

# 基于改进型一致性算法的储能SOC均衡控制方法

方强,程志江,杨涵棣

(新疆大学 可再生能源发电与并网控制教育部工程研究中心,新疆 乌鲁木齐 830017)

**摘要:**针对光伏、风机等具有不受控性和波动性的分布式发电系统(DGS)大规模并入电网造成的储能单元出力不均匀和荷电状态(SOC)不均衡的问题,基于多代理系统提出了一种基于梯度补偿的改进型一致性算法。将DGS、退役动力电池储能系统(DPBESS)和电网共同组建成多代理微电网系统。在改进一致性算法基础上设计了SOC均衡控制方法,加快了无中心控制器的储能系统SOC均衡速度,保证了微电网内能量的供需平衡。通过仿真验证了所提控制方法的有效性。

**关键词:**多代理系统;SOC均衡控制;微电网;能量供需平衡;改进一致性算法

**中图分类号:**TM46 **文献标识码:**A **DOI:**10.19457/j.1001-2095.dqed24191

## Energy Storage SOC Equalization Control Method Based on Improved Consistency Algorithm

FANG Qiang, CHENG Zhijiang, YANG Handi

(Engineering Research Center of Renewable Energy Power Generation and Grid Connection Control, Ministry of Education, Xinjiang University, Urumqi 830017, Xinjiang, China)

**Abstract:** For the problem of uneven power output and unbalanced state of charge (SOC) of energy storage units caused by large-scale integration of distributed generation systems (DGS), such as photovoltaic and wind turbine, into the grid, an improved consistency algorithm based on gradient compensation was proposed based on a multi-agent system. The DGS, decommissioned power battery energy storage system (DPBESS) and the grid were jointly formed into a multi-agent microgrid system. The SOC equalization control method was designed based on the improved consistency algorithm, which speed up the SOC equalization of the energy storage system without central controller and ensured the balance of energy supply and demand in the microgrid. The effectiveness of the proposed control method was verified by simulation.

**Key words:** multi-agent system; state of charge (SOC) equalization control; microgrid; energy supply and demand balance; improved consistency algorithm

随着“碳达峰”、“碳中和”目标的提出,加速能源清洁低碳转型、积极应对气候变化成为全球共同性议题<sup>[1]</sup>。由于可再生能源的不稳定性给电网和用户侧带来了不利影响,通过与储能单元结合,构成混合能源型微电网,可实现能量转移,降低负载高峰,填补发电低谷,减少对电网的影响。退役动力电池储能系统(decommissioned power battery energy storage system, DPBESS)之间的荷电状态(SOC)和容量有所差异,分布式发电系统

(distributed generation systems, DGS)出力的不确定性和负载的多样性等因素会对微电网内的能量供需平衡带来影响。因此,一种合适的能量均衡控制方案和微电网控制策略显得尤为重要。

传统的微电网分层控制方法单一,无法对微电网的各个参数进行优化调节,其能量控制方式主要有:集中式控制、分层控制、分布式控制<sup>[2]</sup>。文献[3]利用集中式控制设置了一个主控制器,能汇集全局的信息,响应速度快。但其通信量大,

**基金项目:**新疆维吾尔自治区自然科学基金(2021D01C046);新疆维吾尔自治区重点实验室开放课题(2021D04011)

**作者简介:**方强(1996—),男,硕士研究生,Email:fangqiang578@163.com

**通讯作者:**程志江(1977—),男,博士,副教授,Email:67078267@qq.com

对安全性、可靠性要求高,一旦主控制器发生通信故障,系统将无法运行。文献[4]通过分层控制将微电网的优化控制分成两层,达到了对每一层的分布式控制的目的,但其控制策略比较复杂,层次过多时系统会产生延迟,影响系统运行效率。多代理分布式控制具有自主决策和协调控制等优点,相比于传统的集中控制和分层控制,其通信成本降低,系统效率提升明显,逐步应用到了微电网中。文献[5]提出了基于多代理系统的分布式控制策略,提高了系统的效率和可靠性,降低了通信成本,但控制方法较复杂。以上方案考虑了能量供需平衡关系但没有考虑储能单元间的SOC均衡控制。

