

安时-卡尔曼交叉运行的电池荷电状态估算策略及其微控制器在环验证^①

罗 勇^② 阚英哲^③ 祝传美 赵小帅 祁朋伟 龙克俊

(重庆理工大学汽车零部件先进制造技术教育部重点实验室 重庆 400054)

摘要 为了提高电池管理系统(BMS)的性能,研究了电池荷电状态(SOC)的估算方法,并根据SOC估算算法精度和系统实时性要求,提出了安时(AH)积分算法-卡尔曼(Kalman)算法(AH-Kalman)交叉运行的SOC估算策略。该策略用开路电压(OCV)法确定SOC初值,以实时性较强的AH积分法为主,采用间歇运行的Kalman滤波法修正安时计量法积分误差。建立了系统仿真模型,验证了卡尔曼滤波算法对安时积分法积累误差的修正作用。将控制算法生成C代码下载到目标控制器,搭建微控制器在环测试验证(PILS)平台,进行了与传统卡尔曼滤波算法的复杂度对比分析。结果表明,所提出AH-Kalman交叉运行的SOC估算策略在保证了SOC估算精度的同时也具有较好的实时性,便于实际应用。

关键词 电池荷电状态(SOC)估算算法,微控制器在环测试,C代码生成,卡尔曼滤波法

0 引言

电池管理系统(battery management system,BMS)是混合动力和电动汽车核心关键技术之一^[1],其性能对电池寿命及能量利用率具有重要影响。电池管理系统具有很强的实时性要求^[2],对某些中断请求响应不及时将可能危及整车安全,因此电池管理系统内部各种算法应尽量减少CPU占用。

电池荷电状态(state of charge,SOC)估算BMS最基础和核心的功能^[3-5],对SOC估算算法的研究在BMS开发中占据重要地位。传统的SOC估算算法如开路电压(open circuit voltage,OCV)法、电荷积分法等具有较强的实时性,但存在估算精度不高、易受外界因素影响等不足。近年来,卡尔曼(Kalman)滤波、神经网络等智能算法被用于SOC估算,这些算法大大提高了SOC估算精度,但也增加

了微控制器运算负担,系统实时性受到很大影响。开发一种既能获得较高估算精度,又具有较强实时性的SOC估算算法对提高BMS性能具有重要意义。

本文综合考虑了SOC估算算法的精度和实时性,提出了用开路电压法(OCV)确定SOC初值,以实时性较强的安时(Ampere hour,AH)积分法为主,采用间歇运行的卡尔曼(Kalman)滤波法修正积分误差的SOC估算策略,在保证算法实时性的前提下提高了SOC估算精度。本文称其为AH-Kalman交叉运行的SOC估算策略。本研究基于实验数据搭建了电池模型,对所提出的算法与传统的智能优化算法进行了算法复杂度和精度对比分析,以验证新算法的有效性。在此基础上,将新算法写成C代码下载到BMS控制器,搭建微控制器在环测试(processor-in-the-loop simulation,PILS)系统对算法在控制器中的实际运行效果进行了验证。

① 国家自然科学基金(51305475),重庆市基础科学与前沿技术研究专项(cstc2013jcyjA60004)和重庆市教委科学技术研究(KJ1500927)资助项目。
 ② 男,1983年生,博士,副教授;研究方向:新能源汽车匹配,仿真与测试,车辆动力传动与控制;E-mail: Luoyong@cqut.edu.cn
 ③ 通讯作者,E-mail: 365086727@qq.com
 (收稿日期:2017-01-10)

1 AH-Kalman 交叉运行的 SOC 估算策略

现有的 SOC 在线估计算法中,开路电压(OCV)法简单易行^[6],但由于电池开路电压具有滞回特性和超电势特性,使开路电压法不适用于动态工况,只适合在汽车启动前或停车等静态工况使用;安时(AH)计量法通过对电流进行积分来估算剩余电量,但由于自放电和测量误差的存在,在运行过程中会产生误差累计,难以自我修正;卡尔曼(Kalman)滤波法的核心思想是对系统状态做出最小方差意义上的最优估计,它能对初始误差进行修正,对噪声有很强的抑制作用,在估算过程中有很好的精度,缺点是涉及大量的矩阵运算^[7,8]。

综合考虑 SOC 估算算法的精度和实时性,针对各种 SOC 估算算法的特点,利用其精确的使用时刻,规避其不精确的使用时刻,本文提出如下的 SOC 估算策略:(1)在汽车刚启动时,动力电池没有滞回现象和超电势现象,此时采用开路电压(SOC)法可以得到精确的 SOC 初值;(2)获得 SOC 初值之后,在车辆运行过程中采用算法复杂度低的安时(AH)计量法估算 SOC,以减小控制器运算负担;(3)为消除自放电和电流读取误差对安时计量法造成的积累误差,间歇性地运行卡尔曼滤波算法对安时法进行修正,以较小的计算量换取 SOC 估算精度的提升。

