15th Triennial World Congress, Barcelona, Spain

DISTRIBUTED WIENER LOGIC PROCESSORS

Enso Ikonen [*,1] **Kaddour Najim** [**] **Urpo Kortela** [*]

[*] *University of Oulu, Finland*
[**] *E.N.S.I.A.C.E.T., Toulouse, France*

Abstract: A distributed Wiener logic processor model structure is considered. Each fuzzy Wiener model consists of a succession of a linear dynamic part and a static steady-state (non-linear) logical part. The model structure and the necessary gradients required by gradient-based parameter estimation methods are given. Parameter projection and a modified threshold method are discussed. A simulation example illustrates the approach in the identification of a nonlinear, non-minimum phase CSTR process where a van der Vusse reaction takes place.

Keywords: fuzzy systems, man/machine interaction, nonlinear models, parameter estimation, process identification

1. INTRODUCTION

The DLP (Distributed Logic Processor) approach in process identification has been considered by several authors, see (Pedrycz *et al.*, 1995), (Ikonen *et al.*, 2000), (Najim and Ikonen, 1999), and references therein. The Takagi-Sugeno fuzzy model can be seen as a special case of the DLP structure (Ikonen, 1996). The main difference to the mainstream of fuzzy rule-based approaches is that the parameter estimation concerns the rule-base of the system, not the partitioning of the input–output spaces. Instead, the emphasis is in finding the logical relationships between the modelling landmarks. Thus, the transparency of the mapping, relying on the familiarity of the concepts used in partitioning, is not lost during the parameter estimation. Note, however, that there are no particular technical constraints that would restrict the use of data-driven methods with the DLP approach also for the selection of an optimal partitioning (see, *e.g.*, (Pedrycz and Valente de Oliveira, 1996)).

The identification of dynamic systems using DLP's is straightforward using the time-series approach (Pedrycz *et al.*, 1995). However, the dimension of the nonlinear mapping can become excessive, prohibiting practical applications. In this paper, we consider a succession of a linear dynamic part and a static steady-state (non-linear) logical part. In the DWLP (Distributed Wiener Logic Processors) approach proposed in this paper, the inputs of each logic processor are filtered using a system composed of $I \times I$ linear IIR filters where I is the number of inputs to the process model. The output of the structure is obtained as a combination of the non-linear dynamic logical mappings, by associating each mapping with an output singleton and computing a weighted average. Since the filters precede the static logical parts, the system is composed of a number of Wiener systems in parallel.

In what follows, the DWLP structure will be considered in detail. Also the Hammerstein case is briefly outlined. The application to process identification is discussed, and a numerical example illustrates the performance of the approach in process modelling, using noisy data from a simulated van der Vusse CSTR (Chen *et al.*, 1995). It is concluded that the approach is appealing from the point of view of both accuracy of the predictions,

[1] Supported by the Academy of Finland, Project no 48545

and simplicity of the analysis of the behavior of the nonlinear dynamic model obtained.

2. DISTRIBUTED WIENER LOGIC PROCESSORS

Consider the following Wiener logic processor (WLP) structure. For each WLP, the system inputs $x_i \in \Re$, $i = 1, 2, ..., I$, are associated with a system of linear filters

$$z_j(k) = \sum_{i=1}^{I} \frac{B^*_{j,i}\left(q^{-1}\right)}{A_{j,i}\left(q^{-1}\right)} x_i\left(k - d_{j,i}\right) \qquad (1)$$

where $z_j \in \Re$, $j = 1, 2, ..., I$, and $B^*_{j,i}\left(q^{-1}\right)$ and $A_{j,i}\left(q^{-1}\right)$ are polynomials in the backward shift operator q^{-1} such that $\lim_{z \to 1} B^*_{j,i}(z)/A_{j,i}(z) = \delta_{j,i}$, $\delta_{j,i} = 1$ if $i = j$, 0 otherwise, and $d_{j,i}$ are the time delays. Let $B^*_{j,i}\left(q^{-1}\right)$ and $A_{j,i}\left(q^{-1}\right)$ be given by:

$$B^*_{j,i}\left(q^{-1}\right) = b_{j,i,0} + b_{j,i,1}q^{-1} + ... \qquad (2)$$
$$+ b_{j,i,n_{B_{j,i}}-1} q^{-\left(n_{B_{j,i}}-1\right)} + b^*_{j,i,n_{B_{j,i}}} q^{-n_{B_{j,i}}}$$

$$A_{j,i}\left(q^{-1}\right) = \qquad (3)$$
$$1 + a_{j,i,1}q^{-1} + ... + a_{j,i,n_{A_i}} q^{-n_{A_{j,i}}}$$

where

$$b^*_{j,i,n_{B_i}} = \delta_{j,i}\left(1 + \sum_{n=1}^{n_{A_{j,i}}} a_{j,i,n}\right) \qquad (4)$$
$$- \sum_{n=1}^{n_{B_{j,i}}-1} b_{j,i,n}$$

The membership $\mu_b \in [0,1]$ in the b'th set partitioning the space of one of the z_i's, $b = 1, 2, ..., B$, is obtained from

$$\mu_b(k) = \mathrm{F}_b\left(z_i(k), \cdot_b\right) \qquad (5)$$

where partitioning is done using a single-variable membership function of form F_b, parameterized with some associated parameters $\cdot_b$. The memberships $\mathbf{M}(k) = [\mu_1(k), \mu_2(k), ..., \mu_B(k)]^T$ are fed to a logic processor (LP).

A logic processor consists of a layer of R AND-nodes

$$\xi_r(k) = \mathrm{T}\left(\mathbf{M}(k)\, \mathrm{S}\mathbf{\Omega}_r\right) \qquad (6)$$

with parameter vectors $\mathbf{\Omega}_r = [\omega_{r,1}, \omega_{r,2}, ..., \omega_{r,B}]^T$, $\omega_{r,b} \in [0,1]$, $r = 1, 2, ..., R$. T and S denote the t-norm and the s-norm, respectively. The layer of AND-nodes is followed by a single OR-node

$$\tau(k) = \mathrm{S}\left(\mathbf{\Xi}(k)\, \mathrm{T}\mathbf{\Lambda}\right) \qquad (7)$$

with inputs $\Xi(k) = [\xi_1(k), \xi_2(k), ..., \xi_R(k)]^T$, $\xi_r(k) \in [0,1]$ and parameters $\mathbf{\Lambda} = [\lambda_1, \lambda_2, ..., \lambda_R]^T$ $\lambda_r \in [0,1]$, $\tau_r \in [0,1]$. In order to enhance the mapping capabilities, an additional monotone non-linear element, such as a parameterized sigmoid function, can follow the OR-node

$$\widehat{\nu}(k) = \frac{1}{1 + e^{-\sigma_2(\tau(k) - \sigma_1)}} \qquad (8)$$

where $\widehat{\nu}(k) \in [0,1]$ and parameters $\sigma_1, \sigma_2 \in \Re$.

Notice, that the WLP can be separated into two blocks $\widehat{\nu}(k) = \mathrm{f}_2(\mathrm{f}_1(\mathbf{x}(k)))$ where f_1 is a linear dynamic function and f_2 is a static mapping. Thus, it is a Wiener structure. An additional restriction was posed by requiring that the steady-state gain of the dynamic part is one, which results in that the static part is a steady-state model of the process.

A distributed WLP (DWLP) network consists of Q WLP's. The output is obtained by computing a weighted average

$$\widehat{y}(k) = \frac{\sum_{q=1}^{Q} \overline{y}_q \widehat{\nu}_q(k)}{\sum_{q=1}^{Q} \widehat{\nu}_q(k)} \qquad (9)$$

weighted with $\overline{y}_q \in \Re$, $q = 1, 2, ..., Q$.

