Reinforcement Learning-Based Home Energy Management System for Resiliency

Naren Srivaths Raman, Ninad Gaikwad, Prabir Barooah, and Sean P. Meyn

***Abstract*— With increase in the frequency of natural disasters such as hurricanes that disrupt the supply from the grid, there is a greater need for resiliency in electric supply. Rooftop solar photovoltaic (PV) panels along with batteries can provide resiliency to a house in a blackout due to a natural disaster. Our previous work showed that intelligence can reduce the size of a PV+battery system for the same level of post-blackout service compared to a conventional system that does not employ intelligent control. The intelligent controller proposed is based on model predictive control (MPC), which has two main challenges. One, it requires simple yet accurate models as it involves real-time optimization. Two, the discrete actuation for residential loads (on/off) makes the underlying optimization problem a mixed-integer program (MIP) which is challenging to solve. An attractive alternative to MPC is reinforcement learning (RL) as the real-time control computation is both model-free and simple. These points of interest accompany certain trade-offs; RL requires computationally expensive off-line learning, and its performance is sensitive to various design choices.**

In this work, we propose an RL-based controller. We compare its performance with the MPC controller proposed in our prior work and a non-intelligent baseline controller. The RL controller is found to provide a resiliency performance—by commanding critical loads and batteries—similar to MPC with a significant reduction in computational effort.

I. Introduction

In the recent past the frequency of extreme weather events like hurricanes, heat waves, and forest fires have increased [1]. The powerful winds associated with hurricanes have been responsible for damage to the transmission and distribution system of the power grid leading to extended power outages [2]. Some examples include Hurricane Irma which led to the loss of electricity for 4.8 million utility customers in Florida, with 1.5 million remaining without electricity for five days or more [3], and Hurricane Maria which led to months-long blackout in Puerto Rico [4], with an estimated death toll in the thousands [5].

Distributed solar generation can provide a resilient energy supply since the sky is often clear immediately after a hurricane. However, as the average household load in the U.S. is quite high $30.5\ kWh/day$ [6], serving the entire household load from an on-site PV+battery system will require a large system, driving up cost substantially.

In our prior work [7] we show that an intelligent controller can reduce the size—and thus, cost—of the PV+battery system to provide resiliency. It is able to do so by exploiting the flexibility in demand and supply along with forecasts. In order to maintain habitable conditions during an extended outage, certain critical loads—like the refrigerator to keep food and medicine safe, lights for illumination, and fans to provide some thermal comfort—need to be serviced, but noncritical loads need not. Among these critical loads, refrigerators have a higher priority over lights and fans. The lower priority loads can be shed in favor of servicing the refrigerators; this offers flexibility in demand. Flexibility in supply comes from the fact that the charging rate of a battery is variable; a battery can be fast charged when low solar irradiance is expected, however, at a cost to battery's health.

The intelligent controller that we proposed in our prior work is based on model predictive control (MPC). A challenge with MPC is that it requires simple yet accurate models as it involves real-time optimization. Moreover, due to the discrete nature of actuation for the residential loads (on/off), the underlying optimization problem in the MPC ends up being a mixed-integer program (MIP). Depending on the planning horizon for MPC, the number of decision variables can be large, and solving such a high dimensional constrained mixed-integer program with limited computing resources can be challenging. During power outages after a hurricane, computing resources are limited. Accessing cloud-based services might not be an option as communication infrastructure might be damaged, and locally available controller hardware might not be powerful.

In this work, we propose a reinforcement learning (RL)-based controller for the same resiliency problem mentioned above. RL is a set of tools used to approximate an optimal policy based on data obtained from a physical system or its simulation. It has two key advantages over MPC; *the real-time computation is both model-free and simple*. This simplicity makes RL an attractive alternative to MPC for our problem.

As in [7], our focus is on designing a controller only for post-disaster scenarios during which grid supply is unavailable. When grid supply is restored, it is assumed that the software will switch to a "normal operating" mode. The normal operating mode may also be a sophisticated controller that seeks to, for instance, minimize the utility bill of the consumer by controlling the PV+battery system. There is a plethora of work in that direction; see [8], [9], [10], [11], and [12], with some recent works using RL [13], [14], and [15]. Therefore we do not consider that problem here. The work [16] presents a rule-based controller for charging the house battery to its maximum before an outage occurs;

The authors are with the University of Florida, Gainesville, Florida 32611, USA. narensraman@ufl.edu, ninadgaikwad@ufl.edu, pbarooah@ufl.edu, meyn@ece.ufl.edu. The research reported here is partially supported by NSF awards 1646229 and 1934322.

however, it does not consider developing a controller for a post-disaster scenario. Works on controlling the PV+battery system to maximize resiliency performance in a post-disaster scenario, the focus of this paper, is extremely limited. To the best of our knowledge, only [17], their follow up work [18], and our prior work [7] consider the problem of operation for resilient energy supply to a house. Both [17] and [18] use MPC, however, they ignore the mixed-integer nature of the optimization problem, and also ignore the capability of a battery to vary charging rate which can be exploited.

We consider a single family house with solar PV panels, a battery energy storage system, and three loads: refrigerator, lights, and fans. These devices are shown in Figure 1, along with other relevant infrastructure. Among the loads we consider refrigerator to be the *primary* load, and lights and fans together as the *secondary* load. The primary goal of a control system during an extended power outage is to maintain the refrigerator temperature and to keep the battery alive. A secondary goal is to service the secondary load as much as possible while following a user-defined load profile and use fast charging of the battery judiciously.

