

PEAR: Power Efficiency through Activity Recognition (for ECG-based Sensing)

Feng-Tso Sun*, Cynthia Kuo*[†] and Martin Griss*

*Carnegie Mellon University

{lucas.sun, martin.griss}@sv.cmu.edu

[†]Nokia Research Center

cynthia.kuo@nokia.com

Abstract—The PEAR (Power Efficiency through Activity Recognition) framework is presented using an ECG-based body sensor network as a case study. PEAR addresses real-world challenges in continuously monitoring physiological signals.

PEAR leverages a wearable sensor's embedded processing power to conserve energy resources. This is accomplished by performing some data processing on the sensor and reducing the overhead of wireless data transmission. A coarse-grained decision tree-based activity classifier was implemented on a sensor node to recognize the sensor wearer's activity level. Using the wearer's activity level, the sensor dynamically manages its activities – sampling of the ECG sensor, processing of the data, and wireless transmission – to minimize overall power consumption.

This paper describes the design and implementation of RR interval extraction and activity recognition modules on a SHIMMER sensor node. An activity-aware energy model is presented along with energy profiling results. The level of energy conservation varies with a wearer's level of activity, and a sensitivity analysis shows that PEAR's advantage over standard body sensor network architectures increases with more activity. In a user study, our participants were active 18%-28% of the time. Based on this level of activity, our implementation of PEAR increases battery life up to 2.5 times when compared to conventional ECG sensing approaches.

This approach is applicable to a broad range of pervasive health applications that incorporate continuous monitoring of physiological signals.

I. INTRODUCTION

Small, wireless sensors are now widely available to monitor many physiological markers. Continuous monitoring of physiological signals promises to improve traditional healthcare in many ways. For example, gathering longitudinal data offers health care providers a more complete understanding of their patients' health. Real-time data streams from pervasive healthcare systems can also help providers make more timely diagnoses. For patients, data from pervasive systems can be used to provide feedback, both to monitor their own health and to help them modify their behavior.

However, a major challenge to the adoption of continuous monitoring systems remains: sensor battery life is limited. A reasonable battery life is necessary for any sensor system that senses, transmits, and processes data in a real-world deployment.

Some pervasive healthcare prototypes sidestep power consumption issues by limiting pilot studies to short time windows. Those prototypes continuously transmit or store large

amounts of data for post analysis. The sensor battery may deplete within hours, particularly if physiological sensors with high data rates or wireless transceivers are used [1]. In real-world deployments, wearers would need to recharge the sensor frequently, at great inconvenience.

In this paper, we present the PEAR (Power Efficiency through Activity Recognition) framework for a wearable, ECG-based continuous monitoring system. PEAR conserves a sensor's battery power by using the wearer's activity context. Activity context is derived from an embedded accelerometer. The sensor automatically adjusts its sampling, processing, and transmission according to the wearer's activity level.

Our contributions in addressing sensor energy efficiency are twofold:

- 1) We implement the processing of raw ECG data on the sensor node itself. On-node feature extraction reduces the amount of data that is sent via wireless transmission, a significant source of energy consumption.
- 2) We also implement a coarse-grained activity classifier on the sensor node. The sensor interprets a wearer's activity level to minimize data collection, processing, and transmission when possible. This is accomplished without loss of accuracy. In contrast, many other systems waste resources by collecting, transmitting, and then filtering out unnecessary data during post-processing.

In the next section, we describe related work in the area of body sensor networks. Section III provides a brief introduction to continuous ECG monitoring. In Section IV, we overview the major components in the PEAR framework. Sections V and VI then focus on the software implementations on our sensor node. In Section VII, we compare the energy model of a typical continuous monitoring system with PEAR's activity-aware energy model. We then describe our power measurement environments and protocols in Section VIII. Energy profiling results are presented in Section IX.

II. RELATED WORK

Many ECG sensors have been integrated into wearable systems for wellness monitoring [2][3]. These systems demonstrate the feasibility of recording, transmitting, and extracting ECG features using mobile devices. Features extracted from the ECG raw data can be used in a wide variety of applications, such as emotion inference and arrhythmia detection [4][5].

However, physical movement often confounds the physiological signals. Some systems transit raw physiological data from the sensor and then filter out the confounded signals during offline analysis [6]. In contrast, our sensor does not transmit data during periods of high physical activity, reducing the collection and processing of unnecessary data.

Using machine learning techniques on accelerometer data for characterizing physical activities is well-known [7][8][9]. These studies use body-worn or mobile phone accelerometer data to distinguish between several postures or activities, usually with high accuracy. Most of these activity inference algorithms require the computational power of a PC or mobile phone. Rather than transmit data to a mobile device for processing, we perform activity classification on the sensor node itself, using a real-time, coarse-grained algorithm suited for the node’s limited computational and power resources.

