A Cloud-Based Energy Management Framework with Local Redundancy Control Capability

Zhikang Li
UM-SJTU Joint Institute
Shanghai Jiao Tong University
Shanghai, P. R. China
zhikang.li@sjtu.edu.cn

Amro Alsabbagh
Polytechnique Montreal
University of Montreal
Montreal, Canada
amro.alsabbagh@hotmail.com

Chengbin Ma
UM-SJTU Joint Institute
Shanghai Jiao Tong University
Shanghai, P. R. China
chbma@sjtu.edu.cn

***Abstract*—With the widely use of renewable energy in household smart grid, cloud-based energy management system has been received more and more attention for its ability to store historical data and predict future power. However, there is still limit on reliable and efficient control of household smart grid system under high latency conditions when command from cloud server can not be sent to local controller in real time. This study aims to combine cloud control and local control to realize a cloud energy management framework with local redundancy. Firstly, we analyzed the model of the proposed framework. Then based on this model, we proposed a redundancy control strategy with a two-layer control loop. A switch mechanism between cloud and local control mode was designed to enable system to switch smoothly under different latency conditions. Simulation results show the effectiveness of proposed framework and control strategy in reducing cost and relieving power fluctuation.**

***Index Terms*—smart grid, cloud computing, energy management, redundancy.**

I. Introduction

Smart grid has been widely promoted worldwide, and it is developing towards more efficient and reliable [1]. Combining smart grid with smart home is a typical application. With the proper use of green energy technology like photovoltaic system (PV) and battery energy storage system (BSS) and electric vehicles (EVs), smart grids can significantly improve efficiency and reduce the cost of household electricity services [2].

However, household smart grid system has high uncertainty both in power generation and power demand, which attaches great importance to the efficient energy management of smart grid. These variations can significantly influence the power demand required from power grid, and lead to burden for the operation of grid. Hence, power prediction is necessary for efficient energy management system (EMS) in household smart grid system.

Accurate power prediction requires historical data and high computation power, which is not suitable for local controller that is equipped for household smart grid system. The widely use of cloud computing provides a solution to relieve the burden in local controller [3]. Based on low latency communication technology, real-time power schedule is able to be carried out on cloud terminal. Several references like [4]and [5] that discuss this control architecture show effectiveness in reducing grid power consumption and saving operation cost.

Although cloud-based EMS has several benefits, it has several drawbacks compared to traditional local EMS. Cloud-based EMS heavily relays on the communication between cloud server and local controller, any latency may cause serious problem when all schedule commands are sent from cloud server. As energy system becoming the foundation in modern smart home, the reliability of EMS has dawn an increasing interest. In household smart grid EMS, there are several references that discuss hardware redundancy like in [6] or software redundancy like in [7]. However, there are few references that consider redundancy in control architecture.

Local controller with offline schedule mode provides a solution for reliable and efficient control in cloud-based EMS. In order to combine the advantage of cloud-based EMS and local-based EMS, we apply the concept of redundancy in control framework design and propose a cloud-based EMS framework with local redundancy control capability. The major work is summarized as follow:

- This paper proposes an EMS framework that contains both cloud and local terminals. Cloud sever takes use of its data storage and computation power to do prediction based power schedule, while local controller takes use of its real-time control ability to check power constraints and carry out schedule results. Cloud computing efficiency and local control redundancy are combined in this framework.
- A two-layer control strategy for cloud server and local controller is designed to increase system reliability under different degrees of latency and offline conditions.
- A switch mechanism is designed to help EMS change smoothly among different conditions. Simulation results show the effectiveness of proposed control strategy and switch mechanism.

II. System Framework and Modeling

A. System Framework

The proposed cloud EMS framework mainly includes two parts, namely local terminal and cloud terminal. The framework diagram is shown in Fig. 1.

