Optimizing Operations per Joule for Energy Harvesting IoT Devices

Jaehyun Park[1], Ganapati Bhat[2], Cemil S. Geyik[2], Hyung Gyu Lee[3], Umit Y. Ogras[2]

[1]School of EE, Ulsan, Korea, jaehyun@ulsan.ac.kr
[2]School of ECEE, Arizona State University, Tempe, AZ, {gmbhat, cemil.geyik, umit}@asu.edu
[3]School of CCE, Daegu University, Gyeongsan-si, Korea, hglee@daegu.ac.kr

March 1, 2019

***Abstract*— Small form-factor and low-cost wearable devices enable a variety of applications including gesture recognition, health monitoring and activity tracking. Since these devices are severely constrained by battery capacity, energy harvesting, and optimal use of the harvested energy are critical to make them practical. This paper considers optimal gesture recognition using self-powered devices. Given a harvested energy budget, we propose an approach to maximize the number of gestures that can be recognized under accuracy constraints.We construct a computationally efficient optimization algorithm with the help of analytical models derived using a detailed energy consumption breakdown of a wearable device prototype. Our empirical evaluations demonstrate up to 2.4× increase in the number of recognized gestures compared to a manual optimization.**

I. Introduction

Wearable internet of things (IoT) devices are becoming popular due to their small form factor and low cost [4]. Small form factor enables interesting applications including gesture-based control, health monitoring, and activity tracking [3], [5], [8], [22]. However, it also limits the battery capacity,which is one of the major obstacles for widespread adoption of wearable IoT devices [23].

Wearable devices cannot rely on high capacity batteries used in smartphones due to their relatively large size and heavy weight (2100 mAh @ 42 g) [11]. Lighter flexible batteries cannot be used alone either, since they have modest capacities (200 mAh @ 1.2 g) [9]. Therefore, harvesting energy from ambient sources is crucial to relieve from the dependence on batteries [7]. Recent research shows that photovoltaic cells (PV-cells) can provide 10–100 mW/cm^2 density [21]. Wearable devices can greatly benefit from this harvesting potential, since they can be personalized for each user. For example, the device can learn the usage patterns, and adapt the operating points to its user.

In this work, we consider wearable devices powered primarily through ambient energy sources, as illustrated in Figure 1. Since the amount of the harvested energy sets the available energy budget, *the device has to* ***maximize the work performed under this energy budget***. To this end, we employ gesture recognition as the target domain, because it has a wide range of wearable applications, such as gesture-based control and interaction with robots assistive devices. More precisely, we maximize the number of gestures dynamically under energy budget and accuracy constraints. This problem is convoluted by three major challenges. First, accurate energy consumption and gesture recognition accuracy models are needed to guide this optimization. Second, this problem should be solved at runtime with minimum implementation overhead. Finally, to be credible, the optimization methodology has to be validated using an energy harvesting device and user subject studies.

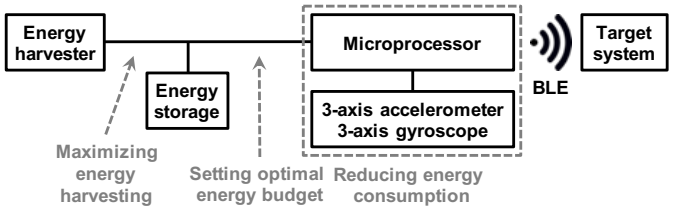


Fig. 1. Wearable gesture recognition system.

This report presents a novel methodology to maximize the number of gestures that can be recognized under energy budget and accuracy constraints. To achieve this goal, we designed the wearable device whose components are illustrated in Figure 1. It consists of an energy harvesting subsystem, a microprocessor, a 3-axis accelerometer, a 3-axis gyroscope and Bluetooth low energy (BLE) interface (detailed in Section III). Using this prototype, we characterize the power consumption of the accelerometer, microprocessor and BLE while performing gesture recognition.Then, we develop a compact energy model that can be used at runtime by the proposed optimization approach. Similarly, we analyze the recognition accuracy as a function of the gesture recognition duration by performing user studies. Finally, we present a computationally efficient algorithm to maximize the number of recognized gestures under the energy budget and accuracy constraints. We show that the proposed approach increases the number of gestures that can be recognized by up to 2.4× compared to manual optimization.

In summary, the major contributions of this report are as follows:

- A detailed energy consumption analysis for wearable gesture recognition devices and analytical models,
- An algorithm to maximize the number of recognized gestures under the given energy budget and accuracy constraints,
- Empirical evaluations using a wearable device prototype demonstrate up to 2.4× increase in the number of rec-

ognized gestures compared to manual optimization.

The rest of the report is organized as follows. We review the related work in Section II. We present the system overview and the proposed algorithm in Section III and Section IV, respectively. Finally, we discuss the experimental results in Section V, and summarize the conclusions in Section VI.

