# Extending the Lifetime of Fuel Cell Based Hybrid Systems[*]

Jianli Zhuo[1], Chaitali Chakrabarti[1], Naehyuck Chang[2], Sarma Vrudhula[3]
[1]Dept. of Electrical Engineering, Arizona State University, Tempe, AZ, 85287, U.S.
[2]School of CSE, Seoul National University, Seoul, Korea
[3]Dept. of CSE, Arizona State University, Tempe, AZ, 85287, U.S.
jianli@asu.edu, chaitali@asu.edu, naehyuck@snu.edu.kr, vrudhula@asu.edu

## ABSTRACT

Fuel cells are clean power sources that have much higher energy densities and lifetimes compared to batteries. However, fuel cells have limited load following capabilities and cannot be efficiently utilized if used in isolation. In this work, we consider a hybrid system where a fuel cell based hybrid power source is used to provide power to a DVFS processor. The hybrid power source consists of a room temperature fuel cell operating as the primary power source and a Li-ion battery (that has good load following capability) operating as the secondary source. Our goal is to develop polices to extend the lifetime of the fuel cell based hybrid system. First, we develop a charge based optimization framework which minimizes the charge loss of the hybrid system (and not the energy consumption of the DVFS processor). Next, we propose a new algorithm to minimize the charge loss by judiciously scaling the load current. We compare the performance of this algorithm with one that has been optimized for energy, and demonstrate its superiority. Finally, we evaluate the performance of the hybrid system under different system configurations and show how to determine the best combination of fuel cell size and battery capacity for a given embedded application.

## Categories and Subject Descriptors

D.4.1 [**Operating Systems**]: Process Management—*Scheduling*

## General Terms

Algorithms

## Keywords

Fuel cell, Battery, Hybrid systems, DVFS system, Task scaling

## 1. INTRODUCTION

Fuel cells are alternative power sources that have recently attracted a great deal of attention. They have been used to provide power in power plants, automobiles and are now being considered for portable applications. Fuel cells have very high energy densities ($>$ 2000 Watt-hour per kilogram) compared to batteries (typically $<$ 200 watt-hour per kilogram) [1]. Consequently, a fuel cell package is expected to generate power longer (4 to 10X) than a battery package of the same size and weight [2, 3].

Fuel cells are designed to generate a fixed amount of output current and have very limited load following capability. This is primarily due to the sluggish reactions in fuel cell electrodes (mainly oxygen electro reduction with exchange current density of $10^{-10}$A/cm$^2$) as opposed to battery electrode reactions (typically metal corrosion reaction whose exchange densities are $10^{-2}$A/cm$^2$) [2, 3]. Thus for embedded applications where there is substantial load current variation, it is important that a hybrid system be considered – the fuel cell would provide the bulk of the current and would serve as the primary source, and the battery would provide the load following capability and serve as the secondary source.

In recent years, a lot of work has been done to improve the power/energy/battery efficiency of embedded systems [4, 5, 6, 7, 8]. There is also some work on systems powered by solar sources [9]. This is different from fuel cell systems because the solar sources are not controllable. The existing work on fuel cells and fuel cell based hybrid power systems are typically for high temperature operations and are used in automobiles or stationary settings such as buildings and plants [10, 11, 12].

In this paper, we consider a hybrid power source built with a sodium borohydride ($NaBH_4$) fuel cell that works at room temperature and a rechargeable Li-ion battery. The fuel cell is operated at the point of maximum efficiency and has a constant output current. The Li-ion battery has a fixed capacity and is capable of providing multiple levels of output current. The hybrid power source provides power to a DVFS processor whose current can be scaled. When the fuel cell current $I_F$ is larger than the load current $i_k$, the Li-ion battery is charged, but once the battery gets fully charged, the excess charge is wasted. When the fuel cell current is smaller than the load current, the Li-ion battery provides the difference $i_k - I_F$. We develop policies for enhancing the lifetime of the fuel cell in such a hybrid system. The policies are based on minimizing the overall charge loss defined as the difference between the loss in the fuel cell stored charge and the gain in the battery charge. We present an algorithm to scale the load current subject to deadline and battery charge constraints, such that the system successfully executes the tasks with minimum charge loss. We consider both single tasks and multiple tasks in a static scheduling environment.

While there has been a significant amount of work in energy optimization of DVFS systems for unlimited energy sources, this is the first work on system level optimization of fuel cell based hybrid systems. While in traditional DVFS systems, the energy consumption due to task execution is minimized, here we consider both the

[*]This research was funded in part by the NSF grant (CSR-EHS 05059540), the Consortium for Embedded Systems, ASU, and LG Yonam Foundation.

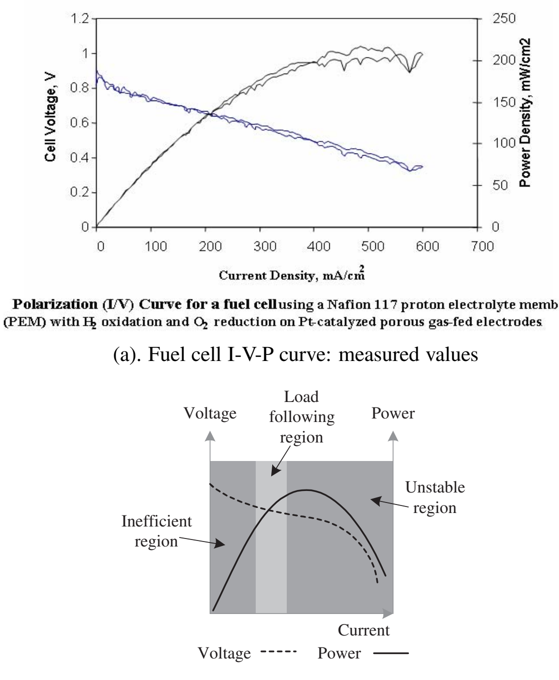
*DAC 2006,* July 24–28, 2006, San Francisco, California, USA.
Copyright 2006 ACM 1-59593-381-6/06/0007 ...$5.00.

hybrid power source and the DVFS consumer, and minimize the total charge loss of the system. The main contributions of this paper are as follows.

