MDPI

Article

Wireless Body Area Network Control Policies for Energy-Efficient Health Monitoring

Yair Bar David [1], Tal Geller [1], Ilai Bistritz [2,*], Irad Ben-Gal [1], Nicholas Bambos [2,*] and Evgeni Khmelnitsky [1]

[1] Department of Industrial Engineering, Tel Aviv University, Tel-Aviv 69978, Israel; yairbd2@gmail.com (Y.B.D.); taljgeller@gmail.com (T.G.); bengal@tauex.tau.ac.il (I.B.-G.); xmel@tau.ac.il (E.K.)

[2] Department of Electrical Engineering, Stanford University, Stanford, CA 94305, USA

[*] Correspondence: bistritz@stanford.edu (I.B.); bambos@stanford.edu (N.B.)

Abstract: Wireless body area networks (WBANs) have strong potential in the field of health monitoring. However, the energy consumption required for accurate monitoring determines the time between battery charges of the wearable sensors, which is a key performance factor (and can be critical in the case of implantable devices). In this paper, we study the inherent trade-off between the power consumption of the sensors and the probability of misclassifying a patient's health state. We formulate this trade-off as a dynamic problem, in which at each step, we can choose to activate a subset of sensors that provide noisy measurements of the patient's health state. We assume that the (unknown) health state follows a Markov chain, so our problem is formulated as a partially observable Markov decision problem (POMDP). We show that all the past measurements can be summarized as a belief state on the true health state of the patient, which allows tackling the POMDP problem as an MDP on the belief state. Then, we empirically study the performance of a greedy one-step look-ahead policy compared to the optimal policy obtained by solving the dynamic program. For that purpose, we use an open-source Continuous Glucose Monitoring (CGM) dataset of 232 patients over six months and extract the transition matrix and sensor accuracies from the data. We find that the greedy policy saves ≈50% of the energy costs while reducing the misclassification costs by less than 2% compared to the most accurate policy possible that always activates all sensors. Our sensitivity analysis reveals that the greedy policy remains nearly optimal across different cost parameters and a varying number of sensors. The results also have practical importance, because while the optimal policy is too complicated, a greedy one-step look-ahead policy can be easily implemented in WBAN systems.

Keywords: wireless body area networks; controlled sensing; energy efficiency; partially observable Markov decision processes (POMDPs); remote health monitoring

1. Introduction

WBANs enable the estimation of the physical and health condition of a patient with unprecedented precision as well as the ability to monitor the evolution of these heath conditions over time [1–5]. The control of WBANs involves activating a set of sensors that are located on or near a patient (e.g., wearable) while collecting data related to their physiological and mental activities. The sensing process is controlled by a computing unit, which selects a subset of sensors to be activated at a given time, analyzes the obtained measurements, and transmits the information and initial results to a central unit (server) for further analysis and decision making.

Recent research and practical developments in sensors and WBAN technology allow efficient implementations of WBANs in modern real-world applications [3,4,6]. One of the main factors that impacts the performance of such systems in the context of health monitoring is the inherent trade-off between minimizing the energy consumption of operating the sensors and maximizing the accuracy of the health state estimation based on the

measured information [1,2]. In this paper, we focus on this trade-off between energy costs and accuracy and propose a tractable model that allows designing efficient algorithms that balance this trade-off.

Our goal is to design an efficient dynamic policy that the controller will use to select a subset of sensors at each epoch, which maintains highly accurate knowledge of the patient's health state while saving as much energy as possible.

We formulate a dynamic health-sensing model based on partially observable Markov decision processes (POMDPs). A POMDP is defined by a set of states, a set of actions, the transition probabilities between states, a cost function, and a set of observations [7].

To convert our POMDP into an MDP, we show that all the information captured in all the past outputs can be summarized using a belief vector, which is a probability vector over the possible health states. This belief state is the state of our MDP. The analysis of the MDP essentially allows computing the optimal policy for the problem by means of dynamic programming.

By discretizing the continuous belief state, we can numerically approximate the optimal policy for our dynamic program with an arbitrary degree of accuracy. However, computing the optimal policy has a very high computational complexity that renders it impractical. Even if the optimal policy can be computed offline given the patient parameters, it requires storing very large policy tables on the controller, which should be a simple and efficient device. An alternative is the greedy one-step look-ahead policy, which is generally a suboptimal compromise. Its performance compared to the optimal policy depends on the problem parameters, among them the transition matrix between the health states, the sensor accuracies, and the cost function parameters.

We study the empirical performance of the greedy policy using an open-source Continuous Glucose Monitoring (CGM) dataset [8], which includes the measurements of 232 patients from different age groups and backgrounds over $\approx$6 months. We estimate the transition matrix and sensor accuracies from the data and then evaluate the greedy policy over these 232 patients. Our empirical study includes sensitivity analysis for the cost parameters and the number of sensors that are external to the data. We also study the performance of the greedy policy using synthetic simulations over a wider range of parameters.

Our results show that while the greedy algorithm is suboptimal in general, it is an appealing choice for CGM. The greedy policy saves $\approx$50% of the energy costs while reducing the accuracy cost by less than 2% (i.e., reducing the numerical cost by a factor of 0.98 and not the accuracy itself). Our results are encouraging, since they suggest that in certain cases, the simple and efficient greedy policy can be used for WBAN control in practice while losing very little compared to the optimal intractable policy.

1.1. Related Work

Optimization techniques for WBANs regularly deal with optimizing the hardware and communication components, such as wireless communication protocols between the sensors and the controlling unit, to reduce energy consumption [9–11]. Another stream of research has aimed to increase energy efficiency by optimizing the controlling algorithms, which usually concern sensor selection based on the information gained through the system's activities [3,12], or optimizing the resource allocation in the system [13–15]. Another important aspect of designing a WBAN system is to ensure the confidentiality and integrity of the data, since any malicious attempt on the data the WBAN system collects can have dire consequences [5,16].

To improve the energy efficiency of the WBAN system, it is also important to reduce the volume of the transmitted data, which has been the focus of several studies [16–19]. In contrast, we focus on the energy costs of sensing and operating the sensors. To design a full system, our approach can be combined with that of [2,4,20,21] in a modular manner.

POMDPs have been used to model the behavior and information transitions in WBANs [2,4,20,21]. Prior work on WBAN control used simplifying assumptions such as allowing only homogeneous or identical sensors and "perfect sensing information" [7,9,11].

We extend the POMDP model to allow multiple non-identical noisy sensors, which yield a vector of sensor outputs at each epoch.

1.2. Outline

The rest of the paper is organized as follows. Section 2 formulates our health sensing Markov model and our joint energy-accuracy objective. Section 3 uses the belief state approach to convert our POMDP problem to a MDP problem. Section 4 utilizes the convenient MDP form to propose a one-step look-ahead policy, and it also shows how to compute the optimal policy, which serves as a benchmark to the proposed policy. Section 5 provides the empirical analysis of the proposed approach tested on a real-time Continuous Glucose Monitoring (CGM) dataset. Section 6 provides extended empirical results, which include sensitivity analysis several generalizations of our model.

2. Health Sensing Model

2.1. Patient Health States

We first discuss the health state transitions. We assume a finite set of patient health states, denoted by $\mathcal{H} = \{h_1, h_2, \ldots, h_J\}$. The health states can represent the process of a given disease or different health conditions or a mixture of both. If the health states represent severity levels of a disease, then the set $\mathcal{H}$ will be ordered such that for each pair $h_j, h_{j'}$ where $j < j'$, h_j is considered less healthy than $h_{j'}$. Accordingly, h_J is considered the healthiest state, and h_1 is considered the least healthy state. In some applications, h_1 can be considered a terminal health state in which the monitoring of the individual is no longer relevant (e.g., hospitalization, patient mortality, etc.). If the health states represent different conditions, then there is no simple order between them; for example, if the states are swine flu, COVID-19, and an allergic response.

We assume that the health state follows a discrete-time Markov chain, and we call each time step t an epoch, which is the time steps when the controller makes a decision. The transition probability between any two states during two consecutive epochs, $h^t, h^{t+1} \in \mathcal{H}$, is given by a transition matrix T, in which the (i, j) element is given by:

$$T_{ij} = Pr(h^{t+1} = h_j \big| h^t = h_i). \tag{1}$$

In practice, the transition probabilities can be estimated using historical clinical data. Therefore, we assume that T is known to the controller and is fixed throughout the sensing (monitoring) period.

2.2. Sensors

Consider a WBAN that consists of N sensors operating over a time horizon of L epochs. Each sensor may be responsible for collecting different types of information (e.g., different vital signs). The information collected by the sensors is used to estimate the patient's health state, which is unknown to the controller at all times. At each decision epoch, t, the controller chooses to activate some subset of the N sensors from the action set $\mathcal{S}$ of all possible activation subsets. Since some combinations of sensors cannot be realized due to physical or physiological limitations, the size of $\mathcal{S}$ might be smaller than 2^N. Naturally, reducing the size of the set $\mathcal{S}$ reduces the complexity of the controller's decision making. The sensor activation status at epoch t is denoted by $\mathbf{s}^t = (s_1^t, s_2^t, \ldots, s_N^t)$, where $s_n^t = 1$ refers to an activated sensor, and $s_n^t = 0$ refers to a deactivated sensor, for $n = 1, \ldots, N$. The active sensors are selected to optimize a cost function (defined in Section 2.3) that considers both the energy consumption and misclassification costs. For ease of notation, the time index may be omitted when discussing the general time-independent properties.

