Quadratic Programming

Optimization in Systems and Control

SC42055

17-10-2018

J.J. Lont | 4424409
C.C.J. van Engelenburg | 4237080

Delft University of Technology

List of symbols

Descriptions of all parameters used are succinctly summarized in the table below.

Table 1: Parameters used to describe the optimization problem

Parameter description	Parameter	Value	Unit
LP problem-defining vector	f	[-]	[-]
State vector	x	[-]	[-]
Terminal cost price	c_N	0.9	$\frac{€}{K^2}$
QP problem-defining matrix Q2	H	[-]	[-]
QP problem-defining matrix Q2	H_c	[-]	[-]
QP problem-defining vector Q4	c	[-]	[-]
QP problem-defining vector Q4	c_c	[-]	[-]
Normal-form constant	c_k	[-]	[-]
Terminal cost	c_N	0.9	$\frac{€}{K^2}$
Equality constraint matrix	A_{eq}	[-]	[-]
Equality constraint vector	b_{eq}	[-]	[-]
Internal tank temperature	T	[-]	K
Minimum internal tank temperature	T^{min}	[-]	K
Maximum internal tank temperature	T^{max}	368	K
Ambient temperature	T^{amb}	[-]	K
Reference temperature	T^{ref}	323	K
System parameter	a_1	[-]	$\frac{1}{s}$
System parameter	a_2	[-]	$\frac{K}{J}$
Input heat to the tank	$\dot{Q}^{in}$	[-]	W
Maximum input heat to the tank	$\dot{Q}^{in}_{max}$	[-]	W
Output heat from the tank	$\dot{Q}^{out}$	[-]	W
Time-step	Δt	3600	s
Discrete-time system matrix	A	[-]	[-]
Discrete-time system matrix	B	[-]	$\frac{K}{W}$
Group-unique parameter	E1	4	[-]
Group-unique parameter	E2	8	[-]
Group-unique parameter	E3	9	[-]
Price per unit of input heat	λ^{in}_k	[-]	$\frac{€}{MWh}$
Hourly horizon	N	360	hours

Table 2: Acronyms

LP	Linear Programming
QP	Quadratic Programming

1
Discretization of the problem

The system dynamics of every particular heat tank can be described using the following ordinary differential equation:

$$\frac{dT(t)}{dt} = a_1 \cdot (T^{amb}(t) - T(t)) + a_2 \cdot (\dot{Q}^{in}(t) - \dot{Q}^{out}(t)) \tag{1.1}$$

where a_1 and a_2 are the system parameters, and where $\dot{Q}^{in}(t), \dot{Q}^{out}(t) > 0$ are the in and ouput heats. Since these system parameters are unknown an option to estimate or identify them is to use system identification methods. To make this possible we preferably need much where all the temperatures and in/out-put heats are measured over a horizon. Since measuring is not instantaneously but sampled at a specified frequency, we have to transform the (continuous time) ODE to a discrete time formulation. The key ingredient to do so is the well known approximation for the discretization of a derivative (in this case specified for the temperature in the tank:

$$\frac{dT_k}{dt} \approx \frac{T_{k+1} - T_k}{\Delta t} \tag{1.2}$$

We can transform 1.1 into:

$$\begin{aligned} \frac{T_{k+1} - T_k}{\Delta t} &\approx a_1 \cdot \left(T_k^{amb} - T_k\right) + a_2 \cdot \left(\dot{Q}_k^{in} - \dot{Q}_k^{out}\right) \\ \Longleftrightarrow T_{k+1} &= T_k + \Delta t \cdot \left(a_1 \cdot \left(T_k^{amb} - T_k\right) + a_2 \cdot \left(\dot{Q}_k^{in} - \dot{Q}_k^{out}\right)\right) \\ &= (1 - \Delta t \cdot a_1) \cdot T_k + \Delta t \cdot a_2 \cdot (\dot{Q}_k^{in} - \dot{Q}_k^{out}) + \Delta t \cdot a_1 \cdot T_k^{amb} \\ &= A \cdot T_k + B \cdot \begin{bmatrix} \dot{Q}_k^{in} \\ \dot{Q}_k^{out} \end{bmatrix} + c_k \end{aligned} \tag{1.3}$$

With the following parameters,

$$A = (1 - \Delta t \cdot a_1), \qquad B = \Delta t \cdot a_2 \cdot \begin{bmatrix} 1 & -1 \end{bmatrix}, \qquad c_k = \Delta t \cdot a_1 \cdot T_k^{amb} \tag{1.4}$$

We are now able to use system identification methods. In the next chapter we aim to identify the parameters using the minimization of the Euclidian norm of the difference between the measured T_{k+1} and the measured $A \cdot T_k + B \cdot [\dot{Q}_k^{out}, \dot{Q}_k^{in}]^T + c_k$ in which the unknown model parameters are the arguments in the minimization.