算法流程如图 1 所示,该 SOC 估算策略以算法复杂度低的安时法为主要方式,在运行一段时间安时法之后,启动卡尔曼滤波算法消除安时法的积累误差,待卡尔曼滤波算法收敛后,则重新启用安时法。以此交叉运行的方式,可在保证 SOC 估算精度的同时大大减轻微控制器的运算负担。

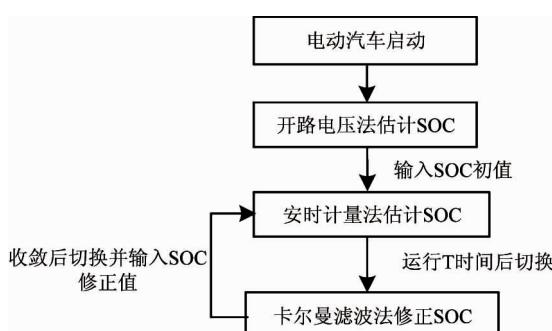


图 1 AH-Kalman 交叉运行 SOC 估算策略算法流程图

2 算法实现

2.1 电池模型的建立

AH-Kalman 交叉运行 SOC 估算策略的卡尔曼滤波算法的实现需要配合高精度电池模型,首先通过理论分析结合实验数据拟合建立锂电池模型。本文将锂电池等效为二阶 RC 电路模型^[5],如图 2 所示。

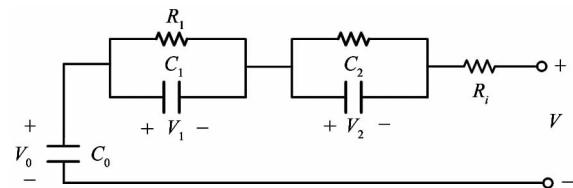


图 2 二阶 RC 等效电路模型

图中电容 \$C_0\$ 为简化电压源, \$C_0\$ 两端的电压 \$V_0\$ 为电池的开路电压; \$R_i\$ 为电池的欧姆内阻; \$R_1\$、\$C_1\$ 和 \$R_2\$、\$C_2\$ 组成二阶 RC 网络,其两端电压分别为 \$V_1\$ 和 \$V_2\$。

根据电路模型能得到二阶 RC 模型的状态方程和输出方程如下:

$$\begin{bmatrix} \dot{V}_0 \\ \dot{V}_1 \\ \dot{V}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 - \frac{1}{R_1 \cdot C_1} & 0 \\ 0 & 0 & 1 - \frac{1}{R_2 \cdot C_2} \end{bmatrix} \cdot \begin{bmatrix} V_0 \\ V_1 \\ V_2 \end{bmatrix} \quad (1)$$

$$+ \begin{bmatrix} -\frac{1}{C_0} \\ \frac{1}{C_1} \\ \frac{1}{C_2} \end{bmatrix} \cdot I \quad (1)$$

$$V = [1 \quad -1 \quad -1] \cdot \begin{bmatrix} V_0 \\ V_1 \\ V_2 \end{bmatrix} - R_i \cdot I \quad (2)$$

取电池模型中的状态变量为

$$x = [x_1 \quad x_2 \quad x_3]^T = [V_0 \quad V_1 \quad V_2]^T \quad (3)$$

系统输入 \$u\$ 为电池的工作电流 \$I\$,且放电为正,系统观测值 \$y\$ 为电池的工作电压 \$V\$,采样时间为 \$T\$。

得二阶 RC 等效电路系统的状态方程和观测方程为

$$\begin{aligned} x_k &= Ax_k + Bu_k + \omega_k \\ y_k &= Cx_k + Du_k + \nu_k \end{aligned} \quad (4)$$

式中, x_k 为系统状态变量 x 在 k 时刻的值; u_k 为系统的输入量; y_k 为系统输出在 k 时刻的值; A 为系统的传递矩阵; B 为系统的输入矩阵; C 为系统的观测矩阵; D 为系统的前馈矩阵; ω_k, ν_k 为系统的过程噪声和观测噪声。

