Design of Multi-Delay Predictive Filters Using Dynamic Programming

Konsta Koppinen[1†], Jari Yli-Hietanen[2] and Peter Händel[3]
Signal Processing Laboratory
Tampere University of Technology
P.O.Box 553, 33101 Tampere, Finland
Tel: +358 3 365 2278, fax: +358 3 365 3857
e-mail: [1] makoko@cs.tut.fi, [2] ypsilon@cs.tut.fi, [3] peter.handel@s3.kth.se

Abstract— **Multi-delay predictive FIR filters utilizing a small number of multipliers are proposed. These filters are shown to have substantially lower noise gain than the standard minimum-length predictors using the same amount of multipliers. These filters are formulated for both arbitrary-order polynomial and sinusoidal signal prediction. The use of dynamic programming for the efficient optimization of these filters is proposed.**

I. Introduction

Several techniques have been developed for the design and optimization of linear filters capable of extrapolating polynomial and sinusoidal signals. Among these are the Heinonen-Neuvo predictors [3], frequency-response optimized FIR predictors [5] and IIR predictors [4]. The Heinonen-Neuvo predictors are a good prototype of predictive filters but are inflexible. The lowpass-predictors have flexible design methods for designing optimal predictors with various constraints but the limitations of the underlying FIR structure cannot be overcome. IIR predictors offer a more general structure for optimization and good performance but currently they have to be optimized using general optimization methods.

In this paper we propose a structure for FIR predictors where only the number of multipliers and the maximum delay are limited, i.e. single-tap delays in a standard FIR filter may be replaced by multi-tap delays. A schematic diagram of the structure is shown is Figure 1.

These structures can be efficiently implemented e.g. using signal processors where the number of multiplications is more critical the the number of delays. As we will shorly see, this structure can significantly reduce the noise gain of the filter when compared with the optimal single-delay Heinonen-Neuvo predictors.

II. Formulation

In this section we formulate the design problem for multi-delay FIR predictors for polynomial and sinusoidal signals.

Consider first the case of polynomial prediction of order L and prediction step p. Thus, when we filter any signal

† This research has been partly funded by the Nokia Foundation.

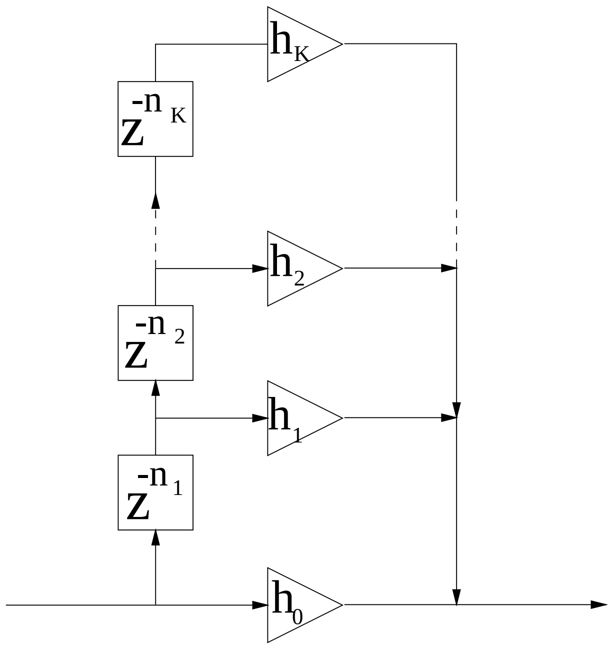

Fig. 1. Direct-form structure of multi-delay FIR filters. The delays z^{-n_k} are integer delays, i.e. $n_k \in \{1, 2, 3, \ldots\}$.

of the form

$$P_L(n) = \sum_{\ell=0}^{L} \alpha_\ell n^\ell,\ \alpha_\ell \in \mathbf{R} \tag{1}$$

we require the output $y(n)$ to be $P_L(n+p)$, where p is not required to be an integer. The filtering operation for linear FIR filters of order M can be expressed as an inner product between the reversed input $x(n)$ and the vector of impulse response coefficients:

$$y(n) = \mathbf{x}(n)^T \mathbf{h}, \tag{2}$$

where $\mathbf{x}(n)$ is the reversed input window

$$\mathbf{x}(n) = \begin{pmatrix} x(n) & x(n-1) & \cdots & x(n-M) \end{pmatrix}^T \tag{3}$$

and $\mathbf{h}$ is the impulse response vector

$$\mathbf{h} = \begin{pmatrix} h(0) & h(1) & \cdots & h(M) \end{pmatrix}^T. \tag{4}$$

