. These two approaches are helpful for battery life prolongation, although they do not predict the lifetime. Boker et al. [2] introduced battery transition systems based on a discretized KiBaM. They proved that for a certain class of these models model checking is decidable but restricted to the discretized model. We consider the undiscretized version of KiBaM and use the analytical solution to it, which promises to be more precise. Heni et al. [11] model a mobile network as an on/off Markov decision chain and use a naive battery model. Panigrahi et al. [19] estimate the battery life of mobile embedded systems with a stochastic process (a discrete time Markov chain by Chiasserini and Rao [3]). In our work we complement Uppaal SMC models with the battery model, thus enabling the battery life prediction for more complicated systems. Xue et al. [25] introduce an energy saving mechanism for the IEEE 802.16e standard for mobile stations. They analyze the current performance by using a Markov chain model of the protocol but do not consider the KiBaM battery model. They evaluate their proposal by simulation in the discrete event simulation tool NS2 while we use a statistical model checker instead. Energy-aware software engineering [7] can model energy on different levels of abstraction: on the instruction set or the LLVM IR. It uses Horn clauses coupled with energy information. If the average values are used, the bounds on the energy consumption can be unsafe. Static energy profiling [10] is a variant of static analysis for energy consumption that gives probability distributions instead of static bounds. Unfortunately, these two approaches are not applicable for network protocol modeling when the source code is unavailable, especially on the design phase. We are not aware of any work on integrating an analytical solution to the existing modelling approaches and most of the methods above either cannot deal with long lifetime battery modelling or consider naive battery models. In our work, we allow to compute the precise analytical solution of KiBaM battery model for several months of system execution and we can provide precise estimates on battery lifetimes.

2 Battery model

The simplest way to model a battery is to assume that its capacity is constant (independent from the consumption pattern) and its charge decreases

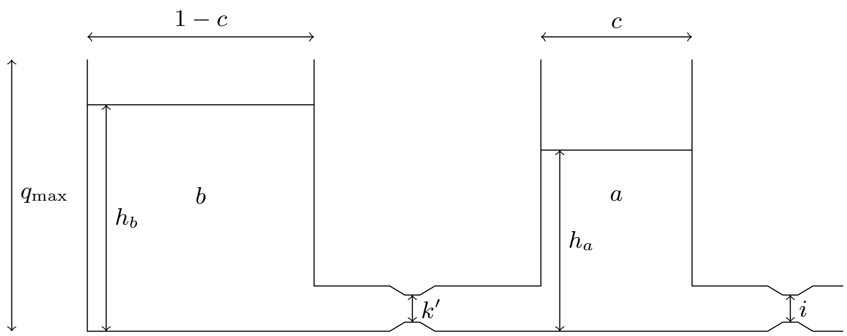


Fig. 1: KiBaM battery model

proportionally with the drawn current. However, the main drawback of this approach is that it ignores nonlinear battery effects. In reality, the delivered battery capacity, or the real amount of energy the battery provides, depends on the actual profile of the current, which is also known as the *rate capacity effect*. Different battery models were developed taking this into account and they can be classified as electrochemical [6], electrical-circuit [9], stochastic [3], and analytical models [18, 20]. We choose to work with the analytical model KiBaM [18] as it is a relatively simple model and for most practical scenarios there is only a small precision gain when considering more advanced models like the diffusion model [14, 16].

2.1 Kinetic battery model

The Kinetic Battery Model (KiBaM) [18] captures nonlinear rate capacity effects by representing the battery as two communicating vessels. The charge is split into two parts: the available charge a and the bound charge b. The charge is modelled as a liquid stored in two interconnected vessels of width c and $1 - c$ that represent the available and bound charges, respectively. The liquid levels correspond to the charges and are written as h_a and h_b. Figure 1 illustrates the model.

The available charge is taken on current i directly, and the bound charge supplies the available charge whenever h_a is smaller than h_b. The flow rate between the available and bound charges is proportional to the difference between their heights and the parameter k'. When the available charge is zero, the battery is considered as discharged even though there may still be some bound charge left. Equation 1 formalizes the relationship between available and bound charge [18].

$$\begin{cases} \dot{a} = -i + k'(h_b - h_a) \\ \dot{b} = -k'(h_b - h_a) \end{cases} \tag{1}$$

As we can see from the model abstraction, the battery life depends on the energy consumption pattern. If energy is drawn rapidly then the available charge can become empty, having a significant amount of the bound charge left. Conversely, a slower energy consumption allows the available charge to restore and it reduces the amount of the bound charge left unused. Considering that $h_a = \frac{a}{c}$ and $h_b = \frac{b}{1-c}$, and substituting k' with $kc(1-c)$, where $0 \leq c \leq 1$ is the available charge fraction, we obtain Equation 2.

$$\begin{cases} \dot{a} = -i + kcb - k(1-c)a \\ \dot{b} = -kcb + k(1-c)a \end{cases} \tag{2}$$

We can solve this equation for a constant current i by using Laplace transforms [18] and obtain

$$\begin{cases} a = a_0 e^{-kt} + \frac{(q_0 kc - i)(1-e^{-kt}) - ic(kt-1+e^{-kt})}{k} \\ b = b_0 e^{-kt} + q_0(1-c)(1-e^{-kt}) - \frac{i(1-c)(kt-1+e^{-kt})}{k} \end{cases} \tag{3}$$

where a_0, b_0, and q_0 are the available, bound, and full charge at the zero-time.

