RAPID PROTOTYPING LIBRARY FOR ADAPTIVE SIGNAL PROCESSING APPLICATIONS

T. Bigg, J. Owen
Entegra Ltd
2 Waltham Road
Maidenhead
Bucks. SL6 3NH
UNITED KINGDOM
Email: tim@entegra.co.uk

R.W. Stewart, D. García-Alís, M. Harteneck, M. Llovet-Vilà
Signal Processing Division
Dept. of Electronic & Electrical Eng.
University of Strathclyde
Glasgow G1 1XW
UNITED KINGDOM
Email: bob@spd.eee.strath.ac.uk

ABSTRACT

In this paper we present a library for the rapid protypying of adaptive signal processing algorithms, architectures and applications. The library is hosted by the DSP simulation software *SystemView* and covers virtually the complete spectrum of linear, and non-linear adaptive algorithms currently in use in contemporary DSP and communications applications. The library can be easily used with real signals, with variable system wordlengths, sampling frequencies and so on. Therefore in this paper we will briefly discuss the design philosphy behind the library and overview the various algorithms and applications that are implemented. The paper will show an implementation of an adaptive multiuser CDMA receiver/ decision feedback equaliser (DFE) as an example of the relevance and rapid development that is possible. Copies of the library and example files can be downloaded from the web following the instructions in the paper.

1. INTRODUCTION

Over the last 25 years the original work of Widrow [3] in defining the least mean squares (LMS) algorithm and architecture has lead to literally tens of thousands of conference, journal and industrial reports on performance, application and usefulness of the LMS. The LMS algorithm is in daily use in modems, landline and digital mobile telephones, teleconferencing, and even in Windows95 (ADPCM audio wav file compression uses an IIR-LMS for prediction). Looking to the next 5 years, the relevance of adaptive signal processing for the emerging W-CDMA standards (in terms of data equalisation, echo control and multiuser adaptive receivers), and in the implementation of acoustic echo controllers/noise suppressors for hands-free communication means that the LMS is likely to continue to be researched and developed in academia and in industry. However, it is arguable if there is much more research and development to be performed, and the key requirement is for efficient deployment and implementation of the adaptive signal processing.

Surprisingly there do not seem to be any comprehensive commercial adaptive libraries available for the main DSP simulation packages used within industry. Therefore we have developed a very flexible adaptive signal processing library in order that rapid prototyping and implementation of adaptive algorithms can be undertaken.

The algorithms in the library cover the key algorithmic strategies found in the literature over the last few years, namely [6]:

- Least Mean Squares (LMS)
- Recursive Least Squares (RLS) / QR
- Kalman
- Newton Gradient Techniques
- “Open Loop” (Wiener Hopf, Least Squares)

The structure of the library elements is such that FIR, IIR, and multichannel implementations using any of the above classes of algorithms is straightforward to implement. Figure 1 provides an overview of the various algorithms that can be directly implemented via parameter and variable specification.

The objectives in developing the library were to allow a user to:

- Rapidly develop a real world relevant simulation using appropriate adaptive signal processing algorithms;
- Investigate systems for stability & algorithm performance;
- Setup algorithm variants (variable step size, updating rate etc.);
- Consider alternative to LMS algorithms (RLS, Kalman, QR etc);
- Undertake a *bit true* real world relevant simulation.

This paper briefly reviews the library components, and presents a number of developed applications. A comprehensive set of adaptive applications and an evaluation version of the library can be downloaded from `http://www.entegra.co.uk/icassp99.htm`.

2. DSP SIMULATION WITH SYSTEMVIEW

The adaptive library was developed to run within the DSP simulation tool, SystemView by Elanix [1]. SystemView is a token based simulator for Windows95/98/NT and has the capability to allow professional development of complex DSP and communication system simulations, and to provide support for post-simulation data analysis. A demonstration version of the tool is available on the web at `http://www.elanix.com`.