1) Local terminal: Local terminal usually locates at house and owned by users. Energy router controls the power balance

978-1-7281-6207-2/21/$31.00 ©2021 IEEE

2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS) | 978-1-7281-6207-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICPS49255.2021.9468249

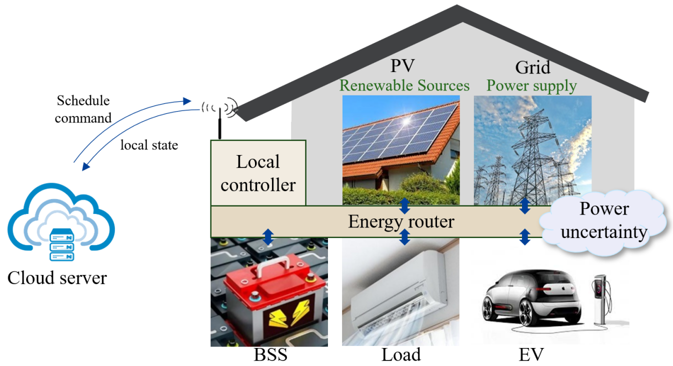


Fig. 1. Cloud energy management framework

among each power component. Without loss of generality, the studied household smart grid system includes grid power supply, PV, BSS, base load, and EV that is charging at home. Energy router is equipped with a local controller that has low computation power considering cost. The local controller only schedules real-time power balance.

2) Cloud terminal: Cloud terminal provides service for several local terminals, and it is equipped with data storage facilities and high computation performance server. Power schedule based on historical data and prediction are carried out on cloud terminal. Real-time power schedule data is sent to local terminal through low latency and high bandwidth technology like 5G.

3) Features of proposed framework: Local controller and cloud server both have advantages and disadvantages. cloud server can make prediction and optimize for long time interval schedule based on historical data [5]. However, its control heavily relies on stable and communication with local controller. Local controller is able to monitor real-time power balance [4]. However, it has limited computation ability and hence can not perform long term energy schedule. The proposed framework combine the advantage of cloud server and local controller, and realize power schedule in two terminals.

The users of household smart grid energy management system can access personal account on cloud server to check private operational data, query scheduling plan and set user behavior based preference. Besides, users can also control facilities manually through energy router.

B. Micro grid Model

In this proposed schedule framework, PV power and load power are assumed to be stochastic variables, BSS charging/discharging power and EV charging power are control variables. The power dynamic of each component is modeled as follows.

1) BSS charging/discharging model: BSS charging model is modeled according to [8], which can be illustrate as follows:

$$SoC_{\tau}^{b} = SoC_{ini}^{b} - \sum\nolimits_{t=0}^{\tau} P_t^b \Delta t / C^b \tag{1}$$

$$SoC_{\min}^{b} \le SoC_{\tau}^{b} \le SoC_{\max}^{b} \ \forall \tau \tag{2}$$

where SoC_{τ}^{b} stands for BSS SoC at time τ, SoC_{ini}^{b} stands for initial SoC, Δt stands for time slots length. C^b stands for battery capacity. SoC_{τ}^{b} satisfies minimum SoC $SoC_{\min}^{b}$ constraint and maximum SoC $SoC_{\max}^{b}$ constraint for any time τ. P_t^b stands for charging/discharging power of BSS, the positive value stands for BSS discharge and the negative value stands for BSS charge. P_t^b satisfies charging and and discharging power limit

$$-P_{\text{ch.,max}}^{b} \le P_t^b \le P_{\text{disch.,max}}^{b} \tag{3}$$

where $P_{\text{ch.,max}}^{b}$ and $P_{\text{disch.,max}}^{b}$ are maximum charging and discharging power respectively.

