Duty Cycle Optimization of Sensors in Wireless Body Area Network Based on Reinforcement Learning

First A. Author, *Fellow, IEEE*, Second B. Author, and Third C. Author, Jr., Member, IEEE

*Abstract*—Blood pressure (BP) is one of many rhythmic physiological indicators, and its home monitoring has great value and long-term significance. This paper establishes a two-dimensional blood pressure change model based on Circadian and Seasonal variation of blood pressure. We use the maximum and trend to characterize the state of blood pressure over a period of time. Blood pressure sensors can be used to sense the BP data in wireless body area network (WBAN). Our goal is to achieve a balance between power saving and low latency by optimizing the sensor duty cycle based on the BP model. To solve the problem, we establish it into a markov decision process (MDP) and then use the Deep Q-learning Network(DQN) method in Reinforcement Learning (RL) to find the best result. To test the validity of the algorithm, we simulated 180 days of blood pressure data for testing. Simulation results show that the algorithm can effectively reduce energy consumption and perform well in sensing delay.

*Index Terms*—blood pressure, duty cycle, Reinforcement Learning, wireless body area network

# INTRODUCTION

As the level of medical care continues to improve, the level of medical services that people enjoy continues to improve. In order to enable more people to enjoy the high-quality services brought by modern medical care, more and more high-tech is being used. The family health monitoring system allows patients to be physically monitored at home or in the office without having to stay in the hospital. At the same time, a more familiar and comfortable environment is also conducive to patient recovery. Wireless body area network (WBAN) is a technology commonly used to implement a home health monitoring system [1]. Taking BP monitoring as an example, Fig. 1 is a schematic diagram of the WBAN model.

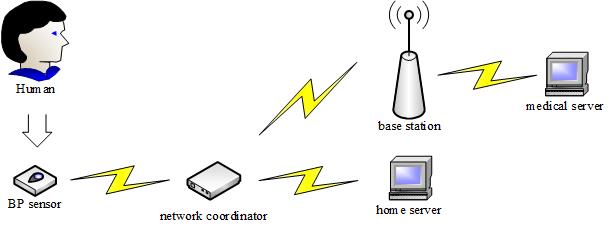


Fig. 1 schematic diagram of the WBAN model

The BP sensor is placed on the surface of the skin or embedded in the human body, and the data is collected and sent to the network coordinator. Finally, the data is sent over the wireless network to the home server and medical server. Patients and medical staff can analyze and process the collected data.

Like the general wireless sensor network (WSN), the sensor node battery life in the WBAN is limited, so it is also necessary to consider the service life of the sensor node [2]. In article [2] the duty cycle is defined as the ratio between the time during which a sensor node is in active state and the total time of active/dormant states. It considered the upper-bound of the WSN lifetime based on random duty cycle. This paper focuses on optimizing the duty cycle of the sensor to save energy while not damaging the immediacy of the induction by understanding the process of physiological index changes.

The major contributions of this paper can be summarized as follows:

* We establish a two-dimensional blood pressure change model based on circadian and seasonal variation of blood pressure, using the maximum and trend to characterize the state of blood pressure over a period of time.
* We design an algorithm based on DQN to optimize the duty cycle of the sensor for energy saving and low latency. It is verified by simulation that the algorithm is effective.

# blood Pressure Change Model

Human physiological indicators are mainly affected by nerves and body fluids (hormones), and have the characteristics of circadian and seasonal variation. We divide a change cycle into multiple observation windows, and calculate several statistics of physiological indicators in the windows, which can effectively help us understand the patient's physical condition. Due to the rhythm of physiological indicators, the corresponding statistics also have a certain change law.

## Circadian and Seasonal Variation

Ambulatory blood pressure measurement plays an important role in monitoring certain diseases. Article [3] studies the changes in BP in patients and gives some images of circadian pattern of systolic BP of normotensive and hypertensive patients. In most individuals, BP rises significantly after getting up in the morning and falls at night.

