*Proceedings of the 10th Mediterranean Conference*
*on Control and Automation - MED2002*
*Lisbon, Portugal, July 9-12, 2002.*

# UNSCENTED KALMAN FILTER IN ADAPTIVE NEURAL MODEL-BASED PREDICTIVE CONTROL

**P. Gil [†]*, J. Henriques [†], H. Duarte-Ramos *, A. Dourado [†]**

[†] CISUC-Informatics Engineering Department,
UCPólo II, Pinhal de Marrocos, 3030 Coimbra, Portugal
Phone: +351 239790000, Fax: +351 239701266
e-mail: pgil@dei.uc.pt

[*] Electrical Engineering Department, UNL
2825-114 Monte de Caparica, Portugal
Phone: +351 212948545, Fax: +351 212948532

## Abstract

An adaptive model-based predictive control scheme is proposed for non-linear systems. This methodology exploits the non-linear modelling capabilities of non-linear state-space neural networks and the online weights adjustment by means of an unscented Kalman filter. Results from experiments show evidences on its good tracking performance even when the system's dynamics change.

**Index Terms**: State-space neural networks; on-line training; model-based predictive control; dual unscented Kalman filter.

## 1. Introduction

In the last few years the development and application of artificial neural networks methodologies in a wide variety of fields, such as in filtering, modelling and control, have witnessed an increasing growth. At the basis for this scenario were undoubtedly the demonstrated approximation capabilities of multi-layer networks [1] and the fact that they are less sensitive to noise and more fault tolerant than other non-linear mappings, such as polynomial or splines models [2].

In system identification, feedforward networks have been used to represent spatio-temporal information by means of the tapped-delay-line method. Nevertheless, it is well established that dynamic neural structures containing a state feedback not only may provide computational advantages, since the corresponding models are likely to possess a smaller number of parameters, but also they can describe a larger class of dynamic systems than the input-output counterparts. Furthermore, it is not always possible to derive an input-output model globally equivalent to a specific state-space representation.

Regarding the control systems design, neural networks can basically be incorporated in two different ways: as a neuro-controller or providing a model of the plant to be placed under control. Within the first group of techniques, a neural network may be used to emulate the behaviour of a particular controller either a human controller or another automatic controller, as well as providing the system's inverse model. For instance, Cavagnari *et al.* [3] have implemented a non-linear receding horizon controller by training a neural network in a supervised way.

With respect to the inverse-model-based techniques, several control structures have been proposed so far [4], all having in common the underlying idea that the process and the controller would result in an identity mapping between the desired output and the system's output.

In another different direction, as a result of their inherent ability to approximate arbitrary non-linear vector functions, neural networks have extensively been used within model-based control techniques providing black-box models from which suitable control-laws can be derived. For instance, in the context of predictive control methodologies, Sørensen *et al.* [5] reported to use a neural input-output black-box model in a generalised predictive control framework. Bearing in mind the advantages of recurrent neural networks for modelling dynamic systems Gil *et al.* [6] reported an implementation of a non-linear neural state-space model based predictive controller (MPC) guaranteeing free static offset by incorporating a pre-filter in the control loop.

In the present work, instead of resorting to an offset compensation, it is proposed to use a real-time neural network weights adaptation within a MPC framework. For system identification, a batch technique is first applied in the estimation of the neural state-space parameters. Next, at each discrete time and using the most recent input-output sample, by means of a dual unscented Kalman filter, not only a new set of parameters is provided to the neural state space predictor, but also a state estimation is made available, which is crucial to the control optimisation stage. Experiments on a laboratory heating system are used to highlight the merits of the proposed methodology, particularly when the controlled system's dynamics is forced to change.

# 2. Neural Network Modelling: Topology and Training

Consider a general deterministic discrete-time non-linear plant described by (1) for which is required to find a model structure and a particular parameterisation:

$$\begin{aligned} x(k+1) &= f\left(x(k),u(k),k\right) \\ y(k) &= h(x(k),k) \end{aligned} \tag{1}$$

where $f:\Re^n\times\Re^m\times\Re\rightarrow\Re^n$ and $h:\Re^n\times\Re\rightarrow\Re^p$ are non-linear functions, assumed to be smooth; $u(k)\in\Re^m$, $x(k)\in\Re^n$ and $y(k)\in\Re^p$ are, respectively, the input vector, the state vector and the output vector, at a discrete time $k$.

