

# Improving the Energy Efficiency of Wearable Computing Units Using on Sensor Fifo Memory

Ozgun Pinarer, Atay Ozgovde

## ► To cite this version:

Ozgun Pinarer, Atay Ozgovde. Improving the Energy Efficiency of Wearable Computing Units Using on Sensor Fifo Memory. International Journal of e-Education, e-Business, e-Management and e-Learning, 2015, 5 (2), pp.105-113. 10.17706/ijeeee.2015.5.2.105-113 . hal-02380167

**HAL Id: hal-02380167**

**<https://hal.archives-ouvertes.fr/hal-02380167>**

Submitted on 10 Feb 2020

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

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

# Improving the Energy Efficiency of Wearable Computing Units Using on Sensor Fifo Memory

Ozgun Pinarer\*, Atay Ozgovde

Department of Computer Engineering, Galatasaray University, Istanbul, Turkey

\* Corresponding author. Tel.: +90(212)2274480(307); email: opinarer@gsu.edu.tr

Manuscript submitted March 5, 2015; accepted March 30, 2015.

doi: 10.17706/ijeeee.2015.5.2.105-113

---

**Abstract:** Proliferation of wearable devices with wide spectrum of sensing capabilities together with commercial availability has increased the applicability of ambient intelligence concepts in practical system designs. Being wearable enforces extra constraints in terms of form factor and weight that limit the computational properties and the battery lifetime. There has been increasingly many number of studies for the energy efficiency of embedded and mobile hardware platforms. Due to the known techniques, increasing the energy consumption of an embedded system inherently requires some components to go into the low energy modes with a certain pattern, which in turn entails performance penalties at the application level. Existing solutions for increasing energy efficiency mainly focus only on a certain component of the system, such as hardware, networking firmware and try to achieve energy efficiency without considering the state the application is dynamically in. In this study, the critical balance between energy efficiency and application performance is handled. Application feedback is merged with energy efficiency and according to the application performance, duty cycle mechanism can be configured dynamically. A memory unit (FIFO) of the sensing component is also involved into the dynamic sleep scheduling mechanism in order to process latest sampled data while microprocessor and radio module of the sensor devices are in sleep mode. In this context, one of the fundamental implementations of ambient application which is based on triaxial accelerometer signal, pedometer is performed. Experiments realized on the dataset proved that it exists an interval where energy efficiency is obtained without degrading application performance under critical level and also usage of FIFO showed a significant impact on application performance and energy gain.

**Key words:** Ambient intelligence, ambient assisted living, signal processing, wireless communication, embedded computation.

---

## 1. Introduction

Ambient Intelligence (AmI) defines the technical bridge between people and their environment which is interconnected, adaptable, dynamic and especially embedded. Reference [1] describes AmI as an intelligent, embedded, digital environment that is sensitive and responsive to the presence of people. In AmI research domain, Activities of Daily Living (ADL) and context awareness are the central issues. Main intention behind AmI studies is commonly to monitor and recognize daily activities of the user; to make several assumptions by realizing AmI algorithms which rely on sensory data coming from the real environment.

AmI environment is surely constructed by various heterogeneous wireless sensors. It is possible to use these wireless sensors for location detection, habitat monitoring, radiation detection, chemicals, humidity,

temperature, velocity, pressure, smart home systems and also for measuring physiological activities like EMG, ECG, GSR for monitoring health states of a person. All of these types of sensors should be quite small and thus should be integrated into almost any AmI application. Especially sensors used in measuring psychological activities should be wearable. Being wearable, having wide spectrum of sensing capabilities and commercial availability have increased the applicability of ambient intelligence.

These wireless devices have three fundamental features: sensing, computing and communicating. Being wearable brings some features like freedom of movement and comfort. The most important constraint is the limited battery. In the literature there exists several energy efficient methods for saving energy consumption, for extending lifetime. However, these methods are component based approaches and impacts of using these methods on the application performance is not considered adequately. Regarding to this problem, in this work, a holistic approach which takes component and application level into account is proposed. Energy consumption and the ways for improving lifetime of sensor device are handled. Main intention in this study is to generate a dynamic sleep scheduling mechanism (DSS) and to find a suitable region where application can work with tolerable performance degradation meanwhile lifetime of the sensor is extended by decreasing energy consumption. While realizing this aim, in contradistinction to existing approaches, temporary memory of the sensing chip is involved into proposed approach and becomes the most important part of the system.

