



RLMan: an Energy Manager Based on Reinforcement Learning for Energy Harvesting Wireless Sensor Networks

Fayçal Ait Aoudia, Matthieu Gautier, Olivier Berder

► To cite this version:

Fayçal Ait Aoudia, Matthieu Gautier, Olivier Berder. RLMan: an Energy Manager Based on Reinforcement Learning for Energy Harvesting Wireless Sensor Networks. IEEE Transactions on Green Communications and Networking, 2018, pp.1 - 11. 10.1109/TGCN.2018.2801725 . hal-01701688

HAL Id: hal-01701688

<https://hal.science/hal-01701688>

Submitted on 6 Feb 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

RLMan: an Energy Manager Based on Reinforcement Learning for Energy Harvesting Wireless Sensor Networks

Fayçal Ait Aoudia, Matthieu Gautier, Olivier Berder Univ Rennes, CNRS, IRISA

Email: {faycal.ait-aoudia, matthieu.gautier, olivier.berder}@irisa.fr

Abstract—A promising solution to achieve autonomous wireless sensor networks is to enable each node to harvest energy in its environment. To address the time-varying behavior of energy sources, each node embeds an energy manager responsible for dynamically adapting the power consumption of the node in order to maximize the quality of service while avoiding power failures. A novel energy management algorithm based on reinforcement learning, named RLMan, is proposed in this work. By continuously exploring the environment, RLMan adapts its energy management policy to time-varying environment, regarding both the harvested energy and the energy consumption of the node. Linear function approximations are used to achieve very low computational and memory footprint, making RLMan suitable for resource-constrained systems such as wireless sensor nodes. Moreover, RLMan only requires the state of charge of the energy storage device to operate, which makes it practical to implement. Exhaustive simulations using real measurements of indoor light and outdoor wind show that RLMan outperforms current state of the art approaches, by enabling almost 70 % gain regarding the average packet rate. Moreover, RLMan is more robust to variability of the node energy consumption.

I. INTRODUCTION

Many applications, such as smart cities, precision agriculture and plant monitoring, rely on the deployment of a large number of individual sensors forming Wireless Sensor Networks (WSNs). These individual nodes must be able to operate for long periods of time, up to several years or decades, while being highly autonomous to reduce maintenance costs. As refilling the batteries of each device can be expensive or impossible if the network is dense or if the nodes are deployed in a harsh environment, maximizing the lifetime of typical sensors powered by individual batteries of limited capacity is a perennial issue. Therefore, important efforts were devoted in the last decades to develop low power consumption devices as well as energy efficient algorithms and communication schemes to maximize the lifetime of WSNs. Typically, a node quality of service (sensing rate, packet rate...) is set at deployment to a value that guarantees the required lifetime. However, as batteries can only store a finite amount of energy, the network is doomed to die.

A promising solution to increase the lifetime of WSNs is to enable each node to harvest energy in its environment. In

this scenario, each node is equipped with one or more energy harvesters, as well as an energy buffer (battery or capacitor) to allow storing part of the harvested energy for future use during periods of energy scarcity. Various energy sources are possible [1], such as light, wind, motion, fuel cells.... As the energy sources are typically dynamic and uncontrolled, it is required to dynamically adapt the power consumption of the nodes, by adjusting their quality of service, in order to avoid power failure while maximizing the energy efficiency and ensuring the fulfilment of application requirements. This task is done by a software module called Energy Manager (EM). For each node, the EM is responsible for maintaining the node in Energy Neutral Operation (ENO) [2] state, i.e., the amount of consumed energy never exceeds the amount of harvested energy over a long period of time. Ideally, the amount of harvested energy equals the amount of consumed energy over a long period of time, which means that no energy is wasted by saturation of the energy storage device.

Many energy management schemes were proposed in the last years to address the non trivial challenge of designing efficient adaptation algorithms, suitable for the limited resources provided by sensor nodes in terms of memory, computation power, and energy storage. They can be classified based on their requirement of predicted information about the amount of energy that can be harvested in the future, i.e., prediction-based and prediction-free. Prediction-based EMs [2]–[5] rely on predictions of the future amount of harvested energy over a finite time horizon to take decision on the node energy consumption. In contrast with prediction-based energy management schemes, prediction-free approaches do not rely on forecasts of the harvested energy. These approaches were motivated by (i) the significant errors from which energy predictors can suffer, which incur overuse or underuse of the harvested energy, and (ii) the fact that energy prediction requires the ability to measure the harvested energy, which incur a significant overhead [6]. Indeed, most of the EMs require an accurate control of the spent energy, as well as detailed tracking of the previously harvested and previously consumed energies to operate properly.

Considering these practical issues, we propose in this paper RLMan, a novel EM scheme based on Reinforcement Learning (RL) theory. RLMan is a prediction-free EM, whose objective is to maximize the quality of service, defined as the packet rate, i.e., the frequency at which packets are generated (e.g., by performing measurements) and sent, while avoiding power

This work was already partially published in the proceedings of the IEEE International Conference on Communications 2017 (IEEE ICC17).

failure. It is assumed that the node is in the sleep state between two consecutive packet generations and sendings, in order to allow energy saving. RLMan requires only the state of charge of the energy storage device to operate and aims to set the packet rate by both exploiting the current knowledge of the environment and exploring it, in order to improve the policy. Continuous exploration of the environment enables adaptation to varying energy harvesting and energy consumed dynamics, making this approach suitable for the uncertain environment of Energy Harvesting WSNs (EH-WSNs). The problem of maximizing the quality of service in EH-WSNs is formulated as a Markovian Decision Process (MDP), and a novel EM scheme based on RL theory is introduced. The major contributions of this paper are:

- A formulation of the problem of maximizing the quality of service in energy harvesting WSNs using the RL framework.
- A novel EM scheme based on RL theory named RLMan, which requires only the state of charge of the energy storage device, and which uses function approximations to minimize the memory footprint and computational overhead.
- An exploration of the impact of the parameters of RLMan using extensive simulations, and real measurements of both indoor light and outdoor wind.
- The comparison of RLMan to three state-of-the-art EMs (P-FREEN, Fuzzyman and LQ-Tracker) that aim to maximize the quality of service, regarding the capacitance of the energy storage device and the variability of the node task energy cost.

The rest of this paper is organized as follows: Section II presents the related work focusing on prediction-free EMs, and Section III provides the relevant background on RL theory. In Section IV, the problem of maximizing the packet rate in energy harvesting WSNs is formulated using the RL framework, and RLMan is derived based on this formulation. In Section V, RLMan is evaluated. First, the simulation setup is presented. Next, exploration of the parameters of RLMan is performed. Finally, the results of the comparisons of RLMan to three other EMs are shown. Section VI concludes this paper.