The RL controller presented is designed to satisfy the goals mentioned above. There are several challenges in applying RL for the resiliency problem. First, there are state constraints like maintaining the refrigerator temperature within bounds, which are hard to impose using a model-free technique like RL. In addition, performance of RL is sensitive to many design choices, such as the states and cost function. The proposed RL formulation addresses some of these challenges. We design the state-space to include certain exogenous inputs along with their forecasts, as they can provide valuable information to the controller. We design the cost function, which helps the controller to learn the state constraints, and thus, maintains the refrigerator temperature within the desired bounds during implementation (even without access to a model).

We compare the performance of the proposed RL controller with the MPC controller from our prior work [7], and a baseline controller that is representative of the commercial systems one can install today. We also test the robustness of the RL controller to forecast errors. Simulations show that the proposed RL controller is able to service the primary load similar to the MPC controller, but with a significant reduction in real-time computational effort. It is also found that the baseline controller fails to service the primary load for several hours each day. The secondary load servicing performance of RL is found to be poorer than MPC.

The rest of this paper is organized as follows. Section II describes the system. Section III presents our proposed RL-based controller. Section IV briefly describes the MPC and baseline controllers used for comparison. The simulation setup is described in Section V. Simulation results are presented and discussed in Section VI. Finally, the main conclusions are provided in Section VII.

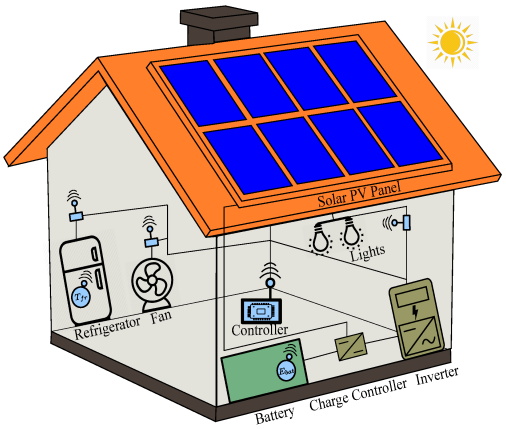


Fig. 1: Hardware involved in the proposed control system.

II. System Description

In order to achieve the goals mentioned in Section I, a control system can command the following: (i) on/off state of the refrigerator ($u_{fr}(k) \in \{0,1\}$), (ii) on/off state of the secondary load (aggregate of lights and fans, $u_s(k) \in \{0,1\}$), (iii) charging/discharging state of the battery ($c(k), d(k) \in \{0,1\}$), and (iv) when charging the battery, the charging mode of the battery ($m(k) \in \{1,2\}$). The battery has two charging modes: normal and fast charging, where fast charging is undesirable as it degrades battery life. Hence, the control commands are,

$$u(k) := [u_{fr}(k), u_s(k), c(k), d(k), m(k)]^T. \qquad (1)$$

Time is discrete, with $k = 0, 1, 2, \ldots$ denoting the time index and ΔT_s denoting the interval (hours or minutes) between k and $k+1$. In the sequel, $E(k)$ (Wh) will denote the energy consumed/generated during the time interval between time indices k and $k+1$, with the subscript specifying the source or consumer of the energy.

Figure 1 shows a schematic of the plant. The various components of the plant and the key quantities associated with them are listed next. (i) Solar PV panels with energy production potential $E_{pv}(k)$, (ii) battery with energy $E_{bat}(k)$ bounded between a minimum and maximum, i.e., $E_{bat}(k) \in [\underline{E}_{bat}, \bar{E}_{bat}]$, (iii) refrigerator (primary load) with internal temperature $T_{fr}(k)$, and with desired range between a minimum ($\underline{T}_{fr}$) and maximum ($\bar{T}_{fr}$), and (iv) lights and fans (secondary load) with a user-defined demand trajectory $E_s(k)$. Moreover, $T_{fr}(k)$ is affected by the house temperature which is denoted by $T_{house}(k)$. Description of their mathematical models are presented in Section II of our prior work [7], and are omitted here.

III. Reinforcement Learning-Based Controller

A. Markov Decision Process (MDP) Formulation

The proposed RL controller is designed to construct a control policy which can satisfy the goals mentioned in Section I. As a way to motivate RL design, we model the problem as a discrete-time Markov decision process comprised of the tuple $(\mathcal{X}, \mathcal{U}, \mathcal{P}, C, \beta)$, where $\mathcal{X}$ denotes the state-space, $\mathcal{U}$ denotes the action-space (a set of all *feasible*

TABLE I: Feasible control inputs U_k.

u_{fr}	0	1	1	0	0	1	1
u_s	1	0	1	0	1	0	1
Γ	-1	-1	-1	0	0	0	0

u_{fr}	0	0	1	1	0	0	1	1
u_s	0	1	0	1	0	1	0	1
Γ	1	1	1	1	2	2	2	2

actions), $\mathcal{P}$ denotes the transition kernel, $C : \mathcal{X} \times \mathcal{U} \times \mathcal{X} \to \mathbb{R}$ denotes the cost function, and β denotes a discount factor. We denote by $X_k \in \mathcal{X}$ the state, and $U_k \in \mathcal{U}$ the control input at time-step k. Next, we define some of the key components of the MDP for our problem.