II. Related Work

Wearable IoT devices have been studied extensively due to their form factor and cost advantages. Researchers have proposed sensor networks, gesture-based control, health monitoring, and activity monitoring as potential applications of IoT devices [12], [18]. Significant amount of research has also focused on wearable devices with energy harvesting [7], [17]. For instance, a jacket with solar and thermal energy harvesting is proposed in [7]. Similarly, a multi-sensor wearable bracelet with body heat harvesting is proposed in [17].

Energy harvesting in IoT devices has necessitated the development of energy management and energy allocation algorithms for wearable IoT devices [6], [14], [20]. For example, the algorithm proposed in [14] allocates the duty cycle of a wireless sensor node for every control interval. The authors employ a linear programming model to maximize the work performed in a day. Allocation of duty cycle is equivalent to energy allocation, since we can easily derive the energy from the duty cycle. Similarly, [6] uses a dynamic programming approach to perform a near-optimal energy allocation for self-powered wearable devices. In this work, we assume that the energy budget for each time horizon is provided by a similar algorithm. Then, we maximize the number of gestures recognized under this energy budget.

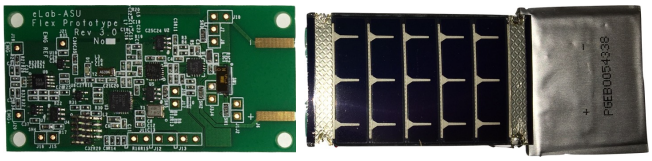
Power-aware computing is critical for wearable devices due to limited energy budget. Therefore, recent research has focused on an accuracy-power trade-off in wearable devices [15], [16], [26]. For instance, the technique presented in [26] uses a dynamic sensor selection to minimize the power consumption of a gesture recognition body area network. This leads to a maximization of the network lifetime. The work in [15] proposes an algorithm to perform optimal feature selection in wearable sensor networks. In contrast to these approaches, we propose a novel runtime algorithm that maximizes the number of gestures that can be recognized in a given time horizon. We first formulate the problem as a nonlinear optimization problem. Then, we use experimental measurements on a wearable device to derive a low complexity solution to the problem.

III. Target System Overview

A. Energy harvesting device prototype

The target wearable device shown in Figure 2 harvests energy using PV-cells and an energy harvesting circuit. Since the harvested energy is intermittent and exhibits significant variations over a day [2], we also employ a 1 g battery with 45 mAh capacity, as shown in Figure 2. The battery stores surplus energy and powers the target device, when the harvested energy is not sufficient. The power circuitry consists of a PV-cell SP3-37 [10], Lithium-polymer battery PGEB0054338 [9] and a MPPT charger TI BQ25504 [24]. In addition, the target system includes a TI CC2650 microprocessor [25] with 1.3 mW average power consumption and Invensense motion processing unit MPU-9250 [13].

When attached to user's hand, the wearable device captures the hand motion using the 3-axis accelerometer. Then, the microprocessor processes the data to recognize the intended gesture. Finally, the decoded gesture is transmitted to the target physical system through the BLE interface. *The amount of harvested energy determines the energy budget that can be exploited by the device*. To be practical, this system has to maximize the number of intended operations (i.e., in our case gesture recognition) *under this budget*, while maintaining a minimum level of recognition accuracy. Therefore, we propose a methodology to maximize the number of recognized gestures with a given energy budget and accuracy constraint.



(a) Mounted prototype **(b) Prototype with a PV-cell and a battery**

Fig. 2. Gesture recognition prototype used in this work.

B. Problem formulation

Given the characteristics of the energy harvesting system, we can determine the energy that can be harvested over a finite horizon t_h. We use this amount as the energy budget E_b available for the wearable device. We define the gesture recognition duration t_g as the time spent by the device to infer a single gesture. The wearable device actively senses the hand motion and processes the data during this period, which takes a portion of t_h. We denote the number of gestures recognized within the finite horizon by $N_g(t_g)$, since it is a function of the gesture recognition duration. The energy consumption per gesture $E_g(t_g)$ is a function of t_g, because t_g determines the active time of the processor and sensor. Similarly, the energy consumption of the device during the idle time is denoted by $E_i(t_g)$.

Finally, the energy consumed for transmitting the recognized gesture is denoted by E_{comm}. With this notation, we formulate the maximization of $N_g(t_g)$ as:

$$\text{maximize} \quad N_g(t_g) \quad such\ that \tag{1}$$

$$E_{total}(t_g) = E_g(t_g) \cdot N_g(t_g) + E_i(t_g) + E_{comm} \leq E_b \tag{2}$$

$$G_{acc}(t_g) \geq G_{acc,min} \tag{3}$$

The first constraint in this formulation ensures that the total system energy consumption is less than the energy budget. The second constraint guarantees that the accuracy of the gesture recognition $G_{acc}(t_g)$ is greater than a minimum accuracy $G_{acc,min}$. Note that $G_{acc}(t_g)$ is a function of t_g, since t_g determines the number of data points used for gesture recognition given the sampling frequency.

