UNIVERSITY OF LIÈGE
SCHOOL OF ENGINEERING

Milestone 4

PROJ0016: Big Data Project

Julien GUSTIN, Romain CHARLES, Joachim HOUYON

March 8, 2022

1 Nomenclature

$$T[r,t] : \text{Air temperature of room r at time } t \quad [°C] \tag{1}$$
$$T_a[t] : \text{Ambient temperature (outdoor) at time } t \quad [°C] \tag{2}$$
$$T_w[t] : \text{Water temperature at time } t \quad [°C] \tag{3}$$
$$T_s[r,t] : \text{Temperature set} \quad [°C] \tag{4}$$
$$\Delta t : \text{Time step length (15min / 900s)} \quad [s] \tag{5}$$
$$t : \text{Time step} \tag{6}$$
$$R_r : \text{thermal resistance (for thermal loss)} \quad [°C/W] \tag{7}$$
$$R_{r,r'} : \text{Thermal resistors coupling the rooms } r \text{ and } n \quad [°C/W] \tag{8}$$
$$C_r : \text{thermal capacities (Heat capacity of the building)} \quad [J/°C] \tag{9}$$
$$m_r : \text{Mass flow rate} \quad [kg/s] \tag{10}$$
$$k_r : \text{Thermal loss coefficient} \quad [\%] \tag{11}$$
$$Q_{heat}[r,t] : \text{Heat transfer from the radiator to the room } r \quad [W] \tag{12}$$
$$c_p : \text{Heat capacity of water (4810)} \quad [kJ/kg/°C] \tag{13}$$
$$a_z : \text{Offset parameter - can be seen as internal heat gains} \quad ([W]) \tag{14}$$
$$irradiance[t] : \text{Solar irradiance at time } t \quad [W/5m^2]) \tag{15}$$
$$occupancy[t] : \text{Whether or not there are people on the house} \quad [0,\ 1] \tag{16}$$
$$area[r] : \text{Area of room } r \quad [m^2] \tag{17}$$

$\forall r \in$ Rooms and $\forall r' \in$ ClusterOf(r)

2 Description of the model

Since the last milestone we have updated our model based on the dataset such that we now take into account the occupancy and the solar irradiance. Indeed, these two variables inform us when there are people in the house and the second one how the house is irridiate by the sun. Considering them improved by a lot our model accuracy.

We could also have used the temperature of the ground but as in the dataset it is a fixed value we thought that it is not really usefull to incorpore it into our model.

$$T[r,t+1] = \left(\frac{Ta[t] - T[r,t]}{R_r * C_r} + \frac{Q[r,t]}{C_r} - \sum_{r' \in ClusterOf(r)} \frac{T[r,t] - T[r',t]}{R_{rr'} * C_r} \right) * \Delta t + T[r,t] \quad (18)$$

Where

$$\begin{aligned} Q[r,t] &= Q_{int}[r,t] + Q_{sol}[r,t] + Q_{heat}[r,t] \\ Q_{int}[r,t] &= a_r * occupancy[t] \\ Q_{sol}[r,t] &= area[r] * irradiance[t] * gs_r \\ Q_{heat}[r,t] &= cp * m_r * (Tr[r,t] - T[r,t]) \end{aligned}$$

$$Tr[r,t+1] = \begin{cases} max(T[r,t+1], Tw[t+1]) & \text{if radiator on} \\ max(T[r,t+1],\ Tr[t] - \Delta t * k[r] * Q_{heat}[r,t]) & \text{if radiator off} \end{cases} \quad (19)$$

$\forall r \in Rooms$ and $\forall t \in T > 0$, initials (i.e. when t=0) values of variable are set to 0.

R_r, C_r, $R_{rr'}$, a_r, gs_r, m_r and k_r are unknown parameters that must me optimized according to the dataset.

The $max(.,.)$ in equations 19 mean that the temperature of the radiator should at least be at ambient temperature.