- Description of a room-temperature fuel cell + Li-ion battery based hybrid system (Section 2)
- Development of the charge based optimization framework for hybrid systems. The objective is to minimize the charge loss of the hybrid system and not the energy consumption of the DVFS processor (Section 3).
- Development of a new algorithm to scale the load current during static scheduling that minimizes the charge loss (Section 3)
- Evaluation of the performance of charge loss based scaling algorithm and its use in determining the best combination of fuel cell size and battery capacity for a given embedded application (Section 4).

## 2. FUEL CELL BASED SYSTEMS

The sodium borohydride ($NaBH_4$) fuel cell system that operates at room temperature [2, 3] consists of (i) a fuel cell based on polymer electrolyte membrane (PEM) with hydrogen as the anode and air as the cathode, and (ii) a hydrogen generator that includes a sodium borohydride based hydrogen storage, low power piezo pump to move the liquid fuel, a Ruthenium based catalytic reaction to release hydrogen from the sodium borohydride solution and a liquid gas separator.



**Polarization (I/V) Curve for a fuel cell** using a Nafion 117 proton electrolyte membrane (PEM) with $H_2$ oxidation and $O_2$ reduction on Pt-catalyzed porous gas-fed electrodes

(a). Fuel cell I-V-P curve: measured values

(b). Fuel cell I-V-P curve: exaggerated view

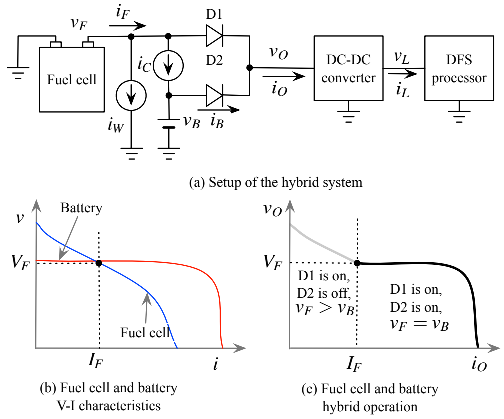
**Figure 1: Polarization curves for room temperature fuel cell**

Fig 1.(a) describes the polarization (I/V, I/P) curves for this fuel cell system using Nafion 117 PEM with hydrogen oxidation and oxygen reduction on Pt-catalyzed porous gas fed electrodes [2, 3]. The characteristics of the I-V-P curves can be explained by electron transfer that occurs at very low currents and mass transport that occurs at high currents.

From the curves, we see that as the current density increases, the voltage drops, and the power density first increases and then decreases. The operating point of the fuel cell is not set at the maximum power density point but at P=$\frac{2}{3}$maxpower. This is because the system is unstable at or beyond the maximum power density point and the system is cost-inefficient at the very low power region. Also, the fuel cell has some load following capability around $\frac{2}{3}$maxpower as shown in the exaggerated plot in Fig 1.(b). But to employ this load following capability, we have to add extra control circuit /devices and apply a fairly involved feedback control scheme. In this paper, we only consider the case where the fuel cell current is fixed.

In general, a fuel cell has ”slow” electrodes, ie, its reactions do not occur as quickly as the current levels that need to be drawn by the load, and so the electrodes polarize and the fuel cell loses voltage. As a result, for applications where the load current fluctuates, the fuel cell has to be supported by a battery which has ”fast” electrodes and is a much better load follower.

The proposed fuel cell based hybrid system consists of the following components: $NaBH_4$ based fuel cell, Li-ion rechargeable battery, DC-DC converter and the system board. Fig 2.(a) shows the schematic of the fuel cell based hybrid system.



(a) Setup of the hybrid system

(b) Fuel cell and battery V-I characteristics

(c) Fuel cell and battery hybrid operation

**Figure 2: Fuel cell based hybrid system**

The operation of this system is explained using Fig 2.(b) and Fig 2.(c). The V-I characteristics of the fuel cell and the battery are shown in Fig 2.(b). While the fuel cell output voltage, $v_F$, drops significantly as the fuel cell current, $i_F$, increases, the battery voltage, $v_B$, remains stable (since Li-ion batteries are good load followers). Let us assume that the fuel cell is designed to operate at $(V_F, I_F)$. Then the battery output voltage, $v_B$, is set to $v_B = V_F$. When the load current is less than $I_F$, the fuel cell current is kept constant at $I_F$ and the difference between the fuel cell current and the load current is used to charge the battery with current $i_C$. If the battery is fully charged, the extra current is discharged to ground in the form of bleeder current, $i_W$. Thus when $i_O < I_F$, $i_F = I_F = i_C + i_W$. If the load current is greater than $I_F$, ($i_O > I_F$), the battery provides the remaining current, $i_B = i_O - I_F$. If the battery charge is depleted, the system fails. However, for a DVFS system, $i_L$ can be scaled down which translates to $i_O$ being scaled down such that $i_O \leq I_F$.

Use of DVFS has its advantages and disadvantages. On the one hand, DVFS stretches the execution of the tasks and reduces the shutdown time of the fuel cell. On the other hand, DVFS reduces the current load and allows the battery to store charge. In the next section, we describe a procedure to judiciously scale the current in a DVFS system such that the overall charge loss in the system is minimized.

