

Interactive teaching of adaptive signal processing

by Robert W. Stewart, Moritz Harteneck and Stephan Weiss

Over the last 30 years adaptive digital signal processing has progressed from being a topic for a strictly graduate-level advanced class in signal processing theory to one that is part of the core curriculum for many undergraduate classes in signal processing. The key reason is the continued advance of communications technology, with its need for echo control and equalisation, and the widespread use of adaptive filters in audio, biomedical, and control applications. This paper reviews the basic theory and applications of adaptive signal processing and also presents information on multimedia teaching methods that are being used on a 24 hour course module on adaptive signal processing. In particular it presents some recent World Wide Web JAVA programmes that were developed for teaching and self-study. Information is provided on how to download and run the JAVA applet.

In this paper we review the key theory and applications of adaptive signal processing. We then focus on some Internet-based simulation learning tools that we have developed recently based on our experience of teaching a Master's level module on adaptive signal processing over the last 8 years. This development follows from our previous experience with computer-based teaching of adaptive signal processing using a custom windows package¹ and from our knowledge of the pragmatic issues of teaching digital signal processing (DSP) to practising engineers².

Recently we have developed a custom suite of adaptive signal processing algorithms that was written in JAVA to run from the World Wide Web (WWW). The result is an adaptive filtering learning tool that has proven to be extremely effective in presenting basic and advanced adaptive signal processing concepts. The key aim of the software is to bridge the gap between the theory and mathematics of textbooks and the practical application and implementations of adaptive DSP. Students are able to run the JAVA program from anywhere with Internet access, such as from university workstations or PCs, or even from home using a modem and PC connected to the WWW. The traditional computer laboratory problems of machine availability, software licensing, portability and so on are clearly circumvented using this approach. The JAVA applet has been used for three years at the University of Strathclyde to teach the Adaptive Signal Processing Master's class. As an example of its portability, in the

summers of 1997, 1998 and 1999, the JAVA adaptive suite was successfully used on a course taught by Strathclyde academic staff using a computer laboratory at the University of Los Angeles, UCLA over the Internet³.

The JAVA applet is freely available on the WWW at http://www.spd.eee.strath.ac.uk/~bob/adaptivejava/java_adaptive_dsp/index.htm and accompanies the version of this paper that is being published in the library section of the IEE's *Computer Forum* at <http://forum.iee.org.uk>.*

