Lehrstuhl Elektrische Netze
und Erneuerbare Energie

Masterarbeit

Development of an Energy Management System for demand response programs within smart DC houses.

am 23.05.2017 vorgelegte Materarbeit
von Abhik Ghosh

Erstprüfer: Prof. Dr.-Ing. habil. M. Wolter
Zweitprüfer: Prof. Dr.-Ing. A. Lindemann

Betreuer: Dr.-Ing.P Lombardi

Aufgabenstellung im Original

Erklärung

Ich versichere, dass ich diese Masterarbeit ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

23 May 2017
………………………………………………………..
Datum Unterschrift

I Abstract

In the future, distributed renewable energy (DRES) sources will play a crucial role in power system. The mission of the European “energy roadmap 2050” is to achieve low carbon economy. However the European “energy roadmap 2050” can not only be achieved through renewable energy generation but also effort in energy efficiency is needed. The smart DC house is a good option to consider as it increases the energy efficiency. A house with direct current (DC) supply distribution is called a DC house. Similarly a house with an alternating current (AC) distribution is called an AC house. In smart DC house, all AC loads are being replaced by DC loads. For example, AC induction motors are being replaced by brushless DC (permanent magnet) motors (BLDC). Instead of constant speed regulation, variable speed drives (VSD) are being used. Devices operating on DC are being called as DC internal loads. All electronics are DC internal loads. Replacing DC internal loads with direct DC eliminates converter losses. The DC house is more efficient than AC house as it eliminates converter AC/DC and AC/DC/AC losses. This saves a lot of electrical energy in DC house. The energy saving can be further consolidated by using an Energy Management System (EMS) which optimally controls the energy consumption of the smart house, the energy generated and stored within a smart house.

The DC house covers all residential loads such as transport load, electronics appliances load, heating and cooling loads. In this DC house, EMS uses various demand response (DR) algorithms to optimize self-consumption (SC) of photovoltaic (PV) energy, battery energy storage system (BESS), home appliances, heat pump, heat storage system and electric vehicle (EV). The demand response based on pricing signal from grid is being studied and being compared with the without demand response condition. Based on the electricity energy prices, algorithm is being devised to minimize the cost of electrical energy, are being implemented.

Further a cost benefits ratio between optimal sizing and optimal SC is being performed. The optimal sizing of PV arrays, BESS and converters are crucial to save the cost of electrical energy. Evaluations of these algorithms is done via numerical simulations based on real world traces from production systems.

When DC smart house is being compared with AC smart house, the results show that DC smart houses saves 24 % of electrical energy. Optimal sizing saves 30 % of initial investment. Optimal sizing of DC smart house reduces the size of PV array by 24 %, battery size by 25 % as compared to AC smart house. A DC house with price function based DR saves 8 % of consumer electricity bill as compared to a DC house without DR. Levelised cost of electricity (LCOE) of the AC house is 9 cent/kWh and DC house is 7.5 cent/kWh.

II Contents

III List of Figures.

IV List of tables.

V Abbreviation

AC house	House with an alternating current (AC) distribution.
DC house	House with an alternating current (DC) distribution.
AC/DC	Rectifier.
DC/AC	Inverter
AC/DC/AC	Converter AC to DC then DC to AC.
EER	Energy efficiencies ratio
VFD	Variable frequency drive.
DR	Demand Response.
Ct/kWh	Cents per kWh.
P	Power.
PV	Photovoltaics.
PPV	Power photovoltaic
Heat grid	Heat pump load
House grid	Home appliances
ESupply	Electrical energy supplied by grid.
TZone	Temperature of zone (inside house Temperature)
TAmbient	Temperature of ambient.
CP	circulation pump.
PCP	Power consumed by circulation pump
ECP	Electrical energy consumed by circulation pump.
qVRef	Reference volume given by controller to the system.
EHeat	Thermal energy required for heating or cooling inside the room.
QHeat	Thermal power required for heating or cooling inside the room.
Price function	Price function which is either active or inactive
Grid	Supply required electrical energy and power to the system
TMedium	Temperature of the fluid (either upstream or downstream)
qvMedium	Flow of the fluid (either upstream or downstream)
Volume controller	Controls volume of reference flow to be given to pump.
EV charging station	Electric vehicle charging station

VI Symbolverzeichnis

$\mathrm{PV}_{\mathrm{gen\_p}}$	Photovoltaic peak power generation in kW
$\mathrm{PV}_{\mathrm{geni}}$	Instantaneous power generation in kW.
$R_{\mathrm{radiation}}$	Radiation density in $\mathrm{W/m^2}$
η_{modules}	Efficiency of PV modules
$\eta_{\mathrm{converter}}$	Efficiency of converter
A_{area}	Area needed for PV modules in $\mathrm{m^2}$
$Energy_{BESS_{t+1i}}$	Energy of BESS in kWh
$Energy_{grid}$	Energy from thr grid
$BESS_{charge}$	Charging energy
$BESS_{discharge}$	Discharging energy
$P_{heatload}$	Power for heat pump kW
$P_{totalConsumptioni}$	Power for total house
$P_{houseload}$	Power for home appliances
$P_{ElectricVechicle}$	Power for EV
P_{Gridi}	Power from grid
P_{BESSi}	Power from BESS
$\mathrm{cost}_{\mathrm{of}_{\mathrm{PV}}}$	Cost of PV modules
CRF	Cost of recovery factor
$\mathrm{Cost}_{\mathrm{of}_{\mathrm{grid}}}$	Grid electricity prices
DOD	Depth of discharge
SOC	State of charge
$\mathrm{Converter}_{\mathrm{size}}$	Size of the converter
Load	Amount of power needed.
$\forall i$	All step size.

1 Introduction

1.1 Motivation

As residential building energy demand is ever increasing, so to fulfill consumer energy demand, only the expansion of renewable energy will not be sufficient but increase in energy efficiency is also needed. In year 2014, 144 billion kilowatt hours [bn kWh] of electricity is being used in private household which $25\ \%$ of 576 bn kWh of total electrical energy in Germany [1]. Private households could get rid of unnecessary energy costs up to a considerable degree.

As per the German government´s energy concept, a main objective of is to halve primary energy consumption by 2050 in comparison to 2008 [1–4]. The Federal Government Energy´s Efficiency Strategy for Buildings (ESG) can be achieved not only through renewable energy resources generation but also increase in efficiency [5].

The smart DC house is a good option to consider as it increase the efficiency of the house by eliminating converter losses. As demand for electricity is increasing and there a rise in semiconductor devices. So the new efficient way to increase the efficiency of energy is to use DC house instead of AC house. In this study, modeling of an AC house and a DC house is being performed. In a DC house, converter power losses AC/DC/AC and AC/DC can be eliminated. For example, in current practice, PV energy is being used after connected to inverter but DC house uses direct current. Devices operating on DC are being called as DC internal loads. All electronics are DC internal loads. Replacing DC internal loads with direct DC eliminates converter losses. Replacing all induction motors with brushless DC (permanent magnet) motors can save $5 - 15\ \%$ of energy used by AC motors [6,7]. Along with variable frequency drives (VFD) can increase the saving up to $30 - 50\ \%$ [6–9].

The main loads of a house are transport, heat loads, cooling loads and home appliances. So a self-sustaining house with distributed renewable energy sources (DRES) (PV arrays), heat pump, home appliances and electric vehicle are being designed. Figure 1 shows this house can sustain all its energy need through PV arrays, DRES, heat storage system and battery energy storage system (BESS).

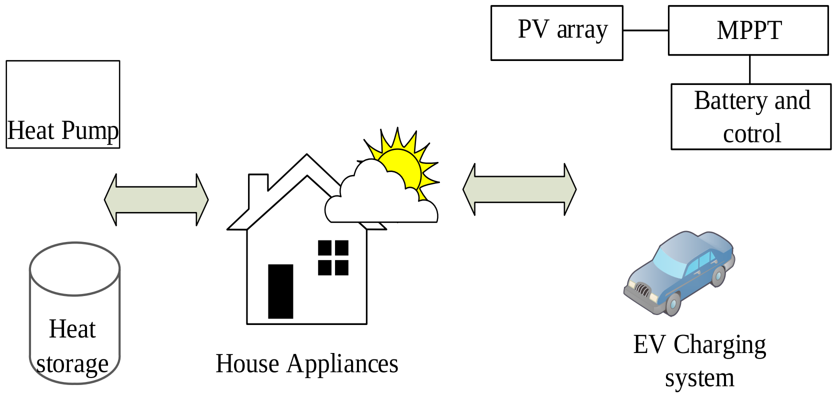

Figure 1 Private Household system with distributed energy sources.

Energy need can not only be sufficient through renewable energy generation but also efficient and optimal consumption is also required. The double peak consumption pattern waste lot of PV energy. The Energy Management System optimally control the consumption pattern of house. It uses demand response algorithm (DR) to optimally control generation and consumption pattern in the house.

EMS uses DR based on self-consumption and grid price. The private household can better use demand response to decrees the electricity billing price. DR in household sector can help in stabilizing grid and incorporate renewable energy and storage system in the grid. So changes in the lifestyles of consumption can create a huge effect Then energy consumption analysis and economic analysis is being performed.

1.2 Objectives and contributions of this study.

The topic of this study is "**Development of an Energy Management System for demand response programs within smart DC houses"**. This research design models of house and simulation of one-year analysis is being done based on Dresden weather data.

Specific objectives are:

- Modelling of smart home system using software SimulationX.
- Development of a mathematical model for the EMS algorithms.
- Development of control algorithm based on electricity price.
- Simulation of the behavior of a DC smart house.
- Analysis of the results.

1.3 Project outline

Chapter 1 gives Motivation and objectives of the study, Chapter 2 gives modeling and simulation. Chapter 3 shows the algorithms of EMS, Chapter 4 shows simulation results and Chapter 5 shows conclusion for this study.

.

2 Modelling and simulation

2.1 Simulation Software.

Software used to use modeling of the system are given below.

SimulationX is a Modelica based program package used for modelling and simulation analysis of the systems. Here Green building library in SimulationX version 7 has been used. This software uses Modelica language [10].

OPENMODELICA is based on Modelica-based modeling and simulation environment [11].

Solver

The **backward differentiation formula** (BDF) is a method for solving of ordinary differential equations. These methods are being implemented for the solution of stiff differential equations [10].

2.2 Modelling.

SimulationX is used for analysis and simulation purpose. SimulationX provides in its Green building library,

2.2.1Process

Figure 2 shows private household energy management system, the house model of floor space of 200 m^2 and height of the zone is 3 m is taken. A DC house and AC house is being modeled with heat pump, demand response controller, heat storage, PV modules, BESS, and electric vehicle. Then weather data for one year (Dresden, Germany) (which include ambient temperature, wind speed, wind direction, solar diffuse radiation and solar direct horizontal radiation) are used to simulate one-year condition with $1 \text{ e} - 6$ second step size.

The algorithms of energy management system based on self-consumption optimizer, BESS optimizer and demand Response (price function algorithm) are being used for the process. A variable tariff is assumed. These process has following components.

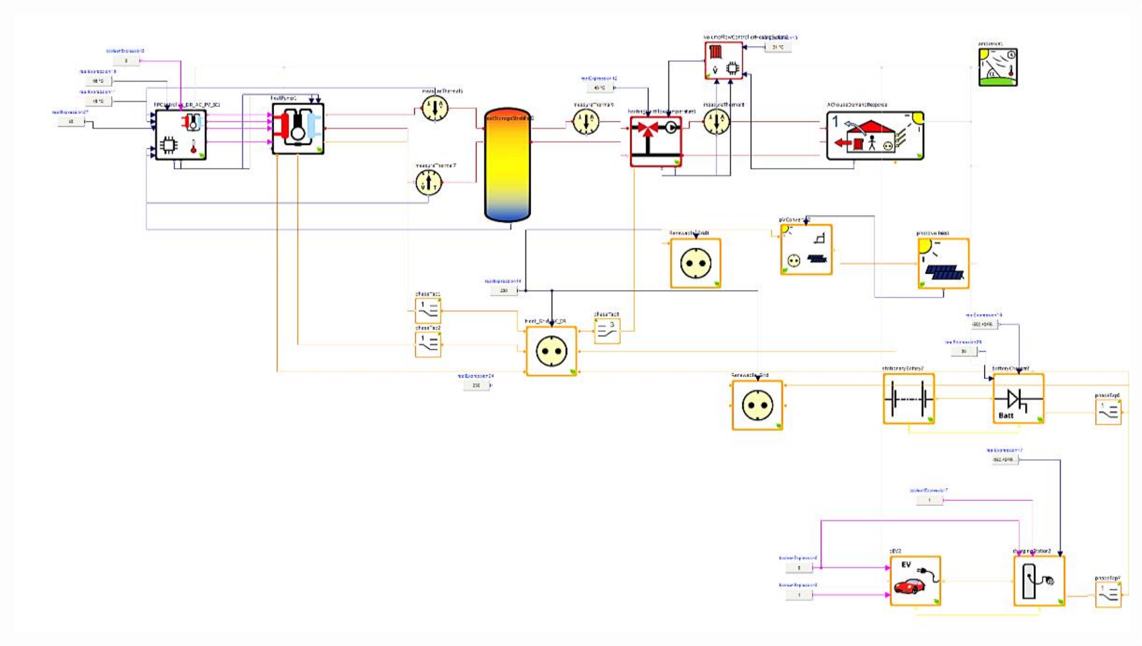

Figure 2 Housing system in SimulationX.

Heat pump use to heat the zonal temperature of the room. Heat pump includes source pump, circulation pump and compressor.
Grid House system is connected to grid which use to distribute required electrical energy and power to the system.
Heat storage is used to store the liquid which is pumped to either heat or cool the zonal temperature of the room, it acts as buffer storage. The storage capacity of 750 l and cylinder diameter of 0.75 m is used. The heating medium is water. Parameters of specific heat capacity and densities of water are used. It can hold 30 kWh heat load
Controller is designed to control the heating and cooling as per the zonal temperature requirement and pricing requirement of grid.
House model is used from SimulationX green building which is designed very close too real model.
Thermal and flow measurement sensor are being used.
Weather data of Dresden, Germany is taken, in that ambient temperature, radiation, wind speed, wind direction and solar radiation diffusion data for one year is inserted.
Electric vehicle: Here in simulation, EV with BLDC and battery.
Battery Energy Storage System: Lithium-ion battery based on ZARC model is being used.
Photovoltaic array: Simulation uses Panasonic HIT® N330HIT (Heterojunction Intrinsic thin layer) which has module conversion efficiency 22.5 %

2.2.2Building zone.

The Building Zone model simulates various disturbances on the internal temperature in a building (See Figure 3). Here influence of person on the heat demand and electricity needed is also taken into account. This building zonal model (one zone) can be coupled with others zones to make a complicated building system.

The zone temperature depends on various disturbances:

- Heat transmission through walls, windows and other boundaries (e.g. doors) (q_{trans})
- Solar yields (Q_{sun})
- Ventilation losses (q_{vent})
- Internal yields and losses via persons as well as electricity and water usage (q_{ele}, q_{person})
- Internal heat storage (air, walls, inner masses) (Q)

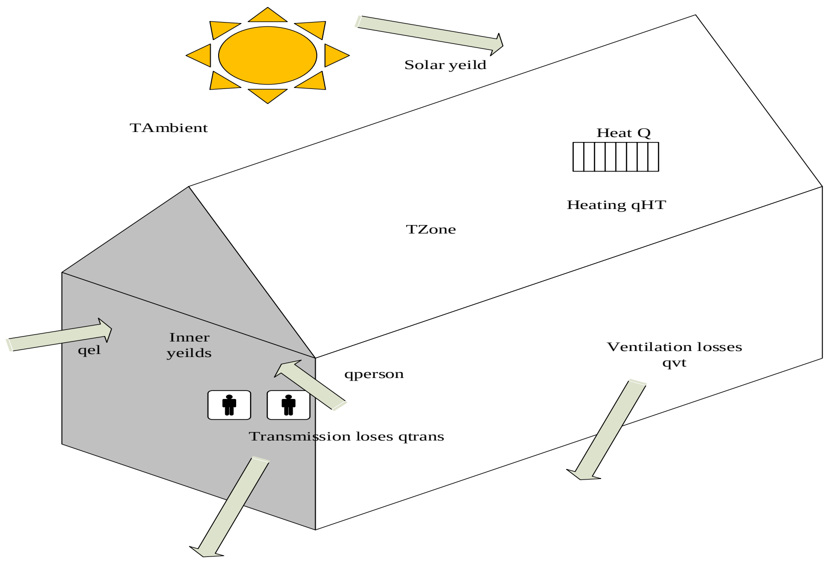

Figure 3 Private house model in which weather data is inserted

Equation 1, equation 2, equation 3, equation 4 and equation 5 shows the behavior of various influences on house model.

$$Q = c_p * A * (T_{zone} - T_{ambient}) \tag{1}$$

$$Q_{sun} = \sum g_c * A * I \tag{2}$$

$$q_{trans} = U * A * (T_{zone} - T_{ambient}) \tag{3}$$

$$q_{vent} = C_p * V_{air} * ß_{air} * (T_{zone} - T_{ambient}) \tag{4}$$

$$Q_{sun} + q_{elec} + q_{person} - q_{vent} - q_{trans} = Q \tag{5}$$

Where

Q	Heat transferred per unit time$\mathrm{W/m^2}$.
c_p	Specific heat capacity.
A	The area of the surface $\mathrm{m^2}$.
T_{zone}	Temperature inside the room in k.
$T_{ambient}$	Temperature of the atmosphere.
q_{vent}	The heat transferred per unit time due to ventilation losses.
V_{air}	Volume of air from boundary $\mathrm{m^3}$.
$ß_{air}$	Density of boundary.
Q_{sun}	The heat transferred per unit time due to sun light.
g_c	Heat transfer coefficient.
I	Rays incident on the surface with respect to time.
q_{elec}	The heat transferred per unit time due to electricity.
q_{person}	Heat transferred per unit time due to person.
q_{trans}	Heat transferred per unit time due to transmission losses.
U	Heat transmission value of boundry.

2.2.3Modelling of AC and DC house

To compare the saving in DC house, an AC house where electrical distribution is AC and a DC house where distribution is DC are being modeled. DC house eradicates DC/AC/DC power conversion losses to DC appliances. Figure 4 and Figure 8 shows an AC house and Direct-DC house.

Power conversion efficiencies

Table 1 gives the details about converter efficiency. These converter efficiencies are being used to calculate the energy losses.

Table 1 Engineering parameters for simulation.

Engineering parameters	**Min**	**Max**	**unit**	**source**
Rectifier efficiency	90 %	95 %		[6–8,12,13,13]
Inverter efficiency	90 %	95 %		[6–8,12,13]
DC-DC efficiency	90 %	95 %		[6–8,12,13]
MPPT	90 %	95 %		[6,7,13,14]
Battery charge efficiency	85 %	90 %		[6,7,13,14]

Battery discharge efficiency	90 %	95 %		[6,7,13,14]
BLDC gain efficiency	5 %	15 %		[6,7,13,14]
VFD gain efficiency	30 %	50 %		[6,7,13,14]
DC condenser gain efficiency	16 %	22 %	EER	[6,7,13,14]
DC heat pump gain efficiency	15 %	25 %	COP	[6,7,13,14]

2.2.3.1 **AC house**

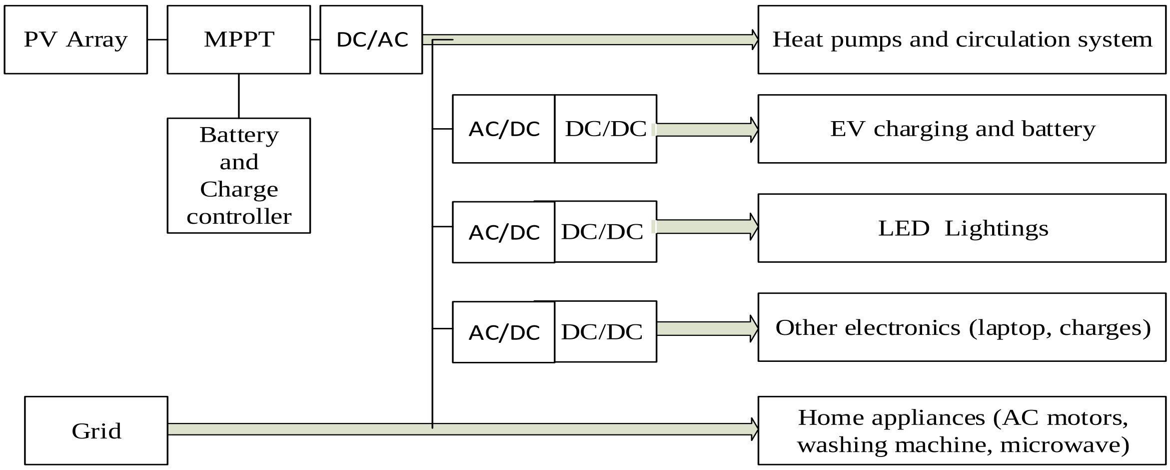

Figure 4 AC house with AC power distribution.

In AC house, all power distribution system is AC distribution and all appliances accept AC supply (either single phase or 3 phase supply). The AC house uses 110 V_{AC} Or 220 V_{AC} configuration. The AC house is connected to grid. PV array are connected to MPPT, battery charge controller and a rectifier. The rectifier converts DC to AC supply and MPPT (Maximum power point tracker) is a DC/DC converter which optimizes the maximum power produced of PV array by tracing voltage and current.

The AC house has following loads:

1. **Heat pumps and circulation system:** Heat pump consists of a compressor, a source pump, a circulation pump and a variable frequency drive. All motors are AC induction motors. The variable frequency drive (VFD) uses rectifier for variable speed control. The AC supply is rectified to DC before given to VFD. In comparison, DC house uses BLDC which is $5 - 15$ % more efficient than AC induction motors and with reduced size [6,15]. VFD uses direct DC supply saving rectifier conversion losses. BLDC with variable speed drives (VSD) can save up to $30 - 50$ % of energy used induction motors. Pumping, refrigeration, space cooling and space heating uses VFD with motors. With Dresden

weather data, simulation is being run for one year and heat pump energy consumption data is obtained[6,15].

2. **Home appliances**[9,15–21] :

Home appliances consist of Refrigeration loads (fridge and deep fridge), washing and drying load e.g. washing machine and dryer with heat pump, cooking loads. Figure 5 distribution of house loads of 3 member house hold. Figure 6 shows the average load profile of average household appliances in Europe.

i. **Refrigeration load** uses vapor-compression cycle which include compressor, circulation pump and VFD. Similar to headloads e.g. Heat pump, it uses all induction motors and rectifier. So the conversion losses take place due to AC induction motor and rectifier.

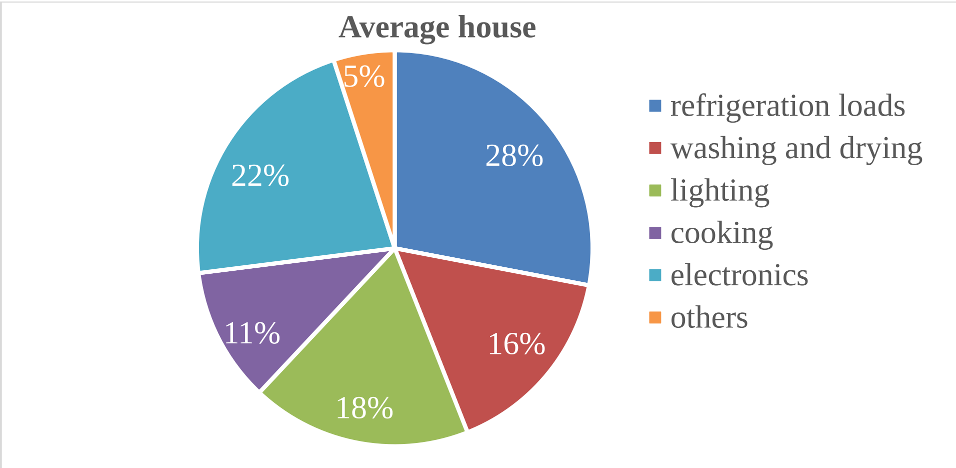

Figure 5 Average house with 3 household members [9,16–22].

ii. **Washing and drying load**: washer uses induction AC motor with VFD. Dryer uses heat pump with VFD. So the conversion losses take place due to AC induction motor and rectifier.

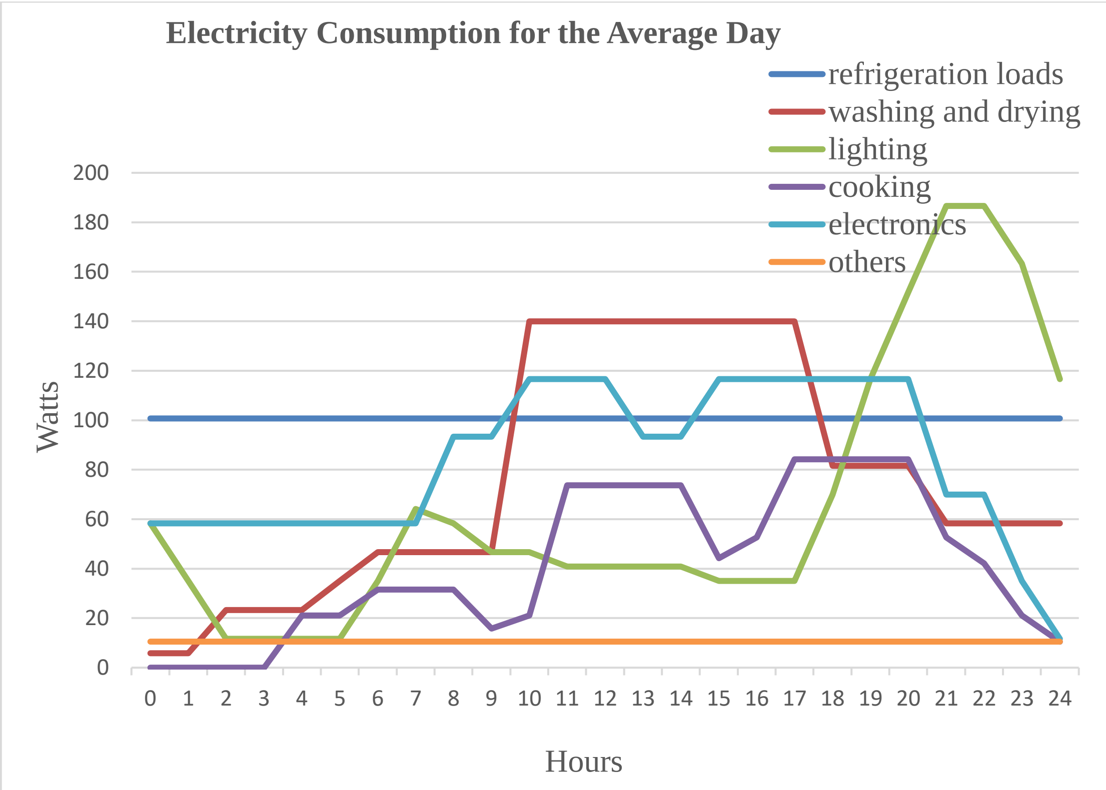

Figure 6 Energy consumption for average day of an average household in Europe [9,16–22].

3. **Other electronics**: This includes Laptops, chargers, microwave oven, other internet of things, controller, computers, phones etc. All these electronics works on direct current but in AC house uses rectifier to convert AC distribution to DC.
4. **Electric vehicle**: Electric vehicle uses BLDC and VFD. The charging station converts AC to DC.
5. **Lighting loads**: AC House uses LED which uses AC/DC and DC/DC converter are required. The PV generated DC is get inverted to AC in AC distribution house. Then this AC rectified to DC before getting use in LED.
6. **BESS**: AC Distribution uses AC/DC and DC/DC converter. These losses due to converter power conversion can be eliminated using DC architecture.

Table 2 Optimized parameters for AC house simulation for one day.

AC house loads	Initial state	unit	source
Heat pump		12 kWh	SimulationX
Home appliances		10 kWh	[9,16–21]
Electric vehicle	50 % SOC	11 kWh	Assumption

Figure 8 shows the AC house loads considered for simulation. Average winter day load profile is taken. The simulation determine the heat load based on weather conditions. Table 2 shows the optimized parameters used for EMS. The heat pump load data is taken from simulationX, average appliances load from various references and vehicle load of 22 kWh is assumed [9,16–22]. Figure 7 shows the loads distribution of an AC house during winter day.

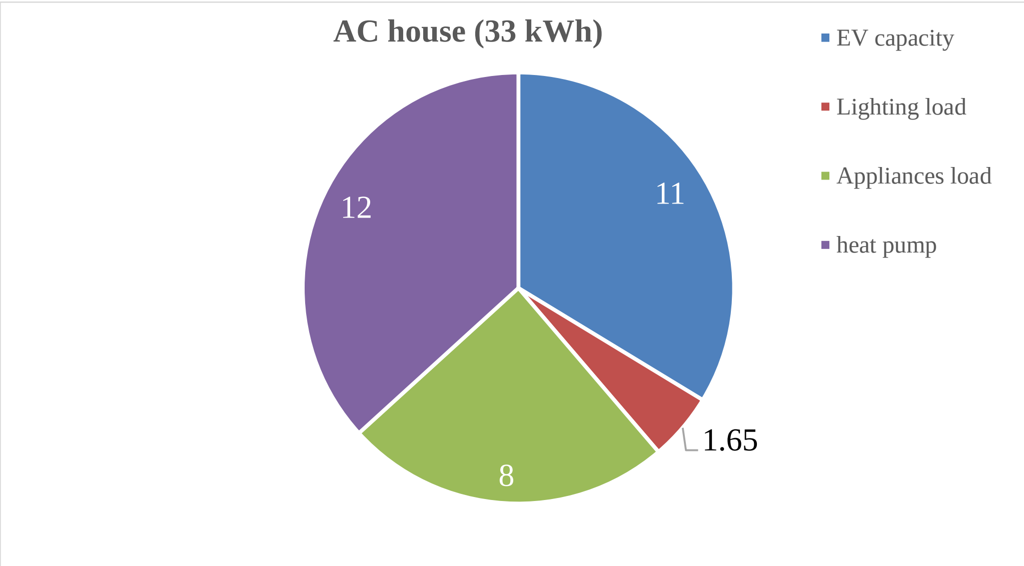

Figure 7 Average energy consumption in AC house during winter.

