Interpolation in Digital Modems—Part I: Fundamentals

Floyd M. Gardner, *Fellow, IEEE*

***Abstract*— Timing adjustment in a digital modem must be performed by interpolation if sampling is not synchronized to the data symbols. This paper describes the fundamental equation for interpolation, proposes a method for control, and outlines the signal-processing characteristics appropriate to an interpolator. The material combines a review of previously known topics, presentation of new results, and a tutorial exposition of the subject.**

A companion paper will treat performance and implementation.

I. Introduction

TIMING in a data receiver must be synchronized to the symbols of the incoming data signal. In analog-implemented modems, synchronization typically is performed by a feedback loop that adjusts the phase of a local clock, or by a feedforward arrangement that regenerates a timing wave from the incoming signal. The local clock or the timing wave is used to sample (or *strobe*) the filtered output of the modem, once per symbol interval. Message data are recovered from the strobes. Timing of the strobes is adjusted for optimum detection of the symbols.

Implementation of the modem by digital techniques (a topic of intense present activity) introduces sampling of the signal. In some circumstances, the sampling can be synchronized to the symbol rate of the incoming signal; see Fig. 1(a) and (b). Timing in a synchronously sampled modem can be recovered in much the same ways that are familiar from analog practice.

In other circumstances, the sampling cannot be synchronized to the incoming signal. Examples include 1) digital processing of unsynchronized frequency-multiplexed signals, or 2) nonsynchronized digital capture and subsequent postprocessing of a signal. For one reason or another, the sampling clock must remain independent of the symbol timing. See Fig. 1(c) for a nonsynchronized-sampling configuration.

How is receiver timing to be adjusted, by digital methods, when it is not possible to alter the sampling clock? One answer is to *interpolate* among the nonsynchronized samples in such manner as to produce the correct strobe values at the modem output—the same strobe values that would occur if the original sampling had been synchronized to the symbols.

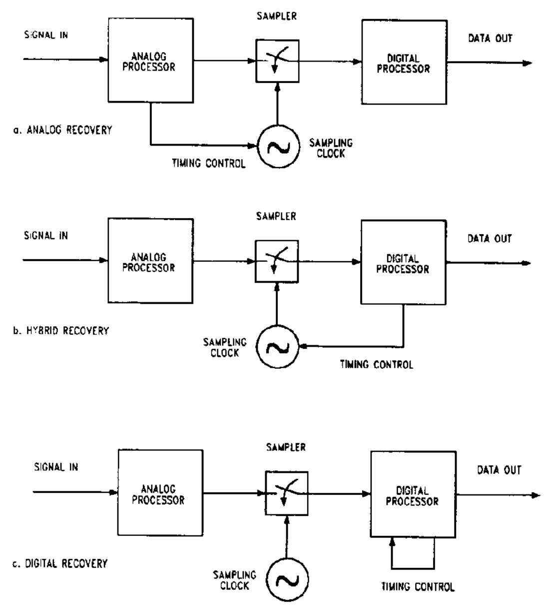


Fig. 1. Timing recovery methods.

Interpolation is a timing-adjustment operation on the signal, not on a local clock or timing wave. In this respect, it is radically different from timing adjustment in the better-known analog modems. Of all the operations in a digitally implemented modem, interpolation is perhaps the one with the least resemblance to established analog methods.

Several issues arise as follows.

—What mathematical model of interpolation can be devised?

—How is interpolation to be controlled?

—What characteristics are desirable in an interpolator for modems?

—How is the interpolator to be implemented?

—What performance can be obtained? How large is the computing burden?

—What conceptual model is appropriate for interpolation?

These are the matters treated in this paper and its companion [1]. The first three issues are addressed here in Part I, and the last three in Part II [1]. Attention is concentrated on *high-speed* methods, defined by a hardware-imposed constraint that no clock frequency can greatly exceed the signal sample rate.

Paper approved by the Editor for Synchronization and Optical Detection of the IEEE Communication Society. Manuscript received December 6, 1990; revised May 23, 1991. This work was supported under Contract 8022/88/NL/DG by the European Space Agency, Noordwijk, The Netherlands. This paper was presented at the Second International Workshop on Digital Signal Processing Techniques Applied to Space Communications (DSP'90), Politecnico di Torino, Turin, Italy, September 24–25, 1990.

The author is with Gardner Research Company, Palo Alto, CA 94301.

