

Dynamic Optimization of Battery Health in IoT Networks

Kazim Ergun^{*}, Raid Ayoub[†], Pietro Mercati[†], and Tajana Rosing^{*}

^{*}*Dept. of Electrical and Computer Engineering, University Of California, San Diego*

[†]*Intel Corporation*

{kergun, tajana}@ucsd.edu , {raid.ayoub, pietro.mercati}@intel.com

Abstract—The reliability and maintainability of the Internet of Things (IoT) devices become highly important as the number of “things” grows rapidly. The majority of the IoT devices have batteries which age, degrade, and eventually require maintenance. Existing work focuses on ensuring that batteries have sufficient amount of stored charge to operate until they can recharge, but does not consider battery degradation. This leads to high replacement and maintenance costs in large IoT networks. In this paper, we formulate the problem of minimizing battery degradation to improve the lifetime of IoT networks and solve it with Model Predictive Control (MPC) leveraging models for battery dynamics and State of Health (SoH). The battery SoH is modeled using a realistic non-linear model while taking ambient temperature into account. We demonstrate that our solution can improve network lifetime up to 68.5% compared to conventional energy consumption focused algorithms, which use simple linear battery models. The proposed approach achieves near-optimal performance in terms of preserving battery health, staying within 8.7% SoH with respect to an ideal oracle solution on average.

I. INTRODUCTION

The Internet of Things (IoT) is a growing network of heterogeneous devices that have the ability to process and transfer data. IoT will connect more than 20 billion “things” by 2020 according to Gartner Inc. [1]. When IoT is fully realized, the maintenance and diagnostics costs will be enormous, and if not addressed, it can limit the scalability of IoT solutions [3]. Since the majority of these devices are battery-powered, a part of these costs is associated with battery maintainability. Even though battery itself might be cheap, battery replacement, especially for large-scale IoT systems, is often not feasible due to logistical constraints. One example is the High-Performance Wireless Research and Education Network (HPWREN), which have battery-powered sensors deployed on canyon walls and mountain peaks with no road access [2]. In such cases battery replacement involves expensive labor & infrastructure, hence the battery life should be improved to keep the network running for as long as possible.

There is often confusion when discussing battery lifetime because the lifetime for rechargeable and non-rechargeable batteries are described in different ways. Non-rechargeable batteries die, and need to be replaced after their initial charge is completely depleted. Therefore, the indicator for remaining battery life is the State of Charge (SoC). On the other hand, rechargeable batteries can withstand hundreds of charge-recharge cycles, allowing operation for extended periods when combined with energy harvesting solutions, such as solar cells

or thermal energy. Despite their ability to be recharged, these batteries still have limited lifetimes, and require replacement due to aging. In this case, instead of SoC, we need to consider their State of Health (SoH) which is a figure of merit of the physical condition of a battery. SoH degrades due to cycle aging (charge-discharge rate & total amount) and calendar aging (ambient conditions, e.g. temperature) which results in deterioration of battery conditions in the form of internal impedance increase, open voltage decrease, and most importantly, capacity fading. Depending on its application, a rechargeable battery reaches its end of life with an SoH between 70%-80% and needs to be replaced.

Most works in the literature concerning lifetime maximization either consider non-rechargeable batteries and deal with SoC, or assume ideal operation for rechargeable batteries, neglecting the effects of SoH degradation in their management strategies. Our main insight is that if maximum lifetime is targeted in the network, specifically the battery SoH should be considered. Particularly, we observe that the techniques which focus on optimizing the energy consumption of a network do not yield optimal battery life. In this work, we formulate the problem of determining the data flow that minimizes SoH degradation of rechargeable batteries for an IoT network where battery-powered devices have the capability of sensing, processing, and communicating data. The amount of data routed through a device affects the power consumed for communication & computation, which in turn influences the rate of degradation. In our formulation, we model batteries from two different perspectives:

- **Battery Degradation:** The focal point of this paper is the fact that battery SoH degrades at different rates depending on how the battery is used. In the light of this, we can intelligently manage the network to prolong battery lifetime. Hence, we have a model that relates current rate, SoC, depth of charge/discharge and temperature to how SoH degrades.
- **Battery Dynamics:** We use the *Temperature Dependent Kinetic Battery Model* (T-KiBaM) [22], a dynamic model which can describe the nonlinear characteristics of available battery capacity. Not just the net amount, but the way in which the power is consumed, that is, the current-extraction patterns and the employed current levels play a significant role in battery depletion [12]. Therefore, to

realistically capture the influence of power consumption on the battery, it is inaccurate to assume linear energy depletion with respect to the power consumed/current drawn, and a dynamic battery model is needed.

As a result of this dynamic behavior, the solution to our problem considers the battery state over time and therefore, is time-dependent rather than fixed. Hence, we adopt an optimal control formulation and propose a model predictive controller (MPC) solution to dynamically control data flow rates in the network to minimize SoH degradation over a predefined horizon. We evaluate our solution using real-world deployment in a smart home and a large scale IoT network HPWREN. We show that our solution can achieve comparable performance to an “oracle” solution which knows all future data. For comparison, we implement a standard network lifetime maximization method [8] which adopts an ideal battery model with linear energy depletion. We also investigate the impact of ambient temperature on SoH degradation and network lifetime. Furthermore, an example extension to the original problem is presented by regularizing the objective function with an end-to-end delay function.

The rest of the paper is organized as follows. In Section II, we review related work on maximum network lifetime routing and battery degradation management. In Section III, we first start with outlining the overall problem and describing our network model. Next, we build the battery dynamics and investigate the mechanisms behind battery degradation to obtain a closed-form, nonlinear mathematical expression for the SoH of a battery. Lastly, we construct a finite horizon optimal control problem with the goal of determining the data flow to minimize the degradation of an IoT network and present our MPC solution. In Section IV, we provide experimental results and conclude our work by discussing these results.

