

一种用于 NS-3 仿真的电池模型及其数值解法

金仁成,周 峰,李 宇

(大连理工大学辽宁省微纳技术及系统重点实验室,辽宁大连 116024)

摘 要:为了解决网络模拟器 NS-3 不能对无线传感器网络节点的电池信息做出准确仿真的问题,吸收 LiIon 电池模型和 KiBaM 电池模型各自的优点,采用 LiIon 电池模型的方法计算 KiBaM 电池模型中有效电荷井的输出电压,得到能够体现电池的速率容量效应和自恢复效应、能够反映输出电压随剩余电量降低而降低的 KiBaM-LiIon 电池模型,并对模型中无法获得解析解的微分方程组求解,基于隐式 Runge-Kutta 方法,提出了易于编程实现的数值解法。仿真和实验的结果表明,模型及其数值解法能够在 NS-3 中对电池使用过程中任何时刻的电压、剩余电量等作出准确预测,且具备计算复杂度低的优点。所提出的电池模型及其数值解法不仅适用于 NS-3,同样也适用于其他网络模拟器,可为无线传感器网络和移动 Ad-hoc 网络的研发提供参考。

关键词:计算机仿真;无线传感器网络;network simulator 3 (NS-3);电池模型;隐式龙格库塔法

中图分类号:TP393.17 **文献标志码:**A

A battery model and its numerical method for the network simulator 3 (NS-3)

JIN Rencheng, ZHOU Feng, LI Yu

(Key Laboratory for Micro/Nano Technology and System of Liaoning Province, Dalian University of Technology, Dalian, Liaoning 116024, China)

Abstract: In order to solve the problem that the network simulator 3 (NS-3) cannot accurately simulate the battery information of the wireless sensor network (WSN) node, the LiIon battery model is used to calculate the output voltage of the available charge well (ACW) of the KiBaM battery model by combining the advantages of the LiIon battery model and the KiBaM battery model, then a KiBaM-LiIon battery model that can reflect the rate capacity effect and recovery effect of the battery, as well as the decrease of the output voltage with the decrease of remaining charge is proposed. For solving the ordinary differential equations without analytical solution in the model, an easy programming numerical solution based on the implicit Runge-Kutta method is proposed. Simulations and experiments show that the proposed model and its numerical solution can accurately predict the voltage and remaining charge at any time during battery use in NS-3, and have the advantage of low computational complexity. The proposed battery model and proposed numerical solution are not only applicable to NS-3, but also to other

收稿日期:2018-12-27;修回日期:2019-03-25;责任编辑:陈书欣

基金项目:国家“十二五”科技支撑计划项目(2011BAG05B02);国家自然科学基金创新群体项目(51621064);中央高校基本科研业务费专项资金(DUT16TD20)

第一作者简介:金仁成(1969—),男,吉林长春人,副教授,博士,主要从事无线传感器网络、物联网等方面的研究。

E-mail:rcjin@dlut.edu.cn

金仁成,周峰,李宇.一种用于 NS-3 仿真的电池模型及其数值解法[J].河北科技大学学报,2019,40(3):233-241.

JIN Rencheng, ZHOU Feng, LI Yu. A battery model and its numerical method for the network simulator 3 (NS-3)[J]. Journal of Hebei University of Science and Technology, 2019, 40(3): 233-241.

network simulators, which provides good technical support and guarantee for the research and development of WSN and mobile Ad-hoc networks (MANET).

Keywords: computer simulation; wireless sensor network (WSN); network simulator 3 (NS-3); battery model; implicit Runge-Kutta

在现代网络通信技术研究中,尤其是大规模网络的研究,在进行网络协议和网络设备的设计、开发及验证时,受设备和场地的限制,通常使用网络模拟技术进行仿真。目前,网络仿真已成为互联网算法性能、协议拓扑最经济快捷的评价方法之一^[1],NS-3(network simulator 3)就是其中一款优秀的网络模拟器。NS-3 是一款比较年轻的网络仿真工具,专门用于教育和研究,基于 GNU GPLv2 许可,可以免费获取、使用和修改^[2]。数以万计的科研工作者将自己的科研成果整合进 NS-3 中,为其面世作出贡献。截止目前,其最新版本为 2018-09-04 的 3.29 版本。

对于一些能量受限的网络,例如无线传感器网络(WSNs),移动 Ad-hoc 网络(MANETs)等,在设计网络协议时需要充分考虑到网络节点的能耗情况^[3]。因此,网络模拟器中必须能够提供精确的电池模型,对网络节点的能量消耗进行精确计算,否则仿真结果便是不可信的。