目前微电网中主要讨论的是对分布式储能单元间的SOC均衡控制。文献[6]把模块化多电平变换器用在储能单元的控制模块,通过改变输出单元占空比来保证SOC均衡。文献[7]在下垂系数的取值上使用幂指数来加快SOC均衡的速度。文献[8]设计了模糊控制器,依据当前的SOC值来计算储能单元所需的下垂系数。上述文献都是采用集中式控制,通信较复杂,而且没有考虑不同容量的储能。文献[9]在传统一致性算法的基础上提出了SOC均衡方案,欠缺均衡速度对系统的影响。

本文采用的分布式多代理通信网络,只需要相邻代理间的通信来获取储能单元的SOC,减少了通信次数,并基于此研究了系统内能量供需平衡问题,形成无约束双向受控制的多端能源流动模式;同时根据函数在某点的梯度方向是该点函数值变化最快的方向,在固有的一致性算法中引入了一种状态变量的梯度项,得到改进型基于梯度补偿的一致性算法,和传统一致性算法的收敛速度做了比较;对不同容量的DPBESS进行基于PQ控制的SOC均衡策略的研究,最后通过仿真验证了所提策略的有效性。

## 1 多代理微电网框架设计

### 1.1 多代理网络的通信结构

在多代理系统中,通常用有向图或无向图来描述系统的通信网络。图1是基本的有向图,把图中的一个节点视为一个多代理,把图中的每条边视为相邻代理之间的信息交流的通路。定义  $G = (V, E, A)$  为有向图来描述多代理的网络拓扑,其中,  $V$  表示每个代理,  $E \subseteq V \times V$  表示边的集

合,  $A$  为邻接矩阵,相邻代理之间的关系用  $A$  中的元素  $a_{ij}$  表示,即  $A=[a_{ij}]_{n \times n}$ 。  $v_i, v_j$  代表每条边的元素,若  $(v_i, v_j) \in E$ , 则  $a_{ij} \neq 0$ , 否则  $a_{ij} = 0$ 。节点  $i$  处的类型矩阵定义为  $F=[f_{ii}]_{n \times n}$ , 是一个对角矩阵,  $f_{ii}$  表示节点  $i$  处连接的代理的类型。若代理是受控的, 则  $f_{ii}=1$ , 否则  $f_{ii}=0$ 。出度矩阵  $O=[o_{ii}]_{n \times n}$  也是一个对角矩阵, 在这里表示为某个节点处对外引出的通信线数<sup>[10]</sup>。

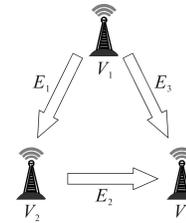


图1 基本有向图

Fig.1 Basic directed graph

### 1.2 多代理微电网网络设计

本文中的系统拓扑如图2所示,其由两台风机发电系统、一台光伏发电系统、两组退役动力电池储能单元以及电网和负载组成。考虑系统实现的具体形式,根据输出功率的特性,风机和光伏易受到自然环境的影响,其输出功率是不受控制的,采用最大功率跟踪控制方案,故将其作为不受控代理,而DPBESS能够稳定控制能量的输入输出,故将其作为受控代理。不受控代理向相邻的受控代理发送自身的功率和本地负载信息,同时受控代理之间也进行双向通信,通过接收的信息进行汇总,根据能量供需平衡原则实时调整输出的功率,进而实现微电网内能量供需平衡关系。电网也作为一个代理,其功能包括两个:一是作为系统与电网通信的枢纽,传递微网中能

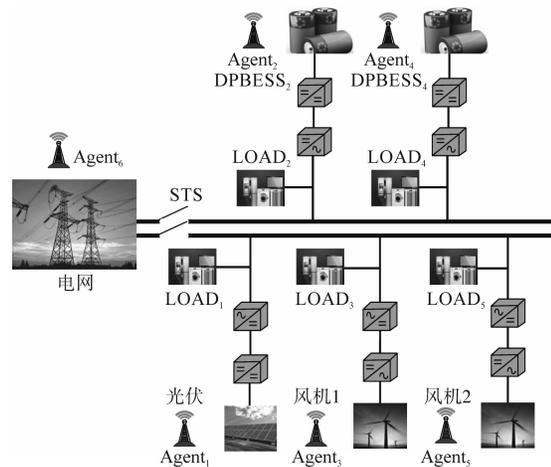


图2 多代理微电网系统结构

Fig.2 Structure diagram of multi-agent microgrid system

量的需求和对系统频率进行支撑;二是对微电网的保护作用,当电网出现故障时,及时切断与系统的联系,减少故障给系统带来的不利因素。各个代理通过功率变换装置并联在电力网络中给本地的负载提供能量。

