World Academy of Science, Engineering and Technology
International Journal of Chemical and Molecular Engineering
Vol:2, No:10, 2008

Affine Radial Basis Function Neural Networks for the Robust Control of Hyperbolic Distributed Parameter Systems

Eleni Aggelogiannaki, and Haralambos Sarimveis

***Abstract*—**In this work, a radial basis function (RBF) neural network is developed for the identification of hyperbolic distributed parameter systems (DPSs). This empirical model is based only on process input-output data and used for the estimation of the controlled variables at specific locations, without the need of online solution of partial differential equations (PDEs). The nonlinear model that is obtained is suitably transformed to a nonlinear state space formulation that also takes into account the model mismatch. A stable robust control law is implemented for the attenuation of external disturbances. The proposed identification and control methodology is applied on a long duct, a common component of thermal systems, for a flow based control of temperature distribution. The closed loop performance is significantly improved in comparison to existing control methodologies.

***Keywords*—**Hyperbolic Distributed Parameter Systems, Radial Basis Function Neural Networks, H_∞ control, Thermal systems.

I. Introduction

IN DPS, inputs, outputs as well as parameters may change temporally and spatially due to diffusion, convection and/or conduction phenomena. These systems are mathematical described by sets of PDE, where time and spatial coordinates are the independent variables. DPSs are quite common in industries (heat exchangers, tubular reactors, fluidized beds and crystallizers). For example, long ducts for fluid transport are key components in many highly utilized industrial thermal systems, where they are often used in tandem with other major components such as, for example, heat exchangers. The energy conservation law in such a system is mathematically described by a PDE from which the temperature (the dependent variable) can be determined as a function of both time and space. If negligible diffusive phenomena and constant fluid velocity throughout the duct are assumed, the resulting PDE is one-dimensional and also belongs to a particular class of distributed parameter systems, known as hyperbolic DPSs.

Control of DPSs has attracted a lot of research during the last years. However, most methodologies focus on parabolic systems, since they can be sufficiently described by finite order approximations. Contrary to that, in hyperbolic systems all the eigenmodes of the corresponding spatial differential operator are of the same level of energy and so an infinite order model is necessary to accurately describe the system dynamics. References [1]-[2] proposed a methodology to model and control hyperbolic systems based on the combination of the method of characteristics and the sliding mode techniques. Reference [3] introduced the concept of characteristic index for the synthesis of states and output feedback controllers. Robust schemes to overcome model uncertainties and unmodeled dynamics have also been proposed in [4]-[5]. Reference [6] used the numerical solution of the Lagrangian form of the PDE in a long duct to analyze the stability of a proportional-integral (PI) controller for the flow-based control of the outlet temperature. Feedback and robust control laws that are based on the method of characteristics were also proposed in [7]. Nonlinear controller was used in [8] to control hot spot temperature in plug flow reactor. Finally, model predictive control methodologies have also been proposed for hyperbolic DPSs in a number of publications [9]-[11].

All the aforementioned control techniques in some way presuppose that the PDEs are known or known subject to uncertainties [4], [5]. If we suppose that no knowledge of the PDEs exists but only input-output data are available to derive a control law, then the identification of the system is firstly required. Among other non linear empirical modeling methodologies, neural networks have also been used for the identification of DPSs in a number of references [12]-[14].

In this work, we implement a RBF neural network for the identification of first order hyperbolic DPSs. Additionally, we utilize an H_∞ controller of proved robust stability that is available in literature [15]-[17]. More analytically, an affine RBF neural network is firstly developed, based purely on experimental data, according to the training method described in [18]. This neural network can predict the process output variables at different locations of the DPS, if it is supplied with a number of input past values and the location the prediction is required. Afterwards, the empirical model is transformed to a nonlinear state space representation, by considering temperature at specific locations as well as past values of velocity as state variables. In this representation the unmodeled dynamic behavior is also included as an additional

Manuscript received August 28, 2006. This work is financial supported by Alexander S. Onassis Public Benefit Foundation.

Authors are with the School of Chemical Engineering, National Technical University of Athens, Greece.

H. Sarimveis is the corresponding author (phone: +30-210-7723237; fax: +30-210-7723138; e-mail: hsarimv@central.ntua.gr).

bounded term. This new representation allows the straightforward implementation of the robust H_∞ control configuration [15]-[17] that ensures stability. The overall proposed methodology is used for the identification and control of the temperature distribution along a duct (key component of thermal systems), when the fluid velocity is considered as the manipulated variable. The performance of the neural network identification method is tested over a number of validation data and through step tests with encouraging results. The same disturbance attenuation examples used in [6] are also used in this work in order to test the control configuration. The advantages of the proposed methodology are that no discretization is performed online, since the model can estimate temperature at any location and that stability is ensured in a region of the origin.