II. RELATED WORK

The first prediction-free EM was LQ-Tracker [7], proposed in 2007 by *Vigorito et al.*. This scheme relies on linear quadratic tracking, a technique from control theory, to adapt the duty-cycle according to the current state of charge of the energy buffer. In this approach, the energy management problem is modeled as a first order discrete time linear dynamical system with colored noise, in which the system state is the state of charge of the energy buffer, the controller output is the duty-cycle and the colored noise is the moving average of state of charge increments produced by the harvested energy. The objective is to minimize the average squared tracking error between the current state of charge and a target residual energy level. The authors used classical control theory results to get the optimal control, which does not depend on the colored noise, and the control law coefficients are learned online using

gradient descent. Finally, the outputs of the control system are smoothed by an exponential weighting scheme to reduce the duty-cycle variance.

Peng et al. proposed P-FREEN [8], an EM that maximizes the duty-cycle of a sensor node in the presence of battery storage inefficiencies. The authors formulated the average duty-cycle maximization problem as a non-linear programming problem. As solving this kind of problem directly is computationally intense, they proposed a set of budget assigning principles that maximizes the duty-cycle by only using the current observed energy harvesting rate and the residual energy. The proposed algorithm requires the current state of charge of the energy buffer as well as the harvested energy to take decision about the energy budget. If the state of charge of the energy buffer is below a fixed threshold or if the amount of energy harvested at the previous time slot is below the minimum required energy budget, then the node is operating with the minimum energy budget. Otherwise, the energy budget is set to a value that is a function of both the amount of energy harvested at the previous time slot and the energy storage efficiency.

In [9] the authors proposed to use fuzzy control theory to dynamically adjust the energy consumption of the nodes. Fuzzy control theory proposes to extend conventional control techniques to ill-modeled systems, such as EH-WSNs. With this approach, named Fuzzyman, an intuitive strategy is formally expressed as a set of fuzzy IF-THEN rules. The algorithm requires as an input both the residual energy and the amount of harvested energy since the previous execution of the EM. The first step of Fuzzyman is to convert the crisp entries into fuzzy entries. The so-obtained fuzzy entries are then used by an inference engine which applies the rule base to generate fuzzy outputs. The outputs are finally mapped to a single crisp energy budget that can be used by the node. The main drawbacks of Fuzzyman are that it requires the amount of energy harvested which can be impractical to measure, and the lack of systematic way to set the parameters.

RL theory was used in the context of energy harvesting communication systems [?], [10]. In [?], the authors addressed the problem of maximizing the throughput at the receiver in energy harvesting two-hop communications. The problem was first formulated as a convex optimization problem, and then reformulated using the reinforcement learning framework, so it can be solved using only local causal knowledge. The well-known SARSA algorithm was used to solve the obtained problem. With RLMan, we focus on maximizing the packet rate in point-to-point communication systems.

Hsu et al. [10] considered energy harvesting WSNs with packet rate requirement, and used Q-Learning, a well-known RL algorithm, to meet the packet rate constraints. In this approach, states and actions are discretized, and a reward is defined according to the satisfaction of the packet rate constraints. The aim of the algorithm is to maximize the overall rewards, by learning the Q-values, i.e., the accumulative reward associated with a given state-action pair. The proposed EM requires the tracking of the harvested energy and the energy consumed by the node in addition to the state of charge. Moreover it uses two dimensional look-up tables to store

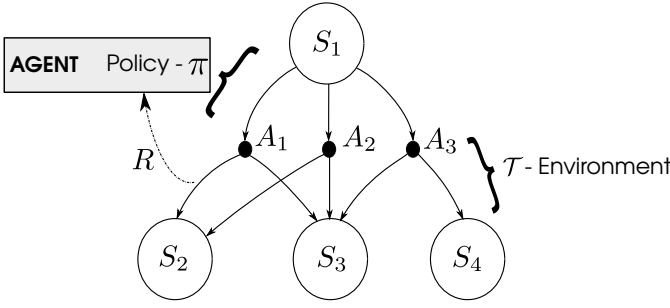


Fig. 1: Markovian decision process illustration.

the Q-values, which incurs significant memory footprint. At each execution of the EM, the action is chosen according to the Q-values using soft-max function, and the Q-value of the last state-action pair observed is updated using the corresponding last observed reward. With RLMAN, we propose an approach that requires only the state of charge of the energy storage device in order to maximize the packet rate. Indeed, measuring the amount of harvested energy or consumed energy requires additional hardware, which incurs additional cost, and increases the form factor of the node. Moreover, linear function approximators are used to minimize the memory footprint and computation overhead, which are critical when focusing on constrained systems such as WSN nodes.

III. BACKGROUND ON REINFORCEMENT LEARNING

RL is a framework for optimizing the behavior of an agent, or controller, that interacts with its environment. More precisely, RL algorithms can be used to solve optimization problems formulated as MDPs. In this section, MDPs in continuous state and action spaces are introduced, and the RL theoretical results required to understand the proposed EM scheme are given.

An MDP is a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$ where \mathcal{S} is the state space, \mathcal{A} is the action space, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ is the state transition probability density function and $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function. In discrete time MDPs, which are considered in this work, at each time step k , the agent is in a state $S[k] \in \mathcal{S}$ and takes an action $A[k] \in \mathcal{A}$ according to a policy π . In response to this action, the environment provides a scalar feedback, called reward and denoted by $R[k+1]$, and the agent is, at the next time slot $k+1$, in the state $S[k+1]$. This process is illustrated in Figure 1. The aim of the RL algorithm is to find a policy which maximizes the accumulated reward called return. In this work, MDPs are assumed to be stationary, i.e., the elements of the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$ do not change over time.

Continuous state and action spaces are considered in this work, however in Figure 1, an illustration of a discrete state and action spaces MDP is shown for clarity. The stochastic process to be controlled is described by the transition probability density function \mathcal{T} , which models the dynamics of the environment. The probability of reaching a state $S[k+1]$ in the region $\mathcal{S}[k+1]$ after taking the action $A[k]$ from the state

$S[k]$ is therefore:

$$\Pr(S[k+1] \in \mathcal{S}[k+1] \mid S[k], A[k]) = \int_{\mathcal{S}[k+1]} \mathcal{T}(S[k], A[k], S) dS. \quad (1)$$

When taking an action $A[k]$ in a state $S[k]$, the agent receives a scalar reward $R[k+1]$ assumed to be bounded. The reward function is defined as the expected reward given a state and action pair:

$$\mathcal{R}(S, A) = \mathbb{E}[R[k+1] \mid S[k] = S, A[k] = A]. \quad (2)$$

The aim of the agent is to find a policy which maximizes the total discounted accumulated reward defined by:

$$J(\pi) = \mathbb{E} \left[\sum_{k=1}^{\infty} \gamma^{k-1} R[k] \mid \rho_0, \pi \right], \quad (3)$$

where $\gamma \in [0, 1]$ is the discount factor, and ρ_0 the initial state distribution. From this last equation, it can be seen that choosing values of γ close to 0 leads to "myopic" evaluation as immediate rewards are preferred, while choosing a value of γ close to 1 leads to "far-sighted" evaluation.