传统的微电网有并网运行和孤岛运行两种模式,其主要是由静态转换开关STS的状态决定的。在本文中,微电网一直处于并网状态,微电网内的各个子系统可以和电网进行能量的输送,以实现系统内能量的供需平衡。

### 1.3 多代理微电网能量供需方案

在微电网的母线节点处部署信息采集和通信装置组成代理决策系统,把采集来的信息进行汇总和分析,再根据自身控制目标,做出相应的决策,自发地对周围环境的变化做出响应或与相邻的代理进行信息交换、协作,共同应对环境的变化,进而实现对被控对象的信息交换和控制。受微电网中接入负载的影响,能量供需方案有以下三种:

1)不受控代理输出的能量能够维持负载的运行,这时只需调节它们的输出即可。不受控代理的逆变器采用恒功率控制,使其输出的功率与负载所需功率相等。

2)不受控代理产出的能量不能够维持负载的运行,受控代理能够满足功率缺额。这时不受控代理的逆变器采用恒功率控制,受控代理采用PQ控制,两者共同维持负载的需求。其能量供需关系如下:

$$\begin{cases} \sum [P^*(k+\tau) \cdot F] + \sum [P(k) \cdot (E-F)] = \sum P_{LOAD}(k) \\ \sum [Q^*(k+\tau) \cdot F] + \sum [Q(k) \cdot (E-F)] = \sum Q_{LOAD}(k) \end{cases} \quad (1)$$

式中: $P(k), Q(k)$ 分别为DGS当前时刻输出的有功功率和无功功率; $P^*(k+\tau), Q^*(k+\tau)$ 分别为下一时刻DPBESS输出的有功功率和无功功率; $P_{LOAD}(k), Q_{LOAD}(k)$ 分别为负载的有功和无功功率。

不受控代理输出的有功功率和无功率加上受控代理下一时刻输出的有功功率和无功率等于该时刻微电网内所有负载需求的有功功率和无功率。由于不受控代理的输出不受控,实际上是由受控代理来补偿功率的差额。为了满足上述的能量供需关系,根据多代理通信网络推导出的受控代理下一时刻的功率参考指令如下:

$$\begin{cases} P^*(k+\tau) \cdot F = P(k) \cdot F + (A^T + F)(O + F)^{-1} \cdot [P_{LOAD}(k) - P(k)] \\ Q^*(k+\tau) \cdot F = Q(k) \cdot F + (A^T + F)(O + F)^{-1} \cdot [Q_{LOAD}(k) - Q(k)] \end{cases} \quad (2)$$

3)不受控代理和受控代理均以最大功率运行时仍满足不了负载的功率需求,这时仅凭微电网自身已无法维持稳定运行,需要电网参与其中能量的调度来维持稳定,功率差额由电网补偿。当受控代理的输出功率达到上限时,继续保持最大功率输出,同时把参考功率和当前功率的差额作为下一时刻的电网调度指令发送给电网代理,电网代理对电网进行调度,使微电网内的能量再次平衡,控制原理如图3所示。此时,主电网的输出功率如下式:

$$\begin{cases} P_{grid}^*(k+\tau) = \sum \{ P(k) \cdot F + (A^T + F) \cdot (O + F)^{-1} \cdot [P_{LOAD}(k) - P(k)] \} \cdot F \\ Q_{grid}^*(k+\tau) = \sum \{ Q(k) \cdot F + (A^T + F) \cdot (O + F)^{-1} \cdot [Q_{LOAD}(k) - Q(k)] \} \cdot F \end{cases} \quad (3)$$

