

Recovering Periodically Spaced Missing Samples

“DSP Tips and Tricks” introduces practical design and implementation signal processing algorithms that you may wish to incorporate into your designs. We welcome readers to submit their contributions to the Associate Editors Rick Lyons (r.lyons@ieee.org) or Britt Rorabaugh (dspboss@aol.com).

There are times when we need to process time-domain sequences that have been corrupted due to missing samples of a desired signal sequence. This article presents a powerful technique to recover (compute) periodically spaced missing samples of a time-domain sequence.

MISSING SAMPLES RECOVERY PROCESS

To clarify our missing samples problem, Figure 1 shows a corrupted $x_c[n]$ time sequence, represented by the black dots. What we desire is an $x_o[n]$ sequence representing correct sampling of the continuous $x_o(t)$ signal at the same sample rate as $x_c[n]$. We represent the missing samples of $x_o[n]$ which are shown by the white circles, using zero-valued samples in $x_c[n]$. We call the black-dot samples of $x_c[n]$ riding on the $x_o(t)$ curve “available samples,” and we refer to $x_c[n]$ ’s zero-valued black-dot samples as “missing samples.”

As shown in Figure 1, the available and missing samples in $x_c[n]$ form a periodic pattern, meaning that $x_c[n]$ contains a repetitive pattern of two available samples, one missing sample, one

available sample, and one missing sample. Each repeating T -length period contains $A = 3$ available samples and $M = 2$ missing samples where, of course, $T = A + M = 5$.

Let us assume that $x_o(t)$ is bandlimited to B Hz, and that a sampling rate of F_s Hz was used to generate the sequence $x_o[n]$, that is, $x_o[n] = x_o(n/F_s)$. Again, our missing sample recovery problem is to compute the desired $x_o[n]$ signal from the corrupted $x_c[n]$ signal. We can solve our missing sample recovery problem if the following two conditions are satisfied:

- $x_c[n]$ contains a T -length periodic pattern of A available samples and M missing samples.
- Frequency B is limited to

$$B \leq \frac{A}{T} \cdot \frac{F_s}{2}. \quad (1)$$

Next we explain how to recover the missing samples using the above example of a periodic pattern where $A = 3$ and $M = 2$. (Our missing samples recovery scheme can be applied to any integer values of A and M .)

RECOVERY PROCESS

When the above two conditions are satisfied, our missing sample recovery technique begins by defining two sam-

pling time vectors for the available and missing samples that reside within the $T = 5$ period. For our example in Figure 1, we define the available sample time vector as

$$t_A = [0, 1, 3],$$

the length of which is $A = 3$, and define the missing sample time vector as

$$t_M = [2, 4],$$

the length of which is $M = 2$.

Next we partition $x_c[n]$ into A subsequences of available samples based on the t_A time vector elements. The first subsequence of available samples is

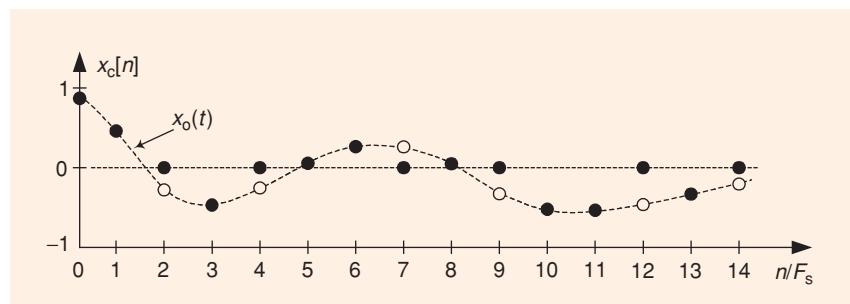
$$x_o[t_A[1] + pT] = x_o[0], x_o[5], \\ x_o[10], x_o[15], \text{etc.}$$