2.2.3.2 **DC house model**

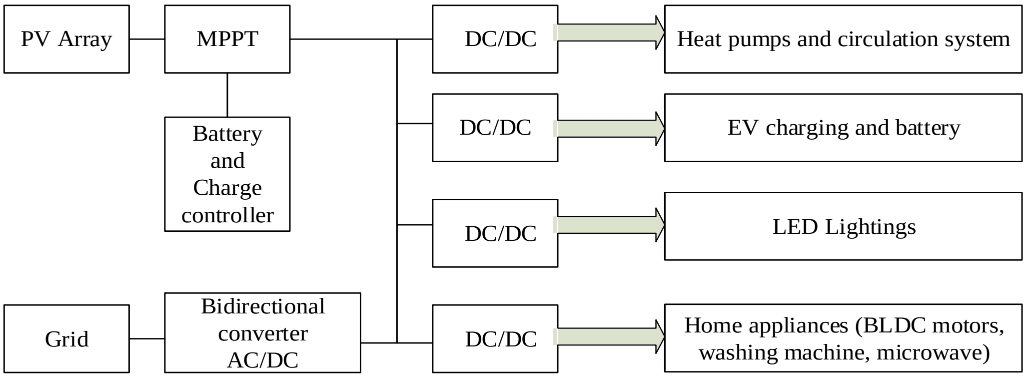

Figure 8 DC house with DC power distribution.

In DC house, all power distribution system is DC architecture and all appliances accept DC supply. The DC house uses 48 V_{DC} or 24 V_{DC} configuration. The house is connected to grid using bidirectional converter (AC/DC or DC/AC). When DC house is taking power from grid, AC/DC Converter is used. Net metering is used, whenever PV array produces excess power, it is send to the grid using DC/AC converter. This bidirectional converter occurs extra losses in DC House. The PV array is connected by MPPT but no rectifier is needed. DC House uses BLDC and VFD.

The DC house has following loads:

1. **Heat pump**: Heat pump works on vapor compression cycle. Compressor, source pump and circulation pump has BLDC and VFD. BLDC has many advantages over AC induction motors. As it is uses VFD, then using BLDC is better option than AC induction motor [20]. The BLDC eliminates the rotor magnetization losses like eddy current losses which is inherit to AC induction motor. So heat losses due to magnetization can be eliminated. The lower cooling needs in BLDC reduces energy consumption. Moreover BLDC motors are more compact. So it reduces the manufacturing cost due to reduction in size as compare to AC induction motors [20]. BLDC with variable speed drives (VSD) can save up to $30 - 50$ % of energy used induction motors.

2. **Home appliances:**

Refrigeration loads (fridge and deep fridge), washing and drying load e.g. washing machine and dryer with heat pump, cooking loads.

i. **Refrigeration load** uses vapor-compression cycle which include compressor, circulation pump and VFD. Here in DC house, heat pump uses BLDC and VFD. So it reduces conversion losses take place due to AC induction motor and rectifier in AC house.

ii. **Washing and drying load**: washer uses BLDC with VFD. Dryer uses heat pump with VFD. So the conversion losses are eliminated due to use of BLDC and rectifier.

3. **Electric vehicle** consists of BLDC and battery system. Here no inverter is required, power from battery can be used directly. This eliminates inverter losses which is required in an AC house.

4. **Lighting loads** consists of LED which can use direct DC using a DC/DC converter. This eliminates rectifier losses which is required in an AC house.

Calculation of DC loads

A new DC load profile is calculated as a function of new DC production and end user efficiency. Equation 6 shows the function to calculate new DC loads as a function of new efficiencies. Compare AC house and DC house and take out the converter losses accounted for.

- **DC load modelling**

$$DC_{load} = \frac{AC_{load} * \eta_{\mathrm{AC_{production}}} * \eta_{\mathrm{AC_{use}}}}{\eta_{\mathrm{DC_{production}}} * \eta_{\mathrm{DC_{enduse}}}} \tag{6}$$

Where

AC_{load}	Energy load in an AC house in kWh.
$\eta_{\mathrm{AC_{production}}}$	Efficiency of produced energy.
$\eta_{\mathrm{AC_{use}}}$	Efficiency of used energy.
DC_{load}	Energy load in a DC house.
$\eta_{\mathrm{DC_{enduse}}}$	Efficiency of produced energy.
$\eta_{\mathrm{DC_{production}}}$	Efficiency of used energy.

Figure 9 shows the new calculated DC loads for DC house. This is average winter loads distribution for DC house. The calculation based on equation 6 calculates new DC load.

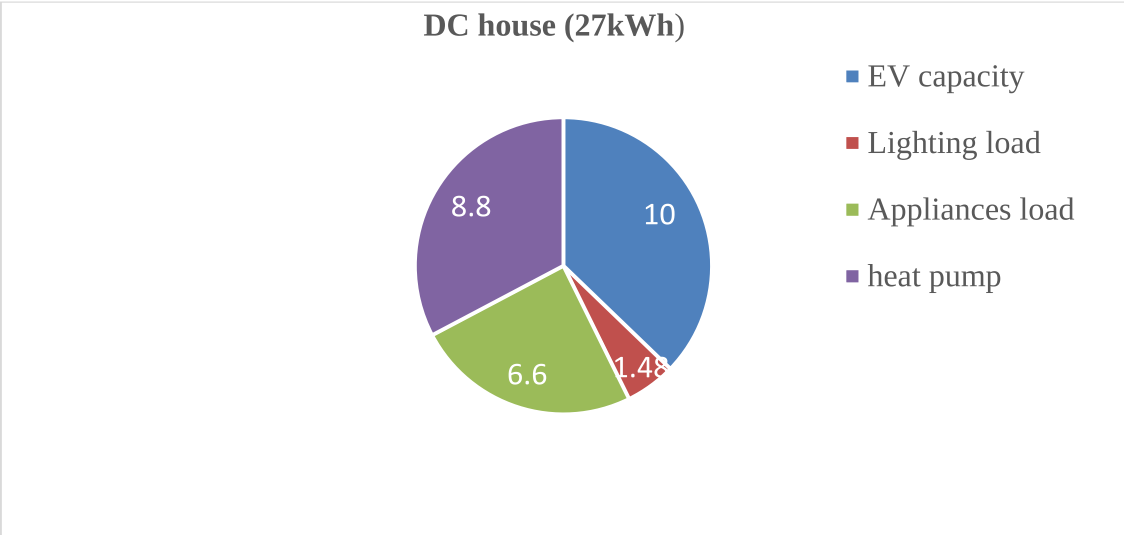

Figure 9 Average energy consumption in DC house during winter.

Saving due to DC house

Figure 10 shows the final structure DC and AC house

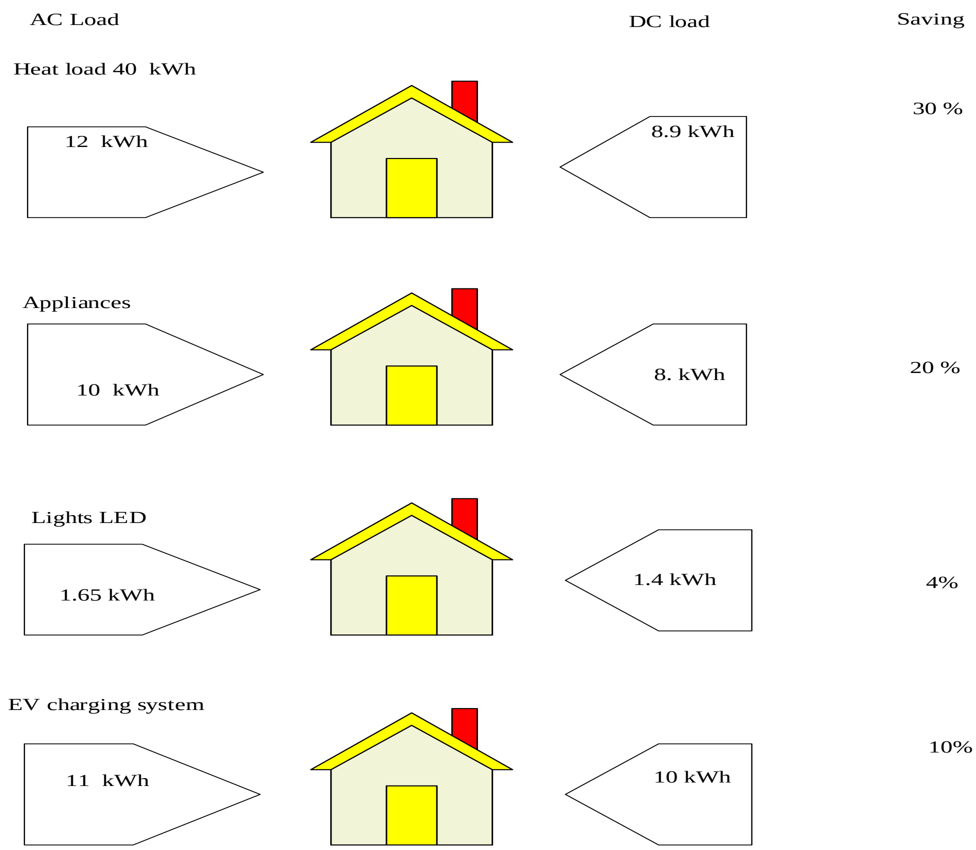

Figure 10 Saving in DC house.

Using DC supply distribution, power converter losses can be eliminated. The following steps reduces consumption and increase the efficiency of the house.

1. Replace all AC internal loads with DC internal loads. E.g. all induction motors with VFD can be replaced by BLDC with VFD. This increases the efficiency by 30-50% compare to AC induction motors with VFD [7,8,13,23]. Figure 10 shows 30% of saving.
2. Use all appliances with DC heat pump with VFD instead of resistance heating.
3. All electronics consume direct DC, replace the AC distribution to DC distribution. DC distribution eliminates AC-DC conversion losses. Figure 10 shows saving of 20% [7,8,13,23].
4. Better Energy management system can drastically improve the efficiency of the house.

2.2.4Weather data

Figure 11 shows the ambient temperature for one year in Dresden, Germany. The temperature varies from $-10°\mathrm{C}$ to$30°\mathrm{C}$. Figure 12 and Figure 13 gives the direct and diffused solar radiation of the location. The direct solar radiation power density radiation varies from0 to 800 $\frac{\mathrm{W}}{\mathrm{m}^2}$. The diffuse solar radiation power density radiation varies from0 to 450 $\frac{\mathrm{W}}{\mathrm{m}^2}$. The graph for one year starting from January is shown.

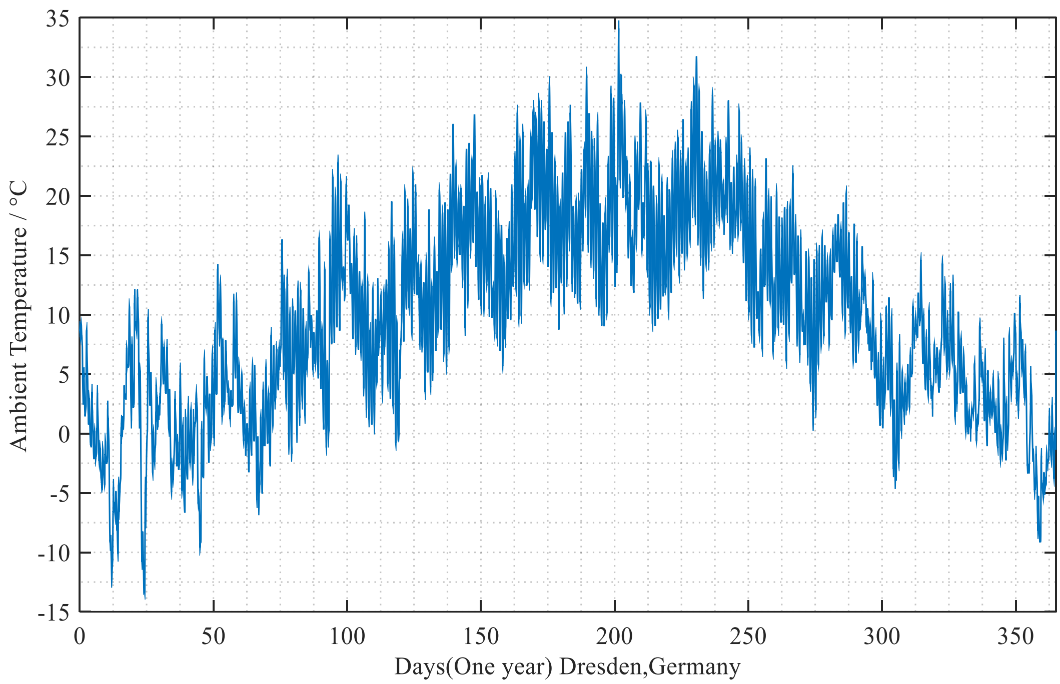

Figure 11 Ambient Temperature of Dresden, Germany.

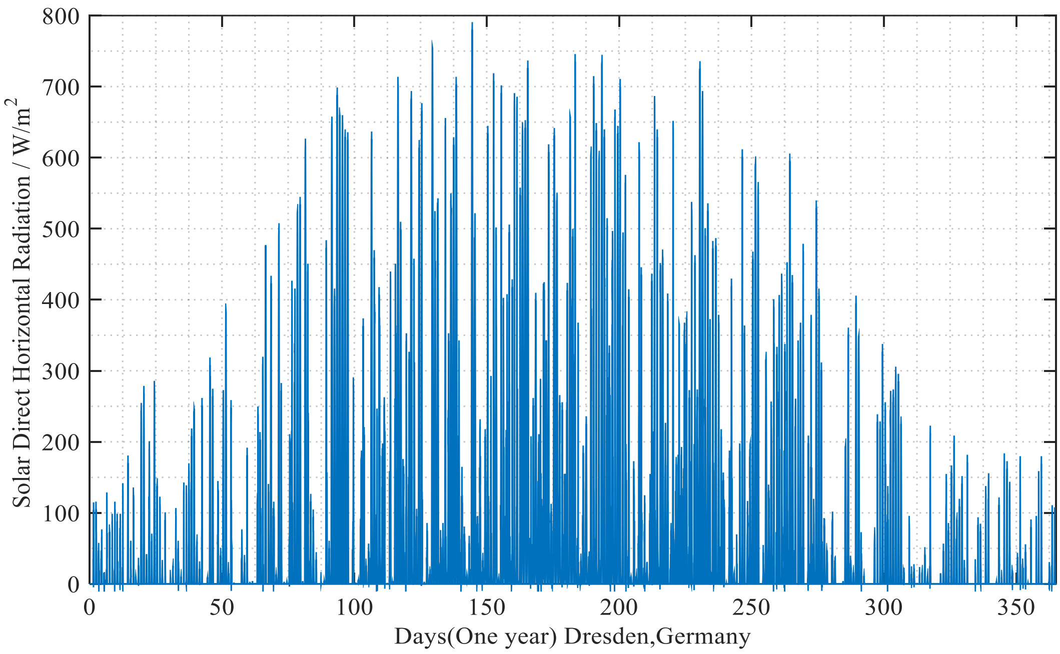

Figure 12 Solar direct horizontal radiation of Dresden, Germany.

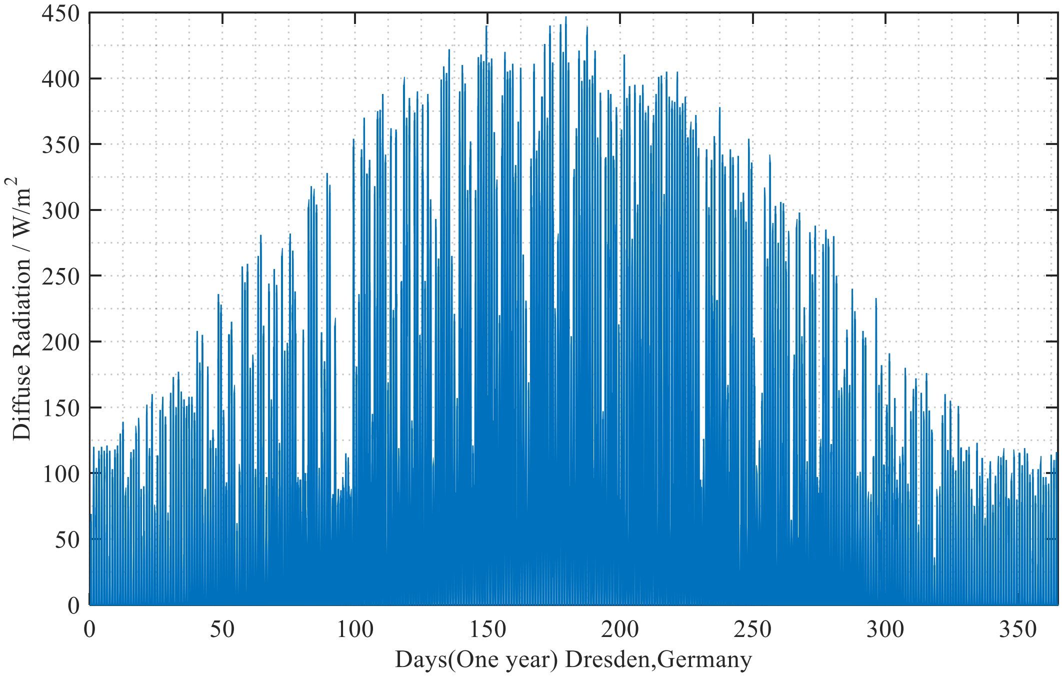

Figure 13 Diffuse radiation of Dresden, Germany.

2.2.5Heat pump

In heat pump, cooling or heating of fluid takes place which is built of six basic components compressor, source pump, expansion valve, evaporator, condenser and circulation pump. The process works on vapor-compression refrigeration cycle. Figure 14 shows the operation of heat pump.

Compressor uses to increase pressure by reducing volume and uses to transport fluid through pipe. It works on Rankine cycle of thermodynamics.

Source pump is an **air source heat pump** (ASHP) is a system which transfers heat from outside to inside a building, or vice versa. Under the principles of vapor compression refrigeration, an ASHP uses a refrigeration system involving a compressor and a condenser to absorb heat at from air and release it at house. These are used as space heaters or cooler, and are sometimes called "reverse-cycle air conditioners".

In household heating system, ASHP absorbs heat from outside air and releases it inside the building, as hot air, hot water-filled radiators, under floor heating and/or domestic hot water supply. The same system can often do the reverse in summer, cooling the inside of the house.

Circulation pump is a pump used to circulate fluids in a closed circuit. It circulates the fluids within a closed system.

There are two modes of working of heat pumps:

- In **heating mode,** evaporator is used as outer coil, while condenser is the indoor coil. The fluid gets heated in evaporator in which heat is added from external source (outside air or soil), this heated fluid is being carried by a pump to heat housing system through heat exchanger. Then fluid is going through an expansion valve and condenser. This cycle is called vapor compression cycle where heat is added and removed as per the requirement.
- In **cooling mode,** the cycle is similar, but now condenser is the outdoor coil and evaporator is the indoor coil (which reaches a lower temperature). This is the familiar mode in which air conditioners operate.

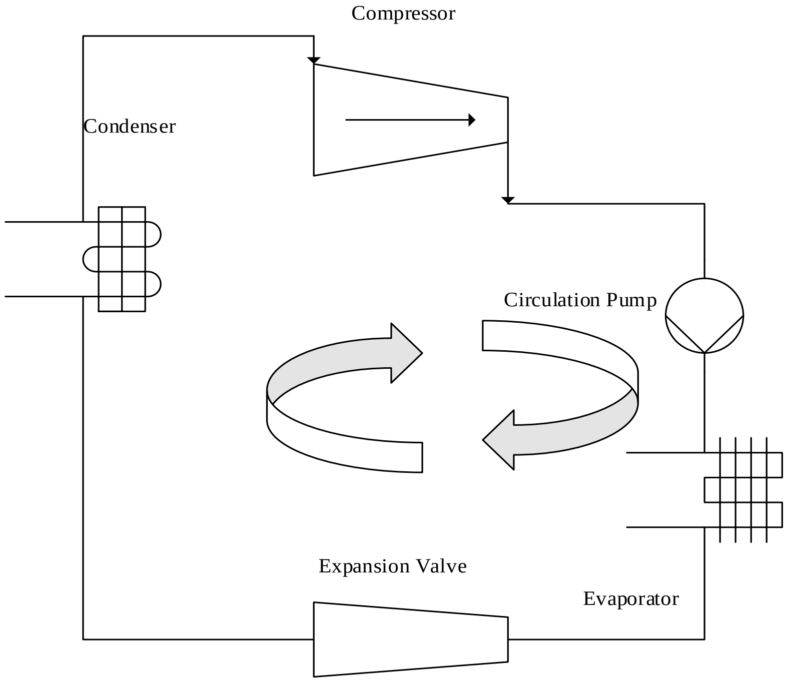

Figure 14 Heat pump process.

Convection law

$$Q = h_c * A * (T_s - T_\infty) \quad (7)$$

Where

Q	Heat transferred per unit time.
h_c	Heat transfer coefficient.
A	Area of the surface.
$T_s - T_\infty$	Temperature difference two surfaces where heat transfer s taking place.

2.2.6 Battery electric vechicle

The distributed renewable resources will play a crucial role in future. EV holds huge potential in DRES. In future, EV will replace all fuel injection vehicle. Here in simulation, EV model with BLDC and battery is being used. EV will fulfill all daily commute needs. Battery is Latium-ion based and depth of discharge is100 %.

2.2.7Battery Energy Storage System

BESS plays an important role in increasing efficiency. BESS acts as DRES. Whenever excess energy is needed, BESS fulfill that energy need. These are DC voltage storage system. Lithium-ion battery based on ZARC model is being used. Lithium cells has a voltage from 3 V to4.2 V. Many cells can be added either in series or parallel or both. The BESS has batteries and charging system. The dynamic impedance model (ZARC) of battery cell contains

1. Ohmic resistance.
2. Voltage open circuit
3. Constant phase element

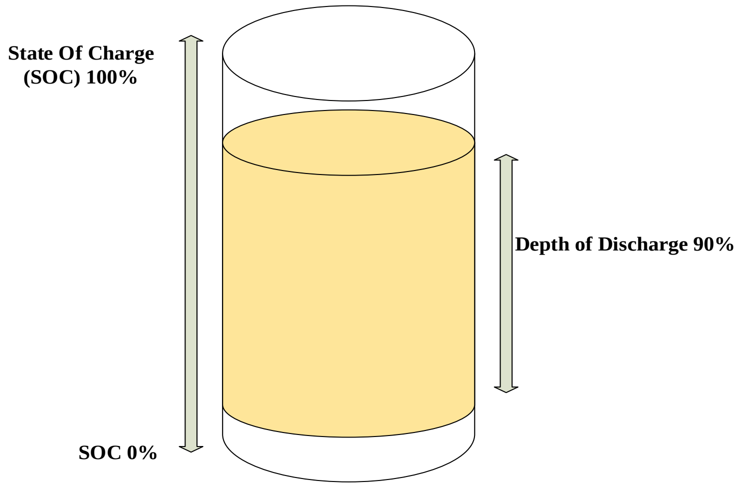

Figure 15 Battery Energy Storage System

These elements varies as per state of charge (SOC) and cell temperature. Figure 15 shows DOD is 90 %(from SOC 5 % to95 %).

2.2.8 Photovoltaic array

The PV converts sun energy to electrical DC energy. These are used as DC voltage sources. When connected to the grid, DC energy needs to be change DC to AC using inverter. Simulation uses Panasonic HIT® N330HIT(Heterojunction Intrinsic thin layer) which has module conversion efficiency 22.5 % [24,25]. The module has inclination angle of 25° (where 90° vertical and 0° horizontal) and orientation angle of 145° (where 0° North, 90° East, 180° South, 270°West).

2.3 Optimal sizing

This forms a multi objective optimization problem. These find set of optimal solutions (Pareto optimal). It uses power and energy balance equations to find optimal size to optimize the self-consumption (SC) of PV energy and optimize the cost of energy generation.

- **PV sizing**

$$PV_{geni} = \eta_{\text{converter}} * \eta_{\text{modules}} * A_{area} * R_{\text{radiation}} \qquad \forall i \qquad (8)$$

- **Sizing of battery**

$$Energy_{BESS_{t+1i}} = Energy_{BESS_{ti}} + BESS_{charge} + BESS_{discharge} \quad \forall i \tag{9}$$

$$Energy_{BESS_{min}} = (1 - \mathrm{DOD}) * Energy_{BESS_{max}} \quad \forall i \tag{10}$$

$$BESS_{charge} = Energy * \eta_{\mathrm{charge}} \tag{11}$$

$$BESS_{discharge} = Energy * \eta_{\mathrm{discharge}} \tag{12}$$

- **Converter sizing**

$$\mathrm{Converter_{size}} = \frac{\mathrm{Load}}{\eta_{\mathrm{converter}}} \tag{13}$$

To determine the optimal sizing following algorithm is being used. This is multi objective problem [26]

$$Min\, Z_1 = \frac{1}{self\ consumption} = \frac{\sum_{i=0}^{24} PV_{geni}}{\sum_{i=0}^{24} min(\, PV_{geni}, P_{totalConsumptioni})} \tag{14}$$

$$Min\, Z_2 = \mathrm{cost_{of_{PV}}} * \mathrm{CRF} * PV_{gen\_p} + \mathrm{cost_{of_{BESS}}} * \mathrm{CRF} * Energy_{BESS} + \mathrm{Cost_{of_{grid}}} * Energy_{grid} \tag{15}$$

•

$$P_{totalConsumptioni} = P_{houseload} + P_{heatload} + P_{ElectricVechicle} \quad \forall i \tag{16}$$

$$P_{BESSi} = P_{totalConsumptioni} - PV_{geni} - P_{Gridi} \quad \forall i \tag{17}$$

Where

$$\text{If } P_{BESSi} \geq 0; \text{Battery discharged}$$

$$\text{If } P_{BESSi} \leq 0; \text{Battery charged}$$

$$\mathrm{MinR_{radiation}} \leq R_{\mathrm{radiation}} \leq \mathrm{MaxR_{radiation}} \tag{18}$$

$$\mathrm{MinA_{area}} \leq A_{area} \leq \mathrm{MaxA_{area}} \tag{19}$$

$$\mathrm{MinP_{totalConsumptioni}} \leq P_{totalConsumptioni} \leq \mathrm{MaxP_{totalConsumptioni}} \tag{20}$$

$$\mathrm{SOC_{min}} \leq SOC_i \leq \mathrm{SOC_{max}} \quad \forall i \tag{21}$$

$$\mathrm{Energy_{BESS_{min}}} \leq Energy_{BESS_i} \leq \mathrm{Energy_{BESS_{max}}} \quad \forall i \tag{22}$$

$$\mathrm{CRF} = \mathrm{Cost_{of_{Recovery\_factor}}} = \frac{(\mathrm{r} * (1 + \mathrm{r})^{\mathrm{n}})}{(1 + \mathrm{r})^{\mathrm{n}} - 1} \tag{23}$$

Where

r Rate of interest

n Number of years.

The description of multi-objective problem are as follows: Equation 14 objective function optimizes SC, equation 15 objective function optimizes energy cost. Then it has constraints based on energy and power balance. Equation 8 shows the PV generation, equation 16 shows power needed for house, equation 17 shows power charges and discharges of BESS. Equation 18 shows constraints on sun radiation, equation 19 is constraints on area available for PV arrays, equation 20 shows constraints on energy consumption, equation 21 shows the constraints battery energy.

The optimal solutions give parameters for EMS. Table 3 shows the engineering parameters

Table 3 Optimized parameters for simulation for an average winter day.

Engineering parameters	**AC house**	**DC house**
PV array [24,25]	7 kW_p	5 kW_p
Battery capacity	8 kWh	6 kWh
Appliances load	9.7 kWh	8 kWh
Electric vehicle	11 kWh	10 kWh
Heat pump	12 kWh	8.8 kWh
Heat load	40 kWh	40 kWh

2.4 Energy Management System.

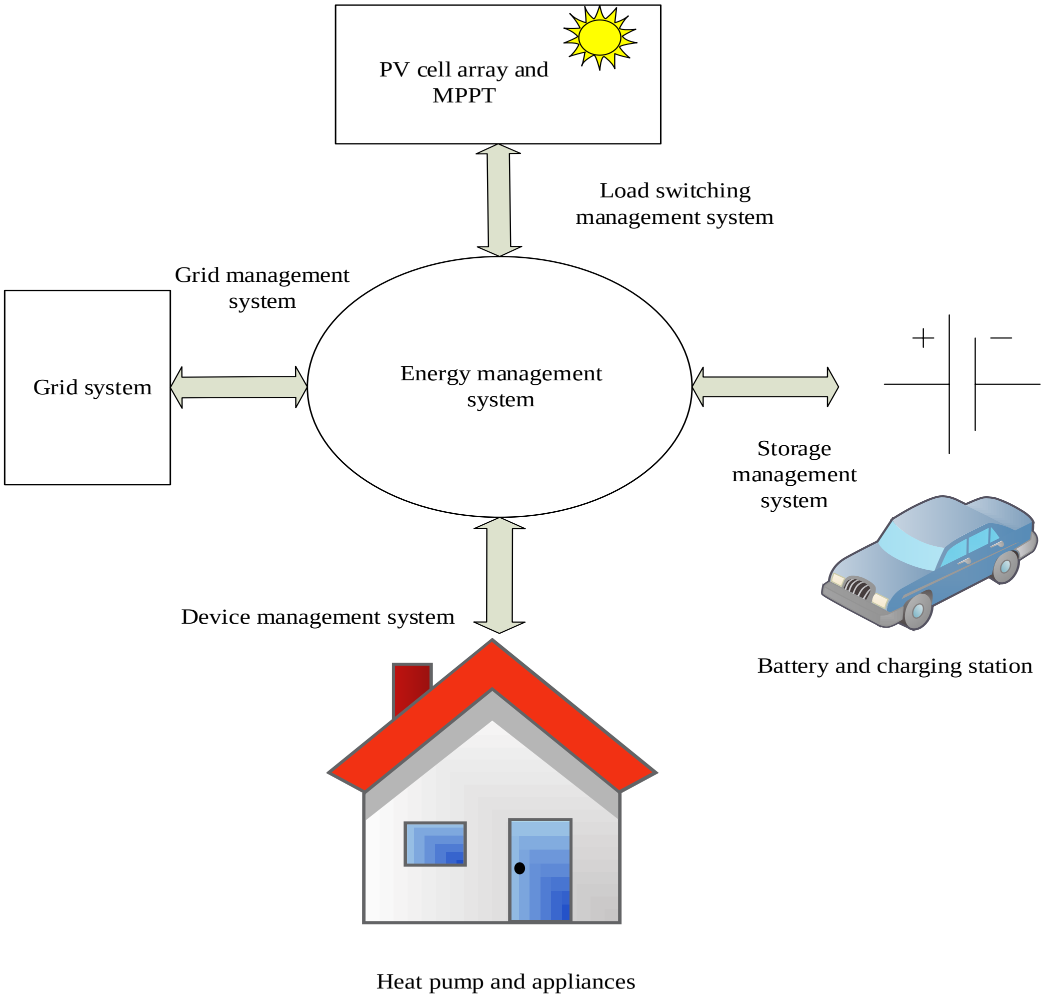

Figure 16 Energy management system of Housing system.

