Energy-Efficient Context Classification With Dynamic Sensor Control

Lawrence K. Au, *Student Member, IEEE*, Alex A. T. Bui, *Member, IEEE*, Maxim A. Batalin, *Member, IEEE*, and William J. Kaiser, *Senior Member, IEEE*

***Abstract*—Energy efficiency has been a longstanding design challenge for wearable sensor systems. It is especially crucial in continuous subject state monitoring due to the ongoing need for compact sizes and better sensors. This paper presents an energy-efficient classification algorithm, based on partially observable Markov decision process (POMDP). In every time step, POMDP dynamically selects sensors for classification via a sensor selection policy. The sensor selection problem is formalized as an optimization problem, where the objective is to minimize misclassification cost given some energy budget. State transitions are modeled as a hidden Markov model (HMM), and the corresponding sensor selection policy is represented using a finite-state controller (FSC). To evaluate this framework, sensor data were collected from multiple subjects in their free-living conditions. Relative accuracies and energy reductions from the proposed method are compared against naïve Bayes (always-on) and simple random strategies to validate the relative performance of the algorithm. When the objective is to maintain the same classification accuracy, significant energy reduction is achieved.**

***Index Terms*—Classification, energy efficiency, hidden Markov model, optimization, partially observable Markov decision process, sensor selection, wearable platform.**

I. Introduction

WEARABLE technology has advanced rapidly in the past decade due to low-cost wireless and sensor electronics. It has increasingly found uses in healthcare applications such as Parkinson's disease management [1], chronic obstructive pulmonary disease (COPD) [2], neurorehabilitation [3], and general health monitoring [4]—most of which require continuous subject monitoring. Furthermore, for many hidden medical conditions, symptoms are often missed during traditional clinical visits; and cumulative, free-living monitoring of physical activities with wearable systems has shown to be a major clinical outcome for patients with chronic diseases [5], [6].

Designing a wearable system for continuous activity monitoring entails multiple conflicting requirements—functionality, package size, battery capacity, and connectivity [7]–[9]. For instance, the combination of sensor data acquisition and signal processing algorithms must provide adequate information that describe the patients, but the wearable system must be lightweight enough to be worn in real life.

Among all the design requirements, energy efficiency has been a major design consideration of wearable platforms [10], [11]. Battery life is often severely limited in wearable applications, yet constant classification and wireless sensing induce significant energy consumption. Many existing applications do not address this problem directly because the length of their experiments and pilot studies are relatively short and the depicted situations are often controlled. Such scenarios may not be applicable in practice: battery energy can be completely depleted within hours if not managed properly [12], and recharging is often unavailable while the systems are in use. Consequently, more energy-efficient techniques are needed to manage battery use while minimizing misclassification costs.

Efficient use of battery energy often depends on applications and available sensors. Often times, not all sensors are needed to recognize certain activities (e.g., running, resting, and walking) [13]. Intelligently activating sensors and other resources on-demand to reduce energy consumption becomes more important. In traditional and mobile computing, a timeout policy is a simple technique for minimizing energy consumption. These techniques, however, often focus on minimizing energy consumption without much control of performance constraints [4], [15]. For wearable platforms, there should be a method to gauge the amount of *useful information* collected for a given energy constraint, because classification accuracy is the ultimate measure of relative performance.

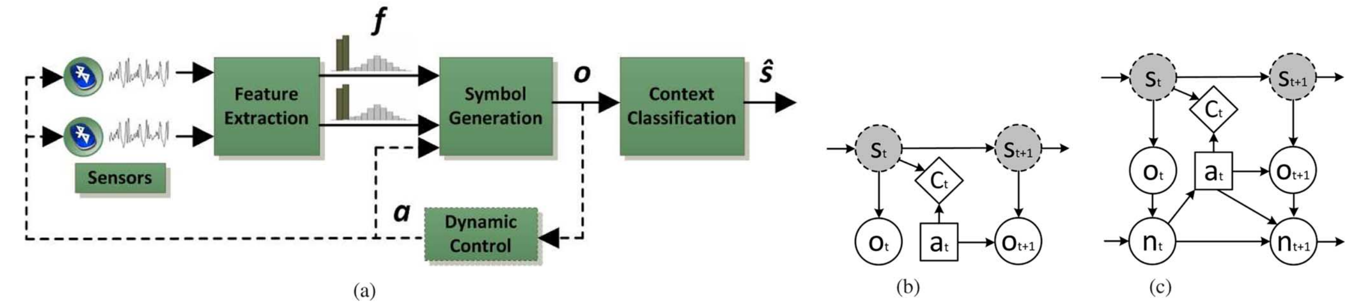
Leveraging *context awareness* has become a viable solution that can help alleviate some of the sensing requirements in continuous monitoring [16]. For instance, Wu *et al.* shows how a hierarchy of sensors is used by keeping track of heart rate variability after exercising through motion sensing [17]. Cakmakci *et al.* [18] uses the IBM WristWatch [19] to demonstrate energy reduction through context awareness (with an accelerometer). Krause [20] and Wang [21], [22] studied the tradeoff between prediction accuracy and power consumption in mobile sensing under a Markov framework. Other studies that discuss similar design processes and tradeoffs include acoustic sensing [23], the autonomous sensor button [24], MEDIC [25], the Smart Module [26], KNOWME system [11], and EEMSS [27]. In many of these examples, the key is to recognize the next best course of action (e.g., activating an ECG sensor) given a particular context (e.g., running) to minimize unnecessary sensor use. The proposed solution leverages context awareness by modeling the dependencies between successive contexts: at a particular point

Manuscript received March 29, 2011; revised June 10, 2011; accepted July 24, 2011. Date of publication October 10, 2011; date of current version April 19, 2012.This work was supported by the National Institutes of Health/National Library of Medicine under Grant NLM T15 LM07356.

The authors are with the Electrical Engineering Department, University of California, Los Angeles, Los Angeles, CA 90095 USA (e-mail: au@ucla.edu; buia@mii.ucla.edu; maxim.batalin@gmail.com; kaiser@ee.ucla.edu)

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TBCAS.2011.2166073



(a) (b) (c)

Fig. 1. System description and influence diagrams. (a) Dynamic sensor control. (b) POMDP. (c) FSC-POMDP.

in time, some contexts are stochastically more likely than others [20]. The main intuition to energy-efficient sensing is to activate sensors episodically (e.g., [28]). This paper proposes a decision-theoretic approach to *select sensor resources for accurate context classification, while keeping the average energy consumption below some budget*. This constrained optimization problem is formulated under the framework of a Partially Observable Markov Decision Process (POMDP). POMDP models the context transitions as a Markov process. Each Markov state generates an observation from sensors via feature extractions. Examples of POMDP include medical diagnosis, decision making, and assistive technology [29]–[31].

II. System Description

The architecture of the proposed framework is illustrated in Fig. 1(a), which resembles a general feedback control system. At each time step (or *epoch*), the system acquires *sensor data* and processes the data through *feature extraction* and *symbol generation*. The output from the processing serves as the input to (a) context classification, and (b) a dynamic control block. *Context classification* computes the most likely state at the current epoch t. The *dynamic control* block generates a sensor control signal that adjusts the operating state of the sensors. The objective is to reach a desired condition in the long run (e.g., minimizing average misclassification cost as $t \rightarrow \infty$). Note that the control policy associated with the dynamic control is computed offline; its execution does not induce significant runtime costs. The following sections discuss each of the major blocks in detail.

A. Sensors and Feature Extraction

First, sensor data are acquired at a sufficient sampling rate. They are then used to generate distinctive *features* for classification. Most feature extraction algorithms select a specific processing window size (e.g., four seconds) to facilitate data analyses; one feature is extracted in every window.

During training data collection, the range of values of each feature is discretized into a finite number of intervals (or *bins*), essentially performing Gaussian cluster discretization [13]. Thus, the true feature value falls into exactly one of these intervals. The number of intervals for each feature is determined primarily using available training data. Such a discretization greatly facilitates supervised learning and computations related to inference [13].

Denote the value $f^{(i)}$ as the f-th bin of the i-th feature; that is, $f^{(i)}$ represents the index of the interval enumeration in the feature discretization step described above. For example, if a feature takes on a Gaussian distribution that is discretized into m bins, then $f^{(i)} \in \{0, 1, \ldots, m\} = \mathbb{F}^{(j)}$. Multiple features can be vectorized (i.e., a *feature vector*), $\mathbf{f} = (f^{(1)}, f^{(2)}, \ldots)$. The *feature space*, $\mathbb{F}$, corresponds to the set of all possible values $\mathbf{f}$ can take on.

B. Observation Symbol Generation

In many applications, the cardinality of $\mathbb{F}$, $|\mathbb{F}|$, can become large. Symbol generation is a method for reducing the cardinality by enumerating $\mathbb{F}$ into a finite number of subsets, with each subset mapped to a numerical observation symbol. Denote $F^{(j)} \subset \mathbb{F}$ as the j-th subset. The process of symbol generation maps each of the $F^{(j)}$ to the respective observation symbol $o = j \in \{0, 1, \ldots\} = \Omega$. Furthermore, $\cup_j F^{(j)} = \mathbb{F}$ and $F^{(j)} \cap F^{(k)} = \{\emptyset\}, \quad \forall j \neq k$. The two conditions ensure that every $\mathbf{f} \in \mathbb{F}$ maps to exactly one observation symbol $o \in \Omega$. For example, one can map each possible $\mathbf{f} \in \mathbb{F}$ to a unique observation symbol.

