PBVI for Heart Rate Motion Artifact Reduction and Sensor System Energy Savings

Jacob Sindorf

***Abstract*—Commercially available sensors, such as the photoplethysmography (PPG), struggle with long term use due to energy constraint criteria. PPG sensors also provide accurate signal readings when the user performs little to no motion, including activities such as sitting, standing, or laying. Activities with high motion such as walking or running receive noisy motion artifacts that potentially skew the data. This work proposes a way to apply filters to reduce the motion artifacts under high motion activities. Through a Partial Observable Markov Decision Process (POMDP) framework and a Point Based Value Iteration (PBVI) algorithm, optimal actions can be selected to either observe the accelerometer data for activity recognition, or choose to apply a filter. Simulations of states show promising results with overall positive reward of .04 when turning the accelerometer on and off. Keeping the accelerometer on at all times received a negative overall reward of -.26. These results indicate that the use of the filters and overall energy savings are possible, bringing up a way to allow the PPG sensor to be used in studies more representative of daily life.**

I. Introduction

A photoplethysmography (PPG) sensor is a commercially available health monitoring sensor with a multitude of uses. It's main functions include a pulse oximeter to measure blood oxygen saturation, and a heart rate sensor (HR). Most available PPG sensors also include an inertial measurement unit (IMU), containing both an accelerometer and a gyroscope. Although PPG sensors are commonly used to take health metrics at home and in health clinics, they require the patient to be in a little to no motion activity state for reading, including sitting, standing, or laying down. PPG sensors, especially HR readings, suffer from motion artifacts and noise when the user begins to transition to more active states such as walking or running. The noisy signals are then rendered unreadable making long term data collection on active patients difficult. Along with difficulty in motion artifacts, the sensor also relies on a single charge making energy efficiency a top priority in long term sensor use.

This work proposes a method to filter noise due to motion artifacts, and save energy in the sensor system. Motion artifacts can be reduced through the application of filters, however these filters must only be applied when motion is detected. As for energy savings, leaving both the accelerometer, and the HR sensor on at all time causes the system to use up the battery reserves. Thus in order to have long term HR signal readings, a method to choose optimal actions is needed. This work proposes a Partial Observable Markov Decision Process (POMDP) and a Point Based Value Iteration (PBVI) method. These allow for an optimal action selection system to apply noise filters in real time, while using the partially observable state through accelerometer signals.

II. Related work

POMDP and PBVI have been well described in [1]. To summarize, a POMDP closely follows a MDP except it's states are not directly observable. The agent can only perceive observations which can be used to convey incomplete information of the state. Common terminology to describe a POMDP include, states, S, actions, U and observations, Y. A state transition probability, $P_{tr} = \mathbb{P}(S_{k+1} = j \mid S_k = i, U_k = u)$, an observation probability distribution, $P_o = \mathbb{P}(Y_k = y \mid S_k = i, U_k = u)$, and a reward function, $\mathbb{E}[\sum_{k=0}^{N-1} r]$, are also needed as a POMDP can be used to optimize action selection. A POMDP can maintain a complete trace of all actions and observations known as a history, that brings up two main issues, the curse of dimensionality, and the curse of history. Both drawbacks bring up the need for an algorithm to use POMDPs in practical applications.

This is where PBVI can be used. To start, a set of beliefs can be defined as the probability distribution of current state and history. The belief can be defined as a sufficient statistic that can predict the future. However, the belief space, $\mathbb{B}$ can be large, bringing the need for a term, $\mathcal{A}$, to represent a suitable finite dimensional set of vectors of α. The value function update can be implemented as a sequence of operations on the set of vectors. However this still contains a large number of vectors. Thus a relevant subset of vectors, $\tilde{\mathcal{A}}$ can be used on a smaller dimensional subset of belief points, $\mathbb{B}$. PBVI starts with the small initial set of belief points to start a set of backup operations. Belief points grow and backup operations continue until an approximate solution is reached. The equations and methods of [1] are explained further in the application of this work.