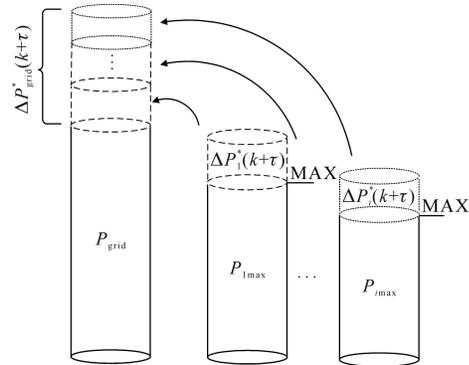


图3 电网与代理间能量的均衡控制

Fig.3 Collaborative control of energy between power grid and agent

## 2 基于梯度补偿的改进一致性算法SOC均衡控制策略

### 2.1 SOC均衡控制策略

本文应用的退役动力电池梯次利用系统不考虑退役动力电池的检测、筛选和成组技术,仅考虑直接使用退役动力电池梯次利用设计构成的电池管理系统(battery management system, BMS)电池包。不受控代理逆变器采用最大功率跟踪控制,以实现可再生能源的最大程度利用。受控代理逆变器的输出功率采用了在传统PQ控制中引入SOC均衡因子的闭环控制策略来进行调

节,控制框图如图4所示。通过调节电流环的参考电流和PWM信号来产生逆变器的驱动信号,进而调整逆变器有功功率的输出来实现SOC均衡。

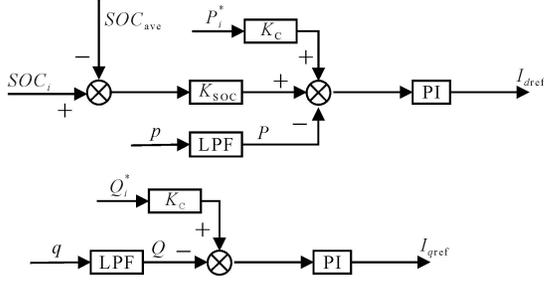


图4 DPBESS逆变器的控制策略框图  
Fig.4 Control strategy block diagram of DPBESS inverter

图4中, $p, q$ 为逆变器输出的有功功率和无功功率; $P_i^*, Q_i^*$ 为维持系统能量供需平衡的DPBESS逆变器输出功率; $I_{dref}$ 和 $I_{qref}$ 为 $d, q$ 轴的参考电流; $K_{soc}$ 为SOC均衡的权重系数; $SOC_i$ 与 $SOC_{ave}$ 的差值为SOC均衡因子,当均衡因子变为0时不影响系统能量供需平衡状态。图4中的各个SOC的估算如下式所示:

$$SOC = SOC_0 - \frac{\int P^* dt}{V_{DC} C_e} \quad (4)$$

式中: $SOC_0$ 为DPBESS的初始荷电状态值; $V_{DC}$ 为逆变器直流侧的电压; $P^*$ 为逆变器输出的有功功率; $C_e$ 为DPBESS的容量值。

SOC均衡的原理如下所述:当假设在第 $n$ 个时刻2台DPBESS的SOC关系为 $SOC_{1,n} > SOC_{2,n}$ ,由式(4)可得此时刻DPBESS的输出功率关系为 $P_{1,n} < P_{2,n}$ 。从图4可知,当 $SOC_{1,n}$ 与 $SOC_{ave}$ 差值大于零时,通过均衡因子的权重系数加入到参考功率的补偿中,在第 $n+1$ 时刻输出的有功功率 $P_{1,n+1} > P_{2,n+1}$ ,由式(4)计算出SOC的变化量关系为 $\Delta SOC_{1,n} > \Delta SOC_{2,n}$ ,说明SOC初始值大的DPBESS释放的有功功率高,SOC初始值小的DPBESS释放的有功功率低,再经过几个时刻这样的调整就能使SOC均衡。当SOC均衡后,SOC均衡因子变为0,不会改变系统的能量供需平衡关系。

### 2.2 基于梯度补偿的改进一致性算法

要得到各个DPBESS的SOC平均值,需要各个受控代理之间进行通信。假设系统里有 $m$ 个代理, $x_i \in \mathbf{R}$ 表示代理 $i$ 的一个状态变量,相邻代理间的状态信息的交换过程可以视为一个线性系统,由传统的一致性算法可得:

$$x_i(k+1) = x_i(k) + \sum_{j \in N_i} a_{ij} [x_j(k) - x_i(k)] \quad (5)$$

其中

$$i = 1, 2, \dots, n \quad j = 1, 2, \dots, m$$

式中: $x_i(k), x_i(k+1)$ 分别为在迭代过程中的第 $k$ 时刻和第 $k+1$ 时刻的状态变量; $x_j(k)$ 为代理 $j$ 在第 $k$ 时刻的状态变量; $a_{ij}$ 为相邻代理 $i$ 和 $j$ 之间传递信息的权重系数,若它们之间有通信,则 $0 < a_{ij} < 1$ ,否则 $a_{ij} = 0$ ;  $N_i$ 为所有与代理 $i$ 相邻的代理的集合,  $N_i = \{j | (v_i, v_j) \in \mathcal{E}\}$ ,  $\mathcal{E}$ 为一个很小的实数。

