Real Domain Adaptive WOS Filtering using Neural Network Approximations

Ioan Tabus, Moncef Gabbouj and Lin Yin
Signal Processing Laboratory
Tampere University of Technology
P.O.Box 553 SF-33 101 Tampere, Finland

Abstract

The problem of optimal Weighted Order Statistics (WOS) filters design is first related to the optimal design in a larger class, called Variable Rank Order Statistics (VROS) filters, the results obtained providing guidelines for the approach to be used in WOS filtering. Since adaptive methods applied directly to WOS filter models have to cope with a very ill conditioned problem, the adaptation will act on a model which belongs to Neural Networks (NN) class. This particular neural model can be trained using an algorithm very similar to the classical Backpropagation algorithm. In the final stage of training, the neural model can be made arbitrarily close to a WOS filter.

1. WOS Filters and Optimal Design Problem

1.1 WOS and VROS filters

The archetype of WOS filter is the order statistics (OS) filter, which processes at any time t the input values inside a window $X(t)$ of length $N = N_1 + N_2 + 1$, including the current input $x(t)$,

$$X(t) = [x(t-N_1)\dots x(t)\dots x(t+N_2)]$$
$$= [X_1 X_2 \dots X_N](t) \quad (1)$$

First, the values in the window are ordered decreasingly, resulting the ordered vector

$$\mathcal{O}X(t) = [X_{(1)}\dots X_{(N)}](t) = [X_{i_1}\dots X_{i_N}](t) \quad (2)$$

where $X_{(k)}$ denotes the k'th value in the ordered set. This operation can be thought of as being implemented by the ordering operator $\mathcal{O}$, using the permutation

$$P(t) = \begin{pmatrix} 1 & 2 & \dots & N \\ i_1 & i_2 & \dots & i_N \end{pmatrix} \quad (3)$$

of the elements in the original window for arranging them in decreasing order. The output of the filter will be $y(t) = X_{(r)}(t)$ with r, the so called rank of the order statistics, a fixed integer value less than or equal to N.

More flexibility can be added to OS filters if the rank r is allowed to be dependent on the actual permutation $P(t)$. This goal can be achieved by assigning to every permutation a rank, e.g. using a look-up table .

Definition 1 A Variable Rank Order Statistics (VROS) filter performs the ordering of the window $X(t)$ using the permutation $P(t)$, then takes the rank r assigned to $P(t)$ and finally gives the output $y(t) = X_{(r(P(t)))}$.

A more refined way to achieve the permutation dependence of the rank is provided by WOS filters.

Definition 2 A WOS filter has a real valued weight $W(X_i) = W_i$ assigned to every entry X_i in the window and a threshold weight, W_{N+1}. The filter will process the samples in the input window in two steps:

- the window is decreasingly ordered ;
- for every $r = 1,\dots,N$, the inequality $\sum_{i=1}^{r} W(X_{(i)}(t)) \geq W_{N+1}$ is tested and the least value of r for which the inequality holds will select the value $y(t) = X_{(r)}(t)$ as the filter output.

It is obvious that r is not anymore fixed as in OS filters (rank order filters); the weights have the effect of changing the rank value when the order in the window changes and it follows that WOS filter class is included in VROS filter class.

WOS filter class is a subclass of stack filters because any WOS filter possesses the stacking property. However, since one can give examples of VROS filters which do not possess the stacking property[2], it results that there are VROS filters which are not WOS filters. This proves that WOS filter class is *strictly* included in VROS filter class.

1.2 The Approach to Optimality

This paper will deal with the adaptive approach to optimality. The proposed adaptive methods minimize two types of criteria : Mean Absolute Error and Mean Squared Error, although extension to other criteria is straightforward. If the arithmetic mean is taken as an estimate of the statistical mean (ergodicity hypothesis) the optimality criterion can also be quoted as Minimum Sum of Absolute Errors and Minimum Sum of Squares Error respectively.