By interconnection of block diagram type tokens SystemView allows the interconnection of more complex systems typically including digital filters, transform coders, quantisers, samplers,

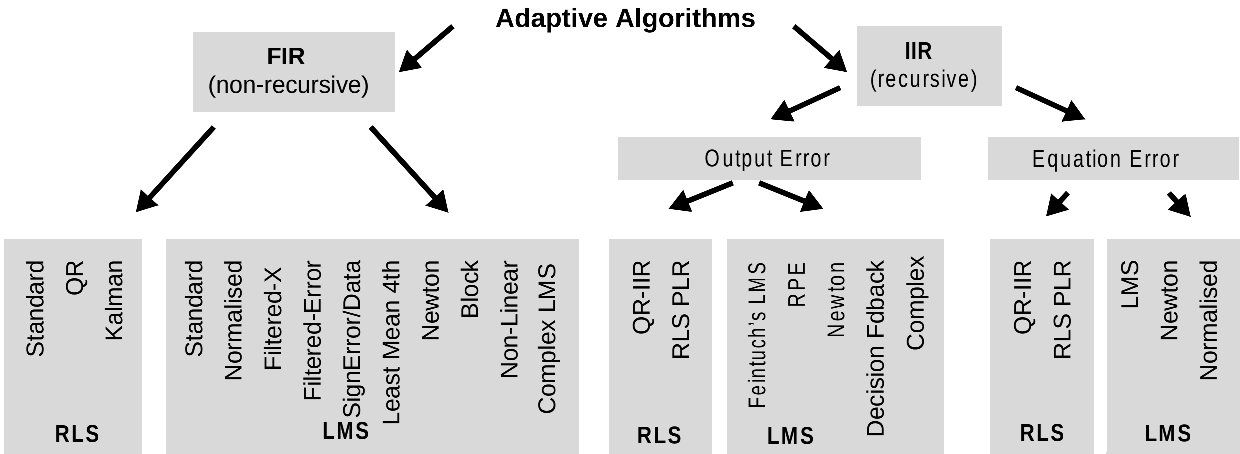


Figure 1: Adaptive library algorithms.

non-linear elements, modulators, and so on. Simulation source data can be generated by source tokens (sine, Gaussian, PRBS etc) and system outputs sent to suitable sink tokens whereupon signals can be post-analysed statistically, using frequency domain analysis tools and so on. Figure 7 (discussed later) presents a screenshot of a SystemView simulation with a number of interconnected tokens.

3. THE ADAPTIVE LIBRARY

In this section we will summarize the algorithm capability of the adaptive library, briefly review the philosophy behind implementation, and thereafter present implementations of the library.

3.1 The LMS Algorithm

The most general tokens in the adaptive library are the Least Mean Squares (LMS) tokens. These tokens allow straightforward implementation of the LMS and its many variants.

For the generic FIR LMS architecture [2] shown in Figure 2, the general filter weight update equation is given by:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu(-\hat{\nabla}_k) \tag{1}$$

where $\mathbf{w}(k) = [w_0, w_1, \ldots, w_{N-1}]_k^T$

The LMS gradient estimate [3] is well known to be:

$$\hat{\nabla}_k = \frac{de^2(k)}{d\mathbf{w}(k)} = 2e(k)\left[\frac{de(k)}{d\mathbf{w}(k)}\right] = -2e(k)\left[\frac{dy(k)}{d\mathbf{w}(k)}\right] = -\mathbf{x}(k) \tag{2}$$

and therefore the LMS algorithm is:

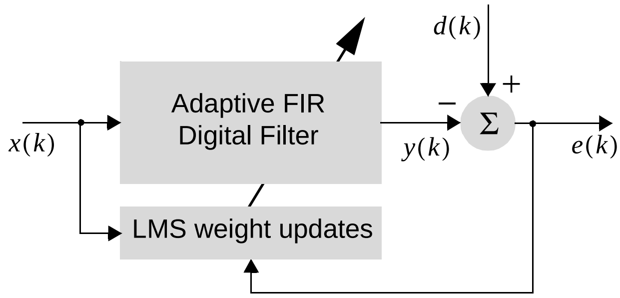
$$\mathbf{w}(k+1) = \mathbf{w}(k) + 2\mu e(k)\mathbf{x}(k) \tag{3}$$

where $\mathbf{x}(k) = [x(k), x(k-1), \ldots, x(k-N+1)]^T$. The LMS algorithm and its many variants are of course used in literally hundreds of applications ranging from simple linear prediction for speech coding to complex arithmetic decision feedback equalisers (DFE) for quadrature channel modulation.

3.2 The General LMS Adaptive Token

The general LMS (titled "GenLMS") adaptive token of the library implements:

$$\mathbf{w}(k+1) = c\mathbf{w}(k) + 2\mu e(k)\mathbf{g}(k) \tag{4}$$



$$y(k) = \mathbf{w}^T(k)\mathbf{x}(k) \qquad \mathbf{w}(k+1) = \mathbf{w}(k) + \mu(-\nabla_k)$$

Figure 2: The Generic LMS processor

Figure 4 illustrates the internal signal flow graph connections for one of the general LMS tokens. $\mathbf{g}(k)$ is an N element vector, i.e. the same dimension as the no. of weights in the filter. (The constant c is the leakage factor [2].) In order to implement the standard LMS of Eq. 3 we simply set $\mathbf{g}(k) = \mathbf{x}(k)$ and calculate $e(k)$ as shown in Figure 3 which shows the token interconnects of SystemView required to produce the generic LMS of Figure 2.

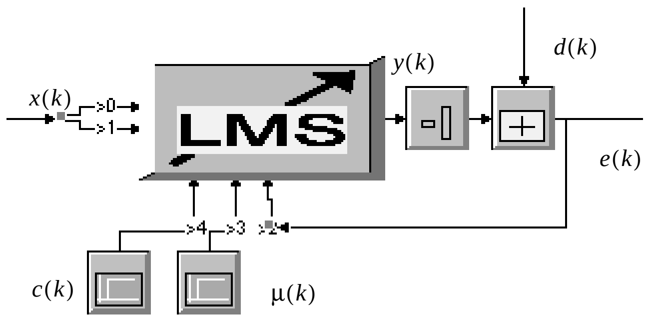
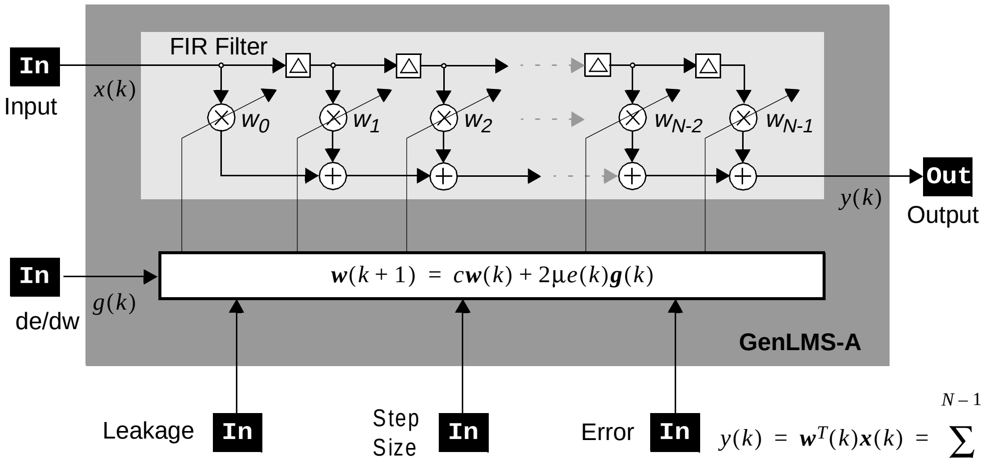


Figure 3: General LMS SystemView implementation.



$$y(k) = \boldsymbol{w}^T(k)\boldsymbol{x}(k) = \sum_{n=0}^{N-1} w_n(k)x(k-n)$$

Figure 4: The general LMS, GenLMS-A token from the adaptive library.