Let l_n^t denote the output of sensor n at time t, while $\mathbf{l}^t$ is the output vector of all the sensors at t, $L(\mathbf{s}^t)$ is the set of all the possible output vectors for a sensor activation vector $\mathbf{s}^t$, and $\{\mathbf{l}^t\}_{t=1}^E$ is a sequence of sensor outputs during E epochs. We assume that given a patient's health state, the probability of a certain sensor output is known. For example,

given the particular health state of a patient with diabetes, the probability that a blood sugar level sensor will return a certain value may be obtained from a known distribution.

WBAN sensors are typically power-constrained and low-cost. To account for this, we consider detectors that can only sense specific threshold crossings. Hence, each sensor can only provide output from a set of discrete outputs with size Λ. While any number of thresholds is possible in our model, we are mainly motivated by a small number of thresholds ($\Lambda = 2, 3$), since then, the sensors are much simpler to implement and operate, which reduces their battery consumption.

For example, if $\Lambda = 3$, then $l_n^t \in \{0, 1, 2, \varnothing\}$, where $\varnothing$ denotes a deactivated sensor. Suppose that there are three sensors, s_1, s_2 and s_3, and only s_1 and s_2 have been activated, so $\boldsymbol{s} = (1, 1, 0)$. Then, $L(\boldsymbol{s}) = \{(2, 2, \varnothing), (2, 1, \varnothing), (2, 0, \varnothing), (1, 2, \varnothing), (1, 1, \varnothing), (1, 0, \varnothing), (0, 2, \varnothing), (0, 1, \varnothing), (0, 0, \varnothing)\}$.

We define $A_n^{\Lambda \times J}$ as the output probabilities matrix. Each component A_{nij} expresses the probability of observing output "i" from sensor n given that the individual is currently in health state h_j:

$$A_{nij} = Pr(l_n = i | h_j) \ n = 1, \dots, N; i = 0, \dots, \Lambda - 1 ; j = 1, \dots, J. \tag{2}$$

Definition 1. *Define the output function of sensor n as:*

$$f_n(l_n, j) = Pr(l_n \,|\, h = h_j) = \begin{cases} A_{n(\Lambda-1)j} & l_n = \Lambda - 1 \\ \dots & \dots \\ A_{n1j} & l_n = 1 \\ A_{n0j} & l_n = 0 \\ 1 & l_n = \varnothing \end{cases}. \tag{3}$$

Note that since a deactivated sensor ($l_n = \varnothing$) provides no information, its output function should return 1 in order to neutralize its impact on the probability of the output of the activated sensors.

We assume independence between the sensor outputs given a patient's true health state. This is a natural assumption, since it is only the health state that couples the sensors' outputs, and each is affected by its own local noise. Therefore, we may define the output function of all sensors as:

$$f(\boldsymbol{l}, j) = Pr(\boldsymbol{l} \,|\, h = h_j) = \prod_{n=1}^{N} Pr(l_n \,|\, h = h_j) = \prod_{n=1}^{N} f_n(l_n, j). \tag{4}$$

2.3. Power and Misclassification Costs

Our goal is to develop algorithms that account for both the cost of activating a set of sensors and the misclassification cost associated with possible errors in determining the patient's health state. The cost of activating a set of sensors, $\boldsymbol{s}$, is denoted by $C(\boldsymbol{s})$. The formulation of the misclassification cost is often more complicated and is related to the false positive and false negative classification errors. If the patient's actual health state is better than the estimated one, the system incurs a false positive error cost, which is proportional to the probability of such error. Similarly, the false negative error cost is proportional to the probability that the actual heath state of the patient is worse than the estimated one. We denote the false positive and false negative cost parameters by C_{FP} and C_{FN}, respectively. Since the actual health state is unknown and random, the system may incur both false positive and false negative costs at the same time. The following definition formulates the considered misclassification cost.

Definition 2. *The misclassification cost per health state, h_j, denoted by $\rho_j\left(\{I^\tau\}_{\tau=1}^t\right)$, is defined as:*

$$\rho_j\left(\{I^\tau\}_{\tau=1}^t\right) = C_{FP} \sum_{i=1}^{j-1} Pr\left(h^{t+1} = h_i \,\middle|\, \{I^\tau\}_{\tau=1}^t\right) + C_{FN} \sum_{i=j+1}^{J} Pr\left(h^{t+1} = h_i \,\middle|\, \{I^\tau\}_{\tau=1}^t\right). \tag{5}$$

As indicated above, we multiply the probability that the patient is in a state worse than a specific health state h_j^t by the false-positive cost parameter C_{FP}, and we multiply the probability that the patient is in a state better than h_j^t by the false negative cost parameter C_{FN}. A more sophisticated cost structure is possible. In particular, a health state's specific costs and/or dynamic costs can be incorporated in the proposed model.

Then, the total estimated misclassification cost is defined as:

$$\rho\left(\{I^\tau\}_{\tau=1}^t\right) = \sum_{j=1}^{J} Pr\left(h^{t+1} = h_j \,\middle|\, \{I^\tau\}_{\tau=1}^t\right) \cdot \rho_j\left(\{I^\tau\}_{\tau=1}^t\right). \tag{6}$$

2.4. Motivating Example

As we show in Section 6, by wisely choosing the subset of sensors to activate, much energy can be saved with only a small reduction in the average misclassification cost compared to activating all sensors. In this subsection, we illustrate this idea given a simple example. We examine a scenario where patient's health condition can be classified into one of three health states, $\mathcal{H} = \{h_1, h_2, h_3\}$. We assume that for each epoch t, the patient may either transit to the neighboring less health state or remain in the current state. For h_1, the patient may either remain in h_1 or transit to the healthiest state h_3. The Markov chain is depicted in Figure 1, for some probabilities $0 < \tau_1, \tau_2, \tau_3 < 1$.
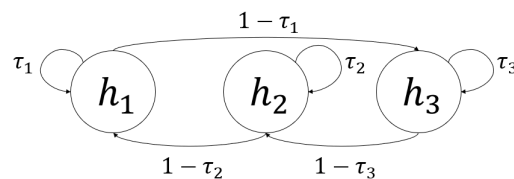

Figure 1. Schematic description of the Markov chain.

We also assume three completely accurate sensors with possible outputs $\{0, 1, \varnothing\}$, where each sensor can perfectly detect a particular health state. The sensors outputs probabilities matrices are:

$$A_1 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix} \quad A_2 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad A_3 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

If the patient is in state h_1, the first sensor would output a signal "1", while the other two sensors would output "0". Similarly, if the patient is in state h_2 or h_3, the corresponding sensor would output "1", while the other two would output "0".

The optimal policy in this scenario only requires activating a single sensor throughout the sensing period (except for at t_0) and still provides perfectly accurate sensing. At first, the controller activates all sensors to identify the initial health state of the patient. From there on, at each epoch, the controller only activates the sensor that corresponds to the current health state to identify whether the patient has transitioned to the next health state or remained in the current health state. Compared to the naïve solution of constantly activating all sensors, the optimal solution results in a significant reduction of energy consumption (a single sensor is activated) while preserving perfect knowledge of the patient's health state. Even with imperfect but still fairly accurate sensors, activating a single sensor in this manner will result in a small misclassification cost while reducing the energy consumption dramatically.

The optimal policy in the above scenario is intuitive and easy to guess. Once the sensor output probabilities matrix is not deterministic or the transition matrix has a different

structure, computing the optimal solution amounts to solving the POMDP, which in general is highly complicated. As done in the literature, the first step to solve the POMDP is to summarize the information captured in the past measurements using a compact belief state on the health state of the patient. This converts the POMDP into a belief state MDP for which we can, in principle, compute the optimal policy by dynamic programming.

3. Belief States

The patient's actual health state is unknown to the controller, and our goal is to monitor and estimate this state. Therefore, the controller uses the sensors' outputs to produce a probability distribution over the set of health states. This distribution is defined as a belief state.

Definition 3. *The belief state $\boldsymbol{b}^t = \left(b_1^t, \ldots, b_J^t \right)$ is defined such that $b_j^t = Pr\left(h^t = h_j \mid \{l^\tau\}_{\tau=1}^t \right)$ for every j.*

The belief state is the probability distribution of the health states of the patient given the past measurements. Next, we prove that tracking belief states instead of all past measurements comes with no loss of optimality (i.e., it is a sufficient statistic) and that it can be recursively updated. During the monitoring period, the belief states evolve based on the outputs, $\boldsymbol{l}^t$, received from the activated sensors, $\boldsymbol{s}^{t-1}$. Below is a schematic description of the order of events in two specific subsequent epochs:

As shown in Figure 2, (a) health states h^t evolve according to the Markov chain transition matrix as an independent process. (b) The subset of activated sensors $\boldsymbol{s}^t$ at each epoch provides outputs $\boldsymbol{l}^{t+1}$ during the following epoch $t + 1$. (c) The activated sensor outputs also depend on the health state. Using the activated sensor outputs $\boldsymbol{l}^{t+1}$ and the belief state $\boldsymbol{b}^t$, we update the belief state $\boldsymbol{b}^{t+1}$. (d) Finally, given the updated belief state, the controller selects the subset of activated sensors $\boldsymbol{s}^{t+1}$ at the next epoch.
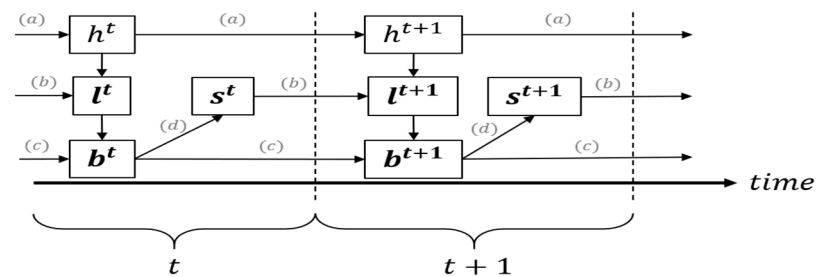

Figure 2. Order of events throughout sensing epochs.