When solar panels are installed in the house, it is important to optimize PV self-consumption. (SC). Self-consumption is defined as ratio of amount of solar energy consumed to total solar energy produced. SC is a way to optimize the use of PV generated energy. Main obstacle for SC is the energy use in house does not match with the time when it is consumed. Due to this, power from grid is being used. Natural self-consumption is 40 % [27] but EMS can optimize the SC. Figure 16 shows the functioning of EMS
Optimized SC stores the surplus energy from the PV generated system to BESS. This not only reduces the energy purchase from grid but also the grid feed in energy (which in turn increases grid stability) [28,29]. The EMS using DR algorithm optimizes the SC to 90 %,

by doing so it reduces the electricity consumption from grid, reduces the electricity bill and contributes to the grid stability [28–31]. When BESS is used, then SC is all of PV generated energy which is either consumed directly or using BESS

.

$$Self\ consumption\ (\%) = \frac{self\ solar\ energy\ consumed\ (\text{kWh})}{Total\ solar\ energy\ produced\ (\text{kWh})} \tag{24}$$

EMS uses following systems (see Figure 16):

1. BESS: The battery system uses different demand response algorithms to increase SC and optimize the energy consumption. DR based on price function and fast charging algorithm is being implemented.
2. Device management system: This optimizes consumption of energy either from PV or from grid.
3. Grid management system: This optimizes the energy consumed from grid.
4. Load switching management system: This optimizes the energy generated from PV.

2.4.1 Algorithm of Heat pump controller.

Heat pump controller consumes the solar power based on appliances priority. Here the priority is as follows

1. Heat pump
2. Home appliances
3. Electric vehicle

When

$$P_{heatload} = PV_{geni} + P_{BESSi}\ or\ P_{grid} \qquad \forall i \tag{25}$$

Equation 25 shows the power consumption of heat pump. Figure 17 shows that heat pump consumes solar energy and pump the heated fluid in the heat storage. The heat storage system pumps to maintain the zonal internal temperature of the house. It maintains the zonal temperature within a temperature band of 19℃ to21℃. Heat pump uses temperature inertia and solar energy to maintain the temperature. The equation shows heat pump uses either solar energy or BESS.

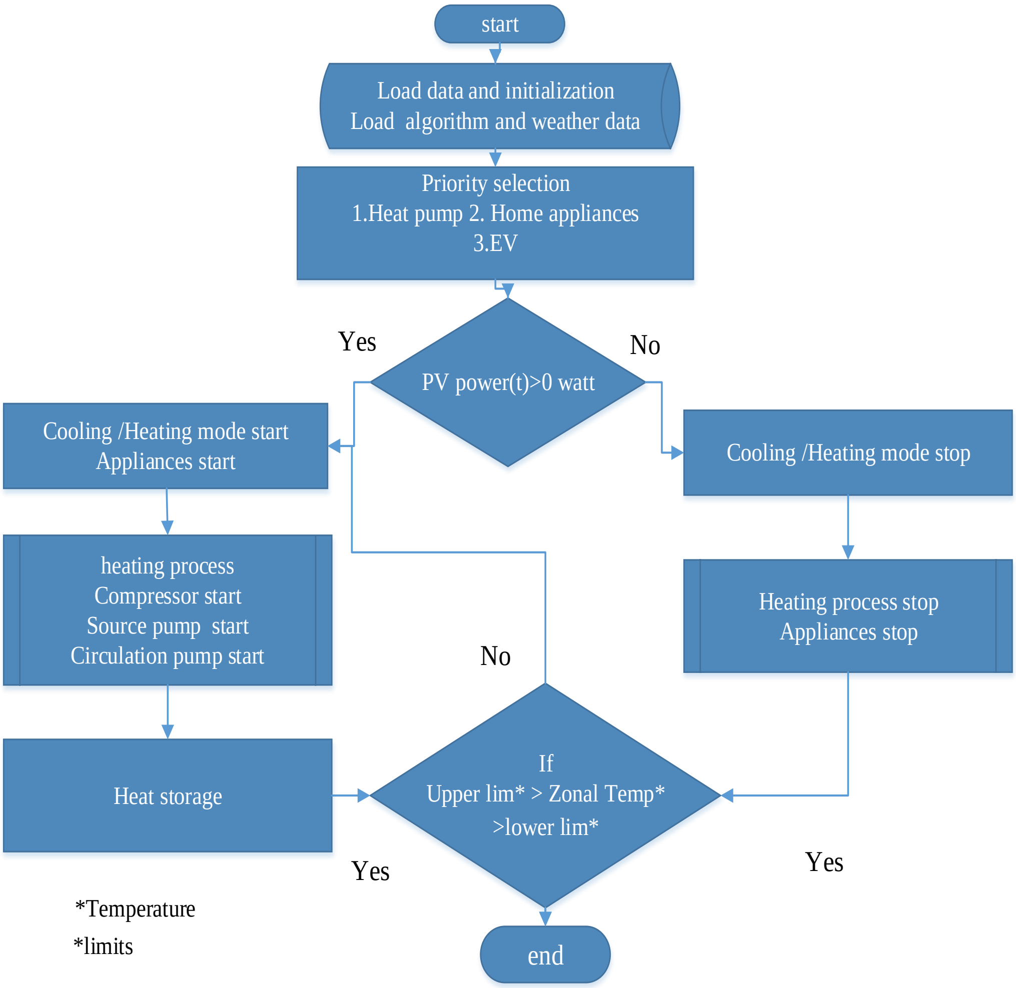

Figure 17 Algorithm of Heat pump controller.

Here for one day simulation Dresden, February average data were taken. Here total PV energy generation is 40 kWh and heat pump load for AC house is 12 kWh and for DC house is 8.9 kWh. Figure 18 shows the consumption behavior of heat pump to optimize self-consumption of solar energy.

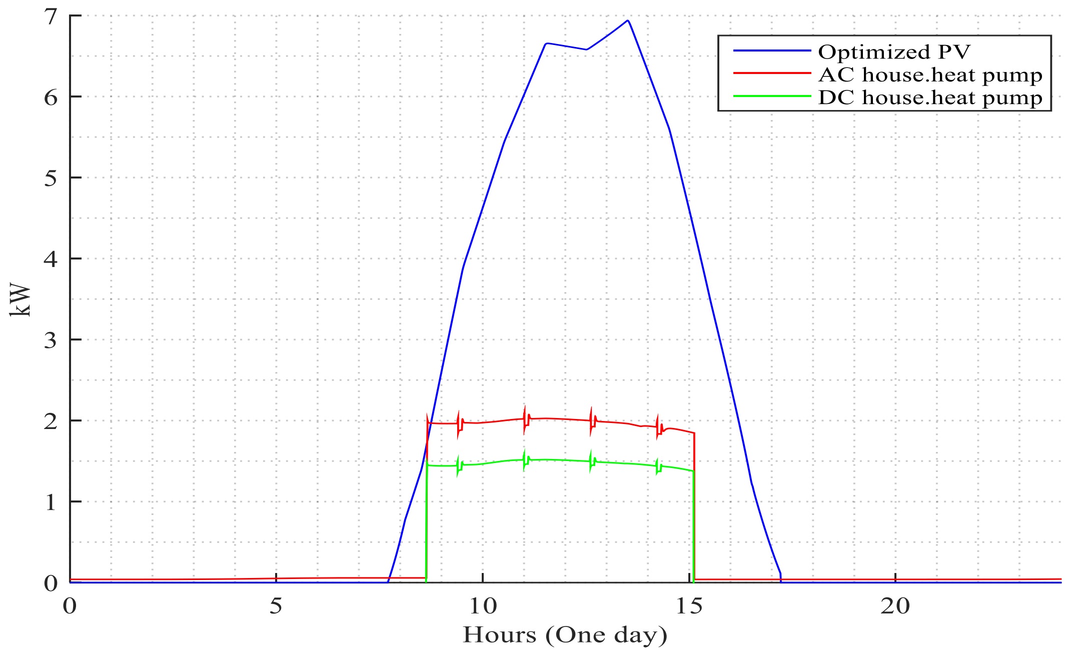

Figure 18 Heat pump consumption pattern with EMS.

Figure 19 shows the zonal temperature inside the house remain at 20°C temperature reference. The heat pump uses internal thermal inertia of house and temperature remains within 1°C temperature band.

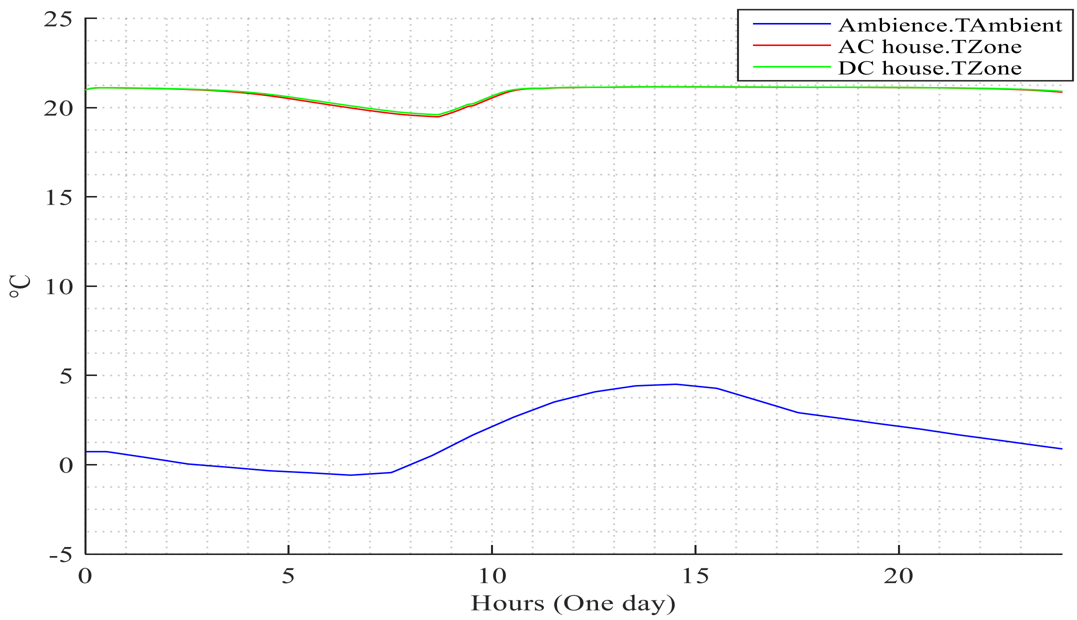

Figure 19 Zonal temperature inside the house.

Figure 20 shows the flow of heat pump. DC house and AC house shows a similar flow pattern. That means the pump loads remain same in spite of using BLDC with VFD.

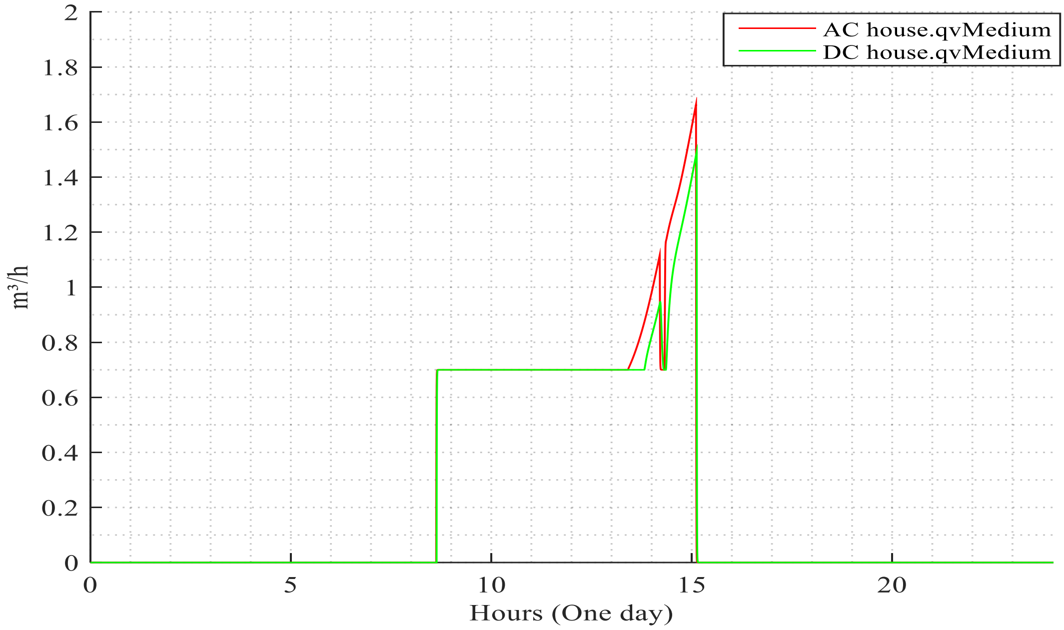

Figure 20 Flow of heat pump.

Figure 21 shows the heat load of 40 kWh is required for the house. The coefficient of performance (COP) for AC heat pump is 3 and for DC house is 4. So there is 35 % increase in COP for DC house.

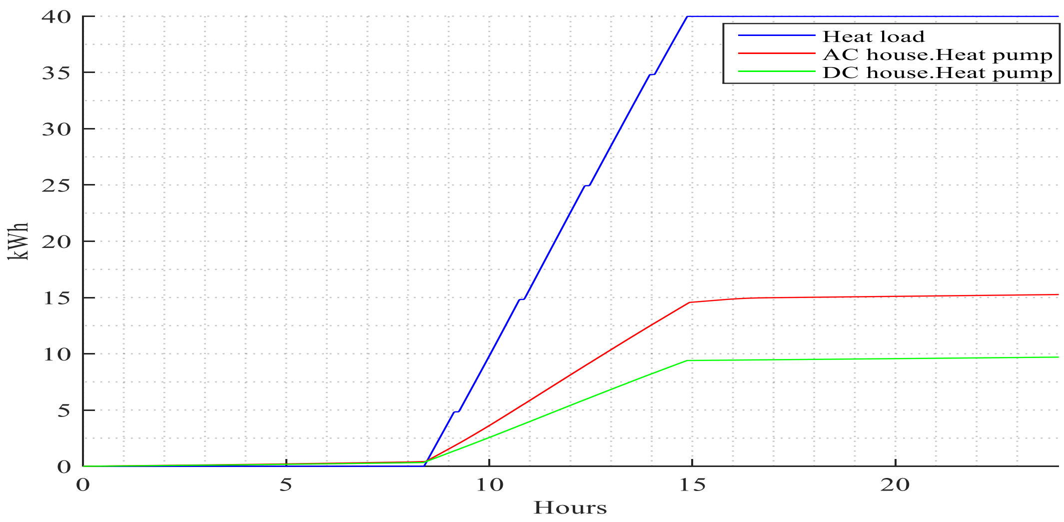

Figure 21 Heat load of house and energy consumption of AC and DC heat pump.

As heat storage is available. So heat pump can store heat energy during day time when solar energy is available and use this during night time. Figure 22 shows charging and discharging pattern of heat storage is shown.

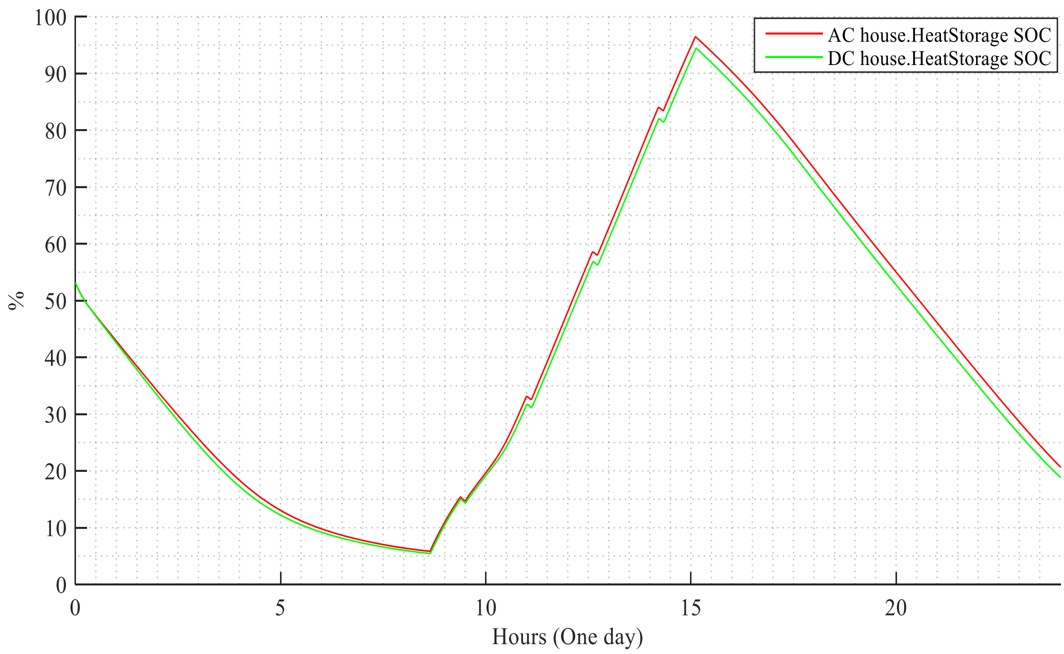

Figure 22 SOC of heat storage system.

2.4.2Algorithm of home appliances controller.

In home appliances controller (see Figure 23), consumes power for home appliances in priority based. Here in the EMS, the priority of home appliances is second. Excess solar power after heat pump consumption is consumed by home appliances. During day time, it consume solar energy and excess energy requirement is compensated by BESS. During night time or when solar power is not available, energy requirement is fulfilled by BESS. Figure 23 shows the load flow curve of home appliances. Equation 26 shows the power consumption of home appliances.

$$P_{totalConsumptioni} = PV_{geni} - P_{heatload} + P_{BESSi} \quad \forall i \tag{26}$$

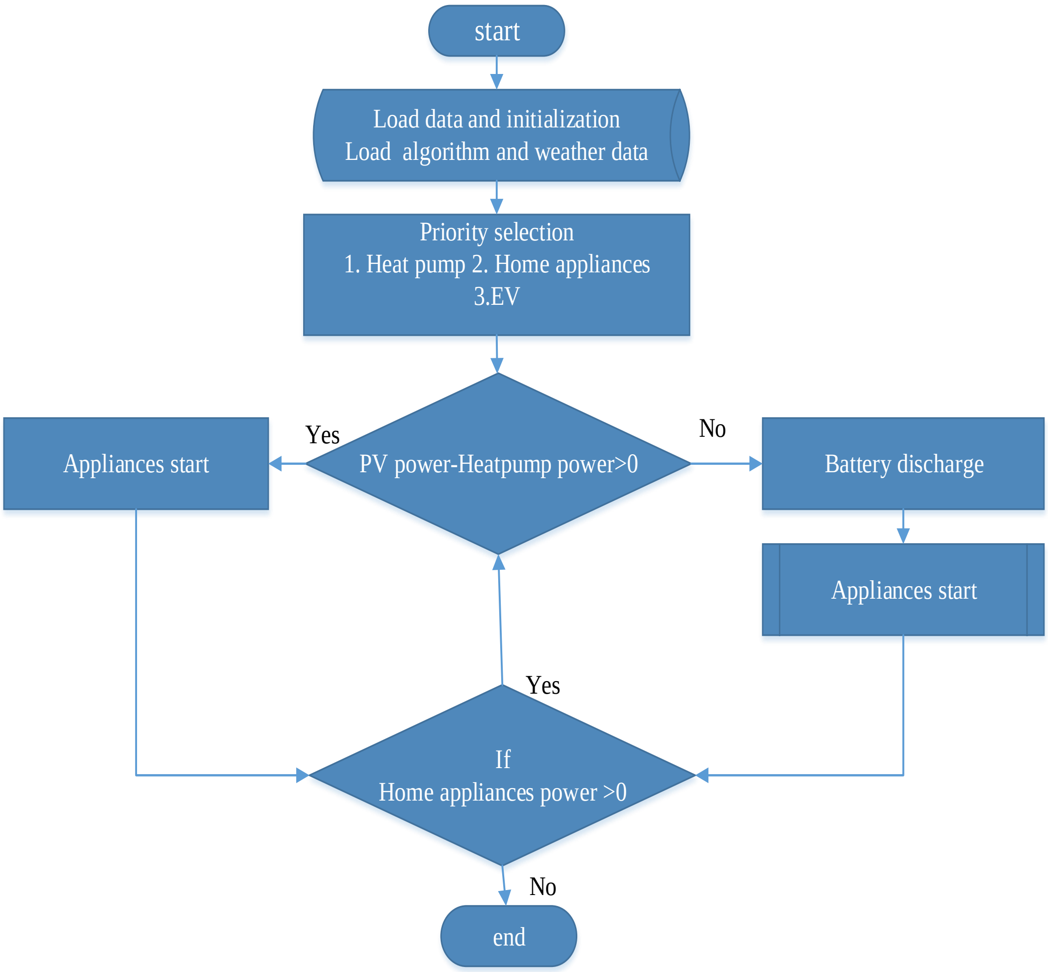

Figure 23 Algorithm of home appliances controller.

Figure 24 shows the load flow of home appliances during 24 hours. The BESS compensates the energy requirement when solar energy is not available.

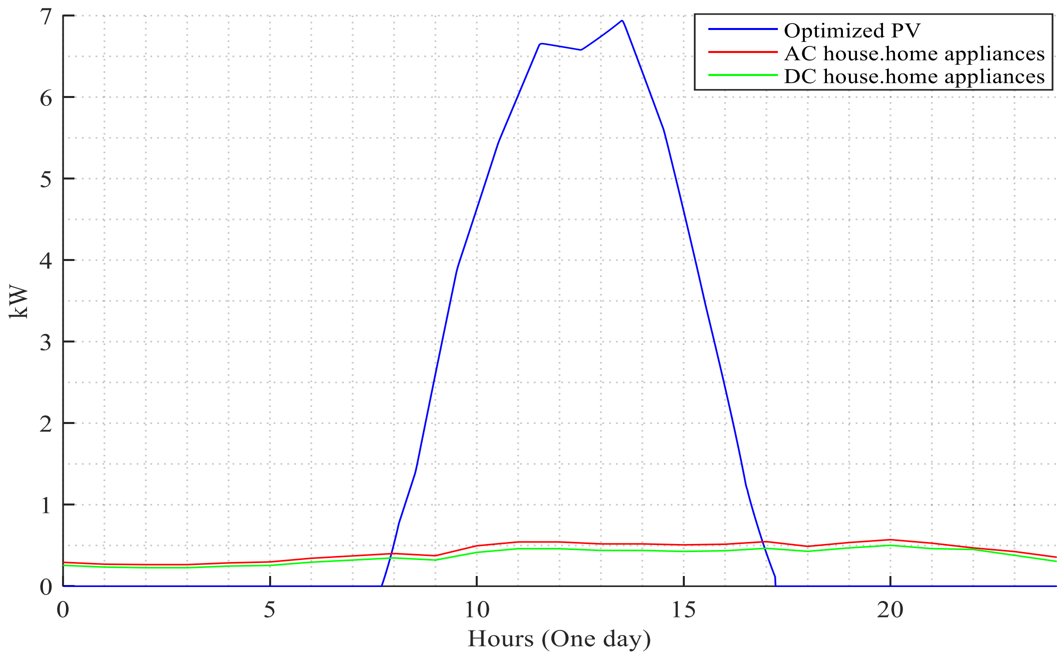

Figure 24 Consumption pattern of house appliances with EMS.

2.4.3Algorithm of Electric vehicle charger.

Electric vehicle charger controller has a priority of third that means that it consumes power after fulfillment of heat pump and home appliances power requirement. See Figure 25 shows that the algorithm works in SOC range from 100 % to 0 % (DOD100 %). Whenever vehicle moves then battery discharges. Equation 27 shows the power consumption of electric vehicle.

$$P_{ElectricVechicle} = PV_{geni} - P_{totalConsumptioni} - P_{heatload} \quad \forall i \qquad (27)$$

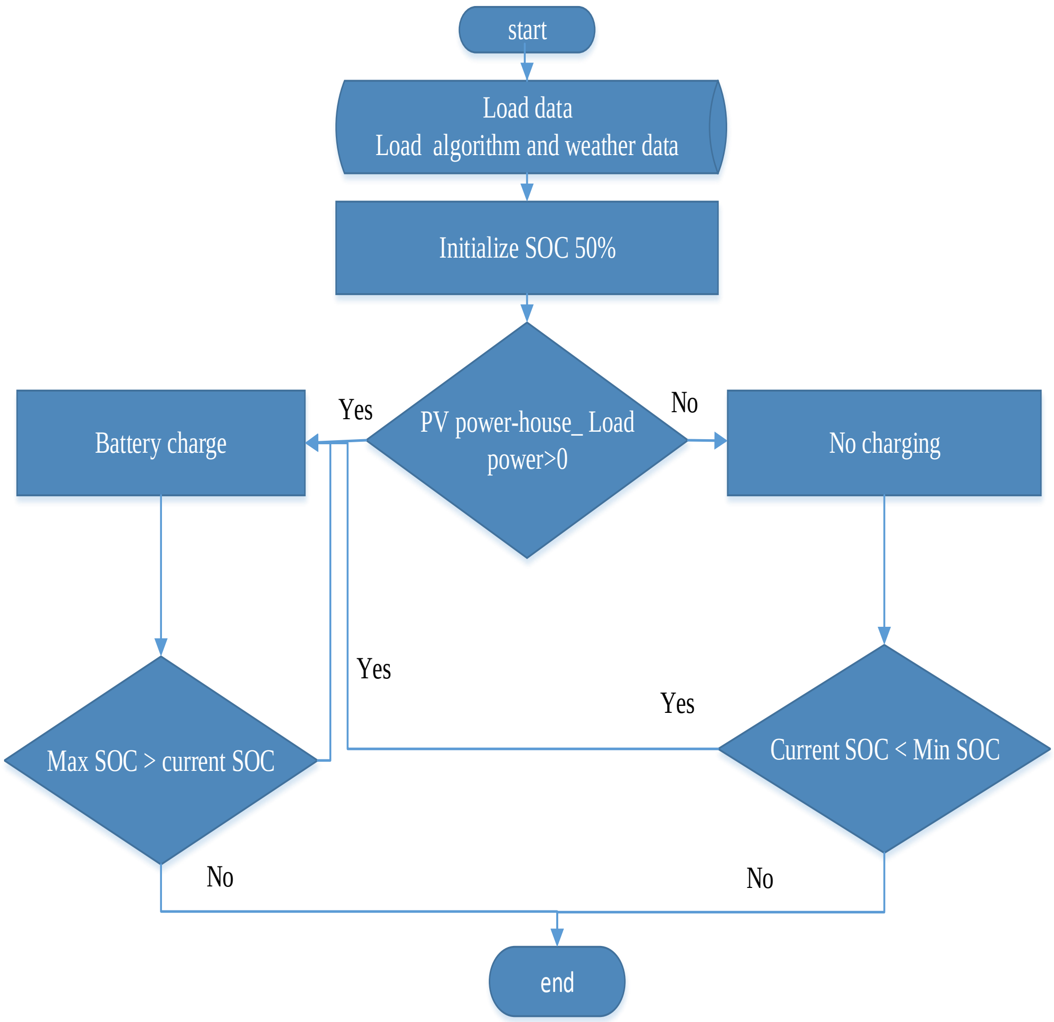

Figure 25 Algorithm of Electric vehicle charger.

Figure 26 shows the power consumption behavior (vehicle charging station) of electric vehicle. The rate of charge is 1 kW/hr. Electric vehicle (Nissan leaf) has 22 kWh load. Initial condition 50 % SOC. EV charges 8.5 kWh in DC house wheareas 7.8 kWh in AC house. Figure 27 shows that the vehicle in DC house charges to 91 % but vehicle in AC house charges till 88 %. The converter losses are6 %. So EV needs 0.28 kWh for 1 % SOC in AC house and 0.22 kWh for 1 % SOC in DC ho use. 20 % Less energy required in DC house.

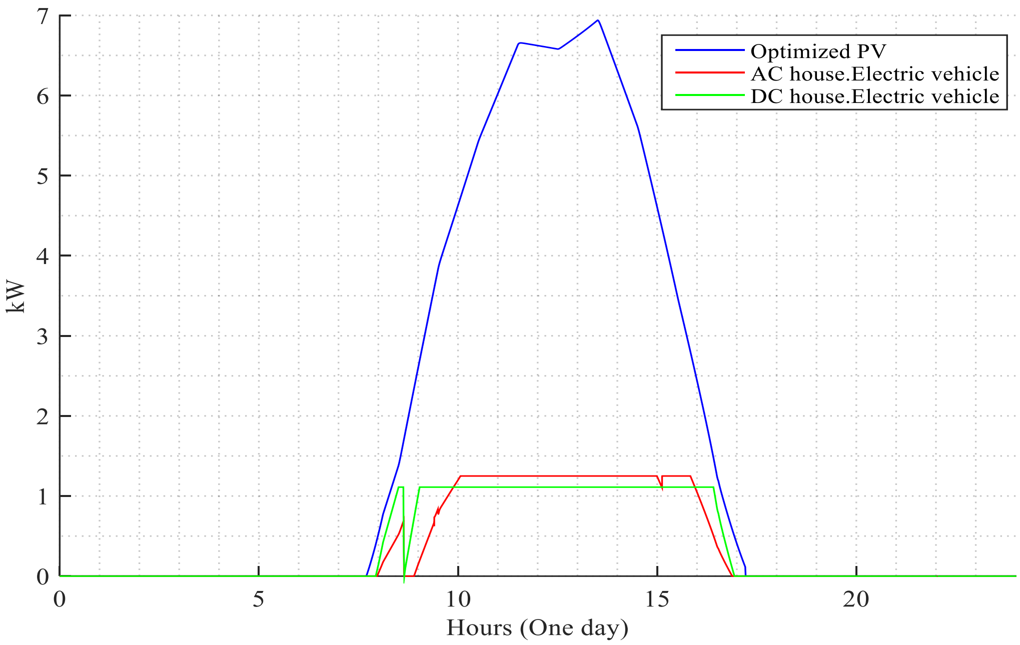

Figure 26 Consumption pattern of electric vehicle.

