Electronics Laboratory STI/ELAB
Ecole Polytechnique Fédérale de Lausanne

ELAB semester project

MPC-based Energy Management System

Student: Nicolas Sierro
Supervisors: Pr. Maher Kayal, PhD student Olivier Van Cutsem

Contents

Abstract

Energy management means provide energy to the consumers while respecting the demand and optimizing the production cost. Supplying power has become more and more complex in a growing network. Several ways exist to adapt to the steadily increasing energy demand. The energy efficiency of the buildings/generators can be improved or new electric factories can be built. However, smarter techniques of managing the grid from the demand side receive an increasing attention by research and industry for a promising economic potential. This report aims to analyze a specific solution to manage the grid energy demand by adapting the power consumption from a smart building side and reduce electricity costs.

1

Introduction

The classical grid system is unidirectional and operates from the utility to the consumer[1]. Contractors usually bill a fixed price to their customers at the end of the month. A limited number of generators try to constantly supply enough energy to respect the demand. During a typical day, two peak consumption occur around lunch time and in the evening. These rush hours are difficult to manage for the grid which must mobilize more resources to satisfy the energy consumed by the households. Therefore, the utility cost is higher which impacts on their customers bill. This issue becomes even more complex with the volatility of renewable energy [2].

The research nowadays turns toward the demand side with an intelligent way to influence the loads. Indeed, the trend tends to the reduction and management of the loads instead of the construction of new generators. Demand side management allows to optimize the human comfort and health in the building while using the grid efficiently and inexpensively. To achieve this purpose, several techniques exist (see figure 1.1). Every strategy applied in a large scale of buildings has a significant impact on the grid and allow to modify the supply/demand balance.

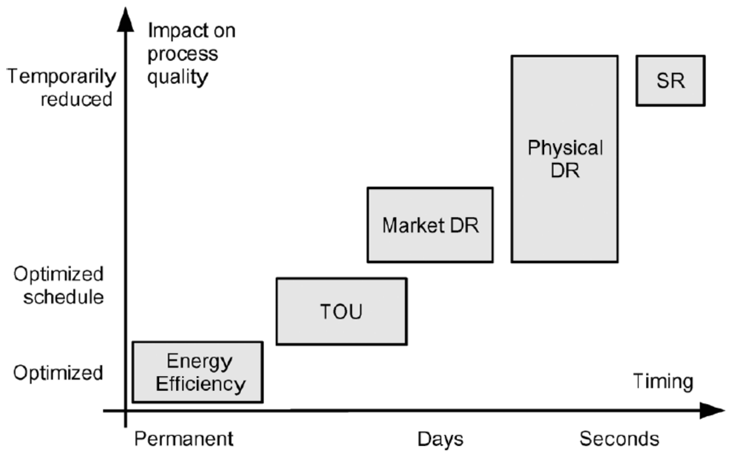


Figure 1.1: Demand side management strategies [1]

The x-axis represents the time to activate the strategy whereas the y-axis evidences the impact on the household comfort. Energy Efficiency improvement is permanent in a building and is achieved by installing better isolation or less high energy devices. Time of Use (TOU) involves a change in the contract/tariffs to penalized consumption during certain hours. The Demand Response (DR) influences the grid less than one day in advance and is therefore more flexible. It exists two types of DR: Market based or Physical/Incentive based. The DR based on the market consists of transmitting the price to the consumers who will adapt their consumption and use the energy when the price is favourable. The consumption pattern is usually planned one day ahead according to the electricity price forecasts. Concerning the Physical/Incentive DR, the utility sends shedding requests to the households. This method is used in prevention of shortage or grid failure and is planned a few minutes ahead. In return, the households receive an economic benefit for the comfort inconvenience. The last category is the Spinning Reserve (SR). When the frequency drops in the grid, it means the energy supply is too low compare to the demand. SR devices measure this change in frequency and consequently adapt the load consumption in the house. A large-scale communication protocol is necessary to coordinate which households shed their loads.

DR is the strategy of interest for this project. This involves the installation of a more enhance equipment such as smart meters[3]. These devices allow a two-ways/real-time communication between the grid and the households to transmit the requests, household consumption or electricity tariffs. Many DR program are ongoing on the European scale and in the US. Significant economic savings were proven thanks to peak demand flattening that reduces the energy costs. Lack of interoperability between smart grid elements and technological immaturity of certain smart grid components represent the most common technical obstacle for the DR technologies.

This study assumes that the DR equipment is available in the household. Electricity tariffs forecast is transmitted from the grid one day ahead. According to this prediction, a price based DR strategy is carried out for one single building. For a large scale, if many households consume energy when the price is low, the DR purpose is achieved and the power peak consumption flatten.

This project aims to maximize the temperature comfort in the building using a Model Predictive Control (MPC) technique while minimizing the electricity bill. Batteries and PV panels are accounted in the energy management strategy.

2
Smart Building Solution

A building containing many rooms with electrical heaters is considered (see figure 2.1). The building uses one or several batteries to store energy and a PV panel generates electricity along the day. To develop a DR strategy minimizing the household bill, the electricity price, temperature and irradiance predictions are required. These variables are available through the smart meters or the weather forecast. A basic energy management consists of preheating the room when the price is low, charging the battery with the energy generated by the PV panel and use the energy stored when the price is high.

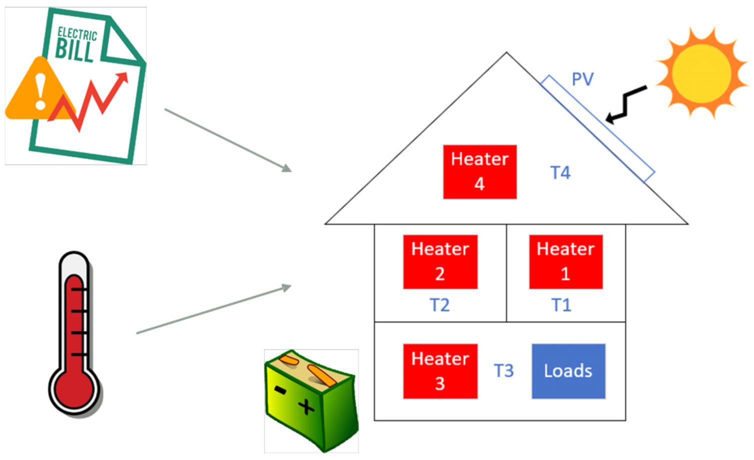


Figure 2.1: Household features

It can be argued that electrical heaters are obsolete and not used in households anymore. However, they are implemented in this project framework because they are power varying loads that maintain a certain comfort inside the house. Generalizing these types of load, heat pumps, air conditioner, ventilator or electrical water boiler can operate in the house with the same DR strategy. Moreover, electrical heaters or water boilers are still installed in isolated building where it is difficult to access to set up a gas or fuel equipment. Finally, with the emergence of clean energy and the demand side management, it is still impossible to predict whether electricity will become more attractive than the other fossil energy in term of heaters. To simplify the development, the building is simulated with a python program. The simulation must work for any kind of building topology.

In this report, first the building model is presented followed by the temperature and battery control strategy. Then, the simulation is introduced. Finally, the results and future steps are discussed.

3

Building Model

The building contains three entities that need to be modeled: The building structure/thermal model, the battery and the PV panel.

3.1 Thermal model:

3.1.1 Thermal equations

The building is characterized by a thermal model according to the conduction and convection equations. All the computations are derived for a single room, considering a homogeneous wall. The surface S groups every piece of walls surrounding the room.

Figure 3.1 shows the physic constants of the wall

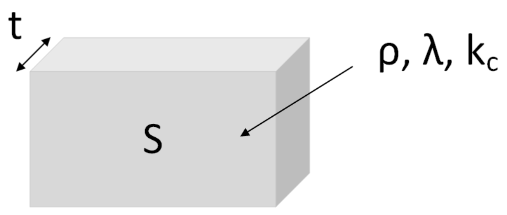


Figure 3.1: Wall properties

t = thickness, S = Surface, ρ = density, λ = thermal conductivity, k_c = convection coefficient

The conduction (3.1) represents the heat transfer inside the wall.

$$\frac{d^2T(x,t)}{dx^2} = \frac{1}{a}\frac{dT(x,t)}{dt} \tag{3.1}$$

T = Temperature inside the wall, a = conduction coefficient, x = position perpendicular to the surface of the wall, t=time

The convection (3.2) represents the heat exchange between the air (either the outside or the room) and the surface of the wall.

$$\phi_{conv} = k_c S(T - T_S) \tag{3.2}$$

ϕ_{conv} = convection flux, T = Inside or outside temperature, T_S = surface temperate

3.1.2 RC circuit

Equations (3.1) and (3.2) are used to define a RC circuit describing the whole system. A thermal resistor R_{th} limits the power flowing inside/outside the wall and a thermal capacitor C emulates a delay for the heat to be transfered inside or outside. A parallel between a thermal model and an electrical model can be deduced:

$T \iff U$

$\phi \iff P$

$R_{th} \iff R_{elec}$

If the wall has a thickness t and a homogeneous layer, the conduction equation can be simplified (3.3).

$$\frac{T(x+\Delta x/2,t) - T(x,t)}{R_{cond}/2} - \frac{T(x,t) - T(x-\Delta x/2,t)}{R_{cond}/2} \simeq C_w \frac{dT}{dt} \tag{3.3}$$

where

$$R_{cond} = \frac{t}{\lambda S} \;\; and \;\; C_w = \rho\, c_m\, t\, S \tag{3.4}$$

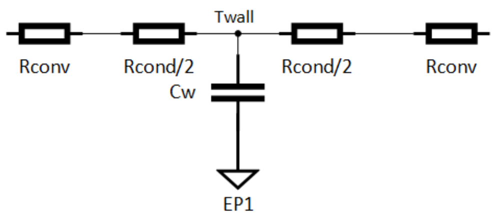
With c_m the specific heat of the wall

From (3.1) the convection thermal resistors is derived (3.5).

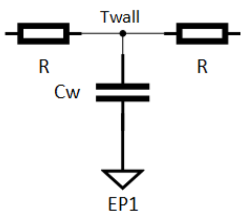
$$R_{conv} = \frac{1}{k_c S} \tag{3.5}$$

From (3.3), a RC circuit can be draw with two resistors and a capacitor in the middle. The convection resistors are added in the two extremities (figure 3.2a). This circuit is simplified by summing the convection and conduction resistors (figure 3.2b).

The temperature do not vary immediately inside the room which is therefore modelled by a capacitor Cr (3.6).



(a) RC wall



(b) RC wall simplified

Figure 3.2: RC circuit of the wall

$$C_r = \rho_{air} V_{room} c_m^{air} \tag{3.6}$$

With c_m the specific heat of the air.

The heater is added to the model (figure 3.3). The power is immediately injected inside the room.

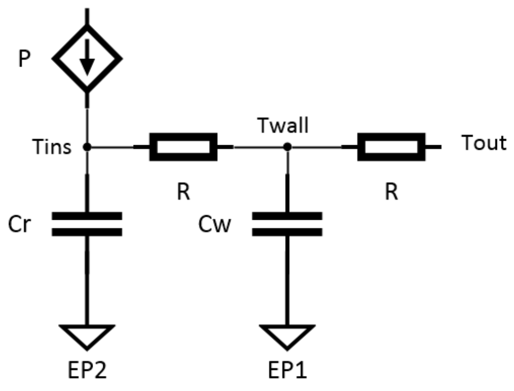


Figure 3.3: Complete RC model

Equation 3.3 was a rough approximation, for further steps the wall can be modeled as figure 3.4 which leads to more complex equations.

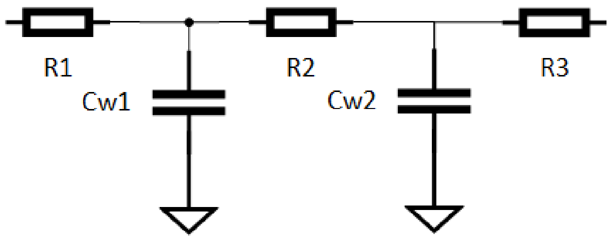


Figure 3.4: Complex RC model

For this project, the model with one wall capacitor is used to remain simple. From the RC circuit (figure 3.3), the differential equations (3.7, 3.8) are derived.

$$C_r \frac{dT_{ins}}{dt} = P_H + \frac{T_w - T_{ins}}{R} \tag{3.7}$$

$$C_w \frac{dT_w}{dt} = \frac{T_{out} + T_{ins} - 2T_w}{R} \tag{3.8}$$

Since a numerical simulation do not work for a continuous model, the system must be discretized. Two different discretization methods are presented.

3.1.3 Euler forward discretization

The derivative is simplified using the value of the next temperature sample (3.9).

$$\frac{dT}{dt} \simeq \frac{T(t+1) - T(t)}{\delta t} \tag{3.9}$$

Applying (3.9) to (3.7) and (3.8) gives (3.10) and (3.11).

$$C_r \frac{T_{ins}(t+1) - T_{ins}(t)}{\delta t} = P_H(t) + \frac{T_w(t) - T_{ins}(t)}{R} \tag{3.10}$$

$$C_w \frac{T_w(t+1) - T_w(t)}{\delta t} = \frac{T_{out}(t) + T_{ins}(t) - 2T_w(t)}{R} \tag{3.11}$$

With this derivative approximation, the sample time δt must be sufficiently small. Otherwise, the differential equation solution diverges.