Solving the optimization problem given by Equations 1–3 at runtime is not easy, since both the objective and constraints

are nonlinear. Moreover, system dependencies make it hard to model the behavior of $E_g(t_g)$ and $E_i(t_g)$.

C. Overview of the proposed approach

The energy consumed per gesture is an increasing function of the gesture recognition duration t_g, since longer duration increases the active time of the sensors and processor. While precise characterization requires a detail model as developed in Section IV-A, it can be conceptually illustrated by the left axis in Figure 3. Hence, the gesture recognition duration t_g is *bounded from above by the given energy budget* E_b. Similarly, the gesture recognition accuracy is expected to improve, if a larger number data samples and longer processing time is used. Again, its precise behavior can be found only after user studies, but we can conceptualize it as a non-decreasing function of the gesture recognition duration, as illustrated by the right axis in Figure 3. Consequently, a *minimum accuracy requirement bounds the gesture recognition duration* t_g *from below regardless of the shape of the curve*. As a result, the feasible region for the optimization problem is the intersection of the regions for energy and accuracy, as highlighted in Figure 3.

To quantify a solution within the feasible region, we need to express the total energy consumption as a function of the gesture recognition duration, i.e., we need to model $E_g(t_g)$ and $E_i(t_g)$. Then, we need to find an expression for $N_g(t_g)$ such that it can be maximized within the feasible region. We solve this optimization problem through following steps:

1) Develop the gesture recognition algorithm on the target hardware and characterize the power consumption of individual components (*Section IV-A*),
2) Using this characterization, construct mathematical energy consumption models (*Section IV-B*),
3) Using the mathematical models, find an expression for $N_g(t_g)$ and its maximum point (*Section IV-C*),
4) Finally, combine the output of step 3 with the lower bound on t_g given by the gesture recognition accuracy $G_{acc,min}$ to find the optimal solution. We characterize $G_{acc}(t_g)$ through user studies presented in *Section V-C*.

IV. Energy-Optimal Gesture Recognition

A. Gesture recognition algorithm and energy characterization

We define five gestures made by hand – backward, forward, left, right, and wave – as shown in Figure 4. In addition, we include a stationary gesture to detect when the device is inactive. To classify these gestures, one can use a variety of supervised learning algorithms, such as support vector machine (SVM), decision tree, logistic regression, and neural network (NN). Selecting the appropriate algorithm depends on the input data size, accuracy and latency requirements, as well as available computational power and memory. In our application, the input is provided by a 3-axis accelerometer with 50 Hz sampling rate. Since common gestures take approximately 0.8 s [1], a baseline implementation with t_g =0.8 s leads to 3 $\times$ 50 Hz $\times$ 0.8 s = 120 input features. We target 90% or higher accuracy on a small IoT device. Finally, we aim at a flexible solution that can be easily extended to more number of gestures and input features. While both SVM and NN implementations meet the accuracy requirement on our test data, we adopt NN due to its flexibility.

Our programmable solution allows changing the number of hidden layers and neurons. The specific instance used in the experiments has a single hidden layer with 4 neurons and sigmoid activation function. These parameters are sufficient to achieve recognition accuracy above 90%. The input layer uses the input features to feed data into the hidden layer. The output layer has 6 neurons, one corresponding to each gesture. The sixth output neuron is added to separate the five gestures from a stationary gesture. The output layer evaluates the probability of each gesture. We employ two versions of the NN for the gesture recognition application:

- **Baseline NN** uses all 120 accelerometer samples collected by the 3-axis accelerometer during t_g as input features.
- **Reduced NN** employs transformed features derived from the raw accelerometer data. We utilize the minimum, maximum and mean values of each axis (x, y, z) over t_g. Hence, this amounts to a total of 9 input features. Since the number of transformed features does not depend on t_g, we can change it at runtime.

Operation and energy measurements: Figure 5 shows the power consumption of the microprocessor and the sensor (i.e., accelerometer) while processing one gesture. The dashed blue and solid red lines represent the measured power consumption of the microprocessor and the sensor, respectively. Initially, the system waits for user motion in the idle state. When the user makes a gesture, the accelerometer sensor wakes the system up, and performs a preprocessing routine to prepare the accelerometer and microprocessor. Then, the accelerometer starts sampling the motion data for a duration of t_g. We observe two different levels of power consumption in the sensor. The sensor power consumption is close to zero during idle state, while it consumes around 2 mW power in the active state. The power consumption also exhibits peaks during the state transitions because of pre- and post- processing of

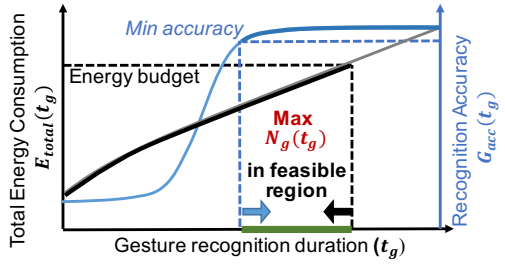


Fig. 3. Energy budget and minimum accuracy requirements constrain the gesture recognition duration t_g from above and below, respectively. Hence, we maximize the number of recognized gestures within the feasible region.

