Smart Grid 智能电网, 2013, 3, 59-67
http://dx.doi.org/10.12677/sg.2013.33011 Published Online June 2013 (http://www.hanspub.org/journal/sg.html)

Design and Implementation of a Battery Management Emulation System*

Miao Wang[1], Junwei Cao[1,2], Chao Wei[1], Chao Lu[3]

[1]Research Institute of Information Technology, Tsinghua University, Beijing
[2]Tsinghua National Laboratory for Information Science and Technology, Beijing
[3]Department of Electrical Engineering, Tsinghua University, Beijing
Email: jcao@tsinghua.edu.cn

Received: Apr. 6th, 2013; revised: Apr. 28th, 2013; accepted: May 7th, 2013

Abstract: Utilization of computer and internet technology for battery management systems is proposed in this paper, which is traditionally based on analog circuit and microcontroller. This brings greater security and longer life for the battery system and reduces hardware complexity and cost. This paper discusses design and implementation process of the battery management system which emulates an energy storage system that consists of a number of battery cells using MATLAB Simulink. Monitoring and controlling the emulated battery system is implemented by software. The system provides web-based user interfaces to control emulated battery and visualizes the change of cell parameters in a friendly way. Experiments are carried out to verify system functions and to explore how to decide computer control period.

Keywords: Battery Management; Simulink Emulation; Computer Control System; Ajax Technology

储能电池管理系统仿真平台的设计与实现*

王　淼[1]，曹军威[1,2]，魏　超[1]，陆　超[3]

[1]清华大学信息技术研究院，北京
[2]清华信息科学与技术国家实验室，北京
[3]清华大学电机工程与应用电子技术系，北京
Email: jcao@tsinghua.edu.cn

收稿日期：2013 年 4 月 6 日；修回日期：2013 年 4 月 28 日；录用日期：2013 年 5 月 7 日

摘　要：本文提出了将计算机和互联网技术引入传统的基于模拟电路和微控制器的电池管理系统的想法，以此为电池系统带来更高的安全性和更长的寿命，降低电池端硬件的复杂度和成本。本文讨论了电池管理系统的详细设计和实现过程。系统通过 MATLAB 仿真工具 Simulink 对由多个电池单元组成的储能系统进行仿真，并通过计算机软件实现对仿真电池系统的监视与控制。在此基础上实现了在具体应用场景中通过 Web 页面对仿真电池系统进行控制，并通过 Web 页面以友好的方式展示控制过程中电池参数的变化。最后通过实验对本系统的功能进行了验证，证明了控制逻辑的正确性，并探讨了计算机软件控制周期的选择问题。

关键词：电池管理；Simulink 仿真；计算机控制系统；Ajax 技术

1. 引言

随着国家加大对下一代智能电网建设的投入，相关技术的研究也逐渐成为热门。其中储能技术，特别是电池储能技术因为具有储存不稳定可再生能源(如风能，太阳能)所产生的电能的能力而受到重视[1]。但是，电池储能系统的运作需要对电池进行管理来保证

*基金项目：国家 973 基础研究计划(2011CB302805 和 2013CB228206)；国家自然科学基金(61233016 和 51037002)。

电池的安全和寿命。

经过近些年的发展，市场上已经有了一些比较成熟的电池管理产品和解决方案。如美国 Metric Mind 公司开发的 Smooth Talk 电动汽车电池管理系统、英国 REAP systems 公司开发的 REAP 系列产品、美国 Manzanita Micro 公司开发的 Mk3 Digital Regulator 系统等。这些产品的共性是多采用纯模拟电路或者简单的数字控制器进行控制，极少引入计算机的控制[2]，因此控制逻辑大多比较简单，不具有应用高级控制逻辑的能力。而且，这些系统很少有能够与计算机交互的接口。

本文提出的系统强调将计算机和互联网技术融入到电池管理系统的设计当中，由计算机完成复杂的控制逻辑运算，为电池系统带来更高的安全性和更长的寿命，而且可以降低电池端硬件的复杂度，降低成本。此外，由于引入了互联网，该系统能够支持通过 Web 访问的方式远程控制储能电站，提高了电池管理系统的使用方便性。系统通过 MATLAB 仿真工具 Simulink 对多个电池单元组成的储能系统进行仿真，并通过计算机软件实现对仿真电池系统的监视与控制。在此基础上利用最新的 Web 技术实现了风储互补和电池平衡两个主要应用的仿真和演示。通过设计和运行实验验证了系统控制逻辑的正确性和可用性，并进一步探讨了计算机软件控制周期的选择问题。

2. 储能电池管理系统设计

本文中的储能电池管理系统是一个 CPS (Cyber-Physical Systems[3-5])的典型应用，该系统通过软件模拟由数量达到一定规模的电池单元串联组成的储能系统，实现对电池细粒度、高实时性的充放电控制，并且能够对电池平衡以及风储互补这两种具体的典型应用提供支持。在系统中引入储能电池系统后，由于电池单元在参数上的差异会随着使用逐渐放大，从而严重影响整个储能系统的循环使用寿命，所以需要对电池进行管理来保证电池的一致性[6,7]。本文的电池管理系统提供了主动使得电池的充电程度(State of Charge，简称 SOC)保持一致的控制逻辑，在下文中将此应用成为电池平衡。系统体系结构如图 1 所示。

