The NEPSAC Nonlinear Predictive Controller in a Real Life Experiment

Levente TAMAS[1], Ioan NASCU[1], Robin De Keyser[2]
[1]Technical University of Cluj-Napoca, Romania, {levente.tamas, ioan.nascu}@aut.utcluj.ro
[2]Ghent University, Belgium, rdk@autoctrl.ugent.be

Abstract-**This paper is intended to give a short overview about the NEPSAC nonlinear predictive control approach. In the first part there are presented the theoretical aspects of the control design procedure, and then in the next section are shown the results for the designed algorithm for a real life experiment. There are also compared the results with other type of controllers like the MPC or PID controller.**

Keywords- **nonlinear predictive control, model based control, parameter estimation, physical modeling.**

I. Introduction

The main objective of this paper is to present the NEPSAC (Nonlinear Extended Prediction Self-Adaptive Control) control algorithm in a real life application. As the name shows it can deal with *nonlinear* systems, using predictive control algorithm with a self adaptive ability. The fact that it can be applied to nonlinear systems is only an extension of the linear EPSAC, the fundamental principles of the algorithm are the same in both cases [1].

The *predictive* part of the algorithm is relying on the fact that it is based on the MBPC principle, i.e. it uses a model for predicting the future response of the system, and based upon this prediction computes the necessary control action for the system.

The *self-adaptive* capability of the algorithm specifies that even for partial miss modeling or for varying systems it is able to function within a normal range, adapting itself to the changes of the modeled system. In other words this ensures a rather high degree of robustness of the algorithm.

A. General description problem

The chosen problem may be divided into the following subparts:

- Choosing the system that will be controlled
- Building the necessary IO interface to the system
- Making identification experiments for the model
- Getting a valid model for the system
- Building a controller for the system
- Testing and optimizing the controller

At the moment of choosing the plant a list of points needs to be analyzed in order to eliminate the possible problems which may emerge at later stages of the design procedure. Such points may be the available measuring devices for the quantities needed to be measured or how many states can be measured and how many are needed to be known; controllability of the plant and the sampling rate of the signals. These are only some of the most fundamental parts of the selection criteria. Based on the mentioned criteria's there was selected a high performance equipment water tank system which will be described in details in the next section.

As the NEPSAC control algorithm is a model based control algorithm it needs a model for the system. The model can be obtained via identification experiments. At this point is already essential to have some basic knowledge about the system, such as the sampling time, the measured variables, and the controlled quantities.

Based upon this priory knowledge it is needed to be selected the type of the identification experiment: step response, stair case, PRBS or other type of excitation [3]. The ideas upon which were selected the input types is presented in the section describing the identification. After selecting the input type it is needed to be performed the proper identification experiments.

Finally, based on the collected data from the identification experiment it can be built a model for the system, which later can be used in the control design procedure. This model building is not a trivial task; it may need also physical modeling knowledge about the system. The model building is treated in details in the part presenting the modeling.

The controller design in the case of EPSAC is generally done based on input/output from a state space model. The currently presented solution has extended this principle.

B. The novelty of this approach

The main novelty of the current solution is that it does not use a state space or transfer function model for the system, instead of it, it can be used any 'black box' model being able to generate a valid output for the model for a certain excitation at a certain operation point. In other words, it uses only the response of a model, and does not care about the description form of the model.

II. Theoretical Presentation

A. Theoretical Aspects of the Model Based Predictive Control

Basically there is an interest to reduce some cost in the plants. In other word these economical problems can be translated to some cost indices used in the control procedure. The model based predictive control (MBPC) approach's main advantage is that it can handle easily these costs and constraints.

However the MBPC control does not want to eliminate, nor replace the 'old good working' PID controllers, rather it offers a chance to reformulate some problems at higher hierarchical

1-4244-1148-3/07/$25.00 ©2007 IEEE.

levels, e.g. economical ones. At this way one may think about predictive controllers as some supervisors, which can provide useful advices to their subordinates in such a way that some long term interest will be satisfied [1].

