DVFS Based Task Scheduling in a Harvesting WSN for Structural Health Monitoring

A. Ravinagarajan
Computer Science and Engineering
University of California San Diego,
La Jolla, CA, USA
aruna@cs.ucsd.edu

D. Dondi
Computer Science and Engineering
University of California San Diego,
La Jolla, CA, USA
ddondi@ucsd.edu

T Simunic Rosing
Computer Science and Engineering
University of California San Diego,
La Jolla, CA, USA
tajana@ucsd.edu

Abstract— **The task scheduler of an energy harvesting wireless sensor node (WSN) must adapt the task complexity and maximize the accuracy of the tasks within the constraint of limited energy reserves. Structural Health Monitoring (SHM) represents a great example of such an application comprising of both steady state operations and sporadic externally triggered events. To this end, we propose a task scheduler based on a Linear Regression Model embedded with Dynamic Voltage and Frequency Scaling (DVFS) functionality. Our results show an improvement in the average accuracy of a SHM measurement, setting it at 80% of the maximum achievable accuracy. There is also an increase of 50% in the number of SHM measurements.**

Keywords - Energy harvester (EH); Task manager; Structural Health Monitoring (SHM); DVFS;

I. INTRODUCTION

THE process of monitoring structures for the purpose of damage identification is known as structural health monitoring (SHM). Structural health monitoring requires knowledge of the undamaged state of the structure as a means of comparison, as well as continual comparison of periodic measurements. This can be separated into two basic categories: *rapid event assessment* and *periodic lifetime monitoring*. While *rapid event assessment or External Request* addresses the need to obtain data from a structure immediately following a significant event (such as an earthquake); *periodic lifetime monitoring or steady state operation* seeks to identify damage that accumulates over long periods of time [1].

The SHM is an excellent sample application for an energy harvesting wireless sensor network (WSN). The adoption of wireless sensor networks in advanced Structural health monitoring (SHM) systems has proliferated in the last few decades [2] due to their ability to operate reliably without human intervention in inaccessible areas. This has been made possible by the usage of wireless communication and environmental energy harvesters (EH) [3]. However, adopting an EH as the main energy supply limits the device's level of activity to the availability of energy in the environment. The most common energy source suited to outdoor SHM applications is solar energy, which is often inconsistent in its availability.

A significant challenge in this type of system is the management and conservation of energy while maintaining the minimum level of QoS required by the particular deployment. The embedded SHM system must consume only as much energy as the energy harvester can collect from the environment [4]. Therefore, the task scheduler must ensure that task allocation is matched to the available energy.

In order to achieve behavior, Kansal et al.[4] present a harvesting theory that uses duty cycling to account for changes in the harvested energy. But for many applications, duty cycling alone is not sufficient. Systems containing a variety of interdependent tasks require a more detailed task model.

Moser et al. [5] present a simulation work in power management in EH systems using discrete service levels associated with specific rewards. The energy harvesting rate in the future K 'frames' is calculated using a predictor algorithm, and a definite set of "service levels" are formulated for a hypothetical workload. They use dynamic programming to select the optimal service level. While optimal, [5] demonstrate their solution with only one constraint (i.e. energy constraint) and not for a number of constraints acting upon the scheduler at the same time. Also, this work has not been extended to any practical application.

The work presented by J.Steck et al. [6] also uses the concept of "discrete service levels", where each level is associated with a "utility" value. This scheduler was applied for a Structural Health Monitoring (SHM) system. They designed an "iterative search" algorithm to meet the task deadlines with multiple performance constraints acting simultaneously. Different service thresholds are set and the energy availability is checked against these thresholds in order to vary task scheduling. The main limitation by setting discrete service levels is the under-utilization of the available energy.

In this paper, we developed a linear regression based algorithm that relates the energy consumption, execution time and data accuracy to the number of tasks and their complexity. This model is then used to maximize the system performance with the constraint of energy availability. By using a linear regression based model, we utilize all the available energy to improve the task accuracy and number of measurements performed. We also incorporate a Dynamic Frequency and Voltage Scaling (DVFS) methodology to increase the efficiency of system energy utilization.

In our results, we demonstrate the superiority of our linear regression algorithm based task scheduler over the iterative search based task scheduler [6] in terms of average accuracy of a measurement, number of performed measurements and number of external requests that can be serviced.

978-3-9810801-6-2/DATE10 © 2010 EDAA

The rest of the paper is organized as follows. Section II provides a description of SHiMmer, the SHM device adopted as target system for the task scheduler, and an overview of the considered SHM tasks. The proposed algorithm for tasks management is described in Section III. Section IV shows the simulation results obtained by implementing the algorithm with and without the DVFS capability. Finally, the conclusions are presented in Section V.