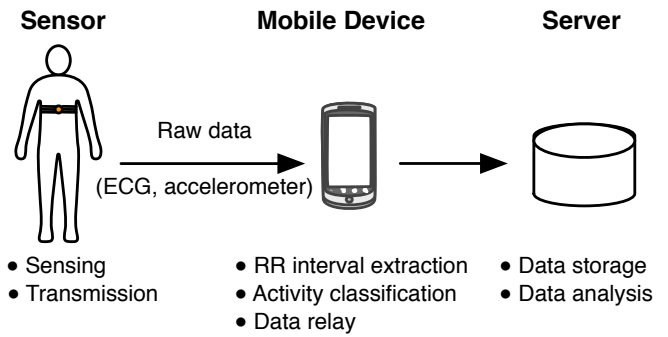
Several techniques for optimizing energy consumption in body sensor networks have been explored. For example, Lorincz et al. designed a wearable sensor network system with eight sensors and a base station (i.e., a PC), where data is selectively transmitted based on the energy consumption of each node and radio link quality [10]. Au et al. focus on energy optimization in wearable sensor-based activity recognition. They additively increase the interval between activity classifications when the classification result remains the same and multiplicatively decrease the interval when the predicted activity changes, resulting in a 7x increase in battery life [11]. The activity classification and control algorithms of both systems are executed on more powerful computing devices, once data has been transmitted from the sensor.

However, evidence suggests that these optimization techniques may be insufficient. Mathur et al. concluded that data processing costs orders of magnitude less energy than radio transmission in a sensor network [12]. More specifically, Albu et al. demonstrated that processing ECG data on a sensor node and transmitting only the extracted features reduces the amount of transmitted data by a factor of 72 [13]. We avoid transmitting unnecessary data by performing feature extraction, in addition to activity classification, on the sensor node. To the best of our knowledge, this is the first implementation of a wearable, continuous ECG monitoring system that processes raw data to reduce transmission overhead and uses the wearer’s activity level to manage data collection, processing, and transmission.

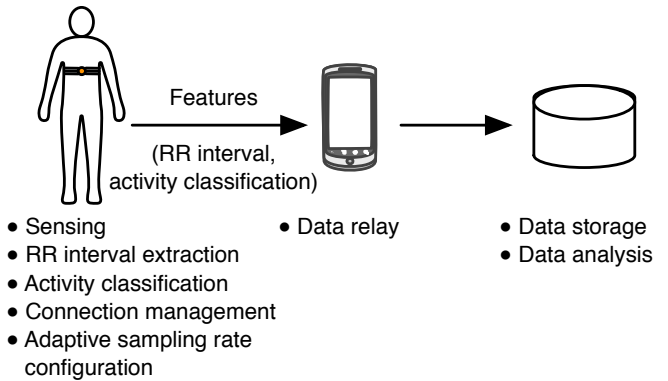
III. CONTINUOUS ECG MONITORING

ECG signal analysis is a common and non-obtrusive method for evaluating cardiac activity. Our system design is based on the insight that ECG-based applications share two properties.

- 1) The basic building block for ECG data analysis is an RR interval. The RR interval represents the time between two heart beats. It is extracted by measuring the time between two “R” peaks in an ECG signal. RR intervals are used to determine heart rate variability (HRV), as well as certain indicators of arrhythmia [14][15].



(a) A typical continuous ECG monitoring system



(b) PEAR system overview

Fig. 1: A comparison of typical framework of continuous ECG monitoring system (a) and PEAR system overview (b).

- 2) Cardiac activity is highly affected by physical activity. Some ECG-based applications, such as those which measure stress or emotional states, need to separate the effects of physical activity from the effects of other stimuli. In these applications, physical activity is a confounding factor.

Figure 1(a) shows a “typical” continuous ECG monitoring system consisting of a wearable, wireless ECG sensor; a mobile device; and a back-end system with a database [16][3]. The on-body ECG sensor measures the electrical activity of the heart at a high sampling rate and transmits the raw, unprocessed sensor data to the mobile device. Temporary data storage and real-time ECG feature extraction are usually handled by the mobile device. The back-end server is generally responsible for long-term data storage and detailed analysis.

In this typical setup, a sensor consumes a large portion of its battery capacity while transmitting raw ECG data to the mobile device. Also, it is wasting energy while collecting and transmitting data that will eventually be filtered out during processing. This data collection scheme can be optimized for real-world deployments.

IV. PEAR SYSTEM OVERVIEW

An overview of a PEAR system is illustrated in Figure 1(b). Like the typical system in Figure 1(a), there are three

components: a wearable sensor, a mobile device, and a back-end system with a database. However, the PEAR system is different from the typical ECG system in two ways. First, the sensor is responsible for processing the raw ECG data and transmitting the extracted RR intervals to the mobile device. Second, the sensor uses accelerometer data to estimate the wearer’s activity level. Activity information is used to (1) manage ECG data collection and processing and (2) activate or deactivate the sensor’s wireless radio.

PEAR has been implemented on the SHIMMER platform, a small wireless sensor designed to support wearable applications. The SHIMMER base board consists of a microprocessor up to 8MHz (with 10kB RAM and 48kB Flash), micro SD card slot, Bluetooth radio module, 3-axis accelerometer, and 280mAh battery [17]. We added an ECG daughter board to the base board for ECG signal acquisition. SHIMMER Rev 2 has an additional circuitry to measure scaled battery voltage and regulator output voltage. The difference between the scaled battery voltage and the regulator output voltage allows us to calculate the instantaneous current. The SHIMMER platform runs TinyOS and is programmed in nesC.

An Android-based mobile phone, a Google Nexus One, serves as a hub in our PEAR system. The application on the mobile phone performs two major tasks. First, it maintains a service that handles Bluetooth connections initiated by the sensor. Once a connection is established, the application then decodes packets of sensor data and stores the information in a local SQLite database. In order to monitor the phone’s battery capacity, we register an Android *intent* receiver. The receiver listens for a broadcast that is triggered when the battery capacity is changed. The phone then transmits the data to a remote server for storage and analysis.

