INTRINSICALLY STABLE IIR FILTERS AND IIR-MLP NEURAL NETWORKS FOR SIGNAL PROCESSING

Paolo Campolucci[*] and Francesco Piazza[+]

Dipartimento di Elettronica e Automatica
Università di Ancona, 60131 Ancona, Italia.
*E-mail: paolo@eealab.unian.it; Internet: http://nnsp.eealab.unian.it

ABSTRACT

This paper presents a new technique to control stability of IIR adaptive filters based on the idea of intrinsically stable operations that makes possible to continually adapt the coefficients with no need of stability test or poles projection.

The coefficients are adapted in a way that intrinsically assures the poles to be in the unit circle. This makes possible to use an higher step size (also named learning rate here) potentially improving the fastness of adaptation with respect to methods that employ a bound on the learning rate or methods that simply do not control stability.

This method can be applied to various realizations: direct forms, cascade or parallel of second order sections, lattice form. It can be implemented to adapt a simple IIR adaptive filter or a locally recurrent neural network such as the IIR-MLP.

1. INTRODUCTION

The capabilities of Locally Recurrent Neural Networks (LRNNs) in performing on-line Signal Processing (SP) tasks are well known [1,3,5,6,10,11,14,15]. In particular one of the most popular architecture is the Multi Layer Perceptron (MLP) with linear IIR temporal filter synapses (IIR-MLP) [3,5,10,11,14,15]. IIR-MLP is theoretically motivated as a non-linear generalization of linear adaptive IIR filters [13] and as a generalization of the popular Time Delay Neural Networks (TDNNs) [1,2,4]. In fact TDNN can be viewed as MLP with FIR temporal filter synapses (FIR-MLP) [2,3,5]. Therefore IIR-MLP are a generalization of FIR-MLP (or TDNN) allowing the temporal filters to have a recursive part. Efficient training algorithms can be developed for general LRNNs and so the IIR-MLP [10,11,14,15]. They are based on Back Propagation Through Time of the error [2] to propagate the sensitivities through time and network layers, and on a local recursive computation of output error (RPE type) [9,13]. They are named Causal Recursive Back Propagation (CRBP) [10,11,15] and Truncated Recursive Back Propagation (TRBP) [14] and they differ in the technique implemented to employ on-line computation. They are both on-line and local in space and in time, i.e. of easy implementation, and their complexity is limited and affordable. They generalize the Back-Tsoi algorithm [3], the algorithms in [4,6], the one by Wan [2] and standard Back Propagation. Even if CRBP and TRBP performs in a quite stable manner if the learning rate is chosen small enough by the user they do not control the stability of the IIR synapses (for the IIR-MLP) or of the recursive filters (for general LRNNs). In the following we will refer mostly to the IIR-MLP case but the extension to other LRNNs are possible and easy in most cases. The same limitation is found in the all literature of learning methods for RNNs and LRNNs, e.g. [2,3,6,9]. The problem with general RNN is that is not even easy to derive necessary and sufficient conditions for the coefficients of the network to assure asymptotic stability even in the time invariant case since the feedback loop include the non-linearity. On the other hand for LRNNs the recursion is usually separated from the non-linearity, as for the IIR-MLP. Therefore in batch mode the overall IIR-MLP is asymptotically stable if and only if each of the IIR filters is asymptotically stable, i.e. all transfer functions poles have a module less than one.

In the time-variant case (on-line adaptation) the above condition is not sufficient anymore. A “slow coefficient-variation” condition must be added to assure stability [8,12]. Even if this is what the theory state, in practice the second condition is often ignored since for practical signals the condition on the poles is sufficient [13] and the cases in which this is not true are “pathological” [13]. The second condition is important when the IIR filter must operate near the instability region [12]. On the contrary in the linear IIR adaptive filter literature there are various techniques that can be employed to control stability [13,8]. The simplest is to monitor the poles and do not adapt at the time when this will bring the poles outside the circle. Since to monitor stability efficient criteria are available such as Jury’s one, that avoids to compute the poles, this method is simpler, but slower and less robust than pole projection methods [13]. To make stability check easier, the IIR filter can be realized as cascade or parallel of second order sections [13,12] or in lattice form [12,7].