根据二阶 RC 等效电路模型可得

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 - \frac{1}{R_1 \cdot C_1} & 0 \\ 0 & 0 & 1 - \frac{1}{R_2 \cdot C_2} \end{bmatrix}, \quad B = \begin{bmatrix} -\frac{1}{C_b} \\ \frac{1}{C_1} \\ \frac{1}{C_2} \end{bmatrix},$$

$$C = [1 \quad -1 \quad -1], \quad D = -R_i$$

根据电池放电实验数据拟合二阶 RC 电池模型的参数如表 1 所示。

表 1 二阶 RC 模型参数

| $R_i (\Omega)$ | $R_1 (\Omega)$ | $R_2 (\Omega)$ | $C_1 (F)$ | $C_2 (F)$ | $C_0 (F)$ |
|----------------|----------------|----------------|-----------|-----------|-----------|
| 0.0055 | 0.0041 | 0.0017 | 21797 | 3634 | 71354 |

电动汽车动力电池在实际使用过程中电流不可能是恒定的,因此我们使用变电流放电来验证模型的适用性,使用不规则脉冲放电来模拟工况,可以验证二阶 RC 电池模型响应的快速性和稳定性。图 3



图 3 电池充放电实验

为电池充放电实验设备,图 4 为不规则脉冲放电电压曲线与二阶 RC 模型拟合曲线结果对比图,图 5 为二阶 RC 模型拟合曲线误差图。实验结果表明,二阶 RC 电池模型估计电压的误差在 0.048 以内,有较好的估计结果,满足电池模型性能的要求。

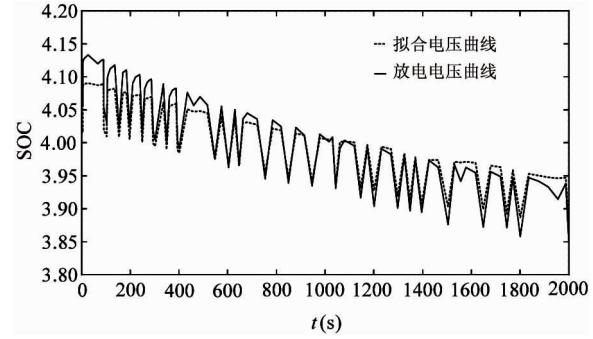


图 4 不规则脉冲放电电压曲线与二阶 RC 模型拟合曲线

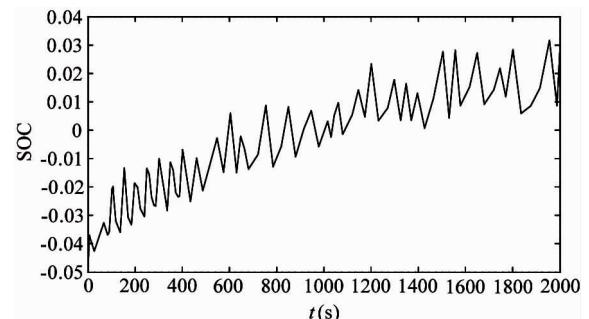


图 5 不规则脉冲放电二阶 RC 模型拟合曲线误差图

2.2 安时-卡尔曼交叉运行算法

本文所述的安时-卡尔曼交叉运行 SOC 估算策略,当车辆正常运行时,采用计算量小的安时计量法估算 SOC 值,SOC 值的计算如下式^[9]:

$$SOC = SOC_0 - \frac{1}{C_A} \int_0^t Z idt \quad (5)$$

式中 SOC_0 为 SOC 初始值, C_A 为电池容量, Z 为库伦效率,公式中的电流符号在放电时为正,充电时为负。

SOC 初始值的准确性对安时计量法的估算精度有很大影响,准确的 SOC 初始值可通过开路电压法获得。开路电压法的原理即电池 SOC 与其开路电压具有一一对应关系,该对应关系由充放电实验获得,如图 6 所示。

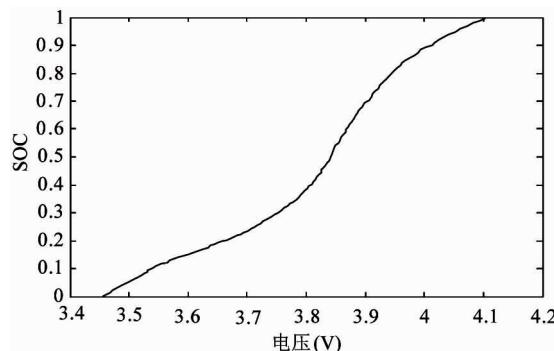


图 6 OCV 与 SOC 关系曲线

由于自放电和测量误差的存在,安时计量法在运行过程中会产生误差累计,难以自我修正,综合考虑算法的精度和计算量,安时-卡尔曼交叉运行的 SOC 估算策略采用间歇运行卡尔曼滤波算法对安时计量法的累积误差进行修正。采用经典卡尔曼滤波法估算电池组的开路电压,再由 OCV-SOC 曲线查表得到电池组的准确 SOC,对安时计量法的累积误差进行修正,具体修正计算过程如下:

(1) 算法初始化: $k = 0$ 时刻, 令 $\hat{x}_0 = \mathbf{E}[x_0]$, $\mathbf{P} = \mathbf{E}[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T]$ 。估计的 k 时刻的状态值和误差协方差在测量更新之前分别为 $\hat{x}_{k+1/k}$ 和 $P_{k+1/k}$, 测量修正后为 \hat{x}_{k+1} 和 P_{k+1} 。

(2) SOC 修正值估算:

$$\hat{x}_{k+1/k} = \mathbf{A}\hat{x}_k + \mathbf{B}u_k \quad (6)$$

$$P_{k+1/k} = \mathbf{A}P_k\mathbf{A}^T + \mathbf{Q}_k \quad (7)$$

$$L_k = P_{k+1/k}\mathbf{C}^T(\mathbf{C}P_{k+1/k}\mathbf{C}^T + R_k)^{-1} \quad (8)$$

$$\hat{x}_{k+1} = \hat{x}_{k+1/k} + L_k(y_{k+1} - \mathbf{C}\hat{x}_{k+1/k} - \mathbf{D}u_{k+1}) \quad (9)$$

$$P_{k+1} = (\mathbf{E} - L_k\mathbf{C})P_{k+1/k} \quad (10)$$

式中, \mathbf{Q}_k 为系统过程噪声矩阵; R_k 为测量噪声误差; \mathbf{E} 为单位矩阵。

2.3 AH-Kalman 交叉运行步长设置

(1) 开路电压法的步长设置: 在 AH-Kalman 交叉运行算法中, 开路电压法的作用是获得动力电池在汽车刚刚启动时 SOC 初始值, 因此开路电压法的步长只需设置很短, 在本文中取 5s。

(2) 卡尔曼滤波法的步长设置: 卡尔曼滤波法的步长设置是根据卡尔曼滤波法矩阵迭代运算收敛至 SOC 真值的速率决定的。卡尔曼滤波算法的步长设置过大会加重微控制器的运算负担, 故运行步

长应设置为确保卡尔曼滤波法收敛至真值附近的最优值。决定卡尔曼滤波算法收敛速度的因素众多, 故步长数一般由针对特定模型的仿真实验获得。通过针对本文模型进行的仿真实验分析, 一般取步长为 200 可确保算法收敛至真值附近。

(3) 安时计量法的步长设置: 安时计量法的运行步长 m 是依据安时积分法的运行结果偏离 SOC 真实值的速率决定的。即安时计量法误差积累的速度越快, 则越频繁地需要卡尔曼滤波法来修正 SOC 估计值, 步长 m 需设置的越短。但是, 随着步长 m 的缩短, 会导致微控制器在相同的运行时间内更多地运行了卡尔曼滤波法, 增加了微控制器的运算负担。因此, 运行步长 m 的取值应在安时计量法积累的误差在估算精度要求的范围内尽可能取一个较大的数值。

2.4 AH-Kalman 交叉运行时序控制逻辑

本算法以保证一定精度的前提下尽量减少微控制器运算负担为原则, 开路电压法、安时积分法和卡尔曼滤波法采用使能子系统封装, 通过时序控制模块产生高低电平控制子系统的开启与关闭, 以此实现各算法交叉运行, 不运行的算法不占用微控制器内存。

时序模块以时钟作为切换算法的依据。在汽车刚起动的前 5s, 所有电池组运行开路电压法获取 SOC 初值。5s 之后根据时序产生模块输出的数字进行安时积分法和卡尔曼滤波算法的交叉运行。时序产生模块通过对时钟的取余运算, 可以依次输出 0 到 6 共计 6 个数字, 间隔时间 200s。时序产生模块输出的数字传输给使能选择模块, 使能选择模块判断接收到的数字是否为 0, 并根据判断结果在三个输出端口输出对应的高低电平信号, 通过控制使能子模块的开启与关闭来控制该电池组当前运行的 SOC 估算方法。图 7 为时序模块的运行方式。

鉴于交叉运行时序的复杂性, 一方面要保证 SOC 初值和修正值的正确传递, 另一方面要保证每个电池组在任一时刻都正确输出 SOC 估算结果, 因此输出选择模块至关重要。为了配置可生成目标微控制器的 C 代码, 输出选择模块选用配有目标语言编译(target language compiler, TLC) 文件的 C MEX S