Markovian Property of Belief States

In this subsection, we show that the past sensor measurements can be summarized as a belief state vector, which is a Markov chain. This converts the POMDP into a belief state MDP. Note that in contrast to a generic POMDP, our cost function is by itself an average taken over the belief states.

Lemma 1. *The belief state b^t is Markovian. For any $0 \leq b \leq 1$, the transition probabilities are given by*

$$Pr\left(b_j^{t+1} = b \mid \{l^\tau\}_{\tau=1}^t \right) = Pr\left(b_j^{t+1} = b \mid \boldsymbol{b}^t \right) = \sum_{l \mid b = b_j^{t+1}(\boldsymbol{b}^t, l)} Pr\left(l^{t+1} = l \right) \qquad (7)$$

where we define

$$b_j^{t+1}\left(\boldsymbol{b}^t, \boldsymbol{l} \right) \triangleq \frac{\sum_i T_{ij} b_i^t f\left(l^{t+1}, j \right)}{\sum_k \sum_m f\left(l^{t+1}, k \right) T_{mk} b_m^t}. \qquad (8)$$

Proof. Observe that

$$
\begin{aligned}
Pr\left(h^{t+1} = j \,\middle|\, \{I^\tau\}_{\tau=1}^t\right) & \\
&= \sum_i Pr\left(h^{t+1} = j \,\middle|\, h^t = i, \{I^\tau\}_{\tau=1}^t\right) \cdot Pr(h^t = i \,|\, \{I^\tau\}_{\tau=1}^t) \underset{(a)}{=} \\
& \sum_i Pr\left(h^{t+1} = j \,|\, h^t = i\right) \cdot Pr(h^t = i \,|\, \{I^\tau\}_{\tau=1}^t) \triangleq \sum_i T_{ij} b_i^t,
\end{aligned}
\tag{9}
$$

where (a) uses the assumption that h is a Markov chain. In a similar manner, given the belief state b^t, the next set of sensor outputs, I^{t+1}, is independent of the past sensor outputs:

$$
\begin{aligned}
Pr\left(I^{t+1} \,\middle|\, \{I^\tau\}_{\tau=1}^t\right) &= \sum_j Pr\left(I^{t+1} \,\middle|\, I_{1:t}, h^{t+1} = j\right) \cdot Pr\left(h^{t+1} = j \,\middle|\, \{I^\tau\}_{\tau=1}^t\right) = \\
& \sum_j Pr\left(I^{t+1} \,\middle|\, h^{t+1} = j\right) \cdot Pr\left(h^{t+1} = j \,\middle|\, \{I^\tau\}_{\tau=1}^t\right) = \sum_j f\left(I^{t+1}, j\right) \cdot \\
& Pr\left(h^{t+1} = j \,\middle|\, \{I^\tau\}_{\tau=1}^t\right) = \sum_j \sum_i f\left(I^{t+1}, j\right) T_{ij} b_i^t
\end{aligned}
\tag{10}
$$

This shows that the belief state summarizes all relevant information given past sensor outputs. Furthermore, the belief state can be recursively updated by:

$$
\begin{aligned}
b_j^{t+1} &= Pr\left(h^{t+1} = j \,\middle|\, \{I^\tau\}_{\tau=1}^{t+1}\right) = \frac{Pr\left(h^{t+1}=j, \, I^{t+1} \,\middle|\, \{I^\tau\}_{\tau=1}^t\right)}{Pr\left(I^{t+1} \,\middle|\, \{I^\tau\}_{\tau=1}^t\right)} \\
&= \frac{Pr\left(h^{t+1}=j \,\middle|\, \{I^\tau\}_{\tau=1}^t\right) \cdot Pr\left(I^{t+1} \,\middle|\, h^{t+1}=j, \{I^\tau\}_{\tau=1}^t\right)}{Pr\left(I^{t+1} \,\middle|\, \{I^\tau\}_{\tau=1}^t\right)} \underset{(a)}{=} \\
& \frac{\sum_i T_{ij} b_i^t f\left(I^{t+1}, j\right)}{Pr\left(I^{t+1} \,\middle|\, \{I^\tau\}_{\tau=1}^t\right)} \underset{(b)}{=} \frac{\sum_i T_{ij} b_i^t f\left(I^{t+1}, j\right)}{\sum_k \sum_m f\left(I^{t+1}, k\right) T_{mk} b_m^t}
\end{aligned}
\tag{11}
$$

where (a) follows from Equation (9) and, since $b_j^t = Pr\left(h^t = j \,\middle|\, \{I^\tau\}_{\tau=1}^t\right)$ by definition, (b) follows from Equation (10). □

Lemma 1 can be straightforwardly applied to simplify the calculation of the misclassification costs:

Corollary 1. *The misclassification costs can be written as*

$$
\rho_j\left(b^{t+1}\right) = \rho_j(b^t, I) = C_{FP} \sum_{i=1}^{j-1} b_i^{t+1}(b^t, I) + C_{FN} \sum_{i=j+1}^{J} b_i^{t+1}(b^t, I)
\tag{12}
$$

so the controller can calculate the total estimated misclassification cost as follows:

$$
\rho(b^t, I) = \sum_{j=1}^{J} b_j^{t+1}(b^t, I) \cdot \rho_j(b^t, I).
\tag{13}
$$

4. Sensor Activation Control

The Markovity of the belief states allows us to pose our problem as a MDP with the belief state as the system state. As opposed to tracking all past measurements, the belief state maintains its dimension over time. This enables us to formalize the dynamic programming equations, resulting in a tractable approximation of the optimal policy. In this section, we discuss two control policies that are based on the belief state MDP. The first is the optimal policy (Section 4.1), and the second is a one-step look-ahead greedy policy (Section 4.2). We prove that with accurate sensors that are all activated or deactivated together, the optimal policy can be computed accurately and efficiently (Section 4.3).

4.1. Optimal Policy

The Markov property of the belief states proven in Lemma 1 shows that our problem can be formulated as a MDP over a finite time horizon of L epochs. Define the value function $V^o(b^t)$, which is the minimum total expected cost of the path that starts at b^t,

also known as the "cost to go" [21]. Then, the optimal solution is given by the following dynamic programming equation on $V^o(\boldsymbol{b}^t)$:

$$
\begin{aligned}
V^o(\boldsymbol{b}^t) &= \min_{\boldsymbol{s}^t}\{(1-\omega)C(\boldsymbol{s}^t) + \int_b \Big(Pr\Big(\boldsymbol{b}^{t+1} = b \,\Big|\, \boldsymbol{b}^t\Big) V^o\Big(\boldsymbol{b}^{t+1}\Big) + \omega\rho_j(b)\Big)db\} \\
&\underset{(a)}{=} \min_{\boldsymbol{s}^t}\left\{ (1-\omega)C(\boldsymbol{s}^t) + \sum_{l\in L(\boldsymbol{s}^t)} \Big(Pr\Big(\boldsymbol{l}^{t+1} = \boldsymbol{l} \,\Big|\, \boldsymbol{b}^t\Big)\Big(V^o\Big(\boldsymbol{b}^{t+1}(\boldsymbol{b}^t, \boldsymbol{l})\Big) + \omega\rho_j(\boldsymbol{b}^t)\Big)\Big)\right\}
\end{aligned}
\tag{14}
$$

where (*a*) follows from Equation (10), which also shows that

$$
Pr\Big(\boldsymbol{b}^{t+1}(\boldsymbol{b}^t, \boldsymbol{l}) = b \,\Big|\, \boldsymbol{b}^t\Big) = Pr\Big(\boldsymbol{l}^{t+1} = \boldsymbol{l} \,\Big|\, \boldsymbol{b}^t\Big).
\tag{15}
$$

Here, ω is the weight the system designer assigns on misclassification costs vs. power costs. Lower values of ω result in emphasizing power cost reduction, while higher values result in minimizing misclassification costs. The default value of ω in our experiments is 0.5.

From Equation (14), we can extract an optimal policy by selecting the $\boldsymbol{s}^t$ that minimizes the value function for the belief state, i.e., $\operatorname{argmin}_{\boldsymbol{s}^t} V^o(\boldsymbol{b}^t)$. Equation (14) can be solved using the value iteration method [21].

Our problem can be formulated with an infinite horizon by adding a discount factor to Equation (14). Alternatively, we can assume a terminal health state, which acts as an absorbing state in the Markov chain. In practice, this state occurs when the WBAN health sensing is no longer relevant: for example, when the patient arrives at an emergency room and requires medical intervention, hospitalization, mortality, etc.

4.2. One-Step Look-Ahead Greedy Policy

The greedy algorithm is a one-step look-ahead policy that at each epoch finds which sensors to activate in order to minimize the immediate cost incurred. The value function of this greedy policy is:

$$
V^g(\boldsymbol{b}^t) = \min_{\boldsymbol{s}^t}\left\{ (1-\omega)\cdot C(\boldsymbol{s}^t) + \omega\cdot\sum_{l\in L(\boldsymbol{s}^t)} Pr\Big(\boldsymbol{l}^{t+1} = \boldsymbol{l}\big|\boldsymbol{b}^t\Big)\cdot\rho(\boldsymbol{b}^t, \boldsymbol{l}) \right\}.
\tag{16}
$$

4.3. Accurate Sensor Use Case

In this subsection, we consider the case of accurate binary sensors, which can either all be activated or deactivated at a certain epoch. Although in general, sensors are not accurate, it is reasonable to assume that sensors manufactured for tracking health states will be fairly accurate. The action space is $\mathcal{S} = \{0, 1\}$, such that all sensors are activated if $a = 1$ and idle if $a = 0$. The sensor output probabilities are $A_{nj} \in \{0, 1\}$ for all n, j. We also note the since we assume binary sensors, a single sensor output probability matrix may be used. In addition, we assume that the sensor output probabilities matrix is invertible, which can be achieved by design. As a result, each possible sensor output vector maps to a single health state. We note that when a single mapping exists between each possible sensor output vector and each health state, the problem can be solved regardless of A's structure. The following lemma shows that in this case, the number of effective belief states is finite and linear in both the time horizon L and the number of health states J. Hence, the dynamic programming problem can be accurately solved.