The server accepts uploaded data through a Java servlet running on Apache Tomcat. The data is stored in a PostgreSQL database for deeper analysis and aggregation.

A. Sensor Software Implementation

Figure 2 outlines the activity-aware energy optimization scheme that we implemented on the sensor node. At all times, accelerometer data is collected and input into an activity classifier. The classifier categorizes the sensor wearer’s activity level into “active” or “inactive.” Stationary wearers, such as those who are sitting or standing, are labeled inactive. Moving wearers, such as those who are walking, climbing stairs, or running, are deemed active. If a sensor’s activity level is declared active (and the previous activity level was inactive), the Bluetooth radio is switched off, and all ECG sampling, processing, and transmission ceases. The sensor continues to monitor the wearer’s activity level by sampling the accelerometer. If the sensor’s activity level is declared inactive (and the previous activity level was active), the Bluetooth radio is switched on. In addition, ECG sampling, feature extraction, and Bluetooth transmission resume.

Note that the sensor node performs activity recognition in order to manage the Bluetooth radio. Radio management is accomplished by assigning the role of master in the Bluetooth

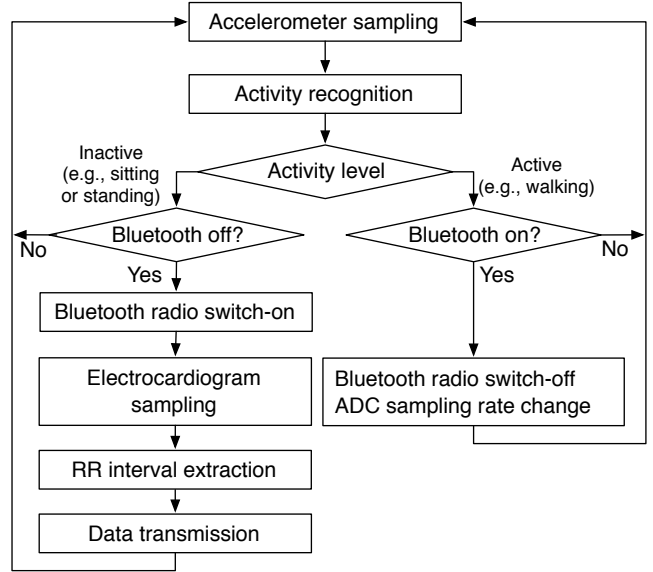


Fig. 2: Flowchart of the activity-aware energy optimization scheme implemented on the sensor.

piconet to the sensor. The mobile device joins the piconet as a slave. This is the reverse of most systems, where wireless sensors are configured to be slaves and the mobile device assumes the role of master. When the sensor acts as the master, however, it does not need to waste extra energy listening for traffic from the mobile device.

In order to utilize the wearer’s activity to control the Bluetooth radio and reduce the transmitted data, we implemented two processing modules on the sensor: A) an activity recognition module and B) an ECG feature extraction module.

V. SENSOR ACTIVITY RECOGNITION MODULE

We implemented a decision tree-based activity classifier to label wearers’ activity levels as active or inactive.

A. Initial Data Collection

Five subjects participated in the initial data collection session. Participants were each instructed to perform three activities: sitting, standing, and walking. The duration of each activity was five minutes. During data collection, the sensor sampled the accelerometer at 100Hz with $\pm 6g$ sensitivity. The data was later downsampled to lower frequencies for analysis. Based on Olguin and Pentland’s findings, the sensor was placed at the waist in order to maximize signal differences between sitting, standing, and walking [18].

B. Feature Extraction for Accelerometer Data

Researchers have designed several time and frequency domain features for accelerometer data that are suitable for distinguishing daily activities [19][9]. Due to limited computational resources on the sensor node, we focused on computing time domain features. The feature vector was calculated within

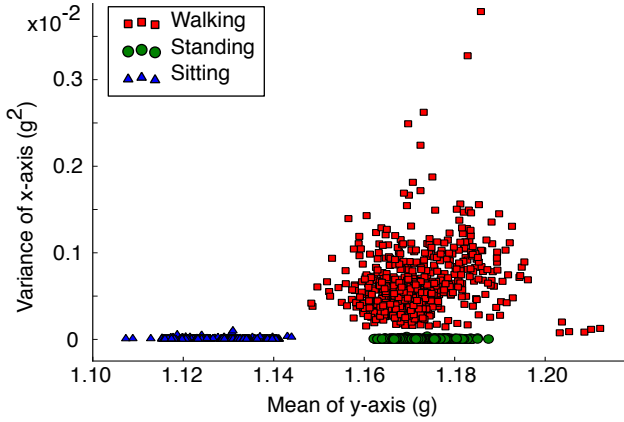


Fig. 3: Training samples for three activities types in the acceleration feature space. There are distinct clusters for each activity.

a time window of five seconds with 50% overlap. We trained the classifier on the following acceleration features:

- *Average*: The mean value of the acceleration signal was calculated for each axis in a 5 second window.
- *Variance*: The variance of the acceleration signal was calculated for each axis in a 5 second window.
- *Signal Magnitude Area (SMA)*: SMA represents the level of dynamics in acceleration signals. Hence, SMA has been regarded as a suitable feature for discriminating sedentary activities, such as sitting or standing, from moving activities, such as walking [9]. SMA is calculated using

$$SMA = \frac{1}{w} \sum_{i=1}^w |x(i)| + |y(i)| + |z(i)| \quad (1)$$

where $x(i)$, $y(i)$, and $z(i)$ are the acceleration signals along x -axis, y -axis, and z -axis, respectively.