以无线传感器网络为例,WSN 节点通常由不可充电或难以充电的电池进行供电,而大量的研究和实验表明,电池并不是理想的、线性的电荷储存器,而是一个复杂的非线性过程^[4],主要表现在^[5-7]:1)电池电压随着放电过程缓慢下降;2)电池能够释放的总电量随着负载电流的增大而减小,该现象被称为电池速率容量效应;3)当电池停止放电或以小电流工作一段时间后,电池的电压反而会上升,该现象被称为电池的恢复特性。为了体现电池的这 3 种特性,NS-3 提供了 3 种电池模型:理想恒压源模型、LiIon 电池模型^[8]、RV 电池模型^[9-10]。理想恒压源模型仅能进行非常粗略的计算,适用于不受能量因素影响的无线网络仿真中。LiIon 电池模型描述了电池电压随放电过程缓慢下降的现象,但并没有体现电池的速率容量效应和自恢复特性。RV 电池模型则描述了电池的非线性特性,但并没有考虑到电池电压受剩余电量的影响。

为提高有关能耗相关的仿真准确度,笔者建立了电池的 KiBaM-LiIon 模型,并给出了该模型适用于 NS-3 以及其他仿真软件的低计算复杂度的数值解法,最后通过仿真和实验验证了所提出方法的有效性。

1 电池模型分析

1.1 LiIon 电池模型

LiIon 电池模型描述了电池在放电过程中的电压下降现象。该模型仅以 SoC(state-of-charge, 荷电状态)作为输入量,可以精确地预测铅酸蓄电池、锂离子电池、镍镉电池、镍氢电池等符合以下特点的任何种类电池的放电曲线。

- 1) 电池内阻恒定,在充放电过程中不发生太大的改变;
- 2) 充电和放电曲线可逆,在一次充放电循环中不考虑电池的寿命衰减;
- 3) 电池的放电量不与放电电流有关,不考虑电池的非线性特性;
- 4) 电池处于恒温状态;
- 5) 不考虑电池的自放电特性;
- 6) 电池没有记忆效应。

LiIon 模型的数学模型如下:

$$\begin{cases} V = E - Ri, \\ E = E_0 - K \cdot \frac{q_r}{q_r - \int i dt} + A \cdot e^{-B \cdot \int i dt}, \\ A = e_f - e_e, \\ B = \frac{3}{q_e}, \\ K = \left| [e_f - e_n + A \cdot (e^{-Bq_n} - 1)] \cdot \frac{q_r - q_n}{q_n} \right|, \\ E_0 = e_f + K + Ri - A, \end{cases} \quad (1)$$

式中: V 是输出电压(V); E 是输出电动势(V); E_0 是电池恒定电压(V); K 是极化电压(V); A 是指数区域幅度(V); B 是指数区域时间常数倒数($A \cdot h^{-1}$); R 是内阻(Ω); e_f 是电池初始电压(V); e_n 是电池名义电压(V); e_e 是指数区域结束位置电压(V); q_r 是电池额定容量($A \cdot h$); q_n 是名义区域结束剩余容量($A \cdot h$); q_e 是指数区域结束剩余容量($A \cdot h$)。

放电曲线的指数区域如图 1 所示。

但是 LiIon 电池模型没有体现电池的速率容量效应和自恢复特性^[11],因此该模型仅能描述电池在恒流放电条件下的状态,并不适用于类似传感器网络节点这种电源时开时关、负载时大时小的场景。

1.2 KiBaM 电池模型

NS-3 提供了另一种能够描述电池的非线性特性的电池模型:RV 电池模型^[9-10]。经大量仿真分析发现,RV 电池模型实际上就是一种连续的 KiBaM 电池模型^[12-13]。KiBaM 电池模型(kinetic battery model)^[14]是一种分析模型,具有较高的抽象性。

KiBaM 利用 2 个井(well)来描述电池的特性,如图 2 所示,2 个井分别被称为束缚电荷井(BCW)和可用电荷井(ACW)。ACW 直接给负载提供能量,而 BCW 只能通过一个“狭小”的通道与 ACW 进行能量交换,能量在通道中的转移速度与两井之间的“高度差”有关,与一个假设的、与电池性质有关的系数有关。该模型通过微分方程组(2)表示。

$$\begin{cases} \frac{dy_1}{dt} = -I + k \cdot (h_2 - h_1), \\ \frac{dy_2}{dt} = -k \cdot (h_2 - h_1), \end{cases} \quad (2)$$

式(2)中 y_1 为图 2 中的 i , y_2 为图 2 中的 j ,其余符号皆如图 2 所示。

电池放电时,按照方程组(2)描述,由 ACW 对外输出能量,ACW 的高度 h_1 下降,与 BCW 之间产生高度差,BCW 开始缓慢地向 ACW 转移能量。如果负载很大,BCW 受通道所限来不及向 ACW 补充能量,ACW 能量就已经耗尽,导致电池释放的总能量降低,符合速率容量效应。若在放电一段时间后停止放电,此时 BCW 仍向 ACW 缓缓补充能量,刚开始时高度差最大,电池的电量恢复速度也最快,随着过程的进行,两井之间的高度差逐渐缩小直到平衡,外在表现就是电池的电量回升速度越来越慢直到不再恢复^[15],符合电池的自恢复效应。