IEEE Log Number 9208042.

II. Background

Interpolation as a Digital Signal Processing (DSP) operation has been covered extensively in the literature; excellent examples and further references may be found in [2] and [3]. By contrast, the role of interpolation in timing adjustment has had comparatively meager attention [2, ch. 6], [4], [5]. In fact, these latter references do not speak of "interpolation", but of "digital phase shifting" [2, ch. 6] and [4], or of "sampling-rate conversion" [2, ch. 2] and [5].

It will be seen presently that the process of timing adjustment includes substantially more than interpolation alone and that "rate conversion" is a more accurate label. Nonetheless, we will apply the term "interpolation" to denote all of the processes that are involved in adjustment of timing.

The term "interpolation" to describe the entire timing-adjustment process appears to have been published first by a group at the Technical University of Aachen [6], [7]. The term is also used by Bingham [8, p. 167].

In light of the extensive DSP literature on interpolation, and of the large number of digitally implemented modems that have been built for voice-frequency telephone-line service, how is it that the literature on digital timing adjustment is so sparse?

Authors in the established DSP literature almost invariably restrict themselves to sampling-rate conversion by a rational factor, which can be modeled as a cascade of interpolation and decimation, each by integer ratios. Thus, the output is synchronized to the input.

But the inherent problem of fully digital timing adjustment is that the signal sampling is not synchronized to the symbol timing; the two rates are *incommensurate* and the sample times never coincide exactly with desired strobe times. Recognition of incommensurability is vital to understanding the timing-adjustment problem.

Limitations of the DSP literature aside, why didn't the timing adjustment problem arise more clearly in the design of digitally implemented telephone-line modems? The answer is that it indeed did arise, and was solved by the adaptive equalizers that play so large a role in those modems. Besides correcting for transmission dispersion, an equalizer almost incidentally also corrects the timing. For that reason, timing adjustment itself does not appear as a widely recognized, distinct problem in the context of telephone-line modems.

Digital implementation is now coming to higher speed communications links which do not require adaptive equalization. The need for digital timing adjustment must be faced by itself, without embedding it inside an equalizer.

III. Model

A. Timing Loop

Consider the feedback timing recovery of Fig. 2. (Feedforward interpolation is also feasible, but not considered here.) A time-continuous, PAM signal $x(t)$ is received. Symbol pulses in $x(t)$ are uniformly spaced at intervals T. For simplicity, $x(t)$ is assumed to be a real, baseband signal, but those restrictions can be removed without difficulty.

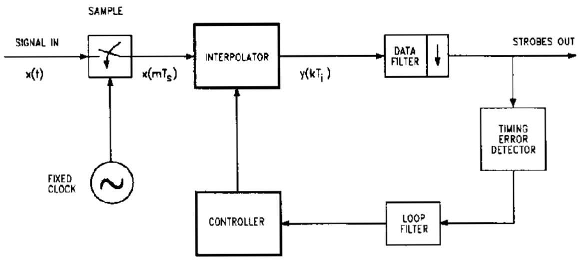


Fig. 2. Elements of digital timing recovery.

Assume $x(t)$ to be bandlimited so that it can be sampled at a rate $1/T_s$ without aliasing. If $x(t)$ is not adequately bandlimited, aliasing will introduce distortion that causes a performance penalty. Interpolation is not an appropriate technique to be applied to wide-band signals.

Samples $x(mT_s) = x(m)$ are taken at uniform intervals T_s. The ratio T/T_s is assumed to be irrational, as indeed will be true in all practical situations where the symbol timing is derived from a source that is independent of the sampling clock. These signal samples are applied to the interpolator, which computes *interpolants*, designated $y(kT_i) = y(k)$ at intervals T_i. We assume that $T_i = T/K$ where K is a small integer.

The data filter employs the interpolants to compute the strobes that are used for data and timing recovery.

In the sequel, the interval T_i between interpolants is treated as a constant, for simplicity of explanation. A practical modem must be able to adjust the interval so that the strobes can be brought into synchronism with the data symbols of the signal; thus, the interpolation interval cannot be constant.

All elements within the feedback loop contribute to the synchronization process. Timing error is measured by the timing error detector and filtered in the loop filter, whose output drives the controller. The interpolator obtains instructions for its computations from the controller.