2) EV charging model: EV charging model can be modeled similar to BSS charging model in [8], the difference is that EV can only be charged when it is connected to charging poles. This process can be modeled as follows:

$$SoC_{\tau}^{ev} = SoC_{ini}^{ev} + \sum\nolimits_{t=0}^{\tau} P_t^{ev} \Delta t / C^{ev} \tag{4}$$

$$SoC_{required}^{ev} \le SoC_{\tau}^{ev} \le SoC_{\max}^{ev} \tag{5}$$

where SoC_{τ}^{ev} stands for EV SoC at time τ, SoC_{ini}^{ev} stands for initial SoC. C^{ev} stands for EV battery capacity. SoC_{τ}^{ev} satisfy minimum required SoC $SoC_{required}^{ev}$ constraint and maximum SoC $SoC_{\max}^{ev}$ constraint when EV finishes charging. P_t^{ev} stands for charing power of BSS. EV can only be charged when it arrives at home and connects to charging pole, the available charging time slot is defined as set Φ. The maximum charging power for EV is $P_{\text{ch.,max}}^{ev}$. The minimum charging power for EV is 0 since we do not consider V2G here. These constraints can be expressed as:

$$\begin{cases} P_t^{ev} = 0 & \text{if } t \notin \Phi \\ 0 \le P_t^{ev} \le P_{\text{ch.,max}}^{ev} & \text{if } t \in \Phi \end{cases} \tag{6}$$

3) PV Power generation: There are already many references that discuss the prediction of PV generation, hence the detailed PV prediction method is beyond the focus of this paper. In this paper we adopt average PV power in [9] as predicted power, and generate real PV power data with 5% error according to [10].

4) Basic load generation: In this paper, load prediction based on historical data is similar to that in PV prediction, the prediction error is set to 10%.

C. Communication latency generation

Timeliness is important in communication between cloud server and local controller. The discrete-time slot is set to 1 minute, and the communication between two terminals is defined as a valid one only if the information is sent and received within one-minute slot. The delayed communication command is invalid and will not be carried out. The time slot between two successive valid communication is modeled as Poisson distribution:

$$P[N(n+1) - N(n) = k] = \frac{e^{-\lambda}(\lambda)^k}{k!} k = 0, 1, ... \tag{7}$$

where λ is arrive rate. The distribution of valid communication reflects the real scenario in cloud-based EMS, in which latency may happen. In order to keep smart grid system reliable under high latency or offline condition, a redundancy control strategy is proposed. The detailed control strategy and switch mechanism are discussed in Section III.

III. Redundancy Control Strategy

A. Information flow

In order to take advantage of cloud server and local controller, a two-layer redundancy control strategy is proposed. This proposed strategy has several features similar to model predictive control (MPC) framework [11] and can be summarized as 1) the prediction of baseload and renewable energy generation is used for future power schedule; 2) the feedback of local states from local terminal to cloud server increases system robustness.

The proposed control strategy contains a cloud control layer and a local control layer, the information flow in this framework is shown in Fig. 2. At each control loop, cloud server sends power schedule result to local controller. Local controller checks power balance among power components and schedules power flow through energy router. Real-time load power, PV power, and BSS states are sent to cloud terminal. Combined with historical database, the predicted load and PV power is calculated. Then power schedule plan for future time slots is optimized and then sent to local controller.

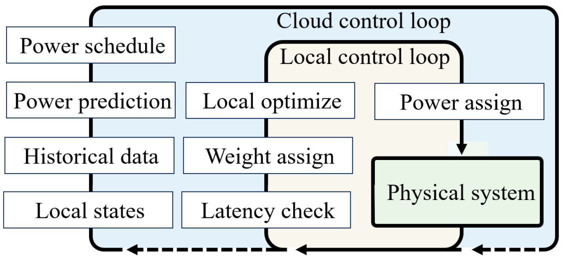


Fig. 2. Information flow of two-layer control loop

The controller algorithm at cloud and local terminal is introduced in detail as follows.

