

# Fault-Tolerant and Low-Power Sampling Schedules for Localized BASNs

Vishwa Goudar, *Student Member, IEEE*, and Miodrag Potkonjak, *Member, IEEE*

**Abstract**—Recent advances in the scope of wearable devices and networks make body area sensor networks (BASNs) an extremely attractive tool to the fields of mobile and tele-health, owing to the range of medical applications they can serve and the diagnostic richness of patient data they can offer. However, for BASNs to achieve true ubiquity, they must be scalable in their support of automated patient data collection, making usability and reliability key considerations. Its designers must wrestle with the tradeoff between usability, hindered by device intrusiveness into the behaviors it measures, and lifetime, enhanced by large power supplies and expensive, sturdy components. Furthermore, the validity and reliability of the collected data are paramount. In this paper, we consider these issues in the context of localized multi-sensory wearable networks and present a method to generate low-power sampling schedules that are resilient to sensor faults while achieving high diagnostic fidelity. We jointly formulate this as a power-constrained sampling problem wherein the number of sensors sampled per epoch are limited, and, a fault tolerant scheduling problem wherein the sampling scheme offers enough redundancy to endure up to a predefined number of sensor faults while maintaining diagnostic accuracy. This formulation is based on, 1) the localized scope of BASNs that engenders strong spatio-temporal interactions in the samples, and, 2) the periodic nature of human behaviors measured. We present our algorithm in the context of gait diagnostics derived from a foot plantar pressure measurement platform and illustrate its performance based on real datasets collected by it.

**Index Terms**—Body area sensor networks, energy-efficient sampling, fault-tolerant sampling, power-constrained sampling.

## I. INTRODUCTION

THE DEMAND for sensor-embedded devices in the medical domain has spawned a promising new subclass of wireless sensor networks (WSNs) known as wearable or body area sensor networks (BASNs). Comprised of networks/arrays of physiological and/or behavioral sensors, they collect detailed and diagnostically rich data which, when used in conjunction with data from patient interviews, on-site observations and tests, enhance diagnostic accuracy, shorten time to detection, and propel preventative healthcare. They also allow medical researchers access to an abundance of high quality data so they may better understand the causes and symptoms of illnesses.

With rising maturity, BASNs will be capable of automatically collecting, processing, and sharing patient medical diagnostic metrics with authorized patient and research databases,



Fig. 1. Pedar plantar pressure map [9] (left), and the Hermes smart shoe platform [16] (right).

and other data sinks in a minimally invasive and unsupervised manner, resulting in devices as inconspicuous and mundane as pacemakers or hearing aids. Therefore, on adoption, they are expected to have enormous impacts on the accessibility, efficiency and quality of care for convalescing patients, geriatric patients, and those with chronic conditions through computer-assisted rehabilitation and continuous health monitoring. However, this integration of mobile healthcare (mHealth) and tele-healthcare requires significant improvements to the reliability and lifetime of these devices, with no compromise to diagnostic integrity or usability.

Currently, BASN usability is stymied by their intrusiveness and lifetime issues. For example, a system comprised of a localized multi-sensory array that monitors foot plantar pressure in elderly adults, such as the Hermes platform [16] (Fig. 1), must be energy-efficient so it may function unattended for extended periods, and light-weight with a small form factor so as not to hinder/disrupt normal activity of the subject wearing the system. Given that such devices are usually battery operated, wherein the rate of power consumption determines the time between recharges, and that power consumption affects the battery size and therefore the form factor, power consumption is crucial to adoptability.

On the other hand, reliability issues are a by-product of physical and economic constraints. Medical sensors can be quite expensive, and over-deploying them towards redundancy and fault-tolerance on space constrained wearable platforms, such as Hermes, is infeasible.

We aim to address these issues with a sampling paradigm that is fault-tolerant and low-power, yet affords high quality medical diagnostic measurements. Our sampling paradigm abides by pre-specified power thresholds by procuring at most  $k$  samples at a time so that the maximum total power drawn by the sensor array is less than this threshold. Furthermore, sensing fault tolerance is realized by ensuring that up to a predefined number of the sampled sensors are allowed to fail, before all

Manuscript received August 22, 2012; revised December 04, 2012; accepted January 03, 2013. Date of publication February 25, 2013; date of current version March 07, 2013. This paper was recommended by Guest Editor F. Koushanfar.

The authors are with the Computer Science Department, University of California, Los Angeles, CA 90095 USA (e-mail: vishwa@cs.ucla.edu; miodrag@cs.ucla.edu).

Digital Object Identifier 10.1109/JETCAS.2013.2243634

the information deduced from their samples is no longer recoverable by other means. We call this number the redundancy factor, or *redFactor*. The former constraint enables device designers to customize power usage based on requisite diagnostic performance, while the latter lets them cater to requisite device lifetime guarantees based on the sensor fault rates, while maintaining diagnostic accuracy throughout the device lifetime.

Our approach strives to maximize the information available for diagnostic extraction under the fault-tolerance and power constraints, by 1) sampling sensors with the most contextually relevant information at the sampled time step, or epoch, 2) utilizing domain-specific spatio-temporal relationships for a given epoch, to infer measurements at un-sampled sensors with high fidelity, 3) sampling/infering a sensor's measurement only if it can also be inferred by at least *redFactor* other sensors, in case the sampled sensor fails, and 4) covering sensors, via measurement or inference, over the majority of epochs when their measurements are most relevant. This last approach affords us the added advantage of improving energy-efficiency by sampling no sensors at epochs deemed irrelevant. We also demonstrate that our algorithm is capable of abiding by the fault-tolerance and power threshold constraints in the context of multi-modal/diagnostic sensing, effectively distributing sensor samples across epochs and among the multiple modalities/diagnostics.

Although we present our algorithm in the context of plantar pressure monitoring systems, it is easily generalized to the monitoring of periodic behaviors, such as those common to BASNs, with devices that wield localized multi-sensory arrays to collect data with strong spatio-temporal dependencies. Also, we limit ourselves to a binary fault model with out-of-band fault detection.

## II. RELATED WORK

WSNs have emerged as a new class of systems that leverage the power of distributed computation, sensing, and actuation towards a host of scientific, military, and engineering applications. However, engineering these systems is not without challenges, a topic surveyed excellently in [13]. As described therein, from the get go, energy has been identified as the single most important resource for this class. Furthermore, communication is recognized as the most demanding operation on this resource, and a number of designs have been proposed to curtail it [18].

The demand for similar sensor-embedded devices in the medical domain has spawned a new subclass of WSNs known as wearable or body area sensor networks (BASNs), the challenges and opportunities for which are surveyed in [2]. BASNs have been applied to host of medical problems from geriatric assistance [10] to emotional health monitoring [24]. Given that it is possible to build systems that can interpret medical diagnostics from electrical signals, attention has also been paid to the energy-efficiency and power consumption of BASNs and their effects on system longevity [3].

Fault tolerance is also a key performance criterion in WSNs. Solutions in this space include self-diagnosis techniques, as well as, co-operative fault recovery techniques. Often, fault-tolerant designs involve redundancy, which, when considered simultaneously with the need for lifetime maximizing/energy-efficient

operation, uncovers a tradeoff. For example, redundant sensor node deployment is usually accompanied by energy-efficient area coverage [20]. Here, the monitored area is covered, in the sensing and communication domains, by multiple distinct and minimally-sized subsets of the over-deployed set that are operated one-at-a-time, thereby achieving high sensing fidelity and redundancy, at low energy costs.

