Controlled-Root Formulation for Digital Phase-Locked Loops

S. A. STEPHENS

J. B. THOMAS
Jet Propulsion Laboratory

In a new formulation for digital phase-locked loops, loop-filter constants are determined from loop roots that can each be selectively placed in the s-plane on the basis of a new set of parameters, each with simple and direct physical meaning in terms of loop noise bandwidth, root-specific decay rate, or root-specific damping. Loops of first to fourth order are treated in the continuous-update approximation ($B_L T \to 0$) and in a discrete-update formulation with arbitrary $B_L T$. Deficiencies of the continuous-update approximation in large-$B_L T$ applications are avoided in the new discrete-update formulation. A new method for direct, transient-free acquisition with third- and fourth-order loops can improve the versatility and reliability of acquisition with such loops.

Manuscript received January 21, 1992; revised May 25, 1993.

IEEE Log No. T-AES/31/1/07982.

This work was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

Authors' address: Jet Propulsion Laboratory, MS 238-600, 4800 Oak Grove Dr., Pasadena, CA 91109-8099.

I. INTRODUCTION

Previous analyses (e.g., [1, 2]) of digital phase-locked loops (DPLLs) are based on the traditions of analog loops and introduce analog considerations, such as loop-filter time constants and uncompensated gain variations, that are unnecessary for "fully digital" loops. This reliance on analog tradition makes digital-loop analysis unnecessarily cumbersome and impedes the progress of analysts with little analog training. Theory for digital loops can be developed from first principles without reference to analog concepts. With an appropriately formulated, fully digital analysis, one discovers that DPLL theory and design become more straightforward and understandable (particularly for third- and fourth-order loops) and that loop performance is more easily controlled for "high-gain" loops.

In the new formulation, loop-filter constants are determined from loop roots that can be selectively placed in the s-plane in pairwise fashion on the basis of a new set of independent parameters, where each parameter has a simple and direct physical meaning in terms of loop noise bandwidth, root-specific decay rate, or root-specific damping. For example, a simple choice of parameter values will automatically give a loop a selected loop bandwidth and supercritically damped behavior (i.e., all roots real, negative, and equal). Thus, the need to solve for root location as a function of traditional loop parameters (e.g., B_L, r, and k for a third-order loop [2]) is eliminated and analysis is simplified. The new parameterization is made feasible in a practical sense by the fact that digital loops can often be designed so that they do not suffer significantly from the effects of amplitude variations. That is, variations in signal amplitude, due to either gain instability or signal-power changes, can often be accounted for by using a normalized phase extractor [3]. In this case, a fully digital DPLL does not require the analysis or precautions [e.g., 2] necessitated in other DPLL designs by potential amplitude variations. Even when amplitude variations cannot be removed, the new formulation can be used to generate a reference or target configuration whose response can then be tested with respect to amplitude variations.

Previous analyses (e.g., [1, 2]) of discrete-update (DU) loops have started with the closed-loop equation in the "continuous-update" (CU) limit in which $B_L T \to 0$, where B_L is the loop noise bandwidth and T is the loop update interval. For sufficiently small $B_L T$ (e.g., $B_L T \leq 0.02$), the CU approximation can provide an adequate starting point for analysis and design of DU loops. When $B_L T$ is increased in this approximation to larger, high-gain values, however, loop roots move away from their initial small-$B_L T$ damping and the loop diverges from expected behavior. Furthermore, actual loop noise bandwidth increases more rapidly than the input "loop parameter bandwidth" and must be separately computed. To overcome these shortcomings, a solution for DU loops is developed in which B_L is true loop noise bandwidth for all allowed values of B_LT, and root locations follow constant-damping paths as a function of B_LT. These features, which are an automatic benefit of the new approach to parameterization mentioned above, can provide, for example, supercritically damped response for all allowed B_LT values. Both the CU and DU approaches can be applied to a "delay-locked" loop, implemented, for example, to steer one clock into synchronization with a second clock on the basis of measured synchronization error.

The analysis is extended to fourth-order loops because of the potential advantages of such loops. In some spacecraft applications, for example, loop bandwidth can be set to a smaller value for a fourth-order loop than for a third. Consequently, lower signal strengths can be reliably tracked. Fourth-order DPLLs, unlike their analog counterparts, are easy to design and implement, given the new parameterization. Accurate placement of loop roots results from a simple selection of parameter values rather than complicated analog circuit design.

Acquisition in third- and fourth-order loops should be carefully crafted. A past approach for a third-order loop has been to acquire first with a second-order loop with "wide bandwidth" and then hand over tracking to a "narrow bandwidth" third-order loop. This approach sacrifices the opportunity of directly acquiring weaker signals with the narrow-bandwidth third-order loop. Given sufficient *a priori* phase information, high-order DPLLs, unlike similar analog loops, can be easily initialized by setting all loop sums so that such digital loops can acquire directly, without first acquiring with a lower-order loop. Furthermore, if sufficient *a priori* information is supplied, DPLLs will start off tracking in-lock, with no transients. *A priori* information, in the form of signal phase and its derivatives, can be supplied, for example, on the basis of fast Fourier transform (FFT) analysis and/or spacecraft trajectory information.

To establish a foundation for analysis, a high-level description of a DPLL is presented in Section II. For loops of first to fourth order, Section III uses the new parameterization to derive a CU-limit solution while Section IV develops a systematic approach from which numerical, controlled-root solutions to the DU loop can be derived. The analysis is organized so that a step-by-step comparison of the DU and CU formulations can be easily made. Solutions are given for loops with either phase and phase-rate feedback or phase-rate-only feedback, and with the computation delay for closing the loop set to either zero or one update interval. To tie in with traditional analysis, the new loop parameters are related to old loop parameters in the CU limit. Section V presents a method for direct transient-free acquisition with third- and fourth-order loops, a method that can improve the versatility and reliability of acquisition with such loops. To illustrate the performance of high-B_LT loops, Section VI presents results for two measures of loop performance: mean time to first cycle slip and steady-state phase error.

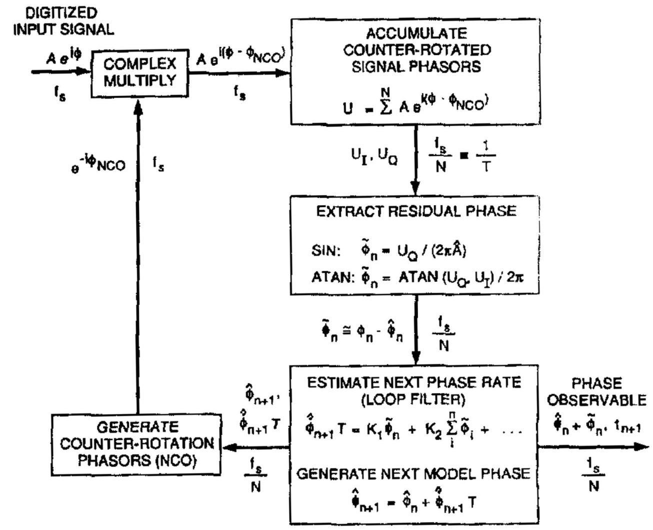

Fig. 1. Schematic illustration of DPLL with phase/phase-rate updates and no computation delay.

II. HIGH-LEVEL DESCRIPTION OF A DPLL

A. Elements of a DPLL

The block diagram in Fig. 1 shows the basic elements of a DPLL. Since detailed explanations of these elements can be found elsewhere (e.g., [3]), they will not be reintroduced here. The example DPLL shown in Fig. 1 is based on "immediate update" of the loop filter (i. e., no computation delay) and feedback of phase as well as phase rate. Alternate DPLL designs might feed back only phase rate and/or have a substantial computation delay ("transport lag").

An incoming signal is sampled in quadrature at a high rate (f_s) and then counter-rotated sample by sample at this high rate with model phasors generated by a number-controlled oscillator (NCO) as directed by loop feedback. The resulting complex counter-rotated signal, which should have very low frequency, is then accumulated over an update interval of length T in order to reduce the data rate. A phase extractor then processes the resulting complex sum to produce a value for residual phase for the given interval (nth). Two choices for phase extractor are shown, arctangent and normalized sine extractor [3]. (In a normalized sine extractor, estimated signal amplitude is used to remove amplitude effects from the Q component.) For an ideal phase extractor, the nth residual phase $\tilde{\phi}_n$ is obtained in units of cycles and is equivalent to the difference of the nth input signal phase ϕ_n and the nth model phase $\hat{\phi}_n$ as obtained through feedback:

$$\tilde{\phi}_n = \phi_n - \hat{\phi}_n \tag{1}$$

with each referenced to the center of the sum interval. Following the usual linear model, the theory presented here will be based on the phase-extractor approximation of (1). In practice, actual residual phase can deviate from this linear model due to nonlinearity in the phase extractor, cycle ambiguities, or inaccurate amplitude normalization. Nevertheless, given a well-designed loop and adequate signal-to-noise ratio (SNR), the approximation of (1) will provide an accurate model for loop behavior when the tracking error is sufficiently small. In a delay-locked loop, the residual delay does not necessarily suffer nonlinearities or ambiguities and the linear model can be an accurate model over the full operating range.

Residual phase is passed to the loop filter to assist in the generation of model phase rate. The loop filter generates the estimate of phase rate for the $(n+1)$th interval in the form of phase change per update interval, $\hat{\dot{\phi}}_{n+1}T$.

In a loop with phase and phase-rate feedback, as in Fig. 1, an estimate of the next model phase, the $(n+1)$th, is projected ahead to interval center by adding this estimated phase change to the previous nth model phase:

$$\hat{\phi}_{n+1} = \hat{\phi}_n + \hat{\dot{\phi}}_{n+1}T. \tag{2}$$

The $(n+1)$th model phase, along with estimated phase rate, is used at the completion of the nth accumulation to initialize the phase and rate registers of the NCO in a manner [3] that causes the NCO to produce over the $(n+1)$th interval, phase values characterized by said rate and center-interval phase. (That is, the rate register of the NCO is initialized with a rate value equivalent to $\hat{\dot{\phi}}_{n+1}T$ and the phase register with a phase value equal to model phase minus one half interval of NCO phase change, $\hat{\phi}_{n+1} - \hat{\dot{\phi}}_{n+1}T/2$. With such initialization, the NCO will generate $\hat{\phi}_{n+1}$ at mid-interval as desired.)

In a loop with rate-only feedback, on the other hand, the NCO rate register is updated at the end of the nth accumulation with a value equivalent to the $(n+1)$th rate estimate. The NCO phase register is left untouched so that NCO phase is "continuous" from interval to interval. The center-interval phase values that are applied by the NCO as a consequence of this feedback approach can be obtained through NCO modeling according to the expression

$$\hat{\phi}_{n+1} = \hat{\phi}_n + \tfrac{1}{2}(\hat{\dot{\phi}}_{n+1}T + \hat{\dot{\phi}}_nT) \tag{3}$$

where a half interval of phase change accumulates due to the nth rate and another half due to the $(n+1)$th rate.

Based on either of these feedback approaches, the loop is closed and a new value for residual phase is produced to repeat the process.

B. Loop Filter

A conventional Nth-order digital loop filter uses residual phase values $\tilde{\phi}_i$ to estimate phase rate for the $(n+1)$th interval according to

$$\hat{\dot{\phi}}_{n+1}T = K_1\tilde{\phi}_{n-n_c} + K_2\sum_{i=1}^{n-n_c}\tilde{\phi}_i + K_3\sum_{i=1}^{n-n_c}\sum_{j=1}^{i}\tilde{\phi}_j + K_4\sum_{i=1}^{n-n_c}\sum_{j=1}^{i}\sum_{k=1}^{j}\tilde{\phi}_k + \cdots \tag{4}$$

where $\hat{\dot{\phi}}_{n+1}T$ is phase change per update interval T and where the sequence extends to the K_N term for an Nth-order loop. The loop constants K_l are specified below. The variable n_c is the computation delay [3], with $n_c = 0$ for "immediate" updates and $n_c = 1$ for a computation delay of one update interval. If update computations are sufficiently fast, immediate updates can be applied, but possibly at the cost of a small amount of sampled data lost while completing the update computations.

