

# Shifted Linear Interpolation Filter

H. Olkkonen<sup>1</sup>, J. T. Olkkonen<sup>2</sup>

<sup>1</sup>Department of Physics and Mathematics, University of Eastern Finland, Kuopio, Finland, <sup>2</sup>VTT Technical Research Centre of Finland, VTT, Finland  
 Email: hannu.olkkonen@uef.fi, juuso.olkkonen@vtt.fi

Received September 21<sup>st</sup>, 2010; revised November 8<sup>th</sup>, 2010; accepted November 15<sup>th</sup>, 2010.

## ABSTRACT

Linear interpolation has been adapted in many signal and image processing applications due to its simple implementation and low computational cost. In standard linear interpolation the kernel is the second order B-spline  $\beta_2(t)$ . In this work we show that the interpolation error can be remarkably diminished by using the time-shifted B-spline  $\beta_2(t-\Delta)$  as an interpolation kernel. We verify by experimental tests that the optimal shift is  $\Delta = 0.25$ . In VLSI and microprocessor circuits the shifted linear interpolation (SLI) algorithm can be effectively implemented by the z-transform filter. The interpolation error of the SLI filter is comparable to the more elaborate higher order cubic convolution interpolation.

**Keywords:** Linear Interpolation, Cubic Convolution, B-splines, VLSI

## 1. Introduction

Linear interpolation plays an important role in modern signal and image processing applications due to its simple implementation and low computational cost. Plonka [1,2] has extensively studied the effect of the shifted interpolation nodes on the approximation error in the B-spline interpolation. According to the theoretical analysis the shift  $\tau = 0$  yields the minimal error norm. Blu *et al.* [3] applied the same idea to the piecewise-linear interpolation. On the contrary, they observed that interpolation error diminishes if the sampling knots are shifted by a fixed amount. Through the theoretical considerations Blu *et al.* [3] determined the optimal shift of the knots is close to  $\tau = 1/5$  of the distance between the knots.

In this work we consider a modification of the linear interpolation. In standard linear interpolation the kernel is the second order B-spline  $\beta_2(t)$ . Instead of shifting the sampling knots we introduce the shifted linear interpolation (SLI), where the B-spline kernel has been time-shifted by a fraction  $\Delta$  of the sampling period. We describe the z-transform SLI filter for efficient implementation of the shifted B-spline kernel  $\beta_2(t-\Delta)$  especially for VLSI and microprocessor circuits.

## 2. Theoretical Considerations

### 2.1. B-Spline Signal Processing

This work was supported by the National Technology Agency of Finland (TEKES).

B-spline approximation of the continuous signal  $x(t)$  is based on the summation [4]

$$x(t) = \sum_k c[k] \beta_p(t-k) \quad (1)$$

where  $c[k]$  are the scale coefficients. The B-spline  $\beta_p(t)$  is constructed by convolving an indicator function  $\beta[t]$  by itself

$$\beta_p(t) = \underbrace{\beta(t) * \beta(t) * \dots * \beta(t)}_{p \text{ times}} \quad (2)$$

where

$$\beta(t) = \begin{cases} 1 & 0 \leq t \leq 1 \\ 0 & \text{elsewhere} \end{cases} \quad (3)$$

The linear interpolation is obtained in the case  $p = 2$

$$x(t) = \sum_k c[k] \beta_2(t-k) \quad (4)$$

where  $\beta_2(t)$  is the triangle function **Figure 1** defined as

$$\beta_2(t) = \begin{cases} t & \text{for } 0 \leq t < 1 \\ 2-t & \text{for } 1 \leq t \leq 2 \\ 0 & \text{elsewhere} \end{cases} \quad (5)$$

### 2.2. Shifted Linear Interpolation

Let us consider the approximation of the continuous-time signal  $x(t)$  by a summation of the shifted B-spline  $\beta_2(t-\Delta)$

$$x(t) = \sum_k c[\Delta, k] \beta_2(t-\Delta-k) \quad (6)$$

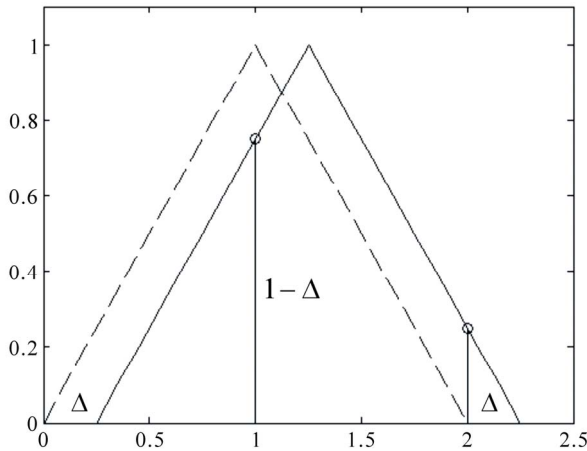


Figure 1. The shifted B-spline  $\beta_2(t-\Delta)$ .