II. SHiMmer: An Embedded System for Structural Health Monitoring

SHiMmer is a standalone embedded system designed for active structural health monitoring. The implemented active SHM consist in analyzing a structure through the application of ultrasonic waves and the acquisition of the wave propagation response of the structure. Collecting information about wave propagation allows detecting and localizing structural damages based on the Lamb Waves method [7].

SHiMmer platform comprises three boards: a digital board based on a ADI BlackFin DSP [8] for data analysis and wireless networking, running a custom Linux-based OS; an analog board that manages 16 independent channels and generates high-voltage active SHM pulse for the lead-zirconate-titanate (PZT) sensors; and a power manager board that collects energy from a solar energy harvester, stores the energy into a Supercapacitor, generates the required power supply rails and manages a Li-Ion battery acting as back-up energy reservoir. Figure 1 shows SHiMmer development board for preliminary laboratory tests.

SHiMmer allows performing ultrasonic SHM analysis stimulating the structure with high-voltage pulses applied through 16 independent PZT sensors. The main advantage of active SHM measurement is the flexibility to adjust the stimulation process to the particular structure conditions.

SHiMmer implements the active SHM measurement following two procedures, *Actuation* and *Sensing*, which are performed recursively over all the possible pairs of PZT sensors. During the *Actuation*, the adopted SHM pulses are custom generated by the DSP, amplified by the analog board up to $30\ V_{pp}$, and transmitted to the structure selecting 1 of the 16 PZT sensors.

During the *Sense*, the structure response is selectively collected through each of the remaining 15 sensors, filtered through an anti-aliasing block, amplified by the analog board up to $\pm 1V$, and acquired by the digital board up to *25 MSPS.* Due to the high-voltage amplification and high sampling-rate acquisition, *Actuation* and *Sense* procedures are high energy consuming activities, representing a significant and irreducible part of the overall SHiMmer power requirement.

SHM data analysis consists of three main steps: reducing the correlated and uncorrelated noises from the SHM signal, extracting the amplitude and frequency information of the measurements, and finally correlating the data provided by the selected set of PZT sensors to detect the presence of damage in the structure. Figure 2 shows the list of tasks involved in the SHM analysis and the accuracy provided by each set of tasks.

The first set of task that is performed is *Actuation and Sensing*, which consists of the iterative measurement with ultrasonic waveform through all the combinations provided by the 16 PZT sensors. The next stages of tasks are responsible for reducing the noise during measurement. These are *averaging* and *filtering*. The former reduces the uncorrelated noise by calculating the average over multiple sequences of samples in each path. The latter filters the signal in the frequency domain by means of a FFT computation of the acquired SHM response and subsequently eliminating the unwanted frequency components, which correspond to noise.

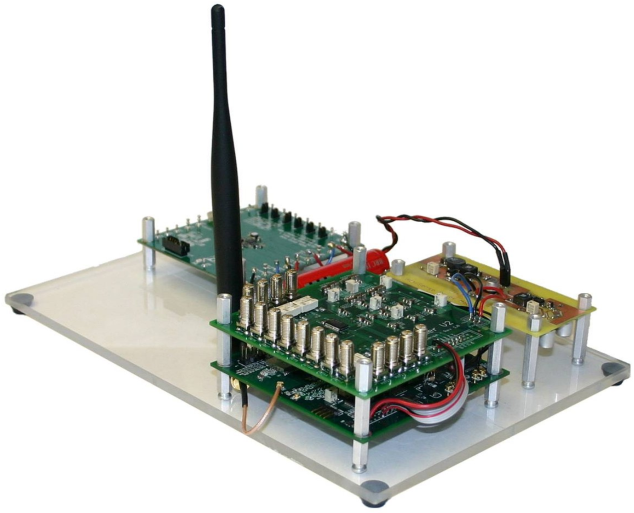

Figure 1. SHiMmer development board for laboratory SHM tests.