III. CONTINUOUS-UPDATE APPROXIMATION

A. Closed-Loop Equation

In many applications, the update interval T is much shorter than all other filter time scales, and considerable insight may be gained by writing (4) in the CU limit, $T \to 0$. To facilitate this, we can define without loss of generality CU loop constants κ_i by means of the relation

$$K_i = \kappa_i T^i \qquad \text{for} \quad i = 1,\ldots,N \tag{5}$$

so that (4) becomes

$$\frac{\hat{\phi}_{n+1} - \hat{\phi}_n}{T} = \kappa_1\tilde{\phi}_{n-n_c} + \kappa_2\sum_{i=1}^{n-n_c}T\tilde{\phi}_i + \kappa_3\sum_{i=1}^{n-n_c}T\sum_{j=1}^{i}T\tilde{\phi}_j + \kappa_4\sum_{i=1}^{n-n_c}T\sum_{j=1}^{i}T\sum_{k=1}^{j}T\tilde{\phi}_k + \cdots \tag{6}$$

where estimated phase rate has been rewritten on the basis of (2) under the assumption that the NCO is updated in both phase and phase rate. In the limit $T \to 0$, T can be replaced by dt, the first term becomes a derivative, and the sums become integrals:

$$\frac{d}{dt}\hat{\phi} = \kappa_1\tilde{\phi} + \kappa_2\int_{t_0}^{t} dt'\tilde{\phi}(t') + \kappa_3\int_{t_0}^{t} dt'\int_{t_0}^{t'} dt''\tilde{\phi}(t'')$$
$$+ \kappa_4\int_{t_0}^{t} dt'\int_{t_0}^{t'} dt''\int_{t_0}^{t''} dt'''\tilde{\phi}(t''') + \cdots \qquad (7)$$

where t_0 is the starting time for accumulation and $\tilde{\phi}$ is the continuous form of (1):

$$\tilde{\phi}(t) = \phi(t) - \hat{\phi}(t). \qquad (8)$$

Thus, in this limit, the basic equation governing "NCO rate" is the same equation that governs the voltage-controlled oscillator (VCO) rate of an analog loop, given perfect integrators. Note that, for sufficiently small T, all DU loops are described by (7) and (8), so that distinctions such as computation delay and phase and phase-rate feedback versus phase-rate-only feedback are not significant with regard to loop behavior.

Solutions of the closed-loop equation can be based on the theory of differential equations after substituting (8) in (7) and differentiating the result $N-1$ times with respect to time:

$$\frac{d^N}{dt^N}\hat{\phi} + \kappa_1\frac{d^{N-1}}{dt^{N-1}}\hat{\phi} + \kappa_2\frac{d^{N-2}}{dt^{N-2}}\hat{\phi} + \cdots + \kappa_N\hat{\phi}$$
$$= \kappa_1\frac{d^{N-1}}{dt^{N-1}}\phi + \kappa_2\frac{d^{N-2}}{dt^{N-2}}\phi + \cdots + \kappa_N\phi. \qquad (9)$$

Solution of this differential equation gives the behavior of model phase $\hat{\phi}$ in response to input signal phase ϕ when $\phi - \hat{\phi}$ is small.

B. Transfer Function and Loop Noise Bandwidth

To find the frequency response of the loop [4], take the Laplace transform of both sides of (9), and utilize the relation

$$\mathcal{L}\left\{\frac{d^n\phi(t)}{dt^n}\right\} = s^n\mathcal{L}\{\phi(t)\} \qquad (10)$$

where $\mathcal{L}\{\ \}$ represents a Laplace transform. This produces

$$[s^N + \kappa_1 s^{N-1} + \kappa_2 s^{N-2} + \cdots + \kappa_N]\hat{\Phi}(s)$$
$$= [\kappa_1 s^{N-1} + \kappa_2 s^{N-2} + \cdots + \kappa_N]\Phi(s) \qquad (11)$$

where $\Phi(s)$ and $\hat{\Phi}(s)$ are the Laplace transforms of $\phi(t)$ and $\hat{\phi}(t)$, respectively. The *closed-loop transfer function* $H(s)$ is defined by $\hat{\Phi}(s) = H(s)\Phi(s)$, so we have

$$H(s) = \frac{\kappa_1 s^{N-1} + \kappa_2 s^{N-2} + \cdots + \kappa_N}{s^N + \kappa_1 s^{N-1} + \kappa_2 s^{N-2} + \cdots + \kappa_N}. \qquad (12)$$

The frequency response of the loop is obtained by substituting $s = i2\pi f$ in (12), where f is frequency in Hertz. The *single-sided loop noise bandwidth* B_L for the closed loop is defined by

$$2B_L \equiv \int_{-\infty}^{\infty} |H(i2\pi f)|^2 df. \qquad (13)$$

This integral can be evaluated on the basis of of [5, eqn. (3.112)], to find B_L as a function of $\kappa_1, \kappa_2, \ldots, \kappa_N$ in the CU limit. The results are summarized in Table I for loops of first to fourth order.

TABLE I
Loop Bandwidth From Loop Constants in CU Approximation

1st order

$$B_L = \frac{\kappa_1}{4}$$

2nd order

$$B_L = \frac{\kappa_1^2 + \kappa_2}{4\kappa_1} = \frac{\kappa_1}{4}(1 + \alpha_2)$$

3rd order

$$B_L = \frac{\kappa_1^2\kappa_2 - \kappa_1\kappa_3 + \kappa_2^2}{4(\kappa_1\kappa_2 - \kappa_3)} = \frac{\kappa_1}{4}\frac{\alpha_2 - \alpha_3 + \alpha_2^2}{\alpha_2 - \alpha_3}$$

4th order

$$B_L = \frac{\kappa_1^2\kappa_2\kappa_3 - \kappa_1\kappa_3^2 - \kappa_1^3\kappa_4 + \kappa_2^2\kappa_3 - \kappa_1\kappa_2\kappa_4 - \kappa_3\kappa_4}{4(\kappa_1\kappa_2\kappa_3 - \kappa_3^2 - \kappa_1^2\kappa_4)}$$
$$= \frac{\kappa_1}{4}\frac{\alpha_2\alpha_3 - \alpha_3^2 - \alpha_4 + \alpha_2^2\alpha_3 - \alpha_2\alpha_4 - \alpha_3\alpha_4}{\alpha_2\alpha_3 - \alpha_3^2 - \alpha_4}$$

C. Solution to the Homogeneous Equation

Solutions to the homogeneous form of (9) (i.e., with $\phi(t) = 0$) provides information as to the transient behavior and stability of the loop, to the extent that the linear approximation of (1) is a valid model for residual phase. When the roots are unequal (i.e., nondegenerate), the solution to the homogeneous equation is

$$\hat{\phi}(t) = \sum_{i=1}^{N} a_i e^{s_i t} \qquad (14)$$

where each s_i is a root of the characteristic equation

$$s^N + \kappa_1 s^{N-1} + \kappa_2 s^{N-2} + \cdots + \kappa_N = 0 \qquad (15)$$

and where the a_is are amplitudes to be determined by the initial conditions. (Similar equations can be developed for the degenerate cases, but such equations are unnecessary if the degenerate cases are approximated as nondegenerate by introducing an infinitesimal offset between equal roots.) For the loop to be stable, the loop constants must be set to values that cause all roots to fall in the left half plane. With such roots, all terms in (14) will decay exponentially.

D. Loop Constants as a Function of Loop Roots

To express the CU loop constants, the κs, as a function of the roots, equate the coefficient of each term in (15) with the coefficient of the like term in the same polynomial factored into its roots:

$$(s-s_1)(s-s_2)\cdots(s-s_N)=0. \tag{16}$$

The κ's are then given by

$$\kappa_1 = -\sum_{i}^{N} s_i \tag{17}$$

$$\kappa_2 = +\sum_{i<j}^{N} s_i s_j \tag{18}$$

$$\kappa_3 = -\sum_{i<j<k}^{N} s_i s_j s_k \tag{19}$$

$$\vdots$$

$$\kappa_N = (-1)^N (s_1 s_2 s_3 \cdots s_N). \tag{20}$$

Thus, if the roots are known, the loop constants can be easily computed.

E. Parameterization of Loop Roots

Loop noise bandwidth B_L provides us with one independent (selectable) loop parameter with useful physical significance. Traditional loop theory specifies additional parameters at this point (e.g., r, k, ζ^2, ω_n, τ_1, τ_2) to complete the parameterization. However, since traditional parameters are either inappropriate for digital loops or have indirect meaning with respect to transient behavior for loops higher than second order, they are less useful than more carefully chosen parameters.

Insight as to a systematic approach for parameterizing the roots of higher order loops can be gained by considering the nature of the roots for first- and second-order loops. The root of a first order loop ($N = 1$) from (15) is a real number that determines the decay rate of the transient response. Thus, a first-order loop can be trivially parameterized with a real number β representing a decay rate, with the root given by $s = -\beta = -\kappa_1$.

For a second-order loop, on the other hand, the two roots are solutions to the quadratic form ($N = 2$) of (15) and are given by

$$s = -\frac{\kappa_1}{2}(1 \pm (1-4\kappa_2/\kappa_1^2)^{1/2}). \tag{21}$$

This result suggests that the simplest form for parameterizing second-order roots is obtained through use of the discriminant $\eta^2 \equiv 1 - 4\kappa_2/\kappa_1^2$ and a decay-rate parameter defined by $\beta \equiv \kappa_1/2$. These new parameters yield

$$s = -\beta(1 \pm \eta). \tag{22}$$

Note that both β and η^2 are real numbers and that η is either a real number or a purely imaginary number depending on the sign of η^2. As is well known from second-order loop theory, the sign of the discriminant determines loop damping:

$\eta^2 > 0$	two real roots: overdamped
$\eta^2 = 0$	two real, equal roots: critically damped
$\eta^2 < 0$	complex conjugate pair: underdamped/oscillatory

Since η^2 determines damping, it is referred to as the *damping parameter*. For a second-order loop, the damping parameter is related to the traditional damping factor ζ [6] by $\zeta^2 = 1/(1-\eta^2)$ while the decay-rate parameter β is related to the traditional variables of loop frequency ω_n [6] and ζ by $\beta = \zeta\omega_n$.

These results for a second-order loop suggest a systematic method for parameterizing the roots of higher order loops. Since the loop constants are real, all complex roots of higher order loops occur in conjugate pairs. Thus, the N roots of an Nth-order loop can be divided into pairs, with each pair parameterized according to (22). With such parameterization, appropriate selection of β and η^2 for a given pair will allow those two roots to be placed at any allowed locations in the s-plane. Loops of odd order will have a last unpaired real root that will be represented by a decay-rate parameter β, in analogy with a first-order loop.

Based on the above considerations, the N roots of an Nth-order loop can be parameterized for even-order loops as

$$\{s_1,s_2;s_3,s_4;\ldots;s_{N-1},s_N\} = \{-\beta_1(1\pm\eta_1);-\beta_2(1\pm\eta_2);\ldots;-\beta_{N/2}(1\pm\eta_{N/2})\} \tag{23}$$

and for odd-order loops as

$$\{s_1,s_2;s_3,s_4;\ldots;s_N\} = \{-\beta_1(1\pm\eta_1);-\beta_2(1\pm\eta_2);\ldots;-\beta_{(N+1)/2}\} \tag{24}$$

The β_i, η_i^2 parameters retain the same meaning for each root pair as the corresponding parameters for a second-order loop. However, it is emphasized that the second-order-loop relations of β and η^2 to loop constants do not persist for higher order loops. As shown in the next subsection, different relations for connecting loop constants and the new parameters must be used for each loop order.