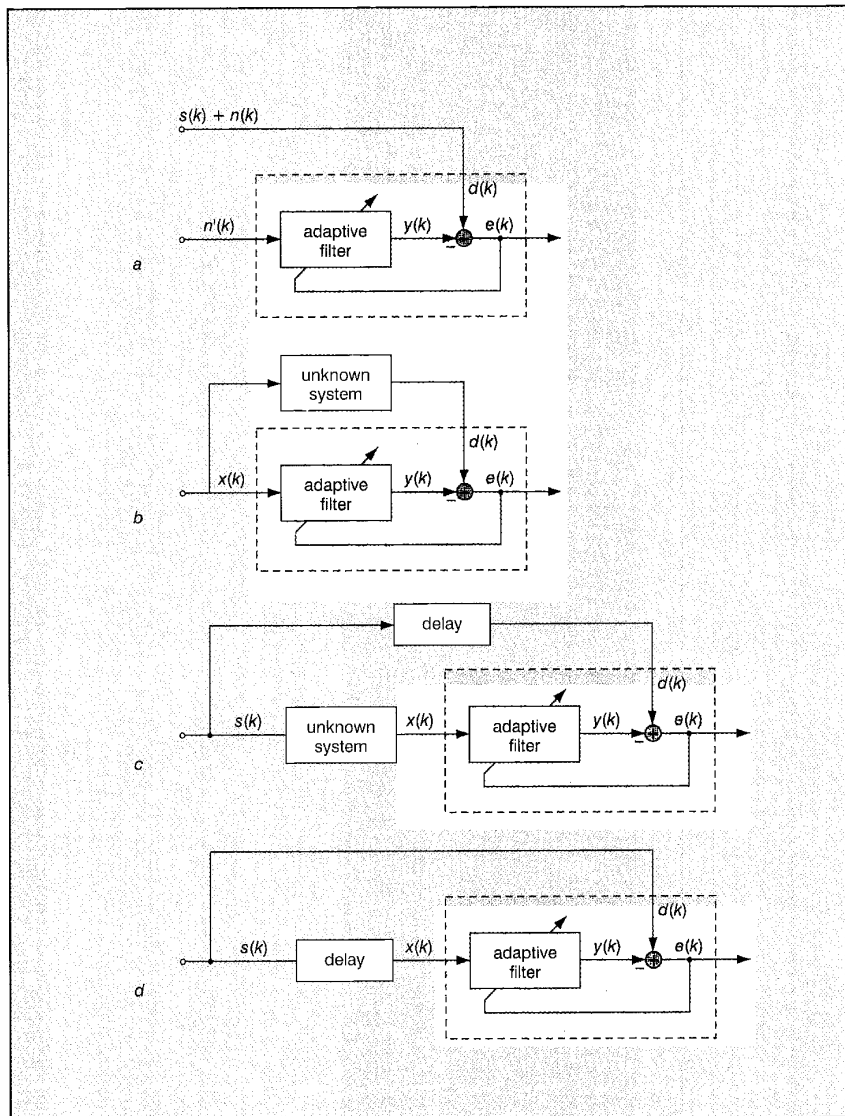
Adaptive signal processing course

Applications of adaptive signal processing

Adaptive signal processing is one of the most important classes of algorithms for modern communication systems. Telephone line modems, for example, now communicate at rates of 56 kbaud and above as a result of the integration of adaptive echo cancellers and adaptive equalisation algorithms. Similarly, the new generation of mobile multimedia systems and set-top boxes will also require the use of adaptive DSP, as will adaptive acoustic echo cancellation, arguably the next key 'plug-in card' for PCs. Adaptive active noise cancellation is another hi-tech and mature technology found in the cabins of some airliners to reduce the level

*Users must register to see content within the *Computer Forum*. Registration is free.

Fig. 1 General adaptive filter applications:
(a) noise cancellation;
(b) system identification;
(c) inverse system identification;
(d) prediction



of noise. More generally, adaptive DSP can be found in biomedical systems, telecommunication systems, industrial control etc.

Fig. 1 shows the general architectures for the key application areas of noise cancellation, system identification, inverse system identification and prediction. Note the common element in these structures is the general adaptive signal processor, as depicted in Fig. 2, with input signal $x(k)$, output signal $y(k)$, desired signal $d(k)$ and error signal $e(k)$.

The structure shown in Fig. 1a would be employed if a signal $s(k)$ and a corrupting noise $n(k)$ had to be separated, and exploits the presence of a reference of the noise signal $n'(k)$. The system identification structure in Fig. 1b can be applied to a number of control problems or to acoustic echo cancellation, where the unknown system would be the transfer function of the teleconferencing room. The set-up given in Fig. 1c is a typical inverse system identification

scenario, which is used, for example, in the problem of equalising telephone lines, where the unknown system is the transfer function of the telephone channel and the adaptive filter should reduce intersymbol interference and other distortions as much as possible. Finally, Fig. 1d depicts the adaptive filter in a predictor set-up. There, the filter tries to predict a sample by using a set of past observations. This set-up is commonly used in coders to reduce the redundancy of a data stream, and thereby increase the coding efficiency. Another widespread example of the application of a predictor is adaptive line enhancement, in which a periodic interference, emanating for example from the mains, can be removed from a signal.

Course outline

The aim of the Adaptive Signal Processing course is to present algorithms and architectures and explain their use in real-world applications. As prerequisites

it is assumed that students have studied discrete and continuous signals and systems, and introductory linear algebra. The syllabus of the course can be summarised as follows:

- Introduction to Adaptive Filtering: a historical perspective; a state of the art perspective
- Statistical Signal Processing Revision: correlation; ergodicity; means, variances; stationarity; wide sense stationarity; periodogramme; frequency response vs. power spectrum
- Matrix Algebra Revision: addition, multiplication and matrix inverses; properties of the correlation/covariance matrix; eigenvalues and eigenvectors; QR algorithm
- Wiener Filter Theory: normal equations; error performance surfaces; orthogonality; minimum mean square errors
- Least Mean Squares (LMS) Algorithm: formulation; convergence; stability criteria. Algorithm variations: normalised algorithm, sign error algorithm, sign data algorithm, leaky algorithm, filtered-X algorithm, variable step-size algorithm
- Applications of the LMS: system identification; room acoustics, control systems; inverse system modelling; modems, telecommunications adaptive equalisation, echo cancellation; adaptive beam-forming (radar, sonar, hearing aids, listening devices); active noise cancellation systems in cars, airplanes, medical systems, communication systems
- Recursive LMS-IIR Algorithms: output error formulation; equation error formulation; full gradient, simplified gradient, SHARE, Feintuch's