# 3. FUEL CELL EFFICIENT SCALING

We begin with the notations that have been used in the rest of this paper. The hybrid power source is characterized by $I_F$, the fuel cell current, and $B^{max}$, the charge capacity of the Li-ion battery.

Let $s_k$ be the frequency scaling factor while executing task $T_k$. The task execution time is then $s_k \times \tau_k$, where $\tau_k$ is the worst case execution time at the highest frequency (corresponding to $s_k = 1$).

The total power consumption of the DVFS processor is given by

$$P = C \cdot V_{dd}^2 \cdot f + P_{on} + V_{dd} \cdot I_{static}$$

where the first term is the dynamic power, the second term is the intrinsic power, and the last term is the static power [7, 8]. If we assume that for scaling factor $s_k$, the voltage scales by $s_k$, and both $P_{on}$ and $I_{static}$ are constant, then the total power consumption of task $T_k$, $P_k(s_k)$ is given by

$$P_k(s_k) = \alpha_1 \cdot P_k(1) \cdot s_k^{-3} + \alpha_2 \cdot P_k(1) + (1 - \alpha_1 - \alpha_2) \cdot P_k(1) \cdot s_k^{-1}$$

where $P_k(1)$ is the total power consumption at $s_k = 1$, $\alpha_1$ and $\alpha_2$ are the ratios of dynamic power to total power and intrinsic power to the total power at $s_k = 1$. In this paper, we assume that $\alpha_1 + \alpha_2 = 0.8$. This implies that the static power is 20% of the total power.

The load current of the fuel cell/ battery hybrid source (the input current $i_O$ of the DC-DC converter in Fig. 2(a)) is $i_k(s_k) = \beta \times P_k(s_k)$, where $\beta$ is a system specified constant in the order of $V^{-1}$; $\beta = (v_O \times \eta)^{-1}$, where $v_O = V_F$ is a constant and $\eta$ is the DC-DC efficiency which is also assumed constant [13].

When $T_k$ starts executing, the battery charge is $B_k^{ini}$, and when $T_k$ finishes, the battery charge is $B_k^{end}$. If the task profile consists of $n$ tasks, then the charge consumed by the task profile is $Q_{task} = \sum_1^n i_k(s_k) \times s_k \times \tau_k$. Assume that the fuel cell starts at $t_{start}$ and shuts down at $t_{end}$, then the total charge loss of the system is the difference between the charge loss in the fuel cell $I_F \times (t_{end} - t_{start})$ and the charge gain in the battery $B_n^{end} - B_1^{ini}$. Thus $Q_{sys} = I_F \times (t_{end} - t_{start}) - \left(B_n^{end} - B_1^{ini}\right)$. Note that $B_n^{end} \leq B^{max}$. The difference between $Q_{sys}$ and $Q_{task}$ is the wasted charge $Q_{waste} = Q_{sys} - Q_{task}$. Table 1 provides a list of all these parameters.

**Table 1: Definition of fuel cell system parameters**

| | |
|---|---|
| $I_F$ | the output current of the fuel cell, it is a constant. |
| $B^{max}$ | the charge capacity of the battery |
| $T_k$ | the $k-th$ task in the task profile |
| $s_k$ | voltage / frequency scaling factor of $T_k$, $s_k \geq 1$ |
| $\tau_k$ | the worst case execution time of $T_k$ |
| $P_k(s_k)$ | the processor power when task $T_k$ is scaled by $s_k$ |
| $\alpha_1$ | the ratio of the dynamic power to the total power when $s_k$=1 |
| $\alpha_2$ | the ratio of the intrinsic power to the total power when $s_k$=1; $\alpha_1 + \alpha_2 = 0.8$ in this paper |
| $\eta$ | the DC-DC converter efficiency, $\eta = 0.8$ here |
| $v_O$ | the load voltage (DC-DC input), $v_O = V_B = 1.5V$ here. |
| $\beta$ | $\beta = (\eta \times v_O)^{-1} = 0.833$ |
| $i_k(s_k)$ | the load current $i_k(s_k) = \beta \cdot P_k(s_k)$ when $T_k$ scaled by $s_k$. |
| $B_k^{ini}$ | the battery charge value when task $T_k$ starts |
| $B_k^{end}$ | the battery charge value when task $T_k$ finishes |
| $Q_{task}$ | charge consumed by the task execution |
| $Q_{sys}$ | total charge loss of the system |
| $Q_{waste}$ | wasted charge given by $Q_{sys} - Q_{task}$ |
| $s_k^{opt}$ | the scaling factor which minimizes $Q_{sys}$ |
| $s_k^{task}$ | the scaling factor which minimizes $Q_{task}$ |

## 3.1 Motivational example

Consider a DVFS system where the frequency can be scaled down from 1 to 2.5 with steps of 0.1. Let the load current $i_k(1) =$ 500mA (corresponding to processor power $P_k(1) = 600$mW) and the power ratios $\alpha_1 : \alpha_2 = 4 : 1$. The power source consists of a fuel cell with $I_F = 500$mA, and a battery with $B^{max} = 2000$mA-min. Let task $T_k$ have an execution time $\tau_k = 10$min and deadline 30min. The initial state of the battery is $B_k^{ini} = 1000$mA-min. We study the effect of two different policies, namely, energy efficient scaling, and fuel cell efficient scaling.

