Abstract—Newton predictors have been used by mathematicians to extrapolate tables of polynomials and transcendental functions. However, the predictors based on this computationally efficient algorithm have considerable gain at the higher frequencies. This property reduces their applicability to practical signal processing where the narrow-band primary signal is often corrupted by additive wide-band noise. In a recent paper, two alternative modifications were proposed to the original algorithm that can be used to extrapolate low-order polynomials. In both approaches, the highest order difference of successive input samples, approximating the constant nonzero derivative, is smoothed before it is added to the lower order differences. The additional smoothers reduce the undesired noise gain of Newton predictors.

In this paper, we extend the Linear Smoothed Newton (LSN) predictor by including a recursive term into the basic transfer function and cascading the rest of the successive difference paths with appropriately delayed extrapolation filters of corresponding polynomial orders. This leads to a new class of computationally efficient IIR predictors with significantly lowered gain at the higher frequencies. The introduced Recursive Linear Smoothed Newton predictor is analyzed in the time and frequency domains, and compared to the original Newton predictor, the LSN predictor, and the optimal Heinonen–Neuvo FIR predictor.

Keywords—Polynomial extrapolation, forward prediction, Newton-type prediction, recursive prediction.

I. INTRODUCTION

A. General

Extrapolation of limited data sets is a subject that dates to antiquity, and comes up frequently in all fields of applied and theoretical science, e.g., in instrumentation and measurement. The common problem is to forecast future samples of a primary signal corrupted by additive white Gaussian noise or uniformly distributed noise [1], [2]. Equation (1) defines the generally available noisy input sequence

$$u(p) = u(p) + e(p)$$

where $u(p)$ is the primary signal, and $e(p)$ the disturbing noise component. The difference equation of a general $n$-step-ahead predictor can be written as

$$\hat{u}(p + n) = \sum_{k=1}^{w} \alpha_k \hat{u}(p + n - k)$$

$$+ \sum_{i=0}^{w} \beta_i u(p - k).$$

The coefficients $\alpha_k$ and $\beta_i$ are real constants. However, in FIR prediction, all the feedback coefficients $\alpha_k$ are set equal to zero. In the noise-free situation we require $\hat{u}(p + n) = u(p + n)$. The notation established above is applied throughout this paper.

There exist two main approaches to design a forward predictor: in the first approach, one tries to fulfill some time-domain requirements (e.g., the prediction error must be zero for an $M$th order polynomial), and in the second approach one approximates the ideal prediction characteristics in the frequency domain (i.e., magnitude equal to one and linear increasing phase). Although the second approach has proven to be the most successful approach for interpolation [3], differentiation, and integration [4], this is no longer true for extrapolation. It can be explained by the fact that it is impossible to approximate general noncausal characteristics satisfactorily by a causal filter. The only way to circumvent this problem is to introduce some model for the primary signal. In this paper, we consider polynomial extrapolation that is a special case of forward prediction. Our signal model is assumed to be of form

$$u(p) = \lambda_0 + \lambda_1 p + \lambda_2 p^2 + \cdots$$

$$+ \lambda_{M-1} p^{M-1} + \lambda_M p^M,$$

where the polynomial coefficients $\lambda_i$, $i \in [0, M]$, are unknown real constants. An important consequence of the explicit modeling of the primary signal is that the prediction result may become poor with more general signal types, for example with band-limited signals. However, many real-world measurement signals can be approximated with sufficient accuracy by low-order polynomials. Polynomial approximations are discussed from the theoretical point of view in [5].

B. Newton-Type Prediction

Newton predictors are attractive for extrapolation of polynomials because of their low computational complex-
ity and the straightforward design process. A general \( n \)-step-ahead Newton predictor of an \( M \)-th order polynomial can be expressed as the \( z \)-domain transfer function [6]

\[
H_n(z) = \sum_{k=0}^{M} (1 - z^{-n})^k.
\]

Successful applications of Newton predictors have been reported in the instrumentation and measurement literature. Athani applied them to sample trend monitoring that was used for process failure detection [7]. In that application, Newton predictors were used to forewarn the operator about an impending crisis in a slow industrial process, and enable him to take precautionary measures before the crisis occurs. Ovaska proposed Newton predictors for transmission error correction in a measuring-oriented communication protocol [8], [9]. In that plain protocol, the conventional handshaking and lost-message repetition were replaced by a forward predictor; this reduced the control character transmission overhead drastically.

