Response-surface-based design space exploration and optimisation of wireless sensor nodes with tunable energy harvesters

Leran Wang, Tom J. Kazmierski
Bashir M. Al-Hashimi, Mansour Aloufi
University of Southampton, UK
{lw04r,tjk,bmah,ma08r}@ecs.soton.ac.uk

Joseph Wenninger
Vienna University of Technology, Austria
wenninger@ict.tuwien.ac.at

Abstract—In an energy harvester powered wireless sensor node, the energy harvester is often the only energy source, therefore it is crucial to configure the microcontroller and the sensor node so that the harvested energy is used efficiently. This paper presents a response surface model (RSM) based design space exploration and optimisation of a complete wireless sensor node system. In our work the power consumption models of the microcontroller and the sensor node are defined based on their digital operations so that the parameters of the digital algorithms can be optimised to achieve the best energy efficiency. In the proposed technique, SystemC-A is used to model the system's analogue components as well as the digital control algorithms implemented in the microcontroller and the sensor node. A series of simulations are carried out and a response surface model is constructed from the simulation results. The RSM is then optimised using MATLAB's optimisation toolbox and the results show that the optimised system configuration can double the total number of wireless transmissions with fixed amount of harvested energy. The great improvement in the system performance validates the efficiency of our technique.

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, it has become widely agreed that energy harvesters are needed for long-lasting sensor nodes [4, 5]. 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 [6]. 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. 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 [7]. A wireless sensor node powered by tunable energy harvester typically has the following key components (Fig. 1): 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 measurements from an accelerometer, and the wireless transceiver or transmitter.
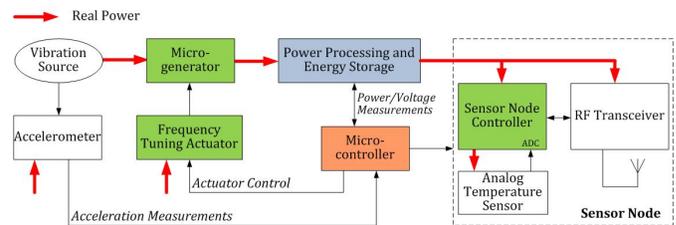

Fig. 1. Block diagram of a tunable energy harvester powered wireless sensor system

Hardware description languages, such as VHDL-AMS and SystemC-A, have been used to model energy harvesters in recent years [8, 9]. HDLs with mixed signal and multi-domain capabilities are suitable for energy harvester modelling because an energy harvester is a mixed-physical-domain system. But none of the existing works includes the sensor node with its power consumption and wireless transmission block into the model. Therefore the inherent link between energy generation and energy consumption of the system is missing in the model and such model reduces the optimization search space. Our focus is to consider both the energy generation and consumption models. The proposed technique models the complete system (Fig. 1) 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. We have identified three system configuration parameters that can influence the energy consumption efficiency. A response surface based design space exploration and optimisation technique has been developed 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. RESPONSE SURFACE MODELLING

Response surface models are constructed from a data set extracted from either physical experiments or computer experiments (simulations) [10]. 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 II-A) and the design of a series of experiments or simulations based on design of experiments (DOE) methodology (Section II-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) ($y \in R^n$) where n is the number of observations, believed to be affected by a vector of independent variables ($a \in R^k$) where k is the number of independent variables, then the relationship between the dependent variable(s) and independent variables can be expressed as:

$$y = f(a_1, a_2, ..., a_k) + \epsilon \qquad (1)$$

where ϵ 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 behaviour of the system function is unknown especially in engineering problems, so the system function $f()$ may be approximated by an empirical model as:

$$y = \hat{y}(a_1, a_2, ..., a_k) + \epsilon \qquad (2)$$

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 (2) (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 (3):

$$x = \frac{a - [a_{max} + a_{min}]/2}{[a_{max} + a_{min}]/2} \qquad (3)$$

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:

$$\hat{y} = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \beta_{ii} x_i^2 + \sum\sum_{i<j} \beta_{ij} x_i x_j \qquad (4)$$

where $\beta_0, \beta_i, \beta_i i, \beta_i j$ 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 (4) 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 (4) can be written as:

$$\hat{\boldsymbol{y}} = \boldsymbol{X}\beta \qquad (5)$$

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 ith observation $\epsilon_i = y_i - \hat{y}_i$ is called the residual for that specific observation. The sum of the squares of the residuals (SSE) is defined as:

$$SSE = \sum_{i=1}^{n} \epsilon^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \qquad (6)$$

Combining equations (5 and 6) and differentiating with respect to β lead to:

$$\frac{\partial SSE}{\partial \beta} = \sum_{i=1}^{n} \left(\frac{\partial}{\partial \beta_i} (y_i - \boldsymbol{X}\beta)^2 \right) \qquad (7)$$

Solving equation (7) for each β_i using least square method (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 [10]. Because D-optimal DOE explores design parameters space efficiently with minimum number of run that enable model construction with good accuracy [11], 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 $\boldsymbol{X}\boldsymbol{X}'$, where $\boldsymbol{X}\boldsymbol{X}'$ is called information matrix [11].

III. SYSTEM DESCRIPTION

Fig. 2 shows the diagram of a wireless sensor node system powered by tunable energy harvester [12]. The wireless sensor node has a temperature sensor and a 2.4GHz radio transceiver. Once activated, the measured data is 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 as an example. 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 detailed tuning algorithms are presented in Section IV-C. The microcontroller also provides energy for the accelerometer and the actuator so that these devices can be turned off when not in use. Table I lists the type and make of the system components.
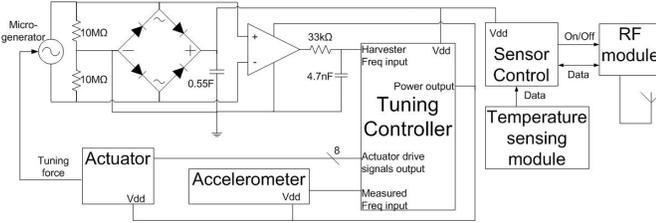

Fig. 2. System diagram of a tunable energy harvester powered wireless sensor system

TABLE I
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

Three parameters have been identified to influence the power consumption of the system: the microcontroller clock frequency, the watchdog timer wakeup time and the wireless node transmission time interval. There are system performance trade-offs between increasing and decreasing each of these parameters therefore optimisation is needed. The optimisation aim is to maximise the number of wireless transmissions with the given amount of harvested energy. Descriptions of each parameter are given below:

1) The microcontroller clock frequency defines how fast the tuning control algorithm runs. High clock frequency means faster command execution but the microcontroller will consume more power. To determine the frequency of input signals, the microcontroller uses a timer/counter to measure the time period of the input signals, therefore the total time needed to finish the counter loop is fixed and higher clock frequency means higher consumed energy. Low clock frequency can save energy but the measurement of the input vibration frequency will be less accurate leading to less effective tuning control algorithm.

2) The watchdog timer wakes up the microcontroller periodically to check if the microgenerator resonant frequency matches the input vibration frequency. To increase the watchdog timer's wakeup time will reduce the power consumption of the microcontroller but longer wakeup time means the response to input frequency change will be slower.

3) The transmission time interval determines the number of transmission in certain time period. The time interval should be chosen so that the sensor node is making as many transmission as possible without depleting the energy storage, i.e consuming more energy than the harvester can generate.

IV. SYSTEM COMPONENT MODELS

A. Tunable microgenerator

The tunable microgenerator is based on a cantilever structure [12]. 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. Detailed description of the microgenerator model can be found in [9]. The control algorithm is modelled as a SystemC digital process as described in Section IV-C.