This paper concentrates on the interpolator and controller alone, with little or no consideration of the data filter, the timing error detector, or the loop filter. One example of digital timing-error detectors may be found in [9], which also has references to other examples. An illustrative loop design and simulation may be found in Part II [1].

The data filter is shown within the feedback loop, after the interpolator. That placement is not essential; the data filter could be outside of the loop, prior to the interpolator. A data filter inside the feedback loop introduces delay, with adverse influence on loop stability.

Post placement may be advantageous when the data filter is more complicated than the interpolator—a likely situation—and when a relatively high sampling rate is employed for interpolation. With postplacement, the data filter can decimate its output to the required strobe rate (just one or two samples per symbol) and thereby save on computing burden. If the data filter is placed before the interpolator, then the sample rate out of the data filter must be maintained high enough to avoid aliasing. On the other hand, simulation results [1] indicate that quite modest sampling rates provide excellent results, even

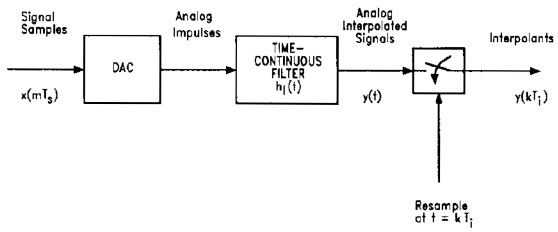


Fig. 3. Rate conversion with time-continuous filter.

with very simple interpolators. Thus, post placement may not often be necessary.

B. Interpolator Equations

To derive a model for the interpolator, we recapitulate the fundamental development of Crochiere and Rabiner [2, ch. 2]. The same basis underlies the adaptive rate convertor in [5].

Refer to Fig. 3, which shows a fictitious, hybrid analog/digital method of rate conversion. Convert the samples to a sequence of weighted analog impulses, which are applied to a time-continuous, analog, *interpolating filter* with impulse response $h_I(t)$. The time-continuous output of the filter is

$$y(t) = \sum_m x(m) h_I(t - mT_s). \tag{1}$$

Observe that $y(t) \neq x(t)$. There is no attempt, and no need to recover the original waveform, contrary to most conventional interpolation. Since a modem is required to perform filtering of signals there is no reason why some of the filtering cannot be included in the interpolator.

Now resample $y(t)$ at time instants $t = kT_i$ where T_i is synchronized with the signal symbols. In general, T_i/T_s is irrational; the sampling and symbol rates are incommensurate.

The new samples—the *interpolants*—are represented by

$$y(kT_i) = \sum_m x(mT_s) h_I(kT_i - mT_s). \tag{2}$$

Although the model includes a fictitious DAC and a fictitious analog filter, the interpolants in (2) can be computed entirely digitally from knowledge of: 1) the input sequence $\{x(m)\}$, 2) the impulse response $h_I(t)$ of the interpolating filter, and 3) the time instants mT_s and kT_i of the input and output samples. These digitally computed interpolants have *identically* the same values as if the analog operations had been performed.

A more useful format is obtained by rearranging the indexing in (2). Recognizing that m is a signal index, define a *filter index*

$$i = \text{int}[kT_i/T_s] - m \tag{3}$$

where int$[z]$ means largest integer not exceeding z. Also, define a *basepoint index*

$$m_k = \text{int}[kT_i/T_s] \tag{4}$$

and a *fractional interval*

$$\mu_k = kT_i/T_s - m_k \tag{5}$$

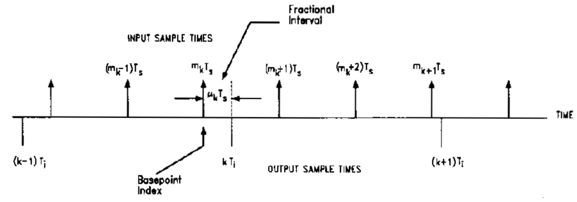


Fig. 4. Sample time relations.

where $0 \le \mu_k < 1$. Timing relations are illustrated in Fig. 4.

Function arguments in (2) become $m = m_k - i$ and $(kT_i - mT_s) = (i + \mu_k)T_s$, and the interpolant is computed at time $kT_i = (m_k + \mu_k)T_s$. Equation (2) can be rewritten as

$$\begin{aligned} y(kT_i) &= y[(m_k + \mu_k)T_s] \\ &= \sum_{i=I_1}^{I_2} x[(m_k - i)T_s] h_I[(i + \mu_k)T_s]. \end{aligned} \tag{6}$$

Equation (6) is the foundation of digital interpolation in modems.