Article [4] studied the seasonal variation in BP over a longer span of time and concluded that systolic BP and diastolic BP are higher in winter than in summer. This process of seasonal variation can be seen as a process of gradual change between day and day, essentially related to climatic factors such as outdoor temperature.

## Observation Windows

As mentioned above, changes in physiological indicators are periodic, and the regularity of changes in each cycle is stable. For example, the change cycle of blood pressure is 24 hours, higher in the morning and lower in the evening. Since the range of blood pressure changes is relatively large, it is necessary to divide a cycle into smaller intervals. As a result the magnitude and complexity of the numerical changes in the interval are reduced, which is beneficial for us to observe the physiological condition of the patient.

As shown in Fig. 2, we divide a cycle T into several windows (i=1,2, ,N). The length of each window is determined according to the situation. Different window lengths may be equal or unequal. Each window has its own sensor duty cycle, which is the focus of our research.

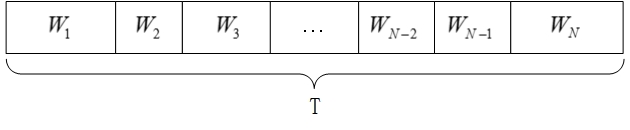


Fig. 2 window lengths may be equal or unequal

Within each window we define multiple time slots of equal length （m=1,2,3,…）and control the sensor for sampling or sleep in each time slot. Each window has its own unique sensor sampling scheme. Obviously, when the proportion of sampling is large, the monitoring effect is better but the energy consumption is larger. When the proportion of sleep is large, the energy consumption is small but has an effect on the monitoring effect. So our goal is to find a balance between energy consumption and monitoring effect. Fig. 3 is a schematic diagram of time slots.

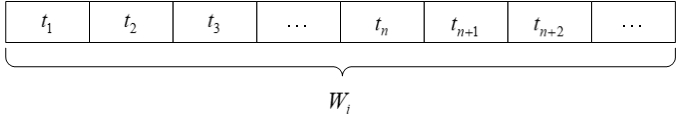


Fig. 3 time slots

## Two-dimensional Windows Model

Taking into account the circadian and seasonal variation of blood pressure, we consider the change in blood pressure within one day as one direction, and the change between day and day as another direction. Assuming that the lengths of the windows are equal, we can obtain the two-dimensional model of observation windows. Fig. 4 is a schematic diagram of the model.

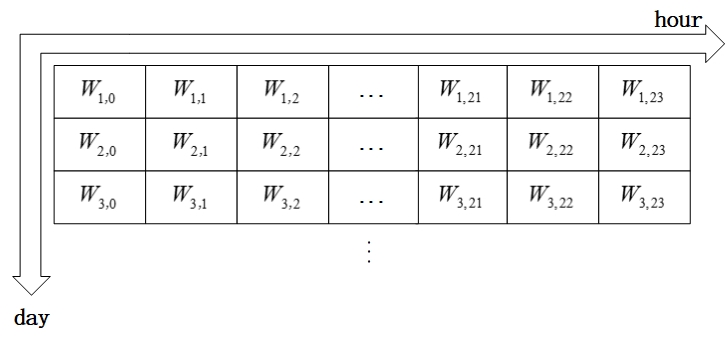


Fig. 4 schematic diagram of model

As mentioned before, each window has a series of blood pressure data, and we will process the data in the window to extract features in later section.

# Window Feature Vector

The BP data in the window is usually too many and not intuitive. In order to better describe the state of BP during this time period, we performed statistical processing on BP data. This process involves calculating the maximum value and trend of the data. After obtaining the statistical results, we divide it into multiple intervals, corresponding to multiple maximum value states *v* and trend states *z*, and finally get the window feature vector (*v*, *z*).

## Maximum Value State

The maximum blood pressure data within the window characterizes the peak reached by the patient's blood pressure during that time period. For the patient, the maximum is the most dangerous moment in the period. When the maximum value is within the interval exceeding the normal value, both the medical staff and the patient should pay sufficient attention to it. Table 1 is the latest WHO standard, which divides systolic BP into multiple classes. We decided to use a similar partitioning criterion to extract the feature from the maximum value and obtain the maximum value state *v*.