B. Control Algorithm

1) Control algorithm for cloud terminal: Cloud server has predicted data for load and PV power, and it has relatively good computation ability, hence it optimizes power schedule for each time slot in a period. The optimization problem in cloud controller aims to minimize electric cost and at the same time reduce fluctuation of BSS and grid power in a period, which is formulated as follows.

$$\min f = \sum_{t=0}^{T} C_t P_t^g \Delta t + \varepsilon_1 \sum_{t=0}^{T} \left| P_t^b - P_{t+1}^b \right| + \varepsilon_2 \sum_{t=0}^{T} \left| P_t^g - P_{t+1}^g \right| \tag{8}$$

subject to (2), (3), (5), (6), and

$$P_t^g + P_t^{pv\prime} + P_t^{bat} = P_t^{load\prime} + P_t^{ev} \quad \forall t \tag{9}$$

where C_t is time of use electric price, P_t^g is grid power, $P_t^{pv\prime}$ and $P_t^{load\prime}$ are predicted PV power and load power at time t respectively. The first term in object function (8) is the total cost for purchase electricity from grid. The second term reduces the variation of BSS, and the third term reduces the variation of grid power. This expression refers to [12] and is simplified to a linear form. ε_1 and ε_2 are user defined constants to weight the priority between minimum economic cost and reduce power fluctuation in object function.

In this optimization problem, battery charing/discharging power P_t^{bat}, EV charging power P_t^{ev} are optimization variables. Constraints (2) and (3) are physical constraints of BSS charging and discharging, constraints (5) and (6) are physical constraints of EV charging, constraint (9) maintains power balance.

2) Control algorithm for local terminal: Local controller has limited computation and storage performance, hence it doesn't store all historical operation data for prediction and only stores data in previous state. The local controller receives power schedule command form cloud server and only the command at current timestamp is carried out in local controller. The utility function of optimization problem in local controller can be formulated as

$$\mathrm{u}_t = \Gamma^{t-t_0} f_1 + (1 - \Gamma^{t-t_0}) f_2 \tag{10}$$

with

$$f_1 = \left\| P_t^b - P_t^{b*} \right\|^2 + \left\| P_t^{ev} - P_t^{ev*} \right\|^2 \tag{11}$$

$$f_2 = C_t P_t^g \Delta t + \gamma \left\| P_t^b - P_{t-1}^g \right\|^2 \tag{12}$$

$$\Gamma^{t-t_0} = \exp(-k(t - t_0)) \tag{13}$$

The weighted sum of f_1 and f_2 composed of two parts of the objective function. P_t^{b*} and P_t^{ev*} are command signal from cloud server for BSS power and EV power, P_t^b and P_t^{ev} are optimize variables of control algorithm for local terminal. f_1 models the difference between actual command and cloud command in quadratic form. f_2 models the cost at current time slot, which includes grid energy cost, and BSS vibration cost. P_{t-1}^g is BSS power at previous time slot. γ is user defined constant. The vibration cost is formulated in quadratic form considering BSS operation condition [12]. Γ^{t-t_0} is a weight factor that changes with the timeliness of cloud command, parameter k scales the change rate of weight factors. As the latency $t - t_0$ increases, the weight of f_1 decrease and the weight of f_2 increase. The utility function gradually change from f_1 to f_2.

The optimization problem in local controller aims to minimize utility function u_t at current time slot with control variable P_t^b and P_t^{ev} while subject to (2), (3), (5), (6), and

$$P_t^g + P_t^{pv} + P_t^b = P_t^{load} + P_t^{ev} \tag{14}$$

where P_t^{pv} and P_t^{load} are real time power for PV and basic load in smart grid.

C. Switch Mechanism

The proposed two-layer redundancy control system is able to switch smoothly under three scenarios, namely normal case, large latency case, and offline case.

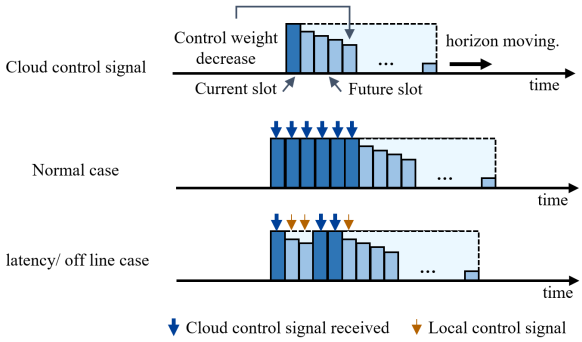


Fig. 3. Switch mechanism under different cases

As shown in Fig. 3, blue arrows represent cloud schedule signal, yellow arrows represent local control signal. Three cases are introduced as follows:

- In normal case, cloud schedule signal is received at each time slot, only f_1 remains in u_t, the best response for local controller is to carry out the schedule result in the first time slot.
- In large latency case, cloud schedule signal can not be received on time. Γ^{t-t_0} will weight between cloud reference and local object according to latency gap $t-t_0$. The utility function is a combination of f_1 and f_2.
- In offline case, no cloud control signal is received, power schedule at each time slot relays on the control of local controller with u_t equals to f_2.

The switch mechanism in local controller combined with power control algorithm can be summarized as Algorithm 1. This algorithm is a real-time algorithm that makes optimal control command considering schedule information from cloud and current states. The reliability under different communication latency states is guaranteed.

IV. Simulation Results and Analysis

A. Simulation setup

The reliability and performance of the proposed schedule framework are evaluated by a case study under different communication latency scenarios. In this case study, a household smart grid that shares the same configuration with Fig. 1 is set. BSS and EV parameters are listed in Table I.

The maximum output power of PV is 7kW, and PV power is picked from a typical day in summer [9]. Home load is picked from typical home load data in [13] and scaled by 2 times to reflect the increase of home load in a few years. The comparison of predicted power and real power used in this case study is shown in Fig. 4. Considering load peak at night and EV charging during midnight, the studied time period is a 36h period chosen from 0:00 AM to 12:00 AM next day. The time of use electricity price in this case study is taken from [14] and is shown in Fig. 5.

Algorithm 1 Two Layer Redundancy Control Algorithm

Initialize local states
while True **do**
  Current time slot $t = t + 1$
  if Receive new cloud command **then**
    Check cloud result time stamp t_0
    Update reference power $P^{b*}_{t-t_0}$ and $P^{ev*}_{t-t_0}$
  else
    Choose last received cloud command whose time stamp is τ
    Update reference power $P^{b*}_{t-\tau}$ and $P^{ev*}_{t-\tau}$
  end if
  Local optimization
  Implement local power command P^b_t and P^{ev}_t
  Send local states to cloud server
end while

TABLE I
SIMULATION PARAMETERS OF BSS AND EV

Parameter	value	Parameter	value
C^b	50kWh	C^{ev}	40kWh
$P^b_{\text{ch.,max}}$	5kW	$P^{ev}_{\text{ch.,max}}$	15kW
$P^b_{\text{disch.,max}}$	10kW	Φ	19:00-5:00
SoC^b_{ini}	0.5	SoC^{ev}_{ini}	0.3

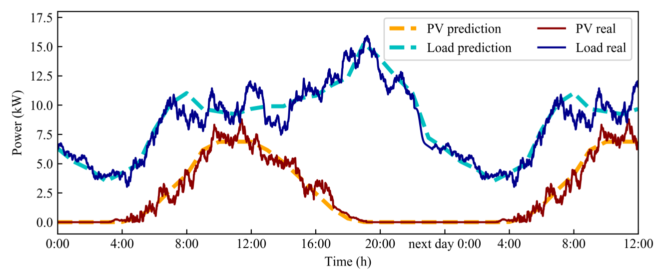


Fig. 4. Predicted and real power of PV and load

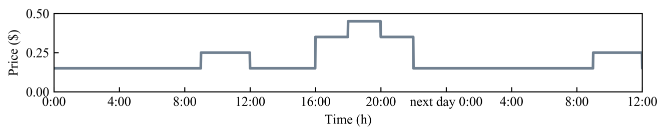


Fig. 5. Time-of-use electrical price

The simulation platform is programmed by Python and its structure is shown in Fig. 6. Two PCs represent cloud and local terminal respectively, and the schedule program on each PC communicates through websockets. Historical data storage and power schedule based on predicted data are realized on cloud terminal. Power balance of each component is realized through power model on local terminal.