The constraint that the filter $\mathbf{h}$ predicts correctly the polynomial input frames

$$\begin{pmatrix} M^\ell & (M-1)^\ell & \cdots & 0^\ell \end{pmatrix}^T, \ \ell = 0, 1, \ldots, L \tag{5}$$

is enough to guarantee prediction of all $P_L(n)$ [5]. Here we define $0^0 = 1$. Thus polynomial prediction can be formulated as the requirement

$$\mathbf{P}^T\mathbf{h} = \mathbf{p}, \tag{6}$$

where $\mathbf{P}$ is a matrix containing the reversed polynomial input frames as columns and $\mathbf{p}$ is the vector of desired outputs

$$\begin{pmatrix} (M+p)^0 & (M+p)^1 & \cdots & (M+p)^L \end{pmatrix}^T. \tag{7}$$

The Heinonen-Neuvo predictors are now obtained as the minimum-norm solution of the underdetermined (if $M > L$) matrix equation (6), explicitly given by

$$\mathbf{h} = \mathbf{P}(\mathbf{P}^T\mathbf{P})^{-1}\mathbf{p}. \tag{8}$$

The constraint of sinusoidal signal prediction of known frequencies can also be given in the form of (6). We require $\mathbf{h}$ to predict correctly any signal

$$S_m(n) = \sum_{\ell=0}^{m} \alpha_\ell \sin(\omega_\ell n + \phi_\ell), \tag{9}$$

where α_ℓ and ϕ_ℓ are arbitrary constants and the frequencies ω_ℓ are known. Again, it can be shown that correct prediction of the basic frames

$$\sin(\omega_\ell n), \ n = 0, 1, \ldots, M, \ \ell = 0, 1, \ldots, m \tag{10}$$

and

$$\cos(\omega_\ell n), \ n = 0, 1, \ldots, M, \ \ell = 0, 1, \ldots, m \tag{11}$$

is necessary and sufficient for the prediction of all $S_m(n)$ [2].

Using these formulations, we can state the problem of multi-delay predictor design using matrix notation. Let the maximum allowable filter order (i.e. maximum delay) be M. We seek a minimum noise gain multi-delay predictor using N multipliers. In (6) this means that we constrain the vector $\mathbf{h}$ to have N non-zero elements.

This leads to a nonlinear optimization problem where (at least implicitly) the following parameters have to be found:

- the optimal delays, i.e. the delays of the non-zero impulse response values.
- the optimal impulse response for these delays.

The second problem is readily solved using the formulation above. Specifically, assume that the N delays of the non-zero impulse response values of the polynomial and/or sinusoidal signal predictor are $n_1, n_2, \ldots, n_N$. By taking into account the zeros in the impulse response vector, the polynomial prediction constraint matrix $\mathbf{P}$ is now

$$\begin{pmatrix} (n_N)^0 & (n_N)^1 & \cdots & (n_N)^L \\ (n_{N-1})^0 & (n_{N-1})^1 & \cdots & (n_{N-1})^L \\ \vdots & \vdots & \cdots & \vdots \\ (n_1)^0 & (n_1)^1 & \cdots & (n_1)^L \end{pmatrix} \tag{12}$$

and the corresponding vector $\mathbf{p}$ is

$$\begin{pmatrix} (N+p)^0 & (N+p)^1 & \cdots & (N+p)^L \end{pmatrix}^T. \tag{13}$$

Let us denote these by $\mathbf{P}_{n_1,\ldots,n_N}$ and $\mathbf{p}_{n_1,\ldots,n_N}$, respectively. Again we can use these in equation (8) to find the non-zero elements of the predictor impulse response vector with minimum norm.

As an example, consider designing the minimum noise-gain predictor of prediction step $p = 1$ for polynomials of order $L = 2$ using just the delays $0, 1, 5, 6$ and 10 and with maximum order 10. Inserting (12) and (13) in (8) results in the vector

$$\begin{pmatrix} 0.9110 & 0.5302 & -0.3041 & -0.3404 & 0.2033 \end{pmatrix}^T, \tag{14}$$

which corresponds to the impulse response

$$h(n) = \begin{cases} 0.9110 & ,n = 0 \\ 0.5302 & ,n = 1 \\ -0.3041 & ,n = 5 \\ -0.3404 & ,n = 6 \\ 0.2033 & ,n = 10 \\ 0 & ,\text{otherwise} \end{cases}. \tag{15}$$

This is actually the optimum filter for the given constraints with 5 multipliers. The sinusoidal prediction case is handled in the same manner by including only the approriate delays in the contraint matrix and desired response vector.