Similar ideas have migrated to the context of BASNs to resolve the tradeoff between sensing fidelity and energy-efficiency, yielding sensor selection approaches wherein a subset of sensors, selected for their ability to jointly predict the measurements at all sensors, are sampled at each epoch [4]. In [14] and [19] the authors proposed an algorithm to further relax the tradeoff between sensing fidelity and "covering" sensor subset size by defining the fidelity only in terms of diagnostic accuracy. However, extending these solutions to provide fault tolerance by constructing multiple distinct sensor subsets is hindered by multiple factors. For one, a BASN sensor's coverage area rapidly changes with time. Also, a BASN designer's ability to overpopulate the wearable platform with sensors is severely limited by the wearable platform size, sensor size, and cost. Under these considerations, we exploit the highly localized nature of body area networks that engenders strong spatio-temporal relationships in the sensed data. This allows us to tackle the problem from the perspective of sample selection, instead of sensor selection, which not only motivates high-fidelity sample inference, but also enables the substitution a sample from a faulty sensor with a sample from an accurate one, yielding low-power, fault-tolerant, and high-fidelity diagnostic metric estimation.

## III. PRELIMINARIES

### A. Human Balance Monitoring via Hermes

We present our algorithm in the context of a human balance monitoring system known as Hermes [16], a smart shoe aimed at extending instability analysis outside of the lab environment. The Hermes platform measures foot plantar pressure via a multi-sensory array comprised of 99 passive resistive pressure sensor purposefully placed at locations specified by the Pedar plantar mapping [9]. Fig. 1 shows the Hermes platform alongside this mapping. The sampling unit for the platform samples each of the sensors at 60 Hz.

While Hermes is capable of continuous measurement, medical professionals are interested in a set of diagnostic metrics for each stride taken by the subject, which do not require all samples for the stride. For example, the GARS-M scale [8], a reliable and valid measure for documenting gait features associated with an increased risk of falling among community-dwelling, frail older persons, relies on guardedness as one of its variables. Measured as the time between heel and toe strikes during a stride, guardedness has also been identified as an indicator of repetitive stress injuries in runners [24]. However, this metric only requires that each sensor be sampled until the detection of foot strike at its respective location, but not after. Similarly, elevated levels of plantar pressure, measured as the average maximum pressure over all sensors for a stride, has been identified as a causative

factor in the development of many plantar ulcers in diabetic patients [12]. Again, the only relevant sample for each sensor is the one that provides the maximum pressure for a stride. In this paper, we focus on guardedness, average maximum pressure (or maximum amplitude), and lateral pressure difference. The last diagnostic metric is an indicator of staggering in the GARS-M. We note that each of these diagnostics is an aggregate over the respective individual sensor metric measurements.

### B. Human Gait

Gait is defined as the way in which movement is achieved by humans with their limbs, such as walking, running, hopping, etc. The gait cycle, or stride, is a functional unit of gait defined as a single sequence of functions of one limb. It is divided into two phases: the stance phase, when the limb is in contact with the ground, and the swing phase, when the limb is in the air for advancement. In order to construct a repeatable sampling scheme, we first define the period of repetition based on the phases of a gait cycle. Given that it is possible to orthogonally detect these phases with a small fixed number of sensors with the method proposed in [15] or via a single large plantar pressure sensor covering the heel as outlined in [14], the sampling scheme produced by our algorithm is defined vis-à-vis the start and end of each of the two phases. Without loss of generality, we note that our selected diagnostics of guardedness, maximum amplitude and lateral pressure difference are all related to samples in the stance phase.

### C. Energy Consumption and Lifetime

While communication is the most demanding operation in WSNs, our work with BASNs focuses on energy consumed in sensor sampling. This is because many BASNs, such as Hermes, are comprised of multi-sensory arrays wherein each node must power significantly more sensors than a WSN node, significantly increasing the energy demands of sampling. For example, the Hermes shoe, built on the MicroLEAP platform, consumes a total of 182.85 mW in active mode, with 72.74 mW consumed by the radio while transmitting at 115.2 kb/s [21] and 100.24 mW consumed in powering the 99 pressure sensors, where sampling power draw is derived from the maximum circuit voltage and force resistance curve for the underlying Flexi-force sensor [11]. Clearly, reducing the number of samples in an epoch from 99 to 1, yields upwards of 54% in power consumption reduction. Furthermore, the resulting reduction in battery discharge rate leads to an increase in the battery capacity [6], and hence its lifetime.

As we shall see, in comparison to the CICA-based algorithm that selects a fixed subset of sensors of cardinality,  $k$ , to be sampled at every epoch [14], our approach offers the added advantage that fewer than  $k$  sensors may be sampled at some epochs. This translates to a further savings in energy over this algorithm. Although it is not considered in our objective function, these savings are measured by the energy savings factor, calculated as the ratio of the maximum number of samples that may be taken to the actual number of samples that are taken.

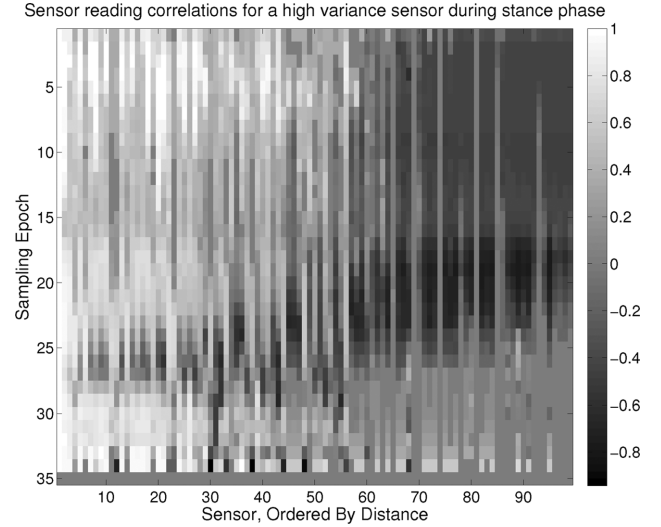


Fig. 2. Spatio-temporal correlations of plantar pressure measurements between a single sensor and all others on a Hermes shoe, over all epochs of the stance phase for a subject.

## IV. DIAGNOSTIC DRIVEN SAMPLE SELECTION

### A. Algorithmic Motivation

We now discuss the motivation behind our algorithmic decisions. The stance phase of the gait cycle is divided into a number of epochs at which we would like plantar pressure to be sampled at various sensors. In order to support pre-specified constraints on maximum power usage, we limit the maximum number of samples,  $k$ , which may be taken at an epoch. A sensor would make a good choice for sampling at an epoch if it can be used to derive the readings at multiple other sensors, or predict them. This predictability of sensors may be determined by a model between readings of sensors. As observed in [14], linear regression models do well to explain the plantar pressure readings of one sensor from another, when the mutual information is high. However, in the general case, our algorithm does not prevent the usage of any other suitable model for this purpose.

Fig. 2 shows the Spearman rank-order correlation coefficients between one sensor,  $s$ , and all 99 others over all epochs of the stance phase from a dataset collected by the Hermes platform. The 99 sensors on the x-axis are ordered by their physical distance from sensor  $s$ . In the spatial domain, we generally observe that nearby sensors are well correlated with the sensor  $s$  compared to distant ones. In the temporal domain, we observe that the level of correlation varies significantly across epochs, even for nearby sensors. The explanation for this hinges on the realization that the correlation between sensors varies by sub-phases of the stance phase.