Problem of Optimal Filter Design Given

- the input set $\{X(t)\}_{t=1}^{T}$,
- the desired output set $\{d(t)\}_{t=1}^{T}$,
- the filter class $\mathcal{C}$,

find the filter parameters θ^* which minimize the criterion

$$J(\theta) = \frac{1}{T}\sum_{t=1}^{T} |d(t) - y(X(t),\theta)|^{\rho} \qquad (4)$$

with $\rho = 1$ for MAE criterion and $\rho = 2$ for MSE criterion. The class $\mathcal{C}$ will be in turn VROS class $\mathcal{V}_N$, with parameter vector $\theta \in \mathcal{V}_N$ and WOS class $\mathcal{W}_N$, with parameter vector $\theta = W \in \mathcal{W}_N$.

1.3 Optimal OS and VROS Design

Since there are only N OS filters, the optimal one can simply be obtained by comparing the N values of the criterion (4).

The procedure for finding the global optimum in VROS filter class - under any performance criterion which depends on the estimation error - can easily be derived, mainly because the selection of a certain rank for a permutation does not restrict the rank selection for other permutations.

Step 1. For all windows $X(t)$ find the associated permutation $P(t)$;
Step 2. Group the identical permutations to obtain the set of distinct permutations $\mathcal{P} = \{P_{ip}\}_{ip=1}^{np}$ and the sets T_{ip} of time instants t for which $X(t)$ associates with P_{ip};
Step 3. For every distinct permutation P_{ip} find the total error resulting if decision is $r(P_{ip}) = i, \quad i = 1\ldots N$

$$J(ip,i) = \sum_{t\in T_{ip}} |d(t) - [\mathcal{O}X(t)]_i|^{\gamma}$$

Step 4. Find for the optimal VROS the rank associated with the permutation P_{ip}

$$r(P_{ip}) = i^* = \arg\min_i J(ip,i)$$

Step 5. Find the rank r of the optimal OS filter

$$r = i^* = \arg\min_i \sum_{ip=1}^{np} J(ip,i)$$

Step 6. For every permutation $P_k \notin \mathcal{P}$, associate the rank $r(P_k)$ found in *step 5.* These complete the specification of the optimal VROS filter.

Table 1: Optimal OS and VROS Design Procedure

The solution provided by the optimal VROS procedure in Table 1 gives a lower bound on the J criterion values for WOS filter class. Some cases offering guidelines for WOS filter design are listed below:

a) If $r(P_{ip}) = (N+1)/2$ for all ip, the optimal WOS filter is the median filter and further adaptive search is not necessary;

b) If $r(P_{ip}) = k$ for all ip, the optimal WOS filter will be the k-rank OS filter and further adaptive search is not necessary;

c) If the relative improvement obtained for J criterion in optimal VROS design over optimal OS filter is not significant, further adaptive search for WOS filter is not justified.

2. Neural Network Approximation : Solution for Adaptive WOS Filter Design

2.1 Criterion Surface for WOS Filter Design

As is well known [4], there is a finite number of distinct WOS filters of a given length (since the number of stack filters of fixed length is also finite). The enumeration of all WOS filters of a given length is a difficult task and their number is very big, even for moderate size windows (greater than 9). This makes direct search over the WOS filter finite set a problem yet impossible to solve in general case.

We consider here the search for the parameters $W = [W_1 \ldots W_N]$ in a continuous space, which can be taken [2] as the N-dimensional hypercube $[0\ ,\ W_{N+1}]^N$, with the threshold fixed at the value $W_{N+1} = N + 0.5$.

The success of any optimum search method applied to minimization problem (4) will be strongly affected by the shape of the hypersurface obtained by evaluating the criterion J in the parameter space.