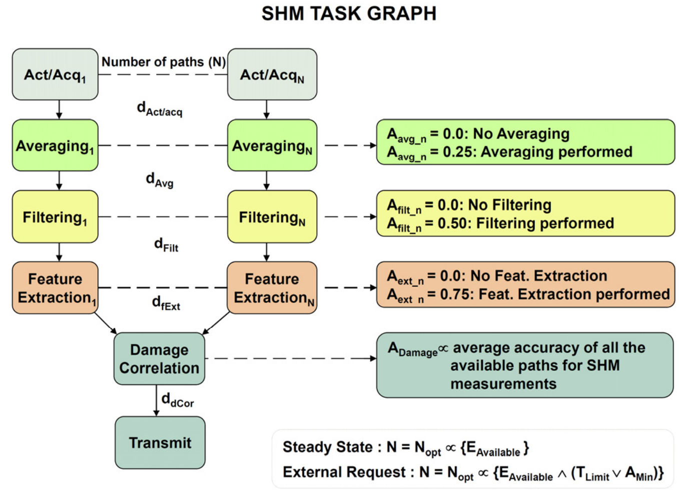
SHM TASK GRAPH
Act/Acq$_1$
Number of paths (N)
Act/Acq$_N$
$d_{Act/acq}$
Averaging$_1$
Averaging$_N$
A_{avg_n} = 0.0: No Averaging
A_{avg_n} = 0.25: Averaging performed
d_{Avg}
Filtering$_1$
Filtering$_N$
A_{filt_n} = 0.0: No Filtering
A_{filt_n} = 0.50: Filtering performed
d_{Filt}
Feature Extraction$_1$
Feature Extraction$_N$
d_{fExt}
A_{ext_n} = 0.0: No Feat. Extraction
A_{ext_n} = 0.75: Feat. Extraction performed
Damage Correlation
$A_{Damage} \propto$ average accuracy of all the available paths for SHM measurements
d_{dCor}
Transmit
Steady State : $N = N_{opt} \propto \{E_{Available}\}$
External Request : $N = N_{opt} \propto \{E_{Available} \wedge (T_{Limit} \vee A_{Min})\}$

Figure 2. The Structural Health Monitoring task graph shows the relation between the tasks and the accuracy of each path, n, for all steps of analysis.

The amplitude and frequency are obtained from a SHM response by convolving the filtered set of data with a set of reference SHM responses, called baseline signals. This task is called *feature extraction.* To detect the presence of damage in the structure, all the SHM data is combined using a damage correlation function, thereby obtaining a map showing the current structure health.

The selection of the ADI BlackFin represents the optimum trade-off between high-performance and low-power characteristics. The former is due to a very efficient design of the core targeted to high speed complex data analysis; the latter is due to the native *dynamic voltage and frequency scaling* (DVFS) algorithm, which allows reducing both the core voltage and frequency.

Based on the DVFS capability, the system has been designed to select one of three different working modes: *Active High*, f_1, where the core is running at 1.2V@300MHz, *Active Low*, f_2, where the core runs at 0.85V@150MHz, and a low-power idle mode where the core is set at 0.85V@75MHz, thereby reducing the system power consumption to 50mW. The reduction of BlackFin core voltage and frequency between the two active modes, allows savings of 30% in the energy consumption.

Table I shows the comparison of energy consumption and computational time of the BlackFin core performing a basic operation of sum and multiplication under the two active modes. The adoption of DVFS and the two active modes, *High* – f_1 and *Low* – f_2, will be described in detail in Section III.

TABLE I. ADI BLACKFIN ENERGY CONSUMPTIONS WITH DVFS

DVFS mode	Sum (Integer value)		Multiply (Integer value)	
	E_{EX} [nJ]	T_{EX} [μs]	E_{EX} [nJ]	T_{EX} [μs]
Active High	274	0.751	336	0.892
Active Low	192	0.932	235	1.098

III. TASK SCHEDULER IMPLEMENTATION

The primary goal of the proposed task scheduler is to maximize the accuracy and number of SHM measurements with respect to the energy consumption and the execution time constraints. In particular, since each measurement has several tasks to be executed, maximizing accuracy and number of SHM measurements consists of performing the maximum number of tasks accordingly to system constraints.

The proposed scheduler maximizes the system performance adopting a set of linear regressions that relate the energy consumption, the execution time and the data accuracy. Indeed, SHiMmer prototype behavior has shown that SHM algorithms' energy consumption, execution time and accuracy are linearly dependent on the amount of analyzed data provided by the set of executed tasks. For this reason, adopted linear regression allows the scheduler to adjust number of task and their complexity to the available energy. Furthermore, in order to improve the efficiency of the energy utilization the proposed task scheduling algorithms uses the DVFS policy provided by the DSP to optimize both the system computation capability and energy consumption.

