# Energy Efficient Sensor Nodes Powered by Kinetic Energy Harvesters – Design for Optimum Performance

Tom J. Kaźmierski, Leran Wang, and Mansour Aloufi

***Abstract*—In an energy harvester powered wireless sensor node system, as the energy harvester is the only energy source, it is crucial to configure the microcontroller and the sensor node so that the harvested energy is used efficiently. This paper outlines modelling, performance optimisation and design exploration of the complete, complex system which includes the analogue mechanical model of a tunable kinetic microgenerator, its magnetic coupling with the electrical blocks, electrical power storage and processing parts, the digital control of the microgenerator tuning system, as well as the power consumption models of sensor node. Therefore not only the energy harvester design parameters but also the sensor node operation parameters can be optimised in order to achieve the best system performance. The power consumption models of the microcontroller and the sensor node are built based on their operation scenarios so that the parameters of the digital algorithms can be optimised to achieve the best energy efficiency. In the proposed approach, two Hardware Description Languages, VHDL-AMS and SystemC-A is used to model the system's analogue components as well as the digital control algorithms which are implemented in the microcontroller and the sensor node. Simulation and performance optimisation results are verified experimentally. In the development of the fast design exploration tool based on the response surface technique, the response surface model (RSM) is constructed by carrying out a series of simulations. The RSM is then optimised using MATLAB's optimisation toolbox and the optimisation results are presented.**

***Index Terms*—Wireless sensor node, Energy harvesting, Performance optimisation, Simulation.**

## I. Introduction

WIRELESS sensor networks (WSNs) have attracted a great research interest in recent years. Since wireless sensor nodes can provide information from previously inaccessible locations and from previously unachievable number of locations, many new application areas are emerging, such as environmental sensing [1], structural monitoring [2] and human body monitoring [3]. Although wireless sensor nodes are easy to deploy, the lack of physical connection means they must have their own energy supply. Because batteries have limited lifetime and are environmentally hazardous, it has become widely agreed that energy harvesters are needed for long-lasting sensor nodes [4]–[6]. The idea is to use energy harvester to capture small amounts of energy from the environment and use the generated energy to power the nodes in wireless sensor networks. Vibration-based energy harvesters are used in many commercial applications since mechanical vibrations are widely present. Most of the reported vibration energy harvester designs are based on a spring-mass-damper system with a characteristic resonant frequency. These devices normally have a high Q-factor and generate maximum power when their resonant frequency matches the dominant frequency of the input ambient vibration [7]. Consequently, the output power generated by the microgenerator drops dramatically when there is a difference between the dominant ambient frequency and the microgenerator's resonant frequency. Tunable microgenerators, which can adjust their own resonant frequency through mechanical or electrical methods to match the input frequency, are therefore more desirable than the fixed frequency microgenerators [8]. A wireless sensor node powered by tunable energy harvester typically has the following key components (Fig. 1) [9]: a microgenerator which converts ambient environment vibration into electrical energy, a power processing circuit which regulates and stores the generated energy, an actuator used for the frequency tuning mechanism, a digital controller that monitors and retunes the tunable energy harvesting system based on vibration

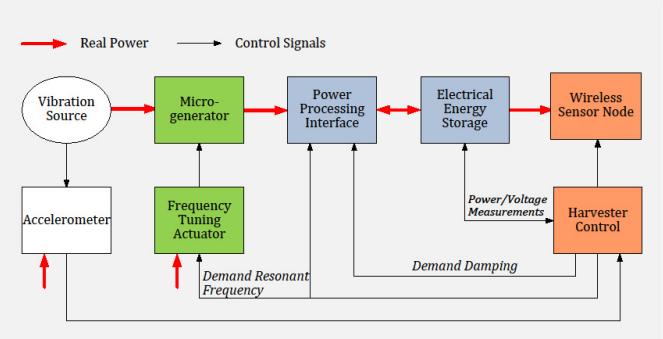



Fig. 1. Components of a energy harvester powered sensor node system [9].

Manuscript received 1 May 2012. Accepted for publication 30 May 2012. Some results of this paper were presented at the 4th Small Systems Simulation Symposium, Niš, Serbia, February 12-14, 2012.

The authors are with the Faculty of Physical and Applied Sciences, University of Southampton, Southampton SO17 1BJ, UK, email: {tjk,lw04r,ma08r}@ecs.soton.ac.uk}.

DOI: 10.7251/ELS1216065K

measurements from an accelerometer, and the wireless transceiver or transmitter.

Hardware description languages, such as VHDL-AMS and SystemC-A, have been used to model energy harvesters in recent years [10], [11]. HDLs with mixed signal and multi-domain capabilities are suitable for energy harvester modelling because an energy harvester is naturally a mixed-physical-domain system. The technique outlined below models the complete system including the analogue mechanical, magnetic and electrical power storage and processing parts, the digital control of the microgenerator tuning system, as well as the power consumption models of sensor node. Additionally, the paper proposes a response surface based design space exploration and optimisation technique so that not only the energy harvester design parameters but also the sensor node operation parameters can be optimised in order to achieve the best system performance

## II. Performance Optimisation

An automated energy harvester design flow must be implemented holistically and based on a single software platform that can be used to model, simulate, configure and optimise an entire energy harvester systems. Such a design flow is outlined in the pseudo-code of Algorithm 1 and also shown in Fig. 2. Naturally, the process starts with initial design specification, such as the available energy source (light, heat, vibration, etc), environmental energy density, device size, minimum voltage level/power output. According to these specifications, HDL models are constructed from component cells available in the component library. The component library contains parameterised models of different kind of micro-generator structures (solar cell, electromagnetic, piezoelectric, etc), various booster circuit topologies and storage elements. The outer loop in the algorithm represents this structure configuration process, which involves examining and comparing those HDL models from the library with the aim of identifying a set of components that meet specific user requirements. The inner design flow loop will then find the best performance of each candidate design by adjusting

**Algorithm 1** Automated energy harvester design flow.

```
Initial design structure and specification
Structure configuration loop:
for all design structures do
  Build HDL model of design
  Optimisation loop:
  repeat
    Simulate and evaluate performance
    if best performance not achieved then
      Update design parameters
    end if
  until best performance achieved
  if there are more structures to try then
    Select new structure
  end if
end for
```

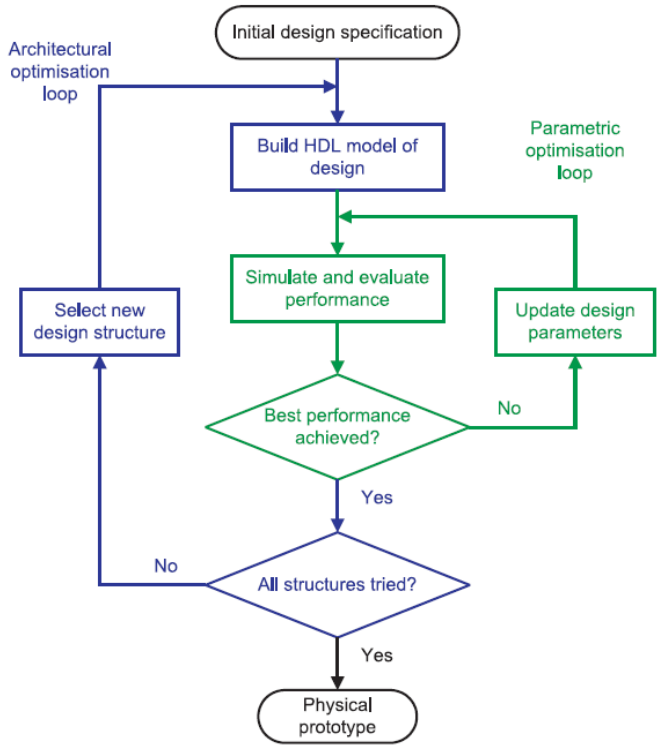



Fig. 2. Energy harvester design flow.

electrical and non-electrical parameters of the design's mixed-technology HDL model. The parametric optimisation of the generated structure will further improve the energy harvester efficiency by employing suitable optimisation algorithms. The design flow ends with the best performing design subject to user-defined performance characteristics.