函数编写,功能是根据时序模块使能信号端口输出的高低电平判断哪一个使能模块正在运行,从而开通该使能子系统的信号通路,进行运算结果数据的传递。图 8 为输出选择模块的运行原理图。其中初值/修正值选择输出模块控制开路电压法获得的初值和卡尔曼滤波法获得的修正值能正确传输至安时积分法模块;SOC 估算结果选择输出模块则是控制将正在运行的估算算法的结果输出,以保证任一时刻都能正确输出高精度 SOC 估算结果。

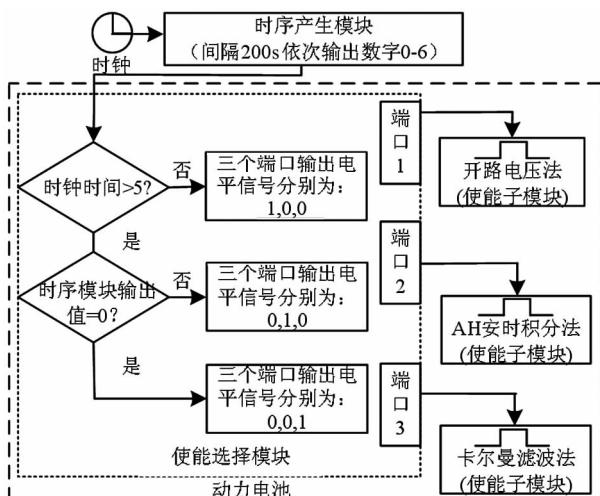


图 7 时序模块

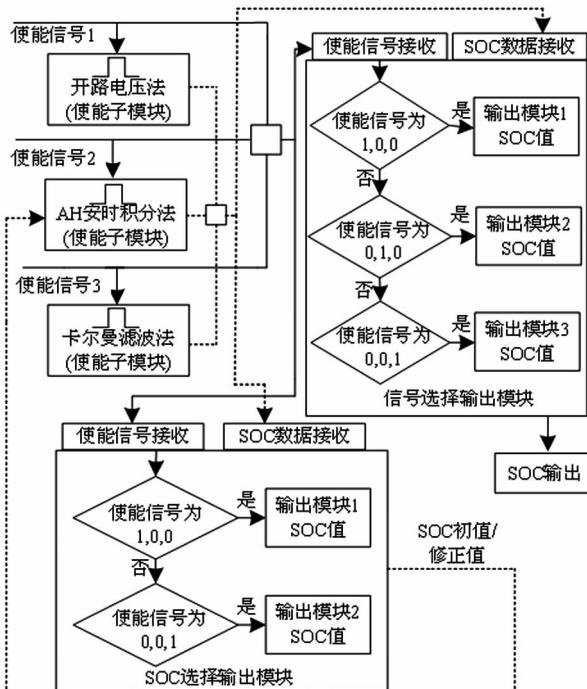


图 8 选择输出模块运行原理图

3 系统建模与仿真研究

根据前面建立的电池模型和 SOC 估算算法,采用 Matlab/simulink 软件搭建系统仿真模型,如图 9 所示。由于电池模型采用实验数据拟合并验证,因而该仿真模型具有较高的计算精度,可以用其验证本文所述算法对 SOC 的估算精度。

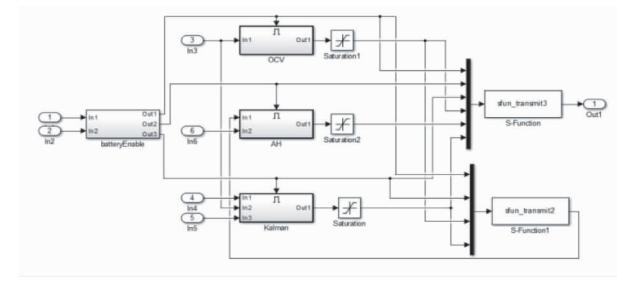


图 9 系统 simulink 仿真模型

为了验证 AH-Kalman 交叉运行算法的 SOC 修正功能,安时积分法不修正库伦效率,即估算 SOC 误差较大,以此来验证运行卡尔曼滤波法修正 SOC 的功能。图 10 为 FUDS 工况下 AH-Kalman 交叉运行算法对 SOC 估算结果与 SOC 真值的对比,其中 SOC 真值为迪卡龙 BNT300 充放电设备记录的实验数据计算获得,图 11 为 AH-Kalman 交叉运行算法 SOC 估计结果与 SOC 实验值的误差。通过 SOC 估算结果对比曲线可以看出,SOC 估算值与测试真值的误差小于 5%,说明 AH-Kalman 交叉运行算法可以较好地修正安时积分法的积累误差,能够较为准确地对 SOC 值进行估算。