Choosing the state space: Based on the system described in Section II and the goals mentioned in Section I, it is natural to include the battery energy level E_{bat} and the refrigerator internal temperature T_{fr} in the state X_k. In addition to these, we include certain exogenous inputs along with their forecasts, as they can provide valuable information to the controller. So we have:

$$\begin{aligned} X_k := [&E_{bat}(k), T_{fr}(k), T_{house}(k), E_s(k), E_{s,1}(k), \\ &E_{s,2}(k), E_{s,3}(k), E_{pv}(k), E_{pv,1}(k), E_{pv,2}(k), \\ &E_{pv,3}(k), E_{pv,4}(k), E_{pv,5}(k), E_{pv,6}(k)]^T \in \mathbb{R}^{14} \end{aligned} \tag{2}$$

where $E_{pv,1}, \ldots, E_{pv,6}$ are 4 hour averages of forecast of the PV energy production potential for the next 24 hours: $E_{pv,i}(k) := \frac{1}{24} \sum_{j=k+(i-1)\times 24+1}^{k+i\times 24} E_{pv}(j)$, and $E_{s,1}, \ldots, E_{s,3}$ are a function of the desired secondary load (lights-fans) trajectory: $E_{s,i}(k) := \sum_{j=k+(i-1)\times 6+1}^{k+i\times 6} E_s(j)$.

Choosing the action space: Based on the system described in Section II, we define the control inputs as:

$$U_k := [u_{fr}(k), u_s(k), \Gamma(k)]^T, \tag{3}$$

where $\Gamma(k) \in \{-1, 0, 1, 2\}$ denotes discharging, idle, charging at normal rate, and fast charging, respectively. The mapping of $\Gamma(k)$ into its constituent battery control commands $[c(k), d(k), m(k)]$ is defined below:

$$c(k) = \begin{cases} 1\,, & \text{if } \Gamma(k) = 1 \text{ or } 2 \\ 0\,, & \text{otherwise} \end{cases} \tag{4}$$

$$d(k) = \begin{cases} 1\,, & \text{if } \Gamma(k) = -1 \\ 0\,, & \text{otherwise} \end{cases} \tag{5}$$

$$m(k) = \begin{cases} 1\,, & \text{if } \Gamma(k) = 1 \\ 2\,, & \text{if } \Gamma(k) = 2 \\ 0\,, & \text{otherwise.} \end{cases} \tag{6}$$

In total, there are 16 combinations of control inputs (U_k) possible. Of these various combinations, $U_k = [0, 0, -1]^T$ is not feasible because if the battery is discharging, then either the primary or the secondary load needs to be served. *So there are 15 feasible combination of control inputs in total which are listed in Table I.*

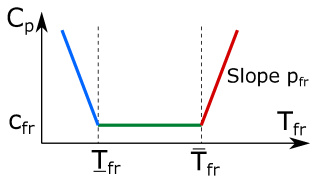


Fig. 2: Cost function for the primary load (refrigerator).

Cost function design: The cost function is designed to capture four key features: (i) maintain the internal temperature of the refrigerator (primary load) within bounds so that the food is safe, (ii) service the secondary load (fans and lights) as desired by the occupants, (iii) keep the battery alive, and (iv) use fast charging judiciously. The overall cost function is the sum:

$$C_{total} := C_p + C_s + C_{bat} + C_c + C_{pv}, \tag{7}$$

where each of the terms are discussed in detail next.

The cost function for the primary load (refrigerator) is:

$$\begin{aligned} C_p(X_k, U_k, X_{k+1}) &= c_{fr} \\ &+ p_{fr}\Big([T_{fr}(k+1) - \bar{T}_{fr}]_+ + [\underline{T}_{fr} - T_{fr}(k+1)]_+\Big) \end{aligned}$$

where c_{fr} is the (low) cost for maintaining T_{fr} within bounds $[\underline{T}_{fr}, \bar{T}_{fr}]$, and p_{fr} is the (large) penalty/cost for violating the bounds; see Figure 2.

The cost function for the secondary load (lights-fans) is:

$$C_s(X_k, U_k, X_{k+1}) = \begin{cases} c_s\,, & \text{if } E_s(k) > 0 \;\&\; E_s^c(k) > 0 \\ p_s\,, & \text{if } E_s(k) > 0 \;\&\; E_s^c(k) = 0 \\ p_s\,, & \text{if } E_s(k) = 0 \;\&\; E_s^c(k) > 0 \\ c_s\,, & \text{if } E_s(k) = 0 \;\&\; E_s^c(k) = 0 \end{cases}$$

where p_s is the (high) cost for not servicing the secondary load when desired or servicing when not desired, c_s is the (low) cost for servicing when desired or not servicing when not desired, and $E_s^c(k)$ is the energy consumed by secondary load.

The cost function for maintaining the battery state of charge is:

$$\begin{aligned} C_{bat}(X_k, U_k, X_{k+1}) &= c_{bat} \\ &+ p_{bat}[\underline{E}_{bat} + \Delta E_{bat} - E_{bat}(k+1)]_+ \end{aligned}$$

where c_{bat} is the (low) cost for keeping the battery alive, and p_{bat} is the (high) cost for allowing the battery to go below a prescribed minimum level of $\underline{E}_{bat} + \Delta E_{bat}$ and thus eventually die.