where  $c[\Delta, k], \Delta \in [0, 1]$  is the scale sequence for the time-shifted kernel. In the sequel we call (6) the shifted linear interpolation (SLI). Usually the data points are known in discrete time intervals  $t = nT (n = 0, 1, 2, \dots)$ . In the  $z$ -transform domain we obtain

$$\begin{aligned} Z\{x[n]\} &= X(z) = C(\Delta, z)Z\{\beta_2(t+\Delta)\} \\ &= C(\Delta, z)[(1-\Delta) + \Delta z^{-1}] \end{aligned} \quad (7)$$

The SLI filter is now obtained from (6) as

$$Z\{x[n-d]\} = C(\Delta, z)[(1-\Delta-d) + (\Delta+d)z^{-1}] \quad (8)$$

By inserting the scale sequence  $C(\Delta, z)$  from (7) we finally have

$$\begin{aligned} Z\{x[n-d]\} &= \frac{(1-\Delta-d) + (\Delta+d)z^{-1}}{1-\Delta + \Delta z^{-1}} X(z) \\ &= SLI(d, \Delta, z) X(z) \end{aligned} \quad (9)$$

where  $SLI(d, \Delta, z)$  denotes the SLI filter.

### 2.3. Inverse SLI Filter

In some applications it is required that the original signal has to be reconstructed from the delayed one. In the general case the SLI filter (9) has no stable inverse since the roots of the nominator may lie outside the unit circle. For  $\Delta = 0.25$  the roots are inside the unit circle in the range  $0 \leq d \leq 0.25$ . The inverse SLI filter can be realized by the inverse filtering procedure described in Appendix I.

## 3. Experimental

### 3.1. Experimental Verification of the SLI Algorithm

We tested experimentally the quality of the SLI filter for a variety of synthetic signals with varying frequency components. The interpolation error was defined as the absolute difference between the analytically calculated and interpolated signal:  $e(t) = |x(t) - x_{int}(t)|$ . The time-shift of the interpolation kernel was altered in the range  $0 \leq \Delta \leq 1$ . The interpolation error was strongly related to the time-shift. Irrespective of the type of the signal the minimum interpolation error was obtained in every case at  $\Delta = 0.25$ . Figure 2 shows an example of the interpolation error versus  $\Delta$  for signal  $\sin(0.1t)$ , when  $d = 0.5$ . The worst case is the linear interpolation

