A Methodical Approach to Hybrid PLL Design for High-Speed Wireless Communications

Yair Linn

University of British Columbia

6335 Thunderbird Crescent, Box 341, Vancouver, BC, Canada, V6T-2G9

e-mail: ylinn@ece.ubc.ca

ABSTRACT

Coherent demodulation in digital wireless communications involves generating a local carrier that is in phase with the received carrier, and then using this local carrier in order to coherently demodulate the received signal. Generation of the local carrier is done via a carrier synchronization Phase Lock Loop (PLL). The receiver also includes a symbol timing synchronization PLL, whose purpose is to determine the optimal times for sampling of the recovered symbol waveforms. PLLs are either: (i) analog; (ii) digital; or (iii) hybrid. In this paper we develop a methodical approach for the application of digital signal processing theory to the design and implementation of hybrid PLLs used in coherent receivers. The methodological nature of our approach will manifest itself in the development of a clear step-by-step procedure for the design of the constituent components of the hybrid PLL. This procedure will explicitly address various aspects and difficulties unique to hybrid implementations.

1. Introduction

Coherent communications has the advantage of being able to provide superior error-rate performance^[1 Chap. 5], with the tradeoff of increased complexity at the receiver, which contains various PLLs. With the advent of faster and better A/D converters, FPGAs, and ASICs, attention in the past two decades has turned from analog to digital implementations^[2-5] - i.e. where sampling of the received IF or near-baseband signal is performed, and the entire demodulation process is done digitally. However, many PLLs have neither completely analog nor completely digital implementations. Looking for example at the carrier PLL, a hybrid implementation is shown in Fig. 1. Hybrid implementations are particularly suited for high-datarates because the sampling rate of a hybrid receiver can be as low as 1 sample/symbol, whereas completely digital demodulation needs at a minimum 2 samples/symbol and more likely around 2.5 samples/symbol for adequate performance^[5]. Moreover, in hybrid receivers the downconversion is done in the analog domain which simplifies the receiver's digital portion.

As for advantages of the hybrid topology vis-à-vis analog implementations, they include: (a) superior repeatability and filter specifications; (b) arbitrarily complicated phase detector and filter structures; (c) the long-term stability and phase noise characteristics achievable with DDS chips are difficult to attain using a VCO; (d) enhanced testability and probing.

This paper focuses on fixed-point hardware implementations. The reason is that, first, such

implementations will always have distinct performance advantages since they can always be made to operate faster than any software and/or floating point implementation. Secondly, more intriguing challenges are present when trying to design a fixed-point hardware system: whereas a software system could be implemented using a high-level language, when implementing a fixed-point hardware system the designer must explicitly address such issues as scaling and overflow, logic resource usage, implementation of mathematical operations, etc.

This paper is a shortened and revised summary of an invited workshop^[6,7], to which the reader is referred for a more detailed treatment. In this paper treat only the carrier PLL, though the derivations are general and can be applied to other PLLs^[6,7].

2. Analog PLL Theory

As a preliminary step before discussing hybrid PLLs, we now establish a step-by-step design procedure for analog PLLs. Subsequently, we shall use this procedure as a basis upon which we shall develop a step-by-step procedure for hybrid PLL design.

The PLL's purpose is to produce an output signal $y_o(t)$ that is in frequency and phase synchronization with the input signal $y_i(t)$. A diagram of an analog PLL is shown in Fig. 2, where $y_i(t) = A \cdot \sin(\omega_i t + \theta_i(t))$ and $y_o(t) = B \cdot \cos(\omega_o t + \theta_o(t))$. The fact that both signals contain the term $\omega_i t$ doesn't necessarily imply that their frequencies are equal; we have simply incorporated the frequency-error induced phase difference, defined as $\Delta\omega \cdot t$, into $\theta_o(t)$ (see [8 Sec. 3.1]).

Fig. 1 - General structure of a hybrid carrier PLL for digital wireless communications. The parts within the dashed line are implemented digitally, while the rest are analog components (the samplers and the DDS are mixed-signal components). $1/T_s$ is the sample rate.

The PLL's purpose is to control the VCO such that $\theta_i(t) = \theta_o(t)$. Note that $y_i(t)$ and $y_o(t)$ are in quadrature, but coherency is easily obtained (when in lock) via a 90° phase shift network applied to $y_o(t)$ (not shown in Fig. 2).

2.1. The equivalent baseband nonlinear model

$\theta_i(t)$, the phase information upon which the PLL operates, is “hidden” within the input signal $y_i(t) = A \cdot \sin(\omega_i t + \theta_i(t))$.