**CASE I – Energy efficient scaling:** Here the DVFS processor is scaled by $s_k$, where $s_k$ is obtained by minimizing the energy function $P_k(s_k) \times s_k \times \tau_k$ [8]. For $\alpha_1 : \alpha_2 = 4 : 1$, this scaling factor is $s_k = 2$. The scaled load current is 170mA and the task finishes at $t = \tau_k \times s_k = 20$min. The fuel cell can be shut down for $30 - 20 = 10$min. The charge metrics are $Q_{task} = 3400$mA-min, $Q_{sys} = 9000$ mA-min, and $Q_{waste} = 5600$ mA-min.

**CASE II – Fuel cell efficient scaling:** If instead, the scaling factor is chosen to be $s_k = 1.1$, the load current is 411mA, and the task finishes at time $t = 10 \times 1.1 = 11$min. The fuel cell can be shut down for 19 min (compared to 10min in case I). Here $Q_{task} = 4521$mA-min, $Q_{sys} = 4521$mA-min, and $Q_{waste} = 0$, which are all significantly lower than the values in Case I.

Thus the scaling factor that minimizes energy should not be used in a fuel cell based hybrid system. Next, we propose a scaling policy that reduces the charge loss and enables the fuel cell to be shut down for a longer time, thereby extending its lifetime.

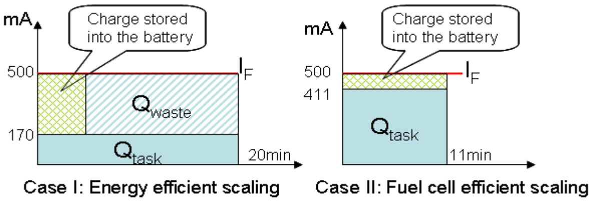


**Figure 3: Motivational Example.**

## 3.2 Scaling algorithm

The input to the task scaling algorithm is a sequence of tasks, along with their specifications (deadlines, WCETs, current), $I_F$, $B^{max}$, and the state of charge of the battery, $B_k^{ini}$. Each task is scaled such that the total charge loss $Q_{sys}$ is minimized subject to the deadline constraint and the battery charge constraint.

### 3.2.1 Determining the scaling factor for a single task

We first consider how to minimize $Q_{sys}$ when we execute a single task. We use the notation $Q_{sys}(s_k)$ when we consider a single task $T_k$. The task duration is $s_k \times \tau_k$, and the charge loss is

$$Q_{sys}(s_k) = I_F \times s_k \times \tau_k - (B_k^{end} - B_k^{ini}) \qquad (1)$$

If $I_F > i_k(s_k)$, then ideally the battery can be charged to $B_k^{ini} + \left(I_F - i_k(s_k)\right) \times s_k \times \tau_k$. Since the battery charge cannot exceed $B^{max}$, we have

$$B_k^{end} = \min\left(B^{max}, B_k^{ini} + \left(I_F - i_k(s_k)\right) \times s_k \times \tau_k\right) \qquad (2)$$

When $I_F < i_k(s_k)$, the battery is discharged to $B_k^{ini} - \left(i_k(s_k) - I_F\right) \times s_k \times \tau_k$. In order to ensure the feasibility, we include a battery charge constraint:

$$Charge\ \ constraint : \left(i_k(s_k) - I_F\right) \times s_k \times \tau_k \leq B_k^{ini} \qquad (3)$$

The total charge loss is then

$$Q_{sys}(s_k) = I_F \cdot s_k \cdot \tau_k - \min\left(B^{max} - B_k^{ini}, \left(I_F - i_k(s_k)\right) \cdot s_k \cdot \tau_k\right)$$

Since $\tau_k$ is a constant for a given task, minimizing the objective function in Equation (1) is the same as minimizing the following objective function:

$$O(s_k) = I_F \times s_k - min\left(\frac{B^{max} - B_k^{ini}}{\tau_k}, s_k \times \left(I_F - i_k(s_k)\right)\right) \qquad (4)$$

The scaling factor which maximizes $O(s_k)$ is defined as the *optimal scaling factor* $s_k^{opt}$.

Next, we study the effect of the following three parameters on $s_k^{opt}$ and $Q_{sys}$: $I_F$, the fuel cell current, $B^{max}$, the charge capacity of the battery, and $\alpha_1 : \alpha_2$, which models the characteristics of the DVFS processor. For this analysis, we assume that the deadline constraint and the battery charge constraint are always valid.

Let $Q_{sys}(s_k^{opt})$ be the optimal charge loss for a *fixed* $I_F$ and $B^{max}$ configuration, and let $Q_{sys}^{min}$ be the minimum possible charge loss among all possible configurations with different values of $I_F$ and $B^{max}$. Note that $Q_{sys}(s_k^{opt}) \geq Q_{sys}^{min}$.

Since $Q_{sys} = Q_{task} + Q_{waste}$, if we minimize both $Q_{task}$ and $Q_{waste}$, then we can achieve $Q_{sys}^{min}$. We call the scaling factor which minimizes $Q_{task}(s_k) = i_k(s_k) \cdot s_k \cdot \tau_k$ as the task optimal scaling factor $s_k^{task}$. If there is unlimited battery capacity, the battery charge at the end of execution of task $T_k$ is $\hat{B}_k^{end} = B_k^{ini} + \left(I_F - i_k(s_k)\right) \cdot s_k \cdot \tau_k$. The wasted charge occurs when $\hat{B}_k^{end} > B^{max}$, and $Q_{waste} = \hat{B}_k^{end} - B^{max}$. Note that the minimum possible charge loss $Q_{sys}^{min}$ occurs when $s_k^{opt} = s_k^{task}$ ($Q_{task}$ is minimized) *and* $Q_{waste} = 0$. If due to any change in the parameters, $s_k^{opt} \neq s_k^{task}$ *or* $Q_{waste} > 0$, then the optimal charge loss $Q_{sys}(s_k^{opt})$ is greater than the minimum possible charge loss $Q_{sys}^{min}$.

