N94- 29669

TDA Progress Report 42-116

February 15, 1994

Effects of Low Sampling Rate in the Digital Data-Transition Tracking Loop

A. Mileant
Telecommunications Systems Section

S. Million and S. Hinedi
Communications Systems Research Section

This article describes the performance of the all-digital data-transition tracking loop (DTTL) with coherent and noncoherent sampling using nonlinear theory. The effects of few samples per symbol and of noncommensurate sampling and symbol rates are addressed and analyzed. Their impact on the probability density and variance of the phase error are quantified through computer simulations.

It is shown that the performance of the all-digital DTTL approaches its analog counterpart when the sampling and symbol rates are noncommensurate (i.e., the number of samples per symbol is an irrational number). The loop signal-to-noise ratio (SNR) (inverse of phase error variance) degrades when the number of samples per symbol is an odd integer but degrades even further for even integers.

I. Introduction

In modern digital communication systems, analog-to-digital (A/D) conversion is performed as far toward the front end as possible using available technology. Usually, the received signal is amplified and then downconverted to the appropriate frequency for digital conversion. Thereafter, various system functions are performed digitally, including carrier, subcarrier, and symbol synchronization, as well as signal detection and decoding. Depending on the application, one can either sample the baseband signals (in-phase and quadrature) or sample the intermediate frequency (IF) signal. Furthermore, the sampling clock can be free running or controlled by the symbol synchronization loop. In the former case, the sampling rate is noncommensurate with the symbol rate. In the latter case, the sampling clock can be adjusted to obtain an integer number of samples per received symbol. All of these issues affect the final architecture and design of a receiver and influence the amount of cross-coupling among the various loops.

A receiver called the Advanced Receiver (ARX) has been developed for Deep Space Network (DSN) applica-

tions [1,2]. In the ARX, the signal is sampled at the IF, and the various tracking loops are implemented digitally, e.g., the "classical" analog integrate-and-dump filters, which are typically part of the loop arms (in-phase and quadrature), are replaced by digital accumulators. This article investigates the performance of the all-digital data-transition tracking loop (DTTL), used for symbol synchronization, for any number of samples per symbol. In the previous version of the Advanced Receiver (ARX I) [1], the sampling was performed synchronously with the symbol rate, and an integer number of samples per symbol were available. In the current version of the Advanced Receiver (ARX II) [2], the sampling is performed asynchronously, and the sampling clock is fixed and independent of the symbol rate. At the highest desired data rate of 6.6 Msymbols/sec and with a fixed 20-MHz processing rate for the ARX II, only about 3 samples per symbol can be obtained. We are interested in the all-digital DTTL response and performance with a small number of samples per symbol.

Some analytical results for the phase error variance of the analog DTTL were derived in [3], where the input is an analog signal and symbol and midphase detection were performed with analog integrate-and-dump filters. Later, the analysis was extended in [4], taking into account variations of the equivalent noise spectrum with respect to normalized phase error. In this article, we extend the previous results for the analog DTTL to the all-digital DTTL. We first note that there are two sampling models. One is to sample the signal instantaneously and the other is to obtain the sample by integration-and-dump (I & D) sampling of the signal. The instantaneous sampling technique can be used when the sampling rate is significantly higher than the symbol rate. The I & D sampling technique should be used when the number of samples per symbol is small [7]. In the absence of prefiltering and noise, the received symbol pulse is a perfect square wave, and instantaneous sampling provides voltage values of equal value. In the I & D case, all samples also have equal value except for the first one after a transition boundary of two symbols with different polarity. The all-digital DTTL can operate on either type of sample.

To illustrate the differences between analog and all-digital DTTLs, we consider the noiseless case first. Note that if the input is an analog signal, the midphase integrator can produce a nonzero error voltage no matter how small the phase error is. Thus, a correction voltage can be generated at every symbol transition whenever a phase error exists. Therefore, the analog DTTL has infinite resolution for phase detection.

The all-digital DTTL, in contrast, has only finite resolution for phase detection. This is illustrated in the following: Suppose that there are an even number of samples per symbol. When a symbol transition occurs, the digital midphase accumulator will produce a nonzero voltage only if the phase error causes sample slipping (assuming equal amplitude samples). As long as the phase error stays within a range of values that avoids sample slipping, the loop always generates a zero error signal. This range of undetectable phase errors accounts for the finite resolution of the all-digital DTTL. The more samples per symbol we use, the higher the resolution we can achieve and the closer the all-digital DTTL is to its analog counterpart. A key question is the impact of the all-digital DTTL's finite resolution on the S-curve and the phase error variance for few samples per symbol (say, three or four samples).

Another issue in an all-digital implementation is the effect of a noninteger number of samples per symbol. If the sampling clock is driven by the symbol synchronization loop, the number of samples per symbol can be made an exact even integer, which reduces the self-noise generated in the midphase accumulator, as will be discussed later. Under that sampling scenario, the sampling clock is constantly adapting as the data rate changes due to Doppler or other effects. One disadvantage of that scheme is that no fixed time base is available in the system. On the other hand, if the sampling clock is free running and is derived from a fixed frequency standard, the sampling period is fixed although the symbol rate may change. This may result in a noninteger number of samples per symbol. A model is derived in this article to analyze the performance of the all-digital DTTL for any sampling and symbol rates. In Section II, a nonlinear analysis of the loop is presented to handle all scenarios along with simulation results. The conclusion is presented in Section III.

II. Analysis

The performance of the all-digital data-transition tracking loop with coherent and noncoherent sampling is analyzed here. The block diagram of the all-digital DTTL is shown in Fig. 1. The input $r(i)$ to the DTTL can be obtained by instantaneous sampling or by I & D sampling. In the subsequent derivation, closed form solutions will be obtained assuming samples of equal value, and results will be verified by simulation assuming samples of both equal and unequal value.

Noncoherent sampling means that the sampling clock runs independently of the estimated symbol phase, i.e.,

the sampling time is independent of the estimated symbol phase. This has minimal impact if there are many samples per symbol. As the number of samples per symbol decreases, the S-curve becomes more and more coarse, and self-noise (to be explained later) increases. Our goal is to quantify the effects of a small number of samples per symbol on the phase error variance. A theory is presented for a first-order DTTL using Markov chains. The approach is to derive the S-curve and to solve the Chapman–Kolmogorov (C–K) equation to get the density function of the phase error. The closed-loop phase error variance and the degradation of the symbol detection can be evaluated from the phase error density function.

To illustrate the phenomenon of self-noise, we consider a simple example shown in Fig. 2, where we have five samples per symbol. We assume no thermal noise and perfect tracking at a particular moment. The output of the symbol transition detector is not zero because it either sums three samples from the first symbol and two samples from the second symbol [Fig. 2(a)] or sums two samples from the first symbol and three samples from the second symbol [Fig. 2(b)]. Notice that this situation occurs for every symbol interval as long as the loop maintains perfect tracking. The nonzero output of the loop filter will drag the loop away from the perfect tracking condition.

In order to quantify this phenomenon, we first introduce three useful parameters. Let β denote the number of samples per symbol, which may not be an integer, and $\alpha(1)$ denote the offset of the first sample mark in a received symbol from the symbol boundary. By convention, $\alpha(k)$ is normalized and is measured as a fraction of the sampling interval. We observe that $\alpha(k)$ remains constant if β is an integer, and it varies from symbol to symbol if β is not an integer. Let us number the received symbols by 1, 2, 3, ... and denote the value of $\alpha(k)$ at the first symbol as $\alpha(1)$, which is referred to as the initial sampling offset. The values of $\alpha(k)$ at the subsequent symbols, namely, $\alpha(2)$, $\alpha(3)$, ..., can be computed from β and $\alpha(1)$. The number of sample marks in a transition detection window and the number of sample marks in a symbol detection window are all functions of $\alpha(k)$. Thus, the output of the symbol detector and that of the transition detector fluctuate from symbol to symbol as $\alpha(k)$. This subject will be discussed later in more detail.

Another important observation about the all-digital DTTL with a low number of samples per symbol is that it can have an irreducible phase error due to a finite number of samples per symbol. To illustrate this phenomenon, let us consider the example shown in Fig. 3, where every symbol contains four samples. We can see that as long as the estimated phase lies between t_1 and t_2, the error signal is always zero (or nearly zero if the received symbol does not have a perfect square waveform or if we have unequal amplitude samples) and the DTTL continues tracking. However, we see that there still exists unresolved phase ambiguity within the interval from t_1 to t_2. Mathematically, this phase ambiguity can be explained by a staircase S-curve. This phenomenon might have little effect on symbol detection performance if we use straight accumulation to detect the symbols. However, if we use weighted accumulation to detect the symbols, the phase ambiguity can introduce misweighting and thus degrade performance [5,6].

Before we proceed to the mathematical analysis, let us examine the all-digital DTTL block diagram again. The error signal accumulator between the loop filter $F(z)$ and the multiplier performs an averaging function so that the subsequent loop filter can operate at a slower speed. The loop bandwidth is determined primarily by the loop filter $F(z)$. Thus, the presence of the accumulator is simply for hardware convenience. In the following analysis, we consider the DTTL without the error-signal accumulator.