Requirements for energy harvester component models are: 1) models need to be computationally efficient for fast performance optimisation when used in complete energy-harvester systems and yet accurate; these are conflicting requirements, 2) models need to capture both theoretical equations and practical non-idealities required for accurate performance estimation. The models should support different mechanical-electrical structures and will be expressed in terms of HDL descriptions. They will be able to predict the behaviour of the actual device accurately while remaining reconfigurable.

A small HDL model library of energy harvester components has been built. It contains two types of micro-generator, each of which can be configured with different coils (wire diameter of 12/16/25 μm), and two types of voltage multipliers that have three to six stages. The voltage transformer has not been included because it cannot be made and tested with available resources. But the simulation based optimisation of energy harvester with voltage transformer has been performed and will be discussed in Section II-A2. The configuration target has been set to find the set of components that can charge the 0.047F super capacitor to 2V in shortest time. These values were chosen because there has been reported energy harvester systems that use 0.047F storage capacitor and 2V working voltage [12]. Simulations of every available energy harvester

configuration were carried out simultaneously and a process has been developed to automatically track the best model. SystemVision VHDL-AMS simulator [13] has been used as the single software platform. The outcome design is listed in Table I.

It is no surprise that the micro-generator II has been chosen because it is larger and stores more kinetic energy. However, it is quite interesting that the coil with the largest wire diameter, which leads to the fewest number of turns, and the VM with the fewest stages have been chosen. To further investigate this result, more simulations have been done and an important trade-off between the electromagnetic micro-generator and the VM voltage booster has been found as explained below.

Fig. 3 shows the charging waveforms of Type I micro-generator connected to the same 5-stage VM but configured with different coils. At the beginning, the energy harvester with 25 µm wire diameter charges the quickest and the 12 µm configuration charges the slowest while the 16 µm one is in between. But the 25 µm configuration also saturates quickly and reaches the 2V mark slower than the 16 µm energy harvester. Due to simulation time limitation, the figure does not show how the other two waveforms end. But it could be foreseen that the 16 µm configuration will also saturate at some point while the 12 µm one reaches highest voltage.

Similar results have been obtained from the voltage booster end. Fig. 4 shows the charging waveforms of Type II micro-generator with 25 µm coil connecting with 3, 4 and 5 stages Dickson VMs. It can be seen that the energy harvester with the 3-stage VM charges the super capacitor to 2V first and the one with the 5-stage VM can reach the highest voltage.

The above results prove that when different components of an energy harvester are combined, the gain at one part may come at the price of efficiency loss elsewhere, rending the whole system less efficient than expected. This observation is very useful for the development of future, more complicated systems and model libraries. Below we evaluate the performance loss due to the close mechanical-electrical interaction (micro-generator and voltage booster) that takes place in the energy harvester system.

TABLE I
PARAMETERS OF THE CONFIGURATION RESULT

| Micro-generator | Type II |
|---|---|
| Wire diameter | 25 $\mu$m |
| Voltage booster | 3-stage Dickson voltage multipler |

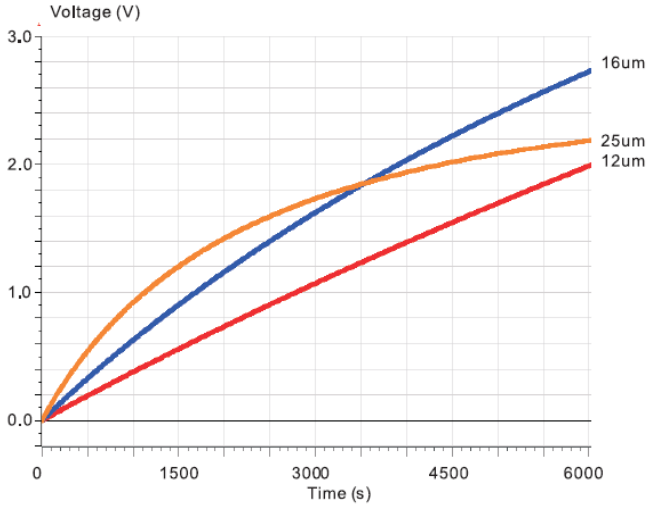



Fig. 3. Simulation of Type I micro-generator with different coils.

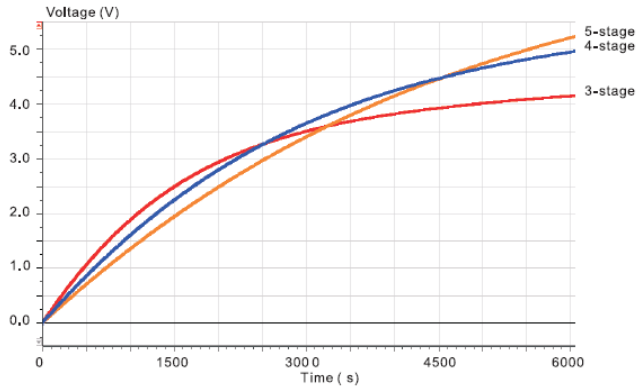



Fig. 4. Simulation of Type II micro-generator with different VMs.

### A. *Performance Optimisation*

The loss expressed in terms of energy harvesting efficiency is:

$$\eta_{Loss} = \frac{E_{Harvested} - E_{Delivered}}{E_{Harvested}} \quad (1)$$

In the proposed design flow, the generated energy harvester design should be parameterised such that automated performance optimisation will be able to further improve the energy harvester efficiency by employing suitable optimisation algorithms. The optimisation objective investigated in the case study below is to maximise the charging rate of the super capacitor.

1) *Exhaustive search:* The micro-generator parameters that can be optimised are related to the coil size, i.e the thickness ($t$) and the outer radius ($R$). Other components such as the magnets and cantilever determine the resonant frequency of the micro-generator and thus should be determined from the application requirements. The optimised parameters of the voltage booster are the capacitor values of each VM stage. The entire energy harvester is optimised as an integrated model and the parameter search space is summarised in Table II.

The optimisation is based on the concurrent simulations of design instances from uniformly sample the search space and track the best result (Fig. 5). Other optimisation algorithms may also be employed and we show in Section II-A2 how a VHDL-AMS based genetic optimisation was successfully applied to the integrated optimisation of an energy harvester system.

To validate the effectiveness of the proposed approach, the

TABLE II
OPTIMISATION SEARCH SPACE

| Coil thickness(mm) | 1.0-1.3 |
|---|---|
| Coil radius(mm) | 2.0-2.45 |
| Capacitor values(uF) | 47/100/150 |

following simulations and experimental measurements have been carried out.

*Original design:* combines Type II micro-generator with a 5 stage Dickson VM. This VM has been reported in literature as the optimal configuration [14]. However, in the original design these two parts are optimised separately, which is quite common in existing energy harvester design approaches. Parameters of the original design are listed in Table III.

*Optimised design:* has been obtained using the proposed design flow (Fig. 5). Table IV gives the new micro-generator and voltage booster parameters.

The impact of these values on improving the energy harvester performance has been validated in both simulation and experimental measurements. According to the optimisation result, a new coil has been manufactured by Recoil Ltd, UK [15] which replaced the original one in the validation (see Fig. 6).

Simulation and experimental waveforms of the original and optimised design are shown in Fig. 7. As can be seen from the figure, there is good a correlation between the simulation and experimental waveforms in both of the energy harvester designs, which validates the effectiveness and accuracy of the proposed design flow. The energy harvester from original design can charge the super capacitor to 2V in 6000 seconds while the optimised design only uses 1500 seconds, which represents a 75% improvement.