POMDP and PBVI are the main focus of this work, however other relevant papers have found success with other methods. It has been shown that motion artifact cancellation can be successful for activities such as walking, running, and jumping through a normalized least means square method. This paper proposes a way to cancel noise using both a PPG sensor, and a motion detection sensor [2]. Being a multi sensor system, long term use would be a difficult task due to the sensors energy constraints. Motion artifact reduction is possible, however it does not take into account the energy of the sensor system. Wireless Body Area Networks (WBAN), are a series of sensors on the body with one central power unit making energy savings critical. This is where success with Dynamic Programming

(DP) has been found. DP can be used to find both accurate and low energy ways to use multi sensor systems [3] [4]. Using MDP techniques has also found success in energy savings in multi sensor systems, with one particular case using a PPG sensor [5]. The results showed promising energy savings using an MDP method indicating the use cases of DP and MDP cases in activity detection and energy savings. To expand upon these ideas, a method of POMDP can be applied to an energy saving and detection problem using a PPG sensor to actively select motion artifact filters.

III. System model

In order to correctly fit the system model into a POMDP framework, the system states, actions, rewards, transitions and observation probabilities must be described. This work uses two separate systems, denoted A and B, for comparison.

Both systems described in this section share the same state space and transition probabilities. Transition Probabilities can be described as $P_{tr} = \mathbb{P}(S_{k+1} = j \mid S_k = i, U_k = u)$, where $S_{k+1} = j$ represents the next state, $S_k = i$ represents the current state, and $U_k = u$ represents the current action. The probabilities and states are modeled after the one's derived in [3] and displayed in the Markov Chain from Zois et al [4]. States have been simplified to $S \in \{1, 2, 3, 4\}$, which corresponds to Sit, Stand, Walk, and Run respectively. The probabilities are action independent and can be simplified into the following transition probability matrix:

$$P_{tr} = \mathbb{P}(S_{k+1} = j \mid S_k = i) = P_{i,j} = \begin{bmatrix} .6 & .1 & .3 & 0 \\ .2 & .4 & .3 & .1 \\ .4 & 0 & .3 & .3 \\ 0 & .1 & .6 & .6 \end{bmatrix}$$

Differences in System A and System B are described in detail in the subsequent sections.

A. System A

System A takes observation of the accelerometer readings at every step to make decisions. It has three possible actions from an action space, $U \in \{1, 2, 3\}$, no filter used, filter 1 used, and filter 2 used respectively. Two filters are used to allow for the difference in walking and running, which may require a different filter to be applied. As the states are not directly observable, it requires an observation space denoted as $Y \in \{1, 2, 3, 4\}$ to represent the interpreted accelerometer data. It can be assumed that the accelerometer correctly recognizes the state and returns an observation value to match the state derived. However, the algorithm to correctly determine state may not always be accurate, which is why an observation probability is used. The observation probability, $\mathbb{P}(Y_k = y \mid S_{k+1} = j, S_k = i, U_k = u)$, does not depend on action or next state, and can simplify to:

$$P_o = \mathbb{P}(Y_k = y \mid S_k = i) = \begin{cases} \epsilon/3 & \forall y \neq i \\ (1 - \epsilon) & \forall y = i \end{cases}$$

Here an epsilon of 0.1 is used to represent an accelerometer system that can correctly guess the state 90% of the time.