If the interpolating filter has finite impulse response (FIR), then I_1 and I_2 are fixed, finite numbers and the digital filter actually used for computing the interpolants has $I = I_2 - I_1 + 1$ taps.

At this point, most DSP accounts of interpolation assume that the ratio T_i/T_s is rational. No such assumption will be made here; real-world symbol rates are almost never synchronous with independent, fixed-rate sampling clocks. Assuming a commensurate ratio tends to obscure broader issues of control and implementation.

When T_i is incommensurate with T_s, the fractional interval μ_k will be irrational and will change for each interpolant. If determined to infinite precision, μ_k takes on an infinite number of values, which *never* repeat exactly. This behavior is contrary to that observed if T_i is assumed very nearly equal to T_s—if sampling is nearly synchronized. Then μ_k changes only very slowly; if μ_k is quantized, it might remain constant over many interpolations. If T_s were commensurate with T_i, but not equal, then μ_k would cyclicly repeat a finite set of values, when the timing loop is in equilibrium.

IV. Control

Fig. 5 presents the timing loop of Fig. 2 with expanded detail for the controller. The interpolator performs the computations of (6). The controller provides the interpolator with information needed to perform the computations. Other essential elements in the loop will not be treated here.

An interpolant is computed from (6) using I adjacent samples $x(m)$ of the signal and I samples of the impulse response $h_I(t)$ of the interpolating filter. The correct set of signal samples is identified by the basepoint index m_k and the correct set of filter samples is identified by the fractional interval μ_k. Thus, the controller of Fig. 5 is responsible for determining m_k and μ_k, and making that information available to the interpolator.

Once m_k and μ_k have been identified by the controller, then other elements load the selected signal and impulse-response

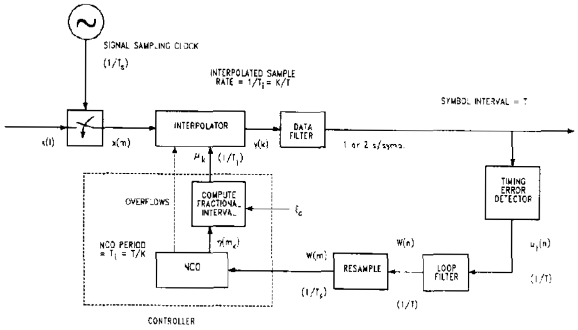


Fig. 5. Timing processor.

samples into the interpolation filter structure for computations. These loading operations are regarded as part of the filter implementation; some options are examined in Part II [1].

The necessary control can be provided by a number-controlled oscillator (NCO). Assume that the signal samples are uniformly clocked through a shift register at rate $1/T_s$ and that the NCO is clocked at a rate synchronized to $1/T_s$.

Provided that the interpolator is never called upon to perform upsampling, then the NCO clock period can be T_s. If upsampling is ever required, then a higher NCO clock rate is needed. Further discussion will concentrate on NCO clocking at rate $1/T_s$ (downsampling only); modifications needed to accommodate upsampling are readily devised once the basic principles are established.

The NCO is operated so that its average period is T_i. Recycling of the NCO register indicates that a new interpolant is to be computed, using the signal samples currently residing in the interpolator's shift register. Thus, basepoint index is identified by flagging the correct set of signal samples, rather than explicitly computing m_k.

A. Extraction of μ_k

Fractional interval μ_k can be calculated from the contents of the NCO's register upon recycling, as will now be explained.

Designate the NCO register contents computed at the mth clock tick as $\eta(m)$, and the NCO control word as $W(m)$. Then the NCO difference equation is

$$\eta(m) = [\eta(m-1) - W(m-1)]\text{mod-}1. \qquad (7)$$

(A decrementing NCO is employed because it affords a minor simplification in computation of μ_k as compared to an incrementing NCO.)

