VOL. 12, NO. 23, DECEMBER 2017 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

LINEAR AND NONLINEAR PREDICTIVE CONTROL ALGORITHMS APPLIED TO A HEATED TANK SYSTEM

Diego F. Sendoya-Losada and Johan Julián Molina Mosquera
Department of Electronic Engineering, Faculty of Engineering, Surcolombiana University, Neiva, Huila, Colombia
E-Mail: diego.sendoya@usco.edu.co

ABSTRACT

In this work a number of model based predictive controllers have been designed in order to regulate a (nonlinear) heated tank system. First, two controllers according to the EPSAC algorithm were designed, one with fixed and one with variable time delay. This algorithm requires a linear model, so the model was linearized around a certain equilibrium point. This gives bad results when the setpoint lies far from the equilibrium output temperature. The results obtained with a variable delay are better than when the time delay is assumed to be constant. Secondly, a NEPSAC controller was designed. A big advantage is that no linearization is required. Consequently, a correct model is available at each point. This explains why NEPSAC gives the best results of all controllers: a low settling time, no overshoot and equally good results for all setpoints. The influence of the prediction horizon was also investigated. A higher value for the prediction horizon results in a calmer system because the controller takes into account more future values. There are less fluctuations in the input and the output converges with less overshoot, but slower. Finally, the NEPSAC controller was tested on the real heated tank system. The tests show that despite a faulty model and a high sensitivity to noise, the controller still gives surprisingly good results. These are comparable to the simulation results. It can be concluded that the NEPSAC controller is very robust.

Keywords: EPSAC, MBPC, NEPSAC, temperature control.

1. INTRODUCTION

Today, more advanced control methods are often used to control industrial processes. One of these methods is model based predictive control (MBPC). Its principal is simple. A process model is determined in advance, for example by identification. With the aid of this model the process output corresponding with different inputs is calculated. The input which gives the best results is then applied to the system.

In this project the MBPC strategy is applied to the heated tank system in Figure-1. Cold water enters the tank, where it is heated with a constant amount of heat Q. A mechanical float switch ensures that the outlet and inlet flow are equal. We wish to control the outlet temperature by varying the flow $q(t)$. The output temperature is not measured directly after the outlet but at a distance L. The tube through which the water flows, introduces extra dynamics and a time delay to the system.

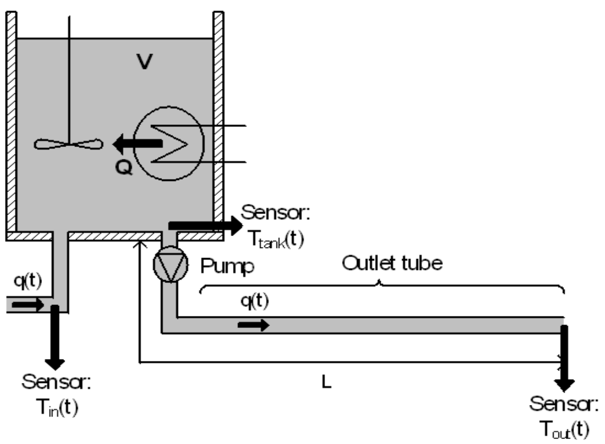

Figure-1. Heated tank system.

There exist many different approaches to MBPC. In this contribution, the EPSAC (Extended Prediction Self-Adaptive Control) algorithm is used, assuming both a constant and a variable time delay. Then a comparison with the NEPSAC (Nonlinear Extended Prediction Self-Adaptive Control) algorithm is done. This is the nonlinear variant of EPSAC. The big difference is that NEPSAC nowhere requires the linearization of the nonlinear process model.

These controllers will be tested and evaluated in simulation. Afterwards the performance of the NEPSAC controller will be tested on the real tank.

2. MATERIALS AND METHODS

2.1 Process model

Figure-2 shows a schematic representation of the heated tank system. Notice that the tube dynamics are viewed separately from the time delay introduced by the tube.

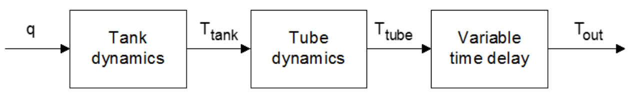

Figure-2. Schematic representation of the system.

The tank dynamics are given by:

$$\rho\, c_p V_{tank} \frac{dT_{tank}(t)}{dt} = Q + \rho\, c_p\, q(t)\big(T_{in} - T_{tank}(t)\big)$$

This subsystem is nonlinear: the right hand side contains a product of the input $q(t)$ and the output $T_{tank}(t)$.The tube dynamics are modeled as follows:

VOL. 12, NO. 23, DECEMBER 2017 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

$$\frac{T_{tube}(s)}{T_{tank}(s)} = \frac{K_{tube}}{\tau_{tube}s + 1}$$

The output temperature is then

$$T_{out}(t) = T_{tube}(t - d(t))$$

where $d(t)$ is the variable time delay, which is dependent on the flow $q(t)$. The flow is equal to the speed of the water multiplied with the tube section: $q = vS$. So the higher the flow, the faster the water goes through the tube and the smaller the time delay is. If the flow was constant, the delay would be constant too and equal to (in discrete time):

$$N_d = \frac{LS}{q\,T_s}$$

In this expression N_d is the time delay in number of sampling periods, T_s is the sampling period and LS the volume of the tube. However, the flow will not be constant. The time delay can then be estimated from:

$$T_s \sum_{i=1}^{N_d} q(t - i) = LS$$

2.2 Predictive control

The usual approach is to predict the process output $y(t + k|t)$ until the prediction horizon $k = N_2'$: $y(t + 1|t)$ to $y(t + N_2'|t)$, based on the following model:

$$y(t) = x(t) + n(t)$$

Here $x(t)$ is the model output and $n(t)$ is the noise on the process. $n(t)$ also contains the influence of both actual noise and modeling errors. This is clarified in Figure-3.

