Trading Off Power Consumption and Prediction Performance in Wearable Motion Sensors: An Optimal and Real-Time Approach

RAMIN FALLAHZADEH and HASSAN GHASEMZADEH, Washington State University, USA

Power consumption is identified as one of the main complications in designing practical wearable systems, mainly due to their stringent resource limitations. When designing wearable technologies, several system-level design choices, which directly contribute to the energy consumption of these systems, must be considered. In this article, we propose a computationally lightweight system optimization framework that trades off power consumption and performance in connected wearable motion sensors. While existing approaches exclusively focus on one or a few hand-picked design variables, our framework holistically finds the optimal power-performance solution with respect to the specified application need. Our design tackles a multi-variant non-convex optimization problem that is theoretically hard to solve. To decrease the complexity, we propose a smoothing function that reduces this optimization to a convex problem. The reduced optimization is then solved in linear time using a devised derivative-free optimization approach, namely cyclic coordinate search. We evaluate our framework against several holistic optimization baselines using a real-world wearable activity recognition dataset. We minimize the energy consumption for various activity-recognition performance thresholds ranging from 40% to 80% and demonstrate up to 64% energy savings.

CCS Concepts: • **Computer systems organization** → **Sensor networks**; *Embedded software*; *Real-time system specification*; • **Mathematics of computing** → *Combinatorial optimization*; • **Computing methodologies** → *Supervised learning*; • **Software and its engineering** → *Embedded software*;

Additional Key Words and Phrases: Embedded systems, body sensor networks, activity recognition, wearable monitoring systems, energy optimization

ACM Reference format:

Ramin Fallahzadeh and Hassan Ghasemzadeh. 2018. Trading Off Power Consumption and Prediction Performance in Wearable Motion Sensors: An Optimal and Real-Time Approach. *ACM Trans. Des. Autom. Electron. Syst.* 23, 5, Article 67 (October 2018), 23 pages.
<https://doi.org/10.1145/3198457>

This work was supported in part by the National Science Foundation under grant CNS-1566359. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding organizations.

Authors' addresses: R. Fallahzadeh and H. Ghasemzadeh, Washington State University, 355 Spokane Street, Pullman, WA 99164-2752; emails: {rfallahz, hassan}@eecs.wsu.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2018 Association for Computing Machinery.

1084-4309/2018/10-ART67 \$15.00

<https://doi.org/10.1145/3198457>

1 INTRODUCTION

Internet of Things (IoT) consists of a large set of application-specific embedded sensing nodes connected through low power channels to work in conjunction and in service of people. Currently, despite the great interest in the research community as well as the industry to grow the ecosystem, there have remained major challenges toward application adaptivity, practicality, and user acceptance. Networked Wearable Systems (NWS), as one of the largest components of IoT, have witnessed an intensive growth in recent years. It is mainly due to their large potential in numerous applications in assistive technology and health care. Due to their capability in providing objective, unobstructed, real-time sensing, processing, and transmission, NWS are rapidly expanding in several application domains such as in-home-patient monitoring, chronic disease management, physical activity monitoring, emergency medical services, vision enhancement, respiratory and heart-rate monitoring, sports, and so on. However, the focus of this article is activity monitoring, as it is currently one of the most popular applications of wearable systems.

One of the major challenges in designing a user-acceptable wearable sensor is the battery lifetime. While the main hardware components inside a wearable sensor is built with the same technology of hand-held devices such as smartphones, unlike those devices, wearables are expected to function for several days with a single charge. Furthermore, in comparison to conventional devices, we have witnessed a substantial decrease in form factor for increased pervasiveness and user acceptance of the wearable technology. Such miniaturization has resulted in significantly less battery capacities that adversely impacts the battery lifetime of the wearable node. Both form factor and battery lifetime are indispensable factors that directly contribute to the user adoption of the technology and are both highly associated with the energy consumption of the NWS as a whole.

Wearables have shown potential to play a pivotal role in automation of the health-care industry, and facilitating self-care and in-home care, resulting in significant saving in health-care costs and boosting patient's quality of care (Kelly and Shahrokni 2017; Wale et al. 2017). While the rapid advances in hardware/software technologies have enabled researchers, clinicians, and industry to have easy access to necessary electronic circuitry and software modules, there are several non-functional design considerations that are preventing most of the proposed wearables to large-scale end-user adoption (Fortino et al. 2013). One of the dominating factors is low user-adherence to these technologies. Designing energy-efficient wearables and ubiquitous sensors is one of the key influences in improving adherence to technology. A number of studies using wearables in remote health monitoring have identified limited battery life and loss of power among the main contributors to low patient adherence (Suh et al. 2011; Worringham et al. 2011). In addition to complications of losing the stream of data and contact with patients, even for a relatively short period of time, prior feasibility studies have reported the loss of motivation and momentum in participants to use the system due to the battery limitation (Desai and Stevenson 2010). Improving the lifetime of the system can reduce the frequency that the end-users are required to recharge the sensors, which may not only improve the adherence but also ensure sustained monitoring and care.

Wearables have been used for a variety of applications, particularly in health and wellness. Activity monitoring is regarded as one of the dominant applications. These systems are typically equipped with low-power MEMS (e.g., accelerometer, gyroscope, magnetometer), a micro-controller (e.g., ATmega328) and a low-energy wireless transmitter (e.g., BLE). These components are, respectively, in charge of sensing, processing, and communication in activity recognition (AR) wearable devices.

Several energy-performance optimization work aimed to reduce the power consumption while maintaining an acceptable prediction performance (Javaid et al. 2013; Rault et al. 2017; Saeedi et al. 2014; Weiss et al. 2010). A comprehensive system optimization platform can inherit the existing single variable optimization techniques and significantly lower the cost of a system by

finding the joint global optimum. To this end, we need a comprehensive multi-variant optimization technique that is not specific to a initial design of the node or its high-level application. Brute-force approach is too complex to be used in real-time adaptation of resource-limited NWS when a possibly large number of variables are considered in the optimization problem (exponential complexity with respect to the number of optimization variables). Once the system is initially optimized and is employed, if a change is made in the network (e.g., a node is disconnected or there is a change in location of a node) or in the application level (e.g., change in desired performance of AR), a low-cost multi-variant optimization approach (ideally with linear complexity) is needed to re-optimize the networked system for an updated optimal energy-performance tradeoff. It is essential with respect to the limited energy and computation resources of wearables.

While several rigid and isolated algorithms have been proposed for optimization of networked wearable systems, the need for a practical multi-variate optimization algorithm with low computational complexity is still unsatisfied. While it may be possible to obtain a global optimum solution for such an optimization problem by performing a brute-force search when the search space is rather small, its exponential computational complexity makes this approach impractical for most emerging embedded and wearable technologies. Even in smaller search spaces, the time complexity is far higher than required for timely adjustments in dynamic settings of wearables.

Wearable sensors are designed to be employed in highly dynamic (human-centered) settings. A low cost optimization algorithm capable of being implemented on wearable systems can further empower the dynamic nature of wearable applications. We propose an efficient approach for holistic optimization of NWS. The framework performs linear search and quickly converges toward the global optimum configuration of the system. Therefore, it provides a practical solution to adaptivity requirement of networked wearable systems while guaranteeing optimal tradeoff of performance and energy consumption. In a recent study (Garcia-Perez et al. 2017), the authors have proposed a holistic scheme to increase the reliability and efficiency of data transmission in mobile health applications. Their methodology is specific to transmission of sensor readings and does not consider power consumption. However, it further shows the efficacy and potential of such holistic approaches in benefiting wearable monitoring systems. To the best of our knowledge, there has not been proposed a generic multi-variant optimization platform for wearables. The main contributions of this article are as follows:

- (1) We overview various system-level energy optimization techniques in wearables and discuss several real-world applications of our framework.
- (2) We propose a computationally simple and practical solution to online multivariate performance-energy optimization problem in wearable motion sensors. Our framework uses smoothing on system variables and transforms the initial non-convex optimization problem into a convex-optimization problem and performs linear search to find the global minimum cost with respect to the desired performance lower-bound.
- (3) We show that complexity of our cyclic coordinate search algorithm grows linearly with respect to the number of optimization variables and provides generic formulation for any finite number of variables. We further discuss the fast global convergence of the algorithm.
- (4) In absence of existing comprehensive optimization approaches, we design several holistic baselines. Using a real-world dataset, we demonstrate the superiority of our framework in comparison to our devised holistic baselines.