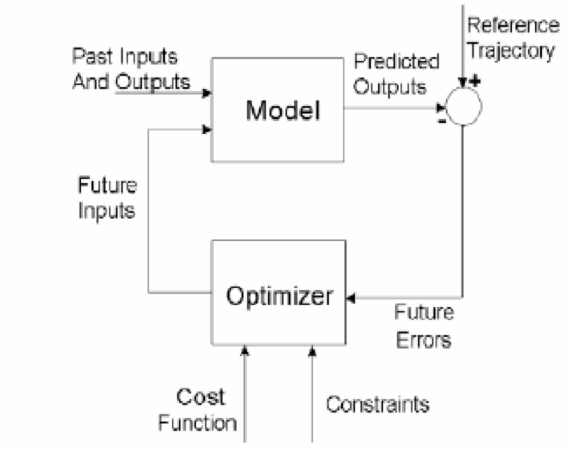

Figure 1 The basic idea of MBPC

The key idea of the MBPC can be summarized on the Figure 1. There can be seen clearly separated the past, present and future control variables. This approach supposes that there is known in advance the reference trajectory for the output state trajectories, at least in some points (dealing with the number of points/shape of the reference trajectory there are available different methods). The system may also have dead time, which can be introduced in the model. The past values for the inputs-outputs are available in the database. The main control objective is to minimize some cost represented by a cost index.

B. *The basic idea of EPSAC*

For the linear case of the predictive control algorithm consider the system as presented on the Figure 2:

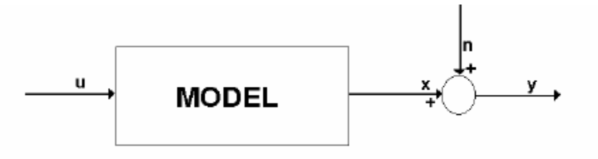

Figure 2 The model for the control

The notations used for the Figure 2 are the followings:

- $y(t)$: (measured) process output
- $u(t)$: process input
- $x(t)$: model output
- $n(t)$: noise

For the discrete model there is considered the CARIMA type one presented in the relation (1):

$$A(q^{-1})y(t) = B(q^{-1})u(t) + \frac{C(q^{-1})}{D(q^{-1})}e(t) \tag{1}$$

Where the e(t) signal is considered white noise, and it represents the disturbance through the filter between the controller and the system.

The equation represented in the relation (1) can be solved using different techniques, like the algebraic equations (Diophantine equation), or by the filtering ones. The two different approaches produce the same result [2].

The GPC approach uses the Diophantine equation techniques to solve the system (1), while the EPSAC uses the filtering one. On the following there will be presented the filtering approach.

C. *The noise prediction in the model*

One of the obvious advantages of the presented approach is that there can be implemented a noise model in the case that there is a priory knowledge about it. By the simple equation (2), the noise can be determined as the difference between the model output and the measured state:

$$n(t) = y(t) - x(t) \tag{2}$$

However it is not as trivial to predict the future noise. One convenient way is to suppose it as being white noise, passing to a suboptimal integration filter. This allows predicting the noise as 0 mean one, i.e. the best prediction being the mean of it, and considering 0.

The white noise e(t) can be recalculated by passing the measured noise through 'inverse' filter used for modeling the disturbance. After having the values for e(t) in the past, the future values can be taken 0, as the best approach for mean of the white noise. With these values, there can be predicted the future noise in a recursive way.

$$n(t+k \mid t) = \frac{C(q^{-1})}{D(q^{-1})}e(t+k \mid t) \tag{3}$$

In the relation (3) the | operator represents the prediction of the signal at the moment *t* for the time *t+k* in the future, e.g. the noise at the *t+k* moment predicted at the time *t*.

There are made N2 prediction points in the prediction horizon.