The system is described as a matrix equation (3.12) and matrix A_d, $B1_d$, $B2_d$ are defined (3.13).

T_{ins} and T_w are called states variables of the system.

$$\begin{pmatrix} T_{ins}(t+1) \\ T_w(t+1) \end{pmatrix} = \begin{pmatrix} 1 - \frac{\delta t}{RC_r} & \frac{\delta t}{RC_r} \\ \frac{\delta t}{RC_w} & 1 - \frac{2\delta t}{RC_w} \end{pmatrix} \begin{pmatrix} T_{ins}(t) \\ T_w(t) \end{pmatrix} + \begin{pmatrix} \frac{\delta t}{C_r} \\ 0 \end{pmatrix} P_H(t) + \begin{pmatrix} 0 \\ \frac{\delta t}{C_w R} \end{pmatrix} T_{out}(t) \tag{3.12}$$

$$\vec{T}(t+1) = A_d\,\vec{T}(t) + B1_d\,P_H(t) + B2_d\,T_{out}(t) \tag{3.13}$$

3.1.4 Exact descretization

(3.7) and (3.8) are written in the continuous matrix form (3.14). A_c, B_c are defined (3.15)

$$\begin{pmatrix} \frac{dT_{ins}}{dt} \\ \frac{dT_w}{dt} \end{pmatrix} = \begin{pmatrix} -\frac{1}{RC_r} & \frac{1}{RC_r} \\ \frac{1}{RC_w} & -\frac{2}{RC_w} \end{pmatrix} \begin{pmatrix} T_{ins} \\ T_w \end{pmatrix} + \begin{pmatrix} \frac{1}{C_r} & 0 \\ 0 & \frac{1}{C_w R} \end{pmatrix} \begin{pmatrix} P_H \\ T_{ext} \end{pmatrix} \tag{3.14}$$

$$A_c = \begin{pmatrix} \frac{1}{RC_r} & \frac{1}{RC_r} \\ \frac{1}{RC_w} & -\frac{2}{RC_w} \end{pmatrix} \; and \; B_c = \begin{pmatrix} \frac{1}{C_r} & 0 \\ 0 & \frac{1}{C_w R} \end{pmatrix} \tag{3.15}$$

This method starts from the state space equation (3.16) and the generalized solution (3.17)[4]. In control theory, x = the states variables and u = the inputs.

$$\dot{x} = A_c x + B_c u \tag{3.16}$$

$$x(t) = e^{A_c(t-t_0)} x(t_0) + \int_{t_0}^{t} e^{A_c(t-\tau)} B_c u(\tau) d\tau \tag{3.17}$$

with $x \iff T$ and $u \iff \begin{pmatrix} P_H \\ T_{ext} \end{pmatrix}$

Then an A/D converter (3.18) and D/A converter/zero order hold (3.19) is plugged in (3.17) which results in (3.20)

$$t_0 = kh \; and \; t = kh + h \tag{3.18}$$

$$u(\tau) = u(kh) \; for \; kh < \tau < kh + h \tag{3.19}$$

$$x(kh + h) = e^{A_c h} x(kh) + \int_{kh}^{kh+h} e^{A_c(kh+h-\tau)} B_c u(kh) d\tau \tag{3.20}$$

With h being the sampling time.

By solving (3.20) and considering $t = kh$ to simplify the notation, the discrete state space is obtained (3.21) and the descrete matrices A_d and B_d.

$$x(t+1) = \underbrace{[e^{A_c h}]}_{A_d} x(t) + \underbrace{[\int_0^h e^{A_c \eta} d\eta \; B_c]}_{B_d} u(t) \tag{3.21}$$

Then (3.21) is applied to the thermal system (3.22).

$$\vec{T}(t+1) = A_d * \vec{T}(t) + B_d * \begin{pmatrix} P_H \\ T_{out} \end{pmatrix} \tag{3.22}$$

This method has the advantage to give the exact solution for any sample time. Matlab or Python contain libraries which perform the discretization without going through all the computation. In python, control.c2d(A,B,C,D,dt) returns the matrices in discrete time. However, the computation cost is higher with the exact descretization because a matrix exponential and an integral must be solved.

With the system in discrete form (3.22), all the information to simulate the model of the room is available.

3.2 Battery model

The maximum power that can be retrieved from a battery is limited and depends on the state of charge (SOC). The curve representing this maximum is highly non-linear and is estimated by two straights. To obtain this approximation, the battery power is sampled for different SOC.

When the battery is controlled, it is important to make sure that the battery set point is always below the maximum. Moreover, to avoid the battery to lose its capacity prematurely, the state of charge is restricted to an area above a minimum and below a maximum. Figure 3.5 represents the estimation of the power curve for the charge (positive values) and discharge (negative values). The green area highlights the power allowed for different SOC. The y-axis scale is highly non-linear for readable purpose.

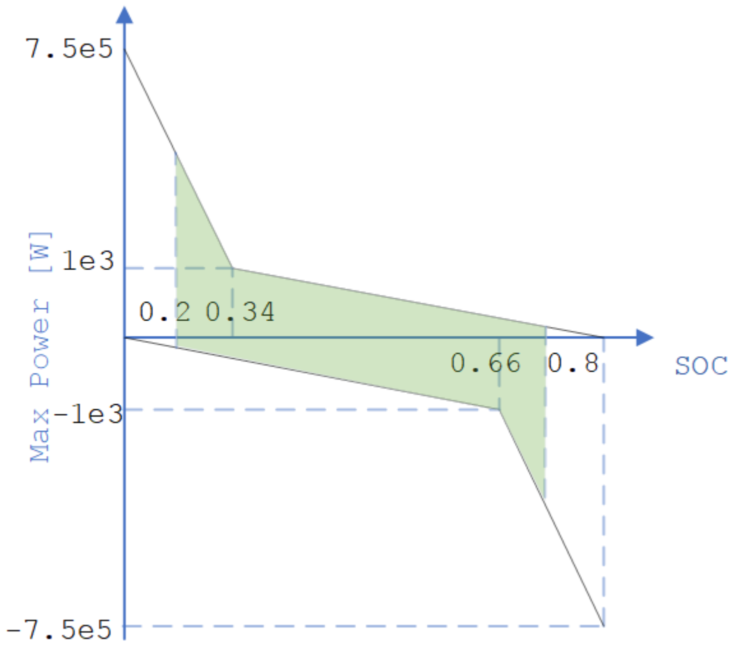


Figure 3.5: Battery model

The energy of the battery E_B and SOC vary according to (3.23) and (3.24)

$$E_B(t+1) = \alpha\, \delta t\, E(t) + \beta\, \delta t\, P_B(t) \tag{3.23}$$

$$SOC(t) = \frac{E_B(t)}{E_{Bmax}} \tag{3.24}$$

α = continuous lost, β = efficiency, dt = sample time, P_B = battery power, E_{max} = energy in the battery when fully charged.

3.3 Generation model

The PV panels are modelled according to their characteristics[5] and assuming a MPPT (Maximum Power Point Tracking). The power production depends on the cell temperature. Cells vary in temperature not only because ambient temperatures change, but also because insolation on the cell changes. To help designers account for changes in cell performance with temperature, it is standard practice for manufacturers to provide an indicator called the nominal operating cell temperature ($NOCT$). The NOCT is the expected cell temperature in a module when ambient temperature is $20°C$ (T_{ref}) and solar irradiation is $1kW/m^2$ (irr_{ref}). The cell temperature formula (3.25) is proportional to the solar irradiance (irr).

$$T_{cell} = T_{amb} + (NOCT - 20)\frac{irr}{irr_{ref}} \tag{3.25}$$

The MPPT potential (3.27) and current (3.29) are derived accounting the thermic losses (3.26, 3.28).

$$V_{loss} = V_{STC}\, \alpha_V (NOTC - T_{ref}) \tag{3.26}$$

With α_V the temperature coefficient of voltage and T_{ref} the reference temperature ($25°C$).

$$V_{MPP} = V_{STC} - V_{loss} \tag{3.27}$$

V_{STC} being the MPPT cell output voltage in standard condition.

$$I_{loss} = I_{STC}\frac{irr}{irr_{ref}}\alpha_I (NOTC - T_{ref}) \tag{3.28}$$

With α_I the temperature coefficient of current.

$$I_{MPP} = I_{STC} \frac{irr}{irr_{ref}} - I_{loss} \tag{3.29}$$

I_{STC} being the MPPT cell output current in standard condition.

Table 3.1 shows typical values of a PV panel.

NOTC	47°C
V_{STC}	34.478 V
I_{STC}	4.65 A
Temperature coeff voltage	0.0046 V/°C
Temperature coeff current	0.0024 A/°C
Nb cells	16

Table 3.1: PV panel characteristics

The power generated is computed accounting the number of cells (3.30).

$$P_G = nb_{cells} * I_{MPP} * V_{MPP} \tag{3.30}$$

4 Energy Management Strategy

As mentioned before, the goal of this project is to find the best solution to minimize the electricity bill while respecting a certain comfort constraint (Inside temperature restricted between two boundaries). In this section, two common controllers regulating temperatures nowadays are described (Thermostat and PI). Those two regulators are associated with a simple rule base control for the battery.

Then, the optimal controller for this study is presented. The performance will be compare with the two first solutions afterwards. For now, a building containing one room and one battery is considered.

4.1 Thermostat controller

A Thermostat is the simplest and most common controller in the households nowadays. The heater is turned on using the maximum power once the temperature reached a minimum and is turned off once the temperature is above a maximum. The battery is used to maximize the self-consumption inside the house.

An example of the temperature regulation with a thermostat is shown in figure 4.1. The temperature is constrained between two bonds.

The battery logic is shown in figure 4.2. If the power generated by the solar panel is greater than the power needed to heat the house, the surplus is stored in the battery. Hence, no power is sold to the utility. The heater consumes the battery energy when the power required is above the generation power.

The thermostat option is not optimal because the electricity price cannot be considered.

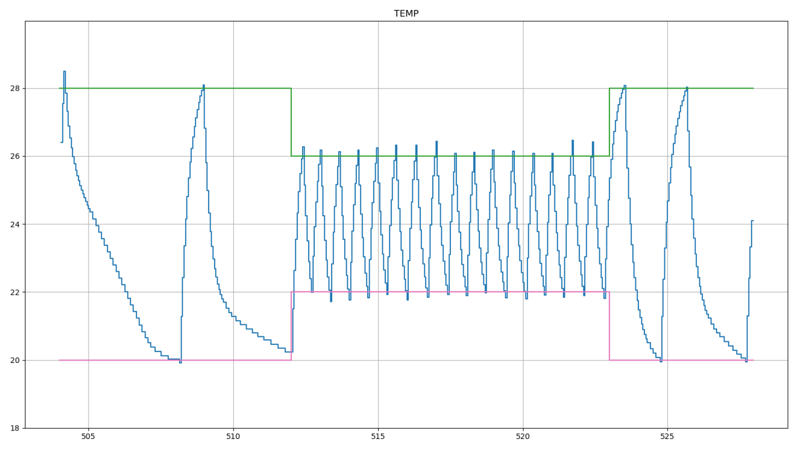


Figure 4.1: Thermostat controller

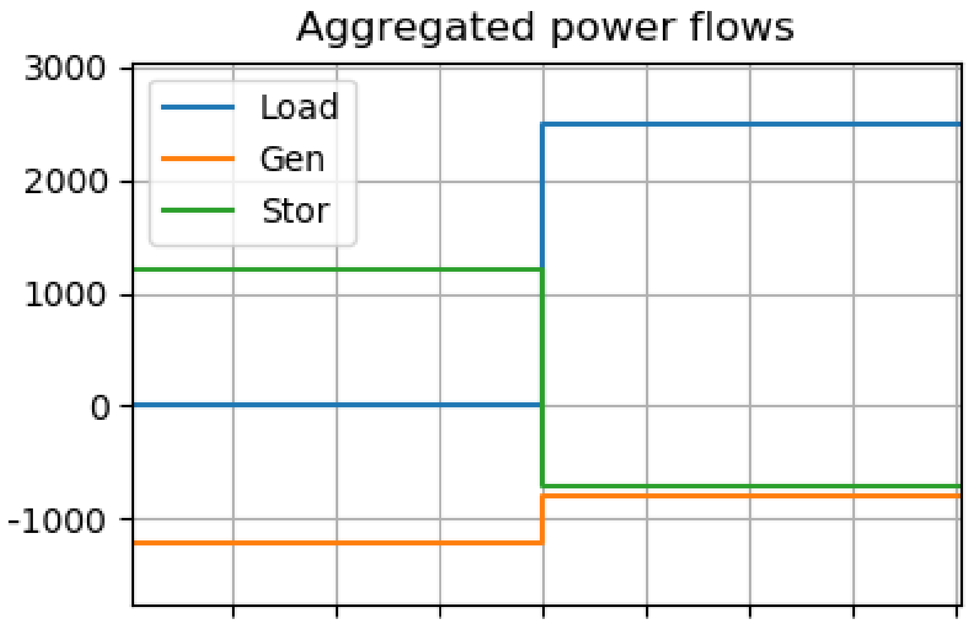


Figure 4.2: Battery logic

4.2 PI controller

A PI is a more advanced controller used for instance in a car to regulate the temperature more precisely. The goal of this controller is to track a reference (T_{ref}) as fast as possible and minimize the error in the steady state.

