

Maximisation of self-consumption of PV power

Pol Boudou Perez

October 2019

1 Introduction

2 Problem statement

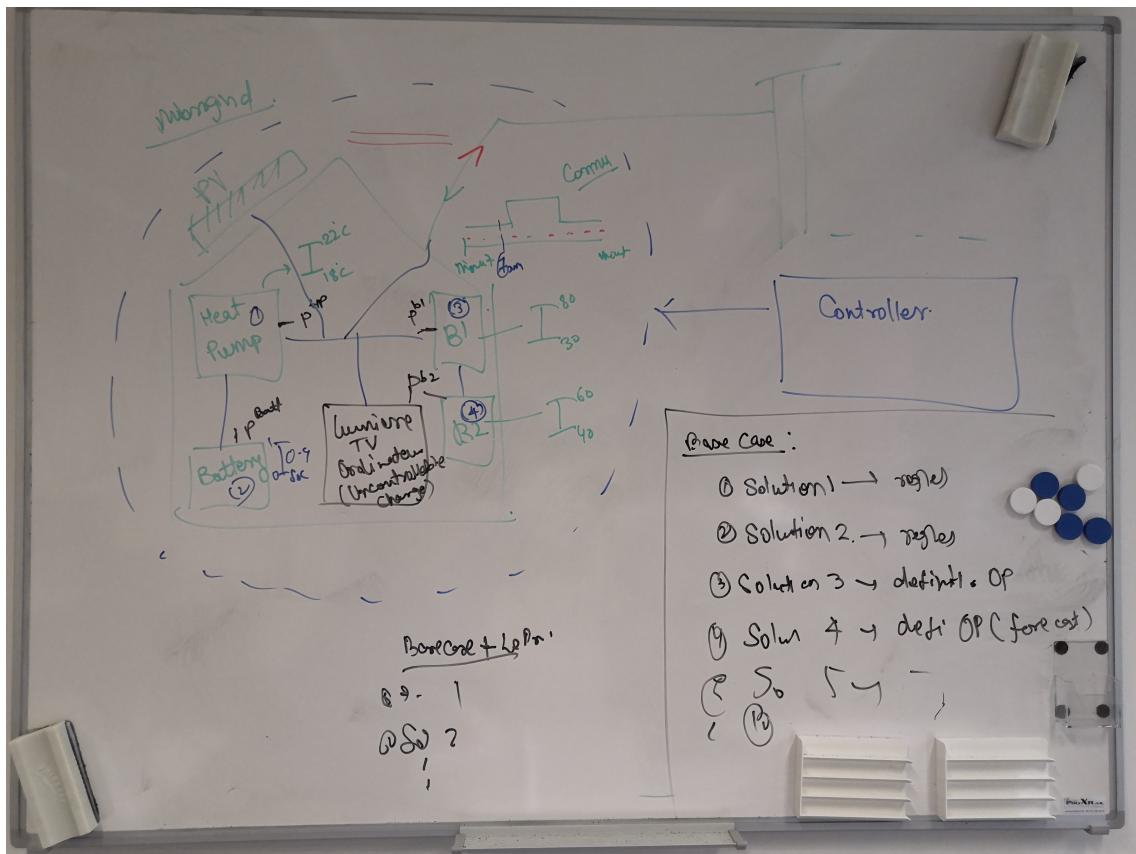


Figure 1: Microgrid sketch (to be changed afterwards)

2.1 System description

The object of the study is a microgrid connected to the distribution grid. In the present case, we consider a simple microgrid that can be seen as a domestic household equipped with a PV system. Because of its power generation capacity, the microgrid can inject excess power into the grid via a single two-ways connection.