## 2.1. Architecture

In this work the black-box model is derived by means of a hybrid recurrent neural network comprising 3 layers, as depicted in Fig. 1. The input and output layers incorporate as much neurons as the number of inputs and outputs of the system, whereas the number of neurons in the hidden layer should be the most appropriate to get a good approximation. In view of selecting the optimal number of hidden neurons, one should mention, without going into details, that understanding the relationship between generalization performance and training error is of crucial importance.

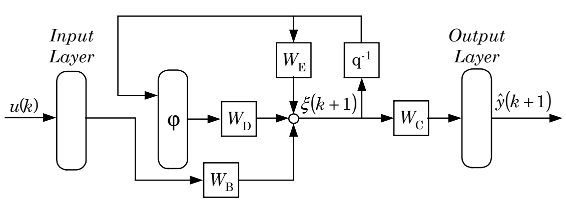


Fig. 1. State-space neural network block diagram.

In this neural network topology $\xi\in\Re^{Nn}$ denotes the neural state-space vector, $\hat{y}\in\Re^{No}$ is the neural output, $u\in\Re^{Ni}$ is the neural external input; $Nn$, $No$ and $Ni$ are, respectively, the number of neurons in the hidden layer, output layer and input layer; $\varphi$ is a non-linear activation function, $q^{-1}$ denotes the backward shift operator. Additionally, the synaptic weights between neurons: $W_B$, $W_C$, $W_D$ and $W_E$ are real-valued matrices having appropriate dimensions.

The resulting dynamic model may be written in the state-space form as:

$$\begin{aligned} \xi(k+1) &= W_D\tanh(\xi(k))+W_E\,\xi(k)+W_B\,u(k); \quad \xi(0)=\xi_0 \\ \hat{y}(k) &= W_C\,\xi(k) \end{aligned} \tag{2}$$

assuming a hyperbolic tangent as a non-linear activation function. It consists in a combination of a linear and non-linear terms, which are likely to perform better than those having only non-linear ones [7].

## 2.2. Training

The evaluation of a suitable parameterisation for a particular model structure is central in many fields of engineering. When a representative observation data set is available in advance, batch learning techniques can be exploited to infer such a relationship. In this context, the back-propagation algorithm, in its various forms has extensively been used to adjust the weights of neural networks. However, since they are all based on a gradient search direction their performance is somehow poor particularly close to local minima compared to second order search direction algorithms.

In the online estimation of recurrent neural networks weights, gradient descent based algorithms have also been applied. Recently, rooted in the intuition that training a neural network can be regarded as a non-linear parameter estimation a number of second-order algorithms have emerged such as the recursive least squares and Kalman filter.

In the present work the Levenberg-Marquardt algorithm is applied to the offline minimisation problem, assuming the following cost function:

$$J(w)=\sum_{k=1}^{N}\left[y(k)-\hat{y}(k|w)\right]^T\left[y(k)-\hat{y}(k|w)\right] \tag{3}$$

where $w\in\Re^{Nw}$ denotes the parameterisation vector and $N$ the total the number of training data.

According to this algorithm, the updating law is given by:

$$\Delta w=-\left(\widetilde{\mathcal{H}}+\lambda\mathrm{I}\right)^{-1}\nabla J(w) \tag{4}$$

with $\widetilde{\mathcal{H}}$ the approximation of the Hessian matrix of cost-function $J(w)$, $\nabla J(w)$ denotes its gradient, $\lambda \in \Re^+$ and $\mathrm{I}$ is an identity matrix of appropriate dimensions.

For online training the neural network it is used a Kalman filter approach based on the unscented transformation (UT) [8]. The unscented transformation enables to compute the statistics properties of a random variable propagated through a non-linear mapping.

Consider then a $Nw$-dimensional real-valued random variable $w$ with mean $\overline{w}$ and covariance matrix $\mathrm{P}_{ww}$ and suppose it is required to predict the mean and the covariance of $y \in \Re^q$ given as:

$$y = \hbar(w) \tag{5}$$

with $\hbar : \Re^{Nw} \to \Re^q$.