Some relevant facts about the surface shape are addressed below:

1) It may be easily seen that modifying every weight value with a small enough quantity, the inequalities which are implied in WOS definition will not change their signs and consequently the WOS

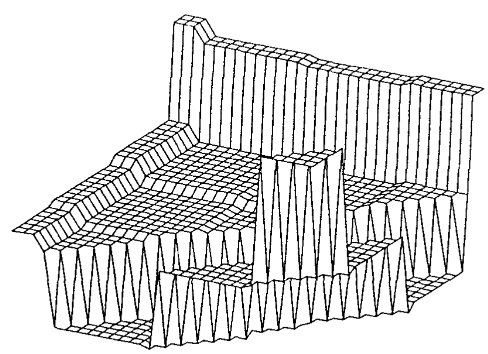

Figure 1: MAE criterion in (W_1, W_2) coordinates

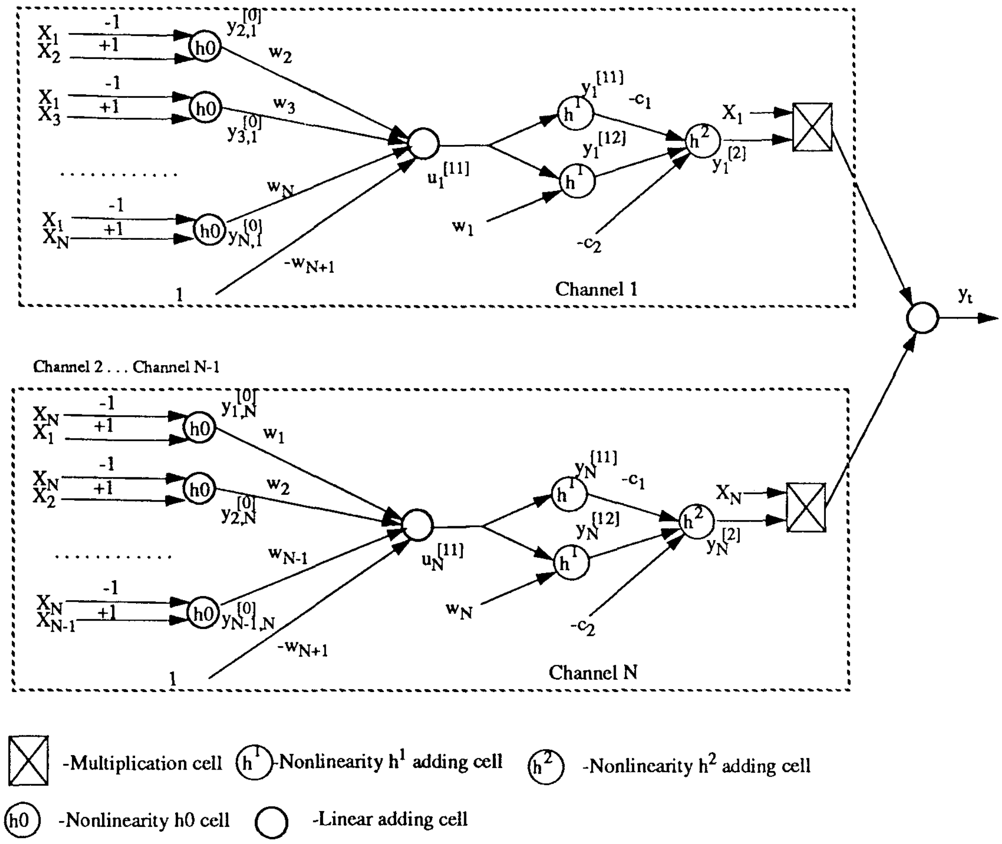

Figure 2: WOS filter parallel implementation

filter with modified weights will be identical to the initial one. This implies that any criterion of the form (4) is a locally constant function almost everywhere, excepting a zero measure set (for which at least one inequality becomes equality) where discontinuities appear. This aspect is illustrated in Figure 1 for a 5-length WOS filter, where the weights were initially assigned to median filter and then the weights W_1 and W_2 were varied in the admissibility region described above.

2) Gradient search methods for this problem will fail, since almost everywhere the gradient is zero.

In order to cope with such an ill conditioned optimization problem, it is obvious that something in the structure of the WOS filter must be controlled, to eliminate the roughness of decision making for a while until significant steps in optimization are made. Smoothing the criterion surface with respect to the parameters will make possible the use of gradient based methods.

We depicted in Figure 2 the structure of the WOS filter in order to show the points where hard decisions occur.

