Accurate Modeling and Prediction of Energy Availability in Energy Harvesting Real-Time Embedded Systems

Jun Lu, Shaobo Liu, Qing Wu and Qinru Qiu
Department of Electrical and Computer Engineering
Binghamton University, State University of New York
Binghamton, New York, USA
{jlu5, sliu5, qwu, qqiu}@binghamton.edu

Abstract — **Energy availability is the primary subject that drives the research innovations in energy harvesting systems. In this paper, we first propose a novel concept of effective energy dissipation that defines a unique quantity to accurately quantify the energy dissipation of the system by including not only the energy demand by the electronic circuit, but also the energy overhead incurred by energy flows amongst system components. This work also addresses the techniques in run-time prediction of future harvested energy. These two contributions significantly improve the accuracy of energy availability computation for the proposed Model-Accurate Predictive DVFS algorithm, which aims at achieving best system performance under energy harvesting constraints. Experimental results show the improvements achieved by the MAP-DVFS algorithm in deadline miss rate. In addition, we illustrate the trend of system performance variation under different conditions and system design parameters.**

Keywords - energy harvest; real time embedded system; task scheduling; effective energy dissipation; sequence prediction

I. INTRODUCTION

Due to rapid progress in semiconductor manufacturing technology, the chip density and operating frequencies have been increasing exponentially, making power consumption a major roadblock of VLSI technology. High power consumption exacerbates the reliability problem by raising the die temperature and by increasing current density on the supply rails. It also reduces battery lifetime which is a key concern in portable devices and mobile computing systems.

Real-time applications (such as audio-video processing and transmission) and systems (such as cell phone and sensor networks) impose new objectives and constraints on low power design by introducing real-time tasks with fixed/random execution times and deadlines. The power management problems of low power design for real-time embedded systems are usually solved at the system level by combining task scheduling/allocation algorithms with *dynamic voltage and frequency scaling/selection* (DVFS) [1][2] techniques, in order to minimize the system energy dissipation, as well as the deadline miss rate, which is a very important performance metric for real-time computing.

In today's applications, most real-time embedded systems are mobile and powered by batteries. Therefore, low power research to maximize the battery lifetime before replacing or recharging the battery is a critical undertaking. Additionally, great interest has risen in powering these systems with renewable energy sources. *Renewable energy* is energy generated from natural resources such as sunlight, wind, rain, tides geothermal heat, etc., which are naturally replenished. *Energy harvesting* (or *energy scavenging*) refers to the process of collecting and converting renewable energy so that it can be utilized by electronic systems. Energy harvesting is a promising technology for overcoming the energy limitations of battery-powered systems and has the potential to allow systems to achieve energy autonomy. Several prototypes such as the Heliomote [3] and the Prometheus [4] have been designed to reveal the superiority of energy harvesting system.

Many technical challenges lie ahead in order to make an energy harvesting system work effectively. Among them is to develop novel power management methods and algorithms dedicated to energy harvesting systems, considering the following distinct features, comparing to the traditional battery-power systems:

1) An energy harvesting system is able to recharge its battery by the harvested power from the environmental energy source such as sunlight, wind, etc.
2) The energy source may present some type of periodic property. For instance, the sunlight has the high intensity at daytime and reduces to zero at nighttime.
3) The energy source is unstable and changing from time to time. The harvested power should be modeled as a time-varying variable. Some energy sources display both stochastic and periodic characteristics.
4) The uncertainty of energy availability. In a battery powered system, we are certain how much energy is left in the storage for use. But for an energy harvesting system, we do not know beforehand exactly how much energy can be utilized by the system.

The energy harvesting systems are exposed to new problems that do not exist in the conventional battery-powered systems. The conventional task scheduling and power management techniques are not designed for the energy harvesting systems and cannot handle the uncertainty in available energy. It is important to develop the novel power management techniques so that the energy harvesting systems are able to operate energy-efficiently as well as achieve energy autonomy.

Design considerations for energy harvesting systems are surveyed by the authors of [3]. Several techniques are proposed to maximize the rewards of the energy harvesting system in [5][6][7]. The authors assume that energy is consumed for obtaining certain level of service, measured in reward; and they focus on how to allocate and consume the energy such that the overall reward is maximized. However these techniques do not target at real-time systems.

Many research activities have been carried out on power management of *real-time energy harvesting embedded systems* (EH-RTES). The authors in [8] proposed an offline algorithm using dynamic voltage and frequency selection (DVFS) based on hypothesis that harvested energy is constant from ambient energy source, which is not a practical assumption. The work in [9] considered the harvesting energy source as time-variant solar power model, restricted which to work only in daytime and nighttime mode. A lazy scheduling algorithm (LSA) is proposed in [10][11] that task execution is optimized at full speed based on as late as possible policy, which did not consider DVFS and take the advantage of task slack for energy savings.

The authors of [12] proposed an energy-aware DVFS (EA-DVFS) algorithm. Based on this algorithm, the task is executed at full speed of the processor if sufficient energy is available to the system; otherwise, the task is slowed down and the processor executes it in a lower frequency and lower power state. The shortcoming of the EA-DVFS algorithm is that it considers available energy and scheduling for only one task, instead of all the tasks in the queue. The authors of [13] proposed an improved adaptive scheduling DVFS (AS-DVFS) algorithm that schedules all the tasks in the queue and assigns frequency and voltage levels to them for evenly distributed workload for the processor. It also adaptively adjusts the scheduling based on run-time energy availability.