In this study, as an application, pedometer based on triaxial accelerometer signal is chosen. Besides, proposed on board algorithm in this study has a capability to recognize the motion and to predict the frequency of the current activity. As distinct from the existing works in the literature, in our perspective, there exists a real-time feedback structure between the sensing unit and the microprocessor (MPU) during the execution process. Depending on the embedded signal processing, sensing unit is updated. Furthermore, memory unit of sensing chip is used for improving sleep scheduling mechanism.

The remainder of this paper is organized as follows: Related works in the literature is given in Section 2. Section 3 presents proposed approach. Chosen dataset and experiment phase are introduced in Section 4. Section 5 shows the outcomes of the experiments and finally, conclusions are discussed in Section 6.

## **2. Related Works on Energy Saving**

Wireless sensors are embedded devices which combine sensing, computation and communication actions. In Wireless Sensor Networks (WSN) domain, these sensors can be used not only outdoor but also indoor. Although being wearable brings in some nice features, it also enforces extra constraints in terms of form factor and weight. These constraints, in turn, heavily limit the computational properties and the battery lifetime of the sensor device. Limited processing ability and low data rate of sensor devices cannot guarantee high performance in some scenarios especially for real-time applications.

Over the past several years, energy efficiency has been a central research theme. Several studies aimed at minimizing the energy consumption, saving energy in order to decrease the total energy consumption of the system and to increase the lifetime of the sensor device. [2] gathers and classifies existing studies on energy consumption of wireless sensor devices in the literature and gives a brief presentation of each approaches. According to Anastasi et al. it is possible to classify existing approaches into three different subsets: usage of duty cycling, data-driven techniques and mobility based methods.

The first technique to conserve energy is using duty cycling. Duty cycling concerns not only sleep scheduling mechanism for power management but also topology control. Topology control and location driven systems are used in multi-hop networks especially for updating neighbour list of each sensor device. It can be considered as a type of periodically self-organization. [3] that topology control protocols can typically increase the network lifetime by a factor of 2–3. In the power management side of the duty cycling

approaches, main intention is to apply a sleep scheduling technique. A number of solutions across multiple layers like energy-efficient MAC protocols, communication strategies, operating systems and energy saving routing mechanisms have been proposed to provide low-cost, high quality and energy efficient wireless access services [4], [5].

Second approach for the energy conservation is based on captured data. One way of the data driven approaches is the data prediction techniques which build a model of the sensed phenomenon, thus queries can be answered by the sink node by using the model instead of the actually sensed data without requiring any communication, reducing the energy consumption [6], [7]. Other ways for energy saving is to use model based sampling and adaptive sampling in order to reduce the number of samples by exploiting spatio-temporal correlations between data [8].

In the existing approaches, several energy saving methods are proposed, whereas influence of energy saving on the application performance is not considered or is ignored. Besides improving energy aware protocols and methods, effects of these improvements on the application performance is an important point that cannot be ignored. Depending on the type and importance of system usage, application performance and reducing energy consumption of sensor device should be balanced. Thus, application performance should be considered as a key metric while testing the proposed energy aware model. In this paper, our work addresses this hidden, unhandled relation.

### **3. Proposed Approach**

In this study, main features of a wireless sensor device (sensing, computing, communicating) are used in order to conserve energy. While realizing this intention, a feedback mechanism is established between the application level and the sensing unit of sensor device. With this correlation, configuration of the sensing unit can be updated, creation and management of dynamic sleep scheduling mechanism can be done in real-time by computing activity signal. As a difference from our previous studies, memory chip of sensing unit is also involved in the proposed approach in order to use dynamic sleep scheduling more efficient and with less error rate.

#### **3.1. Algorithm**

In ambient intelligence applications, wearable sensor devices create opportunity to monitor and recognize users' activities, generate personal outcomes according to the embedded computation realized on MPU of the sensor device. Analyzed activities can be users' daily activities like walking, jogging but also it can be activities which are based in physical athleticism or physical dexterity (analysis of a stroke in tennis, golf or baseball etc). As seen from the sample activities, frequency of these activities are different thus, sleep scheduling should be special for each activity. In this study, pedometer application is chosen to implement our approach and dynamic sleep scheduling mechanism that can be self adapted to the recognized activity is proposed. Our contribution for increasing lifetime of the sensor device is not only to create a self adaptive DSS but also try to maximize the sleep duration by using memory unit of sensing chip which acts as FIFO (First in First Out).