为了降低通信成本,加快平均一致性收敛速度,仿效最优潮流的简化梯度算法,函数在某点的梯度方向是该点函数值变化最快的方向,沿着函数在该点负梯度方向前进时,函数值变化最快。所以在固有的一致性算法中引入状态变量的梯度项 $\nabla_{i,k}[\zeta_i(k)]$ ,得到基于梯度补偿的改进一致性算法,如下式:

$$\begin{cases} \zeta_i(k) = x_i(k) + \sum_{j \in N_i} a_{ij} [x_j(k) - x_i(k)] \\ x_i(k+1) = \zeta_i(k) + \{\lambda_i \nabla_{i,k}[\zeta_i(k)]\} \end{cases} \quad (6)$$

式中: $\zeta_i(k)$ 为迭代中间变量; $\lambda_i$ 为步长因子。

算法的收敛判据如下式:

$$|x_i(k+1) - x_i(k)| < \varepsilon \quad (7)$$

算法流程如图5所示。首先初始化系统信息, $k=0$ ,代理1采集本地的状态信息和接收相邻代理传递的状态信息,通过这些信息来确定此次迭代过程中变量的梯度值来执行改进的一致性算法,从而计算代理1梯度下降的间隔,进行收敛性判断,即当此次迭代值与上次迭代的差值在 $\varepsilon$ 的范围内时,变量收敛,跳出循环并得到一致性收敛的值。

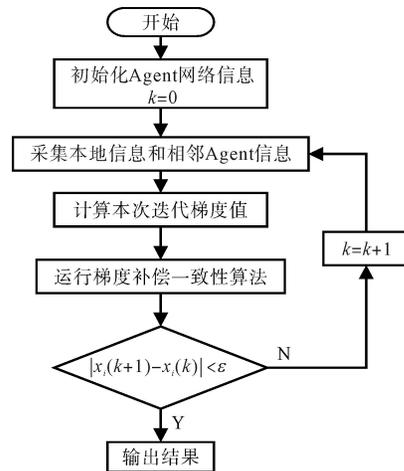


图5 基于梯度补偿的改进一致性算法流程图  
Fig.5 Flow chart of improved consistency algorithm based on gradient compensation

为了验证上述算法的有效性,分别对传统一致性算法和基于梯度补偿的一致性算法的收敛性进行验证,采样的最小时间间隔为0.1 s,验证结果如图6和图7所示。

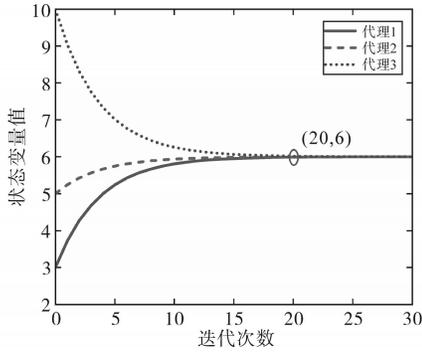


图6 传统一致性算法

Fig.6 Traditional consistency algorithm

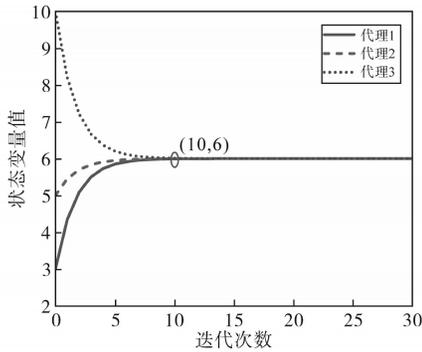


图7 基于梯度补偿的一致性算法

Fig.7 Consistency algorithm based on gradient compensation

由图6和图7可知,在相同的初始状态下,传统一致性算法迭代将近20次时各变量才收敛到平均值附近,而基于梯度补偿的一致性算法经过10次左右就达到收敛状态。说明基于梯度补偿的改进一致性算法的收敛速度更快。