II. RELATED WORK

There is a significant amount of literature addressing the lifetime of Wireless Sensor Networks (WSNs) and IoT networks. Publications in that area usually consider non-rechargeable batteries with limited energy and maximize the time at which the batteries drain out of energy [8], [17]. A common issue with such techniques is that they do not consider the battery dynamics and find a static route based on linear battery energy depletion assumption. Recent studies [20], [7] involved battery dynamics that are able to capture the “non-ideal” behavior of actual batteries in their optimization formulations. Even though they show that one can achieve a significantly longer lifetime with an optimal routing policy using a non-ideal battery model, the solution does not suit systems with rechargeable batteries.

Another set of publications investigate energy harvesting networks with rechargeable batteries. This work usually tries to develop control algorithms to optimally utilize available energy [9], [15]. However, only a handful of studies consider the degradation of batteries, which is the major factor in determining the lifetime of a network of devices with

rechargeable batteries. A Markov model based mathematical characterization of harvesting-based battery-powered sensor devices was provided in [19], particularly focusing on the impact of battery discharge policy on degradation. The authors show that by using this model, a degradation-aware policy significantly improves the lifetime of the sensor compared to “greedy” policies. We instead search for network-level controls (i.e. routing) compared to finding a policy for single sensor node/device. In [23], the issue of battery degradation is approached from a MAC protocol design perspective. Random MAC protocols can generate bursts of transmissions and idleness which may increase battery degradation rate. To solve this problem, they propose an aging aware binary exponential backoff algorithm to avoid excessive fluctuations. This study is tangential to our work since it touches upon the degradation problem with a small modification on MAC protocols. More recently, a technique was presented in [14] to predict SoH in WSN applications from various battery related parameters, which can contribute to building degradation-aware management strategies for IoT networks.

The degradation of batteries in a network control problem is studied primarily in battery energy storage systems, smart grid, and data centers. In [6], the authors include the battery degradation processes in the optimization and propose a linear programming approach for optimization of degradation & performance in offgrid power systems with solar energy integration. In [13], a model predictive control (MPC) based algorithm with an explicit cost function considering battery degradation is implemented for battery energy storage systems. A recent paper [5] presents a distributed control method that can handle multiple batteries connected to the grid using a high accuracy nonlinear battery model. In the context of data centers, [4] and [18] use nonlinear Lithium-ion battery health degradation model for health-aware optimal control. However, these work are not directly applicable to IoT domain because of the different structure of the network, and additional constraints that the network possesses. In those areas, batteries are often modeled in aggregate fashion. In IoT networks, the batteries from different devices are not physically connected and can only supply energy to the associated device. The devices work together to accomplish a network-level task, but their energy demands are individual which differentiates other domains from IoT.

Network lifetime studies up to this point have been mostly State of Charge (SoC) optimization for non-rechargeable batteries with ideal linear models for battery dynamics. The ones that study rechargeable batteries focus on energy management strategies to optimally utilize energy harvesting solutions. Along with just a few other works, we investigate the State of Health (SoH) of batteries. Complimentary to previous works in this area, we control the network to optimize its lifetime by minimizing SoH degradation. To perform a more accurate optimization, we incorporate battery models which capture the temperature-dependent, nonlinear charging/discharging and degradation behavior into the system model.

III. OPTIMAL NONLINEAR BATTERY CONTROL

A. Problem Overview

The goal of this work is to optimize battery health in IoT networks by controlling data flow rates since each device in the network consumes energy for communication and computation as a function of data flow. The energy amount delivered by the battery depends on both short-term battery dynamics and long-term battery wear. Therefore, while the battery dynamics determine the State of Charge, our control algorithm focuses on optimizing State of Health degradation to ensure long term operation. In the following sections, we start by describing our network model, then build the battery dynamics and investigate the mechanisms behind battery degradation to obtain a closed-form, nonlinear mathematical expression for the SoH of a battery. Table I provides the list of symbols that are used throughout this paper.

TABLE I: Nomenclature

Symbol	Definition
S_i	Set of nodes which node i can send data
$d_{i,j}$	Distance between nodes i and j
$w_{i,j}(t)$	Data flow rate from node i to j
$G_i(t)$	Data generation rate of node i
C_r, C_e, C_c	Reception, sensing, computation energy constants
C_f, C_s	Transmission energy constants
$u_i(t)$	Discharge current of battery i
$r_i(t)$	Charge current of battery i
$i_i(t)$	Net current of battery i
$q_A(t)$	Available charge
$q_B(t)$	Bounded charge
$h_A(t)$	Available charge well height
$h_B(t)$	Bounded charge well height
k	Conductance parameter
C_R	Battery rated capacity
$\delta_i(t)$	Difference between heights of two wells of battery i
$\gamma_i(t)$	Total charge of battery i
$SoC_i(t)$	State of Charge of battery i
V_i	Voltage of battery i
T	Temperature
DoD	Depth of discharge
T_{amb}	Ambient temperature
$SoH_i(t)$	State of Health of battery i
$Deg_i(t)$	SoH degradation of battery i

B. Network Model

We model the IoT networks with three layers: top, middle, and bottom. The top layer represents the wireless mesh backbone of the network. The bottom layer contains sensor nodes and the middle layer is composed of a wireless network of gateways. Each gateway node gathers the data coming from the underlying sensors and delivers it to the backbone layer. These nodes can also perform data analysis and processing.