In what follows we make the following non-restrictive assumptions: the partitioning is add-one and made using triangular F_b's; the t- and s-norms are selected as the product ($atb = ab$) and the probabilistic sum ($asb = a + b - ab$). It is now straightforward to find analytic expressions for the gradients with respect to system parameters $\boldsymbol{\theta}$: $a_{j,i,1}$,..., $a_{j,i,n_{A_i}}$; $b_{j,i,1}$,..., $b_{j,i,n_{B_i}-1}$ $\forall j, \forall i$, $\omega_{r,b} \forall r, \forall b$, $\lambda_r \forall r$, σ_1, σ_2 by computing the derivatives of the various model components, and using the chain rule. For simplicity of the notation, the indexing for all q's ($q = 1, 2, ..., Q$) is omitted, as in (1)–(8). For brevity, we omit the expressions for the gradients of the LP's here (see, e.g., (Ikonen *et al.*, 2000)). Given the expressions for the gradients through the static part, the $\frac{\partial \widehat{\nu}}{\partial z_i}(k)$'s, for each q, the gradients with respect to the parameters of the dynamic filters are obtained from

$$\frac{\partial \widehat{\nu}}{\partial a_{j,i,m}}(k) = \frac{\partial \widehat{\nu}}{\partial z_j}(k) \frac{\partial z_j}{\partial a_{j,i,m}}(k) \qquad (10)$$

$$\frac{\partial \widehat{\nu}}{\partial b_{j,i,n}}(k) = \frac{\partial \widehat{\nu}}{\partial z_j}(k) \frac{\partial z_j}{\partial b_{j,i,n}}(k) \qquad (11)$$

where the gradients of the dynamic part are given by

$$A_{j,i}\left(q^{-1}\right)\frac{\partial z_j}{\partial a_{j,i,m}}(k) \quad (12)$$
$$=\delta_{j,i}x_i\left(k-d_{j,i}-n_{B_{j,i}}\right)-v_{j,i}(k-m)$$

$$A_{j,i}\left(q^{-1}\right)\frac{\partial z_j}{\partial b_{j,i,n}}(k) \quad (13)$$
$$=x_i(k-d_{j,i}-n)-x_i\left(k-d_{j,i}-n_{B_{j,i}}\right)$$

where

$$v_{j,i}(k)=\frac{B^*_{j,i}\left(q^{-1}\right)}{A_{j,i}\left(q^{-1}\right)}x_i(k-d_{j,i}) \quad (14)$$

Notice, that the gradients can be approximated using finite differences, too, but this may be far too time-consuming for practical applications.

It is also straightforward to construct distributed Hammerstein logic processors (HLP), using the same ideas as above.

3. DISTRIBUTED HAMMERSTEIN LOGIC PROCESSORS

In the Hammerstein structure, the linear dynamic part follows the static (nonlinear) part. Let the membership $\mu_b \in [0,1]$ in the b'th set be obtained from the partitioning the input space of one of the system inputs x_i, i.e. for $b=1,2,...,B$, it is obtained from

$$\mu_b(k)=\mathrm{F}_b\left(x_i(k),\cdot_b\right) \quad (15)$$

where the input space is partitioned using membership functions of form F_b, parameterized with some associated parameters $\cdot_b$. The logic processor is described by equations (6)–(8).

The output $\widehat{v}(k)$ of the Hammerstein logic processor (Ikonen, 2001) is then obtained by filtering the LP output $\widehat{\nu}(k)\in[0,1]$ with a linear filter

$$A\left(q^{-1}\right)\widehat{v}(k)=B^*\left(q^{-1}\right)\widehat{\nu}(k-d) \quad (16)$$

$\widehat{v}(k)\in\Re$, where the last coefficient of the polynomial B^* is given by

$$b^*=1+\sum_{n=1}^{n_A}a_n-\sum_{n=1}^{n_B-1}b_n \quad (17)$$

The output of a distributed Hammerstein LP network is obtained from (9) by replacing $\widehat{\nu}_q(k)$'s with $\widehat{v}_q(k)$'s.

The Hammerstein LP can be separated into two blocks $v(k)=\mathrm{f}_2\left(\mathrm{f}_1(\mathbf{x}(k))\right)$ where f_1 is a static nonlinear mapping and f_2 is a linear dynamic function. It could be of interest to restrict the output $\widehat{v}$ of the HLP into the unit interval, as this would enable to interpret the HLP as a dynamic logical mapping; for process identification purposes only, this is not necessary, however. Notice, that when approximating nonlinear dynamic MISO systems with the Hammerstein LP's, they do not allow the identification of separate dynamics for each input, nor transient cross-dynamics.