TABLE II
Loop-Filter Constants in CU Approximation

1^{st} order				
$K_1 = 4B_LT$				
2^{nd} order				
$K_1 = 4B_LT\frac{1}{1+\alpha_2}$		$K_2 = \alpha_2 K_1^2$		
	Controlled-root:	$\alpha_2 = \frac{1-\eta_1^2}{4}$		
	Traditional:	$\alpha_2 = \frac{1}{r}$		
3^{rd} order				
$K_1 = 4B_LT\frac{\alpha_2-\alpha_3}{\alpha_2-\alpha_3+\alpha_2^2}$		$K_2 = \alpha_2 K_1^2$	$K_3 = \alpha_3 K_1^3$	
	Controlled-root:	$\alpha_2 = \frac{2\lambda_2 + (1-\eta_1^2)}{(2+\lambda_2)^2}$	$\alpha_3 = \frac{\lambda_2(1-\eta_1^2)}{(2+\lambda_2)^3}$	
	Traditional:	$\alpha_2 = \frac{1}{r}$	$\alpha_3 = \frac{k}{r^2}$	
4^{th} order				
$K_1 = 4B_LT\frac{\alpha_2\alpha_3-\alpha_3^2-\alpha_4}{\alpha_2\alpha_3-\alpha_3^2-\alpha_4-\alpha_2\alpha_4-\alpha_3\alpha_4+\alpha_2^2\alpha_3}$		$K_2 = \alpha_2 K_1^2$	$K_3 = \alpha_3 K_1^3$	$K_4 = \alpha_4 K_1^4$
	Controlled-root:	$\alpha_2 = \frac{4\lambda_2 + (1-\eta_1^2) + \lambda_2^2(1-\eta_2^2)}{(2+2\lambda_2)^2}$	$\alpha_3 = \frac{2\lambda_2(1-\eta_1^2) + 2\lambda_2^2(1-\eta_2^2)}{(2+2\lambda_2)^3}$	$\alpha_4 = \frac{\lambda_2^2(1-\eta_1^2)(1-\eta_2^2)}{(2+2\lambda_2)^4}$
	Traditional:	$\alpha_2 = \frac{1}{r}$	$\alpha_3 = \frac{k}{r^2}$	$\alpha_4 = \frac{a}{r^3}$

We can choose one root factor, β_1, as the *reference decay-rate parameter*, and form new parameters λ_i which indicate magnitude relative to the first:

$$\beta_i = \lambda_i \beta_1 \qquad \text{for} \quad i \geq 2. \tag{25}$$

The root parameterization is now given by

$$\{s_1, s_2; s_3, s_4; s_5, s_6; \ldots\} = \{-\beta_1(1 \pm \eta_1); -\beta_1\lambda_2(1 \pm \eta_2); -\beta_1\lambda_3(1 \pm \eta_3); \ldots\}. \tag{26}$$

The goal here is to create parameters which dictate the *relative* placement of the roots once B_L has been specified. The parameter β_1, which must be a positive number for loop stability, is made a dependent variable and solved for below as a function of B_L. The N independent (selectable) parameters for the loop are B_L, λ_i, and η_i^2 as defined above.

The λ_is, which we will refer to as the *relative decay-rate parameters*, control the relative magnitudes of different root pairs. Furthermore, we will always choose λ_i to be positive to give exponentially decaying solutions to (9), since β_1 will be a positive real value.

Some interesting values of λs and ηs are: all $\lambda_i = 1$ (all real parts of roots equal); all $\eta_i^2 = 0$, $\lambda_i = 1$ (all roots real and equal (supercritically damped)); and all $\eta_i^2 = -1$, $\lambda_i = 1$ ("standard" underdamped response). "Standard" underdamped response for a given root pair corresponds to the response of a 2nd-order loop with $\zeta = 0.707$ (or $r = 2$, since $r = 4\zeta^2$ [6]). For supercritical damping, the response of each root pair corresponds to the response of a critically damped 2nd-order loop ($\zeta^2 = 1$ or $r = 4$).

Thus, this pairwise approach to parameterizing the roots can be easily extended to loops of arbitrary order and can provide direct physical meaning and great flexibility in placing the roots in the s-plane. The next step is to determine the loop constants as a function of the new independent parameters.

F. Loop Constants as a Function of the New Independent Loop Parameters

To begin the process of parameterizing the loop constants, define the higher order CU loop constants in terms of κ_1:

$$\kappa_i = \alpha_i \kappa_1^i \qquad \text{for} \quad i = 2, \ldots, N. \tag{27}$$

No loss of generality is suffered at this step since $N-1$ new parameters α_i replace the $N-1$ κs. Note that this definition makes the αs dimensionless. As shown in Table I for each loop order, when (27) is substituted for the κs in the expression for loop bandwidth, one obtains B_L as a function of κ_1 and the αs. This equation can be used to obtain the DU loop constant $K_1 = \kappa_1 T$ in terms of B_L and the αs, as shown in the first line of equations for each loop order in Table II. Based on this expression for K_1 and (5) and (27),

TABLE III
Loop-Filter Constants for Typical Implementations in CU Approximation

Supercritically damped: $\eta_i^2 = 0$, $\lambda_i = 1$, for all roots

	Loop constants				Traditional parameters		
	K_1	K_2 $(\alpha_2 K_1^2)$	K_3 $(\alpha_3 K_1^3)$	K_4 $(\alpha_4 K_1^4)$	r $(1/\alpha_2)$	k $(\alpha_3 r^2)$	a $(\alpha_4 r^3)$
1st Order	$4\,B_L T$						
2nd Order	$\frac{16}{5} B_L T$	$\frac{1}{4} K_1^2$			4		
3rd Order	$\frac{32}{11} B_L T$	$\frac{1}{3} K_1^2$	$\frac{1}{27} K_1^3$		3	$\frac{1}{3}$	
4th Order	$\frac{256}{93} B_L T$	$\frac{3}{8} K_1^2$	$\frac{1}{16} K_1^3$	$\frac{1}{256} K_1^4$	$\frac{8}{3}$	$\frac{4}{9}$	$\frac{2}{27}$

Standard underdamped: $\eta_i^2 = -1$, $\lambda_i = 1$, for all roots

	Loop constants				Traditional parameters		
	K_1	K_2 $(\alpha_2 K_1^2)$	K_3 $(\alpha_3 K_1^3)$	K_4 $(\alpha_4 K_1^4)$	r $(1/\alpha_2)$	k $(\alpha_3 r^2)$	a $(\alpha_4 r^3)$
1st Order	$4\,B_L T$						
2nd Order	$\frac{8}{3} B_L T$	$\frac{1}{2} K_1^2$			2		
3rd Order	$\frac{60}{23} B_L T$	$\frac{4}{9} K_1^2$	$\frac{2}{27} K_1^3$		$\frac{9}{4}$	$\frac{3}{8}$	
4th Order	$\frac{64}{27} B_L T$	$\frac{1}{2} K_1^2$	$\frac{1}{8} K_1^3$	$\frac{1}{64} K_1^4$	2	$\frac{1}{2}$	$\frac{1}{8}$

the higher order loop constants can be expressed as a function of B_L and the αs, as suggested in the first line of equations for each loop order in Table II.

The variables $\alpha_2,\ldots,\alpha_N$, can now be parameterized in terms of the new independent parameters, B_L, λ_i, and η_i^2. The reference decay-rate parameter β_1 can be expressed as a function of B_L, λ_i, and η_i^2 by substituting in (17) the root expressions in (26) and solving for β_1:

$$\beta_1 = \frac{\kappa_1}{\sum_{\text{roots}} \lambda_i} \tag{28}$$

where $\kappa_1 = K_1/T$ is given in terms of the αs in Table II. Note that the sum over roots counts a λ_i parameter twice for a root pair, once for a simple root. The αs are now obtained by substituting (26), (27) and (28) in (17)–(20). Results for the αs are given for loops of order 1 to 4 in the second line of equations for each loop order in Table II. To tie in with past formulations, the αs are also given in terms of traditional parameters in the third line in Table II.

Thus, if the independent parameters of B_L, λ_i, and η_i^2 have been selected, the loop constants can be obtained as summarized in Table II by first computing the αs on the basis of λ_i and η_i^2, then computing K_1 on the basis of $B_L T$ and the αs, and finally computing the higher order Ks on the basis of the results for those quantities.

Since the transient response of a CU loop is characterized by the solution (see (14)) to the homogeneous equation, knowledge of root locations provides a basis for predicting such transient behavior and settling time. ("Transient response" in this analysis refers to the transient response predicted for the "linear-phase-extractor" model and therefore applies when tracking error is sufficiently small.) Because the η_i^2 and λ_i values, along with the loop bandwidth, completely specify the roots by location in the complex plane, loop transient response is directly selected at the outset when the new loop parameters are chosen. For example, the decay-rate parameter for each root, $\beta_1\lambda_i$, can be computed on the basis of (28) by substituting the appropriate expression for $\kappa_1 = K_1/T$ from Table II. For loops of first to fourth order, Table III presents loop constants for two likely implementations: 1) standard underdamping, where all roots have the same decay rate (all $\lambda_i = 1$) and all $\eta_i^2 = -1$, and 2) supercritical damping, where all roots have the same decay rate and are critically damped (all $\eta_i^2 = 0$). For comparison, Table III also presents the corresponding traditional parameters.

IV. LOOPS WITH DISCRETE UPDATES

A. Closed-Loop Equation

A loop with phase and phase-rate feedback and immediate updates is analyzed in detail while only results are presented for other loop implementations. For loops with phase and phase-rate feedback, (1), (2), and (4) can be combined to relate model phase to input signal phase:

$$\Delta\hat{\phi}_{n+1} + K_1\hat{\phi}_n + K_2\sum_{i=1}^{n}\hat{\phi}_i + K_3\sum_{i=1}^{n}\sum_{j=1}^{i}\hat{\phi}_j + K_4\sum_{i=1}^{n}\sum_{j=1}^{i}\sum_{k=1}^{j}\hat{\phi}_k + \cdots = K_1\phi_n + K_2\sum_{i=1}^{n}\phi_i + K_3\sum_{i=1}^{n}\sum_{j=1}^{i}\phi_j + K_4\sum_{i=1}^{n}\sum_{j=1}^{i}\sum_{k=1}^{j}\phi_k + \cdots \quad (29)$$

where the immediate-update implementation ($n_c = 0$) has been assumed and where the difference operator Δ is defined by

$$\Delta x_n \equiv x_n - x_{n-1}. \quad (30)$$

(An equation analogous to (29) is obtained for a rate-only loop by combining (1), (3), and (4).) To convert (29) to a difference equation, apply the Δ operator $(N-1)$ times:

$$\Delta^N\hat{\phi}_{n+1} + K_1\Delta^{N-1}\hat{\phi}_n + K_2\Delta^{N-2}\hat{\phi}_n + \cdots + K_N\hat{\phi}_n = K_1\Delta^{N-1}\phi_n + K_2\Delta^{N-2}\phi_n + \cdots + K_N\phi_n \quad (31)$$

where N is the loop order. In analogy with (9), solution of this difference equation gives the behavior of model phase $\hat{\phi}$ in response to signal phase ϕ, when $\phi - \hat{\phi}$ is sufficiently small.

B. Transfer Function

To find the frequency response of the loop, take the z-transform of both sides of (31) to obtain

$$z(1-z^{-1})^N\hat{\Phi}_z(z) + (1-z^{-1})^{N-1}K_1\hat{\Phi}_z(z) + \cdots + K_N\hat{\Phi}_z(z) = (1-z^{-1})^{N-1}K_1\Phi_z(z) + \cdots + K_N\Phi_z(z) \quad (32)$$

where Φ_z and $\hat{\Phi}_z$ are the z-transforms of ϕ_n and $\hat{\phi}_n$, respectively. To reach this expression, we have used the relations

$$\mathcal{Z}\{\Delta^N x_n\} = (1-z^{-1})^N\mathcal{Z}\{x_n\} \quad (33)$$

and

$$\mathcal{Z}\{x_{n+1}\} = z\mathcal{Z}\{x_n\} \quad (34)$$

where $\mathcal{Z}\{\ \}$ represents a z-transform. Since the closed-loop transfer function $H_z(z)$ is defined by

$$\hat{\Phi}_z(z) = H_z(z)\Phi_z(z) \quad (35)$$

we find that (32) yields the expression

$$H_z(z) = \frac{D(z) - (z-1)^N}{D(z)} \quad (36)$$

where

$$D(z) \equiv (z-1)^N + (z-1)^{N-1}K_1 + z(z-1)^{N-2}K_2 + z^2(z-1)^{N-3}K_3 + \cdots + z^{N-1}K_N. \quad (37)$$

The frequency response of the loop is obtained by substituting

$$z = e^{i2\pi fT} \quad (38)$$

in (36) where f is the frequency in Hertz. Plots of the transfer function for possible loop implementations are given below.

C. Loop Noise Bandwidth

In analogy with (13), the single-sided noise bandwidth for the closed DU loop is defined by

$$2B_L \equiv \int_{-1/2T}^{1/2T} |H_z(e^{i2\pi fT})|^2\,df. \quad (39)$$

By using the transformation of (38), one can rewrite this integral as a contour integral in the form

$$2B_LT = \frac{1}{2\pi i}\oint H_z(z)H_z(z^{-1})z^{-1}\,dz \quad (40)$$

where the closed path is along the unit circle. (Since the integral is along the unit circle, the conjugate z^* can be replaced by z^{-1}.) This integral can be computed on the basis of residues within the unit circle to obtain B_LT as a function of the poles of the integrand. For simple poles, the integral is given by

$$2B_LT = \sum_i \{(z-z_i)H_z(z)H_z(z^{-1})z^{-1}\}_{z\to z_i} \quad (41)$$

where the sum is over all poles $\{z_i\}$ of the integrand within the unit circle. (For cases with poles of order greater than first, the residue evaluation must be appropriately modified.)