2
Parameter identification

The discrete-time version of the dynamics of the heat tank is found in equation **??**. The model parameters (a_1 and a_2) will be estimated using the following Euclidean norm minimization:

$$\boxed{\underset{a_1,a_2}{\text{minimize}} \quad \sum_{k=1}^{100+E_1} \left(\overline{T}_{k+1} - \left((1-a_1\Delta t)\,\overline{T}_{k+1} + a_2\Delta t\,[1 \quad -1]\left[\overline{\dot{Q}_k^{in}} \quad \overline{\dot{Q}_k^{out}}\right]^T + a_1\Delta t\overline{T}_k^{amb}\right)\right)^2} \tag{2.1}$$

where all quantities come from data and are therefore in bar notation, $\overline{(\cdot)}$. Observe that the only unknown variables are indeed $a = [a_1 \quad a_2]^T$. The square root in the Euclidian norm is removed since it will not contribute to the solution. Mathematically the prove is obvious and yields that

$$\underset{x}{\text{minimize}} \quad \sqrt{\sum_i f(x_i)^2} = \underset{x}{\text{minimize}} \quad \sum_i f(x_i)^2 \tag{2.2}$$

where $f(x)$ is an any arbitrary function which is linear in x. The hypothesis is that 2.1 is a quadratic function in a since we expect a_1^2, a_1a_2 and a_2^2 terms due to the square. Let us investigate if such terms will indeed not vanish by expanding the whole square in the minimization problem formulation. For convenience reasons, we simplify all quantities by removing their bar notation and introduce a new compact quantity,

$$Q_{B,k} := [1 \quad -1]\left[\overline{\dot{Q}_k^{in}} \quad \overline{\dot{Q}_k^{out}}\right]^T \tag{2.3}$$

which clearly represents a scalar. We can now expand the part of interest (within the summation) in 2.1 to obtain,

$$\begin{aligned}
(\cdot)^2 = {} & T_{k+1}^2 - T_{k+1}T_k + T_k^2 && \text{1-term}\\
& + a_1^2\left(T_k^2 + \left(T_k^{amb}\right)^2 - 2T_kT_k^{amb}\right)\Delta t^2 && a_1^2\text{-term}\\
& + a_1\left(2T_{k+1}T_k - 2T_{k+1}T_k^{amb} - 2T_k^2 + 2T_kT_k^{amb}\right)\Delta t && a_1\text{-term}\\
& + a_1a_2\left(-2T_kQ_{B,k} + 2T_k^{amb}Q_{B,k}\right)\Delta t^2 && a_1a_2\text{-term}\\
& + a_2\left(-2T_{k+1}Q_{B,k} + 2T_kQ_{B,k}\right)\Delta t && a_2\text{-term}\\
& + a_2^2\left(Q_{B,k}^2\right)\Delta t^2 && a_2^2\text{-term}
\end{aligned} \tag{2.4}$$

We can now concise it in a quadratic fashion in a and obtain

$$\begin{aligned}
(\cdot)^2 = {} & \underbrace{\frac{1}{2}\begin{bmatrix} a_1 \\ a_2 \end{bmatrix}^T \begin{bmatrix} 2(T_k - T_k^{amb})^2\Delta t^2 & -(T_k - T_k^{amb})Q_{B,k}\Delta t^2 \\ -(T_k - T_k^{amb})Q_{B,k}\Delta t^2 & 2Q_{B,k}^2\Delta t^2 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}}_{a_1^2,\ a_1a_2 \text{ and } a_2^2 \text{ terms}} \\
& + \underbrace{\begin{bmatrix} 2(T_{k+1} - T_k)\left(T_k - T_k^{amb}\right)\Delta t \\ -2(T_{k+1} - T_k)\Delta t \end{bmatrix}^T \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}}_{a_1 \text{ and } a_2 \text{ terms}} \\
& + \underbrace{(T_{k+1} - T_k)^2}_{\text{1-term}}
\end{aligned} \tag{2.5}$$