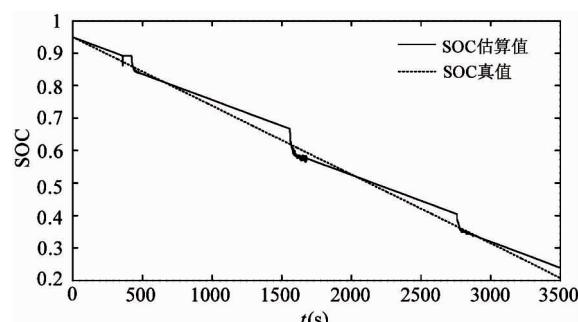


图 10 SOC 估计结果与真值对比曲线

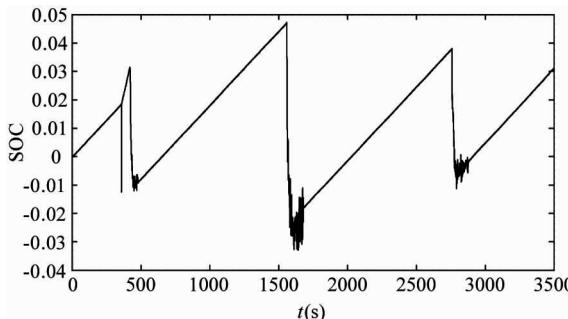


图 11 SOC 估计结果误差图

4 代码时间复杂度分析与微控制器在环验证

4.1 算法时间复杂度分析

代码时间复杂度的高低不仅决定了微控制器内响应各种紧急中断请求的反应能力,而且从微控制器运算能力的方面决定了控制系统实时性的好坏。

本算法的目的在于保证精度的前提下使算法复杂度保持在较低水平,因而有必要对其算法复杂度进行分析验证。根据算法复杂度分析理论,常将算法中语句执行频度作为算法复杂度的度量指标^[10]。

在 AH-Kalman 交叉运行算法中,积分运算的算法是 Forward Euler,此算法输出信号 y 的离散差分公式为:

$$y(n) = y(n-1) + K \times T_s \times u(n-1) \quad (11)$$

式中用 u 表示积分的输入, y 表示输出, K 表示输入信号的增益值, T_s 表示采样时间间隔。根据算法复杂度分析理论,积分算法的执行频度 $f_1(n) = n$ 。由第 2 节中卡尔曼滤波法公式可以计算出若单纯采用卡尔曼滤波算法估算 SOC,其算法执行频度为 $f_2(n) = n^2$ 。

设 AH-Kalman 交叉运行算法运行 2000 步的步长为一个测试区间;设交叉运行算法从开始运行安时计量法到运行卡尔曼滤波法修正结束即将运行下一次安时计量法时为一个循环。在一个循环内,安时计量法的运行步长为 m ,卡尔曼滤波算法的运行步长为 k ,根据 2.3 节中步长设置的分析,此处取 $k = 200$ 。该测试区间内传统卡曼滤波算法的执行频度 $f_4(n) = 2000n^2$,AH-Kalman 交叉运行算法的执行频度为

$$f(n) = \begin{cases} mn & (m > 2000) \\ mn + (2000 - m) & (1800 < m < 2000) \\ 200n^2 + 1800n & (900 < m < 1800) \\ 2mn + (2000 - 2m)n^2 & (800 < m < 900) \\ 1600n + 400n^2 & (533 < m < 800) \\ 3mn + (2000 - 3mn)n^2 & (466 < m < 533) \\ 600n^2 + 1400n & (350 < m < 466) \end{cases} \quad (12)$$

为了方便对比在步长为 2000 步的同一测试区间内, m 取不同值时 AH-Kalman 交叉运行算法和传统卡尔曼滤波法的执行频度,在此取 m 在不同区间的中位数绘制执行频度对比图如图 12。通过曲线对比分析可得出结论,AH-Kalman 交叉运行算法的代码时间复杂度远低于传统的卡尔曼滤波法,且随着 m 数值的增大,AH-Kalman 交叉运行算法的代码时间复杂度越低。