Lemma 2. *Let $\mathcal{S} = \{0, 1\}$, and let the sensors be fully accurate, where $A_{nj} \in \{0, 1\}$ for all n, j. Assume that A is invertible. Then, starting from the belief state $\boldsymbol{b}^0$, only $L\cdot(J+1)$ belief states are reachable.*

Proof of Lemma 2. Let h^t be the health state at time t and $\boldsymbol{h}^t$ be the standard vector that has one at the h^t-th component. If the matrix A is invertible, then we can reconstruct the

health state as $h^t = A^{-1}l$. Hence, $f(l,k) = \mathbf{1}_{\{k=A^{-1}l\}}$. For $j \neq h^{t+1}$, we have $b_j^{t+1} = 0$, and for $j = h^{t+1}$

$$b_j^{t+1} = \frac{\sum_i T_{ij} b_i^t f\left(l^{t+1}, j\right)}{\sum_k \sum_m f\left(l^{t+1}, k\right) T_{mk} b_m^t} = \frac{\sum_i T_{ij} b_i^t \mathbf{1}_{\{j=A^{-1}l\}}}{\sum_k \sum_m \mathbf{1}_{\{k=A^{-1}l\}} T_{mk} b_m^t} = \frac{\sum_i T_{ij} b_i^t}{\sum_m T_{mk} b_m^t} = 1. \quad (17)$$

If the sensors are not activated (i.e., $a = 0$), $f(l,k) = 1$ for all l; hence,

$$b_j^{t+1} = \frac{\sum_i T_{ij} b_i^t}{\sum_k \sum_m T_{mk} b_m^t} \underset{(a)}{=} \sum_i T_{ij} b_i^t \Rightarrow b^{t+1} = T \cdot b^t \quad (18)$$

where (a) follows since the denominator sums over the entire probability space. To summarize:

$$b^{t+1} = \begin{cases} T \cdot b^t & a = 0 \\ h^{t+1} & a = 1 \end{cases} \quad (19)$$

which means that if all sensors are activated, we can perfectly estimate the health state h^t. Hence, only belief states $\mathcal{B}_0 = \left\{ b \middle| b = T^m \cdot b^0 \text{ or } b = T^m \cdot h \text{ for } m \in \mathbb{N}, \, h \in \mathcal{J} \right\}$ are reachable from b^0, where $\mathcal{J}$ is the set of J pure health states. The set $\mathcal{B}_0$ can be computed in advance before solving the dynamic programming equation. $\square$

The implication of the above result is that the dynamic programming problem can be accurately solved with complexity $O(L^2 J)$. To implement this computation, we also exploit the fact that each belief state b^t can transition into two possible belief states: h^{t+1} and $T \cdot b^t$, where T is the transition matrix.

In the general case of inaccurate sensors, the update function of b^{t+1} is not quadratic or even convex. Considering the lack of a convenient structure for the problem, one needs to compute $V\left(b^{t+1}\right)$ for every belief b^{t+1} in the J-dimensional simplex. In practice, this means that only an approximate solution can be obtained by discretizing the belief space. Therefore, the proof of Lemma 2 implies that the general scenario with inaccurate sensors or a complicated action space $\mathcal{S}$ can only be solved numerically.

4.4. Belief State Discretization

The dynamic programming formulation in Equation (14) explores all the possible future states over a finite period. In practice, a continuous belief state results in a very large state space, which results in exponentially growing complexity in J. In this Subsection, we describe the simple discretization procedure for the belief state space we will use to approximate the value function V^o. The procedure first defines a finite set of valid belief state vectors. The vectors can be related to different medical treatments or protocols, as mapped by medical professionals. Then, given an actual belief state vector, the proposed procedure finds the vector from the finite set that is closest to the actual one. Finally, the procedure calculates the value function for each valid belief state vector. The steps of the procedure are the following:

1. Set a level of discretization and define the set B, which contains all valid belief state vectors.
2. Given an actual vector, b, calculate its distance $\sum_j \left| \widetilde{b}_j - b_j \right|$ to each valid vector, $\widetilde{b} \in B$
3. Return the vector, $\widetilde{b}^*$ which is the closest to b.

For example, if the level of discretization is selected to be 0.2, and $b = [0.17, 0.35, 0.40, 0.08]$, then the above procedure returns $\widetilde{b}^* = [0.20, 0.40, 0.40, 0.00]$ as the closest valid vector. After discretizing the state space, we use (14) and calculate $V^o(\widetilde{b})$ for each valid state $\widetilde{b} \in B$.

The fine discretization required to guarantee a good approximation of the optimal policy leads to high dimensional belief space B (as we demonstrate numerically in Section 6.2).

As a result, implementing the optimal policy is complicated, since the computational load grows exponentially with the dimension of *B*. Even solving the dynamic programming offline does not avoid the high complexity, since it requires instead storing large policy tables that map the discretized states to actions.

To avoid discretization, function approximation can be employed, which enables solving the POMDP over a continuous state space [22], although still with high complexity. Since our focus is to study the performance of the greedy policy, we use the simple discretization method described above to provide a "very close to optimal" benchmark.

5. Empirical Analysis on Glucose Data

5.1. Use-Case Description

We now demonstrate the implementation of the WBAN controlling solution in a real-world use case. We use an open-source dataset of continuous glucose measurements collected from a group of diabetes type-1 patients [8]. The measurements were recorded using a wireless glucose level sensor placed on the patients for extended periods. Using our proposed greedy control policy, we aim to reduce energy consumption while maintaining high monitoring accuracies, thus extending the sensors' lifespans and allowing longer sensing periods. The model parameters (sensor accuracies, transition probabilities, health state trajectories) were extracted from the data, allowing us to evaluate the effectiveness of the greedy policy using real-life glucose measurements that represent a potential application.

5.2. The Data

The data used for this analysis were collected as part of a clinical trial aimed at assessing the efficacy of real-time glucose monitoring [8]. The raw data contain glucose measurements for a group of 232 heterogenic patients from all ages with type 1 diabetes and glycated hemoglobin (HbA1c) 7.0–10.0%. For most patients, the measurements were collected over 6 months using three different wireless real-time continuous glucose monitoring (RT-CGM) systems.

Figure 3 displays the glucose level measurements over time for three randomly selected patients over a period of 12 h. For the majority of patients, the time gap between subsequent measurements is 5 min. However, about 16.9% of the records have abnormal time gaps, which could be caused due to malfunctions in the measurements collection, a patient who has removed the sensor, etc. In order to overcome the difficulties when dealing with unequal gaps, we have checked what is the most common time gap per patient and set it as the patient's time frame for measurements collection.

5.3. Data Modeling

5.3.1. Health States

In order to apply the POMDP-based model, we first define the set of health states each patient can occupy. To this end, we discretize the range of glucose measurements by dividing the entire range of glucose values into three smaller ranges, each defined as a health state. The defined ranges are based on medical standards [23]. We note that the defined range structure can be designed according to varying medical needs. In this experiment, the belief states were chosen as $h_3 \leq 100$, $100 < h_2 \leq 180$ and $180 < h_1$.
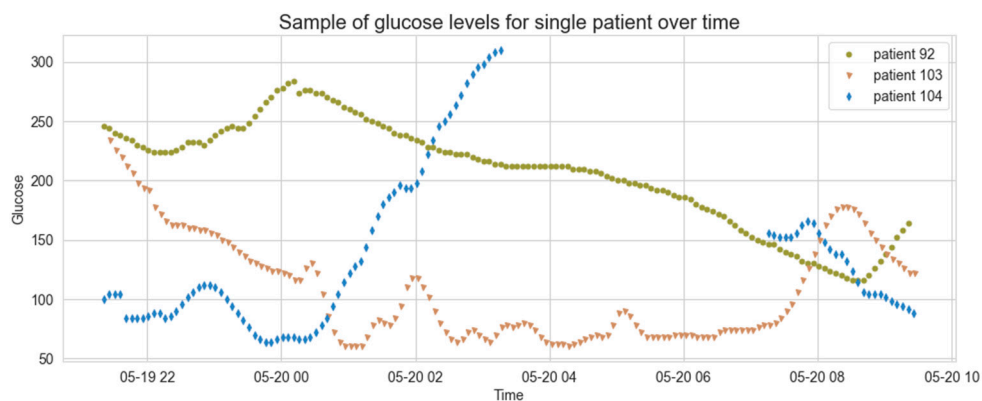
Figure 3. Glucose levels of randomly selected patients over time.