B. Sensor node behaviour 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. 3) 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 II. 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 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.
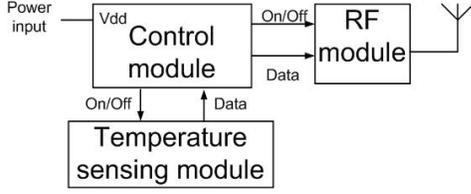

Fig. 3.   Block diagram of the sensor node

TABLE II
SENSOR NODE BEHAVIOUR 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)

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

TABLE III
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

The supply voltage was kept at 2.8V. 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:

$$R_{node} = \begin{cases} 167\,\Omega & \text{when in transmission} \\ 5.8\,\text{M}\Omega & \text{when in sleep} \end{cases} \quad (8)$$

C. Tuning algorithms and power consumption models

In order for a energy harvester powered wireless sensor node (Fig. 2) 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 1. 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 1, 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. 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 1 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 \geq$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 coarse-grain tuning (**Algorithm 2**)
15:         **end if**
16:         Measure the phase different 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-grain tuning (**Algorithm 3**)
21:         **end if**
22:     **end if**
23: **until** Forever

Algorithm 1 contains two subroutines: coarse-grain tuning (Algorithm 2) and fine-grain tuning (Algorithm 3). The coarse-grain tuning measures the frequency of the microgenerator output and moves the actuator to the optimum position according to a predefined lookup table. However, the coarse-grain tuning alone cannot generate the best performance and a fine-grain 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-grain 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-grain tuning algorithm requires more calculation (thus more energy) than the coarse-grain tuning and additional energy is consumed by the accelerometer (see Table IV). Therefore it is not so energy efficient to use only the fine-grain tuning algorithm as the proposed two-subroutine method. In the two-subroutine method, the coarse-grain tuning moves the actuator to the approximate resonant position and the fine-grain tuning finds the exact resonance.

Algorithm 2 Coarse-grain 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 3 Fine-grain 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 $100\mu s$

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 IV). 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.

V. Optimisation results

For a wireless sensor node, it is desirable to make as many transmissions as possible during a fixed time period.

TABLE IV
POWER CONSUMPTION MODELS OF THE SYSTEM COMPONENTS

Component (action)	Operation time(ms)	Current (mA)	Power (mW)	R_{eq} (Ω)	Energy (mJ)
Accelerometer	153	5.1	13.2	509	2.02
Actuator					
(1 step)	5	312	811	8.33	4.06
(100 steps)	500	156	405	16.7	203
Microcontroller					
(Coarse-grain tuning)	149	1.9	5.0	1.38k	0.745
(Fine-grain tuning)	325	5.1	6.5	250	2.11

However, there are other components in the system which also consume the harvested energy as shown in Table IV. Therefore it is important to configure different system parameters in a holistic manner so that the best overall efficiency can be achieved. As described in Section III, three parameters which affect the energy generation and consumption of the wireless sensor node system have been chosen for optimisation. The optimisation aim has been chosen as to maximise the number of transmissions during one hour. The acceleration level of the input vibration is fixed as 60mg and the input frequency changes by 5Hz every 25 minutes (Fig. 5). The value ranges of each parameter and their coded variable symbols are listed in Table V. Each of the three coded variables has three values [-1 0 1] which is the minimum number required to generate a quadratic approximation [10]. The full factorial design requires 27 (3^3) simulations while the D-optimal design only requires 10 simulations. As explained in Section II-B, the D-optimal design points are obtained and 10 simulations have been carried out with the corresponding parameters. The MATLAB response surface toolbox has been used to generate the quadratic equation and the response surface model is:

$$\hat{y}(x_1, x_2, x_3) = 484.02 - 121.79x_1 - 16.77x_2 - 208.43x_3$$
$$+ 120.98x_1^2 + 106.69x_2^2 - 69.75x_3^2$$
$$- 34.23x_1x_2 - 121.79x_1x_3 + 32.54x_2x_3 \qquad (9)$$