Fig. 4. Illustration of the target gestures.

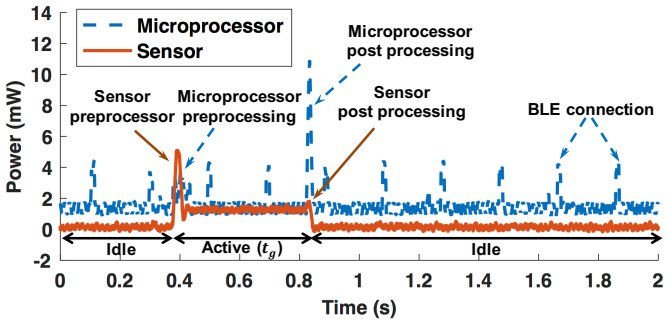


Fig. 5. Power consumption during a gesture recognition when $t_g = 400$ ms.

data acquisition. Once the data acquisition is completed, the microprocessor processes the sensor data, and transmits the recognized gesture using BLE. Unlike the sensor, the power consumption of the microprocessor shows periodic peaks, which are caused by the BLE module to maintain an active connection. In addition, pre-processing and post-processing tasks cause larger peaks in the microprocessor power.

B. Energy modeling

Energy characterization shown in Figure 5 provides useful insights, but it cannot be used directly to solve our optimization problem. Hence, we model the energy behavior of the gesture recognition system with the help of these power consumption measurements.

Active state energy: The energy consumption per gesture $E_g(t_g)$ consists of energy consumption of the microprocessor, $E^{\mu p}_{act}(t_g)$, and of the sensor, $E^{sen}_{act}(t_g)$ in active states, as follows:

$$E_g(t_g) = E^{\mu p}_{act}(t_g) + E^{sen}_{act}(t_g) \tag{4}$$

The active microprocessor energy consumption can be modeled by adding the peak components to the common static energy consumption as follows:

$$E^{\mu p}_{act}(t_g) = P^{\mu p}_{com} \cdot t_g + E^{\mu p}_{pre} + E^{\mu p}_{post}(t_g) \tag{5}$$

where $P^{\mu p}_{com}$, $E^{\mu p}_{pre}$, and $E^{\mu p}_{post}$ are the microprocessor's common static power consumption, preprocessing energy consumption, and post processing energy consumption, respectively. Note that the energy consumption of preprocessing does not depend on t_g.

Similarly, the sensor energy consumption can be written as:

$$E^{sen}_{act}(t_g) = P^{sen}_{com} \cdot t_g + E^{sen}_{pre} + E^{sen}_{acq}(t_g) + E^{sen}_{post}(t_g) \tag{6}$$

where P^{sen}_{com}, E^{sen}_{pre}, $E^{sen}_{acq}(t_g)$ and $E^{sen}_{post}(t_g)$ are the sensor's common static power consumption, the preprocessing energy consumption, the data acquisition energy, and the post processing energy consumption, respectively.

Idle state energy: The energy consumption of the system during the idle state is described as follows:

$$E_i(t_g) = E^{\mu p}_{idle}(t_g) + E^{sen}_{idle}(t_g) \tag{7}$$

where $E^{\mu p}_{idle}(t_g)$ and $E^{sen}_{idle}(t_g)$ are the total energy consumption of the microprocessor and the sensor in idle state, respectively. The idle time of system can be calculated by subtracting total active time from t_h. The energy consumption of microprocessor $E^{\mu p}_{idle}(t_g)$ can be modeled as below:

$$E^{\mu p}_{idle}(t_g) = P^{\mu p}_{com}\left(t_h - t_g \cdot N_g(t_g)\right) \tag{8}$$

Similarly, the sensor does not have any operation during the idle state. Hence, $E^{sen}_{idle}(t_g)$ can be written as:

$$E^{sen}_{idle}(t_g) = P^{sen}_{com}\left(t_h - t_g \cdot N_g(t_g)\right) \tag{9}$$

Communication energy: Since the BLE communication uses a fixed time interval t_{conn} to maintain the connectivity, the wearable system uses the upcoming slot to transmit the data. Hence, the energy consumption caused by BLE communication E_{comm} during the time horizon t_h can be described as follows:

$$E_{comm} = t_h / t_{conn} \cdot E_{conn} \tag{10}$$

where E_{conn} is energy consumption of BLE packet exchange. Note that E_{conn} is the additional energy consumption due to BLE communication. Hence, we have to consider the common static energy consumption when we calculate the energy per bit transmission.

We use the measured energy consumption values to obtain the constant terms in the energy models of the microprocessor and the sensor. Using these values, the energy models are expressed in terms of t_g. Detailed validation of the energy model is presented in Section V-B, while the numeric values are summarized in Table I.