Lastly, reward is denoted with r(i,u), where i is the current state, and u is the current action. The reward structure for system A requires that the accelerometer be kept on, meaning every reward has a cost due to the accelerometer's energy, called $eaccel$. Correct choice of action yields a reward of R, and incorrect choice results in no reward. Using actions 2 and 3 also turn on the filter and gain a small cost due to the filter, labeled $efilt$. The reward structure can organized as follows:

$$r(i, u = 1) = \begin{cases} R - eaccel & i = 1, 2 \\ 0 - eaccel & i = 3, 4 \end{cases}$$

$$r(i, u = 2, 3) = \begin{cases} R - efilt - eaccel & i = u \\ 0 - efilt - eaccel & i \neq u \end{cases}$$

B. System B

As stated, system B shares the same states and transition probabilities as system A, however it has one main difference. That being the accelerometer is not always on. This allows for energy consumption comparison between the two models by analyzing the rewards. The system now has an extra action in the action space, $U \in \{0, 1, 2, 3\}$, where action 0 turns on the accelerometer and allows access to the observation signals. This in turn expands the observation space to $Y \in \{0, 1, 2, 3, 4\}$ where the state estimation of 1 to 4 can only be accessed if action 0 is chosen, else it receives an observation value of 0. This is further shown in the observation model which is now action dependant:

$$P_o = \mathbb{P}(Y_k = y \mid S_k = i, U_k = 1, 2, 3) = \begin{cases} 1 & y = 0 \\ 0 & y \neq 0 \end{cases}$$

$$P_o = \mathbb{P}(Y_k = y \mid S_k = i, U_k = 0) = \begin{cases} \epsilon/3 & \forall y \neq i \\ (1 - \epsilon) & \forall y = i \\ 0 & y = 0 \end{cases}$$

System B also has a different reward structure, where only action 0 can access the accelerometer, as follows:

$$r(i, u = 0) = -eaccel \qquad \forall i$$

$$r(i, u = 1) = \begin{cases} R & i = 1, 2 \\ 0 & i = 3, 4 \end{cases}$$

$$r(i, u = 2, 3) = \begin{cases} R - efilt & i = u \\ 0 - efilt & i \neq u \end{cases}$$

IV. Methodology/analysis

Minimizing energy cost and maximizing accuracy stands as the overall goal to gain. Given the structure of the rewards, it is best to maximize the positive reward, R, in order to obtain accurate guesses. System A and B allow for a comparison between energy consumption of the filters and accelerometer as they apply a cost to the overall reward. A finite horizon problem can be used to approximate the overall reward such

that $\max_{policy} \mathbb{E}[\sum_{k=0}^{N-1} r]$. With k as an index, N as the number of stages, and no terminal reward.

Through a PBVI algorithm, it is possible to derive an optimal action when given a state, observation, and initial belief. In order to satisfy the PBVI algorithm, it requires a set of hyperplanes that have been updated to maximize the value function of a subset of belief points. Those belief points must be a representation of the entire belief space. Thus in order to solve the problem, a belief update function and an algorithm to maximize the value function must be described. Then a one step look ahead problem can be created to execute the algorithms and return an optimal action. The following derivations and equations are in reference to [1].

A. Belief Update Function

A belief at sometime k can be defined as $\beta_k(i) = \mathbb{P}(S_k = i \mid H_k)$, where H is the history of observations and actions. The next belief can then be described as $\beta_{k+1}(j) = \mathbb{P}(S_{k+1} = j \mid H_{k+1})$ and can be written as a function of the current belief, β_k, action, and observation, $\beta_{k+1} = B(\beta_k, U_k, Y_k)$. This can be expressed in the form:

$$\beta_{k+1}(j) = \frac{\sum_i \beta_k(i) P_o P_{tr}}{\sum_i \beta_k(i) P_o} \tag{1}$$

Where P_o represents the observation probability and P_{tr} represents the transition probability as defined earlier. Here β represents a vector of size 1x4 as j can take any value $S_{k+1} \in \{1, 2, 3, 4\}$. so for all j and i combinations, the new β_{k+1} can be created given the old β_k, the action, u, and the observation, y.