Although we describe this insight for sensor  $s$ , we have observed that it generalizes to most sensors in all our datasets. Furthermore, each data point in Fig. 2 depicts the correlation between measurements of two sensors at the same epoch. However, as observed in [4], relative time shifting of the signals from two sensors greatly improves their correlation. As a result, we capture the ability of a sensor's measurement in one epoch to predict others in *senPred*, which is defined in

$$\text{senPred}(e, s) = \{(e', t) | \text{modErr}(s_e, t_{e'}) < \varepsilon_{\text{modErr}}\}. \quad (1)$$

Here, the function *modErr* returns the model error, while predicting the measurements at sensor  $t$  in epoch  $e'$  from those of sensor  $s$  in epoch  $e$ , where  $e'$  may occur before or after  $e$ . Predictability is observed when the estimate is accurate to within  $\varepsilon_{\text{modErr}}$ . In our case, *modErr* returns the root mean squared relative linear regression error.

Further, it is important that we place emphasis on sensors with readings that are more relevant to the medical diagnostic being computed. Given that the diagnostics of interest are aggregates of the metric measurements at each sensor, sensors with a larger spread, e.g., with high entropy, are more valuable when we attempt to reconstruct the medical diagnostic with a partial sets of readings. For example, assuming that we can use the maximum amplitude of some subset of sensors to estimate the average maximum amplitude over all sensors for each stride, the maximum amplitude of sensors whose values tend to vary more across strides is more valuable. For metric  $m$  at sensor  $s$ , this is measured as  $\text{spread}_m(s)$ , which in our case returns the entropy of  $m$  as measured at  $s$ . While the variance or inter-quartile range could be used instead of entropy, we observed superior results with entropy for our datasets owing to the mostly unimodal distribution of sensor measurements.

In arriving at the value of a metric  $m$  at a sensor  $s$ , it is reasonable to sample/predict the measurements of  $s$  at only those epochs when we would expect it to yield the value of  $m$ . In other words, to figure out the maximum amplitude at  $s$ , we would like to sample/predict  $s$ 's measurement only at the epochs when  $s$  is likely to observe maximum amplitude. Given a training dataset, we construct a distribution over epochs,  $e$ , of the likelihood when a sensor  $s$  observes its metric value  $m$  and represent this as  $\text{distr}_m(e, s)$ . Sensors will then be sampled according to  $\text{distr}_m$  allowing us to get away with not sampling any sensor in some epochs. For example, if  $\text{distr}_m(e, s)$  indicates that metric  $m$  is never seen by  $s$  at epoch 30, rather that  $m$  tends to occur around epoch 5, we do not need to sample  $s$  for  $m$  at epoch 30 due to its statistical improbability of occurrence at that epoch. This effect is further enhanced by the spatio-temporal correlations, as a sensor may be sampled at one epoch to recreate the measurements of other sensors at other epochs, vacating the necessity to sample at these other epochs.

Finally, sensing fault-tolerance is provided by applying one sensor's sample towards prediction of another's, only if the latter's sample is also predictable by *redFactor* samples from sensors that are distinct each other and the original predictor. For example, it order for it to be acceptable that  $s$ 's measurement at epoch  $e$  will infer  $t$ 's measurement at epoch  $e'$ , there must exist a set of sensors  $(s_1, \dots, s_{\text{redFactor}})$  distinct from  $s$ , each of which produce at least one sample capable of predicting  $t$ 's measurement at  $e'$ . It is important to note here that the user-imposed power-constraints must simultaneously be adhered to; with the premise that sensor  $s$ 's failure will result in one of the *redFactor* samples being used to recover  $t$ 's reading at epoch  $e'$ , it must be ensured that this replacement sample must occur in an epoch when, in the replacement's absence, at most  $k - 1$  sensors are sampled.

In the following sections, we describe an offline learning approach that combines these considerations. The described approach produces a customized sampling schedule based on a

TABLE I  
ALGORITHM 1 POWER CONSTRAINED SAMPLE SELECTION

---

1:	Input: $\text{senPred}$ ; $\text{distr}_m$ ; $\text{spread}_m$ ; $k$ , the maximum sensors to sample per epoch; $\text{sensSamp}$ , a $\text{numEpoch} \times k$ matrix of sensors to sample; $\text{predMap}$ , a $\text{numEpoch} \times \text{numSensor}$ matrix
2:	Output: $\text{sensSamp}$ ; $\text{distr}_m$ ; $\text{predMap}$
3:	<b>while</b> True <b>do</b>
4:	$\text{sampContr}(e, s) = 0 \ \forall$ epochs $e$ and sensors $s$
5:	<b>for</b> $e$ in $1:\text{numEpochs}$ , $s$ in $1:\text{numSensors}$ <b>do</b>
6:	<b>if</b> $\text{sensSamp}(e, k') \neq 0$ for some $k' \leq k$ <b>do</b>
7:	$\text{sampContr}(e, s) = \sum_{(e', t) \in \text{senPred}(e, s)} \left( \frac{f(\text{spread}_m(t)) \times g(\text{distr}_m(e', t)) \times h\left(\sum_{e''} \text{distr}_m(e'', t)\right)}{\left(\sum_{e''} \text{distr}_m(e'', t)\right)} \right)$
8:	<b>end if</b>
9:	<b>end for</b>
10:	<b>if</b> $\text{sampContr}(e, s) = 0 \ \forall$ epochs $e$ and sensors $s$ , <b>do</b>
	break
11:	<b>end if</b>
12:	Let $\max_{e, s}(\text{sampContr}) = \text{sampContr}(e', s')$
13:	$\text{sensSamp}(e', k') = s'$ where $k'$ is the first index $\leq k$ with non-zero value at $e'$
14:	<b>for</b> all $(e'', t) \in \text{senPred}(e', s')$ <b>do</b>
15:	$\text{predMap}(e'', t) = (e', s')$
16:	$\text{distr}_m(e'', t) = 0$
17:	<b>end for</b>
18:	<b>end while</b>
19:	Return $\text{predMap}$ , $\text{distr}_m$ and $\text{sensSamp}$

---

training dataset of readings, at all sensors and over a few gait cycles of the subject. The resulting schedule may then be applied towards fault-tolerant and power constrained diagnostic estimation for the subject.

### B. Power Constrained Sample Selection

We now describe the power constrained sample selection (PCSS) algorithm, in Table I, which determines the sensors to be sampled at each epoch. The merit of a sensor  $s$ 's assignment to an epoch  $e$  is measured by a heuristic  $\text{sampContr}(e, s)$ . The inputs to the algorithm are *senPred*,  $\text{distr}_m$ , and  $\text{spread}_m$  that are described in the previous subsection, and the maximum samples per epoch,  $k$ . The algorithm returns *sensSamp*, which stores the sensors to be sampled at each epoch, and *predMap* that maps each sensor  $s'$  that will be sampled at epoch  $e'$ , to the sensor  $t$  whose observation at epoch  $e''$  will be predicted by it. For reasons that will be made clear in Subsection IV-D, PCSS also takes *sensSamp* and *predMap* as inputs that can be assumed to be empty for now. Note that we are only interested in a sensor's measurement at epochs when the metric observation likelihood matrix,  $\text{distr}_m$ , indicates that the likelihood is nontrivial (lines 5–9), and when the measurement has not

already been covered (lines 14–17). This latter point is noteworthy as we do not want to value sensor  $s$  at an epoch  $e$  for its ability to predict sensor  $t$ 's readings at an important epoch  $e'$ , if  $\text{predMap}(e', t)$  has already been covered by a different pair  $(e'', s')$  that was selected into  $\text{sensSamp}$ . Furthermore, we are not interested in sampling sensors at epochs when  $k$  samples have already been scheduled (line 6). Finally, sensor selection is terminated when either no unassigned epochs remain in  $\text{sensSamp}$ , or no merited assignments may be made (lines 10–11).