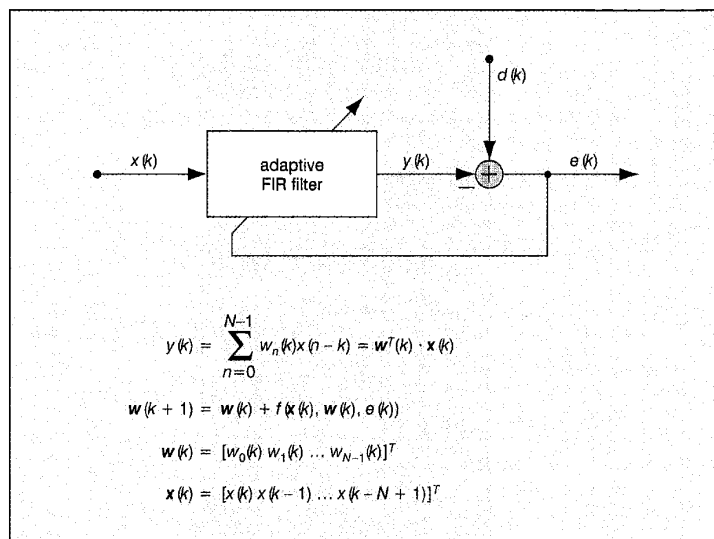


Fig. 2 General adaptive signal processor

- algorithm; applications of recursive LMS algorithms
- Frequency Domain Adaptive LMS: architectures, advantages, and disadvantages
- General Least Squares Solution: least squares solution of general adaptive system; QR algorithm solution
- Recursive Least Squares (RLS) algorithm: RLS formulation; forgetting factors; practical implementations; QR based RLS; numerical stability and integrity issues
- Adaptive Lattice Filters: gradient lattice, RLS lattice
- Non-linear Adaptive Filters: simple LMS neuron, adaptive polynomial filters (Volterra)
- Comparative Analysis: Wiener; LMS-FIR, LMS-IIR; RLS, lattice; frequency domain and neural networks for adaptive signal processing applications
- Implementation of Adaptive Filters: DSP microprocessor implementation; software; custom hardware

