PD Library: Adaptive Filter
Published under GPL License

Authors: Markus Noisternig (noisternig_AT_iem_DOT_at)
Thomas Musil (musi_AT_iem_DOT_at)

1. Introduction

The aim of this project is to extend Miller Puckettes open source real-time computermusic software Pure Data (PD) by a library containing several algorithms for least mean square (LMS) adaptive filtering. This library is published under GPL.

2. Algorithms

2.1 Leaky Normalized Least Mean Square Adaptive Filter (NLMS)

The LMS algorithm is an important member of the family of *stochastic gradient algorithms* and it is widely used in signal processing applications. Significant features of the LMS algorithm are its *simplicity of implementation* and its *model-independent, and therefore robust, performance.* Unfortunately, the main limitation of LMS filters is their *relatively slow rate of convergence*. Two principle factors behave the convergence of the LMS filter: *the step-sice parameter* and the *eigenvalues of the correlation matrix of the tap-input vector.*
The normalized LMS is exactly the same as the standard LMS filter, but the weight controller applies a weight adjustment. This normalization is a manifestation of the *principle of minimal disturbance.*
Additionally, to further stabilize the digital implementation of the NLMS algorithm, we may use a technique known as *leakage*. Leakage prevents overflow in a limited-precision environment by providing a compromise between minimizing the mean square error and containing the energy in the impulse response of the adaptive filter. Furthermore, to prevent *division by zero,* an additional *regularization parameter close to zero* is is introduced.
The leaky NLMS algorithm is summarized in equ. (2.01) to (2.04).

2.2 Coefficient Constrained NLMS Adaptive Filter (NLMSC)

The coefficient constrained NLMS adaptive filter is an extension of the above-mentioned algorithm. Each coefficient growth is constrained by an upper and lower boundary function. This algorithm is used e.g. to improve the robustness of adaptive GSC beamformers. Equ. (2.05) to (2.07) summarize the update rule extensions.

(2.01) $z_k(n) = x_k(n - L_2) - \mathbf{h}_k^T(n)\mathbf{b}(n)$ with $k = 1..K$ input signals

(2.02) $\mathbf{b}(n) = \left[b(n), b(n-1), ..., b(n-J+1)\right]^T$

$$(2.03)\quad \mathbf{h}_k(n) = \left[h_{k,0}(n), h_{k,1}(n), \ldots, h_{k,J-1}(n)\right]^T$$

$$(2.04)\quad \mathbf{h}'_k(n+1) = (1-\alpha\delta).\mathbf{h}_k(n) + \alpha \frac{z_k(n)}{\mathbf{b}(n)^T\mathbf{b}(n)+\varepsilon}\mathbf{b}(n)$$

$$(2.05)\quad \mathbf{h}_k(n+1) = \begin{cases} \boldsymbol{\Xi}_k, & \text{if} \quad \mathbf{h}'_k(n+1) > \boldsymbol{\Xi}_k \\ \boldsymbol{\Psi}_k, & \text{if} \quad \mathbf{h}'_k(n+1) < \boldsymbol{\Psi}_k \\ \mathbf{h}'_k(n+1), & \text{else} \end{cases}$$

$$(2.06)\quad \boldsymbol{\Xi}_k \mathrel{\hat{=}} \left[\Xi_{k,0}, \Xi_{k,1}, \ldots, \Xi_{k,J-1}\right]^T$$

$$(2.07)\quad \boldsymbol{\Psi}_k \mathrel{\hat{=}} \left[\Psi_{k,0}, \Psi_{k,1}, \ldots, \Psi_{k,J-1}\right]^T$$

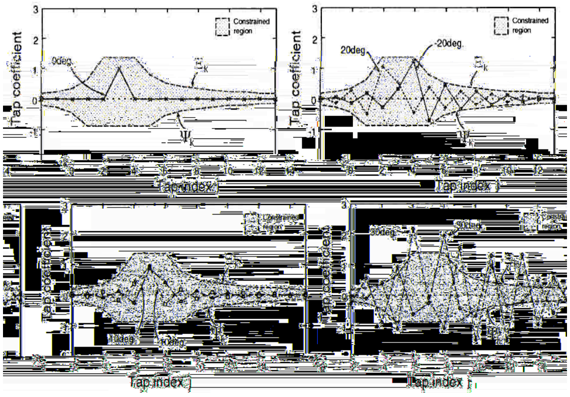


Abb. 2.1: Coefficient Constraining in Adaptive Filtering

2.3 Multi-channel Norm-Constrained NLMS Adaptive Filter (n_CNLMS)

In this multi-channel input single output (MISO) adaptive filter algorithm, the norm constraint by scaling restrains excess growth of tap coefficients. To avoid misadjustment in such multi-channel input systems, implementing leaky coefficient update is inevitable. The n_CNLMS algorithm is summarized in equ. (2.08) to (2.13).