The more difficult procedure in multi-delay predictor design is the determination of the optimal delays. In the next section we propose the use of dynamic programming for this task.

III. Using dynamic programming for delay optimization

Dynamic programming (DP) refers to an optimization method to efficiently determine the sequence of optimal choices for structured problems. Here we will concentrate on finding the optimal non-zero taps of a predictive filter.

Let us now explain the procedure. Our problem is to determine a subgroup of size N from the set of possible delays $0, 1, \ldots, M$. We define a function $\mathcal{F} : \mathcal{X} \mapsto \mathcal{R}$, where $\mathcal{X}$ is the set of subsets of $0, 1, \ldots, M$, which gives the 'cost' of a subset of delays. This function is defined by the minimum possible noise gain for each set of delays and can be explicitly calculated for any $x \in \mathcal{X}$ by

$$\begin{aligned} \mathcal{F}(x) &= \|\mathbf{P}_x(\mathbf{P}_x^T\mathbf{P}_x)^{-1}\mathbf{p}\|^2 & (16) \\ &= \mathbf{p}^T(\mathbf{P}_x^T\mathbf{P}_x)^{-1}\mathbf{P}_x^T\mathbf{P}_x(\mathbf{P}_x^T\mathbf{P}_x)^{-1}\mathbf{p} & (17) \\ &= \mathbf{p}^T(\mathbf{P}_x^T\mathbf{P}_x)^{-1}\mathbf{p}, & (18) \end{aligned}$$

TABLE I

DP ALGORITHM FOR MULTI-DELAY PREDICTOR DESIGN.

DP delay optimization

Initialization:

```
compute all $\binom{M+1}{L+1}$ different sets of $L+1$
   delays into $\mathcal{S}_{L+1}$
for i=$L+2, L+3, \ldots, N$:
   for $(n_1, n_2, \ldots, n_{i-1}) \in \mathcal{S}_{i-1}$:
      for j=$0, 1, \ldots, M$:
         calculate optimal predictor $h$ with
            delays $(n_1, n_2, \ldots, n_{i-1}, j)$
         set NG(j)=noise gain of $h$
      end
      set k=index of the maximum of NG
      set v=vector $(n_1, n_2, \ldots, n_{i-1}, k)$ in ascending order
      set $\mathcal{S}_i = \mathcal{S}_i \cup v$
      store filter and noise gain corresponding
         to these delays
   end
end
find minimum of NG and corresponding filter
```

if the matrix $(\mathbf{P}_x^T \mathbf{P}_x)^{-1}$ is invertible, otherwise we define $\mathcal{F}(x) = \infty$.

Denote by $\mathcal{X}_k$ the subset of $\mathcal{X}$ consists of the delay sets with exactly k elements.

The algorithm keeps track of subsets of $\mathcal{X}$, which are increased in size. Let the number of columns in the constraint matrix $\mathbf{P}$ be $L+1$. The algorithm is initialized with all of the different $\binom{M+1}{L+1}$ combinations of picking $L+1$ delays from $M+1$ positions, which are stored in $\mathcal{S}_{L+1}$. Having determined the sets of i delays, for each element $x \in \mathcal{S}_i$, the value of

$$\mathcal{F}(x \cup d) \tag{19}$$

is determined for all $d \in 0, 1, \ldots, M$. The minimum over d is found and the set $x \cup d$ is added to $\mathcal{S}_{i+1}$.

The dynamic programming algorithm for determining the best multi-delay predictor is summarized in Table I.

The number of function evaluations (filter designs) required in the dynamic programming method is approximately $\binom{M+1}{L+1}(N-L-1)(M+1)$ whereas an exhaustive search would require evaluating $\binom{M+1}{N}$ filters.

The cost function is invariant to the order of the arguments. This is used to further reduce the computational complexity by only adding to $\mathcal{S}_i$ the delays which are unique as sets. This is utilized in the algorithm by sorting the chosen vector $(x \cup d)$ before adding it to $\mathcal{S}_{i+1}$.

Unfortunately we have not been able to prove that the filter resulting from the DP optimization is the true optimum for $L > 0$. The Bellman optimality principle does not hold for the cost function $\mathcal{F}$, since the optimal k delays are not necessarily included in the optimal $k+1$ delays. However, comparison of the DP solution and the optimal solution for several hunderds of thousands of cases with practical values of L and M has resulted in no discrepancies, even with no structure in $\mathbf{P}$ and $\mathbf{p}$. This suggests that the DP algorithm may be used for practical optimization if exhaustive search is infeasible (i.e. $M = 50$, $N = 25$ requires $\approx 1.26e14$ function evaluations, each requiring matrix inversion), but the optimality is still an open question.