Control word $W(m)$ [a positive fraction] is adjusted by the timing-recovery loop so that output of the data filter is strobed at near-optimal timing. Under loop equilibrium conditions, $W(m)$ will be nearly constant. Contents of the NCO register (also a positive fraction) will be decremented by an amount $W(m)$ each T_s seconds and the register will underflow each $1/W(m)$ clock ticks, on average. Thus, the NCO period is $T_i = T_s/W(m)$ and so

$$W(m) \simeq \frac{T_s}{T_i}. \qquad (8)$$

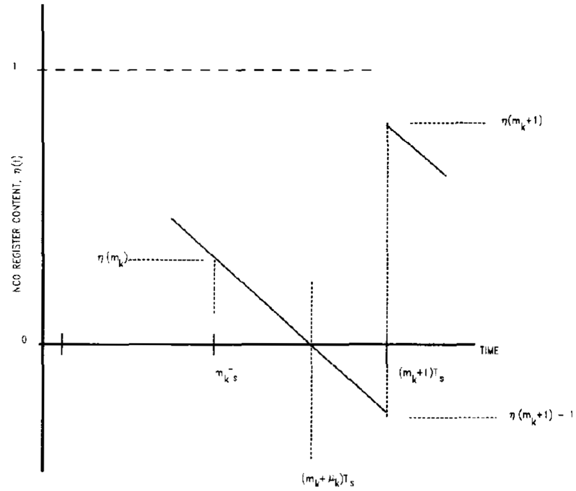


Fig. 6. NCO relations.

That is to say, $W(m)$ is the synchronizer's estimate of the average frequency of interpolation $1/T_i$, expressed relative to the sampling frequency $1/T_s$. The control word is an estimate because it is produced from filtering of multiple, noisy measurements of timing error.

To see how μ_k can be extracted from the NCO, refer to Fig. 6, which is a plot of (fictitious) time-continuous $\eta(t)$ versus continuous time. In the figure, $m_k T_s$ is the time of the sample-clock pulse immediately preceding the kth interpolation time $kT_i = (m_k + \mu_k)T_s$. NCO register contents decrease through zero at $t = kT_i$, and the zero crossing (underflow) becomes known at the next clock tick at time $(m_k+1)T_s$. Register contents $\eta(m_k)$ and $\eta(m_k+1)$ are available at the clock ticks.

From similar triangles in Fig. 6, it can be seen that

$$\frac{\mu_k T_s}{\eta(m_k)} = \frac{(1-\mu_k)T_s}{1-\eta(m_k+1)}$$

which can be solved for μ_k as

$$\mu_k = \frac{\eta(m_k)}{1-\eta(m_k+1)+\eta(m_k)} = \frac{\eta(m_k)}{W(m_k)}. \qquad (9)$$

An estimate for μ_k can be obtained by performing the indicated division of the two numbers $\eta(m_k)$ and $W(m_k)$ that are both available from the NCO. [Equation (9) is an estimate of the exact μ_k because its constituents $W(m_k)$ and $\eta(m_k)$ are both estimates of the true frequency and phase.]

To avoid division, recognize that $1/W(m) \simeq T_i/T_s$; nominal value of this ratio is designated ξ_0. Although the exact T_i/T_s is unknown and irrational, the nominal value ξ_0, expressed to finite precision, can often be an excellent approximation to the true value. Therefore, the fractional interval can be approximated by

$$\mu_k \simeq \xi_0 \eta(m_k). \qquad (10)$$

Represent the deviation in ξ_0 from the true ratio of periods as $\Delta\xi$. This deviation causes a uniformly distributed error with standard deviation $\Delta\xi/(\xi_0\sqrt{12})$ in the calculated value of μ_k.

If the deviation of ξ_0 is too large, then a first order correction

$$\mu_k \simeq \xi_0 \eta(m_k)[2 - \xi_0 W(m_k - 1)] \tag{11}$$

reduces the standard deviation in μ_k to $\Delta\xi^2/(\xi_0^2\sqrt{12})$, again without requiring a division.

Timing errors arising from multiplying by the nominal ξ_0 using (10) instead of dividing by the exact $W(m)$ using (9) cannot accumulate; the feedback loop removes any constant error or trend.

B. Interpolation Jitter

Although the kth interpolation is computed for a time $kT_i = (m_k + \mu_k)T_s$, the interpolant is actually delivered coincident with a clock tick no earlier than $(m_k + 1)T_s$. Therefore, the output exhibits a timing jitter with peak-to-peak fluctuations of T_s, even if the sampling clock and received symbol rate are entirely jitter free.

Timing jitter may be inconsequential if the received data are consumed at the receiver location. A timing clock is provided by underflows of the NCO. Underflow marks give an indication of correct data clocking to any downstream devices because they have the same jitter as the data.