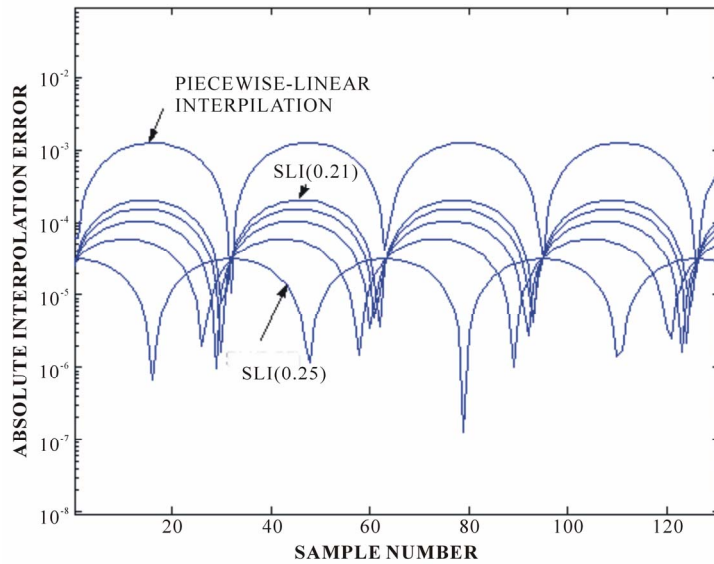
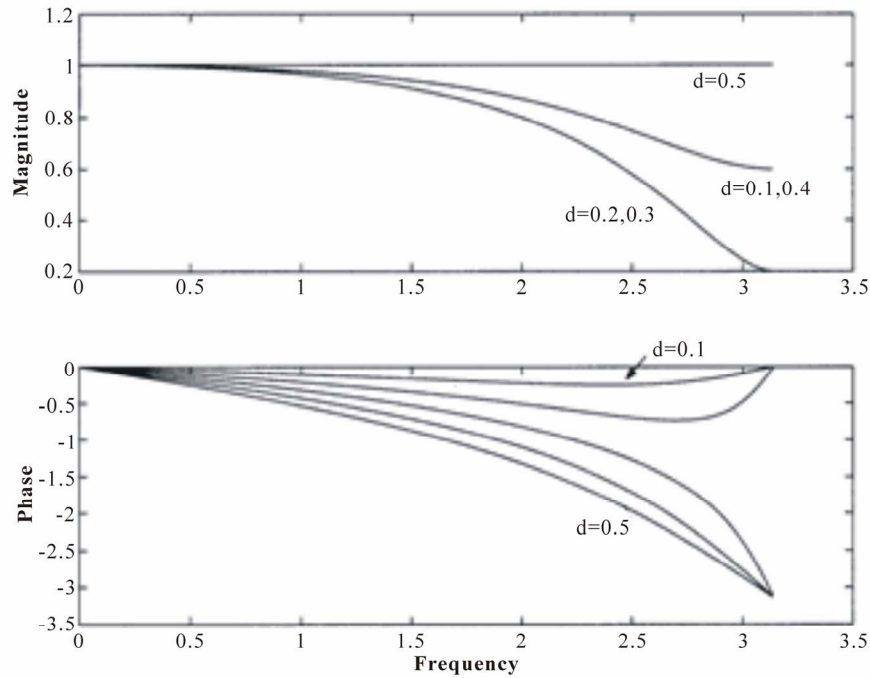


Figure 2. Interpolation error of function  $\sin(0.1n)$  in the case  $d = 0.5$  for piecewise-linear interpolation ( $\Delta = 0$ ) and for SLI filter from  $\Delta = 0.21$  to  $\Delta = 0.25$ .



**Figure 3.** The magnitude and phase responses of the SLI filter for  $d = 0.1, \dots, 0.5$  ( $\Delta = 0.25$ ).

( $\Delta = 0$ ). Then the interpolation error diminishes from  $\Delta = 0.21$  to  $\Delta = 0.25$ . At bigger  $\Delta$  values the interpolation error starts to increase.

The magnitude and phase response of the SLI filter is given in **Figure 3** for  $\Delta = 0.25$  and  $d = 0.1, \dots, 0.5$ . Magnitude response of the SLI filter decreases at higher frequency values, albeit in the case  $d = 0.5$  the magnitude is constant in the whole frequency range. The phase shift is linear in the frequency range  $0 \leq \omega \leq 1$ . For  $d > 0.5$  and  $\Delta = 0.25$  the magnitude response of the SLI filter turns to high-pass and tends to intensify the noise occurring in high frequency range. As a practical solution the interpolated signal can be time-reversed and  $1-d$  can be inserted instead of  $d$  in the SLI algorithm.

### 3.2. Half Delay Filter

An interesting application of the SLI is the half delay filter, which delays the signal half of the sampling period and interpolates the intermediate points between the evenly spaced knots. For  $\Delta = 0.25$  and  $d = 0.5$  we have

$$SLI(0.5, 0.25, z) = \frac{(1/3) + z^{-1}}{1 + (1/3)z^{-1}} \quad (10)$$

which is an all-pass filter having the unity magnitude response. The interpolation error of the half delayed filter (11) is especially low **Figure 2**. **Figure 4** shows the

phase response and the phase error compared with the ideal phase response.

### 3.3. Comparison with the Cubic Convolution Interpolation

The absolute interpolation error using the cubic convolution method [5] and the SLI-algorithm for signals  $x[n] = \sin(n)$  and  $x[n] = \sin(2n)$  are given in **Figures 5** and **6** for  $d = 0.5$  and  $\Delta = 0.25$ . At  $\omega = 1$  the cubic convolution interpolation performs slightly better, but at  $\omega = 2$  the quality of the SLI algorithm equals the cubic convolution.

