HARDWARE-BASED ADAPTIVE GENERAL PARAMETER EXTENSION IN WCDMA POWER CONTROL

Matti T. Tommiska[1*], *Jarno M. A. Tanskanen*[2], *Jorma O. Skyttä*[1]
[1] Signal Processing Laboratory, Helsinki University of Technology
P.O.Box 3000, FIN-02015 HUT, FINLAND
[*] Tel: +358 9 451 2477; fax: +358 9 460 224
[*] e-mail: matti.tommiska@hut.fi
[2] Institute of Intelligent Power Electronics, Helsinki University of Technology
P.O.Box 3000, FIN-02015 HUT, FINLAND

ABSTRACT

An adaptive general parameter (GP) extension to polynomial FIR predictors is proposed for Rayleigh-distributed fading signal prediction. The GP-extended FIR predictor has been implemented on Altera's programmable logic devices, and its performance has been simulated with varying bit widths. The hardware implementation takes advantage of the characteristic properties of programmable logic devices. This results in decreasing the required number of logic gates and in increasing the maximum sampling rate of the predictor. The proposed adaptation can be regarded as the simplest form of adaptive filtering, and is shown to yield improved filtering performance at a low computational cost. A promising application is to use the GP-extended FIR filter in predicting the received transmission power in a WCDMA mobile communications system.

1 INTRODUCTION

Generally, signal processing methods assume an input signal distribution for which they are optimized. In the real world, however, statistics of measured signals may either not be presentable by simple distributions, or at least determining the current distribution requires additional computational resources. Adaptive filtering and soft computing methods, e.g., neural networks, address this problem of processing measured signals with unknown and/or varying signal characteristics. Many signal processing and control applications rely on identified signal statistics, or are forced to employ computationally costly adaptation schemes in order to function properly.

The application example in this paper is transmitter power control of a mobile communication system [1] . In the mobile power control, the employed closed loop control system operation is inherently delay-limited and would benefit from an accurate prediction of the attenuation, or fading, present in the radio channel. Under certain conditions, radio channel fading can be modeled with Rayleigh distribution [2][3], which can be modeled as a piecewise polynomial function. Also, the fading characteristics are greatly time varying while also the frequency content of the received signal varies with the mobile speed due to the Doppler effect. Thus an adaptive polynomial-predictive filtering scheme is a promising approach to employ to decrease the effects of the closed loop control delays while also attenuating noise. Also, it would naturally be beneficial if the adaptation were computationally as inexpensive as possible.

In this paper, a simple adaptive filtering method is attached to an ordinary fixed-coefficient FIR filter. Here polynomial-predictive [4] FIR filters are used as the exemplary basis filter. The adaptation that is attached to a FIR filter consists of a single adaptive general parameter (GP) [5] that is added to each filter coefficient. The GP extension calculation is illustrated in Fig. 1

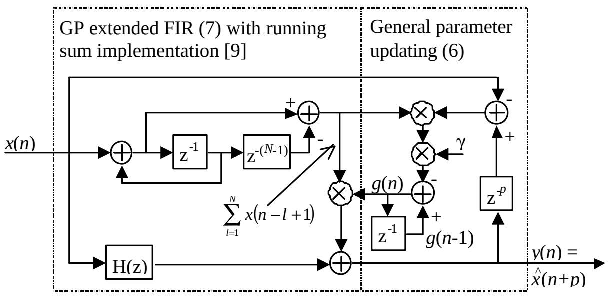

Fig. 1. Structure of the GP-extended FIR with the GP update calculation shown. Notation: input signal $x(n)$, output signal $y(n)$, transfer function of the FIR H(z) providing for the prediction step of p samples with the coefficient vector $\mathbf{h}$ of length N, general parameter $g(n)$, gain factor γ, and hat denoting an estimate.

0-7803-7005-8/01/$10.00 (c) 2001 IEEE

2 BASIS FILTER FOR GENERAL PARAMETER EXTENSION

The types of predictive basis filter employed is a polynomial FIR predictor [4] whose filtering theory is well established.