We can only extract phase information by using a *nonlinearity* in the phase detector. This hampers PLL theoretical analysis since we must contend with *nonlinear* differential equations. Consider the phase detector $PD(y_i(t), y_o(t)) = y_i(t) \cdot y_o(t)$:

$$\begin{aligned} PD(y_i(t), y_o(t)) &= y_i(t) \cdot y_o(t) \\ &= A \sin(\omega_i t + \theta_i(t)) \cdot B \cos(\omega_o t + \theta_o(t)) \\ &= 0.5AB (\sin(\theta_i(t) - \theta_o(t)) + \sin(2\omega_i t + \theta_i(t) + \theta_o(t))) \end{aligned} \quad (1)$$

If we assume that the loop filter eradicates the double frequency term, the effective phase detector output is:

$$PD(y_i(t), y_o(t)) = 0.5AB \sin(\theta_i(t) - \theta_o(t)) \quad (2)$$

Note that (2) is a nonlinear function of $\theta_i(t)$ and $\theta_o(t)$.

We already saw in (2) that the effective phase detector output is a function of $\theta_i(t)$ and $\theta_o(t)$, not of $y_i(t)$ and $y_o(t)$. Therefore, we can analyze the PLL using a baseband model that operates upon phase signals, as shown in Fig. 3. We have made some assumptions regarding the loop components in Fig. 3: (a) the phase detector is a nonlinear device that outputs a function of the phase difference $\theta_i(t) - \theta_o(t)$, that we denote $PD(\theta_i(t) - \theta_o(t))$; (b) the PLL's loop filter is Linear Time Invariant (LTI) with impulse response $f(t)$; (c) the VCO is a device that outputs a sinusoid with a frequency that is a function of its input voltage. Frequency is the derivative of the phase. Therefore, we can write $d(\theta_o(t))/dt = K_V \cdot u_2(t)$ (where K_V is a constant with units rads/(volts·sec) which is dependent upon the VCO characteristics). Thus we have the following:

$$u_1(t) = PD(\theta_i(t) - \theta_o(t)) \quad (3)$$

$$u_2(t) = u_1(t) \otimes f(t) \quad (4)$$

$$\frac{d(\theta_o(t))}{dt} = K_V \cdot u_2(t) \quad (5)$$

We can combine (3)-(5) to reach the differential equation:

$$d(\theta_o(t))/dt = K_V \cdot (PD(\theta_i(t) - \theta_o(t)) \otimes f(t)) \quad (6)$$

Eq. (6) is problematic because it is a *nonlinear* differential equation (due to the function PD). For the current analysis, we assume the phase detector has a sinusoidal function, i.e. that:

$$PD(\theta_i(t) - \theta_o(t)) = K_d \sin(\theta_i(t) - \theta_o(t)) \quad (7)$$

K_d is called the *phase detector gain* (for example for a multiplier phase detector we have from (2) that $K_d = 0.5AB$). As we shall see, we suffer no true loss of generality from the assumption of a sinusoidal phase detector function, since linear-model analysis assumes that the PLL is locked, and then the phase detector curve shape is irrelevant and the only important^[6] parameter is K_d .

Fig. 2 – Analog PLL showing input and output signals.

2.2. The loop filter

The majority of PLLs are 2nd-order loops, for which the loop filters are extremely simple^[9 Sec. 2.4]. We assume that the loop filter is:

$$F(s) = K_a \frac{1 + s\tau_2}{1 + s\tau_1} \quad (8)$$

For discussion of other loop filters see [7], [6], and [8 Chaps. 3-5]. The filter of (8) is easily implemented as a resistor-capacitor network^[9 Chap. 2], but we will not discuss this implementation since we are interested in a *digital* implementation (see Fig. 1).

2.3. Nonlinear PLL equation

We can arrive at a nonlinear differential equation for the analog PLL. Define for convenience the *loop gain* as $K \triangleq K_d K_a K_V$. We can then use (6)-(8) along with the equivalence of the Laplace variable to the derivative operator in the time domain (that is, $s = L\{d(\bullet)/dt\}$) to arrive at^[8 Sec. 3.5]:

$$\tau_1 \frac{d^2 \theta_o}{dt^2} + \frac{d\theta_o}{dt} = K \sin(\theta_i - \theta_o) + K\tau_2 \cos(\theta_i - \theta_o) \left(\frac{d\theta_i}{dt} - \frac{d\theta_o}{dt} \right) \quad (9)$$