Energy availability, i.e., the amount of energy that can be used by the electronic system (e.g., the processor) to complete tasks, is the center of the EH-RTES power management research. All existing research works assume that the available energy is exactly the summation of the harvested energy and the remaining energy in the battery. This assumption, however, is not even close to the situation in real applications. In this paper, we first propose a realistic system model for the EH-RTES. It not only captures the characteristics of harvested energy and battery remaining energy, but also provides the accurate modeling of different sources of energy overhead (waste) in the system.

From the proposed system model we will show that, to accurately calculate the energy dissipation of executing a real-time task, we need to firstly take into account the energy demand by the electronic system. More importantly, this energy demand must be converted to the energy taken from the harvesting source and the battery. To make the conversion accurately, we need to incorporate complex energy flow among different system components and the energy overhead in the flow. To achieve this goal, we propose the concept of *effective energy dissipation* (EED). The importance is that it defines a unique quantity to accurately capture the complex system and component characteristics of the EH-RTES. It also significantly simplifies those scheduling and DVFS algorithms that rely on accurate quantitative information of the system.

In this paper we also address another major issue in this area, which is the real-time prediction of future harvested energy. As we have mentioned, the harvested energy is the most important part that determines the energy availability in an EH-RTES. Almost existing research works either assume that the harvested energy is a constant [8], which is the opposite of in most real applications; or it is already known from profiling [10][11][12][13], which is not realistic either. Recently, several prediction methods are proposed to prediction solar energy. In [14], exponentially Weighted Moving-Average (EWMA) model was designed to predict solar energy based on weighted current time-slot energy and historical average. It was improved in [15] by using Weather-Conditioned Moving Average (WCMA), which considered both solar and weather conditions. The shortcoming of these works is that they do not focus on real-time prediction. Accurate prediction of the near-future harvested energy is crucial to effective power management of the EH-RTES. The efficiency of the energy optimization techniques largely depends on the accuracy of the energy harvesting prediction. In this paper we investigate three common techniques in real time series prediction and their impact on the algorithm design and system performance.

By combining the calculation of effective energy dissipation, the run-time prediction of harvested energy, and the best of existing scheduling/DVFS algorithm, we propose the *Model-Accurate Predictive DVFS* (MAP-DVFS) algorithm. “Model-Accurate” refers to that this is the first approach that is accurate when calculating the energy cost of tasks, according to the EH-RTES system and component models. “Predictive” refers to that this is the first approach that performs run-time prediction on future harvested energy. In our experiments, we will illustrate the improvements achieved by the MAP-DVFS algorithm in deadline miss rate of the system. In addition, we will present results that show the trend of system performance variation under different conditions. We will also compare the effectiveness of different prediction techniques.

The remainder of the paper is organized as follows. In Section II, we discuss the system model of the EH-RTES as well as the models for individual components. Section III gives a detailed definition of *effective energy dissipation*. Section IV presents three *real-time sequence prediction* algorithms for predicting future energy availability. The proposed MAP-DVFS algorithm is discussed in Section V. The experimental results and conclusions are presented in Sections VI and VII, respectively.

II. Background

In this paper, we consider a typical EH-RTES that consists of three major modules: *energy harvesting module* (EHM), *energy storage module* (ESM) and *energy dissipation module* (EDM), as shown in Figure 1. Two *energy conversion modules* (ECM) are used to regulate the voltage to the range which could be used by ESM and EDM.

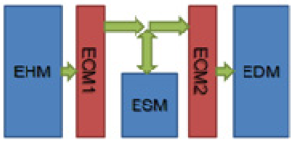


Figure 1. System diagram and energy flow of an EH-RTES.

A. Energy Dissipation Module

Dynamic voltage and frequency scaling/selection (DVFS) is the primary power management technique for real-time embedded systems. Assuming that a DVFS-enabled processor [17] has N discrete operating frequencies f_n: $\{ f_n \mid 1 \leq n \leq N,\ f_{\min} = f_1 < f_2 < \ldots < f_N = f_{\max} \}$; and the supply voltage and power consumption by the processor running at frequency f_n are denoted as V_n and P_n, respectively. We define a slowdown factor S_n as the normalized frequency of f_n with respect to the maximum frequency $f_{\max}$, that is: $S_n = f_n/f_{\max}$. In the remaining parts of the proposal, we use notations f_n & $f(n)$, P_n & $P(n)$, S_n & $S(n)$ interchangeably.

The triplet (a_m, d_m, $wcet_m$) is used for characterizing a real-time task τ_m, where a_m, d_m, $wcet_m$ indicate the *arrival time*, the *relative deadline* and *the worst case execution time* of task τ_m, respectively. On a DVFS-enabled processor, if task τ_m is stretched by a slowdown factor S_n, then its actual execution time at frequency f_n is $wcet_m/S_n$. In real-time applications, a scheduler controls the execution order of the tasks. The system is considered to be preemptive. The task with the earliest deadline has the highest priority and should be executed first; it preempts any other task if needed.