The output of an ideally operating p-step-ahead polynomial FIR predictor is calculated by [4]

$$\sum_{k=1}^{N} h(k)x(n-k+1) = x(n+p) \qquad (1)$$

where $h(k)$ are filter coefficients, and $x(n)$ are input signal samples at discrete time instants n. The FIR predictors are designed to yield exact p-step-ahead prediction of a given polynomial degree, or of a given sinusoidal frequency, and thereafter to minimize the white noise gain, given by

$$NG = \sum_{k=1}^{N} |h(n)|^2 \,. \qquad (2)$$

A set of constraints on filter coefficients can be derived from the filter input-output relations, and closed-form solutions for the FIR filter coefficients can be calculated [4].

The predictor applied in this paper is the first degree Heinonen-Neuvo (H-N) polynomial FIR predictor of length $N = 10$, given in [4]. The filter is selected to represent filters that operate on input signals for which they are not optimized.

3 GENERAL PARAMETER EXTENSION TO FIR PREDICTORS

The general parameter $g(n)$ is added as bias to each FIR coefficient in (1). Generally, the GP extended predictive FIR filtering can be formulated as

$$\sum_{k=1}^{N} (g(n) + h(k))x(n-k+1) = x(n+p), \qquad (3)$$

The general parameter (GP) can be taken out of the filtering summation, and presented as a gain operating on the input signal averaged over N samples

$$g(n)\sum_{k=1}^{N} x(n-k+1) + \sum_{k=1}^{N} h(k)x(n-k+1) = x(n+p) \,.(4)$$

Eqs. (3) and (4) assume prefect prediction of the input signal. Let us denote the actual filter output by $y(n)$. The general parameter is adapted to the signal statistics according to a simple updating rule [5]

$$g(n+1) = g(n) - g\,(y(n-p) - x(n))\sum_{l=1}^{N} x(n-l+1) \quad (5)$$

where γ is gain factor. The predictive filter output is now given by

$$g(n)\sum_{k=1}^{N} x(n-k+1) + \sum_{k=1}^{N} h(k)x(n-k+1) = y(n) = \hat{x}(n+p) \qquad (6)$$

where hat denotes an estimate. Calculation of (5) and (6) is illustrated in Fig. 1. This adaptation resembles the ordinary least mean square filter coefficient adaptation but since there is only a single adaptive parameter $g(n)$ per filter, and not adaptation of all the filter coefficients, input signal statistics are taken into account in a form of average over the filtering window.

Since the GP method generally employs basis filters that are not optimized for the input signal statistics, GP does not generally exhibit convergence to any certain value, like adaptive filtering is ordinarily designed to do, but the GP rather fluctuates along with the input signal in order to adjust the filter to the current input signal statistics.

In [6] the stability bound for GP extended FIRs has been derived. Here, let us state the normalized stability boundary without reproducing the derivations. On the stability boundary, the gain factor is given by

$$\gamma = \frac{2}{\mathbf{x}(n)\mathbf{S}}, \qquad (7)$$

where $\mathbf{S} = \left[\sum_{i=1}^{N} x(n-i+1) \quad \cdots \quad \sum_{i=1}^{N} x(n-i+1) \right]^{\mathrm{T}}$ and $\mathbf{x}(n)$ is a vector consisting of N last input samples. Thus the GP update equation on the stability bound is given as

$$g(n+1) = g(n) - \frac{2}{\sum_{i=1}^{N} x(n-i+1)}(y(n-p) - x(n)) \,. \quad (8)$$

From Fig. 1, it is easy to observe that the adaptive GP extension requires only three additional multipliers, 5 additional summations, and $N + p + 1$ delay units. Thus GP extension can be regarded as a low cost system when it comes to implementation.

4 IMPLEMENTATION ON FPGAs

Implementing digital signal processing (DSP) algorithms on field programmable gate arrays (FPGAs) has many advantages compared to processor-based solutions. These include reduced instruction fetching and decoding, the ability to tailor the bus structure to the operation, the easy design of arithmetic units for nonstandard operations, and the usage of parallel and bit-serial architectures at the same time. Also wide data and memory paths are easy to implement due to the large amounts of continuous routing and memory in the form of EABs (Embedded Array Blocks). [7]

The design process of the GP-extended FIR predictor consisted of iterations between Mathworks' Matlab numerical simulation software and Altera's Max+Plus II [8] programmable logic design software. The original GP-extended FIR predictor design files were programmed in Matlab's own programming language using double precision floating point numbers as the numerical presicio. The GP-extended FIR predictor was implemented in fixed-point arithmetic on Altera's programmable logic chips of the FLEX 10KA device family [9]. The bit width of the GP-extended FIR predictor was designed fully parameterizable.