A. Mathematical Model

Assuming that the carrier and subcarrier (if any) have been removed in an ideal fashion, the received baseband waveform is given by

$$r(t) = \sqrt{S}\sum_k a_k p(t - kT) + n(t) \tag{1}$$

where S is the signal power, T is the symbol time, $n(t)$ is white Gaussian noise with one-sided power spectral density N_0 W/Hz, $a_k = \pm 1$ represents the polarity of the kth symbol, and $p(t)$ is the square-wave function having a value of 1 for $0 \leq t < T$, and having a value of 0 elsewhere. The ith sample can be expressed after normalizing by $1/\sqrt{S}$ as

$$r(i) = a_k + n(i) \tag{2}$$

where we assume the sample is derived from the kth symbol, $n(i)$ is a zero-mean Gaussian random variable with variance $\sigma^2 = N_0/(2ST_s)$, and T_s is the sampling interval. Let the phase error λ (in cycles) be defined as

$$\lambda = \frac{\theta - \hat{\theta}}{2\pi} \tag{3}$$

where θ is the actual received symbol phase and $\hat{\theta}$ is the estimated symbol phase. Note that λ should have a value between -0.5 and 0.5. The error signal is affected by the locations of samples within their respective received symbols. In order to quantify this effect, we define a set of four Δ_k integer-valued functions. They represent the number of sample marks contained in their respective intervals, as illustrated in Fig. 4. The output of the in-phase accumulator $x(k)$ and the midphase accumulator $y(k)$ can be expressed in terms of the Δ_k functions, namely

$$x(k) = \begin{cases} \Delta_1(k)a_k + \Delta_2(k)a_{k+1} + n_1(k) + n_2(k) \\ \quad \text{for } \lambda \geq 0 \\ \Delta_1(k)a_k + \Delta_2(k)a_{k-1} + n_1(k) + n_2(k) \\ \quad \text{for } \lambda < 0 \end{cases}$$

$$x(k+1) = \begin{cases} \Delta_1(k+1)a_{k+1} + \Delta_2(k+1)a_{k+2} + n_3(k) \\ +n_4(k) \quad \text{for } \lambda \geq 0 \\ \Delta_1(k+1)a_{k+1} + \Delta_2(k+1)a_k + n_3(k) \\ +n_4(k) \quad \text{for } \lambda < 0 \end{cases}$$

$$y(k+1) = \Delta_3(k)a_k + \Delta_4(k)a_{k+1} + n_2(k) + n_3(k) \tag{4}$$

where $n_j(k), j = 1, 2, 3, 4$ are zero-mean Gaussian random variables with variances

$$\mathrm{Var}\{n_1(k)\} = \begin{cases} (\Delta_1(k) - \Delta_3(k))\sigma^2 \text{ for } \lambda \geq 0 \\ (\Delta_1(k) + \Delta_2(k) + \Delta_2(k+1) - \Delta_3(k))\sigma^2 \\ \quad \text{for } \lambda < 0 \end{cases}$$

$$\mathrm{Var}\{n_2(k)\} = \begin{cases} (\Delta_3(k) + \Delta_2(k))\sigma^2 \text{ for } \lambda \geq 0 \\ (\Delta_3(k) - \Delta_2(k+1))\sigma^2 \text{ for } \lambda < 0 \end{cases}$$

$$\mathrm{Var}\{n_3(k)\} = \begin{cases} (\Delta_4(k) - \Delta_2(k))\sigma^2 \text{ for } \lambda \geq 0 \\ (\Delta_4(k) + \Delta_2(k+1))\sigma^2 \text{ for } \lambda < 0 \end{cases}$$

$$\mathrm{Var}\{n_4(k)\} = \begin{cases} (\Delta_2(k) + \Delta_1(k+1) + \Delta_2(k+1) \\ -\Delta_4(k))\sigma^2 \text{ for } \lambda \geq 0 \\ (\Delta_1(k+1) - \Delta_4(k))\sigma^2 \text{ for } \lambda < 0 \end{cases} \tag{5}$$

and the Δ_j functions are computed from

$$\Delta_1(k) = \begin{cases} \lfloor 2\beta - \alpha(k-1) \rfloor - \lfloor (1+\lambda)\beta - \alpha(k-1) \rfloor \\ \quad \text{for } \lambda \geq 0 \\ \lfloor (2+\lambda)\beta - \alpha(k-1) \rfloor - \lfloor \beta - \alpha(k-1) \rfloor \\ \quad \text{for } \lambda < 0 \end{cases}$$

$$\Delta_2(k) = \begin{cases} \lfloor (1+\lambda)\beta - \alpha(k) \rfloor - \lfloor \beta - \alpha(k) \rfloor \\ \quad \text{for } \lambda \geq 0 \\ \lfloor \beta - \alpha(k-1) \rfloor - \lfloor (1+\lambda)\beta - \alpha(k-1) \rfloor \\ \quad \text{for } \lambda < 0 \end{cases}$$

$$\Delta_3(k) = \lfloor \beta - \alpha(k) \rfloor - \left\lfloor \left(1 + \lambda - \frac{w}{2}\right)\beta - \alpha(k) \right\rfloor$$

$$\Delta_4(k) = \left\lfloor \left(1 + \lambda + \frac{w}{2}\right)\beta - \alpha(k) \right\rfloor - \lfloor \beta - \alpha(k) \rfloor \tag{6}$$

with w denoting the width of the transition window and $w \leq 1$. As previously mentioned, $\alpha(k+1)$ is the sampling offset and is computed recursively from $\alpha(k)$ using

$$\alpha(k+1) = \lfloor \beta - \alpha(k) + 1 \rfloor - (\beta - \alpha(k)) \tag{7}$$

and $\lfloor y \rfloor$ denotes the greatest integer strictly less than y (i.e., $\lfloor 4.2 \rfloor = 4$, $\lfloor 4 \rfloor = 3$). The derivation of the Δ_i functions are similar and we illustrate only one example here, $\Delta_2(k)$. To derive $\Delta_2(k)$, we use the beginning of the kth received symbol as the reference point. The number of samples in the kth received symbol is $\lfloor \beta - \alpha(k) \rfloor + 1$. The number of samples from the beginning of the kth symbol to the end of the kth estimated symbol is $\lfloor (1+\lambda)\beta - \alpha(k) \rfloor + 1$. Equation (5) follows by observing that the number of samples from the end of the kth received symbol to the end of the kth estimated symbol is $\Delta_2(k)$. The error signal $e(k+1)$ is given by

$$e(k+1) = z(k+1)y(k+1) \tag{8}$$

where $z(k+1)$ denotes the output of the transition detector and $\mathrm{sgn}(x)$ denotes the "signum" function, i.e.,

$$z(k+1) = \frac{\mathrm{sgn}\,[x(k+1)] - \mathrm{sgn}\,[x(k)]}{2} \tag{9}$$

The conditional S-curve is defined by

$$g(\lambda|\alpha(k)) = E_{s,n}\{e(k+1)|\lambda, \alpha(k)\} \tag{10}$$

where $E_{s,n}$ represents the conditional expectation with respect to both the signal (s) and the noise (n), assuming fixed λ and $\alpha(k)$. Following mathematical manipulation similar to that in [4], we get

$$E_{s,n}\{e(k+1)|\lambda,\alpha(k)\} =$$

$$\begin{cases} 0.25[\Delta_4(k)(\text{erf }[r_1(k)] - \text{erf }[r_2(k)] + \text{erf }[r_1(k+1)] \\ + \text{ erf }[r_2(k+1)]) - \Delta_3(k)(\text{erf }[r_1(k)] + \text{erf }[r_2(k)]) \\ - r_3(k)(\Delta_3(k) + \Delta_2(k))(\exp[-r_1^2(k)] + \exp[-r_2^2(k)]) \\ + r_3(k)(\Delta_4(k) - \Delta_2(k))(\exp[-r_1^2(k+1)] \\ + \exp[-r_2^2(k+1)])] \quad \text{for } \lambda \geq 0 \\ 0.25[\Delta_4(k)(\text{erf }[r_1(k+1)] + \text{erf }[r_2(k+1)]) - \Delta_3(k) \\ (\text{erf }[r_1(k+1)] - \text{erf }[r_2(k+1)] \\ + \text{ erf }[r_1(k)] + \text{erf }[r_2(k)]) \\ - r_3(k)(\Delta_3(k) - \Delta_2(k))(\exp[-r_1^2(k)] + \exp[-r_2^2(k)]) \\ + r_3(k)(\Delta_4(k) + \Delta_2(k))(\exp[-r_1^2(k+1)] \\ + \exp[-r_2^2(k+1)])] \quad \text{for } \lambda < 0 \end{cases} \tag{11}$$

where

$$r_1(k) = \sqrt{\frac{R_s(\Delta_1(k) - \Delta_2(k))^2}{\beta(\Delta_1(k) + \Delta_2(k))}}$$

$$r_2(k) = \sqrt{\frac{R_s(\Delta_1(k) + \Delta_2(k))}{\beta}}$$

$$r_3(k) = \sqrt{\frac{\beta}{\pi R_s(\Delta_1(k) + \Delta_2(k))}}$$

$$R_s = \frac{2ST}{N_0}$$

$$\text{erf }(x) = \frac{2\int_0^x e^{-t^2}dt}{\sqrt{\pi}} \tag{12}$$

We observe that $g(\lambda|\alpha(k))$ is the (unconditional) S-curve if $\alpha(k)$ is a constant. If $\alpha(k)$ changes from symbol to symbol, its effect will be smoothed in the loop as long as the loop time constant (inverse of loop bandwidth) is larger than MT. M denotes the number of distinct $\alpha(k)$'s and is discussed in the subsequent section. Therefore, the S-curve is obtained by averaging Eq. (11) over the M values of $\alpha(k)$, which is determined by the initial sampling offset and the number of samples per symbol β, that is,

