

Energy Efficient E-Textile Based Portable Keyboard

Mahsan Rofouei, Miodrag Potkonjak, Majid Sarrafzadeh

Computer Science Department
University of California, Los Angeles (UCLA)
Los Angeles, USA
{mahsan, miodrag, majid}@cs.ucla.edu

Abstract—We have created sensor architecture, data collection and processing techniques for an E-Textile wireless keyboard. We leverage the inherent properties of E-Textiles to produce optimized architecture for energy efficient sensing. Novel techniques such as one where each sensor senses several events (activations of different keys) and each event is sensed by three sensors and flexible interleaved sensing and data processing result in up to a factor of 30 energy reduction over the system where each key is sensed by exactly one sensor. We build the keyboard and test it on multiple subjects.

Keywords- *Soft Keyboard; E-Textiles; Sensing*

I. INTRODUCTION

In the last decade or so, sensor networks emerged as one of the most popular research topics. It seems that almost everything everywhere is going to be instrumented for monitoring and control. Numerous economically important and strategically crucial embedded sensing applications have been advocated from structural building monitoring and water and air pollution tracking to military battlefields and health purposes. The life-time of the sensing system and therefore energy consumption is a premier design metrics. An important observation is that it is more difficult to optimize energy in sensing applications than in other systems since timing constraints are strict. If a particular sample is not collected at a particular time, often there is no way to recover it.

Pressure sensors and smart E-Textiles provide a starting point for a wide spectrum of interesting and important sensing applications including smart shoes, smart bed sheets, instrumented gloves, medical canes and smart dressing items [6, 7]. For all these and many other mobile applications energy is the premier design and operational metrics. An example of such application is a portable keyboard which is designed to be used with small wireless devices such as Personal Digital Assistants (PDAs) and smart phones. Using keyboards with adequate keys to support applications such as writing emails, editing documents and etc. for mobile applications has received a great deal of attention [8,12]. The productivity of the user can be greatly improved by providing full scale and customizable keyboards that are light, low cost and easy to transport and deploy. In addition to enabling the user to use the same keyboard to interact with various devices, there are other important advantages. For example, a large number of diseases are transferred using hand contact. Therefore, shared keyboards such as one used at airports for passenger checking, pose significant health risks that are completely eliminated by E-textile keyboards. In addition, E-textile keyboards enable much better security and user authentication. Finally, the new

keyboards can also collect a variety of statistics about the user that can be used for improving productivity and even possible detection of health induced irregularities.

Our goal in this paper is to study how to organize and operate mobile systems that employ pressure sensors and smart E-Textiles. We design an E-Textile based portable keyboard and energy efficient algorithms for its use. We design sampling and processing methodology that would minimize the energy consumption of the system while detecting all events given samples of different typing sessions on the designed E-Textile keyboard. Our objective can be formulated as in (1) where $E_{Sampling}$ is the energy needed to perform sampling and $E_{Processing}$ is the energy used to perform processing to identify keys.

$$\text{Minimize } E_{Sampling} + E_{processing} \quad (1)$$

We develop architecture, data collection and processing techniques for an E-Textile portable keyboard with ultra low energy requirements. With respect to the base line non-optimized design, we reduce energy up to a factor of 30 in the best case, through development of suitable sensor architecture and development of sampling schedules that decouple the event detection and event identification data collection.

The new paradigm of interleaved and coordinated sampling and data processing poses five technical questions that are essential for minimizing energy while guaranteeing all events are captured and properly processed. We address these questions in the context of the described E-Textile keyboard.

(1) *Which system component and what operational phase to optimize?* We characterized each system component (a single sensor reading and signal processing, one bit computation and communication) in terms of its energy and timing requirements. We see that localized processing is a must and that minimizing the number of sensing readings dominates both overall energy consumption as well as the timing requirements.

(2) *How to create sensing architecture that guarantees to capture all events and requires minimal energy?* It is tempting to allocate a dedicated sensor to each key. However, a huge energy saving is accomplished for some of alternative sensing architectures. Our architecture consists of m vertical and n horizontal sensors that range across the keyboard. Sensor readings are combined as needed using electronic switches. This enables us to create event detection sensor combinations where readings from all sensors are combined. It also enables the organization of effective queries for key identification.