Figure 4.3 shows the bloc diagram of a PI controller

A PI regulator computes the error(e) and apply a control signal to the plant. A

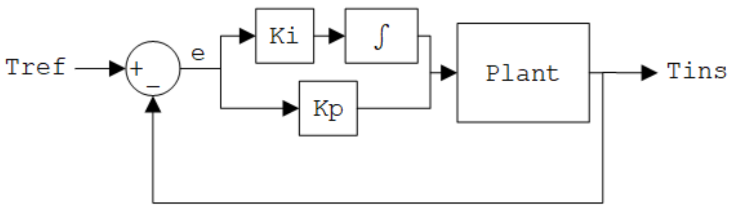


Figure 4.3: PI bloc diagram

negative feedback allows to minimize the error relative to the reference temperature. The sum of the error adjusted by a proportional gain Kp and the integral of the error over time adjusted by a gain Ki is forwarded to the plant. The values of Kp and Ki are tuned to obtain the desire shape of the temperature signal.

These gains can be precisely calculated using a control technique to place the poles of the transfer function of the system and obtain the desire tracking. However, for this project, the controller gains are tune experimentally in a simulation to minimize the temperature overshoot and obtain an accurate tracking. This is good enough for this project purpose.

If Kp and Ki are too high, an overshoot appears when there is an abrupt change in the reference temperature T_{ref}. If Kp is too small, the temperature converges to the reference slowly. If Ki is too small, the controller will never provide enough power to track the reference. A trade-off must be found for the optimal gains.

Figure 4.4 shows the PI with the optimal values ($Kp = 1.6$ and $Ki = 0.015$).

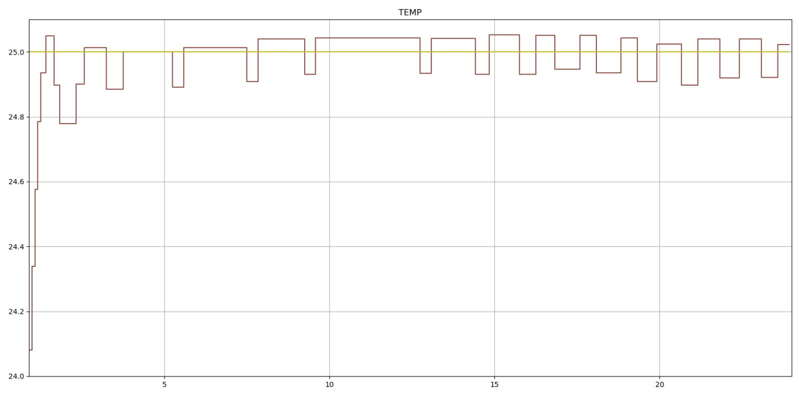


Figure 4.4: PI controller

The error is immediately corrected which gives a signal oscillating around the reference.

The battery logic is the same as the thermostat. A PI controller is a good solution to obtain the best comfort with a very precise temperature but do not account for the electricity price.

4.3 MPC

4.3.1 MPC structure

A MPC (Model Predictive Controller) is chosen to be implemented in the system. This is the only controller able to consider the electricity price to plan an optimal schedule for the heaters and the battery set points. A MPC bloc diagram is shown in figure 4.5.

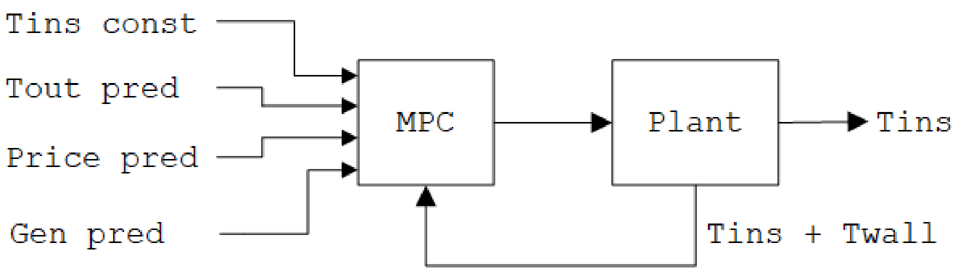


Figure 4.5: MPC controller

This controller takes as input all the predictions, the comfort constraints defined by the resident and the actual states of the plant(the temperature inside the room and the wall). The wall temperature is not measured and therefore not available. A solution will be presented in section 4.3.5 to obtain this value mathematically. For now, this temperature is considered as known.

Acknowledging the model of the rooms and all the input information, the MPC predicts the optimal future powers and temperatures at any simulation step. Finally, only the current battery and heater power computed are forwarded.

A time-line of the controller behaviour is shown in figure 4.6.

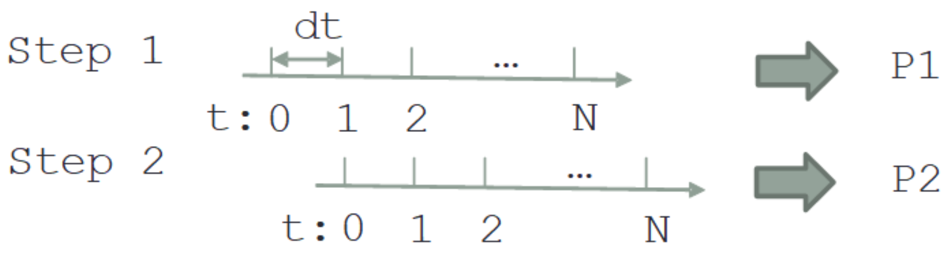


Figure 4.6: Timeline MPC

In step one, temperatures and powers values are computed until a horizon N(number of predicted values defined by the developer). In step two, the actual temperature of the room might not correspond to the predictions in step one. It is due to disturbances or any mismatch between the model and the real building. Therefore, the current state of the room and the predictions are considered again to re-compute the power to apply to the battery and the heaters.

4.3.2 Optimization problem

To implement a MPC, an optimization problem needs to be solved. The mathematical problem is made of two distinct parts to minimize the electricity price while respecting certain constraints:

1. The objective function
   - The electricity price must be minimize.
2. Constraints equations
   - The thermal model of the system must be respected.
   - Comfort constraints defined by the resident must be respected
   - Power for the heater and battery must be bounded

Several libraries exist to help the developer to solve a minimization problem. For this project, cvspy.py library which contains a solver is used in python. The objective function and constraints equations are taken as input. The optimal powers to apply to the system are computed and returned by the solver. Therefore, the way of solving an optimization problem is not detail in this section. However, section 8.1 in annex describes what is behind the solver.

For this problem, (4.1) is the objective function.

$$Cost = \sum_{t=0}^{t=N} c(t) * (P_H(t) + P_B(t) + P_G(t) - P_{sold}(t)) + q_1\epsilon(t)^2 \tag{4.1}$$

With $c(t) = electricity\ price$ at time t, $P_H = heater\ power$, $P_B = battery\ power$, $P_G = power\ generated$ (always negative), $P_{sold} = power\ sold$ (always negative), $q_1\epsilon(t)^2 = quadratic\ form$ associated with constraint (4.6) below.

Below, the constraints of the problem are described.

$$\vec{T}(t+1) = A_d\,\vec{T}(t+1)T(t) + B1_d\,P_H(t) + B2_d\,T_{out}(t) \tag{4.2}$$

$$E_B(t+1) = \alpha E_B(t) + \beta P_B(t)dt \tag{4.3}$$

$$\vec{T}(0) = \vec{T_0}\ \ and\ \ E_B(0) = E_{B0} \tag{4.4}$$

Constraints (4.2) and (4.3) make sure the models described in chapter 3 are respected. Constraint (4.4) initializes the model state values.

$$P_H(t) + P_B(t) + (P_G(t) - P_{sold}(t)) > 0 \tag{4.5}$$

The power purchased is always positive (4.5). $P_G(t) - P_{sold}(t)$ is always negative and represents the power supplied to the house by the generator.

$$T_{inf} - \epsilon(t) < T(t) < T_{sup} + \epsilon(t) \; and \; \epsilon(t) >= 0 \tag{4.6}$$

Constraint (4.6) make sure the temperature constraints are respected. Referring to the quadratic term $q\epsilon(t)^2$ in the objective function (4.1), ϵ is used to relax the constraints and make the problem always feasible even if the temperature is outside the comfort boundaries. Indeed, it is very likely that initially the room temperature is below the minimum and needs to be increased.

If the problem is unfeasible, the simulation aborts and no further computation is performed. By initializing q_1 with a high value, the solver will always try to minimize the quadratic term first and therefore ϵ will tend to zero. Hence, the temperature constraints are reduced until they reached the comfort boundaries defined by the resident. For example, if the initial temperature is below the minimum, the controller will set the heater power at maximum to reach the temperature constraints as fast as possible and minimize ϵ.

$$0 < P_H(t) < P_H max \tag{4.7}$$

$$E_B min < E_B(t+1) < E_B max \tag{4.8}$$

$$P_B min < P_B(t) < P_B max \tag{4.9}$$

Constraints (4.7, 4.8 and 4.9) make sure the battery and heater are bounded as well as the battery energy.

In (4.9) the maximum battery power is set to a constant. This is not true, like it was evidenced in section 3.2, the maximum power is not linear. This is a complex issue because the solver only allow linear constraints (see annex 8.1 for further details). The battery model for the MPC is discussed in section 4.3.4 later.

$$P_G < P_{sold} < 0 \tag{4.10}$$

The power sold (4.10) is always smaller than the power generated by the PV panel.

Figure 4.7 and figure 4.8 show an example of a MPC regulating the temperature inside the room.

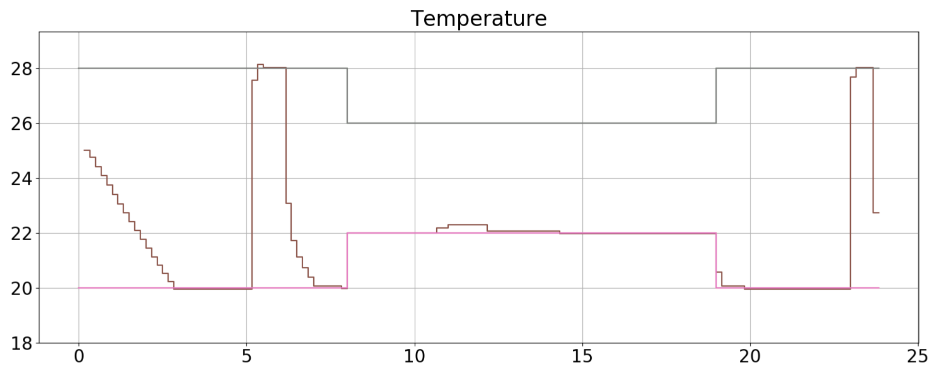


Figure 4.7: MPC temperature

The room is preheated when the electricity price is low and the energy stored thermally is retrieved when the price is high (Figure 4.7).

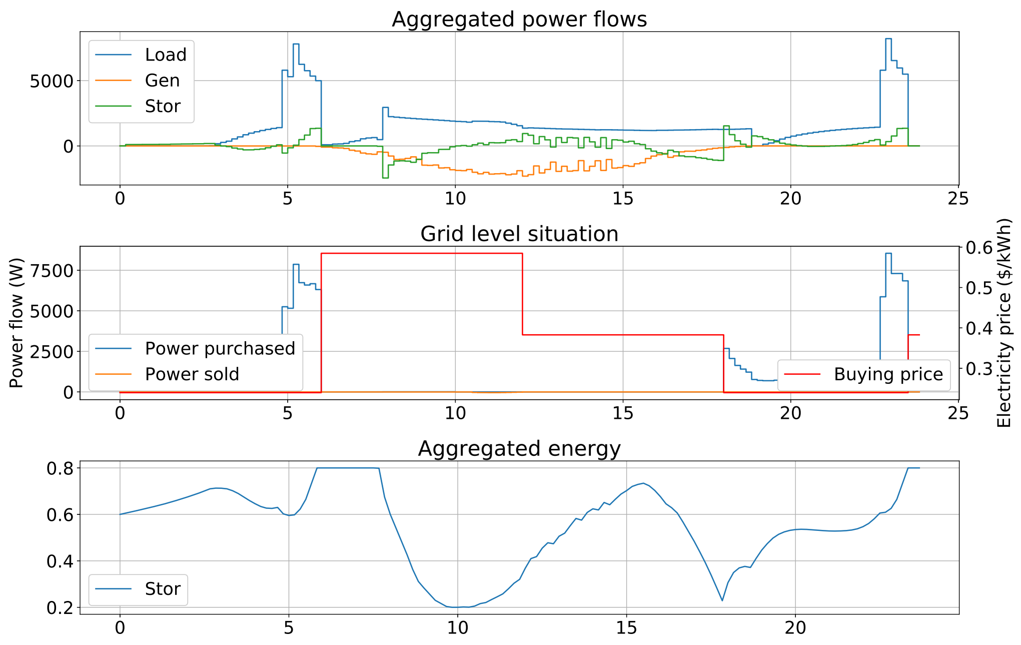


Figure 4.8: MPC power

Figure (4.8) highlights the aggregated power flows, the power purchased and the battery energy. The energy stored in the battery is used when the energy is expensive. When enough energy is provided by the generator to supply the load, the surplus is used to charges the battery. Most of the power is purchased at low cost to anticipate higher electricity price.