We consider a model with multiple source and gateway nodes, one base station, and fixed topology. The network consists of N nodes, where nodes from 1 to $N-1$ denote source and gateway nodes and N denote the base station. We assume that the energy supply of the base station is not constrained but all other nodes have a rechargeable battery that can store a limited amount of energy. $SoH_i(t)$ and $SoC_i(t)$ are respectively the State of Health and State of Charge of the battery of node i , $i = 1, \dots, N$ at time t , and the dynamics

of $SoH_i(t)$ is described with details in the next section. The distance between the nodes i and j are denoted by $d_{i,j}$, and is time-independent since we assume fixed topology. Note that relatively infrequent topology changes can be accounted for by periodically recalculating a new control policy.

Let S_i denote the set of nodes to which node i can send packets. Conditions on S_i can be enforced to constrain the behavior of the network, but we only restrict the transmission distance in our problem. Then, $S_i = \{j : d_{i,j} < d_{max}\}$, where d_{max} is the distance of transmission with maximum power. The notation $j \in S_i$ will be used to show node i can communicate with node j . Let $w_{i,j}(t)$ be the data flow rate from node i to node j at time t . The vector $\mathbf{w}(t) = [w_{1,2}(t), \dots, w_{1,N}(t), \dots, w_{N,N}(t)]^T$ defines the control vector in our problem. Let $G_i(t)$ denote the information generation rate at node i , then we can express the total information that needs to be communicated to the gateway as $G_N(t) = \sum_{i < N} G_i(t)$.

We assume every node in the network has a sensor, CPUs, digital signal processors and a radio link. Since we are dealing with nodes that are sensing, computing and receiving/transmitting, the key energy parameters that contribute to discharge current $u(t)$ of node's battery are: the energy needed to sense a bit E_{sense} , receive a bit E_{rx} , transmit a bit E_{tx} , and compute a bit E_{comp} . For a given distance $d_{i,j}$ between nodes i and j , we compute the energy expenditure as follows:

$$E_{tx} = p(d), \quad E_{rx} = C_r, \quad E_{sense} = C_e, \quad E_{comp} = C_c \quad (1)$$

where C_r, C_e, C_c are given constants dependent on the communication, sensing, and computation characteristics of nodes respectively, and $p(d) \geq 0$ is a function monotonically increasing in d ; the most common such function is $p(d) = C_f + C_s d^\beta$ where C_f, C_s are given constants and β is a constant dependent on the medium [21]. For each sensor node i in the network, we can write the discharge current $u_i(t)$ as in (2), where V_i denotes the voltage of the battery .

$$u_i(t) = \frac{1}{V_i} \sum_{j \in S_i} w_{i,j}(t)(p(d_{i,j}(t)) + C_c) + \frac{1}{V_i} \sum_{j | i \in S_j} w_{j,i}(t)C_r + C_e G_i(t), \quad (2)$$

C. Battery Model

1) *Battery Dynamics*: In this work we use Temperature Dependent Kinetic Battery Model (T-KiBaM), an extension to KiBaM [22]. T-KiBaM is able to accurately characterize the two important effects (Rate Capacity effect, and Recovery effect) that make battery performance nonlinear [12]. The effective capacity of a battery drops for higher discharge rates. This effect is termed as *Rate capacity effect*. If there are idle periods in discharging, the battery can partially recover the capacity lost in previous discharge periods. This effect is known as *Recovery effect*. It was shown in [7] that using battery models which captures these effects results in more accurate optimization algorithms, and leads to improvements in network lifetime.

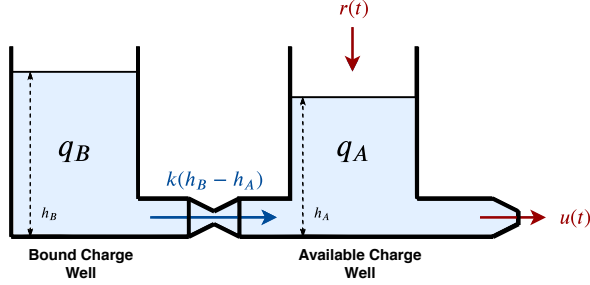


Fig. 1: Temperature Dependent Kinetic Battery Model

As shown in Fig. 1, T-KiBaM models the batteries with two wells, respectively the Bound Charge Well (BCW) and the Available Charge Well (ACW). Three constants are needed for the model: C_R , the rated capacity of the battery; c , the fraction of capacity that may hold available charge; and k , the rate constant. Initially, a part $q_A(0) = cC_R$ of charge is put in the ACW, and a part $q_B(0) = (1-c)C_R$ in the BCW. The charge flow from BCW to ACW through a “valve” with a conductance $k = k_{Arrhenius} = Ae^{-\frac{E_a}{RT_{amb}}}$, a temperature dependent rate constant given by Arrhenius Equation. A is the pre-exponential factor (in s^{-1}), E_a is the activation energy (in KJ/mol), R is the universal gas constant ($8.314 \times 10^{-3} KJ/mol \cdot K$) and T_{amb} is the ambient temperature (in *Kelvin*). The charge flows creating a current $i(t)$ as long as there is a difference between the heights of two wells, i.e. $\delta = h_B - h_A \neq 0$. The heights of these two wells are given by $h_A = q_A/c$ and $h_B = q_B/(1-c)$. Net current $i(t)$ is the difference between an output $u(t)$ representing the discharge outflow due to workload, and a recharge inflow $r(t)$ such that $i(t) = u(t) - r(t)$. The following system of differential equations describes KiBaM.

$$\begin{cases} \frac{dq_A}{dt} = -i(t) + k(h_B - h_A) \\ \frac{dq_B}{dt} = -k(h_B - h_A), \end{cases} \quad (3)$$

For this work we found it convenient to apply a coordinate transformation to variables for using them in the problem formulation. We transform the variables from q_A and q_B to $\delta = h_B - h_A$ (height difference between wells) and $\gamma = q_A + q_B$ (total charge in the battery). Under this transformation, we can write the new differential equations as:

$$\begin{cases} \frac{d\delta}{dt} = \frac{i(t)}{c} - k'\delta \\ \frac{d\gamma}{dt} = -i(t), \end{cases} \quad (4)$$

where $k' = k/c(1-c)$, with initial conditions $\delta(0) = 0$ and $\gamma(0) = C_R$. In the new coordinate system the condition for the battery to be empty is: $\gamma(t) = (1-c)\delta(t)$, meaning that there is no charge left in the available charge well. The two equations in (4) constitute the battery dynamics that is used in optimization problem formulation.