2 SIGNIFICANCE & APPLICATION OF THE PROPOSED FRAMEWORK

When designing a wearable system, an energy optimization scheme for having a prolonged battery lifetime is essential. Recent studies have proposed system-level optimization methods (e.g., greedy

approximation, linear programming, and dynamic programming) each rigidly targeting only one or few aspects of the system (Fallahzadeh et al. 2017; Javaid et al. 2013; Saeedi et al. 2014; Weiss et al. 2010). These techniques include dynamic feature selection (Saeedi et al. 2014), compressed sensing (Fallahzadeh et al. 2017; Majumdar and Ward 2015), communication protocols (Javaid et al. 2013) and distributed cooperative algorithms (Jin et al. 2015). However, a computationally practical solution for multi-variate optimization that can adaptively find the globally optimal configuration has remained a challenge. Such framework can still take advantage of existing solutions to reduce the size of optimization space in pre-processing. For instance, an energy aware feature selection algorithm can help reduce the feature space and as a result improve the time complexity in practice. However, investigating the impact of such algorithms in the speed of global real-time optimization is outside the scope of this study. The state-of-the-art advances in energy-aware wearable AR have been categorized and summarized in Section 5. To the best of our knowledge, the proposed holistic design optimization has not been investigated and researched in prior studies in the field of wearable human activity recognition and this study is the first to propose an effective, generic, and practical solution.

To further highlight the importance of having a generic low-complexity framework capable of actively reoptimizing the NWS, we provide three examples of high-level energy-saving techniques that can be efficiently powered by the proposed framework.

Tiered Performance-Based AR. Activity monitoring wearables are used in various levels, from detecting activities of daily living (e.g., cooking, doing laundry, and dog walking) to ambulatory movements (e.g., sitting, descending and ascending stairs, and walking) and even fine-grained step-count and gait pattern analysis. A collaborative sensing technique exploits these AR performance tiers for energy savings. A master node performs continuous monitoring with low sensitivity and wakes up the system upon detecting the activity of interest for high sensitivity AR. For instance, a smartphone awakens the gait analysis NWS upon fuzzy detection of “walking” activity. In Ghasemzadeh and Jafari (2011), the authors used the same concept to design a low-power template matching module that triggers the main AR module upon detection of target action(s). The modular AR system in Berchtold et al. (2011) uses a similar approach to expand the battery lifetime.

Battery Management in Continuous Monitoring. In applications where continuous 24/7 monitoring is required (e.g., life-logging and daily activeness monitoring), high-performance AR recognition can be sacrificed when the battery percentage is low. A similar approach has been used to adjust the brightness of smart phones and improve the battery lifetime (Shim et al. 2004).

Cost Effective Node Localization. The emerging AR algorithms aim to leverage node localization (NL) algorithms to determine the on-body location of wearables. Given the dynamic nature of NWS, NL technologies allow users to freely wear the sensing nodes on different locations, boosting the usability and robustness of NWS. However, such flexibility in the system has been shown to drastically affect the established energy-performance equation (Saeedi et al. 2014).

The above-mentioned scenarios are a few examples where a low-weight reconfigurable energy-performance tradeoff can provide an efficient basis for ubiquitous adaptation of these technologies.

3 PROPOSED FRAMEWORK

The aim of this study is to globally optimize the data processing pipeline in a wearable AR node. Figure 1 illustrates the processing modules typically found in state-of-the-art AR wearables. The motion sensor data is the input of the processing chain where the time continuous data is sampled at a certain sampling rate. The sampled data, then, form a stream of sensor readings that will be grouped into small segments of data. Typically, one data segment is a few seconds of sensor readings. Segmentation is done using a fixed size and non-overlapping sliding window. From each

Fig. 1. The overview of a data processing pipeline inside a wearable AR node.

segment, a set of informative features are extracted. A feature-set represents the corresponding data segment and is fed into a machine learning algorithm (i.e., AR classifier) for label classification. The classifier predicts a label based on the given feature set.

The accuracy of a classifier (i.e., system's performance) depends on a number of factors including the type of the classifier used, the quality of training data, the quality of data processing and representation, and so on. From a signal processing point of view, we aim to adjust the way that the input signal is processed and represented to the machine learning model (to minimize the energy consumption of the node) while ensuring the desired accuracy need is met. In other words, we aim to optimize the tuning variables $V = v_1, v_2, v_3$ in Figure 1 by trading off energy consumption and performance (classification accuracy).

We formulate the energy-performance optimization problem in AR wearable systems as follows:

$$\text{Minimize } Z(V) \text{ s.t. } V = \{v_1, v_2, \dots, v_n\}, \quad (1)$$

subject to

$$E(V) \leq \beta, \quad (2)$$

where Z is a multivariate energy cost function on bounded region $S \subset R^n$ and V is the finite set of system variables of size n where $v_i \in V$ is on a discrete bounded domain of integer values. The parameter β is an upper-bound on AR errors of the wearable AR system ($E = 1 - \text{Performance}$). The error of classification is formulated as follows:

$$E(V) = 1 - \frac{\text{\# of correctly labeled observations by } (h_V) \text{ classifier}}{\text{Total \# of test observations}}. \quad (3)$$

This problem has two important properties: First, it is a constrained discrete multi-variant optimization problem, which means it cannot be solved using derivative approaches (e.g., gradient descent); second, it has a non-convex hard constraint function, which means the whole optimization problem is non-convex and cannot be solved using local search solutions (e.g., coordinate descent). Local search approaches perform poorly on non-convex functions and do not guarantee the global optimum in presence of local minimum points. Generally, a non-convex multi-variate function introduces multiple local minima.

A continuous function $f(x)$ is a convex if on any interval of its domain (e.g., $[a, b] \in \text{dom}_f$), the median is not greater than the mean of the two ends of the interval. It is defined as follows:

$$f[\lambda x_1 + (1 - \lambda)x_2] \leq \lambda f(x_1) + (1 - \lambda)f(x_2), \quad (4)$$

where $x_1, x_2 \in [a, b]$ and $0 < \lambda < 1$. However, the function $Z(V)$ in Equation (1) is on a discrete domain. In general, a function $f : Z^n \rightarrow R$ is a convex if it can be extended to a continuous convex function. In other words, there is a continuous convex function $\tilde{f} : R^n \rightarrow R$ such that for all $x \in Z^n$, we have $f(x) = \tilde{f}(x)$. Finding $\tilde{f}$, however, is usually difficult and requires finding a perfect fit convexity preserving interpolation. Instead, we use the definition provided in Yüceer (2002). A function $f : S \rightarrow R$, where S is a bounded n -dimensional discrete domain, is convex if and only if for every $x_1, x_2 \in S$, it satisfies the following condition:

$$\lambda f(x_1) + (1 - \lambda)f(x_2) \geq \min_{u \in N(\hat{x})} f(u), \quad (5)$$

where $\lambda \in \{0, 1\}$, $\hat{x} = \lambda x_1 + (1 - \lambda)x_2$, and $N(\hat{x})$ is the set of neighbors of $\hat{x}$. In other words, there can only exist one local minimum in the entire domain S .

To discuss the convexity property in $Z(\cdot)$ and $E(\cdot)$ in our problem formulation (i.e., Equation (1)), we need to understand the details of each function. For each system variable (v_i), our energy model measures the total of the sensing, processing, and communication overhead imposed by having v_i . The total energy consumption is the total energy consumption overhead of V and the initial energy consumption of the system as defined below:

$$Z(V) = \sum_{i=1}^n \sum_{u \in \{s, p, c\}} Z_{v_i}^{(u)} + \sum_{w \in \{s, p, c\}} Z_{idle}^{(w)}, \quad (6)$$

where the subscript in Z refers to the source of energy consumption (i.e., the *idle* system or energy consumption overhead of v_i) and the superscript refers to the type of cost (i.e., sensing, processing, and communication). A more detailed formulation of our cost function is explained in Section 4.2. $Z^{(u)}$ is a summation of non-negative constants and/or linear functions on an increasing domain of positive integers. We know that nonnegative weighted summation of affine functions preserves the convexity, therefore, the results of convexity of Z follows. Examining the convexity of the $E(\cdot)$ is more difficult. This function is a black box that outputs a value between 0 and 1. Measuring the error of the AR is data driven and therefore its convexity cannot be generically accepted or rejected. This can be observed by the fact that, although most classification algorithms are deterministic when trained and test on the same data, the predication outcome and hence the error rate of the system ($E(V)$) cannot be determined for an unseen input. However, intuitively $E(V)$ is expected to demonstrate a convex trend upon increasing the input. For instance, increasing the number of features initially decreases the error to a certain soft point; however, an exceedingly large number of features may potentially skew the dataset and have an adverse effect on the performance. For some variables, after passing a soft point, the performance does not meaningfully improve. For instance, a higher sampling rate reduces the reconstruction error in sub-Nyquist rates but does not improve the performance once it satisfies the Nyquist sampling principle. As in any data-driven function, one should expect minor fluctuations in the output of $E(\cdot)$, potentially resulting in multiple inflection points. As a result, there is no concrete guarantee that the constrains function is a convex. In the near threshold subspace, such distortion in constraint function can lead to an early halting of the optimizer.