4.3.3 MPC objective function

The objective function allows to modify the resident preferences according to priorities. To increase the comfort, the temperature is contained as close as possible to a reference instead of just being between two constraints. The electricity sold to the utility can be also minimize to encourage self-consumption. To account these criterion, another objective function (4.11) is implemented with additional terms.

$$Cost = \sum_{t=0}^{t=N} c(t)*(P_H(t)+P_B(t)+P_G(t)-P_{sold}(t))+q_1\epsilon(t)^2+q_2*(T(t)-T_{ref})^2-q_3*P_{sold} \tag{4.11}$$

Coefficients q_2 and q_3 are tuned according to the objective. The higher q_2 is, the more the reference will be tracked with less consideration to the electricity price. The higher q_3 is, the higher the self-consumption is encouraged and less energy is sold to the grid. Recall that P_{sold} is always negative, hence there is a negative sign before q_3 to minimize the power sold.

4.3.4 MPC battery model

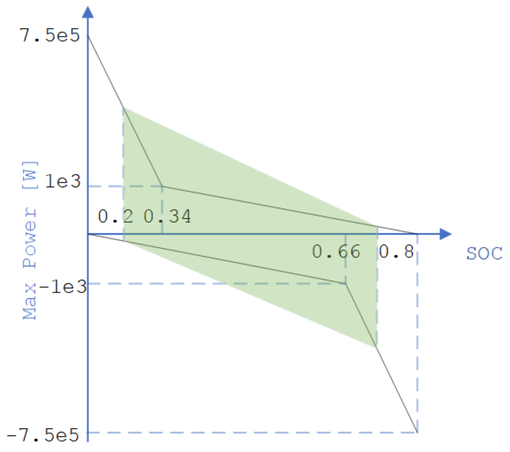
As mentioned in section 4.3.2, the maximum power curve for the battery (figure 3.5) is non-linear and the MPC only accepts linear constraints. Thus, a new model is developed to approximate the maximum battery power curve for the minimization problem.

Three different models were tested (figure 4.9). Those models were simulated in different scenarios to determine which one gives the best approximation.

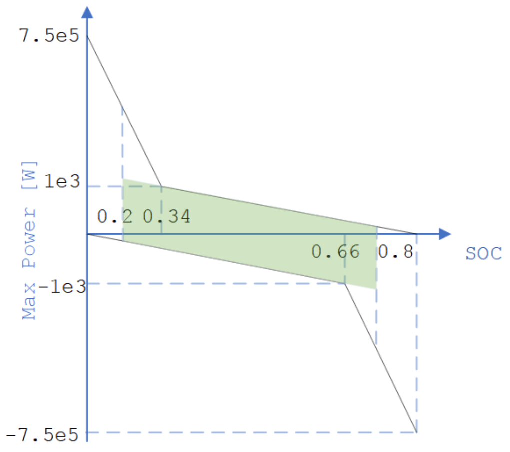
Recall that the y-axis of the plot is not linear. The power used by the MPC to make prediction is highlighted in green. The charge(positive value)/discharge(negative values) maximum power curve are sketched in black. The SOC cannot be below a minimum (0.2) or above a maximum (0.8). The model must limit the MPC computation error on the predictions. The power flowing out of the controller is always restricted to the real power limit at the end of a MPC cycle, even if the power allowed for the computation is beyond the maximum power.

Model A (figure 4.9a) allows the largest area for the MPC computations. This involves that the controller might allow much bigger powers in the predictions and possibly make wrong forecasts. Eventually, if the power given by the controller is bigger than the maximum power curve (in black), this power is limited to the maximum.

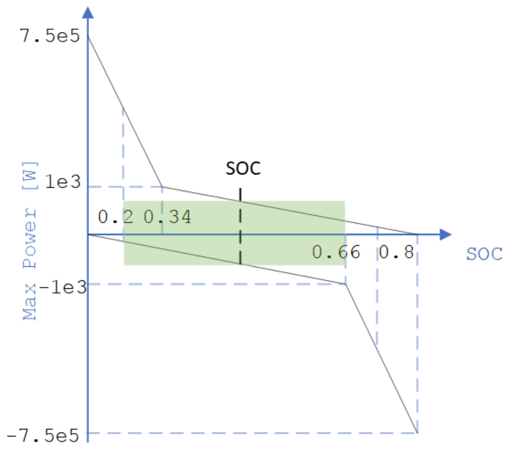
Model B (figure 4.9b) limits the power area for the MPC computations. For the charge curve, there is a large range of value for SOC smaller than 0.34 that the



(a) Battery model A



(b) Battery model B



(c) Battery model C

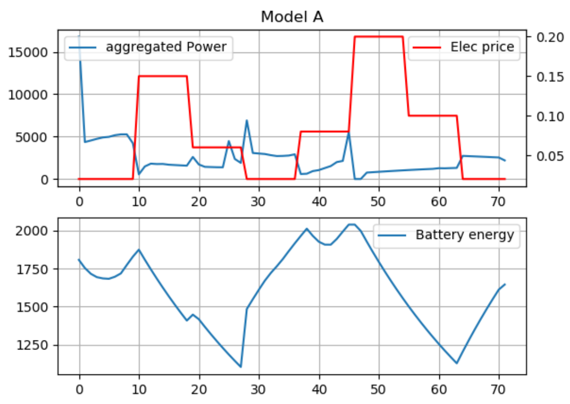
Figure 4.9: Battery models

controller will never consider. However, compare to model A, the MPC will never predict impossible power values.

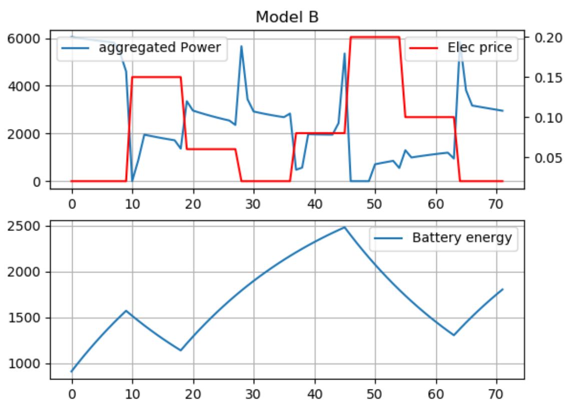
Model C (figure 4.9c) considers varying power allowed by the MPC depending on the current SOC. Considering the charge curve, the maximum power for the current SOC is accounted and the MPC can use any power below this maximum in the future. If the future battery SOC is above the current SOC, the controller might make impossible predictions. If the future battery SOC is below the current SOC, the controller does not take advantage of the whole allowed power.

The three different models are tested for two different scenarios in a simulation lasting 6 hours, 5 minutes time step and 6 hours controller horizon. Scenario 1 starts with a battery SOC of 0.25 and scenario 2 starts with a SOC of 0.75. A fictive electricity price close to a real scenario is used. The outside temperature is also fictive and the room temperature is bounded between 2 values.

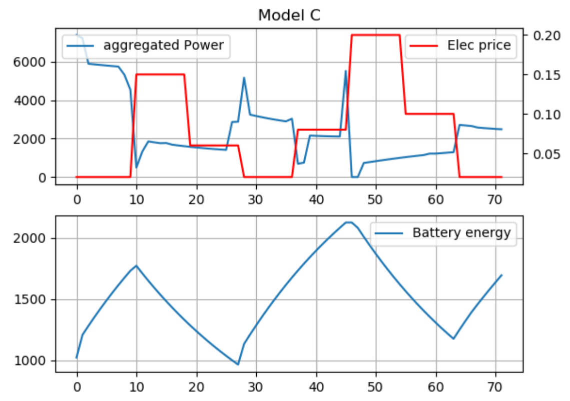
Figure 4.10 shows the scenario 1 power flow for the three models.



(a) Power model A



(b) Power model B



(c) Power model C

Figure 4.10: Power models

In model A (4.10a), it is observed that the battery is initially charged very fast because the power allowed for 0.25 SOC is large. Then the power is relatively restricted for the rest of the simulation but is higher than the other models on average.

Model B (4.10b) and C (4.10c) are relatively similar.

Table 4.1 highlights the fictive cost for the different scenarios.

Model	$SOC_{init} = 0.25$	$SOC_{init} = 0.75$
A	0.750	0.714
B	0.749	0.654
C	0.758	0.690

Table 4.1: PV Battery models cost result

It can be concluded that model B is the best model. Indeed, even if model A performs better for the 0.25 initial SOC scenario, it is only because the battery was charged very fast initially. Model A did not perform well for scenario 2 due to the MPC biased computations.

Model B is always better than model C. It is not surprising because the MPC with model C will either perform wrong predictions or restrict the power available.

4.3.5 MPC observer

As it was mentioned in section 4.3.1, the temperature inside the wall is unknown and is needed for the MPC. This temperature will have to be guessed mathematically being as accurate as possible.

In the controller theory, computations are based on the model which never completely corresponds to the reality. Taking a PI as example, Ki and Kp gains are computed optimally according to the model which is close but but not exactly the real plant. However, this approximation is good enough since the controller is robust to noise and corrects the mismatch model/reality. Of course, this assume that the states variables are known. It is not the case for this MPC and the wall temperature.

Therefore, there are two issues to solve:

- The initial temperature inside the wall is unknown at the beginning of the simulation and must converge as fast as possible to the building wall temperature
- The wall temperature in the simulation must be as close as possible to the building wall temperature (which slightly differs to the wall temperature varying according to the thermal model developped in section 3.1.2)

Thus, a mathematical entity called observer in the control theory is implemented. An observer checks how a physical variable behave by sending an input signal to the plant and watch the change in the output. Then, every states variable are guessed. In this project context, the room temperature is observed according to the power injected by the heater and the wall temperature is guessed. The bloc diagram of the system accounting for the observer is shown in figure 4.11.

The wall temperature in the observer ($Twall\ init$) could be initialized to a random value. However, if this value is close to the reality, the wall temperature will converge faster. Therefore, the wall temperature is initialized as the average of the initial inside and outside temperature.

In control, the states variables are represented by x and the output is y. The states guessed are $\hat{x}$ and the input u ($x=\begin{pmatrix}T_{ins} & T_{wall}\end{pmatrix}^T$, $y = T_{ins}$, $u = P_H$)

The error e between the real states and the guessed state is computed without observer (4.13 and 4.14).

$$\hat{x}(t+1) = A\hat{x} + Bu \tag{4.12}$$

$$e(t) = \hat{x}(t) - x(t) \tag{4.13}$$

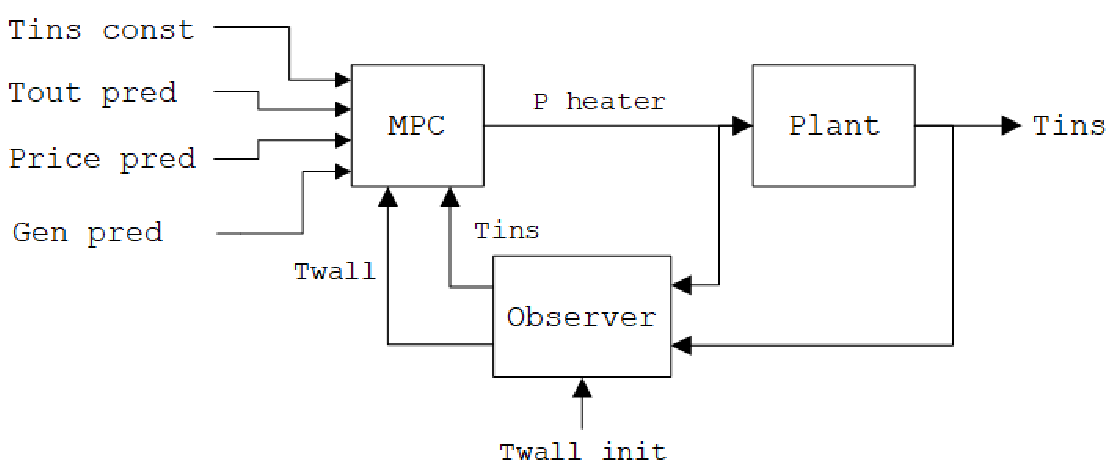


Figure 4.11: MPC with observer

$$e(t+1) = \hat{x}(t+1) - x(t+1) = A(\hat{x}(t) - x(t)) = Ae(t) \tag{4.14}$$

Equation (4.14) shows that the error evolves according to the matrix A. In control, the poles of the transfer function characterized by A influences the convergence speed of the error which must tend to 0 as fast as possible. Since A is fixed, the convergence cannot be adjusted without an observer.