C. Context Classification and Dynamic Control

The observation symbol $o \in \Omega$ is used to (a) compute the state classification output, and (b) issue a sensor control signal to be executed in the next epoch. Classification involves state inference; it outputs the estimated state, $\hat{s}$. Furthermore, based on o, the dynamic control system generates a sensor control signal, a, which determines the availability of different sensors in the next epoch. Both classification and dynamic control are elaborated upon in the subsequent sections.

III. Theoretical Background

POMDP is a discrete-time, controlled hidden Markov model (HMM) where the outcomes are partly stochastic and partly under user control [32]. Note that in the proposed framework, the (hidden) state transitions are *undisturbed* by the actions; that is, the system does not have control over user activities.

Definition 1: A POMDP can be defined as

$\mathbb{S}$: state space
$\mathbb{A}$: action space
Ω: observation space

$T_{ss'} = P(s' \mid s)$: state-transition probability to next state $s' = s_{t+1} \in \mathbb{S}$, given present state $s = s_t \in \mathbb{S}$

$O_{s'ao'} = P(o' \mid s', a)$: observation probability of symbol $o' = o_{t+1} \in \Omega$, given state $s' = s_{t+1} \in \mathbb{S}$ and action $a = a_t \in \mathbb{A}$

$c^{(i)}(s, a)$: the i-th cost function, given present state $s = s_t \in \mathbb{S}$ and action $a = a_t \in \mathbb{A}$

The *state space* $\mathbb{S}$ contains all the states of interest (e.g., user activities), which is usually defined by domain experts based on the application. The *action space* $\mathbb{A}$ comprises the set of all available sensor control signals (e.g., activate/deactivate sensors). Observation space Ω contains the set of all possible observation symbols as extracted from the feature space $\mathbb{F}$. The state-transition probability, $P(s' \mid s)$, models the evolution of the states. The observation probability, $P(o' \mid s', a)$, represents the likelihood of observing o', given state s' and action a. The observation probability is a function of a because the features (and hence symbol) obtained depends on the data acquired from sensors. The i-th cost function, $c^{(i)}(s, a)$, is used to associate the immediate cost of selecting action a when the state is s. The relationships between the POMDP variables are illustrated graphically in an influence diagram [Fig. 1(b)].

A. Objective Function

In POMDP problems, the objective is to search for a sequence of actions (a *policy*), $\pi = (a_0, a_1, \ldots, a_T)$, that minimizes the cost function. In continuous monitoring, one is more concerned with the *long-term average cost per epoch*. If there is only one cost function, it implies searching for a policy π that minimizes the following *average cost criterion*:

$$C_\pi(s_0) = \lim_{T \to \infty} \frac{1}{T} \cdot \mathcal{E}_\pi \left[\sum_{t=0}^{T} c_\pi(s_t) \middle| s_0 \right] \tag{1}$$

where $\mathcal{E}[\cdot]$ denotes the expectation with respect to the policy π, and $c_\pi(s_t)$ represents the immediate cost accrued at time t with state s, with initial state s_0. Note that (1) is evaluated with a *fixed* policy; therefore, action a_t is embedded in the cost function.

B. Policy Approximation Using Finite-State Controller

Solving exactly for the best policy of a general POMDP is intractable due to hidden state space $\mathbb{S}$ [33]. Consequently, selecting the best actions may require the complete history of the action-observation sequence, $(a_0, o_1; a_1, o_2; \ldots)$, which potentially leads to an unbounded amount of memory for decision making [32]. POMDP can be converted to a continuous-state Markov Decision Process (MDP), but the average cost criterion with the equivalent continuous-state variable may not be solved easily [34].

This paper proposes a sub-optimal approach that approximates the policy as a finite-state controller (FSC), and the objective is to search for the optimal policy within this policy representation [34]. The approximation is achieved by discretizing the continuous-state space in the equivalent MDP problem into multiple regions (or *nodes*). Each of the nodes in this *node space* may transition to another node, guided by the observation symbol. By adjusting the number of nodes, one can achieve any arbitrary (and near-optimal) control policy with sufficient memory [34], [35].

Definition 2: A (stochastic) finite-state controller (FSC) can be defined as a policy $\pi = \{\mathbb{N}, \psi, \eta\}$, where

$\mathbb{N}$: node space

$\psi_{na} = P(a \mid n)$: probability of selecting action $a = a_t \in \mathbb{A}$ at node $n = n_t \in \mathbb{N}$

$\eta_{nao'n'} = P(n' \mid n, a, o')$: probability of transitioning from node n to node $n' = n_{t+1}$, with $a \in \mathbb{A}$ and $o' = o_{t+1} \in \Omega$

FSC behaves as a finite-state machine. It issues an action a based on $P(a \mid n)$, and the control action leads to an observation symbol o'. FSC determines the new node n' based on the node-transition probability $P(n' \mid n, a, o')$.

By using a finite-state controller, one can modify the policy based on additional information from the problem (e.g., stochastic versus deterministic actions) to limit the search space. Because all variables are discrete, FSC can be implemented as a lookup table. Most importantly, the joint FSC-POMDP model can be analyzed as a finite-state Markov chain with respect to the new discrete state variable$\bar{s} = \langle n, s \rangle$ [Fig. 1(c)] [34], [35].

IV. Problem Formulation

This section describes the method to acquire the relevant model parameters and the joint FSC-POMDP model. Once the joint model is formed, a corresponding optimization problem is formulated to solve for the control policy.

A. POMDP

1) Training: The state-transition and observation probabilities for POMDP can be obtained using the methods in HMM learning. An HMM can be defined as

$\mathbb{S}$: user contexts (activities)

Ω: observation symbols

$P(s' \mid s)$: state-transition probabilities of reaching next state $s' = s_{t+1} \in \mathbb{S}$, given present state $s = s_t \in \mathbb{S}$

$P(o \mid s)$: probability of observing $o = o_t \in \Omega$ given $s = s_t \in \mathbb{S}$

Suppose there are m features in $\mathbf{f}$. Note that, from the definition of an observation symbol in Section II-B,

$$\begin{aligned} P(o_t = i \mid s_t = s) &= \sum_{\mathbf{f} \in F^{(i)}} P(\mathbf{f} \mid s) \\ &= \sum_{\mathbf{f} \in F^{(i)}} P\left(f^{(1)}, \ldots, f^{(m)} \middle| s \right) \end{aligned} \tag{2}$$

Generally, the conditional joint probability in (2) is estimated through supervised training for all states (similar to the maximum likelihood estimation used in obtaining HMM parameters). If conditional independence is assumed among all the features, then, the joint probability becomes

$$P(o_t = i \,|\, s_t = s) = \sum_{\mathbf{f} \in F^{(i)}} \left[\prod_{j=1}^{m} P\left(f^{(j)} \middle|\, s \right) \right]. \quad (3)$$

The independence assumption reduces the amount of training data needed for parameter estimation. With supervised training, the state-transition and observation probabilities can be computed with annotated data. Specifically, $P(s' \,|\, s)$ and $P(f^{(i)} \,|\, s)$ [using (3)] can be computed using maximum likelihood estimation (i.e., counting) [36]:

$$P(s_{t+1} \,|\, s_t) = \frac{\mathrm{count}(\{s_t = s\} \bigcap \{s_{t+1} = s'\})}{\mathrm{count}(s_t = s)}, \quad \forall t$$
$$P\left(f_t^{(j)} \middle|\, s_t \right) = \frac{\mathrm{count}\left(\left\{ f_t^{(j)} = f^{(j)} \right\} \bigcap \{s_t = s\} \right)}{\mathrm{count}(s_t = s)}, \quad \forall t, j.$$

Furthermore, given initial $P(s' \,|\, s)$ and $P(o \,|\, s)$, an HMM can be re-estimated with new observation symbols using various (unsupervised) learning techniques such as the Baum-Welch algorithm. Because both $P(s' \,|\, s)$ and $P(o \,|\, s)$ are discretized, they can be implemented efficiently as matrices (lookup tables) in practice.

2) Model Parameters: Once the HMM parameters are obtained, it can be integrated into the POMDP model by incorporating appropriate actions and modifying the observation probabilities:

$\mathbb{S}$:	$\mathbb{S}$ from HMM
$\mathbb{A}$:	$\{\mathrm{deactivate,\ activate}\}$
Ω:	Ω from HMM
$P(s' \,\|\, s)$:	$P(s' \,\|\, s)$ from HMM
$P(o' \,\|\, s', a)$:	sensor-dependent observation probability
$c(s, a)$:	misclassification cost
$e(s, a)$:	energy cost

The derivation of $P(o' \,|\, s', a)$ requires further explanation. Control action a has a direct influence on both feature extraction and symbol generation; namely, a controls the availability of sensor data [Fig. 1(a)], and hence both $\mathbf{f} \in \mathbb{F}$ and $o \in \Omega$. In this paper, two control actions are defined: *activate* and *deactivate* all sensors. More control actions can be incorporated; for example, $\mathbb{A}$ can include all possible on/off combinations of the available sensors.