In a recent paper [10], two smoothed versions of the original Newton predictor were proposed. Those predictors were called Linear Smoothed Newton (LSN), and Median Smoothed Newton (MSN). The LSN predictor is recommended to be used with white Gaussian and uniformly distributed additive noises, and the MSN counterpart with impulsive and Laplacian distributed noises. In both cases, the smoothing operation is performed solely for the highest order difference \( (1 - z^{-n})^M \), where \( M \) is the order of the polynomial to be extrapolated, and \( n \) is the desired prediction step. The highest order difference has the largest noise gain, and it approximates the highest order nonzero derivative that attains a constant value if the polynomial is noiseless. All the lower order derivatives are time varying, and all the higher order derivatives are equal to zero. The general \( z \)-domain transfer function of LSN \( n \)-step-ahead predictors can be written as

\[
G_n(z) = S(z)(1 - z^{-n})^M + \sum_{k=0}^{M-1} (1 - z^{-n})^k
\]

where \( S(z) \) is the transfer function of some low-pass filter, preferably a moving averager

\[
S(z) = \sum_{i=0}^{N-1} z^{-i}
\]

that smoothes the noisy \( M \)-th order difference sequence. Insertion of a linear smoother into the original algorithm reduces the high-pass nature and noise gain of Newton predictors without adding any steady-state prediction error to the result. With corrupting mutually uncorrelated additive noise, the noise gain of the one-step-ahead LSN ramp predictor \( n = 1, M = 1 \) approaches unity, as the width of the moving average window approaches infinity. Thus the LSN predictor can never attenuate the noise content of the primary signal.

In this paper, we propose a novel extension to the Linear Smoothed Newton predictor that widens the application range of smoothed Newton predictors. In Section II of this paper, the data flow diagram and the corresponding transfer function of the new Recursive Linear Smoothed Newton (RLSN) predictor are introduced. A brief \( z \)-domain analysis of the RLSN predictors is given in Section III. In Section IV, some numerical comparisons are presented. Section V closes this paper with a conclusion and some comments on future research directions on smoothed Newton-type polynomial extrapolation filters.

II. RECURSIVE LINEAR SMOOTHED NEWTON PREDICTORS

Phase I—Recursive Extension

To enhance the applicability of Linear Smoothed Newton predictors, we first propose a recursive extension to the LSN predictor. The motivation of the present discussion is to obtain a forward predictor that both shares the simplicity of the original LSN predictor, and further reduces the undesired noise gain, thus being suitable to prediction of signals with low signal-to-noise ratio (SNR). In this new approach, we feed back the predicted estimate \( \hat{u}(p) \) of the current input sample \( u(p) \), add it with weighting \( (1 - a) \) to the weighted true input \( a u(p) \), and further add this sum to the sum of the successive differences \( k \in [1, M] \). The difference operators get their inputs \( u(p) \) directly without prescaling. This kind of manipulation does not cause any steady-state error into predictor’s output. The modified \( z \)-domain transfer function can be expressed as

\[
P_n(z) = \frac{a + S(z)(1 - z^{-n})^M + \sum_{k=1}^{M-1} (1 - z^{-n})^k}{1 - (1 - a)z^{-n}}.
\]

For the one-step-ahead ramp predictor, where \( S(z) \) is an \( N \)-length moving averager of (6), the transfer function of (7) can be simplified and written as

\[
P_1(z) = \frac{(a + 1/N) - z^{-N}}{1 - (1 - a)z^{-1}}.
\]

This transfer function has only one additional multiplication and addition if compared to the nonrecursive LSN counterpart—and with appropriate values of parameters \( a \) and \( N \) it has the desired low-pass nature. The width of the passband is mainly controlled by \( N \), and the stopband attenuation by \( a \).