Little remark on clusters, for simplicity we decided to keep the set of cluster[1] that we had from previous milestone, meaning:

1. Dining, Kitchen, Room1, Bathroom
2. Room2, Room3
3. Living

2.1 Parameter fitting procedure

Concerning the parameter fitting procedure we kept exactly what we did from previous milestones but since the dataset is not as large as before we decided to launch 10 batch of 500 optimization on alan cluster[2] with random initial values of a certain range for each parameter, and finally use the parameters that give the best sum of RMSE across room comparing the true value of the room temperature and the one that we predict starting with $T[r, t = 0]$ and using the dataset for other values. This give us a total RMSE of 6.6871°C thus 0.9545°C per room on average.

[1] A cluster is a set of room that are thermally dependent

[2] https://github.com/montefiore-ai/alan-cluster

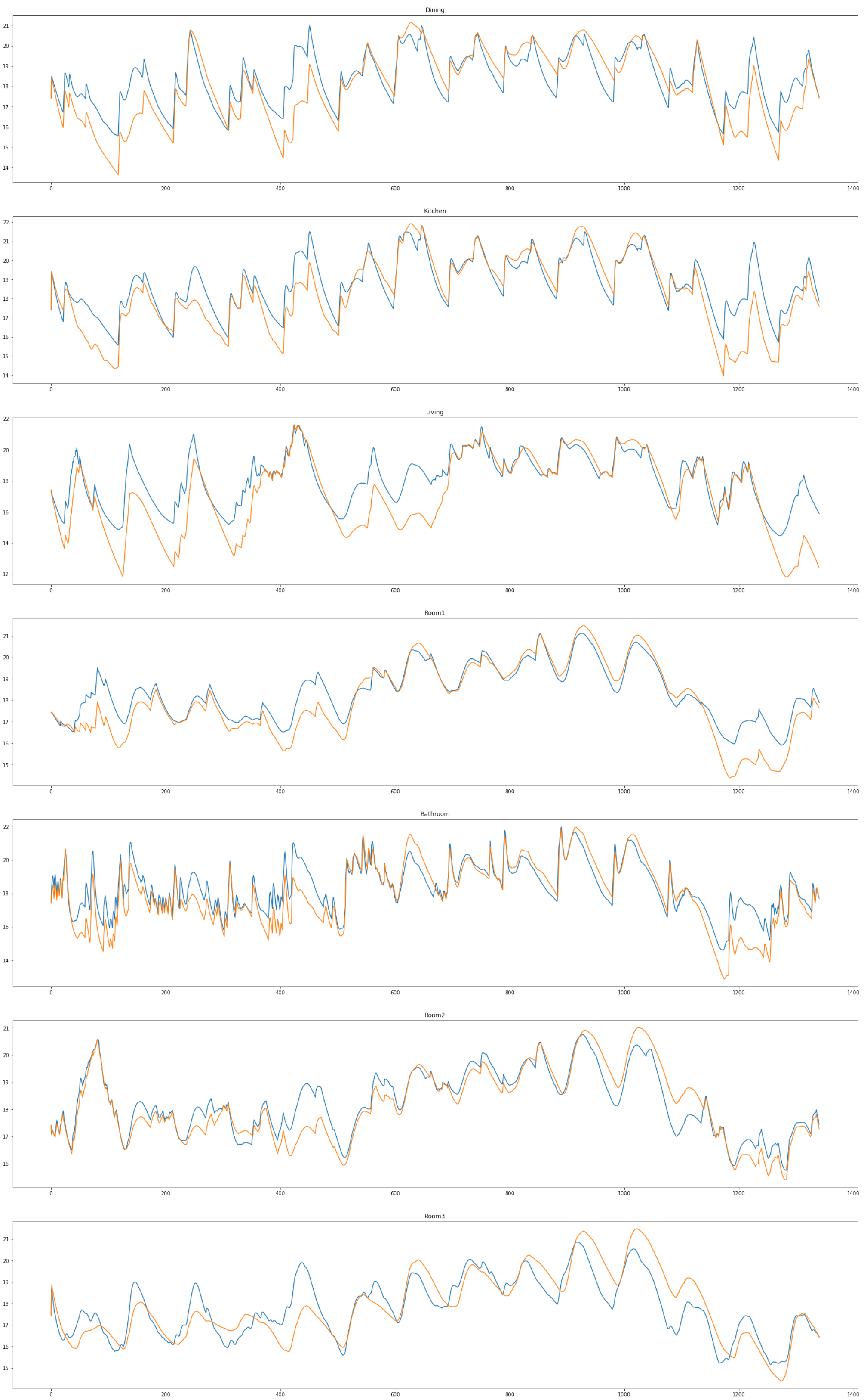


Figure 1. Result of the parameter fitting of our model. In blue is the ground truth and in orange the predicted temperature

This overall result is important in the sense that it allow us to see if our model is physically coherent, but as we will only use the few firsts value of each prediction for the controller it is as much important to see if the small variation are well taken into account.

3 Model predictive control

The goal is to formulate an optimization problem to control the boiler and heaters operations to minimize the difference between the room temperature and its setpoint temperature (model predictive control).