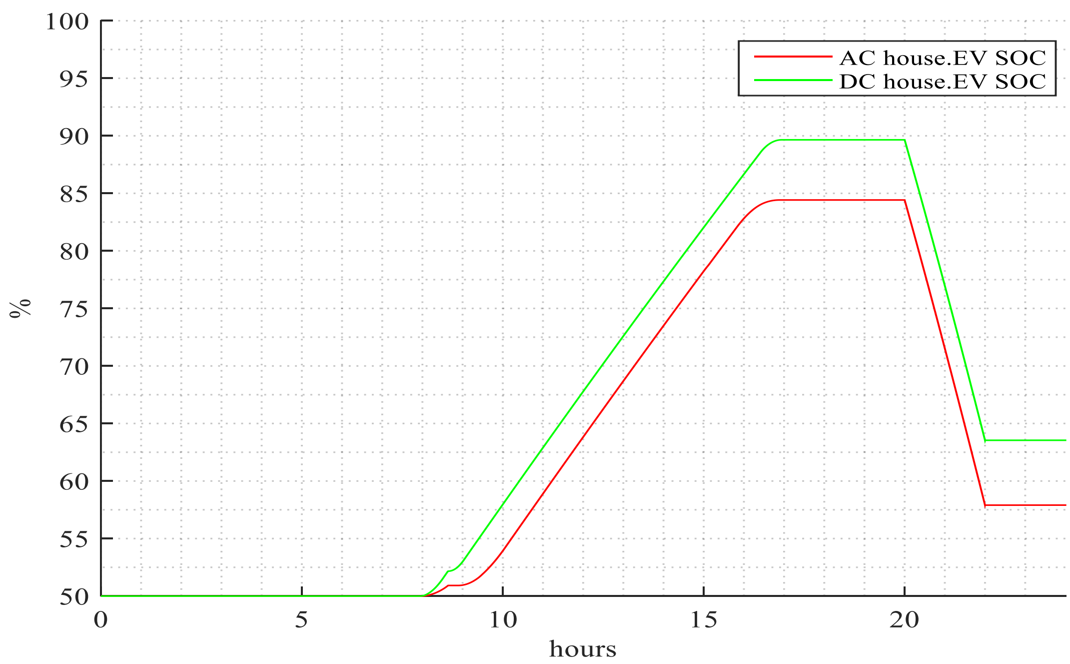

Figure 27 SOC of electric vehicle.

Figure 29 and figure 28 shows that EV requires 5.4 kWh for travelling 60 km. It means that vehicle travels in 0.126 kWh/km in AC house but in DC house, EV requires 0.1 kWh/km. 20 % less energy required in DC house.

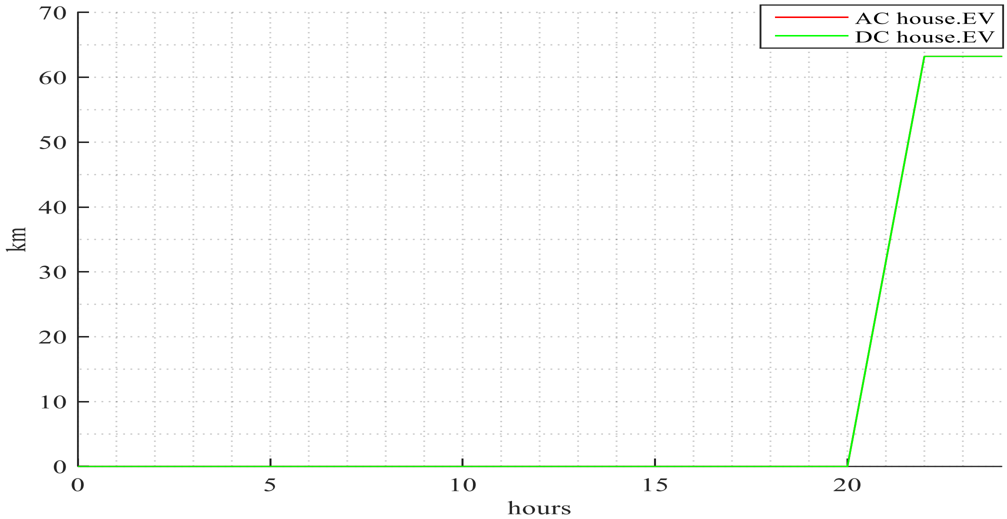

Figure 28 Distance travelled by EV.

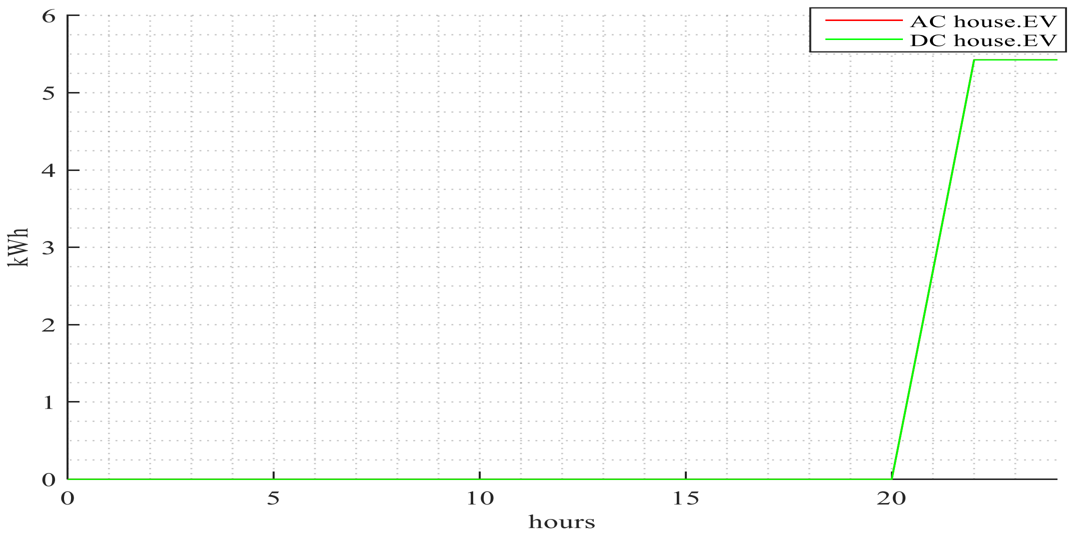

Figure 29 Energy required for travelling.

2.4.4Algorithm of BESS.

Figure 30 shows that the BESS algorithm works in range SOC from 95 % to 5 % (DOD 90 %). Whenever current SOC is less or equal to 5 %, it charges either from grid or PV array. It stops charging whenever the current SOC is more or equal to 95 %. BESS works to fulfill the energy shortage.

$$P_{BESSi} = P_{totalConsumptioni} - PV_{geni} - P_{Gridi} \quad \forall i \tag{17}$$

Where

$$\text{If } P_{BESSi} \geq 0; \text{Battery discharged}$$

$$\text{If } P_{BESSi} \leq 0; \text{Battery charged}$$

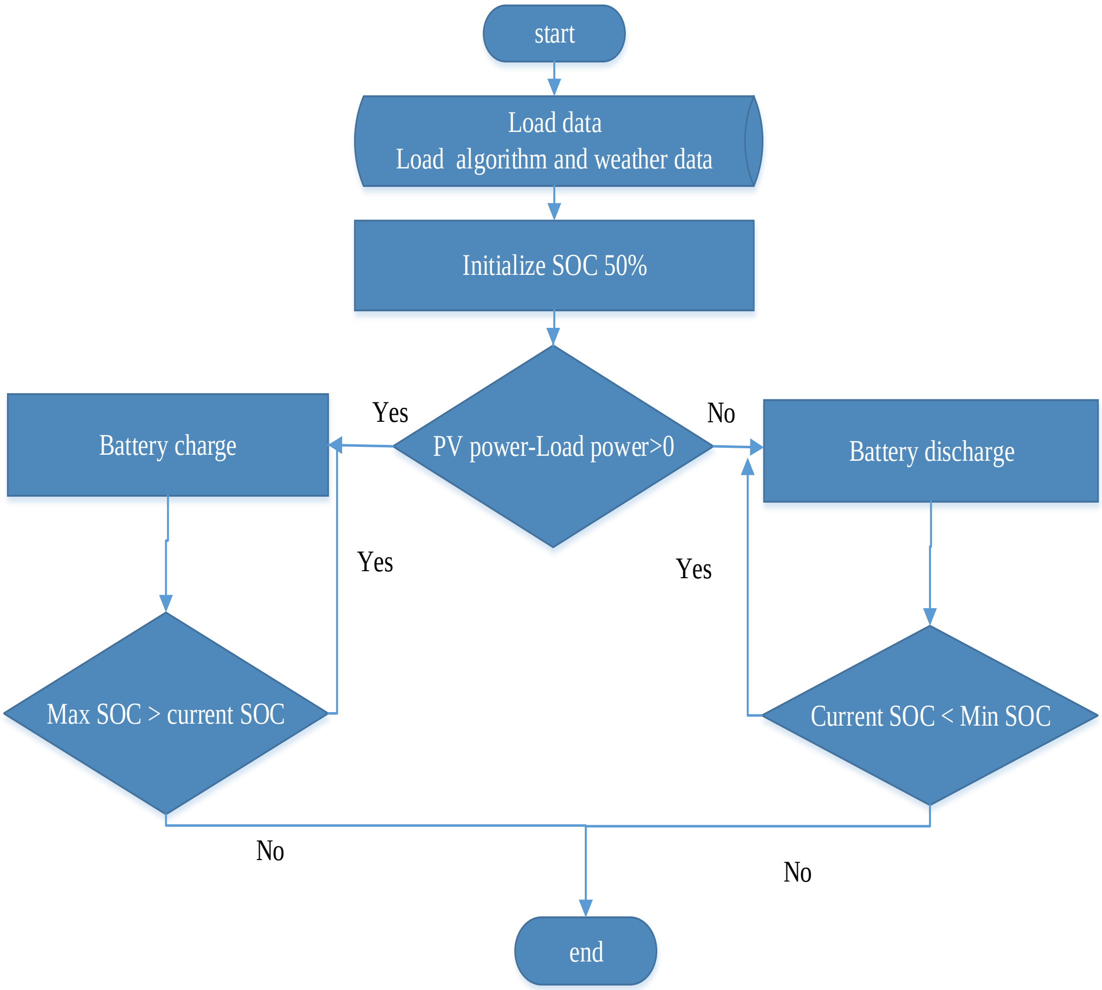

Figure 30 Algorithm of BESS.

Figure 31 shows the charge and discharge cycle of BESS. The figure 31 shows that the battery discharges whenever solar power is not available but charges whenever excess of solar power. As DC house consumption is lesser than AC house, so PV arrays and battery sizing is oversized in DC house. As DC house, PV array is oversized, the battery get charged early as compared to AC house. Due to this, optimum sizing of PV arrays and battery is required.

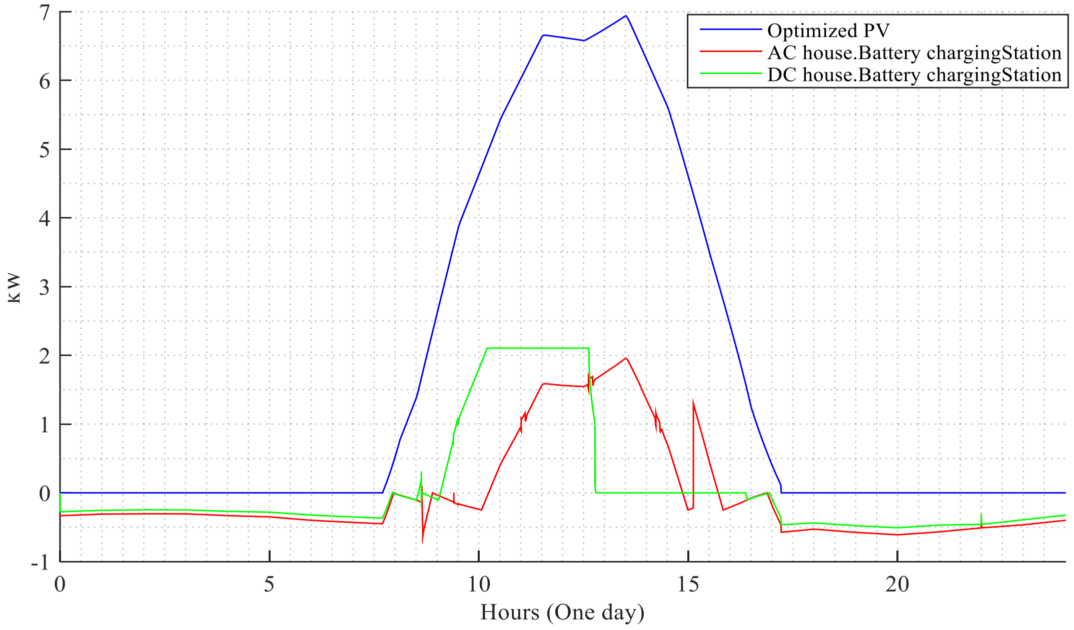

Figure 31 Charging and discharging cycle of BESS.

Figure 32 shows the behavior of SOC in DC house and AC house. The battery charges when solar energy is available and discharges when sufficient solar energy is not available.

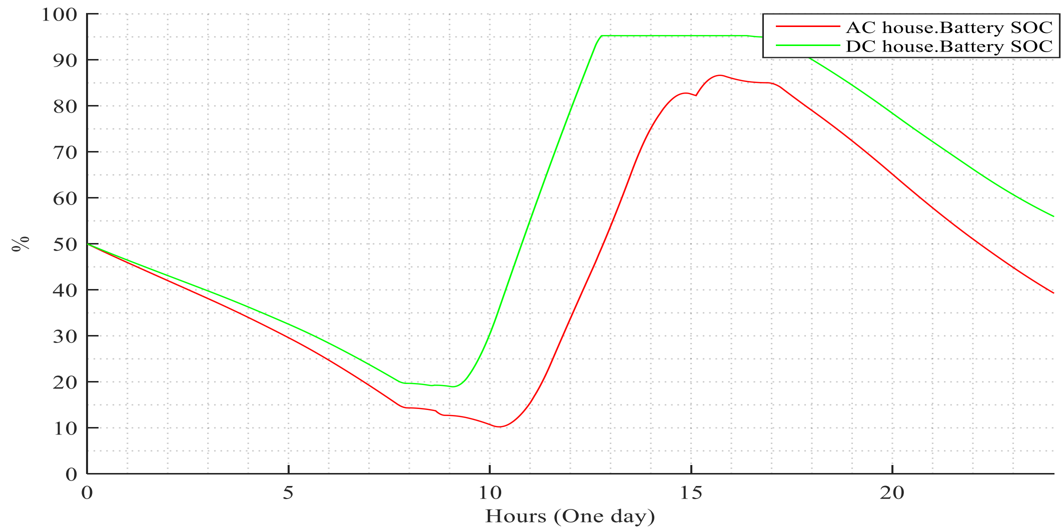

Figure 32 SOC of BESS.

2.5 Energy management system with BESS.

EMS algorithms are being performed in AC and DC house for one day. The EMS optimizes the energy consumption of house. Effect of EMS with and without BESS are being shown in next sections.

2.5.1Total consumption of house without BESS

The main objective is to increase SC and reduce the cost price of electricity. Figure 33 shows that the consumption behavior of DC and AC house without BESS. EMS optimally distributes power throughout the day.

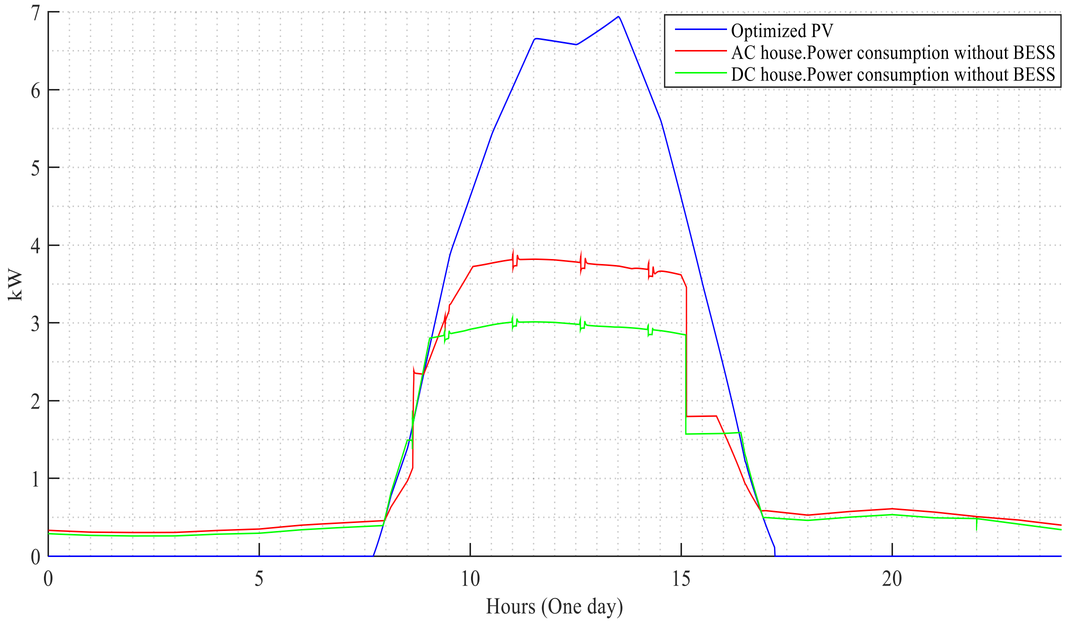

Figure 33 Consumption pattern of house loads.

Figure 34 shows the power difference between energy requirement of the house without BESS and PV generation. As energy requirement of DC house is lesser. So power difference is more in DC house for same PV array size. This shows the oversize of PV in DC house [32,33]. Here grid feed in energy is large which can cause problem in grid stability.

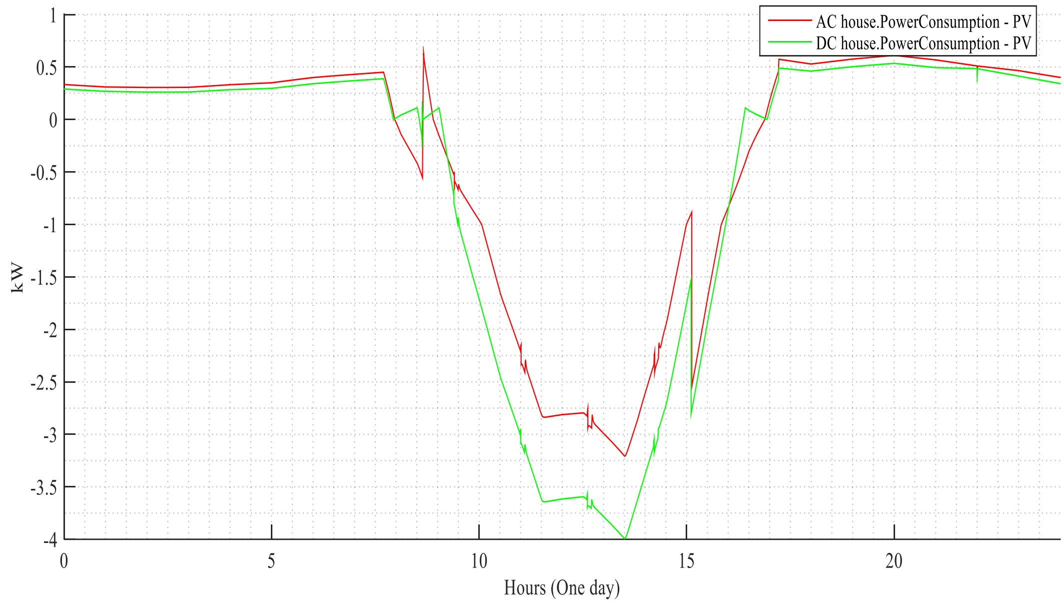

Figure 34 Power difference between house consumption and PV generation without BESS

2.5.2Total consumption of house with BESS

Figure 35 shows the optimal consumption of PV energy using BESS of AC house and DC house. BESS optimizes the SC to 90 %.

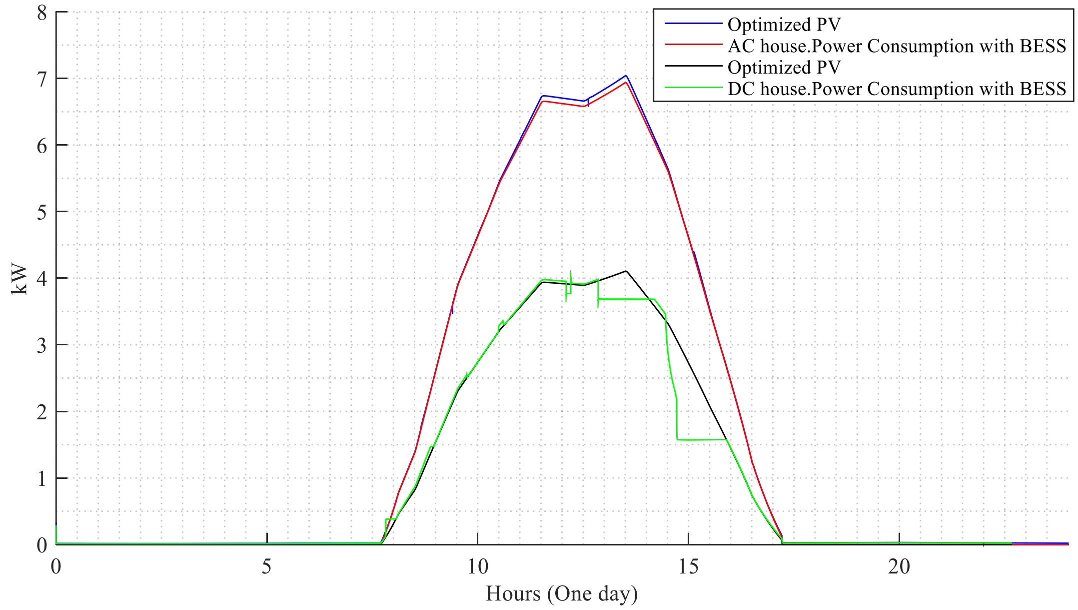

Figure 35 Self consumption of AC house and DC house.

Here optimal sizing of PV system and battery system of AC and DC house is being performed. Figure 35 shows that optimal sizing of DC house, decreases the size of battery by 25 %, PV array by 25 % and converter size by 25 %. In optimized AC house, PV array size of 7 kW_p, battery size is 8 kWh whereas in optimized DC house, PV array size of 5 kW_p, battery size is 6 kWh which increases the optimized SC is 90 %. Due to this the initial investment in DC house is being reduced by 30 % as compare to AC house.
Figure 36 shows that optimized power difference between house energy requirement using BESS and optimized PV arrays generation. The optimized difference says the almost all energy requirement is being fulfilled. This reduces the feed in energy to the grid and contribute in grid stability [34].

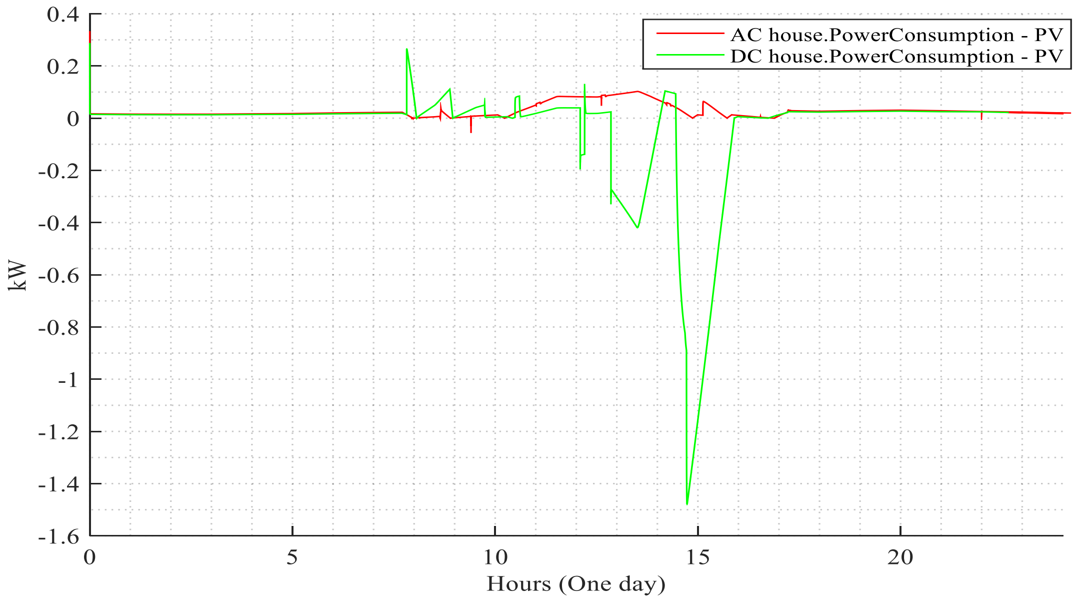

Figure 36 Optimized power difference between house requirement and optimized PV arrays generation.

3 Case study

3.1 One year simulation

3.1.1 Problem formulation

Here simulation on both houses is being run for one year on Dresden, Germany weather data. Figure 37 shows the variation of solar radiation ($\frac{W}{m^2}$) during the year. The solar insolation varies from 4.8 kWh/m^2 /day in July to 0.8 kWh/m^2 /day in January [35]. This has two major problem in designing:

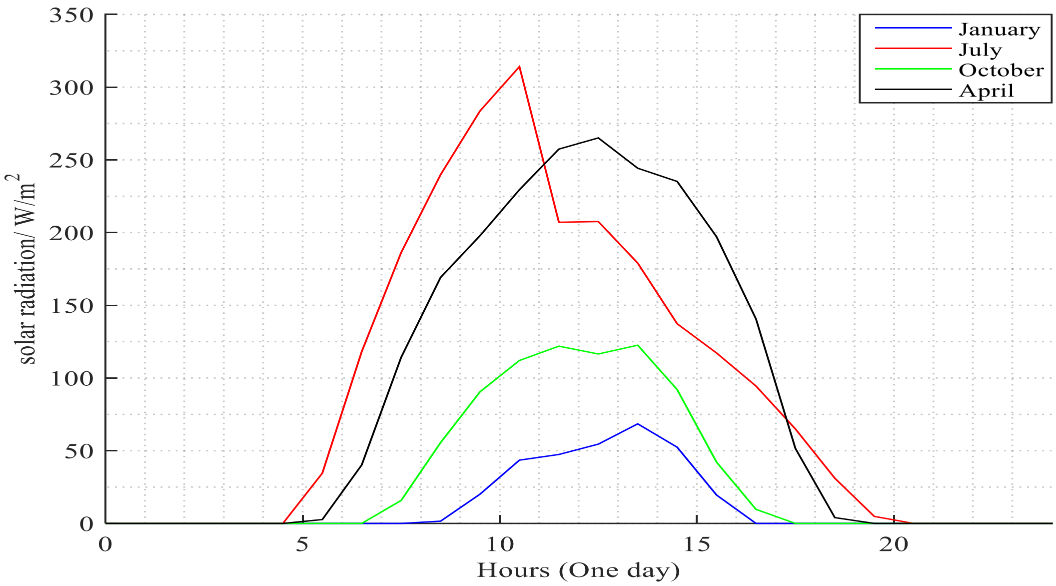

Figure 37 The solar radiation density for seasons.

1. Due to this variation in solar energy, the optimize sizing for a year is difficult. Figure 38 and Figure 39 shows the house consumption: during winter, average energy need per day is 34 kWh in AC house and 27 kWh in DC house. During summer, average energy need per day is 22 kWh in AC house and 18 kWh in DC house. When a house is being designed for winter day or overcast weather then it becomes oversized in summer or sunny day. Figure 38 and Figure 39 shows that when sizing is being determined based on winter days, then it becomes oversize during summer. This requires optimal sizing determination.
2. As number of PV penetration is increasing on the grid. So the grid stability is facing problem related to high penetration of PV generation [29]. Here annual natural SC is 40 % in AC house and 35 % in DC house. This needs an optimized SC. This

research deals with finding an optimal consumer cost benefit ratio between optimal SC and optimal sizing.

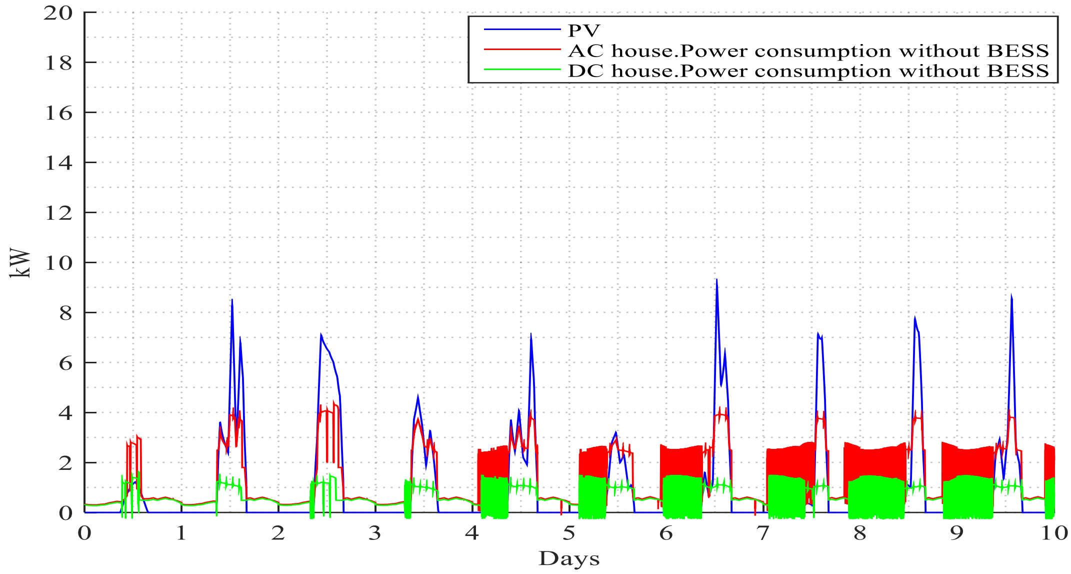

Figure 38 House consumption for winter days.