Similar to the definition in (2), one can define $P(o'|s', a)$ as follows:

$$\begin{aligned} P(o_{t+1} &= i \,|\, s_{t+1}, a_t) \\ &= P(o' \,|\, s', a) \\ &= \sum_{\mathbf{f} \in F^{(i)}} \left[\prod_{j=1}^{m} P\left(f_{t+1}^{(j)} \middle|\, s_{t+1}, a_t \right) \right] \end{aligned} \quad (4)$$

$$\begin{aligned} P&\left(f_{t+1}^{(j)} \middle|\, s_{t+1}, a_t \right) \\ &= \begin{cases} P\left(f^{(j)} \middle|\, s \right) & \text{if } a \text{ enables } f^{(j)} \\ 1/|\mathbb{F}^{(j)}| & \text{if } a \text{ disables } f^{(j)}. \end{cases} \end{aligned} \quad (5)$$

That is, if control action a powers on the sensor that generates $f^{(j)}$, the respective conditional probability is the same probability distribution as derived from HMM training. When a disables the sensor, no additional information is obtained from the sensor. Thus, the corresponding probability distribution is uniform.

3) Misclassification Cost: The objective of our problem is to minimize the cost of misclassifications. A cost function is defined so that it corresponds to the cost (or probability) of misclassifications. For instance, this function can be manually defined by a domain expert who determines the cost of misclassification for each $c(s, a)$ [13]. This paper proposes a (myopic) cost metric using the parameters from the POMDP model. Given $s_t = s$, let the probability of event $\{(S_{t+1} = s') \cup (O_{t+1} = o')\}$ be $P(s', o' \,|\, s)$. This quantity represents the *conditional likelihood* of observing o' and the hidden state is s'. If one estimates S_{t+1} using this measure, then the cost of misclassification can be defined as

$$\begin{aligned} c(s, a) &= 1 - \max_{s', o'} P(s' \,|\, o', s, a) \\ &= 1 - \frac{\max_{s', o'} P(s', o' \,|\, s, a)}{P(o'^{*} \,|\, s, a)} \\ &= 1 - \frac{\max_{s', o'} \left[P(o' \,|\, s', a) \cdot P(s' \,|\, s) \right]}{P(o'^{*} \,|\, s, a)} \end{aligned} \quad (6)$$

where $o'^{*} = \arg\max_{o'} \{ \max_{s'} P(s', o' \,|\, s, a) \}$.

$c(s, a)$ is mainly influenced by the joint probability $P(s', o' \,|\, s)$. The maximum joint probability is then normalized by the marginal probability $P(o'^{*} \,|\, s)$ to obtain the posterior probability (and hence the cost function).

4) Energy Cost: An energy cost function associates each control action a to the corresponding energy consumption of each sensor. For instance, the energy cost can indicate the sensing and transmission energy. Consequently, multiple energy cost functions may exist for multiple sensors. In general, the energy cost function of the i-th sensor can be defined as

$$\begin{aligned} &\mathrm{energy}^{(i)}(s, a) \\ &\quad = \mathrm{energy}^{(i)}(a) \\ &\quad = (\text{energy consumption of the } i\text{-th sensor} \\ &\qquad \text{with } a \in \mathbb{A} \text{ in one epoch}). \end{aligned}$$

Only two control signals are considered: activate/deactivate all sensors. Because the sensor set is homogeneous, only one energy cost function needs to be defined. In practice, a heterogeneous set of sensors requires different energy cost functions, and energy consumption for every $a \in \mathbb{A}$ can be pre-computed from offline measurements.

To make the energy cost functions consistent across all sensors, define a normalized energy cost function:

$$e^{(i)}(s,a) = \frac{\text{energy}^{(i)}(a)}{\max\limits_{a' \in \mathbb{A}} \text{energy}^{(i)}(a')} \tag{7}$$

for the i-th sensor, where $e^{(i)}(s,a) \in [0,1]$. For example, if each sensor has only on/off states, then $e^{(i)}(s,a) = 1.0$ when a activates the i-th sensor, and $e^{(i)}(s,a) = 0.0$ otherwise. If a sensor has multiple operating states, then $e^{(i)}(s,a) \in [0.0, 1.0]$. The normalized energy cost function is used in the optimization problem formulation.

B. FSC

The policy π takes on the definition as mentioned in Section III-B.

1) Time Complexity: Denote the time to execute the FSC-based control policy as t_{ctrl}, then:

$$\begin{aligned} t_{\text{ctrl}}(a) &\\ &= \begin{cases} t_{\text{window}} + t_o + t_{\text{fsc}} + t_{\text{activate}}, & \text{if } a = \text{activate} \\ t_o + t_{\text{fsc}} + t_{\text{deactivate}}, & \text{if } a = \text{deactivate}. \end{cases} \end{aligned}$$

t_{window} denotes the window size used for data acquisition and feature extraction. $t_o = \mathcal{O}(1)$ represents the time required for generating an observation symbol; it is approximately constant once features are extracted. t_{fsc} is the time required to compute the new sensor control signal. Because π can be implemented as a lookup table, $t_{\text{fsc}} = \mathcal{O}(1)$.

t_{activate} and $t_{\text{deactivate}}$ denote the time required to activate and deactivate sensors respectively. In practice, the time to switch sensor states is dependent upon the actual hardware system, and can be assumed to be constant with respect to π and the amount of sensor data. Thus, the time complexities of t_{activate} and $t_{\text{deactivate}}$ are $\mathcal{O}(1)$. The overall time complexity of executing the control policy in an epoch is:

$$\mathcal{O}(t_{\text{ctrl}}) = \begin{cases} \mathcal{O}(t_{\text{window}}), & \text{if } a = \text{activate} \\ \mathcal{O}(1), & \text{if } a = \text{deactivate}. \end{cases}$$

2) Space Complexity: Using the standard big-$\mathcal{O}$ notation, the memory requirement of ψ is $\mathcal{O}(|A||\mathbb{N}|)$. The memory requirement of η is $\mathcal{O}(|\Omega||A||\mathbb{N}|^2)$. Therefore, the space complexity of implementing the control policy π is $\mathcal{O}(|\Omega||A||\mathbb{N}|^2)$.

C. Joint Model

Given a policy, integrating FSC into POMDP results in a finite-state Markov Chain (with respect to $\bar{s} = \langle n, s \rangle$), which is defined as follows [37]:

$$\begin{aligned} &\bar{\mathbb{S}} = \mathbb{N} \times \mathbb{S}: && \text{the set of all } \langle n, s \rangle \text{ pairs} \\ &\bar{\mathbf{T}} = \bar{P}_\pi(\bar{s}' \mid \bar{s}): && \sum_{a,o'} P(a \mid n) P(n' \mid n, a, o') P(o' \mid s', a) P(s' \mid s) \\ &\bar{c}_\pi(\bar{s}): && \sum_{a \in \mathbb{A}} P(a \mid n) c(s,a) \end{aligned}$$

Note that the FSC parameters (ψ and η) are coupled with the original POMDP with this definition. The final control policy π is then solved using these parameters.

D. Context Classification

It has been shown that when updated properly, the *belief state* is a sufficient statistic for context classification [32], [38]. The belief state, $\mathbf{b}$, represents the probability mass function of all states $s \in \mathbb{S}$; that is, $\mathbf{b} = (P(s=0), P(s=1), \ldots)$. As with any probability function, $\mathbf{b}$ satisfies the laws of probability.

Given the present belief state, $\mathbf{b} = \mathbf{b}_t$, the new belief state, $\mathbf{b}' = \mathbf{b}_{t+1}$, can be computed using Bayes rule for estimating *posterior* probability [32]:

$$\begin{aligned} b'(s') &= b'(s'|o', \mathbf{b}, a) \\ &= \frac{P(o'|s', a) \cdot \sum_s P(s'|s) \cdot b(s)}{P(o'|\mathbf{b}, a)}. \end{aligned} \tag{8}$$

In (8), the present belief state, $b(s)$, is used iteratively to compute the new belief state, $b'(s')$. The initial belief state, $\mathbf{b}_0$, can be initialized as a uniform distribution with $b_0(s) = 1/|\mathbb{S}|$ for all s. After initialization, $b(s)$ is computed iteratively. The denominator, $P(o' \mid a, \mathbf{b})$, is the normalization factor that ensures $\mathbf{b}'$ remains a probability distribution:

$$P(o' \mid \mathbf{b}, a) = \sum_{s'} P(o' \mid s', a) \cdot \sum_s P(s' \mid s) \cdot b(s).$$

The context classifier estimates the hidden state $\hat{s}_{t+1}$ by selecting the largest $b'(s')$: $\hat{s}_{t+1} = \arg\max_{s'} b'(s')$.

1) Complexity: Assume that algebraic additions, multiplications, and divisions can execute in constant time $\mathcal{O}(1)$ (independent of the POMDP parameters). The algorithmic complexity of evaluating $\mathbf{b}'$ in real time is equal to $\mathcal{O}(|\mathbb{S}|^2)$, from the definition of (8).

E. Computing Control Policy

The long-term average cost formulation in (1) is an unconstrained problem. For continuous monitoring, the wearable platform wants to limit energy consumption below a certain budget. Literature related to solving such problems include [37], [39]–[41]. The proposed solution technique solves the optimization problem, given a fixed-$|\mathbb{N}|$ FSC.