The observer can be tuned by changing the coefficient of a matrix L. L has two rows an one column for a system with two states variables and one input. Equation (4.15) shows the model of the observer.

$$\hat{x}(t+1) = \underbrace{A\hat{x} + Bu}_{prediction} + \underbrace{L[y(k) - \hat{y}(k)]}_{adjustment} \tag{4.15}$$

$A\hat{x} + Bu$ corresponds to a model running in parallel to the plant to predict the states variables. $L[y(k) - \hat{y}(k)]$ is a factor which adjusts for the mismatch between the model and the reality. Indeed $\hat{y}$ is the output prediction and is an indication of the observer performance. This value is compared with the real output y.

Equation (4.15) can be also written like (4.16) since $\hat{y} = C\hat{x}$. With C the matrix linking the states and output. In the case of the room, C is simply $\begin{pmatrix} 1 & 0 \end{pmatrix}$ because only the inside temperature is measurable ($T_{ins} = C \begin{pmatrix} T_{ins} & T_w \end{pmatrix}^T$).

$$\hat{x}(t+1) = (A - LC)\hat{x} + Ly + Bu \tag{4.16}$$

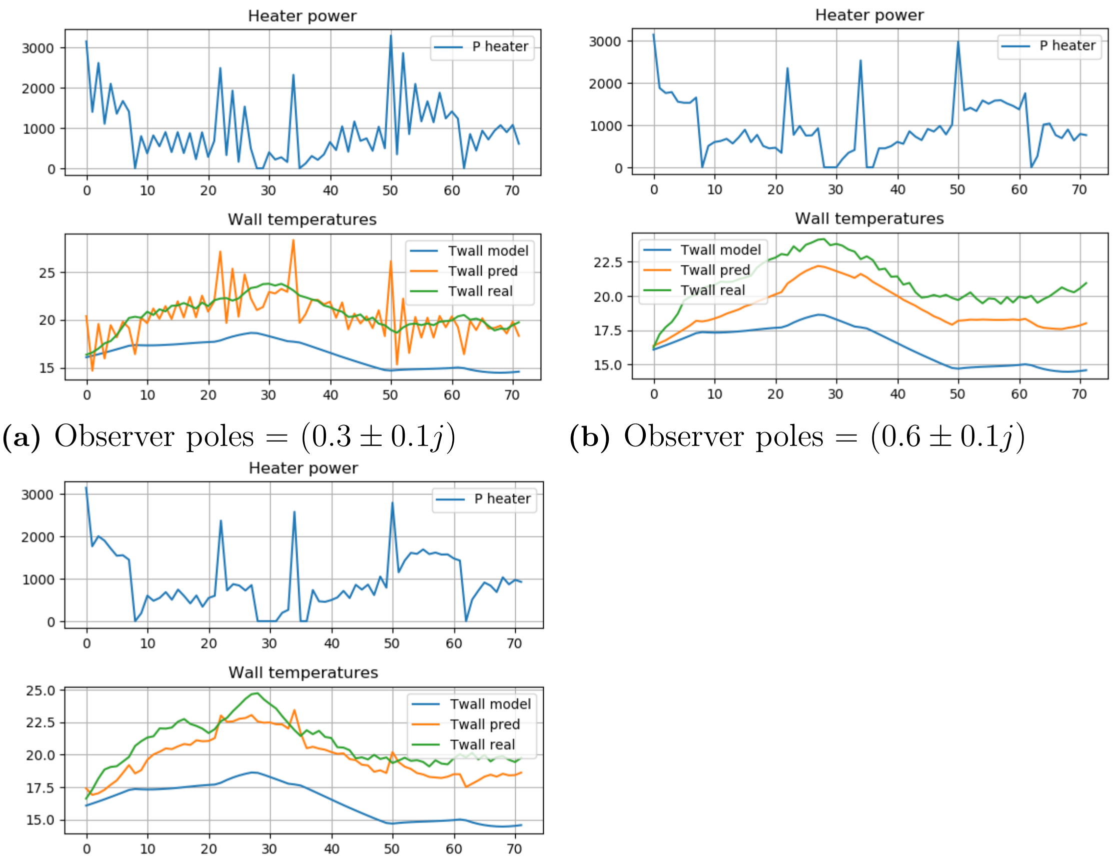
Then, like (4.14) was computed, the error is derived accounting for the observer (4.17).

$$e(t+1) = (A - LC)e(t) \tag{4.17}$$

Equation (4.17) shows that the error evolves according to the matrix $A - LC$. Since the coefficients of L can be chosen, the poles of $A - LC$ are tuned to make the error converge to zero rapidly.

To modify the transfer function of the observer system, two complex conjugate poles of $A - LC$ must be chosen within the complex unit circle. If the poles are close to 0, the wall temperature converges very fast. However, a fast convergence implies a system more sensitive to temperature variation/noise which is always present in a room. If the norm of the complex poles is close to 1, the wall temperature converges very slowly. However, the system is less sensitive to temperature variations and noise. Thus, a trade-off must be found to adjust the complex poles.

To tune the observer, several simulations are performed with different complex conjugates poles values (figure 4.12).



(a) Observer poles $= (0.3 \pm 0.1j)$

(b) Observer poles $= (0.6 \pm 0.1j)$

(c) Observer poles $= (0.9 \pm 0.1j)$

Figure 4.12: Observers models

A noise with a constant biased is added to the temperatures given by the model to simulate the real temperatures (4.18).

$$T_{real} = T_{model} + random[0, 1] \tag{4.18}$$

The observer must be able to guess in which direction is the bias to converge to the building wall temperature.

Figure 4.12a shows the wall temperatures with poles at $(0.3 \pm 0.1j)$. The poles are close to zero and the predicted wall temperature is sensible to noise. It is observed that when the power is abruptly changed by the MPC, it is difficult for the observer to adapt and guess the wall temperature correctly. Indeed, as seen in figure 4.11, the power is an input of the observer and an abrupt change perturbs the system. Since the MPC base the computations on a noisy wall temperature, the power sent to the heater is also noisy.

Figure 4.12b shows the wall temperatures the poles at $(0.9 \pm 0.1j)$. These poles are close to the unit circle, therefore the wall temperature predicted converge slowly to the real temperature. However almost no noise is observed.

Figure 4.12c shows the wall temperatures the poles at $(0.6 \pm 0.1j)$. These poles values evidence a fair balance between fast convergence and noise for this special case. The abrupt change in heater power sent by the MPC remains an issue difficult to manage.

To find the best poles values, the standard deviation between the real and predicted wall temperatures is computed (Table 4.2). Poles at $(0.6 \pm 0.1j)$ clearly perform the best results.

Poles	STD
$0.3 \pm 0.1j$	0.221
$0.6 \pm 0.1j$	0.142
$0.9 \pm 0.1j$	0.194

Table 4.2: Standard deviation observers

Without observer, the standard deviation corresponds to the error between the real and model wall temperature. This value is around 7.63 which is much higher than with an observer.

The observer is also tested with a mismatch between the model and the reality. Figure 4.13 show the observer if in reality, the wall thermal resistor is 1.5 times smaller than what the model expect. This time, because the temperature is less noisy, poles closer to the unit circle are a better choice $(0.8 \pm 0.1j)$

In that case, the wall temperature matches almost perfectly to the building wall temperature.

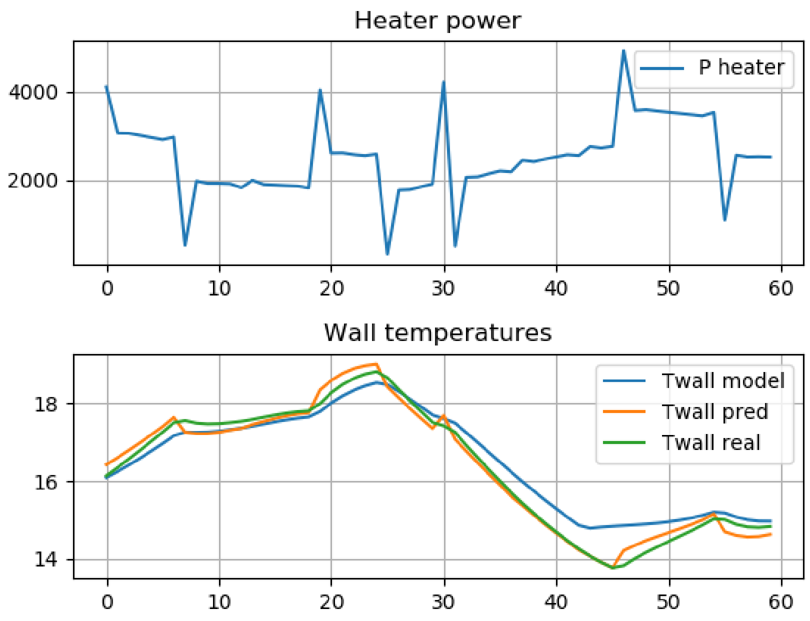


Figure 4.13: Observer mismatch thermal resistor

4.4 Simulation

So far, every simulation was performed using test files to develop the MPC or the observer for one room and one battery. From now, the simulation will be run with the program provided by the ELAB, which allows to generate a building with multiple rooms and different topologies. A server containing the information of the building is available. This database includes the rooms, the wall properties and if the walls contain doors or windows. The electricity price, temperature and sun forecast is also obtained from this database.

The program is composed of two entities:

- The simulator: Emulates the building by using the database information.
- The Energy Management System: Contains the strategy implemented (MPC, PI, Thermostat).

The simulator and the EMS are two process running at the same time. Every time step along the simulation, the desired heaters powers and batteries set points are sent by the EMS to the simulator. Then the simulator communicates the temperature and battery state to the EMS. Figure 4.14 illustrates the whole operation of the building simulation.

Figure 4.15 illustrates the program structure with the main classes and function. All these entities allow to obtain all the data to implement our energy management strategy.

A detailed explanation of the classes and relevant functions are available in annex

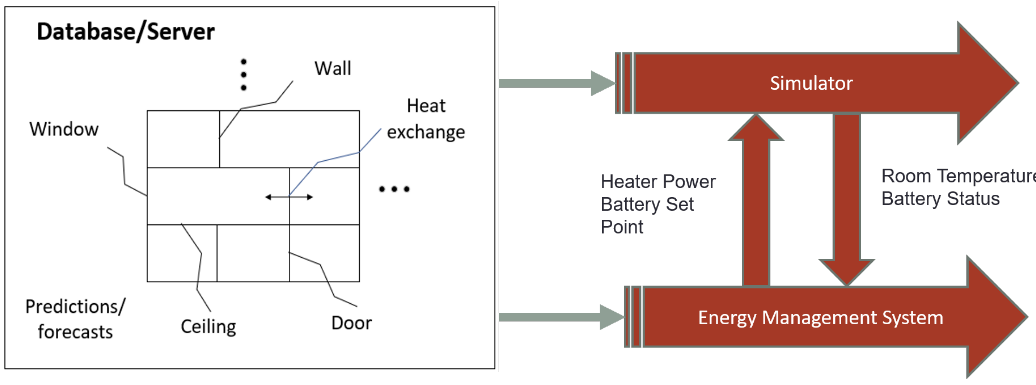


Figure 4.14: Simulation operation

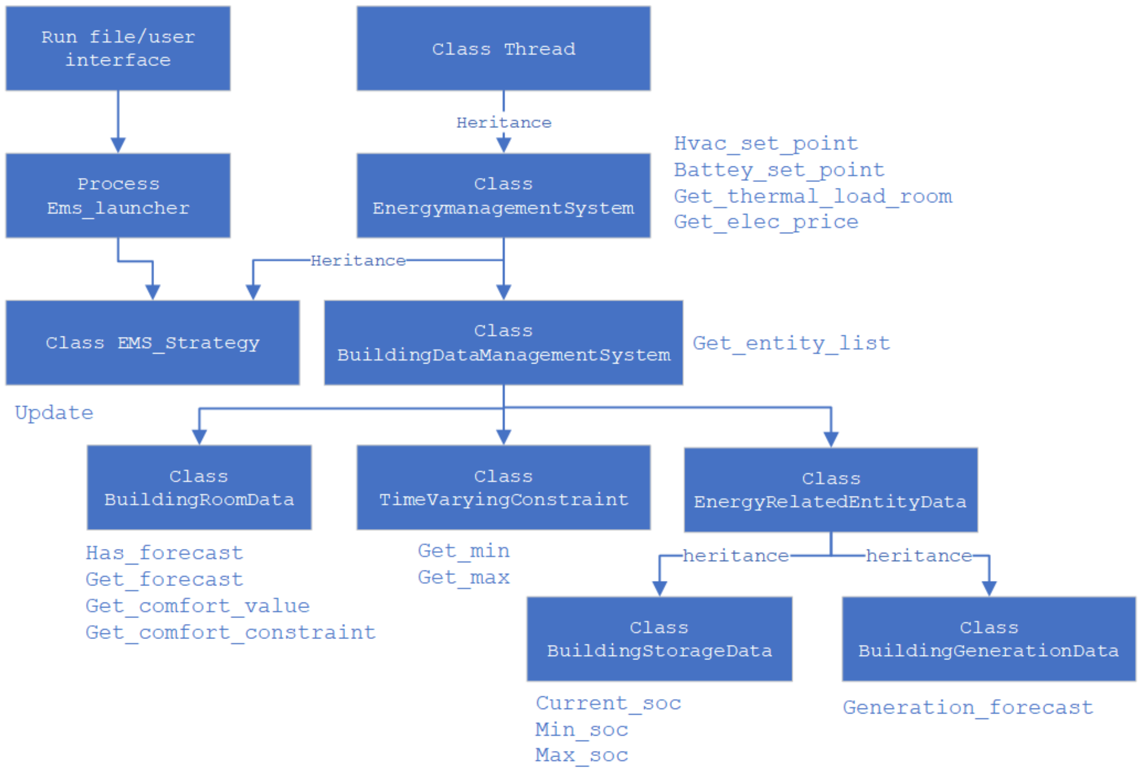


Figure 4.15: Program structure

section 8.2.