系统被分为了 MATLAB 仿真模块，核心程序集，MySQL 数据库，WEB 管理演示平台四个模块。MATLAB 仿真模块主要利用 MATLAB 所包含的基于

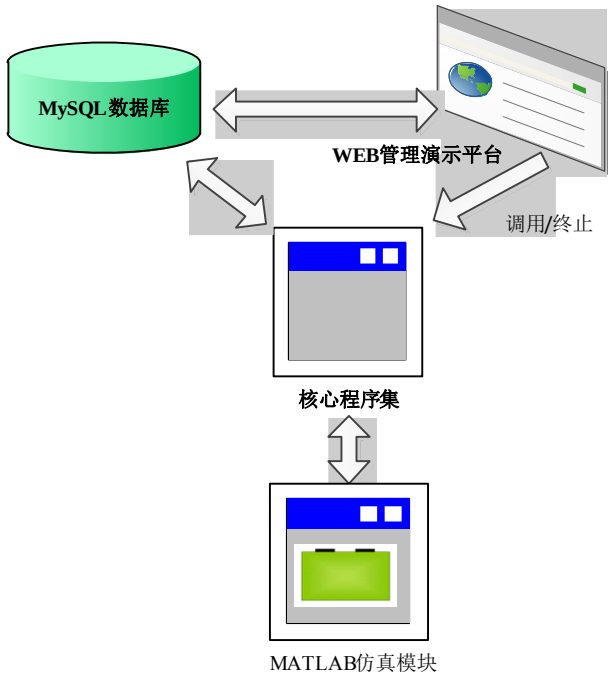

Figure 1. Architecture diagram of the system
图 1. 系统体系结构图

建模的仿真工具 Simulink 中附带的电池模型模拟真实的储能电池系统，能够输出电池的状态参数，并可接受输入，根据输入改变状态参数；核心程序集负责电池管理系统的核心计算，该模块包含三个独立的程序，分别实现数据获取、更新和进程监控，风储互补应用中的相关控制和风电出力－负载曲线模拟以及电池平衡应用中的相关控制；MySQL 数据库用于存储实时更新的电池参数、用户信息，以及核心程序集的状态参数和辅助数据；Web 管理演示平台用于为使用者提供一个友好的电池管理系统入口，不同角色的用户可以方便实时地通过此模块了解系统的运行状态，并发出相应的指令。系统管理员可以通过此平台进行主要应用的演示。各个模块处理独立的功能，在协商好接口设计之后单独实现，互不干扰。下面详细描述各个模块的问题和相关设计。

2.1. MATLAB 仿真模块的设计

2.1.1. 电池仿真的设计

电池仿真通过 Simulink 提供的电池模型来完成，该模型解决了电池充放电特性模拟的问题。考虑到演示系统的规模，将电池单元的数量设定为 10。在该系统中电池模型的主要参数设定如表 1 所示。

Table 1. Main parameters of Simulink battery model
表 1. Simulink 电池模型主要参数

参数	单位	参数值
Battery Type		Lead-Acid
Nominal Voltage	V	2
Rated Capacity	Ah	7
Initial State-of-Charge	%	60
Maximum Capacity	Ah	7.2917
Fully Charged Voltage	V	2.0776
Nominal Discharge Current	A	1.4
Internal Resistance	Ohms	0.0028571
Capacity @ Nominal Voltage	Ah	2.1719
Exponential Zone [Voltage, Capacity]	V, Ah	[2.0362, 0.023333]

2.1.2. 外部接口的设计

基于简化接口的设计原则，MATLAB 仿真模块以纯文本的方式进行输入和输出。其中输入文件被定义为包含 10 个整数的数组，用于控制电池的充放电，其中每个数字对应一个电池单元，数字的具体含义如表 2 所示。

当所有控制位均为 3 时，将由 MATLAB 仿真模块进行电池平衡控制。与之对应的是软件平衡，由核心程序进行细粒度的充放电控制来达到电池平衡的目的。如此设计仿真模块的输入实现了对硬件平衡和软件平衡的同时支持。输出文件被定义为 3 × 10 的浮点数组，用于传递电池的参数，包括电池单元的 SOC、电压和电流。

2.2. 核心程序集的设计

核心程序集包括三个主要模块，一是把 MATLAB 仿真模块产生的实时数据采集起来并送入数据库中的 UPDATE 模块，还有用来完成风储互补和电池平衡两个应用中的控制工作的 WIND 和 EQUALIZATION 模块。

实现 UPDATE 程序时，应选取合适的数据采样的时间间隔 T_u，T_u 与产生电池参数的时间间隔 T_m 之间应满足 $T_u >> T_m$ 的关系，从而保证数据的更新能够满足上层应用的需要，又不至于耗费大量的计算能力。本文在后面的实验环节探讨如何选取最佳 T_u，这里不再赘述。同时为了方便地进行实验，应当支持自定义 T_u。