Solving the Bellman equations exactly with average cost criterion is intractable, and a closed-form solution may not exist. Taking advantage of the problem structure is crucial. By construction, the finite-state Markov chain, $\bar{\mathbf{T}}$, must contain a stationary distribution (asymptotic state occupancy distribution) to have a feasible solution. This distribution can be considered as the long-term state probability distribution when the process runs for a long time. The optimization problem searches for the solution directly by enforcing sufficient conditions for a unique stationary distribution. Given that there exists a unique stationary distribution and bounded costs for all $\bar{s} \in \bar{\mathbb{S}}$, (1) is

TABLE I
FSC-POMDP OPTIMIZATION

variables	$\eta, \psi, \mathbf{p}_\infty$
min	$C = \sum_{n\in\mathbb{N}} \sum_{s\in\mathbb{S}} p_\infty(n,s)\bar{c}(n,s)$
subject to	Energy Cost Constraints
	$\sum_{n\in\mathbb{N}} \sum_{s\in\mathbb{S}} p_\infty(n,s)\bar{e}(n,s) \leq E_{max}$
	Stationary Distribution Constraints
	$\bar{T}_{\bar{s}\bar{s}'} = P(\bar{s}'\|\bar{s}) = \sum_{a,o'} P(a\|n)P(n'\|n,a,o')P(o'\|s',a)P(s'\|s)$
	$\mathbf{p}_\infty^\top \bar{\mathbf{T}} = \mathbf{p}_\infty^\top \quad \forall n \in \mathbb{N}, s \in \mathbb{S}$
	Probability Constraints
	$\sum_{n'} P(n'\|n,a,o') = 1 \quad \forall n,a,o'; \quad P(n'\|n,a,o') \geq 0, \forall n',a,n,o'$
	$\sum_a P(a\|n) = 1 \quad \forall n; \quad P(a\|n) \geq 0, \quad \forall a,n$
	$\sum_{\bar{s}} p_\infty(\bar{s}) = 1 \quad p_\infty(\bar{s}) \geq 0, \quad \forall \bar{s} \in \bar{\mathbb{S}}$

independent of initial state probability distribution $\mathbf{p}_{t=0}$, and is equal to

$$C_\pi = \sum_{\bar{s}} p_\infty(\bar{s}) \cdot \bar{c}_\pi(\bar{s}) = \sum_{n\in\mathbb{N}} \sum_{s\in\mathbb{S}} \left[p_\infty(n,s) \cdot \sum_{a\in\mathbb{A}} P(a|n)c(s,a) \right] \tag{9}$$

where $\mathbf{p}_\infty = (p_\infty(\bar{s} = 0), p_\infty(\bar{s} = 1), \ldots, p_\infty(\bar{s} = |\mathbb{N}||\mathbb{S}| - 1))$ is a column vector that represents the probability mass function of the stationary distribution [34].

The complete optimization problem is shown in Table I. The variables to be solved are the policy (η and ψ) and the stationary distribution $\mathbf{p}_\infty$. The objective function is to minimize the average misclassification cost, and it is defined using (9). The optimization problem is subject to: (a) average energy cost constraint; (b) the stationary distribution constraint; and (c) probability constraints. $E_{\max}$ is the maximum energy budget of the sensors, and is assumed to be constant (and thus independent of initial state probability distribution [34]). $E_{\max} = 0.0$ implies sensors are always off, and $E_{\max} = 1.0$ always on. Thus, $E_{\max}$ may be viewed as the *average duty cycle* of the sensors in the problem definition.

Stationary distribution constraints are enforced in order to guarantee the final solution always reaches a stationary distribution for any control policy (for some $\mathbf{p}_\infty \neq \mathbf{0}$). Specifically, the stationary distribution must satisfy $\mathbf{p}_\infty^\top \bar{\mathbf{T}} = \mathbf{p}_\infty^\top$, where $\mathbf{p}_\infty^\top$ denotes the *transpose* of $\mathbf{p}_\infty$. The probability constraints ensure η, ψ and $\mathbf{p}_\infty$ satisfy probability distributions.

The formulation in Table I is, in general, a quadratically-constrained quadratic program (QCQP). There are efficient solution techniques for solving such problems. In this paper, solutions are obtained using the SNOPT solver [42].

V. Experimental Setup

In the proposed application, the MicroLEAP wearable platform is used for motion data collection [43]. Each subject is asked to wear two MicroLEAPs (at waist level and ankles) and a portable video camera for annotating the ground truth. Data from MicroLEAPs (accelerometer data) are collected for analysis. In the experiments, accelerometer data are acquired at 128 Hz.

Each subject was instructed to carry out free-living activities during (working) day time; and at least three working days (24 hours) of training data are collected for each subject. The length and times of these activities were intentionally left to the discretion of subjects. Initial values for the HMMs are obtained using previously obtained models with ground truth, and state-transition and observation probabilities are re-learned using the Baum-Welch algorithm on training data. Testing data is collected on another working day; ground truth is captured by the portable video camera. All sensor data were synchronized in each session and manually annotated. For each subject, the MicroLEAP that provides the best feature/observation probabilities is selected as the sensor for training and testing.

Features are extracted from the time-series accelerometer data, $y(t)$. One feature value is computed from a four-second window (512 samples). The feature used is the total energy contained in a fixed window (either in time or frequency domain)

$$|Y(\omega)| = \text{total energy in } \mathcal{F}\{y(t)\}$$
$$|y(t)| = \text{total energy in } y(t).$$

Feature discretization and symbol generation are done using the method described in Section II.

Two states (contexts) are incorporated into the model (i.e., $\mathbb{S} = \{\text{static, moving}\}$). *static* represents minimal body movements, while *moving* may include walking and possibly other non-periodic body movements. Because only one sensor is used, $\mathbb{A} = \{\text{deactivate, activate}\}$. Note that the observation space is $\Omega = \{0, 1\}$; they correspond to the indices from the symbol generation. Finally, the control policy is defined as $\pi = \{|\mathbb{N}| = 2, \psi, \eta\}$.

VI. Results

In this section, both simulation and experimental results of a two-state model are illustrated. Specifically, the relative improvement in classifier accuracies (with respect to the POMDP parameters) is studied. A policy comparison is also performed to show the advantage of the FSC-based control policy. Finally, a multi-state model example is provided to show the practicality of the proposed algorithm.

A. Simulation Analysis

Fig. 2(a) and (b) illustrate the effects on classifier accuracy as a function of T. Plots are generated from simulations by executing the policy π on a long sequence of symbols simulated by the underlying HMMs as a function of energy budget.

First, note that the state-transition matrix affects the classifier accuracy. By analyzing with two states, one can gain some intuition as to how the model parameters affect the classification accuracy. For now, assume T is symmetric, and O is equal to $\begin{pmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{pmatrix}$. Obtaining $P(o\,|\,s)$ is often under user control; it can be improved with more accurate feature extraction algorithms. Because T is symmetric, its stationary distribution is uniform ($p_\infty(s) = 1/|\mathbb{S}|$). This is illustrated by the

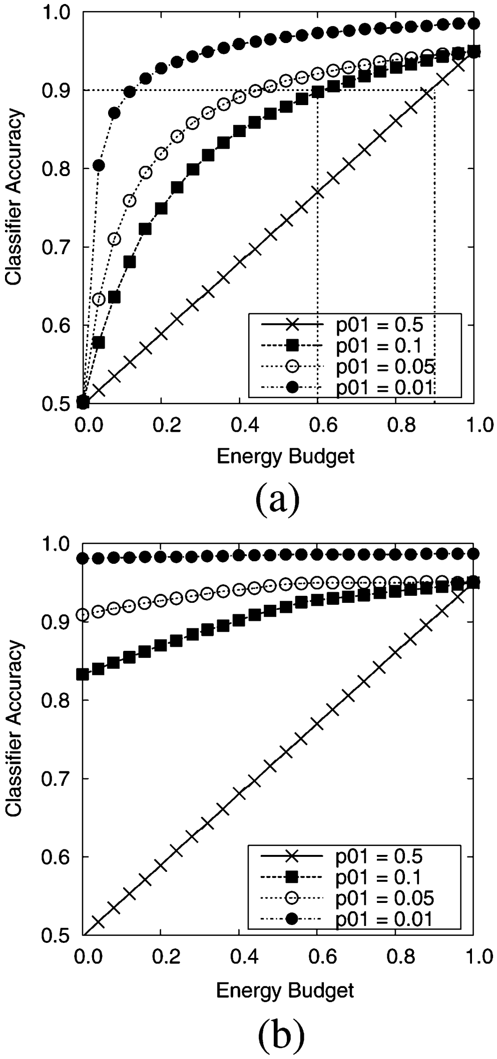


Fig. 2. Classification accuracy versus energy budget: Simulations. (a) T: Symmetric. (b) T: Asymmetric.

50% classifier accuracy with zero energy budget for the two-state problem (i.e., classification based purely on an underlying Markov model, making no additional observations).

Because T is completely parametrized by $p_{01} = P(S_{t+1} = \text{moving} \,|\, S_t = \text{static})$, classifier accuracies increase more rapidly with additional energy budget when $p_{01} \rightarrow 0$, while $p_{01} = 0.5$ increases linearly. This is because the context classifier (Bayesian belief update) takes into account both the state-transition *and* observation probabilities; the stationary distribution only determines the classifier accuracy when the sensors are unavailable. This is the inherent advantage of incorporating the state-transition probabilities into context classification: Bayesian belief update is almost always more accurate than what a naïve Bayes classifier would provide (unless $p_{01} = 0.5$, in which case POMDP provides no additional advantage over the linear relationship provided by naïve Bayes).

The main benefit of the proposed solution is the significant energy reduction for a given classifier's accuracy. For instance, in Fig. 2(a), with a classifier accuracy of 90%, the naïve Bayes classifier requires an energy budget of 0.9 (90% average sensor duty cycle). With $p_{01} = 0.1$, the energy required is reduced to 0.6. As p_{01} approaches zero, the energy reduction becomes more significant. Also note that as $p_{01} \rightarrow 0$, the overall classifier accuracy may exceed what is feasible with observation probabilities alone (when $P(o \,|\, s) < 1.0, \quad \forall s$). This is due to the *filtering* process inherent to the context classifier (analogous to the forward algorithm [44]).

