

An Efficient Scheduling Algorithm for Multiple Charge Migration Tasks in Hybrid Electrical Energy Storage Systems

Qing Xie, Di Zhu, Yanzhi Wang, and Massoud Pedram

Department of Electrical Engineering
University of Southern California
Los Angeles, California, United States, 90089
{xqing, dizhu, yanzhiwa, pedram}@usc.edu

Younghyun Kim, and Naehyuck Chang

Department of EECS/CSE
Soule National University
Soule, Korea, 151-747
{yhkim, naehyuck}@elpl.snu.ac.kr

Abstract—Hybrid electrical energy storage (HEES) systems are comprised of multiple banks of heterogeneous electrical energy storage (EES) elements with distinct properties. This paper defines and solves the problem of scheduling multiple charge migration tasks in HEES systems with the objective of minimizing the total energy drawn from the source banks. The solution approach consists of two steps: (i) Finding the best charging current profile and voltage level setting for the Charge Transfer Interconnect (CTI) bus for each charge migration task, and (ii) Merging and scheduling the charge migration tasks. Experimental results demonstrate improvements of up to 32.2% in the charge migration efficiency compared to baseline setups in an example HEES system.

I. Introduction

None of the existing EES elements satisfy all the desired performance metrics, as shown in Table 1. Hybrid EES (HEES) systems address this shortcoming by using a collection of heterogeneous EES elements suitably organized and accessed. More precisely, appropriate charge management policies, including charge allocation, charge replacement and charge migration, can take advantage of the strengths of individual types of EES elements while hiding their weaknesses [1]–[5]. The charge migration process, which moves energy between the EES banks, is a crucial operation in a HEES system primarily because it alleviates the self-discharge loss by moving stored energy to less leaky EES banks. In addition, it ensures the (energy) availability of the EES banks by controlling the state of charge (SoC) of various EES banks. Finally, it improves the efficiency of the subsequent charge allocation or replacement processes in the HEES system by pre-charging or pre-discharging appropriately selected EES banks so as to efficiently meet the dynamic power generation and load demand. The figure of merit for the charge migration task is the *charge migration efficiency*, which is defined as the ratio of the total target energy migrated to *destination* EES banks to the total energy drawn from the *source* EES banks.

Table 1: Characteristics of typical EES elements in HEES systems.

Metrics	Batteries				Super-capacitor
	Li-ion	NiMH	Lead-acid	NiCd	
power capacity	high	high	high	high	ultra high
energy density	very high	medium	low	low	very low
leakage	low	low	low	low	high
cycle life	medium	short	short	medium	very long
cycle efficiency	high	low	low	high	very high
cost	high	medium	very low	low	very high

The *single-source, single-destination (SSSD) charge migration* problem using a bus called Charge Transfer Interconnect (CTI) has been investigated in reference [4]. The SSSD migration utilizes a single CTI during the whole migration process and does not deal with many simultaneous migrations, where multiple source EES

This work is supported in part by a grant from the U.S. National Science Foundation, the Brain Korea 21 Project, the National Research Foundation of Korea (NRF) grant funded by the Korean Government (MEST) (No. 2011-0016480), and ICT at the Seoul National University.

migration process and can improve the overall charge migration efficiency [7]. The catch is that all the migrations start at the same time and share the same set of CTIs (thus they are bound to use the same CTI voltages), which can cause migration efficiency degradation if charge migration tasks are not scheduled and assigned to various CTIs prudently.

This paper introduces the charge migration scheduling (CMS) problem in HEES systems, which in turn enables a number of concurrent MSMD charge migrations through a fixed number of CTIs while meeting a global deadline constraint. In the CMS problem, we have a number of MSMD migration tasks, where each task is defined as delivering a fixed amount of energy from a given set of source EES banks to another set of destination EES banks within a deadline. The objective is to maximize the overall charge migration efficiency (by minimizing the charge that is extracted from the source banks since the charge that is delivered to the destinations banks is specified as part of the problem statement.)

The solution to the CMS problem requires solving the following sub-problems. First, we must determine the optimal CTI voltage and charging current profile for each MSMD migration task. This also depends on the duration the migration process. Second, we must decide whether two MSMD migration tasks should be merged. Merging two migration tasks will force both of them to utilize the same CTI, and thus, adopt the same CTI voltage during the migration process. To ensure that the overall charge migration efficiency for the merged task is higher than those of the individual (non-merged) migration tasks, we only allow merges between tasks that have similar optimum CTI voltage levels. Third, we assign a CTI to the migration task among the available CTIs and calculate the *CTI usage time*, defined as the duration the migration process. Notice that merging two migration tasks into one reduces the resource overhead (i.e., the number of CTIs used is reduced), but the optimum realization of the merged migration task may also result in an increase in the usage time of the shared CTI. In case of a deadline violation, the CTI usage time for the merged task must be reduced to the given deadline at the expense of a sub-optimal realization of the merged migration task. Unfortunately, the optimal solutions of the three sub-problems are not independent but coupled to each other.

We formulate the CMS problem as an optimization problem and solve it in an iterative manner since the three sub-problems mentioned above are coupled. More precisely, we start from the time-unconstrained migration of each migration task and calculate the optimal migration time. Next we reduce the CTI usage time while minimizing the increase of the total energy drawn from the sources at each step until the deadline constraint is met. During the usage time reduction, we derive the optimal CTI voltage and charging current profile over time by using *non-linear fractional programming*. We will merge multiple migration tasks if the merged migration task yields higher migration efficiency than those of the separate ones. We converge to the final solution for the CMS problem when all the migration tasks are scheduled to complete by the given deadline, and no further merging can improve the charge migration efficiency. Experimental results demonstrate significant charge migration efficiency improvements of up to 32.2% compared to baseline setups in example HEES systems.