The rest of the paper is synthesized as follows: In section II, the structure of the RBF neural network is described and in section III the development of a state–space representation is analyzed. The nonlinear robust control law for temperature and the thermal system that will be used as test example are presented in sections IV and V respectively. The efficiency of the RBF identification method and the performance of the controller are examined in section VI and finally, the conclusions of this work are summarized in the last section of the paper.

II. RBF Neural Network for DPSs

RBF neural networks have been widely used for the simulation and control of nonlinear processes. In order to describe a process using a neural network, a number of input-output data should be collected using appropriate sampling interval t_s. These data are then involved in the training procedure in order to determine the neural network parameters. RBF networks consist of three layers, namely the input layer, the hidden layer and the output layer. Training of an RBF network actually means the determination of the number of nodes in the hidden layer, the hidden node centers and the output weights, so that the deviation between the predicted and the true values of the output variables over the set of the available data be minimal. The produced network is validated using a similar set of input-output data that is not involved during the training procedure.

The training method used in this work is based on the fuzzy partition of the input space, which is produced by defining a number of triangular fuzzy sets on the domain of each input variable. The centers of these fuzzy sets form a multidimensional grid on the input space. A rigorous selection algorithm chooses the most appropriate knots of the grid, which are then used as the hidden node centers in the produced RBF network model. The so called fuzzy means training method does not need the number of centers to be fixed before the execution of the method. Due to the fact that it is a one-pass algorithm, it is extremely fast even in the case of a large database of input-output training data. One additional advantage is that the utilized training algorithm needs only one tuning parameter, namely the number of fuzzy sets that are utilized to partition each input dimension. The fuzzy means training methodology is described in details in [18].

The simulation and control of DPSs requires two modifications of the structure of classical RBF networks. Firstly, the network should be able to estimate the dynamic behavior of controlled variables at any spatial point along the duct. Secondly, to be able to guarantee robust stability (See section IV), the nonlinear model should have an affine structure. To achieve these properties, two different vectors are considered to be the neural network inputs. The first input is an $m\times1$ vector that contains the most recent value of the manipulated variables, while the second vector contains another $m\cdot(np\text{-}1)$ past values of process inputs and the location that the estimation of its outputs is required. Fig. 1 presents the aforementioned structure of the proposed RBF network. More analytically the first and the second inputs would be:

$$\mathbf{u}(t-1)=\left[u_1(t-1)\quad u_2(t-1)\quad \ldots\quad u_m(t-1)\right]^T \tag{1}$$

$$\mathbf{U}(t-1,x_j)=\left[\mathbf{u}^T(t-2)\ \ldots\ \mathbf{u}^T(t-np)\quad x_j\right]^T \tag{2}$$

where, $u_i(t-1)$ i=1,…,m is the deviation from steady state for the i^{th} input at time level t-1, x_j j=1,…, ns is the location of the j^{th} sensor and ns the total number of sensors. The second input vector is filtered in the hidden layer according to the following equation:

$$z_c(t-1,x_j)=f\left(\left\|\mathbf{U}(t-1,x_j)-\mathbf{c}_c\right\|_2^2\right) \tag{3}$$

where z_c is the response of the c^{th} node, f is the Gaussian function, $\mathbf{c}_c$ is the center of the c^{th} node (of dimension $m\cdot(np\text{-}1)+1$) and C is the total number of hidden nodes. The responses of the hidden nodes and the first input are then weighted, also adding a bias, to give the estimation of the n controlled variables (also deviation from steady state) at the current time-level and at the specific spatial point $\hat{\mathbf{y}}_{RBF}(t,x_j)=\left[\hat{y}_{1,RBF}(t,x_j)\quad \ldots\quad \hat{y}_{n,RBF}(t,x_j)\right]$ as follows:

$$\hat{\mathbf{y}}_{RBF}(t,x_j)=\mathbf{w}_1\cdot\mathbf{u}(t-1)+\sum_{c=1}^{C}\mathbf{w}_{c+1}\cdot z_c(t-1,x_j)+\mathbf{w}_{C+2} \tag{4}$$

In the above equation $\mathbf{w}_{c+1}$, c=1,…,C is the $n\times1$ weight corresponding to the response of the c^{th} node, $\mathbf{w}_1$ is the $n\times m$ weight of the most recent value of fluid velocity and $\mathbf{w}_{C+2}$ is a $n\times1$ bias. These weights are calculated by solving a constrained least squares problem that makes sure that for zero inputs, the outputs' deviation from steady state will also be zero. From Eqs. (1)-(4), it becomes clear that the controlled variables estimation at one location does not depend on their estimated values at other locations, as it happens in a model based on finite differences. This allows treating the distributed system as a lumped one, and applying existing robust control

methodologies that take into account the existing model mismatch.