The valuation heuristic  $\text{sampContr}(e, s)$  combines, for each  $(e', t)$  that it covers, the spread of sensor  $t$ 's readings via function  $f$ , and, the metric observation likelihood at  $e'$  via function  $g$ , where the likelihood matrix is adjusted at each iteration based on sampling assignments (line 16). The valuation also takes into consideration, via function  $h$ , the extent to which  $t$  has been covered at epochs with nontrivial metric observation likelihood, the motivation for which will be discussed in Section IV-D. The functions  $f$ ,  $g$ , and  $h$  are used to assign relative importance to their operands in the computation of the valuation heuristic. We use power functions with different exponents to express the relative importance.

### C. Optimal Fault Tolerant Coverage

As described thus far, the PCSS algorithm attempts to cover the most relevant samples under pre-specified power-constraints. However, this coverage is not fault-tolerant. Towards this end, we augment the algorithm with a call to a subroutine, outlined in Table II, while determining the coverage of a sample (line 7 of PCSS). This subroutine, called optimal fault tolerant coverage (OFTC), maximizes the value of the sample set covered by each sample  $(e, s)$  (line 5 of PCSS) under the constraints that each of the covered samples  $(e', t)$  is not only well predicted by the model [by being part of  $\text{senPred}(e, s)$ ], but also well predicted by  $\text{redFactor}$  distinct other sensors that provide redundant coverage. Here, the value of each covered sample is derived as described in the composition of  $\text{sampContr}$  in the previous section (line 7 of PCSS and line 20 of OFTC).

The maximum-valued sample set covered by a sample  $(e, s)$ , or  $\text{covSet}$ , is a subset of all samples that could possibly be covered by  $(e, s)$ , or  $\text{senPred}(e, s)$ . Furthermore, it is a subset which can simultaneously be covered by  $\text{redFactor}$  distinct other sensors, while obeying the power constraints on the maximum number of samples per epoch,  $k$ , even if sensor faults occur. A valuable observation is that each of the samples in  $\text{covSet}$  can admit redundant coverage by a set of samples distinct from the others, as long as the power-constraint is honored. A sample  $(e', t')$  selected into the  $\text{covSet}$  of sample  $(e, s)$  could have redundant coverage from  $(e'', s')$  that is already selected into  $\text{sensSamp}$ , while the sample  $(e''', t'')$ , which cannot be accurately inferred from  $(e'', s')$ , could also be selected into  $\text{covSet}$  of sample  $(e, s)$  due to redundant coverage from a different sample selected into  $\text{sensSamp}$ . This is the basis for computing the redundant coverage from the samples in  $\text{sensSamp}$  for all samples in  $\text{senPred}(e, s)$  (line 5). Note that there is no power cost incurred by this form of collaborative redundant coverage.

TABLE II  
ALGORITHM 2 OPTIMAL FAULT TOLERANT COVERAGE

---

```

1: Input: senPred; distrm; spreadm; sensSamp, a numEpoch
   × k matrix of sensors being sampled ; s, the sampled
   sensor; e, the sampling epoch; redFactor, the
   redundancy factor
2: Output: covSet, the optimal set of samples (e', t')
   covered by (e, s) in a fault-tolerant manner under
   sampContr; bestSetVal, the heuristic value of covSet
   under sampContr; redSet, containing a set of samples,
   for each (e', t') in covSet, that will cover (e', t') if sensor
   s fails
3: fitPred(t'') = NULL ∀ sensors t''
4: bestSetVal = -Inf
5: redSet''(e', t') = set of samples (e'', t'') scheduled in
   sensSamp, s.t. (e', t') ∈ senPred(e, s) ∩ senPred(e'', t'')
6: for each sensor t'' do
7:   fitPred(t'') = senPred(e, s) ∩ senPred(e'', t'')
8: end for each
9: combFitSet = redFactor-combinations of all t'' s.t.
   fitPred(t'') ≠ NULL
10: for each set of sensors fitSet in combFitSet do
11:   redSet'(e', t') = redSet''(e', t') ∪ epochs e' and sensors t'
12:   for each t'' in fitSet do
13:     for each (e', t') in fitPred(t'') do
14:       if (e''', t'') ∉ redSet'(e', t') ∪ epochs e''' do
15:         redSet'(e', t') = redSet'(e', t') ∪ (e, t'')
16:       end if
17:     end for
18:   end for
19: redSet'(e', t') = NULL ∪ epochs e' and sensors t' s.t.
   cardinality(redSet'(e', t')) < redFactor
20: setVal = 
$$\sum_{(e', t') \text{ s.t. } \text{redSet}(e', t') \neq \text{NULL}} \left( \frac{f(\overline{\text{spread}_m(t')}) \times g(\text{distr}_m(e', t')) \times h\left(\sum_{e''} \text{distr}_m(e'', t')\right)}{\left(\sum_{e''} \text{distr}_m(e'', t')\right)} \right)$$

21: if bestSetVal < setVal do
22:   bestSetVal = setVal; redSet = redSet'; covSet = (e', t')
   s.t. redSet(e', t') ≠ NULL
23: end if
24: end for each
25: Return covSet, bestSetVal and redSet

```

---

If the redundant coverage obtained thusly does not satisfy the  $\text{redFactor}$  cardinality constraint for some samples in  $\text{senPred}(e, s)$ , backup samples must occur in epoch  $e$ , so they may replace the faulty sensor  $s$ 's sample without incurring additional power costs. This type of redundant coverage is captured in  $\text{fitPred}$  (lines 6–8). In order to then pick the maximum-valued subset of  $\text{senPred}(e, s)$  that can be covered with redundancy and under power constraints, all  $\text{redFactor}$ -combinations of sensors that can provide redundant coverage for samples in  $\text{senPred}(e, s)$ , while themselves being sampled in epoch  $e$ , are iterated over (lines 9–24). In doing so, it is ensured that each covered sample admits redundant coverage from a distinct set of sensors (line 14).

Support for fault-tolerance, via a call to the OFTC algorithm, involves related changes to the PCSS algorithm. These include taking *redFactor* and *redSet*, the redundant covering set for each sample covered in *predMap*, as additional inputs and returning *redSet* as an additional output. Most importantly, the computation of *sampContr* (line 7 of PCSS) is replaced by a call to OFTC with *senPred*, *distr<sub>m</sub>*, *spread<sub>m</sub>*, *sensSamp*, *s*, *e*, and *redFactor* as inputs. *sampContr*(*e*, *s*) is then represented by the value of *bestSetVal* returned by OFTC. Finally, the set of samples accounted for as covered (lines 14–17 of PCSS) are limited to the OFTC return value of *covSet* for (*e'*, *s'*), while the return value of *redSet* is merged into the corresponding structure of PCSS.

#### D. Coverage Improving Iterative Refinement

While the PCSS algorithm is driven by the ability of samples to predict others, the metric observation likelihood of the covered sensors, and, their metric spreads, it does not ensure that the selected sensors are well covered. Put differently, each sensor that we decide to cover must be covered in most, if not all, epochs when they are likely to observe the metric, barring which we increase the likelihood of using erroneous values in the composition of the diagnostic and increase its error. This is precisely why, once the PCSS algorithm has decided to cover a sensor, function *h* increases its contribution to the valuation of sensors that predict it; the lower the sum of likelihood of metric observation over epochs that have not been covered, the more we would like to ensure that it gets covered. *h* is particularly helpful in breaking ties when multiple sensors with similar spreads and are vying for an epoch where their metric observation likelihood is similar.