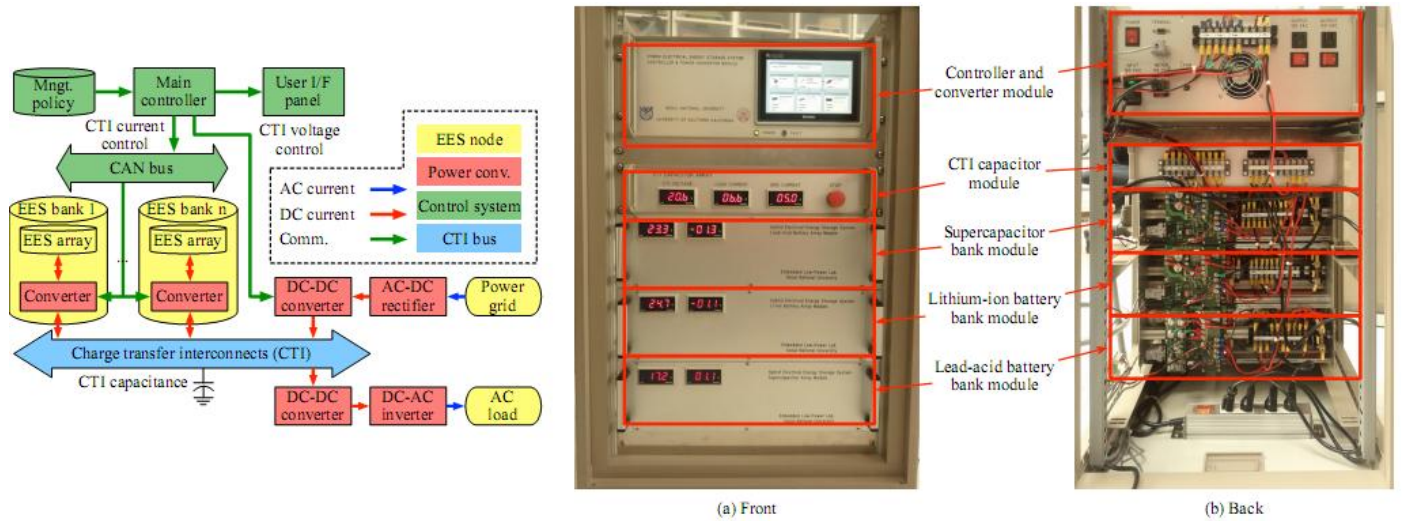


Figure 1 Architecture-level block diagram and photos of our HEES system prototype.

II. HEES System Background

Scheduling problems have been well studied in the past. Previous researches mainly focus on the CPU scheduling problem, which inspires the charge migration scheduling (CMS) problem statement in HEES systems. More precisely, we define the CMS problem such that a HEES system performs multiple, simultaneous charge migrations among EES banks while meeting a deadline constraint. This can be compared to finishing a set of tasks by a given deadline in the CPU scheduling problem. Notice that each migration task must deliver a target amount of energy to the specified destination banks (the target energy plays the same role as the workload of a task in the CPU scheduling problem). A charge migration task occupies the CTI when the transfer is taking place in the same way that a CPU can only perform a single thread at each time instance. The objective is to maximize the charge migration efficiency, which corresponds to a CPU scheduling that minimizes the energy consumption. Hence low-power CPU scheduling [8][9] and CMS problem formulation are analogous.

We formulate the CMS problem so as to utilize algorithmic concepts related to low-power CPU scheduling problem. At the same time, we differentiate CMS from low-power CPU scheduling by taking into account the HEES system characteristics. In particular, previous low-power CPU scheduling methods cannot directly be applied to the CMS problem for the following reasons. Instead of a simple super-linear relationship between the CPU energy consumption and supply voltage level in dynamic voltage and frequency scaling (DVFS)-based scheduling techniques, the charge migration efficiency is a non-convex function of the CTI voltage, charging currents, and SoCs of the involved EES banks. Moreover, we must consider the battery rate capacity, self-discharge, and conversion losses (which are absent from the CPU scheduling problem.) Finally, in the CMS problem, we can merge two migration tasks into a single combined migration task so that two tasks share the same CTI during the migration process. Merging tasks significantly reduces the CTI usage time for the whole migration process while each migration task may also receive extra CTI usage time. This is a key difference between the CMS and low-power CPU scheduling, which does not allow two tasks to simultaneously run on the same CPU. Note that, in this analogy, the CPUs here have single thread execution micro-architectures. Hence, we must develop a new approach for solving the CMS problem that builds on existing approaches for low power CPU scheduling while accounting for the aforesaid differences from the low power scheduling problem.

Figure 1 shows the HEES system prototype that we have built to demonstrate the benefits of a HEES system. The entire HEES

prototype consists of three modules: power converter module, CTI module, and EES bank module.

- The converter module is responsible for the top-level power management between the HEES prototype and the external power sources or load devices. It contains AC to DC conversion for grid power input and DC to AC conversion to support AC loads. DC to DC conversion can also be included into the system to provide ability for DC input and output.
- The CTI module is the path for charge transfer between the power sources, load devices and EES banks themselves. The voltage level setting in CTI very important since it affects the power conversion efficiency of the power converters. Thus we maintain the CTI voltage at a stable and appropriate level to perform our charge management algorithm. To address the instability issue that may caused by the temporary mismatch between the incoming and outgoing current in the CTI, we connect 132,000 uF capacitance to the CTI so that the CTI voltage is robust.
- The EES bank modules are the EES banks that store the electrical energy. We install three heterogeneous EES bank in our prototype: supercapacitor bank, lead-acid battery bank and Li-ion battery bank. An EES bank includes a homogeneous EES element array and a bidirectional converter, which is used to control the charging or discharging current of the EES array. The EES elements are organized in an appropriately constructed two-dimensional array to have proper terminal voltage and capacity. The meters on the front panel display the voltage and current of the EES array.