Table 1 WHO standard of systolic BP

|  |  |
| --- | --- |
| Classes | systolic BP(mmHg) |
| Ideal BP | <120 |
| Normal BP | <130 |
| Normal high BP | 130-140 |
| Hypertension | >140 |

Based on Table 1 we can divide BP into multiple intervals. The patient is hypertension when the systolic BP exceeds 140 mmHg. 130 mmHg to 140mmHg is normal high BP, and 120 mmHg to 130mmHg is normal BP. Systolic BP of less than 120 mmHg is ideal. Considering that the hypotensive standard is generally less than 90 mmHg, we divide the peak of BP into multiple intervals, as shown in (3.1).

 (3.1)

The value of *v* can characterize the peak condition of the current window. Obviously, when *v* equals 4, it indicates that the patient's BP exceeds the normal value in this window，which needs to be vigilant. When *v* is equal to other values, it can also describe the peak situation to a certain extent, which is good for us to judge the physical condition.

## Trend State

There are many statistical tests that can be used to assess the importance of time series trends. One of the commonly used nonparametric trend tests is the Mann-Kendall trend test. It is widely used to distinguish whether a natural process is in natural fluctuations or there is a definite trend. This paper uses this test method to determine whether the BP in the windows shows a significant change trend.

The rank correlation test for a set of observations is formulated as follows. Eq. (3.2) is the calculation method of *S*:

 (3.2)

where

 (3.3)

When n is large enough, the statistic *S* tends to be normally distributed and the variance of *S* is calculated as in Eq. (3.4):

 (3.4)

The Mann-Kendall trend test is calculated as in Eq. (3.5):

 (3.5)

satisfies the standard normal distribution. When the absolute value of is greater than or equal to 1.28, 1.64, 2.32, it means that the confidence is 90%, 95%, and 99% significance test respectively.

With a confidence of 99%, we define three intervals for as three states, as in Eq. (3.6):

 (3.6)

Obviously, the data trend is decreasing when , the data trend is flat when , and the data trend is rising when . We use the value of z to determine the trend of blood pressure in the windows. Combined with the peak state *v* mentioned above, we obtain the window feature vector (*v*, *z*), which can effectively describe the blood pressure data in the windows. In the following sections, we use the vectors to study, so that the sensor adjusts the duty cycle according to the actual situation.

# Coupled Markov Chain Theory

In the section 2 we define the observation windows and in the section 3 define the feature vector describing the state of the data within the windows. Due to the circadian and seasonal variation of BP, the eigenvector sequence can be established as a Markov chain. Because blood pressure changes in two dimensions, the one-dimensional Markov chain cannot fully describe the process, so we need to establish a two-dimensional coupled Markov chain and list the transition probability. Article [5] gives the concept of a coupled Markov chain.

The Markov chain is a probabilistic model that exhibits special dependencies. The future depends on the current and has a transition probability, but is not associated with the past. In the model of this paper, the Markov property is expressed as: the future window feature vector *W* depends on the current *W.* For a single-dimensional Markov chain, such a property can be expressed as in Eq. (4.1):



(4.1)

where is the sequence of window feature vectors and is the state space.

Obviously, represents the probability of transitioning from state to state , where n is the number of states in the model. Thus we can use to represent the one-dimensional state transition probability, and there are two constraints expressed as shown in Eq. (4.2):

 (4.2)

The coupled Markov chain describes the joint behavior of Markov chains in both directions. The current state depends on the previous state in both directions and has a joint transition probability. Similar to the one-dimensional Markov chain, we describe the Markov chain in the horizontal direction as shown in Eq. (4.3):

 (4.3)

where is the window feature vector of the jth window of the ith day.