The case $L = 0$ (i.e. one constraint) for arbitrary $\mathbf{P}$ (now a vector) and $\mathbf{p}$ (a scalar) is

$$\mathrm{argmin}_{v_1,\ldots,v_N \in V} \frac{1}{\mathbf{p}^T(\sum_{n=1}^{N} v_n v_n^T)\mathbf{p}} \tag{20}$$

$$= \mathrm{argmin}_{v_1,\ldots,v_N \in V} \frac{1}{\|\mathbf{p}\|^2(\sum_{n=1}^{N} v_n^2)} \tag{21}$$

$$= \mathrm{argmax}_{v_1,\ldots,v_N \in V} \sum_{n=1}^{N} v_n^2, \tag{22}$$

where $V = \{\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_{M+1}\}$ and it is required that $v_1, \ldots, v_N$ are all different. The solution is to pick the delays corresponding to the N components in $\mathbf{P}$ with largest absolute values. Obviously the dynamic programming algorithm will yield this solution, although it can be obtained more directly as above. Interestingly, the simpler one-at-at-time algorithm, where only one set of delays is retained and updated with the optimal extension, will also work in this case but not (in general) for $L > 0$.

IV. DESIGN EXAMPLES AND RESULTS

A. 3-TAP PREDICTION

Let us first consider designing a predictor for second-order polynomials with prediction step 1. Thus the minimum number of multiplications required is 3. Figure 2 shows the optimal locations of the 3 taps as a function of the maximum predictor order. These always include the 0-delay and maximum-delay taps. The noise gains corresponding to these filters are also shown in Figure 2. We see a sharp decrease in the noise gain for multi-delay predictors when compared to the minimum-delay predictor (which is the first one in Figure 2). The amplitude responses of some of the predictors are shown in Figure 3. Note the quasi-periodicity in the amplitude response due to the multi-delay structure.

B. 6-TAP PREDICTION

The same design was carried out using a maximum of 6 multiplications. The results are shown in Figure 4. It is noteworthy that the optimum delays 'jump' when the maximum order increases from 10 to 11 and from 28 to 29. The reason for this is shown in Figure 5, where the cost function is shown for each possible addition to the set of delays $1, 2, 3, 18, 30$. The nonlinear nature of the optimization problem can be clearly seen (the abrupt discontinuities result from extending the filter with a delay that is already present).

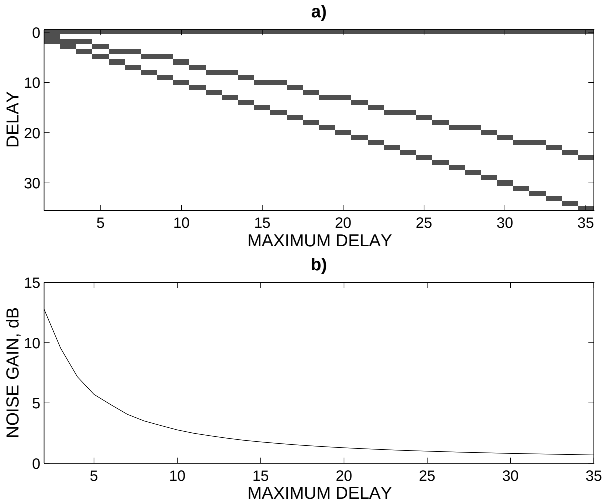

Fig. 2. Top: optimal delay locations for 3-tap prediction of 2^{nd}-order polynomials. The minimum- and maximum-delay taps are always included. Bottom: noise gain as a function of the maximum allowable delay, corresponding to the top figure.

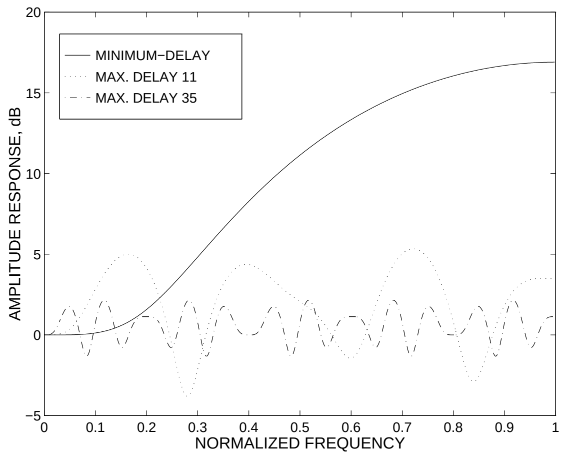

Fig. 3. Amplitude responses of 3-tap predictive filters for 2^{nd}-order polynomials with different maximum delays. The noise gain decreases with increasing maximum delay. At the same time, the prediction bandwidths decrease as well. For the different maximum delays d_{max}, they are: 0.38π for $d_{max} = 2$, 0.10π for $d_{max} = 11$, 0.03π for $d_{max} = 35$.