Policies in RL can be either deterministic or stochastic. A deterministic policy π maps each state to an action: $\pi : \mathcal{S} \rightarrow \mathcal{A}$ in a unique way. When using a stochastic policy, actions are chosen randomly according to a distribution of actions given states:

$$\pi(A \mid S) = \Pr(A[k] = A \mid S[k] = S). \quad (4)$$

Using stochastic policies allows exploration of the environment, which is fundamental. Indeed, RL is similar to trail and error learning, and the goal of the agent is to discover a good policy from its experience with the environment, while minimizing the amount of reward "lost" while learning. This leads to a dilemma between exploration (learning more about the environment) and exploitation (maximizing the reward by exploiting known information).

The initial state distribution is denoted by $\rho_0 : \mathcal{S} \rightarrow [0, 1]$, and the discounted accumulated reward is defined by:

$$J(\pi) = \int_{\mathcal{S}} \rho_{\pi}(S) \int_{\mathcal{A}} \pi(A \mid S) \mathcal{R}(S, A) dA dS, \quad (5)$$

where:

$$\rho_{\pi}(S) = \int_{\mathcal{S}} \rho_0(S') \sum_{k=1}^{\infty} \gamma^{k-1} \Pr[S[k] = S \mid S_0 = S', \pi] dS' \quad (6)$$

is the discounted state distribution under the policy π . During the learning, the agent evaluates a given policy π by estimating the J function (5). This estimate is called the value function of π and comes in two flavors. The state value function, denoted by V^{π} , is a function that gives for each state $S \in \mathcal{S}$ the expected return when the policy π is used:

$$V^{\pi}(S) = \mathbb{E} \left[\sum_{i=1}^{\infty} \gamma^{i-1} R[k+i] \mid S[k] = S, \pi \right]. \quad (7)$$

B. Derivation of RLman

An RL agent uses its experience to evaluate a given policy by estimating its value function, a process referred to as policy evaluation. Using the value function, the policy is optimized such as the long-term obtained reward is maximized, a process called policy improvement. The EM scheme proposed in this work is an actor-critic algorithm, a class of RL techniques well-known for being capable to search for optimal policies using low variance gradient estimates [13]. This class of algorithms requires storing both a representation of the value function and the policy in memory, as opposite to other techniques such as critic-only or actor-only methods, which require only storing the value function or the policy respectively. Critic-only schemes require at each step deriving the policy from the value function, e.g., using a greedy method. However, this involves solving an optimization problem at each step, which may be computationally intensive, especially in the case of continuous action space and when the algorithm needs to be implemented on limited resource hardware, such as WSN nodes. On the other hand, actor-only methods work with a parameterized family of policies over which optimization procedure can be directly used, and a range of continuous actions can be generated. However, these methods suffer from high variance, and therefore slow learning [13]. Actor-critic methods combine actor-only and critic-only methods by storing both a parameterized representation of the policy and a value function.

Figure 2 shows the relation between the actor and the critic. The actor updates a parameterized policy π_ψ , where ψ is the policy parameter, by gradient ascent on the objective function J defined in (3). A fundamental result for computing the gradient of J is given by the policy gradient theorem [14]:

$$\begin{aligned}\nabla_\psi J(\pi_\psi) &= \int_{\mathcal{S}} \rho_{\pi_\psi}(e_r) \int_{\mathcal{A}} Q^{\pi_\psi}(e_r, \chi_g) \nabla_\psi \pi_\psi(\chi_g | e_r) d\chi_g de_r \\ &= \mathbb{E} \left[Q^{\pi_\psi}(e_r, \chi_g) \nabla_\psi \log \pi_\psi(\chi_g | e_r) \mid \rho_{\pi_\psi}, \pi_\psi \right].\end{aligned}\quad (14)$$

This result reduces the computation of the performance objective gradient to an expectation, and allows deriving algorithms by forming sample-based estimates of this expectation. In this work, a Gaussian policy is used to generate χ_g :

$$\pi_\psi(\chi_g | e_r) = \frac{1}{\sigma\sqrt{2\pi}} \exp \left(-\frac{(\chi_g - \mu)^2}{2\sigma^2} \right), \quad (15)$$

where σ is fixed and controls the amount of exploration, and μ is linear with the feature:

$$\mu = \psi\phi. \quad (16)$$

Defining μ as a linear function of the feature enables minimal memory footprint as only one floating value, ψ , needs to be stored. Moreover, the computational overhead is also minimal as $\nabla_\psi \mu = \phi$, leading to:

$$\nabla_\psi \log \pi_\psi(\chi_g | e_r) = \frac{(\chi_g - \mu)}{\sigma^2} \phi. \quad (17)$$

It is important to notice that other ways of computing μ from the feature can be used, e.g., artificial neural networks, in which case ψ is a vector of parameters (the weights of the neural network). However, these approaches incur higher memory usage and computational overhead, and might thus not be suited to WSN nodes.

Using the policy gradient theorem as formulated in (14) may lead to high variance and slow convergence [13]. A way to reduce the variance is to rewrite the policy gradient theorem using the advantage function $A^{\pi_\psi}(e_r, \chi_g) = Q^{\pi_\psi} - V^{\pi_\psi}$. Indeed, it can be shown that [14]:

$$\nabla_\psi J(\pi_\psi) = \mathbb{E} \left[A^{\pi_\psi}(e_r, \chi_g) \nabla_\psi \log \pi_\psi(\chi_g | e_r) \mid \rho_{\pi_\psi}, \pi_\psi \right]. \quad (18)$$

This can reduce the variance, without changing the expectation. Moreover, the TD-Error (Temporal Difference-Error), defined by:

$$\delta = R[k+1] + \gamma V^{\pi_\psi}(e_r[k+1]) - V^{\pi_\psi}(e_r[k]), \quad (19)$$

is an unbiased estimate of the advantage function, and therefore can be used to compute the policy gradient [13]:

$$\nabla_\psi J(\pi_\psi) = \mathbb{E} \left[\delta \nabla_\psi \log \pi_\psi(\chi_g | e_r) \mid \rho_{\pi_\psi}, \pi_\psi \right]. \quad (20)$$

Algorithm 1 Reinforcement learning based energy manager.