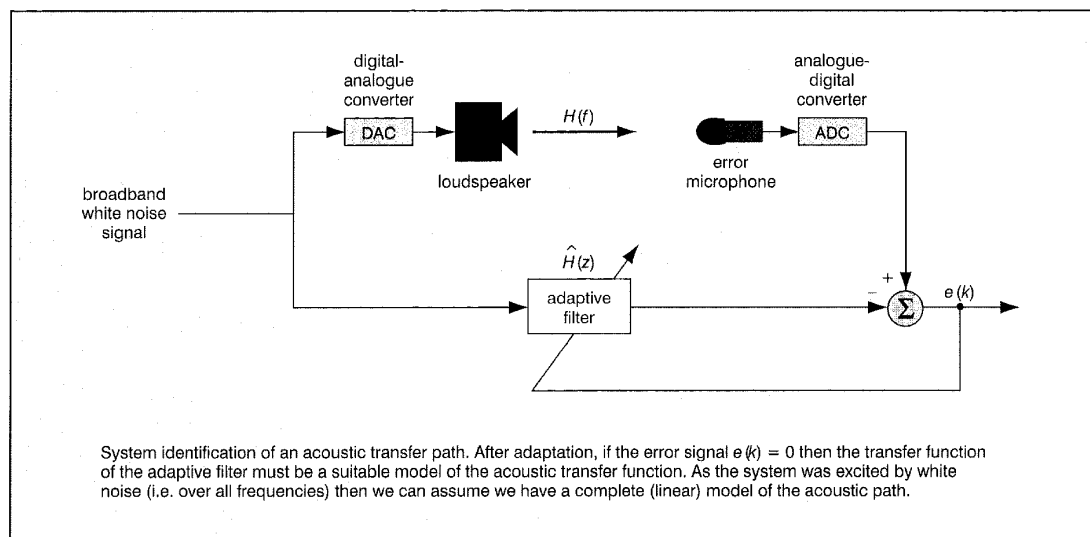


Fig. 3 Adaptive system identification of room acoustics

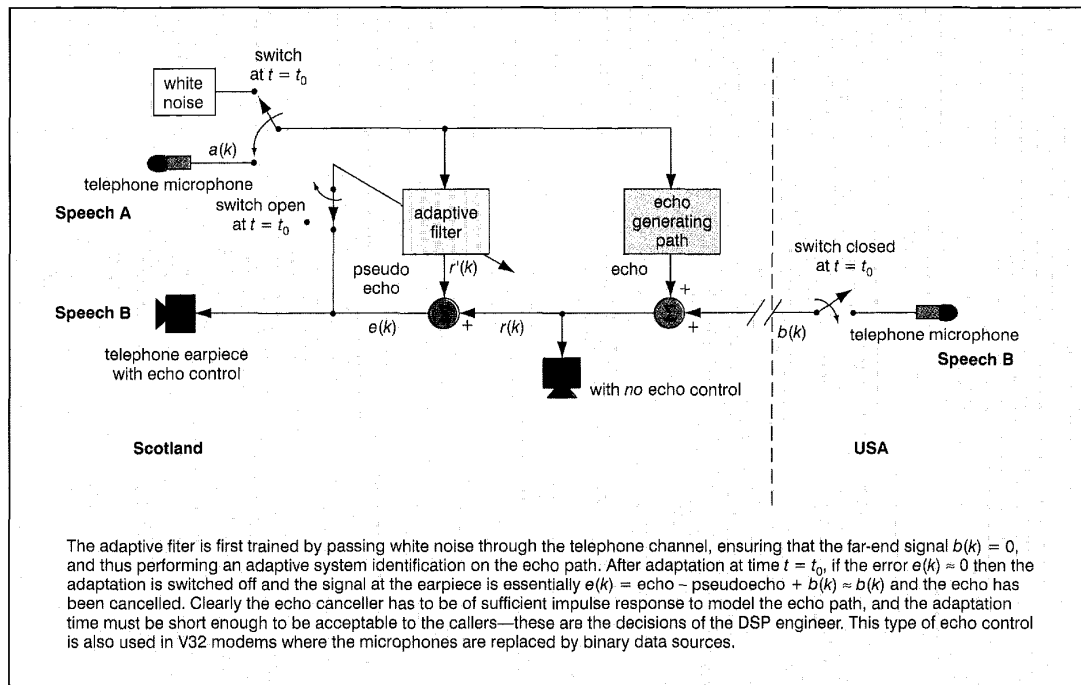


Fig. 4 Adaptive echo cancellation

- Application Examples: examples of the use and performance of adaptive filters are given and demonstrated by audio demonstrations.

The above list outlines the various adaptive algorithms that are presented on the course, and Fig. 1 the different applications considered.

Students are referred to well known texts⁴⁻⁷ for further background in adaptive filter techniques, however none of the texts are compulsory purchases.

Course presentation and chronology

Students are first introduced to the general adaptive model shown in Fig. 2. The standard aim of an adaptive signal processing algorithm is to minimise the power of the error signal $e(k)$. This is done by adapting the coefficients of a filter excited by the input signal $x(k)$ such that the filter output $y(k)$ is as similar as possible to some desired signal $d(k)$. It is straightforward to show students that the mathematically most tractable way forward is to minimise the squared error or the mean squared error. From this model, the main adaptive filtering algorithms described above can be derived and applied to the four main (single channel) adaptive applications in Fig. 1. These applications are then linked and applied to a number of real-world adaptive signal processing problems, for which a few examples are briefly overviewed in Figs. 3–8.

During the teaching of this course, although many students appeared to understand the mathematics of adaptive signal processing, their awareness of how and when to use the various adaptive algorithms was not well developed. When the students later undertook an

adaptive DSP project, the problems they had highlighted their lack of practical and real-world experience. For example, referring back to Figs. 3–8, although the architectures are all quite intuitive, questions such as the following arise in almost every case:

- How important is the length of the adaptive filter?
- What value should the step size be?
- When do I use LMS-FIR, LMS-IIR or RLS?
- What is the MMSE value?
- What adapts faster, an LMS, RLS or IIR LMS? How can I see this?
- Which algorithm can track faster changes of the unknown system?
- What happens when a filter goes unstable?
- Does the error really go to zero?

While a single sentence and mathematical answer is always possible for a specific situation, it is clear that the learning process would be greatly enhanced by a software tool that would allow most questions to be answered by the student within just a few minutes of reasoned simulation. Hence the requirement for a software simulation tool is clear. Unfortunately, no such learning tools were readily available.

JAVA applet

To demonstrate the way adaptive filters perform and to give students an insight into how the different parameters affect the convergence and steady-state behaviour of the algorithms, a JAVA applet has been