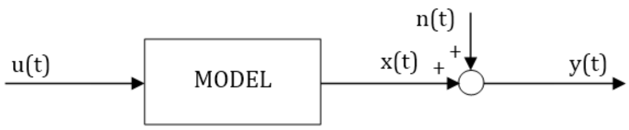

Figure-3. Process model.

The controller minimizes the difference between the predicted process output and a reference trajectory $r(t + k)$ during the coincidence horizon $k = N_1'$ to N_2'. This is done by minimizing the following cost function:

$$\sum_{k=N_1'}^{N_2'} [r(t + k|t) - y(t + k|t)]^2$$

Normally N_1' is chosen equal to the process time delay ($N_d + 1$), because the influence of an input $u(t)$ only becomes visible in the output $y(t + N_d + 1|t)$. N_2' is chosen so that $N_2' - N_1'$ is constant. However, for the heated tank the time delay is variable, so N_1' and N_2' should be adjusted in every step. To avoid this, a slightly different model is used. This is shown in Figure-4. The time delay is separated from the rest of the process dynamics:

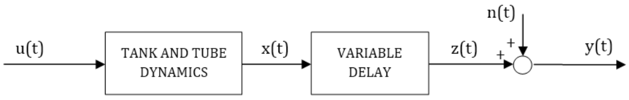

Figure-4. Adjusted process model.

Using this model $z(t) = x(t - N_d)$. So instead of estimating $z(t + N_1'|t)$ to $z(t + N_2'|t)$ it could be just as well estimate $x(t + 1|t)$ to $x(t + N_2' - N_1' + 1|t)$. The noise is modeled in a way that each prediction $n(t + k|t)$ is equal to $n(t|t)$, thus the output prediction can be determined as

$$y(t + k|t) = x(t + l|t) + n(t|t)$$

with $k = N_1'..N_2'$ and $l = N_1..N_2$. Here $N_1 = 1$ and $N_2 = N_2' - N_1' + 1$. This altered model allows to work with constant values for N_1 and N_2.

The state of the system $x(t + k|t)$ is estimated in a parallel way. This means that predictions for $x(t + k|t)$ are determined based on the values of previous inputs u and previous states x. Information present in previous outputs y is not taken into account. This means that there is no check to see if the estimated states are staying aligned with the real output. Based on the information at time t, $x(t|t)$ can be determined exactly. This value is not a prediction but the actual value of the state $x(t)$ and therefore has to be stored.

The noise $n(t)$ is represented by white noise $n_f(t)$ going through a coloring filter.

$$n(t) = \frac{C(q^{-1})}{D(q^{-1})} n_f(t)$$

The choice of the filter is a part of the controller design. Here the simplest filter that eliminates non-zero average disturbances was used.

$$\frac{C(q^{-1})}{D(q^{-1})} = \frac{1}{1 - q^{-1}}$$

Since $n_f(t)$ is white noise, the best prediction for future values $n_f(t + k|t)$ is the noise average, which is zero. With this information the process noise can be predicted as:

$$\begin{aligned} n(t + k|t) &= \frac{C(q^{-1})}{D(q^{-1})} n_f(t + k|t) \\ &= n(t + k - 1|t) + n_f(t + k|t) \\ &= n(t + k - 1|t) \\ &= \ldots \\ &= n(t|t) \text{ for } k = N_1..N_2 \end{aligned}$$

VOL. 12, NO. 23, DECEMBER 2017 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

The value for $n(t|t)$ is determined as the difference between the real output $y(t)$ and the ouput $z(t) = x(t - N_d)$ estimated from the process model. This can be seen easily from Figure-4.

$$n(t|t) = y(t) - x(t - N_d)$$

Through $x(t + k|t)$, the predicted output $y(t + k|t)$ is dependent on the applied control input u. This tries to find the optimal input during the control horizon: $u(t)$ to$u(t + N_u - 1)$, so the first N_u samples of the future input are variable. For the rest of the prediction, the input is held constant at $u(t + N_u - 1)$. The applied control input is split up in two parts.

$$u(t + k|t) = u_{base}(t + k|t) + \delta u(t + k|t) \ \ k = 0..N_2 - 1$$

Due to this, the output contains two parts too.

$$y(t + k|t) = y_{base}(t + k|t) + y_{opt}(t + k|t) \quad k = 1..N_2$$

$y_{base}(t + k|t)$ is the process output due to the basic future control scenario $u_{base}(t + k|t)$. $y_{opt}(t + k|t)$ is the output due to the optimizing future control actions $\delta u(t + k|t)$. The latter are determined so that the resulting output predictions $y(t + k|t)$ minimize the cost function defined earlier. These can be found as follows. If the following vectors are defined,

$$Y = [y(t + N_1|t) \quad ... \quad y(t + N_2|t)]^T$$

$$\bar{Y} = [y_{base}(t + N_1|t) \quad ... \quad y_{base}(t + N_2|t)]^T$$

$$Y_{opt} = [y_{opt}(t + N_1|t) \quad ... \quad y_{opt}(t + N_2|t)]^T$$

Can be stated that

$$Y = \bar{Y} + Y_{opt}$$

$\bar{Y}$can be determined through the above procedure (as the sum of $x(t + k|t)$ and $n(t + k|t)$) since the control input $u_{base}(t + k|t)$ is known. Y_{opt} is dependent on a control input does not know yet. The relation between Y_{opt} and the unknown control input $\delta u(t + k|t)$ can be expressed as follows:

$$Y_{opt} = GU$$

with

$$G = \begin{bmatrix} h_{N_1} & \cdots & h_{N_1 - N_u + 2} & g_{N_1 - N_u + 1} \\ \dots & \cdots & \dots & \dots \\ h_{N_2} & \cdots & h_{N_2 - N_u + 2} & g_{N_2 - N_u + 1} \end{bmatrix}$$

$$U = [\delta u(t|t) \quad ... \quad \delta u(t + N_u - 1|t)]^T$$

In the G-matrix, h_i are the coefficients of the unit impulse response of the system and g_i are the coefficients of the unit step response.