Input: $e_r[k], R[k]$

- 1: $\phi[k] = \frac{e_r[k] - E_r^{f_{ail}}}{E_r^{max} - E_r^{f_{ail}}} \triangleright$ Feature (12)
 - 2: $\delta[k] = R[k] + \gamma\theta[k-1]\phi[k] - \theta[k-1]\phi[k-1] \triangleright$ TD Error (19), (21)
 - 3: \triangleright Critic: update the value function estimate (22), (23):
 - 4: $\nu[k] = \gamma\lambda\nu[k-1] + \phi[k]$
 - 5: $\theta[k] = \theta[k-1] + \alpha\delta[k]\nu[k]$
 - 6: \triangleright Actor: updating the policy (16), (17), (20):
 - 7: $\psi[k] = \psi[k-1] + \beta\delta[k] \frac{(f[k-1] - \psi[k-1]\phi[k-1])}{\sigma^2} \phi[k-1]$
 - 8: Clamp μ_t to $[X^{min}, X^{max}]$
 - 9: \triangleright Generating a new action:
 - 10: $\chi_g[k] \sim \mathcal{N}(\psi[k]\phi[k], \sigma)$
 - 11: Clamp $\chi_g[k]$ to $[X^{min}, X^{max}]$
 - 12: **return** $\chi_g[k]$
-

The TD-Error can be intuitively interpreted as a critic of the previously taken action. Indeed, a positive TD-Error suggests that this action should be selected more often, while a negative TD-Error suggests the opposite. The critic computes the TD-Error (19), and, to do so, requires the knowledge of the value function V^{π_ψ} . As the state space is continuous, storing the value function for each state is not possible, and therefore function approximation is used to estimate the value function. Similarly to what was done for μ (16), linear function approximation was chosen to estimate the value function, as

it requires very few computational overhead and memory:

$$V_\theta(e_r) = \theta\phi, \quad (21)$$

where ϕ is the feature (12), and θ is the approximator parameter.

The critic, which estimates the value function by updating the parameter θ , is therefore responsible for performing policy evaluation. A well-known category of algorithms for policy evaluation are TD based methods, and in this work, the TD(λ) algorithm [15] is used:

$$\nu[k] = \gamma\lambda\nu[k-1] + \phi[k] \quad (22)$$

$$\theta[k] = \theta[k-1] + \alpha\delta[k]\nu[k] \quad (23)$$

where $\alpha \in [0, 1]$ is a step-size parameter.

Algorithm 1 shows the proposed EM scheme. It can be seen that the algorithm has low memory footprint and incurs low computational overhead, and therefore is suitable for execution on WSN nodes. At each run, the algorithm is fed with the current residual energy $e_r[k]$ and the reward $R[k]$ computed using (11). First, the feature and the TD-Error are computed (lines 1 and 2), and then the critic is updated using the TD(λ) algorithm (lines 4 and 5). Next, the actor is updated using the policy gradient theorem at line 7, where $\beta \in [0, 1]$ is a step-size parameter. The expectancy of the Gaussian policy is clamped to the range of allowed values at line 8. Finally, a frequency is generated using the Gaussian policy at line 10, which will be used in the current time slot. As the Gaussian distribution is unbounded, it is required to clamp the generated value to the allowed range (line 11).

V. EVALUATION OF RLMan

RLMan was evaluated using exhaustive simulations. This section starts by presenting the simulation setup. Then, the impact of the different parameters required by RLMan is explored. Finally, the results of the comparison of RLMan to three state-of-the-art EMs are exposed.

A. Simulation setup

The platform parameters used in the simulations correspond to PowWow [16], a modular WSN platform designed for energy harvesting. The PowWow platform uses a capacitor as energy storage device, with a maximum voltage of 5.2 V and a failure voltage of 2.8 V. The harvested energy was simulated by two energy traces from real measurements: one lasting 270 days corresponding to indoor light [17] and the other lasting 180 days corresponding to outdoor wind [18], allowing the evaluation of the EM schemes for two different energy source types. The task of the node consists in acquiring data by sensing, performing computation and then sending the data to a sink. However, in practice, the amount of energy consumed by one execution of this task varies, e.g., due to multiple retransmissions. Therefore, the amount of energy required to run one execution was simulated by a random variable denoted by E_C which follows a beta distribution. The mode of the distribution was set to the energy consumed if only one transmission attempt is required, denoted by E_C^{typ} , and which

All	E_C^{typ}	8.672 mJ
	E_C^{max}	36.0 mJ
	X^{min}	$\frac{1}{300}$ Hz
	X^{max}	5 Hz
	T	60 s
RLMan	α	0.1
	β	0.01
	σ	0.1
	γ	0.9
	λ	0.9
P-FREEN [8]	B_{OFL}	$0.95E_R^{max}$
	η	1.0
Fuzzyman [9]	K	1.0
	η	1.0
	E_B^{eds}	$X^{min}TE_c^{typ}$
	E_B^{min}	$X^{min}TE_c^{typ}$
	E_H^{strong}	$X^{max}TE_c^{typ}$
	E_H^{weak}	$X^{min}TE_c^{typ}$
LQ-Tracker [7]	μ	0.001
	B^*	$0.70E_R^{max}$
	α	0.5
	β	1.0

TABLE I: Parameter values used for simulations. For details about the parameters of P-FREEN, Fuzzyman and LQ-Tracker, the reader can refer to the respective literature.

is considered as the typical case. The highest value that E_C can take, denoted by E_C^{max} , is set to the energy consumed if five transmission attempts are necessary. The standard deviation of E_C is denoted by σ_C , and the ratio of σ_C to E_C^{typ} is denoted by ξ :

$$\xi = \frac{\sigma_C}{E_C^{typ}}. \quad (24)$$

Intuitively, ξ measures the variability of E_C compared to its typical value.

Three metrics were used to evaluate the EMs: the dead ratio, which is defined as the ratio of the duration the node spent in the power outage state to the total simulation duration, the average packet rate denoted by $\bar{\chi}_g$, and the energy efficiency denoted by ζ and defined by:

$$\zeta = 1 - \frac{\sum_t e_{W,t}}{e_{R,0} + \sum_t e_{H,t}}, \quad (25)$$

where $e_{R,0}$ is the initial residual energy, $e_{W,t}$ is the energy wasted by saturation of the energy storage device during the t^{th} time slot, i.e., the energy that could not be stored because the energy storage device was full, and $e_{H,t}$ is the energy harvested during the t^{th} time slot. Table I shows the values of E_C^{typ} and E_C^{max} , as well as the parameter values used when implementing the EMs.