5.3.2. Virtual Sensors and Actions

We think of the glucose data as measured using a single physical sensor, which provided a measurement every 5 min, which we call a time slot. We now describe how our model in Section 2 can be applied to control the measurement frequency of this single sensor. For this purpose, we define a time frame as k consecutive time slots. Then, one can think of each time slot as a virtual sensor and apply our model to these k virtual sensors directly. For example, if only sensor 1 is activated, then the measurement frequency would be $1/k$, and if all sensors are activated, then the measurement frequency would be 1. In general, the measurement frequency does not have to be constant, since we can, for example, activate sensor 1 and sensor 2 and then not activate sensor 3 to sensor k. This way, a time frame is equivalent to the decision epoch as defined in Section 2. Hence, every time frame, the controller can change the subset set of time slots during which the single physical sensor is activated. Intuitively, the higher the sensing frequency, the more energy is consumed and the better the monitoring accuracy. The set of actions is defined by all the possible subsets of the k time slots. However, constraining the set of actions will reduce the complexity of the controller. In our experiment, we chose $k = 6$ so that the decision time frame consists of six time slots. Then, we define the set of actions as $S = \{(4), (2, 6), (1, 3, 5)\}$, where each tuple contains the index of the time slots during which the sensor will carry out a measurement.

Then, the glucose measurements collected throughout each time frame, according to the selected action, are aggregated to a single observation using an aggregation function. The aggregated values are then discretized according to the states defined in the previous sections to provide sensor outputs $l \in \{1, 2, 3\}$. For example, the aggregation function may be defined as an average or maximum over the collected records. This serves as the output obtained by the k virtual sensors. In our experiment, the aggregation function is defined as the average of the collected records.

In principle, more advanced aggregation functions could be designed to average all observations throughout the decision epoch, unless there exists an observation that is relatively far from the other observations throughout the decision epoch. This would allow handling extreme observations differently than decision epochs where the observation values are relatively close to each other, promising that extreme glucose measurements are not missed due to the averaging of observations in the decision epoch.

The system designer can select k according to varying system requirements. Smaller values of k may provide better monitoring performance, since then, controller decisions will be made more frequently. On the other hand, small values would require higher computational complexity.

5.3.3. Transition Matrix

Using a portion of the patient's measurements, the transition matrix can be estimated based on actual health state transitions. The conditional probability T_{ij}, is calculated as the

ratio between the number of transitions between h_i and h_j and the number of transitions from h_i to any of the other health states, over the selected portion. The calculated transition matrix may be used for future monitoring.

5.3.4. Sensor Accuracies

The sensor output probabilities matrix is also calculated using a portion of the patient's data. For each sensor n, we calculate the probabilities to obtain a certain output, given that the patient is in a certain health state. Since each sensor may provide an output $l \in \{1, 2, 3\}$, the sensor output probabilities matrix per sensor n will be of the form $A_n^{3 \times 3}$. For example, if the patient is currently occupying h_1, the probability to obtain an output value $l = 2$ from sensor $n = 3$ is $(A_3)_{1,2}$.

In order to estimate the probability to receive a specific output l if the patient is in health state j, we calculate the ratio between the number of time slots where the patient was in health state j and the sensor returned an output l, divided by the number of time slots the patient was in health state j.

5.4. Model Parameter Extraction

Initially, we have divided the data for each patient to train and test sets to 75% and 25% of the records, respectively. Using the training set, we first extract the transition matrix and sensor accuracies per patient. Then, we use all parameters needed for the POMDP-based model (set of health states H, transition matrix T, sensor accuracies matrix A, set of possible actions S, cost parameters C_{FN}, C_{FP}) to apply a policy over the test set data. Finally, we compare the overall performance of the policy to a baseline. The baseline is defined as an inclusive policy—the sensor is activated at all time slots. This is an accuracy-optimal policy. We also compare with the dynamic programming solution, where the belief state was discretized to 9 levels to approximate the optimal policy using the value iteration algorithm [21].

Then, the collected measurements are aggregated using the aggregation function, thus creating a baseline health state prediction, for each of k time slots. To compare between policies, we will examine the total cost, the power cost, and the health state classification accuracy (calculated using the most likely health state, according to the belief state vector).

We set the following parameters: $k = 6$, $C_{FN} = 100$, $C_{FP} = 50$ and $S = \{(4), (2, 6), (1, 3, 5)\}$, where the last one is the set of possible actions (the time slot indices in which the sensor is activated).

The activation costs are defined to be the number of time slots where the original sensor is activated during the decision time frame of k time slots. For example, if the sensor is activated for three time slots out of a six-time slot decision time frame, the activation cost will be 3. We also note that the misclassification costs were defined to be constant for false positive and false negative scenarios, for simplicity. A more complex cost structure is applicable, such as defining increasing costs by the severity of misclassification. The use of such a cost structure will be demonstrated in Section 6.

5.5. Policy Comparison

Table 1 compares the overall costs of the greedy policy and the inclusive solution, averaged over all patients:

Table 1. Activation costs and accuracy results.

	Activation Costs	Accuracy
Inclusive	100%	85.1%
Greedy	48.33%	83.5%
Difference	**−51.67%**	**−1.6%**

We observe that the activation costs were reduced by 51.67%, whereas the accuracy was reduced by a small factor of 1.6%, resulting in an accuracy score of 83.5%.

Figure 4 compares the performance of value iteration (dynamic programming), greedy, and inclusive. Clearly, the inclusive policy inflicts the highest total costs. However, the misclassification costs induced by the inclusive policy are lower, compared to the greedy and value iteration solutions. In addition, the value iteration solution obtained results similar to the greedy solution. For the scenario presented here and the defined system parameters, the greedy solution provides good performance.
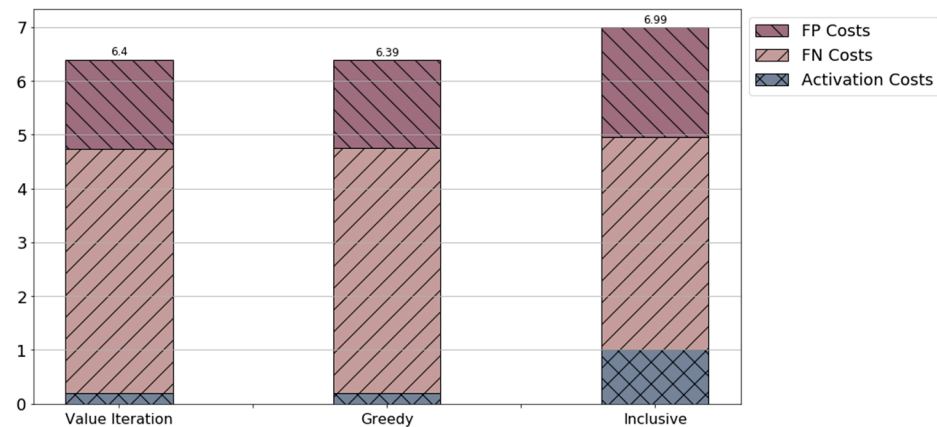

Figure 4. Activation and misclassification cost breakdown obtained by the different proposed policies. The costs are average over all patients. "FP" stands for false positive and "FN" stands for false negative.

Table 2 compares the false positive and false negative probabilities, averaged over all patients and time steps. The greedy solution slightly outperforms the value iteration method for both these measures. We note that this occurs mainly due to the fact that the framework aims at optimizing the total costs obtained in order to reduce activation costs, while maintaining accuracy (as presented in Table 1).

Table 2. Activation costs and accuracy results.

| | Misclassification Probabilities | | |
	Value Iteration	Greedy	Inclusive
False Positive	2.51%	2.47%	0.24%
False Negative	4.19%	4.11%	0.41%

5.6. Sensitivity Analysis

Our results from Section 5.5 suggest that the greedy policy is nearly optimal for the scenario we extracted from the real glucose data. To establish this statement, we now conduct a sensitivity analysis of two main system parameters. We start by varying the parameter ω between 0 and 1, thus changing the weight of the two cost components in Equation (16). Lower values of ω result in emphasizing power cost reduction while higher values result in minimizing misclassification costs. Figure 5 below demonstrates the cost ratio between the greedy and value iteration methods. Each point considers the total costs obtained over all patients, and the error bars show the standard deviation of this ratio over the patients.

One can observe that the performance (total cost) of the greedy policy is never far behind the optimal policy. For the extreme case of $\omega = 1$, in which the controller will aim at minimizing misclassification costs only, the greedy method obtains costs that are approximately 13.5% higher than those of the value iteration method. However, for medical applications, the regime of interest is when the misclassification cost is more important than the energy cost since very small misclassification probabilities are required. When $\omega \leq 0.5$, the cost of the greedy policy is never more than 2% more than that of the optimal

policy. Hence, the greedy policy is nearly optimal for the practical regime of interest in WBAN health monitoring systems in this case.
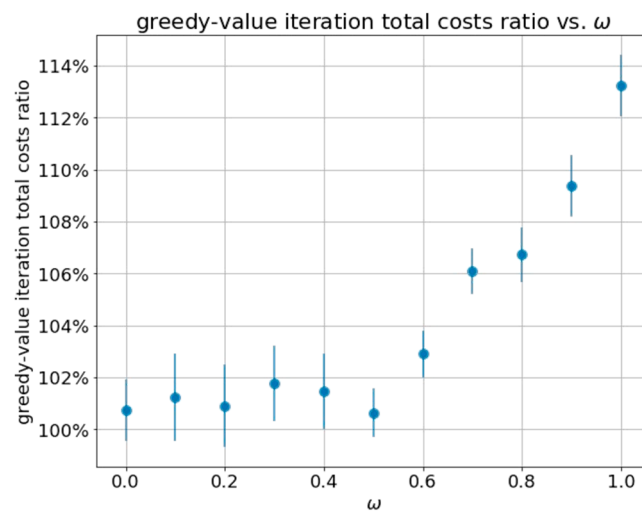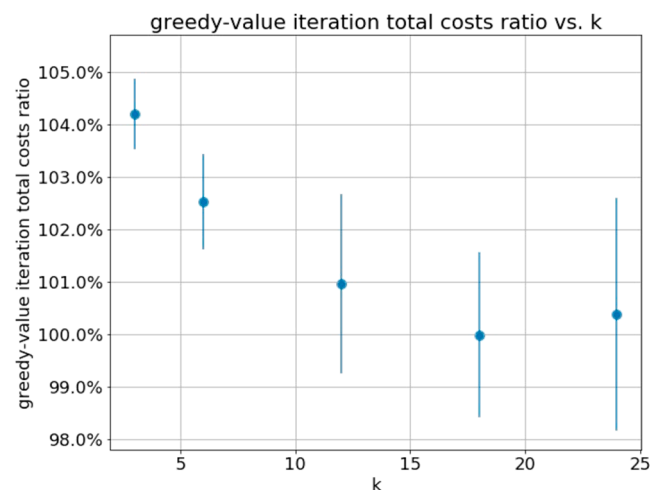

Figure 5. Ratio between total costs obtained by greedy and value iteration methods obtained over all patients for varying ω values, for the following set parameters: $k = 6$, $C_{FN} = 100$, $C_{FP} = 50$.