The task scheduler algorithms manage the allocation of sequences of tasks according to the amount of available energy. At each time interval, the scheduler determines the energy available in the system buffer by tracking the energy collected by the EH as well as the energy consumed by system activities, as expressed by (1).

$$E_{Buffer}\big|_t = E_{Buffer}\big|_{t-1} + E_{EH}\big|_t - E_{Consumed}\big|_t \tag{1}$$

where $E_{Consumed}$ is calculated at run time by the task scheduler, and E_{EH} refers to the amount of energy collected by the energy harvester during the time elapsed since last time interval. E_{EH} takes into account the efficiency of the photovoltaic panels and the leakage of the systems, as shown in (2).

$$E_{EH}\big|_t = E_{PV\ panel}\big|_t \cdot \eta_{system} - E_{Leakage}\big|_t \tag{2}$$

where $E_{PV\,panel}$ is the energy provided by the photovoltaic panel, η_{system} is the efficiency of the power supply sub-system and $E_{Leakage}$ is the sum of the energy leakage of the system, e.g. the supercapacitor self-discharge rate. The proposed task scheduler algorithm exploits the linear regression model and the DVFS functionality in both the Steady State and the External Request operations.

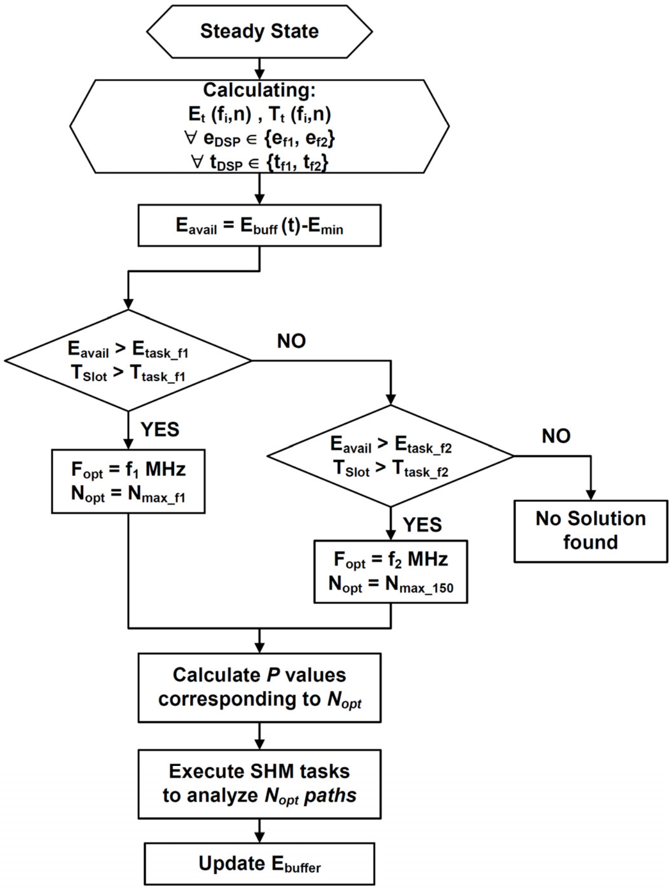
Steady State
Calculating:
E_t (f_i,n) , T_t (f_i,n)
∀ e_{DSP} ∈ {e_{f1}, e_{f2}}
∀ t_{DSP} ∈ {t_{f1}, t_{f2}}
E_{avail} = E_{buff} (t)-E_{min}
E_{avail} > E_{task_f1}
T_{Slot} > T_{task_f1}
NO
YES
F_{opt} = f_1 MHz
N_{opt} = N_{max_f1}
E_{avail} > E_{task_f2}
T_{Slot} > T_{task_f2}
NO
YES
No Solution found
F_{opt} = f_2 MHz
N_{opt} = N_{max_150}
Calculate *P* values corresponding to N_{opt}
Execute SHM tasks to analyze N_{opt} *paths*
Update E_{buffer}

Figure 3. Block diagram of the steady state algorithm, which automatically selects the DVFS working mode and maximize the SHM task allocation.

A. Steady State Algorithm:

Steady state operation is typical of sensor nodes; data is gathered, processed, and transmitted according to a predefined schedule over a long period of time. The goal of steady state operation is to periodically execute a set of tasks at the highest possible accuracy and also to maximize the number of measurements that can be done while maintaining a desired level of energy. The block diagram describing the developed algorithm is shown in Fig. 3.