Since batteries provide higher effective capacities at higher temperatures [22], we use a Correction Factor CF to adjust initial battery capacities (C_R) according to the ambient temperature. CF indicates multiplicative gain or loss of the battery capacity at different temperatures. All parameters (CF ,

c , k , C_R) can be obtained using the battery data-sheets, and through experimental measurements. In this paper, we use the parameters obtained by experimental measurements obtained in [22] for Li-Ion batteries.

2) *SoH Degradation Model*: The State of Health (*SoH*) refers to the condition of the battery and the value of *SoH* declines from 1 (healthy battery) to 0 (dead battery) over time due to degradation. For the *SoH* degradation, we employ the model from [11]. Over continuous battery charge/discharge cycles, significant factors that influence the *SoH* degradation of a battery are temperature T , open circuit voltage V_{OC} , and depth of discharge DoD . Knowing that there is a mapping of V_{OC} from State of Charge (*SoC*), we can consider three aspects of the battery for estimating its degradation in this model: T , DoD , and SoC . SoC is defined as the portion of available battery capacity at a given time and DoD is used to describe how deeply a battery is discharged. The formulation of *SoC* directly comes from our Kinetic Battery Model, where we defined γ as the total charge in the battery. The only difference is that *SoC* represents the normalized charge level of the battery, i.e. $SoC = \frac{\gamma}{C_R} \in [0, 1]$. The degradation model used in this paper makes two assumptions: 1) Each of these effects is independent of the others, and 2) The effects themselves are independent of battery age.

Since the effects of T , DoD and SoC on degradation are assumed to be independent, we can write the total battery degradation as $Deg_{total} = Deg_{SoC} + Deg_{DoD} + Deg_T$. The *SoH* at a given time t is $SoH(t) = SoH(0) - Deg_{total}(t)$, and can be expressed explicitly as shown in (5).

$$SoH(t) = 1 - [\phi_1 SoC_{avg}(t) + \phi_2] + [\theta_1 (\Delta SoC(t))^{\theta_2}] - \left[\int_{t'=0}^t \sigma_1 e^{-\sigma_2 (T_{amb} + \sigma_3 |i(t')|)^{-1}} dt' + \sigma_4 T_{amb} \right], \quad (5)$$

The first bracket expression is the term representing the capacity fade degradation attributable to *SoC*. This is based on an approximation that a time period during which the *SoC* with an average of SoC_{avg} has the same effect on battery life as simply staying at SoC_{avg} for the same time period [11]. The second bracket expression is *DoD* related degradation which accounts for capacity fade resulting from *SoC* swing, i.e. maximum *SoC* minus the minimum over an interval. Finally, the last bracket expression is the degradation due to temperature described with a similar exponential model to Arrhenius relation. The temperature change in the battery is given as a linear function of charge current and ambient temperature (T_{amb}). The absolute value of the current $i(t)$ is used so that the expression is both valid for charging and discharging.

We verified our battery degradation model against NREL Li-Ion battery aging dataset [10]. The repository contains various experiment scenarios and physical measurements of cycling batteries until their capacity is reduced below the industry standard of 80% of their original capacity, which is the point where batteries are considered “dead”. The coefficients in the

expression (5) are obtained by fitting the to the experiments under different temperatures and charge/discharge profiles. We selected $\phi_1 = -10^{-3}$, $\phi_2 = 10^{-8}$, $\theta_1 = 25$, $\theta_2 = 0.017$, $\sigma_1 = 1.4 \times 10^{-4}$, $\sigma_2 = -75$, $\sigma_3 = 0.1$, $\sigma_4 = 4 \times 10^{-5}$. Table II shows the error compared to the measurements from batteries tested at 3 different temperatures.

TABLE II: Battery Model Validation

Battery	Temperature	Error
Li-Ion25	4°C	3.1%
Li-Ion5	24°C	1.6%
Li-Ion49	43°C	4.4%

D. Optimal Control Problem Formulation

Our objective is to minimize the SoH degradation of a network by controlling data flow rates $w_{i,j}(t)$. As a common definition, a network is considered dead when any of the nodes die. To prevent this, we particularly try to minimize the accumulated degradation on the most degraded node, since it is the one that will fail the first. Hence, the cost function is $\underset{\mathbf{w}(t)}{\text{minimize}} \max_{i \in N} \text{Deg}_i(T)$, where degradation at end of interval t is decreed by: $\text{Deg}_i(t) = \text{SoH}_i(0) - \text{SoH}_i(t)$.

Next, we define constraints to represent the battery's physical nature:

- **Current Limit:** The discharge and charge power of a battery is limited, thus there are bounds on charge/recharge current, $Lp_i \leq u_i(t) \leq Up_i$ and $Lp_i \leq r_i(t) \leq Up_i$.
- **Charge Limit :** The charge cannot exceed the maximum capacity of the battery, and as stated in Section IIIC the condition for the battery to be empty is: $\gamma(t) = (1 - c)\delta(t)$. Therefore, the corresponding constraint equation is given as $Lc_i \leq \gamma_i(t) \leq Uc_i$ where $Lc_i = (1 - c)\delta(t)$.