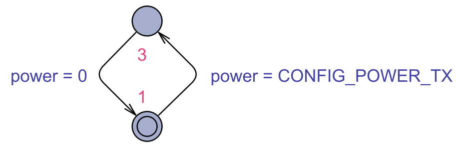
For evaluation purposes we also consider a naive battery model. It represents the battery as a single charge vessel from where the current is drawn. The naive model corresponds to KiBaM with $c = 1$ and where k' becomes insignificant. A battery life computed for the naive model is a theoretical upper bound for the KiBaM battery life. The difference between them shows how much the battery life can be increased by changing the energy consumption pattern. For setting up the KiBaM parameters, we use the data sheet for the battery TADIRAN SL-750 [23]. For the parameter estimation we compute the discharge rate and lifetime pairs for model fitting as explained by Manwell et al. [18]. We use the obtained parameters $c \approx 0.06$, $k \approx 0.46\ h^{-1}$, $q_0 \approx 1.17\ Ah$ in all our experiments.

2.2 Modeling KiBaM in Uppaal SMC

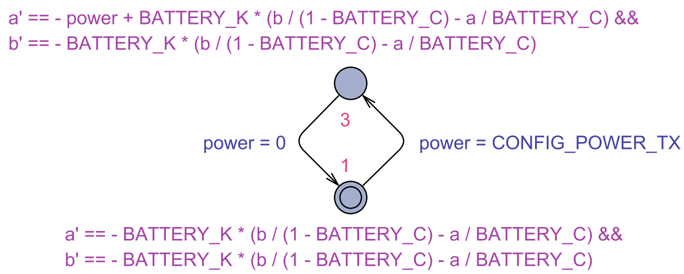
Uppaal SMC [4], the frontend of our work, is a statistical extension of the Uppaal model checker. It combats state space explosion by focusing on stochastic properties of the model. Uppaal models that are used for classical model checking can be adjusted to meet the requirements of Uppaal SMC, namely, input determinism, independent progress, and absence of priority between channels or processes. When all three requirements are met, one can apply both classical and statistical model checking. Uppaal SMC features probability evaluation, hypothesis testing, probability comparison, and simulation framework. In our work we focus primarily on generating simulations.

```
const double INIT_CHARGE = 4.2e+09,
             BATTERY_C = 0.06,
             BATTERY_K = 1.3e-07;
clock a = INIT_CHARGE * BATTERY_C,
      b = INIT_CHARGE * (1 - BATTERY_C);
```

Listing 1.1: Uppaal declarations for a KiBaM



(a) Simple Uppaal SMC model



(b) Uppaal SMC model annotated with battery equations

Fig. 2: Uppaal SMC model and its annotation with KiBaM equations

simulation length (ms)	10^4	10^5	10^6	10^7
model from Fig. 2a	0.125±0.002	1.125±0.011	11.4±0.11	107±1.1
model from Fig. 2b	2.7±0.02	26±0.2	268±2	2615±21
analytical solution to Fig. 2a	1.6±0.05	4±0.1	30±0.6	457±9

Table 1: Simulation time in seconds for the model from Figure 2

Let us consider a simple Uppaal SMC model with two locations shown in Figure 2a. We assume that the power consumption is zero as long as the model stays in the initial location and equal to the current draw `CONFIG_POWER_TX` in the other location. The variable `power` keeps track of the current power consumption level. We now extend this model by introducing constant declaration in Listing 1.1 and by adding invariants, i.e., flow equations to the model locations as shown in Figure 2b. The new declarations include the battery constants q_{max}, c, and k, and clocks that store the available and bound charge values. We map the battery model constants to the constant double type because they do not change over time in our setup. The available charge and the bound charge have to be clocks since their evolution is continuous.

This direct implementation of KiBaM in Uppaal has several practical drawbacks. First, Uppaal SMC solves the differential equations with the Euler method [4], which is less precise than the analytical solution. Second, the simulation of a hybrid model requires a considerable computational effort compared to the non-hybrid one (we will show this in the evaluation section) because Uppaal SMC solves the flow equations numerically. The performance measurements from Table 1 show that the battery equations slow down the simulations even for our simple example by an approximate factor of 25 whereas our implementation of the analytical solution (described below) significantly improves the performance for longer simulations. Finally, a model time unit corresponds to one millisecond of the real time. Estimating battery life (which can last several months) therefore requires a very long simulation. This motivates our effort in finding an analytical solution to battery consumption as described in the next subsection.

2.3 Analytical solution to KiBaM

The KiBaM equations can be solved both numerically and analytically. A numerical solution, e.g., the Euler method, performs a discretization step on which the computations are performed. The discretization step should be reasonably small for precise results. In case of very long simulations, numerical methods become computationally expensive and imprecise due to error accumulations. In contrast, the analytical solution has no discretization step, i.e., the computations can be done only when the current level function changes. It means that the analytical solution can save the computational power on constant or polynomial pieces of the current. Moreover, the analytical solution does not accumulate the discretization error over the simulation time, which is beneficial for long simulations. Hence, we shall develop the analytical solution to KiBaM. In what follows, we provide solutions to KiBaM equations both in general and piecewise cases.