The cost function to minimize then becomes:

$$(R - Y)^T(R - Y) = [(R - \bar{Y}) - GU]^T[(R - \bar{Y}) - GU]$$

which can be solved for U as follows

$$U^* = (G^T G)^{-1} G^T (R - \bar{Y})$$

For linear systems the optimal system input then becomes:

$$u(t + k|t) = u_{base}(t + k|t) + U^*(k + 1) \ k = 0..N_2 - 1$$

This is so because for linear systems the sum of two inputs leads to an output that is equal to the sum of the outputs corresponding with each input applied separately. For nonlinear systems this property is no longer valid. An iterative procedure is used in which u_{base} is adapted as follows:

$$u_{base,new} = u_{base,old} + \delta u$$

in which δu is determined based on $u_{base,old}$. The procedure is repeated until δu becomes virtually zero. The optimal system input is then

$$u(t + k|t) = u_{base}(t + k|t) \quad k = 0..N_2 - 1$$

In both cases only the first value of the calculated optimal input, $u(t|t)$, is applied.

2.3 Simulating and testing

For all controllers a control horizon $N_u = 1$ was used. This simplifies some matters: $u_{base}(t + k|t)$ is defined by one single value $u_{base}(t|t)$, U becomes a scalar and G reduces itself to a $(N_2 - N_1 + 1) \times 1$– column vector containing the system unit step response coefficients.

$$G = [g_{N_1} \quad \cdots \quad g_{N_2}]^T$$

First, an EPSAC controller was designed. This procedure assumes a linear model. Therefore the process model was linearized model around an equilibrium point for the flow. The matrix G is also determined based on this linearized model. In a first step a fixed time delay was assumed, corresponding with the equilibrium flow. Then the time delay was made variable, dependent on the actually used flow.

Second, a NEPSAC controller was implemented. Here, the actual nonlinear process model was used in an iterative procedure to find the optimal input. Note that for nonlinear systems the system step response is dependent on the operating point of the system. G therefore has to be redetermined in every step. This is done based on the

ARPN Journal of Engineering and Applied Sciences

nonlinear model. Take notice that in the NEPSAC algorithm linearization is nowhere required.

Finally, the NEPSAC algorithm was tested on the real heated tank. In order to do that, it was necessary to build in the algorithm in the controller system regulating the tank. For simulating and testing the used sampling time is 4s.

3. RESULTS AND DISCUSSIONS

3.1 EPSAC

EPSAC requires a linear model for the process. The system model consists of two parts: the tank dynamics and the tube dynamics. The tube dynamics are already linear.

$$\frac{T_{tube}(s)}{T_{tank}(s)} = \frac{K_{tube}}{\tau_{tube}s\ +1}$$

The tank dynamics are not. They need to be linearized. The linearization was made around the equilibrium point $q^* = 0.0167$ l/s.

$$\rho\, c_p V_{tank}\frac{dT_{tank}(t)}{dt} = Q + \rho\, c_p\, q(t)\left(T_{in} - T_{tank}(t)\right)$$

For the equilibrium point: $f(q^*, T_{tank}{}^*) = 0$. In this way the equilibrium value for the tank temperature can be found.

$$T_{tank}{}^* = \frac{Q}{\rho c_p q^*} + T_{in} = 30.735\ °C$$

The equilibrium value for the tube temperature ($s = 0$) is then

$$T_{tube}{}^* = K_{tube}T_{tank}{}^* = 30.428\ °C$$

Linearizing the tank dynamics yields the following transfer function:

$$\frac{\overline{T_{tank}}(s)}{\bar{q}(s)} = \frac{T_{in} - T_{tank}{}^*}{V_{tank}s\ + q^*}$$

The input and output have now become the deviations from the equilibrium point, $\bar{q} = q - q^*$ and $\overline{T_{tank}} = T_{tank} - T_{tank}{}^*$. The total system transfer function is then

$$TF = \frac{\overline{T_{tank}}(s)}{\bar{q}(s)}\frac{\overline{T_{tube}}(s)}{\overline{T_{tank}}(s)}$$

with $\overline{T_{tube}} = T_{tube} - T_{tube}{}^*$. Discretizing this transfer function finally gives:

$$dTF = \frac{B(q^{-1})}{A(q^{-1})} = \frac{-\,3.563q^{-1} - 3.336q^{-2}}{1 - 1.814\, q^{-1} + 0.821q^{-2}}$$

Based on this discrete transfer function $x(t) = \overline{T_{tube}}(t)$can be estimated as

$$x(t) = -a_1\, x(t-1) - a_2\, x(t-2) + b_1\, u(t-1) + b_2\, u(t-2)$$

In this $u(t)$ is equal to $\bar{q}(t)$. The actual input and system state are then respectively $u(t) + q^*$ and $x(t) + T_{tube}{}^*$. The G-matrix can be found as the first N_2 values generated by `step(dTF)`.

During the first 200 s the process is brought to its equilibrium by applying a constant input flow equal to q^*. The system output will then, after a certain time, become equal to $T_{tube}{}^*$. After that, it is allowed the controller to regulate the input, and change the system setpoint in order to see its performance.