### 3 仿真验证

根据1.1节中对多代理网络的定义,本文多代理微电网网络中的邻接矩阵 $A$ 、类型矩阵 $F$ 、出度矩阵 $O$ 分别为

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (8)$$

$$F = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (9)$$

$$O = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

根据式(2)、式(8)~式(10)推导出DPBESS的参考有功功率和无功功率为

$$\begin{cases} P_2^*(k+\tau) = P_{LOAD1} + \frac{1}{2}P_{LOAD2} + \frac{1}{2}P_{LOAD3} + \frac{1}{2}P_{LOAD4} - \\ P_1 - \frac{1}{2}P_2 - \frac{1}{2}P_3 + \frac{1}{2}P_4 \\ Q_2^*(k+\tau) = Q_{LOAD1} + \frac{1}{2}Q_{LOAD2} + \frac{1}{2}Q_{LOAD3} + \frac{1}{2}Q_{LOAD4} - \\ Q_1 - \frac{1}{2}Q_2 - \frac{1}{2}Q_3 + \frac{1}{2}Q_4 \end{cases} \quad (11)$$

$$\begin{cases} P_4^*(k+\tau) = \frac{1}{2}P_{LOAD2} + \frac{1}{2}P_{LOAD3} + \frac{1}{2}P_{LOAD4} + \frac{1}{2}P_{LOAD5} + \\ \frac{1}{2}P_2 - \frac{1}{2}P_3 - \frac{1}{2}P_4 - P_5 \\ Q_4^*(k+\tau) = \frac{1}{2}Q_{LOAD2} + \frac{1}{2}Q_{LOAD3} + \frac{1}{2}Q_{LOAD4} + \frac{1}{2}Q_{LOAD5} + \\ \frac{1}{2}Q_2 - \frac{1}{2}Q_3 - \frac{1}{2}Q_4 - Q_5 \end{cases} \quad (12)$$

为了验证上述方案的有效性,在PLECS仿真软件平台搭建了如图8所示的拓扑结构并网微电网系统,设计了变负载运行、DGS输出功率变化、DGS输出中断等工况对所提控制策略的有效性进行验证分析。

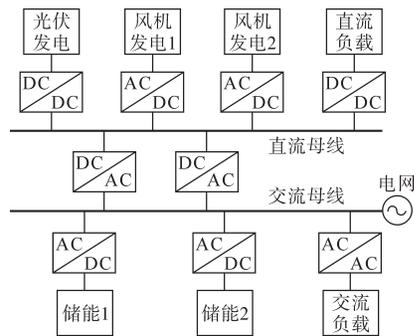


图8 并网微电网拓扑结构图

Fig.8 Topology of grid connected microgrid

仿真模型中,系统参数为:直流侧电压 $V_{dc}=800\text{ V}$ ,并网逆变器输出线电压 $V_{PCC}=380\text{ V}$ ,给定频率 $f=50\text{ Hz}$ ,DPBESS<sub>2</sub>初始 $SOC_2=0.80$ ,DPBESS<sub>4</sub>初始 $SOC_4=0.79$ ,DPBESS<sub>2</sub>容量:DPBESS<sub>4</sub>容量=1:2, $SOC$ 均衡因子系数 $K_{soc}=10^6$ 。代理参数如表1所示。由于仿真条件的限制,设储能单元的 $SOC$ 值相对较小,使其反应速度加快,更加便于观察。

表1 代理参数  
Tab.1 Agent parameters

代理	有功功率/kW	无功功率/kvar
代理1(光伏)	50	20
代理2(DPBESS <sub>2</sub> )	40	20
代理3(风机1)	50	20
代理4(DPBESS <sub>4</sub> )	20	10
代理5(风机2)	40	15

3.1 变负载运行

在此工况下,不受控代理DGS工作在理想环境下,以MPPT模式工作,输出功率恒定,受控代理DPBESS以上述改进型SOC均衡控制策略运行,这时负载变化运行,各微源功率输出波形如图9所示,SOC的验证结果如图10所示。