C. Optimization methodology

The goal of the optimization problem is to maximize $N_g(t_g)$. Therefore, we start with expressing $N_g(t_g)$ as a function of processor and sensor energy consumption. Using Equations 1 and 2, we can express $N_g(t_g)$ as:

$$N_g(t_g) \leq \frac{E_b - E_{comm} - E_i(t_g)}{E_g(t_g)} \tag{11}$$

By substituting $E_i(t_g)$ and E_{comm} using Equations 8-10, we can re-write Equation 11 as:

$$N_g(t_g) \leq \frac{\left(\begin{array}{l} E_b - t_h/t_{conn} \cdot E_{conn} - (P^{\mu p}_{com} + P^{sen}_{com}) \cdot t_h \\ + (P^{\mu p}_{com} + P^{sen}_{com}) \cdot t_g \cdot N_g(t_g) \end{array}\right)}{E_g(t_g)}$$

Now, substituting the components of $E_g(t_g)$ using Equations 5 and 6 we can write:

$$N_g(t_g) \leq \frac{\left(\begin{array}{l} E_b - t_h/t_{conn} \cdot E_{conn} - (P^{\mu p}_{com} + P^{sen}_{com}) \cdot t_h \\ + (P^{\mu p}_{com} + P^{sen}_{com}) \cdot t_g \cdot N_g(t_g) \end{array}\right)}{\left(\begin{array}{l} E^{\mu p}_{pre} + E^{\mu p}_{post}(t_g) + E^{sen}_{pre} + E^{sen}_{acq}(t_g) + E^{sen}_{post}(t_g) \\ + (P^{\mu p}_{com} + P^{sen}_{com}) \cdot t_g \end{array}\right)}$$

Finally, we can simplify the above equation to express $N_g(t_g)$ as:

$$N_g(t_g) \leq \frac{E_b - t_h/t_{conn} \cdot E_{conn} - (P^{\mu p}_{com} + P^{sen}_{com}) \cdot t_h}{E^{\mu p}_{pre} + E^{\mu p}_{post}(t_g) + E^{sen}_{pre} + E^{sen}_{acq}(t_g) + E^{sen}_{post}(t_g)} \tag{12}$$

The numerator of Equation 12 represents the dynamic energy budget for gesture recognition. It is evaluated by subtracting BLE energy consumption and idle energy consumption as we have to spend this energy at a minimum to keep the system running. The denominator of Equation 12 gives the dynamic energy consumption of one gesture recognition. The number of gestures N_g is maximized when we minimize the dynamic energy consumption of one gesture recognition. Next, we will show that maximizing N_g is equivalent to minimizing t_g.

Lemma 1. *The denominator of Equation 12 is positive, and it is an increasing function of t_g.*

Proof. (1) All the addends in the denominator of Equation 12 represent the energy consumptions. Hence, their sum is positive.
(2) The accelerometer acquires data uniformly with a given sampling frequency. Hence, the amount of sensor data increases with t_g. This means that the processor and sensor have to do more processing as t_g increases. Therefore, $E^{sen}_{acq}(t_g)$, $E^{\mu p}_{post}(t_g)$ and $E^{sen}_{post}(t_g)$ are increasing function of t_g, while the remaining two terms are independent of t_g. Therefore, their sum, i.e., the denominator, is an increasing function of t_g. □

Lemma 2. *The numerator of Equation 12 is nonnegative.*

Proof. N_g is nonnegative since it is a physical operation. The denominator of Equation 12 is positive as stated by Lemma 1. Therefore, the numerator of Equation 12, which is a product of a positive and a nonnegative quantity, is nonnegative. □

The following theorem summarizes our major result, which enables a computationally efficient solution for maximizing $N_g(t_g)$.

Theorem 1. *Maximizing $N_g(t_g)$ is equivalent to minimizing t_g.*

Proof. We can see from Equation 12 that the numerator is independent of t_g. Moreover, Lemma 2 establishes that the numerator of Equation 12 is non-negative. Therefore, $N_g(t_g)$ is maximized when the denominator of Equation 12 is minimized. Lemma 1 establishes that the denominator of Equation 12 is an increasing function of t_g. This means that the denominator is minimized when we have the lowest possible t_g. Therefore, maximizing N_g is equivalent to minimizing t_g. □

Theorem 1 states that N_g is maximized when we reduce t_g. At the same time, t_g is bounded from below by the accuracy constraint $G_{acc,min}$. Therefore, the optimization problem is solved by choosing the minimum t_g that meets the accuracy constraint.

V. Experimental Evaluation

A. Experimental setup

Wearable system: We use the in-house wearable prototype described in Section III-A. It features test ports to measure the power consumption of the microprocessor and the MPU separately. Power measurements are performed using NI PXIe-4081 and PXIe-4080 digital multimeter systems [19] with 5 kHz sampling frequency.