In a non-linear multi-variant optimization, a set of solutions (local minima) can be found that satisfies the constraint(s); however, a satisfying solution may not be a global optimum. In an optimization problem with a constrained objective function, a satisfying solution is also a non-dominated solution, meaning that other variables can no longer be optimized without violating the constraint. Using a single-variant optimizer (that optimizes a single variable) or a sequential optimizer (that optimizes one variable at a time) generally will not yield the global optimum because the outcome of locally optimizing one variable is a non-dominated solution. As a result, no further optimization can be performed in higher dimensions of the given multi-dimensional solution space.

In general, a non-convex non-linear optimization problem is a computationally hard problem. As shown in Murty and Kabadi (1987), such discrete optimization is a reduction of the subset sum problem, which is known to be NP-complete. However, if we restrain the size of V to a constant, a brute-force algorithm can solve it in pseudo-polynomial time. However, in practice, a brute-force approach is still expensive even for a moderate input size. An efficient alternative is to transform the initial non-convex optimization problem to a convex problem by finding a convex approximation of the non-convex areas. Assume that function $E(\cdot)$ has become convex by some smoothing function. As a result, the initial non-convex problem is now reduced to a multi-variate convex

optimization. The new problem still preserves the remaining properties of the initial problem. Once a smoothed and convex constraint is given, it will be possible to solve the problem in linear time using the devised cyclic coordinate search algorithm. We further discuss the transformation of the optimization formulation and the devised linear search algorithm in Section 3.1.

3.1 Solution: Cyclic Coordinate Search

Linearity of our solution is crucial to our framework, hence it enables the feasibility of real-time on-node execution. However, with such point-to-point linear optimization, we cannot directly access the smoothed $E(\cdot)$ function, because the coordinate search will only execute the points that it visits along its search path. Alternatively, knowing the upper-bound on E distortion, we can efficiently approximate the constraint and relax the formulation to a weakly constrained optimization problem:

$$\text{Minimize } Z : f(V) \text{ s.t. } V = \{v_1, v_2, \dots, v_n\}, \quad (7)$$

subject to

$$E(V) - \beta \leq \epsilon, \quad (8)$$

where ϵ is the maximum distance of all points in $E - V$ subspace from the smoothed and convex approximation of $E(\cdot)$. In other words, this convex optimization problem approximates the solution to the initial problem and is bounded by ϵ . To find the value of ϵ given enough training data of NWS, we perform a convexity preserving piecewise linear interpolation. Using a piecewise linear interpolation is sufficient, considering the discreteness of domain S . Such interpolation also needs to preserve the convexity of $E(\cdot)$. One efficient approach is to find the convex hull of a set of points $U = \bigcup_{v \in S} (v, E(v))$ (denoted by $CH(U)$) that is defined as the smallest convex set that contains all the points in set U . CH can be efficiently computed using Quickhull algorithm, which uses a divide and conquer approach similar to quick sort algorithm. We compute ϵ by finding the maximum value of the Euclidean distance of each $u \in U$ from $CH(U)$, described below:

$$\epsilon = \max_{x \in CH(U)} \min_{u \in U} \|u - x\|. \quad (9)$$

This can be exactly solved in polynomial time using linear programming. Then, the computed ϵ will be passed to our framework as a constant parameter. We note that the process of constructing the convex hull and finding ϵ is a part of the approximation process (i.e., a one-time computation carried out off-line) and therefore will not affect the computational complexity of the proposed optimization framework.

THEOREM 3.1. *Constraint Equation (8) has a convex behavior.*

PROOF. We can re-write Equation (8) as $E(V) - \epsilon \leq \beta$. We know that by definition $CH(V)$ is convex in $E - V$ subspace. Therefore we only need to show that $CH(V) \leq \beta$ (already known as convex) and $E(V) - \epsilon \leq \beta$ have the same behavior in our problem definition. From Equation (9), we get $E(V) - CH(V) \leq \epsilon$. As a result, we will have $E(V) - \epsilon \leq CH(V)$. Finally, we have $E(V) - \epsilon \leq CH(V) \leq \beta$. On a decreasing interval V , constraint Equation (8) has the same behavior as $CH(V) \leq \beta$. The result follows. $\square$

Figure 2 illustrates an overview of the proposed generic multi-variate optimization framework. In the off-line preprocessing phase, a boundary on convex approximation of the initial non-convex problem is computed and fed into the main algorithm using the training data along with the bounded system variables considered in the optimization ($V \in S$). In the real-time holistic optimization phase, the discrete bounded system variables in S construct a bounded multi-dimensional search space and, using this virtual search space, the low-weight coordinate search algorithm will

Fig. 2. The general optimization framework for energy-aware multi-variate NWS design.

be performed to find the jointly optimal value for each variable with respect to its parameters: the upper-bound on error of classification (β) and the approximation boundary (ϵ).

This optimization problem aims to minimize the cost function in a bounded n -dimensional ($S \subset N^n$) search domain. Constraint Equation (8) guarantees that the overall accuracy of the system will not fall below a certain threshold (i.e., β). This can also be viewed as a tuning parameter that allows to re-optimize the network to accommodate varied accuracy needs depending on a particular application. The optimization is done in terms of NWS variables to carry out the AR task. These variables are the n dimensions in our search space. The algorithm finds the best local solution in each iteration by cycling through the choices in its neighborhood. The technique is similar to the coordinate descent algorithm for derivative optimization problems. Before explaining our algorithm in details, we first formally define the terms used in the algorithm.

Definition 3.2 (Search Space). Denoted by S , is an n -dimensional bounded discrete Cartesian subspace where each axis $x_i = [Min(v_i), Max(v_i)]$. The subspace S^+ is the subset of S that satisfies the constraint Equation (8). Similarly, the subspace S^- is defined as $S^- = S - S^+$.

Definition 3.3 (Local Mesh). The local search space denoted by $M \subset S$, is a convex n -orthotope (e.g., rectangle when $n = 2$) with $P \in S$ as its geometric center. d_{x_i} for dimension i , is the mesh-size in each iteration of the algorithm and is defined as the Euclidean distance from P to M in i th dimension.

Definition 3.4 (M-Neighborhood). A point n_P is a mesh-neighbor of P iff it is a corner in M that has P as its centroid and $n_P \in S$. N is the set of all M -neighbors of P in each iteration.

The cyclic coordinate search algorithm comprises four main steps: (1) initialization, (2) local polling, (3) shrinking, and (4) stopping criteria. Figure 3 illustrates a simple example of the algorithm finding the optimal solution for a bivariate problem. x_1 and x_2 axes are labeled with values of two variables of the optimization problem. The contour plot in the background shows the values of Z for each $P \in (x_1, x_2)$. The optimal point is shown with a green dot that is the lowest cost point in $S^{(+)}$.

Initialization. First, a two-dimensional search space is constructed consistent with Definition 3.2. Next, the search point P is initialized using the upper-bound of V . (white dot in right left corner in Figure 3(a) indicates the starting point $P^{(0)}$). d_{x_1} and d_{x_2} set the initial mesh size ($d_{x_1}^{(0)} = d_{x_2}^{(0)} = 5$ in Figure 3(a)). The initial value of $d_{x_i}^{(0)}$ must satisfy the assumption in the proof of Theorem 3.1.

Fig. 3. The main steps in cyclic coordinate search algorithm, including initialization and local polling (3a), shrinking (3b), and stopping criteria (3c).