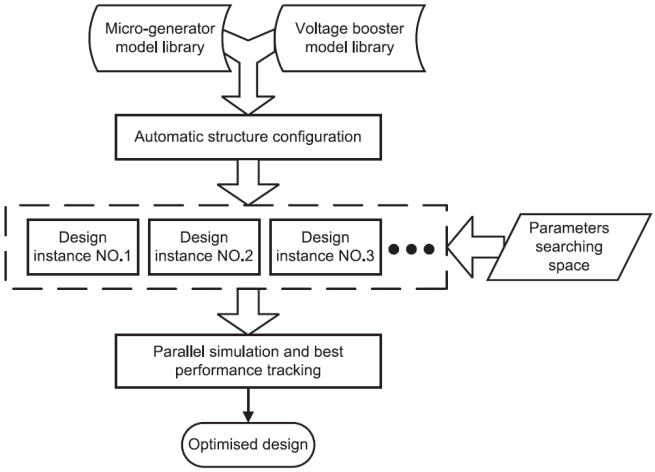



Fig. 5. Implementation of the proposed energy harvester design flow in VHDL-AMS.

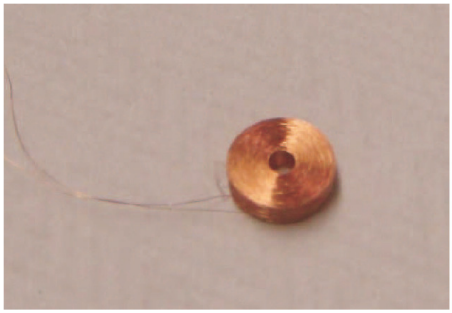


Fig. 6. New coil according to optimisation result ($R$=2.0mm, $r$=0.5mm, $t$=1.3mm, $d$=25μm).

TABLE III
PARAMETERS OF ORIGINAL ENERGY HARVESTER

| Micro-generator | |
|---|---|
| Wire diameter($\mu$m) | 16 |
| Coil thickness(mm) | 1.3 |
| Coil radius(mm) | 2.45 |
| Voltage booster | |
| VM configuration | 5-stage Dickson |
| Capacitor values(C1-C5,uF) | 47,150,150,47,150 |

TABLE IV
PARAMETERS OF OPTIMISED ENERGY HARVESTER

| Micro-generator | |
|---|---|
| Wire diameter($\mu$m) | 25 |
| Coil thickness(mm) | 1.3 |
| Coil radius(mm) | 2.0 |
| Voltage booster | |
| VM configuration | 3-stage Dickson |
| Capacitor values(C1-C3,uF) | 100,100,47 |

2) *Genetic optimization:* This section demonstrates another possible optimisation method to improve the energy harvester efficiency. Fig. 8 shows that in the proposed approach, not only the energy harvester model but also the optimisation algorithm is implemented in a single VHDL-AMS testbench. The parameters used for the optimisation are from both the micro generator and the voltage booster. The optimisation object is to increase the charging rate of the super capacitor. The optimisation algorithm generates design parameters to the model and obtains the charging rate through simulation. The optimisation loop runs continuously until the design parameters reach an optimum.

A super capacitor of 0.22F has been used in the performance optimisation experiment. The micro-generator parameters that can be optimised are the number of coil turns ($N$), the internal resistance ($R_c$) and the outer radius ($R$). The voltage booster circuit here is a voltage transformer. The optimisation parameters are the number of turns and the resistance of the transformer's primary and secondary windings. For proof of concept, a genetic algorithm (GA) [16] has been employed to optimise the energy harvester with a voltage transformer booster. The implemented GA has a population size of 100 chromosomes. Each chromosome has 7 parameters (3 from the micro-generator and 4 from the voltage booster). The crossover and mutation rate are 0.8 and 0.02 respectively. Other optimisation algorithms may also be applied based on the proposed integrated model. The "un-optimised" model parameters are given in Table V.

Applying the proposed modelling and performance optimisation, Table XIV gives the new micro-generator and voltage booster parameters which are referred to as the "optimized" design. The impact of these values on improving the charging of the super is shown in Fig. 9. As can be seen from the simulation results, in 150 minutes the un-optimised energy harvester charges the super capacitor to 1.5V and the optimised energy harvester reaches 1.95V, which represents a 30% improvement.

Performance of the developed GA has been further investigated by comparing the power transfer efficiency before and after optimisation. The maximum average power that can

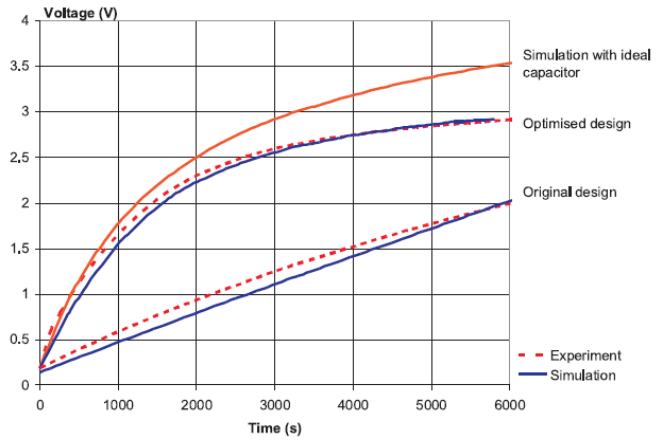



Fig. 7. Simulation and experimental waveforms of original and optimized energy harvesters.

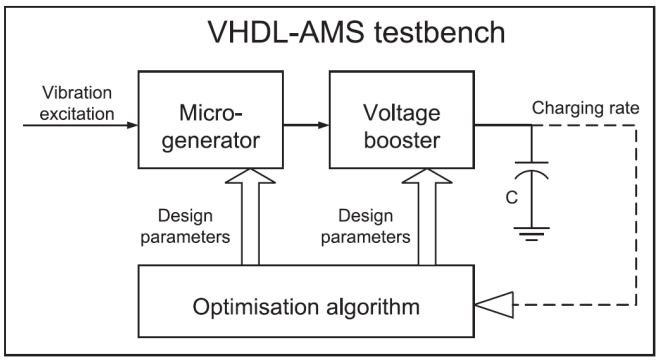



Fig. 8. Integrated performance optimisation in VHDL-AMS testbench.

be delivered to the electrical domain is about 144 μW. Table VII lists the average electrical power output from the micro generator and the voltage transformer. It can be seen that the optimisation improves the efficiency of both the micro generator and voltage booster, which validates the effectiveness of the developed genetic optimisation.

## III. Complete Wireless Sensor Node

Fig. 10 shows the diagram of the wireless sensor node system powered by tunable energy harvester. The wireless sensor node has a temperature sensor and a 2.4GHz radio transceiver. Once activated, the measured data are transmitted to another transceiver which is connected to a PC's USB port. The microgenerator converts the input vibration into electrical energy. The generated AC voltage is rectified by a diode bridge and stored in a 0.55F supercapacitor. The supercapacitor acts as the energy source for the microcontroller that controls the frequency tuning of the microgenerator and for the sensor node. In order to tune the resonant frequency of the microgenerator to match the frequency of the vibration source, the microcontroller uses two input signals, one from the microgenerator and one from the accelerometer. The operational amplifier acts as a comparator to generate square waves from the microgenerator output so that it is easy for the microcontroller to calculate the frequency. The detailed tuning algorithms are presented in

TABLE V
Parameters of Un-optimized Energy Harvester

| Micro-generator | | |
|---|---|---|
| Outer radius of coil ($R$) | | 1.2 mm |
| Coil turns ($N$) | | 2300 |
| Internal resistance ($R_c$) | | 1600 Ω |
| Voltage transformer | | |
| | Resistance(Ω) | No. of turns |
| Primary winding | 400 | 2000 |
| Secondary winding | 1000 | 5000 |

TABLE VI
Parameters of GA Optimized Energy Harvester

| Micro-generator | | |
|---|---|---|
| Outer radius of coil ($R$) | | 1.1 mm |
| Coil turns ($N$) | | 2100 |
| Internal resistance ($R_c$) | | 1400 Ω |
| Voltage transformer | | |
| | Resistance(Ω) | No. of turns |
| Primary winding | 340 | 1900 |
| Secondary winding | 690 | 3800 |

TABLE VII
Energy Harvester Power Efficiency

| | Generated power($\mu$W) | Delivered power($\mu$W) | Overall efficiency |
|---|---|---|---|
| Pre-optimisation | 26.875 | 15.750 | 10.94% |
| Post-optimisation | 29.250 | 19.625 | 13.63% |

Section III-A3. The microcontroller also provides energy for the accelerometer, the operational amplifier and the actuator so that these devices can be turned off when not in use. Table VIII lists the type and make of the system components.