Gesture recognition: The wearable device uses the NN to detect the gesture and transmits it to a host device. The host device stores the detected and the reference gesture. We use 30 data sets from seven users to test the accuracy of the gesture recognition system. Each set has a series of 50 gestures presented in random sequence. 10 data sets are reserved for training the NN. The training data is further divided into 80% training, 10% cross-validation and 10% test data. We obtain 96.5%, 97.4%, and 98.4% accuracy for the training, cross-validation and test data, respectively. The remaining 20 data sets are used for testing the accuracy of the NN. They are never seen by the NN during the training to reliably test the robustness of our gesture recognition system.

B. Energy model validation

The energy consumption of gesture recognition system is measured, when the system runs with baseline and reduced NNs. We sweep the gesture recognition duration t_g of the reduced NN from 400 ms to 800 ms, in increments of 100 ms, while t_g of the baseline NN is 800 ms. Note that the baseline NN does not support different t_g because it uses raw sensor data as inputs of NN.

The energy consumption of the key model parameters and their definitions are summarized in Table I. We use these parameters and the energy models presented in Section IV-B to compute the energy consumption per gesture. Figure 6 compares the modeled energy consumption with the measured data. Each bar shows the average energy consumed per gesture, where the average is taken over 60 random gestures. We achieve a mean percentage error of only 1.1% for the baseline NN. Similarly, the error ranges from 2.5% to 3.1% for the reduced set. Finally, the maximum modeling error observed across all data points is 5.2%.

C. Accuracy analysis

We analyze the accuracy of the gesture recognition application through experiments that involve seven different users. For each experiment, the user is given a series of 50 gestures in a random sequence. Each user repeats the experiment three times to obtain a larger data set. We observe greater than 90% detection accuracy for the reduced NN when the gesture recognition duration $t_g > 380$ ms, as shown in Figure 7. There is a significant degradation in accuracy when t_g is reduced below 380 ms. We observe this behavior because

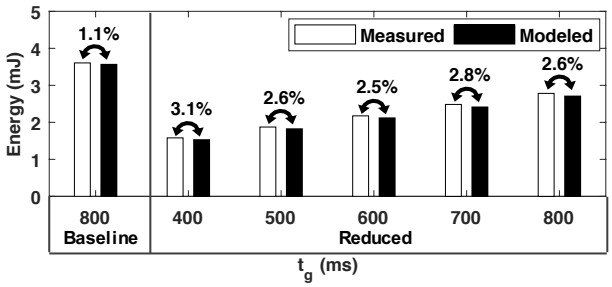


Fig. 6. Comparison of the energy consumed per gesture with the measured data. The number above the bars shows the mean percentage error.

TABLE I
THE ENERGY MODEL PARAMETERS AND VALUES

Parameter	Description	Value	Parameter	Description	Values	
					Baseline	Reduced
E_{conn}	BLE connection energy	28.5 nJ	t_{conn}	BLE connection interval	66 ms	200 ms
$P^{\mu p}_{com}$	Microprocessor common static power	1301.2 nW	$E^{\mu p}_{post}$	Microprocessor post processing energy	187.4 nJ	167.6·t_g−45.7 nJ
P^{sen}_{com}	Sensor common static power	149.4 nW	E^{sen}_{post}	Sensor post processing energy	27.9 nJ	26.9·t_g+7.3 nJ
$E^{\mu p}_{pre}$	Microprocessor preprocessing energy	91.3 nJ	E^{sen}_{acq}	Sensor data acquisition energy	1140.2 nJ	1132.0·t_g+10.0 nJ
E^{sen}_{pre}	Sensor preprocessing energy	110.2 nJ				

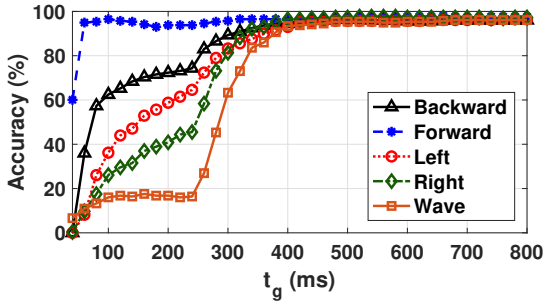


Fig. 7. Accuracy of gesture recognition for all the users.

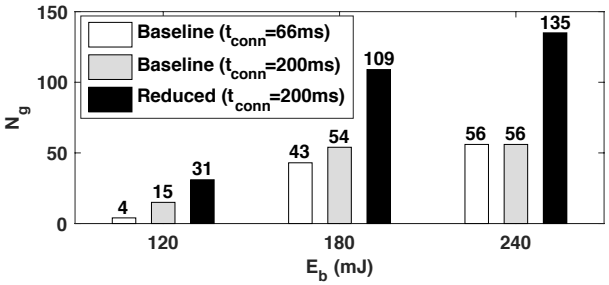


Fig. 8. Comparison of the number of recognized gestures for various energy budgets.

a lower t_g does not allow sufficient time for the NN to sufficiently differentiate between the gestures. Moreover, there may not be sufficient time to complete a gesture, when t_g is not long enough. For example, the accuracy for the wave gesture degrades faster than the rest of the gestures, since a larger number of samples is required to extract its signature. Hence, we use $t_g = 380$ ms as the lower bound of the gesture recognition time. We also note that the accuracy of the baseline NN degrades more rapidly with reducing t_g, since it uses all the data points. We aim to preserve the same level of accuracy (≈90%) for the baseline NN.