**Effect of $I_F$:** Let $\tilde{I}_F$ be the fuel cell current for which $Q_{sys}(s_k^{opt}) = Q_{sys}^{min}$, i.e, $s_k^{opt} = s_k^{task}$ and $Q_{waste} = 0$. If the actual fuel cell current $I_F > \tilde{I}_F$, the charge loss is still $Q_{sys}(s_k^{opt}) = Q_{sys}^{min}$ as long as $\hat{B}_k^{end} \leq B^{max}$. If $I_F$ is large enough that $\hat{B}_k^{end} > B^{max}$, then it can be shown that $Q_{sys}(s_k^{opt}) > Q_{sys}^{min}$. For instance, if $s_k^{opt} = s_k^{task}$ so that $Q_{task}$ is still minimized, $Q_{waste}$ increases, and $Q_{sys}(s_k^{opt}) > Q_{sys}^{min}$. If, on the other hand, $s_k^{opt}$ is decreased such that $\hat{B}_k^{end} \leq B^{max}$, then $Q_{waste} = 0$. But since $Q_{task}(s_k^{opt}) > Q_{task}(s_k^{task})$, $Q_{sys}(s_k^{opt}) > Q_{sys}^{min}$.

Now if $I_F < \tilde{I}_F$, $Q_{waste} = 0$, but as the battery gets discharged more and more, the charge constraint (Eqn.(3)) is violated when $s_k^{opt} = s_k^{task}$. In order to let the task survive, we set $s_k^{opt} > s_k^{task}$, and thus $Q_{sys}(s_k^{opt}) > Q_{sys}^{min}$.

**Effect of $B^{max}$:** Let $\tilde{B}^{max}$ be the capacity for which $Q_{sys}(s_k^{opt}) = Q_{sys}^{min}$. If $B^{max} > \tilde{B}^{max}$, $Q_{waste} = 0$ since $\hat{B}_k^{end} \leq B^{max}$. So $s_k^{opt} = s_k^{task}$. But a large battery increases the package size and weight and may not be a good option.

If $B^{max} < \tilde{B}^{max}$, as long as $\hat{B}_k^{end} \leq B^{max}$, the charge loss is still $Q_{sys}(s_k^{opt}) = Q_{sys}^{min}$. If $B^{max}$ is decreased such that $\hat{B}_k^{end} > B^{max}$, then $Q_{waste} > 0$ and $Q_{sys}(s_k^{opt}) > Q_{sys}^{min}$. Also smaller $B^{max}$ affects the charge constraint (Eqn.(3)) and may result in system failure.

**Effect of power ratio $\alpha_1 : \alpha_2$:** The power ratio $\alpha_1 : \alpha_2$ affects $s_k^{task}$. Note that $Q_{task}$ is minimized when $s_k^{task} = \sqrt[3]{\frac{2\times\alpha_1}{\alpha_2}}$. Thus if $\alpha_1 : \alpha_2 < 0.5$, it is not necessary to scale the task at all, i.e. $s_k^{task}$ should be set to 1.

### 3.2.2 *Determining the scaling factors for a sequence of tasks*

Assume that there are $n$ tasks ($T_1$, $T_2$,...,$T_n$); all tasks arrives at time 0 and share a same deadline $D$. The total charge loss $Q_{sys}$ is

$$Q_{sys}(s_1, s_2, .., s_n) = \sum_{k=1}^{n} Q_{sys}(s_k) \qquad (5)$$

where $Q_{sys}(s_k)$ is given in Eqn (1), and $B_k^{end} = min\left(B^{max}, B_k^{ini} + \left(I_F - i_k(s_k)\right) \times s_k \times \tau_k\right)$ ($\forall k$), $B_k^{ini} = B_{k-1}^{end}$ for $k > 1$, and $B_1^{ini}$ is defined as the charge in the battery before execution of task $T_1$.

The constraints are

$$Deadline \quad : \quad \sum_{k=1}^{n} (s_k \times \tau_k) \leq D$$

$$Charge \quad : \quad \left(i_k(s_k) - I_F\right) \times s_k \times \tau_k \leq B_k^{ini}, \forall k$$

In order to find $s_1, s_2, ...s_n$, we first consider the case when the battery has unlimited capacity and the battery is fully charged. In this case there is no wasted charge, and minimizing $Q_{sys}$ is the same as minimizing $Q_{task}$. Eqn.(5) then reduces to

$$Q_{sys}(s_1, s_2, .., s_n) = \sum_{k=1}^{n} \left(i_k(s_k) \times s_k \times \tau_k\right) \qquad (6)$$

We also assume that the current $i_k(1)$ and the static current ratio $\alpha$ are the same for all the tasks, If there are no deadline constraints, then the objective function is minimized when $s_1 = s_2 .. = s_n = s^{task}$, where $s^{task} = \sqrt[3]{\frac{2\times\alpha_1}{\alpha_2}}$. Now if there is a deadline constraint $\sum_{k=1}^{n} (s_k \times \tau_k) \leq D$, then if $\sum_{k=1}^{n} (s^{task} \times \tau_k) > D$, the objective function is minimized when all tasks have the same scaling factor $s_1 = s_2 = ... = s_n = s$, and $s$ is the largest scaling factor $< s^{task}$; otherwise $s_1 = s_2 = ... = s_n = s^{task}$. Details of the proof are given in [14].

### 3.2.3 *Algorithm description*

The proposed fuel cell efficient task scaling algorithm, *fc_scale*, consists of three main steps. Algorithm 1 describes the pseudo code of this algorithm.