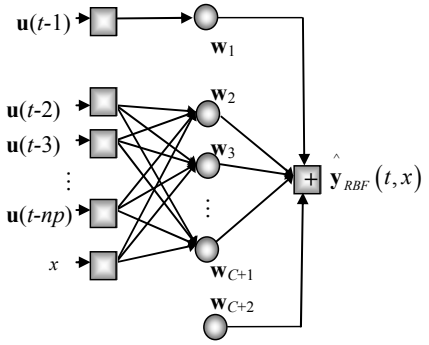

Fig. 1 The RBF neural network architecture for distributed parameter system identification

III. States Space Representation of the RBF Model

In this section a states space representation of the system is derived based on the previously described neural network. The required form of the states space system, in order to apply an existing robust control methodology [15], is:

$$\mathbf{s}(t+1)=\mathbf{f}_1(\mathbf{s}(t))+\mathbf{f}_2(\mathbf{s}(t))\mathbf{u}(t)+\mathbf{f}_3(\mathbf{s}(t))\mathbf{d}(t) \quad (5)$$

$$\boldsymbol{\psi}(t)=\begin{bmatrix}\mathbf{h}_1(\mathbf{s}(t))\\ \mathbf{u}(t)\end{bmatrix} \quad (6)$$

where $\mathbf{s}(t)$ is the vector of states at time level t, $\mathbf{f}_1(\mathbf{s}(t)),\mathbf{f}_2(\mathbf{s}(t)),\mathbf{f}_3(\mathbf{s}(t)),\mathbf{h}(\mathbf{s}(t))$ are continuous nonlinear functions of states with $\mathbf{f}_1(\mathbf{0})=\mathbf{0}$ and $\mathbf{h}(\mathbf{0})=\mathbf{0}$, $\mathbf{u}(t)$ is the vector of m manipulated variables, $\mathbf{d}(t)$ is the model mismatch and $\boldsymbol{\psi}(t)$ are the controlled variables.

Such a model can be developed if outputs $\mathbf{y}(t+1,x_j)$ at locations x_j, j=1,…, ns and the np-1 past values of each input (deviations from steady state) are chosen as state variables:

$$\begin{aligned}\mathbf{s}(t)&=\left[\mathbf{y}^T(t,x_1)\ \ \ldots\ \ \mathbf{y}^T(t,x_{ns})\ \ \mathbf{u}^T(t-1)\ \ \ldots\ \ \mathbf{u}^T(t-np+1)\right]^T\\&=\left[s_1(t)\ \ \ldots\ \ s_{n\cdot ns}(t)\ \ s_{n\cdot ns+1}(t)\ \ \ldots\ \ s_{n\cdot ns+m\cdot(np-1)}(t)\right]^T\end{aligned} \quad (7)$$

From Eq. (7) the number of states is equal to $n\cdot ns+m\cdot(np-1)$, namely the number of locations where estimations of process outputs are required plus np minus one past values of each of the m inputs. At the next time level t+1 the state vector is given by Eq. (8) since there is a mismatch $\mathbf{d}(t,x_j)$, j=1,..., ns between the neural network estimation and the actual process outputs. In Eqs. that follow the symbol $\mathbf{I}$ denotes the identity matrix and $\mathbf{0}$ the zeros matrix of proper dimensions.

$$\mathbf{s}(t+1)=\begin{bmatrix}\mathbf{y}(t+1,x_1)\\ \vdots\\ \mathbf{y}(t+1,x_{ns})\\ \mathbf{u}(t)\\ \vdots\\ \mathbf{u}(t-np+2)\end{bmatrix}^T=\begin{bmatrix}\sum_{c=1}^{C}\mathbf{w}_{c+1}\cdot z_c(t,x_1)+\mathbf{w}_{C+2}\\ \vdots\\ \sum_{c=1}^{C}\mathbf{w}_{c+1}\cdot z_c(t,x_{ns})+\mathbf{w}_{C+2}\\ \mathbf{0}_{m\times 1}\\ s_{n\cdot ns+m+1}(t)\\ \vdots\\ s_{n\cdot ns+m\cdot(np-2)}(t)\end{bmatrix}+\begin{bmatrix}\mathbf{w}_1\\ \vdots\\ \mathbf{w}_1\\ \mathbf{I}_m\\ \mathbf{0}_{m(np-2)\times m}\end{bmatrix}\mathbf{u}(t)+\begin{bmatrix}\mathbf{I}_{n\cdot ns}\\ \mathbf{0}_{m(np-1)\times n\cdot ns}\end{bmatrix}\begin{bmatrix}\mathbf{d}(t,x_1)\\ \vdots\\ \mathbf{d}(t,x_{ns})\end{bmatrix} \quad (8)$$