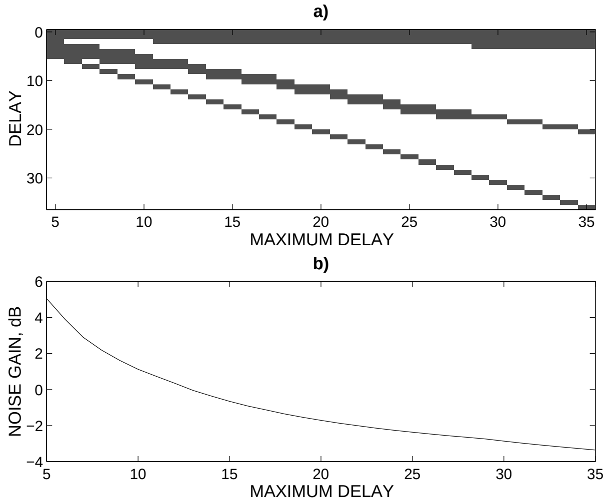

Fig. 4. Top: optimal delay locations for 6-tap prediction of 2^{nd}-order polynomials. Again the minimum- and maximum-delay taps are always included. Bottom: noise gain as a function of the maximum allowable delay, corresponding to the top figure. Note that the noise gain of the minimum-delay 6-tap predictor can be achieved with a 3-tap predictor with maximum delay 6.

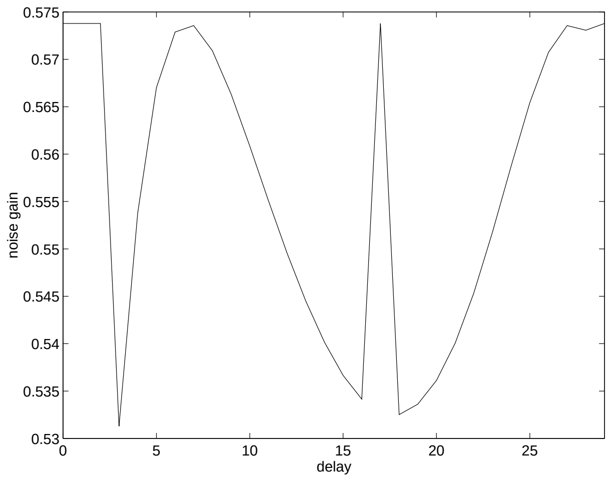

Fig. 5. The minimum noise gain of the filter using delays $1, 2, 3, 18, 30$ extended with an additional delay.

V. Conclusions

We proposed the use of multi-delay polynomial and sinusoidal predictors which have desirable properties in certain implementations. Closed-form formulas can be used to determine the filter tap weights for any given set of delays. We proposed dynamic programming to solve the nonlinear optimization problem of determining the optimal set of delays and showed that the results suggest the optimality of the method, although it is still an open problem. Finally, practical design examples were given which showed the benefits of using multi-delay predictors.

References

[1] P. Händel, “Predictive digital filtering of sinusoidal signals”,*IEEE Transactions on Signal Processing*, Vol. 46, No. 2, February 1998. submitted for publication.

[2] P. Händel, “Predictive FIR/IIR filters for narrow-band waveforms”, *The First European Conference on Signal Analysis and Prediction*, Prague, Czech Republic, 24-27 June, 1997.

[3] P. Heinonen and Y. Neuvo, “FIR-median hybrid filters with predictive substructures”, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 36, No. 6, pp. 892-899, June 1988.

[4] S.J. Ovaska, O. Vainio and T.I. Laakso, “Design of predictive IIR filter via feedback extension of FIR forward predictors”, *38th Midwest Symposium on Circuits and Systems*, Rio de Janerio, Brazil, August, 1995, pp. 370-375.

[5] K. Koppinen, O. Vainio and J. Astola, “Analysis and design of polynomial predictors”, *1996 IEEE Nordic Signal Processing Symposium*, Espoo, Finland, 24-27 September, 1996, pp. 45-48.

[6] G. Hadley, *Nonlinear and Dynamic Programming*, Addison-Wesley, USA, 1964.

[7] G. Nemhauser, *Introduction to Dynamic Programming*, Wiley, New York, 1966.