Clearly we observe that the the quadratic terms do not vanish and hence the prove. Inserting quadratic formulation back in 2.1 results in

$$\boxed{\begin{aligned}
&\underset{a}{\text{minimize}} \quad \frac{1}{2}a^T Ha + c^T a + K = \underset{a}{\text{minimize}} \quad \frac{1}{2}a^T Ha + c^T a \\
&\text{with:} \\
&H = \sum_{k=1}^{100+E_1} \left(\begin{bmatrix} 2(T_k - T_k^{amb})^2 \Delta t^2 & -(T_k - T_k^{amb}) Q_{B,k} \Delta t^2 \\ -(T_k - T_k^{amb}) Q_{B,k} \Delta t^2 & 2Q_{B,k}^2 \Delta t^2 \end{bmatrix} \right) \\
&c = \sum_{k=1}^{100+E_1} \left(\begin{bmatrix} 2\left(T_{k+1} - T_k\right)\left(T_k - T_k^{amb}\right) \Delta t \\ -2\left(T_{k+1} - T_k\right) \Delta t \end{bmatrix} \right) \\
&K = \sum_{k=1}^{100+E_1} \left((T_{k+1} - T_k)^2 \right)
\end{aligned}} \tag{2.6}$$

Since K is a constant it only influences the optimal value of the optimal objective function and not its arguments, we removed it to obtain the quadratic standard form. This quadratic minimization problem can now be solved by a programming algorithm. In MATLAB, quadprog.m with algorithm option 'interior-point-convex' is used to solve the quadratic problem. This algorithm works well for large-scaled problems with sparse matrices and for small-scaled problems with dense matrices. Since we have a small-scaled problem (only two variables to optimize) and a dense H-matrix we can efficiently make use of this algorithm. In our case (E1 = 4) we obtain the following optimal arguments:

$$\begin{aligned}
a_1 &= 1.35 \cdot 10^{-7} \left[\frac{1}{s}\right] \\
a_2 &= 3.71 \cdot 10^{-9} \left[\frac{K}{J}\right]
\end{aligned} \tag{2.7}$$

An other less efficient alternative to solve the quadratic minimization problem is 'trust-region-reflective'. In the case of small-scaled problem this is in most cases more accurate. In Matlab the exact same result is obtained by using this method.

3
Optimizing the energy trade

The cost minimization problem we are dealing with is defined as follows:

$$\boxed{\begin{array}{lll} \min\limits_{\substack{T_2,\dots,T_{N+1},\\ \dot{Q}_1^{in},\dots,\dot{Q}_N^{in}}} & \sum\limits_{k=1}^{N} \lambda_k^{in}\dot{Q}_k^{in}\Delta t & \\ \text{Subject to} & T_{k+1} = A\cdot T_k + B\cdot\left[\dot{Q}_k^{in},\dot{Q}_k^{out}\right]^T + c_k, & k=1,\dots,N, \\ & 0\le \dot{Q}_k^{in} \le \dot{Q}_{max}^{in}, & k=1,\dots,N, \\ & T^{min} \le T_k, & k=1,\dots,N+1 \end{array}} \tag{3.1}$$

Where λ_k^{in} is the price of buying one "unit of input heat" at time k, $\dot{Q}_{max}^{in}$ the maximum input heat to the tank and T^{min} the minimum internal temperature to ensure the heat demand. This optimization problem can be written as a linear minimization problem. This holds since we have a discrete-time 1-dimensional model fit for the internal tank temperature which can be massaged into linear equality constraints. The procedure is given below. The discrete-time 1-dimensional model was found in the previous chapter. Let us first define the states:

$$x = \begin{bmatrix} \dot{Q}_1^{in} \\ \vdots \\ \dot{Q}_N^{in} \\ T_2 \\ \vdots \\ T_{N+1} \end{bmatrix} \tag{3.2}$$