2.4. Linearized PLL model

Eq. (9) is a nonlinear differential equation that is difficult to solve theoretically (although it is very easy to simulate). For theoretical analysis, we assume that the PLL is locked and that the phase error is small enough, i.e. $\theta_e = \theta_i - \theta_o \rightarrow 0$, so we can simplify (9) to the following *linear* differential equation (using $\cos(\theta_i - \theta_o) \rightarrow 1$ and $\sin(\theta_i - \theta_o) \rightarrow \theta_i - \theta_o$):

$$\tau_1 \frac{d^2 \theta_o}{dt^2} + (1 + K\tau_2) \frac{d\theta_o}{dt} + K\theta_o = K\tau_2 \frac{d\theta_i}{dt} + K\theta_i \quad (10)$$

An important conclusion from the assumption of small $\theta_e(t)$ is that the linear model is valid for small $\theta_e(t)$ for *any* phase detector for which $PD(\theta_e(t)) \xrightarrow{\theta_e \rightarrow 0} K_d \theta_e(t)$. In fact, *all* phase detectors that have good performance fulfill $PD(\theta_e(t)) \xrightarrow{\theta_e \rightarrow 0} K_d \theta_e(t)$. Thus, we can conclude that the linear model is valid for all phase detectors, provided that $\theta_e(t) \rightarrow 0$. Hence the only thing we need to find out in order to model a particular phase detector is to ascertain its K_d .

2.5. Solutions of the linearized PLL equation

Given $\theta_i(t)$ we can solve (10) in the time domain^[9 Chap. 4],^[8 Chap. 5], but it is more instructive to solve in the Laplace domain. In the Laplace domain, the VCO transfer function is (from inspection of (5)) $V(s) \triangleq \theta_o(s)/U_2(s) = K_V/s$. The open loop function is (see Fig. 4):

$$G(s) = K_d F(s) V(s) = K_d F(s) K_V / s \quad (11)$$

The closed loop function of the PLL linear model (Fig. 4) is:

$$H(s) = \frac{\theta_o(s)}{\theta_i(s)} = \frac{G(s)}{1+G(s)} = \frac{K \frac{\tau_2}{\tau_1} s + \frac{K}{\tau_1}}{s^2 + \frac{1+K\tau_2}{\tau_1} s + \frac{K}{\tau_1}} \triangleq \frac{P(s)}{Q(s)} \quad (12)$$

This corresponds to the transfer function of a second order system, whose denominator is often written in the form of:

$$Q(s) = s^2 + 2\zeta\omega_n s + \omega_n^2 \quad (13)$$

where ζ the *damping ratio* and ω_n is the *natural (radian) frequency*. ζ and ω_n have been extensively studied in control literature^[9-12]. Comparing (12) and (13), we find that:

$$\omega_n = \sqrt{K/\tau_1} \quad (14)$$

and using (12), (13) and (14):

$$\zeta = 0.5\sqrt{K/\tau_1} \cdot (\tau_2 + 1/K) \quad (15)$$

and by plugging (14) and (15) into (12) we get:

$$H(s) = \frac{(2\zeta - \omega_n/K)\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (16)$$

As is customary, we assume a high-gain loop, as defined by:

$$K \gg \omega_n \quad (17)$$

There is no true loss of generality incurred by making this assumption, as virtually all PLLs (indeed, all control loops) are designed to have a high loop gain. This is because a high loop gain makes the loop bandwidth and damping factor relatively insensitive^[10 Chap. 14] to variations in the gains K_d , K_a , and K_V .

The magnitude of the relationship (17) is usually also determined by the allowable steady-state error^{[6],[8 Chap. 5]}. For example, for a 2nd-order PLL with $F(s)$ of (8) the steady state error to a frequency step of $\Delta\omega$ rads/sec will be $\Delta\omega/K$ rads^[8 Chap. 5]. If the system is such that all frequency steps encountered follow $|\Delta\omega| \leq \omega_n$, then a constraint ensuring an acceptable upper bound on the steady state error is, say, $K = 100\omega_n$. Other steady-state requirements will result in different instantiations for (17).

Using (17), eq. (16) can be reduced to:

$$H(s) = \frac{2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (18)$$

which is a transfer function form that has been analyzed extensively in PLL and control systems literature^[9-12].

3. Choice of Analog PLL Parameters

3.1. Choosing ζ

The choice of ζ is determined by our desire to achieve the fastest PLL response but with minimal overshoot. Moreover, we would like ζ to be optimal in terms of noise performance. We choose $\zeta=0.95$ because: (a) it is a good compromise between the optimal values for several optimization criteria^[9 Table 7.2]; (b) the PLL response is underdamped (i.e. fast response), but without too much overshoot^[9 Chap. 2, 13 Chap. 2], and we get a phase margin of about 75° , which doesn't increase much if we increase ζ further^[13 Fig. 2.4-3].