For that purpose we had to translate our model into a set of constraint, the tricky part was to deal with binary variable. After some research[3] we found a way to enforce binary variable using the big M trick. By using that trick we were able to translate the $if/else$ conditions and the $max(.)$ function of **Equation.19** into a set of constraint.

What we name the discomfort is the absolute difference of the set temperature and the temperature of the zone. Doing it naively would lead to a non-linear objective function, to solve that issue we used the epigraph trick of that function.

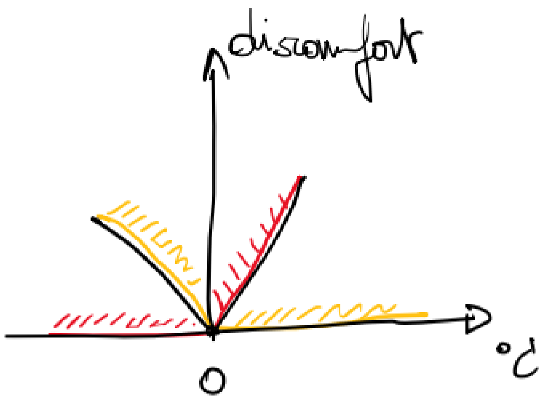


Figure 2. Discomfort

3.1 Optimisation problem

$$\begin{aligned}
\min_{Switch, Tw} \quad & \sum_{t=1}^{T}(\epsilon Tw[t] + \sum_{r \in Rooms}(\gamma|T[r,t] - T_s[r,t]| + \alpha Q_{heat}[r,t])) \\
s.t. \quad & \\
T[r,t+1] \quad & = Model(t) \quad (Eq.18) \\
Tr[r,t] \quad & = max(T[r,t], Switch[r,t]T_w[t] + (1 - Switch[r,t])Loss[r,t]) \\
Loss[r,t+1] \quad & = T_r[r,t] - k_r Q_{heat}[r,t]\Delta t \\
Q_{heat}[r,t] \quad & = cp * m_r * (Tr[r,t] - T[r,t]) \\
\forall t \in T, r \in Rooms \quad &
\end{aligned}$$

where $Switch \in [0,1]^{\#Rooms \times T}$. This formulation is however non-linear because the objective function contains an absolute value, and the radiator constraint is a mixed-integer constraint with also a max function, which is not linear.

γ and α are hyperparameters that must be tuned in order to address the tradeoff between the discomfort and the energy consumption. The presence of Tw with $\epsilon = 10^{-6}$ in the objective function is only to get $Tw[t] = 0$ when all the switches are off.