**Step1:** Calculate the scaling factor $s_k^{even}$ of task $T_k$ based on even distribution of slack to all the remaining tasks (line 3).

**Step2:** Choose the scaling factor based on $s_k^{even}$, $s_k^{opt}$ which minimizes the charge loss, $s_k^{min}$ which ensures that the battery charge constraint is satisfied, and $s_k^{max}$ which ensures that the deadline constraint is satisfied (lines 4-9).

**Step3:** Scale task and compute $B_k^{end}$ (lines 10-11).

**Algorithm 1** Pseudo code of Algorithm *fc_scale*

```
 1: WHILE $k \leq n$ DO
 2: Input: task $T_k$ with $\tau_k$, battery with $B^{max}$ and $B_k^{ini}$.
 3: calculate $s_k^{even}$ by evenly distributing the available slack to all remaining
    tasks.
 4: determine $s_k^{opt}$ which minimizes $O(s)$ in Eqn (4);
 5: determine $s_k^{min}$ by battery charge constraint (Eqn (3));
 6: determine $s_k^{max}$ by deadline constraint;
 7: if $s_k^{min} > s_k^{max}$ then return FAILURE end if
 8: if $s_k^{opt} > min(s_k^{even}, s_k^{max})$ then $s_k^{opt} = min(s_k^{even}, s_k^{max})$ endif
 9: if $s_k^{opt} < s_k^{min}$ then $s_k^{opt} = s_k^{min}$ end if
10: execute task $T_k$ by $s_k^{opt}$ till it finishes;
11: $B_k^{end} = min(B^{max}, B_k^{ini} + (I_F - i_k(s_k^{opt})) \times s_k^{opt} \times \tau_k)$;
12: if $B_k^{end} < 0$ then return FAILURE;
13: else $B_{k+1}^{ini} = B_k^{end}$, k=k+1; end if
14: END WHILE
```

# 4. SIMULATION RESULTS

In this section, we first compare the performance of two algorithms: Algorithm *fc_scale* which is the proposed task scaling algorithm that scales the tasks by $s_k^{opt}$, and Algorithm *en_scale* which is the task scaling algorithm that scales each task to minimize the energy consumption of the DVFS processor. Since we assume that all tasks have the same energy cost function, *en_scale* evenly distributes the slack among all the tasks to minimize energy [4] but ensures that that the frequency scaling is not greater than the optimal scaling factor [8]. Then we compare the performance of Algorithm *fc_scale* under different system configurations.

## 4.1 Experimental setting

The DVFS system supports CPU scaling factor from 1 to 2.5 with steps of 0.1. The load current is set to $i_k(1) = 700$mA, and the power ratio is set to $\alpha_1 : \alpha_2 = 4 : 1$.

The battery capacity $B^{max}$ is varied in the experiments, and the initial charge in the battery is assumed to be half of the capacity. The fuel cell has constant output current $I_F$. We assume that the fuel cell is shut down after all the tasks are completed.

The task set used in the experiments is a task graph (could be dependent or independent task nodes), where all tasks share the same deadline. Here each task sequence consists of 50 to 100 tasks. Each task has $\tau_k$ from 1 to 2 min. The task density is $\mu = \sum_{k=1}^{n} \frac{\tau_k}{D}$, where $D$ is the deadline. $\mu$ varies from 0.3 to 0.9. For each task density value, we run 100 task sequences, and then get the average values for all the feasible cases.

## 4.2 Experiment 1: en_scale vs. fc_scale

Consider the case when $I_F = 800$mA and $B^{max} = 250$mA-hr. Table 2 compares the performance of the two algorithms with respect to the charge metrics: $Q_{sys}$, $Q_{task}$, $Q_{waste}$, along with $E_{task}$, the average energy consumption of the task sequences, and $\delta$, the average time during which the fuel cell can be shut down.

**Table 2: Comparison of the performance of *en_scale* and *fc_scale* when $i_k(1) = 700$mA, $\alpha_1 : \alpha_2$=4:1, $B^{max}$=250mA-hr, $I_F = 800$mA**

| Item | $\mu$ | 0.9 | 0.7 | 0.5 | 0.3 |
|---|---|---|---|---|---|
| $Q_{sys}$**(A-hr)** | **en_scale** | 1.499 | 1.942 | 2.828 | 2.828 |
| | **fc_Scale** | 1.391 | 1.418 | 1.429 | 1.435 |
| $Q_{task}$**(A-hr)** | **en_scale** | 1.169 | 0.969 | 0.879 | 0.879 |
| | **fc_scale** | 1.259 | 1.255 | 1.261 | 1.265 |
| $Q_{waste}$**(A-hr)** | **en_scale** | 0.330 | 0.973 | 1.950 | 1.950 |
| | **fc_scale** | 0.132 | 0.163 | 0.169 | 0.170 |
| $E_{task}$**(kJ)** | **en_scale** | 5.051 | 4.189 | 3.795 | 3.795 |
| | **fc_Scale** | 5.439 | 5.421 | 5.447 | 5.464 |
| **$\delta$ (min)** | **en_scale** | 1.23 | 3.16 | 0 | 147.65 |
| | **fc_Scale** | 9.33 | 42.46 | 104.87 | 252.11 |

The charge consumption results in Table 2 show that even though Algorithm *fc_scale* has higher value of $Q_{task}$, it has significantly lower $Q_{waste}$. As a result, the total charge loss, $Q_{sys}$, of Algorithm *fc_scale* is much lower than that of Algorithm *en_scale*. The energy consumption in the DVFS processor, $E_{task}$, is lower for Algorithm *en_scale*, as expected. Note that $E_{task}$ only accounts for the energy consumed by the DVFS processor. If, instead, the energy consumption of the system defined as $E_{sys} = Q_{sys} \times V_F$ is considered, then Algorithm *fc_scale* has superior performance.