We use a hierarchical cost structure to promote normal charging over fast charging as shown below:

$$C_c(X_k, U_k, X_{k+1}) = \begin{cases} c_n\,, & \text{if } \Gamma(k) \in \{1\} \;\&\; E_{bat}(k) < \bar{E}_{bat} \\ & \quad \;\&\; E_{pv}(k) > u_{fr}(k)E_{fr}(k) + u_s(k)E_s(k) \\ c_f\,, & \text{if } \Gamma(k) \in \{2\} \;\&\; E_{bat}(k) < \bar{E}_{bat} \\ & \quad \;\&\; E_{pv}(k) > u_{fr}(k)E_{fr}(k) + u_s(k)E_s(k) \\ 0\,, & \text{otherwise} \end{cases}$$

where c_f ($> c_n$) is the higher cost for fast charging.

When there is no PV production potential (for example, during nighttime), charging the battery is not possible. Moreover, if a load is trying to be serviced when there is no PV potential, then the battery cannot be idle and needs to discharge. To discourage such undesired behaviors, we define the following cost function:

$$C_{pv}(X_k, U_k, X_{k+1}) = \begin{cases} p_{pv}\ , & \text{if } E_{pv}(k) = 0 \ \&\ \Gamma(k) \in \{1,2\} \\ p_{pv}\ , & \text{if } E_{pv}(k) = 0 \ \&\ \Big(u_{fr}(k) = 1 \text{ or } u_s(k) = 1\Big) \\ & \quad \&\ \Gamma(k) \in \{0\} \\ 0\ , & \text{otherwise} \end{cases}$$

The constants p_{pv}, c_n, c_f, c_{bat}, p_{bat}, c_s, p_s, c_{fr}, and p_{fr} are design choices.

B. Value Function Approximation and Zap Q-Learning

The goal is to obtain a state-feedback policy $\phi^* : \mathcal{X} \to \mathcal{U}$ that minimizes the sum of expected discounted cost:

$$\phi^* := \underset{\phi:\mathcal{X}\to\mathcal{U}}{\operatorname{argmin}} \left\{ \sum_{k=0}^{\infty} \beta^k \mathsf{E}\big[c(X_k, U_k, X_{k+1})\big] \right\} \tag{8}$$

with $U_k = \phi(X_k)$ for $k \geq 0$.

Under the assumption that the underlying problem is an MDP, it is known that the optimal policy satisfies:

$$\phi^*(x) = \underset{u\in\mathcal{U}(x)}{\arg\min}\, Q^*(x,u)\,, \qquad x \in \mathcal{X} \tag{9}$$

where $Q^* : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$ denotes the associated optimal Q function:

$$Q^*(x,u) := \min_{\{U_k\}} \sum_{k=0}^{\infty} \beta^k \mathsf{E}\big[c(X_k, U_k, X_{k+1}) | X_0 = x, U_0 = u\big]$$

where the minimization is over all feasible inputs.

Reinforcement learning algorithms such as Q-learning can be used to estimate an approximation for the Q-function. In this work, we use Zap Q-Learning to approximate Q^* using a parameterized family of functions $\{Q^\theta : \theta \in \mathbb{R}^d\}$ [19]. We employ a linear parameterization, so that,

$$Q^\theta(x,u) = \theta^T \psi(x,u), \quad x \in \mathcal{X},\ u \in \mathcal{U}, \tag{10}$$

where $\psi : \mathcal{X} \times \mathcal{U} \to \mathbb{R}^d$ denotes the "basis functions". For our problem, we choose the basis functions as follows:

$$\begin{aligned} \psi(x,u) := [&f(x)\mathbb{I}_{[0,1,-1]^T}(u);\ f(x)\mathbb{I}_{[1,0,-1]^T}(u); \\ &f(x)\mathbb{I}_{[1,1,-1]^T}(u);\ f(x)\mathbb{I}_{[0,0,0]^T}(u);\ f(x)\mathbb{I}_{[0,1,0]^T}(u); \\ &f(x)\mathbb{I}_{[1,0,0]^T}(u);\ f(x)\mathbb{I}_{[1,1,0]^T}(u);\ f(x)\mathbb{I}_{[0,0,1]^T}(u); \\ &f(x)\mathbb{I}_{[0,1,1]^T}(u);\ f(x)\mathbb{I}_{[1,0,1]^T}(u);\ f(x)\mathbb{I}_{[1,1,1]^T}(u); \\ &f(x)\mathbb{I}_{[0,0,2]^T}(u);\ f(x)\mathbb{I}_{[0,1,2]^T}(u);\ f(x)\mathbb{I}_{[1,0,2]^T}(u); \\ &f(x)\mathbb{I}_{[1,1,2]^T}(u)], \end{aligned} \tag{11}$$

where $\mathbb{I}_A : \mathcal{U} \to \{0,1\}$ is the indicator function of the set A. For $f(x)$, we choose a quadratic function of the states:

$$\begin{aligned} f(x) := [&E_{bat}^2,\ T_{fr}^2,\ T_{house}^2,\ E_{pv}^2,\ E_{bat}T_{fr},\ E_{bat}T_{house}, \\ &E_{bat}E_{pv},\ E_{bat}E_s,\ T_{fr}T_{house},\ T_{fr}E_{pv},\ T_{fr}E_s, \\ &T_{house}E_{pv}, T_{house}E_s,\ E_{pv}E_s,\ E_{bat},\ T_{fr},\ T_{house}, \\ &E_{pv},\ E_{pv,1},\ E_{pv,2},\ E_{pv,3},\ E_{pv,4},\ E_{pv,5},\ E_{pv,6}, \\ &E_s,\ E_{s,1},\ E_{s,2},\ E_{s,3},\ 1]^T \in \mathbb{R}^{29}. \end{aligned}$$