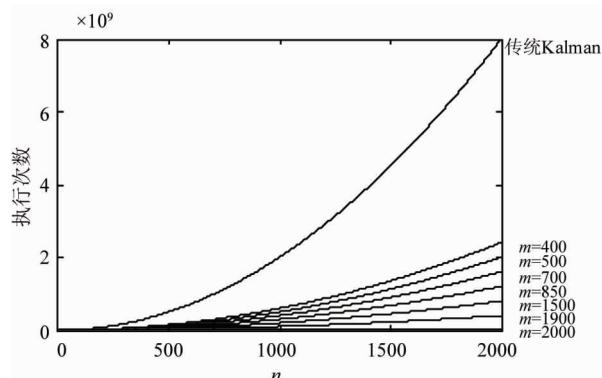


图 12 算法执行频度

4.2 C 代码生成与微控制器在环验证

对于控制算法和控制逻辑的有效性和对 CPU 的占用情况,需要将算法生成 C 代码,下载到实际硬件环境下验证。微控制器在环测试 (PILS) 是将控制算法生成 C 代码并下载到实际控制器中,在工控机中运行通过实验数据建立的高精度电池模型,控制器通过串口通信方式与工控机连接实现数据交换,为控制算法验证提供一个可以接近其真实工作环境的平台^[11]。

本文以 Matlab2013b/simulink 为平台生成目标控制器 C 代码,生成目标代码的配置工作包括:(1)

配置存储类型、数据类型及数据对象和数据词典; (2) 针对目标硬件的代码生成及优化; (3) 编写模型中所用 S 函数的 TLC 目标文件; (4) 整合目标嵌入式硬件及其使用的集成开发环境 (Integrated Development Environment, IDE) 到 simulink 工具链中; (5) 依托微控制器在环验证, 实现需求与模型和代码与模型的双向追踪功能。

下面验证 4.1 中代码时间复杂度的分析结果和对比 AH-Kalman 与传统卡尔曼滤波法对微控制器 CPU 的占用情况, 为了使实验效果更加明显, 通过将算法的运算量扩大 10 倍来增加算法的代码时间复杂度, 设计微控制器在环测试实验如下: (1) 在 simulink 中, 复制 10 组 AH-Kalman 交叉运行算法模型, 配置生成代码下载至目标微控制器, 即微控制器中同时运行 10 组 AH-Kalman 交叉运行算法; (2) 在 simulink 中复制 10 组传统卡尔曼滤波算法模型, 生成代码下载至目标微控制器。

本文选用 Spansion 公司生产的 32 位嵌入式处理器 MB9BF568R。图 13 为本文搭建的微控制器在

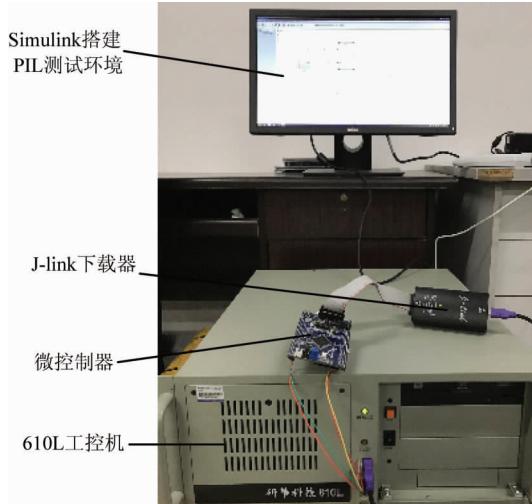


图 13 微控制器在环测试平台

环测试平台, 其中示波器分别接收来自 simulink 模型仿真值和通过串口传回的微控制器运行结果。AH-Kalman 交叉运行算法的 PIL 运行结果如图 14 所示, 其中图 14(a) 为 AH-Kalman 交叉运行算法在 simulink 里的仿真运行结果; 图 14(b) 为 AH-Kalman 交叉运行算法在微控制器内实际运行的结果。AH-

Kalman 交叉运行算法在微控制器内运行估算动力电池的 SOC 值时, 和 simulink 模型仿真结果保持高度一致, 其余 9 组算法运行结果一致, 故在此不一一列出。而 10 组传统卡尔曼滤波同时运行的微控制器已经无法正常传输数据, 微控制器在环测试以系统报错结束。

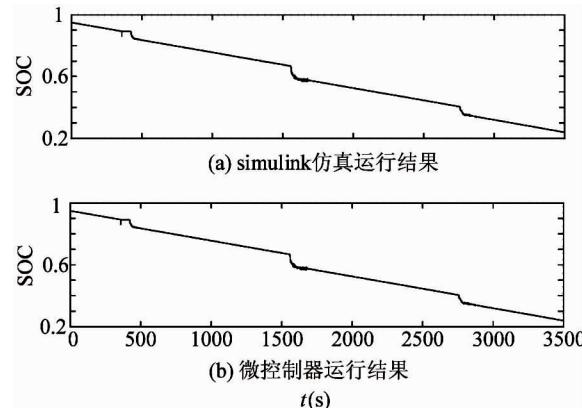


图 14 PIL 运行结果图

实验结果表明, AH-Kalman 交叉运行算法在微控制器内运行估算动力电池的 SOC 值时, 在微控制器 CPU 占用方面优于传统卡尔曼滤波法, 结合 4.1 中的代码时间复杂度分析, 证明 AH-Kalman 交叉运行算法在保证 SOC 估算精度的同时又具有较好的实时性。