D. *The controller configuration*

The response of the system is considered to be one composed from a *base* component and a forced, *optimal* one as in the equation (4). This is valid for linear systems although it can be extended to nonlinear ones too [1].

$$y = y_{base} + y_{opt} \tag{4}$$

This means conceptually the followings:

- The yBase is produced if the system is leaved with the last command to act free over the predicted horizon with the uBase command defined priori. This is not relevant for linear systems however in the nonlinear case this needs to be considered too;

- yOptim will be the response for the optimizing part of the input, i.e. the one which is added to the uBase in order to minimize a certain cost function. This is presented on the Figure 3.

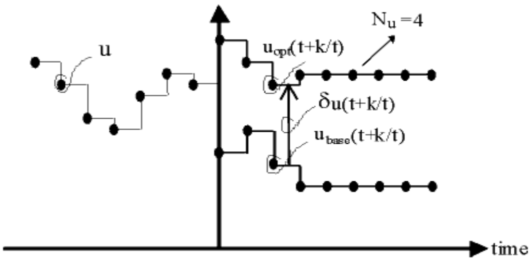

Figure 3 The idea of the base and optimized response

Having in mind that the yOptim is the response to certain special impulses represented by uOptim, with the amplitude of uOptim at moment t while on the other hand these impulses have a finite length, so they can also be interpreted as some sort of step inputs. As a result, the yOptim can be computed as a combination of impulse/step response such in the equation (5)

$$\begin{aligned} y_{opt}(t+k\mid t) &= h_k \delta u(t\mid t) + h_{k-1}\delta u(t+1\mid t) + \cdots + \\ &+ g_{k-N_U+1}\delta u(t+N_u-1\mid t) \end{aligned} \tag{5}$$

Which using a matrix notation it can be represented as:

$$Y_{opt} = \begin{bmatrix} y_{opt}(t+N_1\mid t) \\ y_{opt}(t+N_1+1\mid t) \\ \cdots \\ \cdots \\ y_{opt}(t+N_2\mid t) \end{bmatrix}$$

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

$$U = \begin{bmatrix} \delta u(t\mid t) \\ \delta u(t+1\mid t) \\ \cdots \\ \delta u(t+N_u-1\mid t) \end{bmatrix}$$

$$Y_{opt} = G * U \tag{6}$$

E. The control objective

In the case of the predictive control approach the main aim is to minimize a cost function. Therefore a most common approach is to minimize the following well know ITAE form described in (7):

$$\sum_{N1}^{N2} \left[r(t+k\mid t) - y(t+k\mid t) \right]^2 \tag{7}$$

For the term (7) using the notations from (6) the optimal solution can be written in the following matrix form:

$$U^* = \left(G^T G\right)^{-1} G^T \left(R - \overline{Y}\right) \tag{8}$$

In this way the optimal control part can be determined by simple matrix multiplication using the already defined *G* matrix and the difference between the *R* reference and the *Y* measured values, however special attention should be paid for numerical stability problems.

F. Extension to the non linear case

The superposition principle – the system response may be decomposed in base and optimized responses – presented in the previous chapters which theoretically is valid for linear systems only[1].

Although for nonlinear systems in case that the future *uBase* is selected appropriately, then the optimal and the base response are practically identical, i.e. no superposition is involved, so this results in the optimal response for NEPSAC In order to apply the EPSAC algorithm for nonlinear systems, the following steps have to be done:

- Select the *uBase* in such a way that this is close to the *uOptim*, so there is no superposition involved at the generation of the response.
- Once there is selected the *uBase* there can be computed the *uOptim*, using the above presented algorithm.
- For nonlinear systems the last step might be needed to be repeated in order to get a close *yBase* and *yOptim* response.
- As it can be seen in the case of the nonlinear systems the optimization is done more than once.

III. Practical Approach to the Problem

In this chapter first is presented the plant on which were carried out the measurements, together with the physical modeling, the simulations and the carried out experiments.

A. Presentation of the experiment equipment

On the Figure 4 there is shown the schematics of the water tank system.