First, a set of $2Nw+1$ pairs of weights and translated sigma points $(\Gamma_i, \omega_i)$ is formed according to (6), in such a way that the mean and the covariance of $w$ are preserved.

$$\begin{cases} \left(\kappa(N+\kappa)^{-1}, \overline{w}\right), i = 0 \\ \left(0.5(N+\kappa)^{-1}, \overline{w} + \sqrt{(N+\kappa)(P_{ww})_i}\right), i = 1, \ldots, Nw \\ \left(0.5(N+\kappa)^{-1}, \overline{w} - \sqrt{(N+\kappa)(P_{ww})_i}\right), i = 1, \ldots, Nw \end{cases} \tag{6}$$

with $(\mathrm{P}_{ww})_i$ the $i^{\text{th}}$ column or row of covariance matrix $\mathrm{P}_{ww}$ and $\kappa$ a scaling parameter. These sigma points are then propagated through the non-linear mapping,

$$\Upsilon_i = \hbar(\omega_i) \tag{7}$$

and the corresponding mean and covariance computed as follows:

$$\begin{aligned} \overline{y} &= \sum_{i=0}^{2Nw} \Gamma_i \Upsilon_i \\ \mathrm{P}_y &= \sum_{i=0}^{2Nw} \Gamma_i (\Upsilon_i - \overline{y})(\Upsilon_i - \overline{y})^T \end{aligned} \tag{8}$$

The dual unscented Kalman filter (DUKF) consists in the UT application to the recursive estimation of both the states and the parameters of a non-linear discrete-time dynamic system [9]. In the present work, the non-linear system is assumed to be described as:

$$\begin{aligned} x(k+1) &= f[x(k), u(k), w(k), k] + \upsilon(k) \\ z(k) &= h[x(k), w(k), k] + \eta(k) \end{aligned} \tag{9}$$

where $z \in \Re^p$ denotes the observation vector; $\upsilon \in \Re^n$ is a Gaussian process noise with covariance $Q$ and $\eta \in \Re^p$ a Gaussian measurement noise with covariance $R$.

Like all Kalman filter based algorithms, in DUKF approach the estimates are computed in two stages: in the time update stage one step-ahead predictions are performed whereas in the measurement update phase it is provided a correction to the *a priori* estimates on the basis of the most recent sample. The dual filter equations are given by:

**Weights estimation**

*Time update*:

$$\begin{aligned} \Omega(k \mid k-1) &= \Omega(k-1 \mid k-1) \\ \mathrm{P}_{ww}(k \mid k-1) &= \mu^{-1}\, \mathrm{P}_{ww}(k-1 \mid k-1) \\ \mathcal{Z}^w(k \mid k-1) &= h(\hat{x}(k-1 \mid k-1), \Omega(k \mid k-1), k) \\ \hat{z}_w(k \mid k-1) &= \sum_{i=0}^{2Nw} \Gamma_i^w \mathcal{Z}_i^w(k \mid k-1) \end{aligned} \tag{10}$$

*Measurement update*:

$$\begin{aligned} P_{\nu\nu}^w(k \mid k-1) &= \sum_{i=0}^{2Nw} \Big\{ \Gamma_i^w \left[\mathcal{Z}_i^w(k \mid k-1) - \hat{z}_w(k \mid k-1)\right] \\ &\quad \left[\mathcal{Z}_i^w(k \mid k-1) - \hat{z}_w(k \mid k-1)\right]^T \Big\} + R \\ P_{wz_w}(k \mid k-1) &= \sum_{i=0}^{2Nw} \Big\{ \Gamma_i^w \left[\Omega_i(k \mid k-1) - \hat{w}(k \mid k-1)\right] \\ &\quad \left[\mathcal{Z}_i^w(k \mid k-1) - \hat{z}(k \mid k-1)\right]^T \Big\} \end{aligned} \tag{11}$$

$$\begin{aligned} \mathcal{K}^w(k) &= \mathrm{P}_{wz_w}(k \mid k-1)\left(P_{\nu\nu}^w(k \mid k-1)\right)^{-1} \\ \hat{w}(k \mid k) &= \hat{w}(k \mid k-1) + \mathcal{K}^w(k)[z(k) - \hat{z}(k \mid k-1)] \\ \mathrm{P}_{ww}(k \mid k) &= \mathrm{P}_{ww}(k \mid k-1) - \mathcal{K}^w(k) \mathrm{P}_{\nu\nu}^w(k \mid k-1) \mathcal{K}^w(k)^T \end{aligned}$$