But often the data must be retransmitted over a synchronous link to a remote consumer. The underflow marks usually cannot be retransmitted along with the data. Unless jitter is removed before retransmission, the jitter will be transmitted on the data stream.

Auer [10] has pointed out that a near jitter-free clock can be retrieved from the NCO and used to reclock the data before retransmission. His scheme employs the contents $\eta(m)$ of the NCO register in a direct digital frequency synthesizer. At each NCO clock tick, the register contents are used to address a table of sines to produce a sample sin $2\pi\eta(m)$. These sine samples are applied to a D/A convertor and then filtered to yield an analog, low-jitter sinewave with frequency $1/T_i = K/T$, from which a symbol-rate clock can be derived.

C. Alternative Control Methods

An NCO is not the only possible control structure. An alternative, suggested by M. Moeneclaey, is described in Appendix A.

V. Filter Properties

What properties are desirable in the interpolating filter's impulse response $h_I(t)$ or equivalently, via Fourier transformation, in its transfer function $H_I(f)$? Take heed that the properties sought are those of the fictitious analog filter, despite the fact that all physical operations are performed digitally.

A. Duration of Impulse Response

In general, new filter coefficients [samples of $h_I(t)$] must be reloaded or recomputed for each interpolation. The fractional interval μ_k—which specifies the filter-coefficient sample values—never repeats if T_i and T_s are incommensurate.

If the filter has finite impulse response (FIR), then I filter coefficients and I signal samples must be delivered to the filter structure for each interpolation.

If an infinite impulse response (IIR) filter were employed, a recursive structure would be required so that the computing effort could be finite. Let the filter have p poles and z zeros. Then for each interpolation it would be necessary to load the following information to be able to compute the interpolant:

—$p + z + 1$ filter coefficients, as specified by μ_k.

—$z + 1$ signal samples.

—p past outputs of the filter, calculated with the *present* value of μ_k.

But those past outputs cannot be known for the present μ_k unless they were computed for all possible values of μ_k at every interpolation instant. That would ordinarily be an unacceptable computing burden and so FIR filters are usually preferred.

Other reasons for selecting FIR filters have been given in [3].

B. Ideal Interpolation

It is well known [3] that the bandlimited input signal $x(t)$ (or its samples $\{x(kT_i)\}$ at times $t = kT_i$) could be recovered from the samples $\{x(mT_s)\}$ by using the ideal filter with impulse response

$$h_I(t) = \frac{\sin \pi t/T_s}{\pi t/T_s}$$

and transfer function

$$H_I(f) = \begin{cases} T_s, & |f| < 1/2T_s \\ 0, & |f| > 1/2T_s \end{cases}$$

The ideal filter is IIR and noncausal; it cannot be realized and so perfect recovery of $x(t)$ is not possible with any practical filter. Failure of a realizable filter to reconstruct $x(t)$ would be charged as distortion in conventional applications of interpolation.

But perfect recovery is not required from an interpolator in a modem. It is only necessary that the filtered strobe outputs of the modem have the correct values—a much less stringent requirement than perfect reconstruction of $x(t)$. An interpolating filter in a modem need not be nearly so precise as some of the optimized interpolators found in the DSP literature, such as in [3].

Practical demands on the interpolation filter can be explored by considering its frequency response $H_I(f)$.

C. Stopband Response

The spectrum of the signal samples has periodic images, spaced at a frequency interval $1/T_s$. See Fig. 7. An interpolation filter is required to suppress those images prior to resampling. Any image energy that is not suppressed will be aliased by resampling and, if the sampling and symbol rates are incommensurate, will constitute random interference to the output sequence $\{y(k)\}$.

An ideal interpolation filter completely suppresses all input frequency components above $1/2T_s$ and the same stopband

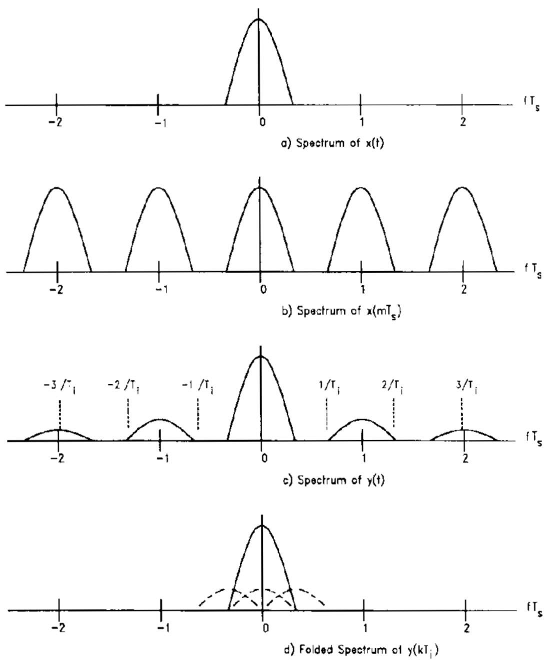


Fig. 7. Signal spectra.