$$g(\lambda) = \frac{1}{M}\sum_{k=1}^{M} g(\lambda|\alpha(k)) \tag{13}$$

The S-curve $g(\lambda)$ is a function of λ, β, and $\alpha(k)$. In general, β can be any real number. When β is an irrational number, then $\alpha(k)$ is nonperiodic. However, when β is a rational number, then $a(k)$ becomes periodic and assumes a finite number of possible values. For example, if β is of the form $X.0000\ldots$, where X is any nonzero integer, then $\alpha(k)$ will have only one value all the time; if β is of the form $X.X000\ldots$, then $\alpha(k)$ will have at most 10 different values; if β is of the form $X.XX000\ldots$, then $\alpha(k)$ will have at most 100 different values, and so forth. For example, consider $\beta = 4.1$ and suppose the initial sampling offset $\alpha(1) = 0.7$. Clearly, $\alpha(2) = 0.6$, $\alpha(3) = 0.5$, $\alpha(4) = 0.4$, $\alpha(5) = 0.3$, $\alpha(6) = 0.2$, $\alpha(7) = 0.1$, $\alpha(8) = 0$, $\alpha(9) = 0.9$, $\alpha(10) = 0.8$, $\alpha(11) = 0.7\ldots$. The more values $\alpha(k)$ takes, the smoother will be the S-curve of the tracking loop.

For a large β, $r_i(k+1) = r_i(k)$ and $i = 1,2,3$; for $\lambda \geq 0$, $\Delta_1(k) \to (1-\lambda)\beta$, $\Delta_2(k) \to \lambda\beta$, $\Delta_3(k) \to (w/2 - \lambda)\beta$, $\Delta_4(k) \to (w/2+\lambda)\beta$; and, as a result, $r_1(k) \to (1 - 2\lambda)\sqrt{R_s}$, $r_2(k) \to \sqrt{R_s}$, $r_3(k) \to 1/\sqrt{\pi R_s}$. Substituting back in Eq. (11) and simplifying, one obtains

$$\frac{E_{s,n}\{e(\lambda)\}}{\beta} = \lambda \,\text{erf}\left(\sqrt{R_s}(1-2\lambda)\right) - \frac{1}{8}[w - 2\lambda]\left[\text{erf}\left(\sqrt{R_s}\right) - \text{erf}\left(\sqrt{R_s}(1-2\lambda)\right)\right] \tag{14}$$

which agrees with the S-curve derived by Simon [4] for the analog loop. For $\lambda \leq 0$, similar steps can be taken to show that the resulting S-curve also agrees with the analog case, with slope at the origin given by

$$m = \beta \left(\text{erf}\left(\sqrt{R_s}\right) - \frac{w}{2}\sqrt{\frac{R_s}{\pi}} e^{-R_s} \right) \qquad (15)$$

Figure 5 depicts the theoretical and simulated S-curves as a function of the phase error for various values of β, namely 4, 5, 10, 4.5, and 4.74, with $w = 1$. Note the staircase shape of the S-curve for all β, with step size proportional to $1/M\beta$. In this case, the phase detector is insensitive to variations of the input phase which occur in the flat regions of the S-curve. For small integers, the steps are large, while for large integers, the steps are small. In general, the S-curve is not symmetric and has a bias (i.e., $g(0) \neq 0$) proportional to $1/M\beta$, except for the case when β is an exact even integer. The sign of the asymmetry depends on $\alpha(1)$, which was set to 0.5 in Fig. 5(e). As β approaches an irrational number, the S-curve becomes smooth due to the averaging over a large set of values and it converges to the S-curve encountered in analog systems. In all these figures, the S-curve has a zero slope at the origin, which prevents a linear analysis because it requires a nonzero slope. Also depicted in Fig. 5(a) are two cases: when the signal and noise $(s + n)$ are filtered prior to the loop and when only the signal is filtered but not the noise. The first scenario represents the case where the main contribution of the filtering is occurring at the receiver, while the latter scenario is representative of filtering at the transmitter. The filter used in the simulations is a first-order lowpass with transfer function $H(z) = h_0/(1 + h_1 z^{-1})$. Note from Fig. 5(a) that the filtering introduces a bias and a slight asymmetry in the S-curve which is mainly due to the signal filtering. The asymmetry in the S-curve can be reduced by using a raised cosine or square-root raised cosine filter [7] that results in a symmetrical pulse shape around the data transition point. In the remaining figures, simulations with both signal and noise filtering are shown to assess the impact of imperfect square pulses.

One way to linearize the loop and, hence, to smooth the shape of the S-curve is to randomly shift by an amount the position of the window relative to the symbol transition point. The effect of this random back and forth shifting is to produce on the average the same amount of samples in both halves of the symbol transition detector, so that β would appear to have an irrational value to the mid-phase detector. The performance of the DTTL with such a detector can be the topic of future work.

We will proceed to perform a nonlinear analysis of the loop using the C–K equation. However, we will need the open-loop variance of the error signal at the input to the loop filter to compute the transitional probability density. The second moment of the error signal is given by

$$e_2(\lambda) = E_{s,n}\left\{ \left(\frac{1}{M} \sum_{k=1}^{M} e(k,\lambda) \right)^2 \right\}$$

$$= \frac{1}{M} \sum_{k=1}^{M} E_{s,n}\{e(k,\lambda)^2\} + \frac{2}{M-1} \sum_{k=1}^{M-1} E_{s,n}\{e(k,\lambda) \times e(k+1,\lambda)\}$$

$$\triangleq \frac{1}{8M} \sum_{k=1}^{M} \left[v_1(k) + v_2(k) \right] + \frac{2}{16(M-1)} \sum_{k=1}^{M-1} \times \left[v_3(k) + v_4(k) + v_5(k) \right] \qquad (16)$$

where the variables $v_i(k)$, $i = 1,2,3,4,5$ are defined in the Appendix. The open-loop variance is then computed using

$$\sigma_e^2(\lambda) = e_2(\lambda) - g^2(\lambda) \qquad (17)$$

The open loop variance $\sigma_e^2(\lambda)$ is depicted in Fig. 6 for various values of β as a function of λ. Again, note the staircase shape of the variance which approaches a smooth function for $\beta = 4.74$. The analysis agrees very well with the simulation points for all values of β.

B. Probability Density Function of the Phase Error

When the number of samples per symbol, β, is low and the decimal part of β has only a few nonzero digits, the S-curve takes a coarse staircase shape, as discussed previously. In this case, the loop behaves nonlinearly, and the phase jitter of the loop cannot be predicted from the loop's noise-equivalent bandwidth. However, when the statistics of the driving noise process are known, the C–K equation permits one to derive the probability density function (pdf) of the closed-loop phase jitter even for a highly nonlinear loop. Once this pdf is found, all the moments of the phase jitter process can be computed, and the noise performance of the loop can be predicted.

In general, an all-digital phase-locked loop can be described by a stochastic difference equation of the form

$$\lambda(k) = u(k) - [g(\lambda(k)) + n(k)]N(z)F(z) \qquad (18)$$

where $\lambda(k)$ is the phase error at the instant k, $u(k)$ is the input signal sequence, $F(z)$ and $N(z)$ are the dis-

crete transfer functions of the loop filter and the numerically controlled oscillator (NCO), respectively, the function $g(\lambda(k))$ represents the nonlinearity of the phase detector, and $n(k)$ is the open-loop noise process. In digital phase-locked loops, this noise process consists of both thermal and quantization noise. For the all-digital DTTL under consideration, the noise $n(k)$ is zero-mean with the variance given by Eq. (17). In general, the product $F(z)N(z)$ is of the form

$$F(z)N(z) = \frac{b_0 + b_1 z^{-1} + \ldots + b_N z^{-N}}{a_0 + a_1 z^{-1} + \ldots + a_N z^{-N}} \qquad (19)$$

Here in the context of the stochastic difference equations, z^{-1} should be thought of as the unit delay operator. Using Eq. (19), Eq. (18) can be rewritten as

$$\begin{aligned}\lambda(k) = {} & [a_0 u(k) + b_0(g(\lambda(k)) + n(k))] \\ & + z^{-1}[a_1 u(k) - a_1\lambda(k) - b_1 g(\lambda(k)) - b_1 n(k)] \\ & + z^{-2}[a_2 u(k) - a_2\lambda(k) - b_2 g(\lambda(k)) - b_2 n(k)] \\ & + \quad \ldots \quad \ldots \quad \ldots \\ & + z^{-N}[a_N u(k) - a_N\lambda(k) - b_N g(\lambda(k)) - b_N n(k)]\end{aligned} \qquad (20)$$

Defining x_1 as the row with z^{-1}, x_2 as the row with z^{-2}, etc., the above equation can now be expressed in terms of the following state and output equations:

$$\begin{aligned}\underline{x}(k+1) &= \mathbf{A}\underline{x}(k) - \underline{b}\, g(\lambda(k)) + \underline{a}\, u(k) - \underline{b}\, n(k) \\ \lambda(k) &= x_1(k) + [a_0 u(k) + b_0 g(\lambda(k)) + b_0 n(k)]\end{aligned} \qquad (21)$$

where $\underline{x}$ is the state vector, $[x_1\, x_2 \cdots x_N]^t$, $\mathbf{A}$ is the transition matrix

$$\mathbf{A} = \begin{bmatrix} -a_1 & 1 & 0 & 0 & \cdot \\ -a_2 & 0 & 1 & 0 & \cdot \\ \cdot & 0 & 0 & \cdot & \cdot \\ -a_N & 0 & 0 & 0 & 0 \end{bmatrix} \qquad (22)$$

$\underline{a} = [a_1\, a_2 \cdots a_N]^t$, and $\underline{b} = [b_1\, b_2 \cdots b_N]^t$ (the superscript t denotes transpose). The need to rewrite the original stochastic difference in vector form stems from the fact that the original equation does not represent a Markov process while its new form is a vector Markov process. For a vector Markov process, the C–K equation relates the probability density function of the state vector $\underline{\mathbf{x}}$ at time $(k+1)$, $p_{k+1}(\underline{\mathbf{x}}(k+1))$, to its probability density function at time k through the following integral:

$$p_{k+1}(\underline{\mathbf{x}}(k+1) = \underline{x}|\,\underline{x}_0) = \underbrace{\int_{-\infty}^{\infty} \cdots \int}_{N-dim} q[\underline{\mathbf{x}}(k+1) = \underline{x}|\,\underline{\mathbf{x}}(k) = \underline{v}]p_k(\underline{\mathbf{x}}(k) = \underline{v}|\,\underline{x}_0)\, dv_1 \cdots dv_N \qquad (23)$$

where $\underline{x}_0$ is the initial condition vector and $q(\cdot)$ is the transition probability density function.