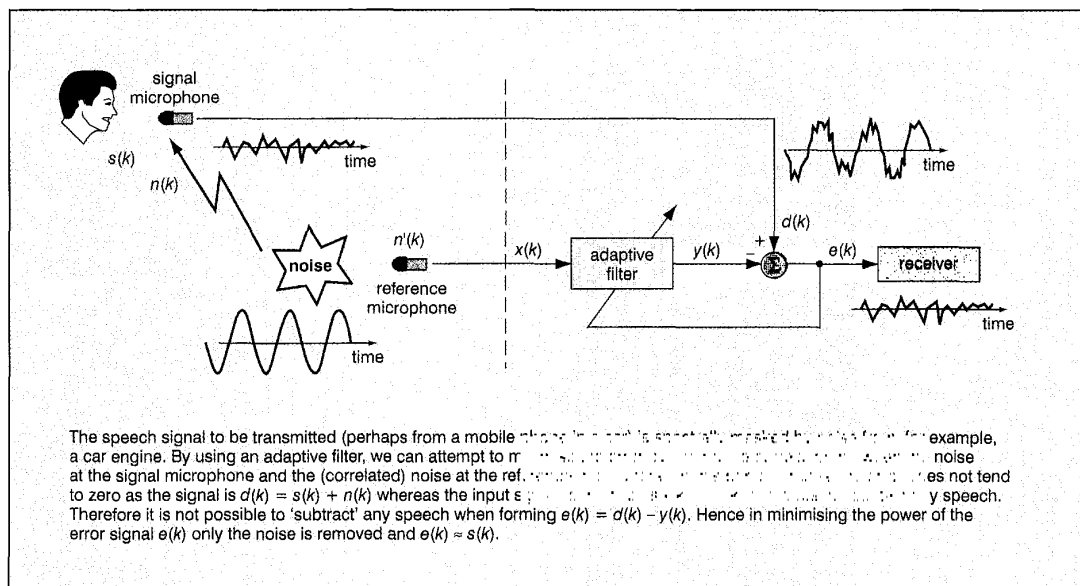


Fig. 5 Adaptive noise cancellation

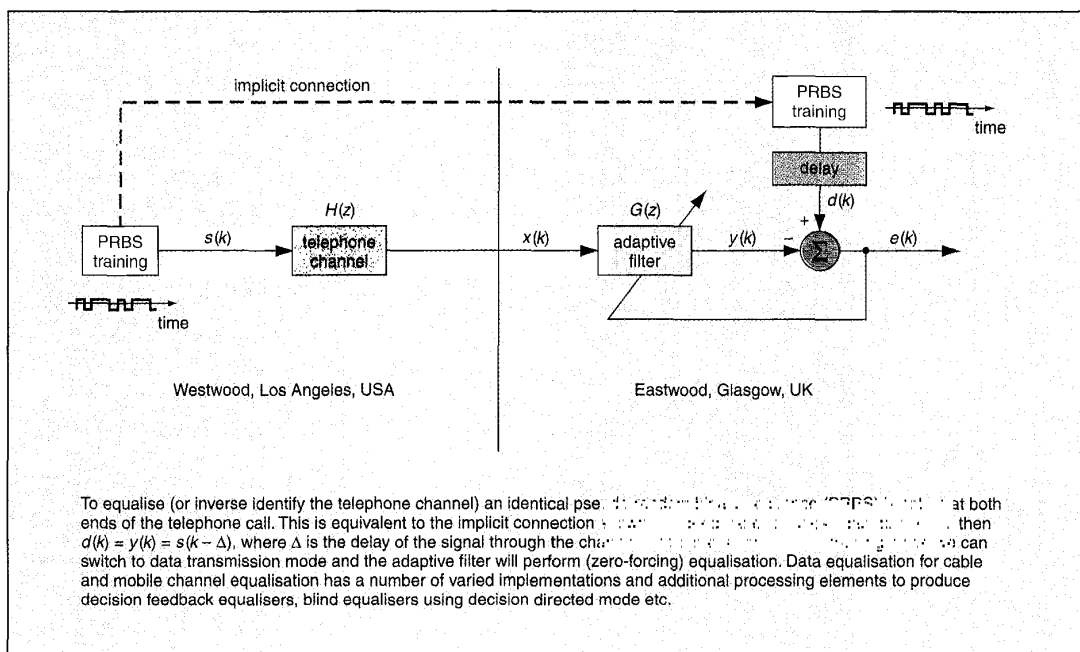


Fig. 6 Adaptive inverse system identification/equalisation

developed which compares two adaptive algorithms for FIR (finite impulse response) filters in a system identification set-up, cf. Fig. 1b. The JAVA environment was chosen for this task as it is platform independent, i.e. the applet can be used on every computer system where a JAVA-capable browser is available. Furthermore, it can be downloaded easily via the Internet so that computing facilities at the university, at remote sites or even at home can be used for teaching purposes. In our teaching, we have also stored the applet on portable PCs and presented it directly in class for demonstrative purposes.

The internal set-up of the demonstration and teaching applet is shown in Fig. 9. It consists of the unknown system, which is represented by a 10 tap FIR system, and two adaptive algorithms, operating in a parallel system identification set-up, so that comparisons of different algorithms and the effects of the parameters can be studied easily. The power of the input signal and the power of the observation noise are controlled by the two scaling factors: *input power level* and *observation noise power*. The signals used for the input signal and the observation noise signal are two independent Gaussian white noise pseudo-random