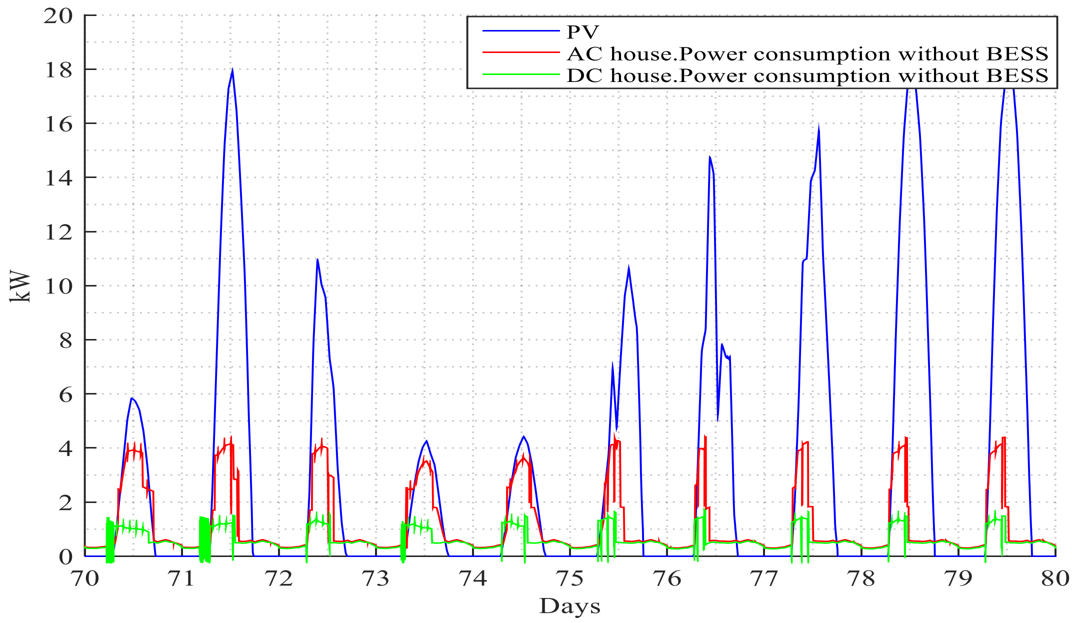

Figure 39 House consumption for summer days.

Three scenarios are being performed to optimize SC and the cost of the energy.

1. **Off grid**: The situation where the house is not connected to the grid. The house generate its own energy using solar array and use BESS to store solar energy. The BESS charges during summer when excess solar power is available and discharges during winter when less solar power is available. This depends on 100% PV and 100% BESS. Here house must be designed for worst possible scenario, e.g. for rainy day or overcast weather
2. **Demand response + electricity from utility + thermal and electrical storage system + photovoltaic cells:** In this case, PV array, BESS and heat storage are used as DRES. It consumes electricity from grid whenever grid electricity price is less. A price function based on variable grid tariff is used to consume electricity from the grid. Demand response algorithm based on price function is being used.
3. **Without demand response + electricity from utility + thermal and electrical storage system + photovoltaic cells:** In this case, PV array, BESS and heat storage are used. It consumes electricity from grid whenever needed.

Simulation is being run for above three scenarios. The results are presented for summer and winter days. Simulation start from January to December (Days are counted from January onwards).

3.2 Off grid

The optimal sizing algorithm determines following parameters to reduce the cost and increase the SC. Table 4 shows the optimal sizing parameters. Figure 40 shows annual consumption requirement for the houses.

Table 4 Optimized parameters for off grid house.

Engineering parameters	**AC house**	**DC house**
PV generation size	12 kW_p	9 kW_p
Battery size	850 kWh	650 kWh
converter size	10.8 kW	8.1 kW

The AC house needs 11000 kWh annually where 1. EV consumes 3000 kWh, 2. Home appliances consumes 3560 kWh, 3. Heat pump consumes 4500 kWh. The DC house needs 9250 kWh annually where 1.EV consumes 3000 kWh, 2.Home appliances consumes 2900 kWh, 3.Heat pump consumes 3350 kWh. Here simulation results for both winter and summer seasons are given.

Off grid system is 100 % PV and 100 % BESS. It must be designed for worst day's e.g. cloudy or rainy or overcast weather. It is 100 % reliable where optimized SC is 90 %. Here total PV generation is 13000 kWh

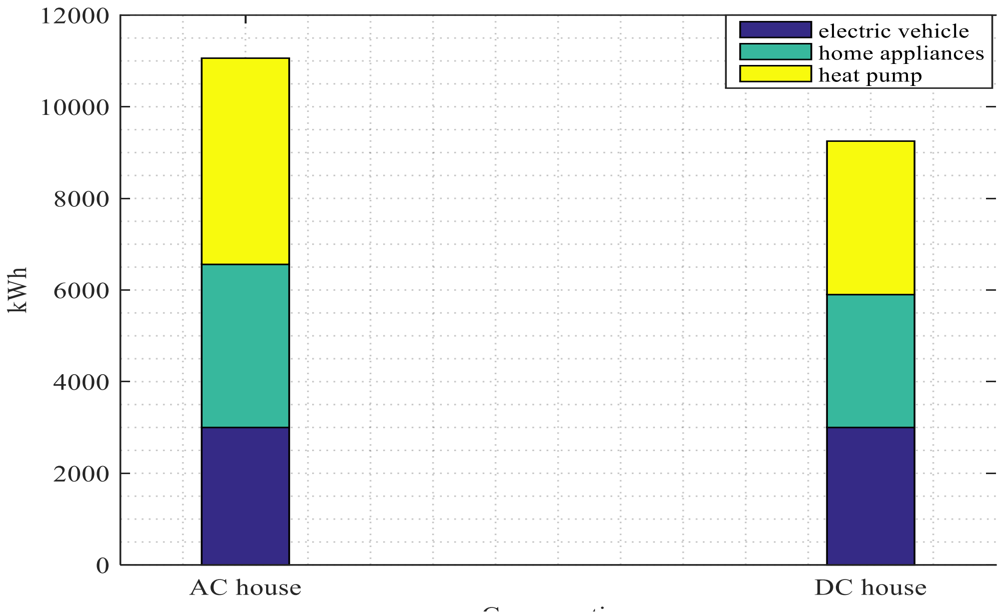

Figure 40 Load distribution of house for one year.

3.2.1 Winter days

Figure 41 shows the consumption requirement for 10 days of January month (1st January to 10th January). It shows the energy demand of house cannot be fulfilled only by PV generation. Figure 42 shows the power difference between house energy and PV generation. The positive power difference means the power deficit and negative power difference means excess power.

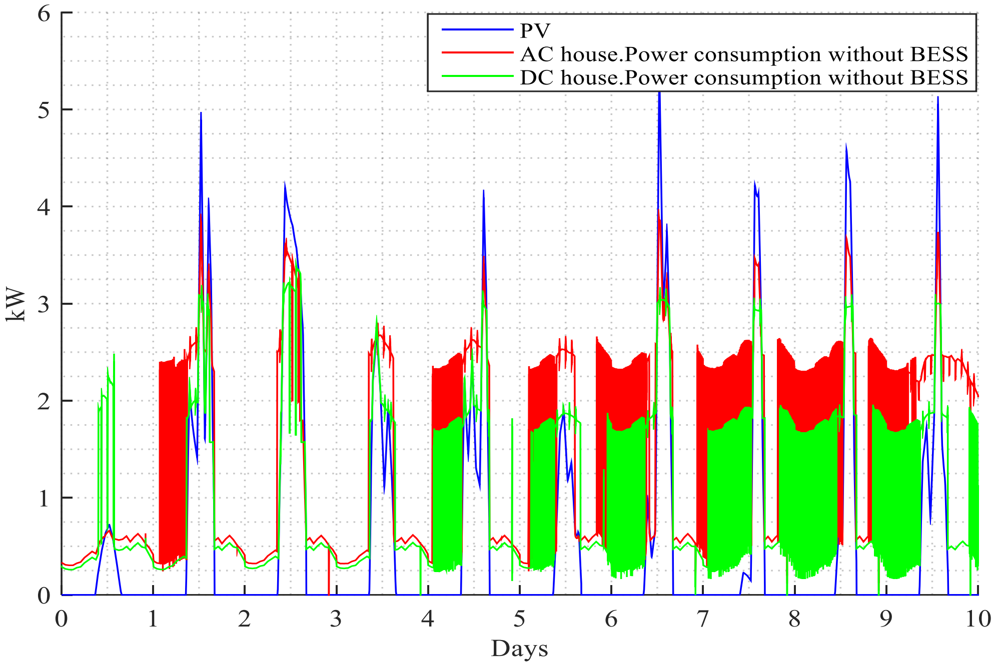

Figure 41 DC and AC house load distribution during winter.

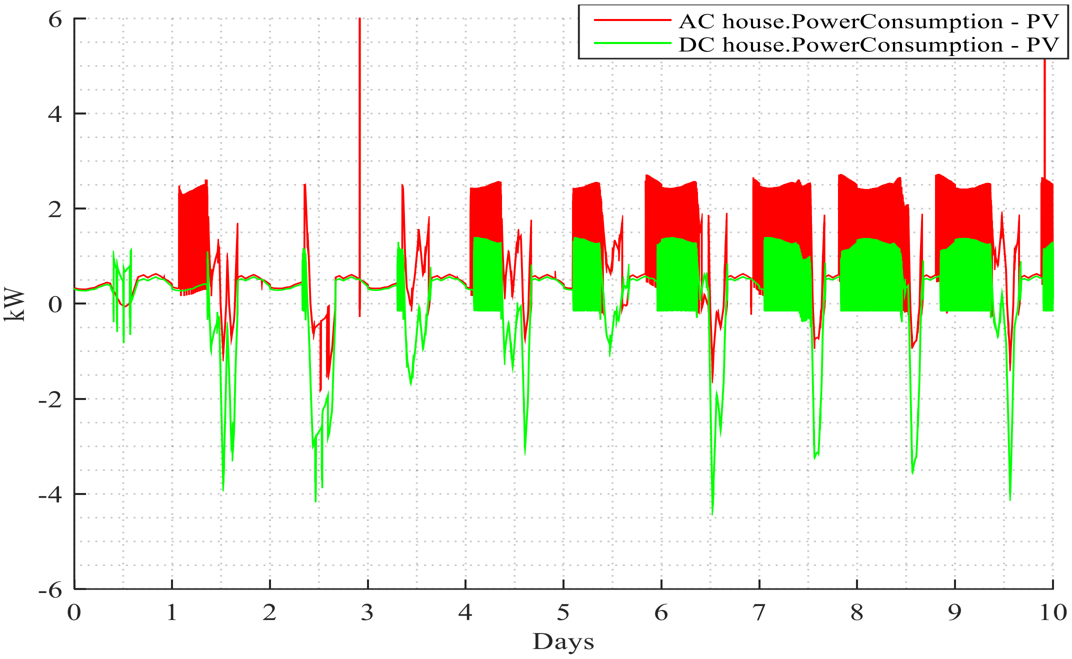

Figure 42 Power difference between house demand without BESS and PV generation.

This positive power difference is being compensated by BESS. Figure 43 shows when BESS is used, then excess energy requirement for the house is being compensated. Here the SC increases to 90 %. Which means it consumes optimal solar energy. Figure 44 shows the power balance after BESS. The power balance indicates that BESS along with PV generation compensates all energy requirement of the house.

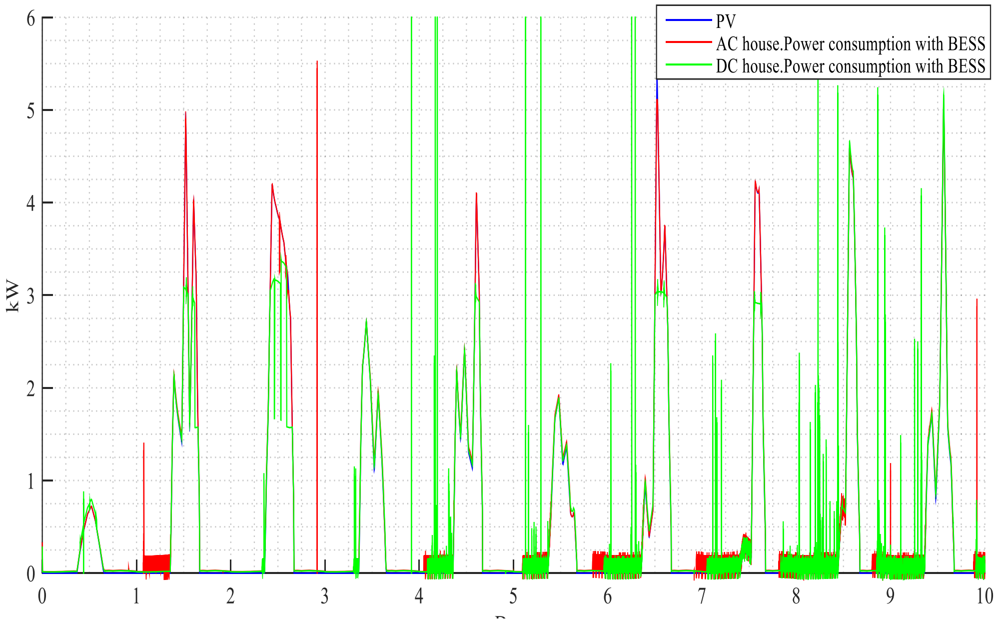

Figure 43 Consumption pattern with BESS in off grid situation during winter.

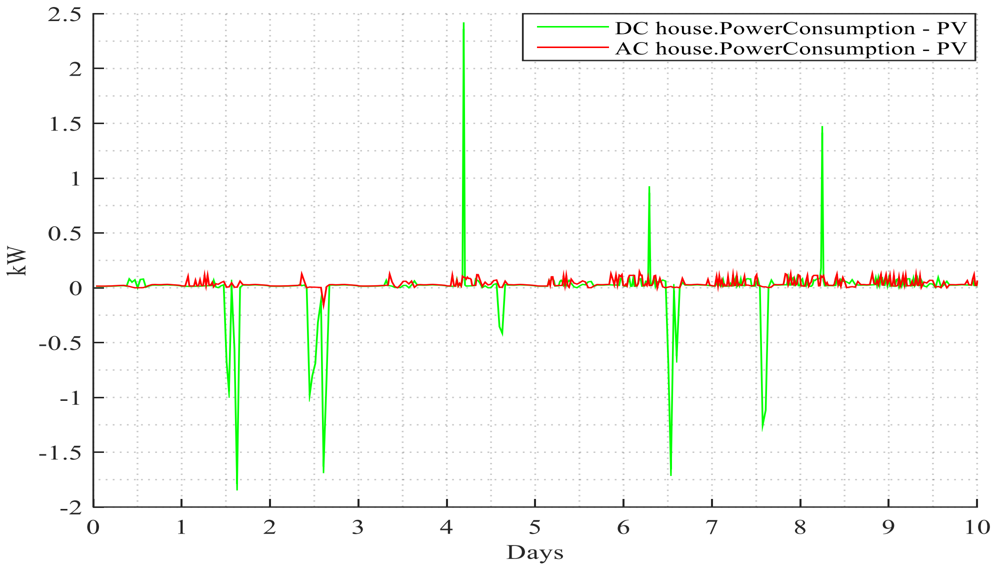

Figure 44 Power difference with BESS in off grid.

Figure 45 shows that zonal temperature inside the house is within 1℃ band width from (19℃ to21℃) where reference temperature is20℃. Heat pump and heat storage are used to keep the zonal temperature inside the room. Thermal intertie of room is also being used.

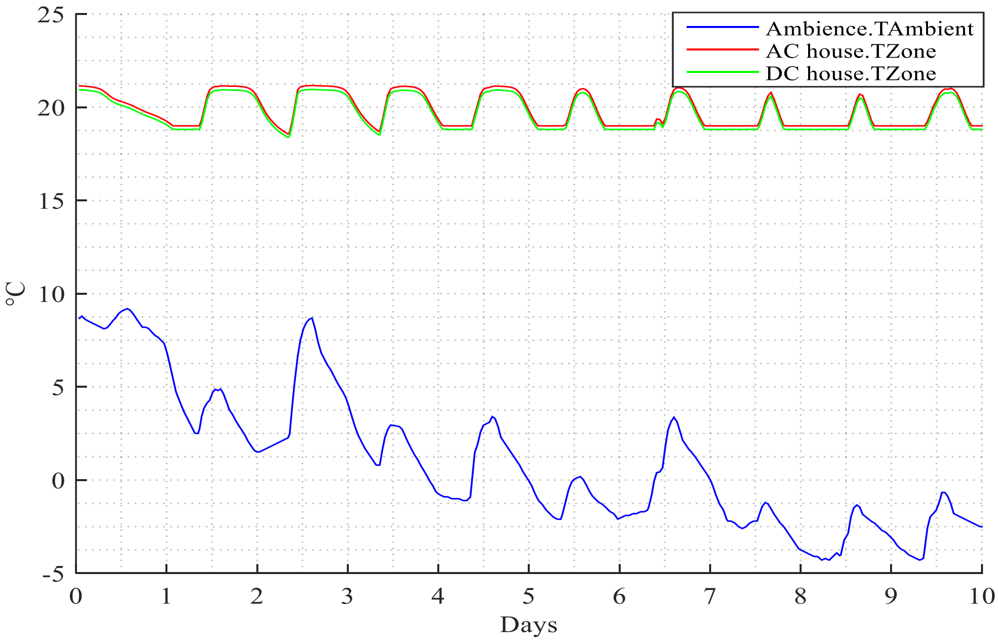

Figure 45 Temperature inside the house during winter.

3.2.2Summer days

Figure 46 shows the consumption requirement for 10 days of March month (11^{th} to 20^{th} March). It shows the PV generation is excess for requirement of house. This happens because heat pump requirement decreases and refrigeration load or cooling loads increases during summer. This excess energy can used to charge BESS. Excess energy of PV can be used during winter using BESS. The idea is to use excess during summer to charge BESS and discharge BESS during winter.

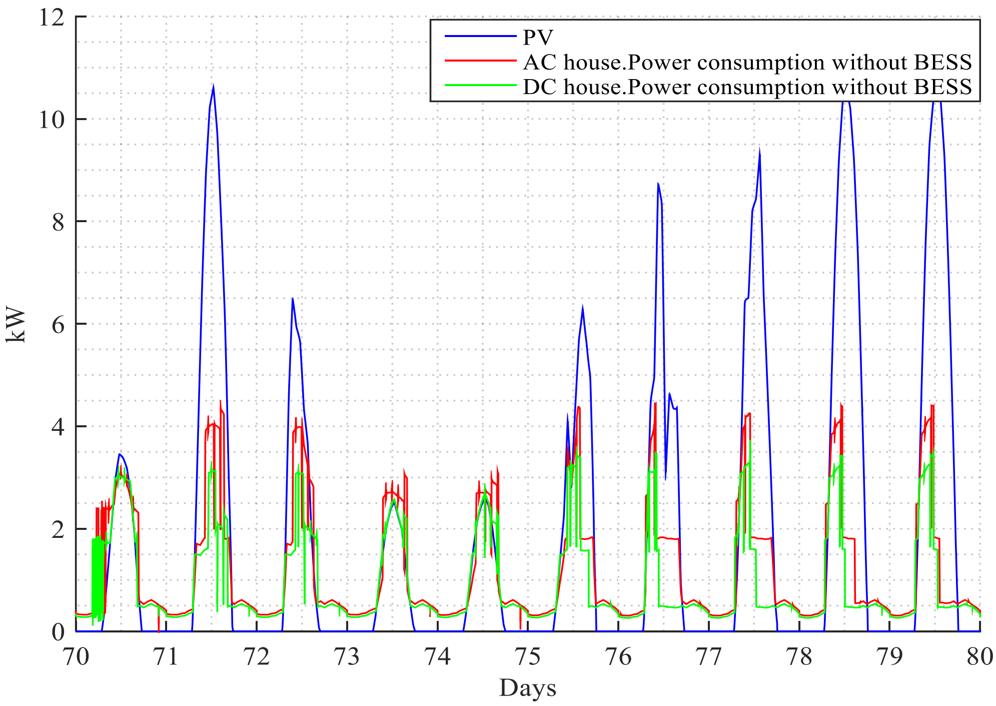

Figure 46 DC and AC house load distribution during summer.

Figure 47 shows the use of BESS to capture PV energy and store. Off grid is not only give independent of one’s energy requirement but also 100 % reliable. This increases the SC to 90 %.

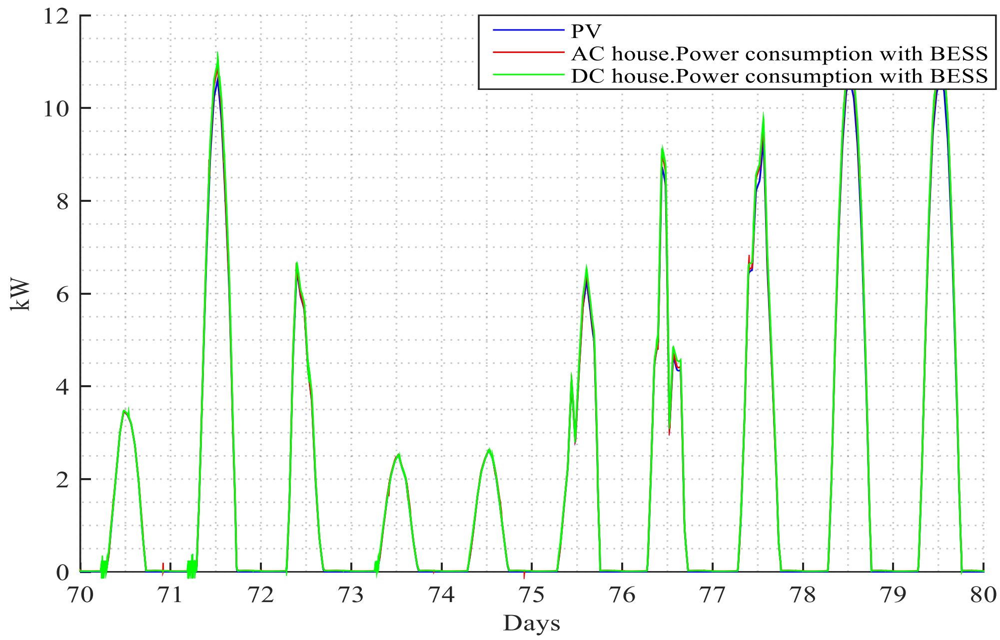

Figure 47 Consumption pattern with BESS in off grid situation during sum-

Figure 48 shows the temperature inside the house remain stable within a 1℃ temperature band width.

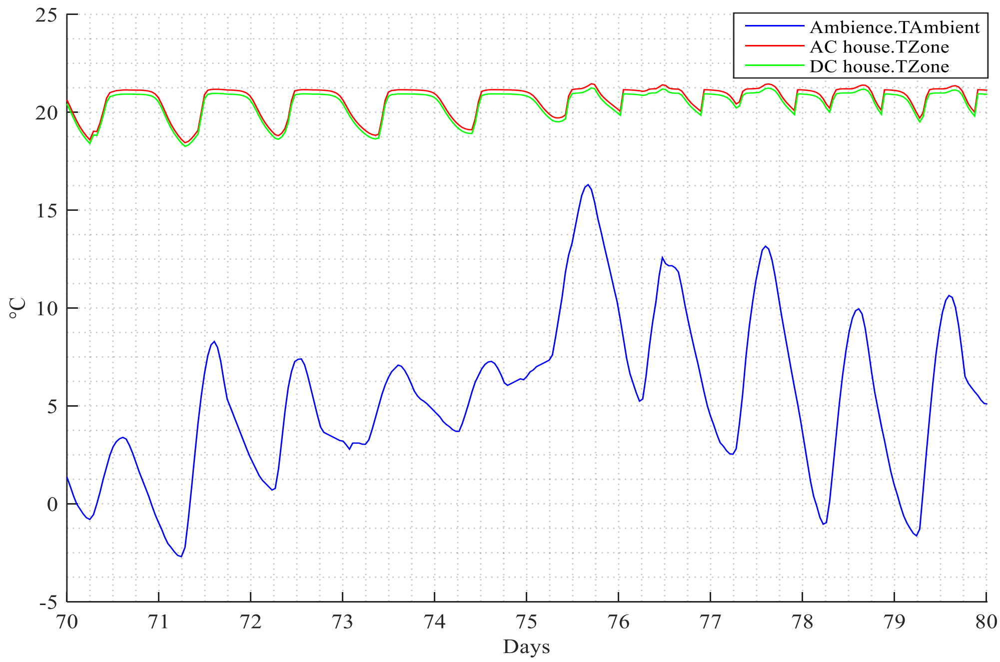

Figure 48 Temperature inside the house during March.

3.2.3One year analysis

Figure 49 shows the SOC of the BESS, a large battery of 850 kWh is being used. During winter, as PV generation is not sufficient. So the energy deficit is being supplied by BESS. During summer, BESS gets charged. DOD is 90 %. The maximum power rate of charging and discharging is 10 kW/h.

This shows that even 850 kWh is falling short as in the end of year, SOC is 50 %. So for continous off grid run, the BESS needed is 1350 kWh for AC house. The DC house require 1000 kWh of BESS.

Figure 49 SOC of BESS for off grid in one year

Figure 50 shows that power difference between house consumption with BESS and PV generation. This shows that during winter, power difference is zero and during summer, excess power is available. So Off grid is self-sufficient and reliable.

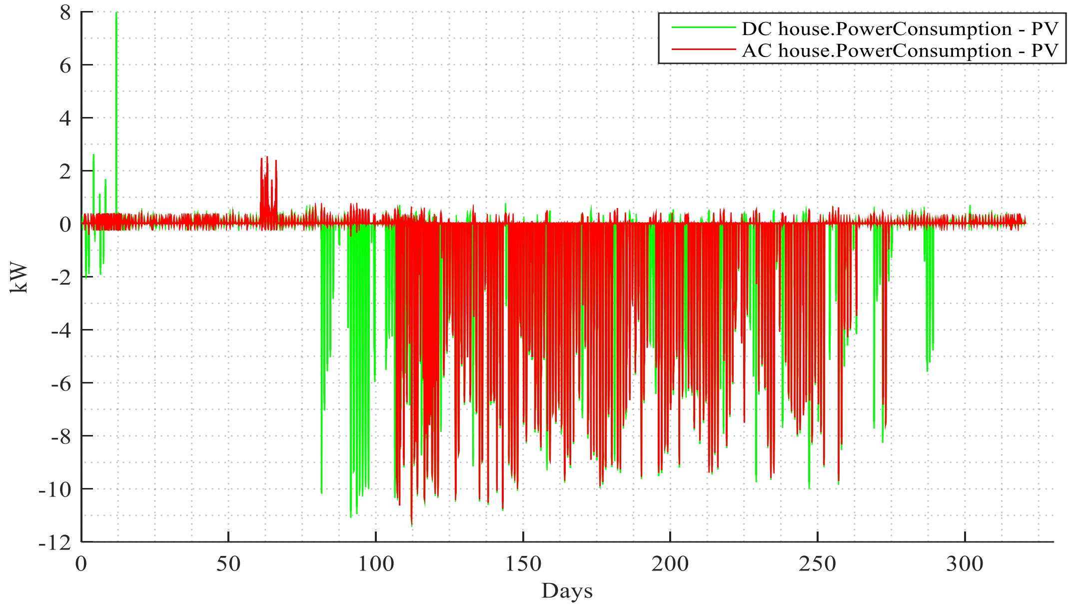

Figure 50 Power balance between home consumption with BESS and PV generation for off grid.

Figure 51 shows that the consumption pattern of house with BESS. This shows BESS along with PV generation compensates all the house consumption.

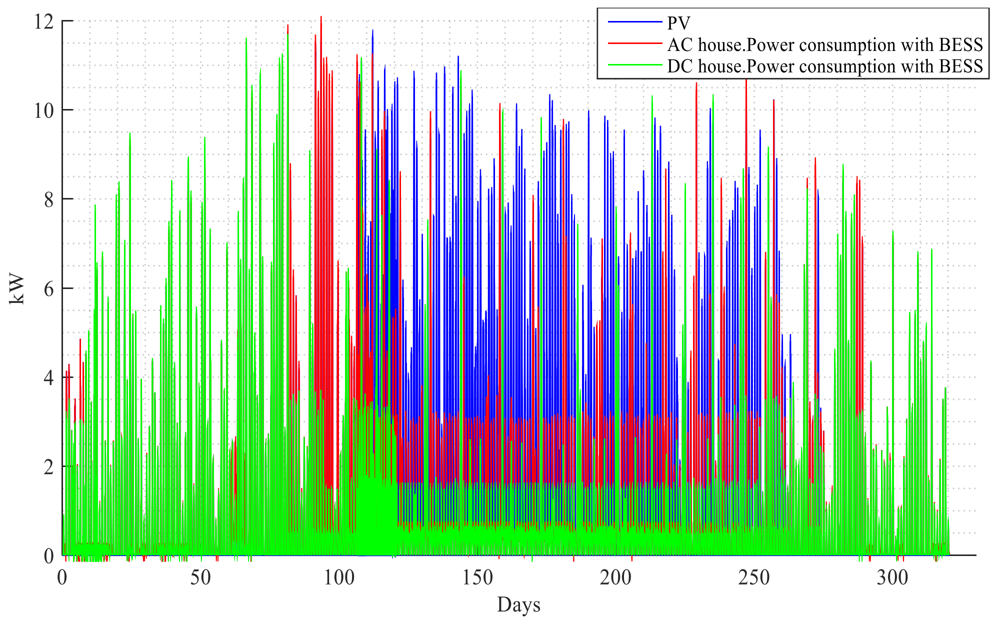

Figure 51 Consumption pattern with BESS in off grid.

3.3 Price function based demand response

This is based on fixed contract. A variable tariff is considered. Figure 52 shows that four slot of pricing has been decided. A reference price is decided. Whenever the grid price is less than reference price then price function is active otherwise it is inactive. Figure 53 shows the price function depending on reference price controls the price during peak hours and saves money.

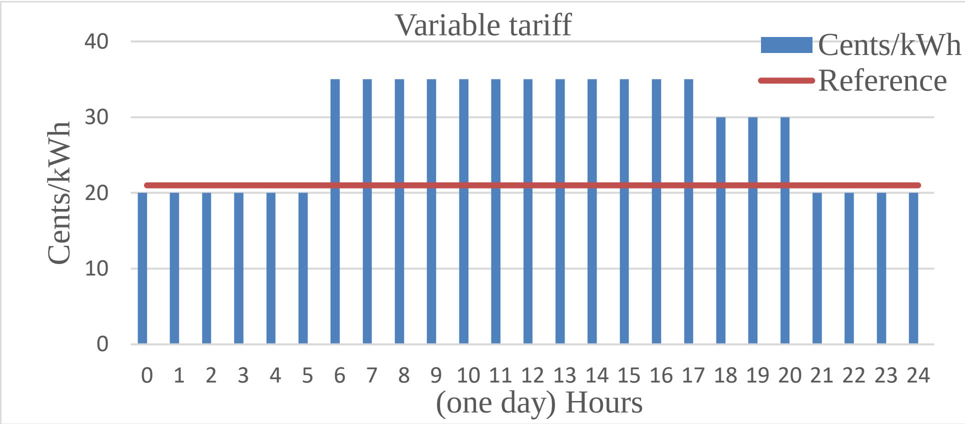

Figure 52 Variable tariff.