Using the battery model, the network model, and constraint equations, the discrete-time optimization problem for a finite interval T is formulated in (6):

$$\underset{\mathbf{w}_t}{\text{minimize}} \quad \max_{i \in N} \text{Deg}_{i,T} \quad (6)$$

$$\text{subject to} \quad \delta_{i,t+1} = \delta_{i,t} + \frac{u_{i,t} - r_{i,t}}{c} - k\delta_{i,t}, \quad \delta_{i,0} = 0, \quad (7)$$

$$\gamma_{i,t+1} = \gamma_{i,t} + (r_{i,t} - u_{i,t}), \quad \gamma_{i,0} - C = 0, \quad (8)$$

$$u_{i,t} = \frac{1}{V_i} \sum_{j \in S_i} w_{i,j}(t)(p(d_{i,j}(t)) + C_c) + \frac{1}{V_i} \sum_{j|i \in S_j} w_{j,i}(t)C_r + C_e G_i(t), \quad (9)$$

$$G_{i,t} = \sum_{j \in S_i} w_{i,j,t} - \sum_{j|i \in S_j} w_{j,i,t}, \quad (10)$$

$$0 \leq w_{i,j,t} \leq W_{max}, \quad (11)$$

$$Lp_i \leq u_{i,t} \leq Up_i, \quad Lp_i \leq r_{i,t} \leq Up_i, \quad (12)$$

$$Lc_i \leq \gamma_{i,t} \leq Uc_i. \quad (13)$$

where (7) and (8) are battery dynamic equations with state variables $\delta_{i,t}$ and $\gamma_{i,t}$ representing node i 's charge level at the time instant t . Workload in terms of data flow for each node i is

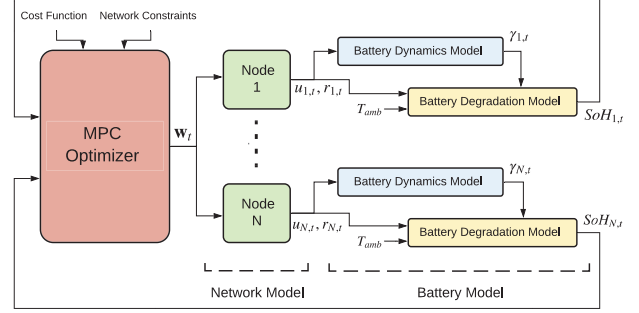


Fig. 2: Block diagram of the proposed MPC solution

expressed by equation (9). Constraints on the control variable \mathbf{w}_t are specified in (10), (11). Finally, (12) and (13) specifies the constraints due to physical limitations of batteries.

Since the solution is based on a finite horizon, two methods are applicable: i) the algorithm is executed once for the complete horizon to get the optimal solution and ii) model predictive control (MPC), where an optimization algorithm is executed at each time interval based on the predicted horizon values and dynamically updated at the next decision interval. Even though the first method gives us the optimal solution for the interval T , it requires knowledge of future (e.g. data generation $G_{i,t}$, current generation $r_{i,t}$, and temperature T_{amb}), thus it is not applicable in practice. For this reason, we employ MPC with the goal of minimizing degradation across the network, and use the first method as a performance benchmark to compare our solution.

As depicted in Fig. 2, the main elements of the discrete time model predictive control are the optimizer and the model. MPC determines the model outputs for the prediction horizon, denoted with M . In the same horizon, the optimizer aims to find the optimal control sequence $\{\mathbf{w}_{k-1+t}, t = 1, \dots, M\}$ for the cost function (6), subject to problem constraints. Only the first element \mathbf{w}_k of the optimized control sequence is applied to the model and the optimization process is repeated at each time step. It is assumed that we have predictions of energy generation, data generation, and ambient temperature for some time into the future within the horizon of the predictive controller. In other words, given a prediction horizon M , we assume knowledge of $r_{i,t}$, $G_{i,t}$, and $T_{amb,i,t}$ for all $t \in \{k, \dots, k + M - 1\}$. When the prediction horizon is less than 24 hours, such an assumption is reasonable as energy generation (e.g solar energy) tends to follow daily patterns and one-day ahead weather predictions can be fairly accurate for ambient temperature.

E. Regularizations

We can regularize the cost function to obtain many different extensions to the original problem of minimizing degradation. Consider an utility function $\phi(\mathbf{w})$, which can be used to model power consumption, delay etc. With a trade-off in network lifetime optimality, the cost function can be regularized as follows to improve network performance in other aspects.

$$\underset{\mathbf{w}_t}{\text{minimize}} \quad \max_{i \in N} \text{Deg}_{i,T} + \lambda \phi(\mathbf{w}) \quad (14)$$

In the following, we show the use of a regularization function for end-to-end delay of a network, although many different utility functions are possible.

End-to-end delay: The end-to-end packet delay from a source node to a sink node depends on the number of hops along its path. As the number of hops increase, the packet will be received by the sink with a higher delay. Queuing delay on the nodes can be assumed negligible because the data rate is low enough for most of the IoT applications to make the number of hops the dominant factor. Instead of directly using the number of hops, we create a metric to provide similar behavior. For a given node i , if its neighbor $j \in S_i$ is farther from the sink than another neighbor $k \in S_i$, then the delay for following a path through node j should be greater compared to node k . Thus, we define:

$$h_{i,j} = \frac{d_{j,N}}{d_{i,N}}, \quad i \in 1, \dots, N-1, \quad j \in S_i \quad (15)$$

To attain the lowest delay, a packet must be forwarded to the neighbor with the minimum h value; the one closest to the sink (node N). A delay function for a node i can be given as:

$$\phi_i(\mathbf{w}) = \sum_{j \in S_i} h_{i,j} w_{i,j}, \quad i \in 1, \dots, N-1, \quad (16)$$

The regularization function should ensure that most of the data traffic is routed through the minimum hop path. Since we are interested in the average delay of the network, the delay function in (16) is summed up over all nodes and averaged over time. The regularization function for end-to-end delay is:

$$\phi(\mathbf{w}) = \frac{1}{T} \sum_{t=1}^T \sum_{i \in N-1} \sum_{j \in S_i} h_{i,j} w_{i,j} \quad (17)$$

IV. EVALUATION

A. Experimental Setup

To illustrate the results of our solution, we consider two examples of real-world deployments: High-Performance Wireless Research and Education Network (HPWREN) [2], and a study on IoT Smart Home developed in our lab. We cover the frequently used mesh and clustered mesh (hierarchical) network topologies with HPWREN and Smart Home cases respectively.