(3) *How to relate events (pressing of a single key) to readings from pressure sensors and organize data processing?*

Our sensor architecture enables us to conduct processing using only binary information about sensor readings; due to experimentally confirmed sensing model where only the closest sensor and its two neighbors detect pressure changes. Therefore, for key identification we use combinatorial search that rapidly eliminates possibilities that are inconsistent with the measurements, from further consideration.

(4) *How to organize measurements such that all timing constraints are satisfied?* We conducted measurements that indicate the maximal speed of typing. We set the longest interval without sampling in such a way that it is minimal, but sufficient for both event detection and additional measurements required for key identification.

(5) *How to exploit information about the events and the user activity?* Application specific information about events can be used to organize data collection in such a way that more likely events are identified earlier in order to save energy.

The rest of the paper is organized as follows: Section II describes related work in this area. We describe preliminaries and sensor modeling using E-Textiles in Section III. Section V describes sensor data processing techniques after presenting a description of design and implementation of the system architecture in IV. We show energy savings in the experimental results section and finally conclusions are drawn in Section VII.

II. RELATED WORK

There are several comprehensive surveys in the area of sensor networks [15]. Energy was considered as one of the most important constraints of sensor networks from early work in this area [11]. Determination of optimal periodic sampling rates and techniques for adaptive sampling has shown its benefits in many embedded sensing environments [13]. [16] has addressed event-driven sensor sampling activation.

There is also a significant body of research on typing characteristics, keyboards and soft keyboards. In the mid of the previous century Shannon information theory attracted tremendous attention. In many fields, model and techniques that leverage Shannon ideas and results were developed. In particular, a number of human psychological models for a variety of tasks and activities have been proposed and analyzed. Among them [3] is one of relatively few that survived test of time. Fitt's model postulated that the speed of human movements is proportional to the time needed for transmission of information through human nervous channels and is calculated in "bits per second". It has been shown experimentally that there is high correlation between the Fitt's index of moving difficulty and actual measured time to complete a task. There have numerous efforts to generalize and improve Fitt's model [4, 5]. Design of soft keyboards using a variety of technologies is an increasingly addressed topic [8, 12]. Also, use of semantic information for more effective text entry [10] and keyboard optimization techniques have been studied [9].

The process of integrating electronics into textiles has been studied for some time now. [14] gives an in-depth overview on smart textiles. [6,7] show some example systems built with E-

Textiles. To the best of our knowledge this is the first effort to create soft keyboards using smart E-Textile pressure sensors. It is also the study that addresses low energy soft-keyboards. Our approach is the first that treats the problem using binary signal processing.

III. PRELIMINARIES

E-Textile is a composite yarn made of fibers coated with conductive polymer. In the event of extra pressure on the surface, the intra fibers are squeezed together and the throughout resistor between the top-bottom surface becomes smaller. Here the resistor is inversely proportional to the pressure imposed. An E-Textile based sensor has a three-stacked layer structure. The sensing material is sandwiched within two conductive pads. In this structure, E-Textile acts like a pressure sensitive resistor. In order to maintain the flexibility and textile feel we use conductive thread and conductive paper for the conductive layer. Figure 1, shows two examples of these sensors made using different conductive material.



Figure 1. The basic sensor structure.

A. Sensor Pressure-Resistance Characteristics

During typing, forces applied are different based on the location of the key or physiological behaviors of users. Here we model the force-resistance relationship in a sensor made with the stacked structure described. Figure 2 shows the curve for forces below 10Ns. [2] performs a detailed study on metrological properties of E-textiles and provides similar curves for other E-Textile material.

B. Sensor Sensitivity with Distance

In fabricating a sensor array, in order to make design easy for mass production, a single sensing layer (middle layer in the three-stacked layer structure) is shared among all array sensors. The result of this design strategy is a dependency of sensors to each other. When applying force on one sensor, the effect would be inevitably posed on its surrounding sensors on account of mechanical linkage. We define the accumulating pressure sensed on each location due to forces applied not directly to itself but to its surrounding locations, the *neighboring effect*.

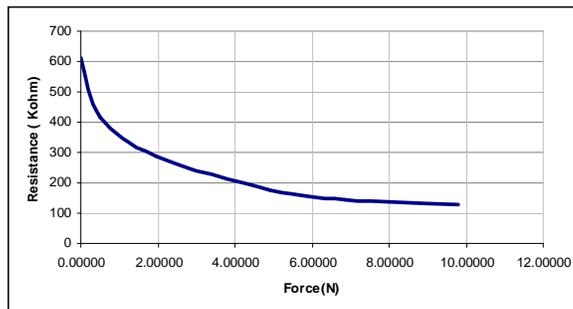


Figure 2. Resistance response based on applied force.