We will now focus on solving the above equation for the steady-state case (large k) and a periodic phase detector. In steady state, the initial condition vector $\underline{x}_0$ is washed out and can be dropped from the C–K equation, $p_{k+1}(\cdot) \rightarrow p_k(\cdot) \triangleq p(\cdot)$, so that the index k can be dropped also. With a periodic phase detector $g(\cdot)$, each state variable takes values only between $0.5 < x_i \leq 0.5$ $(i = 1 \ldots N)$. This requires the "folding" (or collapsing) of the transitional pdf $q[\cdot]$ to the bounded region. We define this new transitional pdf as $\xi[\cdot]$ and rewrite the C–K as follows:

$$\begin{aligned}\dot{p}(\underline{\mathbf{y}} = \underline{y}) &= \underbrace{\int_{-0.5}^{0.5} \cdots \int}_{N-dim} \xi[\underline{\mathbf{y}}(k+1) = \underline{y}|\,\underline{\mathbf{y}}(k) \\ &= \underline{w}]\,\dot{p}(\underline{\mathbf{y}} = \underline{w})dw_1 \cdots dw_N\end{aligned} \qquad (24)$$

where the primed p represents the new pdf restricted to the $\{-0.5, 0.5\}$ region. Even in this simplified form of the C–K equation, closed form solutions can be obtained only for a few special cases. In general, the C–K equation has to be solved numerically, and solutions for N larger than 1 can become very computationally demanding.

In order to proceed, we will restrict our attention to the case of $N = 1$ with $F(z) = b_1$ and $N(z) = z^{-1}/1 - z^{-1}$ (ideal summer), which corresponds to a type-I loop with no

computational delay. Rewriting Eq. (19) for this restricted case, we get

$$F(z)N(z) = \frac{b_1 z^{-1}}{a_0 + a_1 z^{-1}} \tag{25}$$

With $N = 1$, the variables in Eq. (20) are $a_0 = 1$ and $a_1 = -1$ with all others being zero. Since we are not considering any static phase error, we set the input signal $u_k \rightarrow 0$ so that the loop is driven by noise only. Next, we discretize the continuous variable y into $L + 1$ y_i values and, thus, approximate the Markov process by a Markov chain [8]. The larger the value of L, the better is the approximation. The continuous pdf on both sides of the C–K equation can be replaced by discrete probabilities $P(y_i) = \delta p(y = y_i)$ where $\delta = 1/L$, and the transition pdf $\xi[i, j]$ is replaced by transition probability mass distribution $Q[i, j] = \delta^2 \xi[i, j]$. With this substitution, the C–K equation becomes

$$P(y = y_i) = \sum_{j=-\frac{L}{2}}^{\frac{L}{2}} Q[y(k+1) = y_i \,|\, y(k) = w_j]\, P(y = w_j) \tag{26}$$

where $x_i = i\delta$, $i = -L/2, \cdots L/2$. The above equation has to be true for all i. Let $\underline{P}(\cdot)$ be the $(L + 1)$ dimensional vector with elements $P(y = y_i)$ and $\mathbf{Q}$ be the $(L + 1) \times (L+1)$ matrix with elements $Q[i, j]$; then the C–K equation can be written in the compact form $\underline{P} = \mathbf{Q}\underline{P}$, which can be solved by various techniques used in systems of linear equations [9]. For example, one can write the above matrix equation as

$$[\mathbf{Q} - \mathbf{I}]\underline{P} = \underline{0} \tag{27}$$

with $\mathbf{I}$ being the identity matrix and $\underline{0}$ the zero vector. This matrix $\mathbf{Q} - \mathbf{I}$ will have at most $(L+1)$ distinct eigenvalues, and the desired solution, the $\underline{P}$ vector, will be the eigenvector corresponding to the zero eigenvalue. The last step in solving for $\underline{P}$ is to assume a suitable transition probability density function which will generate the elements $Q[i, j]$ of the matrix $\mathbf{Q}$. When thermal noise dominates, the noise process $n(k)$ can be assumed to be Gaussian, for which

$$Q[i, j] = \delta^2 \frac{1}{\sqrt{2\pi\sigma_j^2}} \sum_{l=-\infty}^{\infty} \exp\left[\frac{(x_i + l - \mu_j)^2}{2\sigma_j^2}\right] \tag{28}$$

where $\sigma_j^2 = b_1^2 \sigma_e^2(\lambda)$, $\sigma_e^2(\lambda)$ is the variance given by Eq. (17) at $\lambda = x_j$, and $\mu_j = -a_1 x_j - b_1 g(x_j)$.

However, the Gaussian assumption becomes less accurate when the quantization noise starts to dominate over the thermal noise (the high-SNR case). Our simulations have shown that, at high SNRs, the pdf on the phase error process becomes highly irregular and difficult to describe mathematically. So a trapezoidal or a uniform distribution function with mean and variance as used for the Gaussian gives results close to the ones obtained by simulation. In our computations, the singular value decomposition (SVD) algorithm was used, and L was set to 1100 to achieve acceptable resolution at a high SNR. The computed and simulated pdfs are depicted in Fig. 7 for the case of $\beta = 4, 5, 10, 4.5$, and 4.74. When $\beta = 4$, the pdf exhibits a significant flat region, as expected. However, for $\beta = 5$, the flat region disappears due to the effect of the odd number of samples per symbol. The simulations were carried out with one-sided loop bandwidth B_L set to 0.01 Hz and a symbol rate of 1 symbol/sec. For $\beta = 10$, the flat region is present again but is significantly reduced compared to the case of $\beta = 4$. For $\beta = 4.5$ or 4.74, the pdf is a smooth function, as one would expect. Note the nonzero mean in Fig. 7(e), which is due to the asymmetric S-curve in Fig. 5(e).

The corresponding variances for all cases are shown in Table 1. In Table 1(a), the loop SNR (inverse of tracking variance) is higher at $\beta = 5$ than at $\beta = 4$ and 10 due to the averaging process over the various offsets, $\alpha(k)$'s. Note that the simulations and predictions agree well in all cases, with the largest deviation, of about 0.7 dB, obtained with $\beta = 5$. Table 1(b) depicts the effect of the self-noise which is dominant at high symbol SNRs (such as 25 dB). The model predicts the contribution of the self-noise very accurately for integer as well as noninteger β. This noise is the limiting factor in the tracking performance in any all-digital DTTL implementation and is nonexistent in an analog design.

III. Conclusion

Nonlinear analysis of the all-digital DTTL is used to quantify its performance as a function of the number of samples per symbol β. It is shown that the probability density function of the closed-loop phase error can vary significantly depending upon the number of samples per symbol, β, and the symbol SNR.

The performance of the all-digital DTTL approaches its analog counterpart as β increases and the sampling and

symbol rates are noncommensurate (i.e., β is irrational). The loop SNR (inverse of phase error variance) degrades when β is an odd integer and degrades even further when β is an even integer. In general, the S-curve has a bias proportional to $1/(M\beta)$, but for an even-integer β the bias goes to zero.

Acknowledgment

Credit should be given to Dr. J. B. Thomas at JPL, who suggested the advantage of noncommensurate sampling in systems where the number of samples per symbol is small.

References

[1] D. H. Brown and W. J. Hurd, "DSN Advanced Receiver: Breadboard Description and Test Results," *The Telecommunications and Data Acquisition Progress Report 42-89, vol. January-March 1987*, Jet Propulsion Laboratory, Pasadena, California, pp. 48–66, May 15, 1987.

[2] S. Hinedi, "NASA's Next Generation Deep Space Network Breadboard Receiver," *IEEE Trans. on Comm.*, vol. 41, no. 1, pp. 246–257, January 1993.

[3] W. C. Lindsey and T. O. Anderson, "Digital-Data Transition Tracking Loop," International Telemetry Conference, October 8–10, 1968, Los Angeles, California.

[4] M. K. Simon, "An Analysis of the Steady-State Phase Noise Performance of a Digital Data-Transition Tracking Loop," *Jet Propulsion Laboratory Space Programs Summary 37-55*, vol. III, Jet Propulsion Laboratory, Pasadena, California, pp. 54–62, February 1969.