Table 2 also shows that the fuel cell can be shut down for a much longer time when we use Algorithm *fc_scale*. This is because Algorithm *fc_scale* considers both $B^{max}$ and $I_F$ in choosing the scaling factor. It executes the tasks at a frequency that minimizes the charge loss which in turn leads to an increase in the shutdown duration of the fuel cell. In contrast, Algorithm *en_scale* utilizes the slack as much as possible, which leads to a low value of the load current. Since $I_F$ is constant, the battery charging current is higher and once the battery is fully charged, the charge is wasted.

## 4.3 Experiment 2: Effect of fuel cell current

In Experiment 1, the fuel cell current $I_F$ was set to 800mA, which is higher than the load current at the highest frequency $i_k(1) = 700$mA. So both Algorithms *en_scale* and *fc_scale* had a large amount of wasted charge. Now if $I_F$ is reduced from 800mA in Experiment 1 to 600mA while keeping all other conditions the same, $Q_{waste}$ reduces. Fig 4 describes the charge loss (averaged among 100 random cases) of *en_scale* and *fc_scale* when $\mu = 0.7$. We see that both the algorithms have less $Q_{waste}$ and less $Q_{sys}$. In fact, Algorithm *fc_scale* has $Q_{waste} \approx 0$. This implies that for the given task set, large $I_F$ could result in unnecessarily large $Q_{waste}$. Thus for embedded applications, where the average load current is known, $I_F$ should be chosen carefully.

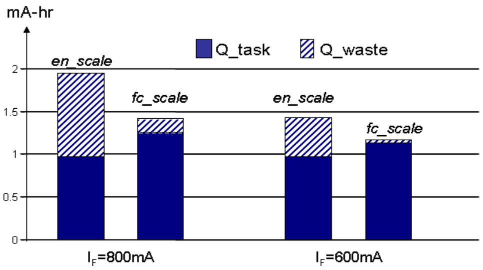


**Figure 4: Comparison of the performance of *en_scale* and *fc_scale* when $I_F = 800$mA and 600mA, for $\mu = 0.7$**

## 4.4 Experiment 3: Effect of different system configurations

We compare the performance of Algorithm *fc_scale* (also averaged among 100 random task graphs) for different values of $I_F$ and $B^{max}$. We vary $I_F$ from 250mA to 700mA with steps of 50mA, and $B^{max}$ from 100mA-min to 204800mA-min (around 3500 mA-hr). As before, the initial battery charge is half of $B^{max}$. We assume that the task current profiles and power ratios are the same as Experiments 1 and 2. Fig 5 shows the total charge loss $Q_{sys}$ when processor utilization is $\mu = 0.7$. The points with $Q_{sys}$=0 correspond to configurations which cannot support the task execution or those where the battery has discharged more than 75% at the end of the task profile.

From Fig 5, we see that for most values of $B^{max}$, $Q_{sys}$ increases as $I_F$ increases. This is to be expected since larger $I_F$ provides larger charge to the battery. But once the battery is fully charged, the excess charge is wasted. When $B^{max}$ is very large, $I_F$ has little effect on $Q_{sys}$. This is because $B^{max}$ is large enough to hold all the excess charge and there is no wasted charge. When $B^{max}$ is very small (a scenario that can occur when the secondary source is implemented by a capacitor), due to the randomness, some cases with large $I_F$ will also fail.

From Fig 5, we also see that the battery capacity $B^{max}$ only matters when $I_F$ is very small or very large. When $I_F = 250$mA, the system can only survive when battery capacity is larger than 204800mA-min (3500mA-hr). In this scenario, $I_F$ is small and the large battery has to discharge to provide the remaining current. When $I_F = 700$mA, the amount of excess charge (beyond what is required by the processor) is large. So when $B^{max}$ is large, the excess charge can be stored in the battery and $Q_{sys}$ is low.

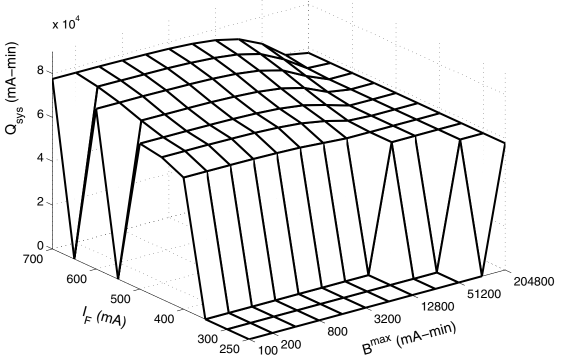


**Figure 5: Variation of $Q_{sys}$ with $I_F$ (linear scale) and $B^{max}$ (log scale) when $\mu = 0.7$**

## 4.5 Experiment 4: Choice of system configuration

Finally, we investigate what is the best system configuration for a set of tasks in an embedded application. We define the best system configuration as the one with the lowest $Q_{sys}$ and the lowest weight (volume).

Assume that we are designing a fuel cell hybrid system which can power the DVFS processor for a duration $\Delta$ (excluding the shutdown time). Then the required energy storage of the fuel cell is $I_F \times V_F \times \Delta$, and that of the battery is $B^{max} \times v_B$. Thus $\Delta$ plays an important role in choosing the best configuration.

Table 4 gives some possible configurations which achieve the lowest $Q_{sys}$ for running a set of tasks for different values of desired lifetime $\Delta$. We assume that $V_F = v_B$ =1.5V, the load current, power ratio and task density are same as that in Exp.3. By using the actual density values in Table 3 [1], we can get the corresponding weight and volume information for each configuration. Note that the total weights and volumes exclude the constant-weight parts such as battery packaging and hydrogen generator.