The (real) process output is generated by a Simulink file which contains the complete nonlinear model of the system. This file requires two vectors, one containing the time and one containing the applied flow, as input. The model output, defined as $z(t)$ in Figure-4, is determined as $x(t - N_d)$ with $x(t)$ estimated as explained above.

The optimal input at time t, $u^*(t|t)$, is then calculated using the methods explained in section 2. As basic input $u_{base}(t) = u(t-1)$ was used. However, there are some constraints on the flow. It has to lie between 0.005 l/s and 0.03 l/s. This was resolved in a suboptimal way, using clipping. The optimal input was determined and if it was too high, 0.03 was put; if it was too low, it was redefined as 0.005.

3.1.1. Fixed time delay

In a first step the time delay was assumed to be constant and equal to the delay corresponding with q^*.

$$N_d = \frac{LS}{q^* T_s} = 15.27\ s$$

The delay has to have a discrete value so it was rounded to $N_d = 15$. Ignoring the dependency of the time delay on the flow applied in the previous steps, will lead to less accurate values of the model output $z(t) = x(t - N_d)$. This, in turn, will lead to worse estimation of the noise $n(t)$.

Figure-5 shows the model output $z(t)$ and the real process output $y(t)$ for a prediction horizon $N_2 = 15$. It also shows the applied control input $u(t)$.

VOL. 12, NO. 23, DECEMBER 2017 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

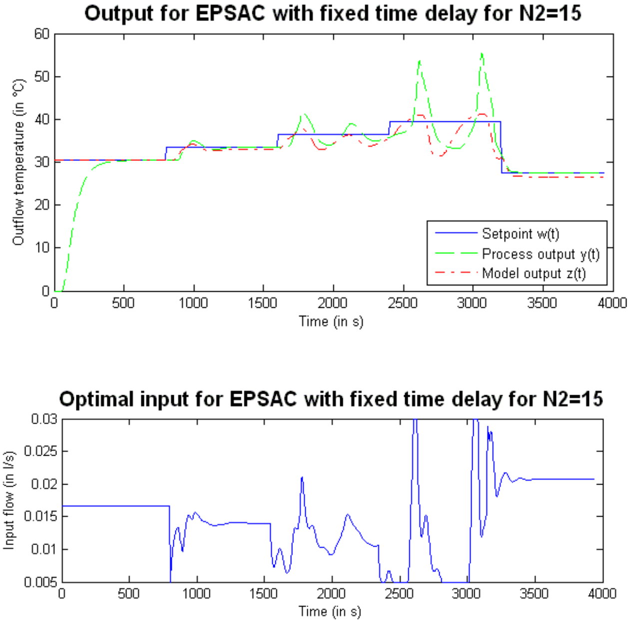

Figure-5. Results for EPSAC with fixed time delay for $N_2 = 15$.

The model is linearized in a point corresponding with an output temperature equal to $T_{tube}{}^{*} = 30.248$ °C. When the setpoint is close to this value, as in the first step (3°C higher than $T_{tube}{}^{*}$), convergence to the setpoint is reasonably fast and doesn't show too much overshoot. But as soon as the setpoint lies a bit further, results worsen fast. The second setpoint step, which is only 6°C higher than $T_{tube}{}^{*}$, already shows a lot of overshoot, as well in the process as in the model output. The output will oscillate for a long time before finally settling on the setpoint. It can be seen a lot of peaks and quick changes in the corresponding input. This explains the oscillatory behavior. At the third step (9°C higher), it can be seen that the system model is now so bad that the output can no longer follow the setpoint. The system becomes unstable. This can be seen in the input, which starts to oscillate between its minimum and maximum values, and in the output: the second peak is higher than the first one. As soon as the setpoint comes in a range of 3°C again, the controller starts giving good results again. The last setpoint change is followed quickly and without overshoot.

Figure-6 shows the same variables (model output, process output and input) but for a controller with a prediction horizon equal to 30.

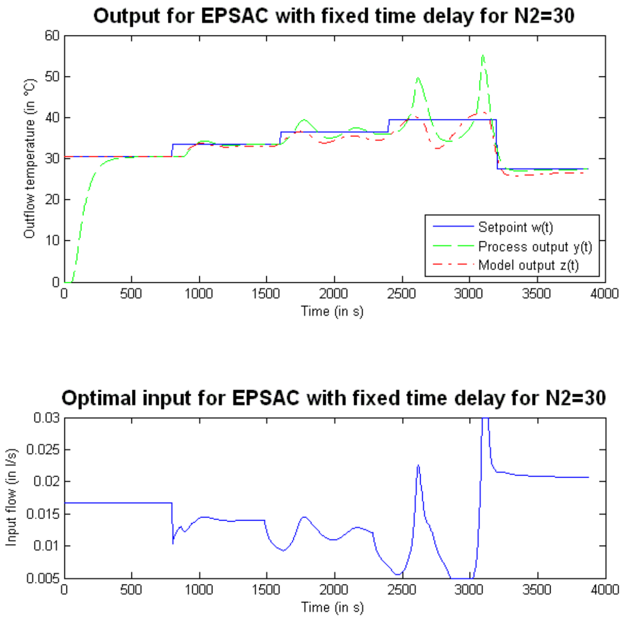

Figure-6. Results for EPSAC with fixed time delay for $N_2 = 30$.

Here it can be seen that in input is a lot less nervous. A higher prediction horizon results in a calmer (but slower) system reaction. The overshoot in the output is a bit lower, but not very much so. Although the input doesn't vary so quickly, it can be still seen it reaching its extremes at the third setpoint step. The system will still become unstable. This can also be seen very clearly here from the fact that the second output peak is a lot higher than the first one.