1) *HPWREN*: HPWREN is a heterogeneous wireless sensor network, deployed in the Southern California area. In HPWREN, there are many types of computing systems ranging from the small wireless sensor nodes, single-board computers, to the high-performance server systems at the UCSD Supercomputer Center. It comprises several subnetworks, but we only simulate the Santa Margarita Ecological Reserve (SMER) network which covers a region of 2500m x 1250m with a mesh topology. There are a total of 15 cameras and 1 acoustic sensor deployed, each generating data of different sizes and at different sampling rates. Data sizes range from 20kB to 2MB with a sampling interval 30sec to 1hour. The devices are equipped with solar panels that supply energy to

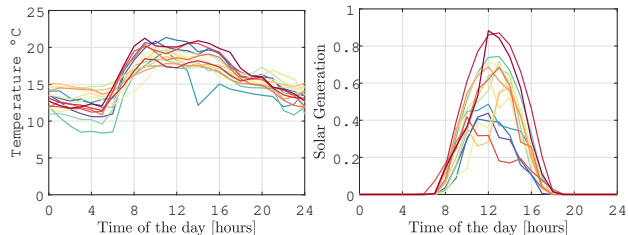


Fig. 3: Temperature and normalized solar generation of 16 nodes in HPWREN during a day.

recharge batteries. We use real temperature data collected from Vaisala WXT520 weather sensors in our battery models as the ambient temperature (T_{amb}) and real solar radiation data from Davis solar sensors to determine the amount of solar power generation ($r_{i,t}$). Fig. 3 depicts the temperature and solar generation profiles of 16 nodes in HPWREN during a day. We estimated the power output of solar panels from solar radiation and used it as the battery charging value for our calculations.

2) *Smart Home*: We have a house instrumented with several off-the-shelf heterogeneous sensors as shown in Fig. 4. Each room in the house has several sensors which help in identifying activities local to that room. These sensors are: (1) kitchen door contact, (2) fridge door contact, (3) kitchen drawer contacts 12, (4) teapot smart-plug, (5) kitchen smart bulb, (6) metasense, (7) airbeam, (8) kitchen angular motion, (9) kitchen locator beacon 1, (10) kitchen cabinet contact 1, (11) kitchen cabinet contact 2, (12) kitchen locator beacon 2, (13) kitchen pantry contact, (14) dining room multi-sensor, (15) dining room locator beacon, (16) living room locator beacon 1, (17) living room motion 1, (18) TV smart plug, (19) living room angular motion, (20) living room motion 2, (21) living room locator beacon 2. There are also two data aggregators (smart hubs), one covering the kitchen and one covering living room & dining room, which aggregate data from the different sensors and send it to the cloud. Since this deployment has the main goal of studying edge processing, all the sensors have a Raspberry Pi Zero associated with them which helps in local processing and data routing. In such a heterogeneous deployment, different sensors send different types of data at various sampling frequencies. Sensors such as door contacts, motion sensors do event based sampling, on the other hand, smart plugs, smart bulbs, angular motion sensors, and air quality sensors sample at constant intervals, ranging from 1/10 sec to 5 sec.

For both experimental scenarios, the coefficients for energy consumption in equation (4) are tuned to fit real sensor hardware according to specifications of the devices. For the battery dynamics model, CF and k parameters are calculated using real ambient temperature data from the temperature sensors. We took $c = 0.5641$ based on the analysis in [22] and set different battery capacities C_R for different devices. The parameters for SoH degradation model is fitted and verified against NREL Li-Ion battery aging dataset [10], and given in Section III.C.

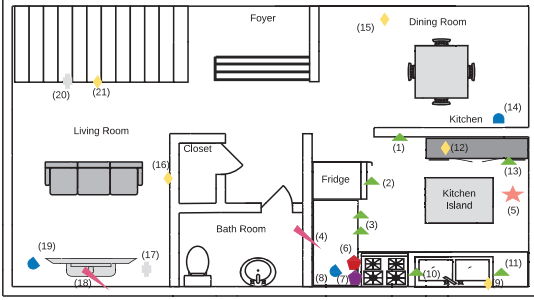


Fig. 4: Sensor deployment in smart home.

B. Experimental Results

In this section, we analyze the amount of SoH degradation in the batteries for our proposed method. For comparison, we have selected: i) an “oracle” optimal solution with knowledge of complete horizon, and ii) an optimization method which involves no degradation model and adopts a “linear” energy depletion assumption as presented in [8]. In contrast to our solution, “oracle” is not applicable in practice, but we use it as a benchmark since it gives the optimal solution over the finite horizon. The “linear” solution aims to optimize the network lifetime by minimizing the total energy consumption of the node with maximum energy consumption. This strategy does not consider the dynamics of the battery or the SoH degradation, and essentially tries to optimize the SoC of the batteries. We denote this method as “linear” since it assumes a linear relation between energy consumption and battery life. We implement all solutions in MATLAB using the YALMIP [16] toolbox.