KiBaM 电池模型没有对电池内部的电化机理建模,而是建立了一个非常有趣的抽象模型,解释了电池的速率容量效应和自恢复效应。但在计算过程中,为了简便却忽略了电池的输出电压与剩余电量之间的关系,将电池看作是恒电压输出,这是十分不准确的。为了得到这种关系,一些文献简单地对输出电压和剩余电量做拟合处理^[16],有效但却缺乏可推广性。

2 KiBaM-LiIon 电池模型与数值解法

2.1 KiBaM-LiIon 电池模型

KiBaM 模型描述了电池的非线性特性却忽略了电压的变化,而电压的变化正是 LiIon 模型所着重考虑的,因此可以将 2 个模型进行结合而弥补各自的缺点。总体思路是:将 KiBaM 模型的 ACW 视作是一块单独的电池,使用 LiIon 模型来计算 ACW 的对外输出电压。

ACW 从初始状态 0 时刻开始到 t 时刻之间的电量变化为

$$q(t) = y_1(0) - y_2(t)。 \quad (3)$$

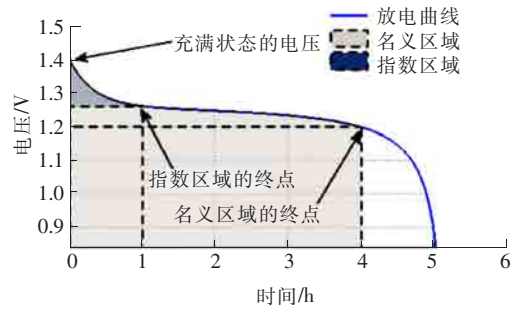


图 1 LiIon 电池模型的放电曲线
Fig.1 Discharge curve of LiIon battery model

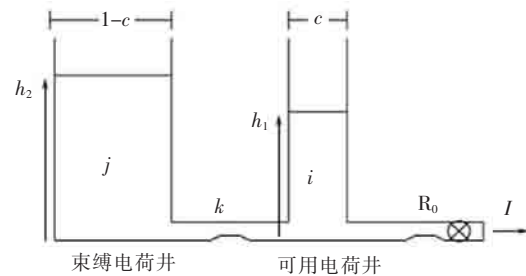


图 2 KiBaM 模型
Fig.2 KiBaM model

在 LiIon 模型中有一项 $\int i dt$, 指的是负载在 0 时刻到 t 时刻之间消耗的电量, 也可以说是从初始状态 0 时刻开始到 t 时刻之间的电量变化, 毕竟在不考虑非线性特性的 LiIon 模型中这两者是相等的。但如果想要用 LiIon 模型对 KiBaM 模型的 ACW 的输出电压进行求解, 就必须采用后一种解释, 因此要将 LiIon 模型与 KiBaM 模型联立, 并将其中的 $\int i dt$ 全部替换为式(4):

$$\begin{cases} V = E - Ri, \\ E = E_0 - K \cdot \frac{q_r}{q_r - q(t)} + A \cdot e^{-Bq(t)}, \\ A = e_f - e_e, \\ B = \frac{3}{q_e}, \\ K = \left[e_f - e_n + A \cdot (e^{-Bq_n} - 1) \right] \cdot \frac{q_r - q_n}{q_n}, \\ E_0 = e_f + K + Ri - A, \\ \frac{dy_1}{dt} = -i - k \left(\frac{y_1}{c} - \frac{y_2}{1-c} \right), \\ \frac{dy_2}{dt} = -k \left(\frac{y_2}{1-c} - \frac{y_1}{c} \right), \\ q(t) = y_1(0) - y_2(t), \end{cases} \quad (4)$$

式中各符号解释同式(1)与式(2)。

2.2 用于 NS-3 的数值解法

KiBaM-LiIon 电池模型中有一段微分方程组, 微分方程组的解析解一般很难得到, 甚至不存在, 因此通常对微分方程模型进行数值求解, 这是工程领域常见的做法^[17]。微分方程的数值解最常用的方法是四阶经典 Runge-Kutta 解法, 但该解法对迭代步长有要求, 只有在迭代步长足够小的情况下才能够保证数值稳定性^[18], 否则误差急剧增加, 掩盖了真值, 使计算不能继续迭代下去。举例说明, 假如取迭代步长 $h=0.001$ 的四阶经典 Runge-Kutta 解法对电池模型进行求解, 对一个包含 50 个节点的网络进行 100 s 的仿真, 单能量部分总计就要进行 500 万次迭代, 这对仿真软件来说是一个巨大的负担, 显然是无法令人接受的。