**States estimation**

*Time update*:

$$\begin{aligned} \mathcal{X}(k \mid k-1) &= f(\mathcal{X}(k-1 \mid k-1), u(k-1 \mid k-1), \\ &\qquad w(k-1 \mid k-1), k) \\ \hat{x}(k \mid k-1) &= \sum_{i=0}^{2n} \Gamma_i^x \mathcal{X}_i(k \mid k-1) \\ \mathcal{X}(k \mid k-1) &= f(\mathcal{X}(k-1 \mid k-1), u(k-1 \mid k-1), \\ &\qquad w(k-1 \mid k-1), k) \\ \mathrm{P}_{xx}(k \mid k-1) &= \sum_{i=0}^{2n} \Gamma_i^x \left[\mathcal{X}_i(k \mid k-1) - \hat{x}(k \mid k-1)\right] \\ \mathcal{Z}^x(k \mid k-1) &= h(\mathcal{X}(k \mid k-1), w(k-1 \mid k-1), k) \\ \hat{z}_x(k \mid k-1) &= \sum_{i=0}^{2n} \Gamma_i^x \mathcal{Z}_i^x(k \mid k-1) \end{aligned} \tag{12}$$

*Measurement update*:

$$\begin{aligned} P_{\nu\nu}^x(k \mid k-1) &= \sum_{i=0}^{2n} \Big\{ \Gamma_i^x \left[\mathcal{Z}_i^x(k \mid k-1) - \hat{z}_x(k \mid k-1)\right] \\ &\quad \left[\mathcal{Z}_i^x(k \mid k-1) - \hat{z}_x(k \mid k-1)\right]^T \Big\} + R \\ P_{\nu\nu}^x(k \mid k-1) &= \sum_{i=0}^{2n} \Big\{ \Gamma_i^x \left[\mathcal{Z}_i^x(k \mid k-1) - \hat{z}_x(k \mid k-1)\right] \\ &\quad \left[\mathcal{Z}_i^x(k \mid k-1) - \hat{z}_x(k \mid k-1)\right]^T \Big\} + R \\ P_{xz_x}(k \mid k-1) &= \sum_{i=0}^{2n} \Big\{ \Gamma_i^x \left[\mathcal{X}(k \mid k-1) - \hat{x}(k \mid k-1)\right] \\ &\quad \left[\mathcal{Z}_i^x(k \mid k-1) - \hat{z}_x(k \mid k-1)\right]^T \Big\} \end{aligned} \tag{13}$$

$$\begin{aligned} \mathcal{K}^x(k) &= \mathrm{P}_{xz_x}(k \mid k-1)\left(P_{\nu\nu}^x(k \mid k-1)\right)^{-1} \\ \hat{x}(k \mid k) &= \hat{x}(k \mid k-1) + \mathcal{K}^x(k)[z(k) - \hat{z}(k \mid k-1)] \\ \mathrm{P}_{xx}(k \mid k) &= \mathrm{P}_{xx}(k \mid k-1) - \mathcal{K}^x(k) \mathrm{P}_{\nu\nu}^x(k \mid k-1) \mathcal{K}^x(k)^T \end{aligned} \tag{14}$$

where $\mu$ denotes the forgetting factor, $\Omega$ the sigma point matrix of *w,* $\Gamma$ the corresponding weight vector, $\mathcal{K}$ the Kalman gain and $\mathcal{X}$ the sigma point matrix of *x*.

# 3. Constrained Model-based Predictive Control

Model-based predictive control is a discrete-time technique for which an explicit dynamic model of the plant is used to predict the system's outputs over a finite prediction horizon $P$ when control actions are manipulated over a finite control horizon $M$ .

At time step $k$ , the optimiser computes an open-loop control action sequence in such a way that the predicted output follows a pre-specified reference while taking into account possible hard and soft constraints. Only the current control action $u(k \mid k)$ is actually fed to the plant over time $[k, k+1)$. Subsequently, the prediction and control horizons are shifted ahead by one step and a new optimisation problem is solved, taking into account the most recent measurements from the plant.

Consider the first order Taylor expansion of a general discrete-time non-linear system (1) as given by:

$$\begin{aligned} x(k+1) &= \Phi x(k) + \Xi u(k) + \mathrm{E} \\ y(k) &= \mathrm{H} x(k) \end{aligned} \tag{15}$$

where $\Phi \in \Re^{n\times n}$, $\Xi \in \Re^{n\times m}$ and $\mathrm{H} \in \Re^{p\times n}$ denote, the state, the input and the output matrices, respectively; $\mathrm{E} \in \Re^{n}$ is the first term of the Taylor series expansion.