3.2. Finding the optimal ω_n

The choice of ω_n in coherent wireless communications is usually determined by the desire to achieve minimal phase-error jitter $\overline{\theta_e^2}$. This is because phase-jitter directly affects the error rate^[3 Fig 2.9]. Formally, suppose that the input carrier

Fig. 3 – Nonlinear equivalent baseband model of the PLL.

Fig. 4 – Linearized analog PLL model.

phase noise has a spectrum $\Phi_{\theta_{pi}}(f)$ and that the thermal-induced phase noise has a spectrum $\Phi_{\theta_{ti}}(f)$. The phase-error variance will be^[14 App. A]:

$$\overline{\theta_e^2} = \int_0^\infty \Phi_{\theta_{ti}}(f) |H(j2\pi f)|^2 df + \int_0^\infty \Phi_{\theta_{pi}}(f) |1 - H(j2\pi f)|^2 df$$

Optimization for minimal jitter is to choose ω_n so that this is minimized. For detailed discussions see [15] and [8 Chap. 8].

4. A Step-by-Step Analog PLL Design Procedure

We now present a step-by-step analog PLL design procedure. Assumed given are the input carrier phase noise characteristics and the input SNR (Signal to Noise Ratio) denoted^{[16 Chap. 6],[6]} as SNR_i . Clearly, the PLL must operate at various SNR_i s, so we should design the PLL for the worst-case value (i.e., the lowest SNR_i). The procedure is shown in Fig. 5. The most difficult step is determination of ω_n , since this involves addressing the input carrier phase noise and the effects of thermal noise^[6, 8 Chap. 8, 15 Chap. 10].

Fig. 5 was developed with no specific modulation in mind, but it is applicable to any modulation type. The only modulation-dependant variables are (a) the phase detector and its gain K_d , and (b) the squaring loss (which influences^[9 Sec. 11.2, 15 Chap. 10] noise calculations and thus the choice of ω_n). Both K_d and the squaring loss are easily derived: K_d is calculated from the slope of the phase detector S-Curve at $\theta_e=0$ (can be done via simple open-loop simulations), while the squaring loss is calculated theoretically or via simple open-loop simulations^{[17],[18],[19],[2 Chap. 5, 6]}.

5. Hybrid Implementation of the PLL

5.1. Preliminary assumptions

This paper's goal is to develop a procedure for the design of fixed-point hardware hybrid PLLs. We assume a high data rate (e.g. above 1 MHz) since (as discussed in the introduction) it is for such high data rates that hybrid receivers are a particularly attractive design choice. A phase detector in digital communications usually produces a new phase estimate for every symbol^{[2],[3],[20]}, and thus the phase detector output sample rate is assumed to be the same order of magnitude as the symbol rate. Thus, denoting the phase detector output sample rate as $f_p = 1/T_p$, we have that $1/T_p \sim 1/T$ where $1/T$ is the symbol rate and “ $\sim$ ” denotes equal orders of magnitude. Again, *this rate is assumed to be at least 1 MHz*.

Carriers that are used in coherent communications generally have phase noise whose content can be assumed non negligible up to a distance from of at most several KHz. For example in the DVB-S2 standard^[21], the specification is -68 dBc/Hz at 1 KHz offset from the carrier. The PLL's natural frequency in Hz $f_n = \omega_n / 2\pi$ is in general the same order of magnitude as the bandwidth of the significant phase-noise content of the received carrier. While f_n is dependent upon the receiver, we can safely say that the *order of magnitude for f_n is at most several KHz*. An important conclusion is that we can safely say that:

$$f_p \gg f_n \quad (19)$$

Eq. (19) is crucial, since it is why the analysis of the PLL discussed in this paper differs from that of just any hybrid control loop.

5.2. The Direct Digital Synthesizer (DDS)

The Direct Digital Synthesizer (DDS) in Fig. 1 is a chip that accepts a digital control word (the *tuning word*) at its input, and outputs a continuous-time sinusoidal waveform whose frequency corresponds to that tuning word. The DDS consists of a Numerically Controlled Oscillator (NCO) followed by a Digital to Analog Converter (DAC), followed by smoothing filter that rejects the image frequencies that appear at the DAC output^{[14 App. A],[22]}. The frequency waveform produced by the DDS is always phase-continuous^[22], regardless of the frequencies through which its output transitions. The linearized PLL model for Fig. 1 is shown in Fig. 6. An example DDS chip is the AD9851 manufactured by Analog Devices^[22].