5 结 论

(1) 针对传统 SOC 估算算法的不足, 综合考虑 SOC 估算精度和系统的实时性, 设计了用开路电压法确定 SOC 初值, 以实时性较强的安时积分法为主, 采用间歇运行的卡尔曼滤波法修正积分误差的 SOC 估算策略。

(2) 建立系统仿真模型, 对所提出的算法进行仿真分析, 结果表明, AH-Kalman 交叉运行算法能对安时法的积累误差起到有效的修正作用, 具有较高的 SOC 估算精度。

(3) 搭建微控制器在环测试系统, 对控制策略的算法复杂度进行验证, 结果表明, AH-Kalman 交叉运行算法在微控制器 CPU 占用方面优于传统卡尔

曼滤波法,在保证 SOC 估算精度的同时又具有较好的实时性。

参考文献

- [1] 杜晓伟,任勇,齐铂金等. ISG 型中混合动力汽车镍氢电池管理系统的研制. 高技术通讯,2010,20(4):422-426
- [2] 辛品,葛元月,薄伟等. 基于单片机的纯电动汽车电池管理系统设计. 农业工程学报,2014, 30 (12) :163-170
- [3] Wu T, Liu L, Xiao Q, et al. Research on SOC estimation based on second-order RC model. *Telkomnika: Indonesian Journal of Electrical Engineering*, 2012,10(7):1667-1672
- [4] Simon S, Nils A, Sebastian S, et al. Particle filter state of charge and state of health estimation for lithium-iron phosphate batteries. *Journal of Power Sources*, 2013,239 (1) :705-710
- [5] Habiballah R, Federico B, Chow M. Online adaptive parameter identification and state-of-charge coestimation for lithium-polymer battery cells. *IEEE Transactions on Industrial Electronics*, 2014, 61(4): 2053-2061
- [6] 徐颖,沈英. 基于改进卡尔曼滤波的电池 SOC 估算. 北京航空航天大学学报,2014,40(6):855-860
- [7] Plett L G. Sigma-point Kalman filtering for battery management systems of LiPB-based HEV battery packs part 2, simultaneous state and parameter estimation. *Journal of Power Sources*, 2006, 161(2):1369-1384
- [8] 张金龙,佟微,漆汉宏等. 平方根采样点卡尔曼滤波在磷酸铁锂电池组荷电状态估算中的应用. 中国电机工程学报,2016,29(1):6246-6253
- [9] 李哲,卢兰光,欧阳明高. 提高安时积分法估算电池 SOC 精度的方法比较. 清华大学学报(自然科学版), 2010,08:1293-1296 + 1301
- [10] 严蔚敏,吴伟民. 数据结构(C 语言版). 北京:清华大学出版社,2010. 1-17
- [11] Kumar D, Kumaresan G. Processor-in-loop simulation for formation flying of multiple unmanned MAVs. *International Federation of Automatic Control*, 2016, 49 (1) : 688-693

A strategy for batteries' state-of-charge estimation using AH - Kalman cross operation and its processor-in-the-loop simulation

Luo Yong, Kan Yingzhe, Zhu Chuanmei, Zhao Xiaoshuai, Qi Pengwei, Long Kejun

(Key Laboratory of Advanced Manufacturing Technology for Automobile Parts,
Chongqing University of Technology, Chongqing 400054)

Abstract

To improve the performance of battery management system (BMS), the estimation of batteries' state of charge (SOC) was studied, and according to the requirement of estimation accuracy and system timeliness, a strategy for SOC estimation using the cross operation of the Ampere hour (AH) integration algorithm and the Kalman algorithm, called the SOC estimation strategy using AH-Kalman cross operation for short, was proposed. The strategy uses the open-circuit voltage (OCV) method to determine the initial SOC value, and uses the real-time AH integral method (playing the main role) and the intermittent operating Kalman method to correct the accumulated error of the AH integral method. The system simulation model was established to verify the correction effect of the Kalman filter algorithm on the accumulative error. The C-code of the control algorithm was generated and downloaded to the target controller, and the processor-in-the-loop simulation (PILS) was built, and the algorithm complexity was compared with the traditional Kalman filter. The results show that the proposed SOC estimation strategy can ensure the accuracy of SOC estimation and has the good real-time performance, which is convenient for practical application.

key words: SOC estimation algorithm, processor-in-the-loop simulation, C code generation, Kalman filter algorithm