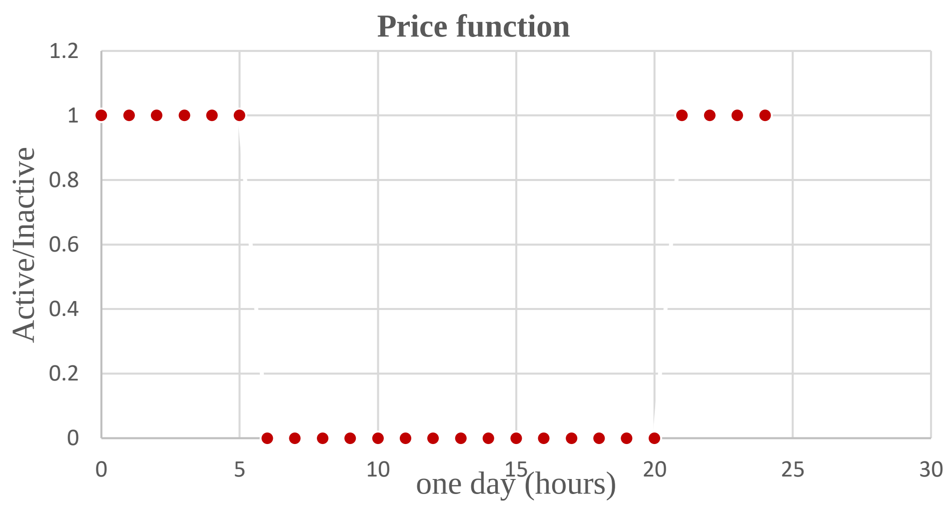

Figure 53 Price function.

3.3.1 Algorithm of optimized heat pump controller.

Figure 54 shows the optimized algorithm of heat pump. In the algorithm, whenever price function is active, heat pump will take power from grid if needed. It not only optimizes the SC of PV generation but also uses internal thermal inertia of the room. It operates within a temperature band. The algorithm finds a cost benefit between optimal sizing and optimal SC: To take in effect of thermal inertia in the house, the zonal temperature is maintained within a temperature bandwidth. Here in current simulation, a band width of ±1°C is being performed. It maintains the zonal temperature within a temperature band of 19°C to21°C. Here reference temperature is 20°C .

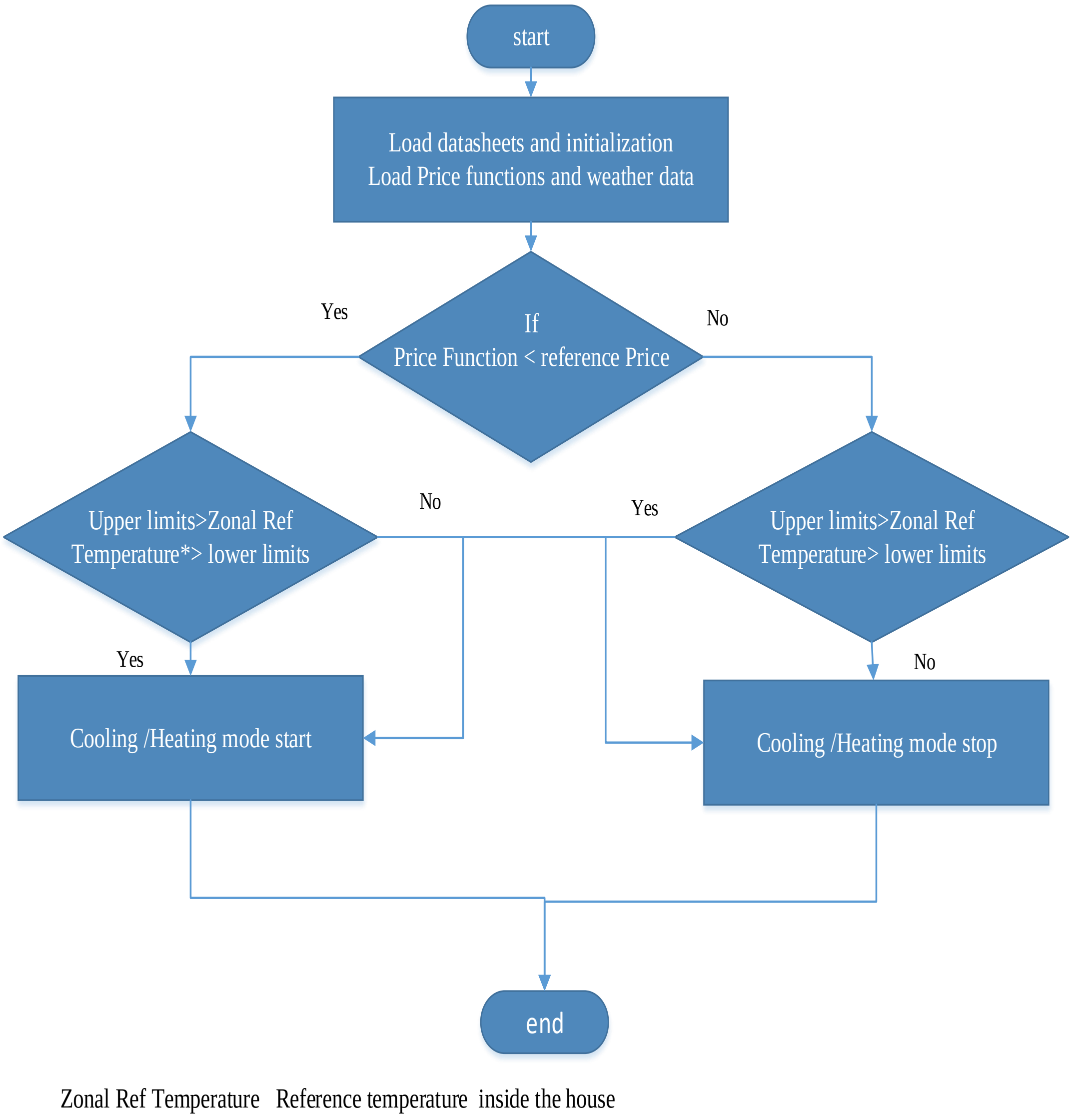

Figure 54 Algorithm for optimized heat pump controller.

3.3.2Algorithm of optimized BESS.

Figure 55 shows the optimized algorithm of BESS. In the algorithm, whenever price function is active, BESS will take power from grid if needed. It not only optimizes the SC of PV generation but also optimizes the cost of energy consumed from grid. The algorithm has different power charging and discharging rate. The fast charging algorithm is used. Here charging rate is more than discharging rate. Here the charging rate is 8 kW/h and discharging rate is 5 kW/h.

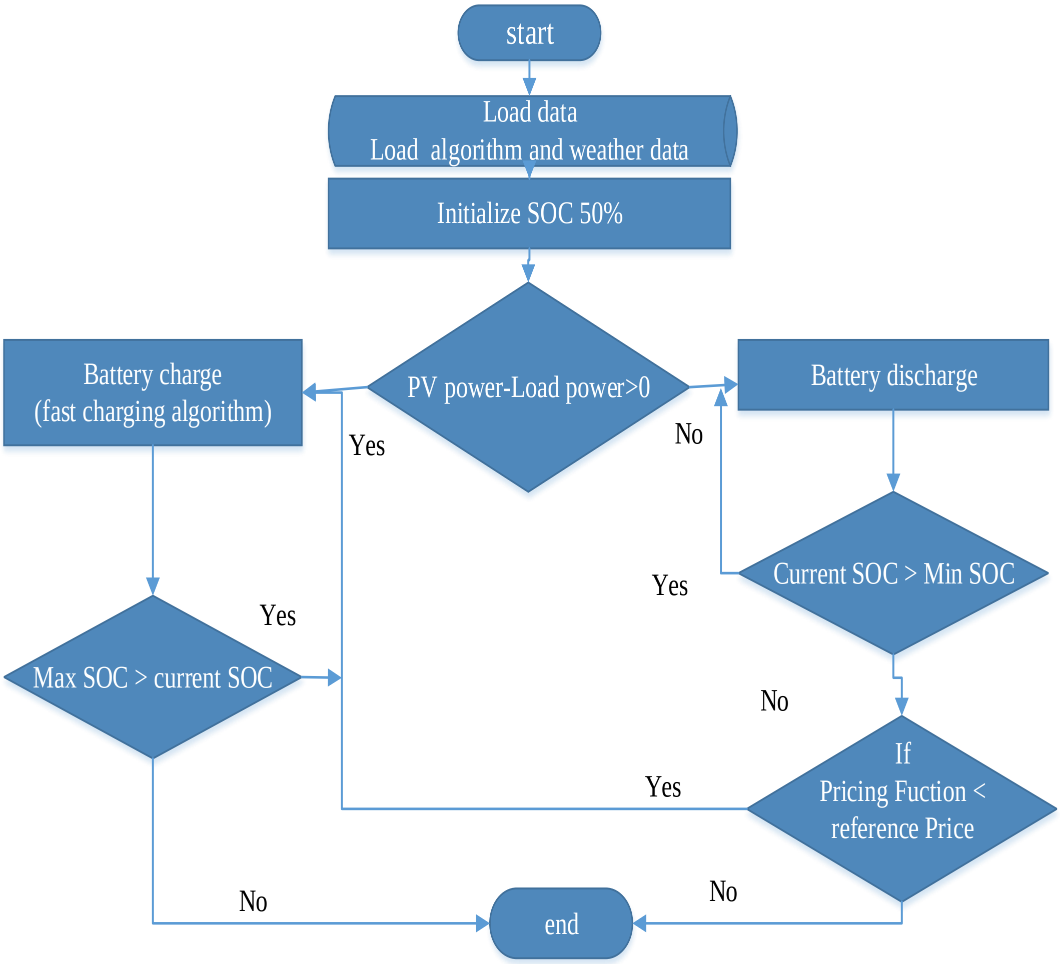

Figure 55 Algorithm of optimized BESS.

Table 5 shows the optimized parameters obtained from optimized sizing algorithms. Figure 57 shows the consumption of the house with BESS. BESS charges when grid prices is less and discharges whenever solar energy is lesser than the energy requirement. So whenever grid price is less, fast charging of BESS and heat pump takes place. Heat pump starts and store thermal energy in heat storage.

Table 5 Optimized sizing parameters in DR house.

Engineering parameters	**AC house**	**DC house**
PV generation size	12 kW_p	9 kW_p
Battery size	20 kWh	15 kWh
converter size	10.8 kW	8.1 kW

3.3.3 Winter days

This Figure 56 shows that consumption pattern of house without BESS.

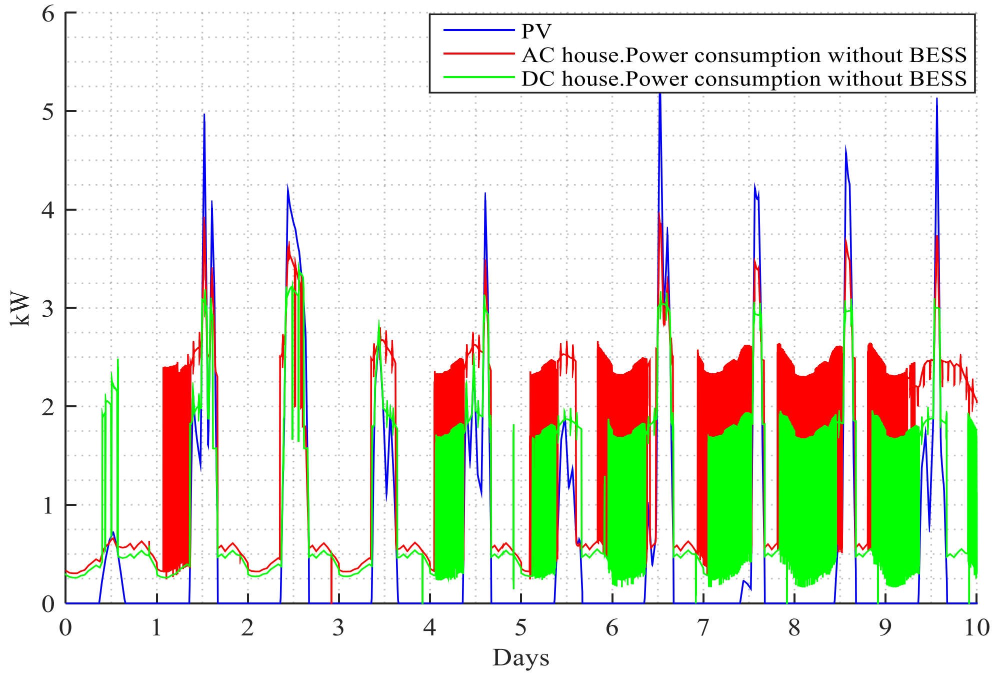

Figure 56 House loads during winter days.

Figure 57 shows that power difference between energy required by the house and solar energy. The energy difference is fulfilled by energy from the grid. The energy from grid is taken whenever grid prices are lower. The price function optimizes the energy consumption from grid.

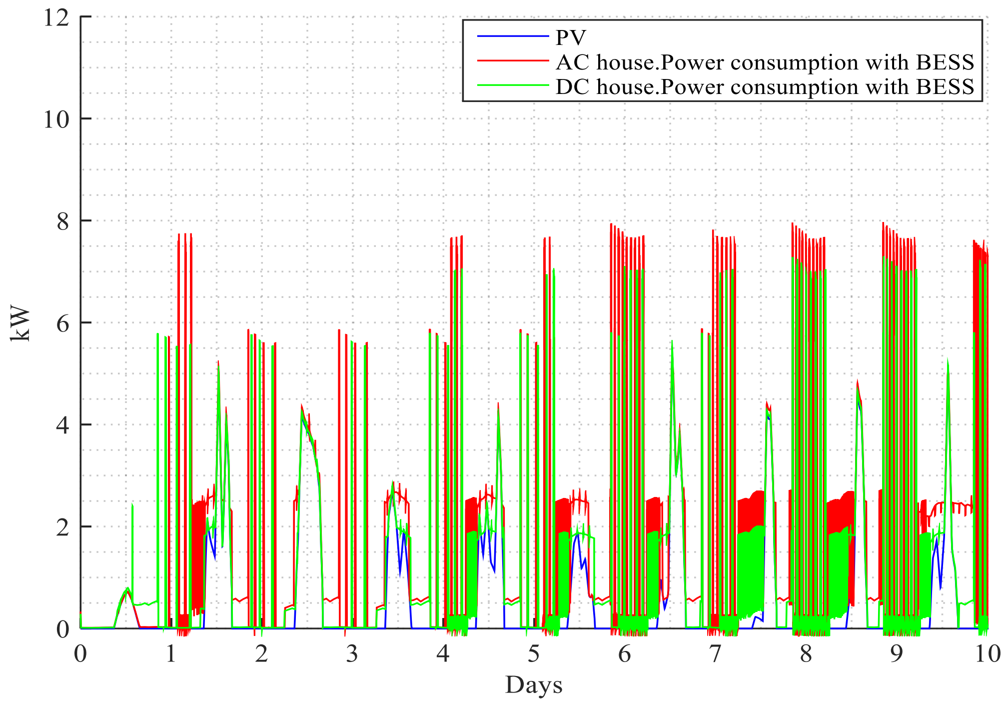

Figure 57 Power required from grid with demand response.

Figure 58 shows the AC house needs 11000 kWh annually where 1. EV consumes 3000 kWh, 2. Home appliances consumes 3560 kWh, 3. Heat pump consumes 4500 kWh. The DC house needs) 9250 kWh annually where 1. EV consumes 3000 kWh, 2. Home appliances consumes 2900 kWh, 3. Heat pump consumes 3350 kWh. Here PV generation is 13000 kWh and energy from grid is 1500 kWh.

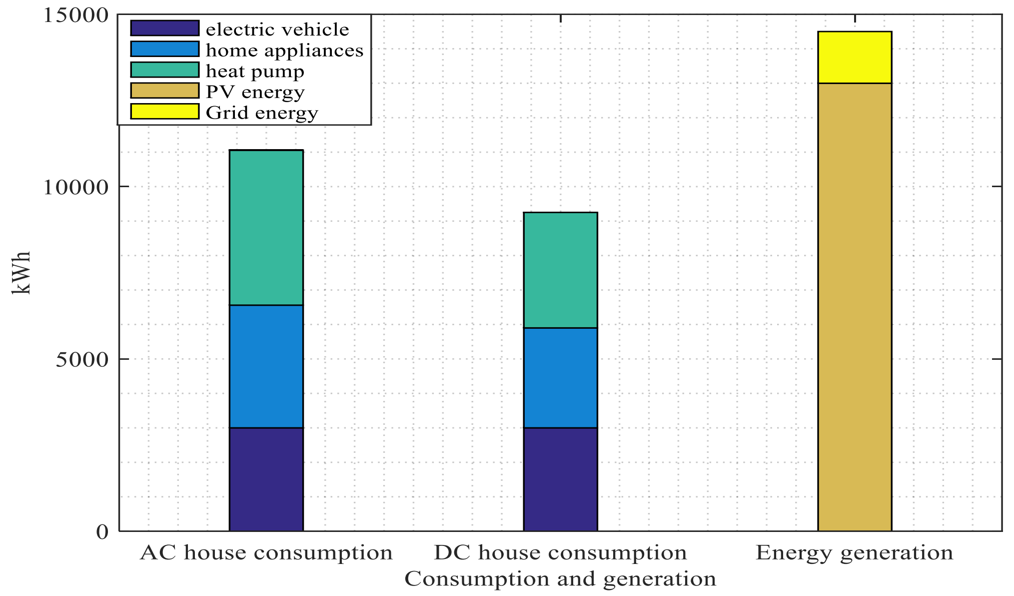

Figure 58 Consumption and generation patterns for one year.

Figure 59 shows that zonal temperature inside the house is within 1℃ width (19 to21℃) where reference temperature is20℃. Heat pump and heat storage are used to keep the zonal temperature inside the room. Thermal intertie of room is being used.

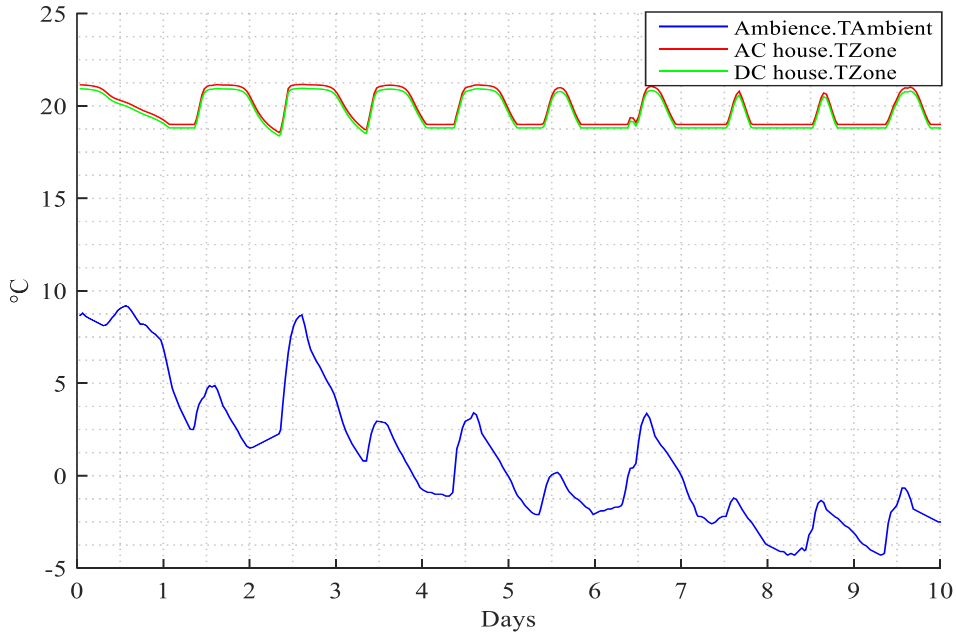

Figure 59 Temperature inside the house during winter days.

Figure 60 shows the variation of SOC of battery with the price function. When price function is active, it fast charges. The DOD during price fiction is 10 %. It charges whenever excess solar energy available or price fiction is active.

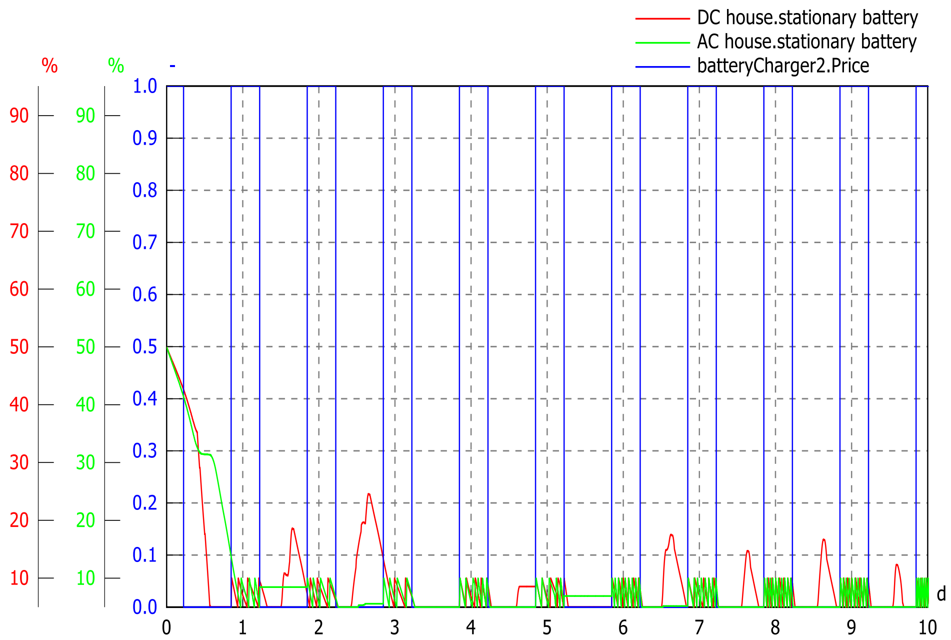

Figure 60 SOC of BESS during winter.

Figure 61 shows the SOC of heat storage.

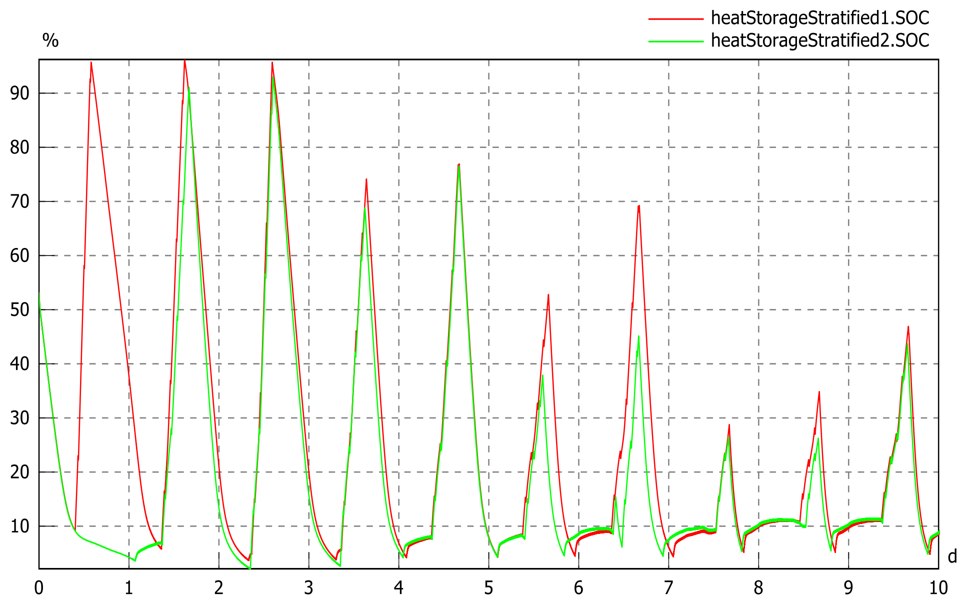

Figure 61 SOC of heat storage system during winter days.

3.3.4Summer days

Figure 62 shows the consumption requirement for 10 days of March month (11th to 20th March). It shows the PV generation is excess for requirement of house. This happens because heat pump requirement decreases and refrigeration load or cooling loads increases during summer. This excess energy can used to charge BESS.

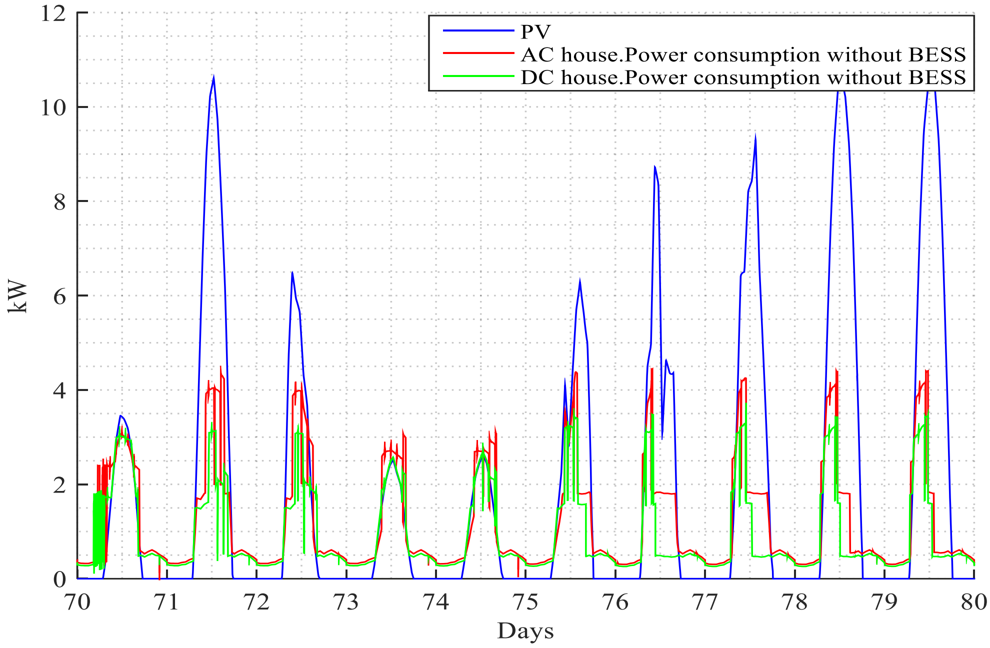

Figure 62 House loads during summer days.

Figure 65 shows the house with BESS captures optimal solar energy. Here SC is 85 % lesser than off grid situation as compare to winter days, energy from grid is less. In winter, energy of grid is required more as compared to sunny days.

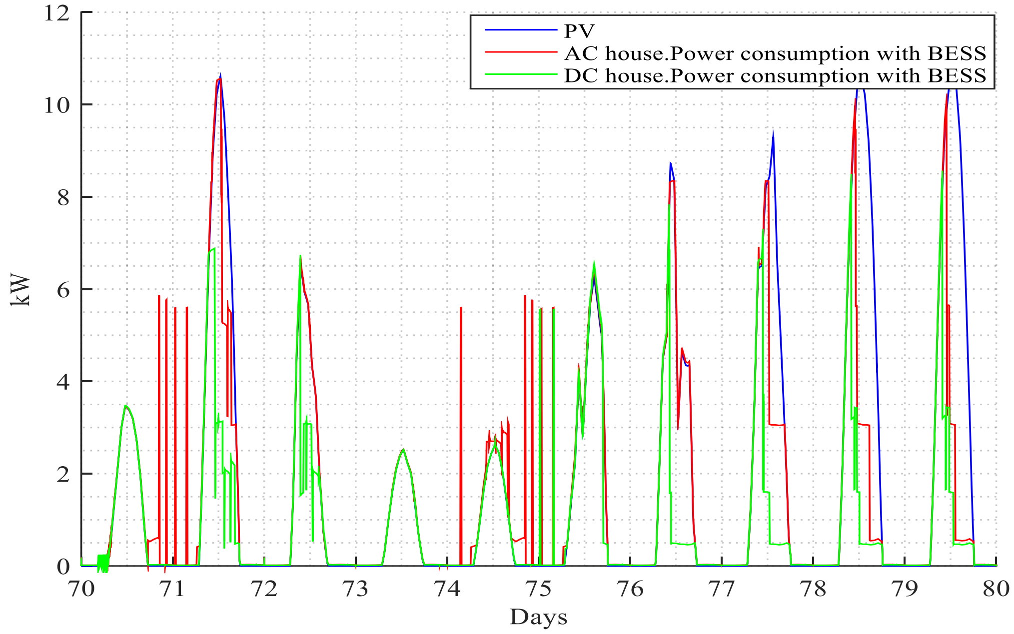

Figure 63 Consumption pattern of house with demand response during summer.

Figure 64 shows the SOC of BESS during summer days. During winter days, solar energy available for BESS charging is less. But during summer days, sufficient solar energy is available for BESS charging.

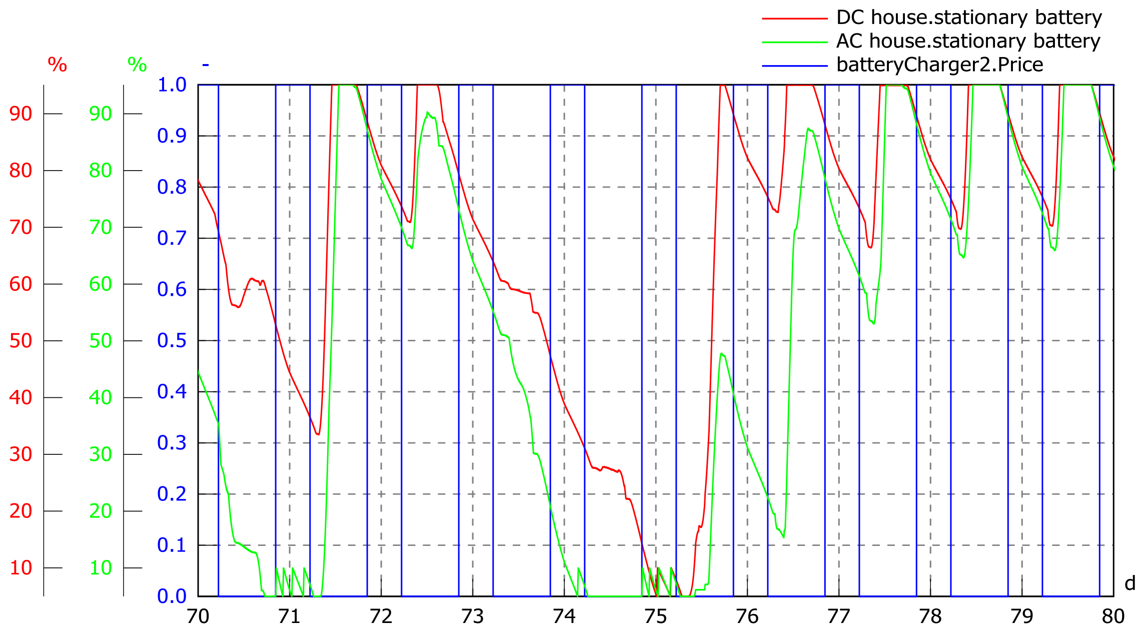

Figure 64 SOC of BESS during summer.