3.1.2. Variable time delay

Now it is taken into account that the time delay is actually variable. N_d is determined as the highest number for which

$$T_s \sum_{i=1}^{N_d} q(t-i) < LS$$

Figure-7 shows the output, the input and the time delay obtained with an EPSAC controller with variable time delay and a prediction horizon $N_2 = 15$.

VOL. 12, NO. 23, DECEMBER 2017 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

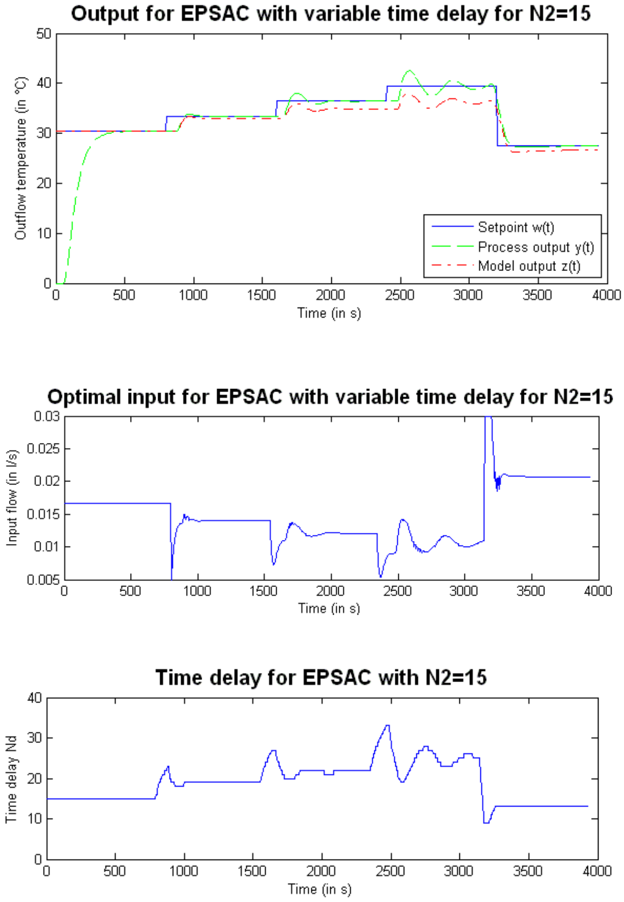

Figure-7. Results for EPSAC with variable time delay for $N_2 = 15$.

In comparison with a fixed time delay, results are already a lot better. The system stays stable everywhere and only at the third step (of 9°C) the system still oscillates when the setpoint changes again. Taking into account the variability of the time delay, is a big improvement for our model. It becomes a lot more realistic. Predictive control heavily relies on the quality of the model at hand. With an improved model the output prediction is better, leading to a more accurate determination of the optimal input. This allows setpoints to be followed in a wider range around the equilibrium point $T_{tube}{}^*$.

It can be clearly seen in the input that every time the setpoint goes up, that the input shows a downward peak. This is logical because if the flow decreases a lot, the water stays in the tank a lot longer, giving it more time to be heated. This results in a higher output temperature. When the setpoint decreases, the opposite effect can be seen. This change in the input and the output in an opposite way was also present in the previous cases, since it is a direct consequence of the system physics, but there it couldn't be seen quite so clearly.

The time delay is, as expected, proportional with the inverse of the input. It can be seen peaks at the same time, but in the opposite direction. The delay is, most of the time, significantly higher than the equilibrium value ($N_d = 15$) at which it was fixed in the previous paragraph. This is again an indication that results for a fixed time delay were bad because of an inaccurate model.

Figure-8 shows the results for an EPSAC controller with variable time delay and prediction horizon $N_2 = 30$.

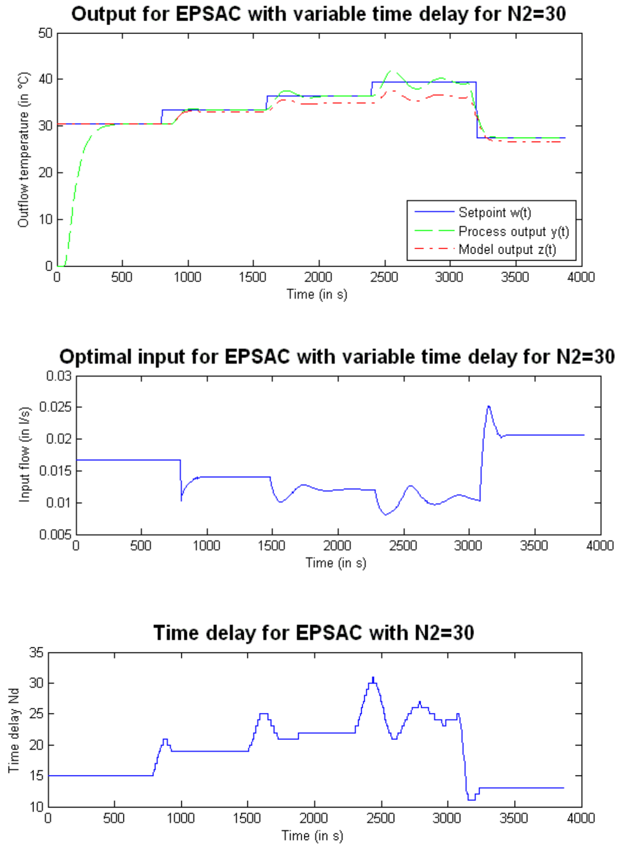

Figure-8. Results for EPSAC with variable time delay for $N_2 = 30$.