In this paper, we discuss CMS problem in a more general HEES system that has multiple EES banks and CTIs. However, the architecture of the general HEES system is the same as our prototype and the simulation parameters in this paper are also obtained from the characterization of the prototype.

III. Problem Statement

3.1 A Scheduling Example

There are two important factors that affect the charge migration efficiency, according to the properties of ESS elements [10] and power converters [4][12]. First, there exists the optimal charging/discharging current for each SSSD migration. Second, the power converters achieve their maximum *conversion efficiency*, defined as the ratio between the output power and input power, when the mismatch of input and output voltage (i.e., $|V_{in} - V_{out}|$) is minimized [12]. These two factors guide the proposed CMS algorithm. The migration tasks typically have different optimal CTI voltages due to the different SoCs of the EES banks. Merging two or more migration tasks allows the scheduler to assign a longer CTI

usage time so that the migration efficiency can be improved. However, merged migration tasks have to accept the same CTI voltage, which causes migration efficiency degradation if their optimal CTI voltages do not quite match each other. Indeed, the proposed CMS algorithm selectively merges those migration tasks whose optimal CTI voltages are nearly the same. Merged tasks, which will share the same CTI, are executed concurrently.

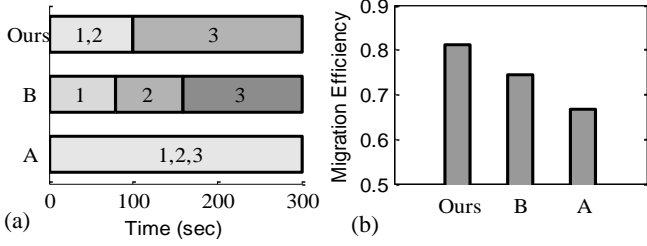


Figure 2: Three scheduling schemes: all merged (A), one at a time (B) and the proposed scheduling (Ours.)

We consider an example of three SSSD migration tasks with the target energy of (1000 J, 1000 J, 750 J) and a deadline constraint of 300 sec. There is only one CTI available. The source and destination EES banks voltages are (15 V, 15 V, 6 V) and (12 V, 12 V, 6 V), respectively. Figure 2(a) compares three scheduling schemes: A) merge all tasks and perform them simultaneously; B) perform them one by one with CTI usage time of (80 s, 80 s, 140 s); and the proposed scheduling. In this example, we merge Task1 and Task2 because they have the same optimal CTI voltage. A longer CTI usage time is assigned to Task 3 to avoid large charging current since it has a low destination EES bank voltage. The Figure 2(b) indicates the charge migration efficiency.

3.2 Problem Formulation

Table 2: Table of important notation and definitions.

The following notation relates to the i^{th} EES bank in the n^{th} migration task (the EES bank is a source bank if it is labeled as src and destination bank if it is labeled as dst).

$V_{array,n,i}^{src(dst),OC(CC)}$	open (closed) circuit voltage of the EES array
$I_{array,n,i}^{src(dst)}$	current between the source (destination) EES array and its discharging (charging) control charger
$I_{bank,n,i}^{src(dst)}$	current between the EES bank and its CTI
$I_{eq,n,i}^{src(dst)}$	equivalent current inside the EES array, considering the rate capacity effect
$P_{c,n,i}^{src(dst)}$	power conversion loss in chargers of the EES bank
$P_{sd,n,i}^{src(dst)}$	self-discharge power loss of the EES array
$P_{drawn,n,i}^{src}$	power drawn from the source EES bank
$P_{gain,n,i}^{dst}$	power pushed into the destination EES bank

The following notation relates to the n^{th} migration task:

$\{Src\}_n, \{Dst\}_n$	set of source or destination EES banks
$V_{CTI,n}$	voltage setting of the CTI
$E_{drawn,n}$	total energy drawn from all the source EES banks
E_{gain}^{tar}	target amount of energy to be pushed into the destination EES bank after the migration process
$t_{CTI,n}$	CTI usage time that is assigned to the migration task
$t_{CTI,n}^*$	global optimal CTI usage time as determined by the time-unconstrained MSMD solution (cf. Section 4.2)
T_A, T_D	arrival time and deadline of the migration process

We define a migration task as: $T(\{Src\}, \{Dst, E_{gain}^{tar}\}, T_A, T_D)$, which draws the energy from $\{Src\}$ and push E_{gain}^{tar} amount of energy into $\{Dst\}$ within the time duration of (T_A, T_D) . Note that the target energy E_{gain}^{tar} is less than the energy drawn from source EES bank due to the energy loss during the migration process.

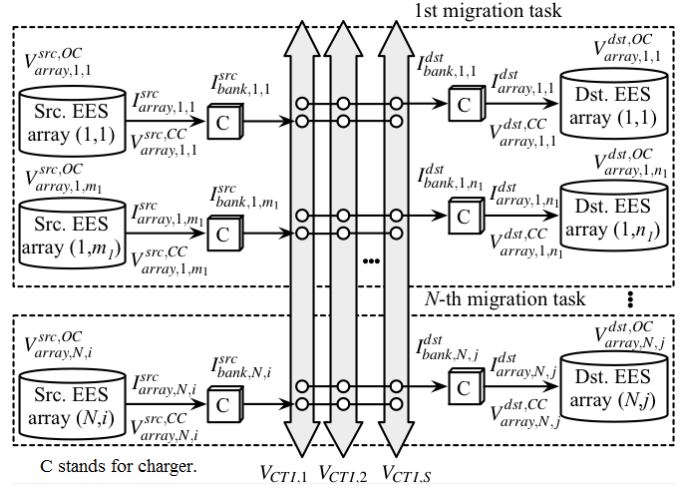


Figure 3: Conceptual diagram of charge migration tasks.