虽然计算精度仍然受迭代步长的影响, 但隐式 Runge-Kutta 解法却有无条件的数值稳定性^[19]。只要把隐式解法显化, 就可以在 NS-3 或其他仿真软件中使用。

对于微分方程组:

$$\begin{cases} \frac{dy_1}{dt} = -i - k \left(\frac{y_1}{c} - \frac{y_2}{1-c} \right), \\ \frac{dy_2}{dt} = -k \left(\frac{y_2}{1-c} - \frac{y_1}{c} \right). \end{cases} \quad (5)$$

写成矩阵形式

$$\frac{d\mathbf{u}}{dt} = \mathbf{A}\mathbf{u} + \mathbf{f}, \quad (6)$$

其中:

$$\mathbf{A} = \begin{bmatrix} -\frac{k}{c} & \frac{k}{1-c} \\ -\frac{k}{1-c} & \frac{k}{c} \end{bmatrix}, \quad (7)$$

$$\mathbf{f} = \begin{bmatrix} -i \\ 0 \end{bmatrix}, \quad (8)$$

$$\mathbf{u} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}. \quad (9)$$

该微分方程组的二级四阶隐式 Runge-Kutta 解法^[20]是:

$$\begin{cases} \mathbf{u}_{n+1} = \mathbf{u}_n + \frac{h}{2}(\mathbf{k}_1 + \mathbf{k}_2), \\ \mathbf{k}_1 = \mathbf{A} \left[\mathbf{u}_n + \frac{h}{4}\mathbf{k}_1 + \frac{3 + 2\sqrt{3}}{12}h\mathbf{k}_2 \right] + \mathbf{f}, \\ \mathbf{k}_2 = \mathbf{A} \left[\mathbf{u}_n + \frac{h}{4}\mathbf{k}_2 + \frac{3 - 2\sqrt{3}}{12}h\mathbf{k}_1 \right] + \mathbf{f}, \end{cases} \quad (10)$$

式中:向量 \mathbf{k}_1 和 \mathbf{k}_2 都是隐式的,只需要将这 2 个向量显化,放在等式的一边,就可以当作是普通表达式在各类仿真软件中进行迭代求解,向量 \mathbf{k}_1 和 \mathbf{k}_2 的显式表达式为

$$\mathbf{k} = (\mathbf{I} - \mathbf{C})^{-1}\mathbf{d}, \quad (11)$$

其中:

$$\mathbf{k} = [\mathbf{k}_1 \quad \mathbf{k}_2]^T, \quad (12)$$

$$\mathbf{C} = \begin{bmatrix} \frac{1}{4}h\mathbf{A} & \frac{3 + 2\sqrt{3}}{12}h\mathbf{A} \\ \frac{3 - 2\sqrt{3}}{12}h\mathbf{A} & \frac{1}{4}h\mathbf{A} \end{bmatrix}, \quad (13)$$

$$\mathbf{d} = [\mathbf{A}\mathbf{u}_n + \mathbf{f} \quad \mathbf{A}\mathbf{u}_n + \mathbf{f}]^T. \quad (14)$$

根据式(11)到式(14)解出向量 \mathbf{k}_1 和 \mathbf{k}_2 的显式表达式,Matlab, Mathematica, R 语言, Python 的 SymPy 库等支持符号运算的语言都可以实现。本文利用 Matlab 的 solve 函数进行求解,最后得到的微分方程组的显式迭代表达式非常长。但得益于隐式解法良好的数值稳定性,可以选用较长的迭代步长,减少迭代次数,因此在迭代次数上减少的运算量远远比公式变长增加的运算量要多。

将解得的向量 \mathbf{k}_1 和 \mathbf{k}_2 代入式(10),得到微分方程组(5)的数值解迭代式的形式如同式(15)。函数 f_1 和 f_2 同样按照上述 Matlab 中符号运算功能得到,具体表达式较长不在本文中给出。

$$\begin{cases} y_{1,n+1} = f_1(y_{1,n}, y_{2,n}), \\ y_{2,n+1} = f_2(y_{1,n}, y_{2,n}). \end{cases} \quad (15)$$

2.3 模型在 NS-3 中的实现方法

NS-3 是 C++ 编写的,添加新的模型需要修改源代码。笔者提出的仿真方案是以 LiIon 电池模型为基础改进而来,因此在编写 NS-3 的 C++ 代码时以 ns3::LiIonEnergySource 类为基础进行修改。ns3::LiIonEnergySource 类的文件地址是 src/energy/model/li-ion-energy-source.h 和 src/energy/model/li-ion-energy-source.cc,在同一文件夹下创建这 2 个文件的副本,命名为 src/energy/model/li-ion-kibam.h 和 src/energy/model/li-ion-kibam.cc,并且将文件中的类名改为 ns3::LiIonKiBaM,同时构造函数和析构函数的函数名也要做相应的改动。除了名称和表 1 提到的修改之外,其余均保持 ns3::LiIonEnergySource 类的原样不变。