First, we solve Equation 1 for an arbitrary current i in order to provide an analytical solution not only for a piecewise constant current intensity but also for piecewise polynomial intensities. Let $i = i(t)$, then Equation 2 can be rewritten as follows.

$$\begin{cases} \dot{a} = -i(t) + kcb - k(1-c)a \\ \dot{b} = -kcb + k(1-c)a \end{cases} \tag{4}$$

We solve Equation 4 by the Laplace transform and obtain the following.

$$\begin{cases} a = a_0 e^{-kt} + q_0 c(1 - e^{-kt}) - \int_0^t i(\tau)d\tau + (1-c)\int_0^t i(\tau)(1 - e^{k(\tau - t)})d\tau \\ b = b_0 e^{-kt} + q_0(1-c)(1 - e^{-kt}) - (1-c)\int_0^t i(\tau)(1 - e^{k(\tau - t)})d\tau \end{cases} \tag{5}$$

Equation 5 is the analytical solution to Equation 2 for an arbitrary current $i = i(t)$. Next, we need to solve the integral $\int_0^t i(\tau)(1-e^{k(\tau-t)})d\tau$ for a polynomial $i(t)$ to obtain the solution to KiBaM for a polynomial current and substitute the polynomial $i(t)$ in Equation 5:

$$\begin{cases} a = a_0 e^{-kt} + q_0 c(1 - e^{-kt}) - \sum_{j=0}^{n} \frac{i_j t^{j+1}}{j+1} \\ \qquad + (1-c)\sum_{j=0}^{n} j! i_j \left[\sum_{l=0}^{j+1} \frac{(-1)^l t^{j-l+1}}{k^l (j-l+1)!} + \frac{(-1)^j e^{-kt}}{k^{j+1}} \right] \\ b = b_0 e^{-kt} + q_0(1-c)(1 - e^{-kt}) \\ \qquad - (1-c)\sum_{j=0}^{n} j! i_j \left[\sum_{l=0}^{j+1} \frac{(-1)^l t^{j-l+1}}{k^l (j-l+1)!} + \frac{(-1)^j e^{-kt}}{k^{j+1}} \right] \end{cases} \tag{6}$$

Equation 6 is now the analytical solution to Equation 2 for a polynomial current $i = \sum_{j=0}^{n} i_j t^j$. For $n = 0$ it coincides with Equation 3 and for $n = 1$, or a linear current, Equation 6 gives the final solution.

$$\begin{cases} a = a_0 e^{-kt} + \frac{(q_0 kc - i_0)(1-e^{-kt}) - i_0 c(kt-1+e^{-kt})}{k} - \frac{i_1(1-c)(kt-1+e^{-kt})}{k^2} - \frac{c i_1 t^2}{2} \\ b = b_0 e^{-kt} + q_0(1-c)(1-e^{-kt}) - \frac{i_0(1-c)(kt-1+e^{-kt})}{k} - \frac{i_1(1-c)(k^2t^2/2-kt+1-e^{-kt})}{k^2} \end{cases} \quad (7)$$

Provided that the charge level of a battery does not affect the system behaviour, the KiBaM data can be computed independently based only on a given electric current function. Assuming that the current intensity is a piecewise constant function of time, we can use the analytical solution of Equation 3 provided above to track the charge of the battery.

2.4 Tool chain for KiBaM battery consumption

We assume that the protocols have been already modeled using the Uppaal SMC formalism. Our task is to design a backend to Uppaal SMC that captures the computed simulation trace (without the battery model but with annotated locations with their power consumption profile), computes the analytical solution to KiBaM, and transparently extends the simulation trace with available and bound charge values in the format used by Uppaal SMC. Such a way of extending the output allows to load the annotated trace back to Uppaal SMC but allows also for its visualization in other tools like VisuAAL [17] (for a demo see a video at `https://youtu.be/hGfMww97xWw` demonstrating a battery drain of network devices sharing the same protocol over the elapsed time). Moreover, we can interpret the simulation trace in a different way for solving other problems, in particular, for computing the battery lifetime or combining the trace with real data as discussed later. Figure 3 shows the data flow for our trace-extender tool. Uppaal SMC is launched in the command line mode having as input a model extended with information on its power consumption and a simulation query requesting the trace of the current. For example, the query

```
simulate 1 [<=1000]{power}
```

generates one simulation trace for the variable `power` lasting 1000 time units. Our extender then combines the settings, model parameters, and information on input and output expressions from a user-defined configuration file.

3 Case study on battery charge in wireless networks

We present now two case studies in order to evaluate the performance of battery charge modelling. Our aim is to evaluate both precision and performance of our implementation of the analytical solution compared to the numerical solution provided in Uppaal SMC. We consider two wireless network protocols used in industry, namely the LMAC [8] and Neocortec's MAC [12] protocols, both

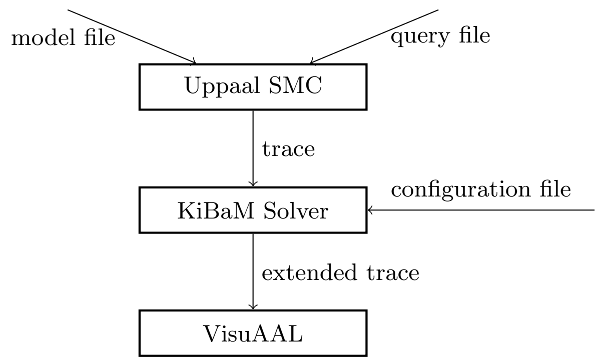


Fig. 3: KiBaM solver data flow

of them being examples of nonadaptive systems. The two protocols differ in the frequency of scheduled transmissions (LMAC is switching more often from sleep to power-consuming modes) and hence the power consumption of both protocols is different. The constants used in the Neocortec's MAC protocol were modified in our model as the exact configuration of the protocol is not publicly available and hence our predicted battery lifetime in this case can differ from the performance of the actual devices sold by the company.