The expressions for the gradients of the Hammerstein LP output with respect to its parameters are obtained from

$$A\left(q^{-1}\right)\frac{\partial\widehat{v}}{\partial a_m}(k) \quad (18)$$
$$=\widehat{\nu}(k-d-n_B)-\widehat{v}(k-m)$$

$$A\left(q^{-1}\right)\frac{\partial\widehat{v}}{\partial b_n}(k) \quad (19)$$
$$=\widehat{\nu}(k-d-n)-\widehat{\nu}(k-d-n_B)$$

and

$$A\left(q^{-1}\right)\frac{\partial\widehat{v}}{\partial w_p}(k)=B^*\left(q^{-1}\right)\frac{\partial\widehat{\nu}}{\partial w_p}(k-d) \quad (20)$$

where $\frac{\partial\widehat{\nu}}{\partial w_p}$ denotes the gradients of the static LP with respect to a generic parameter $w_p\in\{\omega_{r,b},\lambda_r,\sigma_1,\sigma_2\}\ \forall r,\forall b$, and for all $q=1,2,...,Q$ (Notice again that the indexing for q's was omitted, for simplicity of notation).

4. PROCESS IDENTIFICATION USING DWLP

Identification (Ikonen and Najim, 2002) consists of four stages. Assume that a proper data set is available and let us concentrate on some of the issues in structure selection, parameter estimation and validation concerning DWLP models. A similar discussion holds also for the distributed HLP models.

Assume that the sets partitioning the model input-output spaces are given, i.e. $\mu_b(k)$'s can be obtained given $x_i(k)$'s, and $\overline{y}_q$'s are known (This assumption underlines the fact that the main emphasis is to reveal the logical relationships between user-defined concepts; remember that non-parametric regression methods, for example, provide efficient tools for mere function approximation or data-driven optimization problems.) The orders of the feedforward and feedback dynamics, $n_{A_{j,i}}$ and $n_{B_{j,i}}$ need to be specified for all $i,j=1,2,...,I$ and each WLP. It is common to select these based on *a priori* knowledge, although tools such as sub-space methods (Knudsen, 2001) can be useful, too. The number of AND-nodes, R, for each WLP remains to be set. The parameter R controls the size of the rule base within each WLP and its selection is dependent on the resolution provided by the partitioning.

Given the output data $y(k)$, $k = 1, 2, ..., K$, where K is the number of data patterns, the target values $\nu_q(k)$ for the q^{th} WLP can be obtained by assuming a triangular membership function F with its centre placed at $\overline{y}_q$:

$$\nu_q(k) = \max\left[\min\left(\frac{y(k)-\overline{y}_{q-1}}{\overline{y}_q-\overline{y}_{q-1}}, \frac{\overline{y}_{q+1}-y(k)}{\overline{y}_{q+1}-\overline{y}_q}\right), 0\right] \quad (21)$$

and taking $\overline{y}_0 = -\infty$, $\overline{y}_{Q+1} = \infty$. Collecting all the WLP parameters to be estimated in a parameter vector $\boldsymbol{\theta}_q$, we have the following Q separate optimization problems

$$\min_{\boldsymbol{\theta}_q} \sum_{k=1}^{K} \left[\left(\nu_q(k) - \widehat{\nu}_q(k, \boldsymbol{\theta}_q)\right)^2\right] \quad (22)$$

for all q, which can be solved using standard optimization methods.

Our simulations have indicated that the Levenberg-Marquardt method is efficient for solving the problem. However, the problem is actually a constrained optimization problem since we need to have $\omega_{r,b}$, $\lambda_r \in [0,1]$ and $B^*_{j,i}(z)/A_{j,i}(z)$ stable (to ensure the boundedness of the gradients). In order to keep the number of function and gradient evaluations small (due to the computational costs involved), we have found that it is sufficient to use a simple projection method to ensure these constraints. Namely, after each parameter update, the new parameters have been projected into the allowed region

$$\omega_{r,b} = \max\left(\min\left(\omega^{\text{LM}}_{r,b}, 1\right), 0\right) \quad (23)$$

$$\lambda_r = \max\left(\min\left(\lambda^{\text{LM}}_{r}, 1\right), 0\right) \quad (24)$$

where $\omega^{\text{LM}}_{r,b}$ and λ^{LM}_r are the parameters obtained from the unconstrained Levenberg-Marquardt method, and stability is ensured by projecting the feedback polynomial with

$$A_{j,i}\left(q^{-1}\right) = 1 + \gamma a^{\text{LM}}_{j,i,1} q^{-1} + ... + \gamma^{n_{A_i}} a^{\text{LM}}_{j,i,n_{A_i}} q^{-n_{A_{j,i}}} \quad (25)$$

where $0 \ll \gamma < 1$, repeating with the projected parameters until all poles lie in the desired stable region. Note that the constraints on the steady state gain of $B^*_{j,i}\left(q^{-1}\right)/A_{j,i}\left(q^{-1}\right)$ were taken into account in the proper computation of the gradients of the model structure.