**Table 3: Energy density and efficiency [1]**

| | Whr/kg | Whr/l | efficiency | act. Whr/kg | act. Whr/l |
|---|---|---|---|---|---|
| **fuel cell** | 2,500 | 2,500 | 40% | 1,000 | 1,000 |
| **battery** | 200 | 400 | 80% | 160 | 320 |

**Table 4: Weights and volumes of different configurations**

| $\Delta$ | $I_F$ (mA) | $B^{max}$ (mA-hr) | weight(g) | volume(ml) |
|---|---|---|---|---|
| 10hr | 350 | 400 | 9 | 7.13 |
| | 400 | 200 | 7.87 | 6.94 |
| | 700 | 3000 | 38.62 | 24.56 |
| 50hr | 350 | 400 | 30 | 28.13 |
| | 400 | 200 | 31.87 | 30.94 |
| | 700 | 3000 | 80.63 | 66.56 |

From Table 4, it is obvious that the configurations with $I_F =$ 1000mA are bad choices, since the weights and volumes are both much larger than the other configurations. This is because $I_F$ is much higher than the load current, and to achieve the lowest charge loss, the battery has to be also very large! From Table 4, we see that when $\Delta = $ 10hr, the configuration with $I_F =$ 400mA and $B^{max} =$ 200mA-hr has smaller weight and volume compared to the configurations with $I_F =$ 350mA and $B^{max} =$ 400mA-hr. When $\Delta =$ 50hr, the configuration with $I_F =$ 350mA and $B^{max} =$ 400mA-hr has the smallest weight and size. But from Fig 5, we see that the point with $I_F =$ 350mA and $B^{max} =$ 400mA-hr is not stable, so it should be safer to choose $I_F =$ 400mA and $B^{max} =$ 200mA-hr with very small size/weight penalty. However, if the average load current is different, the best system configuration is likely to be different. In general, for embedded applications, the choice of the best system configuration depends on multiple factors including the average load current, processor power ratio, utilization, and energy densities of the fuel cell and battery. A more systematic search is clearly needed.

# 5. CONCLUSION

In this paper, we proposed a hybrid power system consisting of a room temperature fuel cell and a Li-ion rechargeable battery. We developed a charge based optimization framework for the hybrid system structure, and utilized this framework to develop a new algorithm to scale the load current such that the system charge loss was minimized. Experiments showed that the proposed algorithm has better performance than traditional energy aware algorithm for embedded systems. Furthermore, the framework can be used to determine the fuel cell and battery sizes which could achieve the best performance with the smallest total weight and volume.

# 6. ACKNOWLEDGEMENT

We sincerely thank Dr. Don Gervasio and Sonja Tasic *(Flexible Display Center, ASU)*, and Kyungsoo Lee *(School of Computer Science and Engineering, SNU)* for help with the fuel cell setup and measurement.

# 7. REFERENCES

[1] C. K. Dyer, “Fuel cells and portable electronics,” in *Symposium on VLSI circuits (digest of technical papers)*, June 2004, pp. 124–127.

[2] D. Gervasio, S. Tasic, and F. Zenhausern, “A room temperature micro-hydrogen-generator,” *Journal of Power Sources*, vol. 149, pp. 15–21, 2005.

[3] D. Gervasio, “Fuel-cell system for hand-carried portable power,” *International Fuel Cell R&D Forum*, Nov. 2005.

[4] F. Yao, A. Demers, and S. Shenker, “A scheduling model for reduced cpu energy,” in *36th Annual Symposium on Foundations of Computer Science*, Oct. 1995, pp. 374–382.

[5] G. Quan and S. X. Hu, “Energy efficient fixed-priority scheduling for real-time systems on variable voltage processors,” in *38th DAC*, June 2001, pp. 828–833.

[6] R. Rao, S. Vrudhula, and D. N. Rakhmatov, “Battery modeling for energy aware system design,” *IEEE Computer*, vol. 36, no. 12, pp. 77–87, Dec. 2003.

[7] R. Jejurikar, C. Pereira, and R. Gupta, “Leakage aware dynamic voltage scaling for real-time embedded systems,” in *41st DAC*, June 2004, pp. 275–280.

[8] J. Zhuo and C. Chakrabarti, “System-level energy-efficient dynamic task scheduling,” in *42nd DAC*, June 2005, pp. 628–631.

[9] D. Li and P. H. Chou, “Maximizing efficiency of solar-powered systems by load matching,” in *ISLPED*, 2004, pp. 162–167.

[10] R. Hahn, M. Krumm, and H. Reichl, “Thermal management of portable micro fuel cell stacks,” in *19th IEEE Semi-Therm symposium*, March 2003, pp. 202–209.

[11] K. Rajashekara, “Hybrid fuel-cell strategies for clean power generation,” *IEEE Trans. on Industry Applications*, vol. 41, no. 3, pp. 682–689, May-June 2005.

[12] W. Gao, “Performance comparison of a fuel cell-battery hybrid powertrain and a fuel cell-ultracapacitor hybrid powertrain,” *IEEE Trans on Vehicular Tech.*, vol. 54, no. 3, pp. 846–855, May. 2005.

[13] Y. Choi, N. Chang, and T. Kim, “Dc-dc converter-aware power management for battery-operated embedded systems,” in *42nd DAC*, June 2005, pp. 895–900.

[14] Y. Cho, N. Chang, C. Chakrabarti, and S. Vrudhula, “High-level power management of embedded systems with application-specific energy cost functions,” in *43rd DAC*, July 2006.