- *Correlation between each pair of axes*: The correlation between axes can be used to differentiate the orientation of the sensor.

C. Classification Method

We implemented a lightweight classification scheme on the sensor that is appropriate for a resource-constrained environment. We used a WEKA machine learning engine to train the classifier using the J48 Decision Tree [20]. Ten-fold cross validation was used to determine the sampling frequency with optimal classification accuracy. In our analysis, the decision tree achieved 98.3% accuracy with a sampling rate as low as 5 Hz. The decision tree selected the mean adjusted variance of the x -axis and the mean value of the y -axis as the most informative features. Figure 3 plots the activity samples at 5 Hz in the feature space using these two selected features as axes. There are three clearly separated clusters corresponding to three different types of activity.

D. Classification during Real-world Use

To empirically evaluate the classification algorithm, we asked five participants to wear our sensor for a four hour

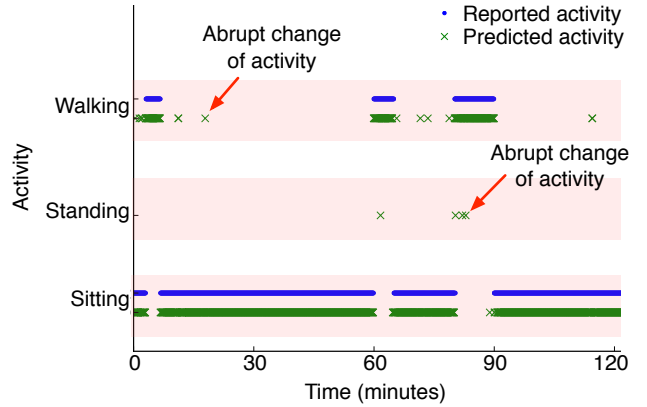


Fig. 4: One participant's self-reported and classified activities. Posture changes and intermittent stops while walking are marked as abrupt changes of activity.

user study. The participants were graduate students whose typical activities include sitting in front of a computer, standing in a line at a restaurant, or walking to a class. An activity annotation tool was provided on a mobile phone. Participants were asked to annotate their activity when transitioning from one activity to another.

During the study, participants were active $20.4 \pm 5\%$ of the time, on average. Figure 4 shows a 120-minute sample of one participant's activity recording. There are two types of situations in which the recognized activity differs from the participant's reported activity. First, sudden posture changes of seated participants may have been misclassified as walking. Sudden posture changes include leaning forward quickly or twisting the torso. Second, intermittent stops while participants were walking are (correctly) classified as standing, but may not have been annotated by participants immediately. If we consider standing to be the correct classification for a pause while walking, the algorithm's classification achieves 91% accuracy.

VI. ECG FEATURE EXTRACTION MODULE

When wearers are inactive, the sensor performs feature extraction on contiguous 5 second windows of raw ECG data.

A. Feature Extraction for ECG Data

The RR interval extraction algorithm is based on a simplified on-node implementation [13] of the Pan-Tompkins QRS detection algorithm [21].

The algorithm is divided into two phases: 1) noise filtering and 2) peak detection. During noise filtering, the potential "R" peaks are enhanced and the background baseline drift is attenuated by performing a bandpass filter on the raw ECG samples. The three-tap bandpass filter is formulated as

$$y(n) = x(n) - 2x(n-1) + x(n-2). \quad (2)$$

Figure 5(a) shows a sample raw ECG signal. Figure 5(b) shows the signal after it has passed through the bandpass filter.

Next, we detect the potential R peaks by calculating the first derivative values of the bandpass filtered signal using

$$fd(n) = 2y(n) + y(n - 2) - y(n - 3) - 2y(n - 4), \quad (3)$$

as pictured in Figure 5(c). In order to remove the negative derivative values, the first derivative signals are squared, producing the exaggerated signal in Figure 5(d).

The algorithm then calculates a moving average over the squared derivative for every 5 samples (i.e., a 50 ms window). The moving window integration computes the smoothed result shown in Figure 5(e). The result includes the location of the QRS candidates and attenuates the random peaks.

Finally, we identify the R peaks in the smoothed signals. We set a minimum distance between R peaks at 300 ms, since RR intervals are almost always longer than 300 ms when wearers are stationary. (An RR interval of 300 ms is equivalent to a heart rate of 200 beats per minute.) We also set an adaptive threshold of half the maximum signal amplitude in a 5 second window. Each sample value greater than the threshold and more than 300 ms from the last peak is recognized as an R peak.

To validate the accuracy of our implementation, we recorded 10 minutes of raw ECG data while sitting and standing participants wore the Shimmer ECG sensor. We ran the recorded raw ECG data through the sensor’s RR interval extraction implementation. The on-node implementation achieves 96.3% precision compared to the result of a Pan-Tompkins implementation in Matlab [21] as the ground truth.

B. Packet Size Reduction

Sending RR intervals instead of raw ECG data significantly reduces the amount of data transmitted from the sensor.

One raw data packet is associated with one sample of raw ECG and accelerometer reading. The raw data packet format contains 22 bytes, including 2 bytes of raw ECG data, 2 bytes of raw accelerometer data, and 18 bytes of packet overhead (e.g., header, timestamp). For a given 5 second window, the sensor generates 11,000 bytes of data that would need to be wirelessly transmitted to a more powerful device in a typical system.

