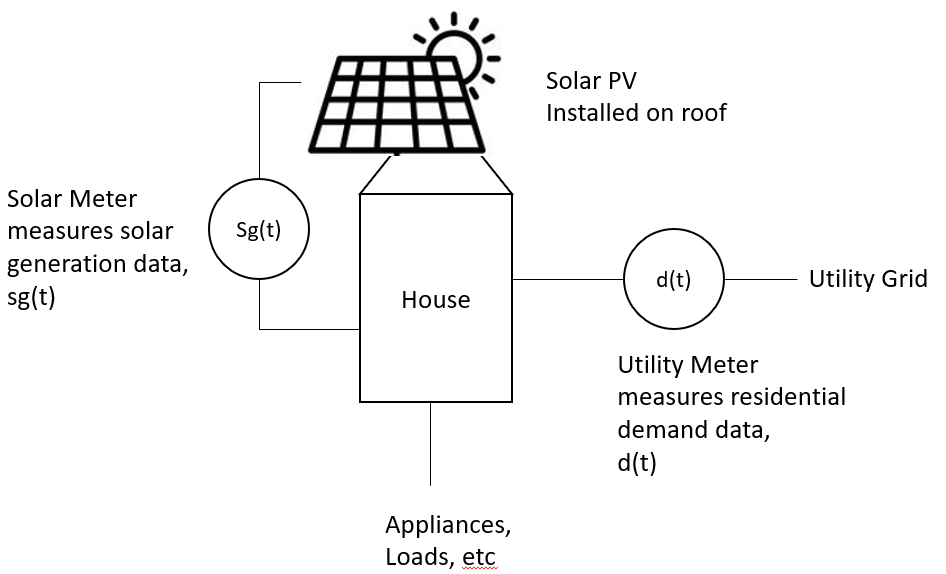
**1. Abstract:**

Power utility companies are investigating energy storage solutions in response to rising energy prices. This project proposal examines an Energy Storage as a Service (ESaaS) model, which divides a large Physical Energy Storage System (PESS) capacity into smaller virtual shares. These virtual shares are rented as Virtual Energy Storage System (VESS) to residential customers who have installed household solar photovoltaic generation (PV). The VESSs mimic residential energy storage systems and assist participants by maximizing self-consumption of the participant’s renewable solar resources. This approach employs Monte Carlo simulations to quantify probability distribution functions to predict household solar generation and load. A stochastic Approximate Dynamic Programming (ADP) technique using a Double Pass algorithm is used to determine the optimal VESS charging and discharging schedule based on random scenarios generated from the Monte Carlo simulations. This approach and the VESS model are applied to a case study in Las Vegas, Nevada, using actual meter data from the locals system peak load month of August. The study will also contrast the Monte Carlo predictions with real data to evaluate the stochastic model's precision and dynamic programming optimization.

**2. Simple electrical diagram:**



**3. Objective Function and constraint equations**

Definition of Variables:

= the resident’s energy cost ($) during time period, t

= the resident’s energy demand (kWh) during time period, t

= the resident’s solar generation (kWh) during time period, t

= the energy charged or discharged (kWh) by the resident’s battery during time period, t

= the residential energy price ($/kWh) during time period, t

= the charge or discharge power rate (kW) of the battery during time period, t

= the State of Charge of a resident’s battery during time period, t

= the one-way charge or discharge efficiency rating

N = number of intervals in a 24 hour day

= interval duration, i.e. for 30-minute intervals

Electricity cost function for the residential household’s electricity costs at time, t:

Objective function: minimize the daily electricity cost for a homeowner

Load and solar data are 30-minute kWh intervals. There are 48 half-hour intervals in a day. N = 48

Subject to the following constraints.

Equations 2 - 4 are constraints that en sure the Approximate Dynamic Programming Algorithm does not choose actions that violate the energy storage battery’s physical requirements.

Equations 5 and 6 define positive Power means battery charging, and negative Power means battery discharging.



Equations 7 and 8 do not allow the battery to be used for selling electricity back to the grid. Equation 7 limits the battery to supply the energy that the solar cannot provide. Equation 8 allows the battery to charge only when there is extra solar energy available.

Equations 9 and 10 limit how much energy that the battery can charge during times when solar is greater than the load.

Equations 11 and 12 limit how much energy that the battery can discharge during times when load is greater than solar.

**4. Requirements**

**Requirement 1** – Building a stochastic Probability Distribution Function to simulate solar and load for Monte Carlo simulations

Real half hourly solar generation data and demand data will be provided. Demand data is random at all hours of the day. It may be zero during some time periods. Solar data is assumed to be random. It is zero during the time of day where there is no sun.

Python code is provided to fit half hourly PDF to the solar data and a PDF to the load data.

**Requirement 2** – Create simulated day-ahead forecast for solar and demand.

Sample the solar generation and demand PDFs to simulate a day of random solar and demand data. This sample data will be used to run a Monte Carlo analysis using many episodes.

**Requirement 3** – Root Mean Square Error, Mean Absolute Error, and Coefficient of Variation.

Calculate the errors (RMSE, MAE) and the Coefficient of Variation (COV) for the real load and real solar data compared to the random simulated load and solar data from Requirement #2. These errors are useful to determine if the PDF functions and random simulated load and solar data are accurate compared to real data.

**Requirement 4** – Monte Carlo analysis of Approximate Dynamic Programming (ADP) using Double Pass Algorithm

See the Pseudocode for instructions. Make any corrections or improvements that are needed to make the algorithm run correctly and efficiently.

For background information about Approximate Dynamic Programming or the Double Pass Algorithm, refer to Sections 2.3 and Algorithms 1 and 2 in Section 2.3.2 in the Approximate Dynamic Programming by Practical Example.pdf

Assumptions:

1. At the beginning of every episode, the energy storage battery begins at a 0% state of charge.
2. The Action Space = array[ -P\_discharge\_max; P\_charge\_max +1; 1]
   1. The possible actions range from -P\_discharge\_max to P\_charge\_max, incrementing by 1 kW.
   2. The battery doing nothing (“idling”) with zero charge is also a possible action.
3. The Value Function is initialized as zeros for all states
4. The Policy is initialized at “idle” for all states.

The Value, V; of any state, s; any action, a; at time, t, is:

And the policy function is:

If the Value Function is initialized to , the first episode of expected future values will be zero:

A note on Value Function Approximation and Monte Carlo methods:

For all episodes after the first episode, there must be a method to solve or approximate the expected value .

A Monte Carlo method is a type of analysis that relies on repeating a large number of trial episodes using randomly sampled data to obtain numerical results. This is why it is very important to randomly simulate load and solar data from the Probability Distribution Functions in Requirement #2.

If the number of episodes, K, is sufficiently large, then running the ADP Double Pass Algorithm K-number of times will perform a Monte Carlo experiment. The expected future Value functions will approach the mean of the future values.

According to the Law of Large Numbers, the expected value of something approaches its average value as the number of episodes increases:

Using Monte Carlo analysis with large K-number of episodes:

And the policy function:

**Requirement 5 – Plotting graphs, charts, etc.**

Plots are needed to visually show the performance of the system. Some useful plots may include:

1. Simulated demand from PDF versus real demand data
2. Simulated solar from PDF versus real solar data
3. Solar(t) and Demand(t) plotted without the battery
4. Solar(t) and Demand(t) plotted with the charging/discharging of the battery
5. Charge and discharge of the battery by itself
6. Any plots that show the convergence of the Approximate Dynamic Programming algorithm
7. Other useful plots