We next quantify the trade-off between the misclassification cost and the activation cost in the glucose use case, following the experiment above. We note that although the policy is obtained given the weighting parameter, the calculated costs reflect the true activation and misclassification costs obtained when applying the solution to the test set (i.e., without the coefficients ω and $1 - \omega$).

The obtained data points have a high fit of $R^2 = 0.997$ to $f(x) = \frac{145.09}{276.96x + 27.65} - 0.07$. This high fit in Figure 6 suggests that $f(x)$ can be used by the system designer to adjust ω for the glucose monitoring case.
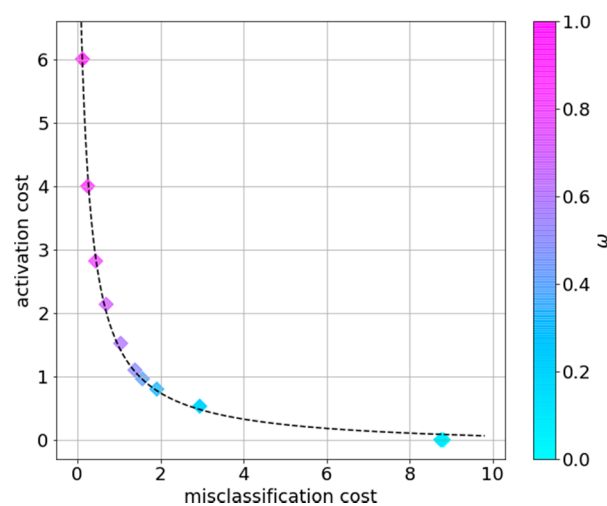
Figure 6. Activation costs vs. misclassification costs per ω, averaged over all patients.

Another main system parameter is the length of the decision epoch k. Higher values of k require lower computational complexity but will introduce more noise into the system, which is caused by the aggregation of more raw observations. Figure 7 shows that the greedy policy is nearly optimal for a wide ranges of values of k.

Figure 7. Ratio between total costs obtained by greedy and value iteration methods obtained over all patients for *k values*, for the following set parameters: $\omega = 0.5$, $C_{FN} = 100$, $C_{FP} = 50$.

6. Empirical Analysis and Sensitivity Analysis on Synthetic Data

In this section, we provide simple numerical examples that will allow us to study the behavior of the greedy policy compared to the optimal policy in more detail, and to provide more intuition and interpretability for the proposed greedy policy. For the glucose data, the transition matrix and sensor accuracies were estimated for the data so they do not represent a degree of freedom. Here, all parameters are arbitrary, allowing for a more general sensitivity analysis.

6.1. WBAN Dynamics for the Greedy Policy

We define the model parameters as follows: $J = 4$ (the number of health states), $N = 5$ (the number of sensors). Additional parameters are as follows:

- $T = \begin{bmatrix} 0.90 & 0.04 & 0.04 & 0.02 \\ 0.02 & 0.90 & 0.04 & 0.04 \\ 0.01 & 0.01 & 0.90 & 0.08 \\ 0.01 & 0.01 & 0.03 & 0.95 \end{bmatrix}$ is the transition matrix, from state i (row) to state j

(column),

- $A = \begin{bmatrix} 0.99 & 0.01 & 0.50 & 0.50 \\ 0.50 & 0.90 & 0.10 & 0.50 \\ 0.50 & 0.50 & 0.90 & 0.10 \\ 0.90 & 0.10 & 0.50 & 0.50 \\ 0.60 & 0.40 & 0.60 & 0.50 \end{bmatrix}$ is the sensor output probabilities matrix,

- $C_s = \begin{bmatrix} 5 & 1 & 1 & 1 & 0.2 \end{bmatrix}$ denotes the sensors' activation costs,

- $C_m = \begin{bmatrix} 0 & 100 & 125 & 150 \\ 25 & 0 & 75 & 100 \\ 50 & 25 & 0 & 50 \\ 75 & 60 & 45 & 0 \end{bmatrix}$ denotes the misclassification costs where element

(i, j)

- denotes the cost of misclassifying health state i as state j. Accordingly, costs above and below the diagonal represent false negative and false positive costs, respectively.

The transition matrix T was chosen based on the reasonable assumption that the probability of remaining in a certain health state is larger than the probability of transitioning between health states. In practice, the transition probabilities can be estimated using historical clinical data as we did for the glucose data in Section 5. The cost parameters were selected to reflect a clear trade-off between the misclassification costs and the sensors' activation costs. In practice, the sensor activation costs can be based on the cost of recharg-

ing the sensors, and the misclassification costs can be estimated based on the cost of the medical care needed due to health state misclassification.

We simulate the greedy algorithm described in Section 4.2 to derive a greedy policy. The most likely state shown in Figure 8 is generated by $h^{ML} = argmax_j\left(b_j^t\right)$. One can observe that the greedy policy generally provides accurate predictions of the actual health states in the proposed example.
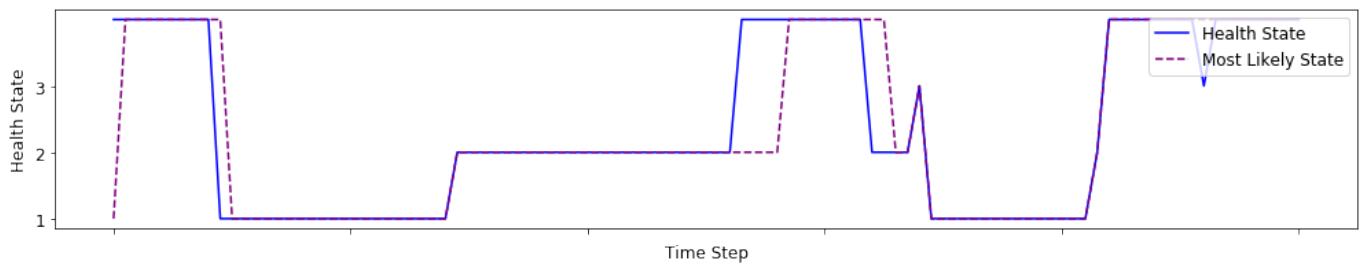

Figure 8. The unknown health state and the most likely health state (using the greedy policy) over 20 epochs.

Figure 9 shows the number of sensors activated throughout the simulation. One can observe that the largest number of sensors is usually activated during health state transitions, i.e., in periods where the patient's health is relatively unstable. In addition, Table 3 summarizes the dynamics of the activated sensors regarding their different accuracies and costs. For example, even though sensor 5 incurs low costs, it is almost never activated due to the low accuracy it provides. Sensors 1 and 4 are informative regarding h_1 and h_2. Although sensor 1 is very accurate compared to sensor 4, it is activated during a smaller number of epochs, which follows due to the lower costs incurred by sensor 4.
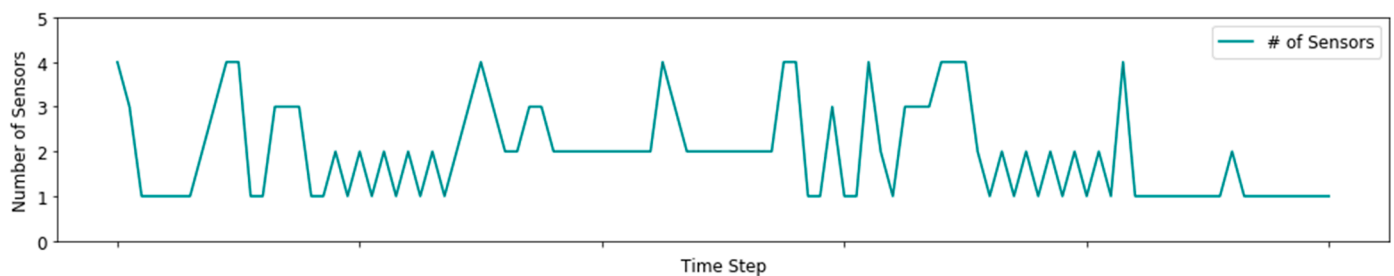
Figure 9. The number of sensors activated throughout the simulation using the greedy policy.

Table 3. Percent of epochs during which each sensor was activated throughout the simulation.

Sensor Usage				
1	**2**	**3**	**4**	**5**
42%	49%	51%	55%	0%

6.2. Comparison of the Greedy Policy to the Optimal Policy

We now compare the proposed greedy policy with the optimal dynamic programming approach, which were presented in Sections 4.1 and 4.2, respectively.