[3] https://math.stackexchange.com/questions/2500415/how-to-write-if-else-statement-in-linear-programming

3.2 Linearization

The objective function is non-linear because of the absolute value. We can apply the epigraph trick: it will replace $|T[r,t] - T_s[r,t]|$ by a new variable $Discomfort[r,t]$ and two additional linear constraints $\forall r \in Rooms, t \in T$:

$$\begin{aligned} Discomfort[r,t+1] &\geq \omega(T[r,t+1] - Ts[r,t+1]) \\ Discomfort[r,t+1] &\geq \beta(Ts[r,t+1] - T[r,t+1]) \end{aligned}$$

The radiator constraint implies the multiplication between a binary and a continuous variable, which can be linearized using a big M constraint:

$$\begin{aligned} T_r[r,t+1] &\leq max(T[r,t+1], Loss[r,t+1]) + M(Switch[r,t+1]) \\ T_r[r,t+1] &\geq max(T[r,t+1], Loss[r,t+1]) - M(Switch[r,t+1]) \\ T_r[r,t+1] &\leq T_w[t+1] + M(1 - Switch[r,t+1]) \\ T_r[r,t+1] &\geq T_w[t+1] - M(1 - Switch[r,t+1]) \end{aligned}$$

Where $M = 100$ is a satisfying upper-bound over the constraints, as we are not expected to deal with temperatures going over 70 degrees. The two first constraints of the linearization are not linear yet. Indeed, we need to deal with the max:

Let $b \in [0,1]^{\#Rooms \times T}$ such that we want $b[r,t]$ to be equal to 1 if and only if $T[r,t] \geq Loss[r,t]$. Associated with the switch, we can replace the constraints implying the max() using a big M constraint:

$$\begin{aligned} T_r[r,t+1] &\geq T[r,t+1] \\ T_r[r,t+1] &\leq T[r,t+1] + M(Switch[r,t+1] + 1 - b[r,t+1]) \\ T_r[r,t+1] &\leq Loss[r,t+1] + M(Switch[r,t+1] + b[r,t+1] \\ T_r[r,t+1] &\geq Loss[r,t+1] - M(Switch[r,t+1]) \end{aligned}$$

Given $Switch[r,t+1] = 0$, the first two constraints ensure that $Tr[r,t+1] = T[r,t+1]$ if $b[r,t+1] = 1$ while the two lasts ensure that $Tr[r,t+1] = Loss[r,t+1]$ if $b[r,t+1] = 0$.

All these tricks lead us to a linear optimization problem, which is convenient because solving these problems are computationally efficient, and it ensures that the problem is convex.

Still, the issue is the computational time, indeed, we can hardly go beyond a look ahead of three step. However, it stills give satisfying results.

3.3 Local tests

Before pushing our model into gradescope we needed to see how our model performed locally, debug and play with hyper-parameters.

The way we simulated was by using a portion of the data set for the forecast of occupancy, outside temperature, ... and using as previous temperature the one computed by our model. This way to simulate may not be the best one as it may be overoptimistic as we consider that our model can predict with fully accuracy the temperature of each room at the preceding time-step. However it is still a great tool for debugging and see how our model behave.