Local Polling. In each iteration, the algorithm updates P by moving to the best local point. It finds the next coordinate by finding the M -neighbor (if it can be found) with lowest $r \times Z$ (and is smaller than $r \times Z(P)$). The next point should, preferably, satisfy the constraint Equation (8). r is a binary penalty coefficient ($r = 1$ if $E(P) - \beta \leq \epsilon$ and $r = -1$ otherwise). Intuitively, P moves toward less Z if $r = 1$ and vice versa. To avoid loops, the points that are inside the search path are stored and avoided in future iterations.

Shrinking. If $r = 1$ (e.g., red dot in Figure 3(a)) or no valid next-point was found with respect to the current d_{x_i} , then the algorithm halves the values of d_{x_i} and continues the search. In Figure 3(b), the two search paths marked with white dots and white triangles have $d = 3$ and $d = 2$, respectively. Each time the algorithm shrinks the d , it erases the points in search path memory.

Stopping Criteria. When for some P , the M cannot be shrunk further ($d_{x_1} = d_{x_2} = 1$) and $r = 1$, we say P is in near optimal area. We keep P as the first candidate. Given the fact that the optimization is convex, from this point forward, only two observations can be made, P is either converging to the optimal solution or diverging from it. In each polling step, upon finding a new candidate (that has a Z smaller than the existing candidate), we update the candidate (i.e., P is still converging). If n consecutive points have $r = -1$, we take this observation as a clue for divergence of P (with respect to the convexity of the problem) and halt the algorithm. Moreover, at any point, if no valid neighbor was found (meaning that P has reached the boundary of S), the algorithm halts and last stored candidate will be the desired solution. The stopping criteria is illustrated in Figure 3(c).

Algorithm 1 describes the generic cyclic coordinate search solution.

ALGORITHM 1: Cyclic Coordinate Search Algorithm

```

1: procedure MAIN( $V$ )
2:   Result:  $P_{opt} : (\underline{v}_1, \dots, \underline{v}_n)$ 
3:   Input:  $V \in S$ ,
       $\beta \in [0, 1)$ ,
       $\epsilon$ ,
       $VP \leftarrow \emptyset$ 
4:   Initialization: construct  $S$ ;  $P \leftarrow P^{(0)}$ ;  $d_{x_i} \leftarrow d_{x_i}^{(0)}$ ;
5:   Polling:
6:   While ( $d=1$  and  $n$  consecutive  $r = -1$  observed) not true:
7:     if  $\forall i, d_i = 1$  and  $P \in S^{(+)}$  and  $Z(P) < Z(P_{opt})$  :
8:        $P_{opt} \leftarrow P$ ;
9:        $VP.add(P)$ ;
10:    If  $P \in S^{(-)}$  is true:
11:       $r \leftarrow -1$ ;  $d_i \leftarrow \lceil d_i/2 \rceil$ ;  $VP.clear()$ ;
12:       $P \leftarrow \text{BestNeighbor}(P, N)$ ;
13:      if  $P = \text{null}$  true: break;
14:    return  $P_{opt}$ 
15: procedure BESTNEIGHBOR( $P, N$ )
16:   If  $P \in S^+$  :  $r \leftarrow 1$  else  $r \leftarrow -1$ 
17:    $N_v \leftarrow \{n \in N | n \notin VP\}$ 
18:    $N^+ \leftarrow \{n \in N_v | n \in S^+\}$ 
19:   If  $N^+ \neq \emptyset$  : return  $n \in N^+$  with lowest  $r \times Z$ 
20:   If  $N^v \neq \emptyset$  : return  $n \in N^v$  with lowest  $r \times Z$ 
21:   return null

```

THEOREM 3.5. *Worst case time complexity of Cyclic Coordinate Search algorithm is $O(n)$ for in small dimensions and is $O(n^2)$ for extremely large problems.*

PROOF. To evaluate the time complexity of the proposed real-time complexity, we have to evaluate the polling step. Given an m -dimensional search space S , the algorithm locally searches for the best solution in its mesh-neighborhood. Regardless of the size of m , in each iteration only one direction is taken (i.e., $\text{BestNeighbor}(P, N)$ function is $O(1)$). The algorithm does not allow revisiting the nodes, meaning that, in the worst case, it traverses along all the axes of S , which is equal

to $O(m \times n)$, where n is the maximum size of input domain V . In smaller problems such as our case study, we can assume $m \ll n$ and therefore the worst case time complexity is $O(n)$. For extremely large problems (where m is comparable to n), the algorithm will run in $O(n^2)$. $\square$

4 VALIDATION

In this section, we first describe our experimental setup and the energy model used to estimate the energy consumption of our system. Then, we elaborate on the experimental results of our optimization framework by comparing against the optimal solution and the devised baselines.

4.1 Experimental Setup

To show the efficacy of our multi-variant optimization framework, we designed an experiment by developing a state-of-the-art AR system. The sensor data are collected from three subjects wearing a motion sensing node placed on their waist. The wearable sensing node is a TelosB mote equipped with a biaxial gyroscope and a triaxial accelerometer. This node is capable of measuring multi-dimensional acceleration and angular velocity of various body movements. Refer to Polastre et al. (2005) for more information on detailed specifications and energy consumption of TelosB mote. Three participants (one male with age 32 and two females with ages 22 and 55) were asked to perform 30 different ambulatory and transitional movements such as “sit to stand,” “sit to lie,” “jumping,” “going upstairs, and the like. Table 1 contains the complete list of all the movements in our experiment protocol. Participants were instructed to repeat each task 10 times. TelosB mote was programmed to sample data at 100Hz where sensor reading was being remotely sent to a base-station (i.e., a computer).

The collected data is segmented into windows of 1.5s of activity. We extracted 10 statistical features as well as 10 morphological features from each activity segment. The features are shown in Table 2 where the energy consumption of computing each feature is also reported. We calculated instruction-level energy consumption of MSP430 micro-controller, which is available on the TelosB mote used in our experiments. The details of energy calculations is based on an instruction-level energy model provided in Ghasemzadeh et al. (2015). Morphological features were equally-spaced samples of the activity signal. Overall, we extracted 100 features from the motion sensor per activity segment.

The objectives of this experiment are (1) to use our framework to reach the highest optimization level from the initial setup while maintaining above the accuracy constraint (this is useful for estimating the amount of energy savings) and (2) validate the performance of our algorithm compared to the ground truth (i.e., the exhaustive search algorithm a.k.a brute-force search) and two devised linear baselines.

4.2 Energy Estimation Model

We define an energy model for our AR system. The main sources of energy consumption in a wearable sensor node comprises the following modules: (1) data sampling (sensing cost), (2) feature extraction, (3) machine learning algorithms (classification), and (4) and information transmission (communication). Using Equation (6), the total energy consumption can be calculated after defining $Z^{(u)}$ where $u \in s, p, c$. The sensing cost is given by

$$Z^{(s)} = \lambda_s \times f, \quad (10)$$

where f denotes the given sampling frequency and λ_s represents the sensing cost of one sample in one sensing node that is available on TelosB mote used in our experiments. The accelerometer consumes 2.64 mW in active mode whereas the power consumption of gyroscope is 31.35mW. The parameter $Z^{(p)}$ includes two processing modules: feature extraction and activity classification. It

Table 1. The List of Ambulatory and Transitional Movements in Our Experiment Protocol