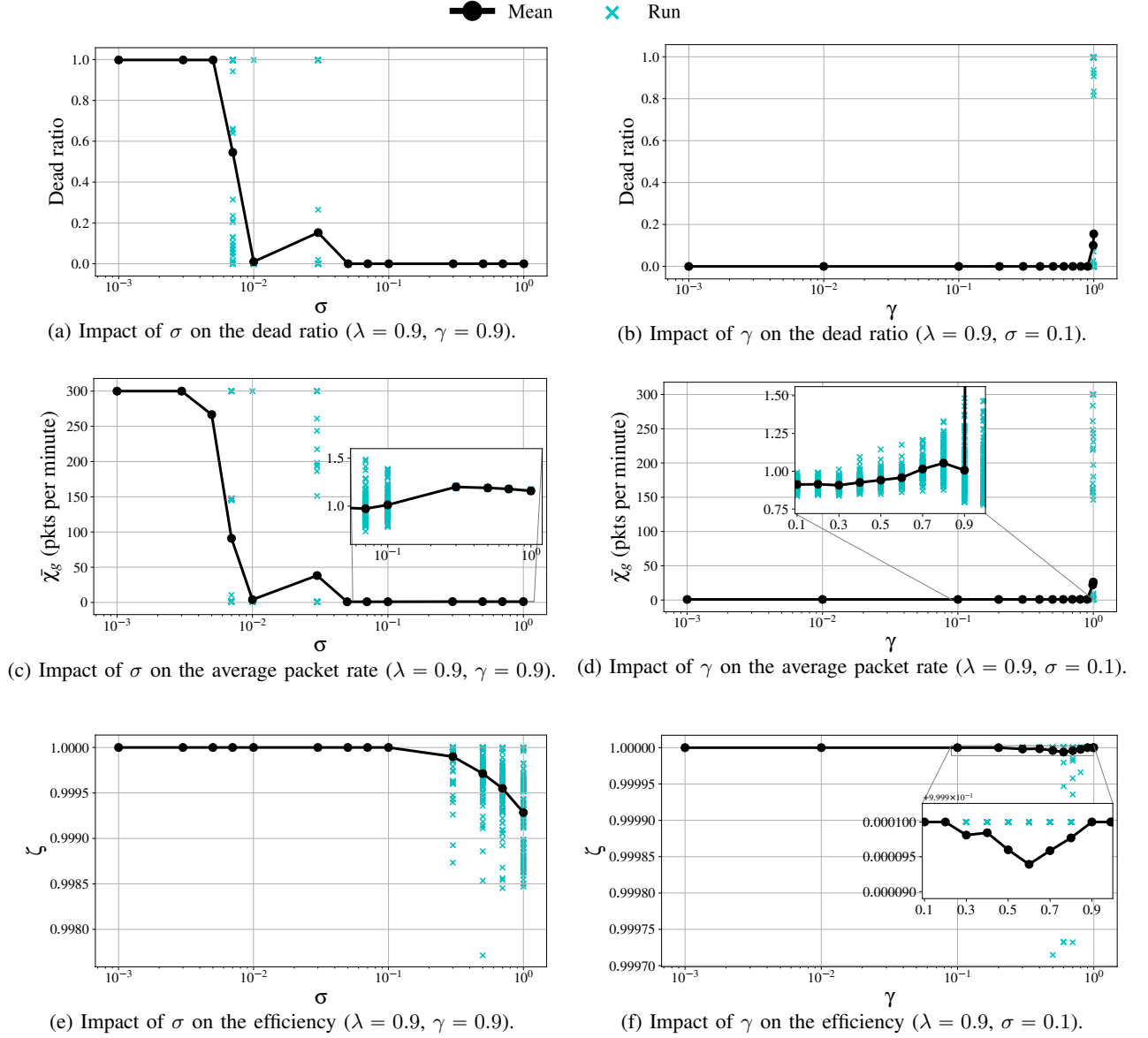


Fig. 3: Impact of the parameters on the performance of RLMan using indoor light energy harvesting.

B. Tuning RLMan

RLMan requires three parameters to be set: the discount factor γ , the trace decay parameter λ and the policy standard deviation σ . Using the setup previously introduced, the impact of these parameters on the performance of the proposed scheme was explored, and the results are shown in Fig. 3. For each value of a parameter, one hundred simulations were run, each performed using different seeds for the random number generators. The capacitance was set to 1 F, as it is the value used by the PowWow platform [16] that was simulated, leading to $E_r^{fail} \approx 3.9$ J and $E_r^{max} \approx 13.5$ J. Each cross in Fig. 3

represents the result of one simulation run, while the dots show the average of all the simulation runs for one value of a parameter. The results show that λ does not significantly impact the performance of RLMan, and therefore the results regarding this parameter are not exposed. Simulations were performed for both indoor light and outdoor wind energy traces, but as the obtained results are similar for both cases, only the results of indoor light are shown.

a) Impact of σ : This parameter is the standard deviation of the Gaussian policy, and therefore controls the amount of exploration performed by the agent. Fig. 3a exposes the impact of σ on the dead ratio. It can be seen that setting this parameter