The methodology of this algorithm is to ascertain a relationship between the number of paths, *n*, frequency, f_i, and the energy, E_t , time consumed, T_t, Accuracy, A_t,. Using curve-fitting, we obtained three linear regression expressions to estimate both the energy consumed and the execution time required to analyze different workloads, as shown in (3).

$$\begin{cases} E_t(f_i,n) = e_{DSP}(f_i,n) \cdot n \\ T_t(f_i,n) = t_{DSP}(f_i,n) \cdot n \\ A_t(f_i,n) = a_{DSP}(f_i,n) \cdot n \end{cases} \tag{3}$$

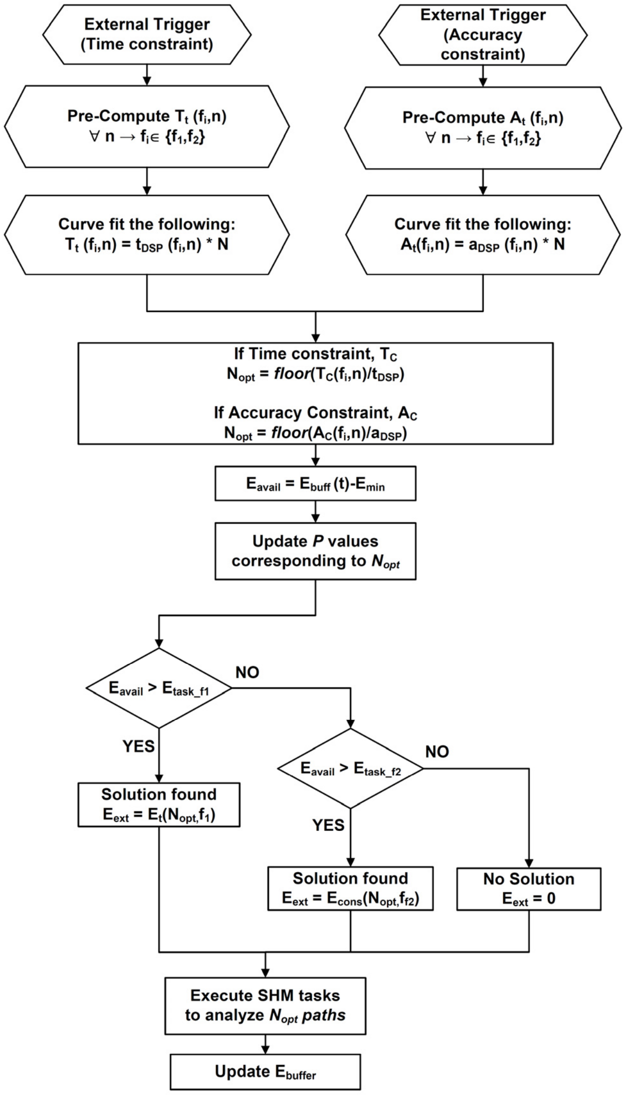
External Trigger (Time constraint)
External Trigger (Accuracy constraint)
Pre-Compute T_t (f_i,n) ∀ n → f_i∈ {f_1,f_2}
Pre-Compute A_t (f_i,n) ∀ n → f_i∈ {f_1,f_2}
Curve fit the following: T_t (f_i,n) = t_{DSP} (f_i,n) * N
Curve fit the following: A_t(f_i,n) = a_{DSP} (f_i,n) * N
If Time constraint, T_C
N_{opt} = floor(T_C(f_i,n)/t_{DSP})
If Accuracy Constraint, A_C
N_{opt} = floor(A_C(f_i,n)/a_{DSP})
E_{avail} = E_{buff} (t)-E_{min}
Update P values corresponding to N_{opt}
E_{avail} > E_{task_f1}
NO
YES
E_{avail} > E_{task_f2}
NO
YES
Solution found E_{ext} = E_t(N_{opt},f_1)
Solution found E_{ext} = E_{cons}(N_{opt},f_2)
No Solution E_{ext} = 0
Execute SHM tasks to analyze N_{opt} paths
Update E_{buffer}

Figure 4. Block diagram of the external request algorithm, which adopts the linear approach to meet the constraint requested by the external device.

where e_{DSP} refers to the amount of energy consumed to analyze a single SHM path, t_{DSP} refers to the execution time for a single SHM analysis and a_{DSP} refers to the achievable accuracy for a single SHM analysis. These linear relations between energy or time with the amount of SHM analysis allow the proposed task scheduler to maximize the number of measurement with respect to the available energy E_{avail}.

$$E_{avail}\big|_t = E_{Buffer}\big|_t - E_{min} \tag{4}$$

where E_{min} is the minimum energy the system needs to guarantee external request response capability. Typically the interval between the two SHM measurements is denoted as T_{total}. However, external trigger requests take a finite amount of time to execute, T_{ER}, thereby effectively reducing the time available for performing the steady state operations. This effective time available is denoted as T_{Slot}.