The second subsequence of available samples is

$$x_o[t_A[2] + pT] = x_o[1], x_o[6], \\ x_o[11], x_o[16], \text{etc.}$$

And finally, the third subsequence of available samples is

$$x_o[t_A[3] + pT] = x_o[3], x_o[8], \\ x_o[13], x_o[18], \text{etc.}$$



[FIG1] Corrupted $x_c[n]$ time sequence ($A = 3$, $M = 2$, and $T = 5$).

where the integer p sequence is 0, 1, 2, 3, etc. (Because $x_0[n]$ can be infinite in extent, index p could range over both positive and negative integers. For convenience, we arbitrarily chose to define $x_0[n]$ as starting at index $p = 0$, with $x_0[n] = 0$ for $n < 0$.)

Using the above definitions, we can recover our missing samples in the form of $M = 2$ subsequences defined as

$$x_0[t_M[1] + pT] = x_0[2], x_0[7], \\ x_0[12], x_0[17], \text{ etc.}$$

and

$$x_0[t_M[2] + pT] = x_0[4], x_0[9], \\ x_0[14], x_0[19], \text{ etc.}$$

We implement our missing sample recovery technique by applying those available sample subsequences to the $M = 2$ reconstruction filterbanks shown in Figure 2. Each filterbank contains $A = 3$ tapped-delay line convolutional subfilters. The output sequences of the reconstruction filter-

THIS ARTICLE PRESENTS A POWERFUL TECHNIQUE TO RECOVER (COMPUTE) PERIODICALLY SPACED MISSING SAMPLES OF A TIME-DOMAIN SEQUENCE.

banks are the desired $x_0[n]$ samples which we set out to recover.

COMPUTING FILTER COEFFICIENTS

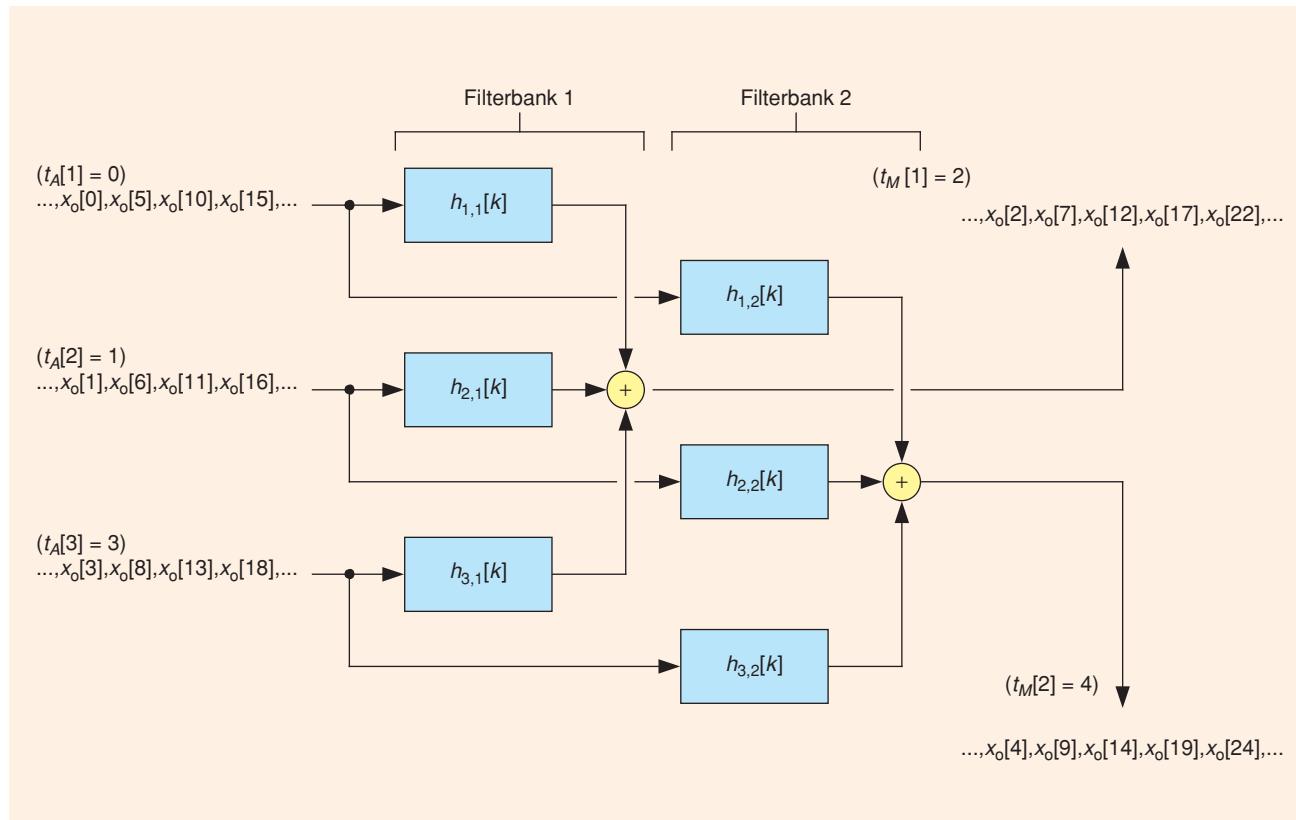
The subfilters used in the filterbanks are all variations of the well-known ideal fractional delay filter, which has an impulse response equal to