As seen in (36), the poles of $H_z(z)$ are the roots of the polynomial $D(z)$ in the denominator of $H_z(z)$ and therefore satisfy the equation

$$D(z) = (z-1)^N + (z-1)^{N-1}K_1 + z(z-1)^{N-2}K_2 + z^2(z-1)^{N-3}K_3 + \cdots + z^{N-1}K_N = 0. \quad (42)$$

Let the N roots of this polynomial be z_i so that $D(z)$ can also be written as

$$D(z) = (z-z_1)(z-z_2)(z-z_3)\cdots(z-z_N). \quad (43)$$

Since the N roots of $D(z)$ must all lie within the unit circle if the loop is to be stable (see next subsection), all poles of $H_z(z^{-1})$ will lie outside the unit circle and will not contribute to the contour integral. (Note that there is no pole at $z = 0$ due to z^{-1}, since $\lim_{z\to 0} H_z(z^{-1}) = K_1 z$.) Thus the contour integral can be evaluated in a straightforward though algebraically tedious fashion on the basis of the N poles of $H_z(z)$ through use of (36), (41), and (43). The resulting expressions for $B_L T$ as a function of the roots z_i are lengthy and relatively uninformative, particularly for the higher-order loops, and are not presented here.

D. Solution to Homogeneous Equation

In analogy with Section IIIC, solutions to the homogeneous form of (31) (i.e., $\phi_n = 0$) provide information as to the transient behavior and stability of the DU loop, given the linear phase-extractor model. The nondegenerate solution to the homogeneous equation is given by

$$\hat{\phi}_n = \sum_{i=1}^{N} a_i z_i^n \qquad (44)$$

where the N amplitudes a_i are to be determined from initial conditions and where the N complex numbers z_i are again the roots of the polynomial $D(z)$ in (42). To see that the roots of (42) provide solutions to the homogeneous equation, substitute $\hat{\phi}_n = z^n$ in the left-hand side of (31), with the right-hand side set equal to zero, and reduce to the form of (42). Thus, the roots from the homogeneous equation are also the poles of the transfer function. (A degenerate case can again be approximated by a nondegenerate solution with infinitesimally separated roots.) In order for the loop to be stable, the loop constants K_i must be set to values for which all the roots to fall within the unit circle. With a modulus less than 1, a root leads to a root-specific transient response that decays.

E. Loop Constants as Function of Loop Roots

To obtain the relationship between the roots and the loop constants, first collect terms according to the power of z in (37) and (43). When the coefficients of like powers of z are equated, one obtains N equations relating roots and loop constants:

$$\sum_{i}^{N} z_i = \binom{N}{1} - K_1 - K_2 - \cdots - K_N \qquad (45)$$

$$\sum_{i<j}^{N} z_i z_j = \binom{N}{2} - \binom{N-1}{1} K_1 - \binom{N-2}{1} K_2 - \cdots - K_{N-1} \qquad (46)$$

$$\sum_{i<j<k}^{N} z_i z_j z_k = \binom{N}{3} - \binom{N-1}{2} K_1 - \binom{N-2}{2} K_2 - \cdots - K_{N-2} \qquad (47)$$

$$\vdots$$

$$\sum_{i}^{N} \frac{z_1 z_2 \cdots z_N}{z_i} = \binom{N}{N-1} - \binom{N-1}{N-2} K_1 - K_2 \qquad (48)$$

$$\prod_{i}^{N} z_i = 1 - K_1 \qquad (49)$$

where $\binom{n}{k}$ is the binomial coefficient. These N equations can be used to solve for each of the N loop constants K_n in terms of the N roots, z_i. Thus, if the roots are known, the loop constants can be calculated.

When the contour integral for $B_L T$ is evaluated as a function of roots for a given loop order, as described in Section IVC, it turns out that the result can be reduced to a form that contains only the functions of z_i found on the left-hand sides of (45)–(49). When this form is reached, $B_L T$ can then be expressed as a function of only the loop constants. As examples, results are presented in Table IV for loops of order one to four with phase and phase-rate feedback and zero computation delay.

F. Parameterization of Loop Roots

Parameterization of loop roots in the case of discrete updates parallels Subsection IIIE for the CU limit. Loop noise bandwidth B_L and the same root-location parameters are adopted as independent loop parameters. The roots are parameterized in the form

$$\{z_1, z_2; z_3, z_4; z_5, z_6; \ldots\} = \{e^{-\beta_1(1\pm\eta_1)T}; e^{-\beta_1\lambda_2(1\pm\eta_2)T}; e^{-\beta_1\lambda_3(1\pm\eta_3)T}; \ldots\} \qquad (50)$$

where λ_i and η_i are the $N-1$ independent parameters specified in Section IIIE, with $\lambda_1 \equiv 1$. These parameters and "normalized" loop bandwidth $B_L T$ will comprise the N independent loop parameters needed to completely specify the loop roots. As in the CU formulation, the quantity β_1 will be made a dependent variable.

G. Loop Constants as a Function of the New Independent Loop Parameters

In (50), the reference decay-rate parameter β_1, which is to be represented in the normalized, dimensionless form $\beta_1 T$ in the DU case, must be

TABLE IV
Loop Bandwidth From Loop Constants for DPLLs With Phase/Phase-Rate Feedback and No Computation Delay

1st order

$$B_LT = \frac{K_1}{2(2-K_1)}$$

2nd order

$$B_LT = \frac{2K_1^2 + 2K_2 + K_1K_2}{2K_1(4-2K_1-K_2)}$$

3rd order

$$B_LT = \frac{4K_1^2K_2 - 4K_1K_3 + 4K_2^2 + 2K_1K_2^2 + 4K_1^2K_3 + 4K_2K_3 + 3K_1K_2K_3 + K_3^2 + K_1K_3^2}{2(K_1K_2 - K_3 + K_1K_3)(8-4K_1-2K_2-K_3)}$$

4th order

$$B_LT = \frac{\left(\begin{array}{c} 8K_1^2K_2K_3 - 8K_1K_3^2 - 8K_1^3K_4 + 8K_2^2K_3 - 8K_1K_2K_4 - 8K_3K_4 + 4K_1K_2^2K_3 + 8K_1^2K_3^2 + 8K_2K_3^2 + 4K_1^2K_2K_4 + 8K_2^2K_4 - \\ 20K_1K_3K_4 - 8K_4^2 + 6K_1K_2K_3^2 + 2K_3^3 + 14K_1^2K_3K_4 + 14K_2K_3K_4 + 4K_1K_2^2K_4 - 10K_1K_4^2 + 11K_1K_2K_3K_4 + \\ 2K_1K_3^3 + 5K_3^2K_4 + 7K_1^2K_4^2 + 6K_2K_4^2 + 5K_1K_3^2K_4 + 5K_1K_2K_4^2 + 4K_3K_4^2 + 4K_1K_3K_4^2 + K_4^3 + K_1K_4^3 \end{array}\right)}{2(K_1K_2K_3 - K_3^2 - K_1^2K_4 + K_1K_3^2 + K_1K_2K_4 - 2K_3K_4 + 2K_1K_3K_4 - K_4^2 + K_1K_4^2)(16 - 8K_1 - 4K_2 - 2K_3 - K_4)}$$

determined as a function of these N independent parameters. In the CU limit, determination of β_1 in terms of B_L, λ_i and η_i^2, could be carried out in closed form, as discussed in Section IIIF. For DU loops, however, the complexity of the equations makes a closed-form solution in the general case impractical. Thus, a numerical solution has been carried out by first selecting a value for β_1T and the $N-1$ independent parameters, λ_i and η_i^2, and then computing numerical values for the N roots z_i through use of (50). The resulting z_i values can be used to compute the normalized loop bandwidth B_LT, as shown in Section IVC, and the loop constants, as shown in Section IVE. Repeating the process in this fashion on the basis of the same λ_i and η_i^2 values, one can vary the parameter β_1T numerically to obtain B_LT and the loop constants as a function of β_1T.

In general, B_LT increases as β_1T increases from zero but can go no higher than a loop-specific maximum value. Plots of B_LT versus β_1T are shown in Fig. 2 for two supercritically damped third-order loops with phase and phase-rate feedback, one with a computation delay of zero, the other with a computation delay equal to one update interval. In the zero-computation delay case, B_LT can get no higher than 9.5, which corresponds to a β_1T value of $+\infty$. In the other case, B_LT reaches a peak value of approximately 0.3 at $\beta_1T = -\ln(3/4)$.

For a given allowed value of B_LT, therefore, one can find the corresponding β_1T. Given λ_i and η_i^2 and this value of β_1T, one can compute the loop roots on the basis of (50). These loop roots can then be inserted in (45)–(49) and the N loop constants can be computed. Thus, loop constants can be determined for given λ_i and η_i^2 as a function of B_LT. Results are presented in Tables V–VIII for loops of order

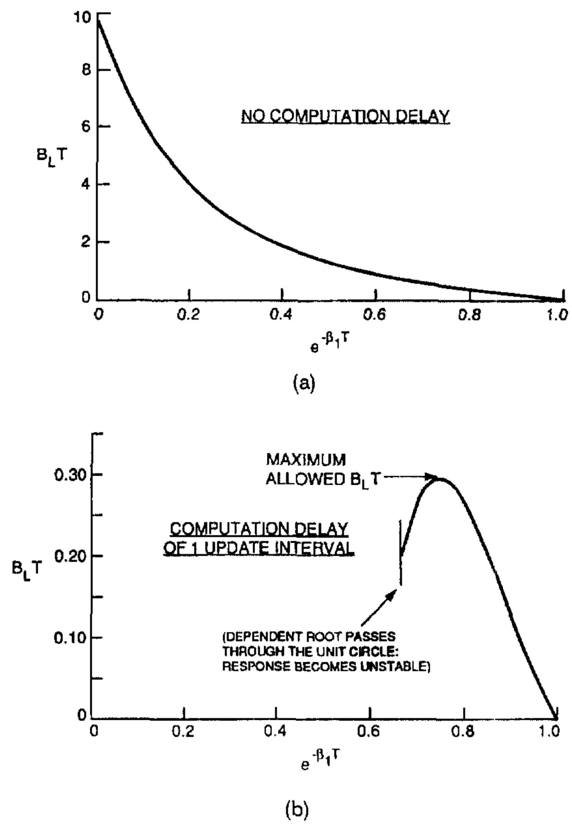

Fig. 2. Normalized loop bandwidth B_LT versus decay-rate parameter β_1T for third-order DPLL with phase/phase-rate feedback and supercritical damping.

one to four, given phase and phase-rate feedback or rate-only feedback, supercritical damping or standard underdamping, and a computation delay of zero or one update interval. For most loops in Tables V–VIII, loop constants are presented over the full range of allowed

TABLE V

Loop-Filter Constants for DPLL With Phase/Phase-Rate Feedback and Supercritically Damped Response