We make two assumptions to make the description succinct. First, each migration task is a MSMD migration that consists of a set of mutually exclusive SSSD migrations, i.e., all source and destination banks are paired up one-to-one. For all the SSSD migrations in one task, they must share the same CTI and commence at the same time. Second, all migration tasks arrive at the beginning of the whole migration process and have the same deadline constraint. However, they may not finish at the same time, since one SSSD migration can disconnect its source bank and destination bank from the CTI once target amount of energy, E_{gain}^{tar} , is delivered to the destination bank. Based on these two assumptions, we can merge multiple migration tasks to form a new migration task:

$$T' = T_1 \cup T_2 \cup \dots \cup T_n = T(\{Src\}_1 \cup \{Src\}_2 \cup \dots \cup \{Src\}_n, \{Dst, E_{gain}^{tar}\}_1 \cup \{Dst, E_{gain}^{tar}\}_2 \cup \dots \cup \{Dst, E_{gain}^{tar}\}_n, T_A, T_D). \quad (1)$$

Figure 3 presents a conceptual diagram of the CMS problem. For better visualization, we place the source banks and destination banks on left and right hand sides, respectively. We have N charge migration tasks, each of them containing m_n ($n \in \{1, 2, \dots, N\}$) source EES banks and m_n ($n \in \{1, 2, \dots, N\}$) destination EES banks. Note that generally the open circuit voltage (OCV) and the closed circuit voltage (CCV) of each EES bank differ from each other due to the internal resistances of the EES elements. We have S number of CTIs. Each EES bank is connected to all the CTIs through S on/off switches, shown as empty circles in Figure 3. For each EES bank, no more than one switch among these S switches can be turned on at any time. If we decide to use the s^{th} ($s \in \{1, 2, \dots, S\}$) CTI for the n^{th} migration task, we turn on the s^{th} switch of all the source and destination banks involved in the n^{th} migration task. In this case, the s^{th} CTI is occupied by the n^{th} migration task and it is unavailable for any other migration tasks unless they are merged to n^{th} task. We only consider a small number of CTIs (no more than four) due to the cost of the switches.

The CMS problem is constrained by the energy conservation relations. (2)~(5) describe the charge migration process from the source banks (on the left side) to the destination banks (on the right side). Considering the rate capacity effect, the power drawn from the source bank $Src(n, i)$ at time t is given by:

$$P_{drawn,n,i}^{src}(t) = V_{array,n,i}^{src,OC}(t) \cdot I_{eq,n,i}^{src}(t) + P_{sd,n,i}^{src}(t) \\ = V_{array,n,i}^{src,OC}(t) \cdot \left(I_{array,n,i}^{src}(t) \right)^{\gamma_{n,i}} + P_{sd,n,i}^{src}(t). \quad (2)$$

The Peukert constant, $\gamma_{n,i}$, is greater than one if the source bank is a battery bank [11] and set to one for a supercapacitor bank. A discharging control charger with a conversion power loss of $P_{c,n,i}^{src}$ is used to control the discharging current of $Src(n, i)$, thus we have:

$$V_{array,n,i}^{src,OC}(t) \cdot I_{array,n,i}^{src}(t) = P_{c,n,i}^{src}(t) + V_{CTI,n}(t) \cdot I_{bank,n,i}^{src}(t). \quad (3)$$

At the destination side, the power provided to the destination bank $Dst(n, i)$ is the power retrieved by its charging control charger from the CTI, excluding $P_{c,n,i}^{dst}(t)$, which is:

$$V_{CTI,n}(t) \cdot I_{bank,n,i}^{dst}(t) = P_{c,n,i}^{dst}(t) + V_{array,n,i}^{dst,OC}(t) \cdot I_{array,n,i}^{dst}(t). \quad (4)$$

Again, the power pushed into destination bank $Dst(n, i)$ is less than the power that is delivered to it due to the rate capacity effect:

$$\begin{aligned} P_{gain,n,i}^{dst}(t) &= V_{array,n,i}^{dst,OC}(t) \cdot I_{eq,n,i}^{dst}(t) - P_{sd,n,i}^{dst}(t) \\ &= V_{array,n,i}^{dst,OC}(t) \cdot \left(I_{array,n,i}^{dst}(t) \right)^{\beta_{n,i}} - P_{sd,n,i}^{dst}(t), \end{aligned} \quad (5)$$

where the constant $\beta_{n,i}$ is less than one if the destination bank $Dst(n, i)$ is a battery bank and equal to one if it is a supercapacitor bank, and $P_{sd,n,i}^{dst}(t)$ is the self-discharge of the EES array [8]. Since the migration task is a set of many SSSD migrations, for each SSSD migration, we have:

$$V_{CTI,n}(t) \cdot I_{bank,n,i}^{src}(t) = V_{CTI,n}(t) \cdot I_{bank,n,i}^{dst}(t). \quad (6)$$

We formulate the CMS problem as follows.

Given: (i) Initial migration tasks $T_n = (\{Src\}_n, \{Dst, E_{gain}^{tar}\}_n, 0, T_D), n \in \{1, 2, \dots, N\}$ at time $t = 0$;

(ii) SoCs of all source and destination EES banks, $SoC_{k,i}^{src}(t)|_{t=0}, SoC_{k,i}^{dst}(t)|_{t=0}, i \in \{1, 2, \dots, m_n\}, n \in \{1, 2, \dots, N\}$;

(iii) S number of available CTIs during the time period $(0, T_D)$.

Find: (i) Merged migration tasks $T'_{n'} = (\{Src\}_{n'}, \{Dst, E_{gain}^{tar}\}_{n'}, 0, T_D, n' \in \{1, 2, \dots, N'\}$ and $N' \leq N$;

(ii) CTI usage time of each migration task, $t_{CTI,n'}, n' \in \{1, 2, \dots, N'\}$;

(iii) Charging current profile and CTI voltage for each destination bank, $I_{array,n',i}^{dst}(t), V_{CTI,n'}(t), i \in \{1, 2, \dots, m_{n'}\}, n' \in \{1, 2, \dots, N'\}, t \in (0, t_{CTI,n'}^{opt})$, for n'^{th} task.