When T is asymmetric ($T = (\begin{matrix} 1 - p_{01} & p_{01} \\ 0.5 & 0.5 \end{matrix})$), more information is available regarding the stationary distribution [as indicated by the classifier accuracy with zero energy budget in Fig. 2(b)]. Specifically, the asymptotic state occupancy distribution is skewed towards one state over the other, and the classifier accuracy at zero energy budget is equal to $p_\infty(s = \text{static})$. As the energy budget increases, the overall classifier accuracy improves. Note that the marginal increase in the overall classifier accuracy is not as significant in this case because T provides *a priori* information regarding frequencies of state occupancy.

B. Analysis From Experimental Data

Fig. 3(a)–(d) show the classifier accuracies as as a function of energy budget for different subjects from experiments described in Section V. In addition to the overall accuracies, the plots show the *sensitivity* of the classification of each state (sensitivity corresponds to how well the classifier detects $S_t = i$, given the true state is i).

First, note that all of the four models from experimental data are asymmetric; this indicates that most subjects are more likely to be sedentary in real life. With zero energy budget, the dominant state ($s^{\text{dom}} = \arg\max_s p_\infty(s)$) is always classified as the true state, and the non-dominant state is always misclassified. In this case, $s^{\text{dom}} = \text{static}$ for all subjects. As energy budget increases, the control policy makes more frequent observations to states that are more difficult to detect; as a result, the sensitivity of the non-dominant state exhibits a non-linear increase in all four plots. The rapid rise of sensitivity is an expected behavior from using POMDP, as opposed to a naïve Bayes model where accuracy is linearly proportional to the energy budget. Such behavior is again attributed to the filtering provided by the classifier. Prior probabilities are no longer assumed to be uniform; events are not completely independent. Another observation is that the sensitivity rises rapidly when energy budget is close to zero; there are only marginal increases when $E_{\max} \geq 0.6$. This indicates that as the sensor is activated more often, less accuracy is gained.

Fig. 4 shows the policy comparison between the FSC-based solution from Section IV-E and a pure random sensor selection policy. The random policy selects sensor action purely based on $E_{\max}$: $P(a_t = \text{activate}) = E_{\max}$ and $P(a_t = \text{deactivate}) = 1 - E_{\max}$. As a result, the random policy is independent of $o \in \Omega, a \in \mathbb{A}$. As expected, the FSC-based policy yields classification accuracies that are as good (or often better) than the random policy across all data sets.

Table II shows the numerical results obtained from the solution on the data collected from the subjects. "Accuracy (NB)" represents the classifier accuracies obtained from using a naïve Bayes classifier. Because NB requires the sensor activation at all times (i.e., $\text{energy budget} = 1.0$), it corresponds to a point on

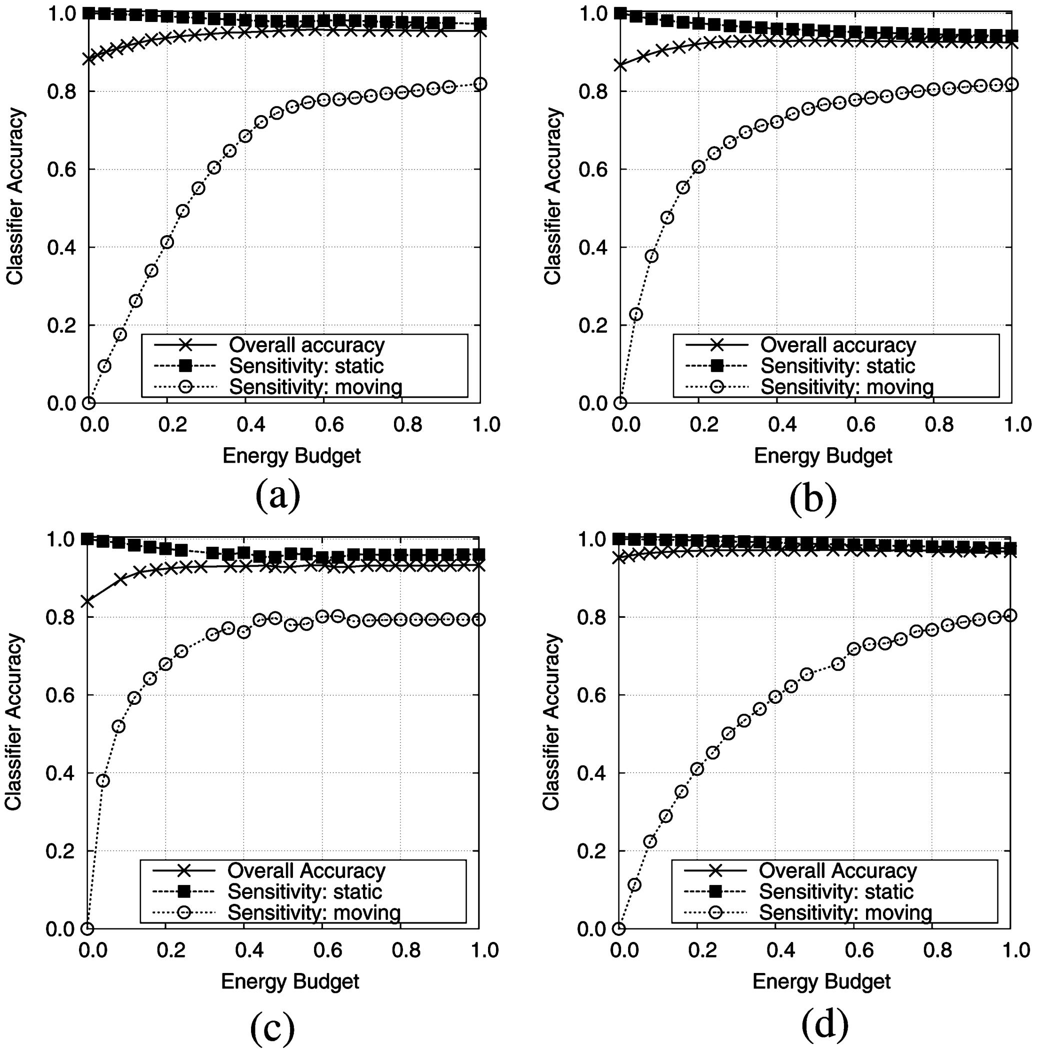


Fig. 3. Classifier accuracy versus energy budget. Subject 1: $T = \begin{pmatrix}0.979 & 0.021\\ 0.064 & 0.936\end{pmatrix}, O = \begin{pmatrix}0.939 & 0.061\\ 0.013 & 0.987\end{pmatrix}$; Subject 2: $T = \begin{pmatrix}0.989 & 0.011\\ 0.076 & 0.924\end{pmatrix}$, $O = \begin{pmatrix}0.989 & 0.011\\ 0.062 & 0.938\end{pmatrix}$; Subject 3: $T = \begin{pmatrix}0.994 & 0.006\\ 0.061 & 0.939\end{pmatrix}$, $O = \begin{pmatrix}0.992 & 0.008\\ 0.092 & 0.908\end{pmatrix}$; Subject 4: $T = \begin{pmatrix}0.980 & 0.020\\ 0.098 & 0.902\end{pmatrix}$, $O = \begin{pmatrix}0.984 & 0.016\\ 0.036 & 0.964\end{pmatrix}$. (a) Subject 1. (b) Subject 2. (c) Subject 3. (d) Subject 4.

the tradeoff curve. "Accuracy (POMDP)" represents the maximum classifier accuracy achieved by the proposed solution. To determine the energy reduction with the proposed technique, the following criteria are used: (a) the overall classifier accuracy from POMDP must be at least as high as that provided by NB, and (b) the sensitivity for each non-dominant state is within 5% of the maximum value.

The experimental data corresponds to the predicted results: a significant energy reduction is achieved. Specifically, using the performance criterion specified above, the proposed technique saves an average of 0.44 unit of energy per epoch while maintaining the specified accuracy. The FSC-based policy π determines how and when the sensors are activated to make observations. Fig. 5 shows a segment of the time-series classification result and how the Bayesian classifier operates. When the sensor is deactivated, both $b(\text{static})$ and $b(\text{moving})$ gradually drifts towards the stationary distribution as dictated by the corresponding HMM model. By making a new observation as governed by FSC, the classifier updates the belief states accordingly. The effect of a new observation depends on its relative value compared to $\mathbf{b}_t$. The key point to note is that on average, more energy (sensor resources) is allocated to detecting the non-dominant state $(S = \text{moving})$. Intuitively, the stationary distribution tilts towards the dominant state, so other states are more difficult to detect. By selecting the appropriate cost functions, energy is spent on detecting ephemeral states to maximize sensitivity.

C. *Multi-State Model*

The POMDP framework can be extended to multiple states. In this section, a simple four-state model is constructed to illustrate the efficacy of the proposed framework. Interested readers should refer to [45].