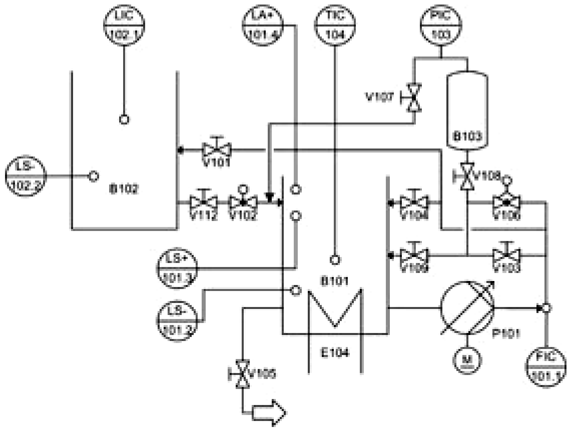

Figure 4 The schematics of the water tank system

As it can be seen there are two water tanks, from which/into which can flow water on several pipes. There is a pump in the system, several valves and a servo valve too. On both tanks there is installed an infrared level sensor

The used IO interface to the tank process is realized via the available modules from IPCON-7000 production series. These are general IO modules which can be interconnected through the RS-485 protocol in a communication bus. They use as communication protocol the DCON specific commands.

DCON Communication Protocol for I-7000 series module works on the principle that there is sent a command to the device which answers for the sent command. The distinction among the modules is done using the address as ID for each of them.

B. *The plant model*

The diagram describing the process is presented on the Figure 5. The rate at which water enters is proportional to the voltage, V, applied to the pump. The rate at which water leaves is proportional to the square root of the height of water in the tank.

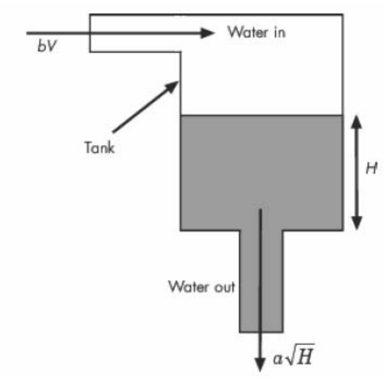

Figure 5 The water tank system overview

The water tank system can be modeled via differential equations represented by

$$\frac{d}{dt}Vol = A\frac{dH}{dt} = bV - a\sqrt{H} \qquad (9)$$

The equation (9) describes the height of water H, as a function of time, due to the difference between flow rates into and out of the tank. The equation contains one state, H, one input, V, and one output. It is nonlinear due to its dependence on the square-root of H. The constants *a* and *b* are specific for the tank, and they need to be identified.

There is also a Simulink model available for the system, which gives a clear view about the nonlinear characteristics in the system. The model using the Simulink blocks is presented on Figure 6.

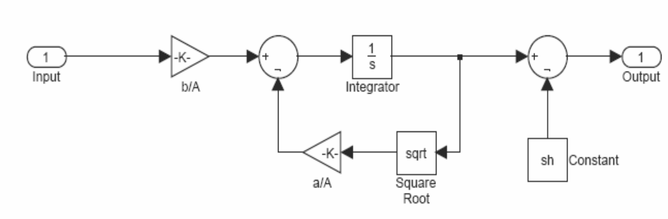

Figure 6 The Simulink schematics of the plant

C. *Parametric identification of the process*

The testing of the sensor and the pump: First there were carried out some experiments in order to determine the characteristics of the sensors and the pump of the system.

The sensor testing consists of checking the linearity of the sensor. As it was mentioned in the specifications of the system, the sensor linear is a linear one, which fact was validated with an experiment.

At the next step there was tested the pump linearity. As it can be seen on the Figure 7, the pump has a rather linear behavior in the range of 0-6. After the pump linearity test, it was checked the non-linear characteristic of the flow out from the tank, which can be also seen on this figure from the time interval 150 to 250.