### A. System Component Models

1) *Tunable microgenerator:* Fig. 11(a) shows a diagram of the electromagnetic microgenerator together with its tuning mechanism. The microgenerator is based on a cantilever structure. The coil is fixed to the base, and four magnets (which are located on both sides of the coil) form the proof mass. The tuning mechanism uses magnetic force to change the effective stiffness of the cantilever which leads to a change of resonant frequency. One tuning magnet is attached to the end of the cantilever beam and the other tuning magnet is connected to a linear actuator. The linear actuator moves the magnet to the calculated desired position so that the resonant frequency of the microgenerator matches the frequency of the ambient vibration. The control algorithm is modelled as a SystemC digital process described in Section III-A3. Fig. 11(b) shows a photo of the microgenerator which is used to validate the proposed technique [17].

The dynamic model of the microgenerator is [18]:

$$m\frac{d^2z(t)}{dt^2}+c_p\frac{dz(t)}{dt}+k_s z(t)+F_{em}+F_{t\_z}=F_a \qquad (2)$$

where $m$ is the proof mass, $z(t)$ is the relative displacement between the mass and the base, $c_p$ is the parasitic damping factor, $k_s$ is the effective spring stiffness, $F_{em}$ is the electromagnetic force, $F_{t\_z}$ is the $z$ component of tuning force $F_t$ and $F_a$ is the input acceleration force. The $z$ component of tuning force is:

$$F_{t\_z}=F_t\frac{z(t)}{l_c} \qquad (3)$$

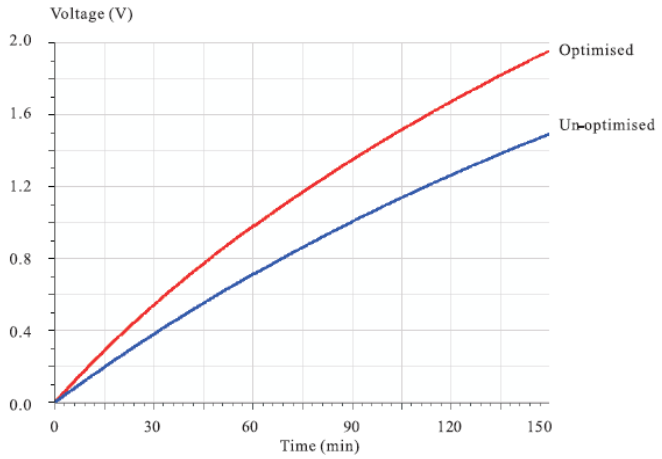



Fig. 9. Simulation waveforms of super capacitor charging by different energy harvester models.

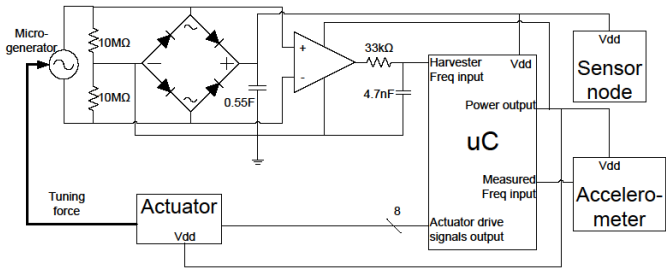



Fig. 10. System diagram of a tunable energy harvester powered wireless sensor node.

where $l_c$ is the length of the cantilever. The resonant frequency $\omega_0$ and damping coefficient $\zeta$ are:

$$\omega_0 = \sqrt{\frac{k_s}{m}} \tag{4}$$

$$\zeta = \frac{c_p}{2\sqrt{mk_s}}. \tag{5}$$

The resonant frequency of the tuned microgenerator ($f_r$') is:

$$f_r' = f_r\sqrt{1+\frac{F_t}{F_b}} \tag{6}$$

where $f_r$ is the un-tuned resonant frequency, $F_t$ is the tuning force between two magnets and $F_b$ is the buckling load of the cantilever. The electromagnetic voltage generated in the coil is:

$$V_{em} = -\Phi\frac{dz(t)}{dt} \tag{7}$$

where $\Phi$=$NBl$ is the transformation factor and $N$ is the number of coil turns, $B$ is the magnetic flux density and $l$ is the effective length. The output voltage is:

$$V_m(t) = V_{em} - R_c i_c(t) - L_c\frac{di_L(t)}{dt} \tag{8}$$

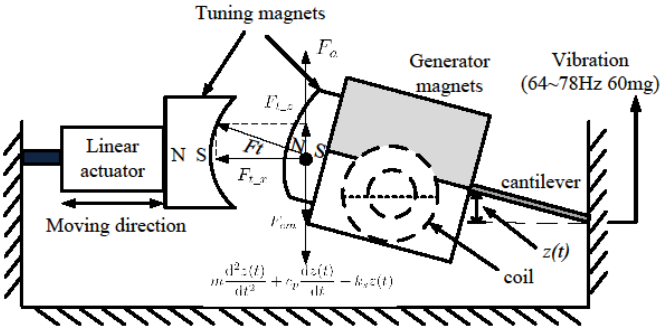
where $R_c$ and $L_c$ are the resistance and inductance of the coil respectively and $i_c(t)$ is the current through the coil. The electromagnetic force is calculated as:

$$F_{em} = \Phi i_c(t). \tag{9}$$

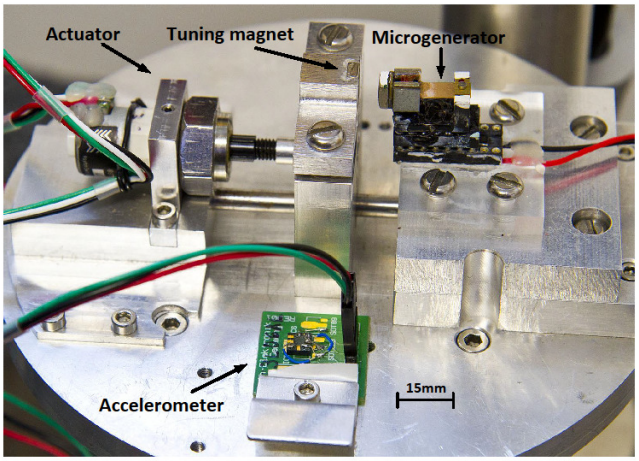
TABLE VIII
SYSTEM COMPONENTS POWERED BY THE ENERGY HARVESTER

| Component | Type | Make |
|---|---|---|
| Microcontroller | PIC16F884 | Microchip |
| Accelerometer | LIS3L06AL | STMicroelectronics |
| Linear actuator | 21000 Series Size 8 stepper motor | Haydon |
| Sensor node | eZ430-RF2500 | Texas Instruments |

2) *Energy-aware sensor node behavior and power consumption model:* The eZ430-RF2500 wireless sensor node from Texas Instruments has been used in the system. The on-board controller is the MSP430F2274 and is paired with the CC2500 multi-channel RF transceiver, both of which are based on low-power design. The sensor node (Fig. 12) monitors the environment temperature as well as the supercapacitor voltage. Once activated, it transmits the temperature and voltage values through the radio link. Transmissions do not involve receiving acknowledgements. A program has been developed for the sensor control module to configure the sensor node in an energy-aware manner, namely that its transmission interval should depend on the available energy on the supercapacitor. The sensor node behaviour is summarised in Table IX. The transmission interval when the supercapacitor voltage is above 2.8V, i.e more energy stored, has been chosen as one parameter for optimisation. Although it is desirable to have as many transmissions as possible during a fixed time period, it may not always be the case that the transmission interval should be set as small as possible. This is because if the transmission is so frequency that the sensor node uses more energy than the harvester can generate, the supercapacitor




(a) Mechanical part




(b) Photo of tunable microgenerator

Fig. 11. Tunable electromagnetic microgenerator.