As an example, the states considered are: $\mathbb{S} = \{\text{sit}, \text{stand}, \text{walk}, \text{run}\}$. A MicroLEAP sensor is strapped on the thigh area of a test subject. Three features (mean value and variance in time domain, fundamental frequency in frequency domain [46]) are extracted and the

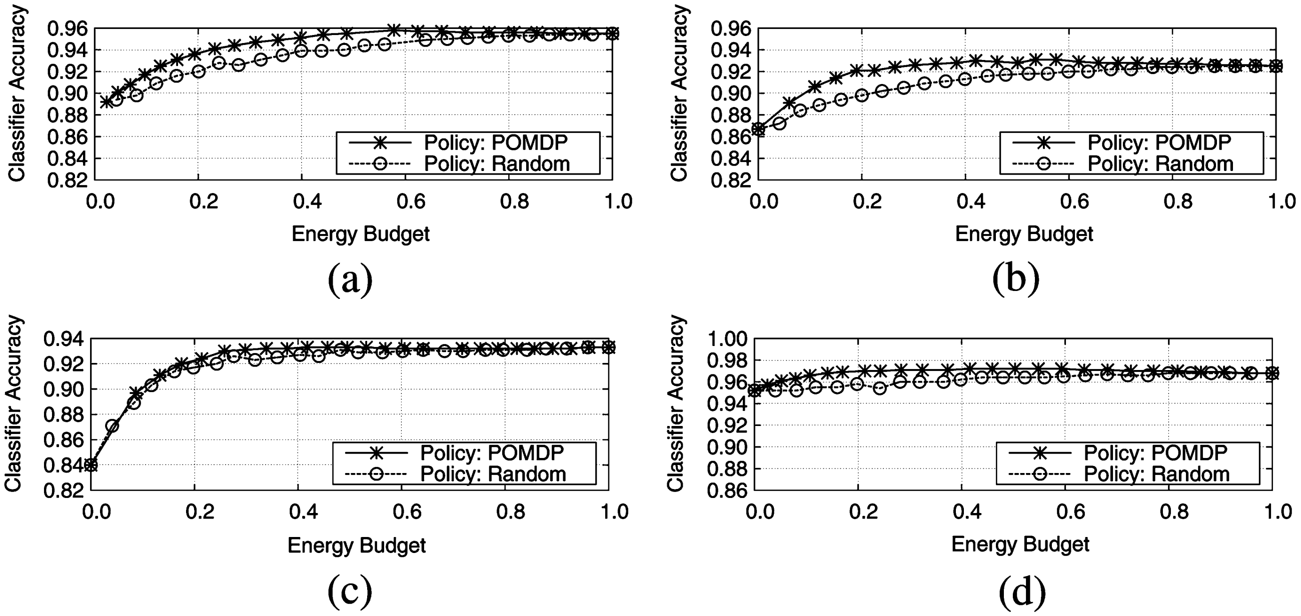


Fig. 4. FSC-based policy versus random policy. (a) Subject 1. (b) Subject 2. (c) Subject 3. (d) Subject 4.

TABLE II

NUMERICAL RESULTS: ACCURACY (NB) INDICATES THE CLASSIFICATION ACCURACY WITH NAÏVE BAYES CLASSIFIER. ACCURACY (POMDP) INDICATES THE MAXIMUM CLASSIFICATION ACCURACY ACHIEVED BY POMDP. ENERGY REDUCTION INDICATES THE CORRESPONDING REDUCTION IN ENERGY BUDGET WHEN THE ACCURACY WITH POMDP REACHES ACCURACY (NB)

Subject	Accuracy (NB)	Accuracy (POMDP)	Energy Reduction
1	0.933	0.958	0.36
2	0.913	0.932	0.48
3	0.925	0.933	0.68
4	0.968	0.972	0.24
		Average Reduction	**0.44**

corresponding symbols are generated. The model defines $|\Omega| = |\mathbb{S}|$, and $\pi = \{|\mathbb{N}| = 2, \psi, \eta\}$. To train for the HMM, the following approach is adopted. First the observation probability $P(o \,|\, s)$ is estimated using supervised training (i.e., annotated data). Because $P(o \,|\, s)$ is assumed to be time-homogeneous and the features are conditionally independent, it can be estimated with a moderate amount of data [13]. Once $P(o \,|\, s)$ is available, $P(s_{t+1} \,|\, s_t)$ can be estimated using the Baum-Welch algorithm (with $P(o|s)$ fixed) [44]. The state-transition and observation probability matrices are shown below

$$T = \begin{pmatrix} 0.973 & 0.025 & 0.001 & 0.001 \\ 0.153 & 0.817 & 0.009 & 0.021 \\ 0.001 & 0.083 & 0.883 & 0.033 \\ 0.030 & 0.166 & 0.116 & 0.688 \end{pmatrix}$$

$$O = \begin{pmatrix} 0.990 & 0.003 & 0.003 & 0.004 \\ 0.001 & 0.980 & 0.015 & 0.004 \\ 0.010 & 0.040 & 0.945 & 0.005 \\ 0.020 & 0.010 & 0.010 & 0.960 \end{pmatrix}.$$

The corresponding tradeoff plot is shown in Fig. 6(a). Fig. 6(b) shows the classifier accuracies as determined by the FSC-based policy and the random policy described previously. From the experiments, FSC performs consistently as good or better than the random sensing policy because the control action is influenced by observations. Part of the future work includes a more analytical method for determining the tradeoffs of different policy representations.

VII. DISCUSSION

Several caveats should be noted when using the POMDP framework. First and foremost, POMDP assumes the underlying context transitions obey the Markov assumption. Within the literature of context modeling, there are related applications that have been experimentally shown to obey the Markov assumption [47]–[51]. Second, POMDP assumes discrete time and that $\mathbb{S}$ and $\mathbb{A}$ are finite. Third, the *average cost criterion* ignores any near-term effects of the Markov process, which may not be appropriate for certain applications.

The memory requirement of FSC-based policy grows linearly with respect to $|\mathbb{A}|$ and $|\Omega|$; a large $|\mathbb{S}|$ does not necessarily imply a large FSC. On the other hand, the size of $|\Omega|$ can grow quickly (due to the number of symbols needed for classification). A key advantage of FSC is that both $|\mathbb{A}|$ and $|\mathbb{N}|$ are under the designer's control. For instance, $\mathbb{A}$ may include different combinations of sensor activations and de-activations. In many FSC-based applications, it has been shown that small $|\mathbb{N}|$ provides near-optimal solutions [52].

In terms of HMM development, multiple types of features (and thus observations) can be integrated into the POMDP model by assuming conditional independence to reduce the amount of training data needed [13]. The solution assumes the policy conforms to a particular representation (FSC), and the solution is derived based on that representation. Finally, the proposed solution provides a general framework for efficient context classification. Additional knowledge regarding the specific application can be used to further limit the search space of the policy representation.

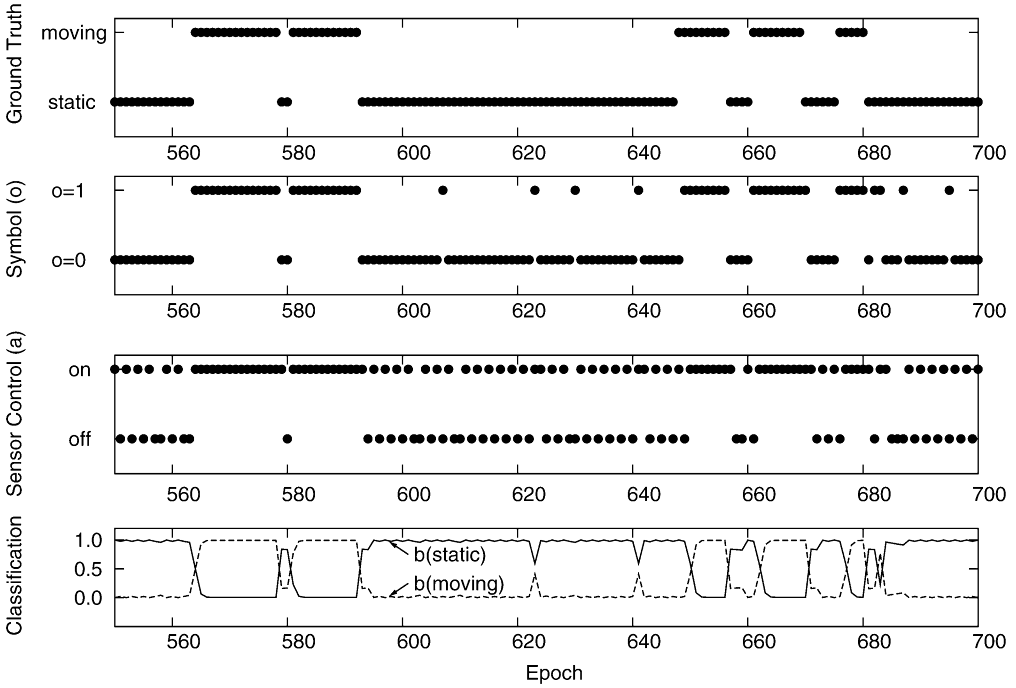


Fig. 5. Segment of temporal data with $E_{\max} = 0.60$. Top two plots indicate the ground truth and corresponding observation symbol $o \in \Omega$. The third plot shows the sensor state. The fourth plot shows the belief state of being in "static" and "moving" (classification output). Note that during the "off" period, the belief states gradually drifts towards the stationary distribution. When the system receives a new o_t during an "on" epoch, the classifier adjusts the belief values accordingly.