IV. SENSOR ARCHITECTURE AND IMPLEMENTATION

In part A we describe how we design our sensing system using the mentioned characteristics of E-Textiles. In part B we describe the design of system architecture.

A. Sensing Circuit Design

The trivial design of a sensor array would be to assign a single sensor to each key with its own dedicated signal line. The disadvantages of this methodology is in excessive signal lines for each key which results in difficult routing structures for larger numbers of sensors. In addition, this design scheme has a larger sensing circuit which increases the energy consumption of the system by needing to sense more elements to conclude the pressed key. An alternative design mechanism is where sensors on the same row/column share the same wire. In this design methodology, a voltage is applied at the column of interest and the current at each row is measured. This design scheme makes the routing much simpler but still needs $n*m$ readings for an n by m sensor array.

Instead of the aforementioned sensor array designs, we use a two layer design in which each layer consists of parallel sensing elements. The sensor array is constructed by placing the two layers on each other in such a way that the sensing elements on the layers become orthogonal to each other. The touching layer of these sensors is connected to V_{cc} and signals are read from each of the parallel sensors on each layer. Figure 3.a shows the arrangement of different layers in this design. This design mechanism gives us the ability to create a sensing circuit with more sensing locations per sensor element and therefore requires fewer readings. Fig. 3.b shows the virtual view of the sensing array produced upon the placement of the two layers on each other. Therefore by placing keys on the virtual intersections of the horizontal and vertical sensor elements, for n vertical and m horizontal sensing elements, only $n+m$ readings will be required for $n*m$ keys. Keys can be either drawn or sewed on the virtual intersection locations.

Using a two-layer design for our prototype keyboard which includes 102 keys, we would need 17 columns and 6 rows to cover all keys. With this design scheme, rows and columns can be scanned simultaneously to identify the location of the pressed key. Therefore, either columns or rows will dominate the querying time depending on the number of sensors they possess. In the case of our current prototype, the querying is dominated by the columns. Another advantage of a balanced architecture is in terms of sensing. With 17 columns and 6 rows, 23 readings are needed, whereas in a more balanced architecture of 9 and 12, only 21 readings are needed.

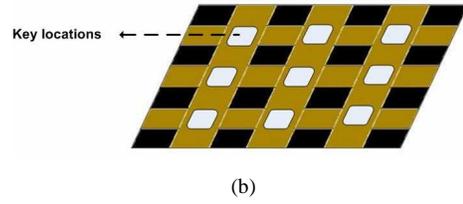
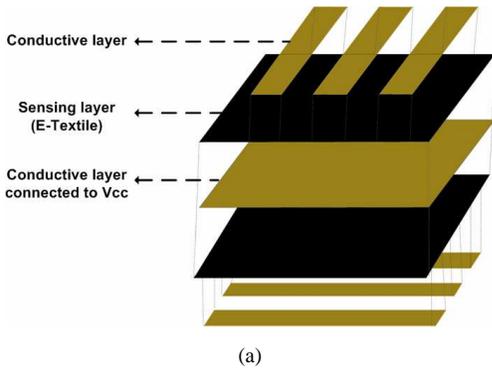


Figure 3. Sensor array design (a) Layer arrangement (b) Virtual view of sensing array and key locations

From the above discussion we conclude that a balanced architecture in terms of number of columns and rows would be more beneficial. For 102 keys, the most balanced architecture would imply 10 columns and 11 rows. However due to geometrical constraints such as the need to preserve the neighboring arrangement of each sensor, a balanced design of 10 and 11 is not trivial. In Fig. 4 we show the sensor arrangements of 9 and 12 columns and rows that produce a fairly balanced architecture. Figure 4.a shows the arrangements of 12 rows instead of the original 6 rows. This layer would become the bottom layer of the keyboard. Note that this arrangement results in a unique pair of sensors for each key. Similarly Fig. 4.b shows the arrangement of 9 columns. Imagining a virtual line in the middle of the layer, each two sensors from each side are merged to create one sensor. This imaginary line is aligned with the line breaking the 6 rows in 12 rows in the bottom layer. Therefore from figures 4.a and 4.b we can see that since 17 is odd we have 8 columns of 12 rows and 1 column of 6.