Movement ID	Details
Move-forward-A	Move forward one step, right leg
Move-forward-B	Move forward one step, left leg
Stand-sit-A	From stand to sit on an armchair
Stand-sit-B	From stand to sit on a dining chair
Sit-stand-A	From sit on an armchair to stand
Sit-stand-B	From sit on a dining chair to stand
Jumping	-
Kneeling-A	Right leg first
Kneeling-B	Left leg first
Sit-lie	Sit to lie
Lie-sit	Lie to sit
Stairs-up	Going up stairs, right leg first (one stair)
Stairs-down	Going down stairs, right leg first (one stair)
Turn-90-A	Turn clockwise 90 degrees and return
Turn-90-B	Turn counterclockwise 90 degrees and return
Turn-360-A	Turn clockwise 360 degrees
Turn-360-B	Turn counterclockwise 360 degrees
Grasp-turn-release-A	Grasp an object with right hand, turn clockwise and release
Grasp-turn-release-B	Grasp an object with two hands, turn clockwise and release
Look-back-A	Look back clockwise and return to the initial position
Look-back-B	Look back counterclockwise and return to the initial position
Reach-up-A	Reach up to a cabinet with right hand
Reach-up-B	Reach up to a cabinet with left hand
Reach-up-C	Reach up to a cabinet with both hands
Rising-A	Rising from kneeling, right leg
Rising-B	Rising from kneeling, left leg
Bend-grasp-A	Bend and grasp from ground with right hand
Bend-grasp-B	Bend and grasp from ground with left hand
Bend-grasp-C	Bend and grasp from coffee table with right hand
Bend-grasp-D	Bend and grasp from coffee table with left hand

is formulated as follows:

$$Z^{(p)} = Z^{(feat. ext.)} + Z^{(class.)} = \left(\sum_{a_i \in A} \lambda_{a_i} \right) + \alpha \times \lambda_{class}, \quad (11)$$

where a_i represents a distinct feature from the set of selected features (denoted by A) and α and λ_{class} denote, respectively, the number of data segments generated in 1s and the classification cost of each data segments. λ_{a_i} are reported in Table 2 and is calculated using instruction-level power numbers of MSP430 micro-controller. After a node performs activity classification, the predicted activity label will be ultimately transmitted to an access point. The communication cost of transmitting the labels are given by

$$Z^{(c)} = \lambda_t \times B, \quad (12)$$

where λ_t represents per-bit transmission cost of ZigBee transmitter used in TelosB mote and is extracted based on ZigBee protocol specifications (Baviskar et al. 2015). The parameter B denotes

Table 2. Energy Consumption of Computing Individual Features

Feature	Description	Energy (nJ)
Amp	Amplitude of signal segment	16386
Med	Median of signal segment	405159
Mean	Mean value of signal segment	8126
Max	Maximum amp of signal segment	8103
Min	Maximum amp of signal segment	8129
P2P	Peak to peak amplitude	16291
Var	Variance of signal segment	38846
Std	Standard deviation	40431
RMS	Root mean square power	29705
S2E	Start to end value	83
Morph	Morphological Samples	41

the number of bits being transmitted. The energy cost of transmitting a 5-bit class label in our experiment is equal to $929.5\mu J$.

4.3 Cyclic Coordinate Search versus Baselines

To empirically validate our framework, we proceed this section with discussing the results of optimization using our scheme applied to a three-variate AR system. We utilize the proposed framework to perform joint optimization with significantly low time complexity on (1) segmentation window size, (2) sampling frequency, and (3) feature selection.

As a benchmark for the proposed optimization solution, we first execute the brute force algorithm to find the ground truth solution to this optimization problem. The brute force algorithm examines all the possible configurations of the system to find the minimum cost sensing and processing pipeline that satisfies the accuracy constraint defined by user (β). Initially, the AR algorithm achieves 0.83 accuracy (or equally $E(P^{(0)}) = 0.17$), which is a significant performance given the large number of classes. The classifier chosen in our experiment is a j48 decision tree, which is suitable for wearable sensors for its efficiency and low computation cost. The evaluation is done off-line on Weka 3.6 machine learning toolkit using 10-fold cross validation scheme. We defined six AR error constraints ranging from 0.2 to 0.6 (equal to classification accuracy of 0.8 to 0.4). Using Equation (9), we reported approximation boundary of $\epsilon = 0.02$ for constraint $E(\cdot)$. The possible values of the input variables (v_i) are set to $v_1 = [0.2 : 1.5]$ seconds (equally spaced by 0.1), $v_2 = [1 : 100]$ Hz, and $v_3 = [1 : 100]$, representing the domain of variables segment size, sampling frequency, and number of features, respectively. For the initial mesh size ($d_i^{(0)}$), we used one-tenth of the range of input variables.

Furthermore, in absence of an existing multivariate optimization solution, we devised two linear multivariate optimization baselines to compare against the proposed framework. We note that the baselines are solutions that offer roughly the same time complexity and therefore, similar to the proposed framework and unlike brute-force method, can be considered practical (however, naive) solutions for real-time execution. The first baseline is all-at-once (AAO) optimization approach, where it is initiated at $P^{(0)}$ and in each iteration, all of the input variables are minimized ($v_i^{(k+1)} = v_i^{(k)} - d$). The parameter d is the distance between consecutive values of v_i (e.g., 0.1 for segment size). The iteration stops once the optimization constraint rejects the next point. The second baseline is the average sequential univariate (ASU) optimization, where all possible orders of $v_i \in V$ are used sequentially to optimize the system. It starts at $P^{(0)}$ and in each iteration, first

Fig. 4. (1) All-at-once optimization. (2) and (3) Sequential univariate optimization. The green dot is the optimal solution.

Fig. 5. The selected segmentation window size for our activity recognition system optimized by cyclic coordinate search versus baseline approaches and exhaustive search for various misclassification thresholds.

input variable is minimized (when other variables are fixed) until it cannot satisfy the optimization constraint, then, by fixing the first variable, it continues by minimizing the next variable. We report the average results over all possible orders. There are six runs of ASU for the optimization instance considered in this section.

Figures 5, 6, and 7, respectively, show the comparison of optimization results in terms of segmentation window size, sampling frequency, and the number of extracted features for various AR error constraints. Several observations can be made. First, comparing our algorithm against the brute-force solution, we confirm the convergence of our solution. However, since we are performing approximation on the initial optimization problem, the two results does not exactly match in several instances. For instance, when only 0.5 accuracy is desired, the proposed algorithm converges to a slightly weaker configuration, violating the hard β threshold set for the initial problem. However, as discussed before, such violation is bounded by ϵ that is the upper-bound on the asymptotic smooth approximation of $E(\cdot)$. Furthermore, the results show the superiority of our method against AAO and ASU in various thresholds. We note that dislike sampling frequency and feature set size, the larger value for segmentation size is more desired as it reduces the energy consumption by decreasing the value of α and B , described in Section 4.2. As previously shown in Figure 4, and empirically demonstrated in our results, the sequential and all-at-once schemes does not guarantee convergence to the global optimum. The reason is because AAO naively assumes that all variables have the same impact in terms of energy consumption overhead and performance,

Fig. 6. The selected sampling frequency for our activity recognition system optimized by cyclic coordinate search versus baseline approaches and exhaustive search for various misclassification thresholds.

Fig. 7. Total number of selected features in our activity recognition system optimized by cyclic coordinate search versus baseline approaches and exhaustive search for various misclassification thresholds.

and ASU over-minimizes the variables with higher ranks. In general, AAO outperforms ASU when lower performance is desired.

Figures 8–11 illustrate the detailed energy consumption of the AR node optimized using the proposed scheme, exhaustive search, and AAO and ASU approaches with respect to each power-hungry module (online data sampling, feature extraction, classification, and transmission, respectively). These figures confirm the previous results with respect to energy consumption of the node with our methodology performing close to optimal and superior to competitors, especially when lower thresholds are allowed.

Figure 12 reports the total cost of running AR on the node for each optimization algorithm with various thresholds on prediction error. As it can be observed, our optimization framework can consistently achieve near optimal energy consumption regardless of the desired prediction performance. AAO is only effective when fuzzy performance is desired and fails in high performance settings. In contrast, ASU's energy consumption is more acceptable in lower thresholds (0.2 and 0.3) but it non-linearly grows when higher thresholds are allowed. The energy savings of 34% and 64% is achieved, respectively, with comparison to AAO and ASU baselines. A higher energy savings (around 45%) against AAO was gained when high performance error was allowed. Moreover, energy consumption of the node under our algorithm has only deviated 5% from the optimal solution, that is essentially the asymptotic approximation factor of our solution for this optimization

Fig. 8. The estimated sensing cost of the activity recognition system optimized by cyclic coordinate search versus baseline approaches and exhaustive search for various misclassification thresholds.

Fig. 9. The estimated classification cost of the activity recognition system optimized by cyclic coordinate search versus baseline approaches and exhaustive search for various misclassification thresholds.

instance, given $\epsilon = 0.02$. We note because the idle energy consumption of the node was shared in all comparisons, we did not consider it in our energy computations. One important observation from this result is that the computation cost of J48 classifier is significantly lower than other modules (less than few hundred μJ). Given the linear growth of the cyclic coordinate search algorithm, it roughly shows the minor energy consumption overhead of the proposed real-time optimization framework, adding the fact that the optimization module will not be activated as frequent as the main AR pipeline.