The “oracle” method requires the knowledge of energy generation, data generation and ambient temperature for all nodes at each time interval in the horizon. The control vector is of size $N \cdot (N - 1) \cdot n_T$, so the solution becomes computationally very expensive for large number of nodes N and long time horizons T , where T consists of n_T time steps. Therefore, we did our simulations over a horizon of 1 month, and time intervals of 1 hour. The SoH value of the node with maximum degradation at the end of 1 month horizon for HPWREN and Smart Home scenarios are given in Table III. Smart Home is divided into kitchen & living room because each room has their own smart hub, so the nodes only send data to their respective hubs creating two clusters in the network. Prediction horizon of 6 hours was used for the proposed MPC solution.

TABLE III: Minimum SoH in the network

	Oracle	Proposed (MPC)
HPWREN	0.9986	0.9983
Smart Home (Kitchen)	0.9987	0.9986
Smart Home (Living Room)	0.9985	0.9985

The overall degradation is very small since the simulation horizon is 1 month (Table III). For the HPWREN experiment, it was observed that the node with the minimum SoH degrades 19.1% more for proposed solution compared to the “oracle”. However, the difference is much smaller in Smart Home, where both methods show nearly identical results.

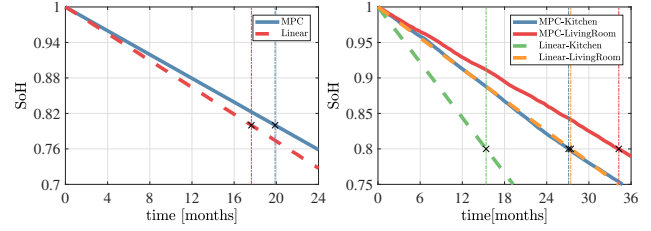


Fig. 5: Minimum SoH in the network simulated until the end of battery life for HPWREN (left) and Smart Home (right)

1) *End of battery life simulations:* Next, we aim to compare with the “linear” method for the time it takes for the first battery to die. In our solution the prediction horizons of MPC are much smaller than the complete horizon, hence we can simulate through the end of battery life without being restricted by computation resources. Since the “linear” method does not consider energy generation and ambient temperature, it should only know the data generation rates for the whole horizon. In both experiment scenarios we have constant data generation rates which makes the solution time invariant. Therefore, we solve the problem in a short horizon and use the same control vector to simulate until end of battery life. The minimum SoH traces are depicted in Fig. 5 for both HPWREN and smart home scenarios. The points where batteries are considered dead ($SoH = 0.8$) shown with vertical lines. By using proposed solution to specifically optimize for SoH of the batteries, we gain 3 months (17.5%) of network lifetime for HPWREN, 11 months (68.7%) for Smart Home (Kitchen), and 7 months (25.0%) for Smart Home (Living Room).

2) *Influence of Prediction Horizon Length:* To study the effect of prediction horizon length on the MPC performance, and for the following sections, we simulated a 50-node network distributed randomly in a square region of size 1000m x 1000m, over 1 month horizon. We assume that we have perfect predictions for the given horizon. Fig. 6b shows that the MPC solution approaches the “oracle” solution as we increase the prediction horizon. If accurate predictions can be made for the disturbances (e.g. ambient temperature, solar radiation) over a long horizon, this can be leveraged in the MPC to improve SoH by increasing the prediction horizon. However, this heavily depends on the use case. For example, in a smart home the usage patterns of the IoT devices may exhibit high variance because of the human factor. In this case it becomes difficult to predict data generation rates of the nodes, and a short prediction horizon should be used. The computation time per MPC update step increases as shown in Fig. 6b. If an hourly update of the controller is preferred as in our experiments, the increasing computation times does not critically affect the choice of the prediction horizon since they are at least an order of magnitude smaller.

3) *Effect of Ambient Temperature on Battery Health:* We compare three cases to analyze the impact of ambient temperature on network life: 1) all nodes are assumed to be under same ambient conditions, 2) nodes have changing ambient temperatures (i.e hourly and daily temperature

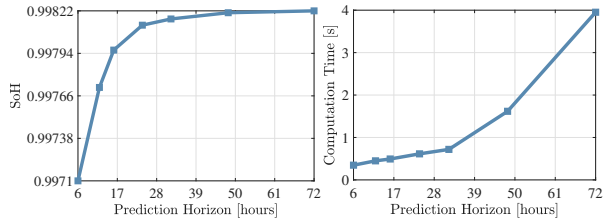


Fig. 6: Influence of prediction horizon length on SoH degradation (left) and computation time (right)

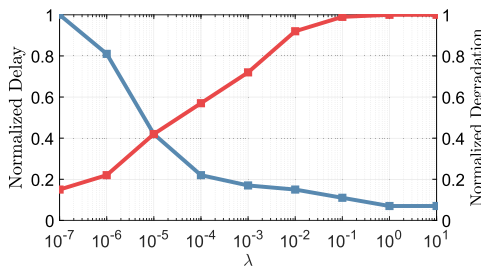


Fig. 7: Normalized delay and normalized degradation by using end to end delay regularization

variations), 3) nodes have constant temperatures, but $\pm 15^\circ\text{C}$ temperature difference with respect to each other. The first case is going to be our reference for assessing the impact of ambient temperature. The second case is the closest to a real life scenario, and the third case may also be plausible if there is an altitude difference between the nodes of the network (e.g mountain top), or if there is an obstacle blocking the sun for one node whereas the other node is exposed to direct sunlight.

The first case with the same constant ambient temperatures for all nodes results in 0.104% SoH degradation for the most degraded node. For varying ambient temperature in case 2, the SoH degradation is 0.118% and very close to the constant temperature scenario. Compared to these two cases, having big temperature differences between nodes generates a much faster degrading network with a SoH degradation of 0.143%.