Therefore, there are $29 \times 15 = 435$ parameters to be learned, i.e., $\theta \in \mathbb{R}^{435}$. Once the basis functions are fixed, the Zap Q-learning algorithm can be used to estimate Q^* using the approximation Q^{θ^*}. We refer the interested reader to Algorithm 1 in [20] for details. The algorithm is implemented using the simulation models presented in Section II of our prior work [7]; given a current state X_k and a control input U_k, the state X_{k+1} at the next time step is obtained using these simulation models, and the tuple (X_k, U_k, X_{k+1}) is used to update the parameters θ.

C. Real-Time Control

The online state-feedback control is computed as follows:

$$\begin{aligned} U_k = \phi^{\theta_T}(X_k) &= \underset{u\in\mathcal{U}(X_k)}{\arg\min}\, Q^{\theta_T}(X_k, u) \\ &= \underset{u\in\mathcal{U}(X_k)}{\arg\min}\, \theta_T^T \psi(X_k, u)\,,\ X_k \in \mathcal{X}, \end{aligned} \tag{12}$$

where θ_T is the estimate of θ^* obtained from the algorithm. *The minimum is over only 15 values, so the optimization problem above is trivial to solve.*

IV. Control Algorithms Used for Comparison

The performance of the RL controller is compared with two others: an MPC controller which was proposed in our prior work [7], and a rule-based baseline controller. These are briefly discussed in the following subsections. See [7] for further details.

A. Model Predictive Control (MPC)

The goal of the MPC controller is the same as that of the proposed RL controller. The controller solves a Mixed-Integer Linear Program (MILP) over a finite planning horizon N to compute the control commands in discrete time steps ΔT_s. The controller uses the following pieces of information to solve this problem: (i) the current value of the states, (ii) forecasts of the exogenous inputs (E_{pv}, T_{house}, and E_s), and (iii) models for refrigerator thermal dynamics and battery energy dynamics. The control commands for the first time step obtained from the solution of this problem are applied to the plant. This process is repeated at the next time step.

The MPC controller has the battery energy level (E_{bat}) and refrigerator internal temperature (T_{fr}) as states, hence we have $x(k) := [E_{bat}(k), T_{fr}(k)]^T$. The control commands are similar to the ones mentioned in II; however the discrete battery control commands $[c(k), d(k), m(k)]^T$ are mapped on to a single continuous command $\Gamma_c(k) \in \mathbb{R}$. The transformation of $\Gamma_c(k)$ to its constituent $[c(k), d(k), m(k)]^T$

is carried out using rule-based logic. The MILP is solved to minimize: refrigerator temperature deviation from its bounds, and battery degradation; and to maximize: battery energy level, and servicing of secondary load; subject to: (i) equality constraints due to battery storage system dynamics model, refrigerator thermal dynamics model, and energy balance model, (ii) box constraints to maintain battery energy level and refrigerator temperature within desired limits, and (iii) various control command constraints. Details of the MPC controller are omitted due to lack of space, see [7] for details.

B. Baseline Controller

The baseline controller consists of two rule-based controllers that are independent of each other. The refrigerator on-off command $(u_{fr}(k))$ is computed by a thermostat controller. The battery charging $(c(k))$ and discharging $(d(k))$ commands are computed by the battery logic controller; it commands the battery to charge when there is excess PV energy and to discharge when the PV energy cannot meet the load demand. The baseline controller does not have a fast charging mode, as certain amount of intelligence is required to exploit the fast charging mode so as to avoid battery degradation. The detailed rule-based logic of both the controllers are presented in [7].

V. Simulation Study Setup

Simulations are conducted for a period of 7 days starting at 00:00 hours from Sept. 11, 2017, to Sept. 17, 2017. The plant is initialized with battery state at $\bar{E}_{bat}$ (i.e., $E_{bat}(0) = \bar{E}_{bat}$) and the refrigerator initial temperature at $2°C$ (i.e., $T_{fr}(0) = 2°C$). The simulation period selected corresponds to the time hurricane Irma made landfall and passed over Gainesville, FL, USA. The source of weather data is National Solar Radiation Database (nsrdb.nrel.gov). The simulations are carried out in MATLAB.

The sizing of the PV battery system is done using a conservative method described in [21]. Canadian Solar CS6K-285 polycrystalline panel, and Trojan SPRE 12 225 (lead acid type) solar battery unit were selected for the system sizing. Lead acid batteries were chosen over Lithium-Ion (Li-Ion) batteries for system cost reduction, as Li-Ion batteries are four times more expensive than lead acid batteries per kWh [22]. The size of the system obtained from this method is as follows: $855\ W$ of PV panels, $5400\ Wh$ of battery storage.

The house described in [23] consists of four bedrooms (1 fan [$65\ W$] and 1 LED [$8\ W$] each), a living room (1 LED), and a kitchen (1 refrigerator [$250\ W$] and 1 LED). The secondary load trajectory for a given day is composed of: LED lights (total units = 6) being on from 18:00 hours to 00:00 hours and fans (total units = 4) running from 21:00 hours to 09:00 hours and is constant for all the days of the simulation.