The Markov chain in the vertical direction is described as shown in Eq. (4.4):

 (4.4)

Coupled Markov chains are obtained by coupling horizontal and vertical chains, but forcing the chains in both directions to reach an equal state. Thus we get the joint transition probability:

 (4.5)

where *C* is a normalizing constant whose value is equal to that shown in Eq. (4.6):

 (4.6)

Thus we get the joint transition probability by combining Eq. (4.5) and Eq. (4.6):

 (4.7)

The joint state transition probability indicates the effect of the window state of the previous day and the previous period on the current state. This means that the current state can be predicted to some extent. In the following sections we will use the RL method to optimize the sensor duty cycle based on the window feature and joint transition probability.

# Optimization Goal

The goal of this paper is to optimize the duty cycle of the sensors based on the state of the windows and the transition probability. In general, the duty cycle of a sensor is the ratio of the length of sampling to the period in a period. A higher duty cycle means that the sleep time is shorter in the same amount of time. As mentioned before, the patient's BP is generally low at night, and if the high duty cycle is still used, the energy of the sensor is wasted. Therefore, we use the RL method to adaptively optimize the duty cycle to reduce energy consumption without affecting the use effect.

In order to facilitate us to judge the effect of a certain duty cycle, we specify the cycle mode of sensors sampling and sleep as shown in Fig. 5. In the window, it is assumed that the number of time slots in sampling in one cycle is , and the number of time slots in sleep is . The acquisition starts from the first time slot of the window, and after collecting slots, it sleeps, and after slots, it starts sampling again. Loop through this mode until the end of the window.

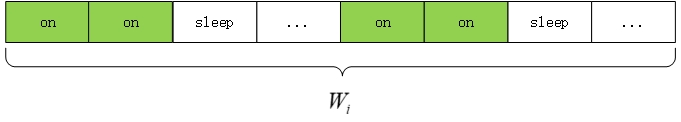


Fig. 5 cycle mode of sensors sampling and sleep(,)

Since and are small relative to the length of the window, we calculate the duty cycle of the window as shown in Eq. (5.1):

 (5.1)

In this paper, we use two calculations when we judge whether the duty cycle in the window is appropriate: energy consumption and delay *.* The final goal is to find the optimal duty cycle based on these two calculations.

Article [2] introduced a formula for calculating the energy consumption (J/s) of a sensor for WSN as shown in Eq. (5.2):

 (5.2)

where is the average sensing rate, is the energy consumption to sense a bit of data, is the transceiver relay data rate, is energy consumed per bit by the transmitter electronics, is the energy consumed per bit in the transmit op-amp, is the transmission distance and is the sleep state energy consumption per second.

In general, the energy consumed by the sensor during sleep is relatively low, so the energy consumption of the sensor is positively related to the duty cycle . When the is large, the energy consumption is large, and vice versa. For most sensor nodes in WSN, the energy of transmitting per bit of data is much greater than the energy consumed by sensing per bit of data. The energy consumption by a sensor node in WSN per second across a distance *d* with path loss exponent *n* is as shown in Eq. (5.3).

 (5.3)

In a specific scenario and application, the transmission power has a maximum and minimum limit as shown in Eq. (5.4).

 (5.4)

After the duty cycle in a window is determined, according to our specified cycle mode of sensors sampling and sleep, we can define the actual delay as the interval between the time when sensor first senses the abnormal blood pressure and the time when the blood pressure actually reaches abnormal in one window. Obviously, indicates the degree of danger of the current duty cycle. If is high, the current duty cycle may cause the patient's condition to be undetected and processed in a timely manner. This can be very dangerous for the patient. So we want to control the within the allowed range as shown in Eq. (5.5).

 (5.5)

Therefore, our goal is to find the right duty cycle to achieve the best balance of energy consumption and delay. To quantify the plausibility of the duty cycle of each window, we define a reward function based on energy consumption and delay as shown in Eq. (5.6).

 (5.6)

The function  is a decreasing function with respect to and . When the values of and are small, the value of the reward function is large. So our goal is to maximize the sum of the reward functions of all windows and meet some constraints, as shown in Eq. (5.7).