Other stability assurance techniques include the hyperstable adaptive recursive filter (HARF) which imposes conditions on the transfer function not easy to guarantee in practice [8].

This paper presents a new technique to control stability of IIR adaptive filters based on the idea of intrinsically stable operations that makes possible to continually adapt the coefficients with no need of stability test or poles projection.

The coefficients are adapted in a way that intrinsically assures the poles to be in the unit circle. This makes possible to use an higher step size (also named learning rate here) potentially improving the fastness of adaptation with respect to methods that employ a bound on the learning rate, e.g. [8] or methods that simply do not control stability.

[+] This research was supported by the Italian MURST.

This method can be applied to various realizations: direct forms, cascade or parallel of second order sections, lattice form. It can be implemented to adapt a simple IIR adaptive filter and on a locally recurrent neural network such as the IIR-MLP.

In the next two sections the Intrinsic Stability technique will be explained and then simulation results will be presented.

2. INTRINSIC STABILITY TECHNIQUE

The basic idea underlying this technique is to restrict the region of the values that the coefficients can take so that the filter is stable. In the following we will assume that the working conditions of the filter are such that the filter stability is assured if the time invariant conditions are satisfied. Otherwise it is always possible to add to this new method a control of the step size that assures the "slow-varying" condition, but the development of such a bound for the step size is beyond the aim of the paper.

2.1 First Order Case

For the sake of clarity the explanation of the method will start from the simplest case, the first order IIR filter.

The input-output relationship is:

$$y[t] = w_0[t]x[t] + w_1[t]x[t-1] + v_1[t]y[t-1] \tag{1}$$

The transfer function is:

$$\frac{w_0[t] + w_1[t]z^{-1}}{1 - v_1[t]z^{-1}} \tag{2}$$

Since the pole in the z plane is $v_1[t]$, the filter is stable if

$$|v_1[t]| < 1 \quad \forall t \tag{3}$$

Now the idea is to introduce a non-linear compressing transformation from a new coefficient to be adapted to $v_1[t]$:

$$v_1[t] = \Psi(\hat{v}_1[t]) \tag{4}$$

$\hat{v}_1[t]$ is named virtual coefficient.

Choosing $\Psi(.)$ as a non-linear squashing function in the range (-1,1) assure $v_1[t]$ to satisfy the stability condition.

A good choice for $\Psi(.)$ is the hyperbolic tangent:

$$g(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \tag{5}$$

Anyway even if other choices are possible other than $\Psi(.) = g(.)$, the $\Psi(.)$ function is requested to be squashing in the range (-1,1), continuos and first order differentiable.

The new parameter $\hat{v}_1[t]$ can be adapted instead of $v_1[t]$ that will be computed by (4). To adapt $\hat{v}_1[t]$ a gradient descent method can be employed with only a little modification with respect to the adaptation of $v_1[t]$, as explained in the following.

$$\hat{v}_1[t+1] = \hat{v}_1[t] - \mu[t]\frac{\partial e^2[t]}{\partial \hat{v}_1[t]} \tag{6}$$

where $e[t] = d[t] - y[t]$, $\mu[t]$ is the step size and $d[t]$ the target signal.

In the following we will assume that the cost function is the instantaneous squared error but an average over a running window can also be easily implemented.

It holds

$$\frac{\partial e^2[t]}{\partial \hat{v}_1[t]} = 2e[t]\frac{\partial e\ [t]}{\partial \hat{v}_1[t]} = -2e[t]\frac{\partial y[t]}{\partial \hat{v}_1[t]} \tag{7}$$

and

$$\frac{\partial y[t]}{\partial \hat{v}_1[t]} = \frac{\partial y[t]}{\partial v_1[t]}\frac{\partial v_1[t]}{\partial \hat{v}_1[t]} = \frac{\partial y[t]}{\partial v_1[t]}\Psi'(\hat{v}_1[t]) \tag{8}$$

The well known recursive expressions to compute the derivative of the IIR filter output with respect to the denominator coefficients (output error RPE) can still be used with the only correction of the multiplication by the derivative of the $\Psi(.)$ function as stated by the last expression. For the hyperbolic tangent it simply holds:

$$g'(x) = \frac{1}{2}[1 - g^2(x)] \tag{9}$$

That is very simple to compute since $g(x)$ is already computed by (4). Therefore after adapting the $\hat{v}_1[t]$ coefficient the corresponding $v_1[t]$ must be computed by (4). Then only $v_1[t]$ will be used to compute the new filter output by (1).