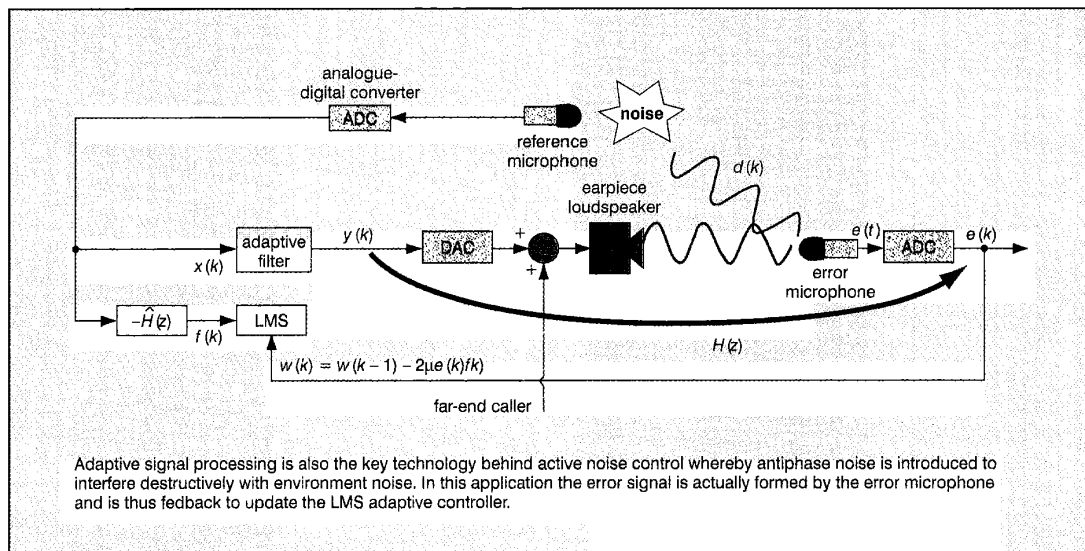


Fig. 7 Adaptive active noise cancellation

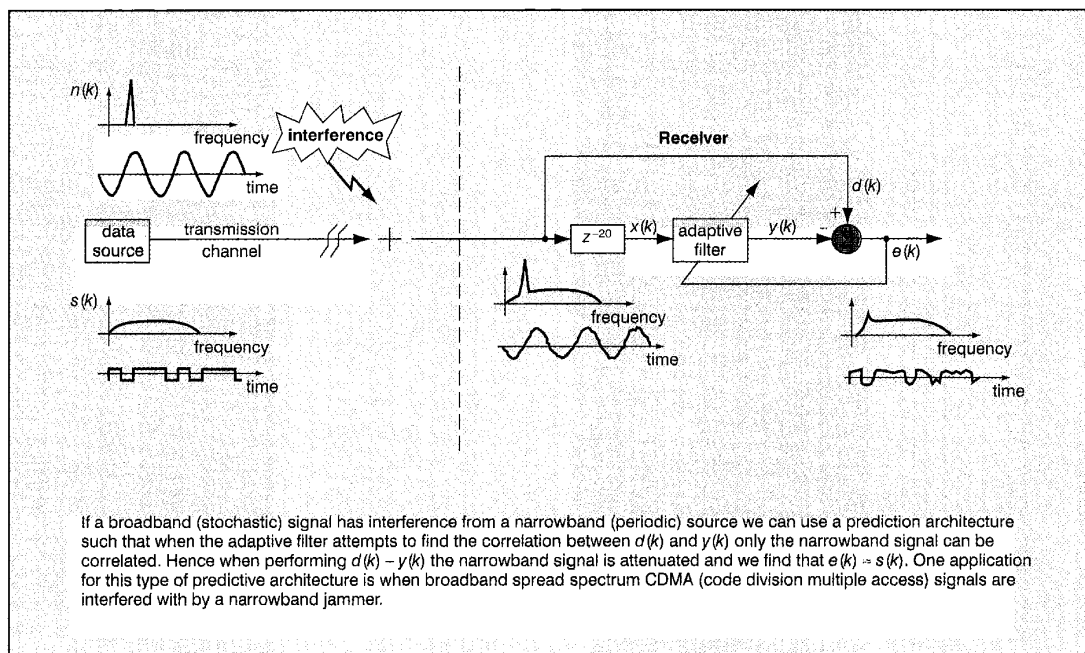


Fig. 8 Adaptive prediction/line enhancer

sequences. Adaptive algorithms belonging to the class of finite impulse response (FIR) least mean squares (LMS) and least squares (LS) algorithms have been implemented in various versions. The versions include the standard least mean squares algorithm⁵, the normalised least mean squares (NLMS) algorithm⁵, the variable step size least mean squares (VSLMS) algorithm⁸ and a variable step size normalised least mean squares (VSNLMS) algorithm, as well as the recursive least squares (RLS) algorithm⁴ and the recursive least squares algorithm solved by a QR decomposition (QR-RLS)⁴.

When reaching the corresponding page in the

Internet, a button to start the applet and a set of instructions about how to use the applet appears. After pressing the button, a window, as shown in Fig. 10, is created which is the core element of the applet. The window is partitioned into three parts.

At the top, the impulse response of the unknown system is shown together with controls affecting the environment of the adaptive filters, i.e. the unknown system and the scaling factors for the input signals. The parameter *Scale* is a scaling factor for the error signals so that the final convergence can be observed more easily even if the error signals are already very small. The unknown system can be changed either by