Maximize: Charge migration efficiency, defined as

$$\eta_{CM} = \frac{\sum_{n'=1}^{N'} \int_0^{t_{CTI,n'}} \sum_{i=1}^{m_{n'}} P_{gain,n',i}^{dst}(t) dt}{\sum_{n'=1}^{N'} \int_0^{t_{CTI,n'}} \sum_{i=1}^{m_{n'}} P_{drawn,n',i}^{src}(t) dt}. \quad (7)$$

Subject to:

(i) Energy constraint: $Dst(n', i)$ receives:

$$\int_0^{t_{CTI,n'}} P_{gain,n',i}^{dst}(t) dt = E_{gain,n',i}^{tar} \quad (8)$$

for all $i \in \{1, 2, \dots, m_{n'}\}, n' \in \{1, 2, \dots, N'\}$ and

(ii) Deadline constraint:

$$\begin{aligned} t_{CTI,n'} &\leq T_D, \forall n' \in \{1, 2, \dots, N'\}, \\ \sum_{n'=1}^{N'} t_{CTI,n'} &\leq S \cdot T_D. \end{aligned} \quad (9)$$

Note that we can break the CTI usage time $t_{CTI,n'}$ of a migration task into many short time periods and finish them separately. Therefore, (9) ensures that we can arrange all migration tasks without violating the deadline constraint.

(iii) Energy conservation: (2) ~ (6) are satisfied;

(iv) EES array charge conservation:

$$SoC(t) = SoC(0) + 1/C_p \int_0^t (I_{eq}(\tau) - I_{sd}(\tau)) d\tau, \quad (10)$$

where C_p is the nominal capacity (in unit of $A \cdot h$, usually defined as the capacity at one-ampere discharge rate).

IV. The Scheduling Algorithm

In this section, we present the proposed scheduling algorithm that contains three sub-algorithms: i) finding the optimal charging current profiles and CTI voltage for a single migration task; ii) determining the appropriate CTI usage time for each migration task under the deadline constraint; and iii) merging the migration tasks so

that the merged tasks have longer CTI usage time while the overall CTI usage time of all migration tasks is reduced. We present each sub-algorithm individually and combine them together to construct the overall solution.

4.1 Solving Single Migration Task

A single migration task T_n has a set of source EES banks $\{Src\}_n$, a set of destination EES banks, and the target energy $\{Dst, E_{gain}^{tar}\}_n$ to be migrated. It is a time-constrained MSMD charge migration problem that is formulated as a non-linear fractional programming problem and solved using the approach proposed in [7]. The solution, including the charging current profiles and CTI voltage, depends on the CTI usage time $t_{CTI,n}$ that is assigned to the n^{th} migration task. We will call this algorithm *MSMDSolver*:

$$\begin{aligned} [E_{drawn,n}^{opt}(t_{CTI,n}), \{I_{array,n,i}^{dst,opt}(t)\}, V_{CTI,n}^{opt}(t)] \\ = MSMDSolver(T_n, t_{CTI,n}), \end{aligned} \quad (11)$$

where $i \in \{1, 2, \dots, m_n\}, t \in (0, t_{CTI,n})$, and $E_{drawn,n}^{opt}(t_{CTI,n})$ is the total amount of energy drawn from all source banks $\{Src\}_n$ in task T_n when the optimal operating conditions are applied. To ensure that we meet the deadline constraint, $t_{CTI,n} \leq T_D$.

4.2 Determining CTI Usage Time

In this section, we introduce our algorithm to determine the CTI usage time for each migration task such that all migration tasks are completed without violating the deadline constraint. Our algorithm is based on the three propositions as follows.

Proposition 1: There exists a *global optimal CTI usage time*, $t_{CTI,n}^*$, such that for each charge migration task $n, \forall t_{CTI,n} > t_{CTI,n}^*, E_{drawn,n}^{opt}(t_{CTI,n}) = E_{drawn,n}^{opt}(t_{CTI,n}^*)$.

Proof: As we explained in Section 3.2, there exists the optimal charging current for each SSSD migration and thus a corresponding optimal CTI usage time. We define the global optimal CTI usage time for a migration task as $t_{CTI,n}^* = \max \{t_{CTI,n,i}^*, i \in \{1, 2, \dots, m_n\}\}$. The $t_{CTI,n}^*$ is solved by *MSMDSolver* while setting $t_{CTI,n}$ to infinity. Assigning the global optimal CTI usage time to this migration task achieves the highest charge migration efficiency η_{CM} . Increasing $t_{CTI,n}$ beyond $t_{CTI,n}^*$ is not helpful in improving η_{CM} . On the other hand, η_{CM} decreases progressively as $t_{CTI,n}$ goes below $t_{CTI,n}^*$. ■

Although it is possible to achieve the highest migration efficiency by using $t_{CTI,n}^*$, we typically have $\sum_{n=1}^N t_{CTI,n}^* > S \cdot T_D$ so that we cannot assign $t_{CTI,n}^*$ to each task directly. Here we define the set of CTI usage times that can achieve the optimal charge migration efficiency under the deadline constraint (ii) as *constrained optimal CTI usage time set*, denoted by $\{t_{CTI,1}^{opt}, t_{CTI,2}^{opt}, \dots, t_{CTI,N}^{opt}\}$.

Proposition 2: For an constrained optimal CTI usage time set $\{t_{CTI,1}^{opt}, t_{CTI,2}^{opt}, \dots, t_{CTI,N}^{opt}\}$, there must exist $t_{CTI,n}^{opt} \leq t_{CTI,n}^*, \forall n \in \{1, 2, \dots, N\}$.

Proof by contradiction: If there exists such an optimal usage time $t_{CTI,n}^{opt} > t_{CTI,n}^*$, we can always reduce it to $t_{CTI,n}^*$ without causing any charge migration efficiency degradation according to Proposition 1, and assign this slack time to other migration tasks to further improve the charge migration efficiency. Thus the new time set becomes the constrained optimal CTI usage time set. ■

Proposition 3: For $t < t_{CTI,n}^*, dE_{drawn,n}^{opt}(t_{CTI,n})/dt < 0$.