The effect of the squashing function is the following. When the pole of the filter $v_1[t]$ is going near the stability region boundary, $\Psi(.)$ will be computed in the flat region so that $\Psi'(.)$ will be small and the adaptation slower. A part of the effect is like decreasing the step size when going to instability but it must be stressed that this technique cannot be viewed just as a control of the step size. The reason is that not only the step size is decreased but also the coefficient is bounded by (4) therefore assuring stability even if the step size is not decreased too much. It appear reasonable to us that any technique that just controls the step size even by some accurate bound will have to reduce the step size more than the technique implementing (4) and so it will be slower. The Intrinsic Stability (IS) technique allows to avoid any stability test or pole projection method.

2.2 Higher Order Cases

This method can be easily applied to the lattice form since in that case the stability conditions state that the reflection coefficients must be less than one in absolute value. Therefore for any order of the filter the IS method can assure the stability of the lattice form simply applying the squashing transformation to each of the reflection coefficients and proceeding as explained.

The IS technique can be easily extended to second order sections, allowing the use of parallel and cascade form, as in the following.

The transfer function of a second order IIR filter is

$$\frac{w_0[t]+w_1[t]z^{-1}+w_2[t]z^{-2}}{1-v_1[t]z^{-1}-v_2[t]z^{-2}} \tag{10}$$

The necessary and sufficient conditions for stability are now

$$\begin{cases} |v_2[t]|<1 \\ |v_1[t]|<1-v_2[t] \end{cases} \tag{11}$$

for each *t*.

Therefore the squashing transformation is to be done as:

$$v_2[t]=\Psi(\hat{v}_2[t]) \tag{12}$$

$$v_1[t]=(1-v_2[t])\Psi(\hat{v}_1[t]) \tag{13}$$

Where the second equation must be computed after the first one, since the new value of $v_2[t]$ must be available.

The last two expressions allow satisfying exactly the stability conditions with *no restriction of the stable region*.

The two $\hat{v}_1[t]$ and $\hat{v}_2[t]$ coefficients can be independently adapted by gradient descent and then $v_1[t]$ and $v_2[t]$ can be computed by the last two expressions.

To generalize the IS method to a general order IIR filter in direct form the poles must be represented in module and phase form. The modules can be squashed in the range (0,1) by a unipolar sigmoid:

$$f(x)=\frac{1}{1+e^{-x}} \tag{14}$$

or again by $g(x)$ in the range (-1,1) to make easier to realize the pole in the origin.

On the other hand the phases can be adapted with no restriction since they do not influence stability.

A gradient descent in the virtual modules (i.e. before compression) and in the real phases of the poles can be implemented. Then the real modules must be found evaluating $\Psi(.)$ and then the coefficients of the denominator of the transfer function computed from the poles. Must be stressed that although is necessary to implement the transformation from poles to coefficients is not necessary to implement the inverse from coefficients to poles that is much more complex.

The other overhead of the IS technique in the general order case with respect to the first or second order case is that the derivatives of the coefficients with respect to the real modules and phases of the poles must be computed to implement the gradient descent in the poles space as a modification of the standard output error RPE. The formulas are not complex to derive and compute but they cannot be shown here, for the sake of space limitation.

3. SHAPE ADAPTATION OF THE SQUASHING FUNCTION

In any of the realization forms of the IIR filter a number of squashing functions must be implemented equal to the order of the filter for a single IIR filter or for each of the synapses of the IIR-MLP neural network.