The objective function can now be concisely represented as,

$$\sum_{k=1}^{N} \lambda_k^{in}\dot{Q}_k^{in}\Delta t \;=\; [\lambda_1^{in}\dots\lambda_N^{in} 0\dots 0]\cdot \begin{bmatrix} \dot{Q}_1^{in} \\ \vdots \\ \dot{Q}_N^{in} \\ T_2 \\ \vdots \\ T_{N+1} \end{bmatrix} \;=\; f^T\cdot x \tag{3.3}$$

To formulate the equality constraint let us proceed from the discrete-time model for the temperature:

$$
\begin{aligned}
& T_{k+1} = A \cdot T_k + B \cdot \left[\dot{Q}_k^{out}, \dot{Q}_k^{in}\right]^T + c_k \\
\Longleftrightarrow & T_{k+1} - A \cdot T_k - a_2 \cdot \Delta t \cdot \dot{Q}_k^{in} = -a_2 \cdot \Delta t \cdot \dot{Q}_k^{out} + c_k \\
\Longleftrightarrow & a_2 \cdot \Delta t \cdot \dot{Q}_k^{in} - T_{k+1} + A \cdot T_k = a_2 \cdot \Delta t \cdot \dot{Q}_k^{out} - c_k \\
\Longleftrightarrow & \begin{cases} \dot{Q}_1^{in} \cdot (a_2 \cdot \Delta t) - T_2 = \dot{Q}_1^{out} \cdot (a_2 \cdot \Delta t) - A \cdot T_1 - c_1 & \text{for} \quad k = 1, \ldots, N, \\ \dot{Q}_k^{in} \cdot (a_2 \cdot \Delta t) + A \cdot T_k - T_{k+1} = \dot{Q}_k^{out} \cdot (a_2 \cdot \Delta t) - c_k & \text{for} \quad k = 2, \ldots, 2N, \end{cases} \\
\Longleftrightarrow & \underbrace{\begin{bmatrix} a_2 \cdot \Delta t & 0 & \ldots & \ldots & 0 & -1 & 0 & \ldots & \ldots & 0 \\ 0 & a_2 \cdot \Delta t & \ddots & \ddots & \vdots & A & -1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots & 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 & \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \ldots & \ldots & 0 & a_2 \cdot \Delta t & 0 & \ldots & 0 & A & -1 \end{bmatrix}}_{A_{eq}} \cdot x = \underbrace{\begin{bmatrix} \dot{Q}_1^{out} \cdot (a_2 \cdot \Delta t) - A \cdot T_1 - c_1 \\ \dot{Q}_2^{out} \cdot (a_2 \cdot \Delta t) - c_2 \\ \vdots \\ \vdots \\ \dot{Q}_N^{out} \cdot (a_2 \cdot \Delta t) - c_N \end{bmatrix}}_{b_{eq}}
\end{aligned}
\tag{3.4}
$$

The standard form of the linear problem can now be formulated:

$$
\boxed{
\begin{aligned}
\min_x \quad & f^T \cdot x \\
\text{Subject to} \quad & A_{eq} x = b_{eq} \\
& \begin{bmatrix} 0 \\ \vdots \\ 0 \\ T_{min} \\ \vdots \\ T_{min} \end{bmatrix} \leq x \leq \begin{bmatrix} \dot{Q}_{max}^{in} \\ \vdots \\ \dot{Q}_{max}^{in} \\ \infty \\ \vdots \\ \infty \end{bmatrix}
\end{aligned}
}
\tag{3.5}
$$

In matlab linprog.m with option 'dual-simplex' is used to solve the minimization problem. 'Dual-simplex' is an efficient linear optimization solver for large-scaled problems with sparse matrices.

3.1. OPTIMIZATION RESULTS

Clearly the minimization problem is linear as represented in 3.5. Furthermore, the prices are given in $\frac{€}{MWh}$. In order to have it in the order we are working with ($\frac{€}{Ws}$) it needs to be multiplied with $\left(10^6 \left[\frac{W}{MW}\right] \cdot 3600 \left[\frac{s}{h}\right]\right)^{-1}$. In Figure 3.1, the results of the optimization are shown in three subplots. In the top plot the Temperature is shown in blue. In the same plot the lower bound on the Temperature is shown and it can be seen that the Temperature always stays above the lower bound as would be expected. In the second subplot of Figure 3.1 is shown how much input heat is bought at each time-step in combination with the price at every time-step. Note that energy is not bought when the prices are high and relatively much energy is bought when the price is low. This result makes sense because the objective is to minimize the product of Q_k^{in} and λ_k over the complete horizon of 15 days. It can also be seen that the upper bound on input energy is taken into account.In the third subplot of Figure 3.1 the investments are shown at each time-step, combined with the growing sum of all investments in red. The overall optimal price resulting from the optimization is €127,58.