3.3.5One year analysis

Figure 65 shows the consumption pattern of both house for one year. The consumption loads are higher during winter as compare to summer. The difference in the consumption pattern of AC and DC house is visible. The excess solar energy available can be captured by using BESS.

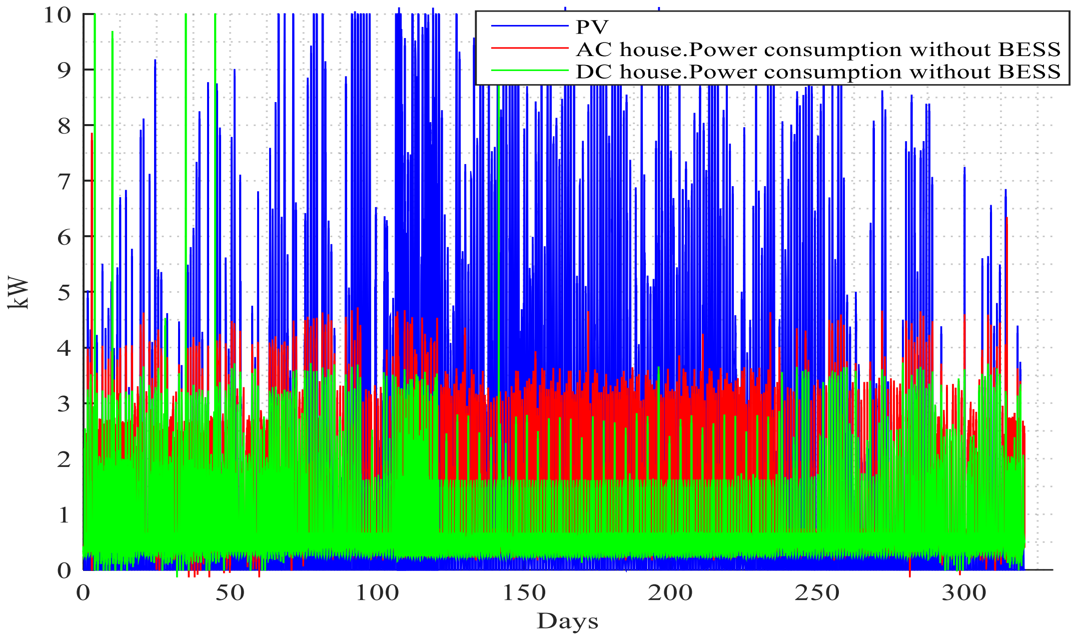

Figure 65 Consumption pattern of house with demand response during one year.

Figure 66 shows that with the use of BESS, the SC can be increased to 80 %. The BESS captures excess energy of solar. The consumption pattern is improved using BESS.

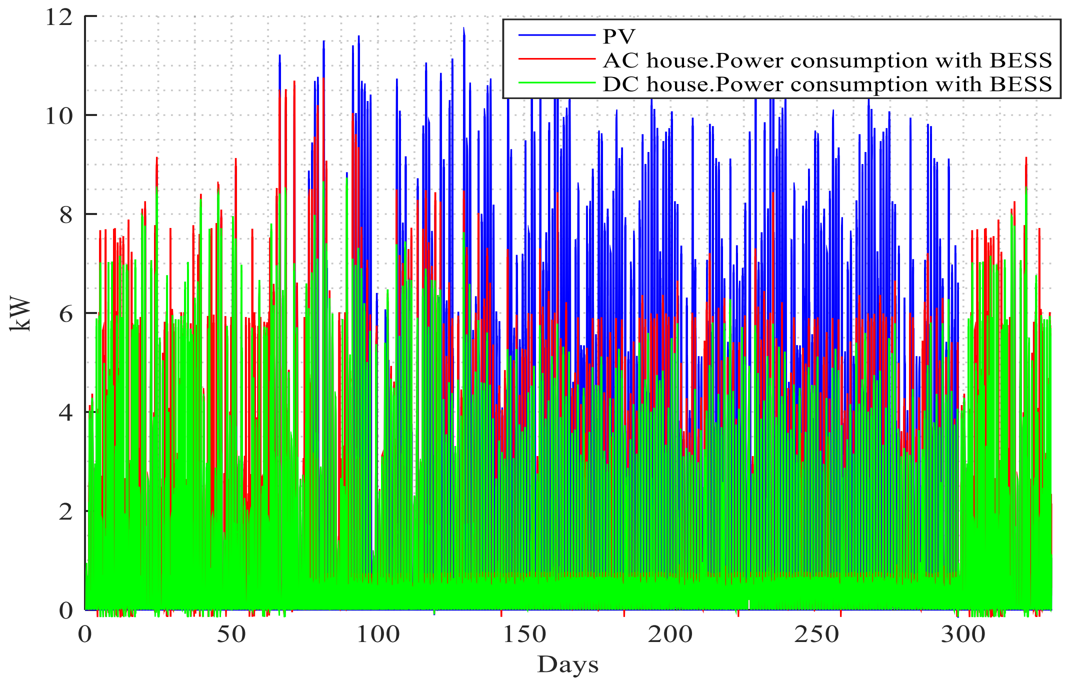

Figure 66 Consumption pattern of house with BESS.

Figure 67 shows the charging and discharging cycle of BESS. The winter and summer pattern shows that during winter, PV generation is not sufficient whereas in summer, PV generation is self-stuffiest. The fast charging algorithm in BESS is visible during winter days. BESS charges from utility grid or micro wind turbine during winter season as sufficient solar energy is not available but during summer, the PV generation is self-sufficient, so the BESS charges from solar energy. Here during summer, the BESS works in DOD 40 %. The effect of off grid and demand response algorithms on battery life is to be further studied. The effect of fast charging algorithms on battery life time is to be further studied.

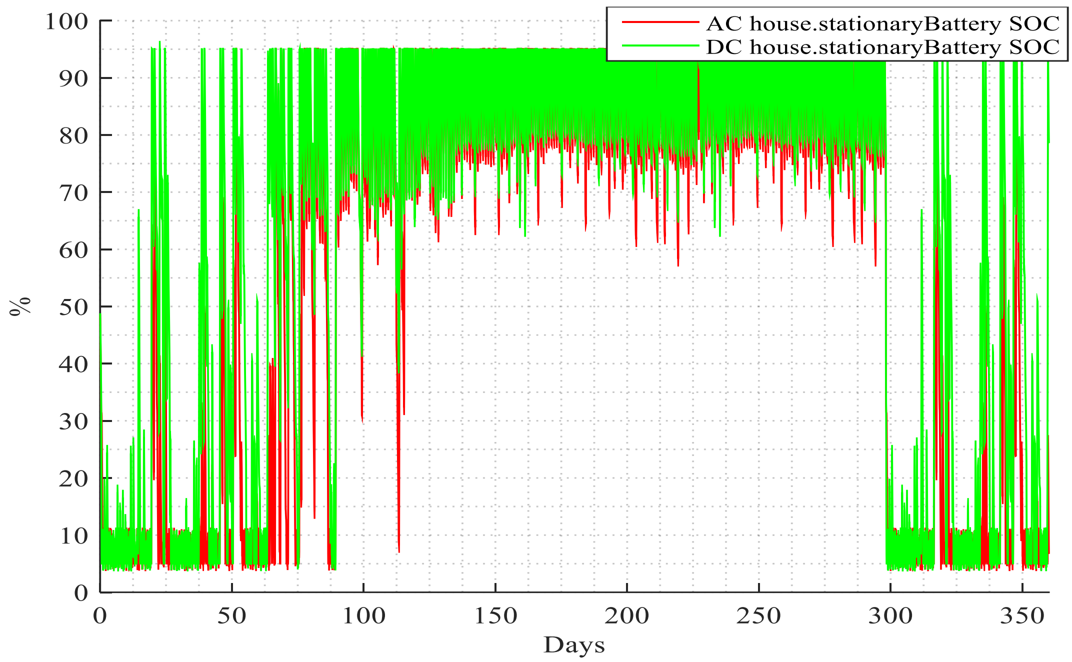

Figure 67 SOC of BESS in DR house.

Figure 68 shows the power difference between houses required energy and PV generated energy. The energy deficit is either taken from grid. The positive power difference shows the amount of energy needed from utility grid.

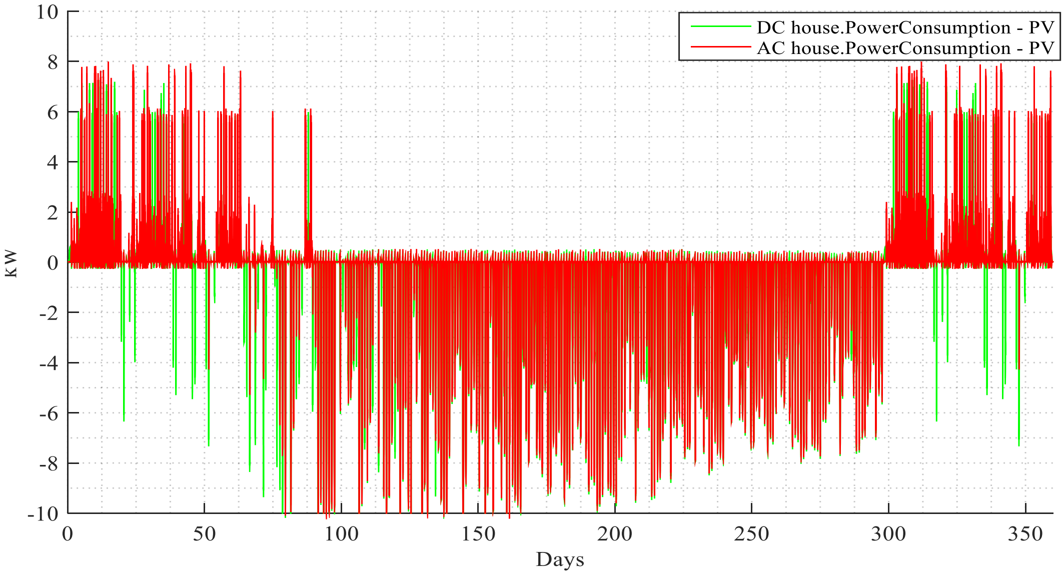

Figure 68 Power difference with BESS.

4 Economic analysis

4.1 Price function based demand response

Table 6 gives the data for economic calculations. Installation cost for AC house is 70000 € and DC house is 51100 € [7,36]. This shows that initial expenditure in DC house is 30 % lesser than AC house. Further cost can decreased by using price based demand response. In the simulation, AC house consumes 1500 kWh of energy from the grid. It is found that demand response saves 8 % of consumer electricity bill annually.

Electricity bill without demand response: $29 \text{ cents/kWh} * 1450 \text{ kWh} = 421€$

Electricity bill with demand response:

$$35 \text{ cents/kWh} * 600 \text{ kWh} + 20 \text{ cents/kWh} * 900 \text{ kWh} = 390 \text{ €}$$

Table 6 Economic parameters.

Economic parameters	**cost**	**unit**	**source**
PV module cost	910	€/kW	[6]
PV balance of system cost	4200	€/kW	[6]
Inverter cost	310	€/kW	[6]
Rectifier cost	310	€/kW	[6]
Bidirectional inverter cost	620	€/kW	[6]
AC circuit breaker cost	12	€/unit	[6]
DC circuit breaker	17	€/unit	[6]
battery cost	500	€/kWh	[6]
Discount rate	0	%	[6]
Maintenance cost	1% of installation cost		[6]
Feed in tariff	14	cents/kWh	[5,6,16]
EEG surcharge	6.3	cents/kWh	[5,6,16]
Electricity from grid	29	cents/kWh	[16,37]
Lifetime parameters	**lifetime**	**unit**	**source**
PV panel	25	Years	[6]
Balance of system	25	Years	[6]
Inverter	10	Years	[6]
Rectifier	10	Years	[6]
Bidirectional inverter	10	Years	[6]
Battery	10	Years	[6]
AC appliances	10	Years	[6]
DC appliances	10	Years	[6]
circuit breaker	20	Years	[6]

$$\text{LCOE} = \frac{\text{I}_\text{o} + \sum_{t=1}^{n} \frac{A_t}{(1+\text{r})^t}}{\sum_{t=1}^{n} \frac{M_t}{(1+\text{r})^t}} \tag{28}$$

Where

I_o	Initial investment.
LCOE	Levelised cost of energy in cent/kWh
r	Rate of interest in %
n	Number of life time years.
A_t	Annual cost in euro (electricity charges and surcharges are included).
M_t	Produced quantity of energy in kWh.

LCOE for an AC house is

$$\text{LCOE}_{\text{AC}} = \frac{-70000\,€ + \sum_{t=1}^{25} \frac{13000\ \text{kWh} * 0.3\ €/\text{kWh}}{(1+0.0)^t}}{\sum_{t=1}^{25} \frac{13000\ \ \text{kWh}}{(1+0.0)^t}} = 9\ \text{cent/kWh}$$

LCOE for a DC house is

$$\text{LCOE}_{\text{DC}} = \frac{-50000\,€ + \sum_{t=1}^{25} \frac{9500\ \text{kWh} * 0.3\ €/\text{kWh}}{(1+0.0)^t}}{\sum_{t=1}^{25} \frac{9500\ \ \text{kWh}}{(1+0.0)^t}} = 7.5\ \text{cent/kWh}$$

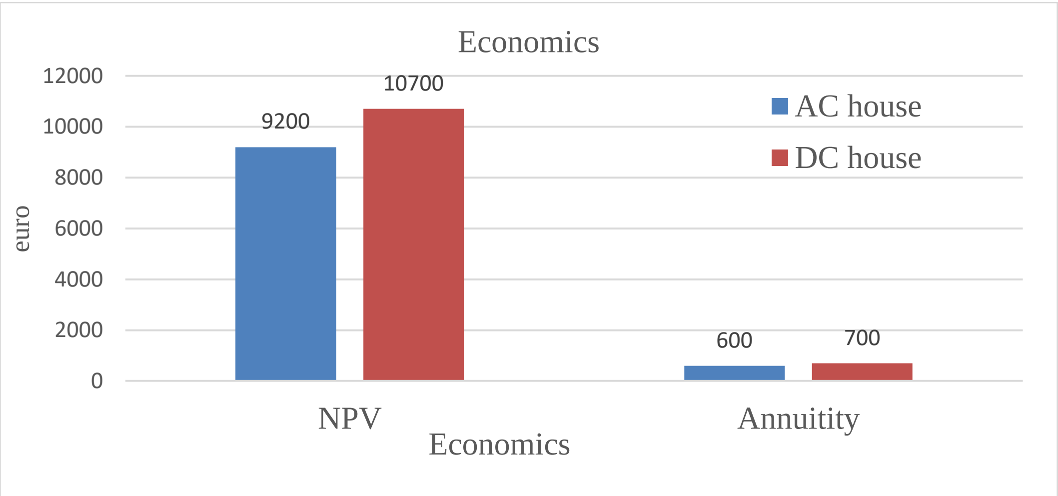

Figure 69 Net present value and Annuity.

Figure 69 shows Net present value (NPV) of AC house is 9,200 € and DC house is 10,700 €. Annuity of AC house is 600 € and DC house is 700 €. Equation 28 gives levelised cost of electricity. Figure 70 shows Levelised cost of energy for AC house is 9 cent/kWh and for DC house is 7.5 cent/kWh.

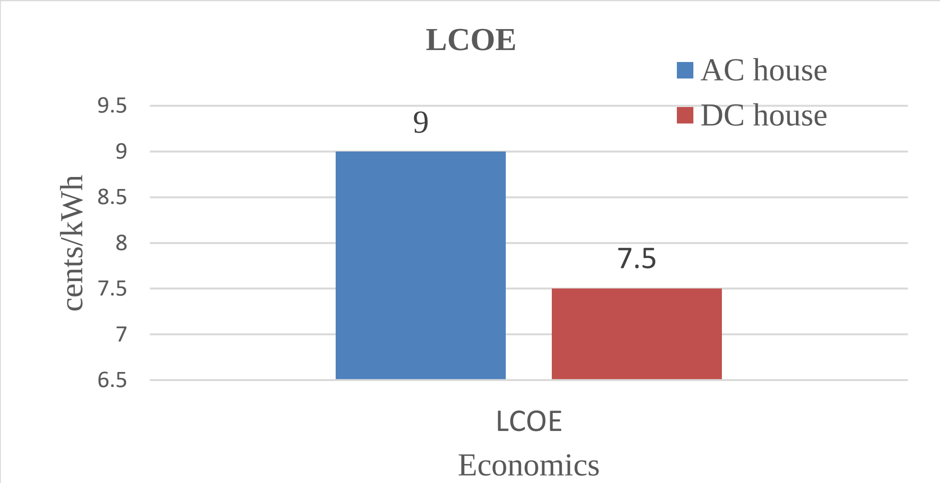

Figure 70 Levelised cost of energy

4.2 Off Grid

As the off grid system needs a large BESS system, so it become oversize and increase the initial investment.

The Initial investment in AC house comes $500{,}000$ € and DC house comes$350{,}000$ €. [7, 23]. The NPV for AC house is -400500 € and DC house is -250000 €. so this is not an economic viable option.

5 Conclusion

In future decentralized renewable energy sources (DRES) will play a crucial role in power system. Decentralized power system will not only need to be stable but also more reliable. The micro grid formed by DRES, BESS and thermal energy storage system holds key to make power system more reliable, stable and independent. Reliability and independence decides the cost of the power system. For example power system connected to utility grid is more reliable but not independent but the off grid power system is more independent but less reliable. Finding optimum solution between reliability and independence holds the key to more viable and stable power system.

Here a DC smart house is being simulated for off grid system and demand response with utility grid system. The cost benefits between optimum SC and optimum sizing is being analyzed.

Energy management system uses various demand repose algorithms to optimize the energy distribution. EMS optimizes the SC and distribution system. EMS optimizes SC to 90 % in off grid system and 80 % in price function based demand response house. The EMS saves energy and reduces the cost.

When DC smart house is being compared with AC smart house, the results show that DC smart houses saves 24 % of electrical energy. Optimal sizing saves 30 % of initial investment. Optimal sizing of DC smart house reduces the size of PV array by 24 %, battery size by 25 % as compared to AC smart house.

In this study, the demand response of housing system which is based on external signal (a pricing signal based) is being analyzed. Comparison between demand response and without demand response housing system has been done. A DC house with price function based DR saves 8 % of consumer electricity bill as compared to a DC house without DR. Levelised cost of electricity (LCOE) of the AC house is 9 cent/kWh and DC house is 7.5 cent/kWh. Although sometime there is more energy consumption in DR but it can save money.

This open doors towards more reliable and independent virtual micro grid. In future work, advanced controller based on machine learning can be implemented. Better design on BESS and EMS not only save money but also increases the efficiency of the system.

6 References

[1] Renewable Energy Germany | German Energy Transition. [April 17, 2017]; Available from: https://1-stromvergleich.com/strom-report/renewable-energy-germany/.

[2] Federal Ministry for Economics Affairs and Energy. BMWi - Federal Ministry for Economic Affairs and Energy - Publications. [April 17, 2017]; Available from: http://www.bmwi.de/Navigation/EN/Service/Publications/publications.html.

[4] supervised by: Univ. Prof. Dr. techn. Reinhard HAAS. Battery Energy Storage System Technology, Applications and Economy of Large Projects.

[5] Federal Ministry for Economics Affairs and Energy. BMWi - Federal Ministry for Economic Affairs and Energy - Homepage. [May 01, 2017]; Available from: http://www.bmwi.de/Navigation/EN/Home/home.html/.

[6] Vossos V, Garbesi K, Shen H. Energy savings from direct-DC in U.S. residential buildings. Energy and Buildings 2014;68:223–31.

[7] Glasgo B, Azevedo IL, Hendrickson C. How much electricity can we save by using direct current circuits in homes? Understanding the potential for electricity savings and assessing feasibility of a transition towards DC powered buildings. Applied Energy 2016;180:66–75.

[8] Optimizing energy savings from "Direct-DC" in U.S. residential buildings.

[9] Hayn M, Bertsch V, Fichtner W. Electricity load profiles in Europe: The importance of household segmentation. Energy Research & Social Science 2014;3:30–45.

[10] www.simulationx.com. [May 01, 2017]; Available from: https://www.simulationx.com/.

[11] Administrator. Welcome to OpenModelica - OpenModelica. [May 01, 2017]; Available from: https://www.openmodelica.org/.

[12] Garbesi K. Catalog of DC Appliances and Power Systems.

[13] Garbesi K. Catalog of DC appliances and power systems 2012.

[14] Messenger RA, Ventre J. Photovoltaic systems engineering, 3rd ed. Boca Raton, Fla.: CRC; 2010.

[15] Godoy-Shimizu D, Palmer J, Terry N. What Can We Learn from the Household Electricity Survey? Buildings 2014;4(4):737–61.

[16] What German households pay for power. [May 01, 2017]; Available from: https://www.cleanenergywire.org/factsheets/what-german-households-pay-power.

[17] Muthuvel P, Daniel SA, Yazhini DG. Retrofitting domestic appliances for PV powered DC Nano-grid and its impact on net zero energy homes in rural India. Engineering Science and Technology, an International Journal 2016;19(4):1836–44.

[18] TANG Z, LIU Y, LIU J, LI R, WEN L, ZHANG G. Multi-stage sizing approach for development of utility-scale BESS considering dynamic growth of distributed photovoltaic connection. J. Mod. Power Syst. Clean Energy 2016;4(4):554–65.

[19] Desroches L-B. Max Tech and Beyond: Maximizing appliance and equipment efficiency by design 2012.

[20] Desroches L-B, Garbesi K. Max Tech and Beyond: Maximizing Appliance and Equipment Efficiency by Design; 2011.

[21] COMM/ENV. Residential Monitoring to Decrease Energy Use and Carbon Emissions in Europe - Intelligent Energy Europe - European Commission. [March 19, 2017]; Available from: https://ec.europa.eu/energy/intelligent/projects/en/projects/remodece.

[22] List of Eligible Inverters per SB1 Guidelines - Go Solar California. [May 01, 2017]; Available from: http://www.gosolarcalifornia.ca.gov/equipment/inverters.php.

[23] COMM/ENV. Residential Monitoring to Decrease Energy Use and Carbon Emissions in Europe - Intelligent Energy Europe - European Commission. [May 01, 2017]; Available from: https://ec.europa.eu/energy/intelligent/projects/en/projects/remodece.

[24] Panasonic debuts high-powered photovoltaic module HIT® N330 for UK and European markets, sets new module efficiency record. [April 09, 2017]; Available from: http://news.panasonic.co.uk/pressreleases/panasonic-debuts-high-powered-photovoltaic-module-hit-r-n330-for-uk-and-european-markets-sets-new-module-efficiency-record-1229674.

[25] Panasonic Solar Panels Support - HIT Solar Panels | Panasonic. [April 09, 2017]; Available from: http://business.panasonic.com/solutions-hvac-hitsolarpanels-solartechnology?sc_sp=business-category_featuresSolutions_products-hvac-solarpanels_Panasonic%20Solar%20Technology.

[26] Fleer J, Stenzel P. Impact analysis of different operation strategies for battery energy storage systems providing primary control reserve. Journal of Energy Storage 2016;8:320–38.

[27] Thomas Thierschmidt, Martin Rothert, Felix Kever, Scott Partlin. Optimised Storage for increased PV Self Consumption.

[28] Moshövel J, Kairies K-P, Magnor D, Leuthold M, Bost M, Gährs S et al. Analysis of the maximal possible grid relief from PV-peak-power impacts by using storage systems for increased self-consumption. Applied Energy 2015;137:567–75.

[29] Appen J von, Braun M, Stetz T, Diwold K, Geibel D. Time in the Sun: The Challenge of High PV Penetration in the German Electric Grid. IEEE Power and Energy Mag. 2013;11(2):55–64.

[30] Castillo-Cagigal M, Caamaño-Martín E, Matallanas E, Masa-Bote D, Gutiérrez A, Monasterio-Huelin F et al. PV self-consumption optimization with storage and Active DSM for the residential sector. Solar Energy 2011;85(9):2338–48.

[32] Aghamohammadi MR, Abdolahinia H. A new approach for optimal sizing of battery energy storage system for primary frequency control of islanded Microgrid. International Journal of Electrical Power & Energy Systems 2014;54:325–33.

[33] Almeida A de, Fonseca P, Schlomann B, Feilberg N. Characterization of the household electricity consumption in the EU, potential energy savings and specific policy recommendations. Energy and Buildings 2011;43(8):1884–94.

[34] Velik R. The influence of battery storage size on photovoltaics energy self-consumption for grid-connected residential buildings. IJARER International Journal of Advanced Renewable Energy Research 2013;2(6).

[35] Solar Irradiance Calculator. [May 06, 2017]; Available from: http://solarelectricityhandbook.com/solar-irradiance.html.

[36] Ghosh I. FLA Presentation for Research Review Meetings.

[37] Federal Ministry for Economics Affairs and Energy. Overview of legislation governing Germany's energy supply system. [May 01, 2017]; Available from: https://www.bmwi.de/Redaktion/EN/Publikationen/gesetzeskarte.html.