$$D_\Delta[k] = \text{sinc}(\Delta + k), \\ k = \dots, -1, 0, 1, \dots, \quad (2)$$

where the delay parameter Δ is in the range of $-1 < \Delta < 1$. Just like the sampled-sinc low-pass filter, the fractional delay filter is ideal in the sense

that perfect implementation of this filter (for example, on a computer) is impossible. However, the approximation of the fractional delay filter has been studied extensively, and many approximation methods are known [1]. As always, the right choice of the fractional delay filter approximation depends strongly on the application.

We now proceed with the example in Figure 1 by using the simple but effective windowing approach to approximate the fractional delay. This results in a tapped-delay line subfilter with a finite number of coefficients [a finite impulse response (FIR) filter]. For windowing the impulse response, the window must also be delayed in time by the fractional delay value Δ . We assume a definition of the window function $w(t)$ that is centered at 0, nonzero on the interval $[-1, 1]$, and zero elsewhere. Any window that is defined by sampling a function with a continuous time parameter t can be used (this includes, for example, Hanning or Hamming windows). Using the indexing variable $k = -K, -K + 1, \dots, 1, 0$,



[FIG2] Reconstruction filters for missing sample recovery.

$1, \dots, K-1$, K and (2), the impulse response of the windowed fractional delay filter is given by

$$d_{\Delta}[k] = D_{\Delta}[k] \cdot w\left(\frac{\Delta+k}{K}\right). \quad (3)$$

This makes computing the coefficients of the convolutional subfilters straightforward.

Next we define two indices that identify the individual subfilters. Because there are A subfilters in each filterbank, we will define a subfilter index of $a = 1, 2, \dots, A$ and a filterbank index of $m = 1, 2, \dots, M$. Using those indices, the coefficients of a single subfilter are represented by $h_{a,m}[k]$. For example, using this two-dimensional subscripted index notation, the coefficients of the third subfilter in the second filterbank are denoted as $h_{3,2}[k]$. Given this notation, the coefficients of the subfilters are defined by

$$h_{a,m}[k] = G_{a,m} \cdot d_{\Delta_{a,m}}[k] \cdot \cos(\pi k(A-1)), \quad (4)$$

where $d_{\Delta_{a,m}}[k]$ is given by (3) with Δ replaced by $\Delta_{a,m}$,

$$G_{a,m} = \prod_{q=1, q \neq a}^A \frac{\sin\left(\pi \cdot \frac{t_M[m] - t_A[q]}{T}\right)}{\sin\left(\pi \cdot \frac{t_A[q] - t_A[q]}{T}\right)} \quad (5)$$

$$\Delta_{a,m} = \frac{t_M[m] - t_A[q]}{T}. \quad (6)$$

Integer k ranges from $-K \leq k \leq K$.