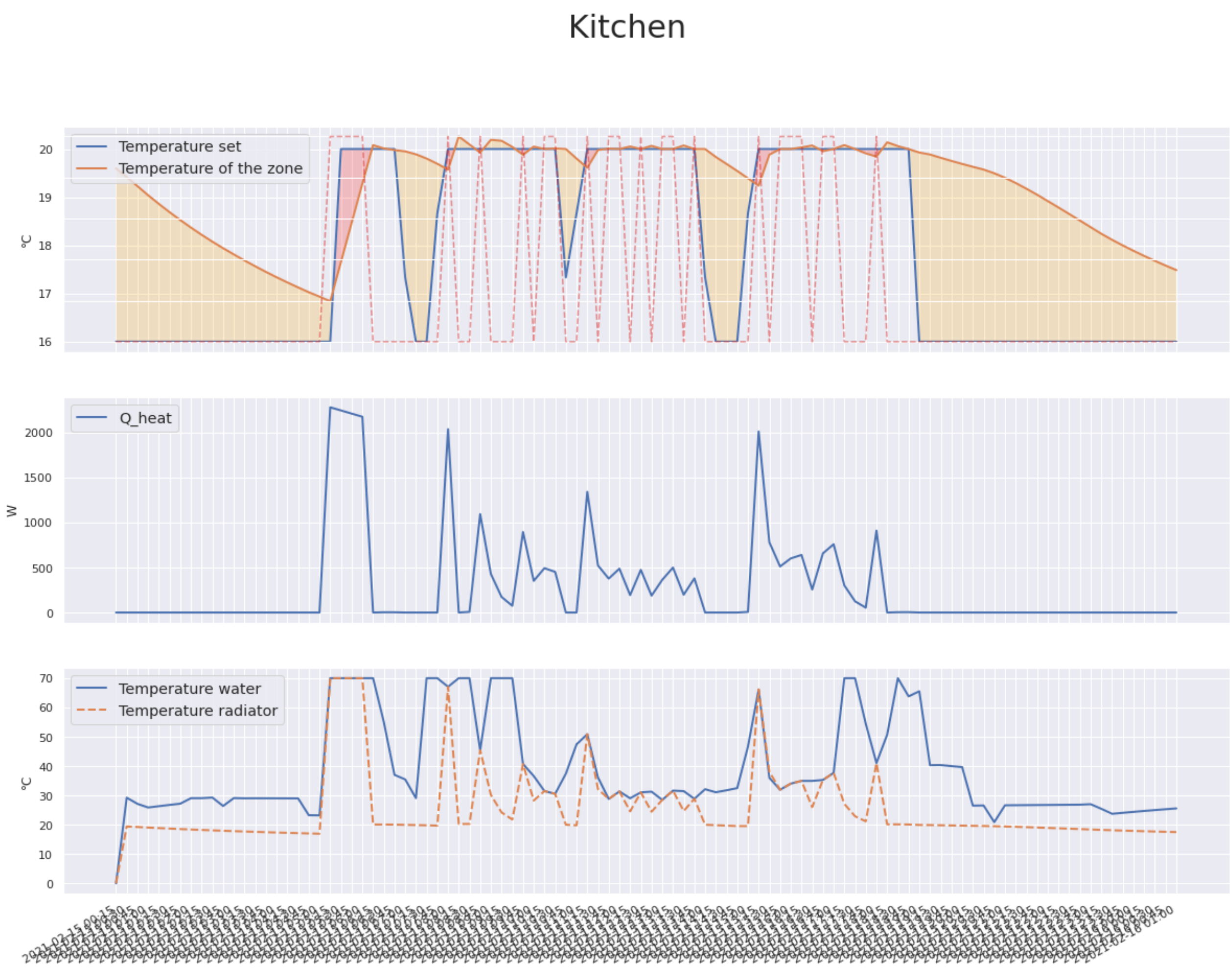
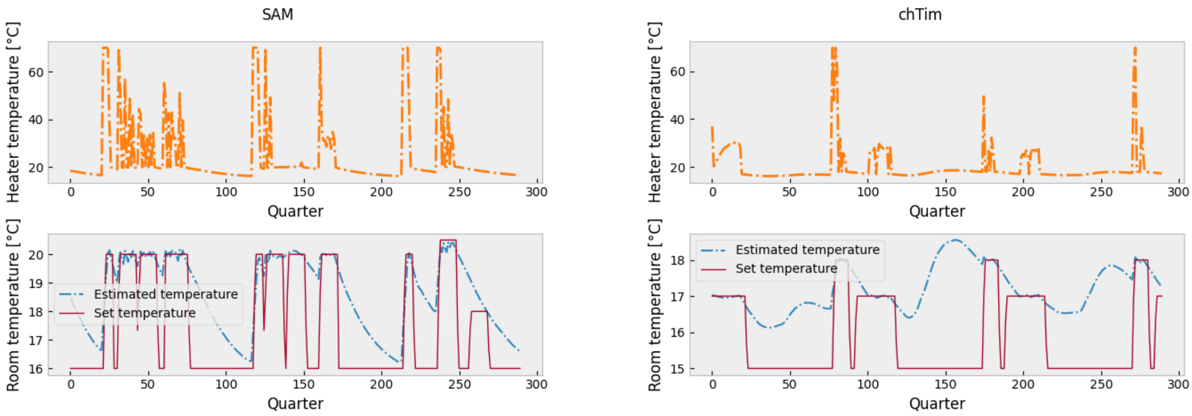