Delayed cascading of the basic sections (8) offers a viable way to further increase the stopband attenuation. It is also possible to obtain ramp predictors with a complex conjugate pole pair by cascading two basic sections with \( a = a_e + ia_m \) and \( a = a_e - ia_m \), respectively. The
resulting transfer function can be expressed as

\[ L(z) = \frac{a^2 + 2a_m + a_{im} + 1}{1 - (-2a_e + 2)z^{-1} - (-a_e + 2a_m - a_{im} - 1)z^{-2}}. \] (9)

The coefficients of a cascade structure can be tailored to a specific application using some constrained optimization procedure, e.g., the interior penalty function method [11] coupled with the Fletcher-Powell algorithm [12]. This penalty function extension ensures stable transfer functions if the starting point \( \{a_e, a_{im}\}_{\text{INITIAL}} \) is inside the unit circle.

**Phase 2—Smoothing with Delayed Extrapolators**

Although the ramp predictor of the form (8) may have the low-pass nature, the predictors (7) of higher order polynomials still have gain greater than or equal to unity at the higher frequencies. The reason for this is that only the primary input signal and the highest order successive difference are smoothed: the primary signal by an "exponential" averager, and its highest order non-zero derivative by a "uniform" averager. All the other difference paths have disturbing gain at the higher frequencies. Smoothing of the approximations of time-varying derivatives cannot be carried out by a conventional low-pass filter or moving averager, because they all introduce lag into the primary signal and thus deteriorate the prediction function [10]. To overcome this problem, we propose an enhanced version of the basic recursive transfer function of (7). In the new transfer function, the lower order successive differences corresponding to the time-varying derivatives are smoothed by delayed polynomial extrapolators. The delay is needed to compensate the forward prediction step, i.e., the output \( \hat{u}(p) \) of a delayed predictor is an estimate of the underlying primary signal \( u(p) \) corresponding to the noisy present input \( u(p) \). First, we need an additional ramp predictor to obtain a predictor for second-order polynomials; then we use both the ramp predictor and the second-order polynomial predictor to obtain the predictor for third-order polynomials, etc. Thus the general transfer function can be written in a form that contains a recursion

\[ K_m(z) = a + S(z) \frac{(1 - z^{-1})^M}{1 - (1 - a)z^{-1}} + z^{-1} \sum_{k=1}^{M-1} K_{M-k}(z)(1 - z^{-1})^k. \] (10)

We call this new class of polynomial extrapolation filters Recursive Linear Smoothed Newton (RLSN). The RLSN extrapolator, depicted in Fig. 1, offers great flexibility to the user to optimize its characteristics for different signal shapes, noise conditions, and computational environments.

**III. Z-Domain Analysis of the RLSN Predictor**

In this section, we give expanded forms of the z-domain transfer functions of some low-order one-step-ahead
RLSN predictors, and study their poles and zeros to gain understanding on the frequency response characteristics. Although those equations may first look complicated, a more detailed study shows that in their final substituted numerical forms the numerators have sparse impulse responses with several coefficients equal to zero. The relative sparsity increases as \( N \) increases. Let us assume that \( a \) is a real constant, and \( N \) is a positive-valued integer.

### A. First-Order Predictor

The simplest form of RLSN predictors is the one-step-ahead ramp predictor \( K_1(z) \), or \( P_1(z) \) of (8). In that case there is no recursion involved in the transfer function. \( K_1(z) \) has a single pole at \( z = 1 - a \), and \( N \) zeroes equally spaced in angle on a circle of radius \((1 + Na)^{-1/N}\), located symmetrically with respect to the real axis. As \( a \) approaches zero from the positive direction, the complete transfer function approaches the recursive running-sum structure

\[
T(z) = \frac{1}{N} \frac{(1 - z^{-N})}{1 - z^{-1}}
\]

with all the zeros located on the unit circle. \( T(z) \) can be interpreted as the sum of the geometric series \( S(z) \) of (6). The recursive running sum is no more a ramp predictor but a level predictor instead. For larger values of \( a \), the low-pass nature of \( K_1(z) \) diminishes as the pole moves toward the origin. The magnitude response \( |K_1(e^{j\omega}, a)| \) is shown as a function of \( a \) with \( N = 16 \) in Fig. 2(a).