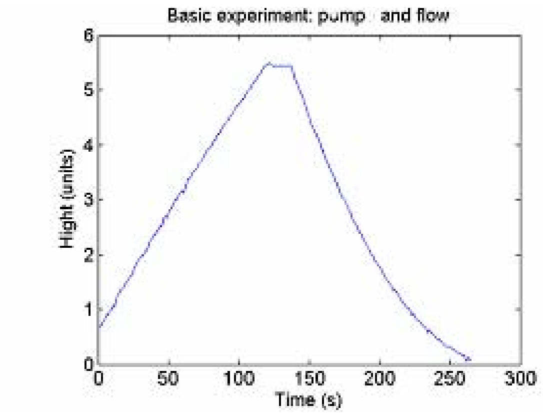

Figure 7 The linearity of the pump

After performing the two experiments there was performed a measurement with stair-case excitation to the system.

The aim of the stair case experiment was to get the nonlinearities in the system, and to be able to validate later the obtained models. The obtained stair case response is plotted on Figure 8.

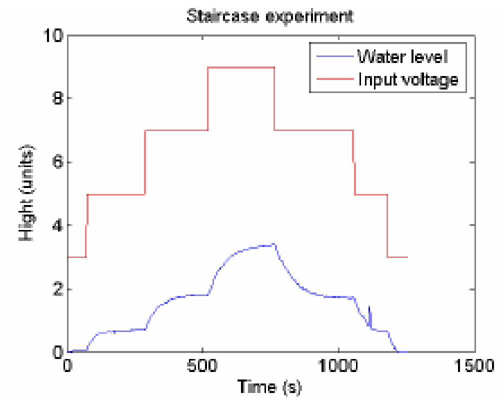

Figure 8 The stair-case experiment

The Simulink control design tool was used for the parametric identification of the plant using the dataset from the staircase experiment. Using this toolbox it can be done the identification in an automatic way, i.e. after the signal specifications, the toolbox estimates the parameters of the model based on certain types of optimization methods [5].

In the next step as a validation for the obtained model this was tested against the measured values, as this can be seen on the Figure 9. As it can be seen on this figure, the simulated output is close to the measured one.

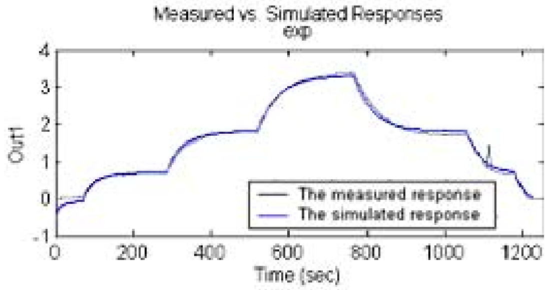

Figure 9 The response of the identified model

D. The controller design

The PID Controller design was performed using a graphical design tool based on frequency response of the system called FRTool [4]. As the plant is a nonlinear one, and the control algorithm is for linear plants, there was chosen a local operation point around which it was considered a model of the plant.

Having this model, the frequency response of the controlled system is presented on the Figure 10. As it can be seen on this figure, the response of the system (thin blue line) is satisfying the robustness constraint (thick blue line), as well as the overshoot constrain (thick red line) and the imposed response time (green line).

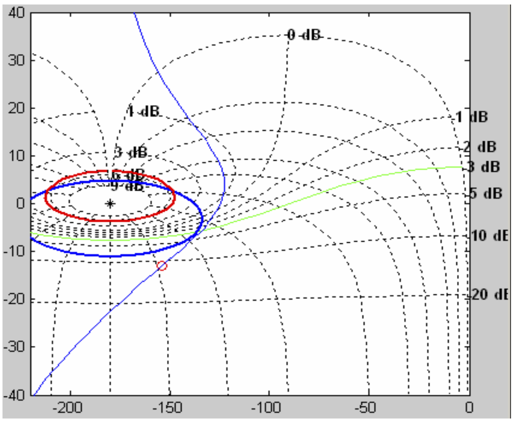

Figure 10 The frequency response of the system with the PID controller

The NEPSAC controller design was performed without making any explicit linearization of the plant, instead of this, using the referenced output generated from the model.