Consequently:

$$\mathbf{f}_1(\mathbf{s}(t))=\begin{bmatrix}\sum_{c=1}^{C}\mathbf{w}_{c+1}\cdot\exp\left(-\left(\sum_{i=1}^{m\cdot(np-1)}\left(s_{n\cdot ns+i}(t)-c_{c,i}\right)^2+\left(x_1-c_{c,m\cdot(np-1)+1}\right)^2\right)\Big/\sigma_c\right)+\mathbf{w}_{C+2}\\ \vdots\\ \sum_{c=1}^{C}\mathbf{w}_{c+1}\cdot\exp\left(-\left(\sum_{i=1}^{np-1}\left(s_{n\cdot ns+i}(t)-c_{c,i}\right)^2+\left(x_{ns}-c_{c,m\cdot(np-1)+1}\right)^2\right)\Big/\sigma_c\right)+\mathbf{w}_{C+2}\\ 0\\ s_{n\cdot ns+m+1}(t)\\ \vdots\\ s_{n\cdot ns+m\cdot(np-2)}(t)\end{bmatrix} \quad (9)$$

$$\mathbf{f}_2(\mathbf{s}(t))=\mathbf{F}_2=\left[\mathbf{w}_1^T\ \ \ldots\ \ \mathbf{w}_1^T\ \ \mathbf{I}_m\ \ \mathbf{0}_{m\times m(np-2)}\right]^T \quad (10)$$

$$\mathbf{f}_3(\mathbf{s}(t))=\mathbf{F}_3=\begin{bmatrix}\mathbf{I}_{n\cdot ns}\\ \mathbf{0}_{m(np-1)\times n\cdot ns}\end{bmatrix} \quad (11)$$

$$\mathbf{h}_1\left(\mathbf{s}(t)\right)=\begin{bmatrix}\mathbf{I}_{n\cdot ns} & \mathbf{0}_{n\cdot ns\times m\cdot(np-1)}\\ \mathbf{0}_{m\cdot(np-1)\times n\cdot ns} & \mathbf{0}_{m\cdot(np-1)\times m\cdot(np-1)}\end{bmatrix}\mathbf{s}(t)=\mathbf{H}_1\cdot\mathbf{s}(t) \quad (12)$$

The system of Eqs. (8)-(12) now has the required form of (5), (6). The condition $\mathbf{f}_1(\mathbf{0})=\mathbf{0}$ is ensured by using the constrained least squares method to determine the output weights during the neural network training procedure. From (12), it is also valid that $\mathbf{h}(\mathbf{0})=\mathbf{0}$. The linearization of $\mathbf{f}_1\left(\mathbf{s}(t)\right)$ will give the square matrix, $\mathbf{F}_1=\left.\frac{\partial\mathbf{f}_1(\mathbf{s})}{\partial\mathbf{s}}\right|_{\mathbf{s}=\mathbf{0}}$.

IV. Robust H_∞ Control Law

The state space representation of the previous section allows the straightforward application of robust H_∞ control, where the control law is expressed as follows [15]:

$$\mathbf{u}(t)=-\left[\mathbf{I}_m \quad \mathbf{0}_{m\times n\cdot ns}\right]\cdot\mathbf{R}^{-1}\cdot\left[\mathbf{F}_2 \quad \mathbf{F}_3\right]^T\cdot\mathbf{P}\cdot\mathbf{f}_1\left(\mathbf{s}(t)\right) \quad (13)$$

where P is a positive definite symmetric $\left(n\cdot ns+m\cdot(np\text{-}1)\right)\times\left(n\cdot ns+m\cdot(np\text{-}1)\right)$ square matrix that satisfies the following matrix inequality:

$$-\mathbf{P}+\mathbf{F}_1^T\mathbf{P}\mathbf{F}_1+\mathbf{H}_1^T\mathbf{H}_1-\mathbf{F}_1^T\mathbf{P}\left[\mathbf{F}_1 \quad \mathbf{F}_2\right]\mathbf{R}^{-1}\left[\mathbf{F}_1 \quad \mathbf{F}_2\right]^T\mathbf{P}\mathbf{F}_1<\mathbf{0} \quad (14)$$