[5] R. Sadr and W. J. Hurd, "Detection Of Signals by the Digital Integrate-and-Dump Filter with Offset Sampling," *The Telecommunications and Data Acquisition Progress Report 42-91, vol. July-September 1987*, Jet Propulsion Laboratory, Pasadena, California, pp. 158–173, November 15, 1987.

[6] R. Sadr, "Detection Of Signals by Weighted Integrate-and-Dump Filter," *The Telecommunications and Data Acquisition Progress Report 42-91, vol. July-September 1987*, Jet Propulsion Laboratory, Pasadena, California, pp. 174–185, November 15, 1987.

[7] E. A. Lee and D. G. Messerschmitt, *Digital Communication*, Boston, Massachusetts: Kluwer Academic Publishers, 1988.

[8] W. B. Davenport and W. L. Root, *Random Signals and Noise*, New York: McGraw-Hill Book Co., 1958.

[9] G. Strang, *Linear Algebra and Its Applications*, 3rd ed., Florida: Harcourt Brace Jovanovich, Inc., 1988.

Table 1. Predicted and simulated variances: (a) white noise dominated region and (b) self-noise dominated region.

(a)

β	Symbol SNR, dB	Variance, simulation	Variance, theory
4.0	3	0.00635	0.006324
5.0	3	0.000954	0.00114
10.0	3	0.00234	0.002434
10.0	10	0.000925	0.000972
4.5	3	0.00235	0.0026
4.74	3	0.00253	0.002720

(b)

β	Symbol SNR, dB	Variance, simulation	Variance, theory
4.0	25	0.004448	0.004578
5.0	25	0.000028	0.000030
10.0	25	0.000910	0.000972
4.5	25	0.000021	0.000024
4.75	25	0.002633	0.002750

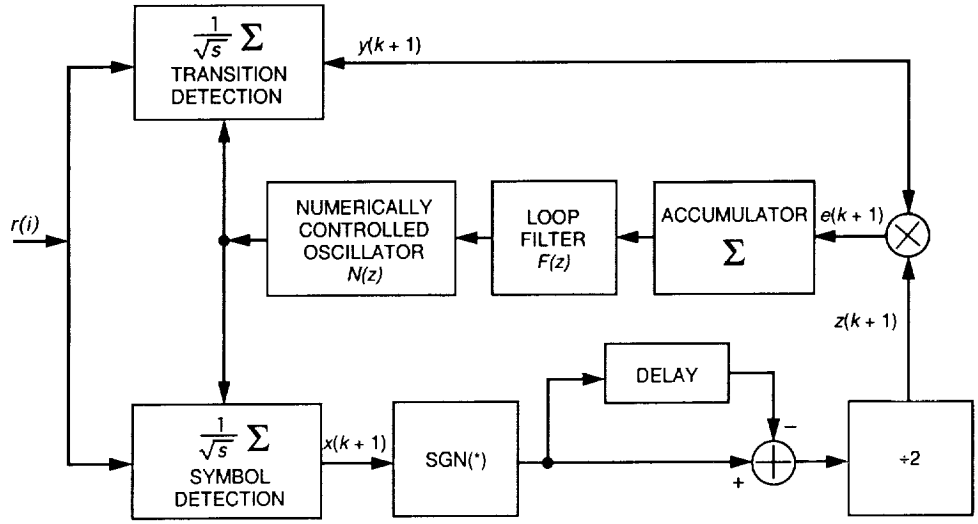

Fig. 1. Block diagram of the all-digital DTTL.

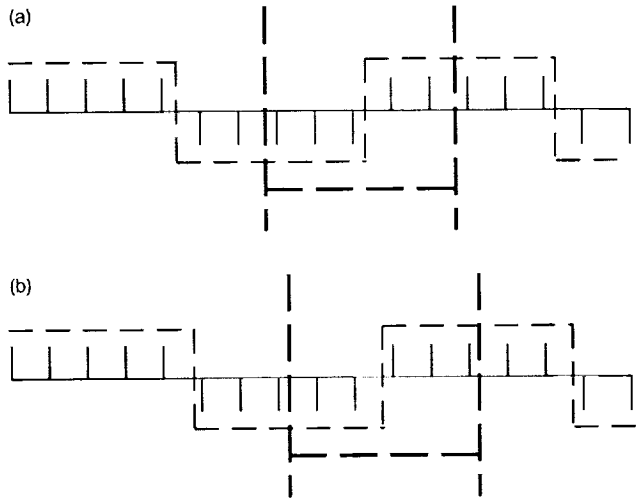

Fig. 2. The effect of an odd number of samples per symbol: (a) three samples from the first symbol and two samples from the second symbol and (b) two samples from the first symbol and three samples from the second symbol.

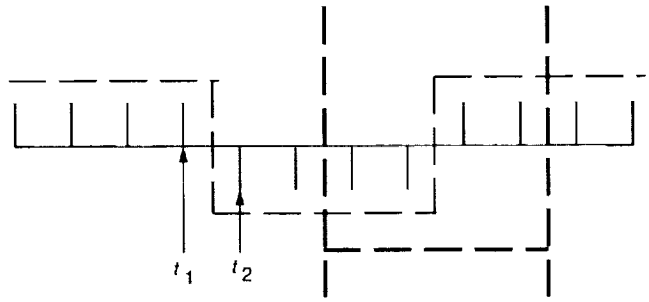

Fig. 3. The phase ambiguity phenomenon.

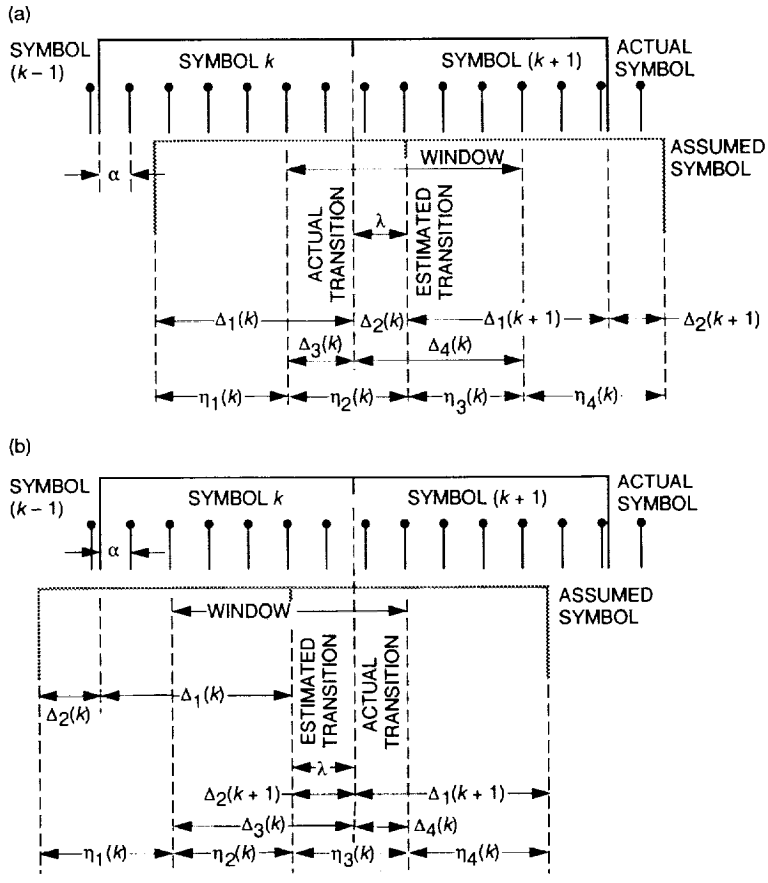

Fig. 4. Definitions of Δ_i functions: (a) $\lambda \geq 0$ and (b) $\lambda \leq 0$.