The whole algorithm's idea is an intuitive one, although it is important to make a suitable data representation for the past, present and future values of input/outputs, states, and noise.

The convention that was used in this paper is the following:

Table1

Past(Nx)	Present(1)	Future(N2)

In Table1 *Nx* represents the number of elements that will be stored from the past. This depends mainly on the length of the structure of the modeled system. The values for the present are just one dimensional, while the predicted values have N2 length, having in mind that all the inputs/states/noise are being predicted on a horizon of N2 points.

Although there is no local linearization performed using this algorithm, the step response at each sampling time for the model is recalculated. In order to have a fast algorithm in the first approach there was made a trade off, considering a local model for the system of the form presented in equation (10)

$$\frac{K}{s+a} \tag{10}$$

This form can be easily deduced from the physical model of the system. The problem with this approach was that the response of the system will get more and more far away from the real one. In this case the response had steady state error, due to the approximations done at the previous phase. As the prediction horizon grows, it grows the steady state error too. A possible solution for this problem may be to include an integrator element in the noise model; however this may reduce the phase margin of the system. So this approach is not a useful one, and it can be used only in the case that N2 is small, as this is also visible on the Figure 11.

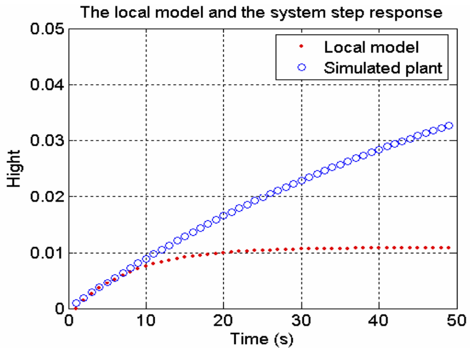

Figure 11 The locally operation point response

Instead of the local operation point it was used the step response of the model starting from an initial state, which response can be generated from the Simulink representation of the plant. This approach leaded to a correct behavior of the controller.

After having done the two controllers design phases, in the next step there were compared in simulations the outputs from the two controlled systems.

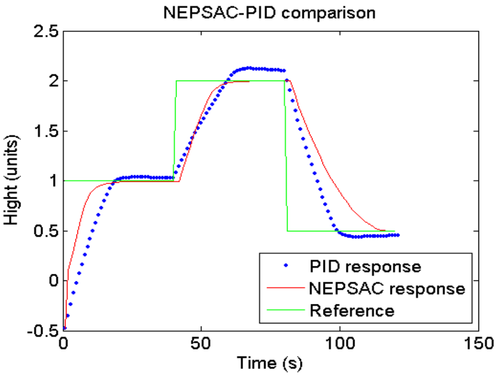

Figure 12 Comparison of the PID and NEPSAC simulations

As it can be seen on the Figure 12 the response of the PID controlled system is fast, doing well around the operation point where it was designed, but it gets with a slow convergence to the set point in different operation points. On the other hand, the nonlinear predictive control algorithm handles in correct mode the nonlinearities in the system.

E. *The real time experimental set up*

The experiment was performed for different controllers the control algorithm running on a PC. The link between the plant and the PC was done with the IPCON data acquisition board, with a sampling time of 1 second. In case of the NEPSAC algorithm it was necessary to check the real time constraints of the algorithm which is a rather time consuming one. On the Figure 13 there is shown the result of the algorithm profiling. As it can be seen, the algorithm runs in about 0.5 seconds, in this way satisfying the real time constraints for the control application with the 1 second sampling time.