The constrained open-loop optimisation problem can be stated as follows:

$$\begin{aligned} \min_u J = \min_u \Big\{ & \sum_{i=1}^{P} \| y(k+i \mid k) - r(k+i) \|_{Q_i}^2 \\ & + \sum_{i=0}^{P-1} \| u(k+i \mid k) \|_{\mathcal{R}_i}^2 + \sum_{i=0}^{M-1} \| \Delta u(k+i \mid k) \|_{\mathcal{S}_i}^2 \Big\} \end{aligned} \tag{16}$$

subject to the system dynamics (15) and to the following constraint inequalities:

$$\begin{aligned} y_{\min} \le y(k+i|k) \le y_{\max},\ & i = 1,\ldots,P,\ k \ge 0 \\ u_{\min} \le u(k+i|k) \le u_{\max},\ & i = 0,\ldots,P-1,\ k \ge 0 \\ \left| \Delta u(k+i|k) \right| \le \Delta u_{\max},\ & i = 0,\ldots,M-1,\ k \ge 0 \\ \left| \Delta u(k+i|k) \right| = 0,\ & i = M,\ldots,P-1,\ k \ge 0 \end{aligned} \tag{17}$$

with $Q_i \in \Re^{p\times p}$, $\mathcal{R}_i \in \Re^{m\times m}$, $\mathcal{S}_i \in \Re^{m\times m}$; $\Delta u \in \Re^{m}$ is the control action increment vector and $r \in \Re^{p}$ the reference vector.

As a result of the optimisation problem convexity any particular solution is a global optimum and hence the open-loop optimal control problem can be rewritten as a quadratic programming problem (18).

$$\begin{aligned} &\textit{minimise} && J(\Delta\tilde{u}) = h^T \Delta\tilde{u} + \frac{1}{2}\Delta\tilde{u}^T \mathcal{H}\, \Delta\tilde{u} \\ &\textit{Subject to} && A^T \Delta\tilde{u} \le b \end{aligned} \tag{18}$$

where $A \in \Re^{mM\times(4mM+2pP)}$, $b \in \Re^{(4mM+2pP)}$; $\Delta\tilde{u} \in \Re^{mM}$ denotes the extended control increment vector over the control horizon. Expressions for the cost function gradient $h \in \Re^{mM}$ and Hessian $\mathcal{H} \in \Re^{mM\times mM}$ can be found in [6].

# 4. Experiments

## 4.1. Process Description

The laboratory process set-up used for assessing the adaptive MPC technique performance is the heating system depicted schematically in Fig. 2. Air drawn from the local atmosphere is forced to circulate by means of a centrifugal fan through a length of duct, being heated in a heater grid just after the inlet. This is a non-linear process with a pure time delay that depends on the position of the temperature sensor and the air flow rate, which is a function of the damper position $\phi$. The input to the system is a voltage on the heating device consisting of a mesh of resistor wires and the system's output is the outlet air flow temperature.

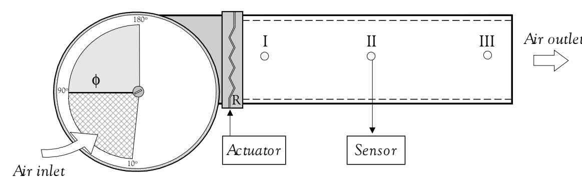

Fig. 2. Heating system schematics.

## 4.2. Results

For offline plant identification purpose, open-loop experiments have been carried out on the heating system in order to collect input-output data to be used in the parameter estimation stage. The heating system damper was set to $\phi = 20°$ and the detector probe placed at the intermediate position (140 *mm*).

The experiments were conducted by feeding step and pseudo random binary signals to the system and choosing a sampling interval of 0.15 second. Two of the collected records were picked out: one for neural network training and the other for cross-validation.

The structure of the neural network is that of presented in Fig. 1, having one neuron in both the input and output layers and three neurons in the hidden layer. After being trained, this neural model is able to predict quite well the behaviour of the plant, as can be inferred from Fig. 3.