Again there are less fluctuations in the input. It changes more gradually. The peaks in the input are also smaller. The input stays between 0.007 l/s and 0.026 l/s, whereas for $N_2 = 15$ it still reaches its limits at some points. The output shows less overshoot too. So it is again confirmed that a higher prediction horizon leads to a calmer system response. It is perhaps a bit slower, but that is hardly noticeable here.

3.2 NEPSAC

Just as for the EPSAC controller, the system was going to its equilibrium point in the first 200s by applying q^* without controlling the process. After 200s there is a setpoint change and the NEPSAC controller starts to regulate the input. The reference trajectory is the same sequence of setpoints as used for simulating the EPSAC controller.

Unlike EPSAC, NEPSAC doesnot require a linear model. The system equations donot need to be linearized. The (real) process output is again generated by

VOL. 12, NO. 23, DECEMBER 2017 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

a Simulink file. The prediction for the system state $x(t)$ is generated by a second Simulinkfile, which contains the nonlinear plant model without the time delay or anything to do with it. Both need the previous applied input and the time as input vectors. The variable time delay is then again estimated as the largest number for which

$$T_s \sum_{i=1}^{N_d} q(t-i) < LS$$

The noise is determined as the difference between the model output $z(t) = x(t - N_d)$ and the real process output. With predictions for $x(t+k|t)$ and $n(t+k|t)$, $y_{base}(t+k|t)$ can be calculated. As initial basic input it was used again $u_{base}(t) = u(t-1)$. $u_{base}(t)$ is then modified in an iterative procedure, as explained in section 2. This keeps iterating as long as the optimizing input $\delta u(t)$ is higher than 10^{-6}. This is sufficiently smaller than the minimal value for the input (0.005 l/s) to be able to be regarded as zero. The final value for $u_{base}(t)$ is then the optimal input which has to be applied at time t. This value is again limited between 0.005 l/s and 0.3 l/s.

To find the optimizing input, the G-matrix with the system step response coefficients is required. Because of the nonlinear model, this matrix is no longer constant. The system response to a step will be different in each operating point. In every iteration G is determined based on the nonlinear system model. First, the system is simulated with a constant input equal to $u_{base}(t)$, so the system stays in its operating point. This is done by the second Simulink file. The values for $x(t+N_1|t)$ to $x(t+N_2|t)$ are stored as the matrix G_2. Next, a constant input $u_{base}(t) + \text{step}$ is applied, so a step is applied on top of the operating flow. $x(t+N_1|t)$ to $x(t+N_2|t)$ are stored in G_1. The matrix G, containing the unit step responses g_{N_1} to g_{N_2}, is then

$$G = \frac{G_1 - G_2}{step}$$

The system response is also dependent on the size of the applied step. To obtain good predictions, the size of this step should be in the same order of magnitude as the changes in the input during a single time step. Based on the simulations for the EPSAC controller, a step equal to 0.001 l/s was taken.

Figure-9 shows the process and the model output obtained with a NEPSAC controller with $N_2 = 15$. It also shows the corresponding input, the variable time delay and the number of iterations needed in each step.

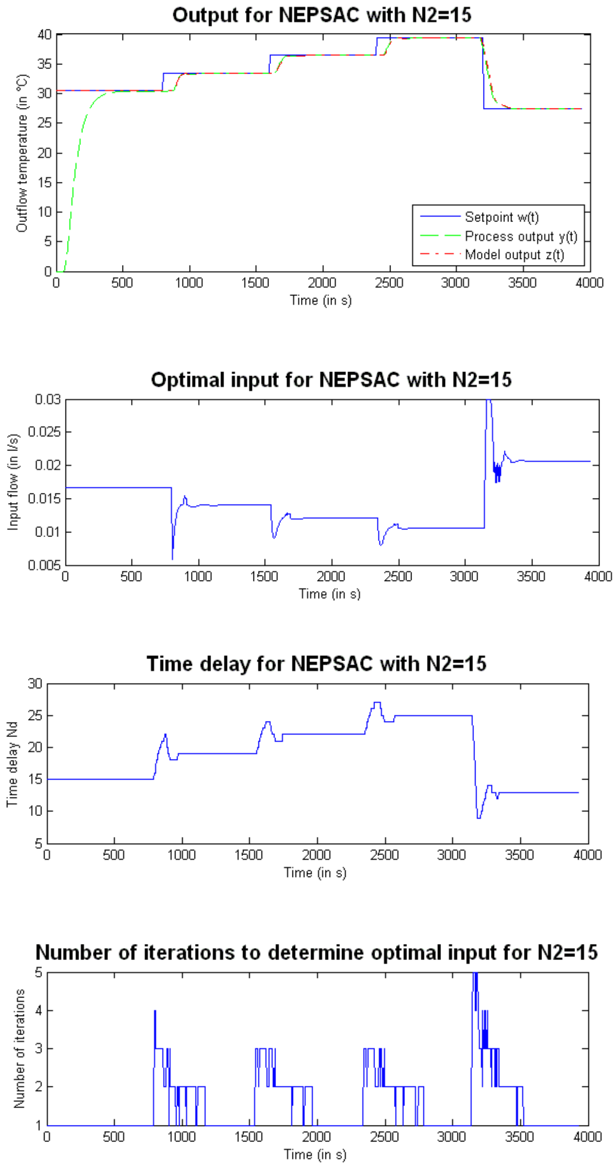

Figure-9. Results for NEPSAC with $N_2 = 15$.

It can be seen that this controller gives very good results. The output follows the setpoint changes without overshoot and within a small time. Notice that results are equally good for all setpoints. This is due to the fact that the model was not linearized. The correct model is being used in all points. It also accounted for the variability of the time delay. Consequently, the model is very good. This leads to good predictions and a very good control of the process.