表 1 ns3::LiIonKiBaM 类相比 ns3::LiIonEnergySource 类所作修改

Tab.1 Modification of ns3::LiIonKiBaM compared to ns3::LiIonEnergySource

名称	类型	访问权限	操作	说明
LiIonKiBam()	构造函数	Public	修改	对新增成员变量的初始化
void CalculateRemainingEnergy()	函数	Private	修改	NS-3 通过定时调用该函数来计算剩余能量
m_c	double	Private	新增	式(4)中的 c
m_k	double	Private	新增	式(4)中的 k
m_y1	double	Private	新增	式(15)中的 $y_{1,n+1}$
m_y1b	double	Private	新增	式(15)中的 $y_{1,n}$
m_y2	double	Private	新增	式(15)中的 $y_{2,n+1}$
m_y2b	double	Private	新增	式(15)中的 $y_{2,n}$
m_i1	double	Private	新增	上一次迭代计算时刻的电流
m_i2	double	Private	新增	本次迭代计算时刻的电流

函数 CalculateRemainingEnergy() 在函数 UpdateEnergySource() 中被调用, UpdateEnergySource() 的函数体结尾通过调用 Simulator::Schedule() 函数来设定 UpdateEnergySource() 函数的下一次被调用的时间, 达到定时执行的效果。在 ns3::LiIonEnergySource::CalculateRemainingEnergy() 中仅仅是将剩余能量减去功率与调用间隔的乘积, 因为该函数被等时间间隔调用, 可以将其改造成迭代计算。图 3 是修改后的函数 ns3::LiIonKiBaM::CalculateRemainingEnergy() 主要部分的伪代码和注释。

```

1 I2 ← CalculateTotalCurrent()           #获取当前电流
2 duration ← Simulator::Now() - m_lastUpdateTime #获取迭代步长
3
4 m_y1 ← m_y1b + ...                     #按照式 (15) 进行计算本次迭代结果
5 m_y2 ← m_y2b + ...                     #公式较长在此省略
6 energyToDecreaseJ ← (m_y1b - m_y1) * m_supplyVoltageV #计算从上次迭代到本次迭代时间内消耗的能量
7
8 如果 energyToDecreaseJ > m_remainingEnergyJ #如果电池耗尽
9     m_remainingEnergyJ ← 0              #
10    m_y1 ← 0                             #
11 否则                                     #否则更新电量信息
12    m_remainingEnergyJ ← m_remainingEnergyJ - energyToDecreaseJ #
13    m_drainedCapacity ← m_drainedCapacity - ((m_y1b - m_y1) / 3600) / m_c #
14
15 m_supplyVoltageV ← GetVoltage(I2)      #计算当前输出电压
16 I1 ← I2                                 #将本次迭代结果赋值给上次迭代
17 m_y1b ← m_y1                            #
18 m_y2b ← m_y2                            #

```

图 3 函数 LiIonKiBaM::CalculateRemainingEnergy() 主体的伪代码与解释

Fig.3 Pseudocode and explanation of the main part of function LiIonKiBaM::CalculateRemainingEnergy()

简便起见, 本文没有编写该能量模型的 Helper 类, 因此在给设备指定能量模型时, 需要先用 ns3::CreateObject<T>() 模版函数分别创建 ns3::EnergySourceContainer 类和 ns3::LiIonKiBaM 类的实例, 然后调用 EnergySourceContainer::Add() 函数, 将指向 ns3::LiIonKiBaM 实例的指针添加到 ns3::EnergySourceContainer 的实例中。

3 仿真与实验评价

3.1 数值解法的仿真评价

用仿真的方式来评价本文 2.2 节提出的数值解法性能, 仿真平台是 Matlab R2018a 64 位版。由于仅仿真数值解法的性能, 为了排除模型因素的影响, 采用提出的数值解法对原始的 KiBaM 模型进行求解。电池电压 3.7 V, 电池容量 200 A·s, 放电状态是 10 s, 往复周期 2 A 电流间歇放电。选取 Matlab 提供的迭代步长 0.001 s 的 ode 求解器作为对比基准, 这是在无法得到解析解的条件下所能达到的最好精度的微分方程数值解^[21]。作为对比, 迭代步长 0.1 s 的四阶经典 Runge-Kutta 解法和迭代步长 1 s 的本文提出的解法进行比较, 仿真结果如图 4 所示。