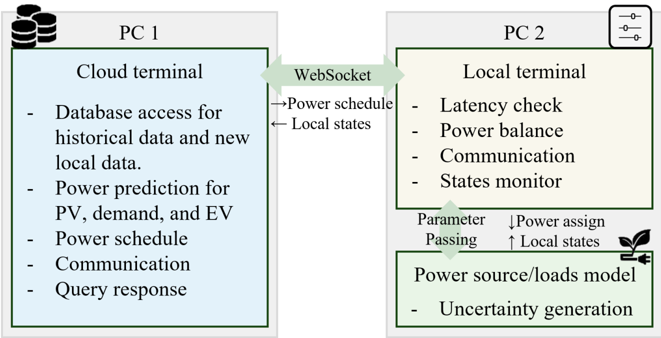


Fig. 6. Simulation configuration for cloud and local terminal

B. Comparison of different work states

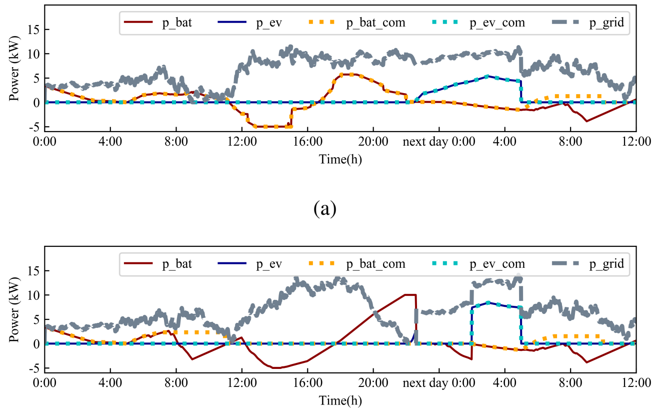
The scheduling results of household smart grid system under normal case and offline case are compared in Fig. 7. In normal case (Fig. 7(a)), cloud server and local controller carry out effective and real-time communication at each time slot, while in offline case (Fig. 7(b)), from 6:00 AM to 2:00 AM next day, cloud server is offline and hence only local controller works. The gray curve stands for grid power, the blue curve stands for PV power, and the red curve stands for BSS power. The orange dash line and light blue dash line stand for cloud server power command for BSS and PV respectively.

As it can be seen, In normal case (Fig. 7(a)), local controller follows real-time schedule command from the cloud server. The cloud schedule algorithm comprehensively considers grid price and power demand/supply relationship: BSS charge from 12:00 to 16:00 when electricity price is low and discharge from16:00 to 22:00 when electricity price is high, and the charging/discharging power of BSS is properly controlled so that the grid power demand is more stable compared to original load demand as shown in Fig. 4. In offline case, the schedule results of BSS is decreased. The reason is that in offline case, local controller has no predicted information for future slots, and the optimization object focuses on current slot. Even though, the smart grid energy system can still work under the redundancy of local controller.

The SoC states of BSS and EV in two cases are shown in Fig. 8. Combining Fig. 7 and Fig. 8, we can find that cloud-based EMS with predicted information can reduce grid fluctuation with less BSS usage, which is good to relieve BSS degeneration. The comparison of grid cost, total BSS supply power during this 36h period is shown in Table II. Both normal case and offline case achieve better economic benefits than the case without BSS schedule.

TABLE II
SIMULATION RESULTS IN THREE SCENARIOS

Scenario	Grid cost	BSS supply
Normal	50.05 ($)	33.79 (kWh)
Offline	50.86 ($)	40.01 (kWh)
Without battery	61.06 ($)	/



(a)

(b)

Fig. 7. Power of BSS, EV, and grid in different cases. (a) Normal case. (b) Offline case.