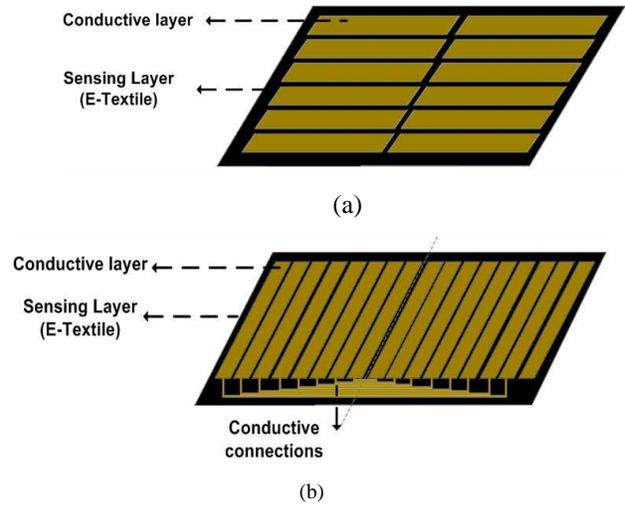


Figure 4. Balanced Sensor Array (a) Rows (b) Columns

The widths of the sensing elements are chosen similar to available keyboards (around 1.3cm). However the choice of the distance between parallel sensing elements is important due to its effect on our detection algorithms. In Section III we described the neighboring effect inherent in sensing elements that share their sensing layers. We choose the distance between consecutive sensing elements in such a way that the neighboring effect is observable on its neighbor sensor but no further. To elaborate, in this design a key press, would result in value changes on maximum of 6 readings (3 columns and 3 rows). By conducting experiments we found that placing the

center of sensors within 1.5 cm spacing will produce the desired neighboring effect. The detection mechanism that leverages this design is described in detail in Section V.

As it will be described in Section V, in our proposed detection algorithm we often need to read sensors in groups. This is done by using summing amplifiers. We chose summing amplifiers since they keep the interaction between inputs minimum and produce an output voltage which is proportional to the algebraic sum of their inputs.

B. System Architecture

Our proposed system is composed of a client part and a host part. In the client part, the pressure sensor array is scanned with the use of multiplexers. After the sensing data is acquired by the microcontroller, it is packaged and transferred to the host receiver side through wireless RF circuit. The microcontroller we chose is MSP430f2274. The A/D converter's resolution is 10 bits, and the sample rate is 10 Hz. The communication protocol used to transfer the data is SimpliciTI and CC2500 is used for the wireless chip. SimpliciTI network protocol is a proprietary low-power radio frequency protocol targeting simple, small RF networks. This network protocol can be considered a complement to ZigBee. The client is connected to the sensing circuit described in the previous section. Typing information is transferred wirelessly to the host side where it is received by the RF chip. The prototype is powered by two AAA batteries. A UART to USB converter (based on MP2010) is designed for interface compatibility. Figure 5 shows the top view image of our prototype E-Textile keyboard.



Figure 5. Prototype keyboard

Power Consumption Model: In this part we derive a power consumption model for the system described above. The total power consumption is composed of the power consumed in the sensing circuit and the transmission sub-system (P_{Com}) shown in equation 2.

$$P_T = P_{Sens} + P_{Com} \quad (2)$$

The sensing power (P_{Sens}) is composed of the power driven by the sensing circuit and sampling power. The former is calculated using resistance in the face of asserted pressure while the later is proportional to the power consumed by ADCs as each sensor reading is through a 10-bit ADC. As it can be seen in Table I, P_{Sens} is dominated by ADC power. This is why group readings are beneficial as they only consume one ADC unit. In the event of group readings the power consumed by summing amplifiers are also added. Table I is calculated according to datasheets [17,18].

From Data transmission power which is in the order of ADC power we see that localized processing is a must and that minimizing the number of sensing readings dominates overall energy consumption.