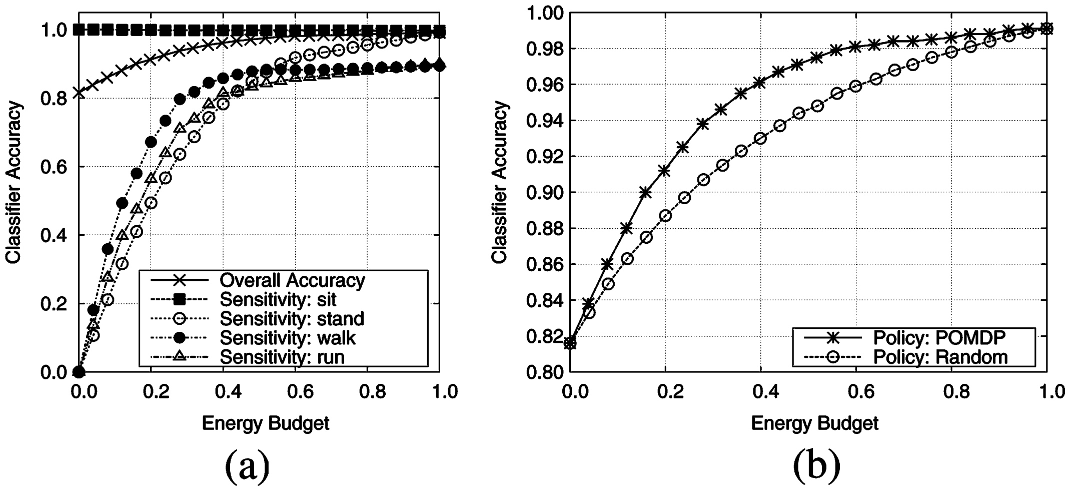


Fig. 6. Classification results with $|\mathbb{S}| = 4$. (a) Tradeoff. (b) Policy comparison.

VIII. Conclusion and Future Work

This paper presents a decision-theoretic approach to efficiently manage sensor energy for context classification under a POMDP framework. Human contexts are modeled as an HMM, and it is integrated into POMDP to take into account the effects of dynamic sensor control. The proposed solution is formulated as a constrained optimization problem, and the results demonstrate significant energy reduction with the same classification accuracy in all of the experiments. The control policy, which is represented by a finite-state controller, is practical for implementation in embedded devices because of its low time and space complexities. The control policy makes observations through the sensors and dynamically adjusts the use of sensors. POMDP takes advantage of the probabilistic nature of the phenomenon and employs expected values as the metric for efficiently finding the best control policy.

Experiments were conducted under free-living conditions with multiple subjects. Wearable motion sensors were used in sensor data collection. Ground truth was obtained through manual annotations with portable video recording. Due to the nature of the application, the HMM for each subject was re-learned using the Baum-Welch algorithm. For the two-state model presented, the Baum-Welch algorithm performed consistently well. Part of the future work will include evaluation of its efficacy with multi-state models in our applications. The results demonstrated that significant energy reduction is possible if the objective is to maintain the overall classifier accuracy. In all four subjects, less than 100% energy budget is required to maintain the same accuracy provided by naïve Bayes classifier. These results are encouraging because POMDP provides a systematic approach to reduce sensor resource usage without impacting the classification accuracy.

Future work includes implementing the proposed solution in embedded devices and wearable sensors to verify the algorithm in real time. Furthermore, more states will be included in POMDP to demonstrate the practicality of the proposed method.

ACKNOWLEDGMENT

The authors would like to thank V. Chen, J. Larson, D. Singh, and C. Xu for volunteering in data collection, and an anonymous reviewer for his comments.

REFERENCES

[1] S. Patel, K. Lorincz, R. Hughes, N. Huggins, J. Growdon, D. Standaert, M. Akay, J. Dy, M. Welsh, and P. Bonato, "Monitoring motor fluctuations in patients with Parkinson's Disease using wearable sensors," *IEEE Trans. Inf. Technol. Biomed.*, vol. 13, no. 6, pp. 864–873, Nov. 2009.

[2] D. M. Sherrill, M. L. Moy, J. J. Reilly, and P. Bonato, "Using hierarchical clustering methods to classify motor activities of COPD patients from wearable sensor data," *J. Neuroeng. Rehabil.*, vol. 2, p. 16, 2005.

[3] P. Bonato, "Advances in wearable technology and applications in physical medicine and rehabilitation," *J. Neuroengi. Rehabil.*, vol. 2, no. 2, 2005.

[4] M. Ermes, J. Parkka, J. Mantyjarvi, and I. Korhonen, "Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions," *IEEE Trans. Inf. Technol. Biomed.*, vol. 12, no. 1, pp. 20–26, Jan. 2008.

[5] P. Bonato, "Wearable sensors/systems and their impact on biomedical engineering," *IEEE Eng. Med. Biol. Mag.*, vol. 22, no. 3, pp. 18–20, May–June 2003.

[6] M. L. Moy, K. Matthess, K. Stolzmann, J. Reilly, and E. Garshick, "Free-living physical activity in COPD: Assessment with accelerometer and activity checklist," *J. Rehabil. Res. Dev.*, vol. 46, pp. 277–286, 2009.

[7] U. Anliker, J. Beutel, M. Dyer, R. Enzler, P. Lukowicz, L. Thiele, and G. Trster, "A systematic approach to the design of distributed wearable systems," *IEEE Trans. Comput.*, vol. 53, no. 8, pp. 1017–1033, Aug. 2004.

[8] G. Troster, "The Agenda of Wearable Healthcare," in *IMIA Yearbook of Medical Informatics*. Stuttgart, Germany: Ubiquitous Health Care Syst., 2005, pp. 125–138.

[9] L. Gatzoulis and I. Iakovidis, "Wearable and portable ehealth systems," *IEEE Eng. Med. Biol. Mag.*, vol. 26, no. 5, pp. 51–56, Sept.–Oct. 2007.

[10] T. Starner, "The challenges of wearable computing: Part 1," *IEEE Micro*, vol. 21, no. 4, pp. 44–52, Jul./Aug. 2001.

[11] G. Thatte, M. Li, A. Emken, U. Mitra, S. Narayanan, M. Annavaram, and D. Spruijt-Metz, "Energy-efficient multihypothesis activity-detection for health-monitoring applications," in *Proc. IEEE Annu. Int. Conf. Engineering in Medicine and Biology Society*, Sep. 2009, pp. 4678–4681.

[12] K. Lorincz, B.-R. Chen, G. W. Challen, A. R. Chowdhury, S. Patel, P. Bonato, and M. Welsh, "Mercury: A wearable sensor network platform for high-fidelity motion analysis," in *Proce. 7th ACM Conf. Embedded Networked Sensor Systems*, New York, 2009, pp. 183–196, ACM.

[13] W. Wu, A. Bui, M. Batalin, D. Liu, and W. Kaiser, "Incremental diagnosis method for intelligent wearable sensor systems," *IEEE Trans. Inf. Technol. Biomed.*, vol. 11, no. 5, pp. 553–562, Sep. 2007.

[14] M. Srivastava, A. Chandrakasan, and R. Brodersen, "Predictive system shutdown and other architectural techniques for energy efficient programmable computation," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 4, no. 1, pp. 42–55, Mar. 1996.

[15] B. Kveton, P. Gandhi, G. Theocharous, S. Mannor, B. Rosario, and N. Shah, "Adaptive timeout policies for fast fine-grained power management," in *Proc. Nat. Conf. Artificial Intelligence*, 2007, vol. 2, pp. 1795–1800.

[16] N. Bricon-Souf and C. R. Newman, "Context awareness in health care: A review," *Int. J. Med. Inf.* vol. 76, no. 1, pp. 2–12, 2007 [Online]. Available: http://www.sciencedirect.com/science/article/B6T7S-4J9N13W-1/2/c72a2c111fbbd9b08d95145392830141

[17] W. Wu, M. Batalin, L. Au, A. Bui, and W. Kaiser, "Context-aware sensing of physiological signals," in *Proc. IEEE 29th Annu. Int. Conf. Engineering in Medicine and Biology Society*, Aug. 2007, pp. 5271–5275.

[18] O. Cakmakci, J. Coutaz, K. V. Laerhoven, and H. W. Gellersen, "Context awareness in systems with limited resources," in *Proc. 3rd Workshop Artificial Intelligence in Mobile Systems*, 2002, pp. 21–29.

[19] N. Kamijoh, T. Inoue, C. Olsen, M. Raghunath, and C. Narayanaswami, "Energy trade-offs in the IBM wristwatch computer," in *Proc. 5th Int. Symp. Wearable Computers*, 2001, pp. 133–140.

[20] A. Krause, M. Ihmig, E. Rankin, D. Leong, S. Gupta, D. Siewiorek, A. Smailagic, M. Deisher, and U. Sengupta, "Trading off prediction accuracy and power consumption for context-aware wearable computing," in *Proc. 9th IEEE Int. Symp. Wearable Computers*, Oct. 2005, pp. 20–26.

[21] Y. Wang, B. Krishnamachari, Q. Zhao, and M. Annavaram, "The tradeoff between energy efficiency and user state estimation accuracy in mobile sensing," in *Proc. Int. Conf. Mobile Computing, Applications, and Services*, San Diego, CA, Oct. 2009.

[22] Y. Wang, B. Krishnamachari, Q. Zhao, and M. Annavaram, "Markov-optimal sensing policy for user state estimation in mobile devices," in *Proc. 9th ACM/IEEE Int. Conf. Information Processing in Sensor Networks*, New York, 2010, pp. 268–278.