When implementing DSP algorithms with finite precision arithmetic, care must be taken to ensure that the results are compatible with theoretical observations. The analysis of finite wordlength effects is well-established, and is presented compactly in [10].

The usage of two different number systems made it necessary to program custom-made conversion functions and batch files in the C programming language and `awk`. This facilitated both importing initial simulation values from Matlab into Max+Plus II and exporting results from Max+Plus II back into Matlab for graphical inspection. The design process is presented in Fig. 2.

To prevent over- and underflows from occurring, the internal calculation precision was designed larger than the bit widths of the input and output lines. When necessary, internal values are either truncated or sign-extended.

Multiplications and divisions are costly to implement in hardware, since they either require a lot of area or are slow to execute. The GP-extended FIR predictor overcomes some of these difficulties by using multiplications with a constant. These are used in the implementation of the basis FIR filter $H(z)$.

Multiplication with a constant is considerably easier to fit into an area-constrained logic chip than a multiplier with varying inputs [11]. For example, LMS and RLS filters require multipliers whose both inputs vary [12], and thus their implementation uses more logic resources than the GP-extended FIR predictor.

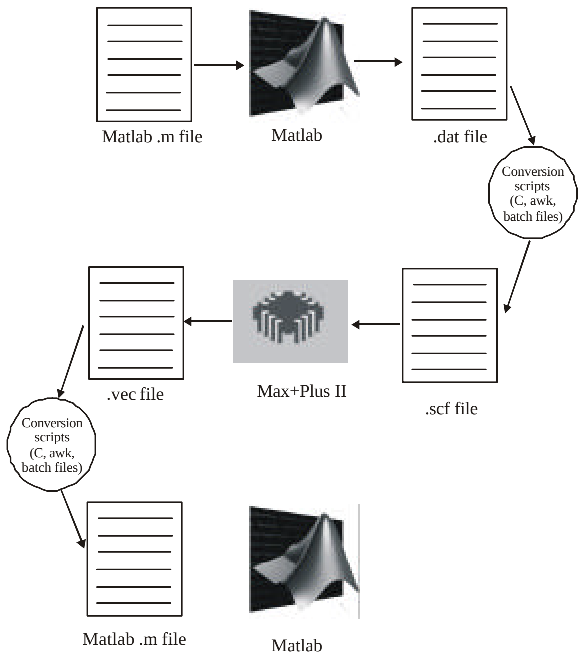

Fig. 2. The design flow consisted of the continous flow of temporary results between two design software packages (Matlab and Max+Plus II) and consequently, of two different numbering systems. To speed up the design process, custom-made conversion scripts were written to manipulate the ASCII text files used as both inputs and outputs of the design sofware packages

Using constant coefficient multipliers is also a special case, where a Field Programmable Gate Array (FPGA) implementation is more flexible than an Application Specific Integrated Circuit (ASIC) implementation. If the coefficients of the basis FIR filter change, all that is needed is a reprogramming of the look-up tables on an FPGA chip. This would be much harder to accomplish on a fixed ASIC platform.

The design of GP-extended FIR predictor was split into two major blocks, the basis FIR predictor and an implementation of Eq. 8. These two blocks required most of the area resources, since the additional blocks are either very compact to design or can be effectively and easily implemented with Altera's LPM (Library of Parameterized Modules) functions. The implementation of Eq. 8 uses an iterative divider, where division requires n-1 clock cycles with n representing the bit width of the divisor and dividend. Thus, the general parameter (GP) is added to to the basis FIR output on every nth clock cycle.

The high-level block diagram of the hardware-based GP-extended FIR predictor is depicted in Fig. 3. Also the internal accuracy when using eight bits wide inputs is presented. Using wider internal variables prevents overflows from occurring.