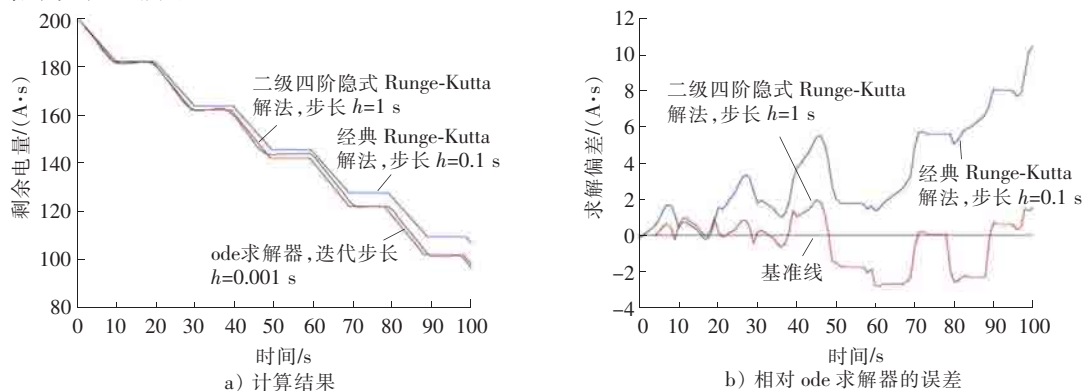


图 4 3 种微分方程数值解法的计算结果

Fig.4 Calculation results of three kinds of numerical solutions of differential equations

从仿真结果可以看出,即使求解精度提高了 10 倍,经典 Runge-Kutta 解法的误差仍然比二级四阶隐式 Runge-Kutta 解法要大。提出的解法以经典解法 1/10 的求解精度,得到了标准差仅为经典解法 32.27% 的准确结果,最大偏差仅为经典解法 26.98% 的准确结果。且经典方法存在严重的累积误差问题,随着仿真时间的推移,累积误差会越来越大,最终得到发散的结果。而本文 2.2 节提出的解法具有更小的计算开销和更高的求解精度,如表 2 所示。

表 2 两种解法相对 ode 求解器的偏差统计值

Tab.2 Deviation statistics of two solutions relative to the ode solver

方法	标准差	最大偏差	终点偏差
经典解法	4.301 5	10.434 6	10.434 6
本文提出的解法	1.388 4	2.815 6	1.517 9

3.2 电池模型的实验评价

为了体现模型的泛用性,被测电池为 2 块不同的电池,标称电压均为 3.7 V,标称容量均为 600 mA · h,但内阻等其他参数不同。测量仪器为 EBC-A05+ 电池测试仪、安捷伦 34401A 数字万用表,如图 5 所示。



a) 被测电池一

b) 被测电池二

c) EBC-A05+

d) 安捷伦 34401A

图 5 实验设备

Fig.5 Experimental equipment

从电池电量充满的状态开始,分别以恒流 1, 0.5, 0.25, 0.1 A 和间歇 0.1 A 的负载进行实验并记录每时刻的电压值,并在 NS-3 中分别进行上述同条件仿真。由于 NS-3 不具备图形界面,没有绘图功能,所以需要把 NS-3 的输出数据拷贝到 windows 计算机并在 Matlab 中读取,才能绘出电池在放电过程中的实测曲线和仿真曲线。为了评价仿真的准确度,在曲线中等间距取 3 000 个点,定义它们的均方根误差(RMSE)作为评价标准,实验结果如表 3 和图 6 所示。

表 3 放电实验与仿真的偏差统计值

Tab.3 Deviation statistics of discharge experiments and simulations

实验条件	均方根误差/V	最大偏差/V
电池一 1 A 恒流放电	0.050 9	0.088 5
电池一 0.5 A 恒流放电	0.037 7	0.172 7
电池一 0.25 A 恒流放电	0.040 2	0.494 2
电池一 0.1 A 恒流放电	0.049 6	0.520 9
电池一 0.1 A 间歇放电	0.031 9	0.100 0
电池二 0.25 A 恒流放电	0.029 6	0.258 5
电池二 0.5 A 恒流放电	0.030 3	0.158 4
电池二 1 A 恒流放电	0.056 1	0.085 9

由实验结果可知,该电池模型能够精准地描述任意电池在使用过程中的状态,平均误差 0.04~0.05 V。最大偏差出现在电池即将耗尽阶段,但在实际使用时,节点不会将电量耗尽,而是会在即将耗尽电量时退出网络,所以这种较大误差的情况实际上是没有影响的,进一步说明本文提出的电池模型的准确度较高。