As seen in Fig. 7, the tuning word controls the output frequency by determining by how much the phase of the output wave will increase each rising edge of the reference clock. Denoting this clock rate as $f_D = 1/T_D$, the number of bits in the phase accumulator register as N and the tuning word as w_D , the output frequency is:

$$f_w = w_D / (2^N \cdot T_D) = (w_D / 2^N) \cdot f_D \quad (20)$$

Of course, (20) is only valid if the Nyquist criterion is observed, i.e. that there are more than two samples of each period of the output sinusoid. This means that (20) holds if:

$$f_w = (w_D / 2^N) \cdot f_D < f_D / 2 \quad (21)$$

Thus the largest tuning word that can produce a unaliased sinusoid will be $w_D = 2^{N-1}$, and that waveform will have a frequency of $f_w = ((2^{N-1} - 1) / 2^N) \cdot f_D \xrightarrow{N \rightarrow \infty} 0.5 f_D$. However, this limit cannot be achieved in practice because the post-DAC filter needed to suppress the first image frequency (which is^[22] at $f_D - f_w$) will be unrealizable. A more reasonable limit is $f_{w,max} \approx 0.4 f_D$.

6. Modeling of the DDS

The tuning word of the DDS is updated at a rate of $f_u = 1/T_u$ Hz that is much slower than f_D , because several samples of the NCO must be traversed between updates of the tuning word for those updates to be expressed at the DDS output. Thus we may write:

$$f_u \ll f_D \quad (22)$$

The operation of the DDS within a PLL is illustrated in Fig. 8. The mathematical model of the DDS is shown in Fig. 9. To see why this model is valid, we compare the behavior of the

Fig. 5 – Step-by-Step PLL design procedure for analog PLLs.

systems in Fig. 9 and Fig. 8 (right), and realize that that behavior is identical. It is emphasized that neither the DAC nor the VCO in Fig. 9 are in fact present in the DDS; they are purely virtual mathematical constructs. As a consequence, we can define the parameters of the DAC and VCO as we wish, so long as we insure that the input and output of the model in Fig. 9 behaves identically to that of the DDS. With that constraint in mind, we now characterize the virtual DAC and VCO. First, regarding the DAC, since the DDS is updated at a rate of f_u that is the conversion rate of the virtual DAC in Fig. 9. Furthermore, since the DDS output frequency remains constant between tuning word updates, the DAC in Fig. 9 must be of the ZOH (Zero Order Hold) type^[11 Chap. 11]. We further decide that the DAC has unit gain, i.e. its output (which has the dimensions of Volts) is simply w_D Volts when it has the tuning word w_D at its input. Since the DAC is updated every T_u seconds, we have^{[6],[7]}:

$$H_{ZOH}(s) = T_u e^{-\frac{sT_u}{2}} \frac{\sinh(sT_u/2)}{sT_u/2} \quad (23)$$

The DAC receives a digital signal as its input and outputs a continuous time signal, and hence strictly speaking it cannot

Fig. 6 – Hybrid PLL equivalent linearized baseband model.

be represented in the Laplace domain. To circumvent this obstacle, the DAC is modeled by a system with a transfer function of (23) that is preceded by a module that converts the DAC's input into a train of impulses in continuous time. Denoting the DAC's input as $v(n)$ and ZOH input as $\eta(t)$, we have $\eta(t) = \sum_{n=-\infty}^{\infty} v(n)\delta(t - nT_u)$.

Turning now to the DDS model's virtual VCO, two parameters are of interest: its center frequency and its tuning sensitivity. The center frequency is unimportant for the equivalent linearized baseband PLL analysis. To characterize the tuning sensitivity K_V , we note that the output phase of the VCO is:

$$\begin{aligned}\theta_{out}(t) &= 2\pi \int_0^t f_w(\tau) d\tau \\ &= (2\pi / 2^N) f_D \int_0^t w_D(\tau) d\tau\end{aligned}\quad (24)$$

Taking the Laplace transform of both sides of (24), we have:

$$\theta_{out}(s) = \frac{2\pi}{2^N} f_D \cdot w_D(s) / s \quad (25)$$

Remembering that we have set the DAC in Fig. 9 to have unit gain, we have that the voltage at the input of the VCO is $V_D(t) = w_D(t)$ and consequently $V_D(s) = w_D(s)$. Plugging the last equality into (25) and dividing both sides by $V_D(s)$, we have that the transfer function of the VCO is

$$H_V(s) \triangleq \theta_{out}(s) / V_D(s) = (2\pi / 2^N) f_D / s.$$

Comparing this to the transfer function form of a VCO, which is K_V/s we have by inspection:

$$K_V = (2\pi / 2^N) f_D \text{ Radians} \times \text{Sec}^{-1} \times \text{Volt}^{-1} \quad (26)$$