Table 3 lists the energy consumption of the Cyclic Coordinate Search Algorithm in runtime. In addition, we note that the real-time optimization will not be executed as frequently as other processing components, which further emphasizes the insignificance of having this small energy overhead.

Fig. 10. The estimated feature extraction cost of the activity recognition system optimized by cyclic coordinate search versus baseline approaches and exhaustive search for various misclassification thresholds.

Fig. 11. The estimated transmission cost of the activity recognition system optimized by cyclic coordinate search versus baseline approaches and exhaustive search for various misclassification thresholds.

Table 3. Energy Consumption Overhead of the Proposed Real-Time Optimization Algorithm for Various Thresholds

$\beta \rightarrow$	0.2	0.3	0.4	0.5	0.6
Energy consumption (uJ)	317.6	376.9	383.8	402.1	390

4.4 Cyclic Coordinate Search versus a Single-Variant Optimizer

Additionally, we designed an experiment to empirically show the limitation of single-variant optimization in a wearable activity recognition node. We implemented and validated the power-aware optimal feature selection algorithm presented in Ghasemzadeh et al. (2015). We used the same dataset, energy model, and experiment setup as in previous experiments (see Sections 4.1 and 4.2). Table 4 shows the produced outcome in terms of number of selected features, the energy consumption of feature extraction module, and the total energy consumption, given various thresholds. In comparison with our algorithm, this approach produced a more efficient feature-set but, in return, did not allow other two variables to be minimized. It is because any further optimization in other

Fig. 12. The estimated total energy consumption of the activity recognition system optimized by cyclic co-ordinate search versus baseline approaches and exhaustive search for various misclassification thresholds.

Table 4. Energy Consumption of a Single Variant Optimizer for Various Thresholds versus the Proposed Global Optimization

$\beta \rightarrow$	0.2	0.3	0.4	0.5	0.6
# of features	18	6	3	2	2
Feature selection cost (μJ)	260.75	86.91	43.46	28.97	28.97
Total cost (μJ)	3.49×10^4	3.475×10^4	3.47×10^4	3.46×10^4	3.46×10^4
Total energy cost versus proposed	2.13x	2.34x	7.96x	9.47x	9.47x

modules would violate the overall classification error threshold. While solely looking at the feature optimization task, the results of this single-variant optimization seems significant, its total energy savings is far outperformed by our algorithm.

5 PRELIMINARIES

In this section, we first categorize the existing power-aware wearable design techniques and then discuss the two commonly adopted energy evaluation models in NWS.

5.1 Related Work

Wearables are typically equipped with low-power sensors such as MEMS (e.g., accelerometer, gyroscope, and magnetometer), a micro-controller (e.g., ATmega328), and a low-energy communication module (e.g., Bluetooth Low Energy, a.k.a BLE). These components are, respectively, in charge of sensing, processing, and communication in wearable and wireless sensing devices. Bounded by limited resources, energy-aware techniques have been researched and proposed to lower the cost of system and increase the device lifetime (Dabiri et al. 2008; French et al. 2007; Ghasemzadeh et al. 2015; Rault et al. 2017; Wang et al. 2015). The following overviews state-of-the-art power-reduction methods proposed in AR literature.

Node Selection. Depending on the nature of activities being monitored, a subset of nodes in the network can be utilized to efficiently monitor the activity with an acceptable accuracy. For instance, reducing the active nodes to half in a NWS when the target activity is focused on upper body could result in 50% overall energy savings. This subset of nodes may change as the target activity (or

activities) changes. Note that node selection process can also be regarded as the problem of finding the best on body node location(s) that guarantees the best AR outcome. Chowdhury et al. (2010), Lu et al. (2010), Noshadi et al. (2010), and Wang et al. (2013) are examples of the studies using such a technique.

Frequency Scaling. Current Micro Electro-Mechanical Sensors (MEMS) are capable of sensing and capturing data in various sampling frequencies ranging from 1 Hz to few hundreds (Ha et al. 2017; Kilani et al. 2016). Reducing the frequency significantly reduces the current in MEMS. For example, reducing the sampling rate from 100 Hz to 10 Hz in accelerometer decreases the current from 10 mA to 4 mA, resulting in 60% energy savings.

Cost-Sensitive Data Analysis. Transmitting raw data from sensing nodes will result in excessive communication cost. Therefore, prior to transmission, each node performs different signal processing tasks to extract only the useful data for further analysis (e.g., data fusion and classification) at destination. These signal processing includes data segmentation, feature extraction, and feature selection. Several studies have proposed methods for lower cost data processing and reduced communication cost in nodes (Javaid et al. 2013; Saeedi et al. 2014; Weiss et al. 2010).

Operational Mode. Existing MEMS function in various operational modes such as high sensitive, economy, and idle mode. While some applications require higher sensitivity, there are other applications (e.g., motion wake) that can be achieved with idle/standby mode. Having a sensor in idle mode will reduce the current by one order of magnitude on average (Williamson et al. 2015).

Sensor Selection. Depending on the application of interest, the type of sensors needed for efficient monitoring may change. While in applications such as activity monitoring and motion wake the use of accelerometer is sufficient, 3D motion capture requires both accelerometer and gyroscope and navigation needs all three sensors. Disabling power hungry sensors has been used in several prior studies (Kim et al. 2014, 2010; Nath 2012; Sun et al. 2011).

Selective Sampling. Due to limited power recourses in wireless sensors, it is often impractical to perform continuous data sensing. Inspired by the fact that human body is a dynamically stable entity, it is possible to selectively sense the data and reduce the duty cycle of the sensors. This will not only reduce the sensing cost, but also the storage, processing and communication costs drastically (Hossain et al. 2017; Shahmohammadi et al. 2017). Selective sampling techniques have been studied priorly and reported energy savings up to 95% compared against the continuous sampling (French et al. 2007).

Compressive Sensing and Communication: Sub-Nyquist sparse sampling (i.e., compressed sensing) along with data compression techniques, aim to reduce the communication load for less power consumption by the transmitter. Data compressions methods compress the signals that were initially sampled at Nyquist rate, locally, to reduce the transmission cost (Fallahzadeh et al. 2017; Marcelloni and Vecchio 2009a; Wang et al. 2015). Compressive sensing is computationally less complex and does not require the data to be stored, however, it is a lossy compression technique (Wang et al. 2016).

Dynamic Voltage Scaling. In addition to different operational modes, dynamic voltage scaling (DVS) techniques have been proposed to further improve the efficiency of embedded processing units by adaptively manipulating the source voltage and operating frequency in real-time (Chung et al. 2016; Dabiri et al. 2008).

Collaborative Sensing/Processing. It is a master-slave technique where a master node (typically an access point or a node with more computation and energy resources) is in charge of more complex or energy consuming task(s). For instance, multiple body nodes could be in charge of sensing and related feature extraction while the master (e.g., a smartphone) could be in charge of utilizing those features for the classification task, which is usually more burdensome (collaborative processing) (Aldeer et al. 2017; Fortino et al. 2015; Park et al. 2011). One other example is when

a smartphone performs continuous AR and upon detecting the activity of interest, triggers other nodes for higher accuracy data acquisition (collaborative sensing).

Communication Optimization. There is a large body of research on optimizing the communication protocol for wireless body sensor network (Brajos Lopez et al. 2014; Marcelloni and Vecchio 2009b). It involves designing efficient routing in mobile ad-hoc networks, opportunistic communication, and adaptive power controls. Body area networks typically use low frequency signals with limited energy and range.

5.2 Energy Evaluation Methods

Over the past few years, researchers have adopted different methods to estimate the power consumption of their proposed systems/methods. In this part, we categorize those efforts into system-level energy modeling and physical-level energy analysis, described below:

System-Level Energy Modeling. In this approach, the energy consumption will be formulated in terms of the task load (e.g., no. of bits transmitted, number of low-level computations performed) multiplied by the a normalized value (usually a constant, e.g., J/bit and J/instruction). The formulation can be presented as a linear programming problem.

Physical-Level Energy Analysis. In this case, the design is synthesized (e.g., using Synopsys tool (Synopsys 2010)) with a standard cell library. Using Register Transfer Level codes, a gate-level simulation of the system is fed to hardware synthesis tools and power compilers to simulate the power consumption of the design. This method is usually more trusted in power analysis studies; however, it is still dependent on the simulation and synthesis parameters (e.g., the standard cell technology).