Proof: The migration current for CTI usage time $t < t_{CTI,n}^*$ is greater than that of $t_{CTI,n}^*$ since shorter CTI time causes larger charging/discharging currents. According to (2)~(5), the energy loss increases super-linearly as the migration current increases. Thus we have $dE_{drawn,n}^{opt}(t_{CTI,n})/dt < 0$. ■

According to Proposition 2 and 3, we are only interested in the left side of $t_{CTI,n}^*$, where $dE_{drawn,n}^{opt}(t_{CTI,n})/dt < 0$. We propose the following algorithm to determine CTI usage time:

Algorithm 1: CTI usage time reduction (UTR)

Inputs: migration task set $\{T_n\}$, $\{t_{CTI,n}, E_{drawn,n}^{opt}(t_{CTI,n})\}$, $n \in \{1, 2, \dots, N\}$, deadline T_D , number of CTIs S , time step length Δt .
if $(t_{CTI,n} > T_D, \forall n \in \{1, 2, \dots, N\})$ //solve constraint (ii.1) violation
| then $t_{CTI,n} \leftarrow T_D$;
while $(\sum_{n=1}^N t_{CTI,n} > S \cdot T_D)$ //solve constraint (ii.2) violation
| For all n , $t'_{CTI,n} \leftarrow t_{CTI,n} - \Delta t$;
| For all n , $E_{drawn,n}^{opt,temp}(t'_{CTI,n}) \leftarrow MSMDSolver(T_n, t'_{CTI,n})$;
| $k \leftarrow FindMinIndex(\{E_{drawn,n}^{opt,temp}(t'_{CTI,n}) - E_{drawn,n}^{opt}(t_{CTI,n})\})$;
| $t_{CTI,k} \leftarrow t'_{CTI,k}$, $E_{drawn,n}^{opt}(t_{CTI,n}) \leftarrow E_{drawn,n}^{opt,temp}(t'_{CTI,k})$;
Return: $t_{CTI,n}^{opt} \leftarrow t_{CTI,k}$, $E_{drawn,n}^{opt}(t_{CTI,n}^{opt}) \leftarrow E_{drawn,n}^{opt}(t_{CTI,n})$, $n \in \{1, 2, \dots, N\}$

Algorithm 1 is based on local neighborhood search. We use $E_{drawn,n}^{opt}$ as the cost function and start from the global optimal solution set $\{t_{CTI,n}^*\}$, $n \in \{1, 2, \dots, N\}$. If the global optimal solution set satisfies the deadline constraint (ii), we simply return it. Otherwise, we iteratively search all moves (defined as a reduction of CTI usage times) and identify the one that brings us the least amount of cost increase. The algorithm is terminated when we move to a solution set that satisfies the deadline constraint (ii). Furthermore, the gradient of $E_{drawn,n}^{opt}(t_{CTI,n})$ versus $t_{CTI,n}$ is always negative for $t_{CTI,n} < t_{CTI,n}^*$; therefore, the cost function does not have local minima in this region. The solution set when we exit our algorithm is the optimal solution set subject to the deadline constraint (ii).

The time complexity of this algorithm is $O(MN)$, where $M = (\sum_{n=1}^N t_{CTI,n} - S \cdot T_D) / \Delta t$ is the total number of moves. Using smaller step length Δt in Algorithm 1 can achieve better results at the expense of longer runtime. We use an adaptive time step (initial at hundreds of seconds then gradually reduce to tens of second) considering both the runtime and quality.

4.3 Merging Tasks

Merging tasks allows one to assign a longer CTI usage time for each migration task, which is helpful in lowering the energy loss due to the rate capacity effect while meeting the deadline constraint. However, we should not merge two tasks if the energy loss due to the non-optimal CTI voltage setting is greater than the energy saved. Therefore, we propose the following algorithm based on the optimal CTI voltage, denoted by $V_{CTI,n}^{opt,mean}$, of difference between tasks.

Algorithm 2: Tasks merging (TM)

Inputs: set of all migration tasks $\{T_n\}$, $\{t_{CTI,n}, E_{drawn,n}^{opt}(t_{CTI,n}), V_{CTI,n}^{opt}(t)\}$, $n \in \{1, 2, \dots, N\}$, number of CTI S , check set size K .
if $(N \geq S)$ //do not merge if number of tasks is no more than CTIs
| $V_{CTI,n}^{opt,mean} \leftarrow \int_0^{t_{CTI,n}} V_{CTI,n}^{opt}(t) dt / t_{CTI,n}$;
| for all $\binom{2}{N}$ pairs, find $\Delta V_{ij} \leftarrow |V_{CTI,i}^{opt,mean} - V_{CTI,j}^{opt,mean}|$;
| sort all $\{\Delta V_{ij}\}$ in ascending order;
| for the first K pairs in sorted list $\{\Delta V_{ij}\}$, do
| | $E_{drawn,ij}^{opt} \leftarrow MSMDSolver(T_i \cup T_j, t_{CTI,i} + t_{CTI,j})$;
| | $\Delta E_{ij} \leftarrow E_{drawn,i}^{opt}(t_{CTI,i}) + E_{drawn,j}^{opt}(t_{CTI,j}) - E_{drawn,ij}^{opt}$;
| pair $(k, l) \leftarrow FindMaxPair(\{\Delta E_{ij}\})$, among K pairs;
| if $(\Delta E_{kl} > 0)$
| | $T_k \leftarrow T_k \cup T_l$, $\{T'_n\} \leftarrow \{T_n\} / T_l$, $N' = N - 1$;
Return: set of all migration tasks $\{T'_n\}$, $n' \in \{1, 2, \dots, N - 1\}$.