TABLE II: RL parameters.

β	ρ	c_{fr}	p_{fr}	c_{bat}	p_{bat}
0.95	0.8	-1	12500.25	-1	162340.9
c_n	c_f	c_s	p_s	p_{pv}	T
-0.25	-0.125	-0.5	500	500	10^7

A. Simulation Parameters

The parameters for the plant system models are mentioned here concisely due to lack of space. For detailed simulation parameters refer our prior work [7].

PV panels: $P_{pv}^{rated} = 285\ W$ (rated power output of PV module); Battery: $\underline{E}_{bat} = 1080\ Wh$, $\bar{E}_{bat} = 5400\ Wh$, $\eta_{bat} = 0.9$ (battery efficiency) ; Refrigerator: $\underline{T}_{fr} = 0\ °C$, $\bar{T}_{fr} = 4\ °C$.

RL parameters: The various parameters used in the RL controller are listed in Table II, where T is the number of training iterations. During learning, the Q function parameters are seen to settle after 10^7 iterations, and takes about 7.5 hours to train. For these learning simulations, we use 2016 weather data for Gainesville, Florida, obtained from the National Solar Radiation Database (nsrdb.nrel.gov).

MPC parameters: A planning horizon of 24 hours is used with a time-step of 10 minutes (i.e. $\Delta T_s = 10$ mins, $N = 144$). Hence, the MILP optimization problem central to the MPC has a total of 1008 decision variables comprising of 720 continuous and 288 binary decision variables.

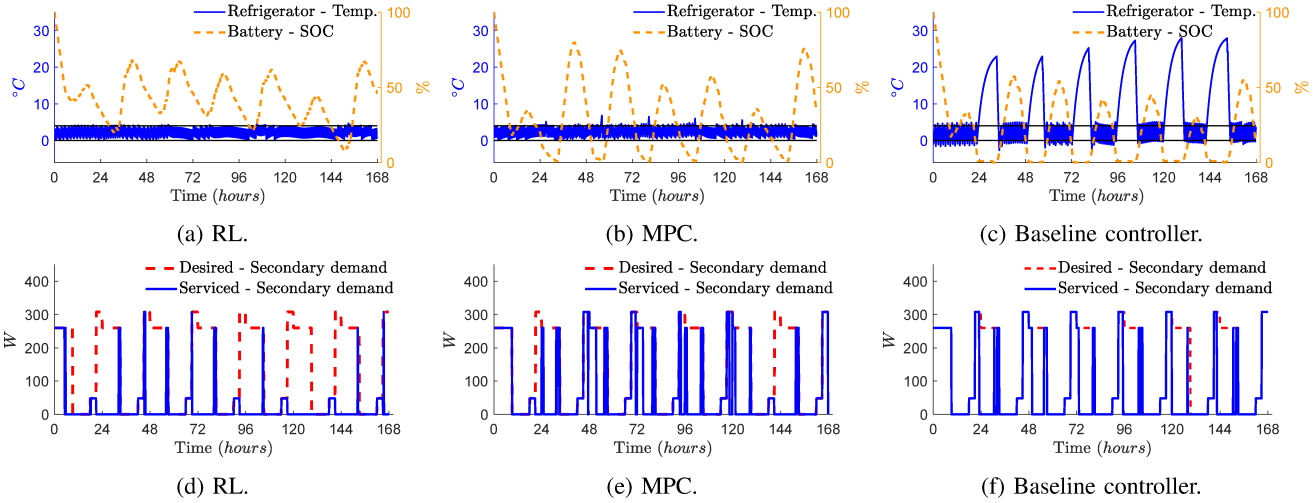
VI. Results and Discussion

A. Load Servicing Performance

Figure 3 shows the simulation results when using the RL, MPC, and baseline controllers. Both the proposed RL controller and the MPC controller keep the refrigerator temperature within the prescribed limits for the entire 7 days with negligible excursions; see Figures 3a and 3b. In contrast, the baseline controller fails to keep the refrigerator temperature within bounds for elongated periods; see Figure 3c. The average daily refrigerator temperature violation is 8.37 hours/day for the baseline controller, but none for the proposed RL controller; see Table III. Note that the Centers for Disease Control and Prevention state that perishable foods (including meat, poultry, fish, eggs and leftovers) in the refrigerator should be thrown away if the power has been off for 4 hours or more [24]. Thus, while the RL and MPC controllers will be able to keep perishable foods fresh for the entire seven days of the outage, with the baseline controller, the stored food will get spoiled after the very first day without grid power.

Figures 3d, 3e, and 3f show the trajectories of the secondary load serviced by the RL, MPC, and baseline controllers respectively. It can be seen that none of the controllers are able to meet the secondary load for the desired duration. The MPC controller has a better performance than the RL controller in servicing the secondary load; see Table III.

Hence, the proposed RL controller demonstrates similar performance in servicing the primary load compared to



(a) RL. (b) MPC. (c) Baseline controller.

(d) RL. (e) MPC. (f) Baseline controller.

Fig. 3: Comparison of controllers' performances for the primary load (top row) and secondary load (bottom row). The weather data used is for the week after Hurricane Irma made landfall in Gainesville, FL.