3.1 Protocol descriptions

The LMAC [8] protocol is a medium access control protocol for multi-hop, energy-constrained wireless sensor networks. This protocol addresses energy efficiency, self-configuration, and distributed operation. The main concept of the LMAC is scheduling. The protocol tries to avoid idle listening by dedicating a time slot to each node. Neocortec's MAC protocol is an alternative attempt on designing an energy-efficient wireless network protocol. The patent [12] discloses some of the protocol details, however, a more detailed description was provided at [17]. We emphasize that this explanation is an approximation of the actual protocol behaviour and hence cannot be used for drawing conclusions about the actual Neocortec implementation of the protocol. Neocortec's MAC protocol is designed to broadcast data between neighbouring nodes and it synchronizes the nodes while performing beacon scans. As the LMAC, Neocortec's MAC protocol benefits from scheduling the data transmission, however does so with a focus on energy preservation. We use both protocols for our evaluation purposes because they show two degrees of power change densities: Neocortec's MAC protocol exhibits very sparse communications with long sleep periods, whereas the LMAC protocol has a more uniform activity with around 50 times more frequent power state changes than Neocortec's MAC. Moreover, the protocols have different power consumption. Thus, in the setup explained in Section 4.2 the LMAC and Neocortec's MAC consume the average current of 13 mA and 0.5 mA respectively.

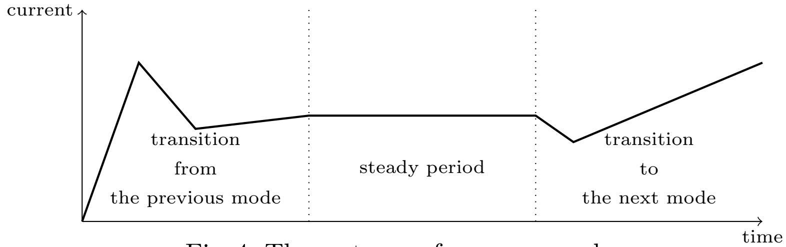


Fig. 4: Three stages of a power mode

3.2 Power modes in protocols

Power consumption modelling of network protocols by using power modes provides a simple yet precise abstraction of the underlying physical phenomenon. In our approach, a network protocol is represented as an automaton where each state is annotated with a power mode. For example, the Neocortec's MAC protocol has the transmission (TX), the receive (RX), and the sleep (Sleep) power mode. The power modes represent the basic operations of the protocol (TX and RX) as well as their absence (Sleep). Usually, the current is assumed to be constant within a power state. This is a simplification of the actual profile of the current consumption and this simplification can have some bias regarding the current behaviour with frequent power mode shifts. Hence we propose a more precise interpretation of power modes capturing this effect.

We distinguish three stages in a power mode: transition from the previous mode, a steady period, and transition to the next mode. Figure 4 illustrates this division (based on the actual measurements provided to us by Neocortec). We assume that the steady period is constant and can be of arbitrary duration. With this interpretation, we can use real measurements in our simulations and model the reality more precisely. By extending and compressing the steady period we can fit an arbitrarily long power mode that appears in the simulation trace. However, our approach may restrict the shortest possible duration for an operation. Eliminating the steady period completely, we obtain the power mode consisting of transitions only. Its duration defines the lower bound of the power mode duration.

3.3 Experiments

We use the Uppaal SMC model of Neocortec's MAC protocol described in [17] and the LMAC protocol model from [8]. We annotate the locations in the models with power consumption levels in order to evaluate the performance of the KiBaM model. We integrate the polynomial analytical solution into our tool KiBaM Solver, which is available on GitHub via link `https://github.com/DIvanov503/KiBaM-Solver.git`.

First, we evaluate how much the numerical solution computed by Uppaal SMC differs from the precise analytical solution, i.e., accumulates the error,

simulation length (seconds)	10	30	50	70	90
Neocortec's MAC avail., mAms	60±0.6	243±0.6	502±0.9	811±1.3	1401±1.8
Neocortec's MAC bound, mAms	66±0.1	270±0.4	527±0.7	837±1.1	1402±1.8
LMAC avail., mAms	85±0.3	788±2.5	2249±6.7	4442±13.0	8997±26.6
LMAC bound, mAms	92±0.2	833±2.2	2310±6.3	4520±12.5	9191±25.7

Table 2: Absolute difference between the Euler method and the analytical solution for available and bound charges