For system A, the beta update does not require an action, and will always use the form shown in eq. 1. System B only chooses eq. 1 if action 0 is selected as it now has access to the observation signal. For actions $u \in 1, 2, 3$, eq.1 can be simplified as $\sum_i \beta(i) = 1$ and the P_o for those actions are 1 when y = 0. This simplifies to:

$$\beta_{k+1}(j) = \sum_i \beta_k(i) P_{tr} \tag{2}$$

Where the next belief represents a 1x4 vector of all j,i pairs. Using both eq. 1 and 2, the belief update function, $\beta_{k+1} = B(\beta_k, U_k, Y_k)$, can be created for both system A and B.

B. Approximate PBVI

To initialize, a Q-dimensional subset of belief points can be determined where $\mathbb{B} = \{\beta^1, \beta^2, ..., \beta^Q\}$, indexed with β^ℓ. Each β must sum to 1, and can be created either randomly, or by discretization. Random belief points require a larger Q value to maintain the same accuracy, so it is advisable to use a discretized set. The discretized set ensures the space covers all values between 0 and 1 and can be created with $\mathbb{B} = [\frac{a}{Z}; \frac{b}{Z}; \frac{c}{Z}; 1 - \frac{a}{Z} - \frac{b}{Z} - \frac{c}{Z}]$, where $a, b, c \in \{0 : Z\}$ and $(a + b + c) \leq Z$. This creates a subset of belief points of size 1 to Z, where it can be seen later that Z = 5 yields a Q = 56. For each of the belief points created, an α vector can be associated with it with the subset of relevant hyperplanes as $\tilde{\mathcal{A}}_N = \{\alpha_N^\ell : \ell = 1, ..., Q\}$ where N is the number of stages. This set can be initialized with $\alpha^\ell = [0; 0; 0; 0]$.

Using a PBVI algorithm gets a sufficiently accurate set of hyperplanes denoted as $\tilde{\mathcal{A}}_0$ by iteratively updating $\tilde{\mathcal{A}}_k$ from $k = N - 1 : 0$. The following equations utilize a PBVI algorithm formulated as a finite horizon problem allowing it to reach 0. The equations can be looped for $k = N - 1 : 0$ to create $\tilde{\mathcal{A}}_0$. To save memory, all past $\tilde{\mathcal{A}}_k$ values can be forgotten and only the final $\tilde{\mathcal{A}}_0$ set needs to be kept.

The max between the inner product of the next belief and α gets the previous value function as seen in eq. 3. This can then be used to get the current value function in eq. 4.

$$\tilde{V}_{N-k-1}(B(\beta^\ell, u, y)) = \max_{\alpha \in \tilde{\mathcal{A}}} \langle B(\beta^\ell, u, y), \alpha \rangle \tag{3}$$

$$\begin{aligned} \tilde{V}_{N-k}(B(\beta^\ell)) &= \max_{u \in U} \sum_i \beta^\ell(i)[r_k(i, u) \dots \\ &+ \sum_y \mathbb{P}(Y_k = y \mid S_k = i, U_k = u) \tilde{V}_{N-k-1}(B(\beta^\ell, u, y))] \end{aligned} \tag{4}$$

The index of the maximum when calculating $\tilde{V}_{N-k}$ yields an optimal action, called u^*. Using the optimal action, the vectors associated to the future value function can be calculated as:

$$\alpha_{k+1}^y = arg \max_{\alpha \in \tilde{\mathcal{A}}} \langle B(\beta^\ell, u^*, y), \alpha \rangle, \quad \forall y \in Y \tag{5}$$

Now the new vectors can be computed as:

$$\begin{aligned} \alpha_k^\ell &= [r_k(i, u^*) + \dots \\ &\sum_{j,y} \mathbb{P}(Y_k = y, S_{k+1} = j \mid S_k = i, U_k = u^*) \alpha_{k+1}^y(j)]_{i \in S} \end{aligned} \tag{6}$$

Eq. 6 can be simplified to $\alpha_k^\ell = [r_k(i, u^*) + \sum_{j,y} P_o P_{tr} \alpha_{k+1}^y(j)]_{i \in S}$. These updated α vectors can be stored in $\tilde{\mathcal{A}}_k = \{\alpha_k^\ell : \ell = 1, ..., Q\}$. With each new k, it returns a new $\tilde{\mathcal{A}}_k$ until the final $\tilde{\mathcal{A}}_0$ is reached.