Table 2. The control commands of MATLAB simulation module
表 2. MATLAB 仿真模块控制命令定义

命令	别名	含义
0	IDLE	空闲(停止充放电)
1	DISCHARGE	放电
2	CHARGE	充电
3	EQUALIZATION	硬件平衡

WIND 程序为风储互补应用提供计算支持。WIND 程序的功能之一是在较短的时间内模拟一天 24 小时风力发电场的出力与负载状况。MySQL 数据库预先存储了 100 组来自真实风电场在某个 24 小时内的出力－负载数据，WIND 程序将以自定义的时间间隔(设为 T_w，默认为 1 秒)从数据库中获取一组新的出力－负载数据，这样就能实现在 $T_w \times 100$ 的时间内模拟了风电场 24 小时的出力与负载状况。

在每次获取一组新的出力－负载数据之后，WIND 程序需要对当前状态做出判断，然后对 MATLAB 仿真模块输出控制命令。

EQUALIZATION 程序为风储互补应用提供计算支持，功能比较单一，只需对电池进行平衡控制。EQUALIZATION 程序需要实现硬件平衡和软件平衡两种控制模式，并通过命令行参数的形式确定采用哪种模式。此外，EQUALIZATION 程序也采用固定时间间隔的控制模式，设时间间隔为 T_e。由于其控制命令是根据数据库中的数据得出的，而数据库中的数据更新周期为 UPDATE 程序的更新周期，即 T_u。如果取 $T_e < T_u$，则会出现多组控制命令是根据同一组数据得出的，没有意义；如果取 $T_e > T_u$，则会造成某些组的数据被略过，也就是说 UPDATE 程序浪费了计算资源。所以，在实践中，应当取 $T_e = T_u$。

2.3. MySQL 数据库设计

数据库模块的存在是为核心程序集和 Web 管理演示平台提供一个数据的“桥梁”。在数据库中我们分别设计了以下几张表：Battery 数据表用于存储电池组的状态参数；Wind 数据表用于存储 100 组出力–负载参数；User 数据表用于存储用户信息，辅助 WEB 管理演示平台实现简单的用户管理功能；还有两个用于存储核心程序集运行状态的数据表，分别为 equal_

status 和 wind_time。这两个数据表都只包含一个字段，存储一个单一的值。其中，equal_status 只存储 1 条记录，用来表示 EQUALIZATION 程序正在运行中或者尚未运行；wind_time 也只存储 1 条记录，用来记录 WIND 程序在上一次运行结束时执行到了哪一条出力－负载状态。

2.4. Web 管理演示平台

Web 管理演示平台是整个系统的最前端，在实现功能的同时还要考虑界面用户体验的设计。图 2 中展示了管理员权限的用户所能使用的功能项。导航栏的下方根据具体功能需要，以及当前应用的运行状态，动态生成各功能页面的菜单。数据图表统一使用开源项目 Open Flash Chart 的柱状图和折线图，尺寸根据页面需要确定。为了加强风储互补和电池平衡两个应用的演示效果，在页面中加入动态示意图。当应用尚未启动时，示意图处于静止状态。当应用处于运行状态时，示意图处于动态的状态，并能根据系统状态发生变化。

3. 储能电池管理系统的实现

3.1. MATLAB 仿真模块的实现

MATLAB 仿真模块实现电池状态的实时仿真，能够接受控制命令并发送电池单元的状态。该模块的输入与输出统一由一个自定义的 S-function 处理，电池状态参数由基于模型设计的电池单元组与附加电路进行计算。仿真模块的整体运行流程如图 3 所示。

上述流程中最核心的部分是控制电池的部分。该部分的计算完全由电池单元与充放电控制模型进行，共有 10 个这样的模型，各模型彼此相同。该模型是通过 Simulink 的模型设计工具实现的，如图 4 所示。