Figure 3. Simulation of the model behavior on the Kitchen from 15/02/2021 at 00:15 to 16/02/2021 at 01:00, using three time step horizon.

The above example is quite interesting because we can see that a bit before the temperature is below the set point our model set the switch to ON in order to increase the temperature of the room. Then, try to keep the room temperature as close as possible to the set-point.

4 Results

Below are presented the results of our model on two different rooms: the dining room and one of the bedrooms.



(a) Dining room

(b) Hum...Tim's bedroom?

Figure 4. Model predictive control applied to the rooms with a look ahead of three step. With an accuracy score of 91.418 / 100 on Gradescope

Concerning the dining room, we can notice that the estimated temperature of the model is doing well at reaching most of the temperatures set points. Indeed, when the next set-point is higher than the estimated temperature, the model turns up the radiator switch and sets the boiler to the necessary number of degrees. Once the set-point is reached, the model maintains its temperature to the set-point. However, the model struggles at reaching set-points that are low compared to the current set-point. Indeed, we have the example between Quarter 250 and 270 of **Fig.**4a, where it fails at reaching the set-point on time. This is due to the physics of the radiator.

Regarding the Tim's bedroom **Fig.**4b, our model seems to perform very poorly. The simple explanation is that the room's temperature is mainly controlled by the outside temperature (and maybe other outliers), for which we have no control.

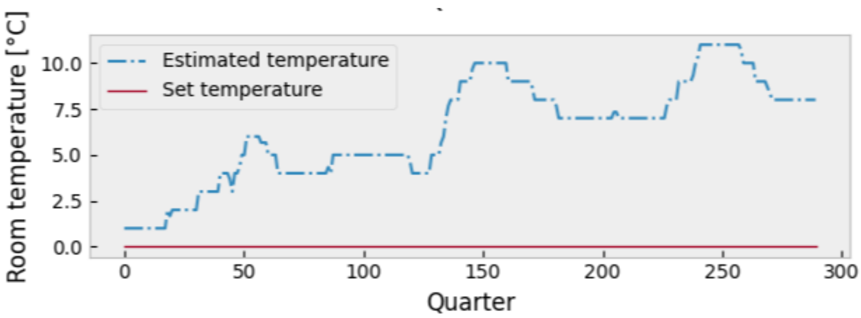


Figure 5. Outside temperature, we can see a high correlation with Tim's bedroom temperature

If the house was equipped with a cooling system, we would be able to address the loss and the outside temperature problems because we have no control over them and they are a problem for reaching a better comfort. We could be able to model a cooling system and use it in order to reach lower temperatures.

5 Conclusion

In conclusion we are quite proud of what we have done and it is fascinating to see that with just a data set and some knowledge about physics, we can end up with a functional model that can be used in real life.

In the end we have learned a lot of things with this project even if we had never heard about heat transfer and thermal physics before.

5.1 Improvements

Our model could be subject to some improvements:

1. **Re-arrange the clusters:** We did not try to find the new best combination of rooms, for simplicity we kept clusters as previously computed with the previous dataset. Re-arranging the cluster could improve the model predictions.
2. **Model a cooling system:** If the house choose to add a cooling system, this one could be modeled in order to address the issue regarding the no control over the loss of the radiator and the external temperatures.
3. **Using reinforcement learning**: Instead of using optimization in order to control the switch and boiler, we could use reinforcement learning. This would allow us to increase the look ahead horizon which may improve our results.