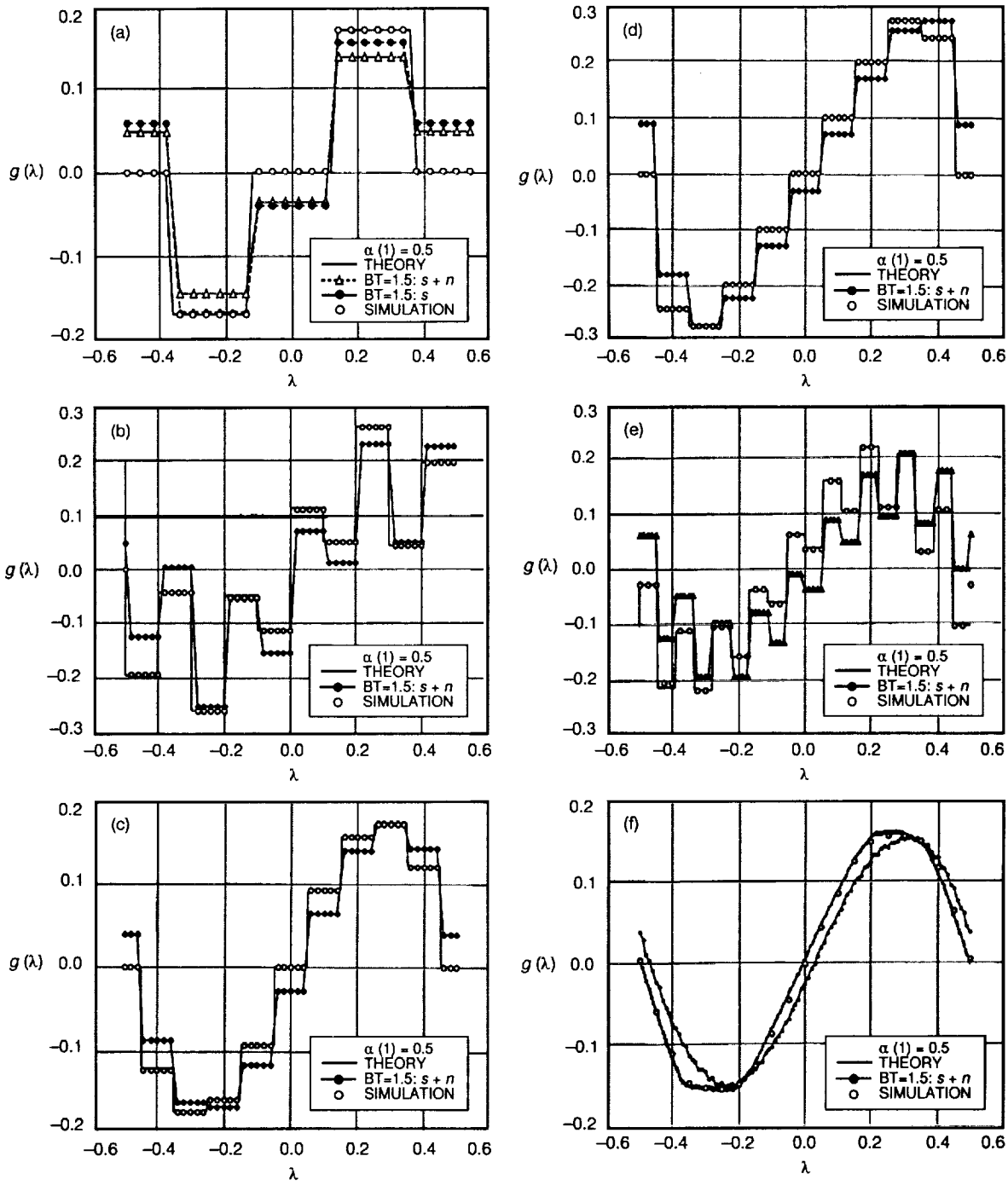

Fig. 5. S curves for various β's: (a) $\beta = 4$, symbol SNR = 3 dB, (b) $\beta = 5$, symbol SNR = 3 dB, (c) $\beta = 10$, symbol SNR = 3 dB, (d) $\beta = 10$, symbol SNR = 10 dB, (e) $\beta = 4.5$, symbol SNR = 3 dB, and (f) $\beta = 4.74$, symbol SNR = 3 dB.

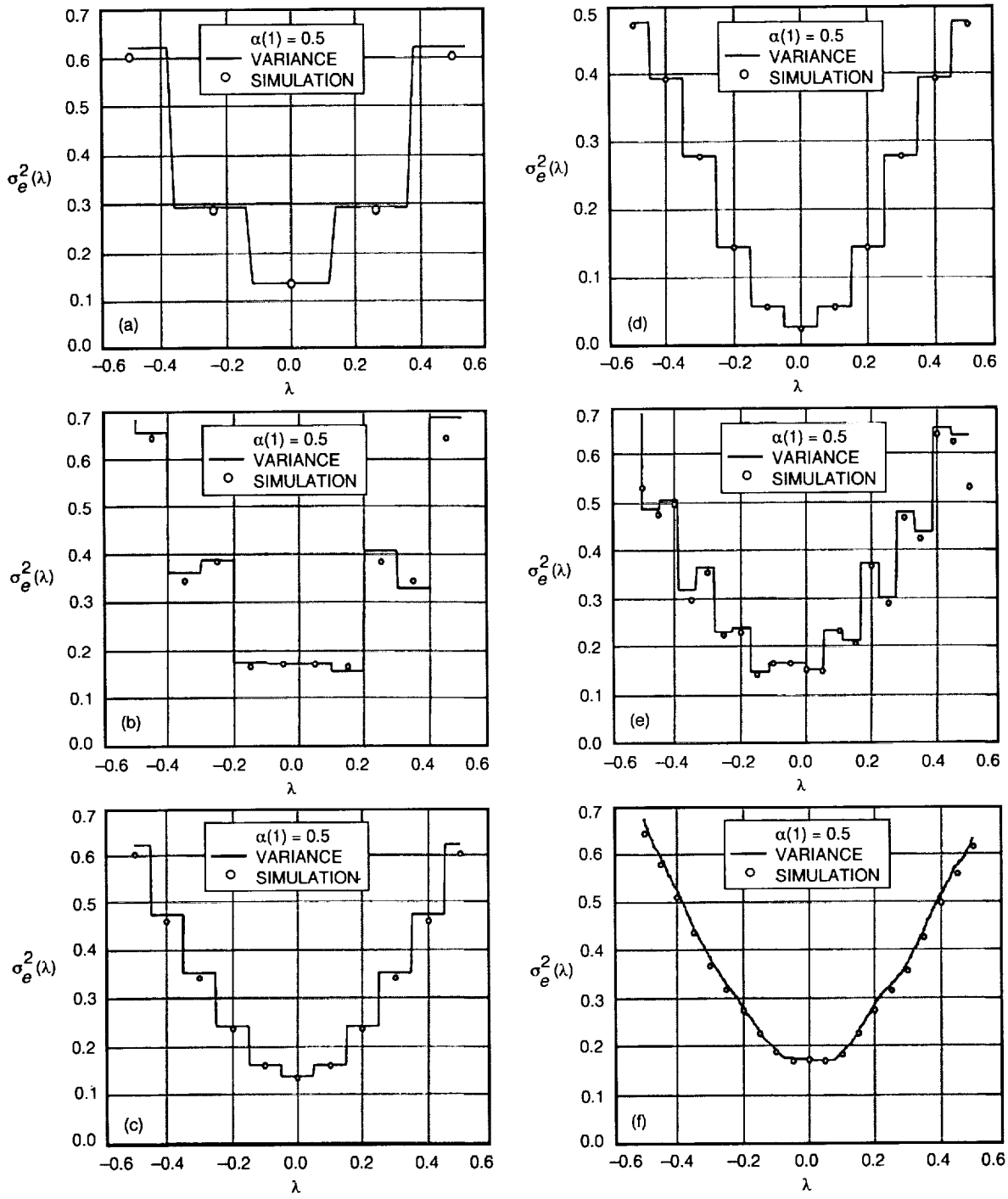

Fig. 6. Open-loop variance $\sigma_e^2(\lambda)$: (a) $\beta = 4$, symbol SNR = 3 dB, (b) $\beta = 5$, symbol SNR = 3 dB, (c) $\beta = 10$, symbol SNR = 3 dB, (d) $\beta = 10$, symbol SNR = 10 dB, (e) $\beta = 4.5$, symbol SNR = 3 dB, and (f) $\beta = 4.74$, symbol SNR = 3 dB.