One of the main advantages of the DWLP's is the transparency of the resulting model. The dynamics are captured in the parameters of the linear transfer functions, from where the pole-zero maps etc. can be easily obtained. Note that also non-linear dynamics can be expressed with the DWLP approach, provided that the dynamics do not exhibit output-multiplicity. Due to the unit steady-state gain constraints in the dynamic parts, there is no redundancy in the gains of the static and dynamic parts.

The parameters of the logic processors are easy to interpret and their examination as such reveals much of the logical relationships. To simplify further, the logic processors can be converted into a binary-weighted rule-base form using the threshold-method (Pedrycz, 1993). In the threshold-method, all $\omega_{r,b} \leq \alpha$ are set to 0's and others to 1's; similarly, all $\lambda_r \geq 1-\alpha$ are set to 1's and others to 0's. Using the unit-element properties of t- and s-norms, the network contents can then be expressed as rules for each output class consisting of simple AND and OR connectives. However, the existence of the sigmoid function complicates slightly the approach. The essence of the approach is recovered by remapping the upper and lower bounds specified by the threshold parameter α through the inverse of the nonlinear function. For the sigmoid function a modified threshold method can be used: all $\omega_{r,b} \leq \alpha_*$ are set to 0's and others to 1's; similarly, all $\lambda_r \geq 1-\alpha^*$ are set to 1's and others to 0's, where

$$\alpha_* = \sigma_1 - \frac{1}{\sigma_2}\ln\left(\frac{1-\alpha}{\alpha}\right) \quad (26)$$

$$\alpha^* = 1 - \sigma_1 + \frac{1}{\sigma_2}\ln\left(\frac{\alpha}{1-\alpha}\right) \quad (27)$$

5. A NUMERICAL EXAMPLE

In the following, we illustrate the approach with a simple numerical example using data from a simulated continuous stirred-tank reactor (CSTR) where a series and parallel van der Vusse reaction

$$A \overset{k_1}{\rightarrow} B \overset{k_2}{\rightarrow} C \quad (28)$$

$$2A \overset{k_3}{\rightarrow} D \quad (29)$$

takes place, for exact details see (Chen *et al.*, 1995). The plant input consisted of step changes in the normalized feed flow, $\frac{\dot{V}}{V_R}$. The concentration, c_B, of the component B was measured and it represented the output. For simplicity, only the SISO case was considered. At the considered region, $\frac{\dot{V}}{V_R} \in [3, 15]$, the plant exhibits non-minimum phase dynamic behavior dependent on the point of operation (for a nice presentation, see (Gatzke and Doyle III, 1999)). Data for identification was generated by simulating the plant with a ramp of step changes from $\frac{\dot{V}}{V_R} = 3\text{h}^{-1}$ to $\frac{\dot{V}}{V_R} = 15\text{h}^{-1}$ using $\Delta\frac{\dot{V}}{V_R} = 1\text{h}^{-1}$ and $\Delta T_{\text{step}} = 30\,\text{min}$, and backwards, and measuring the input and output

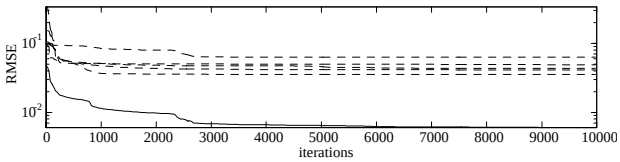

Fig. 1. Evolution of the RMSE.

signals corrupted with $N\left(0, 0.0015^2\right)$ with a sampling time of 1 min. This resulted in a data set of 780 input–output patterns.