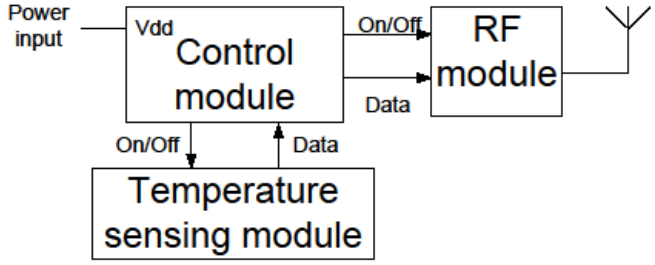



Fig. 12. Block diagram of the sensor node.

TABLE IX
SENSOR NODE BEHAVIOR BASED ON SUPERCAPACITOR VOLTAGE

| Supercapacitor voltage | Wireless transmission interval |
|---|---|
| Below 2.7V | No transmission |
| Between 2.7 and 2.8V | Every 1 minute |
| Above 2.8V | Every 5 seconds (parameter for optimisation) |

voltage will drop below 2.8V and the transmission interval will increase in order for the energy storage to recover. Other factors such as frequency tuning also uses stored energy and therefore will affect how much energy is available for the sensor node.

In order to characterise the power consumption model of the sensor node, the current draw of the sensor node has been measured during each transmission. The results are listed in Table X.

The supply voltage was kept at 2.9V. So during each transmission lasting 4.5 ms, the sensor node consumes 227 μJ of energy and the equivalent resistance of its energy consumption model is:

$$R_{node} = \begin{cases} 167\ \Omega & \text{when in transmission} \\ 5.8\ \text{M}\Omega & \text{when in sleep} \end{cases} \quad (10)$$

3) *Tuning algorithms and power consumption models:* In order for a energy harvester powered wireless sensor node (Fig. 1) to work autonomously, all the system components need to be powered by the harvested energy. The pseudo code of the tuning algorithm is shown in Algorithm 2. Standard SystemC modules were used to model the digital control process and in the experimental verification the control algorithm was implemented in a PIC16F884 microcontroller. As can be seen in Algorithm 2, a watchdog timer wakes the microcontroller periodically and the microcontroller first detects if there is enough energy stored in the supercapacitor. If there is not enough energy, the microcontroller goes back to sleep and waits for the watchdog timer again. If there is enough energy, the microcontroller will then compare the frequency of the microgenerator signal, which is close to the input vibration frequency, to the microgenerator's resonant frequency. When a difference is detected between the vibration frequency and the resonant frequency, the microcontroller

TABLE X
CURRENT DRAW OF THE SENSOR NODE

| Operation | Time | Current |
|---|---|---|
| Sleep mode | N/A | 0.5$\mu$A |
| Wake-up | 1 ms | 4.5 mA |
| Sensing | 1.5 ms | 13.4 mA |
| Transmission | 2 ms | 26.8 mA |

**Algorithm 2** Harvester tuning control algorithm

```
1: repeat
2:    Energy generation while waiting for watchdog timer: 320 seconds (parameter for optimisation)
3:    if Enough energy stored in the supercapacitor ($V_s$ ≥2.6V, where 2.6V is the minimum voltage for the actuator to start) then
4:       Turn on Timer1 (clock frequency as parameter for optimisation)
5:       repeat
6:          Measure microgenerator period
7:       until 8 cycles have been measured
8:       Turn off Timer1
9:       Calculate input vibration frequency from 8 measurements
10:      Find optimum position (8-bit) of tuning magnet through look-up table which has been pre-obtained and stored in the microcontroller memory
11:      if Current position of tuning magnet matches optimum position (the accuracy is $1/2^8$) then
12:         Goto 2
13:      else
14:         Perform rough tuning (Algorithm 3)
15:      end if
16:      Measure the phase difference between the accelerometer signal and the microgenerator signal
17:      if The phase difference is less than 100$\mu$s then
18:         Goto 2
19:      else
20:         Perform fine tuning (Algorithm 4)
21:      end if
22:   end if
23: until Forever
```

retrieves the new desired position of the tuning magnet from a look-up table and begins a tuning process by controlling the actuator to move the tuning magnet to the new position (Fig. 11(a)). The watchdog timer and the microcontroller's clock frequency have been chosen as parameters for optimisation. Because these two parameters determine how much energy the microcontroller consumes and how quickly the system can response to the input vibration frequency change.

Algorithm 2 contains two subroutines: rough tuning (Algorithm 3) and fine tuning (Algorithm 4). The rough tuning measures the frequency of the microgenerator output and moves the actuator to the optimum position according to a predefined lookup table. However, the rough tuning alone cannot generate the best performance and a fine tuning algorithm is needed. This is because the measurement of the frequency of the microgenerator signal does not represent the input vibration frequency accurately enough and, in addition, there may also be a phase difference between the input vibration and the microgenerator motion that prevents the microgenerator from working at the resonance. The fine tuning takes another input, the raw vibration data from the accelerometer and moves the actuator to minimize the phase

**Algorithm 3** Rough tuning algorithm

1: **repeat**
2: Send the optimum position as 8-bit control signal to the actuator
3: The actuator moves tuning magnet
4: Wait 5 seconds for the microgenerator signal to settle down
5: Compare the current position and optimum position
6: **until** Current position of tuning magnet matches optimum position

**Algorithm 4** Fine tuning algorithm

1: **repeat**
2: Send the direction of movement that can reduce phase difference to the actuator
3: The actuator moves tuning magnet by 1 step
4: Wait 5 seconds for the microgenerator signal to settle down
5: Measure the phase of the accelerometer signal
6: Measure the phase of the microgenerator signal
7: Calculate the phase difference
8: **until** The phase difference is less than 100$\mu$s

difference between the microgenerator signal and the accelerometer signal so that the microgenerator is working as resonance. It can be seen that the fine tuning algorithm requires more calculation (thus more energy) than the rough tuning and additional energy is consumed by the accelerometer (see Table XI). Therefore it is not so energy efficient to use only the fine tuning algorithm as the proposed two-subroutine method. In the two-subroutine method, the rough tuning moves the actuator to the approximate resonant position and the fine tuning finds the exact resonance.

To tune the resonant frequency of the microgenerator effectively, the system incorporates a microcontroller, a linear actuator and an accelerometer. These three components need to be powered by the energy harvester in order to make an autonomous system. To characterise the power consumption models of these components, current measurements have been taken and power/energy consumptions have been calculated (Table XI). According to the current and voltage values together with their operational times, the equivalent resistances for the power consumption models of these devices have been obtained.

## IV. HDL Implementation

### A. Analogue Part

The SystemC-A language [19] is used to build the system models. It is an extension to the SystemC language with analogue and mixed-signal (AMS) capabilities. The digital part is modeled using standard SystemC modules. The analogue part, consisting of non-linear differential and algebraic equations, is handled using the extended syntax where the user defines the behaviour of each analogue component by specifying the *build* methods that contribute to the analogue equation set of whole system. In Systemc-A, the

TABLE XI
Power Consumption Models of the System Components

| Component (action) | Operation time(ms) | Current (mA) | Power (mW) | $R_{eq}$ ($\Omega$) |
|---|---|---|---|---|
| Accelerometer | 153 | 5.1 | 13.2 | 509 |
| Actuator | | | | |
| (1 step) | 5 | 312 | 811 | 8.33 |
| (100 steps) | 500 | 156 | 405 | 16.7 |
| Microcontroller | | | | |
| (Rough tuning) | 149 | 1.9 | 5.0 | 1.38k |
| (Fine tuning) | 325 | 5.1 | 6.5 | 250 |

*build* method is provided to support the automatic equation formulation of the user-defined system models. It is a virtual method in the abstract *component* base class and inherited by all derived components. It consists of two functions, *BuildM()* and *BuildRhs()*. SystemC-A uses the *BuildM()* method to add the Jacobian entries to the analogue equation set and *BuildRhs()* method to build the equations, i.e. the right hand side of the Newton-Raphson linearized equation set. The microgenerator equations and corresponding Jacobian matrix entries to be included in the SystemC-A model are listed in Table XII.