## 4. Discussion

The linear interpolation has been used in many micro-processor and VLSI applications due to its simple and fast implementation. The new approaches based on higher degree B-spline interpolation kernels [6-8] suffer from computational complexity and usually require the time-reversed inverse filtering procedure, which is not possible to implement in real-time. In this work we examined the shifted linear interpolation, which outperforms the standard linear interpolation **Figure 2**. The SLI algorithm can be implemented via the z-transform filter (9) with integer coefficients. The interpolation error of the SLI filter is comparable to the more elaborate cubic convolution interpolation **Figures 5,6**, which is a standard image and signal processing tool. The computational complexity of the

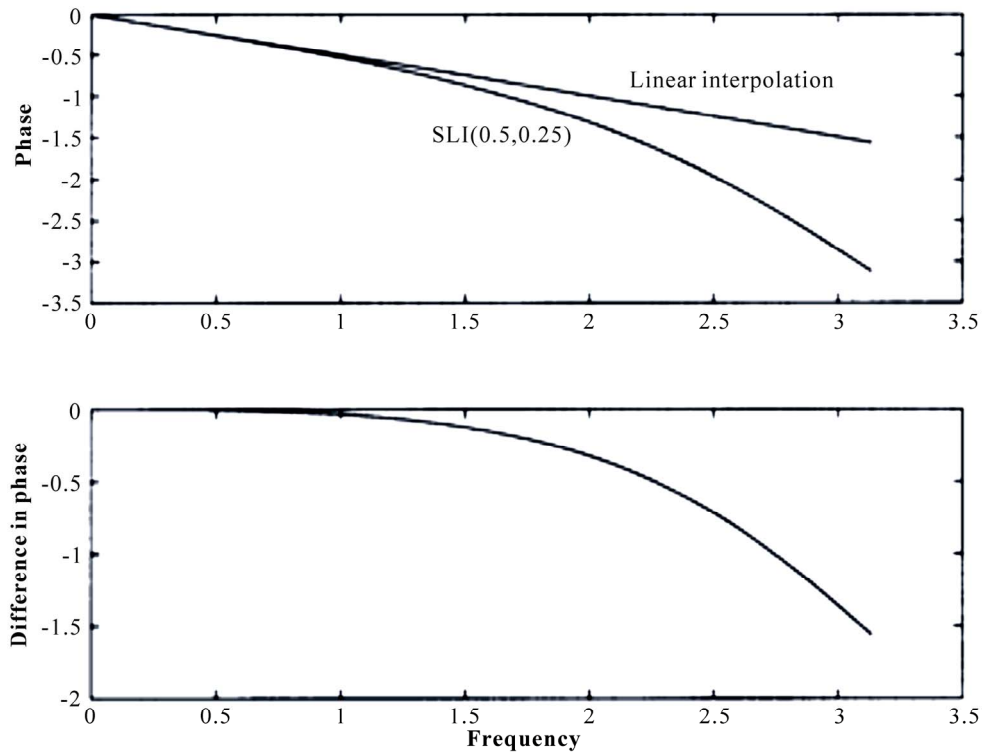


Figure 4. Comparison of the phase responses of the piecewise-linear interpolation and the SLI filter( $d = 0.5, \Delta = 0.25$ ).

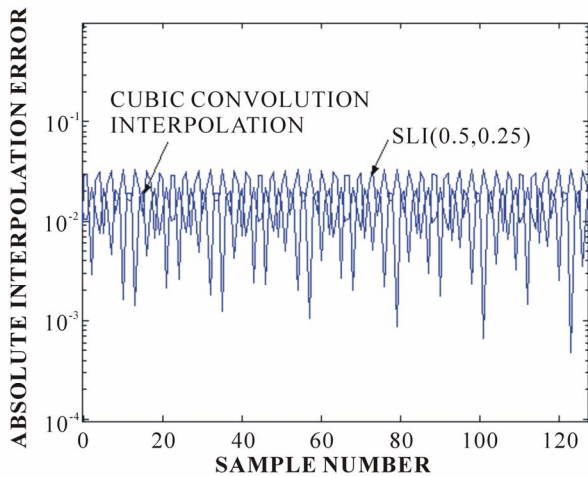


Figure 5. The absolute interpolation error yielded by the SLI algorithm ( $d = 0.5, \Delta = 0.25$ ) and the cubic convolution interpolation for the sinusoidal signal  $x[n] = \sin(n)$ .