As for the energy consumption of the household, this one can be divided in two well defined classes: controllable and non-controllable loads. The latter describes all appliances that are human-switched (TV, lights, coffee machine, etc.). The former makes reference to those flexible loads that are going to be controlled in order to consume whenever there is PV production. For instance, to ensure thermal comfort, the household is equipped with two boilers and a heat pump (HP). The storage capacity from those thermo-electric units allows operating them with certain freedom. Hence they can be employed under certain coordination, to actively respond to the power system fluctuations. Similarly, an energy storage battery system brings more flexibility to the system.

The objective of the energy management strategy is to control the flexible loads in order to reduce electricity costs by maximising self-consumption of PV power.

2.2 Modelling of the various components and syntax

Control variables are named u , whereas measurement variables will be named p . Capital letters will designate constant parameters. Indexes will refer to specific energy systems. Time value of the variables will be noted $[h]$, where h designates the current discrete time period, and goes from 0 to H .

PV system

p_{PV} : PV power (kW) (positive variable)

Energy storage system

u_b : Battery target power (kW). Positive if battery is charging, negative if discharging.

p_b : Measured battery power. Same sign as u_b

x_b : Battery state of charge (kWh)

\bar{C}_b : Max. battery capacity (kWh)

\underline{C}_b : Min. battery capacity (kWh)

\bar{P}_b : Max. charging power of the battery (kW)

\underline{P}_b : Max. discharging power of the battery (kW)

Non-controllable loads

p_{nc} : power demanded by non-controllable loads (kW) (positive)

Thermo-electric loads

$u_{B,k}$: target power of electric boiler k (kW)

$p_{B,k}$: power input of electric boiler k (kW)

$\bar{P}_{B,k}$: Max. power of electric boiler k (kW)

u_{HP} : target power of electric heat pump (kW)

p_{HP} : power input of electric heat pump (kW)

\bar{P}_{HP} : Max. power of electric heat pump (kW)

Building comfort constraint must be respected and indoor temperature $T_{in}[h]$ must remain within a band around temperature bounds $(\underline{T}_{in}, \bar{T}_{in})$.

As for the boilers, flexibility is again achieved by imposing upper and lower limits to the hot-water temperature $(\underline{T}_{B,k}, \bar{T}_{B,k})$.

3 Control strategy

3.1 Premise

The microgrid automatically balances power supply and demand. We assume any PV generated power not consumed by the building is absorbed by the grid without the necessity of any control action. Similarly, any power demand from loads that cannot be supplied by the PV system is supplied by the grid.

3.2 Strategy 1

For the moment, such strategy does not look at grid's electricity price.

This being the first and most basic strategy, it does not demand models for thermo-electric units. In a straightforward manner, the charging of those flexible loads is done whenever temperature approaches lower limits. Defining a certain tolerance ϵ_{T_k} for each of the thermo-electric units will ensure (not in a robust manner) that such limits are not attained. Measurements of power inputs p_{B1} , p_{B2} and p_{HP} are therefore not needed in this approach, and temperatures are the only variables used to determine the adequate control action.

Let's define the number of thermo-electric units as N_c , which equals 3 in the presented system. In order to simplify the algorithm syntax, vectors will be used to express variables representing those controllable loads. For instance, $\hat{p}_c[h]$ includes input power of all N_c loads, and $\hat{x}^c[h]$ regroups their state. From a controller device perspective, it is assumed that a pre-processing unit is handling thermal sensors outputs and normalizing them according to upper and lower limits of each load. In such a way, all $\hat{x}^c[h]$ elements are bonded between 0 and 1.

3.2.1 Summary

The controller is configured by establishing rules determined by all possible system operating scenarios.

For feeding uncontrollable loads p_{nc} , PV power is the prior source, followed by battery storage and finally the main grid.

The variable $p_x = p_{PV} - p_{nc}$ refers to the excess PV power in the microgrid available to feed flexible loads. The dealing of such excess power follows a set of priority rules:

1. If around lower limits, thermo-electric units are prioritized.
2. Battery charging has priority until a certain SOC threshold is reached (80% for example). A binary variable $s_b[h]$ decides if battery has priority over thermal loads ($s_b[h] = 1$) or if thermal loads should be supplied before battery ($s_b[h] = 0$).
3. When loading thermo-electric units, those with the lower energy levels (normalized) are prioritized.