B. Energy Harvesting Module

In this paper, solar panels are chosen to be the primary energy harvesting technology for EH-RTES. Solar energy harvesting is a process with high uncertainty. The energy harvesting rate is heavily dependent on the operating environment and fluctuates in real time. Figure 2 shows daytime solar irradiation profiles that have been collected by the authors from four different days. If we denote $P_H(t)$ as the net power output from the energy harvesting unit. The harvested energy $E_H(t_1, t_2)$ at time interval $[t_1, t_2]$ can be calculated as:

$$E_H(t_1, t_2) = \int_{t1}^{t2} P_H(t) dt \qquad (1)$$

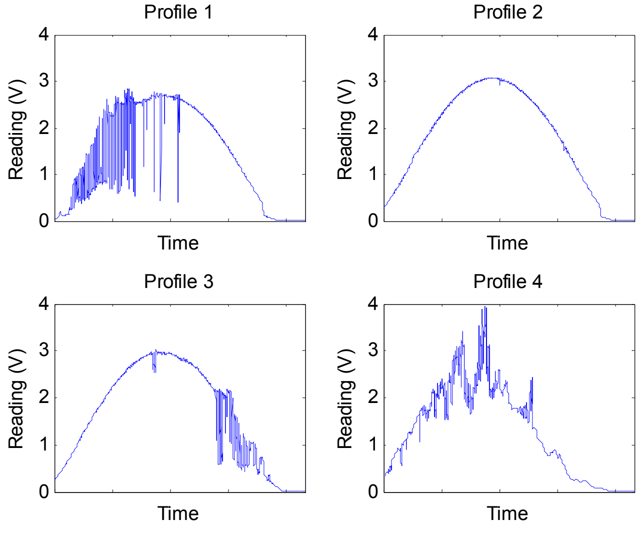


Figure 2. Four different solar irradiation profiles.

C. Energy Storage Module

Energy storage is usually required in EH-RTES not just because we need it to continue operation even when there is no energy to harvest (e.g., at night for a solar-powered system). Rechargeable batteries [18] and ultracapacitors (or supercapacitors) [19] are predominant choices for the ESM.

Let us define η as the battery efficiency factor and the actual energy removed from the battery can be written as:

$$E_{ACT} = \int_{t1}^{t2} \frac{1}{\eta} \cdot V_S \cdot I_S(t) dt \qquad (2)$$

in which V_S is the battery output voltage and $I_S(t)$ is the discharge current. At the same time, the battery efficiency factor is also a function of the current. In a linear approximation, we have:

$$\eta = 1 - \alpha \cdot I \qquad (3)$$

in which α is a positive constant. Then we can re-write the actual energy equation as:

$$E_{ACT} = \int_{t1}^{t2} \frac{V_S \cdot I_S(t)}{1 - \alpha \cdot I_S(t)} dt \qquad (4)$$

The battery charging process can also be modeled with the similar approach.

D. Energy Conversion Moduels

As shown in Figure 1, we consider two electrical energy conversion units in the EH-RTES. The ECM1 converts energy from the output of the EHM so that it can be used by the ESM. Depending on the type of energy harvesting technology, ECM1 can be either DC/DC or AC/DC converter. ECM2 is usually a DC/DC converter that regulates the supply voltage level of the EDM. For DVFS-enabled processors, the output voltage of ECM2 should be controllable.

III. Effective Energy Dissipation

In this section, we propose a new concept of effective energy dissipation that captures not only the interactions among the EHM, ESM and EDM, but also the efficiency factors (energy overhead) of the ESM and ECMs.

Let us first add some key parameters to the system diagram in Figure 1. They are listed as follows.

V_H, I_H	Output voltage and current of the EHM
V_S, I_S	Output voltage and current of the ESM
V_D, I_D	Supply voltage and current of the EDM
V_1, I_1, η_1	Output voltage, current and conversion efficiency of ECM1
V_2, I_2, η_2	Input voltage, current and conversion efficiency of ECM2
C_S, η_S	Total capacity and charge/discharge efficiency of the ESM
E_H, E_S, E_D	Harvested energy from EHM, remaining energy in ESU, and the effective energy dissipation

We introduce a new concept of "*effective energy dissipation,*" denoted as E_D, as a unified energy model that captures not only the actually energy needs by the EDM, but also the energy harvesting speed, and the efficiency factors of the ECMs and ESM.

Consider a short period of time dt, we define E_{DA} as the actually energy needed by the EDM, we have:

$$E_{DA} = V_D I_D dt \qquad (5)$$

The energy needed at the input of ECM2 is:

$$E_{D2} = V_D I_D dt / \eta_2 \qquad (6)$$

To determine the final E_D, we need to look at two different situations. In the first situation, if E_{D2} is smaller than the harvested energy at the output of ECM1:

$$E_{H1} = \eta_1 V_H I_H dt \tag{7}$$

then E_D is energy solely from the EHM. On the other hand, if E_{D2} is larger than E_{H1}, it means that we need to draw extra energy from the ESU:

$$E_{S1} = (E_{D2} - E_{H1})/\eta_S \tag{8}$$

An additional consideration for E_{S1} is that since the dissipated energy from the ESM has to be restored in the future, we should also take into account the charging efficiency. Therefore we have:

$$E_{S1} = (E_{D2} - E_{H1})(1 + 2(1 - \eta_S)/\eta_S) \tag{9}$$

Effectively we are penalizing the usage of energy in the ESM by the "*round-trip*" energy overhead. Finally the effective energy dissipation E_D can be calculated as:

$$E_D = \begin{cases} \dfrac{V_D I_D dt}{\eta_1 \eta_2}, if\ E_{D2} \leq E_{H1} \\ V_H I_H dt + (V_D I_D dt/\eta_2 - \eta_1 V_H I_H dt)\left(1 + \dfrac{2(1-\eta_S)}{\eta_S}\right), otherwise \end{cases} \tag{10}$$

Depending on the accuracy requirements in different applications, the efficiency factors can be either approximated by constants or modeled exactly as functions of input/output voltage/current.

The importance of the effective energy dissipation is that it defines a unique quantity to accurately capture the complex system and component characteristics of the EH-RTES. It also significantly simplifies those scheduling and DVFS algorithms that rely on accurate quantitative information of the system.

