Optimization in Systems and Control: Quadratic Programming Assignment

Meric Erceylan (4305558) Taeke de Haan (4316568), Group 75

October 16, 2018

1. The system dynamics are described with the following differential equation

$$\frac{dT(t)}{dt} = a_1\left(T^{amb}(t) - T(t)\right) + a_2\left(\dot{Q}^{in}(t) - \dot{Q}^{out}(t)\right) \tag{1}$$

This is transformed into a linear discrete-time version by using the following approximation

$$\frac{dT_k}{dt} \approx \frac{T_{k+1} - T_k}{\Delta t} \tag{2}$$

With this we get

$$\begin{aligned} \frac{T_{k+1} - T_k}{\Delta t} &= a_1\left(T_k^{amb} - T_k)\right) + a_2\left(\dot{Q}_k^{in} - \dot{Q}_k^{out}\right) \\ T_{k+1} &= a_1\Delta t\left(T_k^{amb} - T_k\right) + T_k + a_2\Delta t\left(\dot{Q}_k^{in} - \dot{Q}_k^{out}\right) \\ T_{k+1} &= \left(1 - a_1\Delta t\right)T_k + \begin{bmatrix}-a_2\Delta t & a_2\Delta t\end{bmatrix}\begin{bmatrix}\dot{Q}_k^{out} \\ \dot{Q}_k^{in}\end{bmatrix} + a_1\Delta t T_k^{amb} \end{aligned} \tag{3}$$

Thus we can see that the equation may be written as

$$T_{k+1} = AT_k + B\begin{bmatrix}\dot{Q}_k^{out} \\ \dot{Q}_k^{in}\end{bmatrix} + c_k \tag{4}$$

Where

$$\begin{aligned} A &= 1 - a_1\Delta t \\ B &= \begin{bmatrix}-a_2\Delta t & a_2\Delta t\end{bmatrix} \\ c_k &= a_1\Delta t T_k^{amb} \end{aligned} \tag{5}$$

2. Parameters a_1 and a_2 are determined using the following quadratic optimization problem

$$\min_{a_1,a_2} \sum_{k=1}^{100+E_1}\left(\bar{T}_{k+1} - \left(A\bar{T}_k + B\begin{bmatrix}\dot{Q}_k^{out} \\ \dot{Q}_k^{in}\end{bmatrix} + c_k\right)\right)^2 \tag{6}$$

To solve this problem it needs to be rewritten into the quadratic problem normal form

$$\min_x y(x) \tag{7}$$

Where

$$y(x) = \frac{1}{2}x^T Hx + f^T x \tag{8}$$

we can see that H simply is the hessian of the cost function

$$H = \frac{\partial^2 y}{\partial x^2} \tag{9}$$

Furthermore we can determine f with

$$f^T = \frac{\partial y}{\partial x} - x^T H \tag{10}$$

Both f and H can now easily be computed using MATLAB's symbolic toolbox, and solved using the quadratic programming solver, this results in;

- $a_1 = 2.0786 \cdot 10^{-7}$
- $a_2 = 3.7853 \cdot 10^{-9}$

3. The following problem is given

$$\min \sum_{k=1}^{N} \lambda_k^{in} \dot{Q}_k^{in} \Delta t \tag{11}$$

$$\text{s.t. } T_{k+1} = AT_k + B[\dot{Q}_k^{out}, \dot{Q}_k^{in}]^T + c_k \quad k = 1, ..., N \tag{12}$$

$$0 \leq \dot{Q}_k^{in} \leq Q_{max}^{in} \quad k = 1, ..., N \tag{13}$$

$$T^{min} \leq T_k \quad k = 2, ..., N+1 \tag{14}$$

Where

- λ_k price of energy in [€/J], note that 1 [€/J] = $3.6 \cdot 10^9$ [€/MWh]
- Δt time interval of sequential measurements, 3600 [s]
- $\dot{Q}_k^{in}$ input energy in [W]
- $\dot{Q}_k^{out}$ output energy in [W]
- T_k temperature in [K]

a) We can see that there are no quadratic terms in the cost function. Therefore this problem is linear, and con be solved using a linear programming solver.