A one step look ahead problem can be used to solve the execution of the algorithm and get the overall optimal action. With $\tilde{\mathcal{A}}_0$ calculated, the following execution can be performed:

$$\begin{aligned} U_k &= arg \max_{u \in U} \sum_i \beta_k(i)[r_k(i, u) \dots \\ &+ \sum_y \mathbb{P}(Y_k = y \mid S_k = i, U_k = u) \max_{\alpha \in \tilde{\mathcal{A}}} \langle B(\beta^\ell, u, y), \alpha \rangle \end{aligned} \tag{7}$$

Now with a given state, observation, initial β value, and $\tilde{\mathcal{A}}_0$, eq. 7 can be used to pick the optimal action.

V. Simulations/Experiments

In order to test the algorithm, a simulated environment can be created to simulate a sequence of states and observations. To simulate states, a Markov chain generator can be created using the transition probability and an initial distribution of $P_0(i) = .25$. To provide a uniform distribution, a Gumbel max trick can be used, where $G(i) = -log(-log(Uniform(01)))$.

Then the index of the max can be chosen as the next state as follows:

$$s_k = arg\max_{i \in S}[G(i) + log(P_{tr})] \tag{8}$$

With a generated state, an observation can be generated given the observation probabilities. Thus 90% of the time the observation y_k matches the state s_k, otherwise it has an uniform chance to become any of the other states $\in S$.

With a simulated environment defined, multiple simulations can be ran to compare System A and B. For each test, a thousand states and observations can be simulated. Then, 100 realizations can be done, giving 100 different simulated tests each of length 1000. Each realization uses the same simulated sequence of states and observations for every test in order to have a fair comparison of results. For each state and observation, a reward can be calculated based off the reward functions defined previously. Then for each realization a reward value can be calculated to get the average reward per stage, such that $\frac{1}{1000}\sum_{k=0}^{1000} r(s_k, u_k)$. The mean across all realizations can also be calculated to see the average of each.

System A and B both use the same belief space, Q = 56 and N = 100, when calculating the $\tilde{\mathcal{A}}_0$ values prior to the simulations. Both use a reward, R, of 1, an $eaccel$ cost of 0.5, and an $efilt$ cost of 0.2. For each of the 1000 iterations, an action can be determined using eq. 7. System A always has access to the observation, y_k, however system B only has access to y_k when an action of 0 is chosen. For all other actions, $u \in \{1, 2, 3\}$, in system B, a $y_k = 0$ must be given to the belief update function in eq. 7.

In order to compare the results from system A and B, 4 other policies were created. Three of which use the reward set up from system A, and one that uses the rewards from system B. The first three are guess policies that choose an action randomly. The first, called off, chooses action 1 at all times. The second, called even, chooses each action, $u \in \{1, 2, 3\}$, uniformly. The Third, called uneven, chooses action 1 50% of the time, and 2 and 3 25%. Fig. 1 displays the results comparing the guess policies with system A. The policy that uses system B's rewards is called periodic. At every even time step, this system chooses action 0 and gains access to the observations. It then chooses the next action based on the observation. Fig. 2 displays the comparison between the periodic policy, system A, and system B.