There are N similar processing channels, the i'th channel testing if the final decision will be the selection of X_i as output of the filter, only one of the channel outputs being different from 0, the processing flow is described by the set of equations in Table 2.

for k = 1 to N

  for j = 1 to N

$$y_{jk}^{[0]} = h^0(X_j - X_k) = \begin{cases} 1, & \text{if } X_j - X_k > 0; \\ 1, & \text{if } X_j - X_k = 0 \text{ and } j < k; \\ 0, & \text{else.} \end{cases} \quad \text{(WOS 1)}$$

$$u_k^{[11]} = \sum_{\substack{i=1 \\ i \neq k}}^{N} W_i y_{ik}^{[0]} - W_{N+1} \quad \text{(WOS 2)}$$

$$y_k^{[11]} = h^1(u_k^{[11]}) = \begin{cases} 1, & \text{if } u_k^{[11]} > 0; \\ 0, & \text{else.} \end{cases} \quad \text{(WOS 3)}$$

$$y_k^{[12]} = h^1(u_k^{[11]} + w_k) \quad \text{(WOS 4)}$$

$$u_k^{[2]} = y_k^{[12]} - 0.4 \cdot y_k^{[11]} - 0.7 \quad \text{(WOS 5)}$$

$$y_k^{[2]} = h^2(u_k^{[2]}) = \begin{cases} 1, & \text{if } y_k^{[12]} - 0.4 y_k^{[11]} - 0.7; \\ 0, & \text{else.} \end{cases} \quad \text{(WOS 6)}$$

$$y_t = \sum_{k=1}^{N} X_k y_k^{[2]} \quad \text{(WOS 7)}$$

Table 2: WOS Filter Algorithm

2.2 Adaptive WOS Filter Design using Neural Network Training

The WOS filter class will be replaced in our minimization problem with another class which contains the former and for which the minimization is feasible. The optimization will be carried in such a way that at the final search stage, the optimum will be a WOS filter. In order to use a gradient based minimization method, the gradient direction must be defined in almost every point in the parameter space. This can be obtained if in all cases when a weight changes, the criterion value changes too. If the step nonlinearities will be replaced with smooth nonlinearities in the third and forth level in Figure 2 (h^1 and h^2 cells), any change in the weights value will produce a variation in the output, which will affect the criterion value.

Forward Pass

for k = 1 to N

for j = 1 to N

$$y_{jk}^{[0]} = h^0(X_j - X_k) = \begin{cases} 1, & \text{if } X_j - X_k > 0; \\ 1, & \text{if } X_j - X_k = 0 \text{ and } j < k; \\ 0, & \text{else.} \end{cases} \quad \text{(FOR 1)}$$

$$u_k^{[11]} = \sum_{\substack{i=1 \\ i \neq k}}^{N} W_i y_{ik}^{[0]} - W_{N+1} \quad \text{(FOR 2)}$$

$$y_k^{[11]} = s^1(\alpha, u_k^{[11]}) = \frac{1}{1 + e^{-\alpha u_k^{[11]}}} \quad \text{(FOR 3)}$$

$$y_k^{[12]} = s^1(\alpha, u_k^{[11]} + W_k) \quad \text{(FOR 4)}$$

$$u_k^{[2]} = y_k^{[12]} - 0.4 \cdot y_k^{[11]} - 0.7 \quad \text{(FOR 5)}$$

$$y_k^{[2]} = s^2(\gamma, u_k^{[2]}) = \frac{1}{1 + e^{-\gamma u_k^{[2]}}} \quad \text{(FOR 6)}$$

$$y_t = \sum_{k=1}^{N} X_k y_k^{[2]} \quad \text{(FOR 7)}$$

$$e_t = d_t - y_t \quad \text{(FOR 8)}$$

$$J_t = e_t s_{-1,1}(\beta, e_t) = e_t \frac{1 - e^{-\beta e_t}}{1 + e - \beta e_t} \quad \text{(FOR 9)}$$