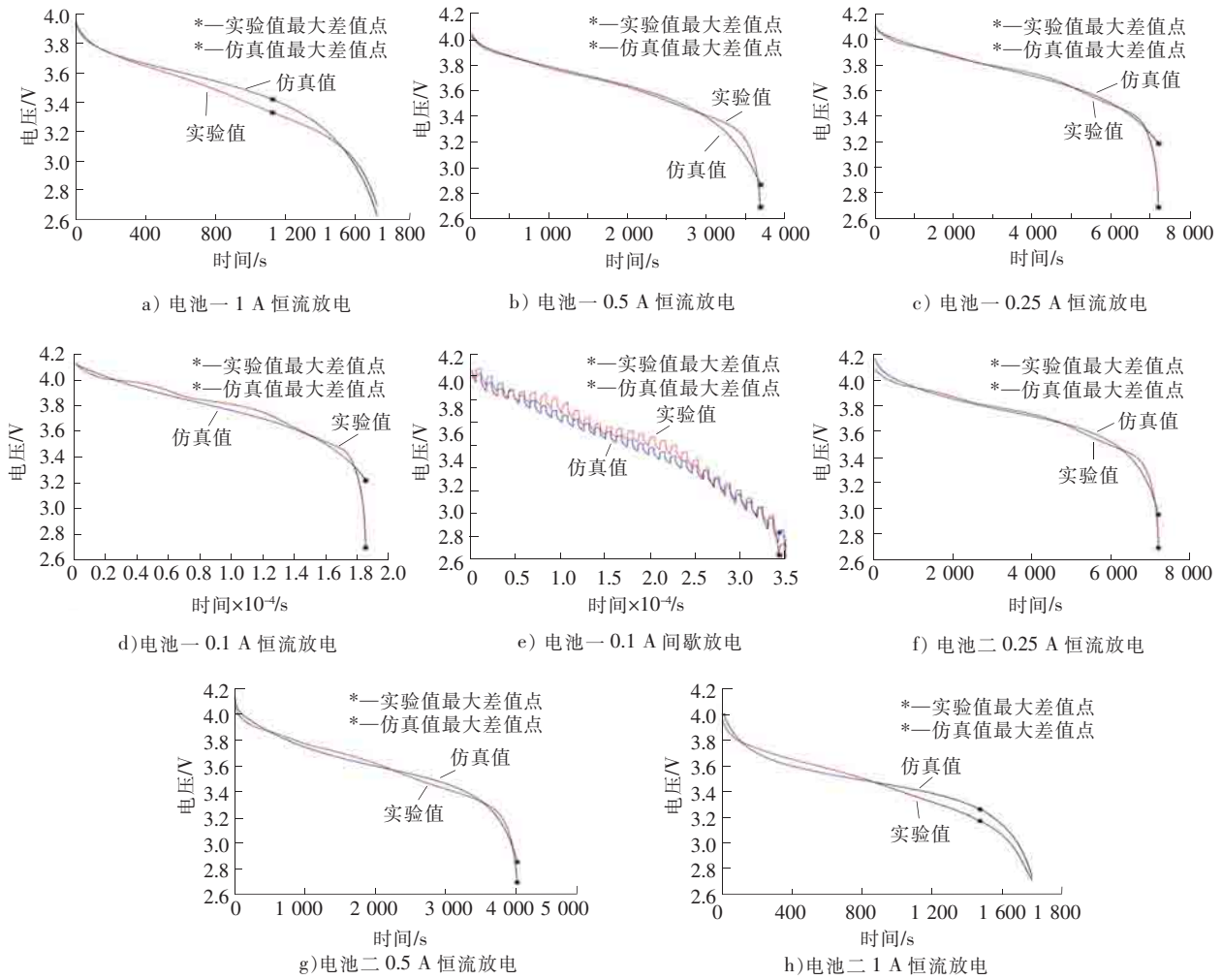


图 6 实验结果与仿真结果在各实验条件下的对比

Fig.6 Comparison of experimental results and simulation results under various experimental conditions

4 结 语

笔者提出的电池模型及其数值解法解决了在 NS-3 中无线传感器网络和移动 Ad-hoc 网络仿真过程中对节点能耗估计不准确的问题,其有效性和准确性得到了实验的证明和支持。它不仅适用于 NS-3,同样也适用于其他网络模拟器,为无线传感器网络和移动 Ad-hoc 网络的研究与发展提供了良好的技术支持和保障。但该电池模型没有考虑电池的寿命衰减,忽略了电池随长期使用而出现的老化现象,需针对此问题进一步深入研究。

参考文献/References:

[1] 茹新宇,刘渊.网络仿真器 NS3 的剖析与探究[J].计算机技术与发展,2018,28(3):72-77.
 RU Xinyu, LIU Yuan. Analysis and research on network simulator 3[J]. Computer Technology and Development, 2018, 28(3): 72-77.