IV. Predicting Harvested Energy

Accurate prediction of the near-future harvested energy is crucial to effective power management of the EH-RTES. It has been acknowledged in references [3][14] that the efficiency of the optimization techniques of EH-RTES largely depends on the accuracy of the energy harvesting profiling and prediction. A good prediction model for the EH-RTES must have high accuracy, low computation complexity and low memory requirement.

In this paper, we investigate three different time series prediction techniques that meet the above mentioned conditions.

A. Regression Analysis

Regression analysis [20] is a statistical technique for modeling and investigating the relationship among observed variables. It is used to estimate and predict the value of one variable by taking into account the other related. Forecasting using simple regression is based on equation:

$$x = b_0 + b_1 z + \varepsilon \tag{11}$$

It is proven that by the method of minimizing least squares to estimate b_0 and b_1,

$$\widehat{b_1} = \frac{\sum_{i=1}^{n}(z_\mathrm{i} - \overline{z}) \sum_{i=1}^{n}(x_\mathrm{i} - \overline{x})}{\sum_{i=1}^{n}(z_\mathrm{i} - \overline{z})^2} \tag{12}$$

$$\widehat{b_0} = \overline{x} - \widehat{b_1}\overline{z} \tag{13}$$

where (x_1, z_1), (x_2, z_2), … ,(x_n, z_n) are the n observations available. In this paper, they are the sunlight intensity and time, respectively. And $\overline{x}$ and $\overline{z}$ are the arithmetic mean of conrespondent x and z.

The fitted simple linear regression model is:

$$\hat{x} = \widehat{b_0} + \widehat{b_1} z \tag{14}$$

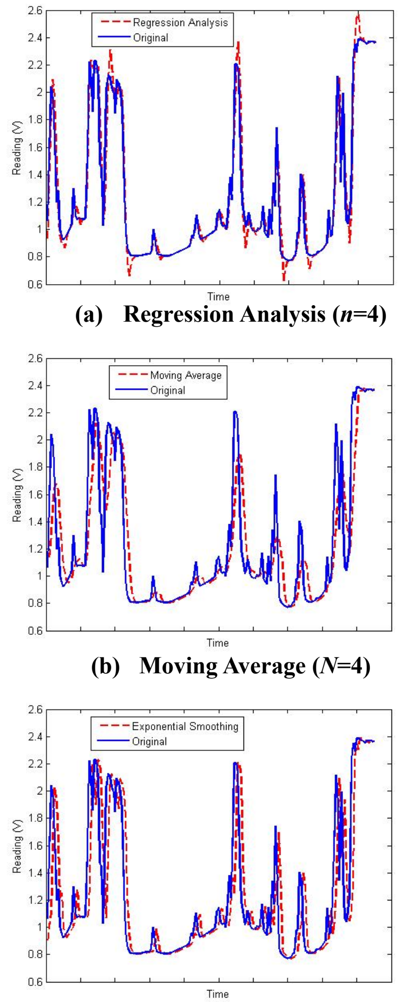
For any value of the predictor variable, Equation (14) will produce the corresponding predicted value of response.

B. Moving Average

The main objective of the *moving average* technique [20] is to predict future values based on averages of the past values. It is useful in reducing the random variations in the observation data. The simple moving average uses the N past observation data to calculate the next prediction time series value, as shown in Equation (15).

$$\hat{x} = \frac{x_{(t)} + x_{(t-1)} + \cdots + x_{(t-N+1)}}{N} \tag{15}$$

The property of simple moving average depends on the number of past observations to be averaged, but it gives equal weight to all past data, of which a large number to be stored and used for forecasts.



(a) Regression Analysis (*n*=4)

(b) Moving Average (*N*=4)

(c) Exponential Smoothing (*α*=0.2)

Figure 3. Predicted solar power sequence versus real data.

C. Exponential Smoothing

The exponential smoothing approach [21] is widely used for short-time forecasting. Although it also employs weighting factors for past values, the weighting factors decay exponentially with the distance of the past values of the time series from the present time. Simple exponential smoothing can be obtained by:

$$\hat{x} = x_{e(t)} = \alpha x_{(t)} + (1-\alpha)x_{e(t-1)} \qquad (16)$$

where $x_{e(t)}$ is called the exponentially smoothed value, $x_{(t)}$ is the observed value at the same point of time. The fraction α is called the smoothing constant, and $x_{e(t-1)}$ is the previous exponentially smoothed value.

Figure 3 shows three sample plots of the run-time predicted solar irradiation versus the actual values. To show the differences visually among the three prediction techniques, we picked a short period of time (a few minutes) when there are frequent and large changes in solar power. From visual inspection we can see that the "Regression Analysis" predictor shows best fit with the real data. The other two predictors have certain delay (1~2 seconds) at the data points with large changes. The quantitative comparison on prediction technique versus system performance will be discussed in Section VI.

V. The MAP-DVFS Alogrithm

In this section we introduce the proposed three-step *Model-Accurate Predictive DVFS* (MAP-DVFS) algorithm which evolves from the AS-DVFS algorithm. The key idea of this approach is that it considers the more accurate system energy availability by using the "effective energy dissipation" defined in Equation (10). This algorithm also includes the run-time prediction of future harvest energy, instead of assuming that it's already known.

A. Create Initial Schedule

All tasks in the ready task queue Q are sorted in the ascending order of the task deadline. In this step, all tasks are assumed to be executed at full speed of the processor. The lazy scheduling policy [10] is used to schedule tasks in Q and tasks are always executed as late as possible.