$$\mathbf{R}=\begin{bmatrix}\mathbf{F}_2^T\mathbf{P}\mathbf{F}_2+\mathbf{I}_m & \mathbf{F}_2^T\mathbf{P}\mathbf{F}_3\\ \mathbf{F}_3^T\mathbf{P}\mathbf{F}_2 & \mathbf{F}_3^T\mathbf{P}\mathbf{F}_3-\gamma^2\mathbf{I}_{n\cdot ns}\end{bmatrix} \quad (15)$$

and $\mathbf{R}$ is a $(m+n\cdot ns)\times(m+n\cdot ns)$ matrix. The control law of Eq. (13) guarantees that the closed loop system of (9)-(12) and (13) has a finite L_2-gain $\le\gamma$ in a neighborhood of the origin of the form $\Omega_a=\left\{\mathbf{s}\,\middle|\,\mathbf{s}^T\mathbf{P}\mathbf{s}\le a\right\}\subseteq\Omega=\left\{\mathbf{s}\,\middle|\,\|\mathbf{s}\|\le r\right\}$ for every $\mathbf{d}(t)\in\mathbf{D}=\left\{\mathbf{d}\in\mathbb{R}^{n\cdot ns},\ \|\mathbf{d}\|^2\le\gamma_1^2\|\boldsymbol{\psi}\|^2\right\},\gamma_1<1/\gamma$. From that follows that the closed loop system is robustly stable in Ω_a.

Remark 1. Parameter γ_1 can be determined from the input-output data that are used for the validation of the neural network since $\mathbf{d}(t)$ represents only the model mismatch.

V. Application in Thermal Systems

Fig. 2 presents a thermal system which will be used to test the proposed control methodology. It is a one-dimensional long duct with circular cross section, being in an environment of constant ambient temperature. The temperature at the entrance of the duct is considered constant, as well as the heat transfer coefficient, the fluid properties and the geometry diameter of the duct. In addition, any diffusive phenomena are neglected. Under these assumptions, the temperature distribution depends only on the fluid velocity. Application of the energy conservation law in this system results in a first order hyperbolic partial differential equation. After appropriate nondimensionalization [6], a governing equation of the following form is obtained:

$$\frac{\partial T(t,x)}{\partial t}+v(t)\frac{\partial T(t,x)}{\partial x}+T(t,x)=0 \quad (16)$$

$0\le x\le L$ and $t\ge 0$.

where L is the length of the duct, x denotes the distance from the duct entrance, t denotes time, $T(t,x)$ is the temperature and $v(t)$ is constant value of the fluid velocity throughout the duct at time t. The boundary condition is $T(t,0)=T_{in}=1$. In order to obtain the initial temperature steady state distribution Eq. (16) is solved using the finite differences method with suitably chosen spatial and temporal intervals, Δx and Δt respectively, so that the Courant-Friedrichs-Lévy stability condition $CFL=v(t)\Delta t/\Delta x\le 1$ is satisfied. So, for $v=1$, $\Delta t=0.001$ and $\Delta x=0.01$, the numerical solution of Eq. (16) until t=3 gives the distribution depicted in Fig. 3.

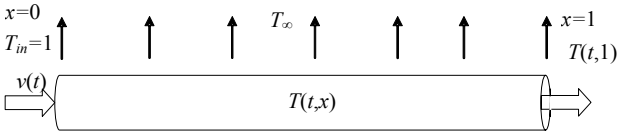

Fig. 2 The schematic representation of a long duct with constant ambient temperature and variable fluid velocity

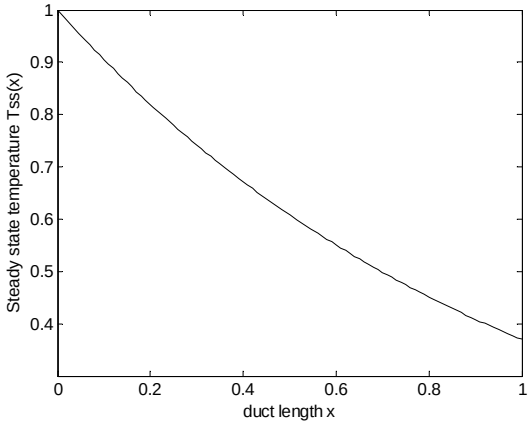

Fig. 3 Temperature steady state distribution for v=1

VI. Simulation Results

A. Neural Network Training

Eq. (16) is also used to obtain the training and the validation sets for the development of the neural network. A set of 3000 random velocity values in the interval $0.9\le v\le 1.1$ were produced and 3000×10 temperature values were collected at equidistant locations and with sampling time $t_s=0.2$. The last 100 input-output data were used for validation purposes, while the rest of them were used for training the network. Table I, presents the performance of the