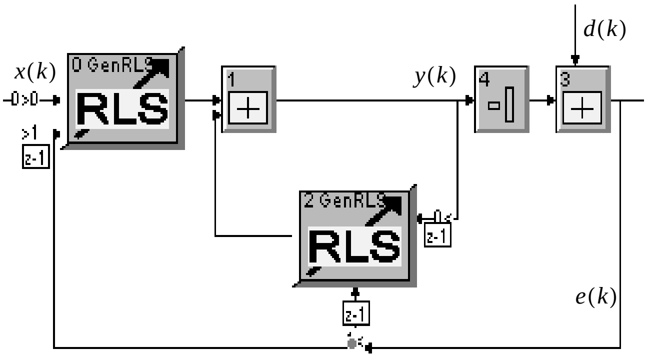


Figure 5: Using two RLS tokens to create an IIR-RLS.

Using this general LMS token, we can easily setup LMS variants such as the sign error (by inserting a slicer before feedback into the LMS, filtered-error, delay LMS, variable step size (by setting up a variable parameter step size source token), filtered-X LMS and so on. We can also interconnect two tokens to produce IIR systems, or multiple LMS tokens to produce multichannel systems.

3.3 Other Adaptive Library Tokens

In addition to the general tokens for the LMS we also provide general tokens to implement the RLS, Newton and Kalman algorithms [2]. Full support is also provided for multichannel and complex LMS. Once again complete parameter and input signal control is provided. Therefore if a user so desires, algorithms such as a (bespoke!) *filtered-X multichannel RLS* can be quickly interconnected and implemented and compared to a similar LMS setup [5]. Figure 5 illustrates how two RLS tokens can be interconnected to produce a (pseudo-linear regression) PLR RLS algorithm [7]. Or for example to, implement and adaptive blind DFE equaliser using Bussgang techniques, we can simply insert a suitable non-linear element in the decision feedback loop.

4. ADAPTIVE APPLICATIONS

In Table 1 we list a number of applications that can be simulated using SystemView and the AdaptLib library. Demonstration versions of these files can be downloaded from `http://www.entegra.co.uk/icassp99.htm`.

As an example of rapid simuation development, in Figure 7 we present a screenshot of an adaptive multiuser receiver for CDMA implemented with an LMS token from the library. One of the major drawbacks of the traditional DS-CDMA receiver is the so-called near-far problem. A weak signal can be overwhelmed by a stronger one coming from a nearby interferer. Using the adaptive library, we can set up an LMS receiver with an appropriately downsampled impulse response [8]. This receiver is basically an adaptive filter that goes through a training period in which the filter adapts in a mean square error sense to a training sequence which is transmitted by the user of interest and also used as desired signal in the adaptive filter. Once adaption has been achieved, the training sequence is switched off and the filter starts operating in a decision directed mode, much in a way of an adaptive equalizer, decoding the received data. The library can then be conviently reviewed in order to evaluate other algorithms DS-CDMA, such as QR, RLS, Newton, or simply varying parameters of the LMS.

5. CONCLUSIONS

This paper has introduced a versatile adaptive library for DSP and communications system simulation and algorithm prototyping. The library runs within the professional DSP simulation tool SystemView and allows virtually any linear and non-linear adaptive applications and algorithm to be set conveniently set up and simulated. To view the capability and relevance of the library we would suggest that users download the library, the examples and associated PDF manuals from `http://www.entegra.co.uk/icassp99.htm`. Examples of all of the applications named in Table 1 are included in the download.

REFERENCES

[1] SystemView by Elanix. http://www.elanix.com.