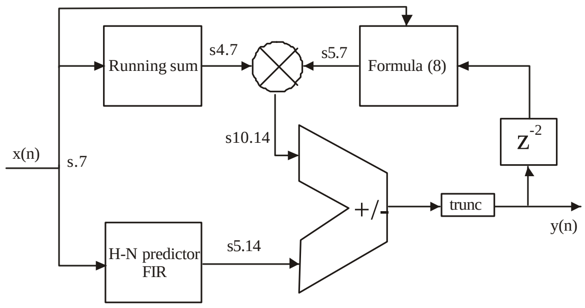

Fig. 3. The high-level block diagram of the FPGA implementation of the GP-extended FIR predictor. The two most important blocks are the basis FIR predictor with constant coefficient multipliers and the implementation of Eq. 8 (Formula 8 in the figure). The internal accuracy is also shown in the format sx.y, where s=sign bit, x=decimal part, y=fractional part.

The maximum sampling rate of the design was 19.6 million samples per second (Msps) when the GP-extended FIR predictor was implemented in 8 bits and 5.4 million Msps when the implementation used 16 bits. The design required hardware resources as follows: the 8 bits wide implementation required 723 logic elements (LEs), which corresponds to approximately 40000 logic gates. Since multiplications by a constant 8 bits wide number fit easily into the internal SRAM-based memory of the programmable logic chip, the implementation can be kept quite compact. The 16 bits wide implementation required 3679 LEs. The attainable speed and required area resources are presented in Table 1, where it can be seen that most of the area resources are required by the updating of *g(n)* (Eq. 8).

Device	Accuracy	Required area resources (in logic elements) Basis FIR	Updating of g(n)	Rest	Total	Maximum speed (Megasamples per second)
EPF10K30ATC144-1	8 bits	280	365	78	723	19,6
EPF10K100ARC240-1	16 bits	1540	2050	89	3679	5,4

Note: The internal accuracy is larger to avoid overflows

Table 1. The area and timing characteristics of the GP-extended FIR predictor in both 8 bits and 16 bits wide implementations .

5 SIMULATION RESULTS

The quality criterion in the presented simulations is the signal-to-noise power ratio (SNR) even though it should more rationally be called signal-to-error power ratio since the example presented is noise free, and the error power results only from the input signal delays and distortion caused by the filters.

The test signal is a Rayleigh distributed signal created according to Jakes' model [2], with signal mean removed for the filtering. This is a model of received power in a mobile communications system [3], where the mobile is set to move at 5 km/h in a system with 1.8 GHz carrier frequency. The Jakes' Rayleigh fading is illustrated in Fig. 4 without mean removal. Sampling is performed at 1.5 kHz [13], which is the sampling rate in the fast power control loop of 3G WCDMA systems.

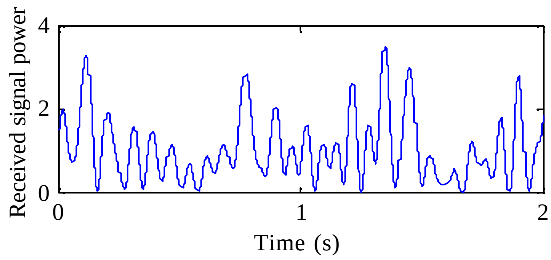

Fig. 4. Two seconds of the Rayleigh distributed signal used to model received signal power in a mobile communications system with carrier frequency of 1.8 GHz and mobile speed 5 km/h. Mean removal is performed before filtering.

Fig. 5 gives the results of applying a H-N predictor along with its GP-extended counterpart to a Rayleigh fading signal shown in Fig. 2. From Fig. 3, it is seen that the pure H-N predictor is able to predict the Rayleigh fading, and that the GP extension improves the performance.

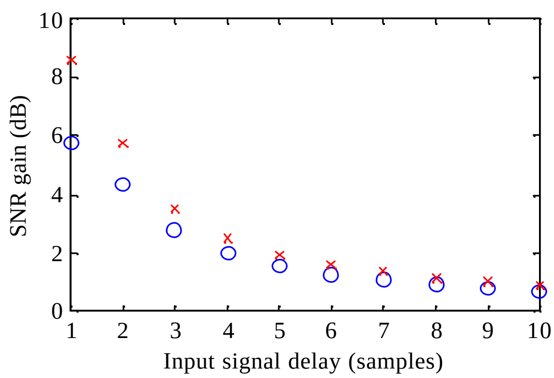

Fig. 5. SNR improvements of predicting 1 to 10 samples delayed Rayleigh fading signals with a one-step-ahead H-N predictor with $N = 10$ (circles) and with a corresponding GP extended predictor with $\gamma = 0.001$ (crosses).