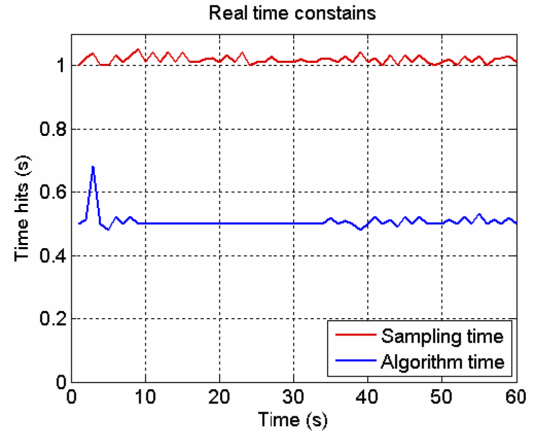

Figure 13 The algorithm running time and the sampling time

The result of the PID controlled system is presented on the Figure 14. As it can be seen on this figure, the controller has quite a good response near the operation point where it was tuned, but in other points, the response is pretty poor. Also the control effort has great variations compared to the NEPSAC case.

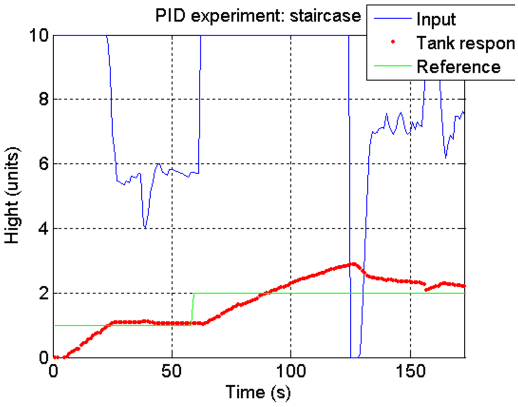

Figure 14 The PID response in the experiment

On the Figure 15 there is shown the output of the controlled system with the NEPSAC controller in case of the prediction horizon N2=5 and the control horizon N1=1.

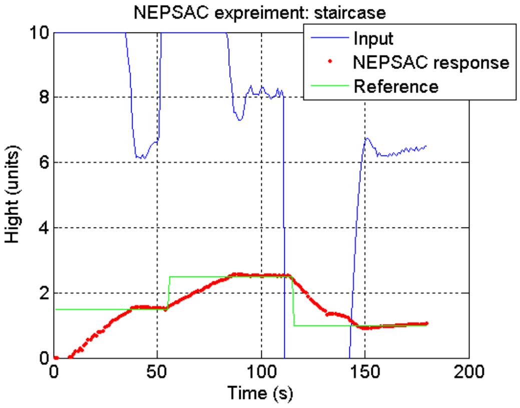

Figure 15 the NEPSAC response of the experiment

Although the response of the controlled plant with these tuning parameters is slower than in the case of the PID controller, there are no stationary errors in this case. The input to the plant has less variation than in the previous case, and in this way the control effort is also less.

Conclusions

As a conclusion it can be said, that the NEPSAC control algorithm is performing well in different operation points in case of nonlinear plants, and it can be also estimated the computational effort for this type of control algorithm as this is a deterministic one.

Acknowledgment

At this point I would like to thank for the interesting and helpful courses of Prof. Robin De Keyser about the predictive control, and to Prof. Ioan Nascu and his assistant Tudor Buzdugan to their help in the experimental set up.

References

[1] R. De Keyser: Model Based Predictive Control. Invited Chapter in ”UNESCO Encyclopedia of Life Support Systems (EoLSS) ”. EoLSS Publishers Co Ltd, Oxford, 2003 (www.eolss.net).
[2] Maciejowski J, *Predictive Control with Constraints.* Prentice Hall London, 2002.
[3] Ljung, L., *System Identification - Theory for the User*, Prentice Hall, Upper Saddle River, N.J., 2nd edition, 1999. pp.68-73.
[4] De Keyser R., C. Ionescu, "FRtool : a frequency response tool for CACSD in MatLab", IEEE Conf. on Computer Aided Control Systems Design (CACSD-CCA-ISIC) Muncher, Germany, pp. 2276-2280, 2006
[5] www.mathworks.com