4) *End-to-end delay*: The normalized delay and degradation results with the additional delay regularization term in the cost function is given in Fig. 7. Results show that there is a trade-off between degradation and delay for different values of λ , which controls the level of regularization. Depending on the application needs, this trade-off can be exploited to design IoT network routing schemes with desired performance & lifetime.

V. CONCLUSION

In this paper, we formulated the problem of minimizing battery degradation to improve the lifetime of IoT networks. We proposed a solution with Model Predictive Control (MPC), leveraging models for battery dynamics and State of Health (SoH). Our work includes the effect of ambient temperature on degradation, and the models we use can accurately capture the nonlinear behavior of actual batteries.

VI. ACKNOWLEDGEMENT

This work was supported by Semiconductor Research Corporation (SRC) task 2805.001.

REFERENCES

- [1] Gartner. (2017) Leading the IoT. https://www.gartner.com/imagesrv/books/iot/iotEbook_digital.pdf.
- [2] High Performance Wireless Research and Education Network (HP-WREN). <http://hpwren.ucsd.edu/>.
- [3] Jasper.(2016)The Hidden Costs of Delivering IIoT Services. https://www.cisco.com/c/dam/m/en_ca/never-better/manufacture/pdfs/hidden-costs-of-delivering-iiot-services-white-paper.pdf.
- [4] B. Aksanli, T. Rosing, and E. Pettis. Distributed battery control for peak power shaving in datacenters. In *2013 International Green Computing Conference Proceedings*, pages 1–8, June 2013.
- [5] A. S. Akyurek and T. S. Rosing. Optimal distributed nonlinear battery control. *IEEE Journal of Emerging and Selected Topics in Power Electronics*, 5(3):1045–1054, Sep. 2017.
- [6] C. Bordin, H.O. Anuta, A. Crossland, I. Lascurain Gutierrez, C.J. Dent, and D. Vigo. A linear programming approach for battery degradation analysis and optimization in offgrid power systems with solar energy integration. *Renewable energy*, 101:417–430, November 2016.
- [7] C. G. Cassandras, T. Wang, and S. Pourazarm. Optimal routing and energy allocation for lifetime maximization of wireless sensor networks with nonideal batteries. *IEEE Transactions on Control of Network Systems*, 1(1):86–98, March 2014.
- [8] Jae-Hwan Chang and L. Tassiulas. Maximum lifetime routing in wireless sensor networks. *IEEE/ACM Transactions on Networking*, 12(4):609–619, Aug 2004.
- [9] M. Gatzianas, L. Georgiadis, and L. Tassiulas. Control of wireless networks with rechargeable batteries [transactions papers]. *IEEE Transactions on Wireless Communications*, 9(2):581–593, February 2010.
- [10] K. Goebel, B. Saha, A. Saxena, J. R. Celaya, and J. P. Christophersen. Prognostics in battery health management. *IEEE Instrumentation Measurement Magazine*, 11(4):33–40, August 2008.
- [11] A. Hoke, A. Brissette, D. Maksimovic, A. Pratt, and K. Smith. Electric vehicle charge optimization including effects of lithium-ion battery degradation. In *2011 IEEE Vehicle Power and Propulsion Conference*, pages 1–8, Sept 2011.
- [12] Marijn R Jongerden and Boudewijn R Haverkort. Which battery model to use? *IET software*, 3(6):445–457, 2009.
- [13] M. Koller, T. Borsche, A. Ulbig, and G. Andersson. Defining a degradation cost function for optimal control of a battery energy storage system. In *2013 IEEE Grenoble Conference*, pages 1–6, June 2013.
- [14] Rafael Lajara, Juan J Pérez-Solano, and José Pelegrí-Sebastiá. Predicting the batteries' state of health in wireless sensor networks applications. *IEEE Transactions on Industrial Electronics*, 65(11):8936–8945, 2018.
- [15] L. Lin, N. B. Shroff, and R. Srikant. Asymptotically optimal energy-aware routing for multihop wireless networks with renewable energy sources. *IEEE/ACM Transactions on Networking*, 15(5):1021–1034, Oct 2007.
- [16] J. Lofberg. Yalmip : a toolbox for modeling and optimization in matlab. In *2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No.04CH37508)*, pages 284–289, Sep. 2004.
- [17] R. Madan and S. Lall. Distributed algorithms for maximum lifetime routing in wireless sensor networks. *IEEE Transactions on Wireless Communications*, 5(8):2185–2193, Aug 2006.
- [18] A. Mamun, I. Narayanan, D. Wang, A. Sivasubramaniam, and H. K. Fathy. Multi-objective optimization to minimize battery degradation and electricity cost for demand response in datacenters. page V002T26A004, 10 2015.
- [19] N. Michelusi, L. Badia, R. Carli, L. Corradini, and M. Zorzi. Energy management policies for harvesting-based wireless sensor devices with battery degradation. *IEEE Transactions on Communications*, 61(12):4934–4947, December 2013.
- [20] S. Pourazarm and C. G. Cassandras. Energy-based lifetime maximization and security of wireless-sensor networks with general nonideal battery models. *IEEE Transactions on Control of Network Systems*, 4(2):323–335, June 2017.
- [21] Theodore S Rappaport et al. *Wireless communications: principles and practice*, volume 2. prentice hall PTR New Jersey, 1996.
- [22] L. M. Rodrigues, C. Montez, R. Moraes, P. Portugal, and F. Vasques. A temperature-dependent battery model for wireless sensor networks. *Sensors*, 17(2), 2017.
- [23] Roberto Valentini, Marco Levorato, and Fortunato Santucci. Aging aware random channel access for battery-powered wireless networks. *IEEE Wireless Communications Letters*, 5(2):176–179, 2016.