TABLE I. ENERGY CONSUMPTION OF SUBMODULES

Sub-module	Energy (J)
ADC(per sample)	9×E-9
Data Transmission(per sample)	7.2×E-9
Sensing circuit (average per sensor)	4.8×E-10

V. SENSOR DATA PORCESSING

The trivial method to identify key presses is to read all columns and rows at an extremely fast sampling rate not to miss any keys pressed. However this method requires $n+m$ readings (in our prototype keyboard $17+6=23$) at a fast sampling rate which consumes a lot of energy. In order to reduce the amount of energy consumption we decouple event detection from event identification. We propose methods that would reduce energy consumption in the case of an event.

A. Event Detection

There are two different approaches in identifying events which is dependent on the definition of an event. When events are defined as any pressure asserted on the surface of a keyboard, adding an additional sensing layer would be sufficient for detecting events. Here the design described in Section IV would change to a three layer design where the third layer is a single sensor with an area with the size of the keyboard. However with this solution, forces which are not directed at key locations are also detected as events and the event identification is performed on them.

However if we only target to consider events as forces applied at key locations, we can use a summing amplifier as discussed in Section IV to add the sum of all rows (or all columns) , and only sample this value. In our keyboard design we use the latter method described. We call this sensing element, the global event detector. With this mechanism instead of sampling 23 values per keyboard reading, we only sample one value.

B. Event Identification using Binary Signal Processing

Figure 6 shows readings of the global event detector sensor, columns 2 (V_2) and 3 (V_3) of the keyboard. In the scenario shown in this figure, keys 'A' and 'S' are pressed consecutively. As it can be seen, due to the neighboring effect, when V_2 is activated we see a relatively smaller activation on V_3 and vice versa. In order to minimize the number of samples in order to find the activated key, we use the below reasoning:

We perform a binary mapping from values 0-255 using a found threshold. Values above this threshold are mapped to 1 and the remaining values are considered 0. With this mapping we can claim that any single key press will result in either a pattern of three consecutive 1s or two consecutive 1s at the boundaries. This is shown below, where we show the binary representation of 17 columns in the presence of a single press.

0000000111000000 , or 1100000000000000

This mapping will help in reducing the number of sensor readings. For example, instead of performing n sensor readings to find a key pressed among n sensors, reading every other sensor (plus 1, to cover both boundaries in case of even number of sensors) will suffice ($\lceil n/2 \rceil + 1$).

As described, our architecture design enables group readings of sensors. A group reading of n sensors after the binary mapping stage, will result in a '1' if one of its n sensors is equal to 1. In other words, a group reading of n is an OR of its n inputs. We leverage the above group reading ability and binary representation of sensors to minimize the number of searches needed to identify the pressed key in the case of an event.

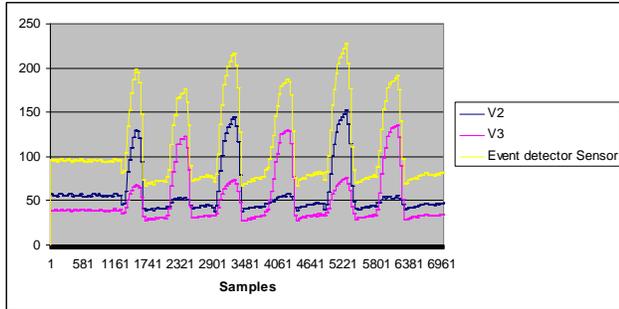


Figure 6. Global event detector sensor, column 2 and 4 readings while pressing 'A' and 'S' keys consecutively.

For each event detected by the global event detector sensor, depending on the available time to perform processing, the number of sensor readings can be reduced. To elaborate, if the available time restricts all the reasoning to be performed in one epoch (time unit), then taking into account the binary representation of sensors, by reading every other sensor we can reason which key is pressed. However if the available time allows two epochs for reading and reasoning, using group readings, we can narrow down the search space by a factor of 2,3 or more, in the first epoch and then search the reduced space with every other readings in the next epoch. This mechanism can be extended to more epochs with same reasoning. Since we have a 1 MIPS processor, we assume all the computations necessary for key identification to be performed in one epoch.

Using this reasoning we have calculated the optimal number of readings with group reading arrangements for different epochs for rows and columns of our prototype keyboard using a onetime static exhaustive algorithm. The expected value of number of readings required for rows and columns are shown in Table II.

TABLE II. COMBINATORIAL SEARCH METHOD.