TABLE V
SYSTEM PARAMETERS FOR OPTIMISATION

Description	Value range	Coded symbol
Microcontroller clock frequency(Hz)	125k - 8M	x_1
Watchdog timer wakeup time(sec)	60 - 600	x_2
Transmission time interval(sec)	0.005 - 10	x_3

The fitted model in equation (9) reflects the effects of each design parameters as well as the interactions effects between design parameters. Fig. 4 plots each single design parameter against the total number of transmissions while holding the other two parameters constant (green solid lines). Fig. 4 also shows the design space of the system parameters (red dash lines).

Two algorithms from the MATLAB optimisation toolbox have been used to maximise the number of transmission, i.e maximise equation (9). The chosen algorithms are Simulated
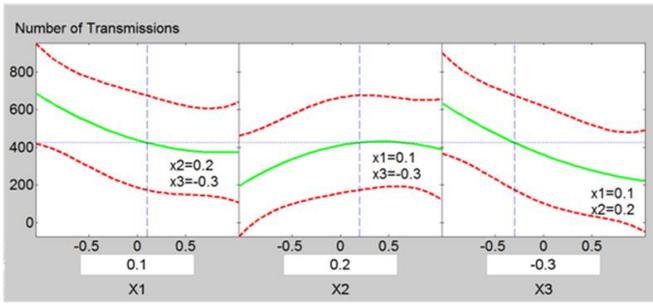
Fig. 4. Design space exploration

Annealing and Genetic Algorithm, both of which are capable of global searching. The optimisation results, together with the original design, are listed in Table VI. It can be seen that both of the optimised design greatly improved the system performance. The total number of transmissions doubled with the optimised designs. To demonstrate the effect of RSM optimisation on the real system in physical domain, the SystemC-A model is simulated using the optimised parameters. Fig. 5 shows the simulated supercapacitor voltage of the original and Simulated Annealing optimised designs. We show the waveforms of the supercapacitor voltage because it can represent both the energy generation and consumption of the system.

TABLE VI
OPTIMISATION RESULTS

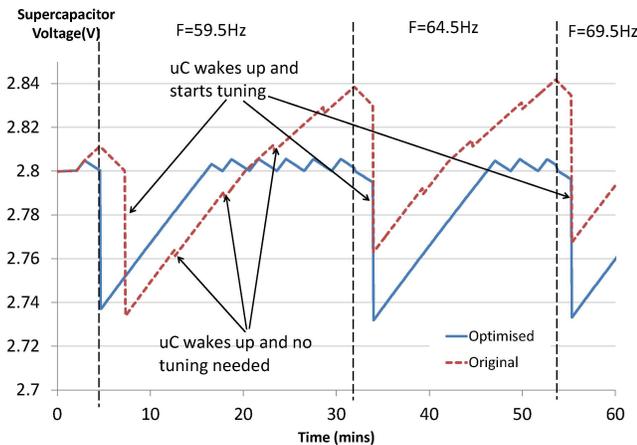|	Original design	Simulated Annealing	Genetic Algorithm
Microcontroller clock frequency(Hz)	4M	8M	125k
Watchdog timer wakeup time(sec)	320	60	600
Transmission time interval(sec)	5	0.005	3.065
No. of transmissions	405	899	894

Fig. 5. Supercapacitor voltage of original and optimised designs

VI. CONCLUSION

This paper highlights the importance of energy consumption in an energy harvester powered wireless sensor node system. The identified design parameters are investigated using response surface model based design space exploration and optimisation. We use SystemC-A to 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. Roundy, P. K. Wright, and J. M. Rabaey, *Energy scavenging for wireless sensor networks: with special focus on vibrations*. Norwell, MA: Kluwer-Academic, 2003.

[7] 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.

[8] 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.

[9] 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.

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

[11] 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.

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