$$T_{Slot} = T_{total} - T_{ER} \tag{5}$$

To calculate the maximum number of paths that can be executed within the specified time limit, T_{Slot}, for the operating frequencies f_1 and f_2, the energy availability is checked against various pre-defined thresholds to determine the optimal (N_{opt} , f_{opt}) pair. The thresholds are calculated using (6) and (7).

$$E_t(f_i) = e_{DSP}(f_i) \cdot T_{Slot}/t_{DSP}(f_i) \tag{6}$$

If the energy available exceeds E_{f1}, then the tasks are executed at the frequency f_1 with a maximum of $T_{Slot} / t_{DSP} (f_1)$ or $N_{max\text{-}f1}$ number of paths. If the energy available is less than E_{f2}, then the tasks are executed at the frequency f_2 with N_{opt} number of paths, obtained from (7).

$$E_t(f_i, n) = e_{DSP}(f_i, n) \cdot N_{opt} \tag{7}$$

Since f_2 consumes less energy than f_1 for performing the task with the same accuracy level, we choose f_2 for lower energy available.

On the other hand, if the E_{avail} falls between E_{f1} and E_{f2}, then tasks are executed at a frequency f_2 at $T_{Slot} / t_{DSP} (f_2)$ or $N_{max\text{-}f2}$ number of paths. In this manner, the scheduler selects the optimal number of paths $N_{opt} (f_i)$ and frequency f_i.

Depending on the N_{opt} calculated, the scheduler generates the corresponding task priority values, P. The $E_t (N_{opt} , f_{opt})$ is calculated for this set of P values by executing the tasks as per the SHM task graph Fig.2. The buffer energy for the next time instant is updated into the energy buffer using (1).

B. *External Request Algorithms :*

The external request mode provides direct control of SHiMmer's capabilities to an external device. Because the relative importance of each task may change depending on the system's current conditions, the task's priority is expected to change in a system over time. By doing so, the algorithm dynamically selects the proper workload to accomplish energy neutrality, while satisfying the performance constraints. The block diagram describing the developed algorithm is shown in Fig. 4.

We have modeled our external request scheduler algorithm around two different performance constraints. First, the algorithm determines the maximum accuracy given that it needs to complete the task within a time limit (time constraint). Second, the algorithm can execute tasks at a specified accuracy value (accuracy constraint).

This algorithm determines the set of task utilities based on a constraint limit. The heuristic consists of two basic steps: 1) In the case of *Time constraint:* Finding the maximum number of paths executable within the time limit. In the case of *Accuracy constraint*: Finding number of paths that guarantee the requested accuracy 2) Satisfying the energy constraint.

Just as in the steady state algorithm, the external request algorithm also pre-computes the parameters such as energy consumed $E_{task} (n, f)$, accuracy $A(n)$ and time consumed $T_{task} (n, f)$ for a number of data paths, n, and frequency, f, associated with a node. By curve-fitting, we obtained two linear regression expressions to estimate the time and energy required to analyze different workloads. These are represented in (3).

a) Time Constraint Algorithm (TC): When an external device sets a maximum execution time limit, T_C, optimal number of paths, $N_{opt,TC}$ (f_i) is calculated by extending inferences from (3) into (8) as shown.

$$N_{opt,TC}(f_i, n) = TC/t_{DSP}(f_i, n) \tag{8}$$

The energy $E_{task}\,(n, f)$ corresponding to $N_{opt,TC}\,(f_i)$ is calculated from (9).

$$E_{task}(f_i, n) = e_{DSP}(f_i, n)\ \cdot\ N_{opt,TC}(f_i, n) \tag{9}$$

The time constraint algorithm tries to maximize the accuracy of the measurement by servicing the request at the higher DVFS active mode. This allows the algorithm to maximize the number of external request that can be successfully served. It first checks if the energy required by the f_1 fits the energy budget. If not, f_2 is selected.

b) Time Constraint Algorithm (TC): When an external device sets a maximum execution time limit, A_C, optimal number of paths, $N_{opt,AC}\ (f_i)$ is calculated by extending inferences from (3) into (10) as shown.

$$N_{opt,AC}(f_i, n) = AC/a_{DSP}(f_i, n) \tag{10}$$

The energy $E_{task}\,(n,f)$ corresponding to $N_{opt,AC}\,(f_i)$ is calculated from (10).

$$E_{task}(f_i, n) = e_{DSP}(f_i, n)\ \cdot\ N_{opt,AC}(f_i, n) \tag{11}$$

Unlike the time constraint algorithm, the accuracy constraint algorithm tries to maximize the energy savings of the measurement by servicing the request at the lower DVFS active mode. It first checks if the energy required by the f_2 fits the energy budget. If not, f_1 is selected. This is done since the accuracy is directly dependent on the number of path. Hence, reducing the DSP working mode allows performing the same analysis with lower energy.