7 APPENDIX A

7.1 Modelica code of coltroller with Price function demand response is shown.

```
model HPController_DR_AC_PV_SC "HPControl-
ler_DR_AC_PV_SC.mo"
    Boolean Price "pRICING FUNCTION on/off-switch";
    Modelica.Blocks.Tables.CombiTable1D EPricing(
        tableOnFile=true,
        tableName=PriceTable,
        fileName=HPFile) "Pricing function of grid" annota-
tion(
        Placement(transformation(extent={{25,5},{45,25}})),
        Dialog(
            group="Coefficient of Performance",
            tab="Results 1"));
    parameter String HPFile=GreenBuilding.Utilities.Func-
tions.getModelDataDirec-
tory()+"\\heat_pump\\hp_data\\Stiebel_El-
tron\\WPL_10_ACS\\SE_WPL_10_ACS2.txt" "File name for heat
pump characteristics" annotation(Dialog(
        group="Heating Power",
        tab="HP - Power Data"));
    Boolean state "PV power  based model state on/off" an-
notation(Dialog(
        group="I/O",
        tab="Results 1",
        showAs=ShowAs.Result));
    GreenBuilding.Interfaces.Ambience.AmbientCondition Am-
bientConditions if FlowControl or SourceAir "Ambient Condi-
tions Connection" annotation(Placement(
        transformation(extent={{600,-685},{620,-665}}),
        iconTransformation(extent={{90,190},{110,210}})));
    protected
        AverageTemperature TAverage(
            SupportPoints=SupportPointsAmbientTemperature,
            StepTime=TimeStepAmbientTemperature) if Flow-
```

```
Control "Temperature averaging model" annotation(Place-
ment(transformation(extent={{-150,100},{-130,110}})));
        DeIcing DEICING(
            tDeIcing=tDeIcing,
            tOperateIce=tOperation,
            tOn=tDeIcingOn,
            tOff=tDeIcingOff) if SourceAir "De-icing con-
troller" annotation(Placement(transformation(extent={{-
150,100},{-130,110}})));
        CirculationPump CPcontrol(
            tCPComp=tDelayCPComp,
            tCPIce=tDelayCPIce,
            TBoundIce=if SourceAir then TDeIcingBound else
280.15) "Circulation pump controller" annotation(Place-
ment(transformation(extent={{-150,100},{-130,110}})));
        SourcePump SPcontrol(
            tSP=tDelaySPComp,
            TBoundIce=if SourceAir then TDeIcingBound else
280.15) "Source pump controller" annotation(Place-
ment(transformation(extent={{-150,100},{-130,110}})));
        Compressor CompControl(
            tMinOff=tHPminOff,
            tMinOn=tHPminOn,
            tCompCP=tDelayHP,
            tCompSP=tDelaySP,
            TBoundIce=if SourceAir then TDeIcingBound else
280.15) "Compressor controller" annotation(Placement(trans-
formation(extent={{-150,100},{-130,110}})));
    public
        input Modelica.Blocks.Interfaces.RealInput TRef(
            quantity="Thermics.Temp",
            displayUnit="°C") "Reference temperature" anno-
tation(
            Placement(
                transformation(extent={{585,-390},{625,-
350}}),
                iconTransformation(extent={{-220,130},{-
180,170}})),
            Dialog(
```

```
                group="Temperature",
                tab="Results 1",
                showAs=ShowAs.Result));
        input Modelica.Blocks.Interfaces.RealInput TAct(
            quantity="Thermics.Temp",
            displayUnit="°C") "Actual temperature" annota-
tion(
            Placement(
                transformation(extent={{585,-440},{625,-
400}}),
                iconTransformation(extent={{-220,80},{-
180,120}})),
            Dialog(
                group="Temperature",
                tab="Results 1",
                showAs=ShowAs.Result));
        input Modelica.Blocks.Interfaces.RealInput TFlow(
            quantity="Thermics.Temp",
            displayUnit="°C") "Flow temperature" annota-
tion(
            Placement(
                transformation(extent={{585,-495},{625,-
455}}),
                iconTransformation(extent={{-220,-120},{-
180,-80}})),
            Dialog(
                group="Temperature",
                tab="Results 1",
                showAs=ShowAs.Result));
        input Modelica.Blocks.Interfaces.RealInput TReturn(
            quantity="Thermics.Temp",
            displayUnit="°C") "Return temperature" annota-
tion(
            Placement(
                transformation(extent={{585,-550},{625,-
510}}),
                iconTransformation(extent={{-220,-170},{-
180,-130}})),
            Dialog(
```

```
                group="Temperature",
                tab="Results 1",
                showAs=ShowAs.Result));
        input Modelica.Blocks.Interfaces.RealInput TFlow-
Ref(
            quantity="Thermics.Temp",
            displayUnit="°C") if not FlowControl "Reference
flow temperature" annotation(
            Placement(
                transformation(extent={{585,-600},{625,-
560}}),
                iconTransformation(
                    origin={-100,200},
                    extent={{-20,-20},{20,20}},
                    rotation=-90)),
            Dialog(
                group="Temperature",
                tab="Results 1",
                showAs=ShowAs.Result));
        input Modelica.Blocks.Interfaces.RealInput
EPriceAverage(quantity="price") "price" annotation(
            Placement(
                transformation(extent={{585,-345},{625,-
305}}),
                iconTransformation(
                    origin={-200,50},
                    extent={{-20,-20},{20,20}},
                    rotation=-90)),
            Dialog(
                group="I/O",
                tab="Results 1",
                showAs=ShowAs.Result));
        parameter String PriceTable="Pricing" "Table name
for Eprice " annotation(Dialog(
            group="Heating",
            tab="HP - Power Data"));
    protected
        Real TFlowControl(
            quantity="Thermics.Temp",
```

```
            displayUnit="°C") "Control temperature for
flow" annotation(Dialog(
            group="Temperature",
            tab="Results 1",
            showAs=ShowAs.Result));
    public
        output Modelica.Blocks.Interfaces.RealOutput qvRef(
            quantity="Thermics.VolumeFlow",
            displayUnit="m³/h") "target volume flow" anno-
tation(
            Placement(
                transformation(extent={{825,-380},{845,-
360}}),
                iconTransformation(
                    origin={-50,-200},
                    extent={{-10,-10},{10,10}},
                    rotation=-90)),
            Dialog(
                group="Volume Flow",
                tab="Results 1",
                showAs=ShowAs.Result));
        output Modelica.Blocks.Interfaces.RealOutput
qvSourceRef(
            quantity="Thermics.VolumeFlow",
            displayUnit="m³/h") "Reference volume flow of
source pump" annotation(
            Placement(
                transformation(extent={{825,-355},{845,-
335}}),
                iconTransformation(
                    origin={0,-200},
                    extent={{-10,-10},{10,10}},
                    rotation=-90)),
            Dialog(
                group="Volume Flow",
                tab="Results 1",
                showAs=ShowAs.Result));
        input Modelica.Blocks.Interfaces.BooleanInput Ena-
ble "External control input" annotation(
```

```
            Placement(
                transformation(extent={{585,-345},{625,-
305}}),
                iconTransformation(
                    origin={0,200},
                    extent={{-20,-20},{20,20}},
                    rotation=-90)),
            Dialog(
                group="I/O",
                tab="Results 1",
                showAs=ShowAs.Result));
        output Modelica.Blocks.Interfaces.BooleanOutput
HPon "Switch-on/off of HP" annotation(
            Placement(
                transformation(extent={{825,-430},{845,-
410}}),
                iconTransformation(ex-
tent={{190,140},{210,160}})),
            Dialog(
                group="I/O",
                tab="Results 1",
                showAs=ShowAs.Result));
        output Modelica.Blocks.Interfaces.BooleanOutput
CPon "Switch-on/off of circulation pump" annotation(
            Placement(
                transformation(extent={{825,-465},{845,-
445}}),
                iconTransformation(ex-
tent={{190,40},{210,60}})),
            Dialog(
                group="I/O",
                tab="Results 1",
                showAs=ShowAs.Result));
        output Modelica.Blocks.Interfaces.BooleanOutput
AUXon if AuxHeat "Auxiliary heat switch on/off" annotation(
            Placement(
                transformation(extent={{825,-500},{845,-
480}}),
                iconTransformation(extent={{190,-
```

```
10},{210,10}})),
            Dialog(
                group="I/O",
                tab="Results 1",
                showAs=ShowAs.Result));
        output Modelica.Blocks.Interfaces.BooleanOutput
SPon "Source pump switch on/off" annotation(
            Placement(
                transformation(extent={{825,-535},{845,-
515}}),
                iconTransformation(ex-
tent={{190,90},{210,110}})),
            Dialog(
                group="I/O",
                tab="Results 1",
                showAs=ShowAs.Result));
        output Modelica.Blocks.Interfaces.BooleanOutput DE-
ICINGon if SourceAir "Switch on/off de-icing" annotation(
            Placement(
                transformation(extent={{825,-565},{845,-
545}}),
                iconTransformation(
                    origin={200,-50},
                    extent={{-10,-10},{10,10}})),
            Dialog(
                group="I/O",
                tab="Results 1",
                showAs=ShowAs.Result));
    protected
        Boolean HP "HP state on/off" annotation(Dialog(
            group="I/O",
            tab="Results 1",
            showAs=ShowAs.Result));
        Boolean AUX "AUX state on/off" annotation(Dialog(
            group="I/O",
            tab="Results 1",
            showAs=ShowAs.Result));
        Boolean CP "CP state on/off" annotation(Dialog(
            group="I/O",
```

```
            tab="Results 1",
            showAs=ShowAs.Result));
        Boolean SP "SP state on/off" annotation(Dialog(
            group="I/O",
            tab="Results 1",
            showAs=ShowAs.Result));
    public
        Modelica.Blocks.Tables.CombiTable1D FlowTempera-
ture(
            tableOnFile=FlowControl,
            table=[0,0],
            tableName=FlowTable,
            fileName=FlowFile) if FlowControl annotation(
            Placement(transformation(extent={{-10,-
10},{10,10}})),
            Dialog(
                group="I/O",
                tab="Results 1"));
        input Modelica.Blocks.Interfaces.RealInput COP if
COPcontrol "Actual COP of heat pump" annotation(
            Placement(
                transformation(extent={{585,-295},{625,-
255}}),
                iconTransformation(extent={{-220,-20},{-
180,20}})),
            Dialog(
                group="Coefficient of Performance",
                tab="Results 1",
                showAs=ShowAs.Result));
    protected
        Real tHPStart(
            quantity="Basics.Time",
            displayUnit="s") "Starting time of HP" annota-
tion(Dialog(
            group="Timing",
            tab="Results 1",
            showAs=ShowAs.Result));
        Real tBiv=if AuxHeat then tBivalence else 0 "Auxil-
iary variable 1";
```

```
        Real deltaTAuxUp=if AuxHeat then deltaTAuxRefUp
else 0 "Auxiliary variable 2";
        Real deltaTAuxLow=if AuxHeat then deltaTAuxRefLow
else 0 "Auxiliary variable 3";
    public
        parameter Boolean SourceAir=true "If true, heat
pump uses ambient air for heat source" annotation(Dialog(
            group="Heat Pump Type",
            tab="Controller Configuration"));
        parameter Boolean COPcontrol=false "If true,
switches on if COP is greater than a certain threashold
level" annotation(Dialog(
            group="Control Strategy",
            tab="Controller Configuration"));
        parameter Real COPmin=2.5 if COPcontrol "Minimum
COP when heat pump can be used if it is COP-controlled" an-
notation(Dialog(
            group="Control Strategy",
            tab="Controller Configuration"));
        parameter Boolean FlowControl=false "If true, flow
temperature controlled by ambient temperature, else manu-
ally controlled" annotation(Dialog(
            group="Flow Temperature Control",
            tab="Controller Configuration"));
        parameter Boolean AuxHeat=false "If true, auxiliary
heat power supply, else none" annotation(Dialog(
            group="Auxiliary Heat Control",
            tab="Controller Configuration"));
        parameter Real tDelayHP(
            quantity="Basics.Time",
            displayUnit="s")=15 "Delay time between cir-
cultion pump and compressor starting" annotation(Dialog(
            group="Heat Pump",
            tab="Timing Control"));
        parameter Real tHPminOn(
            quantity="Basics.Time",
            displayUnit="s")=24 "Minimum turn-on time of
HP" annotation(Dialog(
            group="Heat Pump",
```

```
            tab="Timing Control"));
        parameter Real tHPminOff(
            quantity="Basics.Time",
            displayUnit="s")=30 "Minimum turn-off time of
HP" annotation(Dialog(
            group="Heat Pump",
            tab="Timing Control"));
        parameter Real tDelayCP(
            quantity="Basics.Time",
            displayUnit="s")=600 "Delay time between HP and
circulation pump shutdown" annotation(Dialog(
            group="Circulation Pump",
            tab="Timing Control"));
        parameter Real tDelayCPComp(
            quantity="Basics.Time",
            displayUnit="s")=600 "Delay time between shut-
down of compressor and circulation pump" annotation(Dialog(
            group="Circulation Pump",
            tab="Timing Control"));
        parameter Real tDelayCPIce(
            quantity="Basics.Time",
            displayUnit="s")=100 "Delay time between shut-
down of compressor and circulation pump during de-icing
process" annotation(Dialog(
            group="Circulation Pump",
            tab="Timing Control"));
        parameter Real tBivalence(
            quantity="Basics.Time",
            displayUnit="s")=3600 if AuxHeat "Delay time
while not reaching reference temperature until auxiliary
heating system starts" annotation(Dialog(
            group="Auxiliary Heating",
            tab="Timing Control"));
        parameter Real tDelaySP(
            quantity="Basics.Time",
            displayUnit="s")=60 "Delay time between source
pump and HP starting" annotation(Dialog(
            group="Source Pump",
            tab="Timing Control"));
```

```
        parameter Real tDelaySPComp(
            quantity="Basics.Time",
            displayUnit="s")=60 "Delay time between shut-
down of compressor and source pump" annotation(Dialog(
            group="Source Pump",
            tab="Timing Control"));
        parameter Real TFlowMax(
            quantity="Thermics.Temp",
            displayUnit="°C")=333.14999999999998 "Maximum
flow temperature" annotation(Dialog(
            group="Boundaries Flow and Return",
            tab="I/O - Control"));
        parameter Real TReturnMax(
            quantity="Thermics.Temp",
            displayUnit="°C")=328.14999999999998 "Maximum
return temperature" annotation(Dialog(
            group="Boundaries Flow and Return",
            tab="I/O - Control"));
        parameter Real deltaTActRefLow(
            quantity="Thermics.TempDiff",
            displayUnit="K")=-5 "Lowmost temperature dif-
ference between actual and reference temperature (<0:
TLow<TRef, >0: TLow>TRef)" annotation(Dialog(
            group="Reference Temperature",
            tab="I/O - Control"));
        parameter Real deltaTActRefUp(
            quantity="Thermics.TempDiff",
            displayUnit="K")=5 "Upmost temperature differ-
ence between actual and reference temperature (<0:
TUp<TRef, >0: TUp>TRef)" annotation(Dialog(
            group="Reference Temperature",
            tab="I/O - Control"));
        parameter Real deltaTAuxRefUp(
            quantity="Thermics.TempDiff",
            displayUnit="K")=3 if AuxHeat "Temperature dif-
ference between actual temperature and reference tempera-
ture for auxiliary heating switch-off" annotation(Dialog(
            group="Auxiliary Heating",
            tab="I/O - Control"));
```

```
        parameter Real deltaTAuxRefLow(
            quantity="Thermics.TempDiff",
            displayUnit="K")=-3 if AuxHeat "Temperature
difference between actual temperature and reference temper-
ature for auxiliary heating switch-on" annotation(Dialog(
            group="Auxiliary Heating",
            tab="I/O - Control"));
        parameter String FlowFile=GreenBuilding.Utili-
ties.Functions.getModelDataDirectory()+"\\heat_pump\\con-
trol\\flow_control.txt" if FlowControl "File name for flow
temperature control by ambient temperature" annotation(Dia-
log(
            group="Flow Temperature",
            tab="flow control"));
        parameter String FlowTable="T_flow_c" if FlowCon-
trol "Table name for flow temperature control by ambient
temperature" annotation(Dialog(
            group="Flow Temperature",
            tab="flow control"));
        parameter Real TAmbientMax(
            quantity="Thermics.Temp",
            displayUnit="°C")=289.14999999999998 if Flow-
Control "Maximum temperature for flow control" annota-
tion(Dialog(
            group="Ambient Temperature",
            tab="flow control"));
        parameter Real TAmbientMin(
            quantity="Thermics.Temp",
            displayUnit="°C")=257.14999999999998 if Flow-
Control "Minimum temperature for flow control" annota-
tion(Dialog(
            group="Ambient Temperature",
            tab="flow control"));
        parameter Real SupportPointsAmbientTemperature=168
if FlowControl "Used support points for temperature averag-
ing" annotation(Dialog(
            group="Ambient Temperature",
            tab="flow control"));
        parameter Real TimeStepAmbientTemperature(
```

```
            quantity="Basics.Time",
            displayUnit="s")=3600 if FlowControl "Used time
step for temperature averaging" annotation(Dialog(
            group="Ambient Temperature",
            tab="flow control"));
        parameter Real qvSource(
            quantity="Thermics.VolumeFlow",
            displayUnit="m³/h")=0.90000000000000002 "Con-
stant velocity of source pump" annotation(Dialog(
            group="Source Pump",
            tab="flow control"));
        parameter Real deltaTFlowRefMax(
            quantity="Thermics.TempDiff",
            displayUnit="K")=3 "Temperature difference
(TFlow-TFlowRef) when qvRef=qvMax" annotation(Dialog(
            group="Boundary Temperatures for Volume Flow
Control",
            tab="Volume Flow Control"));
        parameter Real deltaTFlowRefMin(
            quantity="Thermics.TempDiff",
            displayUnit="K")=-10 "Temperature difference
(TFlow-TFlowRef) when qvRef=qvMin" annotation(Dialog(
            group="Boundary Temperatures for Volume Flow
Control",
            tab="Volume Flow Control"));
        parameter Real qvMax(
            quantity="Thermics.VolumeFlow",
            displayUnit="m³/h")=0.00083333333333333328
"Maximum volume flow of circulation pump" annotation(Dia-
log(
            group="Volume Flow Boundaries",
            tab="Volume Flow Control"));
        parameter Real qvMin(
            quantity="Thermics.VolumeFlow",
            displayUnit="m³/h")=0.00019444444444444443
"Minimum volume flow of circulation pump" annotation(Dia-
log(
            group="Volume Flow Boundaries",
            tab="Volume Flow Control"));
```

```
        parameter Real TDeIcingBound(
            quantity="Thermics.Temp",
            displayUnit="°C")=280.14999999999998 if
SourceAir "Boundary temperature for change between active
and passive de-icing" annotation(Dialog(
            group="Boundary Temperature",
            tab="De-Icing"));
        parameter Real tDeIcing(
            quantity="Basics.Time",
            displayUnit="s")=270 if SourceAir "Duration of
de-icing process" annotation(Dialog(
            group="Timing Control",
            tab="De-Icing"));
        parameter Real tOperation(
            quantity="Basics.Time",
            displayUnit="s")=5400 if SourceAir "Statisti-
cally operation time between two de-icing processes" anno-
tation(Dialog(
            group="Timing Control",
            tab="De-Icing"));
        parameter Real tDeIcingOn(
            quantity="Basics.Time",
            displayUnit="s")=60 if SourceAir "Switch-on
time of de-icing process" annotation(Dialog(
            group="Timing Control",
            tab="De-Icing"));
        parameter Real tDeIcingOff(
            quantity="Basics.Time",
            displayUnit="s")=60 if SourceAir "Shutdown time
of de-icing process" annotation(Dialog(
            group="Timing Control",
            tab="De-Icing"));
    protected
        model AverageTemperature "Calculation of average
ambient temperature"
            GreenBuilding.Interfaces.Ambience.AmbientCondi-
tion AmbientConditions "External Ambient Conditions Con-
nector";
            function arrayChange
```

```
                input Real array[:];
                input Real newValue;
                input Integer sizeArray;
                output Real arrayNew[sizeArray];
                protected
                    Real saveArray[sizeArray];
                algorithm
                    for i in 1:(sizeArray-1) loop
                        saveArray[i]:=array[i+1];
                    end for;
                    saveArray[sizeArray]:=newValue;
                    arrayNew:=saveArray;
            end arrayChange;
            function arrayAverage
                input Real array[:];
                input Integer sizeArray;
                output Real average;
                protected
                    Real save;
                algorithm
                    save:=0;
                    for i in 1:sizeArray loop
                        save:=save+array[i];
                    end for;
                    average:=save/sizeArray;
            end arrayAverage;
            protected
                discrete Real ArrayTemperature[Sup-
portPoints];
                discrete Real TimeStep;
            public
                Real TAverage "Average temperature for time
period";
                parameter Integer SupportPoints=168 "Number
of support points for averaging" annotation(Dialog(
                    group="parameter",
                    tab="temperature averaging"));
                parameter Integer StepTime(
                    quantity="Basics.Time",
```

```
                    displayUnit="s")=3600 "Time step be-
tween each data point" annotation(Dialog(
                    group="parameter",
                    tab="temperature averaging"));
            initial equation
                TimeStep=time.start;
                for i in 1:SupportPoints loop
                    ArrayTemperature[i]=AmbientCondi-
tions.TAmbientAverageAct;
                end for;
                TAverage=arrayAverage(ArrayTemperature,Sup-
portPoints);
            equation
                when (time-TimeStep > StepTime) then
                    TimeStep=time;
                    ArrayTemperature=arrayChange(ArrayTem-
perature,AmbientConditions.TAmbient,SupportPoints);
                    TAverage=arrayAverage(ArrayTempera-
ture,SupportPoints);
                end when;
            annotation(Icon(coordinateSystem(extent={{-
100,50},{100,-50}})));
        end AverageTemperature;
        model DeIcing "De-icing controller V1.0"
            function Count_DEICING
                input Integer Count;
                input Integer Counter;
                output Integer CountMore;
                algorithm
                    CountMore:=Count+Counter;
            end Count_DEICING;
            protected
                discrete Real tHeatStart;
                Boolean IceOn;
                discrete Real tIceStart;
                discrete Real tIceEnd;
                Integer CountIceOn;
            public
                Boolean DEICING "Dontrol signal de-icing";
```

```
                Boolean HEAT "Control signal heating";
                parameter Real tDeIcing(
                    quantity="Basics.Time",
                    displayUnit="s")=270 "Duration of de-
icing process" annotation(Dialog(
                    group="control time periods",
                    tab="DE-ICING"));
                parameter Real tOperateIce(
                    quantity="Basics.Time",
                    displayUnit="s")=5400 "Duration between
two de-icing processes" annotation(Dialog(
                    group="control time periods",
                    tab="DE-ICING"));
                parameter Real tOn(
                    quantity="Basics.Time",
                    displayUnit="s")=60 "Switch-on time of
de-icing process" annotation(Dialog(
                    group="control time periods",
                    tab="DE-ICING"));
                parameter Real tOff(
                    quantity="Basics.Time",
                    displayUnit="s")=60 "Switch-off time of
de-icing process" annotation(Dialog(
                    group="control time periods",
                    tab="DE-ICING"));
            initial equation
                CountIceOn = 0;
                tHeatStart = time.start;
                tIceStart = time.start;
                tIceEnd = time.start;
                IceOn = false;
            equation
                        when not pre(HEAT) then
                            CountIceOn=0;
                        elsewhen not pre(IceOn) then
                            if time>time.start then
                                CountIceOn=Count_DE-
ICING(pre(CountIceOn),1);
                            else
```

```
                                CountIceOn=Count_DE-
ICING(pre(CountIceOn),0);
                            end if;
                        end when;

                        when pre(HEAT) then
                            tHeatStart=time;
                        end when;

                        when pre(IceOn) then
                            tIceStart=time;
                        end when;

                        when not pre(IceOn) then
                            tIceEnd=time;
                        end when;

                        when (((time-tHeatStart) > (tOper-
ateIce/2)) and pre(HEAT) and (not (pre(CountIceOn) > 0)))
then
                            IceOn=true;
                        elsewhen (((time-tIceEnd) > tOper-
ateIce) and pre(HEAT) and (pre(CountIceOn) > 0)) then
                            IceOn=true;
                        elsewhen ((time-tIceStart) >
(tOn+tDeIcing+tOff)) then
                            IceOn=false;
                        elsewhen (not pre(HEAT)) then
                            IceOn=false;
                        end when;

                        DEICING = IceOn;
            annotation(Icon(coordinateSystem(extent={{-
100,50},{100,-50}})));
        end DeIcing;
        model CirculationPump "Circulation pump control
V1.0"
            Boolean HEAT "Heat signal";
            Boolean DEICING(
```

```
                start=false,
                fixed=true) "De-icing signal";
            Boolean CP "Switch-on/off of circulation pump";
            Boolean COMP(
                start=false,
                fixed=true) "Switch-on/off of compressor";
            Real TAmbient "Ambient temperature";
            protected
                discrete Real tEndComp;
                discrete Real tStartIce;
                Boolean CPon;
            public
                parameter Real tCPComp(
                    quantity="Basics.Time",
                    displayUnit="s")=600 "Follow-up time of
circulation pump after compressor operation" annotation(Di-
alog(
                    group="switching and operation time pe-
riod",
                    tab="HEATING"));
                parameter Real tCPIce(
                    quantity="Basics.Time",
                    displayUnit="s")=100 "Follow-up time of
circulation pump while di-icing" annotation(Dialog(
                    group="switching and operation time pe-
riod",
                    tab="DE-ICING"));
                parameter Real TBoundIce(
                    quantity="Basics.Unitless",
                    displayUnit="-")=7 "Boundary tempera-
ture for active/passive de-icing" annotation(Dialog(
                    group="boundary temperature for ac-
tive/passive de-icing",
                    tab="DE-ICING"));
            initial equation
                tEndComp=time.start;
                tStartIce=time.start;
                CPon=HEAT;
            equation
```

```
                when not pre(COMP) and not pre(HEAT) then
                    tEndComp=time;
                end when;

                when pre(DEICING) and (TAmbient <
TBoundIce) then
                    tStartIce=time;
                end when;

                when not pre(HEAT) and (time-tEndComp)>tCP-
Comp then
                    CPon=false;
                elsewhen pre(DEICING) and TAmbi-
ent>=TBoundIce and (time-tStartIce)>tCPIce then
                    CPon=false;
                elsewhen not pre(CPon) and not pre(DEICING)
and pre(HEAT) then
                    CPon=true;
                end when;

                CP=CPon;
            annotation(Icon(coordinateSystem(extent={{-
100,50},{100,-50}})));
        end CirculationPump;
        model SourcePump "Source pump control V1.0"
            protected
                discrete Real tEndComp;
                discrete Real tStartIce;
                Boolean SPon;
            public
                parameter Real tSP(
                    quantity="Basics.Time",
                    displayUnit="s")=60 "Follow-up time of
source pump after compressor operation" annotation(Dialog(
                    group="switching and operation time pe-
riod",
                    tab="HEATING"));
                parameter Real TBoundIce=7 "Boundary tem-
perature for active/passive de-icing" annotation(Dialog(
```

```
                    group="boundary temperature for ac-
tive/passive de-icing",
                    tab="DE-ICING"));
                Boolean HEAT "Heat signal";
                Boolean DEICING(
                    start=false,
                    fixed=true) "De-icing signal";
                Boolean SP "Switch on/off of source pump";
                Real TAmbient "Ambient temperature";
                Boolean COMP(
                    start=false,
                    fixed=true) "Compressor on/off";
            initial equation
                tEndComp=time.start;
                tStartIce=time.start;
                SPon=HEAT;
            equation
                when not pre(COMP) and not pre(HEAT) then
                    tEndComp=time;
                end when;

                when pre(DEICING) and (TAmbient <
TBoundIce) then
                    tStartIce=time;
                end when;

                when pre(HEAT) and not pre(DEICING) then
                    SPon=true;
                elsewhen not pre(HEAT) and (time-tEnd-
Comp)>tSP then
                    SPon=false;
                elsewhen pre(DEICING) and TAmbi-
ent<TBoundIce and (time-tStartIce)>tSP then
                    SPon=false;
                elsewhen not pre(SPon) and not pre(DEICING)
and pre(HEAT) then
                    SPon=true;
                end when;
```

```
                SP=SPon;
            annotation(Icon(coordinateSystem(extent={{-
100,50},{100,-50}})));
        end SourcePump;
        model Compressor "Compressor control V1.0"
            Boolean HEAT "Heat signal";
            Boolean DEICING(
                start=false,
                fixed=true) "De-icing signal";
            Boolean COMP "Switch-on/off of compressor";
            Boolean SP(
                start=false,
                fixed=true) "Switch-on/off of source pump";
            Boolean CP(
                start=false,
                fixed=true) "Switch-on/off of circulation
pump";
            Real TAmbient "Ambient temperature";
            discrete Real tStartComp;
            discrete Real tEndComp;
            discrete Real tStartCP;
            discrete Real tStartSP;
            discrete Real tStartHeat;
            Boolean COMPon;
            parameter Real tMinOff(
                quantity="Basics.Time",
                displayUnit="s")=24 "Minimum turn-off time"
annotation(Dialog(
                group="switching and operation time pe-
riod",
                tab="HEATING"));
            parameter Real tMinOn(
                quantity="Basics.Time",
                displayUnit="s")=30 "Minimum turn-on time"
annotation(Dialog(
                group="switching and operation time pe-
riod",
                tab="HEATING"));
            parameter Real tCompCP(
```

```
                    quantity="Basics.Time",
                    displayUnit="s")=15 "Delay time compared to
circultion pump" annotation(Dialog(
                    group="switching and operation time pe-
riod",
                    tab="HEATING"));
               parameter Real tCompSP(
                    quantity="Basics.Time",
                    displayUnit="s")=60 "Delay time compared to
source pump" annotation(Dialog(
                    group="switching and operation time pe-
riod",
                    tab="HEATING"));
               parameter Real TBoundIce=7 "Boundary tempera-
ture for active/passive de-icing" annotation(Dialog(
                    group="boundary temperature for active/pas-
sive de-icing",
                    tab="DE-ICING"));
               initial equation
                    if HEAT then
                         tStartHeat = time.start-
max(tCompCP,tCompSP)-1;
                         tStartCP = time.start-tCompCP-1;
                         tStartSP = time.start-tCompSP-1;
                         tStartComp = time.start-tMinOn-1;
                         tEndComp = time.start;
                    else
                         tStartHeat = time.start;
                         tStartCP = time.start;
                         tStartSP = time.start;
                         tStartComp = time.start;
                         tEndComp = time.start-tMinOff-1;
                    end if;
                    COMPon = HEAT;
               equation
                    when pre(HEAT) then
                         tStartHeat=time;
                    end when;
```

```
                when pre(CP) then
                    tStartCP=time;
                end when;

                when pre(SP) then
                    tStartSP=time;
                end when;

                when pre(COMPon) then
                    tStartComp=time;
                end when;

                when not pre(COMPon) then
                    tEndComp=time;
                end when;

                when (time-tStartCP)>tCompCP and (time-
tStartSP)>tCompSP and pre(HEAT) and (time-tEndComp)>tMinOff
and not pre(DEICING) then
                    COMPon=true;
                elsewhen (time-tStartComp)>tMinOn and not
pre(HEAT) then
                    COMPon=false;
                elsewhen pre(DEICING) and TAmbi-
ent>=TBoundIce then
                    COMPon=false;
                elsewhen not pre(DEICING) then
                    COMPon=true;
                end when;

                COMP=COMPon;
            annotation(Icon(coordinateSystem(extent={{-
100,50},{100,-50}})));
        end Compressor;
    initial equation
        assert(deltaTFlowRefMax>deltaTFlowRefMin,"qvRef not
computable");
        if (AuxHeat) then
```

```
            assert(deltaTAuxRefUp<deltaTActRefUp, "Tempera-
ture regulation limits must be within system control lim-
its");
        end if;

        tHPStart = time.start;

        if (((TAct-TRef)<deltaTActRefUp) and Enable) then
            CPcontrol.HEAT=(not COPcontrol or (COPcontrol
and (COP>COPmin)));
            SPcontrol.HEAT=(not COPcontrol or (COPcontrol
and (COP>COPmin)));
            CompControl.HEAT=(not COPcontrol or (COPcontrol
and (COP>COPmin)));
            DEICING.HEAT=(not COPcontrol or (COPcontrol and
(COP>COPmin))) and SourceAir;
        else
            CPcontrol.HEAT=false;
            SPcontrol.HEAT=false;
            CompControl.HEAT=false;
            if SourceAir then
                DEICING.HEAT=false;
            else
                DEICING.HEAT=true;
            end if;
        end if;

        AUX = (((TAct-TRef)<deltaTActRefUp) and Enable) and
(not COPcontrol or (COPcontrol and (COP>COPmin))) and (Aux-
Heat and (TAct-TRef)<deltaTAuxLow);
    equation
                    EPricing.u[1] = AmbientCondi-
tions.HourOfDay;
                               if ( EPriceAverage > EPric-
ing.y[1] ) then
                                              Price = true;
                    else
                                       Price=false;
                        end if;
```

```
        if FlowControl then
            connect(AmbientConditions,TAverage.AmbientCon-
ditions);
        end if;

        if FlowControl then
            FlowTemperature.u[1]=TAverage.TAverage;
        end if;

        if FlowControl then
            TFlowControl=FlowTemperature.y[1];
        else
            TFlowControl=TFlowRef;
        end if;

        when pre(CompControl.HEAT) then
            tHPStart=time;
        end when;

        when (((TAct-TRef)<deltaTActRefLow) and Enable and
(not COPcontrol or (COPcontrol and (COP>COPmin))))  then
            CPcontrol.HEAT=true;
            SPcontrol.HEAT=true;
            CompControl.HEAT=true;
            if SourceAir then
                DEICING.HEAT=true;
            end if;
        elsewhen (((TAct-TRef)>deltaTActRefUp) or
(TFlow>TFlowMax) or (TReturn>TReturnMax) or not Enable or
(COPcontrol and (COP<=COPmin))) then
            CPcontrol.HEAT=false;
            SPcontrol.HEAT=false;
            CompControl.HEAT=false;
            if SourceAir then
                DEICING.HEAT=false;
            else
                DEICING.HEAT=true;
            end if;
```

```
        end when;

        when pre(CompControl.HEAT) and Enable and AuxHeat
and (time-tHPStart)>tBiv and (TAct-TRef)<deltaTAuxLow then
            AUX=true;
        elsewhen (not Enable or not pre(CompControl.HEAT)
or ((TAct-TRef)>deltaTAuxUp)) and pre(AUX) then
            AUX=false;
        end when;

        if SourceAir then
            CPcontrol.DEICING=DEICING.DEICING;
            SPcontrol.DEICING=DEICING.DEICING;
            CompControl.DEICING=DEICING.DEICING;
        else
            CPcontrol.DEICING=false;
            SPcontrol.DEICING=false;
            CompControl.DEICING=false;
        end if;

        if SourceAir then
            CPcontrol.TAmbient=AmbientConditions.TAmbient;
            SPcontrol.TAmbient=AmbientConditions.TAmbient;
            CompControl.TAmbient=AmbientConditions.TAmbi-
ent;
        else
            CPcontrol.TAmbient=280.15;
            SPcontrol.TAmbient=280.15;
            CompControl.TAmbient=280.15;
        end if;

        CPcontrol.COMP=CompControl.COMP;
        SPcontrol.COMP=CompControl.COMP;

        CompControl.CP=CPcontrol.CP;
        CompControl.SP=SPcontrol.SP;

        if Enable or Price then
            HP=CompControl.COMP;
```

```
            CP=CPcontrol.CP;
            SP=SPcontrol.SP;
        else
            HP=false;
            CP=false;
            SP=false;
        end if;

        if SP  then
            qvSourceRef=qvSource;
        else
            qvSourceRef=0;
        end if;

        if CP  then
            if HP then
                qvRef=max(min((qvMin+(qvMax-
qvMin)*(((TFlow-(TFlowControl+deltaTFlowRefMin))/(deltaT-
FlowRefMax-deltaTFlowRefMin)))),qvMax),qvMin);
            else
                qvRef=qvMin;
            end if;
        else
            qvRef=0;
        end if;

        HPon=HP;

        if SourceAir then
            DEICINGon=DEICING.DEICING;
        end if;

        CPon=CP;
        SPon=SP;

        if (AuxHeat) then
            AUXon=AUX;
        end if
```