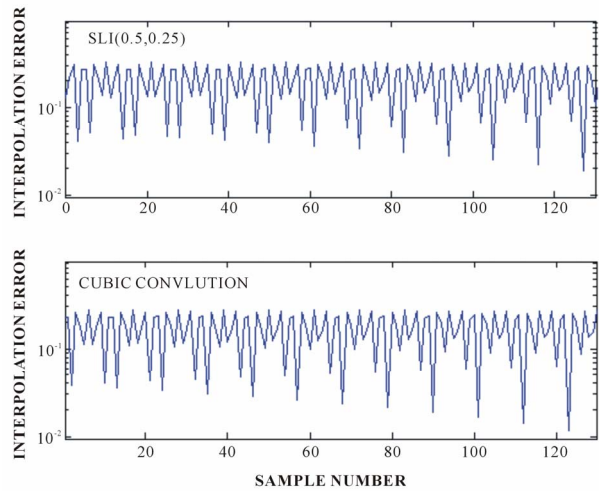


Figure 6. The absolute interpolation error yielded by the SLI algorithm ( $d = 0.5, \Delta = 0.25$ ) and the cubic convolution interpolation for the sinusoidal signal  $x[n] = \sin(2n)$ .

cubic convolution method is significantly higher compared with the SLI algorithm.

The interpolation function used as a kernel in the SLI algorithm correspond the B-spline order  $p = 2$  **Figure 1**. The B-spline  $\beta_2(t)$  has the z-transform  $(1-\Delta) + \Delta z^{-1}$ .

Probably the main reason for the effectiveness of the SLI comes from the two term z-transform, which allows the solution of the scale coefficient sequence  $C(\Delta, z)$  in (7). This leads to the definition of the SLI filter (9). As a sur-

prising experimental observation of the SLI algorithm is the strong dependence of the interpolation error on the time shift  $\Delta$ . Our comparisons using a variety of functions with different waveforms and frequency content showed that every time the interpolation error attains its minimum at  $\Delta = 0.25$ . Previously Plonka [1,2] has studied the B-spline interpolation with shifted nodes. In the theoretical analysis the shift parameter  $\tau = 0$  yielded the minimal error norm. Blu *et al.* [3] conducted theoretically the piecewise-linear interpolation error versus the shift  $\tau$  in the measurement knots. Their analysis predicted the optimal choice  $\tau = 0.21$ . Blu *et al.* [3] used a pre-filter to shift the measurement knots and then the standard piecewise linear interpolation for computation of the interpolated signal. An advantage in this procedure is that the prefiltered signal can be directly implemented by any existing linear interpolation hardware, while the implementation of the SLI filter needs a new hardware design.

An essential observation in this work is that the SLI filter has low-pass frequency response characteristics only in the range  $0 \leq d \leq 0.5$ . For  $d > 0.5$  the frequency response of the SLI filter turns to high-pass and intensifies the noise occurring in high frequency range. The noise interference for  $d > 0.5$  can be avoided by time reversing the interpolated signal. Then  $1-d$  can be inserted instead of  $d$  in the SLI algorithm.

A shortcoming in the SLI algorithm is the phase error occurring in the high frequency range compared with the linear interpolation. Fortunately, in most of the signal processing applications the signal is band limited and the high frequency band  $2 \leq \omega \leq \pi$  contains only some noise components.

An interesting observation is that for  $d = 0.5$  and  $\Delta = 0.25$  the half delay SLI algorithm reduces to the simple all-pass filter (10), which has the unity magnitude response and the phase response is close to the ideal phase response in the range  $0 \leq \omega \leq 1$ . If the signal bandwidth is limited to this range the SLI algorithm yields interpolation precision and accuracy, which is comparable to the more elaborate interpolation methods. The interpolation of intermediate points between the uniformly distributed knots is probably the most frequently used operation is image and signal processing. The half delay filters are also important tools in construction of shift invariant wavelet transform [9]. The half delay SLI filter (10) can be readily applied to replace the more elaborate half delay filters based on the Thiran filters [10]. For  $0 \leq d \leq 0.25$  and  $\Delta = 0.25$  the SLI filter has a stable inverse. If the original data points have to be reconstructed from the interpolated ones, the inverse SLI filter for higher values of  $d$  can be realized by using the

inverse filtering procedure (Appendix I).