5

Results

5.1 MPC vs PI vs Thermostat

In this section, the MPC without temperature tracking is compared with the PI and thermostat. A house with two rooms and one battery is considered.

Figure 5.1 shows the outside temperature which is the same for every simulation. The outside temperature and electricity price are retrieved from a typical day in California.

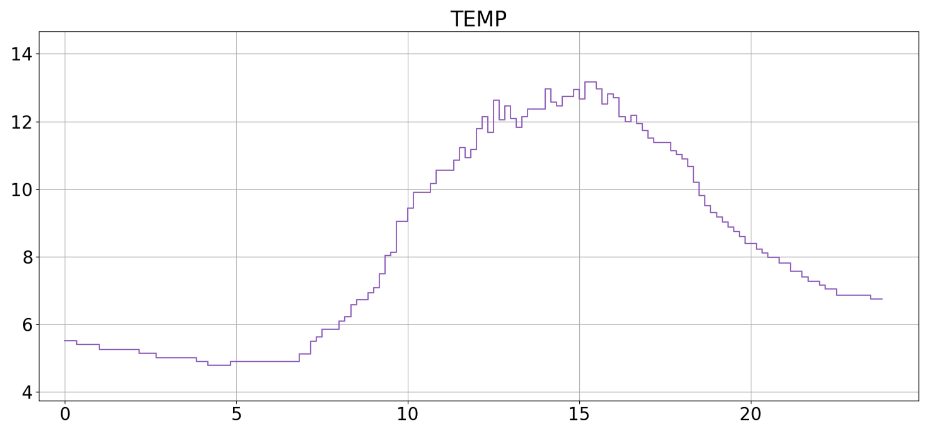


Figure 5.1: Outside temperature

The controllers are compared for a simulation lasting 24 hours with a 10 minutes time step. It was chosen that sending power to the grid do not bring money back.

Figure 5.2 and 5.3 show the temperature and power plots for the thermostat developed in section 4.1. The battery is used to maximize the self-consumption.

It is observed in figure 5.2 that between 12h and 16h, the battery is full. When all heater are turned off, the energy generate has no other choice than be sold to the grid.

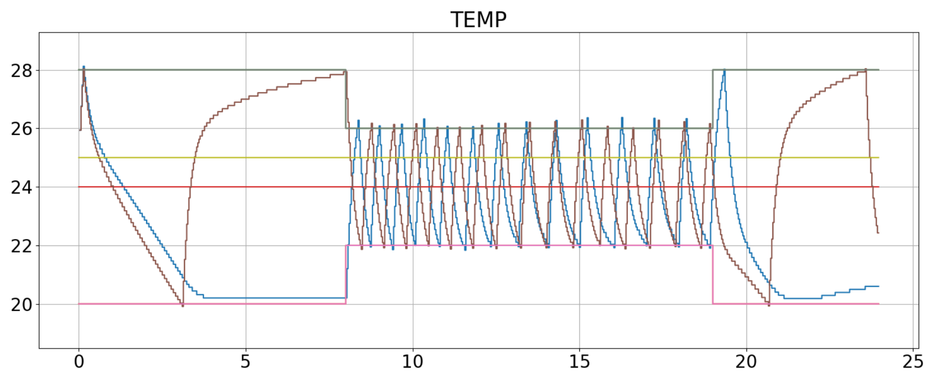


Figure 5.2: Thermostat temperature

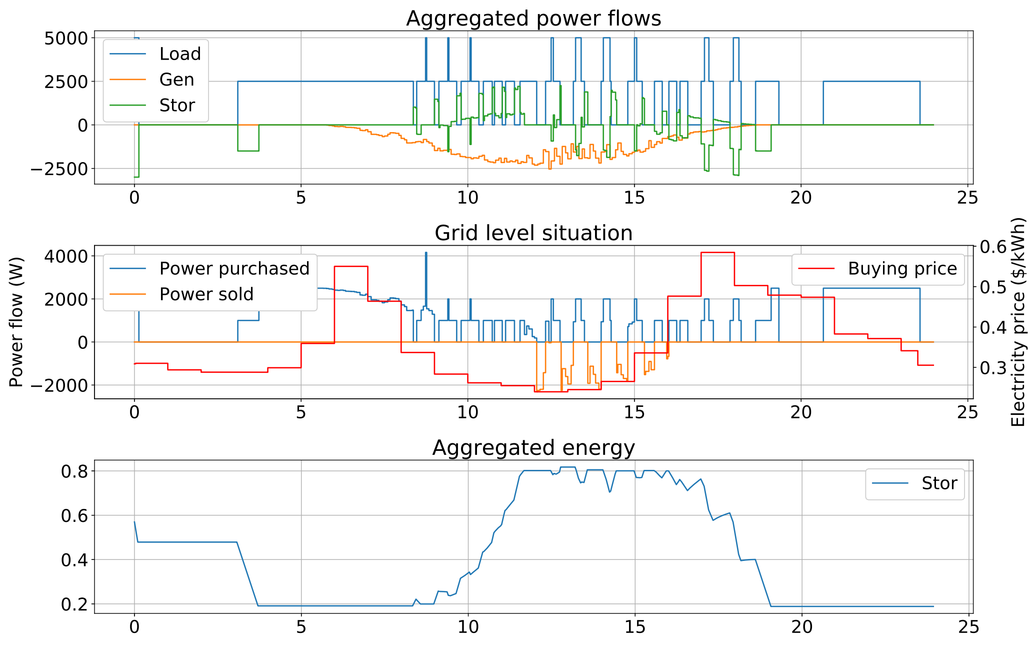


Figure 5.3: Thermostat power

Figure 5.4 and 5.5 show the temperature and power plots for the PI controller developed in section 4.2.

Like the thermostat, the battery maximizes the self-consumption. The power required to heat both rooms remains almost constant. Therefore, self-consumption can be achieved.

Figure 5.6 and 5.7 illustrate the temperature and power flow for the MPC without tracking developed in section 4.3. The MPC computation horizon is 6 hours. The two rooms are preheated at the same time and the battery energy is used when the price is expensive. There is no power sold to the grid since it does not bring money

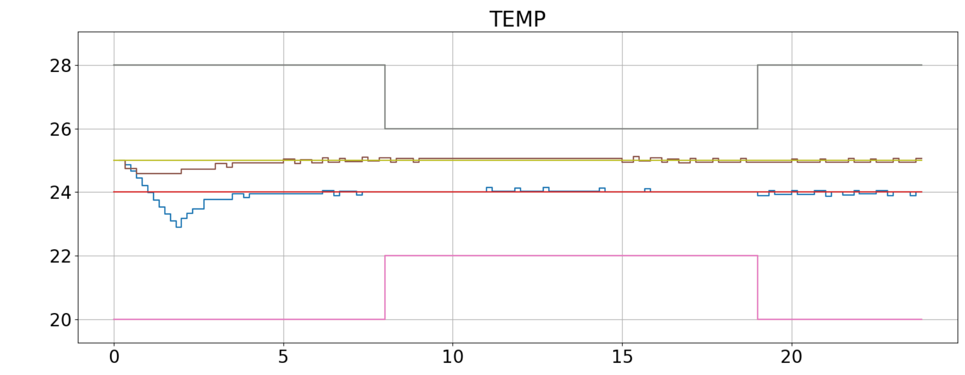


Figure 5.4: PI temperature

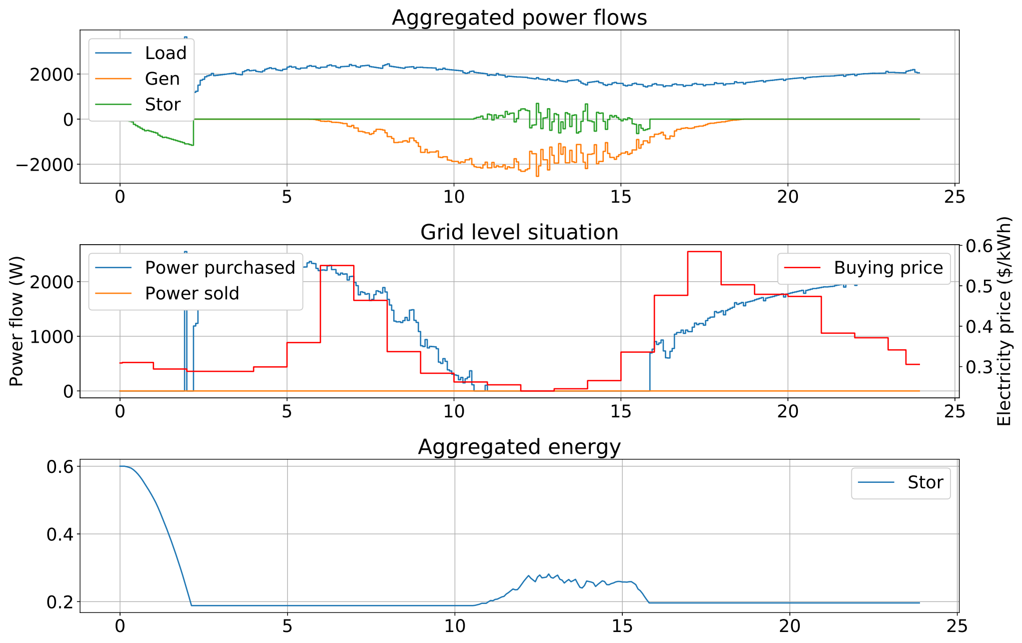


Figure 5.5: PI power

back.

Table 5.1 shows the grid and thermal analysis for the different controllers. The standard deviation between the reference temperature and the room temperature is evidenced (err T_{R1} and err T_{R2}) for both rooms and corresponds to a comfort measurement.

The thermostat performs better than the PI in term of energy purchased. The MPC saves almost half of the electricity cost compare to the thermostat which is significant. However, this study do not account for the difference in comfort between the controllers.