Assuming that there are M tasks in Q, the initial starting time (ist_m) and initial finishing time (ift_m) of each task τ_m (m=1,2,…, M) are required. They will be determined in a reverse order such that ist_M and ift_M are calculated first, while ist_1 and ift_1 last. Based on the lazy scheduling policy, for the last task τ_M, we have: $ift_M = a_M+d_M$ and $ist_M = a_M+d_M-wcet_M$. For all the remaining tasks, the initial schedule can be easily obtained by the following equations, ift_M = min (a_M+d_M, ist_{m+1}) and ist_M = min(max($a_M+d_M-wcet_M$, a_M), $ist_{m+1}-wcet_m$).

In order to schedule practically, the ist_m cannot be smaller than a_m. Note that $a_m+d_m-wcet_m$ is no less than a_m, otherwise there is no feasible scheduling for task τ_m. Therefore, ist_M = min ($a_M+d_M-wcet_M$, $ist_{m+1}-wcet_m$), which means that task τ_m starts either at time instance $a_M + d_M - wcet_M$, when its deadline minuses its worst case execution time, or at the time instance, $ist_{m+1}-wcet_m$, when the starting executing time of its next task ist_{m+1} minus its worst case execution time $wcet_m$.

B. Workload Balancing and Energy Minimization

The initial schedule puts all tasks at full speed, i.e., with the same slowdown factor index SI_m equal to N. In this step, we try to "stretch" these task executions to minimize their energy dissipation by executing at lower energy and frequency levels of the processor.

This step involves N rounds of DVFS optimization, where N is the number of available operating frequencies to the processor. For a given round, the starting time (st_m) of task τ_m for execution is determined by:

$$st_m = \begin{cases} \max(a_1, current_{time}), m = 1 \\ \max(a_m, ft_{m-1}), m = 2, \dots, M \end{cases} \qquad (17)$$

However, its finishing time (ft_m) is more complicated to obtain. Before calculating ft_m, two questions need to be answered: 1. Can the slowdown factor index SI_m for task τ_m be further reduced? If the following inequality holds, $st_m+wcet_m/S(SI_m-1) < ift_m$, then the deadline can still be met after further stretching task τ_m. 2. Are the slowdown factors for tasks indexed from m+1 to M still valid (no deadline misses) if τ_m gets further stretched? If the answers to these two questions are "yes", then SI_m for task τ_m is decremented by 1, thus the operating frequency of task τ_m is reduced to $f(SI_m-1)$ from $f(SI_m)$. Otherwise, SI_m is kept as it is. Now ft_m can be calculated as: $ft_m = st_m + wcet_m/S(SI_m)$. The workload balancing and DVFS step is shown in Algorithm 1.

Algorithm 1: Workload Balancing and DVFS

```
Require: get the initial schedule for $M$ tasks in $Q$
for $n$ = 1:$N$ do
  for $m$ =1:$M$ do
    if $m$ == 1, then
      $st_m = \max(a_m, current\_time)$
    else
      $st_m = \max(a_m, ft_{m-1})$
    end if
    if $st_m+w_m/S(SI_m-1) < ift_m$&&the slowdown factors for tasks with lower priority is valid, then
      $SI_m = SI_m - 1$
  end if
  $ft_m = st_m + w_m/S(SI_m)$
  end for
end for
```

C. Check Energy Availability and Tune Scheduling

In this step, we need to check at run-time the energy availability and adjust the scheduling. If the available energy is not enough for a task's scheduling/DVFS, we must make adjustments to deal with the energy shortage.

If the scheduling/DVFS from Step 2 is invalidated by energy shortage, the task should not be removed from Q right away. Instead, we are going to first try to delay the task execution. For example, if we assume that task τ_m is the first task whose scheduling/DVFS is invalidated by the energy shortage, it means:

$$E_S(st_m)+E_H(st_m, ft_m) < E_D(st_m, ft_m) \qquad (18)$$

where $E_S(st_m)$ is the remaining energy in the ESU at time st_m; $E_H(\mathrm{st_m}, ft_m)$ is the predicted harvested energy between st_m and ft_m using one of the prediction methods in section IV, and $E_D(st_m, ft_m)$ is the effective energy dissipation of τ_m. It is worthwhile to mention that because E_D has captured all energy overhead in the system, we don't need to worry about adding them into the different equations in the algorithm.

The algorithm first tries to reschedule task τ_m by delaying dl_m until the following equality holds:

$$E_S(st_m) + E_H(st_m, ft_m+dl_m) = E_D(st_m+dl_m, ft_m+dl_m)) \quad (19)$$

If the deadline of task τ_m can still be met, which is: $ft_m+dl_m \le a_m+ d_m$, and the slowdown factors for tasks indexed by m+1, …, M are still valid, then task τ_m is executed at time interval [st_m+dl_m, ft_m+dl_m] at the frequency

$f(SI_m)$. The schedule for tasks with lower priority is updated accordingly, as shown in Lines 6~9 of Algorithm 2.

On the other hand, task τ_m is removed from Q and counted as one missed deadline if deadline of task τ_m is violated by delaying dl_m, as shown in Line 12.