The input is comparable to that for the EPSAC with variable time delay ($N_2 = 15$) but the control efforts are even more concentrated around the setpoint changes. The time delay is again complementary to the input.

Maximum five iterations are used. It can be seen that the number of iterations goes up every time the setpoint changes. If the output is in regime, on the setpoint, one iteration is sufficient. This is logical since the

VOL. 12, NO. 23, DECEMBER 2017 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

previous input $u(t-1) = u_{base}(t)$ will then already be the optimal input.

Figure-10 shows the results for a NEPSAC controller with a prediction horizon $N_2 = 30$.

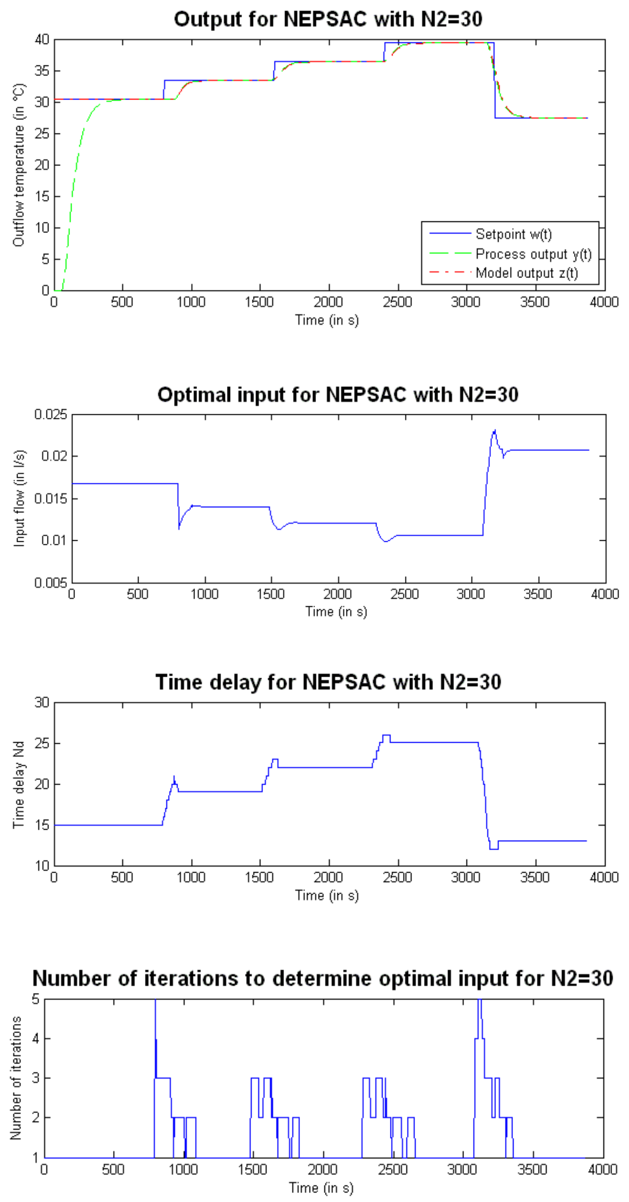

Figure-10. Results for NEPSAC with $N_2 = 30$.

The input is again very calm. There are no fluctuations present, not even little ones, whereas for$N_2 = 15$ there are. The peaks in the input, when the setpoint changes, are also a lot smaller. The controller can better estimate the effect of the input, because it looks further ahead. Even the number of iterations shows less fluctuations.

The output reacts earlier but slower to changes in the setpoint. At the first setpoint change, the real delay between input and output can be seen. The controller only starts working at the moment the setpoint changes. It immediately adjusts the input but the effect only becomes visible after a certain time. For the other setpoint changes, the output starts converging to the new setpoint immediately. This is because the controller looks far enough into the future to see the setpoint change ahead. It already starts to adjust the input before the setpoint change, at a moment such that the effect becomes visible at the setpoint change. For $N_2 = 15$ changes will also be anticipated, but they will be detected later. This results in the fact that there is still a delay between the moment the setpoint changes and the moment the output starts to converge, though it is smaller than the delay at the first setpoint step. Note that, although the output for the controller with $N_2 = 15$ starts reacting later, it reaches the setpoint at the same time as the output for $N_2 = 30$. This implies that for the second controller ($N_2 = 30$) the output reacts more slowly.

Now it is wanted to observe what the absolutely necessary number of iterations to obtain reasonable results is. The NEPSAC controller with $N_2 = 30$was applied to the system, limiting the number of iterations to one. This was done by commenting the "while dU>epsilon"-loop.

Figure-11 shows the obtained results. These are identical to the ones in Figure-10. It can be concluded that, in simulation, more than one iteration isn't even necessary. In a single step the calculated input is already close enough to the optimal input to give excellent results.

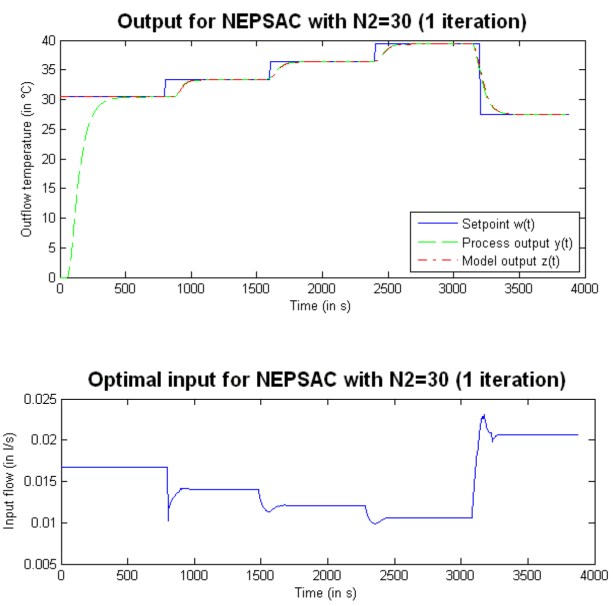

Figure-11. Results for NEPSAC ($N_2 = 30$) with the number of iterations limited to 1.