3.2. 核心程序集的实现

核心程序集用来实现对 MATLAB 仿真模块的监视与控制。在实现核心程序的过程中，为了使程序的

Figure 2. The navigation bar: WEB management platform
图 2. WEB 管理演示平台导航栏设计

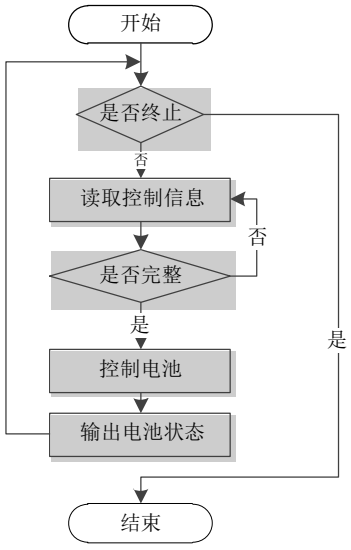

Figure 3. The simulative process of MATLAB
图 3. MATLAB 仿真的运行流程

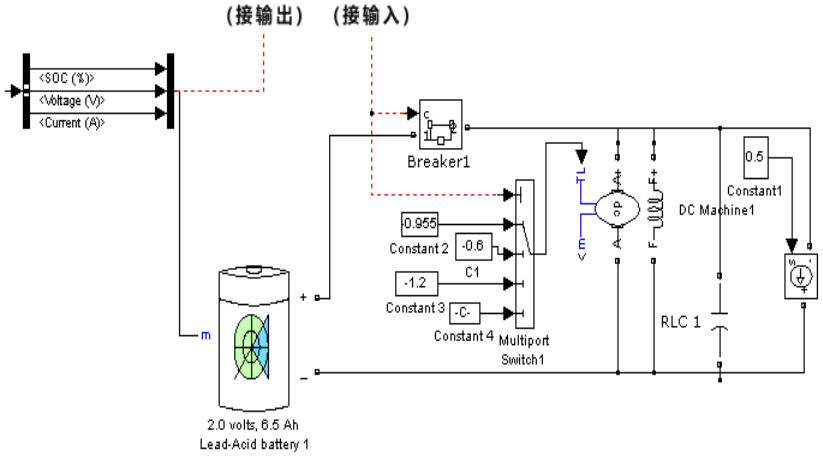

Figure 4. The battery sells and the model of charge-discharge
图 4. 电池单元与充放电控制模型

结构更清晰，对一些公共的对象进行了封装[8]。下面分别描述一下各个子程序的实现。

3.2.1. UPDATE 程序的实现

UPDATE 程序实现了将 MATLAB 仿真模块产生的参数更新到 MySQL 数据库中，并监控 WIND 和 EQUALIZATION 的运行状态，确保系统在没有应用运行时对仿真模块发出 IDLE(空闲)命令。UPDATE 程序的流程如图 5 所示。由于 UPDATE 需要在后台一直运行，所以被设计成死循环，当通过停止脚本杀死 UPDATE 程序的进程时，UPDATE 程序才能结束。

3.2.2. WIND 程序的实现

WIND 程序实现了按照时间序列读取出力–负载状态，然后根据状态对 MATLAB 仿真模块发出控制命令。WIND 程序的流程如图 6 所示。

3.2.3. EQUALIZATION 程序的实现

EQUALIZATION 程序实现了根据电池各个单元的当前状态，对 MATLAB 仿真模块发出平衡控制命令，分硬件平衡和软件平衡两种。EQUALIZATION 程序的流程如图 7 所示。