The SystemC-A code of the tunable microgenerator model, which is according to Table XII, is listed below:

```
generator::generator(){} //constructor
generator::generator(char nameC[5],TerminalVariable
*node_a,TerminalVariable *node_b,double value,double
Freq): //node_a is Vm, node_b is Im, value is the
tuning force, Freq is the input frequency
component(nameC,node_a,node_b,value){
ztQ = new Quantity("ztQ");
//quantity zt is relative displacement
ytQ = new Quantity("ytQ");
//quantity yt is velocity
itQ = new Quantity("itQ");
//quantity it is inductor current
Fin=value; //tuning force
omega=Freq*2*3.14159;}

void generator::build(){ //model equations
t=TS->get_time(); //current time point
S=TS->get_S();
//time derivative, S=2/h for trapezoidal integration
mpytdotdot=-Mp*Yam*omega*omega*sin(omega*t);
//input acceleration force

zt=X(ztQ);
yt=X(ytQ); //X() return previous value
it=X(itQ);

ztdot=Xdot(ztQ); //Xdot() return previous time
derivative
ytdot=Xdot(ytQ);
itdot=Xdot(itQ);

BuildM(ztQ,ztQ,-Ks); //Jacobian of equation (2)
BuildM(ztQ,ytQ,-Cp-Mp*S);
BuildM(ztQ,itQ,-Phi);
BuildRhs(ztQ,mpytdotdot+Mp*ytdot+Cp*yt+Ks*zt+Phi*it)
;
//Right hand side of equation (2)

BuildM(ytQ,ztQ,S);
BuildM(ytQ,ytQ,-1);
BuildM(ytQ,itQ,0);
BuildRhs(ytQ,yt-ztdot);

BuildM(itQ,ztQ,0); //Jacobian of equation (8)
BuildM(itQ,ytQ,-Phi);
BuildM(itQ,itQ,Rc);
BuildRhs(itQ,-Rc*it-Lc*itdot-vt+Phi*yt);
//Right hand side of equation (8)
}
```

TABLE XII
EQUATION FORMULATION OF THE MICROGENERATOR MODEL

| | $z(t)$ | $\frac{dz(t)}{dt}$ | $i_L(t)$ | Equation |
|---|---|---|---|---|
| $z(t)$ | $-k_s$ | $-c_p - mS$ | $-\Phi$ | $m\frac{d^2z(t)}{dt^2} + c_p\frac{dz(t)}{dt} + k_s z(t) + \Phi i_L(t) + F_{t\_z} - F_a$ |
| $\frac{dz(t)}{dt}$ | $S$ | -1 | 0 | 0 |
| $i_L(t)$ | 0 | $-\Phi$ | $R_c$ | $-R_c i_L(t) - L_c\frac{di_L(t)}{dt} + \Phi\frac{dz(t)}{dt}$ |

### B. Digital Part

The pseudo code of the tuning algorithm is shown in Algorithm 2. Standard SystemC modules were used to model the digital control process and in the experimental verification the control algorithm was implemented in a PIC16F884 microcontroller. As can be seen in Algorithm 2, a watchdog timer wakes the microcontroller periodically and the microcontroller first detects if there is enough energy stored in the supercapacitor. If there is not enough energy, the microcontroller goes back to sleep and waits for the watchdog timer again. If there is enough energy, the microcontroller will then compare the frequency of the microgenerator signal, which is close to the input vibration frequency, to the microgenerator's resonant frequency. When a difference is detected between the vibration frequency and the resonant frequency, the microcontroller retrieves the new desired position of the tuning magnet from a look-up table and begins a tuning process by controlling the actuator to move the tuning magnet to the new position (Fig. 11(a)).

Algorithm 2 contains two subroutines: rough tuning (Algorithm 3) and fine tuning (Algorithm 4). The rough tuning measures the frequency of the microgenerator output and moves the actuator to the optimum position according to a predefined lookup table. However, the rough tuning alone cannot generate the best performance and a fine tuning algorithm is needed. This is because the measurement of the frequency of the microgenerator signal does not represent the input vibration frequency accurately enough and, in addition, there may also be a phase difference between the input vibration and the microgenerator motion that prevents the microgenerator from working at the resonance. The fine tuning takes another input, the raw vibration data from the accelerometer and moves the actuator to minimize the phase difference between the microgenerator signal and the accelerometer signal so that the microgenerator is working as resonance. It can be seen that the fine tuning algorithm requires more calculation (thus more energy) than the rough tuning and additional energy is consumed by the accelerometer (see Table XI). Therefore it is not so energy efficient to use only the fine tuning algorithm as the proposed two-subroutine method. In the two-subroutine method, the rough tuning moves the actuator to the approximate resonant position and the fine tuning finds the exact resonance.

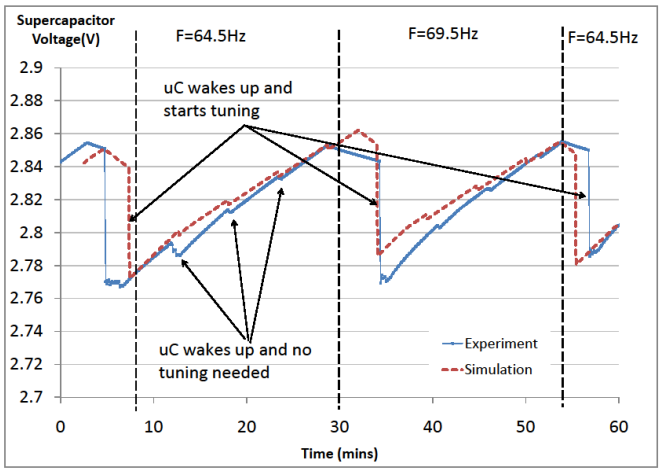
## V. SIMULATION RESULTS AND EXPERIMENTAL VERIFICATION

A SystemC-A model of the complete system has been built and simulated. The SystemC-A code of the top-level testbench is listed below. The system components include the microgenerator, the diode bridge, the supercapacitor and the equivalent variable resistances of the actuator, the accelerometer, the microcontroller and the sensor node.

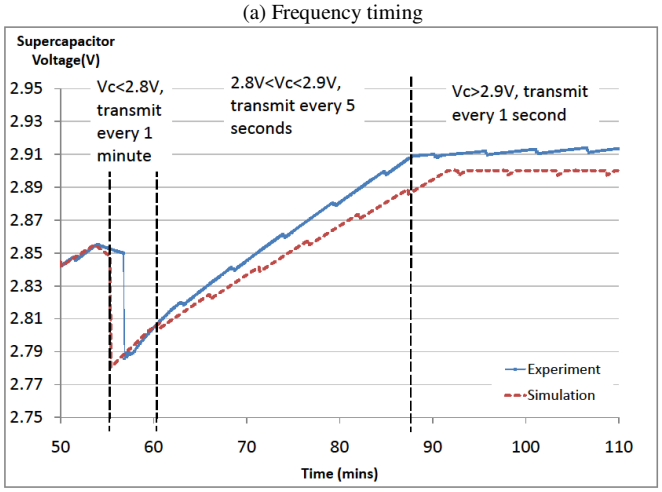
```
void testbench::system(){

ACT=new actuator;
ACM=new accelerometer;
uC=new control;
NODE=new sensor;
n0 = new Node("0");//don't write n0
n1 = new Node("n1");
n2 = new Node("n2");
n3 = new Node("n3");
n4 = new Node("n4");
n5 = new Node("n5");
n6 = new Node("n6");
//microgenerator generator *G1 =new
generator("G1",n1,n2,0.3192,64);
//diode bridge
diode *D1 =new diode("D1",n0,n1,2.117e-7,1.015);
diode *D2 =new diode("D2",n0,n2,2.117e-7,1.015);
diode *D3 =new diode("D3",n2,n3,2.117e-7,1.015);
diode *D4 =new diode("D4",n1,n3,2.117e-7,1.015);
resistor *R1 =new resistor("R1",n1,n0,10e6);
resistor *R2 =new resistor("R2",n2,n0,10e6);
//super capacitor model
resistor *Ri =new resistor("Ri",n3,n4,0.204);
resistor *Rd =new resistor("Rd",n3,n5,84.0);
resistor *Rl =new resistor("Rl",n3,n6,4375.0);
cap_ini *Ci0 =new cap_ini("Ci0",n4,n0,0.35,1.65);
cap_vary *Ci1 =new cap_vary("Ci1",n4,n0,0.21,1.65);
cap_ini *Cd =new cap_ini("Cd",n5,n0,0.21,1.65);
cap_ini *Cl =new cap_ini("Cl",n6,n0,0.06,1.65);
//power consumption models for actuator,
accelerometer, microcontroller and sensor node
res_vary *RAct =new res_vary("RAct",n3,n0,1.0e9);
res_vary *RAcc =new res_vary("RAcc",n3,n0,1.0e9);
res_vary *RuC =new res_vary("RuC",n3,n0,1.0e9);
res_vary *RNode =new res_vary("RNode",n3,n0,1.0e9);
}
```