### B. Second-Order Predictor

The transfer function of a second-order RLSN predictor can be expressed as

\[
K_2(z) = \frac{1}{N} \frac{(1 + aN) + (a^2N + a - 1)z^{-1} - (aN + a)z^{-2} - z^{-N} + (1 - a)z^{-(N+1)} + az^{-(N+2)}}{(1 - (1 - a)z^{-1})^2}.
\]

It has a double pole at \( z = 1 - a \). Since the order of the numerator polynomial is \( N + 2 \), the transfer function has \( N + 2 \) zeroes. For small positive values of \( a \), these are located close to the unit circle, evenly spaced, providing essentially the same frequency response characteristics as \( K_1(z) \). For larger values of \( a \), there is a region of elevated gain in the vicinity of \( \pi/2 \), as shown in Fig. 2(b). This undesired phenomenon is considerably strengthened in the higher order predictors.

### C. Higher Order Predictors

The transfer function of a third-order predictor can be reduced to

\[
K_3(z) = \frac{(1 + aN) + (2a^2N + 2a - 2)z^{-1} + (a^3N + a^2 - 3aN - 4a + 1)z^{-2} + (2a + 2aN - 2a^2 - 3a^2N)z^{-3}}{N(1 - (1 - a)z^{-1})^3} + \frac{(a^2N + a^2)z^{-4} - z^{-3} + (2 - 2a)z^{-(N+1)} + (4a - a^2 - 1)z^{-(N+2)} + (2a^2 - 2a)z^{-(N+3)} - a^2z^{-(N+4)}}{N(1 - (1 - a)z^{-1})^3}.
\]

There is a triple pole at \( z = 1 - a \), and \( N + 4 \) zeroes. In a similar way, we can show that \( K_4(z) \) has a quadruple pole at \( z = 1 - a \), and \( N + 6 \) zeroes (coefficients of \( K_4(z) \) are given in Table I). The magnitude responses \( |K_3(e^{j\omega}, a)| \) and \( |K_4(e^{j\omega}, a)| \) are shown in Figs. 2(c) and (d), respectively. Note that the range of \( a \) is made narrower in these plots because for larger values of \( a \), the high-frequency gain becomes unpractically large for applications with poor SNR.

Zoomed-phase responses of several RLSN predictors are shown in Fig. 3. This gives us an indication of the effective prediction bandwidth of these predictors. The bandwidth \( \Delta f \) of the region with linear increasing phase increases as the order \( M \) of the polynomial model (3) increases. For example, with \( M = 4 \) the value of \( \Delta f \) is approximately 0.02 times the Nyquist frequency. Thus these predictors work solely with narrow-band primary signals—this is a general characteristic of polynomial extrapolators [6], [10].

### IV. Numerical Comparisons

Let us compare the computational complexities and noise gains of different second- and fourth-order polynomial extrapolation filters. Assuming that the additive zero-mean input noise has a flat power spectrum, i.e., the noise is mutually uncorrelated, the noise gain (NG) can be computed either in the time or frequency domain [13]:

\[
NG = \sum_{k=\infty}^{-\infty} |h(k)|^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} |H(e^{j\omega})|^2 d\omega
\]

where \( h(k) \) is the impulse response of the prediction filter and \( H(e^{j\omega}) \) the corresponding frequency response. In Tables II and III, we give the number of arithmetic operations and unit delays required to implement different types of one-step-ahead predictors. The following predictors are included in the comparison: the optimal FIR predictor [2], [14], the basic Newton predictor (4), and three types of smoothed Newton predictors discussed in this paper (5), (7), and (10). The coefficients of the optimal FIR predictors for polynomial orders two and four are given in (15).
Fig. 2. Magnitude responses of the first-, second-, third-, and fourth-order RLSN predictors: (a) \( M = 1, n = 1, N = 16, a \) varies in the range \((0, 1]\); (b) \( M = 2, n = 1, N = 16, a \) varies in the range \((0, 1]\); (c) \( M = 3, n = 1, N = 16, a \) varies in the range \((0, 0.5]\); (d) \( M = 4, n = 1, N = 16, a \) varies in the range \((0, 0.2]\).