Eqs. (23) and (26) allow us to express Fig. 9 in Laplace terms, shown in Fig. 10. Hybrid PLL analysis is now facilitated by inserting the model of Fig. 10 into Fig. 6 which results in Fig. 11. There is an unresolved issue in Fig. 11: the sampling rates at the input and output of $B(z)$ are f_p and f_u , respectively, and these aren't necessarily equal. To establish a relationship between f_p and f_u , recall that $f_p \gg f_n$. Thus, we make the assumption that:

$$f_p \geq f_u \gg f_n \quad (27)$$

The magnitude discrepancy expressed in (27) can be quite dramatic. For example, consider a 100 MHz symbol rate data system ($f_p = 100 \cdot 10^6$ Hz). Typical values for f_n are $f_n \sim 2000$ Hz. The DDS update rate (as we shall see shortly), would be, say, $f_u = 2 \cdot 10^6$ Hz. Given the magnitude difference between f_p and f_u , we desire two things. First, we wish to find the minimum f_u that allows the PLL to function acceptably. Secondly, we wish to implement the loop filter at the much lower clock rate of f_u (rather than at rate f_p). It is thus evident that the sampling rate

Fig. 7 – Explanation of DDS Operation.

of the data at the input of $B(z)$ must be reduced, through use of decimation. This is shown in Fig. 12. The decimation ratio in Fig. 12 is:

$$M = T_u / T_p = f_p / f_u \quad (28)$$

In Fig. 12 we assume implicitly that the spectral content of $\theta_e(nT_p)$ is fully contained in the continuous frequency domain within $[-f_u/2, f_u/2]$. To see why this always holds, define $f \in [-f_{\max}, f_{\max}]$ as the bandwidth of $\theta_e(t)$. It is easily shown that $f_{\max}$ is the same order of magnitude as f_n . But from (27) $f_p \geq f_u \gg f_n$ so we are ensured $f_p \geq f_u \gg f_{\max}$.

7. Calculation of the Digital Loop Filter

The models obtained in previous sections allow us to now compute $B(z)$ so that a desired closed loop transfer function for the PLL is achieved. In this section we assume that the virtual DAC (which is part of the DDS equivalent model), the decimation filter, and the decimator are all ideal (later we derive conditions that ensure that these assumptions are appropriate). Under those assumptions, the series of operations sampling at rate f_p -decimation filter-decimator by M is

tantamount to sampling the signal $K_d\theta_e(t)$ at rate f_u . Thus, we may simplify Fig. 12 to Fig. 13.

Comparing Fig. 13 to Fig. 4 it is seen by inspection that $\hat{F}(s)$ in Fig. 13 corresponds to $F(s)$ in Fig. 4. Since all other loop components are equal, if we can design a filter $B(z)$ so that $\hat{F}(s) = F(s)$ this will imply that the closed loop transfer function of both systems will be identical. Consequently, from inspection of the structure of $\hat{F}(s)$ it is evident that $B(z)$ may be deduced from $F(s)$ by using the *bilinear transformation method*^[23 Sec. 7.1]. We now proceed to find $B(z)$.

$F(s) = K_a \frac{1+s\tilde{\tau}_2}{1+s\tilde{\tau}_1}$ has a pole at $s = -1/\tilde{\tau}_1$ and a zero at

$s = -1/\tilde{\tau}_2$. To use the bilinear method, we pre-warp^[23 Sec. 7.1]:

$$\frac{1}{\tilde{\tau}_1} = \frac{1}{\pi T_u} \tan\left(\frac{\pi T_u}{\tau_1}\right) \quad \text{and} \quad \frac{1}{\tilde{\tau}_2} = \frac{1}{\pi T_u} \tan\left(\frac{\pi T_u}{\tau_2}\right) \quad (29)$$

and then construct the pre-warped transfer function of the analog filter $D(s) = K_a \frac{1+s\tilde{\tau}_2}{1+s\tilde{\tau}_1}$. Now we can employ the

bilinear transformation^[23 Sec. 7.1] in order to determine $B(z)$ as follows:

$$\begin{aligned} B(z) &= D(s) \Big|_{s=\frac{2}{T_u} \left(\frac{1-z^{-1}}{1+z^{-1}} \right)} = K_a \frac{1 + \frac{2}{T_u} \left(\frac{1-z^{-1}}{1+z^{-1}} \right) \tilde{\tau}_2}{1 + \frac{2}{T_u} \left(\frac{1-z^{-1}}{1+z^{-1}} \right) \tilde{\tau}_1} \\ &= K_a \left(\frac{1 + \frac{2\tilde{\tau}_2}{T_u}}{1 + \frac{2\tilde{\tau}_1}{T_u}} \right) \cdot \left(\frac{1 + \left[\left(1 - \frac{2\tilde{\tau}_2}{T_u} \right) / \left(1 + \frac{2\tilde{\tau}_2}{T_u} \right) \right] z^{-1}}{1 + \left[\left(1 - \frac{2\tilde{\tau}_1}{T_u} \right) / \left(1 + \frac{2\tilde{\tau}_1}{T_u} \right) \right] z^{-1}} \right) \end{aligned} \quad (30)$$

Now let us define $\beta_1 = (1 - 2\tilde{\tau}_2/T_u)/(1 + 2\tilde{\tau}_2/T_u)$,

$\alpha_1 = (1 - 2\tilde{\tau}_1/T_u)/(1 + 2\tilde{\tau}_1/T_u)$, and also

$\gamma = K_a (1 + 2\tilde{\tau}_2/T_u)/(1 + 2\tilde{\tau}_1/T_u)$, then $B(z)$ may be

expressed as $B(z) = \gamma \cdot \left(\frac{1 + \beta_1 z^{-1}}{1 + \alpha_1 z^{-1}} \right)$. The direct-form II

implementation^[23 Chap. 6] is shown in Fig. 14.

8. Determination of DDS Update Rate

The determination of the previous section assumed that the (virtual) DAC converter in Fig. 9 is ideal. However, that DAC is not ideal, and actually that DAC is of the ZOH type (Fig. 10). This profoundly affects the PLL and constrains the rate f_u above a certain minimum, which will now be calculated. Modifying Fig. 13 to account for the DAC's non-ideality, while still assuming ideal decimation, results in Fig. 15.

While amplitude interference from aliases passed by the non-ideal virtual DAC will cause a degradation in performance, it will not in general lead to PLL instability^[6]. In contrast, the effect upon the phase response of the signals in the PLL is much more important and will lead to a higher

Fig. 8 – DDS operation vs. operation of an analog VCO. $f_i = \omega_i/2\pi$ is the input signal's frequency; f_o is the analog VCO output frequency; f_w is the DDS's output frequency. Left: operation of analog PLL with VCO (PLL in perfect lock); Right: operation of hybrid PLL with DDS (PLL in perfect lock). The granularity of the frequency steps is exaggerated in order to demonstrate the effects of the update rate on the DDS output frequency.

Fig. 9 – Equivalent mathematical model of a DDS.

Fig. 10 – Laplace domain model of the DDS.

Fig. 11 – Hybrid PLL model with DDS equivalent model inserted.

necessary minimum f_u . We analyze the effect upon the phase response by considering the effect on the Phase Margin (PM) of the PLL.

We now assume that the aliases contribute negligibly to the signal at the output of the DAC (see [6] for justification). Furthermore, we assume that the ZOH magnitude distortion of the primary reconstructed signal is negligible (this is true due to $f_u \gg f_{max}$; see Sec. 6 and [6]). Under these assumptions, we can reduce Fig. 15 to Fig. 16. A comparison of Fig. 16, Fig. 13, and Fig. 4, shows that in Fig. 16 the open loop transfer function will be:

$$\hat{G}(s) = K_d \hat{F}(s) K_v / s = K_d e^{-0.5sT_u} F(s) K_v / s = e^{-0.5sT_u} G(s) \quad (31)$$

The phase margin of a PLL is defined as the number of degrees above -180 that the phase of the open loop response possesses when its magnitude is unity^[13 Chap. 2]. For the systems of Fig. 13 and Fig. 4, if f_c , which we call the *crossover frequency*, is the frequency where $|G(j2\pi f_c)| = 1$, then the phase margin for those systems is:

$$PM(G) = \angle G(j2\pi f_c) - (-180) \text{ Degrees} \quad (32)$$

where $\angle G(j2\pi f_c)$ is in degrees. If we now turn our attention to the system of Fig. 16, we have from (31) that:

$$|\hat{G}(j2\pi f_c)| = |e^{-j\pi f_c T_u} G(j2\pi f_c)| = |G(j2\pi f_c)| = 1 \quad (33)$$

that is, the crossover frequency remains unchanged.