The input–output partitioning was given by triangular fuzzy sets with centers

$$\overline{\frac{\dot{V}}{V_R}}_b = \{1, 3, 5, ..., 17\}\,\mathrm{h}^{-1} \tag{30}$$

$(b = 1, 2, ..., 9)$ for the input and

$$\overline{c_{Bq}} \in \{0.80, 0.95, 1.00, 1.05, 1.10\}\,\mathrm{mol/l} \tag{31}$$

$(q = 1, 2, ..., 5)$ for the outputs. The number of AND nodes R for each WLP was set to 7. The orders of the filters were chosen as $n_A = 2$, $n_B = 3$ for each WLP, with $d = 1$. This setting resulted in 77 parameters to be estimated from data, for each of the 5 WLP's.

The parameters were estimated using the Levenberg-Marquardt method, and projecting the parameters to the allowed region after each iteration (see previous section). The parameters of each of the WLP's were estimated 25 iterations at a time, and the performance of the overall DWLP was tested, until a maximum number of iterations (10000) was reached.

Figures 1–3 illustrate the simulation. The evolution of the RMSE for each WLP (dashed lines) and the DWLP (solid line) is shown in Fig. 1. The error decreases rapidly during the first iterations, but slows down as the number of iterations increases.

The prediction using the estimated WLP's (solid line) and the corresponding data (thin line) are shown in Fig. 2, together with negative absolute errors. The prediction by the DWLP network (solid line) and the corresponding output data (thin line) are shown at the top of Fig. 3; below are shown the absolute error, $|\widehat{c_B}(k) - c_B(k)|$, and process input $\frac{\dot{V}}{V_R}(k)$. From approximation accuracy point of view, a reasonable model was obtained: Clearly, the logical mappings between the input and output classes are properly identified, as well as rough approximations of the non-minimum phase dynamics.

However, the essence of the distributed fuzzy modelling approach is not in the accuracy but in the transparency of the estimated mapping. For example, from the estimated WLP's we can obtain

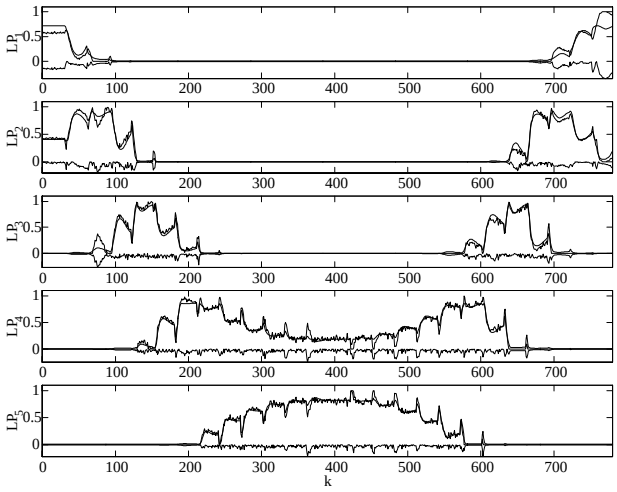

Fig. 2. Behaviour of the WLP models.

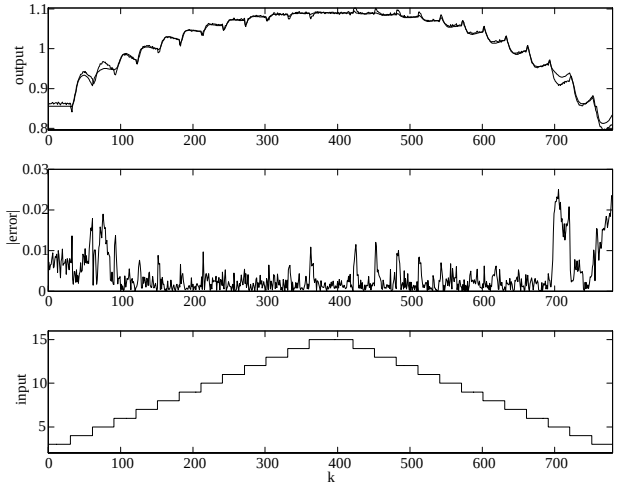

Fig. 3. Behaviour of the DWLP model.

the following type of rules for the static logical mapping ($\alpha = 0.1$):

$$\begin{aligned} &\text{if } \left(\frac{\dot{V}}{V_R} \text{ is } X_2\right) \\ &\text{or } \left(\frac{\dot{V}}{V_R} \text{ is } X_3\right) \\ &\text{or } \left(\frac{\dot{V}}{V_R} \text{ is } X_3 \text{ and } x \text{ is } X_4\right) \\ &\text{then } c_B = 0.95 \end{aligned} \tag{32}$$

where the linguistic labels X_b are associated with the sets defined using the $\overline{\frac{\dot{V}}{V_R}}_b$'s. Similar simplifications can be extracted for all other WLP's.