b) The energy price λ_k is given in €/MWh, we can see that this needs to be converted to €/J to make this data compatible with the equations. Since 1 MWh equals $3.6 \cdot 10^9 J$ we simply need to apply

$$\lambda_k^{new} = \lambda_k^{old}[euro/MWh] \cdot \frac{1}{3.6 \cdot 10^9} \frac{[MWh]}{[J]} = \frac{\lambda_k^{old}}{3.6 \cdot 10^9} \frac{[euro]}{[J]} \tag{15}$$

c) To solve this linear problem we will use

$$x = \left[\dot{Q}_1^{in}, ..., \dot{Q}_N^{in}, \dot{T}_1^{in}, ..., \dot{T}_{N+1}^{in}\right]^T \tag{16}$$

We can clearly see that constraint 13 and 14 can be enforced by putting

$$lb \leq x \leq ub \tag{17}$$

Where:

$$\begin{aligned} lb^T &= [0, ..., 0, T_1, T^{min}, ..., T^{min}]^T \\ ub^T &= [Q_{max}^{in}, ..., Q_{max}^{in}, T_1, \infty, ..., \infty]^T \end{aligned} \tag{18}$$

The initial temperature is assigned to T_k by assigning T_1 as both an upper and lower boundary. Now we only need to implement the first constraint

$$\begin{aligned} T_{k+1} &= AT_k + B[\dot{Q}_k^{out}, \dot{Q}_k^{in}]^T + c_k \\ T_{k+1} - AT_k - B_2\dot{Q}_k^{in} &= B_1\dot{Q}_k^{out} + c_k \end{aligned} \tag{19}$$