$$J = J + J_t \quad \text{(FOR 10)}$$

Backward Pass

$$\delta = \frac{dJ_t}{dy_t} = -(s_{-1,1}(\beta, e_t) + 0.5 e_t (1 - s_{-1,1}(\beta, e_t))(1 + s_{-1,1}(\beta, e_t))) \quad \text{(BAK 1)}$$

for k = 1 to N

$$\delta_k^{[2]} = \frac{dJ_t}{du_k^{[2]}} = \delta x_k \beta y_k^{[2]}(1 - y_k^{[2]}) \quad \text{(BAK 2)}$$

$$\delta_k^{[11]} = \frac{dJ_t}{du_k^{[11]}} = \delta_k^{[2]}(-c_1)\alpha y_k^{[11]}(1 - y_k^{[11]}); \quad \text{(BAK 3)}$$

$$\delta_k^{[12]} = \frac{dJ_t}{du_k^{[12]}} = \delta_k^{[2]} \alpha y_k^{[12]}(1 - y_k^{[12]}) \quad \text{(BAK 4)}$$

$$dJ_t/dW_k = \delta_k^{[12]} + \sum_{\substack{i=1 \\ i \neq k}}^{N} (\delta_i^{[11]} + \delta_i^{[12]}) y_{ki}^{[0]} \quad \text{(BAK 5)}$$

$$(dJ/dW_k)_t = (dJ/dW_k)_{t-1} + \frac{dJ_t}{dW_k} \quad \text{(BAK 6)}$$

Weights Update

$$(W_k)^* = Pr_{W^{adm}}[(W_k)^* - \eta_i (\frac{dJ}{dW_k})_T], \text{ for } k = 1 \text{ to } N \quad \text{(WU 1)}$$

Table 3: Neural Network model and gradient computation

The structure in Fig.2 will be modified such that the hard decision elements with nonlinearities h^1 and h^2 will be replaced with sigmoidal type nonlinearities

$$s(\alpha, x) = 1/(1 + e^{-\alpha x}). \quad (5)$$

Increasing the value of α, the sigmoidal function can be made as close as needed to a hard nonlinearity, in the late stages of the search.

For the MAE criterion, the gradient computation is not well defined when the error is zero due to nondifferentiability at origin of the absolute value function. To overcome this, the absolute value function will be smoothed also using one sigmoidal-type nonlinearity, but with different asymptotic levels, - 1 and 1, instead of those asymptotic levels used in (5) which are 0 and 1:

$$\begin{aligned} |x| &= x\text{sign}(x) \\ &\approx x s_{-1,1}(\beta, x) \\ &= x \frac{1 - e^{-\beta x}}{1 + e^{-\beta x}} \end{aligned} \quad (6)$$

The MAE criterion will become

$$J(W, \alpha, \beta, \gamma) = \sum_{t=1}^{T} J_t = \sum_{t=1}^{T} |d_t - -y_t(W, \alpha, \beta, \gamma)|, \quad (7)$$

where J_t denote the "smoothed" mean valued of the error at time t.

We are now able to compute the gradient for the criterion (7) as an application of the general Forward Backward Training Algorithm [3] to the structure in Figure 2.

The usual problem in Neural Network modelling is to approximate, with a NN model, a given input-output pattern distribution. Our problem is different in the major aspect that the input-output pattern distribution must be obtained for a *specific structure model* (with known parameters in most of the layers, and with hard nonlinearities).This fact determines the specific structure of the Search strategy in Fig.4.

The innermost loop (Loop 2 in Fig. 4) updates the vector W using the gradient of criterion (7) with respect to the parameters in vector W. It tries to override the traps represented by the local minima in the following way: after the gradient direction is computed, a search on this direction is performed, looking for an "optimal" step. If the criterion does not improve in any of the points tested along the negative gradient direction, the iteration of Loop 2 will be stopped; otherwise the parameter updating will be continued, using the optimal step.