Therefore a question should be addressed: is it reasonable to implement exactly the same squashing function for all the IIR filters or IIR-MLP network ? Or may be the function should be optimized in each case ?

We believe that a much improved technique should adapt the squashing function depending on the parameter to be squashed. The simulations performed confirm this idea.

A good way to optimize $\Psi(.)$ is to make it automatically adaptable because the number of coefficients can be so big that is not realistic to optimize each function by hand. Moreover there is no way to a-priori decide for the shape of $\Psi(.)$ without accounting for the current adaptation process.

Therefore we propose the use of non-linear squashing functions that can be automatically and independently adapted.

To save complexity it is very reasonable to implement $\Psi(.)$ as an hyperbolic tangent with adaptive slope, i.e.:

$$g(x,s)=\frac{1-e^{-sx}}{1+e^{-sx}} \tag{15}$$

where *s* is named slope which must be adapted by gradient descent independently from the coefficients of the filter.

In this way the slope of the compression can be optimized for each coefficient to be squashed and the permitted range is (-1,1) unchanged.

Since it holds:

$$\left.\frac{\partial g(x,s)}{\partial x}\right|_{x=0}=\frac{s}{2} \tag{16}$$

it is clear that increasing *s* make the function $g(.)$ closer to the threshold and therefore the IS technique can easier, i.e. faster, place poles near the unit circle.

This can be important when the requested behavior is near the instability that moreover is just the case when a stability control should be employed. The simulations verify this intuition.

The gradient descent on *s* can be employed by the analogous of (6), (7) and the following expression in the first order case (extension is feasible)

$$\frac{\partial y[t]}{\partial s[t]}=\frac{\partial y[t]}{\partial v_1[t]}\frac{\partial v_1[t]}{\partial s[t]}=\frac{\partial y[t]}{\partial v_1[t]}\frac{\partial \Psi(\hat{v}_1[t],s[t])}{\partial s[t]} \tag{17}$$

For the hyperbolic tangent it holds:

$$\frac{\partial g(x,s)}{\partial s}=\frac{x}{2}[1-g^2(x,s)] \tag{18}$$

that again is easy to compute since $g(x,s)$ is already calculated.

The IS technique with Adaptive slope will be named AIS method.

4. SIMULATIONS RESULTS

In this section the AIS method will be applied to the training of an IIR-MLP neural network by the TRBP algorithm [14].

The test problems chosen are two difficult problems of on-line non-linear dynamical system identification.

The neural networks used have 2 layers, 3 hidden sigmoidal neurons, 1 output linear neuron.

The results are shown in terms of plots of the MSE (in dB) vs. iterations (each iteration is an entire learning epoch) averaged over 20 runs each with a different coefficient initialization. The variance is also shown on the top right corner of the plots.

The first experiment is a non-linear ARMA system identification problem proposed in [16], simulated under the same conditions. The network used has all the synapses with 2 zeros and 2 poles.

Fig. 1 shows the improved performances of the AIS technique over not use any stability control. In this case the adaptation of the slope is necessary.

The second test is the identification of a 16-PAM transmission channel in presence of a non-linearity, see [10] for details.

The network used has all the synapses with 4 zeros and 2 poles.

Fig. 2 again shows the improved performances of the AIS technique over not use any stability control.

5. REFERENCES

[1] S. Haykin, "*Neural Networks Expand Sp's Horizons*" IEEE Signal Processing Magazine, vol.13 no. 2, March 1996.

[2] S. Haykin, "*Neural Networks: a comprehensive foundation*", IEEE Press-Macmillan, 1994.

[3] A.D. Back, A.C. Tsoi, "*FIR and IIR synapses, a new neural network architecture for time series modelling*", Neural Computation 3: 375-385, 1991.

[4] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, K.J. Lang, "*Phoneme recognition using time-delay neural networks*", IEEE Trans. on Acoustic, Speech, and Signal Processing, Vol. 37, No.3, March 1989.

[5] A.C. Tsoi, A.D. Back, "*Locally recurrent globally feedforward networks: a critical review of architectures*", IEEE Transactions on Neural Networks, vol. 5, no. 2, 229-239, March 1994.