In comparison, performing RR interval extraction on the sensor reduces the amount of data by roughly two orders of magnitude. The exact size of a feature packet depends on the number of RR intervals in a window. For example, suppose a wearer’s heart rate is 120 bpm. There are 10 RR intervals for every 5 second window. One RR interval is represented in 2 bytes. Thus, the feature packet for this window contains 18 bytes of packet overhead, plus 10 RR intervals \cdot 2 bytes yields a packet totaling 38 bytes.

The reduction in data size is more than 99.5% in 5 second window. The data transmission savings are slightly offset by the resources required to process the data on the sensor. However, in the next section, we show that the amount of energy required for feature extraction on the sensor is still 64% less than the amount of energy that would be consumed during wireless transmission of the raw ECG data.

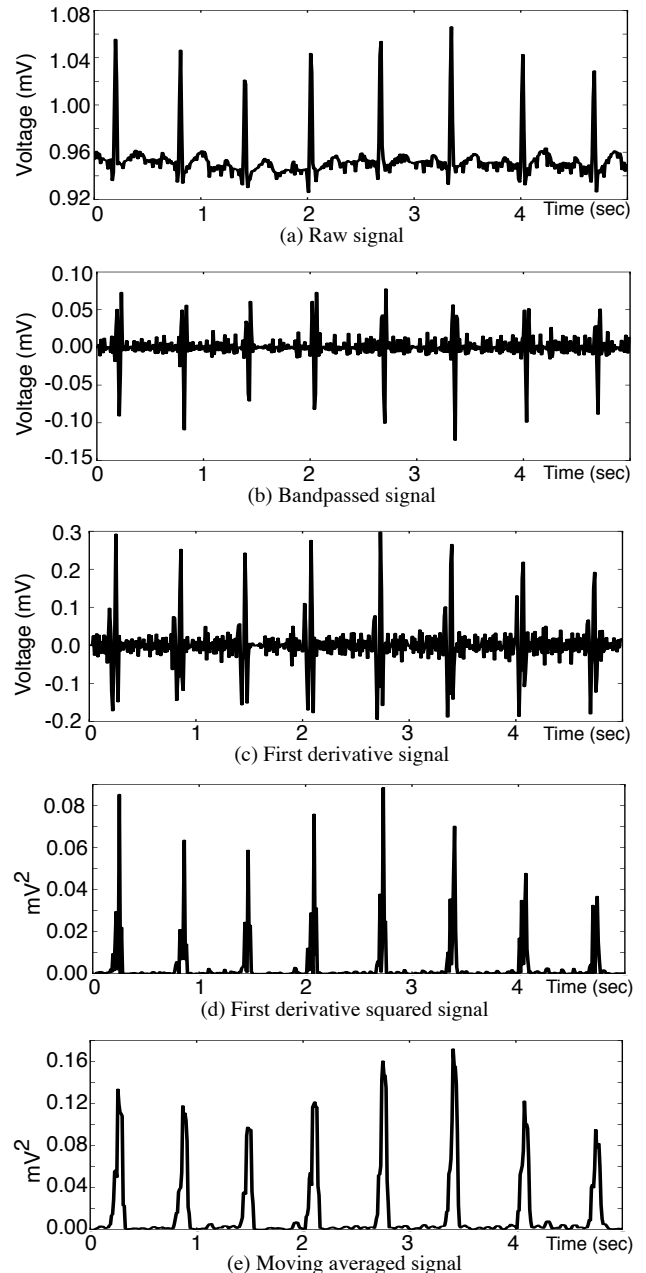


Fig. 5: RR interval feature extraction process. (a) shows the initial raw ECG signal. (b)-(e) illustrate the result of each processing step.

VII. ENERGY MODEL

To evaluate PEAR, we modeled the sensor’s energy consumption and created an energy profile for its activities. Below, we compare PEAR to the typical body sensor network shown in Figure 1(a).

A. Energy Model for a Typical Continuous Monitoring System

In many body sensor networks, sensor nodes are responsible for measuring physiological signals and directly transmitting the raw data to a more powerful computing device, such as a mobile device or a base station. Two basic operations

consume battery power in this model: sensor sampling and radio transmission.

The amount of energy consumed by sensor sampling depends on the number of the actuators, sampling frequency, and the duty cycle configured on the sensor node.

The amount of energy consumed by radio transmission often dominates power consumption in the sensor node [12]. Radio transmission can be broken down into two categories: first, the energy when the radio module is powered on and listening for traffic, but no data are in transit; and second, the energy when the radio module is powered on and transmitting sensor data.

B. PEAR Energy Model

In comparison, the energy model for PEAR is slightly more complex. There are two possible states based on the activity classifier: active and inactive. In each state, the sensor is engaged in a subset of the following activities: sampling data, processing data, storing data in the local flash memory, or wirelessly transmitting the processed data.

This energy model includes two additional components over the previous model. The first is the storage operation, a relatively minor addition with respect to energy consumption. Factors that affect energy consumption during flash memory operations are the size of the data and how frequently data are stored or retrieved. The second component is data processing. The microprocessor on the sensor performs two non-trivial processing tasks, activity recognition and RR interval detection. Data processing is sensitive to the complexity of the microprocessor code and how often the application requests a computational result.