IV. Simulation Results

To validate the proposed algorithms, a full SHM task scheduler has been implemented and simulated. The input to the system is a real distribution of solar energy, comprising sunny, cloudy and variable weather conditions. The various measurement thresholds of the Supercapacitor and the DVFS active frequency mode values were obtained from the SHiMmer development board, shown in Table I. The task sequence listed in the SHM task graph in Fig.2 was also given as an input to the scheduler.

Figure 5 shows the simulated solar energy distribution over ten days. The proposed algorithm based on linear regression and DVFS is compared against the iterative search algorithm of Steck et al. [6]. In addition, the usefulness of DVFS is also demonstrated.

A. Steady State Results

Figure 6 shows the comparison of the task scheduler algorithms in terms of total number of SHM measurements performed in a day. By implementing a linear task allocation to vary the number of task paths, the scheduler is able to maximize the usage of the available energy. It is note-worthy that this method performs 50% more number of SHM measurements as compared to the iterative approach. Moreover, as shown in Fig. 6, the adoption of DVFS policy allows a further increase of 15-20% in the number of SHM measurements. This is due to the versatility of the system to select the low power active mode during low energy conditions, such as night, to allocate a higher number of measurements.

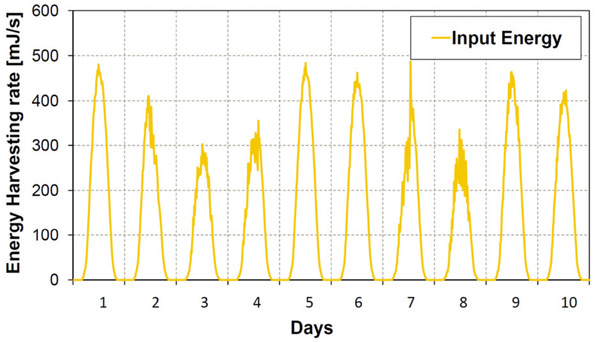

Figure 5. Input solar energy distribution over a simulation duration of 10 days.

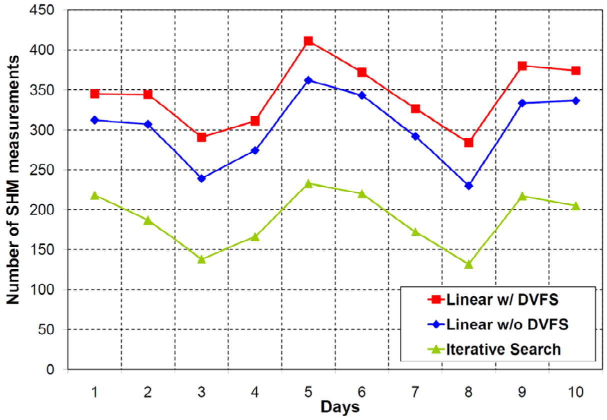

Figure 6. Comparison of the maximum number of SHM measurements between the three algorithms over 10 days of SHM activity.

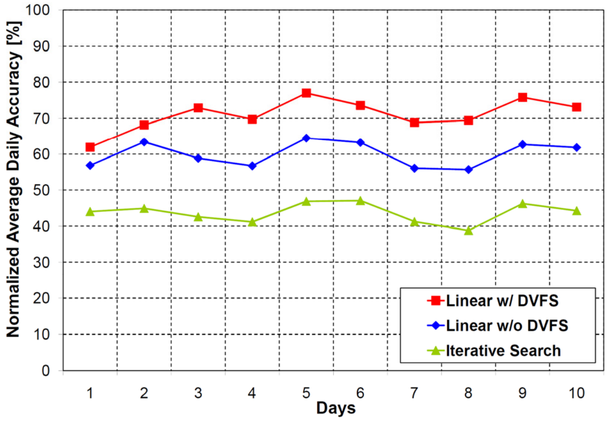

Figure 7. Comparison of the normalized daily average SHM accuracy between the three algorithms over 10 days of SHM activity.