No computation delay

	1st order	2nd order		3rd order			4th order			
B_LT	K_1	K_1	K_2	K_1	K_2	K_3	K_1	K_2	K_3	K_4
0.001	0.003992	0.003193	2.553e-06	0.002903	2.812e-06	9.084e-10	0.002747	2.833e-06	1.299e-09	2.234e-13
0.005	0.01980	0.01582	6.309e-05	0.01438	6.941e-05	1.118e-07	0.01361	6.988e-05	1.597e-07	1.369e-10
0.01	0.03922	0.03130	0.0002488	0.02845	0.0002733	8.778e-07	0.02692	0.000275	1.251e-06	2.138e-09
0.02	0.07692	0.06125	0.0009677	0.05567	0.001060	6.765e-06	0.05269	0.001065	9.617e-06	3.265e-08
0.03	0.1132	0.08993	0.002118	0.08174	0.002312	2.220e-05	0.07738	0.002321	0.0000312	1.578e-07
0.05	0.1818	0.1438	0.005576	0.1307	0.00605	9.485e-05	0.1237	0.006059	0.0001337	1.113e-06
0.075	0.2609	0.2051	0.01176	0.1864	0.01267	0.0002936	0.1766	0.01265	0.0004113	5.055e-06
0.1	0.3333	0.2607	0.01965	0.2369	0.02101	0.0006405	0.2245	0.02094	0.0008915	1.439e-05
0.15	0.4615	0.3572	0.03931	0.3248	0.04147	0.001847	0.3080	0.04113	0.002540	5.978e-05
0.2	0.5714	0.4379	0.06264	0.3983	0.06523	0.00378	0.3780	0.06443	0.005139	0.0001570
0.25	0.6667	0.5061	0.08835	0.4606	0.09089	0.006432	0.4375	0.08942	0.008652	0.0003222
0.3	0.7500	0.5643	0.1155	0.5139	0.1175	0.00976	0.4885	0.1152	0.01300	0.0005671
0.35	0.8235	0.6142	0.1436	0.5598	0.1444	0.01371	0.5327	0.1411	0.01808	0.0008996
0.4	0.8889	0.6575	0.1720	0.5998	0.1712	0.01821	0.5712	0.1668	0.02380	0.001325
0.45	0.9474	0.6952	0.2006	0.6348	0.1977	0.02321	0.6050	0.1920	0.03006	0.001844
0.5	1.0	0.7282	0.2291	0.6657	0.2235	0.02864	0.6349	0.2166	0.03679	0.002459
0.6		0.7828	0.2851	0.7173	0.2732	0.04059	0.6852	0.2634	0.05133	0.003967
0.7		0.8257	0.3394	0.7585	0.3196	0.05371	0.7258	0.3069	0.06692	0.005832
0.8		0.8599	0.3915	0.7920	0.3629	0.06769	0.7590	0.3471	0.08318	0.008026
0.9		0.8874	0.4414	0.8196	0.4030	0.0823	0.7865	0.3842	0.09983	0.01052
1.		0.9096	0.4890	0.8426	0.4402	0.09735	0.8097	0.4184	0.1167	0.01328
1.2		0.9425	0.5779	0.8782	0.5065	0.1283	0.8462	0.4790	0.1503	0.01951
1.4		0.9645	0.6588	0.9042	0.5636	0.1597	0.8734	0.5308	0.1832	0.02650
1.6		0.9793	0.7328	0.9237	0.6130	0.191	0.8942	0.5754	0.2151	0.03410
1.8		0.9889	0.8006	0.9386	0.6559	0.222	0.9106	0.6141	0.2457	0.04216
2.		0.9950	0.8631	0.9502	0.6935	0.2525	0.9236	0.6479	0.2749	0.05058
2.5		1.0	1.0	0.9697	0.7689	0.3258	0.9466	0.7158	0.3419	0.07270
3.				0.9811	0.8248	0.3947	0.9612	0.7667	0.4011	0.09566
3.5				0.9881	0.8672	0.4591	0.9710	0.8058	0.4533	0.1189
4.				0.9925	0.8997	0.5194	0.9778	0.8366	0.4996	0.1422
4.5				0.9953	0.9248	0.576	0.9827	0.8613	0.5409	0.1652
5.				0.9971	0.9444	0.6291	0.9864	0.8814	0.5779	0.1879

Computation delay = 1 update interval

	1st order	2nd order		3rd order			4th order			
B_LT	K_1	K_1	K_2	K_1	K_2	K_3	K_1	K_2	K_3	K_4
0.001	0.003976	0.003181	2.538e-06	0.002892	2.796e-06	9.015e-10	0.002737	2.816e-06	1.288e-09	2.211e-13
0.005	0.01942	0.01554	6.135e-05	0.01414	6.749e-05	1.077e-07	0.01339	6.795e-05	1.537e-07	1.306e-10
0.01	0.03779	0.03023	0.0002357	0.02752	0.0002588	8.165e-07	0.02606	0.0002604	1.163e-06	1.952e-09
0.02	0.07177	0.05734	0.0008737	0.05224	0.0009552	5.897e-06	0.04951	0.0009599	8.364e-06	2.745e-08
0.03	0.1027	0.08185	0.001832	0.07461	0.001993	1.809e-05	0.07076	0.002000	2.554e-05	1.231e-07
0.05	0.1571	0.1245	0.004476	0.1136	0.00482	7.032e-05	0.1078	0.004819	9.836e-05	7.611e-07
0.075	0.2148	0.1685	0.008742	0.1538	0.009282	0.0001955	0.1462	0.009237	0.0002700	3.007e-06
0.1		0.2046	0.01371	0.1869	0.01433	0.0003895	0.1778	0.01420	0.0005309	7.609e-06
0.15		0.2594	0.02487	0.2377	0.02515	0.0009724	0.2265	0.02467	0.001289	2.610e-05
0.2				0.2740	0.03595	0.001784	0.2618	0.03495	0.002295	5.906e-05
0.25				0.3000	0.04617	0.002793	0.2878	0.04452	0.003473	0.0001076
0.3							0.3071	0.05316	0.004765	0.0001722
0.35							0.3211	0.06074	0.006129	0.0002543

values of B_LT, to the extent allowed by computational quantization of B_LT. In Tables V and VI, however, B_LT has been extended only up to 5.0 for third- and fourth-order loops, even though much larger values are feasible (e.g., up to $B_LT = 9.5$ for third order and 24.5 for fourth order in Table V).

Once the loop roots are known, the transfer function can be computed on the basis of (36) and (43). Results are plotted in Figs. 3 and 4 for two loop configurations.

H. Example of Straying Roots at Large B_LT Values in the CU Approximation

As B_LT increases in the CU approximation for loop constants (e.g., Table III), the paths of the

TABLE VI

Loop-Filter Constants for DPLL With Phase/Phase-Rate Feedback and Standard-Underdamped Response

No computation delay

	1st order	2nd order		3rd order			4th order			
B_LT	K_1	K_1	K_2	K_1	K_2	K_3	K_1	K_2	K_3	K_4
0.001	0.003992	0.002661	3.545e-06	0.002603	3.016e-06	1.311e-09	0.002367	2.804e-06	1.661e-09	4.923e-13
0.005	0.01980	0.01319	8.752e-05	0.01290	7.437e-05	1.611e-07	0.01173	6.907e-05	2.037e-07	3.008e-10
0.01	0.03922	0.02609	0.0003448	0.02552	0.0002926	1.264e-06	0.02321	0.0002716	1.594e-06	4.693e-09
0.02	0.07692	0.05106	0.001338	0.04996	0.001133	9.720e-06	0.04545	0.001051	1.222e-05	7.147e-08
0.03	0.1132	0.07499	0.002922	0.07338	0.002469	3.156e-05	0.06679	0.002289	3.955e-05	3.447e-07
0.05	0.1818	0.1199	0.007658	0.1174	0.006445	0.0001355	0.1070	0.005966	0.0001687	2.420e-06
0.075	0.2609	0.1713	0.01607	0.1677	0.01345	0.0004177	0.1530	0.01243	0.0005159	1.093e-05
0.1	0.3333	0.2179	0.02670	0.2133	0.02226	0.0009073	0.1949	0.02054	0.001112	3.096e-05
0.15	0.4615	0.2991	0.05288	0.2929	0.04370	0.002595	0.2683	0.04022	0.003136	0.0001273
0.2	0.5714	0.3675	0.08345	0.3599	0.06842	0.005269	0.3305	0.06282	0.006285	0.0003313
0.25	0.6667	0.4258	0.1166	0.4171	0.09491	0.008901	0.3838	0.08697	0.01049	0.0006738
0.3	0.7500	0.4760	0.1511	0.4662	0.1222	0.01341	0.4299	0.1118	0.01562	0.001176
0.35	0.8235	0.5196	0.1862	0.5089	0.1496	0.01871	0.4701	0.1367	0.02156	0.001851
0.4	0.8889	0.5577	0.2213	0.5463	0.1768	0.02470	0.5056	0.1613	0.02817	0.002705
0.45	0.9474	0.5914	0.2561	0.5793	0.2034	0.03129	0.5370	0.1853	0.03534	0.003738
0.5	1.0	0.6214	0.2902	0.6085	0.2294	0.03838	0.5650	0.2087	0.04296	0.00495
0.6		0.6721	0.3560	0.6580	0.2788	0.05381	0.6128	0.2532	0.05920	0.007884
0.7		0.7134	0.4181	0.6983	0.3247	0.07047	0.6521	0.2944	0.07632	0.01145
0.8		0.7475	0.4763	0.7315	0.3671	0.08796	0.6848	0.3325	0.0939	0.01558
0.9		0.7762	0.5306	0.7593	0.4061	0.1060	0.7124	0.3675	0.1116	0.02021
1.		0.8007	0.5813	0.7829	0.4421	0.1243	0.7360	0.3997	0.1293	0.02527
1.2		0.8399	0.6727	0.8205	0.5059	0.1612	0.7743	0.4569	0.1639	0.03644
1.4		0.8699	0.7524	0.8490	0.5603	0.1979	0.8038	0.5058	0.1971	0.04869
1.6		0.8936	0.8223	0.8713	0.6072	0.2337	0.8272	0.5479	0.2286	0.06171
1.8		0.9127	0.8839	0.8891	0.6478	0.2684	0.8461	0.5846	0.2582	0.07525
2.		0.9286	0.9385	0.9035	0.6832	0.3019	0.8618	0.6167	0.2860	0.08913
2.5		0.9587	1.051	0.9297	0.7544	0.3802	0.8910	0.6817	0.3482	0.1245
3.		0.9827	1.134	0.9470	0.8076	0.4510	0.9111	0.7310	0.4013	0.1599
3.5				0.9592	0.8485	0.5151	0.9256	0.7695	0.4470	0.1945
4.				0.9680	0.8805	0.5735	0.9366	0.8004	0.4867	0.2281
4.5				0.9747	0.9061	0.6269	0.9452	0.8256	0.5213	0.2605
5.				0.9798	0.9267	0.6760	0.9520	0.8465	0.5519	0.2917

Computation delay = 1 update interval

	1st order	2nd order		3rd order			4th order			
B_LT	K_1	K_1	K_2	K_1	K_2	K_3	K_1	K_2	K_3	K_4
0.001	0.003976	0.00265	3.516e-06	0.002594	2.997e-06	1.299e-09	0.002358	2.784e-06	1.645e-09	4.859e-13
0.005	0.01942	0.01299	8.487e-05	0.01270	7.229e-05	1.549e-07	0.01155	6.717e-05	1.959e-07	2.862e-10
0.01	0.03779	0.02533	0.0003248	0.02474	0.0002769	1.170e-06	0.02252	0.0002573	1.478e-06	4.259e-09
0.02	0.07177	0.04827	0.001194	0.04709	0.001019	8.391e-06	0.04293	0.0009475	1.058e-05	5.935e-08
0.03	0.1027	0.06919	0.002479	0.06739	0.002121	2.554e-05	0.06154	0.001971	3.210d-05	2.636e-07
0.05	0.1571	0.1060	0.005937	0.1030	0.005094	9.769e-05	0.09425	0.004732	0.0001220	1.594e-06
0.075	0.2148	0.1448	0.01130	0.1401	0.009726	0.0002657	0.1285	0.009020	0.0003292	6.120e-06
0.1		0.1775	0.01725	0.1709	0.01489	0.000518	0.1571	0.01379	0.0006363	1.503e-05
0.15		0.2300	0.02963	0.2193	0.02572	0.001236	0.2021	0.02372	0.001494	4.860d-05
0.2		0.2713	0.04155	0.2554	0.03630	0.002167	0.2360	0.03335	0.002579	0.0001037
0.25		0.3071	0.05222	0.2833	0.04616	0.003232	0.2622	0.04227	0.003795	0.0001781
0.3				0.3055	0.05518	0.004373	0.2832	0.05037	0.005075	0.0002685
0.35				0.3242	0.06338	0.005530	0.3002	0.05766	0.006374	0.0003710
0.4							0.3143	0.06420	0.007664	0.0004821
0.45							0.3263	0.07006	0.008927	0.0005986
0.5							0.3366	0.07534	0.01016	0.0007167

loop roots can stray from their original intended courses and true loop noise bandwidth can exceed the B_L "parameter" value used to compute the loop constants. This deviation is illustrated in both the sT-plane and z-plane in Fig. 5 for a second-order DU loop with phase and phase-rate feedback, standard underdamping, and no computation delay. The thick straight lines, which are based on the exact DU solution presented in preceding subsections, show the paths taken by standard-underdamped roots as (true) B_LT increases from 0 to 1.4. The thin curved lines, which are produced by the CU approximation, follow the DU lines until about $B_LT = 0.1$ and then curve toward the real axis. (Note that the CU-approximation