Thus, the energy model for the PEAR is expressed as:

$$E_{overall} = NumSegments_{active} \cdot E_{active} + NumSegments_{inactive} \cdot E_{inactive} \quad (4)$$

The overall energy consumption $E_{overall}$ is the sum of the sensor's energy consumption with respect to the wearer's activity level. The terms $NumSegments_{active}$ and $NumSegments_{inactive}$ represent the number of *segments* in each activity level, where each segment consists of data collected over a 5 second sampling interval.

Equation 5 shows how the energy model accounts for sampling, processing, and storage when the wearer is active.

$$E_{active} = T_{segment} \cdot (P_{sample} \cdot f_{sample} + P_{process} \cdot f_{process} + P_{store} \cdot f_{store}) \quad (5)$$

We use $T_{segment}$ (where $T_{segment} = 5$ seconds) to denote the time interval spanned by a segment. P is the amount of power consumed by a specific operation, such as sampling or processing. The corresponding term f is the fraction within interval $T_{segment}$ in which the specific operation occurs.

Equation 6 shows the energy model for an inactive wearer. The energy cost of local data storage is replaced with the cost of wireless transmission.

$$E_{inactive} = T_{segment} \cdot (P_{sample} \cdot f_{sample} + P_{process} \cdot f_{process} + P_{transmit} \cdot f_{transmit}) \quad (6)$$

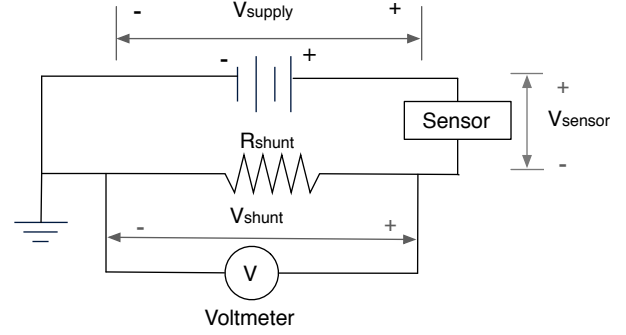


Fig. 6: Shunt circuit used for energy profiling.

On one hand, there are additional processes that draw on the sensor's battery power. On the other hand, the wireless radio is used less frequently and transmits less data. By replacing expensive processes with less expensive ones, we expect to see an increase in the sensor's battery life. The model predicts that this gain should be more pronounced as wearers' activity levels increase.

In the next section, we describe our energy profiling setup. This is followed by quantitative results that compare the energy profiles of the PEAR to the standard system configuration.

VIII. ENERGY PROFILING

A. Setup

We adopted a common shunt circuit, illustrated in Figure 6, to measure the energy consumption of each major operation on the sensor node. During runtime, we measure the voltage drop V_{shunt} across the shunt resistor R_{shunt} with a Lecroy WaveSurfer 44Xs-A oscilloscope. R_{shunt} has a resistance of 4.7Ω and a tolerance of 2%. The oscilloscope samples the voltage drop V_{shunt} at 500 samples per second.

We calculate the value of the instantaneous current drawn by the sensor based on Ohm's law. Then, we apply Joule's law (as shown in Equation 7) to calculate the power dissipated by the sensor node.

$$P_{sensor} = V_{sensor} \cdot I_{shunt} = (V_{supply} - V_{shunt}) \cdot (V_{shunt}/R_{shunt}) \quad (7)$$

B. Profile

Using the profiling setup described above, we ran each of the following eight operations for five minutes. The power consumption is derived from the supplied voltage and the current draw averaged over five minutes.

- *ADC sampling (5Hz)*: The accelerometer sensor is polled at a 5Hz Analog-to-Digital-Conversion (ADC) sampling rate. The ECG sensor is not sampled in this ADC sampling configuration. During the sampling test, the Bluetooth radio and flash read/write functionalities are disabled.
- *ADC sampling (100Hz)*: The ECG sensor is polled at a 100Hz Analog-to-Digital-Conversion (ADC) sampling

rate. Note that even when the ADC sampling rate is 100Hz, we downsample accelerometer data to 5Hz for activity recognition. During the sampling test, the Bluetooth radio and flash read/write functionalities are disabled.

- *Activity recognition*: The sensor computes the accelerometer’s time domain features and traverses the decision tree to classify the wearer’s activity level. The activity classifier continuously operates over 5 second intervals of buffered raw data.
- *RR interval extraction*: The sensor continuously extracts the RR intervals from ECG data over 5 second windows.
- *Flash access*: The sensor continuously writes data packets into flash memory. The size of each packet is 30 bytes, which includes 18 bytes of packet overhead and six RR intervals represented in 2 bytes.
- *Continuous transmission of raw data*: This test configuration simulates the “typical” setup. The sensor samples acceleration and ECG data at 100Hz and immediately transmits the data via Bluetooth to a mobile device. Each packet is 22 bytes, including raw accelerometer data, raw ECG data, and packet overhead.
- *Periodic transmission of extracted features*: This configuration includes sampling, processing, and transmission to simulate when the wearer’s activity level is inactive. The Bluetooth radio is enabled. The sensor samples acceleration and ECG data at 100Hz. The sensor performs RR interval extraction and generates the feature packet over each five second window. The packet size is 18 bytes of overhead + $2 \cdot n$ bytes, where n is the number of RR intervals detected in a five second window.
- *Local storage of extracted activity features*: This configuration includes accelerometer sampling, processing, and storage to simulate when the wearer’s activity level is active. The Bluetooth radio is disabled. The sensor samples the accelerometer at 5 Hz over 5 second windows and immediately stores the data in local flash memory. The 24 byte packets include two extracted acceleration features, activity classification, and packet overhead.