From Fig. 1, it can be seen that overall the average cost from system A exceeds just randomly guessing an action, which would make sense as it would accumulate more rewards, R, for its correct guesses. Due to the accelerometer always being on, it finds a negative reward overall as it would consume more accelerometer energy which begins to outweigh the correct guess. Fig. 2 also includes system A, however in comparison it has a much lower reward than the periodic and system B. As the accelerometer is selectively turned on, system B is able to gain a better balance between reward and energy cost. The periodic system uses the accelerometer every even step which also lowers its overall reward. It can be seen from both

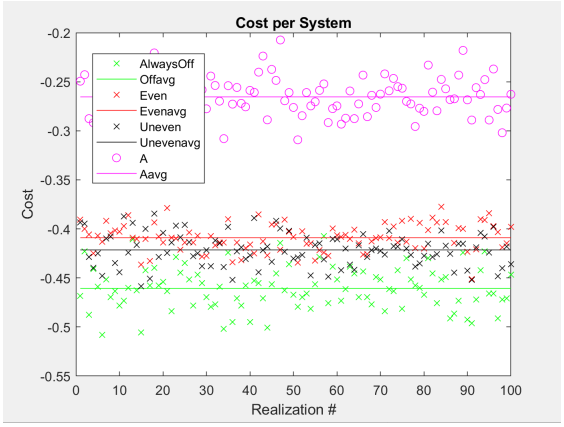

Fig. 1. Cost versus realization to display the cost per stage. Displays the result of each realization as well as the average to compare system A to the 3 guess policies.

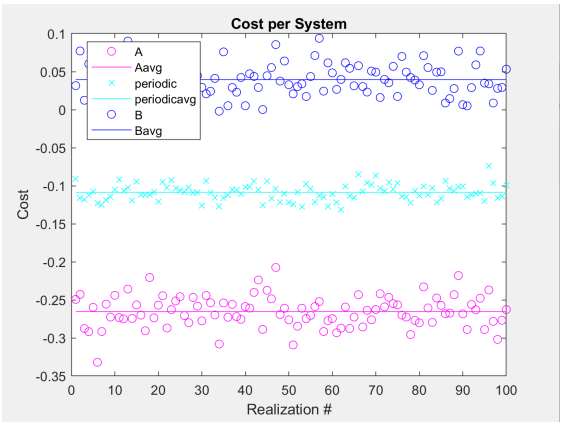

Fig. 2. Cost versus realization to display the cost per stage. Displays the result of each realization as well as the average to compare system A to system B and the periodic policy.

figures that a higher cost can be achieved when selecting the accelerometer specifically as in system B. However, the $eaccel$ value was fixed at 0.5, which is about half of the positive reward for correct guess. To see the effect the $eaccel$ value has on each system, Fig. 3 compares the mean over all 100 realizations for $eaccel$ values from 0 to 1 by 0.2.

By changing the $eaccel$ cost value, it is possible to further visualize the performance of system A and B. For low accelerometer costs, system A receives higher rewards. This would be due to having full access to the accelerometer at all times, and having little to no negative cost due to accelerometer energy. System A quickly decreases with a linear trend as the cost of $eaccel$ is increased to 1. System B on the other hand remains constant regardless of the $eaccel$ value. It can be seen that system B would outperform system A in most cases as the cost of the $eaccel$ has little to no effect on the final outcome of reward.

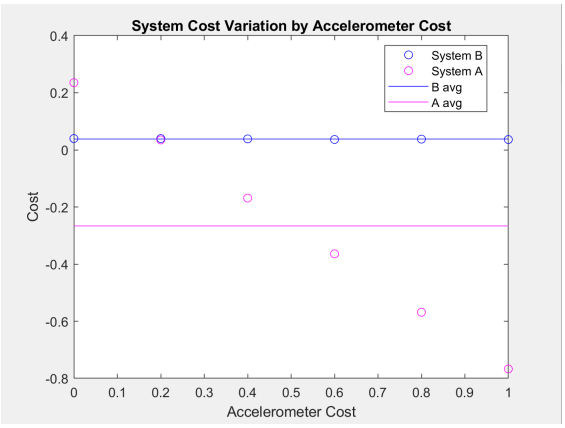

Fig. 3. Accelerometer cost effect on System A and System B. Displays the mean of the 100 realizations for different values of $eaccel$ costs.