We do not merge tasks if the initial number of tasks is equal to or less than the number of available CTIs. In this case, we simply assign one CTI to each migration task, which achieves the optimal migration efficiency since each task uses its optimal CTI voltage.

Otherwise, in Algorithm 2, we check the K pairs of migration tasks that have the closest optimal CTI voltage settings since checking all pairs are time-consuming and typically unnecessary, and merge the pair that brings us the largest improvement in terms of the migration efficiency. Experimental results show that generally merging the pair of tasks that have the closest optimal CTI voltage yields the highest improvement. Therefore, we set $K \leq 5$ in practice to avoid high complexity of the merging algorithm.

4.4 Scheduling Algorithm

We propose scheduling algorithm to find the scheduling of charge migration tasks in an iterative manner. In each iteration, we start from the global optimal CTI usage time set and use Algorithm 1 to reduce it until the deadline constraint is met. Then we use the constrained optimal CTI usage time set as the baseline and check whether we can improve the charge migration efficiency through tasks merging. If merging is performed, the number of migration tasks decrease by one and we start next iteration. We converge to a scheduling when deadline constraint is met and no further merging can improve the charge migration efficiency. The integrated algorithm is shown below:

Algorithm 3: Charge migration tasks scheduling (CMScheduling)

Inputs: initial migration tasks set $\{T_n\}$, $n \in \{1, 2, \dots, N\}$, deadline T_D , number of CTI S , time step length Δt , check set size K .
 $flag_{merge} \leftarrow true$;
while $(flag_{merge})$
| for each migration task n , do
| | $[t_{CTI,n}^{opt}, E_{drawn,n}^{opt}(t_{CTI,n}^*), V_{CTI,n}^{opt}(t)] \leftarrow MSMDSolver(T_n, \infty)$;
| | $\{t_{CTI,n}^{opt}, E_{drawn,n}^{opt}(t_{CTI,n}^{opt})\} \leftarrow$
| | $UTR(\{T_n, t_{CTI,n}^*, E_{drawn,n}^{opt}(t_{CTI,n}^*)\}, T_D, S, \Delta t)$;
| | $flag_{merge} \leftarrow false$;
| | $\{T'_n\} \leftarrow TM(\{T_n, t_{CTI,n}^{opt}, E_{drawn,n}^{opt}(t_{CTI,n}^{opt}), V_{CTI,n}^{opt}(t)\}, S, K)$;
| | if $(\{T'_n\} \neq \{T_n\})$ then $\{T_n\} \leftarrow \{T'_n\}$, $flag_{merge} \leftarrow true$;
for each migration task n' , do
| $[\eta_{MSMD}^{opt}(t_{CTI,n'}), E_{drawn,n'}^{opt}(t_{CTI,n'}), \{I_{array,n',i}^{dst,opt}(t)\}, V_{CTI,n'}^{opt}(t)]$
| $\leftarrow MSMDSolver(T_{n'}, t_{CTI,n'}^{opt})$;
Return: merged migration tasks set $\{T'_n\}$, $t_{CTI,n'}^{opt}$, $\{I_{array,n',i}^{dst,opt}(t)\}$, $V_{CTI,n'}^{opt}(t)$, $i \in \{1, 2, \dots, m_{n'}\}$, $t \in (0, t_{CTI,n'}^{opt})$, $n' \in \{1, 2, \dots, N'\}$.

The proposed scheduling algorithm returns a constrained optimal CTI usage time set that satisfies the deadline constraint (ii). We still need an arrangement with specified starting time, ending time and CTI ID number for each migration task. In practice, the energy overhead due to the CTI voltage regulation is small compared to the energy loss during migration. Therefore, different arrangements of the migration tasks on the CTIs do not affect the overall charge migration efficiency by much. We can randomly break migration tasks to fully utilize all the CTIs.

V. Experimental Results

5.1 Migration Tasks Derivation

We perform preemptive charge migrations from the EES banks with low charge/discharge efficiency but high energy capacity to the EES banks of high charge/discharge efficiency but low energy capacity before a high power demand load kicks in. For example, lead-acid battery banks typically have efficiency around 70% while supercapacitor banks have efficiency as high as 95% [1].

We take the typical load profiles of home kitchens in United State as an example and consider two cases: I without refrigerator (daily energy 4.11 kWh) and, II with refrigerator (daily energy usage 8.03 kWh) [6]. The average load profile of home kitchens has two peaks: one at 8am in the morning and the other at 7pm in the evening

and about 50% of the total kitchen energy is consumed during these two periods. In case I, we use a HEES system I with 20 EES banks. The total capacities of the HEES systems are designed to match the daily energy consumption of the home kitchen. Assuming enough energy is stored in low efficiency banks (charged from grid or photovoltaic array), we migrate 0.3 kWh of energy in six hours from low efficiency banks to high efficiency banks during off-peak hours. Thus roughly 30% ($0.3/(4.11/2 \times 50\%)$) of energy is provided by high efficiency banks during the peak hours, which makes the energy stored in HEES system more efficient since: 1) it is more energy efficient to use the high efficiency banks ($80\% \times 95\% > 70\%$, assuming that the charge migration efficiency is 80%) and, 2) less energy is wasted due to the rate capacity effect during the peak hours [5]. In case II, we use a 40-bank HEES system II and set the total target energy to be 0.6 kWh and total migration time to be six hours. We compare the charge migration efficiency of the proposed scheduling algorithm with some baseline scheduling schemes in both for both of these two cases.