IX. RESULTS

Figure 7 shows the measured current draw for each individual operation. Operations that include radio transmission consume more power than processing or flash access. Continuously transmitting raw accelerometer and ECG data at 100Hz draws the most current, at 9.46 mA. In comparison, transmitting extracted features every 5 seconds reduces the current draw by 64%. Even though additional data processing on the sensor consumes energy, wireless transmissions costs are significant. The overall effect is a reduction in energy consumption. The benefits of transmitting less data may be more significant when the physiological sensor’s sampling rate is higher (e.g., 250Hz or 500Hz).

Using the current draw characteristics in Figure 7, we can estimate the SHIMMER sensor’s battery life. Figure 8 shows the estimated battery life of the SHIMMER sensor node under three transmission schemes.

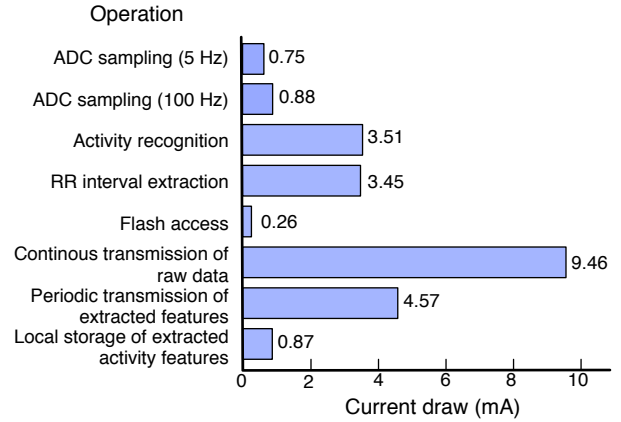


Fig. 7: Current draw characteristics for each operation.

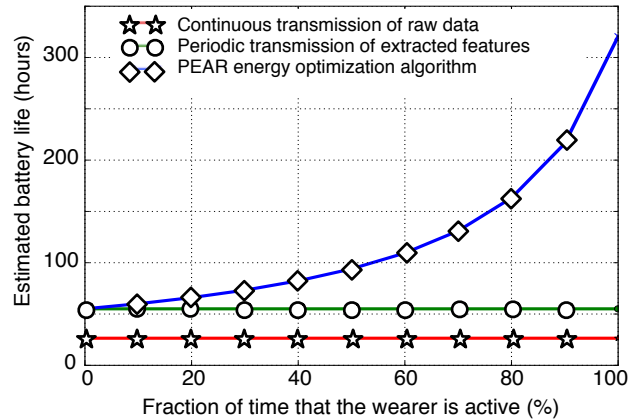


Fig. 8: Sensitivity analysis showing the estimated battery life for a SHIMMER sensor node (280mAh battery capacity), based on the fraction of time that wearers are active. In a small user study, participants were active 20.4% of the time on average.

- 1) Continuous transmission of raw data: The sensor continuously samples raw ECG and accelerometer data at 100 Hz and transmits raw signals via Bluetooth. The sensor’s 280mAh battery will last approximately 26 hours. The battery life will be shorter when the sensor is sampling at higher rate.
- 2) Periodic transmission of extracted features: The sensor continuously samples raw ECG and accelerometer data at 100 Hz, performs feature extraction on the data, and transmits the extracted features via Bluetooth every five seconds. The sensor’s battery life extends to 55 hours.
- 3) PEAR energy optimization algorithm: The sensor continuously samples raw ECG and accelerometer data at 100 Hz and performs feature extraction on the data. The sensor transmits the extracted features via Bluetooth only when the wearer is in inactive state. Otherwise, the sensor stores the data into the local flash.

With PEAR’s energy optimization scheme, the sensor’s battery life extends as the time in which the wearer is active increases. The baseline of PEAR’s estimated battery life is equal to the periodic transmission of extracted features. In

other words, the periodic transmission of extracted features is equivalent to PEAR when the wearer is perpetually inactive. The baseline condition is clearly impractical. The datasets from our user study show that participants were in active state, on average, 20.4% of the time. This would extend the sensor's battery life to approximately 66 hours, representing a 20% increase in battery life over the baseline and a 154% increase over continuous sampling and transmission.

PEAR's energy model omits the power consumed when the wireless radio is turned on and turned off. Radio transitions generate a spike in current draw, but they are short-lived and occur relatively infrequently. In our user studies, the overall time interval at that radio transition occurs is less than 0.8% based on the time of the data. In most cases, the effect of radio transitions is negligible on battery life.

X. CONCLUSION

PEAR leverages a sensor node's embedded processing power to reduce its energy consumption in a continuous monitoring application. We present an implementation of PEAR for a wearable ECG sensor. We developed an on-node, coarse-grained activity recognition module that adapts the behavior of the sensor based on the wearer's context. The sensor interprets a wearer's activity level to minimize data collection, processing, and transmission when possible. We also implemented RR interval extraction on the sensor node to minimize the overhead of wireless data transmission. In other physiological monitoring applications, RR interval extraction would be substituted with the appropriate data features.

PEAR's activity-aware body sensor network architecture extends the battery life of a sensor up to an estimated 2.5 times. The energy profile of our implementation clearly demonstrates the benefit of using on-node processing to reduce the amount of data transmitted wirelessly. Furthermore, activity information extracted from raw accelerometer data is useful for controlling the quality of data collection and strategically reducing the sensor's overall energy consumption at runtime.