When excess power is negative, charging such controllable loads demands electricity from the main grid.

3.2.2 Algorithm

Algorithm 1: Strategy 1

Executed every time-step h
 Inputs: $x_b[h], s_b[h], s_b[h - 1], \hat{x}^c[h], p_{PV}[h], p_{nc}[h]$
 Control variables: u_b, \hat{u}^c
Initialize: $u_b \leftarrow 0, \hat{u}^c \leftarrow 0$

```

if  $x_b[h] > 80\%SOC$  then
  |  $s_b[h] = 0$  (battery has not priority)
end
if  $x_b[h] \leq 70\%SOC$  then
  |  $s_b[h] = 1$  (battery has priority)
else
  |  $s_b[h] = s_b[h - 1]$ 
end
 $p_x = p_{PV}[h] - p_{nc}[h]$ 
call LowerConstraintHandling( $p_x, \hat{x}^c[h]$ )
call TreatExcessPower( $p_x, x_b[h], \hat{x}^c[h]$ )
update Control variables
end
.
```

Algorithm 2: Lower constraint handling

Control variables: $\hat{u}^c[h]$
function *LowerConstraintHandling* ($p_x, \hat{x}^c[h]$) :
for each energy system k state in $\hat{x}^c[h]$ ($B1, B2, HP$) **do**

```

if  $x_k[h] \in [0; \epsilon_k^T]$  then
  | if  $p_x > 0$  then
    | |  $u_k \leftarrow \min(p_x, \bar{P}_k)$ 
    | | if  $(p_x > p_k)$  then
      | | |  $p_x = p_x - \bar{P}_k$ 
    | | end
    | | else
    | | |  $u_k \leftarrow \bar{P}_k$ 
    | | end
  | end
end
.
```

Algorithm 3: Treatment of microgrid's excess power

Control variables: u_b, \hat{u}^c

function *TreatExcessPower* ($p_x, x_b[h], \hat{x}^c[h]$) :

$diff = p_x - p_b[h]$

if $p_x > 0$ **then**

if ($x_b[h] < 80\%SOC$) **and** ($s_b[h] == 1$) **then**

$u_b \leftarrow \min(u_b[h-1] + diff, \bar{P}_b, \frac{1}{dt}(\bar{C}_b - x_b[h]))$
 (unconstrained, power constrained, energy constrained)

$p_x = p_x - u_b$

if $p_x > 0$ **then**

 | call *PriorityChargingFlexibleLoads*($p_x, \hat{x}^c[h]$)

 | **break**

end

else

 call *PriorityChargingFlexibleLoads*($p_x, \hat{x}^c[h]$)

if returned $p_x > 0$ **then**

 | $u_b \leftarrow \min(u_b[h-1] + diff, \bar{P}_b, \frac{1}{dt}(\bar{C}_b - x_b[h]))$

end

end

end

else

$u_b \leftarrow \max(u_b[h-1] + diff, \underline{P}_b, \frac{1}{dt}(\bar{C}_b - x_b[h]))$
 (unconstrained, power constrained, energy constrained)

end

Algorithm 4: Controllable loads priority rules

Control variables: \hat{u}^c

function *PriorityChargingFlexibleLoads* ($p_x, \hat{x}^c[h]$):

 sort $\hat{x}^c[h]$ in increasing order

for k in $\hat{x}^c[h]$ **do**

if $x_k^c \in [1 - \epsilon_k^T; 1]$ **then**

 | return(p_x)

end

$p_k \leftarrow \min(p_x, \bar{P}_k)$

$p_x = p_x - p_k$

if $p_x \leq 0$ **then**

 | return(p_x)

end

end