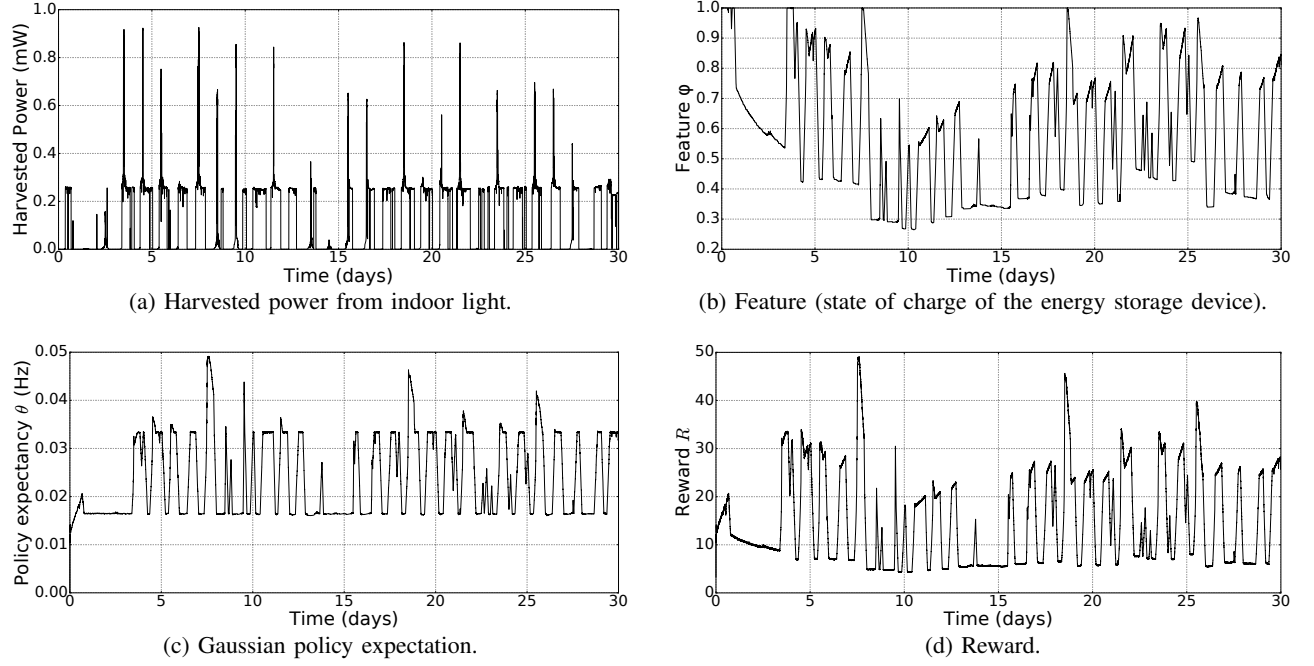


Fig. 4: Behavior of the EM scheme the first 30 days.

to values less than 0.05 leads to non-null dead ratio, which can be explained in Fig. 3c, which shows that values of σ lower than 0.05 lead to too high average packet rates. On the other hand, choosing values of σ higher than 0.1 leads to energy waste, as illustrated in Fig. 3e. As a consequence of these results, σ was set to a value of 0.1 in the rest of this work.

b) Impact of γ : This parameter is the discount factor of the objective function, and therefore controls how much future rewards lose their value compared to immediate rewards. It can be seen on Fig. 3b that values of γ higher than 0.9 lead to power outages. Fig. 3d shows that γ strongly impacts the average packet rate, especially when higher than 0.3. The steep increase of the average packet rate that can be seen for values of γ higher than 0.9 explains the power outages. Moreover, Fig. 3f reveals that choosing values of γ higher than 0.2 and lower than 0.9 leads to energy waste. Therefore, considering these results, γ was set to 0.9 in the rest of this work.

Fig. 4 shows the behavior of the proposed EM during the first 30 days of simulation using the indoor light energy trace, and with $\lambda = 0.9$, $\gamma = 0.9$ and $\sigma = 0.1$. Fig. 4a shows the harvested power, and Fig. 4b shows the feature (ϕ), corresponding to the normalized residual energy. Fig. 4c exposes the expectancy of the Gaussian distribution used to generate the packet rate (μ), and Fig. 4d shows the reward (R), computed using (11). It can be seen that the first day the energy storage device was saturated (Fig. 4b), as the average packet rate was progressively increasing (Fig. 4c), leading to higher rewards (Fig. 4d). As during the second and third days

the amount of harvested energy was low, the residual energy dropped, causing a decrease of the rewards while the policy expectancy was stable. Starting the fourth day, energy was harvested again, enabling the node to increase its activity, as it can be seen on Fig. 4c. Finally, it can be noticed that if a lot of energy was wasted by saturation of the energy storage device the first 5 days, this is no longer true once this period of learning is over.

C. Comparison to state-of-the-art

RLMan was compared to P-FREEN, Fuzzyman, and LQ-Tracker, three state-of-the-art EM schemes that aim to maximize the packet rate. P-FREEN and Fuzzyman require the tracking of the harvested energy in addition to the residual energy, and were therefore executed with perfect knowledge of this value. RLMan and LQ-Tracker were only fed with the value of the residual energy. Both the indoor light and the wind energy traces were considered. First, the EMs were evaluated for different capacitances of the energy storage device, as it impacts the cost and form factor of the WSN nodes. Then, the EMs were compared regarding ξ , which allows us to evaluate the robustness of the algorithms regarding the variability of E_C .

1) Impact of the capacitance of the energy storage device:

The EMs were evaluated for capacitance sizes ranging from 0.5 F to 3.5 F, and for a value of ξ of 0.16. All the EMs successfully avoid power failure when powered by indoor light or outdoor wind. Fig. 5 exposes the comparison results. As it