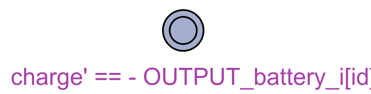


Fig. 5: Naive model of battery in Uppaal SMC

as we increase the length of the simulation. We simulate the two protocols 50 times for 90,000 time units (90 seconds of real time). Table 2 shows the evolution of the average absolute difference between the numerical and analytical solution with 95% confidence intervals. We can see that for both protocols, the difference grows over time as the numerical solution gets less and less precise. The difference for the LMAC protocol is on average higher than for Neocortec's MAC because the more intensive power consumption of the LMAC protocol causes larger accumulated truncation errors of the Euler method. This relatively short simulation does not give us an estimate of the precision gain for the battery life estimation as the drop of batter capacity within 90 seconds is negligible. However, assuming an average battery lifetime to be 100 days for Neocortec's MAC and 3 days for the LMAC protocol, we can express the simulation time and the absolute error of the available and the bound charge as fractions of the battery life and initial charge values respectively. For Neocortec's MAC the simulation models $1 \cdot 10^{-3}\%$ of the average battery life, and the Euler method accumulates an error of $5.5 \cdot 10^{-4}\%$ for the available charge, and $3.5 \cdot 10^{-5}\%$ for the bound charge. The percentages for the LMAC protocol are $3.5 \cdot 10^{-2}\%$, $1.6 \cdot 10^{-3}\%$, and $1.1 \cdot 10^{-4}\%$ respectively. Assuming that the truncation error grows linearly, we can extrapolate our measurements to the duration of the average battery lifetime. Thus, we can predict the available and bound charge for the Neocortec's MAC to have numerical errors of 55% and 3.5%. For the LMAC the estimate is better: 4.6% and 0.3%. We see that the relative error for Neocortec's MAC gets very high. One explanation to this phenomenon is that the average battery lifetime is too long for the Euler method with the given discretization step. Same applies to the LMAC protocol experiment showing a smaller relative error because of the 8.5 times shorter average battery lifetime compared to Neocortec's MAC. The difference between relative errors for the available and bound charges can arise because we use the parameter $c \approx 0.06$ reflecting the available/bound charge ratio is very small meaning that the same absolute error is larger for the available charge than for the bound charge.

# nodes	5	10	50	100
Neocortec's MAC, numeric. KiBaM	54±0.5	119±0.8	686±6.6	2159±19.4
Neocortec's MAC, numeric. naive	42±0.3	87±0.4	544±4.2	1671±14.1
Neocortec's MAC, analyt. KiBaM	0.6±0.01	1.9±0.33	20.6±0.18	121.6±1.01
LMAC, numeric. KiBaM	22±0.1	45±0.1	288±1.0	713±2.1
LMAC, numeric. naive	0.8±0.02	2.1±0.02	41.6±0.23	145.7±0.59
LMAC, analyt. KiBaM	3±0.1	4±0.1	33±0.2	108±0.4

Table 3: Time (in seconds) required for simulating 10,000 time units (average of 100 repetitions with 97% confidence interval)

Next, we compare the performance of the KiBaM model to the naive battery model. The naive model represents the battery as a single power vessel, which can be considered as a KiBaM with $c = 1$ and non-significant k, simplifying the Uppaal SMC model for the naive battery to the one shown in Figure 5 where `charge` represents the remaining charge of the battery. Table 3 shows the time required to simulate 10000 units of time (10 seconds of real time) of the Uppaal SMC model annotated with the KiBaM and the naive battery model and compared to our analytical solution for KiBaM. The problems a scaled by the number of nodes in randomly generated network topologies.

The experiments show in case of Neocortec's MAC significantly faster running times of the analytical approach, compared to numerical solution. We also notice that the performance gain of the analytical solution depends on the density of power mode changes—the more changes occur, the longer time it takes to compute the analytical solution. The data for LMAC indicate a lot smaller performance advantage of our analytical solution since the LMAC model changes the power state around 50 times more frequently than Neocortec's MAC, however, the advantage becomes more obvious when we consider larger networks like e.g. 100 nodes in the table.

4 Simulation traces for estimating long battery lifetime

In our approach we focus on nonadaptive protocols, meaning that the behaviour of the protocol/system does not depend on the actual charge of the battery. This allows us to extend an existing protocol model with the information on power modes and use a statistical model checker like Uppaal SMC to generate the trace for further analysis. However, the generation of battery lifetime-long simulations is problematic since the model time unit corresponds to one millisecond of the real time and that makes estimating battery life (which can last several months) to require a very long simulation exceeding the typical time and memory constraints. Hence, we designed an extrapolation technique that allows us to expand shorter simulation traces to longer ones.

method \ protocol	Neocortec's MAC	LMAC
trace	1852±6	8746±153
const	238±1	535±9
const6	2127±9	—
const5	2551±12	—
const4	3624±13	—
const3	6178±27	—
const2	7428±41	—
lin2	11814±49	—

Table 4: Execution time (in seconds) for generating a 10,000,000 time units long trace with Uppaal SMC and computing battery lifetimes with trace extrapolation and various current mode modelling approaches

4.1 Extension of simulation traces

We shall introduce an extrapolation technique where we first generate a time-bounded trace and repeat the latter half of it until the battery discharges. We exploit that network protocols have a warm-up period and a steady state. The warm-up period is not representative for the model behaviour in the long term. In contrast, the steady period has a nondeterministic regularity: the variable distributions are similar from time to time. If the steady period part of the trace is long enough to assume approximate statistical independence, it can be repeated to model the behaviour on arbitrarily long timeline. We use this idea to continue battery simulation until it fully discharges.

We assume the first half of the simulation to be a warm-up period due to the initial routines that are performed by the protocols. The fraction of the trace to be skipped as the warm-up period can be adjusted depending on the trace duration.