produced neural network models with respect to the number of hidden nodes. Fig. 4 compares the true values with the corresponding predictions produced by the neural network consisting of 58 nodes. A step change of the manipulated variable was used to further test the accuracy of the model. The fluid velocity was increased from 1 to 1.05 and the RBF model predictions are compared with the true temperature in Fig. 5. Both final temperature distributions (after 2 time units) and the responses of the outlet temperature are depicted. The RBF model was again proved very accurate and will be used in the next paragraph for the calculation of the control law.

B. Performance of the Proposed Control Methodology

The proposed controller is now tested in disturbance rejection test cases. Firstly, temperature is assumed to be measured in real time only at the outlet of the duct, in order to check the produced results against those in [6], where the same case has been examined. So, in the nonlinear configuration of Section IV, the values of ns=1 and x=1 were used. The fluid velocity was increased from 1 to 1.01. The desired steady state was that corresponding to v=1. Parameter γ_1 was estimated from the data used to train the neural network. Table II gives the maximum obtained value of γ_1 and also summarizes the rest of control law parameters. The matrix calculated from Eqs. (14)-(15) is:

$$\mathbf{P} = \begin{bmatrix} 1.0000 & 0.0000 & 0.0000 & 0.0000 & -0.0000 \\ 0.0000 & 0.5709 & 0.3826 & 0.0951 & -0.0254 \\ 0.0000 & 0.3826 & 0.3066 & 0.0914 & -0.0223 \\ 0.0000 & 0.0951 & 0.0914 & 0.0360 & -0.0079 \\ -0.0000 & -0.0254 & -0.0223 & -0.0079 & 0.0018 \end{bmatrix} \quad (17)$$

Fig. 6 presents the temperature response and Fig. 7 depicts the control moves that were chosen by the controller to lead the system to the desired steady state. The results are significantly improved in comparison to those presented in [6] (for K_i=-1, K_p=1). Not only the time needed by the system to return to its initial steady state is substantially reduced (around three time units are required), but also the peak temperature value is much lower. An additional significant advantage of the proposed methodology is that the stability of the controller is ensured, by the described offline stabilization procedure. The same example was repeated by considering temperature at five locations $\mathbf{x} = [0.2 \quad 0.4 \quad 0.6 \quad 0.8 \quad 1]$ as controlled variable. In this case ns= 5 and the number of states is equal to 9 (**P** is a 9×9 matrix). Fig. 8 depicts the performance of the system as far as the dynamic behavior of the outlet temperature is concerned and Fig. 9 shows the control moves. It can be noticed that the response is further improved in this case. Indeed the capability to estimate and control temperature in locations inside the duct accelerates the response at exit.

TABLE I
SUM OF SQUARE ERRORS BETWEEN PREDICTIONS AND REAL VALUES

Number of fuzzy sets	Number of hidden nodes C	SSE calculated on 100 examples
3	5	$9.681\ 10^{-4}$
4	16	$1.898\ 10^{-4}$
5	30	$3.915\ 10^{-5}$
6	58	$1.213\ 10^{-5}$
7	93	$1.001\ 10^{-5}$

TABLE II
CONTROL LAW PARAMETERS VALUES

Control Law Parameters	Values
m	1
n	1
np	5
v_{ss}	1
max γ_1	0.22
γ^2	2
r	0.0224
a	0.0006