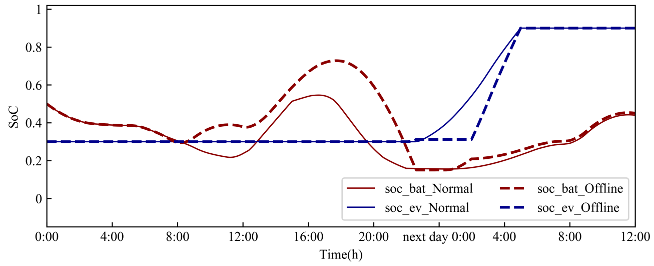


Fig. 8. SoC states of BSS and EV in normal and offline cases

C. Comparison of different latency

In order to compare the performance of the system under different latency cases, three latency level is generated from 6:00 to 2:00 next day by selecting different value of arrival rate λ in latency generation model. As shown in Fig. 9, case1-3 stands for λ equals to 0.2, 0.05, and 0.01 respectively. The green slot stands for the slot in which data is successfully received, the light red slot stands for the slot in which communication is invalided. The simulation results of case1-3 are shown in Fig. 10(a)-(c) respectively, and the SoC states of BSS and EV are shown in Fig. 11.

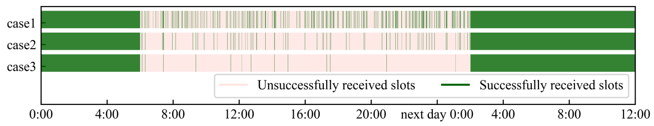


Fig. 9. Successfully received time slots distribution in three cases

The simulation shows that as the failure frequency increase, the system performance gradually shifts from cloud control case to local control case. The total cost of cases 1-3 is 48.9$, 49.2$ and 50.0$, respectively. The proper use of BSS will increase the efficiency and stability of household smart grid system.

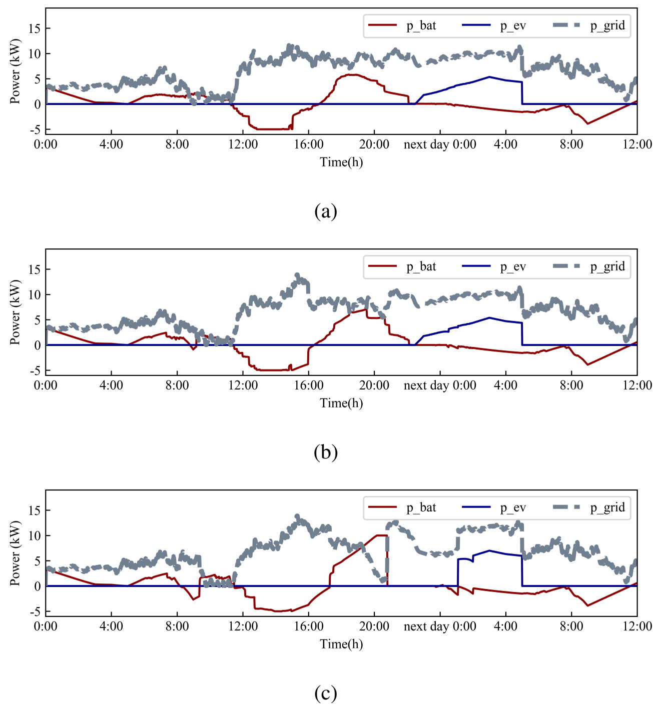


Fig. 10. Power of BSS, EV, and grid in different cases. (a) case 1. (b) case 2. (c) case 3.