Figure 7 shows the comparison between the normalized daily average SHM accuracy of the three schedulers. The normalization adopted in these simulations consists of scaling the average accuracy with the maximum accuracy achievable. These results are in line with the results for the number of SHM measurements. The adoption of a linear task allocation allows the execution of tasks with a higher accuracy, which in turn leads to a higher daily average accuracy. We also observe that the adoption of DVFS policy increases the performance by another 10%, thereby fixing the average SHM accuracy at 70-80% of the maximum achievable accuracy.

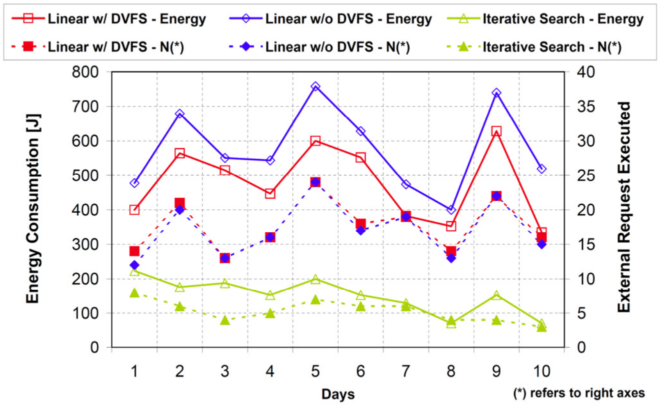

Figure 8. Comparison of the number of served external request and energy consumption for the proposed algorithms in external request operative mode.

B. *External Request Results*

In order to examine the behavior of the algorithms for the external request operation, the three schedulers have been simulated considering two metrics: the number of external requests the system can service and the amount of energy consumed to service the requests.

Figure 8 shows the behavior of the three algorithms for an average number of 25 external requests per day. It can be seen that the iterative search algorithm can service only between 15-30% of the requests, due to its discrete service level classification. On the other hand, our linear algorithm can adjust the system performance to optimize its capability to allocate extra measurements. In particular, our linear model (with and without DVFS) can service up to 95% of the requests. It is also seen from Fig.8 that by adopting the DVFS policy into the scheduler, the system can reduce the amount of energy needed to service the same external constraints, thereby, lowering the system energy requirement.

V. Conclusions

In this paper we proposed a task scheduler for SHM applications designed to maximize both the number and the accuracy of SHM measurements on the energy harvested wireless embedded device. The proposed scheduler uses a Linear Regression based algorithm along with DVFS to adjust system workload and energy consumption to both available environment energy and desired SHM measurements QoS.

The simulation results of the implemented linear algorithm show an increase of more than 50% in the number of daily SHM measurements over the iterative search method [8]. The linear algorithm with DVFS can get within 20% of the maximum achievable daily average accuracy. Finally, our proposed scheduler can satisfy up to 95% of the external requests and, using the DVFS feature, is able to significantly reduce the amount of energy needed to perform the additional requested measurements.

References

[1] C. Farrar, K. Worden, “An introduction to structural health monitoring,” *in Phil. Trans. Roy. Soc. London*, vol. A365, Issue 1851, p. 303-315, December 2006

[2] J. A. Paradiso and, T. Starner, “Energy scavenging for mobile and wireless electronics,” *IEEE Pervasive* Computing, 2005.

[3] Dondi, D.; Bertacchini, A.; Larcher, L.; Pavan, P.; Brunelli, D.; Benini, L., "A solar energy harvesting circuit for low power applications," *Sustainable Energy Technologies, 2008. ICSET 2008. IEEE International Conference on* , pp.945-949, 24-27 Nov. 2008.

[4] A. Kansal, J. Hsu, M. Srivastava and, V. Raghunathan, “Harvesting aware power management for sensor networks,” in *Proceeding of the 43rd annual conference on Design and Automation,* 2006.

[5] C. Moser, L. Thiele and, J. J. Chen “Power management in energy harvesting systems with discrete service levels,” In *Proceedings of the ACM/IEEE conference on Internationl symposium on low power elctronics and design*, 2009

[6] J. B. Steck and, T. Rosing, “Adapting Task Utility in Externally Triggered Energy Harvesting Wireless Sensing Systems,” to be presented at the *6th International Conference on Networked Sensing Systems*, 2009

[7] S. S. Kessler, S. M. Spearing and, C. Soutis, “Damage detection in composite materials using Lamb Wave methods,” in *Proceeding of the American Society for Composites*, Sept. 2001

[8] Analog Devices BlackFin DSP, http://www.analog.com/en/embedded-processing-dsp/blackfin/content/index.html