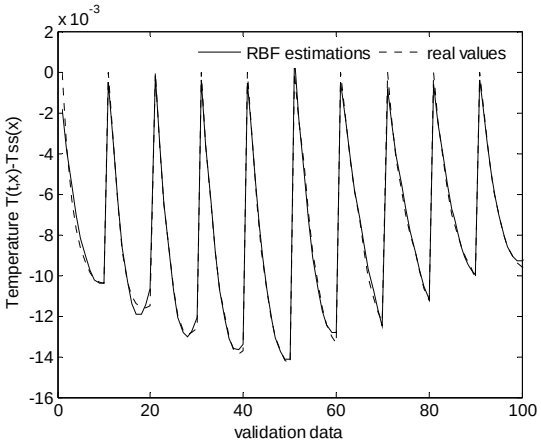

Fig. 4 Actual values and predictions of the neural network model

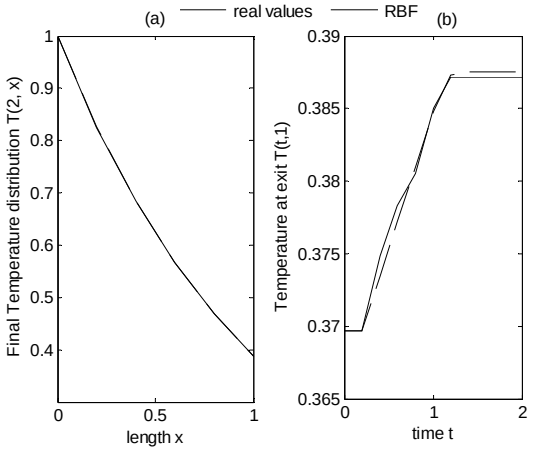

Fig. 5 (a) The temperature distributions after 2 time units, (b) the outlet temperature responses to a velocity step change

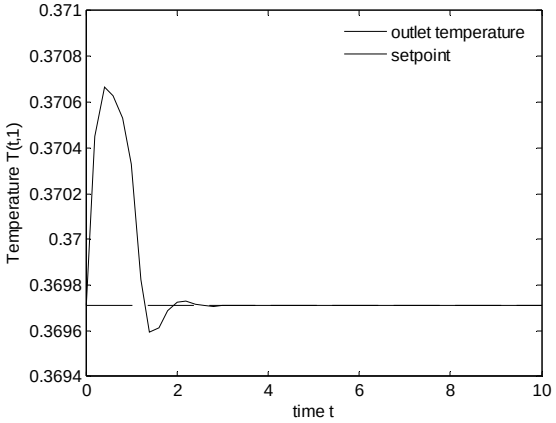

Fig. 6 Outlet temperature response in the operating area around v=1

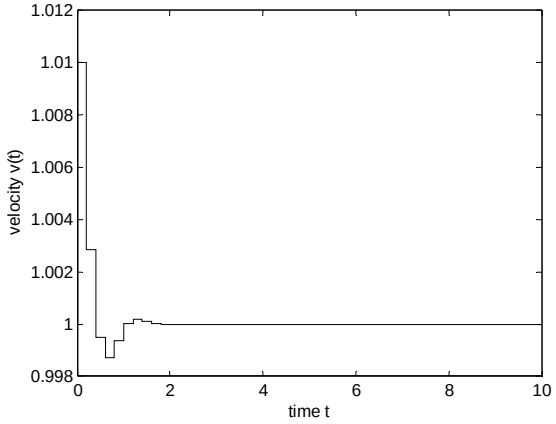

Fig. 7 Control moves in the operating area around v=1

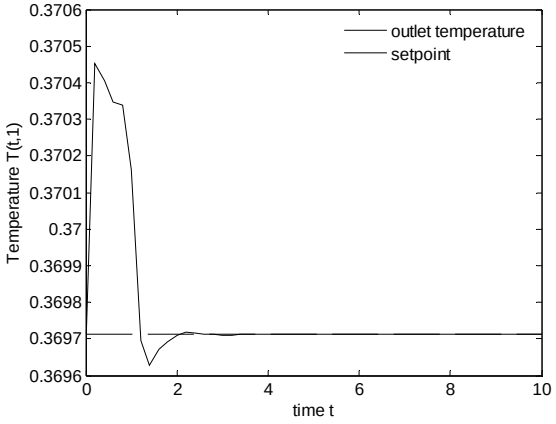

Fig. 8 Outlet temperature response, considering temperature at x = [0.2 0.4 0.6 0.8 1] as controlled variables

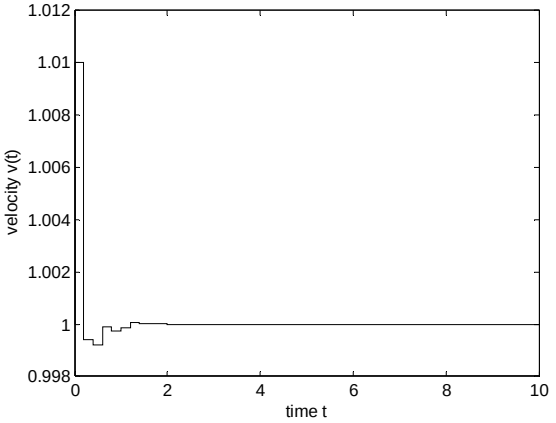

Fig. 9 Control moves, considering temperature at x = [0.2 0.4 0.6 0.8 1] as controlled variables

VII. Conclusion

An alternative approach for modeling and control hyperbolic DPSs has been presented in this work. An affine RBF neural network model, based on input –output data, is proposed to identify the distributed system, avoiding the solution of the corresponding PDEs that are not always known. A state space representation of the RBF network is then obtained that allows the implementation of an H_∞ control methodology. The proposed control takes into account the model mismatch and so can guarantee robust stability in a region of the original steady state. The proposed methodology is applied in a long duct, key component of many industrial thermal systems, and is proved to overcome delay effects and to improve the response of the outlet temperature by considering the temperature distribution along the duct. The control methodology could be easily extended to other DPSs of hyperbolic form, such as convection-reaction processes, by developing each time an adequate empirical model.