The Loop 1 in Fig. 4 is used to modify the steepness of the nonlinearities, using the parameters α, β, γ. These parameters determine the closeness

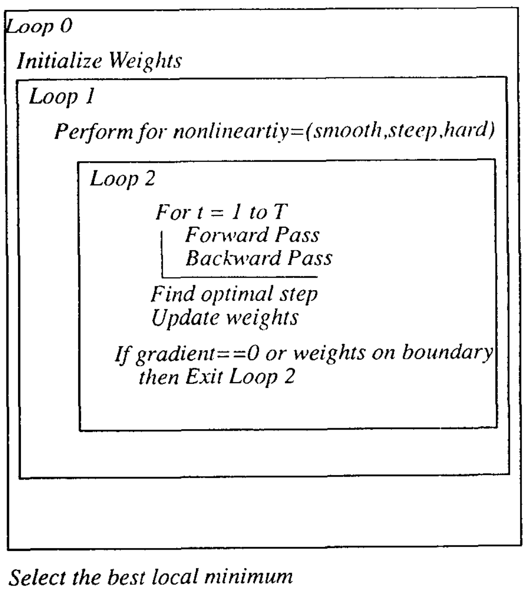

Figure 4: Search Strategy

of the NN model to a WOS filter but also affect criterion (7). We allow only the increasing of steepness, even if this will increase the value of criterion (7); this is similar to a "deterministic simulated annealing" procedure since the α, β and γ parameters can be seen as inverse proportional to the "temperature" in the corresponding NN layers.

The outermost loop, Loop 0 in Fig. 4 will be used for restarting the algorithm after a local optimum NN model (very close to a WOS filter) was found. The starting parameter point will be randomly generated uniformly in the admissible space described in 2.1. In the final stage of the algorithm, the best local minimum point will be selected from the local minimum points obtained during training.

3. Simulation results

In this section some results relevant for a 5-length window filter design will be presented, for an artificial training set in which :

- The desired input is a 10-sample period square wave of levels -20 and 20.
- The input signal $x(t)$ is the sum of the desired signal and an independent and identically distributed noise with Gaussian-contaminated pdf

$$\epsilon_t \sim (1-\xi)\mathcal{N}(0,3) + \xi\mathcal{N}(0,3/\xi)$$

with contamination $\xi = 0.1$.

3.1 Optimal OS and VROS filters

The running of the optimal OS and VROS design procedure from Table 1, produced the following results : the optimal OS filter is the median filter, for which the value of the MAE criterion is $J_{median} = 2.61$. In the data set there were 108 different permutations, for which we associated the optimal ranks as in Table 1. To the 12 permutations which were not found in the data set it was associated the rank 3, as in the optimal OS (median); the value of the MAE criterion for this filter was J=1.29. Since optimal criterion value obtained in VROS filter class shows a significant improvement over the optimal OS filter class, one can expect that the optimal WOS filter also will constitute a significant improvement over OS class.

3.2 Optimal WOS filter

An exhaustive search procedure was applied in order to evaluate all different WOS filters for criterion (4), with $\rho = 1$ and $\rho = 2$. From Figure 5 one can see that the MAE and MSE criteria give similar results in ranking the various WOS filters. The optimal MAE criterion, J=2.008, is obtained for $W = [\ 1\ 2\ 4\ 2\ 1\ ]$. The Figure 5 reveals that in this example there are more than 180 filters which behaves better than optimal OS filter, but also there are more than 1100 filters which behaves much worse.

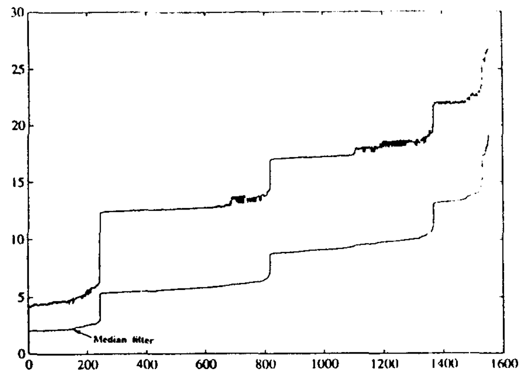

Figure 5: MAE and MSE criteria for all different 5-length WOS filters

3.3 Adaptive WOS Design

We have applied the procedure described in section 2, using 23 different initial starting points, selected uniformly in the admissible parameter space and using a policy of steepening the nonlinearities in three steps: $(\alpha_1 = 1; \gamma_1 = 50; \beta_1 = 50)$, $(\alpha_2 = 10; \gamma_2 = 100; \beta_2 = 100)$ and $(\alpha_3 = 100; \gamma_3 = 200; \beta_3 = 200)$. The results, plotted in Figure 6, show the efficient way in which the parameters are modified in order to improve the criterion values.