getControlStr()方法遍历当前所有电池单元的 SOC 参数，求得平均值，并按照表 3 为每一个电池单

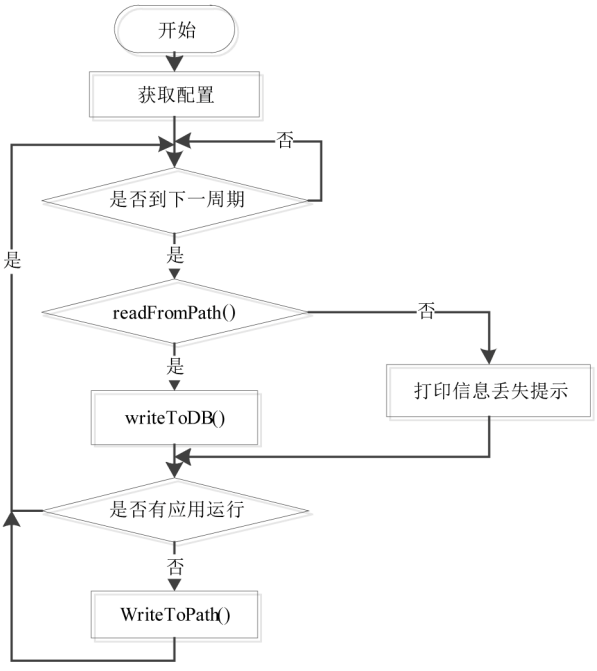

Figure 5. The process of UPDATE
图 5. UPDATE 程序流程

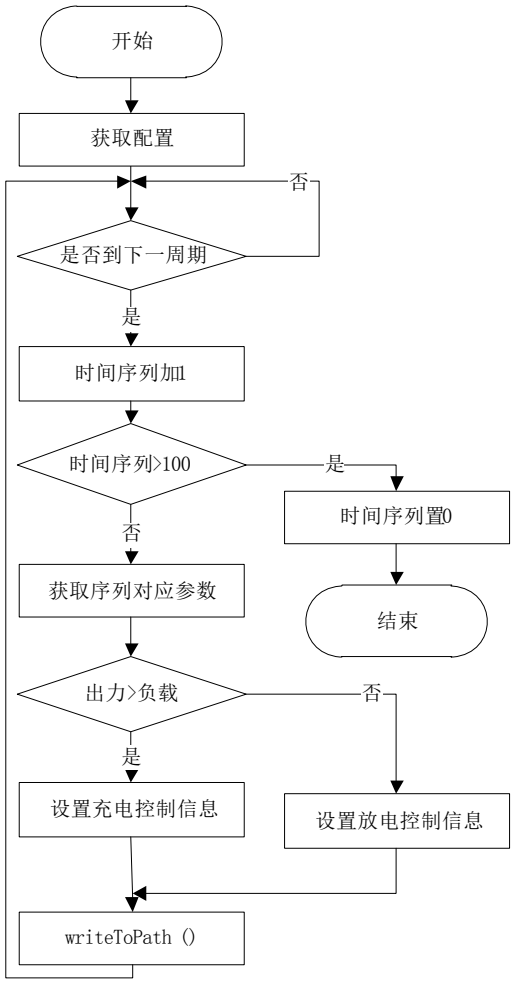

Figure 6. The process of WIND
图 6. WIND 程序流程

Table 3. The decision table of getControlStr()
表 3. getControlStr()方法的决策表

条件	命令
$SOC_i < \overline{SOC} - \frac{\mathrm{MAX}_{diff}}{2}$	CHARGE(充电)
$\overline{SOC} - \frac{\mathrm{MAX}_{diff}}{2} \le SOC_i \le \overline{SOC} + \frac{\mathrm{MAX}_{diff}}{2}$	IDEL(空闲)
$SOC_i > \overline{SOC} + \frac{\mathrm{MAX}_{diff}}{2}$	DISCHARGE(放电)

元设置控制信息。其中 SOC_i 表示第 i 个电池单元的 SOC 参数，$\overline{SOC}$ 表示当前所有电池单元 SOC 的平均值，MAX_{diff} 表示最大平衡差异。

isEqualizationRequired()方法遍历当前所有电池单元的 SOC 参数，求得最大值与最小值，然后计算最大值减去最小值，并与最大平衡差异(所允许的最大电池 SOC 差异)比较，如果前者小于后者，则认为电

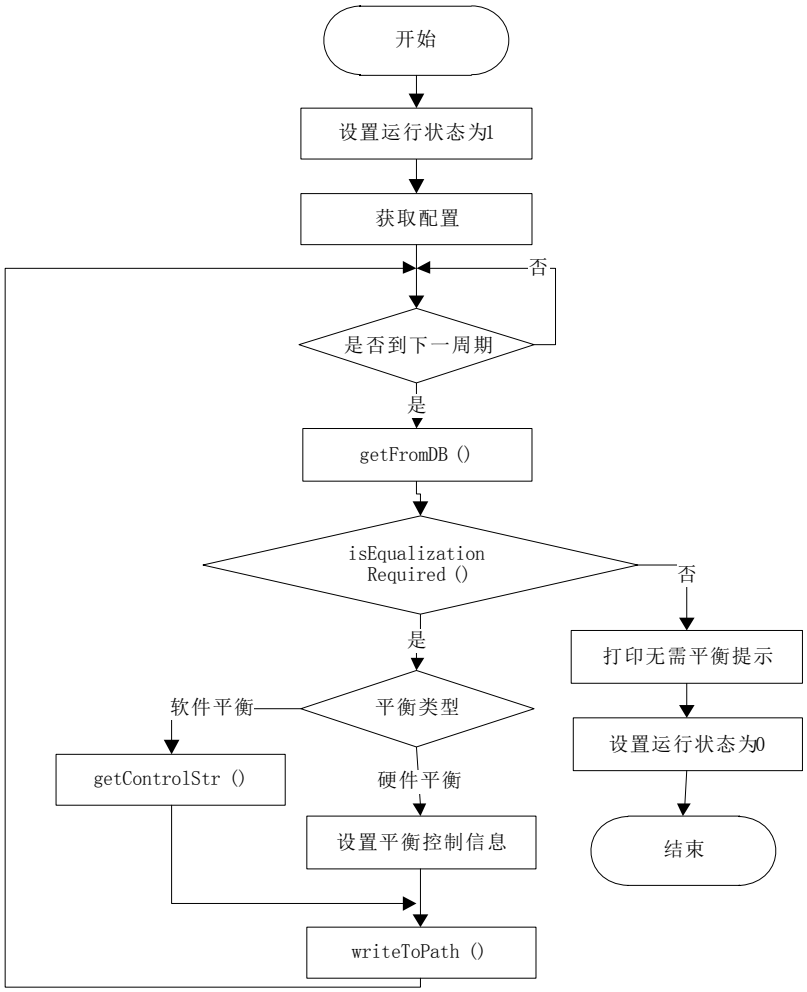

Figure 7. The process of EQUALIZATION
图 7. EQUALIZATION 程序流程

池已经达到平衡状态，返回 false，表示不需要进行电池平衡；否则认为电池尚未达到平衡状态，返回 true，表示仍需要电池平衡。

3.3. WEB 管理演示平台的实现

3.3.1. 电池状态页面的实现

在电池状态页面当中，实现了对当前电池状态参数的动态更新显示。页面将以 1 秒为时间周期，对页面中的 Open Flash Chart[9]中的数据进行更新。此外，用户可以在电压，电流，SOC，电压和 SOC 几种参数组合中选择一种显示，如图 8 所示。

为了实现该页面的效果，首先需要通过 Open Flash Chart 正确显示电池参数，其次是通过 Ajax[10]对图表进行异步刷新，最后通过 JavaScript 的 setInterval 方法实现自动异步刷新。本系统统一使用 data.php[11]