The nonlinearity in the identified DWLP dynamics is clear to see by looking at the pole-zero maps for the WLP linear dynamic parts, Fig. 4 shows a part of the map collecting the poles and zeros of all WLP's. Clearly, for WLP's $2-5$, $\overline{c_{Bq}} \in \{0.95, 1.00, 1.05, 1.10\}$, stable non-minimum phase filters were estimated, whereas the estimation for

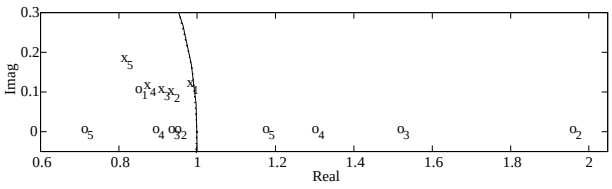

Fig. 4. Pole-zero map. Subscripts refer to the corresponding WLP index q.

the 1st WLP, $\overline{c_{Bq}} = 0.80$, resulted in a stable oscillating filter with all zeros inside the unit circle, but with nonzero imaginary parts. This *a priori* unexpected behavior can be addressed to the small amount of information rich data for the 1st WLP.

6. CONCLUSIONS

A distributed fuzzy Wiener model structure was considered. The model structure and the necessary gradients required by gradient-based parameter estimation methods were given. Parameter projection and a modified threshold method were discussed. A simulation example illustrated the approach in process identification.

Even if the computational costs are somewhat increased, the distributed Wiener logic processor approach is appealing from the point of view of accuracy of the predictions, transparency of the model (role of its components, interpretation of its parameters), as well as the simplicity of the analysis of the behavior of the nonlinear dynamic model obtained (predictable interpolation and extrapolation properties, applicability of the analysis tools for linear dynamic systems).

7. REFERENCES

Chen, H, A Kremling and F Allgöwer (1995). Nonlinear predictive control of a benchmark CSTR. In: *Proceedings of the European Control Conference*. Rome, Italy. pp. 3247–3252.

Gatzke, E and F Doyle III (1999). Multiple model approach for CSTR control. In: *Proceedings of the 14th IFAC World Congress*. Beijing, China.

Ikonen, E (1996). Algorithms for Process Modelling Using Fuzzy Neural Networks: A Distributed Logic Processor Approach. PhD thesis. University of Oulu.

Ikonen, E (2001). Distributed fuzzy Hammerstein modelling. In: *Proceedings of the TOOLMET '01*. University of Oulu.

Ikonen, E and K Najim (2002). *Advanced Process Identification and Control*. Marcel Dekker Inc.. New York.

Ikonen, E, K Najim and U Kortela (2000). Neuro-fuzzy modelling of power plant flue-gas emissions. *Engineering Applications of Artificial Intelligence* **13**, 705–717.

Knudsen, T (2001). Consistency analysis of subspace identification methods based on a linear regression approach. *Automatica* **37**(1), 81–89.

Najim, K and E Ikonen (1999). Distributed logic processors trained under constraints using stochastic approximation techniques. *IEEE Transations on Systems, Man, and Cybernetics - A* **29**, 421–426.

Pedrycz, W (1993). Fuzzy neural networks and neurocomputations. *Fuzzy Sets and Systems* **56**, 1–28.

Pedrycz, W and J Valente de Oliveira (1996). Optimization of fuzzy models. *IEEE Transactions on Systems, Man and Cybernetics – Part B* **26**(4), 627–636.

Pedrycz, W, P Lam and A Rocha (1995). Distributed fuzzy system modelling. *IEEE Transactions on Systems, Man and Cybernetics* **25**, 769–780.