The test scenario has been divided into two parts. During the first half of the test, the input vibration frequency changes by 5Hz every 25 minutes (Fig. 13(a)). The main objective of this part of the test is to demonstrate the frequency tuning capability of the microgenerator. It can be seen that after the input frequency changes, the supercapacitor voltage drops because the generated voltage is not high enough to charge the supercapacitor. Then the microcontroller wakes up and tunes the resonant frequency of the microgenerator, which uses much of the energy stored on supercapacitor but the retuned microgenerator starts to charge the supercapacitor again. During the second half, the input frequency is fixed and the performance of the sensor node is being tested (Fig. 13(b)). The sensor node transmits at different time intervals according to the different voltage levels on the supercapacitor (Table IX). The transmission interval is reflected on the supercapacitor charging slope. The shorter transmission interval is, the more gradual charging slope gets. Experimental measurements have been carried out and the waveforms are also shown in Fig. 13. The comparison between the simulation and experimental waveforms of the supercapacitor voltage represents both the energy generation and consumption of the system. In both figures the simulation results correlate well with the experimental measurements which validate the presented technique.




(a) Frequency timing




(b) Node beahvior

Fig. 13. Simulations and experimental measurements of the supercapacitor voltages.

## VI. Fast Design Exploration Using a Response Surface Model

Response surface models are constructed from a data set extracted from either physical experiments or computer experiments (simulations) [20]. Due to space limitations, only two major steps of the methodology are given below, namely the formation of an approximated mathematical model by fitting the response under study in terms of design parameters using regression analysis (Section VI-A) and the design of a series of experiments or simulations based on design of experiments (DOE) methodology (Section VI-B). Discussions of the statistical assessment of the goodness of fit and the fitted model reliability are omitted in this paper.

### A. Response Surface Mathematical Model

Suppose there is a dependant variable(s) ($y \epsilon R^n$) where $n$ is the number of observations, believed to be affected by a vector of independent variables ($a \epsilon R^k$) where $k$ is the number of independent variables, then the relationship between the dependent variable(s) and independent variables can be expressed as:

$$y = f(a_1, a_2, ..., a_k) + \varepsilon \tag{11}$$

where $\varepsilon$ represents the model errors, $a_1$, $a_2$, ... , $a_k$ are independent variables and $f()$ is called system function that relates dependant variable to independent variables. In most cases, the exact behavior of the system function is unknown especially in engineering problems, so the system function $f()$ may be approximated by an empirical model as:

$$y = \hat{y}(a_1, a_2, ..., a_k) + \varepsilon \tag{12}$$

where $\hat{y}$ are a low order polynomials or a multi-dimensional splines, and this is called the response surface model. The independent variables or design parameters in equation (12) (i.e $a_1$, $a_2$, ... , $a_k$) are expressed in their corresponding physical units and must be converted to a dimensionless quantities with zero mean and the same standard deviation before proceeding with further RSM analysis such as regression. These new quantities are called coded variables (i.e $x_1$, $x_2$, ... , $x_k$) of original design variables (parameters). The transformation process between natural representations and coded representations is achieved via equation (13):

$$x = \frac{a - \left[a_{\max} + a_{\min} / 2\right]}{\left[a_{\max} + a_{\min} / 2\right]} \tag{13}$$

where $a_{max}$ and $a_{min}$ are the maximum and minimum value in the range of that specific design parameter. Now the approximated function $\hat{y}$ is expressed in term of coded variables ($x_1$, $x_2$, ... , $x_k$) and how to choose such a model $\hat{y}$ determines the success of applying RSM methodology. Typically, most engineering problems $\hat{y}$ can be approximated by a quadratic multi-variable polynomials as follows:

$$\hat{y} = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \beta_{ii} x_i^2 + \sum \sum_{i<j} \beta_{ij} x_i x_j \tag{14}$$

where $\beta_0$, $\beta_i$, $\beta_{ii}$, $\beta_{ij}$ are the coefficients of the intercept, linear, quadratic and interaction in the regression model respectively, $x_i$, $x_j$ are the design parameters in their coded format. The coefficients of the polynomial in equation (14) are determined through $n$ simulation runs for the SystemC-A energy harvester model. The design points of the $n$ runs are determined using DOE technique based on D-Optimal criteria. Using matrix notation, equation (14) can be written as:

$$\hat{\boldsymbol{y}} = \boldsymbol{X}\boldsymbol{\beta} \tag{15}$$

where $\boldsymbol{X}_{n \times p}$ is $n \times p$ design matrix, $p$ is the number of coefficients in the approximated polynomial, $n$ is the number of simulation runs. $\beta_{p \times 1}$ are the unknowns parameters need to be solved. The difference between the observed values $\boldsymbol{y}$ and fitted values $\hat{\boldsymbol{y}}$ for the $i$th observation $\varepsilon_i = y_i - \hat{y}_i$ is called the residual for that specific observation. The sum of the squares of the residuals (SSE) is defined as:

$$SSE = \sum_{i=1}^{n} \varepsilon^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 . \tag{16}$$

Combining equations (15) and (16) and differentiating with respect to $\beta$ lead to:

$$\frac{\partial SSE}{\partial \beta} = \sum_{i=1}^{n} \left( \frac{\partial}{\partial \beta_i} (y_i - \boldsymbol{X}\boldsymbol{\beta})^2 \right). \tag{17}$$

Solving equation (17) for each $\beta_i$ using least square method

TABLE XIII
SYSTEM PARAMETERS FOR OPTIMISATION

| Description | Value range | Coded symbol |
|---|---|---|
| Microcontroller clock frequency(Hz) | 125k - 8M | $x_1$ |
| Watchdog timer wakeup time(sec) | 160 - 480 | $x_2$ |
| Transmission time interval(sec) | 1 - 10 | $x_3$ |

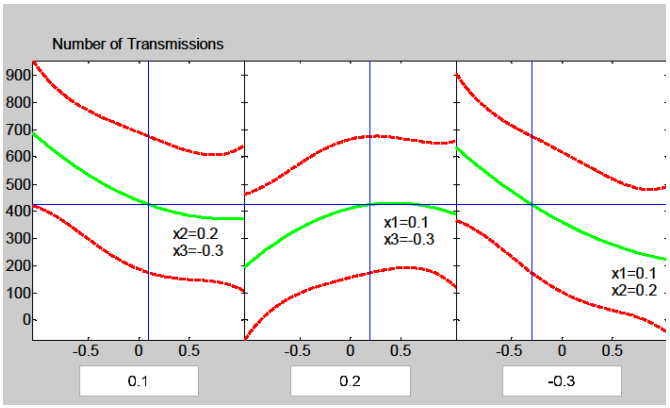



Fig. 14. The effect of each design parameter on system performance (total number of transmissions during one hour).

(LSM) will lead to an accurate model $\hat{y}$ that satisfy the condition of minimum residuals (i.e best fit).

### B. D-Optimal Experimental Design

In the design matrix $\boldsymbol{X}_{n\times p}$, each specific run is represented by a single row and each column contains a specific design parameter that varies in each row based on predefined designed points. How to choose the predefined design points efficiently is called design of experiments (DOE) methodology. There are different types of design of experiments, such as full factorial, central composite design (CCD), Box Behnken designs (BBD) and computer generated designs, such as D-optimal design [20]. Because D-optimal DOE explores design parameters space efficiently with minimum number of run that enable model construction with good accuracy [21], it has be used for the study in this paper. The algorithm of D-optimal criterion optimise the feasible potential design points to form a subset of D-optimal points that will be used in simulation runs. This optimisation is based on maximizing the determinant of ***XX'***, where ***XX'*** is called information matrix [21].