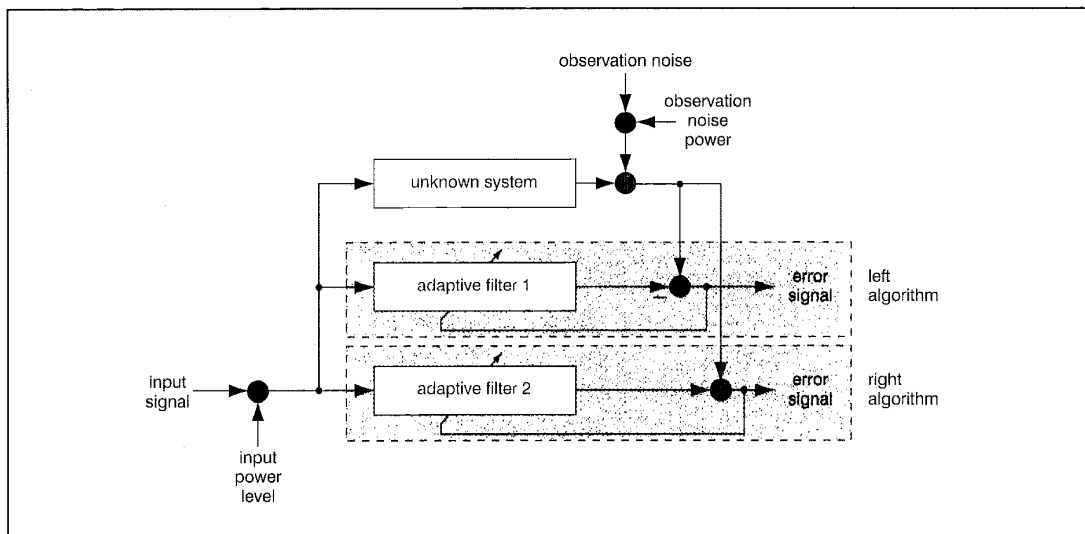


Fig. 9 System identification set-up

clicking directly on the plot showing the weights so as to adjust single weights or by pressing the INVERT button, which multiplies every weight of the unknown system by -1 . The RESET button in the control area sets both adaptive algorithms into a predefined state, i.e. all delay lines are initialized to zero and the weights are set to zero. On the right hand side of the control panel, two choices are given where the student can select the adaptive algorithm to be used, i.e. select among LMS, NLMS, VSLMS, VSNLMS, RLS and QR-RLS.

At the bottom of the window, two adaptive algorithms operate side by side. The algorithms are selected in the control section and, for each algorithm, the time-varying adaptive weights and the *a posteriori*

error signals are shown. The controls setting the number of taps, the step sizes and the forgetting factors differ according to the algorithm chosen in the control section. For each algorithm, a RESET button exists which puts the particular algorithm into a defined state. This might be necessary, for example, after the algorithm has gone unstable.

Using this applet, the initial convergence behaviour, the behaviour after a step change of the unknown system, and various effects, such as a single weight change of the unknown system and the results on the adaptive algorithms, can be observed and studied easily. Furthermore, the behaviour of the algorithms with additional observation noise and in over- and under-

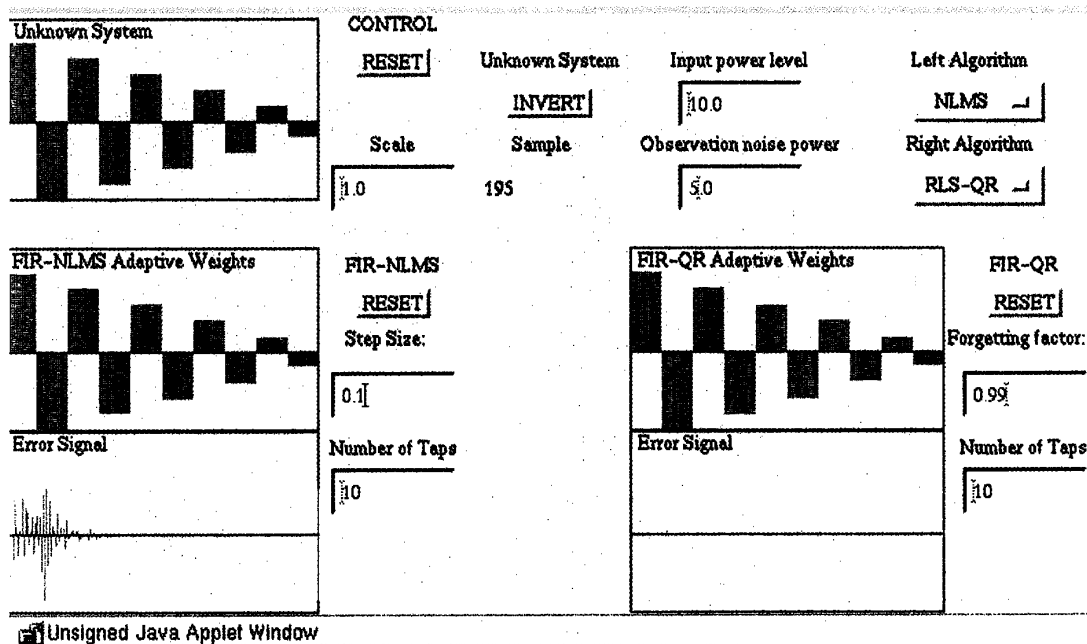
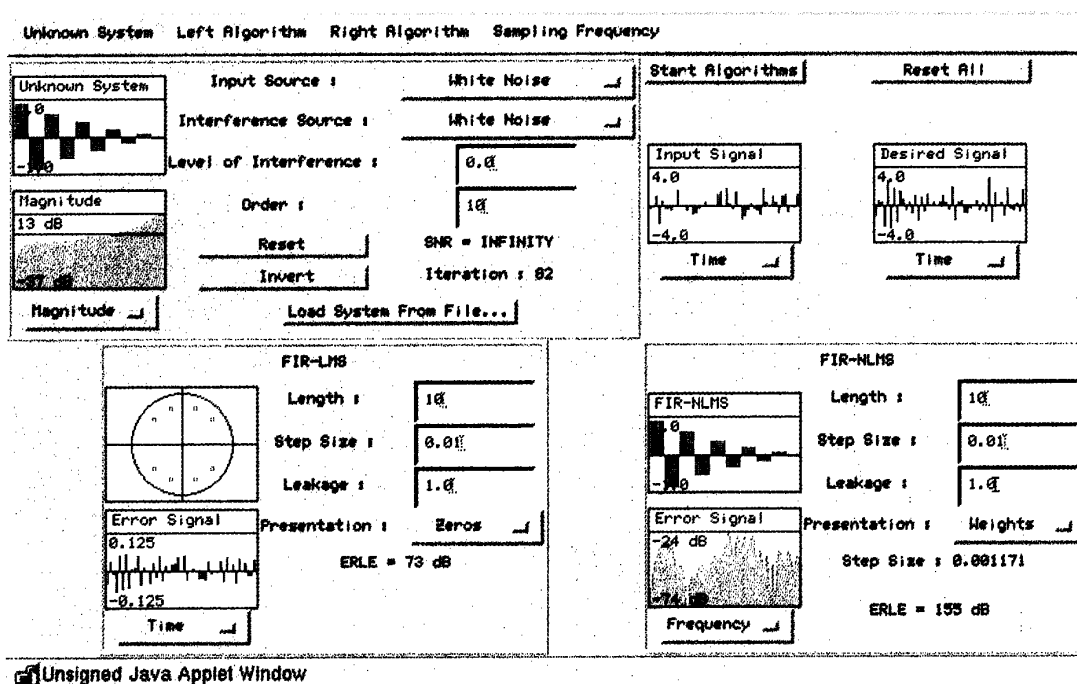