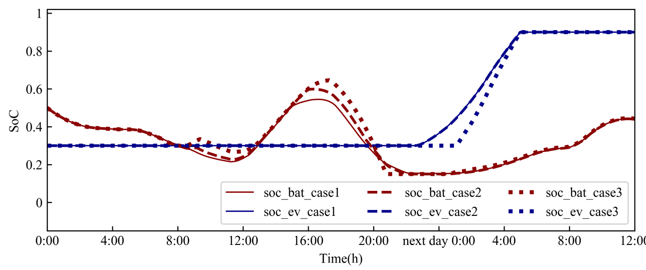


Fig. 11. SoC states of BSS and EV in different cases

V. Conclusion

Efficient energy management of household smart grid requires accurate prediction to relieve the uncertainty caused by renewable energy. Cloud-based EMS provides data storage facility and high computation power to carry out effective power prediction and power schedule. However the reliability of Cloud-based control for smart grid system is challenged when real-time communication between cloud and local terminal can not be guaranteed. Cloud control with local redundancy provides a new perspective for household smart grid control. In this paper, we propose a cloud-based EMS framework with local redundancy, and a two-layer control strategy to coordinate cloud server and local controller. The designed switch mechanism enables system to switch smoothly between cloud control mode and local control mode under different communication latency conditions. Simulation results show the effectiveness of proposed EMS framework in reducing cost and relieving power fluctuation. The proposed concept of local redundancy control capability for cloud-based EMS has further extensions such as considering the role of users' actions in participating in household smart grid system, which will be discussed in future works.

References

[1] J. Baek, Q. H. Vu, J. K. Liu, X. Huang, and Y. Xiang, "A secure cloud computing based framework for big data information management of smart grid," *IEEE Trans. Cloud Comput.*, vol. 3, no. 2, pp. 233–244, 2015.

[2] V. Monteiro, J. G. Pinto, and J. L. Afonso, "Operation modes for the electric vehicle in smart grids and smart homes: Present and proposed modes," *IEEE Trans. Veh. Technol.*, vol. 65, no. 3, pp. 1007–1020, 2016.

[3] M. H. Yaghmaee Moghaddam and A. Leon-Garcia, "A fog-based internet of energy architecture for transactive energy management systems," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1055–1069, 2018.

[4] C. Ju, P. Wang, L. Goel, and Y. Xu, "A two-layer energy management system for microgrids with hybrid energy storage considering degradation costs," *IEEE Trans. Smart Grid*, vol. 9, no. 6, pp. 6047–6057, 2018.

[5] Y. Chen and J. M. Chang, "Emaas: Cloud-based energy management service for distributed renewable energy integration," *IEEE Trans. Smart Grid*, vol. 6, no. 6, pp. 2816–2824, 2015.

[6] D. Niyato, Q. Dong, P. Wang, and E. Hossain, "Optimizations of power consumption and supply in the smart grid: Analysis of the impact of data communication reliability," *IEEE Trans. Smart Grid*, vol. 4, no. 1, pp. 21–35, 2013.

[7] Y. Liu, X. Chen, Y. Wu, K. Yang, J. Zhu, and B. Li, "Enabling the smart and flexible management of energy prosumers via the energy router with parallel operation mode," *IEEE Access*, vol. 8, pp. 35 038–35 047, 2020.

[8] A. Alsabbagh, D. Yan, S. Han, Y. Wang, and C. Ma, "Behaviour-based distributed energy management for charging evs in photovoltaic charging station," in *2018 IEEE International Conference on Industrial Electronics for Sustainable Energy Systems (IESES)*, 2018, pp. 339–344.

[9] Renewable resource data center, national renewable energy laboratory. [Online]. Available: https://www.nrel.gov/rredc/

[10] P. P. Varaiya, F. F. Wu, and J. W. Bialek, "Smart operation of smart grid: Risk-limiting dispatch," *Proc. IEEE*, vol. 99, no. 1, pp. 40–57, 2011.

[11] A. Parisio, E. Rikos, and L. Glielmo, "A model predictive control approach to microgrid operation optimization," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 5, pp. 1813–1827, 2014.

[12] J. Chen, X. Huang, S. Tian, Y. Cao, B. Huang, X. Luo, and W. Yu, "Electric vehicle charging schedule considering user's charging selection from economics," *IET Gener., Transm. & Dis.*, vol. 13, no. 15, 2019.

[13] Base load datasets. [Online]. Available: http://en.openei.org

[14] A. Alsabbagh, H. Yin, and C. Ma, "Distributed electric vehicles charging management with social contribution concept," *IEEE Trans. Ind. Informat.*, vol. 16, no. 5, pp. 3483–3492, 2020.