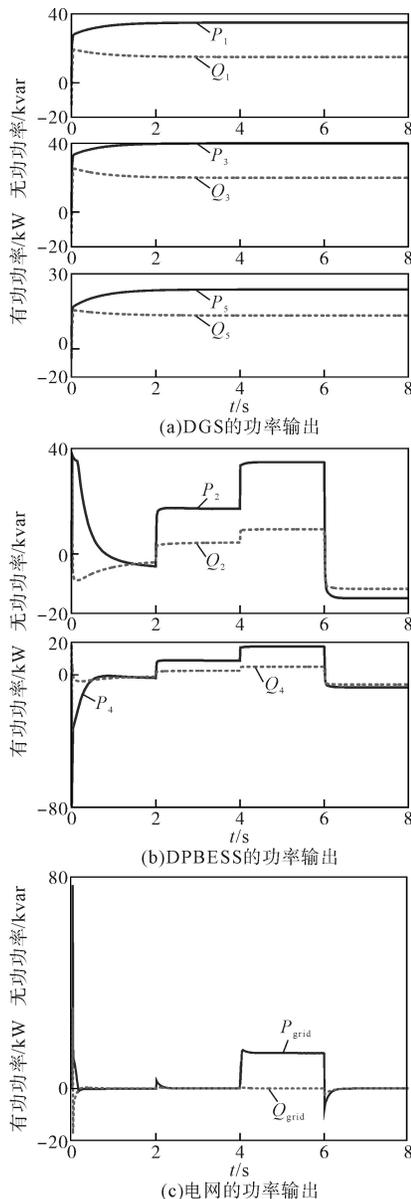


图9 负载改变时的各微源功率输出波形图  
Fig.9 Power output waveforms of each source when the load changes

从图9中分析可知,在2s以前,负载需求的能量可以由不受控代理DGS提供,这时电网输出的功率为零,DPBESS处于充电状态,充电功率按容量分配,SOC高的充电功率低;2s时,负载需求功率加大,此时不受控代理DGS和受控代理DPBESS提供的能量能满足负载需求,DPBESS处在放电状态,放电功率按容量分配,SOC低的放电功率低;4s时负载需求功率继续加大,此时不受控代理DGS和受控代理DPBESS提供的能量已不能满足负载需求,电网依照式(3)开始参与到系统能量的调度;6s时负载功率进一步减小,电网恢复零出力的状态,DPBESS恢复充电状态,整个系统供需侧能量平衡。

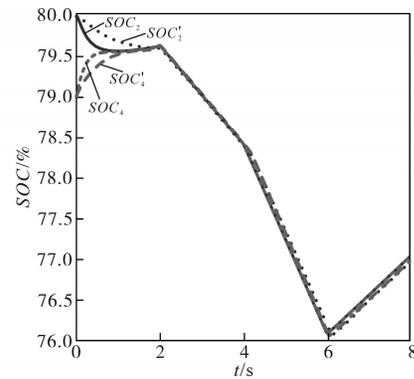


图10 负载改变时的荷电状态

Fig.10 State of charge when the load changes

图10中,SOC<sub>2</sub>和SOC<sub>4</sub>是传统一致性算法的仿真波形。从图中可知,传统的一致性算法下,两个DPBESS的SOC约在2s达到均衡状态。而在基于梯度补偿的改进一致性算法下,SOC在1s左右就达到均衡状态。相比之下后者的SOC均衡效果要更优。在负载变化的过程中,SOC也能继续保持均衡状态。

3.2 天气变化和变负载运行

在实际应用中,受到自然环境等因素的影响,风机和光伏发电会随之波动,在极端天气环境中,变化更剧烈。基于此工况模拟了不受控代理的功率输出,负载变化同3.1节,各微源功率输出波形如图11所示,SOC的验证结果如图12所示。

由图11分析可知,极端环境下的不受控代理DGS的输出功率在变化,此时的电网和DPBESS的输出功率也随之变化;2s之前DGS的输出功率较3.1节略有增加,SOC上升速度加快;3s—6s间,三台DGS的输出功率都发生了变化,总功率相对减少,DPBESS的输出功率增加,达到额定功