However, if the PCSS algorithm chooses to cover too many distinct sensors in the first few iterations, it is likely that it will be difficult to ensure high coverage of all these sensors in the end, due to an increase in competition among the covered sensors for each epoch. This effect is exacerbated as *k* increases; as more samples may be scheduled per epoch, PCSS's initial focus is to choose to cover sensors with high entropy at epochs where the metric observations likelihood is high. By the time the algorithm alters course to increase coverage, the competition may be too high. We resolve this by running the PCSS algorithm in the context of an iteratively refinement strategy for sensor coverage, that we call the Coverage Improving Iterative Refinement (CIIR) algorithm and outline in Table III.

The CIIR algorithm eliminates those sensors from consideration for coverage at each successive run of the PCSS algorithm, which have failed to meet a coverage threshold,  $\tau_{cov}$ , over previous runs (lines 12–14). This is achieved by removing them from the *senPred* structure and a corresponding cleanup of the *predMap*, *redSet*, and *sensSamp* structures. This reduces competition and subsequently improves coverage for the shortlisted sensors. Note that the PCSS algorithm uses the *senPred* structure to determine valuation and assign coverage in lines 7 and 14–17, respectively, of the PCSS algorithm. In lieu of iterative improvement of coverage, the CIIR algorithm introduces two parameters  $\tau_{mincov}$  and  $\tau_{maxcov}$  that provide the range of coverage threshold values to be iterated over. Here,  $\tau_{maxcov}$  serves as a tuning knob between accuracy via coverage of more sensors, versus, accuracy via improved precision of metric esti-

TABLE III  
ALGORITHM 3 COVERAGE IMPROVING ITERATIVE REFINEMENT

---

```

1: Input: senPred; distrm; spreadm; k, the maximum sensors
   to sample per epoch; equivm, a map of equivalent sensors;
   redFactor, the redundancy factor
2: Output: sensSamp, a numEpoch × k matrix of sensors to
   sample; predMap, a numEpoch × numSensor matrix of
   pairs indicating the pair (e, s) used to predict
   measurements at (e', t); redSet, a numEpoch × numSensor
   matrix of sets indicating the set of samples that provide
   backup for the prediction of measurements at (e', t)
3:  $\tau_{cov} = \tau_{mincov}$ 
4: Set predMap(e, s) = NULL  $\forall$  epochs e and sensors s
5: Set sensSamp(e, k') = NULL  $\forall$  epochs e and k' in 1:k
6: Set redSet(e, s) = NULL  $\forall$  epochs e and sensors s
7: for k' in 1:k do
8:   senPred' = senPred
9:   while  $\tau_{cov} \leq \tau_{maxcov}$  do
10:    Run PCSS with input predMap, sensSamp, redSet,
    senPred', spreadm, distrm and k' and store the return
    values in predMap', redSet', distrm' and sensSamp'
11:    Calculate coverage(s) =  $1 - \sum_{e \in 1:numEpochs} \text{distr}_m'(e, s)$ 
12:    predMap'(e, s) = NULL, redSet'(e, s) = NULL and
    distrm'(e, s) = distrm(e, s)  $\forall$  epochs e and sensors s s.t.
    coverage(s) <  $\tau_{cov}$ 
13:     $\forall$  epochs e and k'' in 1:k', set sensSamp'(e, k'') =
    NULL if predMap'(e', s') = NULL for all (e', s')
    covered by sensSamp'(e, k'')
14:     $\forall$  epochs e and sensors s, remove (t, e') from
    senPred'(e, s) if coverage(t) <  $\tau_{cov}$ 
15:     $\tau_{cov} = \tau_{cov} + \delta$ 
16:  end while
17:  $\forall$  epochs e and sensors s, remove (t, e') from
    senPred(e, s) if t in equivm(r) for any r s.t. coverage(r)
     $\geq \tau_{maxcov}$ 
18:  sensSamp = sensSamp'; predMap = predMap';
    redSet = redSet'; distrm = distrm'
19: end for
20: Return predMap, redSet and sensSamp

```

---

mates at the covered sensors. Also,  $\tau_{mincov}$  supplies the starting point for iterative improvement and may be set to the minimum coverage value observed, in *coverage* (line 11), after the first iteration.

To further improve coverage quality when multiple samples may be taken in an epoch, the CIIR algorithm calls the PCSS algorithm with stepwise increments, *k'*, of the maximum number of samples per epoch, *k* (lines 7 and 10). Between such increments, sensors selected for coverage must be covered satisfactorily. Consequently, the covered sensors and sampling schedule constructed for one value of *k'* must be maintained as a subset of the covered sensors and sampling schedule for subsequent values of *k'* (lines 10 and 18). In addition to improving coverage, this approach allows us the opportunity to keep from simultaneous covering sensors that are well correlated or equivalent. The structure *equiv<sub>m</sub>* maps each sensor to the set of sensors

TABLE IV  
SENSOR SELECTION PERFORMANCE FOR MAXIMUM AMP

Dataset	Sensors Sampled	Sensors Predicted	Mean % of max. entropy	Energy Savings Factor
Left 1	13	10	87	1.85
Right 1	11	9	96	2.08
Left 2	12	3	77	1.56
Right 2	8	2	88	1.78
Left 3	8	2	80	1.53
Right 3	6	3	80	2.5
Left 4	8	4	91	1.96
Right 4	4	2	95	2.0
Left 5	15	3	85	5.55
Right 5	9	2	89	5.26

that it is equivalent to. For our application, we define equivalency of sensors  $s$  and  $t$  by a low root mean squared relative linear regression error for the metric values at  $s$  and  $t$ . To avoid concurrent coverage of equivalent sensors,  $senPred$  is updated, between increments of  $k'$ , by filtering out all sensors that are equivalent to each covered sensor (line 17). Lastly, the point of having the PCSS algorithm make sampling decisions for any value up to  $k'$  (line 6 of the PCSS algorithm), is that epochs left empty by PCSS when run with one value of  $k'$  may come in handy at a subsequent value of  $k'$ .

The CIIR algorithm terminates to produce the sampling scheme, in  $sensSamp$ , the resulting map of sensor measurements to predict, in  $predMap$ , and a multimap of  $redFactor$  backup samples from distinct sensors for each predicted measurement, in  $redSet$ . By sampling measurements at each stride based on the schedule in  $sensSamp$ , the diagnostic metric is composed, based on a model, from the metric values at the sensors covered in  $predMap$ . In our case, similar to [14], we use multiple linear regression as the model as it provides a good fit. In the case of sensor faults,  $redSet$  is consulted to alter  $predMap$ , and possibly  $sensSamp$ , so that diagnostic metric composition can be continued at the same level of accuracy.

The individual metric value at each covered sensor is derived from its sampled/predicted measurements, in a manner identical to the case when its measurements are available for all epochs. Note that, by construction, we expect these sparsely sampled/predicted values to have high accuracy, since the mapping in  $predMap$  derives from the strongly linked samples in  $senPred$ . Consequently, and given that our coverage of the relevant epochs for these sensors is near complete, we expect the predicted metric values at these sensors to be accurate as well.