For this section, we define a more compact problem, which will allow deeper analysis. The parameters are defined as follows: $J = 3$, $N = 2$. Additional parameters are as follows:

- $T = \begin{bmatrix} 0.90 & 0.06 & 0.04 \\ 0.04 & 0.90 & 0.06 \\ 0.01 & 0.04 & 0.95 \end{bmatrix}$ and $A = \begin{bmatrix} 0.90 & 0.10 & 0.50 \\ 0.50 & 0.90 & 0.10 \end{bmatrix}$

- $C_s = \begin{bmatrix} 2 & 2 \end{bmatrix}$ and $C_m = \begin{bmatrix} 0 & 33 & 42 \\ 8 & 0 & 25 \\ 17 & 8 & 0 \end{bmatrix}$.

The selected policy gives the number of sensors activated given a certain belief state vector.

Figure 10 maps the belief states into the subset of activated sensors. We observe that the controller tends to activate both sensors in the mid-ranges of the belief states, i.e., in areas where the information concerning the patient's active health state is not definitive. As the belief states move toward the edges of the graph, fewer sensors are needed because there is higher certainty concerning the patient's active health state. In the far corners of the graph, no sensors are activated. We note that the distinction between the use of the first and second sensors, when activating only one sensor, is related to each sensor's ability to differentiate between a different pair of health states.
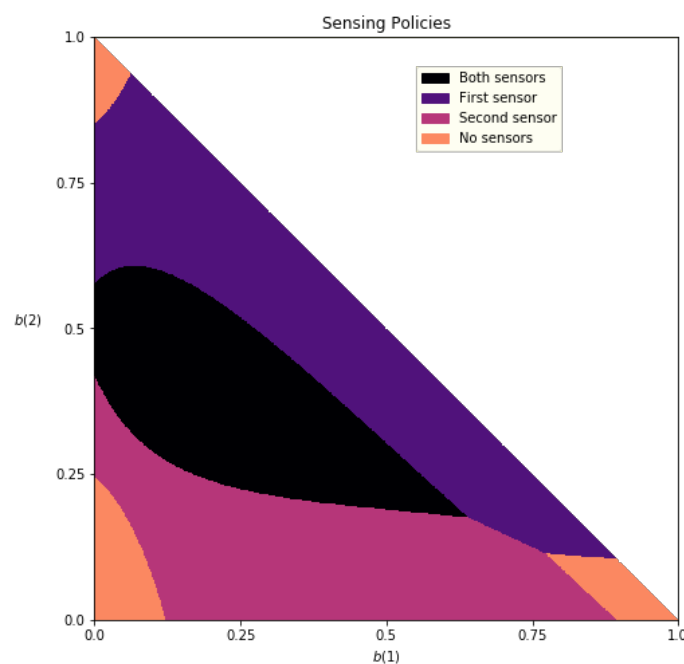
Figure 10. Greedy sensing policy over the belief state space b. Since $b_3 = 1 - (b_1 + b_2)$, the belief space forms a simplex and may be described by only the belief over the first two states.

We now examine the different behaviors displayed by the proposed greedy policy and the dynamic programming policy. For this purpose, we use the average total costs obtained by both solutions over 30 random transition matrices as the main performance indicator. In addition, for simplicity of analysis from this point forward, we apply a simpler cost structure where the FP and FN costs are set to constant values—50 and 100, respectively. We note that all transition matrices share the property that the probabilities of remaining in certain health states are generally higher than those of transitioning to other health states. As discussed in Section 4.4, the optimal dynamic programming solution requires belief state discretization. Higher discretization resolutions will allow more accurate health state monitoring but will incur an exponentially higher computational load.

Figure 11 demonstrates the effect of varying the discretization resolution on the performance of the dynamic programming solution. The *x*-axis shows the "binning" resolution defined for the discretized belief state space. Higher discretization levels lead to higher accuracy when approximating actual belief states. We can see that for lower values of discretization levels, the greedy policy outperforms the dynamic programming formulation, while for higher levels, the value iteration solution outperforms the greedy solution and converges to an average total cost that is approximately 3% lower than that of the greedy solution. *Overall, the greedy solution shows a maximum gap of less than 4% from the dynamic programming solution for high discretization levels.*
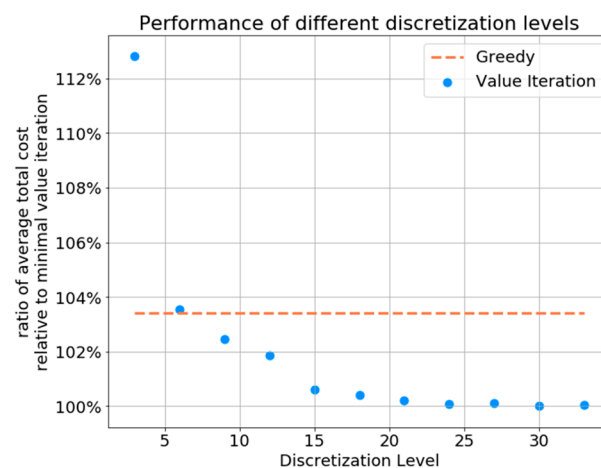
Figure 11. Performance of the value iteration and greedy policies as a function of the discretization level, averaged over 100 i.i.d. Monte Carlo simulations.

We observe also that there is a downward trend up to a discretization level of 15, where the trend plateaus. From this graph, we can see that a discretization level of 15 (meaning each element in the belief state vector could be approximated by a factor of $\frac{1}{15}$) provides sufficient performance while maintaining a lower computational load compared to larger discretization levels. Hence, in the rest of the simulations, we will use 15 as the discretization level.

6.3. Sensor Accuracy Sensitivity Analysis

We now analyze the effect of the sensor accuracies on the policies' performances. Generally, using sensors with poor accuracies introduces more noise into the systems that hinders their performance. We define a group of output probability matrices using a parameter ϵ as follows:

$$A = \begin{bmatrix} 1 - \epsilon & \epsilon & \epsilon \\ \epsilon & 1 - \epsilon & \epsilon \end{bmatrix}, 0 \leq \epsilon \leq 0.5$$

Hence, each of the two sensors differentiates between a pair of health states. Larger values of ϵ result in weaker differentiation between the pairs of health states.

As expected, higher values of ϵ result in higher costs for all policies, whereas near-zero values provide low costs down to the point where the sensors are perfect, which results in identical low costs for all the policies (see Figure 12). For ϵ values higher than 0.3, the sensors yield very low-accuracy outputs, causing the controller to deactivate the sensors throughout the sensing period. For the majority of examined ϵ values, the proposed greedy policy was nearly optimal. Only for very inaccurate sensors with $\epsilon \geq 0.25$ is the total cost of the greedy policy more than 5% higher than that of the (approximately) optimal value iteration. This is encouraging, since medical sensors are typically required to be highly accurate in practice due to the possible implications.
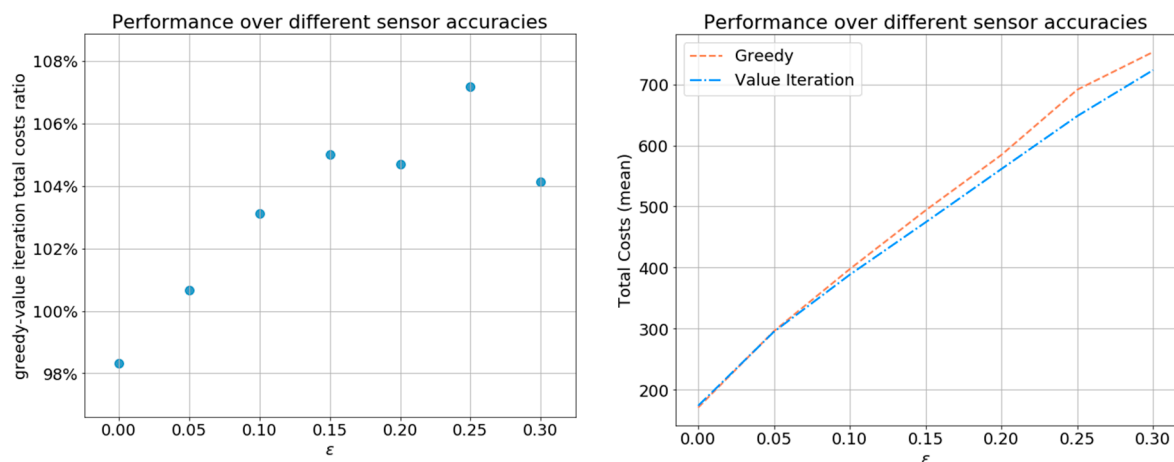
Figure 12. Total costs of the greedy policy and the value iteration policy over different sensor output probability matrices A defined by $0 \leq \epsilon \leq 0.3$ (step size 0.05), averaged over 100 i.i.d. Monte Carlo simulations.