D. Optimization results

Inputs to the proposed optimization methodology are the time horizon and corresponding energy budget. Since the harvested energy can fluctuate rapidly due to environmental conditions, we assume a 1-minute finite horizon and analyze the optimization results for energy budget E_b={120 mJ, 180 mJ, 240 mJ}. We also note that larger time horizon does not change the percentage savings significantly, as it does not change the proposed algorithm. For comparisons, we use the baseline NN and a manually optimized version of the baseline NN by increasing the BLE connection interval t_{conn} to reduce the BLE overhead. Our solution (labeled as *Reduced*) uses the proposed optimization algorithm and the same t_{conn} as the manually optimized baseline. Throughout the experiments, we enforce a minimum gesture recognition accuracy of 90%.

When the energy budget is 120 mJ, the baseline NN is able to recognize only 4 gestures in one minute, since the static energy and BLE communication consume 72.5% and 21.6% of the energy budget, respectively. The baseline method can recognize 15 gestures by reducing BLE communication energy with longer t_{conn}. The proposed optimization provides an additional 2× boost and increases N_g to 31, as shown in Figure 8. Increasing the energy budget to 180 mJ and 240 mJ benefits the baseline NN significantly. Nevertheless, our optimization approach still provides 2× and 2.4× improvement over the optimized baseline, respectively. In particular, when the energy budget is 240 mJ, the maximum number of gestures that can be recognized by our approach is limited by the 1-minute time horizon, *not the energy budget*.

We illustrate the optimization results in more detail in Figure 9(a) and (b). The implicit upper bound induced by t_h is shown with the dotted curve, while the accuracy constraint is illustrated by the vertical dashed line. The result obtained with the baseline NN is the point labeled with the △ marker. In contrast, our optimization approach enables us to vary the number of gestures $N_g(t_g)$ along the solid curve. This curve is a decreasing function of t_g, as the proof of Theorem 3.1 shows. Hence, the optimal point is determined as the minimum gesture recognition duration, as stated by Theorem 3.1. When the energy budget is increased to 240 mJ, $N_g(t_g)$ curve is shifted up, as shown in Figure 9(b). This means larger number of recognized gestures as expected. We note that, $N_g(t_g)$ starts intersecting the timing constraint given by the dashed curve. As a result, the constraint due to time horizon ($t_g \cdot N_g(t_g) \leq t_h$) determines the maximum number of gestures. Hence, the optimal point is at the corner of the feasible region.

VI. CONCLUSIONS

Wearable IoT devices are becoming popular in interesting applications such as gesture-based control due to their small

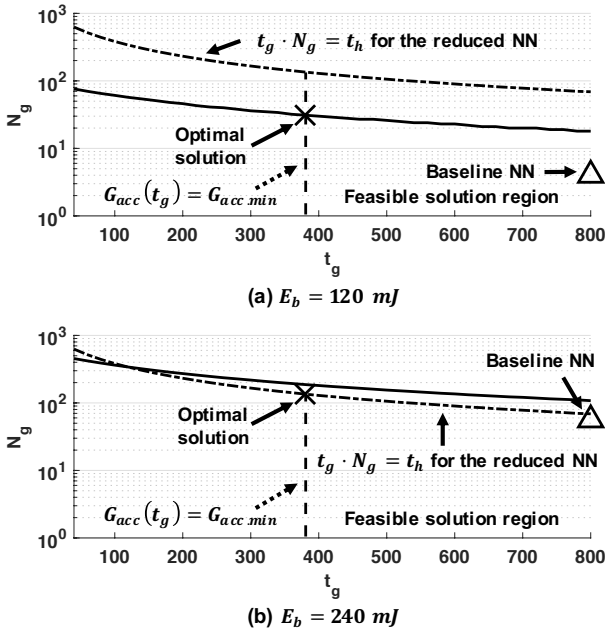


Fig. 9. Illustration of the optimal solution for different energy budgets.

form factor and low cost. Battery life limitation is one of the major issues of wearable devices. Hence, energy harvesting and optimal use of the harvested energy are critical. We presented an optimization approach to maximize the number of gesture can be recognized under the energy budget and accuracy constraints. We show that the proposed algorithm shows up to 2.4× improvement in the number of recognized gestures over the optimized baseline.

REFERENCES

[1] A. Akl, C. Feng, and S. Valaee, "A Novel Accelerometer-Based Gesture Recognition System," *IEEE Trans. Signal Process.*, vol. 59, no. 12, pp. 6197–6205, 2011.