In this study, with the wearable sensor devices, activity recognition is been recognized. Frequency of the current activity (for instance stride frequency) is calculated on board by using a temporary buffer that stores previous same type of activities (like previous strides). When required information is achieved to calculate the activity frequency, a sub procedure runs to estimate next possible action. This estimation gives a chance to put the sensor device in a sleep mode(either just the radio module which consumes the biggest part of the energy budget or (radio+MPU) while waiting for the next action(for instance next stride for a pedometer application). The most important time period is when the sensor device is woken up and continues processing. When the sleep period is over, MPU can not recognize if it missed an action or not

while the sleep duration. In this point, we propose usage of memory located in sensing chip. Even MPU is in a sleep mode, sensing unit may store the latest samples in that memory which acts as a FIFO. This gives MPU to process the latest samples while it was in a sleep mode.

Furthermore, system contains also a comparison procedure that can compare the current action with the stored ones in its temporary buffer. This procedure makes a decision: either the current action is a continuation of the previous ones or the current action is a start of a new type of activity (ex: passage from jogging to walking). If the taken decision announces that it's a new type of activity, it triggers a new procedure to reset the current algorithm in order to adapt itself to the new activity. It drains temporary buffer, disables sleeping mechanism if it is enable.

As mentioned before, proposed algorithm makes sensor device to configure itself dynamically with the feedback mechanism coming from other internal components. Thanks to that, it may create and update a dynamic sleep scheduling according to the activity frequency. With the usage of memory unit of the sensing chip, DSS can run more accurately. The advantage of this approach is that it's independent from a certain activity or sensor device. As all existing approaches have, proposed method contains some risks like missing action which causes a degradation of application performance obviously. To prevent that, usage of FIFO of the sensing unit comes into prominence.

Fig. 1 shows the architecture of the proposed approach. Upper side of the figure represents the accelerometer chip of the sensor device and the bottom side is the MPU side. Sensing unit captures raw data and stores it into the FIFO. MPU fetches raw data from FIFO and use this data in the activity recognition module and the dynamic sleep scheduling module. They interact with the temporary buffer (it stores previous captured actions) in order to compare with the current one and to make an outcome as a configuration commands for radio module and MPU itself.

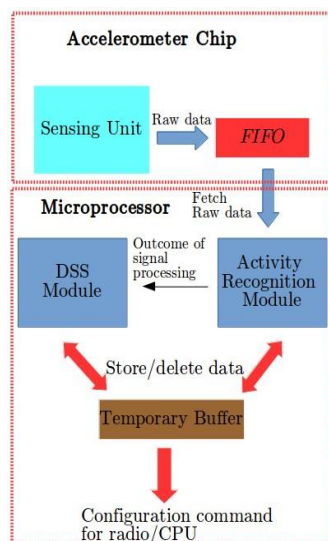


Fig. 1. Architecture of the proposed approach.

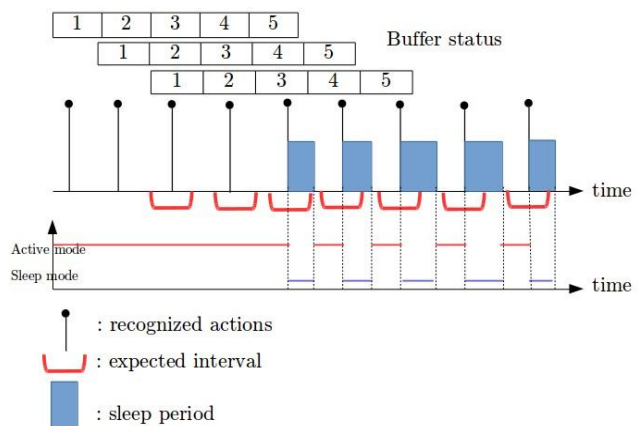


Fig. 2. Process of DSS mechanism.

### 3.2. Dynamic Sleep Scheduling Parameters

DSS mechanism has four main parameters that should be initialized at the beginning of the system. These parameters are the following:

1) length of FIFO: In this study, as an improvement of our previous works, memory located in the sensing chip is in use. It acts as FIFO. Every captured raw data is put into sensor register but also put into FIFO (needs to be configured at the beginning). MPU fetches raw data from either sensor register or FIFO (also needs to be configured). FIFO allows MPU to process the latest samples while it was in a sleep mode.