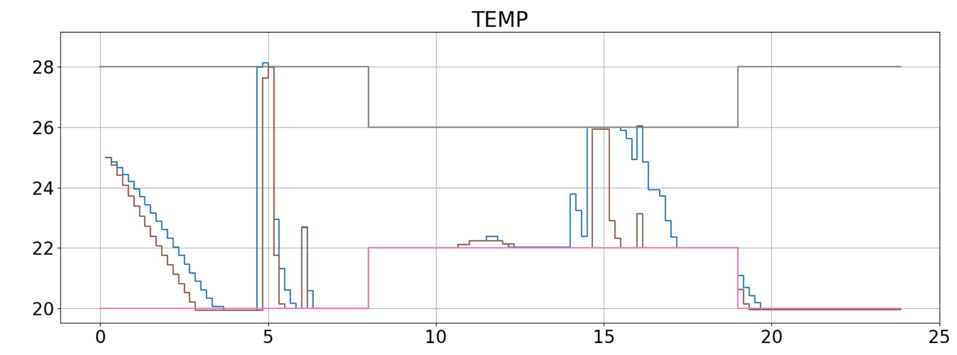


Figure 5.6: MPC temperature

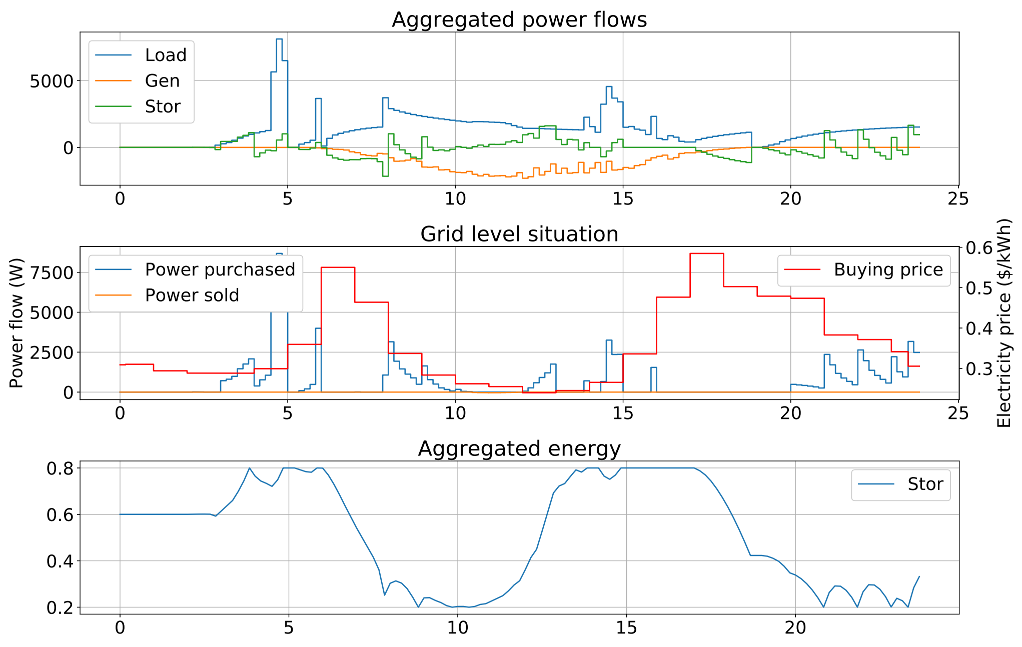


Figure 5.7: MPC power

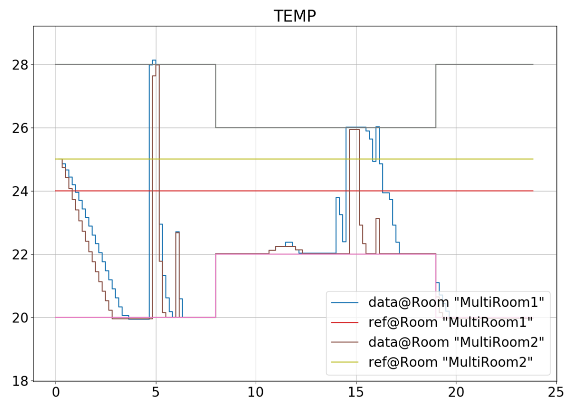
	E purchased	Price	E sold	err T_{R1}	err T_{R2}
Thermostat	26.73 kWh	99.76$	2.70kWh	2.53	2.28
PI	27.26 kWh	109.88$	0.0kWh	0.23	0.13
MPC	15.65 kWh	50.04$	0.042kWh	2.84	3.86

Table 5.1: Report controllers

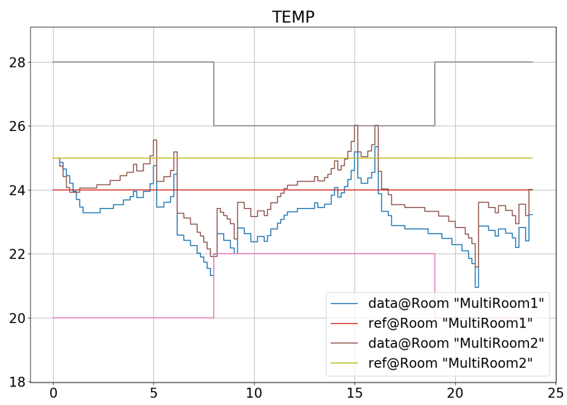
5.2 MPC tracking vs PI

In this section, the MPC with the tracking coefficient q_2 (see objective function section 4.3.3) is compared to the PI controller. The tracking coefficient allows to tune the balance between the electricity cost minimization and a good reference tracking. The simulation parameters are the same as section 5.1. The system is compared for $q_2 = 20$, $q_2 = 50$ and $q_2 = 100$

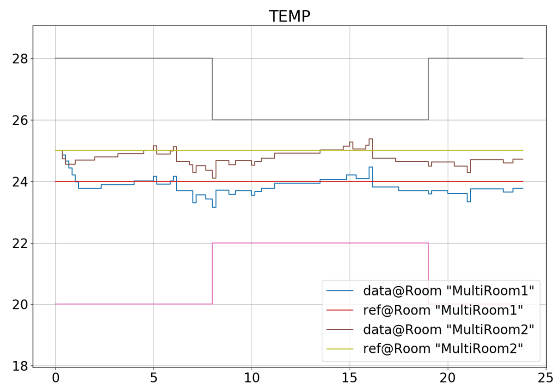
Figure 5.8 highlights the difference in the inside temperature depending on how much the comfort is fostered with q_2.



(a) $q_2 = 0$



(b) $q_2 = 20$



(c) $q_2 = 100$

Figure 5.8: MPC tracking

Table 5.2 shows the grid and thermal analysis for the different controllers and value of the tracking coefficient q_2.

	E purchased	Price	E sold	err T_{R1}	err T_{R2}
MPC $q_2 = 0$	15.65 kWh	50.04$	0.042kWh	2.84	3.86
MPC $q_2 = 10$	19.87kWh	63.19$	0.0kWh	1.96	2.52
MPC $q_2 = 20$	23.58kWh	78.0$	0.0kWh	1.19	1.43
MPC $q_2 = 50$	26.06kWh	90.23$	0.0kWh	0.53	0.61
MPC $q_2 = 100$	26.95kWh	95.06$	0.0kWh	0.30	0.31
PI	27.26 kWh	109.88$	0.0kWh	0.23	0.13

Table 5.2: Report tracking controllers

The PI controller is still the best to optimize the comfort but has the highest cost. By setting q_2 to a high value, the electricity price and comfort of the MPC approach the PI values. Depending on q_2 the the tracking error is reduced while the cost increases.

5.3 Influence of horizon

The horizon for which the MPC computes the optimal powers has a relevant impact on the price. If the horizon is too small, the controller might not see that the price will be higher in the future and preheat the room/charge the battery. If the horizon is too large, the simulation completion time is too long.

Moreover, it might be useless to use a very large horizon time. Indeed, if the MPC detects a peak consumption in 4 hours and preheat the room, it is not useful to know that another peak electricity price will occur 8 hours later.

Figure 5.9 and 5.10 show the difference in power for a horizon of 1.5h and 9h

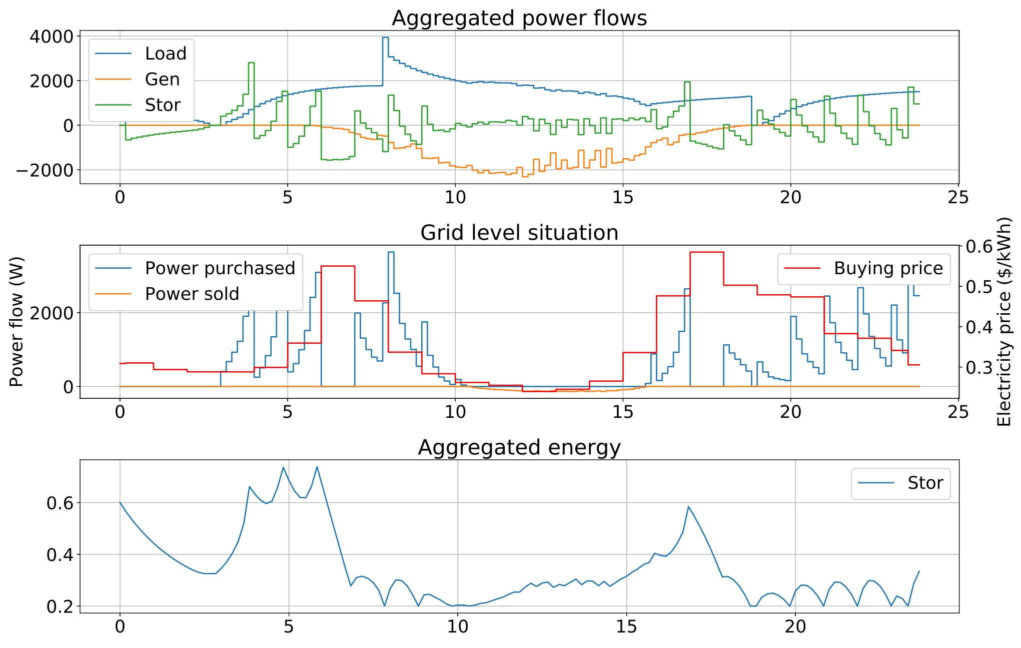


Figure 5.9: MPC horizon = 1.5h

With a small horizon, it is evidenced that the power is used right before a price increase or right after a price decrease. This is due to a lack of long term price consideration. With smaller horizon, there is less planning and the energy is stored less in the battery. For higher horizon, even if the prices increases in a short term, no power is used in prevention of more favorable time to preheat the room or charge the battery.

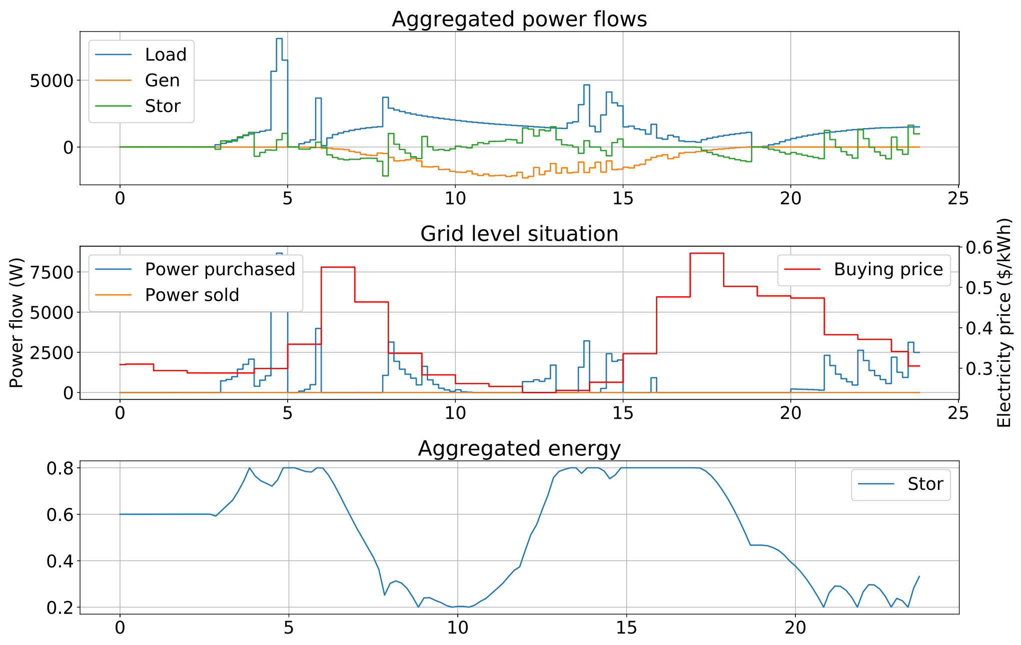


Figure 5.10: MPC horizon = 9h

Table 5.3 compare the simulation reports with the different horizons.

Horizon	E purchased	Price	E sold	err T_{R1}	err T_{R2}	simulation time
1.5h	15.43 kWh	57.45$	0.53kWh	2.85	3.83	0:57 min
3h	15.22 kWh	52.61$	0.287kWh	2.85	3.84	1:32 min
4.5h	15.37 kWh	50.96$	0.138kWh	2.83	3.86	2:05 min
6h	15.65kWh	50.04$	0.02kWh	2.84	3.86	2:36 min
9h	15.78kWh	49.72$	0.0kWh	2.84	3.85	3:50 min
12h	15.75kWh	49.68$	0.0kWh	2.82	3.85	4:57 min

Table 5.3: Horizon comparison

It is observed that the price is smaller when the horizon grows. Small horizons even sell energy due to a bad planning. However, the difference in price is small, especially for higher horizon. The simulation time gets significantly higher with larger horizon.

Comparing 6h horizon and 12h horizon, there is only 0.36$ cost difference and a simulation time difference of 2:21 minutes. Between 4.5h and 6h horizon, there is a simulation time gain of 0:31 for a cost lost of 0.92$ which would remain significant for a month (27.6$ in average). Thus, a 6h horizon is a good compromise between simulation duration and minimal cost.

6

Future steps

6.1 Robust control

The MPC computations are based on the electricity price and the weather forecast. It is very likely that these predictions are not accurate and will not occur as planned in reality. Therefore, the MPC computes wrong forecasts and the temperature might get out of the comfort constraints. The issue will be to find a way to always respect these temperature boundaries even with the worst predictions. A new model based on probability must be developed. One strategy would be to create fictive constraints which are more restricted than those defined by the resident (figure 6.1).

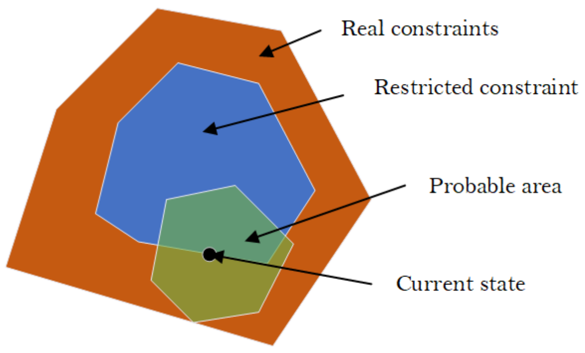


Figure 6.1: Robust MPC constraints

In orange is the area for which the temperature constraints are defined respecting the temperature bounds if the forecasts are correct. In green is the probable state that the inside temperature can possibly reach the next simulation step taking into account the worst prediction scenario. In blue is the restricted constraint. If the inside temperature stays in this area, it is sure that even with the worst predictions the constraints are respected.