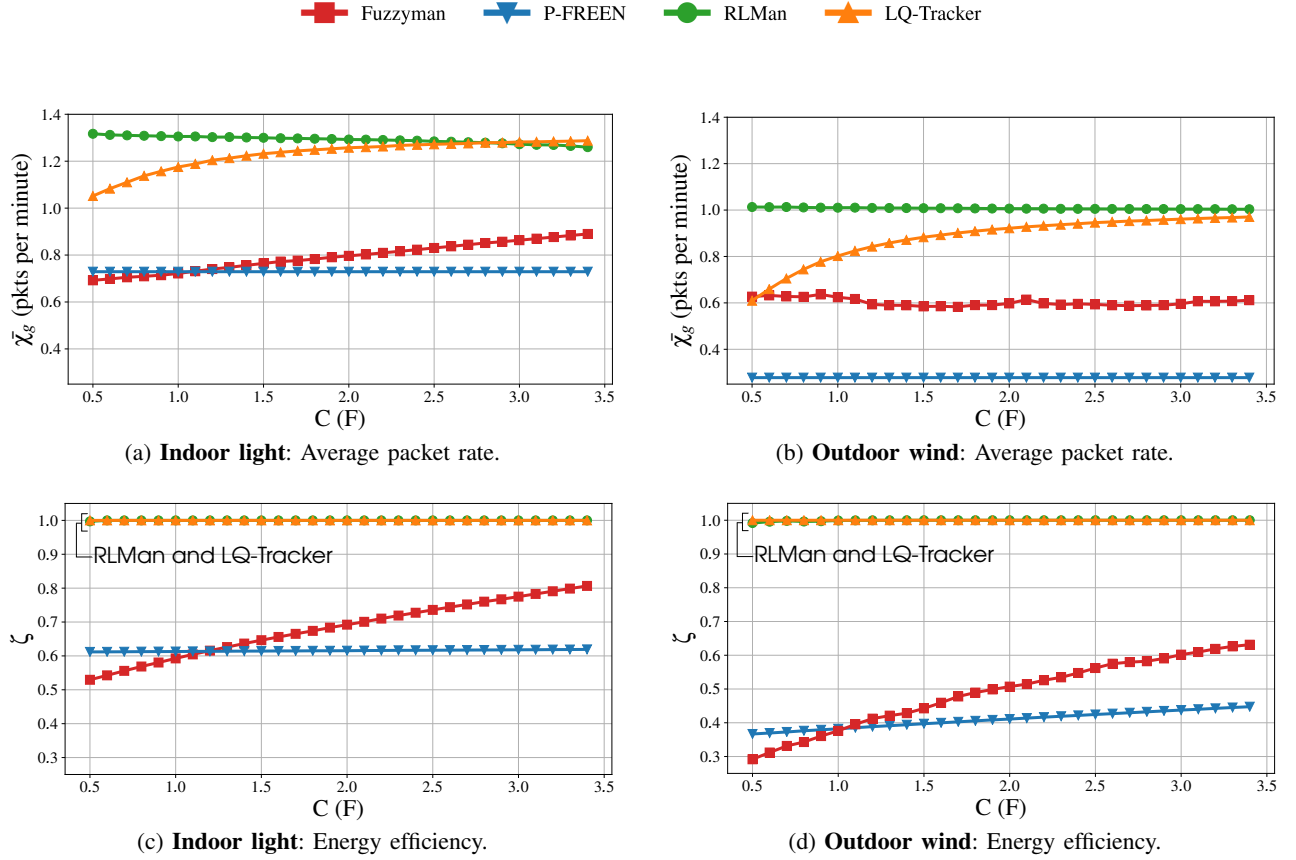


Fig. 5: Average packet rate and energy efficiency for different capacitance values, in the case of indoor light and outdoor wind.

can be seen on Fig 5c and Fig. 5d, both RLMAN and LQ-Tracker achieve more than 99.9 % efficiency, for indoor light and outdoor wind, for all capacitance values, and despite the fact that they require only the residual energy as an input. In addition, when the node is powered by outdoor wind, RLMAN always outperforms the other EMs in terms of average packet rate for all capacitance values, as shown in Fig. 5b. When the node is powered by indoor light, RLMAN also outperforms all the other EMs, except LQ-Tracker when the values of the capacitance are higher than 2.8 F. Moreover, the advantage of RLMAN over the other EMs is more significant for small values of the capacitance. Especially, the average packet rate is more than 20 % higher compared to LQ-Tracker in the case of indoor light, and almost 70 % higher in the case of outdoor wind, when the capacitance value is set to 0.5 F. This is encouraging as using small capacitance leads to lower cost and lower form factor.

2) *Impact of the variability of E_C :* The EMs were evaluated for different values of ξ . E_C^{typ} and E_C^{max} were kept constant, while σ_C was changed by varying the parameters of the beta distribution. The capacitance of the energy storage device was set to 1.0 F, as it is the value used by PowWow. Results are shown in Fig. 6. It can be seen from Fig. 6c and Fig. 6d that

in both the cases of indoor light and outdoor wind, increasing the variability of E_C does not affect the efficiency of RLMAN and LQ-Tracker, both of which achieve an efficiency of 1. Concerning Fuzzyman and P-FREEN, increasing ξ enables higher efficiencies. Indeed, P-FREEN and Fuzzyman outputs are an energy budget, from which a packet rate must be calculated. The calculation of the packet rate was performed based on a typical cost of a task execution E_C^{typ} . However, as ξ increases, E_C tends to be higher than E_C^{typ} more often, which causes less energy waste as more energy is required to achieve the same quality of service.

Fig. 6a and Fig. 6b reveal that RLMAN achieves higher quality of service in term of packet rate. In the case of indoor light, both packet rates decrease as ξ increases. However, the gain of RLMAN over LQ-Tracker becomes more significant as ξ increases. Regarding the case of outdoor wind, the packet rate is not affected by ξ when using RLMAN, but decreases when using LQ-Tracker, showing the benefits enabled by RLMAN. LQ-Tracker output is a duty-cycle in the range $[0, 1]$, from which the packet rate must be calculated. Similarly to what was done with P-FREEN and RLMAN, the typical value of E_C , E_C^{typ} was used for computing the packet rate from the duty-cycle.