Number of samples that can be fitted in FIFO:  $s$  is given in Equation (1):

$$l = (m/8) \times a$$

$$s = (N/l) = [N / (m/8)] \times a = (N \times 8) / (m \times a) \quad (1)$$

where  $N$  signifies the length of FIFO (in terms of bytes),  $m$  represents m-bit accelerometer measurements (ex:16 bits),  $a$  is the number of axes of the accelerometer signal (3 axes),  $l$  represents length of one sampling and  $s$  is the number of sampling that can be fitted into FIFO.

2) Length of Buffer ( $\beta$ ): Buffer is a temporary memory allocated by MPU for the purpose of storing recognized activities. In order to calculate the activity frequency and also start-up of DSS, system should store previous recognized activities and their timestamps values. This buffer can contain information of same type of activities. If the decision mechanism, that is mentioned in the previous subsection, announces that type of the captured activity is different from the previous ones, in that case, buffer will be drained in order to store data about new captured activity.

Main usage of this temporary memory is to store information about same activity so that the MPU can calculate Activity Frequency. Calculation of Activity Frequency is given in Equation (2):

$$\text{Activity Frequency (AF)} = \frac{\sum_{i=1}^{n-1} (ta_{i+1} - ta_i)}{n-1} \quad (2)$$

where  $ta_i$  represents the time value of action  $i$  ( $i$  th item of the buffer),  $n$  is the length of the buffer. Average of time differences between the captured activities is calculated with the given equation.

3) Alpha( $\alpha$ ): Alpha ( $\alpha$ ) is a coefficient to calculate the sleep duration. The relation between the coefficient alpha and the sleep duration is given in Equation (3). When the buffer becomes full, DSS is enabled. With the proposed approach, MPU is able to estimate the next possible action time with calculated activity frequency. With the defined alpha value, sleep duration is determined and radio module and/or MPU are put in sleep mode for a calculated time. Goal of this sleep duration is to consume less energy while waiting the next action instead of consuming energy in an idle state. For instance, suppose that the activity frequency is calculated 2 seconds by the MPU. That means the sensor device expects next possible action 2 seconds later. If coefficient alpha is set to 0.5, it means that sensor device will be in sleep mode for 1 sec, if alpha is equal to 0.7, then  $ta$  sleep duration will be 1.4 sec. Thus, instead of consuming energy in an idle state for 2 sec, energy saving can be realized by DSS.

$$\text{Sleep Duration} = \alpha \times \text{Activity Frequency} \quad (3)$$

4) Tolerance ( $\gamma$ ): The last parameter of the DSS is tolerance value. While taking decision of continuation of the same activity or not, coefficient tolerance has a big impact. As the MPU calculates the activity frequency of the activity, expected time for the next possible action becomes into an interval of time when the tolerance value is involved into the calculation. This interval is time interval. For the MPU, next action of the same activity is expected to occur in that interval, otherwise it will be treated as a new type of activity which will trigger reset procedure that disables sleep scheduling mechanism and drains the buffer. Calculation of this time interval is given in Equation (4):

$$\text{estimated time interval} = \text{now} + [\text{min}; \text{max}]$$

$$\text{min} = \text{Activity Frequency} \times (1 - \gamma)$$

$$\text{max} = \text{Activity Frequency} \times (1 + \gamma)$$

$$(ta_{i+1} - ta_i) \in [\text{AF} \times (1 - \gamma); \text{AF} \times (1 + \gamma)] \quad (4)$$



where min and max stand for lower and upper bound of the interval.  $ta_i$  represents the last recognized action and  $ta_{i+1}$  is the next possible action (expected action). Suppose that coefficient tolerance ( $\gamma$ ) is set to 20% and the activity frequency is calculated 1sec, in this case, time interval for the estimation of the next action (same type of activity) becomes [0.8; 1.2]. Thus, time difference between the next possible action  $ta_{i+1}$  and the last captured action  $ta_i$  is :  $ta_{i+1} - ta_i \in [0.8; 1.2]$ .

Visual representation of the DSS is given in Fig. 2 (where length of buffer is set to 40 bytes which is equivalent to storage of 5 actions). There are many reasons for recognizing the next activity out of the expected time interval. It is not always because of a change of activity type. Chosen coefficients alpha and tolerance may cause this problem as well. For example, a very low tolerance value may cause an error in comparison procedure of the activities or an coefficient alpha which is chosen quite high may cause a long sleep duration which may result with missing an action. For the sleep duration case, usage of FIFO may correct that error.