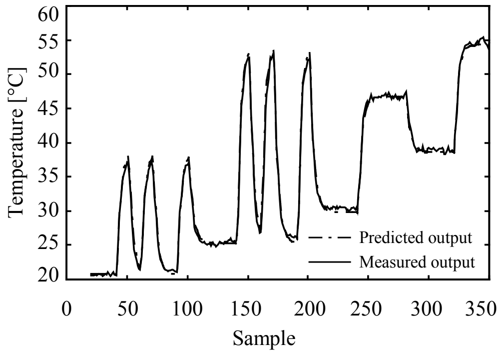


Fig. 3. Model validation.

The neural state-space predictor is used within the local instantaneous linear model-based predictive control (LIMPC) framework as a seed for linear discrete-time models. At each sampling time, a new model is provided to the optimiser by means of a first-order Taylor series expansion.

The constrained open-loop optimal control (18) is solved by taking the prediction horizon and the control horizon, respectively, as $P = 3$ and $M = 1$ time steps, choosing 0.15 second for sampling time and imposing the following constraints:

$$\begin{aligned} &15°\mathrm{C} \le y \le 60°\mathrm{C} \\ &0\ \mathrm{V} \le u \le 10\ \mathrm{V} \\ &\left|\Delta u\right| \le 2.0\ \mathrm{V} \end{aligned} \tag{19}$$

The cost functional weight matrices were chosen as:

$$\begin{aligned} &\mathcal{Q}_i = \begin{cases} 30, & i < P \\ 100, & i = P \end{cases} \\ &\mathcal{R}_i = 10^{-4}\ , \quad \forall i \\ &\mathcal{S}_i = 10^{-4}\ , \quad \forall i \end{aligned} \tag{20}$$

In the first group of control experiments the neural non-linear state-space model with fixed weights, i.e. not adjusted online, is used within the LIMPC framework together with an unscented Kalman filter observer. This control scheme is then applied to control the laboratory heating system, being one of these experiments plotted in Fig. 4. As can be observed for some particular operating ranges the control system is unable to remove completely static offsets. These deviations are mainly attributed to model/plant mismatch and thus by further training the neural network, making use of fresh data, one should be able to reduce those modelling errors and, in turn, achieving a better control performance.

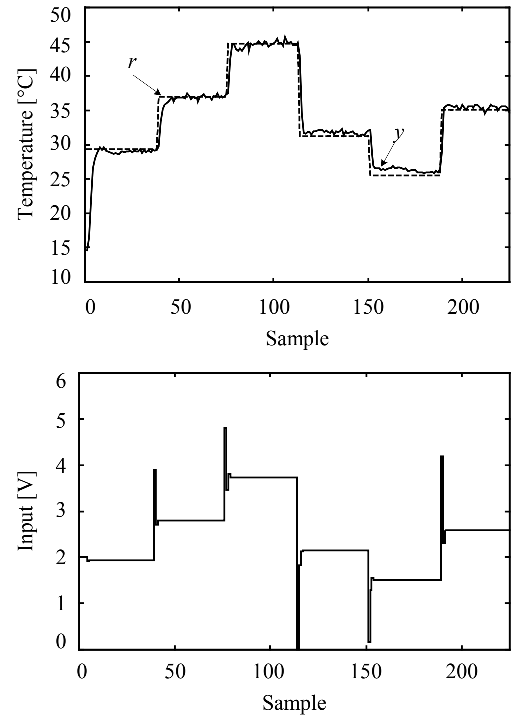


Fig. 4. Non-adaptive LIMPC

In fact, as shown in Fig. 5, the proposed adaptive model-based predictive controller not only guarantees a stable time response but also the deterministic steady-state errors are significantly reduced or even removed. In this experiment a dual unscented Kalman filter was used for updating the neural network weights and estimating the system states.

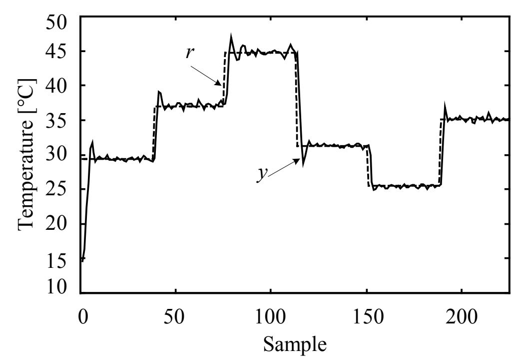


Fig. 5.a) Adaptive LIMPC – Output.