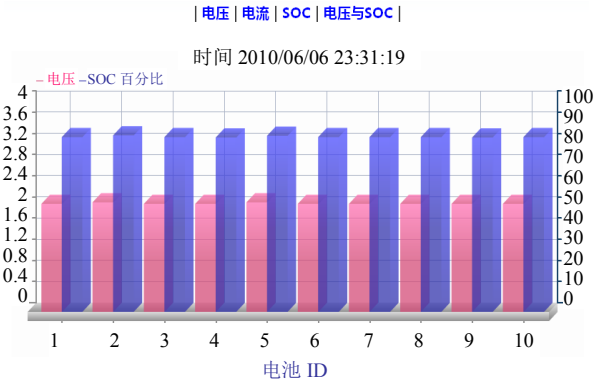

Figure 8. The page layout of battery status
图 8. 电池状态页面效果图

作为所有图表的数据来源。

3.3.2. 电池平衡页面的实现

由动态菜单、电池 SOC 参数图表和动态示意图

[12,13]组成，其实现如图 9 所示。

3.3.3. 风储互补页面的实现

风储互补页面与电池平衡页面类似，由动态菜单，动态示意图，电池 SOC 参数图表，以及出力－负载曲线图组成。其中，前三部分的实现与电池平衡页面中的对应部分的实现类似。出力－负载曲线图能够根据时间变化不断生长，系统默认时间间隔为 1 秒(代表 0.24 小时)，图 10 是系统运行到第 92 秒时的出力－负载曲线。

4. 系统实验分析

本文提出的储能电池管理系统的设计目的是探索将计算机和互联网技术引入电池管理系统的可行性。我们将通过实验对系统各模块能否实现统一功能进行验证，并在此基础之上，通过实验探讨软件离散化的控制能否与模拟电路连续的控制达到相同的效果，以及为了达到相同的效果需要将系统控制周期设定在什么范围内。

4.1. 功能验证实验

实验的目的在于验证风储互补和电池平衡两个应用是否能够按照拟定的逻辑对电池进行控制。其中，对风储互补的验证实验采用的方案是：完整运行 100 秒的风储互补应用，采集 10 个电池单元在每一秒的 SOC 参数平均值，绘制成折线图，然后与出力－负载曲线进行比较，考察系统是否按照出力－负载状况成功地对电池进行控制。

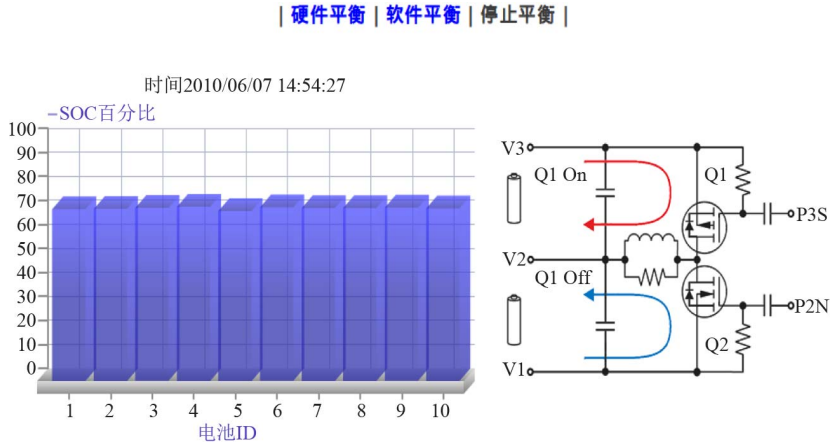

Figure 9. The page layout of battery balance
图 9. 电池平衡页面效果图

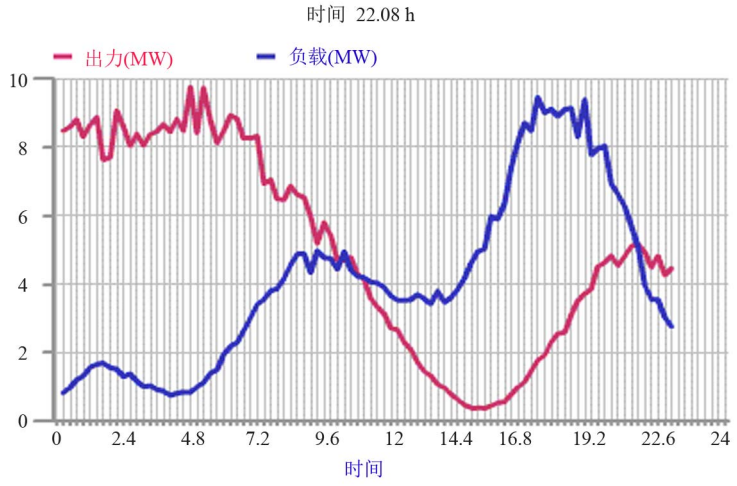

Figure 10. Curve: output-load
图 10. 出力－负载曲线效果图

对电池平衡的验证实验采用的方案是：预先设定 10 个电池单元的 SOC 参数，使各电池的 SOC 参数具有较大差异。设定系统控制周期 $T_u(T_e)$为 200 毫秒，最大平衡差异 MAX_{diff}为 1%。然后以软件平衡的方式运行电池平衡应用，记录每一秒的 SOC 参数的最大值与最小值的差值，直到系统达到平衡状态。最后将差值和 MAX_{diff}绘制成折线图进行比较，考察差值是否在逐渐向 MAX_{diff}靠近。