6.2 Load profile

So far, the loads considered were only the battery and the heaters. However, a building contains multiple loads such as computers, lights, wash machines or possibly electric cars. Some of the loads are planned according to the habit of the resident and some other could be planned according to a load scheduling (which is another project developed at the ELAB). Given this load profile which changes every day, the program would have to change the heater and battery planning to avoid consuming at the same time as other loads. The minimization problem/MPC would have to account for some ranges during the day that are less optimal to use power.

7

Conclusion

Demand Response will play a major role in Smart Grids implementation and will allow benefits on system operation and on market efficiency. A smart building solution evidenced a way to minimize the electricity bill of one residence while in a large scale supporting the grid to avoid peak energy consumption.

An intelligent way to optimize the comfort temperature inside the rooms by using a MPC strategy with a PV panel and a battery was reported. The building was modelled and simulated in a generic manner for any floor topology. The predictive controller enables many possibilities to trade off between a better comfort or more economy savings. A broad research on the MPC operation was carried out.

The study evidenced a significant economic benefit of using a MPC instead of basic controllers such as a thermostat or a PI. A more robust control accounting for other loads in the house will be required to enable more savings while respecting the comfort temperature constraints.

8

Annex

8.1 Linear programming

8.1.1 Solver

This section aims to explain how a solver works using linear programming. In other words, how to optimize a function under constraints. In linear programming, the solver can find a solution only if the objective function is convex and the constraints are linear. Therefore, the solution is always located at the boundaries of the constraints or at one single point inside the area defined by the constraints if the objective function is quadratic. To illustrate, a linear function $f(x) = 1 - x1 - x2$ is used with simple constraint forming a square on x1, x2 plane (figure 8.1). The problem is feasible only if the area defined by the constraints is not empty.

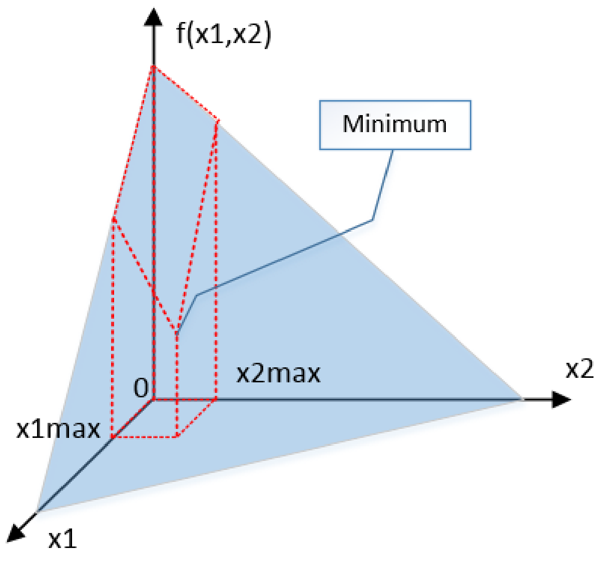


Figure 8.1: Convex function

A basic solution would be to test every point to find the minimum. A more advanced algorithm called simplex is used in many solvers. This algorithm consider the gradient of the function to orientate the research in the right direction and eliminate the points that cannot be a minimum. However, the worst case of this strategy needs the same amount of iterations as the basic solution if the research starting point have the worst position.

8.1.2 Solution for thermal problem

The solver requires linear equations for the constraints to solve the problem and return the optimal powers. The thermal model equations developed in section 3.1.2 gives the thermal behaviour for one time step. However, the constraints equations must describe the system for every time step.

This section describes how the constraints matrices are built for a 3 time steps horizon and only for the thermal model (for readable purpose). It can be generalized for a N time steps horizon including the battery equations using the same method. The constraints equations must be on the linear equation form $A_{sys}\vec{x} = B_{sys}$.

Equation (8.1) shows the matrix equation for 3 time steps built from the thermal model equations. A,$B1$, $B2$ refer to the thermal model matrices equations (3.13).

$$\underbrace{\begin{pmatrix} T_{ins}(1) \\ T_w(1) \\ T_{ins}(2) \\ T_w(2) \\ T_{ins}(3) \\ T_w(3) \\ P_H(0) \\ P_H(1) \\ P_H(2) \end{pmatrix}}_{\vec{x}} = \underbrace{\begin{pmatrix} A_{2x2} & 0_{2x2} & 0_{2x2} & B1_{2x1} & 0_{2x1} & 0_{2x1} \\ 0_{2x2} & A_{2x2} & 0_{2x2} & 0_{2x2} & B1_{2x1} & 0_{2x1} \\ 0_{2x3} & 0_{2x2} & A_{2x2} & 0_{2x2} & 0_{2x2} & B1_{2x1} \\ 0_{1x1} & 0_{1x1} & 0_{1x1} & 1_{1x1} & 0_{1x1} & 0_{1x1} \\ 0_{1x1} & 0_{1x1} & 0_{1x1} & 0_{1x1} & 1_{1x1} & 0_{1x1} \\ 0_{1x1} & 0_{1x1} & 0_{1x1} & 0_{1x1} & 0_{1x1} & 0_{1x1} \end{pmatrix}}_{M_{9x9}} \begin{pmatrix} T_{ins}(0) \\ T_w(0) \\ T_{ins}(1) \\ T_w(1) \\ T_{ins}(2) \\ T_w(2) \\ P_H(0) \\ P_H(1) \\ P_H(2) \end{pmatrix} + \underbrace{\begin{pmatrix} B2_{2x1}T_{out}(0) \\ B2_{2x1}T_{out}(1) \\ B2_{2x1}T_{out}(2) \\ 0_{1x1} \\ 0_{1x1} \\ 0_{1x1} \end{pmatrix}}_{b_{9x1}} \tag{8.1}$$

The vector multiplying M_{9x9} is written (8.2) to evidenced the unknown vector x on the left side of (8.1)

$$\begin{pmatrix} T_{ins}(0) \\ T_w(0) \\ T_{ins}(1) \\ T_w(1) \\ T_{ins}(2) \\ T_w(2) \\ P_H(0) \\ P_H(1) \\ P_H(2) \end{pmatrix} = \underbrace{\begin{pmatrix} 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&1 \end{pmatrix}}_{S_{9x9}} \begin{pmatrix} T_{ins}(1) \\ T_w(1) \\ T_{ins}(2) \\ T_w(2) \\ T_{ins}(3) \\ T_w(3) \\ P_H(0) \\ P_H(1) \\ P_H(2) \end{pmatrix} + \underbrace{\begin{pmatrix} T_{ins}(0) \\ T_{ins}(1) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}}_{c_{9x1}} \tag{8.2}$$

Plugging (8.2) in (8.1) gives (8.3)

$$\vec{x} = M_{9x9}S_{9x9}\ \vec{x} + b_{9x1} + c_{9x1} \tag{8.3}$$

Re-ordering (8.3) finally gives the full linear constraints equations used by a linear programming solver (8.4).

$$\underbrace{(I_{9x9} - M_{9x9}S_{9x9})}_{A_{sys}}\ \vec{x} = \underbrace{b_{9x1} + c_{9x1}}_{B_{sys}} \tag{8.4}$$

8.2 Python code explanations

Figure 8.2 recall the python program structure.

Runfile/userinterface: The program can be started either by running the file *run_nogui.py* or with the user interface. The file *simulation_config.py* allows to modify the simulation parameters such as the duration and time steps. This file starts the two processes: the EMS (*EMS_launcher*) and the simulator.

EMS_Strategy: Thread launched by *EMS_launcher*. In this class the strategy (MPC, thermostat, PI) is coded. This class heritates from *EnergyManagementSystem* class which contains many information and functions to interact with the building. *EMS_strategy* must overwrite the function update managed by the *EnergyManagementSystem* thread.

EnergyManagementSystem: Contains the function to set the battery and heater power and obtain the electricity price. The location of the heaters is also retrieved from this class. A *BuildingDataManagementSystem* object is created in this entity to obtain all the building data.

BuildingDataManagementSystem: Includes all the building entities (Rooms,

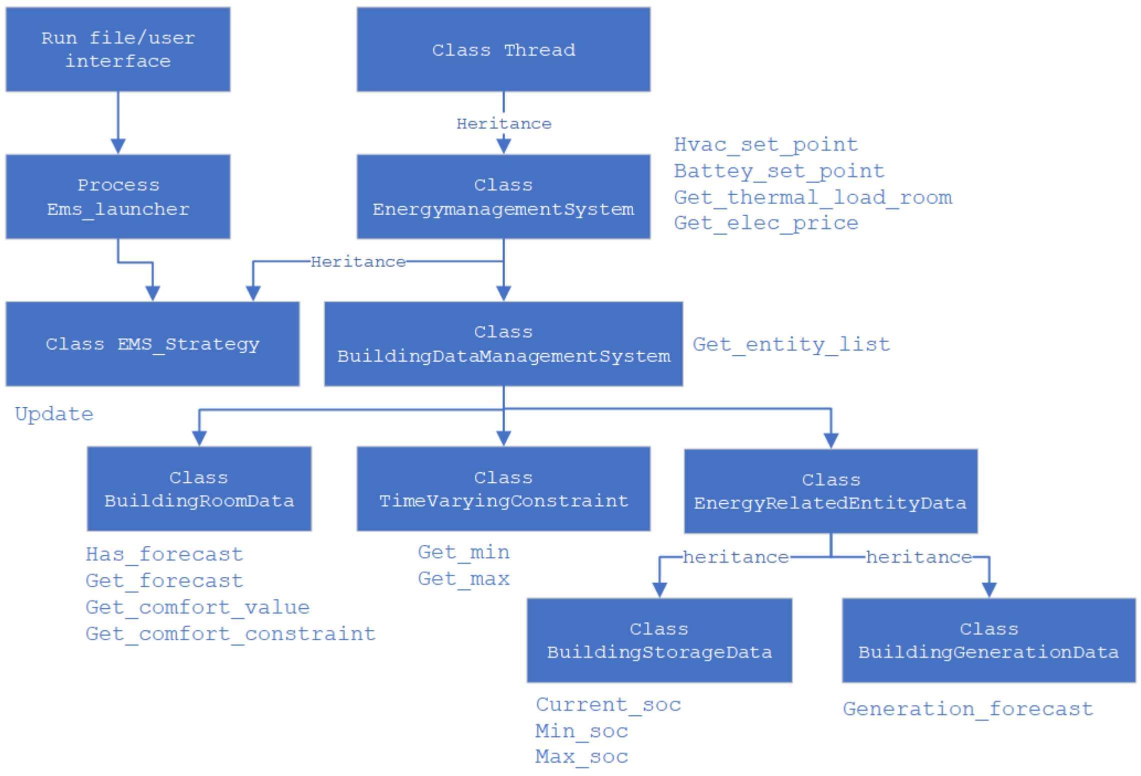


Figure 8.2: Program structure

heaters, battery, constraints). The function $get\_entity\_list("ENTITY\_NAME")$ returns every entity according to the parameters $ENTITY\_NAME$. It can be the loads(heaters), batteries or PV panels.

$BuildingRoomData$: Contains the data of the rooms. An object $BuildingRoomData$ contains the method $hasForcast(r\_id)$ which tells if a room (with ID r_id) is the outside. Indeed, the outside is considered as a room for which the temperature is known. This temperature is retrieved with $get\_forcast$ function. The $get\_comfort\_value$ method returns the current temperature of the $BuildingRoomData$ object. The function $get\_comfort\_constraint$ returns a $TimeVaryingConstraint$ object which is used to obtain the temperature boundaries.

$TimeVaryingConstraint$: The methods $get\_min(t)/get\_max(t)$ return the minimum and maximum temperature constraints at a given time t.

$EnergyRelatedEntityData$: This is the mother class of the two subclass $BuildingStorageData$ and $BuildingGenerationData$. The property $current\_power$ is used for any children of this class.

$BuildingStorageData$: Contains the data of the battery such as the current SOC, the minimum/maximum SOC authorized as well as the maximum power at a given SOC.

$BuildingGenerationData$: The method $generation\_forecast$ returns a list of power generation predicted for a given time and given irradiance.

Bibliography

[1] P. Palensky and D. Dietrich. Demand side management: Demand response, intelligent energy systems, and smart loads. *IEEE Transactions on Industrial Informatics*, 7(3):381–388, Aug 2011.

[2] Pierluigi Siano. Demand response and smart grids—a survey. *Renewable and Sustainable Energy Reviews*, 30(Supplement C):461 – 478, 2014.

[3] Bin Zhou, Wentao Li, Ka Wing Chan, Yijia Cao, Yonghong Kuang, Xi Liu, and Xiong Wang. Smart home energy management systems: Concept, configurations, and scheduling strategies. *Renewable and Sustainable Energy Reviews*, 61(Supplement C):30 – 40, 2016.

[4] Denis Gillet. *Multivariable Control and Coordination System, Discretization.* EPFL, October 2017.

[5] Gilbert M. Masters. *Renewable and efficient electric power system.* IEEE Press, 2013.