[2] D. Balsamo *et al.*, "Hibernus: Sustaining Computation During Intermittent Supply for Energy-Harvesting Systems," *IEEE Embedded Systems Letters*, vol. 7, no. 1, pp. 15–18, 2015.

[3] H. Banaee, M. U. Ahmed, and A. Loutfi, "Data Mining for Wearable Sensors in Health Monitoring Systems: A Review of Recent Trends and Challenges," *Sensors*, vol. 13, no. 12, pp. 17 472–17 500, 2013.

[4] D. Bandyopadhyay and J. Sen, "Internet of Things: Applications and Challenges in Technology and Standardization," *Wireless Personal Commun.*, vol. 58, no. 1, pp. 49–69, 2011.

[5] G. Bhat, R. Deb, and U. Y. Ogras, "OpenHealth: Open source platform for wearable health monitoring," *IEEE Design & Test*, 2019.

[6] G. Bhat, J. Park, and U. Y. Ogras, "Near Optimal Energy Allocation for Self-Powered Wearable Systems," in *Proceedings of International Conference on Computer Aided Design*, 2017.

[7] Q. Brogan, T. O'Connor, and D. S. Ha, "Solar and Thermal Energy Harvesting With a Wearable Jacket," in *Proc. IEEE Intl. Symp. Circuits and Syst.*, 2014.

[8] V. Custodio, F. J. Herrera, G. López, and J. I. Moreno, "A Review on Architectures and Communications Technologies for Wearable Health-Monitoring Systems," *Sensors*, vol. 12, no. 10, pp. 13 907–13 946, 2012.

[9] DMI International Distribution Ltd. (2017) Curved lithium thin cells. http://www.dmi-international.com/data%20sheets/Curved%20Li%20Polymer.pdf.

[10] FlexSolarCells. (2013) SP3-37 Datasheet. http://www.flexsolarcells.com/index_files/OEM_Components/Flex_Cells/specification_sheets/01_FlexSolarCells.com_PowerFilm_Solar_SP3-37_Specification_Sheet.pdf.

[11] Guangzhou Markyn Battery Co., Ltd. (2011) GM555375. https://www.powerstream.com/poly/GM555375.pdf.

[12] S. Hiremath, G. Yang, and K. Mankodiya, "Wearable Internet of Things: Concept, Architectural Components and Promises for Person-Centered Healthcare," in *Proc. Int. Conf. Wireless Mobile Commun. and Healthcare*, 2014, pp. 304–307.

[13] InvenSense. (2017) Motion processing unit. https://www.invensense.com/products/motion-tracking/9-axis/mpu-9250.

[14] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, "Power Management in Energy Harvesting Sensor Networks," *ACM Trans. Embedd. Comput. Syst.*, vol. 6, no. 4, p. 32, 2007.

[15] A. Krause *et al.*, "Trading Off Prediction Accuracy and Power Consumption for Context-Aware Wearable Computing," in *Proc. IEEE Int. Symp. Wearable Comput.*, 2005, pp. 20–26.

[16] Y. Liang, X. Zhou, Z. Yu, and B. Guo, "Energy-Efficient Motion Related Activity Recognition on Mobile Devices for Pervasive Healthcare," *Mobile Netw. and Appl.*, vol. 19, no. 3, pp. 303–317, 2014.

[17] M. Magno *et al.*, "InfiniTime: Multi-Sensor Wearable Bracelet With Human Body Harvesting," *Sustainable Computing: Informatics and Systems*, vol. 11, pp. 38–49, 2016.

[18] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of Things: Vision, Applications and Research Challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497–1516, 2012.

[19] National Instruments, "Pxi digital multimeter," http://www.ni.com/en-us/shop/select/pxi-digital-multimeter, 2017.

[20] D. Noh, L. Wang, Y. Yang, H. Le, and T. Abdelzaher, "Minimum Variance Energy Allocation for a Solar-Powered Sensor System," *Distributed Comput. in Sensor Syst.*, pp. 44–57, 2009.

[21] J. A. Paradiso and T. Starner, "Energy scavenging for mobile and wireless electronics," *IEEE Pervasive Computing*, vol. 4, no. 1, pp. 18–27, 2005.

[22] M. Patel and J. Wang, "Applications, Challenges, and Prospective in Emerging Body Area Networking Technologies," *IEEE Wireless Commun.*, vol. 17, no. 1, 2010.

[23] R. Rawassizadeh, B. A. Price, and M. Petre, "Wearables: Has the Age of Smartwatches Finally Arrived?" *Commun. ACM*, vol. 58, no. 1, pp. 45–47, 2014.

[24] TI BQ25504. (2017) http://www.ti.com/lit/ds/symlink/bq25504.pdf.

[25] TI CC2650. (2017) http://www.ti.com/lit/ds/symlink/cc2650.pdf.

[26] P. Zappi *et al.*, "Activity Recognition From On-Body Sensors: Accuracy-Power Trade-Off by Dynamic Sensor Selection," *Lecture Notes in Comput. Sci.*, vol. 4913, p. 17, 2008.