风储互补实验结果如图 11 所示。

从图中可以看出，当出力大于负载时，电池各单元充电；当出力小于负载时，电池各单元充电放电。也就是说电池单元的 SOC 参数的平均值是随着出力－负载状况变化的，这就说明了 WIND 程序成功地对电池进行了控制。有一点值得注意，在应用开始阶段和出力曲线和负载曲线交汇的地方，电池单元平均 SOC 的变化会滞后 3~4 个周期，经过分析，这种滞后现象的产生并非由于控制命令不能及时产生或是及时传达到 MATLAB 仿真模块，而是因为仿真模块进行充放电转换时需要 3~4 个周期的时间，而这与真实的储能电池系统的情况是一致的。

电池平衡实验结果如图 12 所示。

电池平衡的执行过程持续了 94 个周期，也就是 18.8 秒。从上图可以看出，电池单元间 SOC 的最大差异从大约 20%逐渐向 MAX_{diff}靠近，最终成功地达到了平衡条件。该实验说明了当系统控制周期 T_u 取 200 毫秒，最大平衡差异 MAX_{diff}取 1%时，软件控制的频率是符合要求的。但是是否是最优的，将在下面探讨。

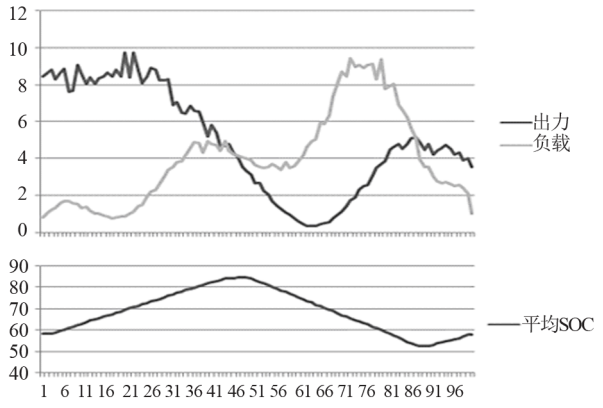

Figure 11. The results of the wind power and energy storage complementarity
图 11. 风储互补实验结果

4.2. 控制周期实验

我们在前面曾讨论了控制周期的选取问题，如果取较小的控制周期，则会带来更高的控制频率，控制效果也更好。相反，如果取较大的控制周期，则会降低对计算机硬件的要求，但是控制效果将会变差，甚至无法及时对系统的变化做出反应。

为了量化控制效果，在实验中引入了对照组实现——硬件平衡。所谓硬件平衡，即在 MATLAB 仿真模块中实现平衡控制逻辑，这样，控制频率与 MATLAB 的仿真频率是一致的，因此可以认为硬件平衡是实时的。将在同样初始条件下的硬件平衡时间设为 t_h，软件平衡时间设为 t_s，如果 $t_s \le 1.3 \times t_h$，则认为软件控制的效果是可以接受的。

在实验中，将从 $T_u = 200$ ms 开始进行测试，然后逐级增加 T_u，每次记录 t_s，直到 $t_s > 1.3 \times t_h$ 为止。测试的结果如图 13 所示。

从上面的折线图中可以看出，硬件平衡始终保持稳定的时间。对于软件平衡，当 $T_u \le 1200$ ms 时，软件平衡所需要的时间与硬件平衡时间保持一致。而当 $T_u \ge 1400$ ms 时，平衡所需要的时间剧烈增大，迅速

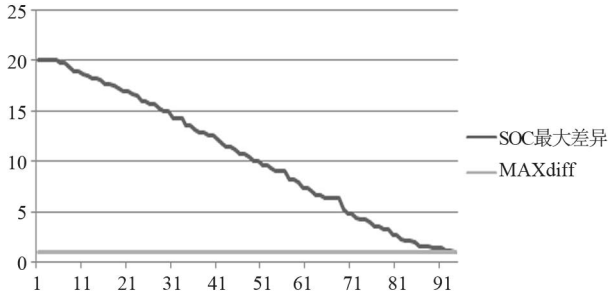

Figure 12. The results of battery balance
图 12. 电池平衡实验结果

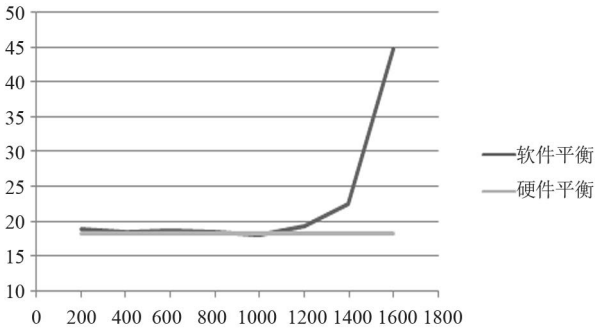

Figure 13. The results of control cycle
图 13. 控制周期测试结果

破坏了平衡效果可以令人接受的条件。因此，对于本系统当前的设定来说，将系统控制周期设定为 1200 毫秒是综合考虑控制效果和经济性的最佳选择。补充一组对 T_u = 2000 ms 的情况所做的实验，达到平衡条件所需要的时间如表 4 所示，从中可以看出，达到平衡所需要的时间呈高度的随机性，甚至出现了超过 200 秒仍未达到平衡被手动终止的情况(在表中用 N/A 表示)。