#### 4. Experiments

In this study, for the test phase of the proposed approach, pedometer application is chosen. Pedometer is a device, usually portable and electronic or electromechanical, that counts each step a person takes by detecting the motion of the person's hips or legs. Pedometer devices are used originally by sports and physical fitness enthusiasts and they are now becoming popular as an everyday exercise measures and motivator. Pedometers can be a motivation tool for people wanting to increase their physical activity as well. Pedometers have been shown in clinical studies to increase physical activity, and reduce blood pressure levels. With the accelerometer signal, it is possible to detect each step done by foot left or right although the signal coming from the ankle which holds the sensor device is stronger and more characteristic. Steps are being detected by the embedded algorithm run on the MPU. In this study, dataset of WISDM laboratory ("Wireless Sensor Data Mining") from Fordham University is chosen [9]. With a mobile program written for Android based mobile devices, activities of the users are captured and analyzed. Proposed approach (DSS +FIFO) is tested on that dataset. Details of this dataset is given in Table I. During the experiment phase of the study, influences of DSS parameters on application performance and on the energy gain are examined. For the energy consumption, related equation given in Equation (5) is used. Total amount of energy consumption of the sensor device is sum of consumption of each component  $\epsilon_i$ .

$$\begin{aligned}\epsilon_{\text{totalconso}} &= \sum_{i=1}^n \epsilon_{\text{component}_i} \\ \epsilon_{\text{conso}} &= \epsilon_{\text{active}} + \epsilon_{\text{sleep}} \\ \epsilon_{\text{active/sleep}} &= V_{\text{DrainSupply}} \times I_{\text{activeMode/sleepmode}} \times t_{\text{active/sleepmode}}\end{aligned}\tag{5}$$

where  $\epsilon_{\text{active}}$ ,  $\epsilon_{\text{sleep}}$  are respectively the energy consumption while sensor device is in active mode and sleep mode.  $V$  and  $I$  stand for the voltage and the current. Finally  $t$  shows the duration for the relevant mode.

Table 1. Details of the Dataset

Recognized Activities	Walking, Jogging, Ascending stairs, Descending stairs, Sitting, Standing
Number of subjects/ total number of samples	33/ 1,098,207
Distribution of raw data	Walking 424,400 (38.6%) Jogging 342,177 (31.2%) Ascending stairs 122,869 (11.2%) Descending stairs 100,427 (9.1%)

	Sitting 59,939 (5.5%) Standing 48,395 (4.4%)
Sensor device in use/location of device	Android Mobile Device/Subjects' front pants leg pocket
Processed signal/Sampling frequency	3 axe accelerometer signal/50Hz
Tagging raw data	[subjectID],[ActivityTag],[timestamp],[x-accel],[y-accel],[z-accel]

## 5. Experiment Results

As mentioned in the DSS parameters, there are four main parameters: length of FIFO, length of buffer ( $\beta$ ), coefficient alpha ( $\alpha$ ), coefficient tolerance ( $\gamma$ ).

Buffer is for storage of the same type of activity. It contains main actions of the current activity like step time information with its accelerometer signal data ([timestamp], [x- ccel], y- accel], [z-accel]). The length of this temporary memory defines how many steps will be stored in this buffer. If the length of buffer is chosen to small, calculation of ( $\beta$ ) may not be correct since small length of buffer means less samples stored. On the other hand, if the length of the buffer is chosen too long, it will delay DSS mechanism which is the main goal of the system. According to the several preliminary tests, a temporary memory with 40 bytes is optimal for walking activity. A buffer with 40 bytes may contain 5 samples according to the Equation 1 (variables m and a).

The experiments are done on the chosen dataset. As the majority of the dataset consists of walking data (38%), proposed approach is applied on walking data. Main idea behind these experiments is to analyze the impact of the usage of FIFO on the energy optimization and at the same time evolution of application performance with DSS.