behavior is desirable in a practical interpolation filter. Of course, no realizable filter can provide infinite attenuation over an entire stopband. Therefore, any practical filter will introduce some penalty because of incomplete suppression of images.

Fig. 7 illustrates spectra of various signals in the modem. The top line of the figure shows the bandlimited spectrum of the input signal $x(t)$. Sampling generates periodic spectral images, as in the second line. Absence of aliasing is indicated by the non-overlap of the images.

The time-continuous interpolating filter attenuates the images in varying degree, so that the spectrum of $y(t)$—the third line—consists of a main lobe around zero frequency, plus partially suppressed images at all integer multiples of $1/T_s$.

Upon resampling at rate $1/T_i$, all residual images fold in onto the desired signal. Fig. 7(d) sketches that part of the spectrum (not to scale) lying in the vicinity of zero frequency. The actual spectrum repeats with a period of $1/T_i$. If T_i/T_s is irrational, the folded images are uncorrelated with the desired signal and will impair recovery of the data. Relative power in the folded images, or equivalently, image attenuation by $H_I(f)$, is a measure of the adequacy of the stopband response of the filter.

D. Passband Response

An ideal interpolator would pass all frequencies from 0 to $1/2T_s$ with flat attenuation and with linear phase. In a modem where signal filtering is to be performed anyhow, there is no need for flat transmission in the filter's passband. The interpolator merely contributes a portion of the filtering that is required for the receiver. Any reasonable passband characteristic is permissible, provided that it can be compensated without penalty by other linear filters in the system.

This relaxation in the passband means that interpolating filters for use in modems can have much less stringent requirements than would be imposed upon interpolation filters that attempted to recover the orginal time function $x(t)$. The passband filtering allowable in a modem interpolator is not counted as distortion.

VI. Conclusion

If sampling in a digital modem is not synchronized with the data symbols, timing must be adjusted by interpolating new samples among the original ones. "Interpolation" is really a more-involved process that combines interpolation and subsequent decimation by resampling.

A useful conceptual model includes a digital-to-analog convertor, an analog, time-continuous interpolating filter, and a resampler, all fictitious, to produce the desired interpolants. Exactly the same interpolants can be computed entirely digitally from the input samples and knowledge of the sampled impulse response of the fictitious analog filter. Equation (6) underlies interpolation operations in digital modems.

An individual interpolant is specified by the signal samples (the *basepoint set*) that contribute to its value, and the filter samples used for the computation. The basepoint set is identified by a *basepoint index*, and the filter samples are identified by the *fractional interval*. These two pieces of information must be delivered to the digital interpolating structure by a controller. A number-controlled oscillator (NCO) can provide these parameters via control algorithms presented in the text.

Because the NCO is clocked synchronously with the signal samples, the modem output will exhibit timing jitter. This jitter is inconsequential if the data are consumed locally to the modem, because the NCO can provide a symbol clock with the same jitter as the data.

If the data must be retransmitted synchronously, the jitter may be intolerable. A jitter-free analog clock can be recovered from the NCO and used to reclock the jittered data prior to retransmission.

The fictitious analog interpolating filter should be FIR and should provide good stopband suppression of the periodic images of the sampled input signal. Passband response of this filter is part of the overall filtering of the modem. In consequence, non-flat response in the passband is not charged as distortion, as it would be in a classical interpolator. A designer has wide latitude in distributing overall filter response between the interpolating filter and other filters in the modem.