TABLE VII
Loop-Filter Constant for DPLL With Rate-Only Feedback and Supercritically Damped Response

No computation delay

	1st order	2nd order		3rd order			4th order			
B_LT	K_1	K_1	K_2	K_1	K_2	K_3	K_1	K_2	K_3	K_4
0.005	0.01976	0.01575	6.275e-05	0.01431	6.890e-05	1.109e-07	0.01353	6.927e-05	1.579e-07	1.351e-10
0.010	0.03918	0.03109	2.475e-04	0.02823	2.707e-04	8.696e-07	0.02669	2.718e-04	1.235e-06	2.109e-09
0.015	0.05822	0.04602	5.486e-04	0.04175	5.978e-04	2.874e-06	0.03947	5.992e-04	4.069e-06	1.039e-08
0.020	0.07689	0.06054	9.607e-04	0.05488	0.001043	6.668e-06	0.05187	0.001044	9.411e-06	3.194e-08
0.025	0.09520	0.07466	0.001479	0.06763	0.001599	1.275e-05	0.06392	0.001598	1.794e-05	7.585e-08
0.030	0.1132	0.08840	0.002097	0.08003	0.002259	2.157e-05	0.07562	0.002254	3.025e-05	1.530e-07
0.035	0.1308	0.1018	0.002812	0.09207	0.003018	3.354e-05	0.08699	0.003006	4.688e-05	2.758e-07
0.040	0.1481	0.1148	0.003619	0.1038	0.003868	4.902e-05	0.09803	0.003848	6.831e-05	4.579e-07
0.045	0.1651	0.1274	0.004513	0.1152	0.004806	6.836e-05	0.1088	0.004774	9.496e-05	7.139e-07
0.050	0.1818	0.1397	0.005491	0.1262	0.005826	9.186e-05	0.1192	0.005778	1.272e-04	1.059e-06
0.060	0.2143	0.1634	0.007683	0.1474	0.008089	1.524e-04	0.1392	0.008000	2.097e-04	2.083e-06
0.070	0.2456	0.1858	0.01016	0.1675	0.01062	2.325e-04	0.1581	0.01047	3.179e-04	3.663e-06
0.080	0.2758	0.2070	0.01291	0.1865	0.01339	3.336e-04	0.1760	0.01317	4.533e-04	5.936e-06
0.090	0.3051	0.2271	0.01590	0.2044	0.01636	4.569e-04	0.1929	0.01605	6.170e-04	9.039e-06
0.100	0.3333	0.2461	0.01911	0.2214	0.01952	6.033e-04	0.2089	0.01909	8.095e-04	1.311e-05
0.150		0.3268	0.03784	0.2938	0.03718	0.001697	0.2773	0.03591	0.002207	5.227e-05
0.200		0.3864	0.05992	0.3489	0.05646	0.003398	0.3298	0.05393	0.004280	1.323e-04
0.250				0.3904	0.07588	0.005680	0.3703	0.07179	0.006918	2.628e-04
0.300				0.4208	0.09441	0.008524	0.4014	0.08870	0.009991	4.495e-04
0.350							0.4250	0.1041	0.01337	6.970e-04
0.400							0.4422	0.1177	0.01696	0.001013

Computation delay = 1 update interval

	1st order	2nd order		3rd order			4th order			
B_LT	K_1	K_1	K_2	K_1	K_2	K_3	K_1	K_2	K_3	K_4
0.005	0.01939	0.01547	6.103e-05	0.01406	6.698e-05	1.067e-07	0.01331	6.738e-05	1.521e-07	1.290e-10
0.010	0.03773	0.03003	2.344e-04	0.02729	2.562e-04	8.084e-07	0.02583	2.573e-04	1.147e-06	1.924e-09
0.015	0.05510	0.04372	5.064e-04	0.03974	5.515e-04	2.582e-06	0.03762	5.530e-04	3.651e-06	9.080e-09
0.020	0.07157	0.05663	8.656e-04	0.05147	9.388e-04	5.800e-06	0.04873	9.399e-04	8.171e-06	2.679e-08
0.025	0.08724	0.06880	0.001302	0.06253	0.001406	1.075e-05	0.05922	0.001405	1.509e-05	6.118e-08
0.030	0.1022	0.08031	0.001806	0.07299	0.001942	1.765e-05	0.06912	0.001939	2.468e-05	1.188e-07
0.035	0.1164	0.09120	0.002371	0.08287	0.002539	2.666e-05	0.07850	0.002530	3.715e-05	2.066e-07
0.040	0.1300	0.1015	0.002990	0.09224	0.003188	3.792e-05	0.08738	0.003172	5.262e-05	3.311e-07
0.045	0.1431	0.1113	0.003657	0.1011	0.003882	5.149e-05	0.09580	0.003857	7.119e-05	4.991e-07
0.050	0.1556	0.1205	0.004367	0.1095	0.004614	6.745e-05	0.1038	0.004578	9.289e-05	7.169e-07
0.060	0.1790	0.1377	0.005898	0.1251	0.006175	1.066e-04	0.1186	0.006108	1.457e-04	1.326e-06
0.070		0.1531	0.007553	0.1392	0.007834	1.555e-04	0.1320	0.007725	2.108e-04	2.200e-06
0.080		0.1671	0.009307	0.1520	0.009561	2.139e-04	0.1442	0.009400	2.877e-04	3.378e-06
0.090		0.1797	0.01114	0.1636	0.01133	2.817e-04	0.1553	0.01111	3.757e-04	4.890e-06
0.100		0.1911	0.01305	0.1741	0.01313	3.585e-04	0.1654	0.01284	4.741e-04	6.761e-06
0.150				0.2142	0.02208	8.655e-04	0.2044	0.02130	0.001094	2.205e-05
0.200							0.2296	0.02889	0.001860	4.804e-05

curves are marked by both true B_LT values and parameter B_LT values. As explained in [3], true B_LT is the actual normalized noise bandwidth for a DU loop, while parameter B_LT is the value used to compute the loop constants in Table III.) Where the curves separate, the CU approximation starts to diverge from standard underdamping, and loop damping changes. The inset plot illustrates this divergence in terms of the corresponding damping parameter η^2. For the CU approximation, η^2 starts at the intended underdamped value of -1 at $B_LT = 0$, increases to a critically damped value of 0 at true $B_LT = 0.8$, and then approaches $+1$ at true $B_LT \approx 1.2$. Thus, loop response at high B_LT values does not match the original intended response. In contrast, as indicated by the thick lines, the DU exact solution maintains standard underdamping (i.e., $\eta_i^2 = -1$) for allowed values of B_LT.

V. TRANSIENT-FREE ACQUISITION WITH DPLLS

If the signal phase and its time derivatives are accurately known at start-up, it is possible to initialize the loop sums and loop phase so that the loop starts

TABLE VIII
Loop-Filter Constants for DPLL With Rate-Only Feedback and Standard-Underdamped Response

No computation delay

	1st order	2nd order		3rd order			4th order			
B_LT	K_1	K_1	K_2	K_1	K_2	K_3	K_1	K_2	K_3	K_4
0.005	0.01976	0.01312	8.661e-05	0.01283	7.365e-05	1.590e-07	0.01165	6.831e-05	2.006e-07	2.951e-10
0.010	0.03918	0.02589	3.396e-04	0.02530	2.886e-04	1.242e-06	0.02299	2.673e-04	1.561e-06	4.572e-09
0.015	0.05822	0.03831	7.482e-04	0.03742	6.356e-04	4.084e-06	0.03399	5.879e-04	5.117e-06	2.237e-08
0.020	0.07689	0.05039	0.001302	0.04918	0.001106	9.429e-06	0.04468	0.001021	1.177e-05	6.827e-08
0.025	0.09520	0.06214	0.001992	0.06062	0.001691	1.794e-05	0.05506	0.001560	2.233e-05	1.610e-07
0.030	0.1132	0.07358	0.002810	0.07172	0.002383	3.020e-05	0.06516	0.002195	3.747e-05	3.225e-07
0.035	0.1308	0.08472	0.003746	0.08252	0.003175	4.674e-05	0.07496	0.002921	5.780e-05	5.774e-07
0.040	0.1481	0.09556	0.004793	0.09302	0.004060	6.800e-05	0.08450	0.003731	8.382e-05	9.522e-07
0.045	0.1651	0.1061	0.005944	0.1032	0.005032	9.438e-05	0.09377	0.004619	1.160e-04	1.474e-06
0.050	0.1818	0.1164	0.007191	0.1132	0.006085	1.262e-04	0.1028	0.005578	1.546e-04	2.173e-06
0.060	0.2143	0.1362	0.009950	0.1322	0.008410	2.075e-04	0.1201	0.007691	2.526e-04	4.218e-06
0.070	0.2456	0.1551	0.01302	0.1503	0.01099	3.137e-04	0.1365	0.01003	3.796e-04	7.321e-06
0.080	0.2758	0.1731	0.01637	0.1674	0.01380	4.460e-04	0.1521	0.01256	5.367e-04	1.171e-05
0.090	0.3051	0.1902	0.01995	0.1837	0.01679	6.055e-04	0.1669	0.01526	7.244e-04	1.761e-05
0.100	0.3333	0.2066	0.02372	0.1991	0.01995	7.924e-04	0.1809	0.01809	9.428e-04	2.523e-05
0.150		0.2788	0.04471	0.2657	0.03734	0.002135	0.2417	0.03356	0.002476	9.483e-05
0.200		0.3387	0.06740	0.3183	0.05595	0.004103	0.2899	0.04990	0.004650	2.270e-04
0.250		0.3912	0.09013	0.3606	0.07452	0.006583	0.3288	0.06604	0.007307	4.273e-04
0.300		0.4464	0.1108	0.3951	0.09238	0.009451	0.3606	0.08142	0.01030	6.937e-04
0.350				0.4241	0.1093	0.01259	0.3870	0.09580	0.01351	0.001020
0.400				0.4499	0.1253	0.01584	0.4093	0.1091	0.01683	0.001397
0.450							0.4282	0.1213	0.02021	0.001815
0.500							0.4446	0.1325	0.02359	0.002264
0.600							0.4726	0.1523	0.03034	0.003197

Computation delay = 1 update interval

	1st order	2nd order		3rd order			4th order			
B_LT	K_1	K_1	K_2	K_1	K_2	K_3	K_1	K_2	K_3	K_4
0.005	0.01939	0.01292	8.403e-05	0.01263	7.161e-05	1.529e-07	0.01148	6.644e-05	1.930e-07	2.808e-10
0.010	0.03773	0.02514	3.200e-04	0.02453	2.731e-04	1.150e-06	0.02232	2.533e-04	1.448e-06	4.150e-09
0.015	0.05510	0.03669	6.856e-04	0.03575	5.859e-04	3.646e-06	0.03255	5.431e-04	4.581e-06	1.940e-08
0.020	0.07157	0.04763	0.001162	0.04635	9.942e-04	8.132e-06	0.04222	9.210e-04	1.019e-05	5.668e-08
0.025	0.08724	0.05801	0.001732	0.05637	0.001484	1.496e-05	0.05137	0.001374	1.870e-05	1.281e-07
0.030	0.1022	0.06787	0.002383	0.06587	0.002045	2.440e-05	0.06005	0.001891	3.041e-05	2.465e-07
0.035	0.1164	0.07725	0.003101	0.07487	0.002665	3.660e-05	0.06829	0.002463	4.550e-05	4.243e-07
0.040	0.1300	0.08619	0.003878	0.08342	0.003336	5.169e-05	0.07612	0.003081	6.409e-05	6.735e-07
0.045	0.1431	0.09472	0.004703	0.09154	0.004051	6.970e-05	0.08358	0.003738	8.621e-05	1.006e-06
0.050	0.1556	0.1029	0.005569	0.09928	0.004802	9.066e-05	0.09068	0.004428	1.119e-04	1.431e-06
0.060	0.1790	0.1182	0.007397	0.1137	0.006391	1.413e-04	0.1039	0.005886	1.735e-04	2.595e-06
0.070		0.1322	0.009316	0.1268	0.008066	2.032e-04	0.1160	0.007418	2.483e-04	4.226e-06
0.080		0.1452	0.01129	0.1388	0.009795	2.757e-04	0.1271	0.008996	3.354e-04	6.368e-06
0.090		0.1573	0.01330	0.1498	0.01156	3.581e-04	0.1372	0.01060	4.336e-04	9.049e-06
0.100		0.1685	0.01531	0.1599	0.01333	4.494e-04	0.1466	0.01221	5.418e-04	1.228e-05
0.150		0.2164	0.02502	0.2003	0.02203	0.001012	0.1841	0.02007	0.001195	3.663e-05
0.200				0.2290	0.02994	0.001683	0.2108	0.02716	0.001958	7.290e-05
0.250				0.2532	0.03710	0.002344	0.2307	0.03332	0.002754	1.177e-04
0.300							0.2461	0.03863	0.003547	1.677e-04
0.350							0.2588	0.04325	0.004319	2.185e-04

tracking in-lock, with no transients. Such information on signal phase could be supplied, for example, on the basis of FFT analysis or *a priori* trajectory estimates. This section presents a technique for transient-free initialization of a loop with phase and phase-rate feedback. A similar derivation can be carried out for a loop with rate-only feedback.