[2] 顾洁,朱宗卫,徐友庆,等. NS-3 仿真环境中 802.11 服务区分机制的研究与实现[J]. 计算机工程,2018:0051676.
 GU Jie, ZHU Zongwei, XU Youqing, et al. Research on the service differentiation mechanisms in wireless local area networks by using NS-3[J]. Computer Engineering, 2018: 0051676.

[3] CHE Zhiping, JIN Rencheng, ZHU Ming, et al. Battery optimal scheduling based on energy balance in wireless sensor networks[J]. IET Wireless Sensor Systems, 2015, 5(6):277-282.

[4] 王震. 基于双能源的高效节能 WSN 节点设计[D]. 大连:大连理工大学, 2015.

- WANG Zhen. The Design of Efficient Energy-Saving WSN Node Based on Dual-Energy[D]. Dalian: Dalian University of Technology, 2015.
- [5] LAHIRI K, RAGHUNATHAN A, DEY S, et al. Battery-driven system design: A new frontier in low power design[C]// International Conference on Design Automation Conference. [S.l.]: IEEE, 2002:994932.
- [6] 许参,李杰,王超. 一种高效的多电池系统放电方法[J]. 电路与系统学报, 2005, 10(5): 144-148.
XU Shen, LI Jie, WANG Chao. A effective multiple battery system discharge schedule[J]. Journal of Circuits and Systems, 2005, 10(5): 144-148.
- [7] 鲍曜. 一种低功耗网络控制收发系统的设计与研究[D]. 杭州:浙江大学, 2017.
BAO Yao. The Design and Research of A Low-power Network Control Mail System[D]. Hangzhou: Zhejiang University, 2017.
- [8] TREMBLAY O, DESSAINT L A, DEKKICHE A I. A generic battery model for the dynamic simulation of hybrid electric vehicles[C]// Vehicle Power and Propulsion Conference. [S.l.]: IEEE, 2007:4544139.
- [9] RAKHMATOV D N, VRUDHULA S B K. An analytical high-level battery model for use in energy management of portable electronic systems[C]// IEEE/ACM International Conference on Computer Aided Design. [S.l.]: IEEE, 2001:968687.
- [10] RAKHMATOV D N, VRUDHULA S B K. Energy management for battery-powered embedded systems[J]. ACM Transactions on Embedded Computing Systems, 2003, 2(3):277-324.
- [11] 张彦琴, 刘汉雨, 卢明哲. 锂离子电池统一充电模型研究[J]. 电源技术, 2015, 39(2):283-284.
ZHANG Yanqin, LIU Hanyu, LU Mingzhe. Modeling of unified charging of lithium-ion batteries[J]. Chinese Journal of Power Sources, 2015, 39(2): 283-284.
- [12] JONGERDEN M, HAVERKORT B R. Which battery model to use? [J]. IET Software, 2010, 3(6):445-457.
- [13] MANWELL J F, MCGOWAN J G. Lead acid battery storage model for hybrid energy systems[J]. Solar Energy, 1993, 50(5):399-405.
- [14] RAO V, SINGHAL G, KUMAR A, et al. Stochastic battery model for embedded systems[C]// Proceedings of the IEEE International Conference on VLSI Design. [S.l.]: IEEE, 2005:61.
- [15] 王闯. 锂电池 SOC 估算方法及串联电池组一致性研究[D]. 沈阳:东北大学, 2014.
WANG Chuang. Lithium-ion Battery SOC Estimation Method and the Equilibrium of Series Battery Pack Research[D]. Shenyang: Northeastern University, 2014.
- [16] 余文正. 动力电池模型分析及其快速充放电策略研究[D]. 成都:电子科技大学, 2013.
YU Wenzheng. Power Battery Model Analysis and Its Fast Charge and Discharge Strategy Study[D]. Chengdu: University of Electronic Science and Technology of China, 2013.
- [17] 倪素环, 杨雪. 冲击振动单边单质量破碎系统的非线性动力学分析[J]. 河北科技大学学报, 2017, 38(5):469-473.
NI Suhuan, YANG Xue. Nonlinear dynamic analysis of single-sided and single-mass crushing system under impact and vibration[J]. Journal of Hebei University of Science and Technology, 2017, 38(5):469-473.
- [18] 张宏伟. 计算机科学计算[M]. 北京:高等教育出版社, 2013.
- [19] CHEN D J L. The efficiency of Singly-implicit Runge-Kutta methods for stiff differential equations[J]. Numerical Algorithms, 2014, 65(3):533-554.
- [20] BUTCHER J C. Implicit runge-kutta processes[J]. Mathematics of Computation, 1964, 18:50-64.
- [21] 马泽涛, 文鹏, 施琳达, 等. 基于 Matlab 常微分方程数值解的分析与比较[J]. 大学教育, 2017(12):50-52.