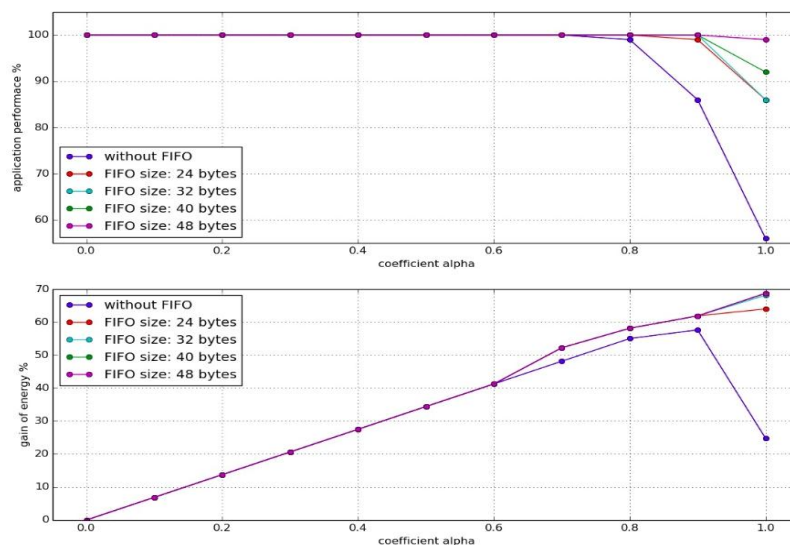


Fig. 3. Variation of application performance and the energy gain with different length of FIFO and coefficient alpha.

As mentioned before, coefficient alpha defines the sleep duration. Propose algorithm is able to calculate the activity frequency by using stored data in the buffer and make an estimation on the possible next action of the same activity. This estimation combined with the coefficient tolerance results with a creation of a time interval that the MPU expects the next action to be occurred. When the buffer becomes full, DSS mechanism is turned on. In that case, sleep duration is calculated according to the Equation (3). The coefficient alpha decides how long the sensor device will be in sleep mode while waiting the next action. Upper graph of Fig. 3 given below presents the variation of application performance while increasing



coefficient alpha for different length of FIFO (24, 32, 40, 48 bytes) (buffer length is set to 40 bytes and coefficient tolerance is set to 0.2 which means 20% tolerance degree).

According to the obtained results, for all lengths of FIFO, while the coefficient alpha  $\alpha \in [0; 0.7]$ , application performance is 100%. When the coefficient alpha is set to 0.8, in that case system starts not to recognize the action (for instance missing a step for pedometer application). With the usage of FIFO, it is possible to process the latest samples while the MPU was in a sleep mode. When we increase the coefficient alpha, system misses action and without usage of FIFO, application performance decreases. This degradation can be fixed by the FIFO which gives a chance to MPU to process latest data. For the case where coefficient alpha is equal to 1.0, application performance become 56%. FIFO makes the system to perform better and raise up the application performance. Application performance becomes 99% if the system uses a FIFO with a length 48 bytes.

Bottom graph of Fig. 3 shows the energy gain while increasing the coefficient alpha. Again the length of buffer is set to 40 bytes and coefficient tolerance is set to 0.2. Setting coefficient alpha to 0 means that there won't be an sleep duration even if the DSS is on. Thus, it is impossible to talk about the energy gain (0%). When the coefficient alpha increases, MPU is put into sleep mode for a given duration. As a connection to the previous figure, when the system misses an action, DSS mechanism is turned off and the buffer is drained. This causes a decrease in terms of application performance but also energy save. That's explains the decrease of energy gain when the coefficient alpha approaches to 1.0 (when the FIFO is not in use). Fig. 4 and Fig. 5 present the the triples (fifo length-alpha coefficient-application performance and energy gain). In these graphs, the focus of our attention is more visible. From these plots, it is shown that it exists a region where application performance and energy gain are suitable.

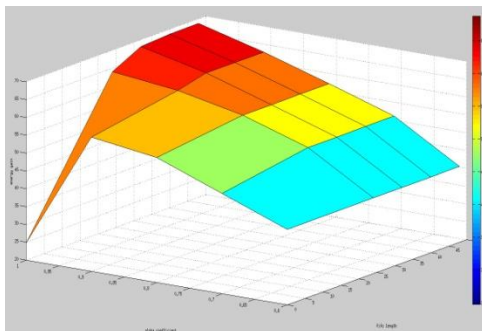


Fig. 4.  $f(\text{fifo length}, \alpha) = \text{application performance (\%)}$ .

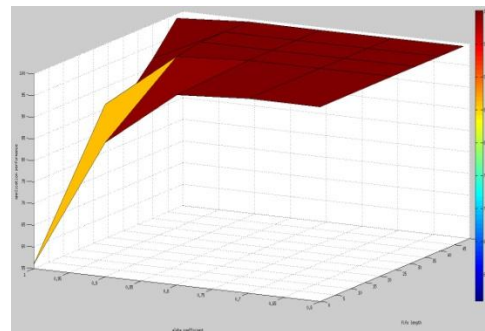


Fig. 5.  $f(\text{fifo length}, \alpha) = \text{energy gain (\%)}$ .