4.2 Experiments with long battery lifetime prediction

As a proof of concept, we perform two case studies investigating the battery lifetime of the already discussed protocols: Neocortec's MAC and LMAC. We implement a battery life estimator on top of our KiBaM Solver. For battery life evaluation, we generate in VisuAAL a network consisting of 25 nodes with a random topology. We generate 100 traces of 10,000,000 time units corresponding to 2.8 hours of real time in Uppaal SMC. Then we apply the trace extrapolation technique (Section 4.1) to continue the modelling up to the end of the battery lifetime. We also investigate how a more complex power consumption modelling from Section 3.2 affects performance of battery lifetime prediction. We discretize current consumption graphs from real wireless sensor network nodes and manually divide power modes into stages. We interpolate the stages with splines and vary the number of points a spline interpolates and the maximal degree of splines. Thus, we construct a sequence of interpolation schemes of ascending complexity. First, we use constant splines and increase their number until each spline interpolates two points only. Then we replace the constant splines with linear ones. We perform these refinement steps for Neocortec's MAC only because we do not have graphs for idle mode in LMAC.

Table 4 shows the computation time required for generating an input trace with Uppaal SMC and predicting the battery lifetime. Here "const" is a usual piecewise constant approach without distinguishing stages in power modes, "const$i$" stands for a refined piecewise constant interpolating i points with a single constant function, i.e., a smaller i corresponds to a more complex interpolation, "lin$i$" is same as "const$i$" but uses linear splines. We emphasize that computation of the KiBaM solution takes time comparable to generating a relatively short trace in Uppaal SMC but reaches a much longer prediction horizon: between 83 and 123 days for Neocortec's MAC and between 1.8 and 4.7 days for the LMAC. The table also shows the performance of the battery life estimation involving a more precise power mode model. While decreasing i in "const$i$" we add constant steps to the transition state approximations and finally switch to linear splines in "lin2." Our observations show that our trace extrapolation method allows for using more complex input for a reasonable performance penalty but most importantly, by the combination of our techniques, we are able to predict the battery lifetime in the duration of several months.

5 Conclusion and future work

We investigated two approaches to improve the precision and performance of battery modelling in hybrid automata simulation tools like Uppaal SMC. We developed an analytical solution of the flow equations allowing us to use a separate postprocessing unit and we implemented a tool for computing the analytical solution to the kinetic battery model. Moreover, we proposed a method of computing the battery lifetime by extrapolating simulation traces. This technique allows estimating battery lifetimes without simulating the model for the whole battery discharge period, which is infeasible in practice. As a proof of concept we applied our implementation of the analytical solution for comparing the battery lifetimes of two energy-efficient wireless sensor network protocols: LMAC and Neocortec's MAC.

As a part of the detailed current modelling, we solved the KiBaM equations in the general case. From the general solution we obtained the solution for polynomial currents and we proposed a method for using real current measurements with simulation traces. The evaluation part of our work included various setups. We showed that our implementation of the analytical solution has a better performance than the numerical one and we demonstrated that the difference between the numerical and the analytical solution grows over time.

We suggest several directions of the future work. First, we can continue refining the modelling approach. The trace extrapolation technique we use now is rather simplistic. More advanced statistical methods can be applied to our problem. For instance, one can try to obtain a discrete-time Markov chain from the simulation trace and perform the battery life estimation stochastically, similarly to stochastic battery models by Chiasserini and Rao [3]. Second, we can use a more advanced battery model. For example, we can take the temperature-dependent KiBaM [21] to capture the temperature influence. Furthermore, we can investigate the impact on the battery lifetime of various protocol optimizations, e.g. the node can nondeterministically select the neighbours it is going to interact with and ignore the others. Such an optimization can reduce the power consumption of the node but also can deteriorate the network characteristics.

Acknowledgements. We thank Kevin H. Jørgensen, Niels B. Christoffersen, Rasmus D. Petersen and Tim H. Gjøderum for their help with integrating our tool with VisuAAL and numerous discussions about the annotation of battery modes in Uppaal SMC models of the two wireless network protocols. We thank Neocortec for allowing us to use their MAC protocol in our experiments and Thomas Steen Halkier for discussions about the battery consumption patterns and for providing us with current consumption graphs. The work was funded by the center IDEA4CPS, the Innovation Fund Denmark center DiCyPS and ERC Advanced Grant LASSO. The last author is partially affiliated with FI MU, Brno.

References

1. Bisgaard, M., Gerhardt, D., Hermanns, H., Krčál, J., Nies, G., Stenger, M.: Battery-aware scheduling in low orbit: the GomX–3 case. In: Formal Methods, 2016. LNCS, vol. 9995, pp. 559–576. Springer (2016)
2. Boker, U., Henzinger, T.A., Radhakrishna, A.: Battery transition systems. In: ACM SIGPLAN Notices. vol. 49, pp. 595–606. ACM (2014)
3. Chiasserini, C.F., Rao, R.R.: Energy efficient battery management. IEEE Journal on Selected Areas in Communications 19(7), pp. 1235–1245 (2001)
4. David, A., Larsen, K.G., Legay, A., Mikučionis, M., Poulsen, D.B.: Uppaal SMC tutorial. International Journal on Software Tools for Technology Transfer 17(4), pp. 397–415 (2015)
5. David, A., Larsen, K.G., Legay, A., Mikučionis, M., Poulsen, D.B., Van Vliet, J., Wang, Z.: Statistical model checking for networks of priced timed automata. In: Formal Modeling and Analysis of Timed Systems - 9th International Conference. LNCS, vol. 6919, pp. 80–96. Springer (2011)
6. Doyle, M., Fuller, T.F., Newman, J.: Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell. Journal of the Electrochemical Society 140(6), pp. 1526–1533 (1993)
7. Eder, K., Gallagher, J.: Energy-aware software engineering. In: ICT - Energy Concepts for Energy Efficiency and Sustainability. pp. 103–127. InTechOpen (2017)
8. Fehnker, A., van Hoesel, L., Mader, A.: Modelling and verification of the LMAC protocol for wireless sensor networks. In: Integrated Formal Methods. LNCS, vol. 4591, pp. 253–272. Springer (2007)
9. Gold, S.: A PSPICE macromodel for lithium-ion batteries. In: Twelfth Annual Battery Conference on Applications and Advances. pp. 215–222. IEEE (1997)