Algorithm 2: Scheduling Adjustment on Energy Availability

```
Require: $M$ tasks in $Q$
$E_D$ : the effective energy dissipation calculated by Equation (10)
$E_H$ : the predicted harvested energy from time span of current task's execution
1.  if $E_C(st_m) + E_S(st_m, ft_m) < E_D(st_m, ft_m)$, then
2.    calculate $dl_m$ from equation (16)
3.    if $ft_m + dl_m \le d_m$ && the slowdown factors for tasks with lower priority is valid, then
4.      $st_m = st_m + dl_m$
5.      $ft_m = ft_m + dl_m$
      //update schedule for succeeding tasks
6.      for $i = m+1:M$ do
7.        $st_i = \max(st_i, ft_{i-1})$
8.        $ft_i = st_i + exe_i$;
9.      end for
10.   else
11.     remove task $\tau_m$ from queue $Q$
12.   end if
13. end if
```

D. Overall MAP-DVFS Algorithm

The complete *MAP-DVFS* algorithm is shown in Algorithm 3. It is basically the sequential execution of Section V.A, V.B and V.C. The initial schedule guarantees that tasks meet their deadline requirements. The workload balancing and DVFS algorithm minimizes energy dissipation by trading the task slack time for energy savings. The adaptive adjustment method in Algorithm 3 makes sure that the system has sufficient energy to execute the task based on the computation of *effective energy dissipation* and the prediction on future harvested energy.

Algorithm 3: Overall MAP-DVFS Algorithm

```
Require: maintain a ready task queue $Q$
set task queue $Q$ empty
while (true) do
if new task coming, then
  push new task in $Q$,
  get initial schedule for tasks in $Q$ using Algorithm 1
  balance workload using Algorithm 2
  adjust scheduling using Algorithm 3
end if
execute first task in the task queue
if the task is finished, then
  remove task from the ready task queue $Q$
end if
end while
```

VI. Experimental Results and Analysis

In this section, the performance of the proposed algorithm and the impact of prediction methods are evaluated based on simulations. We have developed a discrete event-driven simulator in C++ and implemented the proposed MAP-DVFS algorithm, for comparison purposes, the AS-DVFS algorithm is also implemented.

A. Simulation Setup

A DVFS-enabled XScale processor is used as the EDM, Please refer to [17] for the power, voltage and frequency of Intel XScale processor. The overhead between the processor switches to different operating voltage/frequency settings is negligible and ignored in the experiment.

The ESM is assumed to be a rechargeable battery or supercapacitor. Without loss of generality, we assume that the charging/discharging efficiency of the ESM is fixed to be 0.9, the conversion efficiency of ECM1 and ECM2 are 0.9. The capacity of the ESM is set to be 1000J. To speed up the simulation, the ESM has 500J energy to start each simulation.

The EHM in the simulation is set to be a solar panel with area of 10*cm* by 10*cm* and conversion efficiency of 10%. Figure 2 in Section II shows four different daytime (7:00AM~7:00PM) solar irradiation profiles that we have collected during February and March of 2010. The data are collected every 5 seconds and the readings are in "*Volt*". We use linear interpolation to generate more data points to fit the simulation step size. The readings represent the possible power output from the solar panel, i.e., P_H. If we denote the reading as Y, we can convert using the following equation:

$$P_H = Y * \mathrm{U} * \mathrm{A} * \varphi \qquad (20)$$

where "U" is a constant value 250 (with unit of $W/(m^2V)$) determined by the solar sensor [19], "A" is the area of the solar panel (0.01 m^2) and "φ" is the conversion efficiency, which is 10%.

Similar to most of previous research work in this area, we also use synthetic task sets for simulation [11]. Each synthetic task set contains the arbitrary number of periodic tasks. In a specific synthetic task set, the period p_m of a task τ_m is randomly chosen from the set {10s, 20s, 30s, …, 100s}, and the relative deadline d_m is set to its period p_m. The worst case execution time w_m is randomly drawn in the interval [0, p_m]. Note that we assume large (in the time magnitude of seconds) tasks to maintain reasonable CPU time for simulation.

To design different workload categories, we introduce the notation of utilization U, which can be calculated as:

$$U = \sum_m \frac{w_m}{p_m} \qquad (21)$$

where w_m and p_m are the worst case execution time, and the period of task τ_m, respectively. The utilization U characterizes the percentage of the busy time of the processor if applications are executed at the full speed. Thus U cannot be greater than 1. In the simulation, we generate different U values by scaling up/down the worst case execution time of each task in the synthetic task set.

For a given solar profile and utilization setting, the simulation run covers the complete one-day solar profile. This process is repeated for 1000 times, each time with a new random task set. The results presented here are the average of these runs.

B. Impact of Using Effective Energy Dissipation

The first set of experiments focus on illustration the impact of using effective energy dissipation (EED). We compare the deadline miss rate (DMR), between the proposed algorithm without prediction and the AS-DVFS algorithm [13]. To study the impact of EED, we are not predicting harvested energy in this setup. Like all previous work, we assume that P_H can be accurately predicted by profiling. Without the prediction, we use the name of MA-DVFS for the algorithm in this setup. The experimental results are shown in Figure 4.

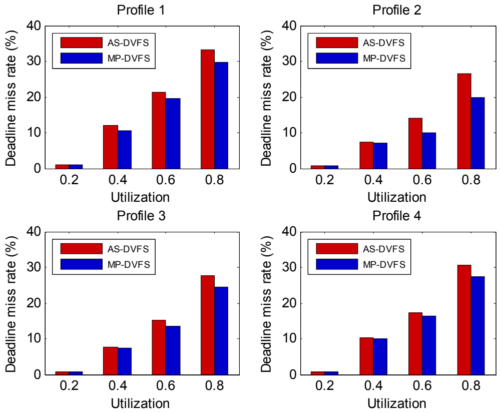


Figure 4. **Comparisons of deadline miss rate.**

We can see that the absolute deadline miss rate reduction increases when the utilization decreases, and at high utilization settings, the MA-DVFS algorithm achieves significant reduction in deadline miss rate over the AS-DVFS algorithm. Without considering the sources of energy overhead in the system, the AS-DVFS algorithm intend to over-estimate the available energy when it comes to the Step 3 of the algorithm. As a result, it sometimes tries to execute a task without enough energy, which leads to wasted energy and a deadline miss. The MA-DVFS algorithm does not make this mistake because it uses the effective energy dissipation for energy availability calculation.