References

[1] E.M. Hanczyc and A. Palazoglu, "Nonlinear Control of a Distributed Parameter Process: The Case of Multiple Characteristics," *Ind. Eng. Chem. Res.*, vol. 34, pp. 4406-4412, 1995.

[2] E.M. Hanczyc and A. Palazoglu, "Sliding Mode Control of Nonlinear Distributed Parameter Chemical Processes," *Ind. Eng. Chem. Res.,* vol. 34, pp. 557-566, 1995.

[3] P. Christofides and P. Daoutides, "Feedback control of hyperbolic PDE systems," *AIChe J.*, vol. 42, no.11, pp. 3063-3086, 1996.

[4] P. Christofides and P. Daoutides, "Robust control of hyperbolic PDE systems," *Chemical Engineering Science*, vol. 53, no. 1, pp. 3063-3086, 1998.

[5] P. Christofides, *Nonlinear and Robust Control of PDE Systems: Methods and Applications to transport reaction processes*. Boston: Birkhäuser, 2001.

[6] S. Alotaibi, M. Sen, B. Goodwine and K.T. Yang, "Flow based control of temperature in long ducts," *International Journal of Heat and Mass Transfer,* vol. 47, pp. 4995-5009, 2004.

[7] H. Shang, J.F. Forbes and M. Guay, "Feedback control of hyperbolic distributed parameter systems," *Chemical Engineering Science,* vol. 60, pp. 969 – 980, 2005.

[8] I. Karafyllis and P. Daoutidis, "Control of hot spots in plug flow reactors," *Computers & Chemical Engineering,* vol. 26, no. 7–8, pp. 1087–1094, 2002.

[9] A.A. Patwardhan, G.T. Wright and T.F. Edgar, "Nonlinear model predictive control of distributed parameter systems," *Chemical Engineering Science,* vol. 47, no. 4, pp. 721-735, 1992.
[10] S. Dubljevic, P. Mhaskar, N.H. El-Farra and P. Christofides, "Predictive control of transport-reaction processes," C*omputers and Chemical Engineering*, vol. 29, pp. 2335-2345, 2005.
[11] H. Shang, J.F. Forbes and M. Guay, "Model predictive control of Quasi-linear Hyperbolic Distributed Parameter Systems," *Ind. Eng. Chem. Res*., vol. 43, pp. 2140-2149, 2004.
[12] R. Gonzáles-García, R. Rico-Martínez, I. Kevrekidis, "Identification of distributed parameter systems: A neural net based approach," *Comp. Chem. Eng.*, vol. 22, pp. 965-968, 1998.
[13] R. Padh, S. Balakrishnan and T. Randolph, "Adaptive-critic based optimal neuro control synthesis for distributed parameter systems," *Automatica*, vol. 37, pp. 1223-1234, 2001.
[14] E. Aggelogiannaki, H. Sarimveis, "Model predictive control for distributed parameter systems using RBF neural networks," in: *Proc. 2nd International Conference on Informatics in Control, Automation and Robotics*, Barcelona, 2005.
[15] L. Magni, G. De Nicolao, R. Scattolini, F. Allgöwer, "Robust model predictive control for nonlinear discrete-time systems," *International Journal of Robust and Nonlinear Control,* vol. 13, pp. 229-246, 2003.
[16] W. Lin and C.I. Byrnes, "Discrete-time Nonlinear H_∞ Control with Measurement Feedback", *Automatica*, vol. 31, pp. 419-434, 1995.
[17] W. Lin and L. Xie, "A link between H_∞ control of a dicrete time nonlinear system and its linearization", *International Journal of control*, vol. 69, no. 2, pp. 301-314, 1998.
[18] H. Sarimveis, A. Alexandridis, G. Tsekouras and G. Bafas, "A fast and efficient algorithm for training radial basis function neural networks based on a fuzzy partition of the input space," *Industrial Engineering Chemistry Research*, vol. 41, pp. 751-759, 2002.