### TABLE I
COEFFICIENTS OF THE RLSN PREDICTOR FOR FOURTH-ORDER POLYNOMIALS

<table>
<thead>
<tr>
<th>Order of the Coefficient</th>
<th>Numerator</th>
<th>Denominator</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>((1 + aN)/N)</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>((-3 + 3a + 3a^2N)/N)</td>
<td>(-4 + 4a)</td>
</tr>
<tr>
<td>2</td>
<td>((3 - 9a - 6aN + 3a^2 + 3a^3N)/N)</td>
<td>(-4 + 12a + 6a^2)</td>
</tr>
<tr>
<td>3</td>
<td>((-1 + 9a + 8aN - 9a^2 - 12a^2N + a^3 + a^4N)/N)</td>
<td>(-4 + 12a - 12a^2 + 4a^3)</td>
</tr>
<tr>
<td>4</td>
<td>((-3a - 3aN + 9a^2 + 12a^2N - 3a^3 - 6a^3N)/N)</td>
<td>(-4 + 4a + 6a^2 - 4a^3 + a^4)</td>
</tr>
<tr>
<td>5</td>
<td>((-3a^3 - 3a^2N + 3a^3 + 4a^3N)/N)</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>((-a^4 - a^3N)/N)</td>
<td>-</td>
</tr>
<tr>
<td>N</td>
<td>(-1/N)</td>
<td>-</td>
</tr>
<tr>
<td>N + 1</td>
<td>((3 - 3a)/N)</td>
<td>-</td>
</tr>
<tr>
<td>N + 2</td>
<td>((-3 + 9a - 3a^2)/N)</td>
<td>-</td>
</tr>
<tr>
<td>N + 3</td>
<td>((1 - 9a + 9a^2 - a^3)/N)</td>
<td>-</td>
</tr>
<tr>
<td>N + 4</td>
<td>((3a - 9a^2 + 3a^3)/N)</td>
<td>-</td>
</tr>
<tr>
<td>N + 5</td>
<td>((3a^2 - 3a^3)/N)</td>
<td>-</td>
</tr>
<tr>
<td>N + 6</td>
<td>(a^4/N)</td>
<td>-</td>
</tr>
</tbody>
</table>
Fig. 3. Zoomed-phase responses of the first-, second-, third-, and fourth-order RLSN predictors: solid line: \( M = 1, n = 1, N = 10, a = 0.15 \); dashed line: \( M = 2, n = 1, N = 10, a = 0.15 \); dash-dot line: \( M = 3, n = 1, N = 10, a = 0.15 \); dotted line: \( M = 4, n = 1, N = 10, a = 0.15 \).

**TABLE II**

<table>
<thead>
<tr>
<th>Predictor Type</th>
<th>( k/N )</th>
<th>NG</th>
<th>MUL</th>
<th>ADD</th>
<th>DEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimal FIR</td>
<td>( k=16 )</td>
<td>0.73</td>
<td>5</td>
<td>13</td>
<td>21</td>
</tr>
<tr>
<td>Basic Newton, Eq. (4)</td>
<td></td>
<td>19.00</td>
<td>4</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>LSN, Eq. (5)</td>
<td>( N=16 )</td>
<td>5.39</td>
<td>4</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>LSN-FB, Eq. (7)(1)</td>
<td>( N=16 )</td>
<td>1.36</td>
<td>5</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>RLSN, Eq. (10)(1)</td>
<td>( N=16 )</td>
<td>0.23</td>
<td>4</td>
<td>7</td>
<td>20</td>
</tr>
</tbody>
</table>

(1) with \( a = 0.1 \)  
(2) computationally efficient Campbell-Neuvo implementation structure [14]

*Note:* MUL = number of multiplications; ADD = number of additions/subtractions; DEL = number of unit delays.