Which is written into matrix form

$$A_{eq}x = b_{eq} \tag{20}$$

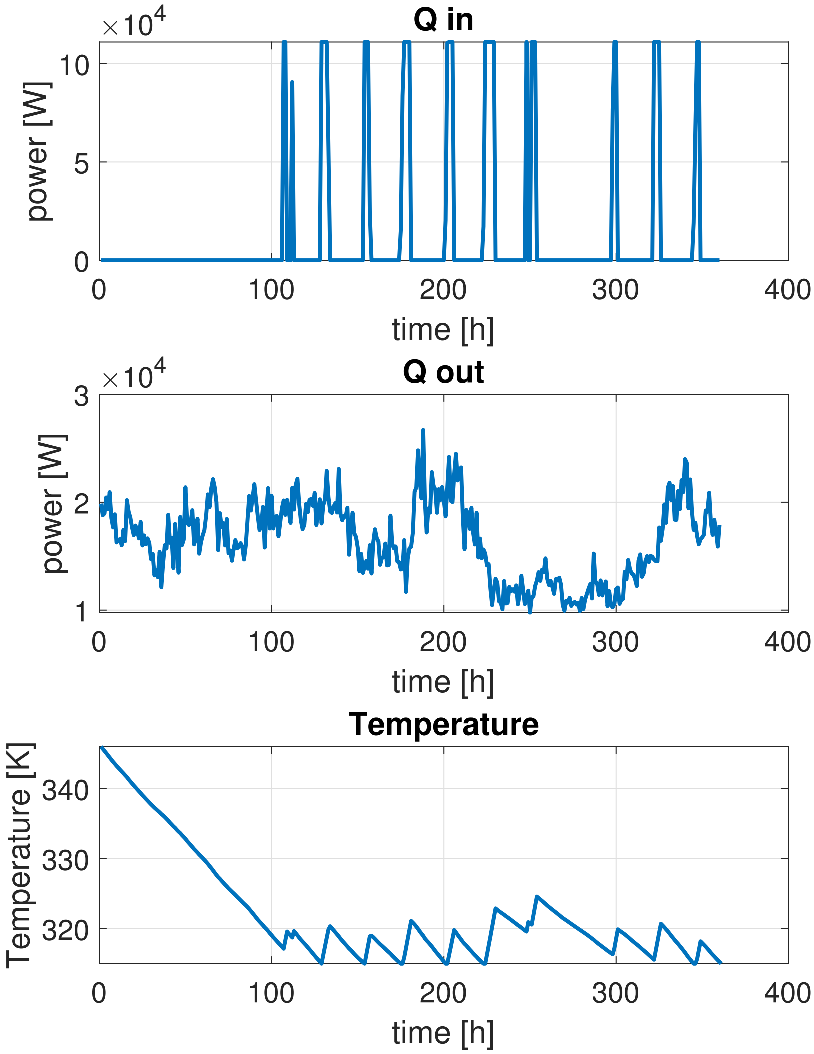

Figure 1: Optimal buying strategy of input energy

Where

$$A_{eq} = \begin{bmatrix} -B_2 & 0 & \dots & 0 & -A & 1 & 0 & \dots & 0 & 0 \\ 0 & -B_2 & \dots & 0 & 0 & -A & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & -B_2 & 0 & 0 & 0 & \dots & -A & 1 \end{bmatrix}$$

$$b_{eq} = \begin{bmatrix} B_1 \dot{Q}_1^{out} + c_1 \\ B_1 \dot{Q}_2^{out} + c_2 \\ \vdots \\ B_1 \dot{Q}_N^{in} + c_N \\ B_1 \dot{Q}_N^{in} + c_N \end{bmatrix} \tag{21}$$

Thus we can finally see that the problem can be formulated as

$$\min_x f^T x \begin{cases} A_{eq} x = b_{eq} \\ lb \leq x \leq ub \end{cases} \tag{22}$$

Which is solved using MATLAB's linear programing solver, the optimal buying strategy of input energy is shown in figure 1. We can compute the final price with

$$p = \sum_{k=1}^{N} \lambda_k^{in} \dot{Q}_k^{in} \Delta t \tag{23}$$

Which results in a total cost of 110.5426 €

4. The situation slightly changes
   - An upper boundary is added to the temperature in the tank $T^{max} = 368[K]$.
   - An extra cost is added which is equals to a certain constant multiplied with the squared error between the final temperature T_{N+1} and the reference temperature $T_{ref} = 323K$.

   a) We can see that the change in cost function makes this a quadratic problem which can be formulated as

$$\min \sum_{k=1}^{N} \lambda_k^{in} \dot{Q}_k^{in} \Delta t + C\left(T_{N+1} - T_{ref}\right)^2 \tag{24}$$

$$\text{s.t. } T_{k+1} = AT_k + B[\dot{Q}_k^{out}, \dot{Q}_k^{in}]^T + c_k \quad k = 1, ..., N \tag{25}$$

$$0 \leq \dot{Q}_k^{in} \leq Q_{max}^{in} \quad k = 1, ..., N \tag{26}$$

$$T^{min} \leq T_k \leq T^{max} \quad k = 2, ..., N+1 \tag{27}$$

Where:

- $C = \left(0.1 + \frac{E_2}{10}\right)[€/\text{K}^2]$ is a constant used for the cost

The cost function may be reformulated as

$$\begin{aligned}
&\min_x \sum_{k=1}^{N} \lambda_k^{in} \dot{Q}_k^{in} \Delta t + C\left(T_{N+1} - T_{ref}\right)^2 \\
&\min_x \sum_{k=1}^{N} \lambda_k^{in} \dot{Q}_k^{in} \Delta t + C\left(T_{N+1}^2 - 2T_{N+1}T_{ref} + T_{ref}^2\right) \\
&\min_x \sum_{k=1}^{N} \lambda_k^{in} \dot{Q}_k^{in} \Delta t + CT_{N+1}^2 - 2CT_{N+1}T_{ref}
\end{aligned} \tag{28}$$