10. Haemmerlé, R., López-García, P., Liqat, U., Klemen, M., Gallagher, J.P., Hermenegildo, M.V.: A transformational approach to parametric accumulated-cost static profiling. In: International Symposium on Functional and Logic Programming. pp. 163–180. Springer (2016)
11. Heni, M., Bouallegue, A., Bouallegue, R.: Energy consumption model in ad hoc mobile network. International Journal of Computer Networks & Communications 4(3), pp. 207–217 (2012)
12. Jaeger, M.G.B.: Wireless network medium access control protocol (2012), https://www.google.com/patents/US20120120871, US Patent App. 12/945,989
13. Jongerden, M., Haverkort, B., Bohnenkamp, H., Katoen, J.P.: Maximizing system lifetime by battery scheduling. In: 2009 IEEE/IFIP International Conference on Dependable Systems & Networks. pp. 63–72. IEEE (2009)
14. Jongerden, M.R., Haverkort, B.R.: Which battery model to use? IET Software 3(6), pp. 445–457 (2009)
15. Jongerden, M.R., Haverkort, B.R.: Battery aging, battery charging and the kinetic battery model: A first exploration. In: Quantitative Evaluation of Systems. pp. 88–103. Springer (2017)
16. Jongerden, M.R., Haverkort, B.R.: Battery aging, battery charging and the kinetic battery model: A first exploration. In: Quantitative Evaluation of Systems: 14th International Conference. pp. 88–103. Springer (2017)
17. Jørgensen, K.H., Christoffersen, N.B., Petersen, R.D., Gjøderum, T.H.: VisuAAL - an application for visualizing realistic mesh network protocol behavior through Uppaal simulations (2017), Master Thesis, Aalborg University, Department of Computer Science
18. Manwell, J.F., McGowan, J.G.: Lead acid battery storage model for hybrid energy systems. Solar Energy 50(5), pp. 399–405 (1993)
19. Panigrahi, D., Dey, S., Rao, R., Lahiri, K., Chiasserini, C., Raghunathan, A.: Battery life estimation of mobile embedded systems. In: Fourteenth International Conference on VLSI Design, 2001. pp. 57–63. IEEE (2001)
20. Rakhmatov, D.N., Vrudhula, S.B.: An analytical high-level battery model for use in energy management of portable electronic systems. In: Proceedings of the 2001 IEEE/ACM International Conference on Computer-Aided Design. pp. 488–493. IEEE (2001)
21. Rodrigues, L.M., Montez, C., Moraes, R., Portugal, P., Vasques, F.: A temperature-dependent battery model for wireless sensor networks. Sensors 17(2), 422 (2017)
22. Stoer, J., Bulirsch, R.: Introduction to numerical analysis, vol. 12. Springer Science & Business Media (2013)
23. TADIRAN SL-750 Data Sheet. https://tadiranbatteries.de/pdf/lithium-thionyl-chloride-batteries/SL-750.pdf, accessed 2018-06-12
24. Wognsen, E.R., Haverkort, B.R., Jongerden, M.R., Hansen, R.R., Larsen, K.G.: A score function for optimizing the cycle-life of battery-powered embedded systems. In: Formal Modeling and Analysis of Timed Systems - 13th International Conference. LNCS, vol. 9268, pp. 305–320. Springer (2015)
25. Xue, J., Yuan, Z., Zhang, Q.: Traffic load-aware power-saving mechanism for IEEE 802.16 e sleep mode. In: 4th International Conference on Wireless Communications, Networking and Mobile Computing, 2008. pp. 1–4. IEEE (2008)

Appendix A An analytical solution to KiBaM

A.1 The general case

We solve Equation 1 from [18] for an arbitrary current i to provide an analytical solution not only for a piecewise constant current intensity but also for piecewise polynomial intensities.

Let $i = i(t)$, then Equation 2 can be rewritten:

$$\begin{cases} \dot{a} = -i(t) + kcb - k(1-c)a \\ \dot{b} = -kcb + k(1-c)a \end{cases} \tag{8}$$

Let us apply the Laplace transform to Equation 8. We use the differentiation property to derive image functions for $\dot{a}$ and $\dot{b}$:

$$\begin{cases} pA - a_0 = -I + kcB - k(1-c)A \\ pB - b_0 = -kcB + k(1-c)A \end{cases} \tag{9}$$

where $A = \mathcal{L}(\{a\})$, $B = \mathcal{L}(\{b\})$, and $I = \mathcal{L}(\{i\})$ are image functions of the Laplace transform applied to a, b, and i respectively.