## 5. Conclusions

In signal and image processing applications linear interpolation has been frequently used due to its simple implementation and low computational cost. In standard linear interpolation the kernel is the second order B-spline  $\beta_2(t)$ . We show in this work that the interpolation error can be remarkably reduced by adapting the time-shifted B-spline  $\beta_2(t - \Delta)$  as an interpolation kernel. By experimental tests we verify that the optimal shift is  $\Delta = 0.25$ . The shifted linear interpolation (SLI) algorithm can be effectively implemented by the z-transform filter in VLSI and microprocessor applications. The interpolation error of the SLI filter is comparable to the more elaborate cubic convolution interpolation. The future research goal is to extend the time-shifted interpolation kernel to higher order B-splines.

**Appendix I.** Implementation of the inverse filter  $SLI^{-1}(d, \Delta, z)$  for  $\Delta = 0.25$  and  $d > 0.25$ .

By denoting the interpolated signal by  $Y(z) = Z\{x[n-d]\}$  the inverse filter  $SLI^{-1}(d, \Delta, z)$  is

$$SLI^{-1}(d, \Delta, z) = \frac{1 - \Delta + \Delta z^{-1}}{(1 - \Delta - d) + (\Delta + d)z^{-1}} = \frac{X(z)}{Y(z)} \quad (11)$$

The inverse filter  $SLI^{-1}(d, \tau, z)$  is not stable which is not stable for  $\Delta = 0.25$  and  $d > 0.25$ , since the root in the denominator is outside the unit circle. However, the inverse filter can be implemented by the following inverse filtering procedure. First we replace  $z$  by  $z^{-1}$

$$SLI^{-1}(d, \Delta, z^{-1}) = \frac{X(z^{-1})}{Y(z^{-1})} = \frac{\Delta + (1 - \Delta)z^{-1}}{(\Delta + d) + (1 - \Delta - d)z^{-1}} \quad (12)$$

The inverse filter  $SLI^{-1}(d, \Delta, z^{-1})$  is stable for  $\Delta = 0.25$  and  $d > 0.25$ . The  $X(z^{-1})$  and  $Y(z^{-1})$  are the z-transforms of the time reversed signal  $x[N-n]$  and the interpolated signal  $y[N-n]$ . The following Matlab program **ISLI.m** demonstrates the computation procedure.

```
function x = ISLI(y,d,delta)
y = y(end:-1:1);
x = filter([delta 1-delta],[delta+d 1-delta-d],y);
x = x(end:-1:1);
```

## REFERENCES

- [1] G. Plonka, "Periodic Spline Interpolation with Shifted Nodes," *Journal of Approximation Theory*, Vol. 76, No. 1, 1994, pp. 1-20.
- [2] G. Plonka, "Optimal Shift Parameters for Periodic Spline Interpolation," *Numerical Algorithms*, Vol. 6, No. 2, 1994,

- pp. 297-316.
- [3] I. J. Schoenberg, "Contribution to the Problem of Approximation of Equidistant Data by Analytic Functions," *Quarterly of Applied Mathematics*, Vol. 4, No. 1, 1946, pp. 45-99, 112-141.
  - [4] T. Blu, P. Thevenaz and M. Unser, "Linear interpolation revitalized", *IEEE Trans. Image Process.* Vol. 13, No. 5, pp. 710-719, May 2004.
  - [5] R. G. Kay, "Cubic Convolution Interpolation for Digital Image Processing," *IEEE Transactions on Acoustics, Speech, Signal Processing*, Vol. 29, No. 6, December 1981, pp. 1153-1160.
  - [6] P. Thevenaz, T. Blu and M. Unser, "Interpolation Revisited," *IEEE Transactions on Medical Imaging*, Vol. 19, No. 7, July 2000, pp. 739-758.
  - [7] J. T. Olkkonen and H. Olkkonen, "Fractional Delay Filter Based on B-Spline Transform," *IEEE Signal Processing Letters*, Vol. 14, No. 2, February 2007, pp. 97-100.
  - [8] J. T. Olkkonen and H. Olkkonen, "Fractional Time-Shift B-Spline Filter," *IEEE Signal Processing Letters*, Vol. 14, No. 10, October 2007, pp. 688-691.
  - [9] H. Olkkonen and J. T. Olkkonen, "Half Delay B-Spline Filter for Construction of Shift Invariant Wavelet Transform," *IEEE Transactions on Circuits and Systems II*, Vol. 54, No. 7, July 2007, pp. 611-615.
  - [10] I. W. Selesnick, "The Design of Approximate Hilbert Transform Pairs of Wavelet Bases," *IEEE Transactions on Signal Processing*, Vol. 50, No. 5, May 2002, pp. 1144-1152.