Epochs	Number of readings for columns (9)	Number of readings for rows (12)
1	5	7
2	3.55	4
3	3.22	3.66
4	3.22	3.66

Here, the question that arises is how we can determine the number of available epochs for measurement and reasoning. This number depends of the global event detector sampling

mechanism. In an extreme case, a very fast sampling in the global event detector sensor will capture events early enough to enable multiple epoch processing. On the other extreme, sampling at a very slow rate that is only fast enough to capture the presence of events, would allow one epoch processing. This can be further explained with the example in Fig. 7. This figure shows an event happening in the global event detector signal where point C is the end of the event. If the event detector sampling can assure the discovery of an event at an early point such as A, the time available for event identification would be t_{AC} . However if the global event detector sampling is only fast enough to guarantee the capture of events as late as points such as B, the remaining time for event identification becomes only t_{BC} .

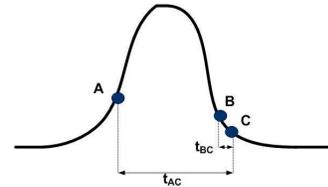


Figure 7. Sampling points on event detection signal.

Therefore to minimize the overall energy consumption of the system, we need to solve an optimization problem. More epochs available, enable more optimized measurements and therefore less energy consumption in the event identification phase. On the other hand increasing the number of epochs requires a higher sampling rate on the event detection sensor and therefore higher energy consumption in the event detection phase. This optimization problem can be solved depending on different parameters and application constraints.

Please note that deciding the optimal querying is performed in the keyboard design stage and thus is static. Based on optimal querying mechanism, necessary summing amplifiers are used in the circuit. The sampling mechanism in the global event detector sensor will determine the number of epochs available for processing. We set the longest interval without sampling in such a way that it is minimal, but sufficient for both event detection and additional measurements required for key identification. To formulate this, if T_S represents the time of typing session and T_p is considered minimum duration of a typing pulse, we can approximate the number of samples needed to be performed by the global event detector sensor as shown in equation 3. In this case, depending on different choices of available time ($T_{Available}$) different stages can be chosen from Table II to identify the pressed key.

$$N_{Samples} = \frac{T_S}{T_p - T_{available}} \quad (3)$$

The method described above is for single key presses at instant. In some situations a user may press two or more keys simultaneously. We can address the situation when two keys are pressed using an optimal algorithm that is similar to the case of single key. Resolving situations where three or more keys are simultaneously pressed induces too high energy overhead. Since it happens rarely in actual use of keyboard, we leave it up to the typist to correct such instances.

VI. EXPERIMENTAL RESULTS

In this section we show the energy savings resulting from sampling and processing strategies described. We gathered typing data from different people and performed statistical analysis to find the necessary parameters for our proposed sampling scheme. One of these parameters is the duration of a key press pulse. This amount is different among different people and is also different for the same person in different situations using different keys. A direction for future work is studying these pulses under different typing situations (eg. stress) and using them for customization. Here, we only consider the minimum pulse duration we found over all people which is 250ms. We use this in determining the sampling frequency of the global sensor detector using equation 3.

We compare three different strategies in Table III. Method 1 is energy results for using non-intelligent sampling of all sensors at 10 Hz sampling intervals for key identification. However this method is still better than a trivial non-optimized method since it uses a near optimized sensing architecture which only requires $n+m$ readings for $n*m$ keys as described. Method 2, uses statistical information on duration of typing forces to reduce sampling intervals on the optimized sensing architecture. In Method 3, we use our proposed combinatorial search in binary representation of signals with epoch time intervals. The session time in below experiments is 20 seconds.

TABLE III. ENERGY CONSUMPTION OF SAMPLING TECHNIQUES (NJ)

Typing Speed (chars per sec)	Method 1	Method 2	Method 3
1.75	41.4	15.12	2.89300
0.5	41.4	15.12	1.34500
2.2	41.4	15.12	3.45028
5	41.4	15.12	6.91780

As Table III shows using Method 3 we are able to achieve savings from 5x to 30x in energy consumption depending on typing speeds compared to the non-optimized version (Method 1). Comparing to Method 2 which uses statistical data of typing force durations, we achieve 2x to 11x savings in energy consumption. Overall as expected, if the user is an excellent typist the savings are minimal (for our fastest typist, still 5x) and if the user is a slow typist the savings are very significant. To show the efficiency of our combinatorial search sensing we show the different situations of Method 3 based on available epochs and their energy consumptions in Table IV.