取其中第 2 次的数据，绘制成形如图 14 所示的 SOC 最大差异与 MAX_{diff} 的对照图。

从图 14 中可以看出，SOC 最大差异始终在 MAX_{diff} 上方波动，直到 SOC 最大差异“很幸运”地落入 MAX_{diff} 下方才能结束平衡控制。产生这种情况的原因是控制周期过长，在当前周期应当充电的电池单元在下一个周期开始的时候其 SOC 早已超过平均值；同理，在当前周期应当放电的电池单元在下一个周期开始的时候其 SOC 已经大大低于平均值，所以就造成了图 14 中的情况。解决这个问题的方法是减小控制周期或者提高 MAX_{diff}(放松收敛条件)。前者需要更高的计算机性能，后者需要牺牲控制的精确度。

5. 结论

本文作为一种将计算机和互联网技术引入传统的基于模拟电路和微控制器的电池管理系统的探索和尝试，提出了储能管理系统演示与仿真平台。在分析现有电池管理系统的共性的不足和限制基础之上，结合储能系统的具体需求，提出了由计算机对电池单元进行细粒度的监视与控制的电池管理系统，并将给予 web 的管理演示平台引入到系统的设计中，提高了使用系统的方便性。利用 MATLAB 仿真工具 Simulink 的模型库，设计并实现了具有统一输入和输出端口的电池储能系统仿真模块，能够实时地输出电池单元的电压、电流、SOC 参数，并能根据输入对电池单元进行独立的充放电控制。实现了风储互补和电池平衡两个主要应用的仿真与演示，系统各模块能协调统一地实现系统整体的功能。最后通过实验验证了系统控制逻辑的正确性和可用性。在此基础上，探讨了计算机控制系统中如何设计控制周期的问题，通过设计实验成功获得了本系统的最佳控制周期。

未来将进一步研究如何将计算与物理过程更加紧密地结合在一起，以提高物理系统的安全与性能，解决传统物理系统在应用上的一些障碍。

Table 4. The balance time of T_u = 2000 ms
表 4. T_u = 2000 ms 时平衡所需时间

次序	1	2	3	4	5	6	7	8
平衡时间(秒)	20	74	56	48	N/A	102	N/A	24

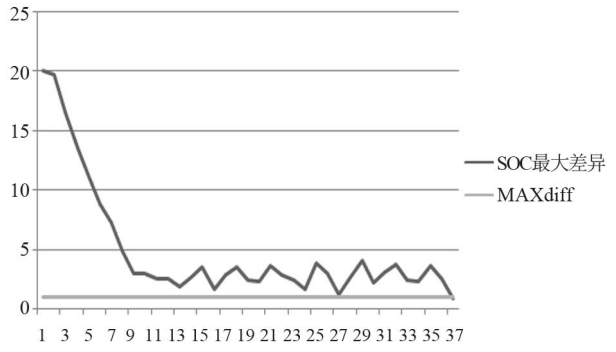

Figure 14. The second experiment results of T_u = 2000 ms
图 14. T_u = 2000 ms 第 2 次实验数据

参考文献 (References)

[1] 陈树勇, 宋书芳, 李兰欣等. 智能电网技术综述[J]. 电网技术, 2009, 33(8): 1-7.
[2] 互动百科. 计算机控制系统[URL]. http://www.hudong.com/wiki/计算机控制系统
[3] Y. Tan, S. Goddard and L. C. Pérez. A prototype architecture for cyber-physical systems. ACM SIGBED Review, 2008, 5(1): 1-2.
[4] E. A. Lee. Cyber-physical systems—Are computing foundations adequate? In: Position Paper for NSF Workshop on Cyber-Physical Systems: Research Motivation, Techniques and Roadmap, 16-17 October 2006, Austin: 9.
[5] 何积丰. Cyber-physical systems [J]. 中国计算机学会通讯, 2010, 6(1): 25-29.
[6] 张文亮, 丘明, 来小康. 储能技术在电力系统中的应用[J]. 电网技术, 2008, 32(7): 1-9.
[7] 吴东兴. 电池管理系统的设计与实现[D]. 湖南大学, 2006.
[8] S. B. Lippman, J. Lajoie, E. MooBarbara, 著, 李师贤, 蒋爱军, 梅晓勇, 林瑛, 译. C++ Primer (第 4 版)[M]. 北京: 人民邮电出版社, 2006.
[9] Open flash chart [URL]. http://teethgrinder.co.uk/open-flash-chart/
[10] J. J. Garrett. Ajax: A new approach to web applications [URL], 2005. http://www.adaptivepath.com/ideas/essays/archives/000385.php
[11] M. Achour, F. Betz, et al. PHP manual [URL], 2013. http://php.net/manual/en/index.php
[12] Available BMS systems [URL]. http://sourceforge.net/apps/mediawiki/tumanako/index.php?title=EVD5_BMS
[13] CellBalanceVisualizer [URL]. http://liionbms.com/balance/index.html