DISCUSSION

We note that the process of removing the M missing samples from $x_0[n]$ is a form of decimation. Any noise components in $x_0[n]$ above the frequency B in (1) are aliased into the signal band by this decimation. Similarly, the recovery process using the filterbanks is a general form of upsampling and interpolation and generates a signal that is bandlimit-

WE PRESENT A NOVEL METHOD TO RECOVER PERIODICALLY MISSING SAMPLES FROM APPROPRIATELY BANDLIMITED SIGNALS.

ed to B Hz, even if the original signal (which we are trying to recover) was not bandlimited. Some noise (images caused by upsampling) may appear above B Hz due to the finite attenuation of the FIR filters.

Although similar methods for recovering missing samples have been investigated before, to the best of our knowledge the powerful recovery technique presented here is new. The idea of using filterbanks to resample recurrent nonuniformly sampled signals is discussed in [2]; however, instead of recovering missing samples in oversampled sequences, those authors develop a filterbank that resamples the recurrent nonuniform samples to uniform samples spaced at a sampling period equal to $1/(2B)$. This can also be achieved using the method described in this article, by placing the “missing samples” at uniform intervals in one period.

The special case of recovery of an $M = 1$ missing sample per recurrence period was solved in [3] with a single symmetric FIR filter, also under the $x_0(t)$ signal bandwidth limitations in (1).

CONCLUSIONS

We presented a novel method to recover periodically missing samples from appropriately bandlimited signals. We used filterbanks with fractional delay filters as building blocks, and described a windowed-sinc method to approximate the fractional delays with FIR subfilters. Because the implementation of filterbanks and the approximation of fractional delay filters are well-studied problems, this makes the proposed recovery method flexible and easy to implement.

Space limitations do not allow us to present the complete derivation of the subfilter coefficient expression in (4). However, to aid in the reader’s understanding and software modeling, a derivation (4), MATLAB code to compute the reconstruction filterbanks’ coefficients using the windowing method for the example in Figure 1, and sample values of the sequences in (2) through (6) for the example in Figure 1 are available on the *IEEE Signal Processing Magazine* Web site included in the references below.

ACKNOWLEDGMENTS

Many thanks to Rick Lyons for his suggestions on content and generous help with the formatting of this article. I also acknowledge the newsgroup *comp.dsp*, which spawned the idea put forth in this article with a thread about reconstructing equally spaced missing samples.

AUTHOR

Andor Bariska (andor.bariska@zhwin.ch) is a research associate with the Institute of Data Analysis and Process Design of the Zurich University of Applied Sciences Winterthur, Switzerland. Before joining them, he spent 12 years with Weiss Engineering Ltd., developing and implementing DSP algorithms for their audio processing products. He enjoys living with his wife, three-year-old daughter, and a cat in Zurich.

REFERENCES

More details, software, and data samples are available at <http://apollo.ee.columbia.edu/spm/?i=external/tipsandtricks>.

- [1] T. Laakso, V. Välimäki, M. Karjalainen, U. Laine, “Splitting the unit delay,” *IEEE Signal Processing Mag.*, vol. 13, no. 1, pp. 30–60, Jan. 1996 [Software Online]. Available: <http://www.acoustics.hut.fi/software/fdtools>
- [2] Y. Eldar and A. Oppenheim, “Filterbank reconstruction of bandlimited signals from nonuniform and generalized samples,” *IEEE Trans. Signal Processing*, vol. 48, no. 10, pp. 2864–2875, Oct. 2000.
- [3] R. Adams, “Nonuniform sampling of audio signals,” *J. Audio Eng. Soc.*, vol. 40, no. 11, pp. 886–894, Nov. 1992.