[6] P.Frasconi, M.Gori, G.Soda, "*Local Feedback Multilayered Networks*", Neural Computation 4: 120-130, 1992.

[7] A. D. Back and A. C. Tsoi, "*An Adaptive Lattice Architecture for Dynamic Multilayer Perceptrons*", Neural Computation 4, 922-931, 1992.

[8] A. Carini, V. J. Mathews, G. Sicuranza, "*Sufficient Stability Bounds for Slowly varying Discrete-Time Recursive Linear Filters*", Proc. of ICASSP-97 IEEE Int. Conference on Acoustic, Speech and Signal Processing, Munich (Germany), April 1997.

[9] R.J. Williams, D. Zipser, "*A Learning Algorithm for Continually Running Fully Recurrent Neural Networks*", Neural Computation 1: 270-280, 1989.

[10] P. Campolucci, F. Piazza, A. Uncini, "*On-line learning algorithms for neural networks with IIR synapses*", Proc. of the IEEE International Conference of Neural Networks, Perth, Nov. 1995.

[11] P. Campolucci, A. Uncini, F. Piazza, "*Causal Back Propagation Through Time for Locally Recurrent Neural Networks*", IEEE International Symposium on Circuits and Systems, Atlanta, May 1996.

[12] P. A. Regalia, "*Adaptive IIR filtering in Signal Processing and Control*", Marcel Dekker Inc., New York, 1995.

[13] J.J. Shynk, "*Adaptive IIR filtering*", IEEE ASSP Magazine, April 1989.

[14] P. Campolucci, A. Uncini, F. Piazza, "*A new IIR-MLP learning algorithm for on-line signal processing*" Proc. of ICASSP-97 IEEE Int. Conference on Acoustic, Speech and Signal Processing, Munich (Germany), April 1997.

[15] P. Campolucci, A. Uncini, F. Piazza, "*Fast adaptive IIR-MLP neural networks for signal processing applications*" Proc. of ICASSP-96 IEEE Int. Conference on Acoustic, Speech and Signal Processing, Atlanta (USA), May 1996.

[16] A.D. Back, A.C. Tsoi, "*A simplified gradient algorithm for IIR synapse Multilayer Perceptron*", Neural Computation 5: 456-462, 1993.

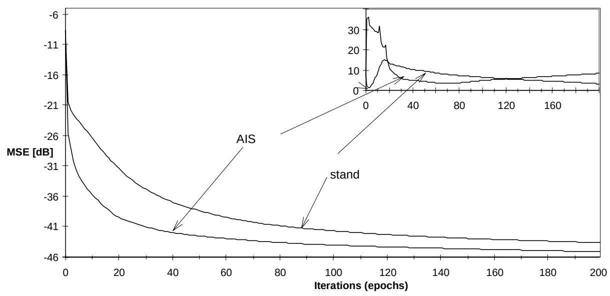

Figure 1. IIR-MLP learning performances identifying the Back-Tsoi test system by the TRBP(8,2) algorithm implementing: standard method (no stability control) and AIS technique. μ=0.03.

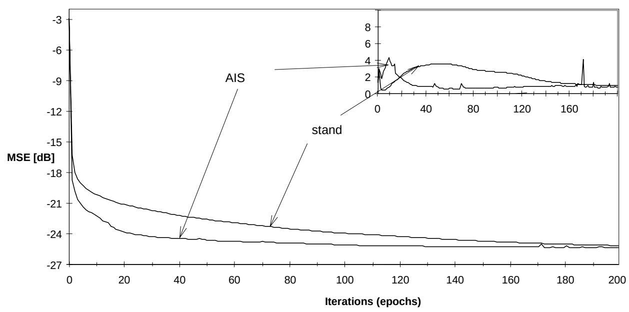

Figure 2. IIR-MLP learning performances identifying the 16-PAM test system by the TRBP(8,2) algorithm implementing: standard method and AIS technique. μ=0.01 for TRBP(8,2) standard. For AIS: μ=0.05.