**TABLE III**

<table>
<thead>
<tr>
<th>Predictor Type</th>
<th>( k/N )</th>
<th>NG</th>
<th>MUL</th>
<th>ADD</th>
<th>DEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimal FIR</td>
<td>( k=16 )</td>
<td>3.66</td>
<td>9</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>Basic Newton, Eq. (4)</td>
<td></td>
<td>251.00</td>
<td>4</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>LSN, Eq. (5)</td>
<td>( N=16 )</td>
<td>73.53</td>
<td>8</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>LSN-FB, Eq. (7)(1)</td>
<td>( N=16 )</td>
<td>22.58</td>
<td>9</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>RLSN, Eq. (10)(1)</td>
<td>( N=16 )</td>
<td>0.50</td>
<td>18</td>
<td>17</td>
<td>22</td>
</tr>
</tbody>
</table>

(1) with \( a = 0.1 \)  
(2) direct form implementation

and (16), respectively [15]:

\[
h_{2nd}(i) = \frac{9k^2 + (9 - 36i)k + 30i^2 - 18i + 6}{k^3 - 3k^2 + 2k},
\]

\( i \in [1, k]. \) (15)

\[
h_{4th}(i) = \frac{[25k^4 + (50 - 300i)k^3}{1} + (1050i^2 - 450i + 275)k^2
\]

\[
- (1400i^3 - 1050i^2 + 1150i - 250)k
\]

\[
+ 630i^4 - 700i^3 + 1050i^2 - 50i + 120
\]

\[
\times [k^5 - 10k^4 + 35k^3 - 50k^2 + 24k]^{-1},
\]

\( i \in [1, k]. \) (16)

It is clearly visible that the RLSN predictor offers the lowest noise gain with low computational complexity. The RLSN is also the only predictor that has true noise attenuation in fourth-order prediction—all the other predictors are gaining the input noise.

Finally, let us equalize the noise gains of the optimal FIR predictor (the coefficients are optimized for minimum output noise power) and the RLSN predictor, and compare their computational complexities. Choosing, for example, the second-order case and requiring that \( NG = 0.5 \), we get \( k = 22 \) for the FIR predictor, and \( a = 0.27 \) with \( N = 16 \) for the RLSN predictor. The number of arithmetic operations (MUL + ADD) in the FIR case is 18, while in the RLSN case it is only 11 (\( \approx 40\% \) less). The number of unit delays is almost equal in both cases.
We can conclude that the RLSN predictors have excellent noise attenuation characteristics and low computational complexity when compared to other commonly used polynomial predictors.

V. CONCLUSION

We have introduced an IIR counterpart of the recently presented Linear Smoothed Newton predictor. The new Recursive Linear Smoothed Newton predictor offers excellent noise attenuation characteristics with low-order polynomials, thus being suitable to instrumentation and measurement applications with poor SNR. The computational complexity of the RLSN predictors is usually low because the numerator of the transfer function has a sparse impulse response, i.e., most of the coefficients of the tapped delay line are equal to zero, and the order of the denominator is typically no more than three.

Several applications exist for the RLSN predictor. These include use as a stand-alone predictor and a real-time curve fitter. Velocity prefiltering, commonly needed in motion control systems, is one potential application area of RLSN predictors. An appropriately delayed RLSN predictor can be used as a “delayless” prefilter or an estimator of a noisy tachometer signal if the underlying primary velocity pattern approximates some polynomial form. This is the case, e.g., in elevator and crane control [16], and in many other machine automation applications.

Our future research will concentrate on developing adaptive extensions to the basic LSN and RLSN structures, and performing corresponding studies with the Median Smoothed Newton predictor. Also, the behavior of the RLSN predictor with nonpolynomial input signals is an important subject of future research. Although we have analyzed solely the properties of RLSN predictors in the one-dimensional case, the proposed approach can be applied to two-dimensional signal processing with similar benefits.

ACKNOWLEDGMENT

The authors thank one of the reviewers for his detailed comments and valuable suggestions aimed at improving the paper.

REFERENCES