#### E. Power Constrained Multi-Modal/Diagnostic Sampling

Our algorithm may easily be extended to the multi-modal or multiple diagnostic metric cases, such as the composition

TABLE V  
SENSOR SELECTION PERFORMANCE FOR GUARDEDNESS

Dataset	Sensors Sampled	Sensors Predicted	Mean % of max. entropy	Energy Savings Factor
Left 1	13	19	59	2.0
Right 1	10	12	73	2.32
Left 2	4	2	65	2.5
Right 2	6	2	67	2.0
Left 3	7	3	75	1.27
Right 3	2	8	70	1.0
Left 4	8	3	62	1.41
Right 4	2	6	74	1.0
Left 5	2	3	89	1.0
Right 5	2	6	87	1.0

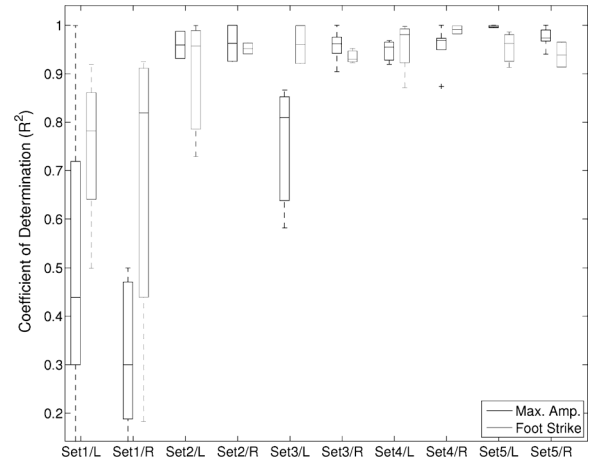


Fig. 3. Coefficient of determination ( $R^2$ ) for predicted sensors.

of guardedness and average maximum amplitude from plantar pressure readings. In order to accomplish this, we first note that there must be one set each, of the following structures, per metric:  $senPred$ ,  $sensSamp$ ,  $predMap$ ,  $redSet$ ,  $sampContr$ , and  $coverage$ . We construct a vector of metric weights that will be used to switch between the metrics in the main while loop of the PCSS algorithm (lines 3–18). A weight,  $w_1$ , for maximum amplitude and,  $w_2$ , for guardedness will assign up to  $w_1/(w_1 + w_2)$  of the assigned epochs to samples for the maximum amplitude metric and up to  $w_2/(w_1 + w_2)$  of the assigned epochs to samples for guardedness. However, fewer epochs may be assigned to a metric, if only those many valuable samples may be obtained. The weights are derived from the width of the likelihood curves  $distr_m$  for each metric; if a metric requires samples from more epochs, it deserves a higher weight. The assignments of epochs to samples for individual metrics are made in tandem. In addition, PCSS is modified to allow for the same sensor to be sampled at a given epoch for multiple diagnostic metrics. In this case, the construction of the redundant covering



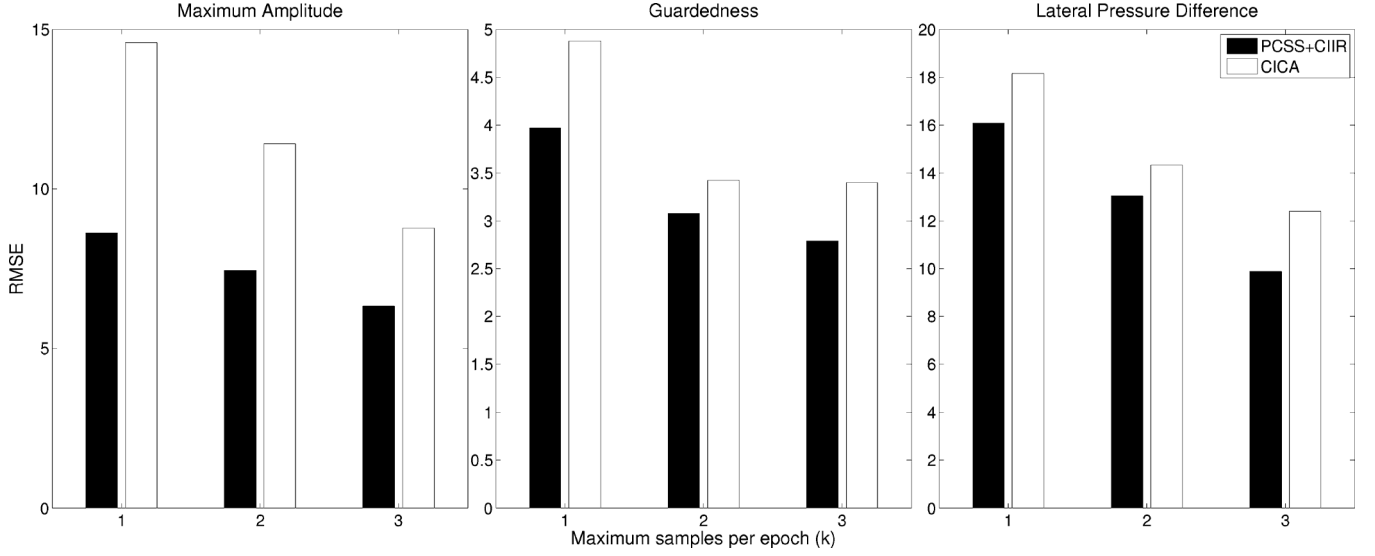


Fig. 4. RMSE of max amplitude, guardedness, and lateral pressure difference diagnostics for PCSS+CIIR versus CICA [14].

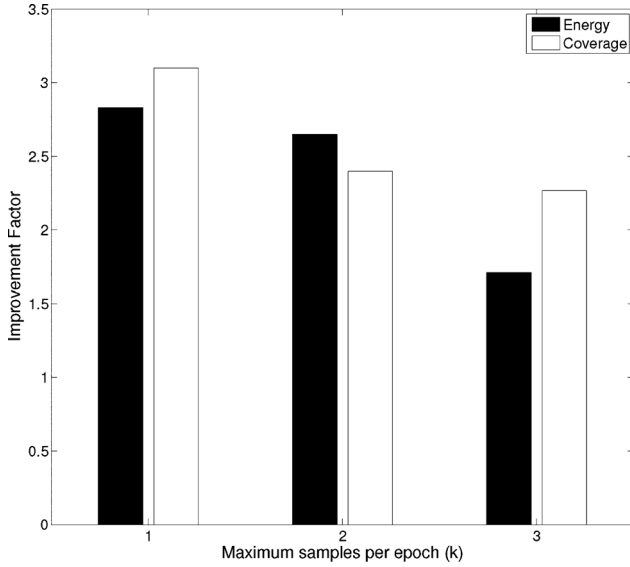


Fig. 5. Sensor coverage and energy savings factors for different  $k$ .

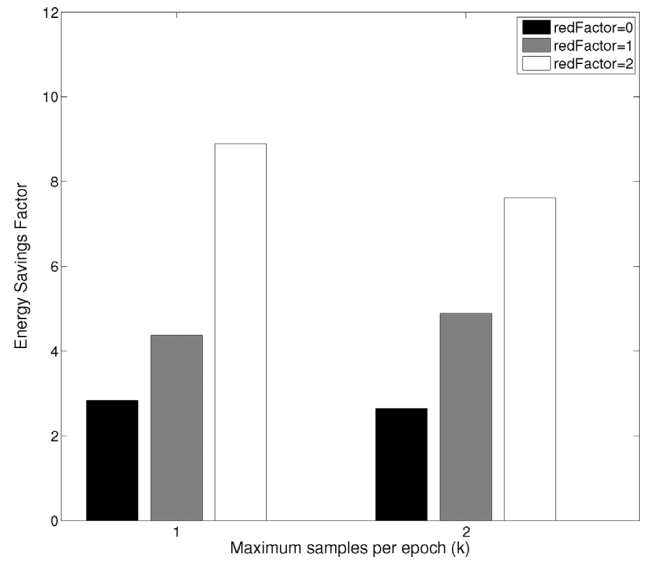


Fig. 6. Energy savings factors for different levels of fault-tolerance.

set, in *redSet*, to replace such a sample must consider the predicted measurements for each such diagnostic metrics.