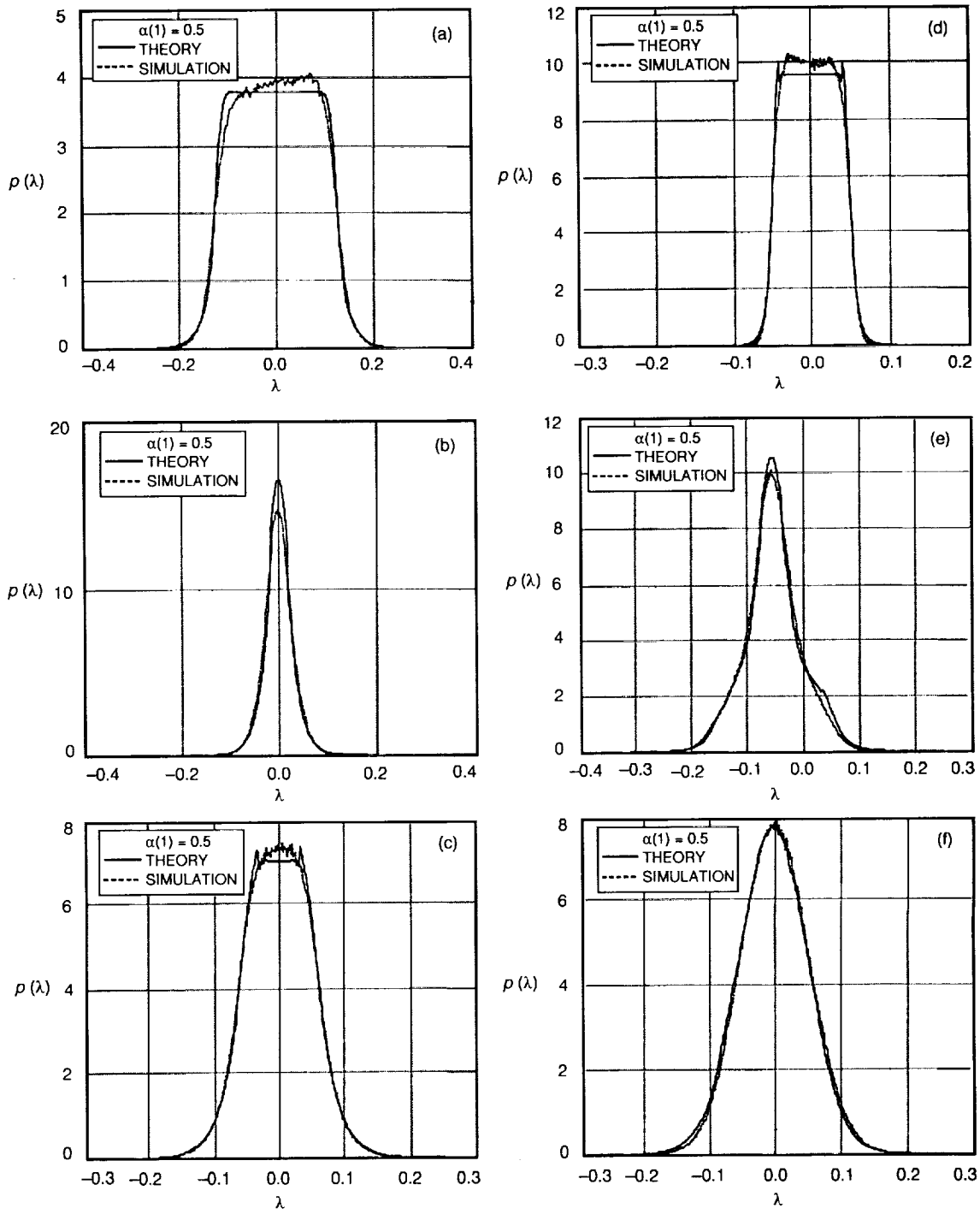

Fig. 7. Probability density function $p(\lambda)$: (a) $\beta = 4$, symbol SNR = 3 dB, (b) $\beta = 5$, symbol SNR = 3 dB, (c) $\beta = 10$, symbol SNR = 3 dB, (d) $\beta = 10$, symbol SNR = 10 dB, (e) $\beta = 4.5$, symbol SNR = 3 dB, and (f) $\beta = 4.74$, symbol SNR = 3 dB.

Appendix

Definition of Variables $v_1(k)$, $v_2(k)$, $v_3(k)$, $v_4(k)$, and $v_5(k)$

$$
\begin{aligned}
v_1(k) = {} & \left(\left[\Delta_3^2(k) + \Delta_4^2(k) + \frac{\beta[\Delta_3(k) + \Delta_4(k)]}{2R_s}\right]\right. \\
& \left.\times\,[4 + (\operatorname{erf}\,[r_1(k)] - \operatorname{erf}\,[r_2(k)])(\operatorname{erf}\,[r_1(k+1)] + \operatorname{erf}\,[r_2(k+1)])]\right) \\
& - 2\Delta_3(k)\Delta_4(k)(\operatorname{erf}\,[r_1(k)] + \operatorname{erf}\,[r_2(k)])(\operatorname{erf}\,[r_1(k+1)] + \operatorname{erf}\,[r_2(k+1)]) \\
& + (\Delta_3(k) + \Delta_2(k))^2 r_3(k)\left((\operatorname{erf}\,[r_1(k+1)] + \operatorname{erf}\,[r_2(k+1)])\right. \\
& \left.\times\left(-\frac{\Delta_1(k) - \Delta_2(k)}{\Delta_1(k) + \Delta_2(k)}\exp\,[-r_1^2(k+1)] + \exp\,[-r_2^2(k+1)]\right)\right) \\
& - (\Delta_4(k) - \Delta_2(k))^2 r_3(k+1)\left((\operatorname{erf}\,[r_1(k)] - \operatorname{erf}\,[r_2(k)])\right. \\
& \left.\times\,\frac{\Delta_1(k+1) - \Delta_2(k+1)}{\Delta_1(k+1) + \Delta_2(k+1)}(\exp\,[-r_1^2(k+1)] + \exp\,[-r_2^2(k+1)])\right) \quad \text{for } \lambda \geq 0
\end{aligned}
\tag{A-1}
$$

$$
\begin{aligned}
v_1(k) = {} & \left(\left[\Delta_3^2(k) + \Delta_4^2(k) + \frac{\beta[\Delta_3(k) + \Delta_4(k)]}{2R_s}\right]\right. \\
& \left.\times\,[4 + (\operatorname{erf}\,[r_1(k)] - \operatorname{erf}\,[r_2(k)])(\operatorname{erf}\,[r_1(k+1)] + \operatorname{erf}\,[r_2(k+1)])]\right) \\
& - 2\Delta_3(k)\Delta_4(k)(\operatorname{erf}\,[r_1(k)] + \operatorname{erf}\,[r_2(k)])(\operatorname{erf}\,[r_1(k+1)] + \operatorname{erf}\,[r_2(k+1)]) \\
& + (\Delta_3(k) - \Delta_2(k))^2 r_3(k)\left((\operatorname{erf}\,[r_1(k+1)] + \operatorname{erf}\,[r_2(k+1)])\right. \\
& \left.\times\left(-\frac{\Delta_1(k) - \Delta_2(k)}{\Delta_1(k) + \Delta_2(k)}\exp\,[-r_1^2(k+1)] + \exp\,[-r_2^2(k+1)]\right)\right) \\
& - (\Delta_4(k) + \Delta_2(k))^2 r_3(k+1)\left((\operatorname{erf}\,[r_1(k)] - \operatorname{erf}\,[r_2(k)])\right. \\
& \left.\times\left(\frac{\Delta_1(k+1) - \Delta_2(k+1)}{\Delta_1(k+1) + \Delta_2(k+1)}\exp\,[-r_1^2(k+1)] + \exp\,[-r_2^2(k+1)]\right)\right) \quad \text{for } \lambda < 0
\end{aligned}
\tag{A-2}
$$

$$
\begin{aligned}
v_2(k) = {} & 2\Delta_3(k)(r_3(k)[\Delta_3(k) + \Delta_2(k)](\text{erf}\,[r_1(k+1)] + \text{erf}\,[r_2(k+1)]) \\
& \times (\exp\,[-r_1^2(k+1)] - \exp\,[-r_2^2(k+1)]) - r_3(k+1)[\Delta_4(k) - \Delta_2(k)] \\
& \times (\text{erf}\,[r_1(k)] + \text{erf}\,[r_2(k)])(\exp\,[-r_1^2(k+1)] + \exp\,[-r_2^2(k+1)])) \\
& + 2\Delta_4(k)(-r_3(k)[\Delta_3(k) + \Delta_2(k)](\text{erf}\,[r_1(k+1)] + \text{erf}\,[r_2(k+1)]) \\
& \times (\exp\,[-r_1^2(k+1)] + \exp\,[-r_2^2(k+1)]) + r_3(k+1)[\Delta_4(k) - \Delta_2(k)] \\
& \times (\text{erf}\,[r_1(k)] - \text{erf}\,[r_2(k)])(\exp\,[-r_1^2(k+1)] + \exp\,[-r_2^2(k+1)])) \quad \text{for } \lambda \geq 0
\end{aligned} \tag{A-3}
$$

$$
\begin{aligned}
v_2(k) = {} & 2\Delta_3(k)(r_3(k)[\Delta_3(k) + \Delta_2(k)](\text{erf}\,[r_1(k+1)] - \text{erf}\,[r_2(k+1)]) \\
& \times (\exp\,[-r_1^2(k+1)] + \exp\,[-r_2^2(k+1)]) - r_3(k+1)[\Delta_4(k) - \Delta_2(k)] \\
& \times (\text{erf}\,[r_1(k)] + \text{erf}\,[r_2(k)])(\exp\,[-r_1^2(k+1)] + \exp\,[-r_2^2(k+1)])) \\
& + 2\Delta_4(k)(-r_3(k)[\Delta_3(k) + \Delta_2(k)](\text{erf}\,[r_1(k+1)] + \text{erf}\,[r_2(k+1)]). \\
& \times (\exp\,[-r_1^2(k+1)] + \exp\,[-r_2^2(k+1)]) + r_3(k+1)[\Delta_4(k) - \Delta_2(k)] \\
& \times (\text{erf}\,[r_1(k)] + \text{erf}\,[r_2(k)])(\exp\,[-r_1^2(k+1)] - \exp\,[-r_2^2(k+1)])) \quad \text{for } \lambda < 0
\end{aligned} \tag{A-4}
$$

$$
\begin{aligned}
v_3(k) = {} & \Delta_3(k)\Delta_3(k+1)[(\text{erf}\,[r_1(k)] + \text{erf}\,[r_2(k)])(\text{erf}\,[r_1(k+1)] + \text{erf}\,[r_2(k+1)])] \\
& - (\Delta_3(k)\Delta_4(k+1)(\text{erf}\,[r_1(k)] + \text{erf}\,[r_2(k)])((\text{erf}\,[r_1(k+1)] - \text{erf}\,[r_2(k+1)]) \\
& + (\text{erf}\,[r_1(k+2)] + \text{erf}\,[r_2(k+2)]))) \\
& - \Delta_3(k+1)\Delta_4(k)(4 + [(\text{erf}\,[r_1(k)] - \text{erf}\,[r_2(k)])(\text{erf}\,[r_1(k+1)] \\
& + \text{erf}\,[r_2(k+1)])] + (\text{erf}\,[r_1(k+1)] - \text{erf}\,[r_2(k+1)]) \\
& \times (\text{erf}\,[r_1(k+2)] + \text{erf}\,[r_2(k+2)])) \\
& + \Delta_4(k)\Delta_4(k+1)((\text{erf}\,[r_1(k)] - \text{erf}\,[r_2(k)])((\text{erf}\,[r_1(k+1)] - \text{erf}\,[r_2(k+1)]) \\
& + (\text{erf}\,[r_1(k+2)] + \text{erf}\,[r_2(k+2)])) + (\text{erf}\,[r_1(k+1)] + \text{erf}\,[r_2(k+1)]) \\
& \times (\text{erf}\,[r_1(k+2)] + \text{erf}\,[r_2(k+2)])) \quad \text{for } \lambda \geq 0
\end{aligned} \tag{A-5}
$$