Let us add the second equation to the first one and express A:

$$A = \frac{q_0}{s} - \frac{I}{s} - B \tag{10}$$

Let us substitute Equation 10 in the second equation in System 9 and solve it for B:

$$B = \frac{b_0}{p+k} + \frac{q_0 k(1-c)}{p(p+k)} - k(1-c)\frac{I}{p(p+k)} \tag{11}$$

Then we substitute Equation 11 in Equation 10 and obtain a solution for image functions A and B:

$$\begin{cases} A = \frac{q_0}{p} - \frac{I}{p} - \frac{b_0}{p+k} - \frac{q_0 k(1-c)}{p(p+k)} + k(1-c)\frac{I}{p(p+k)} \\ B = \frac{b_0}{p+k} + \frac{q_0 k(1-c)}{p(p+k)} - k(1-c)\frac{I}{p(p+k)} \end{cases} \tag{12}$$

We apply partial-fraction decomposition: $\frac{1}{p(p+k)} = \frac{1}{k}(\frac{1}{p} - \frac{1}{p+k})$. Using Laplace transform for constants and exponents, and the convolution property, we apply the inverse Laplace transform and obtain:

$$\begin{cases} a = a_0 e^{-kt} + q_0 c(1-e^{-kt}) - \int_0^t i(\tau)d\tau + (1-c)\int_0^t i(\tau)(1-e^{k(\tau-t)})d\tau \\ b = b_0 e^{-kt} + q_0(1-c)(1-e^{-kt}) - (1-c)\int_0^t i(\tau)(1-e^{k(\tau-t)})d\tau \end{cases} \tag{13}$$

Equation 13 is the analytical solution to Equation 2 for an arbitrary current $i = i(t)$.

A.2 The polynomial case

We need to solve the integral $\int_0^t i(\tau)(1-e^{k(\tau-t)})d\tau$ for a polynomial $i(t)$ to obtain the solution to KiBaM for a polynomial current. Let us find $\int_0^t \tau^j(1-e^{k(\tau-t)})d\tau$ for $j \geq 1$:

$$\int_0^t \tau^j(1-e^{k(\tau-t)})d\tau = \frac{t^{j+1}}{j+1} - e^{-kt}\int_0^t \tau^j e^{k\tau}d\tau \tag{14}$$

We apply integration by parts to the subtrahend:

$$X_j = \int_0^t \tau^j e^{k\tau}d\tau = \frac{1}{k}\tau^j e^{k\tau}\Big|_0^t - \frac{j}{k}\int_0^t \tau^{j-1}e^{k\tau}d\tau = \frac{1}{k}t^j e^{kt} - \frac{j}{k}X_{j-1} \tag{15}$$

We obtain a recurrent formula for X_j. We derive a direct formula for X_j considering that $X_1 = \frac{1}{k}te^{kt} - \frac{1}{k^2}(e^{kt}-1)$:

$$X_j = \frac{t^j e^{kt}}{k} + e^{kt}\sum_{l=1}^{j-1}\frac{(-1)^l j! t^{j-l}}{k^{l+1}(j-l)!} + \frac{(-1)^j j!}{k^{j+1}}(e^{kt}-1) = e^{kt} j! \sum_{l=0}^{j}\frac{(-1)^l t^{j-l}}{k^{l+1}(j-l)!} + \frac{(-1)^{j+1}}{k^{j+1}} \tag{16}$$

Then we substitute X_j in Equation 14:

$$\int_0^t \tau^j(1-e^{k(\tau-t)})d\tau = \frac{t^{j+1}}{j+1} - j!\sum_{l=0}^{j}\frac{(-1)^l t^{j-l}}{k^{l+1}(j-l)!} - \frac{(-1)^{j+1}e^{-kt}}{k^{j+1}} \tag{17}$$

Equation 17 can be transformed into a more convenient form:

$$\int_0^t \tau^j(1-e^{k(\tau-t)})d\tau = j!\left[\sum_{l=0}^{j+1}\frac{(-1)^l t^{j-l+1}}{k^l(j-l+1)!} + \frac{(-1)^j e^{-kt}}{k^{j+1}}\right] \tag{18}$$

Let $i(t) = \sum_{j=0}^n i_j t^j$. If we substitute the polynomial $i(t)$ in Equation 13 and use Equation 18 we get:

$$\begin{cases} a = a_0 e^{-kt} + q_0 c(1-e^{-kt}) - \sum_{j=0}^n \frac{i_j t^{j+1}}{j+1} \\ \qquad +(1-c)\sum_{j=0}^n j! i_j \left[\sum_{l=0}^{j+1}\frac{(-1)^l t^{j-l+1}}{k^l(j-l+1)!} + \frac{(-1)^j e^{-kt}}{k^{j+1}}\right] \\ b = b_0 e^{-kt} + q_0(1-c)(1-e^{-kt}) \\ \qquad -(1-c)\sum_{j=0}^n j! i_j \left[\sum_{l=0}^{j+1}\frac{(-1)^l t^{j-l+1}}{k^l(j-l+1)!} + \frac{(-1)^j e^{-kt}}{k^{j+1}}\right] \end{cases} \tag{19}$$

Equation 19 is the analytical solution to Equation 2 for a polynomial current $i = \sum_{j=0}^n i_j t^j$.