[2] S. Haykin. *Adaptive Filter Theory*, 3rd Edition. Prentice Hall, 1996.

[3] B. Widrow and S. D. Stearns. *Adaptive Signal Processing.* Prentice Hall, 1995.

[4] P. M. Clarkson. *Optimal and Adaptive Signal Processing*. CRC Press, 1993.

[5] S. J. Elliot, C. C. Boucher and P. A. Nelson. The behaviour of a multiple channel active control system. *IEEE Transactions on Signal Processing*, Vol. 40, No. 5, pp. 1041-1052, . May 1992.

[6] M. Moonen and I. Proudler. *The Adaptive Filter Cookbook*. http://www.esat.kuleuven.ac.be/~moonen/asp_course.html

[7] P. A. Regalia. *Adaptive IIR Filtering in Signal Processing and Control*. Dekker, New York, 1994.

[8] G. Woodward, B.S. Vucetic. Adaptive Detection for DS-CDMA. Proceedings of the IEEE, vol. 86, No. 7, pp. 1413-1434, July 1998.

System Identification	Inv Sys Identification	Prediction	Noise Cancellation
• Telephone channel ident.	• Channel equalisation	• CDMA Inteference Sup	• Active noise control
• Modem Echo Canc.	• Decision Directed Equ.	• LPC Speech Coding	• Background noise
• Acoustic echo control	• Blind Equalisation	• Period Noise Suppress	• ECG noise control
• Radio channel ident.	• Decision Feedback	• Spectral Whitening	• Acoustic beamforming
• Room acoustic ident.	• Fractional Equalisers	• Adaptive CDMA receiver	• Multimedia noise control

Figure 6: Applications simulated with the SystemView adaptive library.

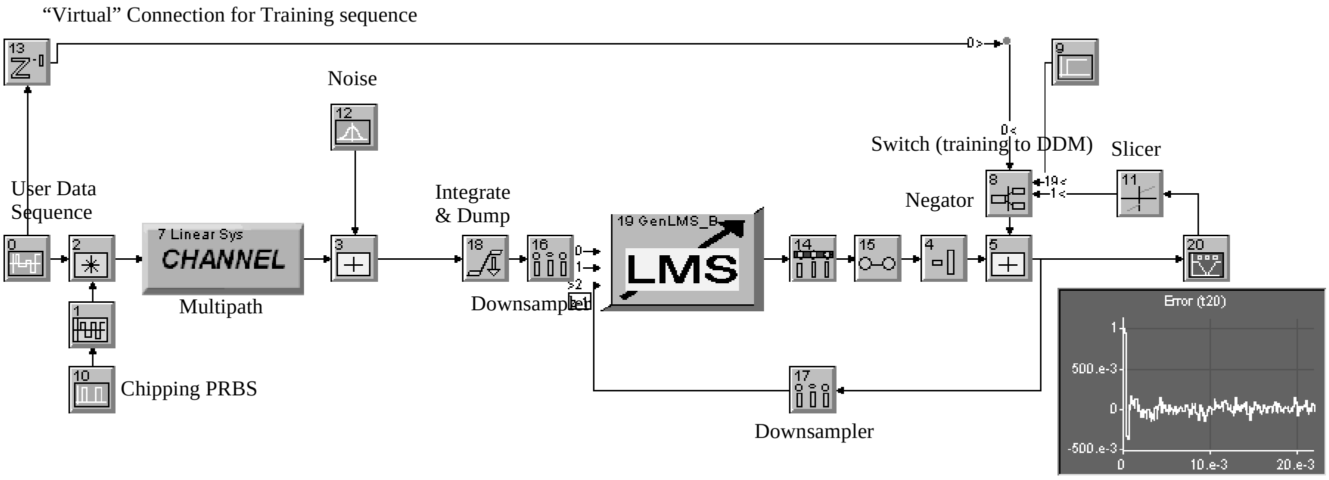


Figure 7: Adaptive multiuser receiver for single user CDMA.