VII. Appendix A: Alternative Control Method

M. Moeneclaey has pointed out an alternative control scheme that does not use an NCO. Two successive interpolations are performed for time instants

$$kT_i = (m_k + \mu_k)T_s$$

$$(k+1)T_i = (m_{k+1} + \mu_{k+1})T_s. \quad \text{(A.1)}$$

Subtracting these two expressions and rearranging slightly gives the recursion

$$m_{k+1} = m_k + T_i/T_s + \mu_k - \mu_{k+1}. \tag{A.2}$$

By definition, m_{k+1} is an integer. Then, since $0 \le \mu_{k+1} < 1$,

$$m_{k+1} + \mu_{k+1} = m_k + T_i/T_s + \mu_k < m_{k+2} \tag{A.3}$$

whence the increment in sample count from one interpolation to the next is

$$m_{k+1} - m_k = \text{int}[T_i/T_s + \mu_k]. \tag{A.4}$$

Notice that a practical scheme must work with the increment rather than the sample count m_k. Any finite-length counter of m_k would overflow eventually.

To compute the fractional interval μ_k, recognize that the fractional part $fp[\ ]$ of the increment is zero

$$fp[m_{k+1} - m_k] = 0 = fp[T_i/T_s + \mu_k - \mu_{k+1}]$$

from which one may conclude

$$\mu_{k+1} = \left[\mu_k + \frac{T_i}{T_s}\right] \text{mod-1}. \tag{A.5}$$

The true T_i/T_s is not available. Instead, the synchronizer produces a control word $V(m_k) \simeq T_i/T_s$ to be used in the recursions (A.4) and (A.5). This control word is the synchronizer's estimate of the true interpolation period T_i relative to the sampling period T_s.

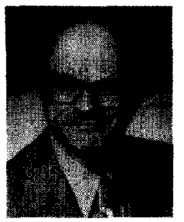
The alternative control method may be most useful in systems where the data are consumed at the same location as the data receiver, without reclocking. It is not immediately apparent how a jitter-free, time-continuous clock for retransmission could be synthesized easily without the phase $\eta(m)$ that accumulates in an NCO.

ACKNOWLEDGMENT

I wish to thank Dr. R. Harris and L. Erup of the European Space Agency for their helpful critiques of the work as it progressed.

REFERENCES

[1] L. Erup, F. M. Gardner, and R. A. Harris, "Interpolation in digital modems—Part II: implementation and performance," to be published.
[2] R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1983.
[3] R. W. Schafer and L. R. Rabiner, "A digital signal processing approach to interpolation," *Proc. IEEE*, vol. 61, pp. 692–702, June 1973.
[4] R. E. Crochiere, L. R. Rabiner, and R. R. Shively, "A novel implementation of digital phase shifters," *Bell Syst. Tech. J.*, vol. 54, pp. 1497–1502, Oct. 1975.
[5] F. Takahata *et al.*, "A PSK group modem for satellite communication," *IEEE J. Select. Areas Commun.*, vol. SAC-5, pp. 648–661, May 1987.
[6] M. Oerder, G. Ascheid, R. Haeb, and H. Meyr, "An all digital implementation of a receiver for bandwidth efficient communication," in *Signal Processing III* (Eusipco 1986), I. T. Young *et al.* Ed., pp. 1091–1094, Elsevier, 1986.
[7] G. Ascheid, M. Oerder, J. Stahl, and H. Meyr, "An all digital receiver architecture for bandwidth efficient transmission at high data rates," *IEEE Trans. Commun.*, vol. 37, pp. 804–813, Aug. 1989.
[8] J. A. C. Bingham, *The Theory and Practice of Modem Design.* New York: Wiley, 1988.
[9] F. M. Gardner, "A BPSK/QPSK timing-error detector for sampled receivers," *IEEE Trans. Commun.*, vol. COM-34, pp. 423–429, May 1986.
[10] E. Auer, "An advanced, variable data rate modem for Intelsat IDR/IBS services," Paper 1-3, *Proc. 2nd Int. Workshop Digital Signal Processing Techniques Appl. Space Commun.*, Turin, Italy, 24–25 Sept. 1990.



Floyd M. Gardner (S'49–A'54–SM'58–F'80) received the B.S.E.E. degree from the Illinois Institute of Technology, Chicago, IL, in 1950, the M.S.E.E. from Stanford University, Stanford, CA, in 1951, and the Ph.D. degree from the University of Illinois, Urbana, IL, 1953.

He has been a independent consulting engineer since 1960, active in the fields of communications and electronics. He is a specialist in synchronization and in phase-lock loops, and is the author of the book *Phaselock Techniques* (New York: Wiley, 2nd edition, 1979). In recent years he has been investigating algorithms for digitally implemented modems.

Dr. Gardner is a Registered Professional Engineer in the State of California.