VI. Concluding Remarks

Using a PBVI algorithm provides a way to select optimal actions on a sensor system. Here a simulated experiment of a PPG sensor shows how accounting for energy cost and state estimation is possible. This work takes and compares two main systems for filtering Heart Rate noise while observing state from accelerometer data. Although the filters and processing of the sensor data is outside the scope of this work, it would be possible to use the PBVI framework in real time. Having two possible filters also allows for the final system to be robust enough to filter small and large noise obstructions from motion.

From the results, it can be shown that a system similar to system B can provide action choices and energy saving. By selectively choosing when to use the accelerometer as well as the filters, the system is capable of saving energy. System B also showed constant results regardless of accelerometer energy cost, which could point to the use of PBVI algorithms in high energy costing sensors. Noise reduction in the heart rate sensor would also be possible given the right filters, allowing for PPG sensors to be used in more activity cases with high motion artifacts to monitor heart rate.

Overall the system shows promising results for use in real time sensor systems. Its ability to reduce accelerometer cost and provide optimal action choices provide a framework for other energy saving sensor systems. Energy savings and filter choice also point towards the use of this algorithm in long term research involving a PPG. As it can be robust enough to apply multiple motion artifact filters, and save energy for use over long periods of time.

VII. Future Work

Active selection of noise canceling filters and energy savings provides a promising framework for future projects. As this project provides a proof of concept through simulation, application in real time would be a major area of future work. That would include using real data from a PPG sensor or sensor system, as well as noise filter selection. The accelerometer data would have to be analyzed and a guess of state would have to be output in real time for use in this work. Heart rate monitoring would also be an important area as it would allow for visualization of filter selection. Once proper filters are tested offline, they can be applied to the system and in real time the system can be verified by the proper noise filter application. Real time activity tests could be performed to see how well the filters are applied when specific states are reached.

As well as noise cancellation, the other important area of study would be the energy consumption. Energy consumption in sensor systems remains an issue in long term studies using energy constrained sensor systems. The results from this work show how energy costs of certain sensors may not fully change the outcome of the system. Thus future work on larger sensor systems could be done to test the effectiveness on energy savings. This would allow the proposed system to be used in multiple cases and sensor setups to provide long term energy savings.

Further contributions would be to compare the these results to state of the art work in terms of complexity and accuracy. New algorithms can also be explored to test their overall results to those obtained with PBVI. Some areas of interest could be Reinforcement Learning, where the system can be trained to recognize and apply noise filters.

The work done in this research paper provide a framework for future studies. The promising results indicate that a PBVI algorithm can be used in real time for both accurate filter selection and active energy savings.

References

[1] J. Pineau, G. J. Gordon, and S. Thrun, "Anytime point-based approximations for large pomdps," *CoRR*, vol. abs/1110.0027, 2011. [Online]. Available: http://arxiv.org/abs/1110.0027

[2] T. Shimazaki, S. Hara, H. Okuhata, H. Nakamura, and T. Kawabata, "Cancellation of motion artifact induced by exercise for ppg-based heart rate sensing," in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2014, pp. 3216–3219.

[3] G. Thatte, M. Li, S. Lee, B. A. Emken, M. Annavaram, S. Narayanan, D. Spruijt-Metz, and U. Mitra, "Optimal time-resource allocation for energy-efficient physical activity detection," *IEEE Transactions on Signal Processing*, vol. 59, no. 4, pp. 1843–1857, 2011.

[4] D.-S. Zois, M. Levorato, and U. Mitra, "Energy-efficient, heterogeneous sensor selection for physical activity detection in wireless body area networks," *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1581–1594, 2013.

[5] D. Amiri, A. Anzanpour, I. Azimi, M. Levorato, P. Liljeberg, N. Dutt, and A. M. Rahmani, "Context-aware sensing via dynamic programming for edge-assisted wearable systems," *ACM Trans. Comput. Healthcare*, vol. 1, no. 2, mar 2020. [Online]. Available: https://doi.org/10.1145/3351286