率后保持不变,电网出力也随之增加,系统保持能量供需平衡。

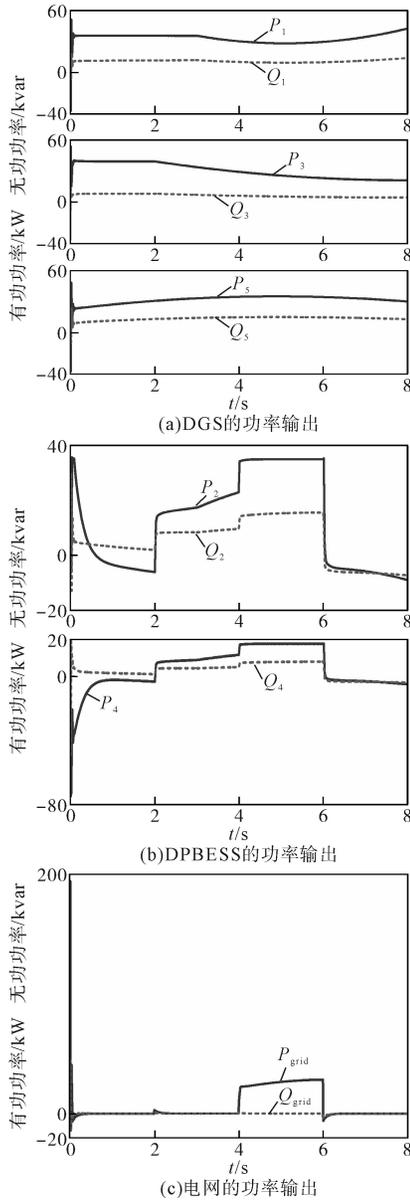


图11 天气和负载同时改变时各微源功率输出波形图

Fig.11 Power output waveforms of each source when weather and load change at the same time

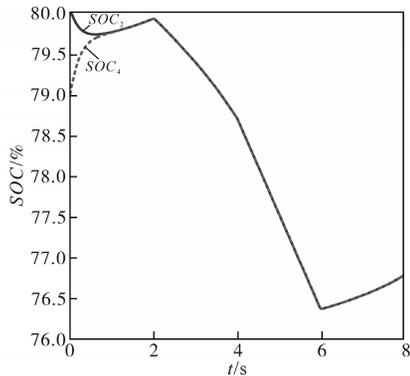


图12 天气和负载同时改变时的荷电状态

Fig.12 State of charge when weather and load change at the same time

分析图12知,相比于3.1节中,0 s—2 s内DGS的输出功率大,DPBESS<sub>4</sub>充电功率变大,SOC达到均衡后两个DPBESS继续保持充电状;2 s—6 s后负载加大,开始放电;6 s后负载变小,两个DPBESS恢复充电状态。整个过程中SOC总体下降趋缓,均衡控制效果良好。

### 3.3 即插即用性

在实际的工程应用中会出现某台或多台DGS故障的极端情况,从而退出微电网系统。基于此工况模拟了DGS的功率输出情况,并伴随着3.2节中的负载和DGS的功率变化,各微源功率输出波形如图13所示,SOC的验证结果如图14所示。

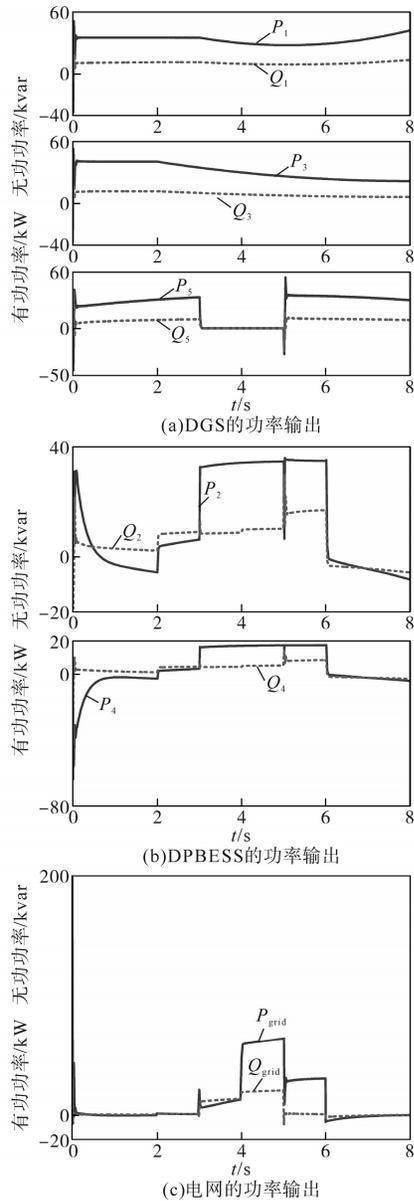


图13 即插即用性验证中各微源功率输出波形图

Fig.13 Power output waveforms of each micro source in plug and play verification