6.4. Transition Matrix Sensitivity Analysis

Finally, we demonstrate the effect of stability in the patient's health state trajectories on the policies' performances. Less stable health state trajectories, in which the patient tends to change health states often, are modeled by a more equally distributed transition matrix. For that purpose, we define a group of transition matrices using a parameter $\widetilde{T}$:

$$
T = \begin{bmatrix} \widetilde{T} & \frac{1-\widetilde{T}}{2} & \frac{1-\widetilde{T}}{2} \\ \frac{1-\widetilde{T}}{2} & \widetilde{T} & \frac{1-\widetilde{T}}{2} \\ \frac{1-\widetilde{T}}{2} & \frac{1-\widetilde{T}}{2} & \widetilde{T} \end{bmatrix}, 0 \leq \widetilde{T} \leq 1.
$$

Note that $\widetilde{T}$ determines the stability of the health state trajectories. For lower values of $\widetilde{T}$, the obtained transition matrix is more uniform. For the value $\widetilde{T} = 0.3$, the policies yield the highest possible costs, since then, T provides no information concerning the patient's potential health state trajectory. For higher values of $\widetilde{T}$, the transition matrix T causes higher-stability health state trajectories. For the majority of $\widetilde{T}$ values examined, the greedy policy is nearly optimal, showing the largest gap when the total costs of both policies decline for higher values of $\widetilde{T}$. For the most extreme value of $\widetilde{T} = 1.0$ (not shown in the left graph in Figure 13), for which the heath state is static, the total cost of the greedy policy is 450% compared to the value iteration policy.
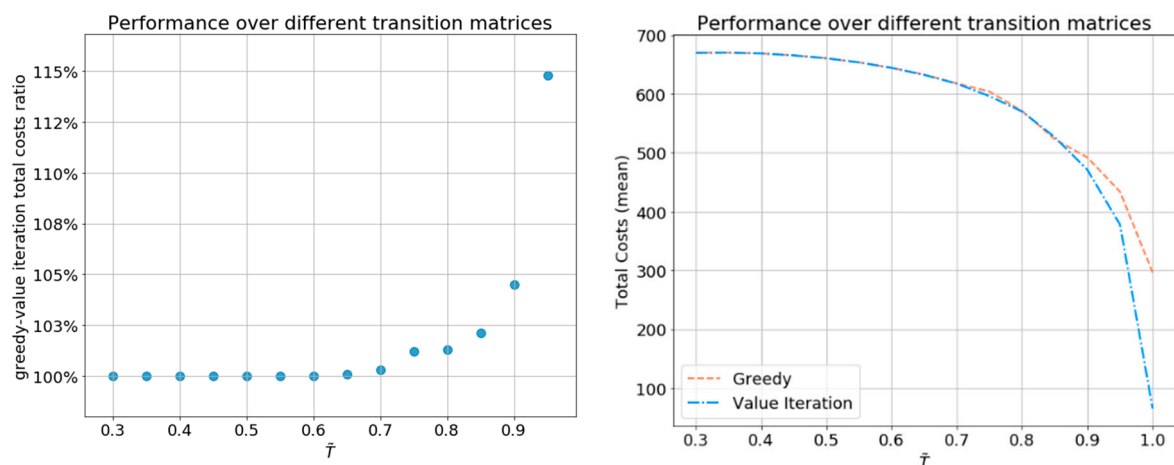

Figure 13. Total costs of the greedy policy and the value iteration policy over different transition matrices T, defined by $0.3 \leq \widetilde{T} \leq 1$ (step size 0.05), averaged over 100 i.i.d. Monte Carlo simulations.

7. Conclusions

We studied the energy consumption versus accuracy trade-off in WBAN health monitoring systems. In our setting, the controller decides which subset of sensors to activate at every epoch. The objective is to monitor the unknown patient's health state while minimizing the power and misclassification costs over time.

We studied the performance of the simple one-step look-ahead approach for sensor activation. This simple policy is significantly easier to implement and offers better interpretability than the optimal one based on dynamic programming. Furthermore, since the optimal policy works with a continuous belief state space, it must be discretized, which degrades the performance and requires storing large policy tables.

Our empirical study on the glucose levels of 232 patients over 6 months reveals that the proposed greedy policy saves 50% of the energy cost while losing only 1.6% of the misclassification cost compared to the policy that activates all sensors at all times (i.e., an accuracy-optimal policy). The sensitivity analysis confirms that the greedy policy is nearly optimal as long as the weight of the misclassification cost is at least twice that of the energy cost. Since in medical application, the misclassification probabilities must be very low, this weight regime includes the practical regime.

Our findings suggest that the greedy policy can be an efficient policy to use in practice. The sensitivity analysis suggests that the reason the greedy policy is so effective is that as we estimate from the data, the health state transitions are not extremely slow, and the sensor accuracies are not extremely bad. Since this is typically the case for various medical applications, studying the performance of the greedy policy on other medical datasets will further establish the greedy policy as a practical health monitoring control for WBAN systems.

As future research, additional cost structures can improve the applicability and real-world value of the solutions. For example, costs that dynamically change over time or different misclassification costs for more severe health states could be used. Additional extensions include generalizing for unknown POMDP parameters (transition matrix, patient health states, etc.). For example, assuming the transition matrix is unknown to the controller, sensor outputs from a group of patients may be used to initially estimate a generalized transition matrix.

References

1. David, Y.B.; Geller, T.; Ward, A.; Miller, D.; Khmelnitsky, E.; Bambos, N.; Ben-Gal, I. Optimal Health Monitoring via Wireless Body Area Networks. In Proceedings of the 57th IEEE Conference on Decision and Control (CDC), Miami Beach, FL, USA, 17–19 December 2018; pp. 6800–6805.
2. Miller, D.; Zhou, Z.; Bambos, N.; Ben-Gal, I. Optimal Sensing for Patient Health Monitoring. In Proceedings of the 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018; pp. 1–7.
3. Chen, M.; Gonzalez, S.; Vasilakos, A.; Cao, H.; Leung, V.C.M. Body Area Networks: A Survey. *Mob. Netw. Appl.* **2010**, *16*, 171–193. [CrossRef]
4. Mohammed, Y.O.; Baroudi, U.A. Partially observable Markov decision processes (POMDPS) and wireless body area networks (WBAN). *KSII Trans. Internet Inf. Syst.* **2013**, *7*, 1036–1057.
5. Kurian, A.; Divya, R. A survey on energy efficient routing protocols in wireless body area networks (WBAN). In Proceedings of the 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore, India, 17–18 March 2017.
6. Krishnamurthy, V. *Partially Observed Markov Decision Processes*; Cambridge University Press: Cambridge, UK, 2016.
7. Jovanov, E. A survey of power efficient technologies for Wireless Body Area Networks. In Proceedings of the 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Vancouver, BC, Canada, 20–25 August 2008; p. 3628.
8. JDRF CGM Study Group. Jdrf randomized clinical trial to assess the efficacy of realtime continuous glucose monitoring in the management of type 1 diabetes: Research design and methods. *Diabetes Technol. Ther.* **2008**, *10*, 310–321. [CrossRef] [PubMed]
9. Xing, J.; Zhu, Y. A Survey on Body Area Network. In Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing, Beijing, China, 24–26 September 2009; pp. 1–4.
10. Miao, L.; Mao, J.; Cassandras, C.G. Optimal Energy-Efficient Downlink Transmission Scheduling for Real-Time Wireless Networks. *IEEE Trans. Control Netw. Syst.* **2017**, *4*, 692–706. [CrossRef]
11. Tian, D.; Georganas, N.D. A coverage-preserving node scheduling scheme for large wireless sensor networks. In Proceedings of the 1st ACM International Workshop on wireless sensor networks and applications, Atlanta, GA, USA, 28 September 2002; pp. 32–41.
12. Kim, H.; Villardi, G.P.; Ma, J. Energy efficient dynamic resource allocation for wireless systems using receiver puncturing technique. *J. Commun. Netw.* **2018**, *20*, 366–373. [CrossRef]
13. Cassandras, C.G.; Wang, T.; Pourazarm, S. Optimal Routing and Energy Allocation for Lifetime Maximization of Wireless Sensor Networks with Nonideal Batteries. *IEEE Trans. Control Netw. Syst.* **2014**, *1*, 86–98. [CrossRef]
14. Ren, X.; Johansson, K.H.; Shi, D.; Shi, L. Quickest Change Detection in Adaptive Censoring Sensor Networks. *IEEE Trans. Control Netw. Syst.* **2018**, *5*, 239–250. [CrossRef]
15. Kang, J.J.W. An Inference System Framework for Personal Sensor Devices in Mobile Health and Internet of Things Networks. Ph.D. Thesis, School of Information Technology, Deakin University, Geelong, Australia, 2017.
16. Kang, J.J.; Larkin, H. Inference of Personal Sensors in the Internet of Things. *Int. J. Inf. Commun. Technol. Appl.* **2016**, *2*, 1–23. [CrossRef]
17. Sasirekha, S.; Swamynathan, S. Cluster-chain mobile agent routing algorithm for efficient data aggregation in wireless sensor network. *J. Commun. Netw.* **2017**, *19*, 392–401. [CrossRef]
18. Kang, J.J.; Luan, T.; Larkin, H. Data Processing of Physiological Sensor Data and Alarm Determination Utilising Activity Recognition. *Int. J. Inf. Commun. Technol. Appl.* **2016**, *2*, 108–131. [CrossRef]
19. Panangadan, A.; Ali, S.M.; Talukder, A. Markov decision processes for control of a sensor network-based health monitoring system. In Proceedings of the National Conference on Artificial Intelligence, Ho Chi Minh City, Vietnam, 19 May 2015; AAAI Press: Menlo Park, CA, USA; MIT Press: Cambridge, MA, USA; London, UK, 1999; Volume 20, p. 1529.
20. Fei, X.; Boukerche, A.; Yu, R. A POMDP Based K-Coverage Dynamic Scheduling Protocol for Wireless Sensor Networks. In Proceedings of the IEEE Global Telecommunications Conference GLOBECOM, Miami, FL, USA, 6–10 December 2010; pp. 1–5.
21. Bertsekas, D.P. *Dynamic Programming and Optimal Control*; Athena Scientific: Belmont, NY, USA, 1995; Volumes 1–2.
22. Koppel, A.; Warnell, G.; Stump, E.; Stone, P.; Ribeiro, A. Policy Evaluation in Continuous MDPs with Efficient Kernelized Gradient Temporal Difference. *IEEE Trans. Autom. Control* **2017**, *66*, 1853–1863. [CrossRef]
23. Standards of Medical Care in Diabetes-2006. *Diabetes Care* **2006**, *29* (Suppl. 1), s4–s42. Available online: https://care.diabetesjournals.org/content/29/suppl_1/s4 (accessed on 20 June 2021).