Fig. 10 Core JAVA window



Unsigned Java Applet Window

Fig. 11 Advanced JAVA applet

modelled situations can be explored by the student.

Conclusion and current work

In this paper we have reported on an adaptive filtering class that is taught every year as part of the Communications, Control and Digital Signal Processing Master of Science course at the University of Strathclyde. While teaching the course it has been noted that the understanding of the students could be greatly enhanced by using simulations and interactive presentation techniques.

Therefore a JAVA applet has been developed that demonstrates the internal working of adaptive algorithms and is now used in the class and in external courses. However, for an advanced student this applet can be too rigid to perform simulations in a more complex set-up. Therefore, our current work is extending our ideas towards a more advanced applet which can be viewed and downloaded from the WWW. Fig. 11 shows the interface to the advanced applet, which also incorporates frequency domain observations and accepts the use of signals submitted by the student. Any comments on the applets are welcome.

Acknowledgments

The authors gratefully acknowledge the input of Professor J. McWhirter and Dr. I. Proudler of DERA, Malvern to this paper and the support of Dr. M. Harteneck through contract CSM/140. We also acknowledge the input from D. Garcia-Alis, E. Pfann, Dr. D. Crawford and Dr. E. Roy.

References

- 1 STEWART, R. W.: 'Practical DSP for scientists and engineers'. Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Minneapolis, USA, April 1993, 1, pp.32-35
- 2 STEWART, R. W., and AASERUD, P.: 'Adaptlab: an adaptive signal processing environment for teaching and development'. Proc. 28th Asilomar Conf. on Signals, Systems and Computers, Monterey, USA, November 1994, 2, pp.1320-1324
- 3 STEWART, R. W.: 'Digital signal processing: algorithms and applications'. UCLA Extension Short Course, University of California at Los Angeles, Extension Program. See <http://www.unex.ucla.edu>
- 4 HAYKIN, S.: 'Adaptive filter theory' (Prentice Hall, 1991, 2nd edn.), ISBN 0 13 012236 5
- 5 WIDROW, B., and STEARNS, S. D.: 'Adaptive signal processing' (Prentice Hall, Englewood Cliffs, 1985)
- 6 ALEXANDER, S. T.: 'Adaptive signal processing, theory and applications' (Springer-Verlag, 1986), ISBN 3 540 96480 4
- 7 SHYMK, J. J.: 'Adaptive IIR filtering', *IEEE Acoustics, Speech, Signal Process. Mag.*, April 1989, 6, (4), pp.4-21
- 8 MATHEWS, V. J., and XIE, Z.: 'A stochastic gradient adaptive filter with gradient adaptive step size', *IEEE Trans. Signal Process.*, June 1993, 41, (6), pp.2075-2087

© IEE: 2000

Dr. Stewart and Dr Harteneck are with the Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow G1 1XW, UK.

Dr. Weiss was formerly at the University of Strathclyde and is now with the Department of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK. E-mail: s.weiss@ecs.soton.ac.uk