C. *Performance Trend Study*

In real applications, the size of solar panel can be changed due to application and technology, which causes the harvested power from solar panel changes in a wide range. Therefore, in this set of experiments, we test the MA-DVFS algorithm with four different harvesting power settings derived from Profile 3 in Figure 2: P_H, $2P_H$, $3P_H$, and $4P_H$. Also, four different capacity of ESM are tested, from 500J to 2000J with step of 500. The simulation results show that in addition to the utilization ratio U, the harvested power and the ESM capacity (E_{CAP}) also have significant impact on the deadline miss rate.

Figure 5 shows the plots of sweeping both harvest power and E_{CAP} for four different utilization ratios of 0.2, 0.4, 0.6 and 0.8. We have the following observations:

1. With increase of the harvest power and/or E_{CAP}, the deadline miss rate decreases. The most significant decreases happen between P_H and $2P_H$, and between E_{CAP} of 500J and 1000J.
2. Deadline miss rate reduction increases when the processor utilization ratio increases, this trend is also shown in Section VI.B.
3. Deadline miss rate reduction is not linear when harvest power or E_{CAP} increases.

D. *MAP-DVFS with Different Prediction Techniques*

In this third set of experiments, we compare the three prediction methods, discussed in section IV, in the MAP-DVFS algorithm with both effective energy dissipation and prediction. The time step of the prediction is set to be 1 second. Since these three prediction algorithms have low computational complexity, their impact on system performance and energy dissipation can be ignored.

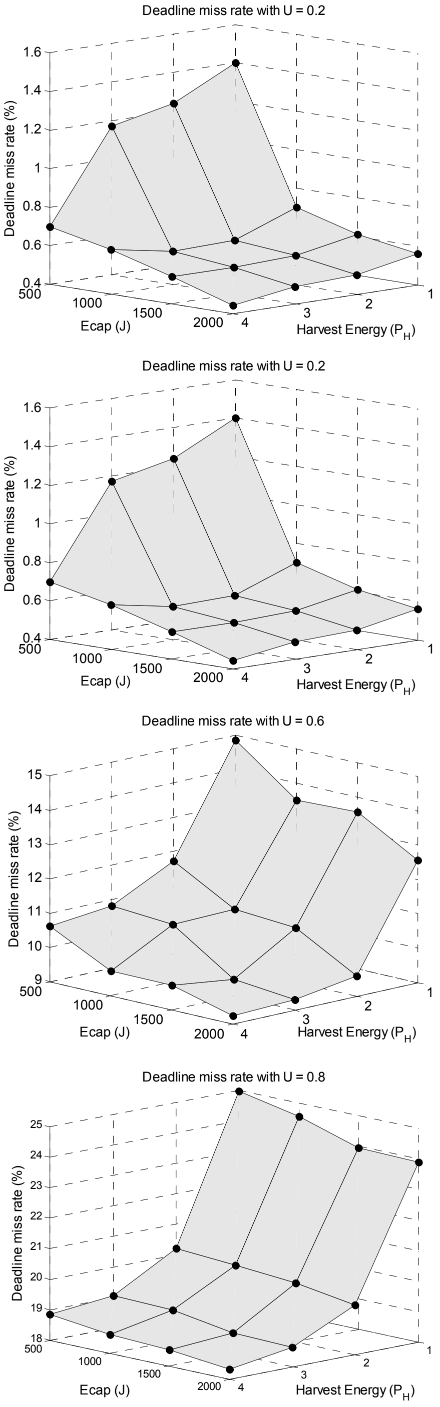


Figure 5. **Sweeping harvest energy and the capacity of ESM.**

In Table 1, the cells in 3rd, 4th and 5th columns include the deadline miss rate results of different prediction methods in MAP-DVFS, regression analysis (RA), moving average (MA) and exponential smoothing (ES), respectively. The last column is the deadline miss rate of the MA-DVFS, which assumes that P_H can be accurately predicted. Although it is not possible for the MA-DVFS algorithm to work well in real applications without prediction, we use the results just for reference purposes.

From Table 1, we can see that the ES has higher deadline miss rate, especially when utilization is higher, comparing to RA and MA. Also, the DMR values of RA and MA show minor differences in all profile and U settings. The results show that RA and MA have better accuracy in predicting solar energy, comparing to the ES technique. According to [21], single ES does not work efficiently when a remarkable trend component is present in the time series pattern.

Table 1. Comparisons of deadline miss rate.