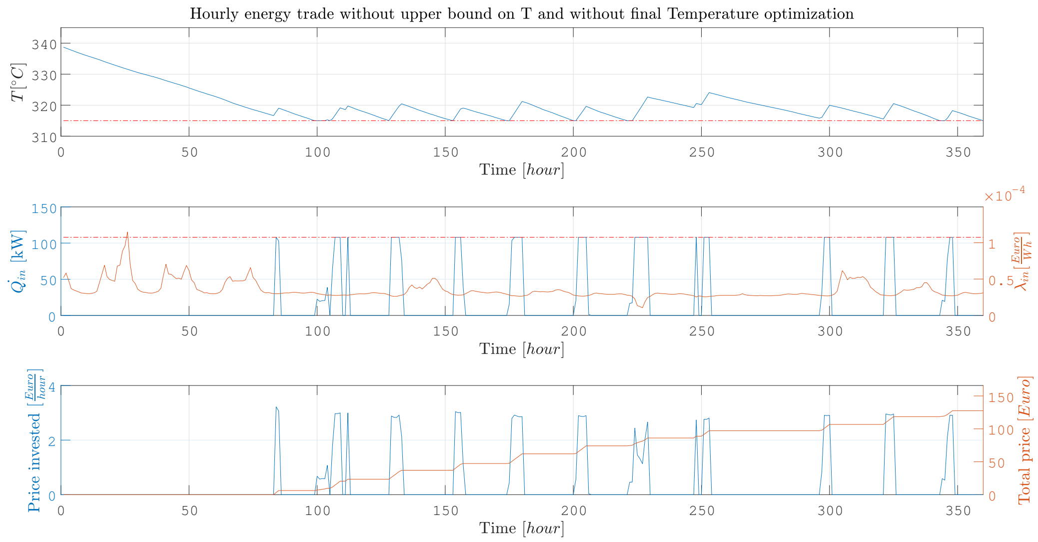

Figure 3.1: The upper graph shows how the temperature develops, the middle graph the optimal input heats which are found by the 'interior-point-convex' algorithm and the cost at which they are bought and the lower graph shows the development of the investments and its total price. All are evaluated for increasing times. The dotted red lines represent the bounds on the temperature and heat.

4 A ADDITIONAL TERMINAL COST

Introducing a quadratic terminal cost and an upper bound on the input heat results in the following minimization problem:

$$\boxed{\begin{array}{ll} \min\limits_{x} & f^T \cdot x + \left(T_{N+1} - T_{ref}\right)^2 c_N \\ \text{Subject to} & A_{eq} x = b_{eq} \\ & \begin{bmatrix} 0 \\ \vdots \\ 0 \\ T_{min} \\ \vdots \\ T_{min} \end{bmatrix} \leq x \leq \begin{bmatrix} \dot{Q}^{in}_{max} \\ \vdots \\ \dot{Q}^{in}_{max} \\ T_{max} \\ \vdots \\ T_{max} \end{bmatrix} \end{array}} \tag{4.1}$$

with c_N, the terminal cost ($0.9\frac{€}{K^2}$ in our case). The minimization can be massaged to a standard quadratic minimization problem. The steps are shown below:

$$\begin{aligned} f^T x + \left(T_{N+1} - T_{ref}\right)^2 c_N &= f^T x + \left(x_{2N} - T_{ref}\right)^2 c_N \\ &= f^T x + \left(x_{2N}^2 - 2x_{2N} T_{ref} + T_{ref}^2\right) c_N \\ &= \underbrace{x_{2N}^2 c_N}_{\text{quadratic term}} + \underbrace{f^T x - 2x_{2N} T_{ref} c_N}_{\text{linear term}} + \underbrace{T_{ref}^2 c_N}_{\text{1-term}} \\ &= \frac{1}{2} x^t \underbrace{\begin{bmatrix} 0 & \dots & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & 0 & 0 \\ 0 & \dots & 0 & 2c_N \end{bmatrix}}_{H_c} x + \underbrace{\left[\lambda_1^{in}, \dots, \lambda_N^{in}, 0, \dots, -2c_N\right]}_{c_c^T} x + T_{ref}^2 c_N \end{aligned} \tag{4.2}$$