$$
\begin{aligned}
v_3(k) = {} & \Delta_4(k)\Delta_4(k+1)[(\text{erf}\,[r_1(k+1)] + \text{erf}\,[r_2(k+1)]) \\
& \times (\text{erf}\,[r_1(k+2)] + \text{erf}\,[r_2(k+2)])] \\
& - (\Delta_3(k)\Delta_4(k+1)(\text{erf}\,[r_1(k+2)] + \text{erf}\,[r_2(k+2)])((\text{erf}\,[r_1(k)] + \text{erf}\,[r_2(k)]) \\
& + (\text{erf}\,[r_1(k+1)] - \text{erf}\,[r_2(k+1)])))) \\
& - \Delta_3(k+1)\Delta_4(k)(4 + [(\text{erf}\,[r_1(k)] + \text{erf}\,[r_2(k)])(\text{erf}\,[r_1(k+1)] \\
& - \text{erf}\,[r_2(k+1)])] + (\text{erf}\,[r_1(k+2)] - \text{erf}\,[r_2(k+2)]) \\
& \times (\text{erf}\,[r_1(k+1)] + \text{erf}\,[r_2(k+1)])) \\
& + \Delta_3(k)\Delta_3(k+1)((\text{erf}\,[r_1(k)] + \text{erf}\,[r_2(k)])((\text{erf}\,[r_1(k+2)] - \text{erf}\,[r_2(k+2)]) \\
& + (\text{erf}\,[r_1(k+1)] + \text{erf}\,[r_2(k+1)])) + (\text{erf}\,[r_1(k+1)] - \text{erf}\,[r_2(k+1)]) \\
& \times (\text{erf}\,[r_1(k+2)] - \text{erf}\,[r_2(k+2)])) \quad \text{for } \lambda < 0
\end{aligned} \tag{A-6}
$$

$$
\begin{aligned}
v_4(k) = {} & (r_3(k)[\Delta_3(k) + \Delta_2(k)]r_3(k+1)[\Delta_3(k+1) + \Delta_2(k+1)](\exp\,[-r_1^2(k+1)] \\
& + \exp\,[-r_2^2(k+1)])(\exp\,[-r_1^2(k+1)] + \exp\,[-r_2^2(k+1)])) \\
& + (\Delta_3(k)[\Delta_4(k) - \Delta_2(k)]r_3(k+1)[\Delta_4(k+1) - \Delta_2(k+1)] \\
& \times (\exp\,[-r_1^2(k+1)] + \exp\,[-r_2^2(k+1)])(\exp\,[-r_1^2(k+2)] + \exp\,[-r_2^2(k+2)])) \\
& - (r_3(k)[\Delta_3(k) + \Delta_2(k)]r_3(k+1)[\Delta_4(k+1) - \Delta_2(k+1)](\exp\,[-r_1^2(k+1)] \\
& + \exp\,[-r_2^2(k+1)])(\exp\,[-r_1^2(k+2)] + \exp\,[-r_2^2(k+2)])) \\
& - (\text{erf}\,[r_1(k)] - \text{erf}\,[r_2(k)])\sqrt{\frac{\beta(\Delta_4(k) - \Delta_2(k))^4}{\pi R_s}} \\
& \times \left(\left(\frac{(\Delta_1(k+1) - \Delta_2(k+1))}{(\Delta_4(k) - \Delta_2(k) + \Delta_3(k+1) - \Delta_2(k+1))^{1.5}} \exp\left[-(\Delta_1(k+1) - \Delta_2(k+1))^2 \frac{R_s}{\beta}\right.\right.\right. \\
& \left.\left.\left. \times \left| \frac{1}{[\Delta_3(k) + \Delta_2(k)]\left[1 + \frac{[\Delta_3(k+1) + \Delta_2(k+1)]}{[\Delta_4(k) - \Delta_2(k)]}\right]} - \frac{1}{[\Delta_4(k) - \Delta_2(k)]} \right| \right]\right) \\
& + \left(\frac{(\Delta_1(k+1) + \Delta_2(k+1))}{(\Delta_4(k) - \Delta_2(k) + \Delta_3(k+1) - \Delta_2(k+1))^{1.5}} \exp\left[-(\Delta_1(k+1) + \Delta_2(k+1))^2 \frac{R_s}{\beta}\right.\right.
\end{aligned}
$$

$$\times \left| \frac{1}{[\Delta_3(k)+\Delta_2(k)]\left[1+\frac{[\Delta_3(k+1)+\Delta_2(k+1)]}{[\Delta_4(k)-\Delta_2(k)]}\right]} - \frac{1}{[\Delta_4(k)-\Delta_2(k)]} \right| \Bigg) \Bigg)$$

$$+ (\text{erf}\,[r_1(k+2)] + \text{erf}\,[r_2(k+2)]) \sqrt{\frac{\beta(\Delta_4(k)-\Delta_2(k))^4}{\pi R_s}}$$

$$\times \left(-\left(\frac{(\Delta_1(k+1)-\Delta_2(k+1))}{(\Delta_4(k)-\Delta_2(k)+\Delta_3(k+1)-\Delta_2(k+1))^{1.5}} \exp\left[-(\Delta_1(k+1)-\Delta_2(k+1))^2 \frac{R_s}{\beta} \right.\right.\right.$$

$$\times \left| \frac{1}{[\Delta_3(k)+\Delta_2(k)]\left[1+\frac{[\Delta_3(k+1)+\Delta_2(k+1)]}{[\Delta_4(k)-\Delta_2(k)]}\right]} - \frac{1}{[\Delta_4(k)-\Delta_2(k)]} \right| \Bigg)$$

$$+ \left(\frac{(\Delta_1(k+1)+\Delta_2(k+1))}{(\Delta_4(k)-\Delta_2(k)+\Delta_3(k+1)-\Delta_2(k+1))^{1.5}} \exp\left[-(\Delta_1(k+1)+\Delta_2(k+1))^2 \frac{R_s}{\beta} \right.\right.$$

$$\times \left| \frac{1}{[\Delta_3(k)+\Delta_2(k)]\left[1+\frac{[\Delta_3(k+1)+\Delta_2(k+1)]}{[\Delta_4(k)-\Delta_2(k)]}\right]} - \frac{1}{[\Delta_4(k)-\Delta_2(k)]} \right| \Bigg) \Bigg) \tag{A-7}$$

$$\begin{aligned}
v_5(k) = {} & r_3(k)\,[\Delta_3(k)+\Delta_2(k)](\exp\,[-r_1^2(k+1)] + \exp\,[-r_2^2(k+1)])(\Delta_3(k+1)(\text{erf}\,[r_1(k+1)] + \text{erf}\,[r_2(k+1)]) \\
& - \Delta_4(k+1)[(\text{erf}\,[r_1(k+1)] - \text{erf}\,[r_2(k+1)]) + (\text{erf}\,[r_1(k+2)] + \text{erf}\,[r_2(k+2)])]) + \Delta_3(k+1) \\
& \times [\Delta_3(k+1)+\Delta_2(k+1)](\Delta_3(k)(\text{erf}\,[r_1(k)] + \text{erf}\,[r_2(k)])(\exp\,[-r_1^2(k+1)] + \exp\,[-r_2^2(k+1)]) \\
& - \Delta_4(k)[(\text{erf}\,[r_1(k+2)] + \text{erf}\,[r_2(k+2)])(\exp\,[-r_1^2(k+1)] - \exp\,[-r_2^2(k+1)]) \\
& + (\text{erf}\,[r_1(k)] - \text{erf}\,[r_2(k)])(\exp\,[-r_1^2(k+1)] + \exp\,[-r_2^2(k+1)])]) \\
& + r_3(k)[\Delta_4(k)-\Delta_2(k)](\Delta_4(k+1)[(\text{erf}\,[r_1(k)] - \text{erf}\,[r_2(k)]) \\
& \times (\exp\,[-r_1^2(k+1)] - \exp\,[-r_2^2(k+1)]) + (\text{erf}\,[r_1(k+2)] + \text{erf}\,[r_2(k+2)]) \\
& \times (\exp\,[-r_1^2(k+1)] + \exp\,[-r_2^2(k+1)])] - \Delta_3(k+1)[(\text{erf}\,[r_1(k+2)] + \text{erf}\,[r_2(k+2)]) \\
& \times (\exp\,[-r_1^2(k+1)] - \exp\,[-r_2^2(k+1)]) + (\text{erf}\,[r_1(k)] - \text{erf}\,[r_2(k)]) \\
& \times (\exp\,[-r_1^2(k+1)] + \exp\,[-r_2^2(k+1)])]) \\
& + r_3(k+1)[\Delta_4(k+1)-\Delta_2(k+1)](\exp\,[-r_1^2(k+2)] + \exp\,[-r_2^2(k+2)]) \\
& \times (\Delta_4(k)[(\text{erf}\,[r_1(k)] - \text{erf}\,[r_2(k)]) + (\text{erf}\,[r_1(k+1)] + \text{erf}\,[r_2(k+1)])] \\
& - \Delta_3(k)(\text{erf}\,[r_1(k)] + \text{erf}\,[r_2(k)]))
\end{aligned} \tag{A-8}$$