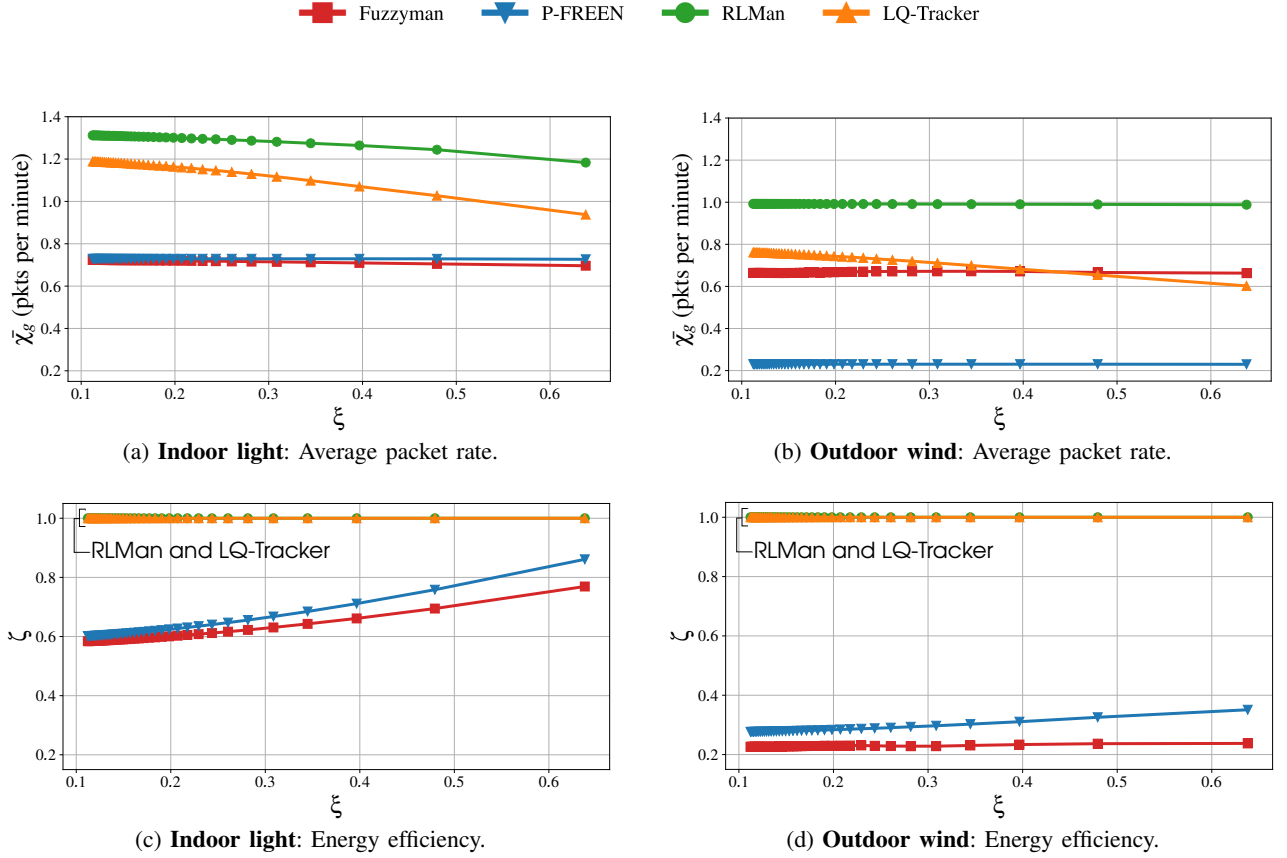


Fig. 6: Average packet rate and energy efficiency for different values of ξ , in the case of indoor light and outdoor wind.

These results show the practical advantages of using RLMan. RLMan does not require any knowledge on the energy consumption of the node. It successfully adapts to variations of the energy consumption of the node task (e.g., due to channel noise or sensors variability), and ensures high energy efficiency and packet rate. One can imagine that the performance of the other schemes could be increased by tracking the energy consumption of the node, and using an adaptive algorithm to compute the packet rate from the output of the EM, but this would require additional resources and increase the complexity, memory and computation footprint of the energy manager. RLMan inherently adapts to the dynamics of both the harvested energy and the consumed energy, and therefore does not require this additional complexity.

VI. CONCLUSION

The problem of designing energy management schemes for wireless sensor nodes powered by energy harvesting is tackled in this paper, and a novel algorithm based on reinforcement learning, named RLMan, is introduced. RLMan uses function approximations to achieve low memory and computational overhead, and only requires the state of charge of the energy storage device as an input, which makes it suitable for

resource-constrained systems such as wireless sensor nodes and practical to implement. It was shown, using exhaustive simulations, that RLMan enables significant gains regarding packet rate compared to state of the art approaches, both in the case of indoor light and outdoor wind. Moreover, RLMan successfully adapts to variations of the consumed energy, and therefore does not require additional energy consumption tracking schemes to keep high efficiency and quality of service.

REFERENCES

- [1] R. J. M. Vullers, R. v. Schaijk, H. J. Visser, J. Penders, and C. V. Hoof, "Energy Harvesting for Autonomous Wireless Sensor Networks," *IEEE Solid-State Circuits Magazine*, vol. 2, no. 2, pp. 29–38, Spring 2010.
- [2] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, "Power Management in Energy Harvesting Sensor Networks," *ACM Transactions on Embedded Computing Systems*, vol. 6, no. 4, 2007.
- [3] A. Castagnetti, A. Pegatoquet, C. Belleudy, and M. Auguin, "A Framework for Modeling and Simulating Energy Harvesting WSN Nodes with Efficient Power Management Policies," *EURASIP Journal on Embedded Systems*, no. 1, 2012.
- [4] T. N. Le, A. Pegatoquet, O. Berder, and O. Sentieys, "Energy-Efficient Power Manager and MAC Protocol for Multi-Hop Wireless Sensor Networks Powered by Periodic Energy Harvesting Sources," *IEEE Sensors Journal*, vol. 15, no. 12, pp. 7208–7220, 2015.

- [5] F. Ait Aoudia, M. Gautier, and O. Berder, "GRAPMAN: Gradual Power Manager for Consistent Throughput of Energy Harvesting Wireless Sensor Nodes," in *IEEE 26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, August 2015.
- [6] R. Margolies, M. Gorlatova, J. Sarik, G. Stanje, J. Zhu, P. Miller, M. Szczodrak, B. Vignham, L. Carloni, P. Kinget, I. Kymissis, and G. Zussman, "Energy-Harvesting Active Networked Tags (EnHANTs): Prototyping and Experimentation," *ACM Transactions on Sensor Networks*, vol. 11, no. 4, pp. 62:1–62:27, November 2015.
- [7] C. M. Vigorito, D. Ganesan, and A. G. Barto, "Adaptive Control of Duty Cycling in Energy-Harvesting Wireless Sensor Networks," in *4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, June 2007.
- [8] S. Peng and C. P. Low, "Prediction free energy neutral power management for energy harvesting wireless sensor nodes," *Ad Hoc Networks*, vol. 13, Part B, 2014.
- [9] F. Ait Aoudia, M. Gautier, and O. Berder, "Fuzzy power management for energy harvesting Wireless Sensor Nodes," in *IEEE International Conference on Communications (ICC)*, May 2016.
- [10] R. C. Hsu, C. T. Liu, and H. L. Wang, "A Reinforcement Learning-Based ToD Provisioning Dynamic Power Management for Sustainable Operation of Energy Harvesting Wireless Sensor Node," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 2, pp. 181–191, June 2014.
- [11] C. Liu, X. Xu, and D. Hu, "Multiobjective Reinforcement Learning: A Comprehensive Overview," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 3, pp. 385–398, March 2015.
- [12] C. R. Shelton, "Balancing Multiple Sources of Reward in Reinforcement Learning," in *Advances in Neural Information Processing Systems 13*, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds. MIT Press, 2001, pp. 1082–1088. [Online]. Available: <http://papers.nips.cc/paper/1831-balancing-multiple-sources-of-reward-in-reinforcement-learning.pdf>
- [13] I. Grondman, L. Busoniu, G. A. D. Lopes, and R. Babuska, "A Survey of Actor-Critic Reinforcement Learning: Standard and Natural Policy Gradients," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1291–1307, November 2012.
- [14] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy Gradient Methods for Reinforcement Learning with Function Approximation," in *Advances in Neural Information Processing Systems 12*, vol. 99, 2000, pp. 1057–1063.
- [15] R. S. Sutton, "Learning to Predict by the Methods of Temporal Differences," *Machine Learning*, vol. 3, no. 1, pp. 9–44, 1988.
- [16] O. Berder and O. Sentieys, "PowWow: Power Optimized Hardware/Software Framework for Wireless Motes," in *International Conference on Architecture of Computing Systems (ARCS)*, February 2010.
- [17] M. Gorlatova, A. Wallwater, and G. Zussman, "Networking Low-Power Energy Harvesting Devices: Measurements and Algorithms," in *IEEE INFOCOM*, April 2011.
- [18] "National Renewable Energy Laboratory (NREL)," <http://www.nrel.gov/>.