## V. RESULTS

We discuss the performance of our algorithm in the context of five plantar pressure datasets obtained from distinct subjects with distinct gait profiles, wearing the Hermes shoe while walking. The number of strides per dataset varies from seven to a couple hundred. We divide each dataset into two subsets, ascribing 80% for training and 20% for testing. All the results discussed here are for the testing subsets. Also, note that each dataset includes data for the left and right feet which are treated independently.

### A. Sensor Selection and Prediction

We assess performance from the perspectives of the quality of sensors selected, energy-efficiency of the proposed sampling

schemes and the accuracy in metric prediction. The presented summaries are for  $k = 1$  and  $\text{redFactor} = 0$ , wherein separate sampling schemes were constructed for each of the metrics. Tables IV and V summarize the performance for the maximum amplitude and guardedness metrics, respectively.

We also observe that we are predicting sensors with a lot of information—the average entropy of the predicted sensors as a percentage of maximum entropy is fairly high, especially in the case of the maximum amplitude metric. This bias is owed to the fact that the weight, in *sampContr*, placed on the spread in function  $f$  is much higher, relative to  $g$  and  $h$ , in the case of maximum amplitude. These weights were selected to achieve better diagnostic accuracy.

Perhaps most importantly, we observe that despite us being able to predict many sensors, the proposed sampling schemes are able to provide energy savings up to a factor of 5.55 with



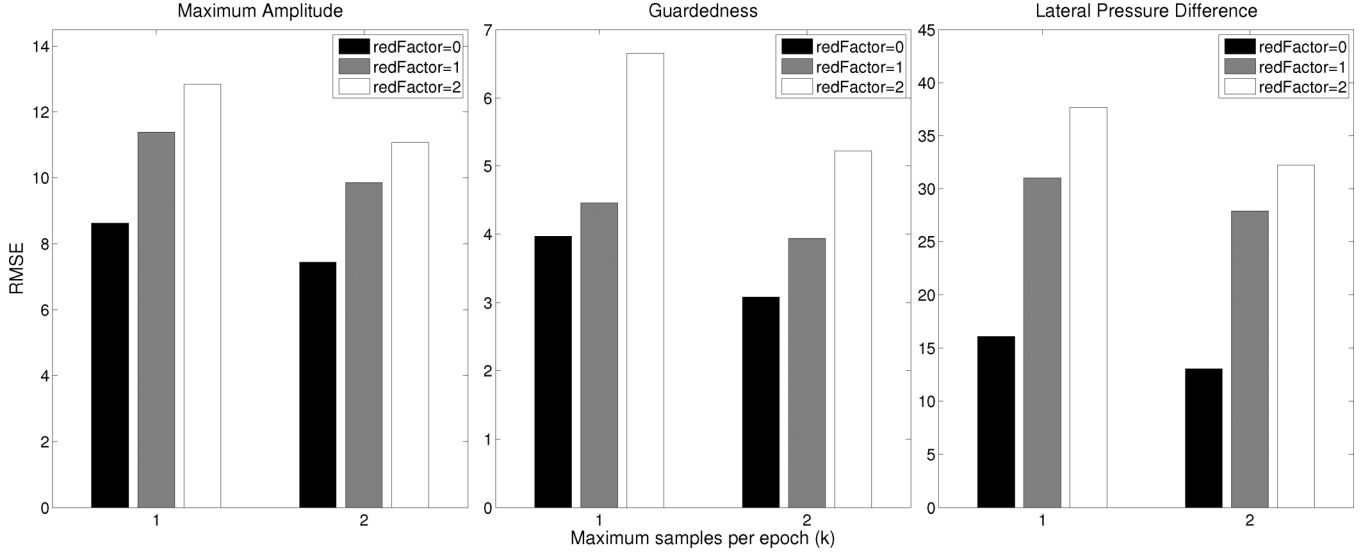


Fig. 7. RMSE of max amplitude, guardedness, and lateral pressure difference diagnostics for different levels of fault-tolerance.

the mean savings hovering around 2 and 1.5 for each of the metrics, when compared to CICA [14]. The energy savings factor is calculated based on the number of epochs during which no samples are taken. Note here, that the CICA-based algorithm samples the same sensor(s) at every epoch. So, it is clear, in that case, that only a single sensor's metric would have been available, given the power constraint. While the quality of the metric for this sensor would be perfect, Fig. 3 shows that in a majority of the datasets, our algorithm's accuracy of prediction, as described by the coefficient of determination, is high (close to 1.0) for all predicted sensors. Low accuracy is observed for the first dataset due to the fact that it is small, comprised of five training strides and two testing strides.

### B. Accuracy of Diagnostic Composition

Next, we compared the energy-efficiency and diagnostic accuracy of our algorithm for the average maximum amplitude, guardedness and lateral pressure difference diagnostics, to the CICA-based algorithm in [14], with  $redFactor$  set to 0. Here, we measure accuracy in terms of the root mean square error (RMSE) in each of the diagnostics metrics, where the RMSE is taken jointly over all datasets. For both the algorithms, all metrics are predicted from a single sampling scheme that is distinct for each dataset. Fig. 4 plots the observed RMSE of the two algorithms, averaged over all datasets, for each of the diagnostic metrics, and, for values of  $k$  between 1 and 3. The improvement over CICA ranges between 10% and 40%. Note that under a linear battery discharge model, the studied values of  $k$  translate to a lifetime increase of between 53% and 54%. Furthermore, we observe significant improvements in energy savings as well as sensor coverage for different values of  $k$  over CICA (Fig. 5). Here, the sensor coverage factor is just the ratio of the total number of sensors sampled over all metrics, to  $k$ . Both savings factors are a weighted average over all datasets, where the weights are the number of strides per dataset. This helped adjust for the large deviations observed in the first dataset (Tables IV and V).

We are able to infer metric values at over twice the number of sensors sampled by CICA. The energy savings vary between 43% and 65% rising up to 74% for one of the datasets when  $k$  is 2. However, despite being able to infer metric values for many more sensors than  $k$ , we do not observe such large improvements in diagnostic accuracy. Whereas CICA picks the most orthogonal set of high-relevance sensors to construct the diagnostic metrics, our algorithm does not exert such fine grained control on the relationship between sensors selected for coverage. We also observe that our improvement over CICA for different values of  $k$  vary across diagnostics. This is due to the difference in the way diagnostic metrics are balanced by the two approaches while selecting a sensor for coverage. CICA is directly data driven and bases its sensor selection, at each iteration, on the improvement in error seen for each diagnostic. Based on our problem formulation, we have chosen instead to weigh sample selection for different metrics based on their relative likelihood distribution across the epochs in a stride.

### C. Performance Under Fault Tolerance

Finally, we compare the energy-efficiency and diagnostic accuracy of our algorithm under different values of  $redFactor$  for each of the diagnostic metrics. As before, accuracy is measured in terms of the RMSE over all datasets for each diagnostic metric, based on a single sampling scheme per dataset. However, in order to determine the diagnostic accuracy in case of sensor faults, we must measure the error under such conditions. We accomplish this by removing a random sensor from  $sensSamp$  and  $predMap$ , replacing its samples with those from  $redSet$  and computing the error based on the resulting  $predMap$ . If  $redFactor$  is greater than 1, this procedure is repeated  $redFactor$  times. Fig. 7 shows the resulting RMSE for each metric, averaged over 100 such testing trials, for values of  $k$  set to 1 and 2, and values of  $redFactor$  ranging from 0 to 2.