Suppose that the input signal can be represented by the appropriate polynomial so that steady-state tracking can develop. In steady-state tracking, residual phase becomes a constant ($\tilde{\phi}_n = \tilde{\phi}_{ss}$ = steady-state phase error) so that (4) becomes

$$\Delta\phi_{n+1} = K_1\tilde{\phi}_{ss} + K_2\sum_{i=1}^{n-n_c}\tilde{\phi}_i + K_3\sum_{i=1}^{n-n_c}\sum_{j=1}^{i}\tilde{\phi}_j + K_4\sum_{i=1}^{n-n_c}\sum_{j=1}^{i}\sum_{k=1}^{j}\tilde{\phi}_k \quad (51)$$

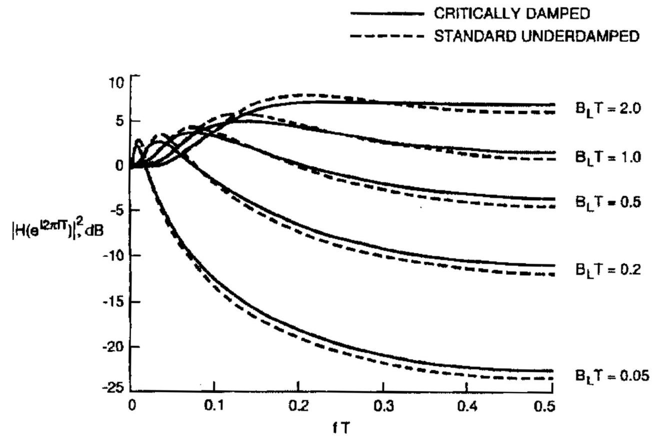

Fig. 3. Power transfer functions for third-order DPLLs with phase/phase-rate feedback and no computation delay.

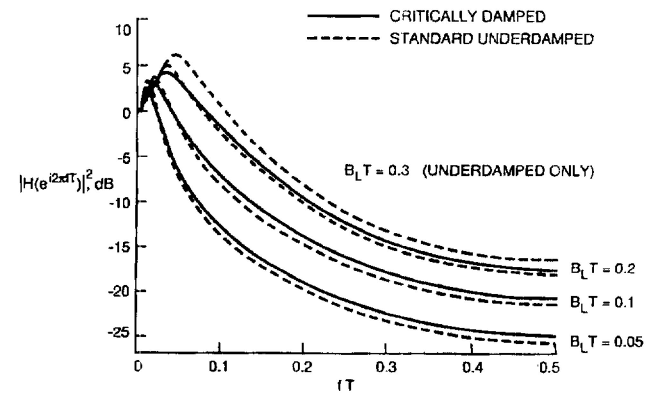

Fig. 4. Power transfer functions for third-order DPLLs with phase/phase-rate feedback and computation delay of one update interval.

for up to a fourth-order loop. Note that estimated model phase rate, $\hat{\dot{\phi}}_{n+1}T$ has been replaced by differenced *input* phase. This substitution is based on (2) and the fact that model phase tracks input phase exactly in steady-state tracking, except for noise and a constant phase offset. Higher order differences of (51) become

$$\Delta^2\phi_{n+1} = K_2\tilde{\phi}_{ss} + K_3\sum_{i=1}^{n-n_c}\tilde{\phi}_i + K_4\sum_{i=1}^{n-n_c}\sum_{j=1}^{i}\tilde{\phi}_j \qquad (52)$$

$$\Delta^3\phi_{n+1} = K_3\tilde{\phi}_{ss} + K_4\sum_{i=1}^{n-n_c}\tilde{\phi}_i \qquad (53)$$

$$\Delta^4\phi_{n+1} = K_4\tilde{\phi}_{ss} \qquad (54)$$

Using a Taylor expansion whose origin is the center of the nth interval ($t = t_n$):

$$\phi(t) = \phi_n + (t-t_n)\phi_n^{(1)} + \frac{(t-t_n)^2}{2}\phi_n^{(2)} + \frac{(t-t_n)^3}{6}\phi_n^{(3)} + \frac{(t-t_n)^4}{24}\phi_n^{(4)} + \cdots \qquad (55)$$

one can show these same phase differences are related to input phase derivatives by

$$\Delta\phi_{n+1} = T\phi_n^{(1)} + \frac{T^2}{2}\phi_n^{(2)} + \frac{T^3}{6}\phi_n^{(3)} + \frac{T^4}{24}\phi_n^{(4)} + \cdots \qquad (56)$$

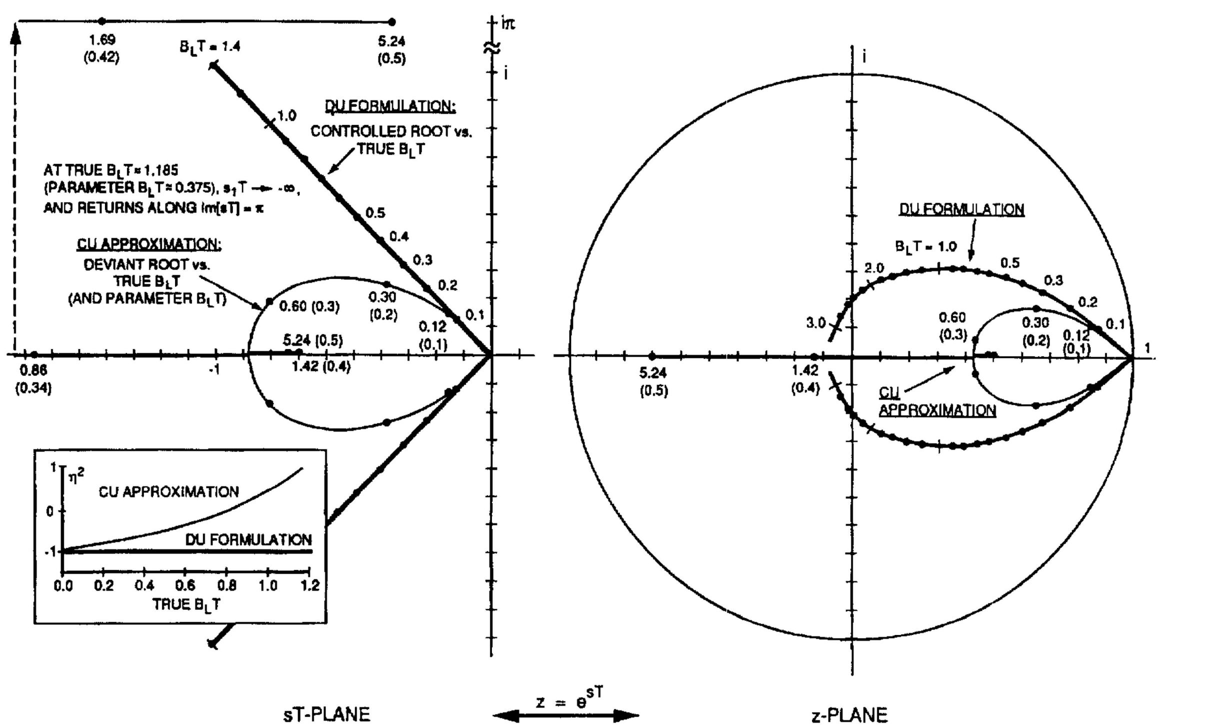

Fig. 5. Root-locus plots for second-order DPLL (phase/phase-rate feedback, no computation delay, and standard underdamping) as function of B_LT.

TABLE IX
Transient-Free Initialization of DPLLs With Phase/Phase Rate Feedback

1st Order

$$\tilde{\phi}_{ss} = \frac{T}{K_1}\phi_n^{(1)}$$

2nd Order

$$\tilde{\phi}_{ss} = \frac{T^2}{K_2}\phi_n^{(2)}$$

$$\sum_{i=1}^{n-n_c}\tilde{\phi}_i = \frac{1}{K_2}(T\phi_n^{(1)} + \frac{T^2}{2}\phi_n^{(2)} - K_1\tilde{\phi}_{ss})$$

3rd Order

$$\tilde{\phi}_{ss} = \frac{T^3}{K_3}\phi_n^{(3)}$$

$$\sum_{i=1}^{n-n_c}\tilde{\phi}_i = \frac{1}{K_3}(T^2\phi_n^{(2)} - K_2\tilde{\phi}_{ss})$$

$$\sum_{i=1}^{n-n_c}\sum_{j=1}^{i}\tilde{\phi}_j = \frac{1}{K_3}(T\phi_n^{(1)} + \frac{T^2}{2}\phi_n^{(2)} + \frac{T^3}{6}\phi_n^{(3)} - K_1\tilde{\phi}_{ss} - K_2\sum_{i=1}^{n-n_c}\tilde{\phi}_i)$$

4th Order

$$\tilde{\phi}_{ss} = \frac{T^4}{K_4}\phi_n^{(4)}$$

$$\sum_{i=1}^{n-n_c}\tilde{\phi}_i = \frac{1}{K_4}(T^3\phi_n^{(3)} - \frac{T^4}{2}\phi_n^{(4)} - K_3\tilde{\phi}_{ss})$$

$$\sum_{i=1}^{n-n_c}\sum_{j=1}^{i}\tilde{\phi}_j = \frac{1}{K_4}(T^2\phi_n^{(2)} + \frac{T^4}{12}\phi_n^{(4)} - K_2\tilde{\phi}_{ss} - K_3\sum_{i=1}^{n-n_c}\tilde{\phi}_i)$$

$$\sum_{i=1}^{n-n_c}\sum_{j=1}^{i}\sum_{k=1}^{j}\tilde{\phi}_k = \frac{1}{K_4}(T\phi_n^{(1)} + \frac{T^2}{2}\phi_n^{(2)} + \frac{T^3}{6}\phi_n^{(3)} + \frac{T^4}{24}\phi_n^{(4)} - K_1\tilde{\phi}_{ss} - K_2\sum_{i=1}^{n-n_c}\tilde{\phi}_i - K_3\sum_{i=1}^{n-n_c}\sum_{j=1}^{i}\tilde{\phi}_j)$$

For all orders:

$$\hat{\phi}_n = \begin{cases} \phi_n - \tilde{\phi}_{ss}, & \textbf{arc tangent phase extractor} \\ \phi_n - arcsin(2\pi\,\tilde{\phi}_{ss})/2\pi, & \textbf{sine phase extractor} \end{cases}$$

$$\Delta^2\phi_{n+1} = T^2\phi_n^{(2)} + \frac{T^4}{12}\phi_n^{(4)} + \cdots \quad (57)$$

$$\Delta^3\phi_{n+1} = T^3\phi_n^{(3)} - \frac{T^4}{2}\phi_n^{(4)} + \cdots \quad (58)$$

$$\Delta^4\phi_{n+1} = T^4\phi_n^{(4)} + \cdots. \quad (59)$$

By respectively equating (56)–(59) with (51)–(54), one obtains a set of equations whose number is equal to the number of unknowns, where the unknowns are $\tilde{\phi}_{ss}$ and the loop sums. Thus, these unknowns can be expressed in terms of the derivatives of input phase for a given a set of loop constants. Results for loops of order one to four are presented in Table IX.