TABLE IV. ENERGY CONSUMPTION FOR COMBINATORIAL SEARCH IN BINARY REPRESENTATION (NJ)

Typing Speed (chars per sec)	1 epoch	2 epochs	3 epochs	4 epochs
1.75	4.5	3.10114	2.89300	2.89594
0.5	1.8	1.40239	1.34500	1.34794
2.2	5.472	3.71269	3.45028	3.45322
5	11.52	7.51789	6.91780	6.92074

Therefore considering altogether savings, we achieve up to a factor of 30 energy reduction compared to the non-optimized

method and up to a factor of 11 compared to Method 2 which uses intelligence on typing force durations. The lifetime of the system using the most optimized sensing mechanism is more than 250 hours with two AAA batteries. This is significant energy reductions comparing to current available foldable keyboards [19, 20] which offer 90 hours life time.

VII. CONCLUSIONS

In this work we described the design and implementation of architecture, data collection and processing techniques for an E-Textile based wireless keyboard. We leveraged the inherent properties of E-Textiles to produce an optimized architecture for sensing by reducing the number of sensing elements and developed sampling techniques that separate event detection from event identification and use minimum number of samples to reduce the system energy usage. We achieved great energy improvements with respect to the base line non-optimized design, up to a factor of 30 based on user typing speed.

ACKNOWLEDGMENT

This work was supported in part by the NSF under awards CNS-0958369, CNS-1059435, and CCF-0926127.

REFERENCES

- [1] H. Beker, and F. Piper, "Cipher Systems: The Protection of Communications", 1982.
- [2] Del Prete, et al. "A novel pressure array sensor based on contact resistance variation: metrological properties". Rev. Scientific Instruments 72, 2001, pp. 1548-1553.
- [3] P.M. Fitts, "The information capacity of the human motor system in controlling the amplitude of movement". Journal of Experimental Psychology, Vol. 47, 1954, pp. 381-391.
- [4] I.S. MacKenzie, "Fitts' law as a research and design tool in human-computer interaction", Human- Computer Interaction, 1992, pp. 91-139.
- [5] W. Soukoreff, et al, "Theoretical upper and lower bounds on typing speeds using a stylus and keyboard", Behaviour & Info Tech, 1995, pp 370-379.
- [6] H. Noshadi, et al. "Energy Optimization in Wireless Medical Systems Using Physiological Behavior", ACM/BMES Wireless Health, 2010, pp 128-136.
- [7] M. Rofouei, et al "Computing with Uncertainty in a Smart Textile Surface for Object Recognition." IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, 2010, pp. 174-179.
- [8] M. Hunter, S. Zhai, B.A. Smith, "Physics-based graphical keyboard design", CHI Human factors in computing systems, 2000.
- [9] M. Raynal, N. Vigouroux, "Genetic algorithm to generate optimized soft keyboard", CHI on Human factors in computing systems, 2005, pp. 1729-1732.
- [10] J. Goodman, et al. "Language modeling for soft keyboards", Intelligent user interfaces, 2002, pp. 419-424
- [11] D. Ganesan, et al. "Complex Behavior at Scale: An Experimental Study of Low-Power Wireless Sensor Networks", Technical Report UCLA/CSD-TR 02-0013, 2002.
- [12] D. Ashbrook "Context sensing with the Twiddler Keyboard". ISWC 1999. pp 197-198.
- [13] M. A. Batalin , et al. "Call and response: experiments in sampling the environment", Proceedings of the 2nd international conference on Embedded networked sensor systems, 2004, pp. 25-38.
- [14] D. Marculescu et al., "Electronic Textiles: A Platform for Pervasive Computing," Proc. IEEE, 2003, pp. 1995-2018.
- [15] D. Culler, M. Srivastava and D. Estrin, Editors, Guest Editors' Introduction: " Overview of Sensor Networks", IEEE 2004, pp. 41-49.
- [16] S. Jevtic , et al. "Lucid dreaming: reliable analog event detection for energy-constrained applications", IPSN 2007, pp. 350-359.
- [17] <http://focus.ti.com/docs/prod/folders/print/msp430-adc10.html>
- [18] <http://focus.ti.com/docs/toolsw/folders/print/ez430-rf2500t.html>
- [19] <http://www.wirelessground.com/blfoblke.html>
- [20] <http://www.onlypre.com/blfoke.html>