Prof.	***U***	**MAP-DVFS**			**MA-DVFS**
		RA	**MA**	**ES**	
1	0.2	1.93	1.93	2.12	0.97
	0.4	11.44	11.68	14.53	10.71
	0.6	19.67	19.95	25.48	19.62
	0.8	30.46	30.92	38.64	29.80
2	0.2	0.68	0.67	0.60	0.58
	0.4	8.59	8.62	8.21	7.12
	0.6	14.26	14.25	15.74	10.11
	0.8	24.58	24.66	29.33	19.74
3	0.2	0.66	0.67	0.62	0.60
	0.4	8.72	8.59	8.61	7.38
	0.6	14.65	14.71	16.76	13.48
	0.8	25.32	25.56	30.50	24.55
4	0.2	1.66	1.62	1.58	0.67
	0.4	10.69	10.43	11.73	10.08
	0.6	17.43	17.32	19.33	16.33

VII. Summaries

In this paper, we have proposed the concept of effective energy dissipation that defines a unique quantity to accurately quantify the energy dissipation of the system. It includes not only the energy demand by the electronic circuit, but also the energy overhead incurred by energy flowing among different system components. We also addressed the need in run-time prediction of future harvested energy. These two contributions significantly improve the accuracy of energy availability computation for the proposed Model-Accurate Predictive DVFS algorithm. Experimental results show the improvements achieved by the MAP-DVFS algorithm in deadline miss rate. Through experiments we have also illustrated the trend of system performance variation under different conditions and system design parameters. The effectiveness of different prediction techniques is also compared by simulation.

References

[1] Hong, D. Kirovski, G. Qu, M. Potkonjak, and M. B. Srivastava, “Power Optimization of Variable-Voltage Core-Based Systems,” *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Jan. 1999.

[2] J. Luo, and N. K. Jha, “Static and Dynamic Variable Voltage Scheduling Algorithms for Real-Time Heterogeneous Distributed Embedded Systems,” *Proc. VLSI Design*, Jan. 2002.

[3] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M. B. Srivastava “Design Considerations for Solar Energy Harvesting Wireless Embedded Systems”, *Proc. of the International Symposium on Information Processing in Sensor Networks*, Apr. 2005.

[4] X. Jiang, J. Polastre, and D. E. Culler, “Perpetual Environmentally Powered Sensor Networks”, *Proc. of the International symposium on Information Processing in Sensor Networks*, Apr. 2005.

[5] C. Moser, J.-J. Chen, and L. Thiele, “Reward Maximization for Embedded Systems with Renewable Energies,” *Proc. of the International Conference on Embedded and Real-Time Computing Systems and Applications*, Aug. 2008.

[6] C. Moser, J.-J. Chen, and L. Thiele, “Optimal Service Level Allocation in Environmentally Powered Embedded Systems,” *Proc. of ACM Symposium on Applied Computing*, Mar. 2009.

[7] C. Moser, L. Thiele, D. Brunelli, and L Benini, “Robust and Low Complexity Rate Control for Solar Powered Sensors,” *Proc. of Design, Automation and Test in Europe*, Mar. 2008.

[8] B. A. Allavena, A. Allavena, and D. Mossé, “Scheduling of Frame-Based Embedded Systems with Rechargeable Batteries,” *Proc. of Workshop on Power Management for Real-time and Embedded Systems*, May 2001.

[9] C. Rusu, R. G. Melhem, and D. Mossé, “Multi-Version Scheduling in Rechargeable Energy-Aware Real-Time Systems”, *Journal of Embedded Computing*, 2005.

[10] C. Moser, D. Brunelli, L. Thiele, and Luca Benini, “Lazy Scheduling for Energy-Harvesting Sensor Nodes,” *Proc. of Fifth Working Conference on Distributed and Parallel Embedded Systems*, Oct. 2006.

[11] C. Moser, J.-J Chen and L. Thiele, “Real-Time Scheduling for Energy Harvesting Sensor Nodes,” *Real-Time System*, vol. 37, no.3, pp.233-260, 2007.

[12] S. Liu, Q. Qiu, and Q. Wu, “Energy Aware Dynamic Voltage and Frequency Selection for Real-Time Systems with Energy Harvesting”, *Proc. of Design, Automation, and Test in Europe*, Mar. 2008.

[13] S. Liu, Q. Qiu, and Q. Wu, “An Adaptive Scheduling and Voltage/Frequency Selection Algorithm for Real-time Energy Harvesting Systems”, *Proc. of Design Automation Conference*, Jul. 2009.

[14] J. Hsu, S. Zahedi, A. Kansal, M. Srivastava, V. Raghunathan, “Adaptive duty cycling for energy harvesting systems,” *Proc of the ISLPED*, Oct., 2006.

[15] J. Recas, C. Bergonzini, D. Atienza, and T. S. Rosing, “Prediction andmanagement in energy harvested wireless sensor nodes,” *Proc. VITAE’09*, May 2009, pp. 6–10.

[16] C. Bergonzini, D. Brunelli, and L. Benini, “Algorithms for harvested energy prediction in batteryless wireless sensor networks,” *Advances in sensors and Interfaces, 2009. IWASI 2009.3rd International Workshop*, June 2009, pp. 144–149.

[17] C. Xian, Y. Lu, Z. Li, “Energy-Aware Scheduling for Real-Time Multiprocessor Systems with Uncertain Task Execution Time,” *Proc. of Design Automation Conference*, 2007.

[18] M. Pedram, Q. Wu, "Design Considerations for Battery-Powered Electronics," *Proc. of Design Automation Conference*, 1999.

[19] A. M. van Voorden, L. M. R. Elizondo, G. C. Paap, J. Verboomen and L. van der Sluis, “The Application of Super Capacitors to relieve Battery-storage systems in Autonomous Renewable Energy Systems,” *IEEE Power Tech Conference*, 2007, pp. 479-484.

[20] D. C. Montgomery, L. A. Johnson, J. S Gardiner, “Forecasting and Time Series Analysis”, McGraw-Hill, 1990.

[21] A. K. Palit, D. Popović, “Computational Intelligence in Time Series Forecasting”, Springer, 2005.

[22] “*Silicon-cell Photodiode Pyranomters*,” Apogee Instruments, Inc., http://www.apogeeinstruments.com/pyr_spec.htm