### C. RSM Optimisation Results

As described in Section III-A, three parameters which affect the energy generation and consumption of the wireless sensor node system have been chosen for optimisation. Their value ranges and coded variable symbols are listed in Table XIII. Each of the three coded variables has three values [-1 0 1] which is the minimum number required to generate a quadratic approximation [20]. The full factorial design requires 27 ($3^3$) simulations while the D-optimal design only requires 10 simulations. As explained in Section VI-B, the D-optimal design points are obtained and 10 simulations have been carried out with the corresponding parameters. The acceleration level of the input vibration is fixed as 60 mg and the input frequency changes by 5 Hz every 25 minutes. The optimisation aim has been chosen as to maximise the number of transmissions during one hour. The MATLAB response surface toolbox has been used to generate the quadratic equation and the response surface model is:

$$\hat{y}(x_1,x_2,x_3) = 469.167 - 108.833x_1 - 18.833x_2 - 209.5x_3 + 71.833x_1^2 + 90.5x_2^2 - 39.0x_3^2 - 32.333x_1x_2 - 71.333x_1x_3 + 43.333x_2x_3 \quad (18)$$

The fitted model in equation (18) reflects the effects of each design parameters as well as the interactions effects between design parameters. Fig. 14 plots each single design parameter against the total number of transmissions while holding other design parameters constant.

Two algorithms from the MATLAB optimisation toolbox have been used to maximise the number of transmission, i.e maximise equation (18). The chosen algorithms are Simulated Annealing and Genetic Algorithm, both of which are capable of global searching. The optimisation results, together with the original design, are listed in Table XIV. It can be seen that both of the optimised design improved the system performance massively. The total number of transmissions doubled with the optimised design, which validates our proposed technique.

## VII. CONCLUSION

Wireless sensor networks are fast developing and energy harvester powered sensor nodes have attracted great research interest. In order to design energy efficient wireless sensor nodes, it is crucial to consider all the components in the context of energy consumption in a complete, autonomous wireless system. This paper presents such an HDL based modeling approach that links the system's energy generation and consumption with its analogue parts as well as digital processes. Simulation and optimisation results of the developed HDL models match well with the experimental measurements and correctly reflect the changing energy flow when the digital processes are carrying out different operations. Future work will focus on the optimisation of both the energy harvester and digital control algorithms so that the system's overall energy efficiency can be improved. This paper also presents an approach to fast design space exploration based on a response surface model. The RSM has been used to optimise a complete wireless sensor node syste using SystemC-A and MATLAB. SystemC-A has been used to

TABLE XIV
OPTIMISATION RESULTS

| | Original design | Simulated Annealing | Genetic Algorithm |
|---|---|---|---|
| Microcontroller clock frequency(Hz) | 4M | 125k | 125k |
| Watchdog timer wakeup time(sec) | 320 | 160 | 480 |
| Transmission time interval(sec) | 5 | 1 | 1 |
| Number of transmissions | 405 | 869 | 809 |

model the system's analogue components as well as the digital processes and MATLAB to generate and optimise the response surface model. As demonstrated by the optimisation results, the proposed technique leads to an efficient optimisation process by combining the power of SystemC-A in modelling multi-domain systems and the power of MATLAB in computation.

## REFERENCES


[1] C. Alippi, R. Camplani, C. Galperti, and M. Roveri, "A robust, adaptive, solar-powered wsn framework for aquatic environmental monitoring," Sensors Journal, IEEE, vol. 11, no. 1, pp. 45–55, 2011.

[2] Q. Ling, Z. Tian, Y. Yin, and Y. Li, "Localized structural health monitoring using energy-efficient wireless sensor networks," Sensors Journal, IEEE, vol. 9, no. 11, pp. 1596–1604, 2009.

[3] A. Sapio and G. Tsouri, "Low-power body sensor network for wireless ecg based on relaying of creeping waves at 2.4ghz," in Body Sensor Networks (BSN), 2010 International Conference on, 2010, pp. 167–173.

[4] S. Roundy, P. K. Wright, and J. M. Rabaey, Energy scavenging for wireless sensor networks: with special focus on vibrations. Springer, 2004.

[5] M. P. Buric, G. Kusic, W. Clark, and T. Johnson, "Piezo-electric energy harvesting for wireless sensor networks," in Wireless and Microwave Technology Conference, 2006. WAMICON '06. IEEE Annual, 2006, pp. 1–5.

[6] S. Ergen, A. Sangiovanni-Vincentelli, X. Sun, R. Tebano, S. Alalusi, G. Audisio, and M. Sabatini, "The tire as an intelligent sensor," Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, vol. 28, no. 7, pp. 941–955, 2009.

[7] P. Mitcheson, T. Green, E. Yeatman, and A. Holmes, "Architectures for vibration-driven micropower generators," Journal of Microelectromechanical Systems, vol. 13, no. 3, pp. 429–440, 2004.

[8] D. Zhu, J. Tudor, and S. Beeby, "Strategies for increasing the operating frequency range of vibration energy harvesters: a review," Measurement Science and Technology, vol. 21, no. 2, 2010.

[9] Energy Harvesting Systems: A Block Diagram (2010, July 16), "Holistic energy harvesting," September, 2011. [Online]. Available: http://www.holistic.ecs.soton.ac.uk/res/ehsystem.php

[10] H. Boussetta, M. Marzencki, S. Basrour, and A. Soudani, "Efficient physical modeling of mems energy harvesting devices with vhdl-ams," Sensors Journal, IEEE, vol. 10, no. 9, pp. 1427–1437, 2010.

[11] L.Wang, T. Kazmierski, B. Al-Hashimi, A.Weddell, G. Merrett, and I. Ayala-Garcia, "Accelerated simulation of tunable vibration energy harvesting systems using a linearised state-space technique," in Design, Test and Automation in Europe (DATE 2011), March 14-18, 2011, pp. 1267–1272.

[12] R. Torah, P. Glynne-Jones, J. Tudor, T. O'Donnell, S. Roy, and S. Beeby, "Self-powered autonomous wireless sensor node using vibration energy harvesting," Measurement Science and Technology, vol. 19, no. 12, pp. ISSN 1361–6501, 2008.

[13] M. G. Corporation, SystemVision User's Manual, ser. Version 3.2, Release 2004.3, July 2004.

[14] R. Torah, M. Tudor, K. Patel, I. Garcia, and S. Beeby, "Autonomous low power microsystem powered by vibration energy harvesting," Sensors, IEEE, pp. 264–267, 28-31 Oct. 2007.

[15] Recoil Ltd, UK, http://www.recoilltd.com/index.htm, Sept. 2008.

[16] M. Mitchell, An Introduction to Genetic Algorithms. Cambridge, Massachusetts: the MIT Press, 1996.

[17] I. A. Garcia, D. Zhu, J. Tudor, and S. Beeby, "Autonomous tunable energy harvester," in PowerMEMS 2009, 1-4 December 2009, pp. 49–52.

[18] D. Zhu, S. Roberts, J. Tudor, and S. Beeby, "Design and experimental characterization of a tunable vibration-based electromagnetic micro-generator," Sensors and Actuators A: Physical, vol. 158, no. 2, pp. 284–293, 2010.

[19] H. Al-Junaid and T. Kazmierski, "Analogue and mixed-signal extension to SystemC," IEE proc. Circuit Devices Systems, vol. 152, no. 6, pp. 682–690, Dec. 2005.

[20] J. Jacquez, "Design of experiments," Journal of the Franklin Institute, vol. 335, no. 2, pp. 259–279, 1998.

[21] R. Unal, R. Lepsch, and M. McMillin, "Response surface model building and multidisciplinary optimisation using d-optimal designs," in Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on multidisciplinary Analysis and optimisation, 1998, pp. 405–411.