6 DISCUSSION AND FUTURE WORK

The motivation behind this work is the fact that performing isolated optimization on NWS can be significantly improved by taking into account the intrinsic dependence of various system design parameters and the highly dynamic nature of wearable sensor. For example, considering the feature selection without looking at node locations will not produce the best optimization results since the information gain of a particular feature could be different as we change the sensor's location. Sensor's location, similarly, impacts the optimal sampling frequency as the intensity of the collected data changes with rearrangement of network topology. Moreover, with the emergence of wearables, each with several applications (e.g., fitness tracking, sleep monitoring, and rehabilitation), real-time and adaptive system optimization becomes more crucial. As another example, consider continuous changes of nodes in the wearable sensor network (e.g., user adds a new wearable sensor or decides not to wear his/her smartwatch today). The proposed platform is an essential first step toward having a computationally low-cost dynamic optimization framework, which can potentially help the NWS to adapt itself to contentious changes in the system associated with both user-environment interactions and application-level needs. We note that detecting change points in a NWS is itself an active research area and does not fit in the scope of this research.

The theoretical analysis stated in Theorem 3.5 provides proof of scalability but does not give concrete evidence of absolute time cost overhead imposed by the algorithm. To further illustrate the real-time application of our methodology, we use the equation below to estimate the real-time computation overhead:

$$t_E = \frac{\sum_i (C_i \times TE_i)}{CLK}, \quad (13)$$

where t_E denotes the CPU utilization of the MSP430 microcontroller on our TelosB motes. The numerator is number of cycles per instruction multiplied by the number of times each instruction is executed. The denominator is the clock frequency of the microcontroller. To investigate the CPU

utilization of our algorithm, we considered an average run of the algorithm (where $\beta = 0.4$) translated for MSP430 gcc compiler compatibility. The number of cycles required for each instruction can be found in Texas Instruments (2010). Using a low-frequency clock rate of 400kHz in our microcontroller, our algorithm will utilize 4.12% of CPU for a single run where $\beta = 0.4$. Taking into account the sparsity of our algorithm execution, we argue that real-time time cost of the proposed algorithm can be regarded as negligible.

We acknowledge that there are still a few parts in our framework design and implementations that need to be further researched in our future studies: (1) The off-line search-space for each system variable is done exhaustively. While performing an exhaustive search for a single variable problem in off-line mode is much more affordable and does not impact the runtime complexity in our real-time optimization, using proven heuristic optimizations such as grammatical evolutionary techniques can make the framework much faster in pre-processing phase, especially when a higher number of design variables are considered. (2) Although the experiments are based on a real-world dataset, the optimization and energy consumption estimations are done retrospectively. While such analysis sheds light on applicability and effectiveness of our methodology, as a future work, we aim to implement and program the framework on the node to have a more precise evaluation of its practicality and energy savings. (3) In absence of an alternative lightweight optimization solution that can be applicable in our case study, we devised two lightweight baselines that have shown to be competitive in some of our experiments. In future work, we will devise more competitive baselines to better highlight our findings. One example could be devising a multi-objective evolutionary algorithm (EA). The use of such methods in this context is novel and yet to be investigated. However, EAs are generally unpredictable and highly dependent on convexity of the problem space and therefore do not guarantee finding an optimal Pareto-front.

7 CONCLUSION

This article proposed an optimal and light-weight framework for holistic and real-time optimization of wearable sensing nodes by devising a cyclic coordinate search algorithm. We reduced the initial hard non-convex optimization problem to a convex multi-objective optimization and introduced a linear time algorithm that runs on a multi-dimensional bounded discrete domain. Using a real-world wearable AR dataset, we showed that the proposed framework can offer an optimal, practical, and comprehensive optimization solution in the presence of multiple design variables. The linear time complexity of our algorithm provides the means for low-cost deployment of a real-time optimization framework that could adaptively optimize the system based on the desired prediction performance. Our experiments using TelosB mote with three participants performing 30 movements showed that the proposed framework can achieve between 34% to 64% energy savings compared to the competing lightweight holistic optimization baselines.

REFERENCES

- Murtadha M. N. Aldeer, Richard P. Martin, and Richard E. Howard. 2017. Tackling the fidelity-energy trade-off in wireless body sensor networks. In *Proceedings of the 2017 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE'17)*. IEEE, 7–12.
- Jaypal Baviskar, Afshan Mulla, Manish Upadhye, Jeet Desai, and Aniket Bhovad. 2015. Performance analysis of ZigBee based real time home automation system. In *Proceedings of the 2015 International Conference on Communication, Information & Computing Technology (ICCICT'15)*. IEEE, 1–6.
- Martin Berchtold, Henning Günther, Matthias Budde, and Michael Beigl. 2011. Scheduling for a modular activity recognition system to reduce energy consumption on Smartphones. In *Proceedings of the 24th International Conference on Architecture of Computing Systems (ARCS'11) Workshops*. Springer, 1–8.
- Ruben Braojos, Ivan Beretta, Jeremy Constantin, Andreas Burg, and David Atienza. 2014. A wireless body sensor network for activity monitoring with low transmission overhead. In *Proceedings of the 12th IEEE International Conference on Embedded and Ubiquitous Computing (EUC'14)*. IEEE, 265–272.

- Atanu Roy Chowdhury, Ben Falchuk, and Archan Misra. 2010. Medially: A provenance-aware remote health monitoring middleware. In *Proceedings of the 2010 IEEE International Conference on Pervasive Computing and Communications (PerCom'10)*. IEEE, 125–134.
- Ching-Che Chung, Wei-Siang Su, and Chi-Kuang Lo. 2016. A 0.52/1 V fast lock-in ADPLL for supporting dynamic voltage and frequency scaling. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 24, 1 (2016), 408–412.
- Foad Dabiri, Navid Amini, Mahsan Rofouei, and Majid Sarrafzadeh. 2008. Reliability-aware optimization for DVS-enabled real-time embedded systems. In *Proceedings of the ISQED 2008. 9th International Symposium on Quality Electronic Design*, 2008. IEEE, 780–783.
- Akshay S. Desai and Lynne Warner Stevenson. 2010. Connecting the circle from home to heart-failure disease management. *The New England Journal of Medicine* 363, 24 (2010), 2364–7.
- Ramin Fallahzadeh, Josue Pagan Ortiz, and Hassan Ghasemzadeh. 2017. Adaptive compressed sensing at the fingertip of internet-of-things sensors: An ultra-low power activity recognition. In *Proceedings of the 2017 Design, Automation & Test in Europe Conference & Exhibition (DATE'17)*. IEEE, 996–1001.
- Giancarlo Fortino, Stefano Galzarano, Raffaele Gravina, and Wenfeng Li. 2015. A framework for collaborative computing and multi-sensor data fusion in body sensor networks. *Information Fusion* 22 (2015), 50–70.
- Giancarlo Fortino, Roberta Giannantonio, Raffaele Gravina, Philip Kuryloski, and Roozbeh Jafari. 2013. Enabling effective programming and flexible management of efficient body sensor network applications. *IEEE Transactions on Human-Machine Systems* 43, 1 (2013), 115–133.
- Brian French, Daniel P. Siewiorek, Asim Smailagic, and Michael Deisher. 2007. Selective sampling strategies to conserve power in context aware devices. In *Proceedings of the 2007 11th IEEE International Symposium on Wearable Computers*. IEEE, 77–80.
- Cesar Garcia-Perez, Almudena Diaz-Zayas, Alvaro Rios, Pedro Merino, Kostas Katsalis, Chia-Yu Chang, Shahab Shariat, Navid Nikaein, Pilar Rodriguez, and Donal Morris. 2017. Improving the efficiency and reliability of wearable based mobile eHealth applications. *Pervasive and Mobile Computing* 40, Supplement C (2017), 674–691. DOI: <http://dx.doi.org/10.1016/j.pmcj.2017.06.021>
- Hassan Ghasemzadeh, Navid Amini, Ramyar Saeedi, and Majid Sarrafzadeh. 2015. Power-aware computing in wearable sensor networks: An optimal feature selection. *IEEE Transactions on Mobile Computing* 14, 4 (2015), 800–812.
- Hassan Ghasemzadeh and Roozbeh Jafari. 2011. Ultra low power granular decision making using cross correlation: Optimizing bit resolution for template matching. In *Proceedings of the 2011 17th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'11)*. IEEE, 137–146.
- Chi-Yuan Ha, Yo-Xian Wang, and Che-Wei Chang. 2017. Dynamic power management for wearable devices with non-volatile memory. In *Proceedings of the 2017 International Conference on Applied System Innovation (ICASI'17)*. IEEE, 37–39.
- H. M. Sajjad Hossain, Md Abdullah Al Hafiz Khan, and Nirmalya Roy. 2017. Active learning enabled activity recognition. *Pervasive and Mobile Computing* 38, Part 2 (2017), 312–330. DOI: <http://dx.doi.org/10.1016/j.pmcj.2016.08.017> Special Issue IEEE International Conference on Pervasive Computing and Communications (PerCom) 2016.
- Texas Instruments. 2010. MSP430x2xx family user's guide (rev. H). (2010).
- Nadeem Javaid, Sarim Hayat, Mustafa Shakir, Mahmood Ashraf Khan, Safdar Hussain Bouk, and Z. A. Khan. 2013. Energy efficient Mac protocols in wireless body area sensor networks-a survey. *arXiv:1303.2072*.
- Yanling Jin, Yongsheng Ding, Kuangrong Hao, and Yaochu Jin. 2015. An endocrine-based intelligent distributed cooperative algorithm for target tracking in wireless sensor networks. *Soft Computing* 19, 5 (2015), 1427–1441.
- Ciara M. Kelly and Armin Shahrokni. 2017. From shelf to bedside—wearable electronic activity monitoring technologies might assist oncologists in functional performance status assessment of older cancer patients. *Clinical Colorectal Cancer* 16, 2 (2017), e115–e118.
- Dima Kilani, Mohammad Alhawari, Baker Mohammad, Hani Saleh, and Mohammed Ismail. 2016. An efficient switched-capacitor DC-DC buck converter for self-powered wearable electronics. *IEEE Transactions on Circuits and Systems I: Regular Papers* 63, 10 (2016), 1557–1566.
- Bongjae Kim, Yookun Cho, and Jiman Hong. 2014. AWNIS: Energy-efficient adaptive wireless network interface selection for industrial mobile devices. *IEEE Transactions on Industrial Informatics* 10, 1 (2014), 714–729.
- Donnie H. Kim, Younghun Kim, Deborah Estrin, and Mani B. Srivastava. 2010. Sensloc: Sensing everyday places and paths using less energy. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*. ACM, 43–56.
- Hong Lu, Jun Yang, Zhigang Liu, Nicholas D. Lane, Tanzeem Choudhury, and Andrew T. Campbell. 2010. The Jigsaw continuous sensing engine for mobile phone applications. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*. ACM, 71–84.
- Angshul Majumdar and Rabab K. Ward. 2015. Energy efficient EEG sensing and transmission for wireless body area networks: A blind compressed sensing approach. *Biomedical Signal Processing and Control* 20 (2015), 1–9.