4. Conclusions

Optimal WOS filter design was shown to be a difficult problem. This paper reveals some aspects which were not considered in the previous attempts to solving this problem.

In many situations there are a many WOS filters which perform better than optimal OS filters, but there also exist a large number of filters with worse behavior. Consequently, the effectiveness of the procedure used for WOS filter design seems to be a crucial criterion for using WOS filters.

Our procedure was shown to cope successfully with the ill conditioned problem of adaptive WOS filter design, providing effective solutions.

A practical approach to use the procedures described in this paper may be the following:

a)Preprocessing level : Use the procedure in Table 1 to find the Optimal OS filter and optimal VROS filter and decide, according to 1.3 whether or not the WOS filter design is needed.

b) Use the adaptive WOS filter design procedure to obtain a sequence of suboptimal WOS filters, until a filter is found which improves significantly the value of the MAE criterion obtained in optimal OS filter.

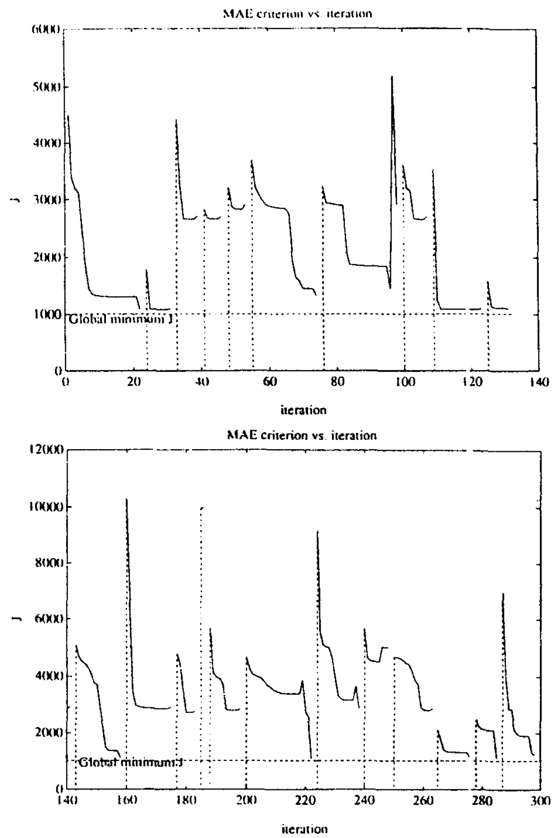

Figure 6: MAE during training

References

[1] O.Yli Harja,J.Astola,Y.Neuvo, " *Analysis of the properties of median and weighted median filters using the threshold logic and stack filter representations*" IEEE Trans. Signal Processing,39,pp.395-410,Feb.1991.

[2] I.Tabus,M.Gabbouj,Y.Lin " *Real Domain WOS Filtering Using Neural Network Approximations*", manuscript in preparation.

[3] D.E.Rumelhart, J.L. McClelland, " *Parallel Distributed Processing*",Cambridge,MA: M.I.T. Press, 1986.

[4] M.K.Prasad,Y.H.Lee " *Analysis of Weighted Median Filters Based on Inequalities Relating the Weights*"Circuits ,Systems,Signal Process.,11,pp.115-136,1,1991.

[5] L.Yin,J.Astola,Y.Neuvo," *Adaptive Weighted Order Statistic Filter Using Backpropagation Algorithm*"Proc.IEEE Int.Symp.Circ.Sys, pp.499-502,New Orleans, Louisiana,1990.