Note that the term CT_{ref}^2 is left out of the equation since it does not influence the location of the minimum. This problem can now be rewritten in a quadratic optimization problem

$$\min_x \frac{1}{2} x^T H x + f^T x \begin{cases} A_{eq} x = b_{eq} \\ lb \leq x \leq ub \end{cases} \tag{29}$$

Where

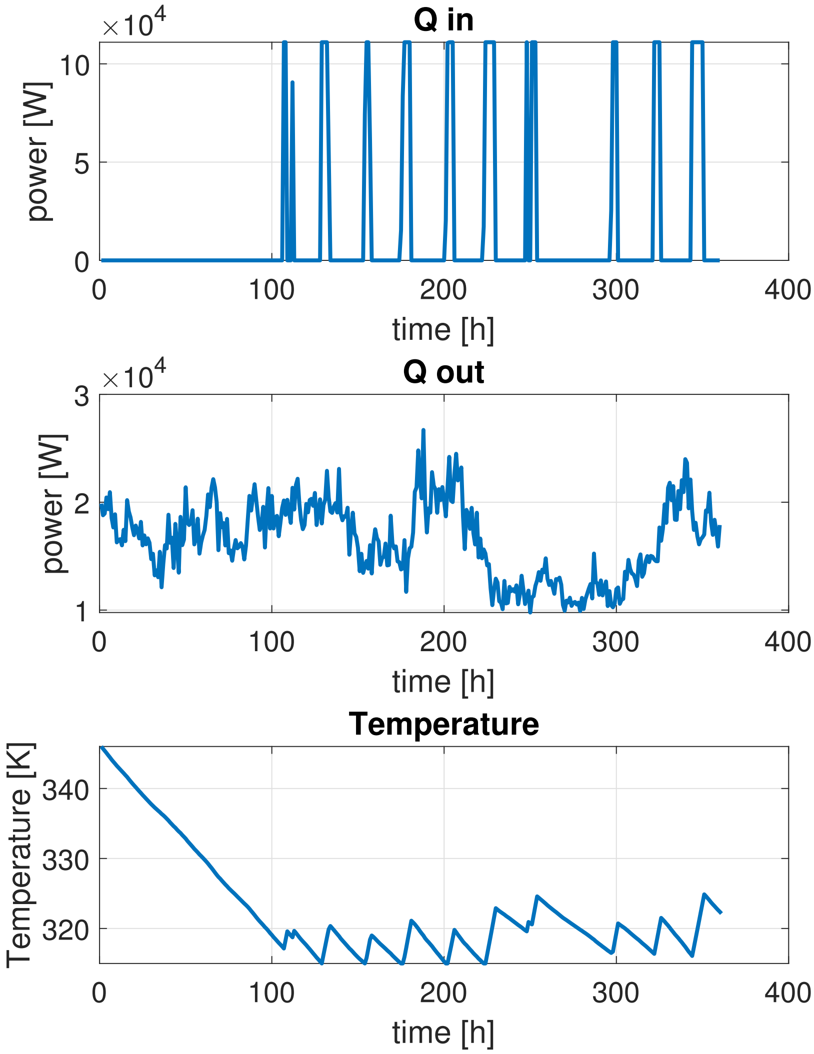

Figure 2: result

$$H = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & - & \dots & 2C \end{bmatrix}, f = \begin{bmatrix} \Delta t \lambda_1 \\ \vdots \\ \Delta t \lambda_N \\ 0 \\ \vdots \\ -2CT^{ref} \end{bmatrix} \tag{30}$$

$$lb^T = [0, \dots, 0, T_1, T^{min}, \dots, T^{min}]^T \tag{31}$$

$$ub^T = [Q^{in}_{max}, \dots, Q^{in}_{max}, T_1, T^{max}, \dots, T^{max}]^T \tag{32}$$

The Matrices A_{eq} and b_{eq} remain unchanged since the overall system dynamics are not influenced.

b) This problem is solved using MATLAB's quadratic programming solver. The optimal power strategy is shown in figure 2. The overall cost equals 126.0078€. Which is slightly higher than our previous result. The terminal cost only accounts for 0.90353€of the total cost.