5.2 Charge Migration Scheduling Results

The baseline setups basically follow two different scheduling schemes: A) merging based, i.e., merge all the initial migration tasks into S number of tasks and assign one task to each CTI, where the CTI usage time for each task is the global deadline T_D ; and B) clustering based, i.e., cluster all initial migration tasks into S number of groups, assign each group to one CTI and perform the migration tasks in a group one at a time while the CTI usage time is determined based on the ratio of target amount of energy transfers of all migration tasks in that group. During the migration process, the OCVs of the battery banks change only slightly while the OCVs of the supercapacitor banks vary by a large amount due to their small energy capacity and strong SoC-OCV dependency. Hence it is difficult for the baseline schemes to predict the OCV in following time instances. The baselines setups use constant CTI voltage during the entire migration process. For each baseline scheduling scheme, we try different merging methods: A.1 (merge tasks that have similar initial OCVs), A.2 (merge tasks so that the total target energy for each merged task is nearly the same) and A.3 (randomly pick the initial tasks and merge them, we did 100 runs and report the average efficiency) and clustering methods: B.1 (cluster tasks that have similar initial OCVs), B.2 (cluster tasks so that the total target energy of each group is nearly the same) and B.3 (randomly pick the initial tasks and cluster them, we did 100 runs and report the average efficiency.)

Figure 4 compares the charge migration efficiencies of the proposed scheduling algorithm and various baselines. According to the results in Figure 4, performance in charge migration efficiency of these two baseline scheduling schemes depend on the problem setup e.g., the number of migration tasks, amount of target energy to be delivered, and the deadline constraint. In addition, the performance fluctuates by about 5% when trying different merging and grouping combinations. As shown in Figure 4, the proposed algorithm for solving the CMS problem consistently outperforms the baseline systems with charge migration efficiency improvement ranging from 11.4% to 32.2%.

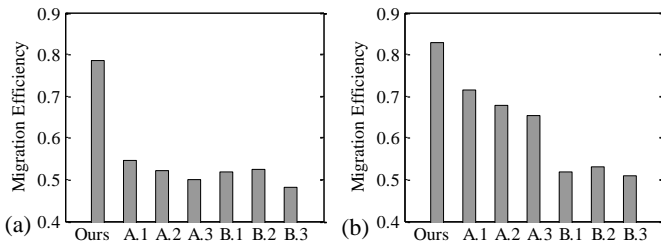


Figure 4: Charge migration efficiencies comparison in 20-bank HEES system (a) and 40-bank HEES system (b).

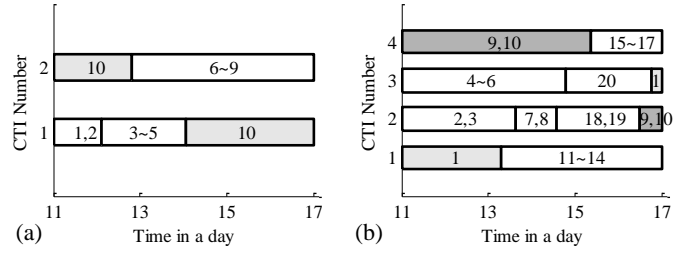


Figure 5: Our migration task scheduling solutions for a 20-bank HEES system (a) a 40-bank HEES system (b).

Figure 5 shows the scheduling of the migration tasks in these two HEES systems. The numbers inside the block denotes the initial migration task IDs. Multiple initial migration tasks have been merged when multiple numbers appear in one block, as shown in Figure 5. To maximize the overall efficiency and finish all migration tasks by the deadline, we sometimes break one migration task into some smaller subtasks, depicted in gray color in Figure 5, so that we can perform them separately and meet the global deadline.

VI. Conclusion

This is the first paper that introduces the charge migration scheduling (CMS) problem for hybrid electrical energy storage (HEES) systems. The scheduling objective is to maximize the overall charge migration efficiency of multiple, simultaneous migration tasks through a fixed number of CTIs while meeting the deadline constraint. We developed three sub-algorithms to: i) determine the optimal CTI voltage and charging current profiles for each migration task, ii) reduce the CTI usage time to satisfy the deadline constraint, and iii) merge the most appropriate task pairs to further improve the migration efficiency. We integrated these three sub-algorithms together and solved for scheduling that achieves high migration efficiency. Experimental results demonstrate significant charge migration efficiency improvements in the example HEES systems.

References

- [1] M. Pedram, N. Chang, Y. Kim, and Y. Wang, "Hybrid electrical energy storage systems," in *ISLPED*, 2010.
- [2] F. Koushanfar, "Hierarchical hybrid power supply networks," in *DAC'10*.
- [3] Q. Xie, Y. Wang, Y. Kim, N. Chang and M. Pedram, "Charge Allocation for Hybrid Electrical Energy Storage Systems," in *CODES+ISSS*, 2011.
- [4] Wang, Y. Kim, Q. Xie, N. Chang, and M. Pedram, "Charge migration efficiency optimization in hybrid electrical energy storage (HEES) systems," in *ISLPED*, 2011.
- [5] Q. Xie, Y. Wang, M. Pedram, Y. Kim, D. Shin, and N. Chang, "Charge replacement in hybrid electrical energy storage systems," in *ASPDAC* 2012.
- [6] R. Hendron, C. Engebrecht, "Building America House Simulation Protocols", <http://www.nrel.gov/docs/fy11osti/49246.pdf>.
- [7] Wang, Q. Xie, M. Pedram, Y. Kim, N. Chang, and M. Poncino, "Multiple-source and multiple-destination charge migration in hybrid electrical energy storage systems," in *DATE*, 2012.
- [8] F. Yao, A. Demers, and S. Shenker, "A Scheduling Model for Reduced CPU Voltage", in *Foundation of Computer Science*, 1995.
- [9] T. Ishihara, H. Yasuura, "Voltage Scheduling Problem for Dynamically Variable Voltage Processors", in *ISLPED*, 1998.
- [10] M. Chen and G. Rincon-Mora, "Accurate electrical battery model capable of predicting runtime and I-V performance," in *TEC*, 2006.
- [11] D. Linden and T. B. Reddy, *Handbook of Batteries*. McGraw-Hill Professional, 2001.
- [12] Y. Choi, N. Chang and T. Kim, "DC-DC converter-aware power management for low-power embedded systems," in *TCAD*, 2007.