## 6. Discussion and Future Work

In this paper, there are two main approaches distinct from other studies aimed to save energy on the wireless sensor devices. Firstly a dynamic sleep scheduling is proposed. This approach includes a holistic approach which takes component and application level into account. Thus, this sleep scheduling mechanism is not specific to certain application. According to the application and especially activity frequency, sleep scheduling mechanism can configure itself and calculate sleep duration dynamically. With this approach, suitable region is found out where application can work with tolerable performance degradation meanwhile lifetime of the sensor is extended by decreasing energy consumption.

Secondly, memory component located on the sensor chip is involved in the proposed approach. While applying dynamic sleep scheduling mechanism, with the usage of memory that acts as FIFO, it is possible to process latest sampled data while MPU and radio are in sleep mode. Proposed approach is tested on pedometer application based on accelerometer signal. The experiments showed that it exists a region where application performance and energy gain are suitable, also impact of FIFO on application performance and energy gain. As a future work, this system will be tested on more complex and various

dataset in order to observe the system performance.

## Acknowledgment

This work is supported by Galatasaray University Research Foundation under the Grant No. 12.401.002..

## References

- [1] Obaidat, M. S., Anpalagan, A., & Woungang, I. (2012). *Handbook of Green Information and Communication Systems*. Academic Press.
- [2] Chu, D., Deshpande, A., Hellerstein, J. M., & Hong, W. (2006). Approximate data collection in sensor networks using probabilistic models. *Proceedings of the IEEE 22nd International Conference on Data Engineering* (pp. 48–48).
- [3] Deshpande, A., Guestrin, C., Madden, S. R., Hellerstein, J. M., & Hong W. (2004). Model-driven data acquisition in sensor networks. *Proceedings of the Thirtieth international conference on Very large Data Bases: Vol. 30*. (pp. 588–599).
- [4] Gaggioli, A. (2005). Optimal experience in ambient intelligence. *Ambient Intelligence*, 35–43.
- [5] Anastasi, G., Conti, M., Francesco, M. D., & Passarella, A. (2009). Energy conservation in wireless sensor networks: A survey. *Ad hoc Networks*, 7(3), 537–568.
- [6] Ganesan, D., Cerpa, A., Ye, W., Yu, Y., Zhao, J., & Estrin, D. (2004). Networking issues in wireless sensor networks. *Journal of Parallel and Distributed Computing*, 64(7), 799–814.
- [7] Demirkol, I., Ersoy, C., & Alagoz, F. (2006). Mac protocols for wireless sensor networks: A survey. *IEEE Communications Magazine*, 44(4), 115–121.
- [8] Kho, J., Rogers, A., & Jennings, N. R. (2009). Decentralized control of adaptive sampling in wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 5(3), 19.
- [9] Kwapisz, J. R., Weiss, G. M., & Moore, S. A. (2011). Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2), 74–82.



**Ozgun Pinarer** received his B.S. and M.S. degrees in Computer Engineering from Galatasaray University in 2010 and 2012, respectively. Currently, he is a research assistant in the Computer Engineering Department of Galatasaray University and he is pursuing his Ph.D. degree at Institut National des Sciences Appliquées de Lyon (INSA de Lyon) in Laboratoire d'Informatique en Images et Systèmes d'Information (LIRIS). His research interests include wireless sensor networks, wearable sensors, mobile and ubiquitous computing, smart building systems and ambient intelligence.



**Atay Ozgovde** received the BS and MS degrees from Bogazici University, Istanbul, in 1995 and 1998, respectively. He worked for Nortel Networks as an R&D engineer in various telecommunications projects between 1998 and 2001. In 2002, he started working as a research assistant in the Computer Engineering Department, Bogazici University, where he completed the PhD degree in the NETLAB research group in 2009. He worked as a researcher at the TAM research center to complete his postdoctoral research where he worked with the WiSE-Ambient Intelligence Group. Currently, he is an assistant professor in the Computer Engineering Department, Galatasaray University. His research interests include wireless sensor networks, embedded systems, distributed systems, pervasive computing, and complex networks. He is a member of the IEEE.