To complete initialization of the loop, an estimate of starting model phase must be computed. To be exact, this estimate must account for steady-state phase error. For an arctangent phase extractor, tracking error is equal to residual phase (neglecting system-noise error and possible cycle ambiguities). For a sine phase extractor, however, steady-state tracking error becomes

$$(\phi_n - \hat{\phi}_n)_{ss} = \arcsin(2\pi\tilde{\phi}_{ss})/2\pi \quad (60)$$

where all phase values are measured in cycles. Thus, model phase for the n^{th} interval, after accounting for tracking error, is given by

$$\hat{\phi}_n = \begin{cases} \phi_n - \tilde{\phi}_{ss} & \text{for an arctan extractor} \\ \phi_n - \arcsin(2\pi\tilde{\phi}_{ss})/2\pi & \text{for a sine extractor} \end{cases}. \quad (61)$$

Based on *a priori* values for signal phase and its derivatives for "phantom" interval n, one can therefore calculate at the "completion" of that phantom interval the values for loop sums and model phase that would have been present under steady-state tracking. These quantities can be used to initialize the $(n+1)$th interval as though the nth interval had been processed. In predicting the $(n+1)$th phase rate by means of (4), however, the residual phase for the phantom nth interval is set equal to $\tilde{\phi}_{ss}$. Loop updates are carried out in the normal fashion for subsequent intervals. In this way, the loop can be initialized for the $(n+1)$th interval as though steady-state lock had been established and no transients will be observed.

The preceding derivation assumed a signal with appropriate polynomial phase so that a steady-state phase error would develop. Under less ideal dynamics, the above initialization process will not eliminate transients, but can greatly assist direct acquisition with higher order loops. Similarly, if the derivatives of signal phase are known, but phase is not, the loop sums can be initialized as prescribed, with initial loop phase arbitrarily set to zero. Again, loop acquisition will be greatly enhanced.

VI. TWO MEASURES OF LOOP PERFORMANCE

A. Mean Time to First Cycle Slip

Simulations have been carried out to determine mean time to first cycle slip, $\langle T_{1st} \rangle$, for loops with phase and phase-rate feedback, no computation delay, supercritical damping, and a sine phase extractor with perfect amplitude normalization. The tracking-error criterion for detecting a cycle slip was $|\phi - \hat{\phi}| > 0.75$ cycles. After each slip, the loop was reinitialized, as described in Section V, with perfect *a priori* so that it would start off in steady-state lock with no transients. A Gaussian random-number generator simulated noise for the counter-rotation sums.

Loops of 1st to 4th order have been simulated on the basis of the loop constants in Table V. Assumed values of $B_L T$ ranged between 0.02 and 2.0 and loop SNR between 0 and 10 dB. Example results are shown in Fig. 6(a), where B_L multiplied by mean time to first cycle slip is plotted versus loop SNR for $B_L T = 0.02$ and 0.5. We define loop SNR, SNR_L, according to the definition that leads to a tracking error variance, $\langle (\phi - \hat{\phi})^2 \rangle$ in radians, equal to $1/\mathrm{SNR}_L$ for high SNR values. In terms of cycles slips, loop performance deteriorates somewhat as loop order increases, given a fixed-loop SNR. For a given loop order and loop SNR, however, cycle-slip performance improves as $B_L T$ increases, as shown in more detail in Fig. 6(b) where $B_L \langle T_{1st} \rangle$ is plotted versus $B_L T$ for 1st- to 4th-order loops, given a loop SNR of 10 dB. With a 3rd-order loop, for example, Fig. 6(b) indicates that $B_L \langle T_{1st} \rangle$ improves by two orders of magnitude when $B_L T$ is increased from 0.02 to 0.5.

As a test of the simulation software, the cycle-slip criteria have been changed to Viterbi's criteria in his exact closed-form solution [7] for a first-order loop in the CU limit, and the simulations rerun. To within a statistical error of about 10%, our first-order loop results for $B_L \langle T_{1st} \rangle$ at low $B_L T$ values agree with Viterbi's theoretical predictions up to the maximum loop SNR tested, $\mathrm{SNR}_L = 4$ dB.

B. Steady-State Phase Error

Loop performance at large values of $B_L T$ has also been assessed in terms of the steady-state phase error (static phase error). In the CU limit, steady-state phase error ($\tilde{\phi}_{ss}$) is proportional to B_L^{-N} for an Nth-order

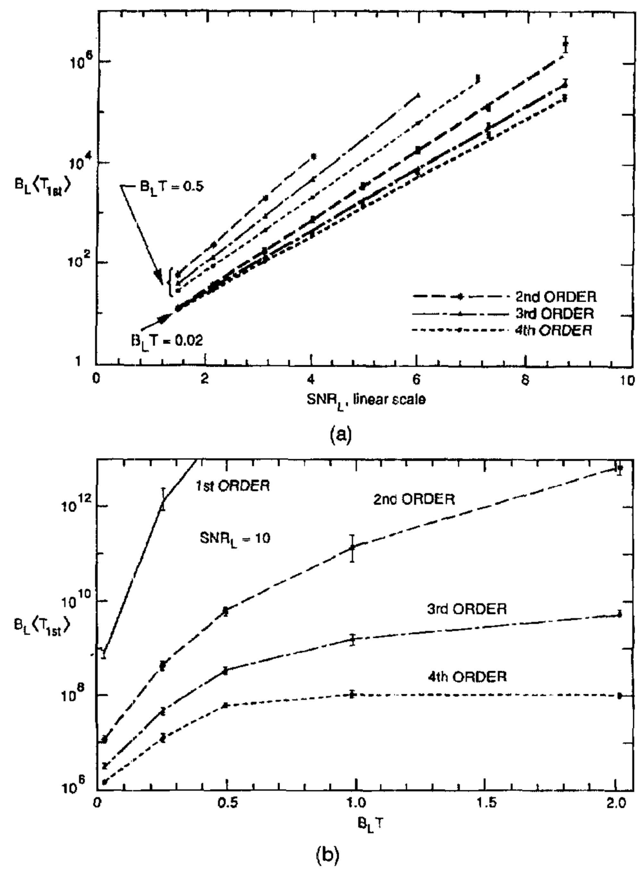

Fig. 6. Mean time to first cycle slip for DPLLs with supercritical damping, phase/phase-rate feedback, no computation delay, and sine phase extractor.

loop, as can be derived by expressing K_N for each loop order in Table II in terms of B_L and then substituting the result in the expression for $\tilde{\phi}_{ss}$ in Table IX. For large values of $B_L T$, however, $\tilde{\phi}_{ss}$ in a DU loop is not proportional to B_L^{-N}, as illustrated in Fig. 7. Fig. 7(a) plots as a function of $B_L T$ the dimensionless coefficient required to multiply the CU-limit form for $\tilde{\phi}_{ss}$. These plots pertain to loops of order 1 to 4, with phase and phase-rate feedback and with the indicated damping and computation delay. At $B_L T = 0$, the coefficient is equal to the CU-limit value. As $B_L T$ increases, the increase in this coefficient relative to the zero-$B_L T$ value is a measure of the "excess" $\tilde{\phi}_{ss}$ relative to the nominal CU-limit values. As Fig. 7(a) indicates, for example, $\tilde{\phi}_{ss}$ at $B_L T = 0.5$ for a third-order, standard underdamped loop, is about three times larger than the CU limit would predict.

Fig. 7(b) plots the corresponding effective loop bandwidth as determined from $\tilde{\phi}_{ss}$, where "effective" denotes the decrease in bandwidth relative to the B_L^N model. For example, the effective bandwidth is about 0.6 times the true loop bandwidth when $B_L T = 0.5$ for a second-order loop. Thus, the effective "dynamic" loop bandwidth, with regard to $\tilde{\phi}_{ss}$, becomes progressively smaller relative to loop noise bandwidth as $B_L T$ increases.

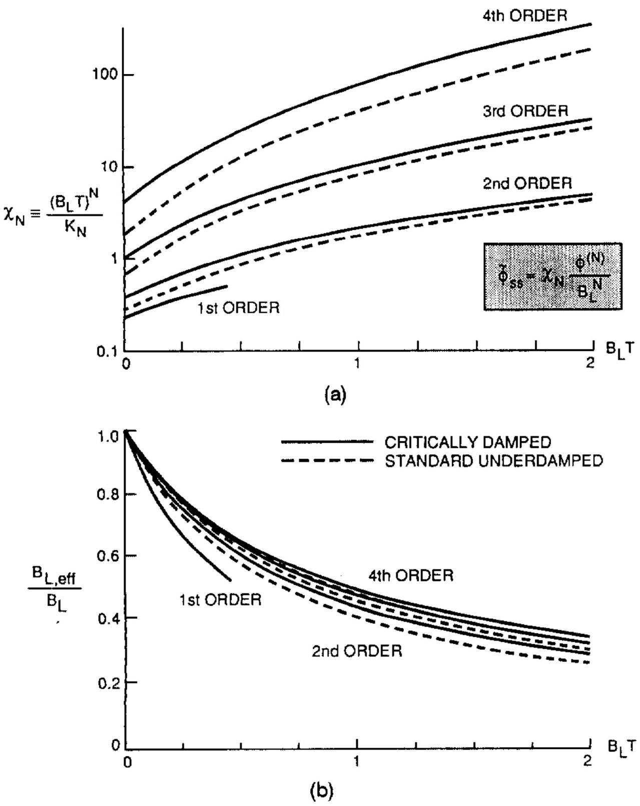

Fig. 7. Quantities related to steady-state phase error in DPLLs with phase/phase-rate feedback and no computation delay.

VII. CONCLUSIONS

A first-principles analysis of DPLLs has led to a new approach for parameterizing loops that is not complicated by analog considerations. Loop constants are determined from loop roots that can be selectively placed in the s-plane on the basis of loop noise bandwidth and new independent parameters that have simple and direct significance relative to root-specific decay-rate and root-specific damping. The formalism can be systematically extended to loops of arbitrary order and provides great flexibility in directly placing roots in the s-plane.

In the CU limit, loop constants for loops of first to fourth order are obtained in closed form as functions of the new parameters. In a solution for a DU loop, however, complexity of the equations leads to a numerical approach. The analysis has been applied to loops with either phase and phase-rate feedback or phase-rate-only feedback, with supercritical damping or standard underdamping and with either zero computation delay or a computation delay equal to one update interval. With the new parameterization, exact selection of true loop noise bandwidth and root-by-root selection of damping and relative decay-rate can be carried out for high order loops, even when $B_L T$ is large. Analysis and design of third- and fourth-order loops becomes more straightforward and understandable.

To improve the versatility and reliability of acquisition in high-order loops, a method for direct, transient-free acquisition has been presented. Given adequate *a priori* estimates of phase and its derivatives, steady-state signal lock can be obtained directly with third- and fourth-order loops without first acquiring with lower order loops. For appropriate applications, use of this method can allow direct and reliable acquisition with a third- or fourth-order loop at smaller loop bandwidth and lower signal strength than traditional approaches.

As a measure of the performance of large-$B_L T$ loops, simulations of loop behavior in terms of mean time to first cycle slip have been carried out for loops of first- to fourth-order based on the new parameterization. The simulated loops have phase and phase-rate feedback, supercritical damping, and no computation delay. For a given loop bandwidth, loops with larger $B_L T$ exhibit a considerably better (larger) mean time to first cycle slip than those with smaller $B_L T$ values.

Loop performance has also been assessed on the basis of steady-state phase error. As $B_L T$ increases for a fixed value of B_L, the steady-state phase error is essentially constant for small $B_L T$ values (e.g., $B_L T \leq 0.02$) but increases for larger values of $B_L T$. Thus, by this measure loop performance with respect to dynamics deteriorates as $B_L T$ increases, given fixed B_L.

ACKNOWLEDGMENTS

The authors would like to thank Dr. W. C. Lindsey for his review of the paper and valuable suggestions.

REFERENCES

[1] Lindsey, W. C., and Chie, C. M. (1981)
A survey of digital phase-locked loops.
Proceedings of IEEE, **69**, 4 (Apr. 1981), 410–431.

[2] Aguirre, S., and Hurd, W. J. (1984)
Design and performance of sampled data loops for subcarrier and carrier tracking.
TDA progress report 42-79, Jet Propulsion Laboratory, Pasadena, CA, July 1984, 81–95.

[3] Thomas, J. B. (1989)
An Analysis of digital phase-locked loops.
Publication 89-2, Jet Propulsion Laboratory, Pasadena, CA, Feb. 1989.

[4] Bracewell, R. (1965)
The Fourier Transform and its Applications.
New York: McGraw-Hill, 225.

[5] Gradshteyn, I. S., and Ryzhik, I. M. (1980)
Table of Integrals, Series, and Products (corrected and enlarged ed.). New York: Academic Press, 218.

[6] Lindsey, W. C. (1972)
Synchronization systems in communications and control.
Englewood Cliffs, NJ: Prentice-Hall, 1972.

[7] Viterbi, A. J. (1966)
Principles of Coherent Communications.
New York: McGraw-Hill, 1966.