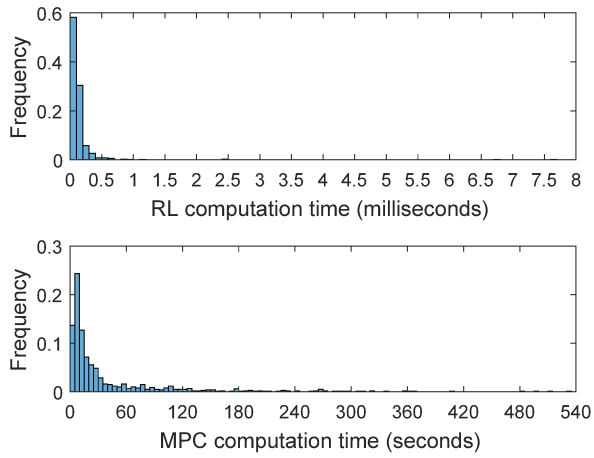


Fig. 4: Histograms of the computation time when using the RL and MPC controllers. Note that for RL, the time is in milliseconds, while for MPC, it is in seconds.

the MPC controller, but MPC performs better than RL in servicing the secondary load.

B. Computational Performance

Figure 4 shows a histogram of the time taken for the real-time computation in the RL and MPC controllers. There is a several-orders (4×10^5) of magnitude reduction in the computational effort when using the RL controller compared to the MPC controller, which makes RL an attractive choice for our problem.

The optimization problem in MPC is solved using GUROBI [25], a MILP solver, on a Desktop Linux computer with 8GB RAM and a 3.60 GHz×8 CPU. Recall that the optimization problem has a total of 1008 decision variables of which 288 are binary. On an average it takes 62.47 seconds for GUROBI to solve the MILP in MPC for one planning horizon; however, the solver stalled for 4.6% of the times. Note that both the computer hardware and the software (GUROBI, a commercial solver) used for solving the optimization problem is quite powerful. Despite using such powerful hardware and software, it takes about 62.47 seconds to solve. During an extended power outage, the computing resources available are going to be much lower, which might make an MILP-based controller quite challenging to use.

On the other hand, the real-time control computation in the RL controller involves finding the minimum over only 15 values, and thus on an average takes only 0.14 milliseconds to solve.

The speed and simplicity of real-time computation in RL comes with computationally complex off-line learning. It takes about 7.5 hours to learn the Q function parameters. However, the learning can happen during a non-contingency situation, i.e., when there is no power outage. Under such conditions, using cloud-based computing resources is also a possibility.

TABLE III: Performance comparison of RL, MPC, and Baseline Controllers.

	RL	MPC	Baseline
Refrigerator temp. violation ($hours/Day$)	0	0.042	8.375
secondary load served ($\%$ $time$)	28	53	48

C. Robustness

During an extended power outage access to accurate weather forecast can be challenging. We test the performance of the RL controller under such conditions to see if it is robust to forecast errors. Recall that the forecast of PV energy production potential ($E_{pv}(k)$, $E_{pv,1}(k)$, ...) is a part of the state in the RL controller. Figure 4 shows E_{pv} used in simulating the plant (actual) and the forecast used in the controller. The RL controller is found to be robust to errors in forecast. The refrigerator temperature is kept within the prescribed limits for the entire 7 days with negligible excursions, and the secondary load is served 30% of the time. The MPC controller is also found to have a similar degree of robustness to forecast errors.

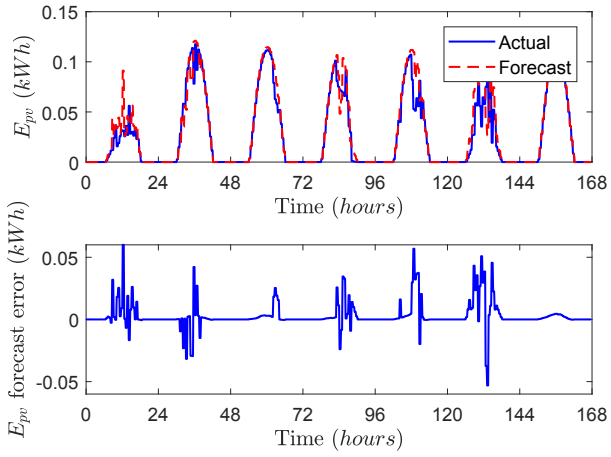


Fig. 5: Comparison of the actual PV energy production potential used in the plant and the forecast used in the RL controller to test its robustness to forecast errors.

In addition, recall that for the RL controller, the learning simulations are done using weather data from 2016, while all the testing simulations are done using weather data from 2017, during which Hurricane Irma occurred.

VII. Conclusion

We presented an RL-based controller for energy management of a house—consisting of solar PV panels and battery energy storage—during an extended grid power failure. Simulation results show that the RL controller performs similar to an MPC controller that was proposed in our prior work in servicing the primary load (refrigerator). The sensing and forecast information used by RL and MPC are similar, and simulations indicate they both have some robustness to forecast errors. But the real-time computational effort in RL is five orders of magnitude lower than that in MPC. No special purpose solver is required for the real-time control computation by the RL controller, and its computations can be performed in a low power processor—even possibly in a microcontroller—whereas the MPC controller requires an MILP solver and a desktop or a laptop computer.

The RL controller's secondary load performance (servicing lights and fans) is found to be poorer than MPC. RL is sensitive to many design choices such as the cost function and the penalties. It might be possible to improve the performance of RL in servicing the secondary load by varying these design choices. We plan to explore this in the future.