- Synopsys. MEDICI User's Manual. 2010. Synopsys Inc. *Mountain View, CA*.
- Francesco Marcelloni and Massimo Vecchio. 2009a. An efficient lossless compression algorithm for tiny nodes of monitoring wireless sensor networks. *Computer Journal* 52, 8 (2009), 969–987.
- Francesco Marcelloni and Massimo Vecchio. 2009b. An efficient lossless compression algorithm for tiny nodes of monitoring wireless sensor networks. *Computer Journal* 52, 8 (2009), 969–987.
- Katta G. Murty and Santosh N. Kabadi. 1987. Some NP-complete problems in quadratic and nonlinear programming. *Mathematical Programming* 39, 2 (1987), 117–129.
- Suman Nath. 2012. ACE: Exploiting correlation for energy-efficient and continuous context sensing. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*. ACM, 29–42.
- Hyduke Noshadi, Foad Dabiri, Saro Meguerdichian, Miodrag Potkonjak, and Majid Sarrafzadeh. 2010. Energy optimization in wireless medical systems using physiological behavior. In *Wireless Health 2010*. ACM, 128–136.
- Taiwoo Park, Jinwon Lee, Inseok Hwang, Chungkuk Yoo, Lama Nachman, and Junehwa Song. 2011. E-gesture: A collaborative architecture for energy-efficient gesture recognition with hand-worn sensor and mobile devices. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*. ACM, 260–273.
- Joseph Polastre, Robert Szewczyk, and David Culler. 2005. Telos: Enabling ultra-low power wireless research. In *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*. IEEE Press, 48.
- Tifenn Rault, Abdelmadjid Bouabdallah, Yacine Challal, and Frédéric Marin. 2017. A survey of energy-efficient context recognition systems using wearable sensors for healthcare applications. *Pervasive and Mobile Computing* 37 (2017), 23–44.
- Ramyar Saeedi, Brian Schimert, and Hassan Ghasemzadeh. 2014. Cost-sensitive feature selection for on-body sensor localization. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*. ACM, 833–842.
- Farhad Shahmohammadi, Anahita Hosseini, Christine E. King, and Majid Sarrafzadeh. 2017. Smartwatch based activity recognition using active learning. In *Connected Health: Applications, Systems and Engineering Technologies (CHASE'17), 2017 IEEE/ACM International Conference on*. IEEE, 321–329.
- Hojun Shim, Naehyuck Chang, and Massoud Pedram. 2004. A backlight power management framework for battery-operated multimedia systems. *IEEE Design & Test of Computers* 21, 5 (2004), 388–396.
- Myung-kyung Suh, Chien-An Chen, Jonathan Woodbridge, Michael Kai Tu, Jung In Kim, Ani Nahapetian, Lorraine S. Evangelista, and Majid Sarrafzadeh. 2011. A remote patient monitoring system for congestive heart failure. *Journal of Medical Systems* 35, 5 (2011), 1165–1179.
- Feng-Tso Sun, Cynthia Kuo, and Martin Griss. 2011. PEAR: Power efficiency through activity recognition (for ECG-based sensing). In *2011 5th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth'11)*. IEEE, 115–122.
- Janet Wale, Anna Mae Scott, Bjørn Hofmann, Sarah Garner, Eric Low, and Lloyd Sansom. 2017. Why patients should be involved in health technology assessment. *International Journal of Technology Assessment in Health Care* (2017), 1–4.
- Aosen Wang, Feng Lin, Zhanpeng Jin, and Wenyao Xu. 2016. Ultra-low power dynamic knob in adaptive compressed sensing towards biosignal dynamics. *IEEE Transactions on Biomedical Circuits and Systems* 10, 3 (2016), 579–592.
- Aosen Wang, Chen Song, Xiaowei Xu, Feng Lin, Zhanpeng Jin, and Wenyao Xu. 2015. Selective and compressive sensing for energy-efficient implantable neural decoding. In *Proceedings of the 2015 IEEE Biomedical Circuits and Systems Conference (BioCAS'15)*. IEEE, 1–4.
- Ning Wang, Geoff V. Merrett, Robert G. Maunder, and Alex Rogers. 2013. Energy and accuracy trade-offs in accelerometry-based activity recognition. In *Proceedings of the 2013 22nd International Conference on Computer Communications and Networks (ICCCN'13)*. IEEE, 1–6.
- Yael Weiss, Yuval Fledel, Yuval Elovici, and Lior Rokach. 2010. Cost-sensitive detection of malicious applications in mobile devices. In *Proceedings of the International Conference on Mobile Computing, Applications, and Services*. Springer, 382–395.
- James Williamson, Qi Liu, Fenglong Lu, Wyatt Mohrman, Kun Li, Robert Dick, and Li Shang. 2015. Data sensing and analysis: Challenges for wearables. In *Proceedings of the 2015 20th Asia and South Pacific Design Automation Conference (ASP-DAC'15)*. IEEE, 136–141.
- Charles Worringham, Amanda Rojek, and Ian Stewart. 2011. Development and feasibility of a smartphone, ECG and GPS based system for remotely monitoring exercise in cardiac rehabilitation. *PLoS One* 6, 2 (2011), e14669.
- Ümit Yüceer. 2002. Discrete convexity: Convexity for functions defined on discrete spaces. *Discrete Applied Mathematics* 119, 3 (2002), 297–304.

Received October 2017; revised March 2018; accepted March 2018