$$(2.08)\quad y(n) = b(n-L_1) - \sum_{k=0}^{K-1} \mathbf{w}_k^T(n)\mathbf{z}_k(n) \qquad \text{with} \qquad k = 1 \ldots K$$

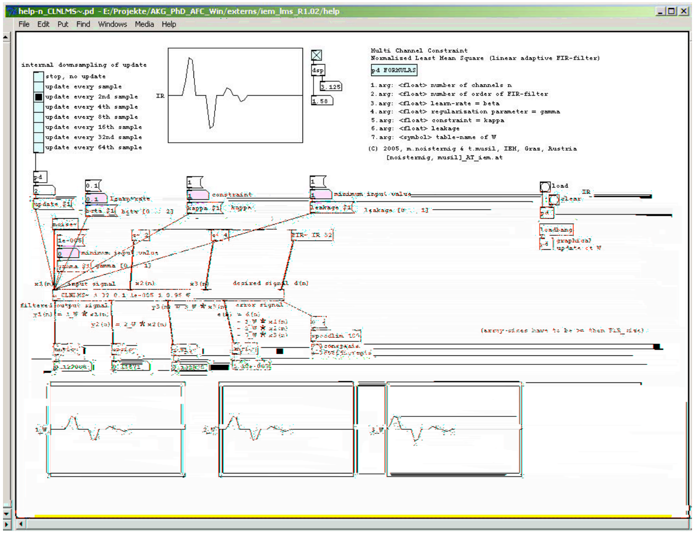
$$(2.09)\quad \mathbf{w}_k(n) = \left[w_{k,0}(n), w_{k,1}(n), \ldots, w_{k,J-1}(n)\right]^T$$

$$(2.10)\quad \mathbf{z}_k(n) = \left[z_k(n), z_k(n-1), \ldots, z_k(n-J+1)\right]^T$$

$$(2.11)\quad \mathbf{w}'_k = (1-\beta\delta).\mathbf{w}_k(n) + \beta \frac{y_k(n)}{\mathbf{z}_k(n)^T\mathbf{z}_k(n)+\varepsilon}\mathbf{z}_k(n)$$

$$(2.12)\quad \Omega = \mathbf{w}'^T_k\,\mathbf{w}'_k$$

$$(2.13)\quad \mathbf{w}_k(n+1) = \begin{cases} \sqrt{\dfrac{\Gamma}{\Omega}}\,\mathbf{w}'_k, & \text{for} \quad \Omega > \Gamma \\ \mathbf{w}'_k, & \text{else} \end{cases}$$



```
internal downsampling of update
stop, no update
update every sample
update every 2nd sample
update every 4th sample
update every 8th sample
update every 16th sample
update every 32nd sample
update every 64th sample
```

```
Multi Channel Constraint
Normalized Least Mean Square (linear adaptive FIR-filter)
pd FORMULAS
1.arg: <float> number of channels n
2.arg: <float> number of order of FIR-filter
3.arg: <float> learn-rate = beta
4.arg: <float> regularisation parameter = gamma
5.arg: <float> constraint = kappa
6.arg: <float> leakage
7.arg: <symbol> table-name of W
(C) 2005, m.noisternig & t.musil, IEM, Graz, Austria
    [noisternig, musil]_AT_iem.at
```

```
n_CLNLMS~ 3 32 0.1 1e-005 1 0.95 W
y1(n) = 1_W * x1(n)
y2(n) = 2_W * x2(n)
y3(n) = 3_W * x3(n)
e(n) = d(n)
 - 1_W * x1(n)
 - 2_W * x2(n)
 - 3_W * x3(n)
(array-sizes have to be >= than FIR_size)
```

Abb. 2.2: PD help for n_CNLMS adaptive filter