In all cases, we see that the error increases with  $redFactor$ . As the need for fault-tolerance rises, there are fewer sensors that can

be covered due to the increased standards of coverage under immutable spatio-temporal redundancy. Fig. 6 provides further evidence to this effect as we see a rise in the energy savings factor (over CICA) with an increasing *redFactor*, owing to reduced coverage. However, we also notice that the error decreases with *k* regardless of *redFactor*, thereby allowing the device designer to cater to accuracy requirements under lifetime constraints by relaxing the power constraints.

## VI. CONCLUSION

We have presented an approach for sample selection under fault-tolerance and power constraints in wearable BASNs. We evaluate our approach for multiple validated diagnostics from plantar pressure measurements, wherein we are able to produce sensor fault-tolerant schedules with significant energy savings and favorable diagnostic accuracy.

## REFERENCES

- [1] V. Goudar and M. Potkonjak, "Energy-efficient sampling schedules for body area networks," in *IEEE Sensors*, Oct. 2012, pp. 1–4.
- [2] M. A. Hanson, "Body area sensor networks: Challenges and opportunities," *Computer*, vol. 42, no. 1, pp. 58–65, Jan. 2009.
- [3] L. Yan, L. Zhong, and N. K. Jha, "Energy comparison and optimization of wireless body-area network technologies," in *Proc. ICST BodyNets*, Jun. 2007, pp. 1–8.
- [4] H. Noshadi *et al.*, "Energy optimization in wireless medical systems using physiological behavior," in *Proc. ACM Wireless Health*, Oct. 2010, pp. 128–136.
- [5] S. Megerian, F. Koushanfar, G. Qu, G. Veltri, and M. Potkonjak, "Exposure in wireless sensor networks: Theory and practical solutions," *J. Wireless Netw.*, vol. 8, no. 5, pp. 443–454, Sep. 2002.
- [6] R. Rao *et al.*, "Battery modeling for energy aware system design," *IEEE Computer*, vol. 36, no. 12, pp. 77–87, Dec. 2003.
- [7] J. B. Wendt, S. Meguerdichian, H. Noshadi, and M. Potkonjak, "Semantics-driven sensor configuration for energy reduction in medical sensor networks," in *Proc. ISLPED*, Jul. 2012, pp. 303–308.
- [8] J. M. VanSwearingen *et al.*, "The modified gait abnormality rating scale for recognizing the risk of recurrent falls in community-dwelling elderly adults," *Phys. Therapy*, vol. 76, no. 9, pp. 994–1002, Sept. 1996.
- [9] Novel.de, Pedar., 2007 [Online]. Available: <http://www.novel.de/>
- [10] W. Wu *et al.*, "The SmartCane system: An assistive device for geriatrics," in *Proc. ICST BodyNets*, Mar. 2008.
- [11] FlexiForce sensor [Online]. Available: <http://www.tekscan.com/flexible-force-sensors>
- [12] P. R. Cavanagh and S. A. Bus, "Off-loading the diabetic foot for ulcer prevention and healing," *JAPMA*, vol. 100, no. 5, pp. 360–368, Sep. 2010.
- [13] D. Culler, D. Estrin, and M. Srivastava, "Guest editors' introduction: Overview of sensor networks," *Computer*, vol. 37, no. 8, pp. 41–49, Aug. 2004.
- [14] J. B. Wendt and M. Potkonjak, "Medical diagnostic-based sensor selection," in *Proc. IEEE Sensors*, Oct. 2011, pp. 1507–1510.
- [15] S. Popovic *et al.*, "A reliable gyroscope-based gait-phase detection sensor embedded in a shoe insole," *IEEE Sensors J.*, vol. 4, no. 2, pp. 268–274, Apr. 2004.
- [16] H. Noshadi *et al.*, "Hermes—Mobile balance and instability assessment system," in *Proc. Biosignals*, Jan. 2010, pp. 264–270.
- [17] R. Dickerson, E. Gorlin, and J. Stankovic, "Empath: A continuous remote emotional health monitoring system for depressive illness," in *Proc. ACM Wireless Health*, Oct. 2011.
- [18] F. Koushanfar, A. Davare, D. Nguyen, A. Sangiovanni-Vincentelli, and M. Potkonjak, "Techniques for maintaining connectivity in wireless ad-hoc networks under energy constraints," *ACM Trans. Embedded Comput. Syst.*, vol. 6, no. 3, pp. 16–26, 2007.
- [19] V. Goudar and M. Potkonjak, "Power constrained sensor sample selection for improved form factor and lifetime in localized bans," in *Proc. ACM Wireless Health*, Oct. 2012.
- [20] S. Megerian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, "Worst and best-case coverage in sensor networks," *IEEE TMC*, vol. 4, no. 1, pp. 84–92, Jan. 2005.
- [21] L. K. Au *et al.*, "MicroLEAP: Energy-aware wireless sensor platform for biomedical sensing applications," in *Proc. IEEE Biomed. Circuits Syst. Conf.*, Nov. 2007, pp. 158–162.
- [22] F. Koushanfar, N. Taft, and M. Potkonjak, "Sleeping coordination for comprehensive sensing using isotonic regression and domatic partitions," in *Proc. IEEE Infocom*, Apr. 2006, pp. 1–13.
- [23] J. B. Wendt, S. Meguerdichian, H. Noshadi, and M. Potkonjak, "Energy and cost reduction in localized multisensory systems through application-driven compression," in *Proc. Data Compress. Conf.*, Apr. 2012, p. 411.
- [24] A. I. Daoud *et al.*, "Foot strike and injury rates in endurance runners: A retrospective study," *Med. Sci. Sports Exer.*, vol. 44, no. 7, pp. 1325–1334, Jul. 2012.
- [25] S. Meguerdichian, S. Slijepcevic, V. Karayan, and M. Potkonjak, "Localized algorithms in wireless ad-hoc networks: Location discovery and sensor exposure," in *Proc. MobiHOC*, Oct. 2001, pp. 106–116.
- [26] S. Slijepcevic and M. Potkonjak, "Power efficient organization of wireless sensor networks," in *IEEE In. Conf. Commun.*, Jun. 2001, vol. 2, pp. 472–476.
- [27] V. Goudar and M. Potkonjak, "Dielectric elastomer generators for foot plantar pressure based energy scavenging," in *Proc. IEEE Sensors*, Oct. 2012, pp. 1–4.
- [28] S. Meguerdichian, F. Koushanfar, G. Qu, and M. Potkonjak, "Exposure in wireless ad hoc sensor networks," in *Proc. MobiCom*, Jul. 2001, pp. 139–150.
- [29] J. B. Wendt, V. Goudar, H. Noshadi, and M. Potkonjak, "Spatiotemporal assignment of energy harvesters on a self-sustaining medical shoe," in *Proc. IEEE Sensors*, Oct. 2012, pp. 1–4.
- [30] S. Meguerdichian, H. Noshadi, F. Dabiri, and M. Potkonjak, "Semantic multimodal compression for wearable sensing systems," *IEEE Sensors*, pp. 1149–1453, Nov. 2010.

**Vishwa Goudar** (S'08) is a Ph.D. degree candidate in computer science at the University of California, Los Angeles, CA, USA.

His research interests lie in the area of power optimization and self-sustaining operation of wearable systems.

**Miodrag Potkonjak** (M'02) received the Ph.D. degree in electrical engineering and computer science from University of California, Berkeley, CA, USA, in 1991.

He is a Professor with Computer Science Department, University of California, Berkeley, CA, USA.