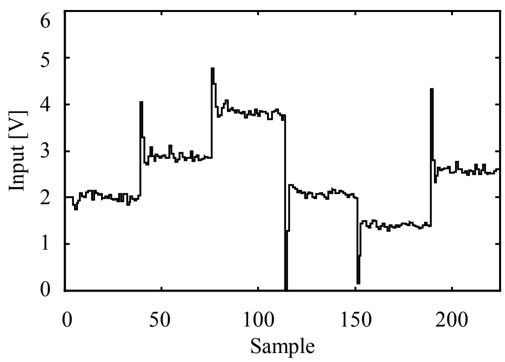


Fig. 5.b) Adaptive LIMPC – Control action.

In order to assess the performance of the adaptive model-based predictive control scheme in time-variant plants, the laboratory heating system dynamics was allowed to change during the control experiments. For this purpose, the blower inlet throttle was changed to $\phi = 30°$ at instant 20 second, returning to its original position ($\phi = 20°$) at instant 40 second. Fig. 6 compares the air flow temperature for the adaptive and non-adaptive model-based predictive controllers. As can be seen, due to the online neural network weights adaptation, the new system dynamics has been captured in a suitable way being a decisive factor in removing those static offsets.

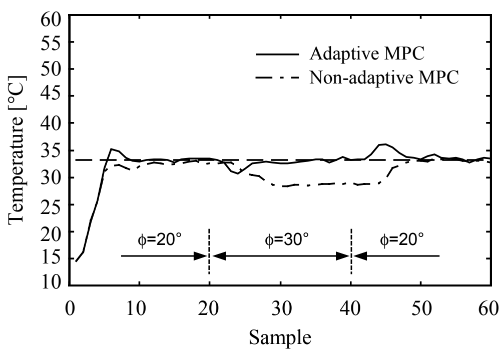


Fig. 6. MPC - Changing the system's dynamics.

## 5. Conclusions

In this paper an adaptive constrained model-based predictive control scheme is proposed on the basis of an online model parameters updating. The black-box model is derived by means of state-space neural network and the online training and states estimation is based on an unscented Kalman filter.

This approach was successfully applied to the control of a laboratory heating system whose dynamics is forced to change during the experiments. Furthermore, as it is clear from comparative studies, the proposed adaptive MPC technique outperforms that of a standard MPC scheme. Finally, since the underlying control design is quite general and flexible this adaptive scheme offers significant potential benefits for time-variant systems control.

## Acknowledgements

This work is partially supported by POSI - Programa Operacional da Sociedade de Informação of Portuguese Fundação para a Ciência e Tecnologia and European Union FEDER and PRAXIS XXI ALCINE Project.

## References

[1] K. Hornik, M. Stinchcombe and H. White, "Multilayer Feedforward Networks are Universal Approximators," *Neural Networks*, vol.2, pp. 359-366, 1989.

[2] K. Narendra, "Neural Networks for Control: Theory and Practice," Proc. of the IEEE, vol. 84, pp. 1385-1406, 1996.

[3] L. Cavagnari, L. Magni and R. Scattolini, "Neural Network Implementation of Nonlinear Receding-Horizon Control," *Neural Computing & Applications*, vol. 8, pp. 86-92, 1999.

[4] M. Hussain, "Review of the applications of neural networks in chemical process control – simulation and online implementation," *Artificial Intelligence in Engineering*, vol.13, pp. 55-68, 1999.

[5] P. Sørensen, M. Nøgaard, O. Ravn and N. Poulsen, "Implementation of Neural Network Based Non-linear Predictive Control," *Neurocomputing*, vol. 28, pp. 37-51, 1999.

[6] P. Gil, J. Henriques, A. Dourado and H. Duarte-Ramos, "Constrained Neural Model Predictive Control with Guaranteed Free Offset," *Proc. 2000 IEEE Int. Conf. on Industrial Electronics, Control and Implementation – IEECON*, Nagoya, Japan, pp. 1991-1996, 2000.

[7] D. Pham and L. Xing, "Neural Networks for Identification, Prediction and Control," Springer-Verlag London Limited, 1995.

[8] S. Julier, J. Uhlmann and F. Durrant-Whyte, "A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators," *IEEE Trans. Aut. Control*, vol. 45, pp. 477-482, 2000.

[9] E. Wan and R. Merwe, "The Unscented Kalman Filter for Nonlinear Estimation," *Proc. IEEE Symp. 2000 Adaptive Systems for Signal Processing, Comunication and Control*, Alberta, Canada, 2000.