 (5.7)

# Algorithm based on reinforcement learning

Reinforcement Learning is a common machine learning method. For the specific problem, it obtains the influence of each action on the state and the reward obtained by the action through the reward function. The goal is to get the optimal policy , which is the sequence of best actions. We designed an algorithm based on Q-Learning in RL and used it to solve our optimization problem.

## Q-Learning

A Markov decision process (MDP) can be characterized by the following variables: state space (which is a finite set), action space (a set of finite actions), state transition probability matrix, reward function, and discount factor. State refers to the state in which the agent is currently located, usually consisting of a series of discrete values that make up its state space. Action is an action taken by agent that affects the state and is usually a limited set. The state transition probability describes the probability of a transition between states. The state transition probability is one of the components describing the Markov process. The reward function is used to describe the degree of advantage of the state, usually a positive value indicates an acceptable state and a negative value indicates a state that should be avoided.

|  |  |
| --- | --- |
| **Algorithm** DQN-Based optimization window duty cycle scheme | |
| 1: | Initial Q-table and *s*; |
| 2: | Initial α,γ, and N; |
| 3: | Initial ; |
| 4: | repeat |
| 5: | Initial ,; |
| 6: | Initial and the sum of reward *R*=0; |
| 7: | Initial and ; |
| 8: | ; |
| 9: | repeat |
| 10: | Randomly generate ; |
| 11 | Calculate *v* and *z* of ; |
| 12: | Calculate ; |
| 13: | If : |
| 14: | Choose a randomly; |
| 15: | If: |
| 16: | ; |
| 17: | ove to next window; |
| 18: | Change and according to *a*; |
| 19: | Obtain *s*’ ,and *r*; |
| 20: | ; |
| 21: | If >: |
| 22: | Back to step 6 |
| 23: | Update Q-table by Eq. (6.1); |
| 24: | until( is the last observation window) |
| 24: | ; |
| 25: | Until() |

Using several elements of MDP, we can create a Q-table.Q-table maps the state space to the action space one by one, and the values in the table represent the maximum future reward expectations for taking the action in that state. The initial value is set to 0. Based on this table, the agent can choose the appropriate decision for each action.

In order to get a Q-table with good enough effect, we use the Q-Learning method. In each episode, we explore the action by using the *ε*-greedy strategy. This strategy chooses exploitation with the probability of *ε* and chooses exploration with a probability of .If it is the exploration phase, we randomly select an action corresponding to the current state. If it is the exploitation phase, we select the action with the largest . After the action selection, the next state and the corresponding reward function are obtained, and the of the current state in the Q-table is updated by Eq. (6.1)

 (6.1)

where is the reward for taking that action at that state, is the maximum expected future reward given the new state and all possible actions at that new state,α is learning rate and is discount rate.

## Optimization Algorithm

Based on the Q-Learning, we designed an optimization algorithm to solve our problem. According to the principle of the algorithm, we initialize the Q-table and each coefficient, and set the *ε-*greedy algorithm to be related to the number of episode. When the number of episode increases, the probability of exploitation will be higher. The reward function is calculated during the episodes and the Q-table is updated according to Eq. (6.1) until the training is completed.

In our algorithm, the state vector *s* is set as (*v*, *z*,,). For a particular window feature vector (*v*, *z*), the final goal of our training is to have a specific duty cycle expressed as a specific combination of and . Action *a* is set to multiple values, corresponding to changing and . At the beginning of the algorithm, the parameters are initialized, and the values in the Q-table are all set to 0. In each episode, we start from the first observation window and calculate *v* and *z*. We calculate the sum of energy consumption based on the duty cycle of the current window. Then we find the corresponding state vector *s* in the Q-table. We set *ε* to the expression associated with episode as shown in Eq. (6.2).

 (6.2)

where is a constant used to control the convergence speed of the algorithm.