References

[1] U.S. Global Change Research Program, "2014: Highlights of climate change impacts in the United States: The third national climate assessment," May 2014.

[2] A. Kwasinski, F. Andrade, M. J. Castro-Sitiriche, and E. O'Neill-Carrillo, "Hurricane Maria effects on Puerto Rico electric power infrastructure," *IEEE Power and Energy Technology Systems Journal*, vol. 6, no. 1, pp. 85–94, 2019.

[3] United States Energy Information Agency, "EIA-930 Hurricane Irma Impact Tracking Report Friday September 15, 2017, 15:00 hours," https://www.eia.gov/special/disruptions/archive/hurricane/irma/pdf/Irma_20170915_1500_report.pdf, 2017.

[4] M. Gallucci, "Rebuilding Puerto Rico's Power Grid: The Inside Story," *IEEE Spectrum*, 2018.

[5] N. Kishore, D. Marqués, A. Mahmud, M. V. Kiang, I. Rodriguez, A. Fuller, P. Ebner, C. Sorensen, F. Racy, J. Lemery, L. Maas, J. Leaning, R. A. Irizarry, S. Balsari, and C. O. Buckee, "Mortality in Puerto Rico after hurricane Maria," *New England Journal of Medicine*, vol. 379, no. 2, pp. 162–170, 2018.

[6] United States Energy Information Agency, "Annual electric power industry report (survey form no. eia-861)," *US Energy Information Administration, Washington, DC, USA*, 2019.

[7] N. Gaikwad, N. S. Raman, and P. Barooah, "Smart home energy management system for power system resiliency," in *2020 IEEE Conference on Control Technology and Applications (CCTA)*, 2020, pp. 1072–1079.

[8] A. Di Giorgio and L. Pimpinella, "An event driven smart home controller enabling consumer economic saving and automated demand side management," *Applied Energy*, vol. 96, pp. 92–103, 2012.

[9] A. Anvari-Moghaddam, H. Monsef, and A. Rahimi-Kian, "Optimal smart home energy management considering energy saving and a comfortable lifestyle," *IEEE Transactions on Smart Grid*, vol. 6, no. 1, pp. 324–332, 2014.

[10] F. Brahman, M. Honarmand, and S. Jadid, "Optimal electrical and thermal energy management of a residential energy hub, integrating demand response and energy storage system," *Energy and Buildings*, vol. 90, pp. 65–75, 2015.

[11] M. Marzband, H. Alavi, S. S. Ghazimirsaeid, H. Uppal, and T. Fernando, "Optimal energy management system based on stochastic approach for a home microgrid with integrated responsive load demand and energy storage," *Sustainable cities and society*, vol. 28, pp. 256–264, 2017.

[12] M. J. Sanjari, H. Karami, and H. B. Gooi, "Analytical rule-based approach to online optimal control of smart residential energy system," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 1586–1597, 2017.

[13] B. V. Mbuwir, M. Kaffash, and G. Deconinck, "Battery scheduling in a residential multi-carrier energy system using reinforcement learning," in *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, 2018, pp. 1–6.

[14] J. R. Vázquez-Canteli and Z. Nagy, "Reinforcement learning for demand response: A review of algorithms and modeling techniques," *Applied Energy*, vol. 235, pp. 1072 – 1089, 2019.

[15] B. V. Mbuwir, D. Geysen, F. Spiessens, and G. Deconinck, "Reinforcement learning for control of flexibility providers in a residential microgrid," *IET Smart Grid*, vol. 3, no. 1, pp. 98–107, 2020.

[16] W. I. Schmitz, M. Schmitz, L. N. Canha, and V. J. Garcia, "Proactive home energy storage management system to severe weather scenarios," *Applied Energy*, vol. 279, p. 115797, 2020.

[17] J.-J. A. Prince, P. Haessig, R. Bourdais, and H. Gueguen, "Resilience in energy management system: A study case," *IFAC-PapersOnLine*, vol. 52, no. 4, pp. 395 – 400, 2019, iFAC Workshop on Control of Smart Grid and Renewable Energy Systems CSGRES 2019.

[18] ——, "Stochastic modelled grid outage effect on home energy management," in *Proceedings of the Conference on Control Technology and Applications (CCTA'20)*, August 2020.

[19] A. M. Devraj and S. Meyn, "Zap Q-learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 2235–2244.

[20] N. S. Raman, A. M. Devraj, P. Barooah, and S. P. Meyn, "Reinforcement learning for control of building HVAC systems," in *American Control Conference*, July 2020.

[21] G. M. Masters, *Renewable and efficient electric power systems*. John Wiley & Sons, 2013.

[22] B. Diouf and C. Avis, "The potential of li-ion batteries in ecowas solar home systems," *Journal of Energy Storage*, vol. 22, pp. 295 – 301, 2019.

[23] B. Cui, C. Fan, J. Munk, N. Mao, F. Xiao, J. Dong, and T. Kuruganti, "A hybrid building thermal modeling approach for predicting temperatures in typical, detached, two-story houses," *Applied Energy*, vol. 236, pp. 101 – 116, 2019.

[24] Centers for disease control and prevention, "Keep food and water safe after a disaster or emergency," https://www.cdc.gov/disasters/foodwater/facts.html, 2019, last accessed: Jan, 17, 2020.

[25] Gurobi Optimization, LLC, "Gurobi optimizer reference manual," 2019. [Online]. Available: http://www.gurobi.com