The minimization problem can now be formulated in a quadratic standard form:

$$\boxed{\begin{array}{ll} \min\limits_{x} & \frac{1}{2} x^T H_c x + c_c^T x \\ \text{Subject to} & A_{eq} x = b_{eq} \\ & \begin{bmatrix} 0 \\ \vdots \\ 0 \\ T_{min} \\ \vdots \\ T_{min} \end{bmatrix} \leq x \leq \begin{bmatrix} \dot{Q}^{in}_{max} \\ \vdots \\ \dot{Q}^{in}_{max} \\ T_{max} \\ \vdots \\ T_{max} \end{bmatrix} \end{array}} \tag{4.3}$$

Since the quadratic minimization problem is large-scaled and has sparse matrices a proper algorithm to use is again the 'interior-point-convex' option in MATLAB. The results are discussed in Section 4.1.

4.1. OPTIMIZATION RESULTS

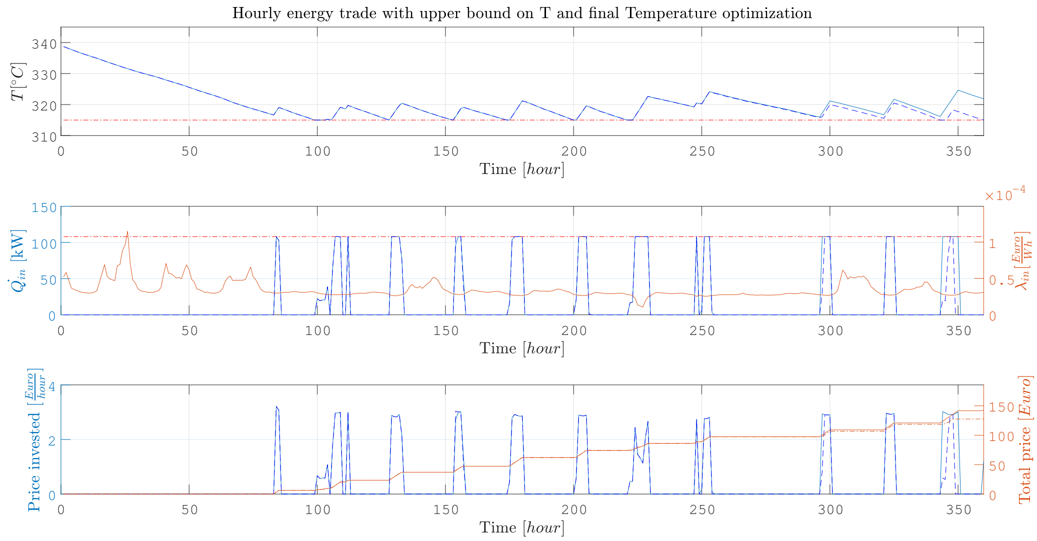

Figure 4.1: The upper graph shows how the temperature develops, the middle graph the optimal input heats which are found by the 'interior-point-convex' algorithm and the cost at which they are bought and the lower graph shows the development of the investments and its total price. All are evaluated for increasing times. The dotted red lines represent the bounds on the temperature and heat and the dotted blue lines represent the optimal values in the linear case (without terminal cost).

One can observe that the total cost is increased with respect to the linear minimization problem. Clearly, the terminal cost contribution increases the total cost, which is expected since the optimal value for the terminal temperature found in part 3 is not equal to T_{ref}. Furthermore, since the terminal cost contribution is (relatively) large (5 $°C$ difference results in about a terminal cost of approximately 25€ and 10 $°C$ difference in about 100 €!) the algorithm indeed tries to minimize the terminal cost by buying 'extra' input heat over the whole horizon: the differences can be observed in 4.1. The largest increases in investments are done at the end. Even though investing on average more over the whole horizon increases the total price, the terminal cost is made relatively much larger than this small (needed) investment since it comes in a quadratic fashion. The total costs over the horizon has increased from €127,58 to €142,78 and in the optimal case the terminal cost is indeed (relatively) small (€1,21 or 0.85 %).