Moving forward, episodic sampling has been shown to increase energy efficiency with tolerable error rate [11]. There is an opportunity to apply these principles to further reduce the footprint of our activity recognition scheme. Also, the increase in battery life is dependent on the fraction of time that participants are actively moving. Further studies will provide a more complete picture of wearers' activity levels.

The principles of PEAR are applicable to a broad range of physiological monitors, such as galvanic skin response, respiration rate, or blood pressure. Extending the battery life of sensors will enable users and health care providers to acquire continuous and reliable physiological data with minimal interruptions.

REFERENCES

[1] M. Hanson, H. Powell, A. Barth, K. Ringgenberg, B. Calhoun, J. Aylor, and J. Lach, "Body area sensor networks: Challenges and opportunities," *Computer*, vol. 42, no. 1, pp. 58–65, Jan 2009.

[2] J. Oresko, Z. Jin, J. Cheng, S. Huang, Y. Sun, H. Duschl, and A. Cheng, "A wearable smartphone-based platform for real-time cardiovascular disease detection via electrocardiogram processing," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 14, no. 3, pp. 734–740, May 2010.

[3] J. Healey and B. Logan, "Wearable wellness monitoring using ECG and accelerometer data," in *Wearable Computers, 2005. Proceedings. Ninth IEEE International Symposium on*, Oct. 2005, pp. 220–221.

[4] B. Cinaz, R. L. Marca, B. Arnrich, and G. Troster, "Towards continuous monitoring of mental load," in *Proceedings of the Int'l Conference on Ubiquitous Computing (UbiComp'10)*. ACM, Sept. 2010, pp. 1–5.

[5] R. Fensli, E. Gunnarson, and T. Gundersen, "A wearable ECG-recording system for continuous arrhythmia monitoring in a wireless tele-home-care situation," in *Computer-Based Medical Systems, 2005. Proceedings. 18th IEEE Symposium on*, 2005, pp. 407–412.

[6] J. Healey, L. Nachman, S. Subramanian, J. Shahabdeen, and M. Morris, "Out of the lab and into the fray: Towards modeling emotion in everyday life," in *Pervasive Computing*, ser. Lecture Notes in Computer Science. Heidelberg, 2010, pp. 156–173.

[7] T. Brezmes, J.-L. Gorricho, and J. Cotrina, "Activity recognition from accelerometer data on a mobile phone," in *Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living*, 2009, vol. 5518, pp. 796–799.

[8] N. Ravi, N. D. P. Mysore, and M. L. Littman, "Activity recognition from accelerometer data," in *In Proceedings of the Seventeenth Conference on Innovative Applications of Artificial Intelligence*. AAAI Press, 2005, pp. 1541–1546.

[9] G. Krassnig, D. Tautinger, C. Hofmann, T. Wittenberg, and M. Struck, "User-friendly system for recognition of activities with an accelerometer," in *Pervasive Computing Technologies for Healthcare (Pervasive-Health), 2010 4th International Conference*, Mar. 2010, pp. 1–8.

[10] K. Lorincz, B. Chen, G. W. Challen, A. R. Chowdhury, S. Patel, P. Bonato, and M. Welsh, "Mercury: a wearable sensor network platform for high-fidelity motion analysis," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '09. New York, NY, USA: ACM, 2009, pp. 183–196.

[11] L. Au, M. Batalin, T. Stathopoulos, A. Bui, and W. Kaiser, "Episodic sampling: Towards energy-efficient patient monitoring with wearable sensors," in *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, Sept. 2009, pp. 6901–6905.

[12] G. Mathur, P. Desnoyers, D. Ganesan, and P. Shenoy, "Ultra-low power data storage for sensor networks," in *Proceedings of the 5th international conference on Information processing in sensor networks*, ser. IPSN '06. New York, NY, USA: ACM, 2006, pp. 374–381.

[13] D. Albu, J. Lukkien, and R. Verhoeven, "On-node processing of ECG signals," in *Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE*, Jan. 2010, pp. 1–5.

[14] R. Barbieri, E. Matten, and E. Brown, "Instantaneous monitoring of heart rate variability," in *Engineering in Medicine and Biology Society, 2003. Proceedings of the 25th Annual International Conference of the IEEE*, vol. 1, Sept. 2003, pp. 204–207 Vol.1.

[15] B. M. Asl, S. K. Setarehdan, and M. Mohebbi, "Support vector machine-based arrhythmia classification using reduced features of heart rate variability signal," *Artif. Intell. Med.*, vol. 44, pp. 51–64, Sept. 2008.

[16] S. Korsakas, J. Lauznis, A. Vainoras, Z. Markovitch, L. Gargasas, Z. Navickas, and R. Ruseckas, "The mobile ECG and motion activity monitoring system for home care patients," in *Computers in Cardiology*, 2006, pp. 833–836.

[17] <http://www.shimmer-research.com>.

[18] D. Olguin and Y. Pentland, "Human activity recognition: Accuracy across common locations for wearable sensors," in *Proc. 10th International Symposium Wearable Computer*, 2006, pp. 11–13.

[19] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *Pervasive*, 2004, pp. 1–17.

[20] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA Data Mining Software: An Update," *SIGKDD Explorations*, vol. 11, 2009.

[21] J. Pan and W. J. Tompkins, "A real-time QRS detection algorithm," *Biomedical Engineering, IEEE Transactions on*, vol. BME-32, no. 3, pp. 230–236, Mar. 1985.