The accuracy of the hardware implementation was investigated by comparing the simulation results of the finite wordlength hardware implementation with the practically infinite precision of Matlab simulations (See also Fig. 2). It was noted, that an implementation with 8 bits precision already yielded sufficient accuracy, and 16 bits precision was practically indistinguishable from the Matlab simulations with double precision floats. These results are illustrated in Fig. 6.

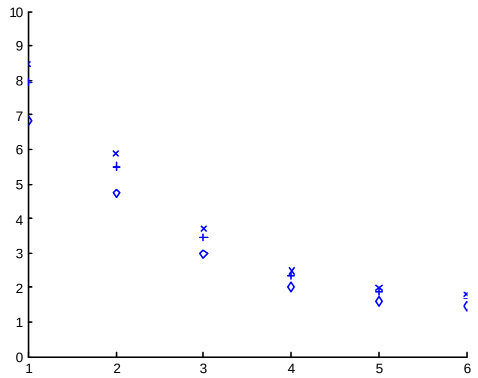

Fig. 6. SNR performance of the one-step-ahead H-N predictor in 64-bit floating point numbers (Matlab, crosses), 16-bit fixed-point numbers (Altera FPGA, plus sign) and 8-bit fixed-point numbers (Altera FPGA, diamond)

6 CONCLUSIONS

Adaptive general parameter extension to polynomial-predictive FIR filters expands the applicability of the filters beyond their nominal input signal delays. The adaptation is of low complexity, and involves only a single parameter that is applied as an adaptive gain to the FIR coefficients. The hardware implementation is simple and straightforward and achieves remarkably good throuhput rates. A promising application worth further research is to use the GP-extended FIR predictor in WCDMA power control.

REFERENCES

[1] J.M.A. Tanskanen, A. Huang, T.I. Laakso, and S.J. Ovaska, "Prediction of received signal power in CDMA cellular systems," Proceedings of the 45th IEEE Vehicular Technology Conference, Chicago, IL, July 1995, pp. 922–926.

[2] W. C. Jakes (ed.), *Microwave Mobile Communications*, John Wiley & Sons, Inc., 1974.

[3] D. Parsons, *The Mobile Radio Propagation Channel*, Pentech Press limited, 1992.

[4] P. Heinonen and Y. Neuvo, "FIR-median hybrid filters with predictive FIR substructures," IEEE Trans. Acoustics, Speech, and Signal Processing, vol. 36, pp. 892–899, June 1988.

[5] A.A. Ashimov and D.J. Syzdykov, "Identification of high dimensional system by the general parameter method," in Preprints 8th Triennial World Congress of Int. Federation of Automatic Control, Aug. 1981, Kyoto, Japan, pp. 32–37.

[6] J.M.A. Tanskanen, O. Vainio, and S.J. Ovaska, "Adaptive general parameter extension for tuning FIR predictor," Proc.of the IFAC Workshop on Linear Time Delay Systems, Ancona, Italy, Sept. 2000.

[7] Implementing FIR Filters in FLEX Devices, Application Note 73, Altera Corporation.

[8] Max+Plus II Programmable Logic Development System & Software, January 1998, Altera Corporation.

[9] Altera FLEX 10K Embedded Programmable Logic Family Data Sheet, ver. 4.02, May 2000, Altera Corporation.

[10] S.J. Mitra, *Digital Signal Processing, a Computer-Based Approach*, McGraw-Hill, 2001, pp. 583-659.

[11] Implementing Multipliers in FLEX 10K EABs, Technical Brief, March 1996, Altera Corporation.

[12] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, 1995.

[13] H. Holma and A. Toskala (editors), *WCDMA for UMTS, Radio Access for Third Generation Mobile Communications*, John Wiley & Sons, 2000, pp. 188-203.