In each episode, a random number is generated in each window and compared with *ε* to determine the selection strategy of the action. According to the *ε*-greedy strategy, if in the exploration phase, one action is randomly selected, and in the exploitation phase, the action with the largest  is selected. After that, the and of the next window are changed according to the action. Then slide the window backwards and calculate the reward function r, which can be used to update the.If the delay exceeds the maximum , go back to the first window and start again.Repeat this process until the number of episode reaches the maximum value N we set at the beginning.

# Simulation results

## Parameter setting

To verify the feasibility of the algorithm, we tested it using the simulated systolic BP data. The simulated systolic BP data is derived from the changes described in articles [3] and [4]. Changes in systolic BP have circadian and seasonal variation. During the day, systolic BP has a peak in the morning and evening, and a trough occurs in the night. On the other hand, the systolic BP in summer is lower than that of winter. Therefore, based on these rules, we generated 180-day data to simulate changes in systolic BP in hypertensive patients from summer to winter. There is one data per second.

As described in the algorithm, we need to define the action space and the reward function *r*. The action *a* is a change of and . When the action is selected, the and of the sensor are also changed, which changes the duty cycle of the sensor at the same time. Since and are small, the change should also be small. Thus, the action space we designed is shown in Eq. (6.3).

 (6.3)

The reward function needs to consider energy consumption and delay. Our goal is to minimize energy consumption when the delay is not greater than the maximum. Therefore, the reward function *r* should be designed as a piecewise function related to the delay. According to Eq. (5.2), the change in the sum of energy consumption after the duty cycle of the sensor is changed can be expressed by Eq. (6.4).

 (6.4)

where and are the sum of the duty cycles of all windows in initial and current episodes.

We divide the reward function into two parts. One is for controlling delay and the other is for energy consumption. the reward function of delay can be designed as shown in Eq. (6.5).

 (6.5)

Within each episode, we calculate of each window, and there is a penalty if the duty cycle within the window produces a delay that exceeds the limit.

The reward function of energy consumption can be designed as shown in Eq. (6.6).

 (6.6)

When the sum of energy consumption is reduced, the reward is positive, otherwise it is negative. By this way we can control the duty cycle to change in the direction of reducing the sum of energy consumption.

Thus, our reward function *r* can be written as shown in Eq. (6.7).

 (6.7)

Obviously, if the delay is large, then our hope is to increase the duty cycle and if the delay is small, the duty cycle should be reduced. In this way, through continuous training, a duty cycle that ensures less delay and less energy consumption can be obtained.

Learning rate α is set to 0.1 and discount rate is set to 0.9.

## Results

Based on our algorithm and parameters, we used the simulated 180-day BP data to test. We observed the variation of reward value R, total energy consumption and maximum delay with E and obtained the image. For e, we set the values of a to 0.1, 1 and 10 to observe the convergence speed and the final convergence value of the algorithm. Fig. 6 is a variation image of the reward function, Fig. 7 is a variation image of the total energy consumption, and Fig. 8 is a variation image of the maximum delay.

References

1. Otto, C. A., E. Jovanov., and A. Milenkovic. “A WBAN-based System for Health Monitoring at Home,” in *IEEE/EMBS International Summer School on Medical Devices & Biosensors*, 2006.
2. Rout, Rashmi Ranjan, and S. K. Ghosh , “Enhancement of Lifetime using Duty Cycle and Network Coding in Wireless Sensor Networks,”*IEEE Transactions on Wireless Communications*, pp. 656–667,Dec.2013.
3. Hermida, R. C., Ayala, D. E., & Portaluppi, F.,“Circadian variation of blood pressure: the basis for the chronotherapy of hypertension,” *Adv Drug Deliv Rev*, vol. 59, no. 9, pp. 904–922, 2007.
4. Woodhouse, P. R., Khaw, K. T., & Plummer, M., “Seasonal variation of blood pressure and its relationship to ambient temperature in an elderly population,” *Journal of Hypertension*, vol. 11, no. 11, pp. 1267–1274, 1993.
5. Chessa, A. G., “A markov chain model for subsurface characterization: theory and applications,” *Mathematical Geology*, vol. 38, no. 4, pp. 503–505, 2006.