3.3 Testing on the real plant

Finally, the NEPSAC controller was tested on the actual heated tank. In order to do this, the algorithm was implemented in a Matlab file. After that, it was possible to use the graphical interface that was available. A prediction horizon $N_2 = 30$ was used. The used number of iterations N_i in each step had to be chosen by the user too. $N_i = 15$was selected. This was the maximum value (otherwise

ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

the calculations would start taking too long), to obtain as good results as possible.

Figure-12 only shows the test results for when the system had stabilized. At first, the output didn't converge to the given setpoint within a reasonable time. By repeatedly changing the setpoint to a value close to the current value of the output temperature, the system was stabilized. Once the system was stabilized, the output followed the setpoint steps nicely, as long as these were not taken too big (a few degrees). The settling time is approximately 250s (a little less). This is comparable with the simulations. It is now being compared with the first setpoint step in the simulations, because on the real plant, the controller doesn't know beforehand when the setpoint will change. This results in a situation comparable with only starting the control at the moment of a setpoint change.

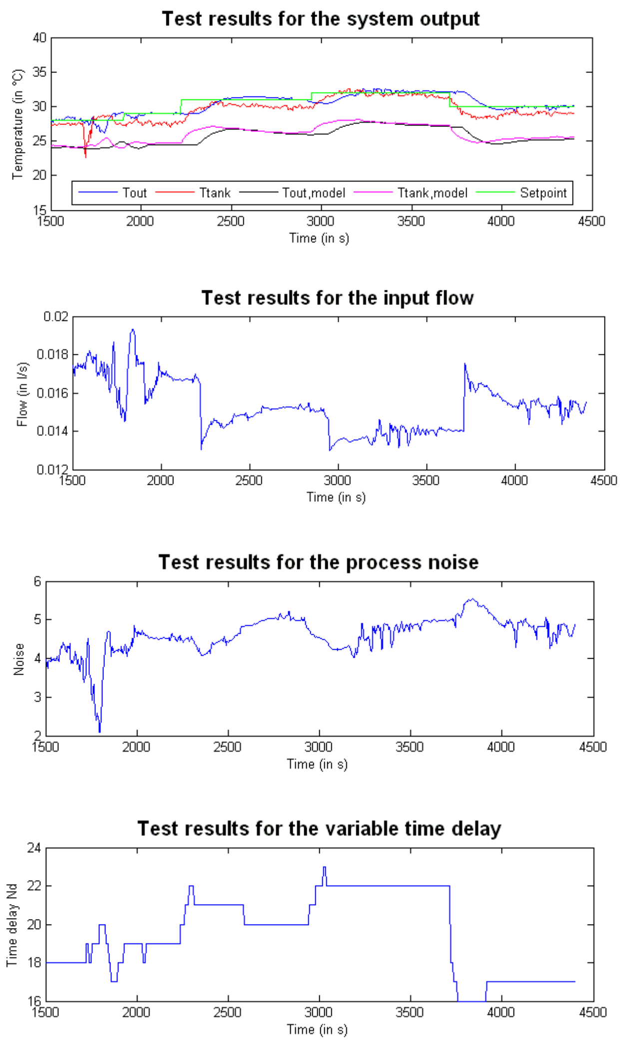

Figure-12. Test results for the NEPSAC controller.

There is a significant difference between the real tank and tube temperatures and those estimated from the model. The estimated temperatures are systematically 4 to 5°C lower than in reality. This can also be seen in the process noise which is more or less constant around 4.5°C. As stated before, the noise also represents modeling errors. The system model was identified some time ago. It is clear that the system parameters have changed since. The fact that the real temperatures are always a few degrees higher than expected, leads us to suspect that the amount of heat added in the tank has increased over the time. Nevertheless, the results obtained with this faulty model are pretty good. This indicates that the NEPSAC controller is very robust.

Another conclusion that can be drawn from the tests is that the system is highly sensitive to noise. There are lot of little oscillations in the process output and the input flow. Simply leaning on the Table (at 3250 s) results in visibly higher oscillations in the results for the output, the input and the noise. At 1700 s, someone touched the mixer. This resulted in a big noise peak that is visible in all plots. Otherwise, the noise plot contains a more or less constant value and some peaks at the moments the setpoint changes. Looking at the input, it can be seen that the input is very similar to that in the simulations for NEPSAC with $N_2 = 15$, noise aside. This is again an indication that the NEPSAC is working correctly despite a bad model and a lot of noise.

REFERENCES

De Keyser, R. 2003. A Gent'le Approach to predictive control, UNESCO Encyclopedia of Life Support Systems (EoLSS), Eolss Publishers Co Ltd, Oxford.

Normey-Rico J. E. 1999. Predicción para Control, Doctoral Thesis, Universidad de Sevilla, Spain.

Normey-Rico J.E., Camacho E. F. 2007. Control of dead-time processes, Springer-Verlag, London, U.K.

Sendoya-Losada D. F., Robayo Betancourt F., Salgado Patrón J. 2016. Time delay estimation for BIS monitor used in general anesthesia. ARPN Journal of Engineering and Applied Sciences. 12(7): 2120-2129.

Sendoya-Losada D. F., Robayo Betancourt, F., Salgado Patrón J. 2016. Application of a predictive controller with variable time delay in general anesthesia. ARPN Journal of Engineering and Applied Sciences. 12(8): 2661-2667.