[23] M. Stäger, P. Lukowicz, and G. Tröster, "Power and accuracy trade-offs in sound-based context recognition systems," *Pervasive Mob. Comput.*, vol. 3, no. 3, pp. 300–327, 2007.

[24] N. Bharatula, U. Anliker, P. Lukowicz, and G. Trster, "Architectural tradeoffs in wearable systems," in *Architecture of Computing Systems—ARCS 2006*, ser. Lecture Notes in Computer Science, W. Grass, B. Sick, and K. Waldschmidt, Eds. Berlin, Germany: Springer, 2006, vol. 3894, pp. 217–231.

[25] W. H. Wu, A. A. T. Bui, M. A. Batalin, L. K. Au, J. D. Binney, and W. J. Kaiser, "Medic: Medical embedded device for individualized care," *Artif. Intell. Med.*, vol. 42, no. 2, pp. 137–152, 2008.

[26] A. Smailagic, D. Reilly, and D. P. Siewiorek, "A system-level approach to power/performance optimization in wearable computers," in *Proc. IEEE Computer Society Annu. Workshop VLSI*, Washington, DC, 2000, p. 15.

[27] Y. Wang, J. Lin, M. Annavaram, Q. A. Jacobson, J. Hong, B. Krishnamachari, and N. Sadeh, "A framework of energy efficient mobile sensing for automatic user state recognition," in *Proc. 7th Int. Conf. Mobile Systems, Applications, and Services*, New York, 2009, pp. 179–192, ACM.

[28] L. Au, M. Batalin, T. Stathopoulos, A. Bui, and W. Kaiser, "Episodic sampling: Towards energy-efficient patient monitoring with wearable sensors," in *Proc. IEEE Annu. Int. Conf. Engineering in Medicine and Biology Society*, Sep. 2009, pp. 6901–6905.

[29] J. Pellegrini and J. Wainer, On the Use of POMDPs to Model Diagnosis and Treatment of Diseases, Instituto de Computacao—UNICAMP, Brazil, Tech. Rep., 2003.

[30] J. Hoey, P. Poupart, C. Boutilier, and A. Mihailidis, "POMDP models for assistive technology," in *Proc. AAAI Fall Symp. Caring Machines: AI in Eldercare*, Arlington, VA, Nov. 2005.

[31] J. E. Goulionis and A. Vozikis, "Medical decision making for patients with Parkinson disease under average cost criterion," *Aust. New Zealand Health Policy*, vol. 6, p. 15, 2009.

[32] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artif. Intell.*, vol. 101, no. 1–2, pp. 99–134, 1998.

[33] C. Papadimitriou and J. N. Tsitsiklis, "The complexity of Markov decision processes," *Math. Oper. Res.*, vol. 12, no. 3, pp. 441–450, 1987.

[34] H. Yu and D. P. Bertsekas, "On near optimality of the set of finite-state controllers for average cost pomdp," *Math. Oper. Res.* vol. 33, pp. 1–11, Feb. 2008 [Online]. Available: http://portal.acm.org/citation.cfm?id=1527933.1527934

[35] N. Meuleau, K.-E. Kim, L. P. Kaelbling, and A. R. Cass, "Solving POMDPs by searching the space of finite policies," in *Proc. 15th Conf. Uncertainty in Artificial Intelligence*, Stockholm, Sweden, Nov. 1999.

[36] K. Murphy, "Dynamic Bayesian networks: Representation, inference and learning," Ph.D. dissertation, Univ. California, Berkeley, 2002.

[37] D. Aberdeen and J. Baxter, "Scalable internal-state policy-gradient methods for POMDPs," in *Proc. 19th Int. Conf. Machine Learning*, San Francisco, CA, 2002, pp. 3–10.

[38] K. Aström, "Optimal control of Markov decision processes with incomplete state estimation," *J. Math. Anal. Appl.*, pp. 174–205, 1995.

[39] J. D. Isom, S. P. Meyn, and R. D. Braatz, "Piecewise linear dynamic programming for constrained POMDPs," in *Proc. 23rd Nat. Conf. Artificial Intelligence—Volume 1*, 2008, pp. 291–296 [Online]. Available: http://portal.acm.org/citation.cfm?id=1619995.1620043

[40] Q. Zhao and K. Liu, "Detecting, tracking, and exploiting spectrum opportunities in unslotted primary systems," in *Proc. IEEE Radio and Wireless Symp.*, Jan. 2008, pp. 491–494.

[41] C. Watkins, "Learning from delayed rewards," Ph.D. dissertation, King's College, Cambridge, U.K., 1989.

[42] P. E. Gill, W. Murray, and M. A. Saunders, "Snopt: An sqp algorithm for large-scale constrained optimization," *SIAM Rev.* vol. 47, pp. 99–131, Jan. 2005 [Online]. Available: http://portal.acm.org/citation.cfm?id=1055334.1055404

[43] L. Au, W. Wu, M. Batalin, D. Mclntire, and W. Kaiser, "MicroLEAP: Energy-aware wireless sensor platform for biomedical sensing applications," in *Proc. IEEE Biomedical Circuits and Systems Conf.*, Nov. 2007, pp. 158–162.

[44] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, pp. 257–286, Feb. 1989.

[45] L. Au, A. Bui, M. Batalin, X. Xu, and W. Kaiser, "CARER: Efficient dynamic sensing for continuous activity monitoring," in *Proc. IEEE Engineering in Medicine and Biology Society Annu. Int. Conf.* , 2011, accepted for publication.

[46] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," *Lecture Notes Comput. Sci.*, vol. 3001/2004, pp. 1–17, 2004.

[47] T. Duong, H. Bui, D. Phung, and S. Venkatesh, "Activity recognition and abnormality detection with the switching hidden semi-Markov model," in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, Jun. 2005, vol. 1, pp. 838–845, vol. 1.

[48] N. Nguyen, D. Phung, S. Venkatesh, and H. Bui, "Learning and detecting activities from movement trajectories using the hierarchical hidden Markov model," in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, Jun. 2005, vol. 2, pp. 955–960, vol. 2.

[49] P. Natarajan and R. Nevatia, "Coupled hidden semi Markov models for activity recognition," in *Proc. IEEE Workshop Motion and Video Computing*, Feb. 2007, p. 10.

[50] W. Zhang, F. Chen, W. Xu, and Z. Cao, "Decomposition in hidden Markov models for activity recognition," *Lecture Notes Comput. Sci.*, vol. 4577/2007, pp. 232–241, 2007.

[51] X. Liu and C.-S. Chua, "Multi-agent activity recognition using observation decomposedhidden Markov models," *Image Vis. Comput.* vol. 24, no. 2, pp. 166–175, 2006 [Online]. Available: http://www.sciencedirect.com/science/article/B6V09-4HM7RW0-5/2/8458ff7a694e1a8102533fbd9f4fbb30

[52] C. Amato, D. S. Bernstein, and S. Zilberstein, "Solving POMDPs using quadratically constrained linear programs," in *Proc. 5th Int. Joint Conf. Autonomous Agents and Multiagent Systems*, New York, 2006, pp. 341–343.

Lawrence K. Au (S'03) received the B.S., M.S. and Ph.D. degrees in electrical engineering from the University of California, Los Angeles (UCLA), in 2004, 2007, and 2011, respectively.

He was a recipient of the NLM Biomedical Informatics Research Training Fellowship at UCLA. He held internships at Intel and Qualcomm, focusing on the development of software algorithms for wireless wearable sensor platforms. His research interests include embedded sensor systems design, signal processing, stochastic modeling, and optimization.

Alex A. T. Bui (M'01) received the Ph.D. degree in computer science from the University of California, Los Angeles (UCLA) in 2000.

Currently, he is a Professor of Radiological Sciences at UCLA and a lead member of the UCLA Medical Imaging Informatics (MII) Group, with research interests focusing on distributed information architectures for biomedical research and clinical environments, probabilistic data modeling, and the visualization of clinical information. He is also Director of UCLA's NIH/NLM Medical Informatics Training Program.

Maxim A. Batalin (M'05) received the B.S. degree in mathematics and computer science from the University of Oregon, Eugene, in 2001, the M.S. and Ph.D. degrees in computer science from the University of Southern California (USC), Los Angeles, in 2002 and 2005, respectively, and the Master of Management degree from Tavriya National University, Simferopol, Ukraine, in 2002.

He is a Postdoctoral Fellow in the Department of Electrical Engineering, University of California, Los Angeles (UCLA). His research interests include intelligent embedded systems, mobile robotics, and sensor networking.

William J. Kaiser (SM'97) received the Ph.D. degree in solid state physics from Wayne State University, Detroit, MI, in 1984.

From 1977 through 1986, as a member of Ford Motor Company Research Staff, his development of automotive sensor and embedded system technology resulted in large volume commercial sensor production. At Ford, he also developed the first spectroscopies based on scanning tunneling microscopy. From 1986 through 1994, at the Jet Propulsion Laboratory, he developed and demonstrated the first electron tunnel sensors for acceleration and infrared detection and initiated the NASA/JPL microinstrument program. In 1994, he joined the faculty of the Electrical Engineering Department at the University of California, Los Angeles (UCLA). At UCLA, he initiated the distributed networked embedded sensor field via many large collaborative programs across several departments. These combined UCLA research activities have now led to the creation of many new programs within DARPA, NSF, NASA, and in commercial technology corporations. He served as UCLA Electrical Engineering Department Chairman from 1996 through 2000. He has authored more than 100 publications, 100 invited presentations and 21 patents.