Evaluating the phase margin of $\hat{G}(s)$, we have:

$$PM(\hat{G}) = \angle \hat{G}(j2\pi f_c) - (-180) = \angle e^{-j\pi f_c T_u} G(j2\pi f_c) - (-180) = PM(G) - 180 \cdot f_c T_u \text{ Degrees} \quad (34)$$

We thus observe a decrease in the phase margin by $180 \cdot f_c T_u$ degrees. A decrease in the phase margin is detrimental, since it may cause an underdamped response of the PLL or its outright instability^[13 Chap. 2]. Thus we must find a condition on f_u so that that decrease is tolerable. If we are willing to tolerate a phase margin decrease of d_A degrees, from (34):

$$f_u > (180/d_A) \cdot f_c \quad (35)$$

It can be shown that we have^[13 Sec. 2.4]:

$$f_c \approx 2\zeta f_n \quad (36)$$

Thus from (35) and (36) the constraint is:

$$f_u > (180/d_A) \cdot 2\zeta f_n \quad (37)$$

Exact linear-systems analysis of models which include terms of the type $\exp(-s\tau)$ is impossible due to the non-polynomial nature of this term^[10 Sec. 7.12]. Instead, we compute the effect on the PM and then using the relationship^[13 Fig. 2.4-3] $\zeta \approx 0.01 \cdot PM$ we define an *effective* ζ , denoted as ζ_{eff} , defined as $\zeta_{eff} \triangleq 0.01 \cdot PM$. We can then think of the PLL with as a second-order system with damping factor ζ_{eff} . The allowable phase margin degradation will be determined from the values of ζ_{eff} that the designer is willing to accept (usually $0.5 \leq \zeta_{eff} \leq 0.75$). Hence the design starts by determining the acceptable range for ζ_{eff} , which then (taking into account other PM degradation sources (such as latency, see Sec. 10) and considering that the initial PM is^[13 Fig. 2.4-3] at most 75°) determines the allowable PM degradation due to the DDS, denoted d_A , which in turn determines the DDS update rate via (37).

Acceptable^[9 Chap. 7] values of ζ are $0.8 \leq \zeta \leq 1.3$ so we can't play much with ζ in (37) but only with f_u . For example, to limit the degradation to $d_A = 3^\circ$ then, for $\zeta = 0.95$, using (37) we get $f_u > 114f_n$. Once f_u is determined, the phase margin is computed and the system is considered a second-order system with natural frequency f_n and a damping factor of $\zeta_{eff} = 0.01 \cdot PM$.

9. Implementation of the Decimation Filter

We shall now analyze the decimation filter and find a structure for its efficient implementation in hardware. The decimation filter $h_{DF}(n)$ is an FIR filter^[24] whose length we shall denote as L , and thus its output $y(k)$ as a function of its input $x(n)$ is:

$$y(k) = \sum_{m=0}^{L-1} h_{DF}(m) x(Mk - m) \quad (38)$$

The direct-form^[23 Chap. 6] realization of (38) often requires too many resources for it to be implemented in fixed-point hardware, because it has many multipliers and addition circuits that need to operate at rate f_p . Some improvements can be attained by using alternate topologies^[24 Figs. 6.28, 6.31], but even those structures often require excessive resources. To make efficient hardware implementation possible, consider the unit-gain rectangular window filter, that is:

$$h_{DF}(n) = \begin{cases} 1/L & 0 \leq n \leq L-1 \\ 0 & \text{otherwise} \end{cases} \quad (39)$$

which has the transfer function^{[6],[7]}:

Fig. 12 – Hybrid PLL model with decimation.

Fig. 13 – Hybrid PLL model, assuming ideal virtual DAC, ideal decimation filter, and ideal decimator.

Fig. 14 – Direct-form II implementation of $B(z)$.

Fig. 15 – PLL model, with ideal decimation but non-ideal DAC.

$$H_{DF}(e^{j\Omega}) = e^{-j\Omega \frac{L-1}{2}} \frac{1}{L} \cdot \frac{\sin(\Omega L / 2)}{\sin(\Omega / 2)} \quad (40)$$

which has a (one-sided) passband of $[0, 2\pi/L]$, so for a passband of $[0, \pi/M]$ we choose $L = 2M$, and from (40):

$$H_{DF}(e^{j\Omega}) = e^{-j\Omega \frac{2M-1}{2}} \frac{1}{2M} \cdot \frac{\sin(\Omega M)}{\sin(\Omega / 2)} \quad (41)$$

In $H_{DF}(e^{j\Omega})$ there is some distortion in the passband, and the sidelobes are relatively high. However, it can be shown^[6] that these effects cause an increase of at most 10.9% (≈ 0.5 dB) in the noise power inside the PLL, which for nearly all applications is acceptable. Moreover, dramatic savings in logic resources can be attained if (39) is implemented wisely, as we now show. Combining (38) and (39), and since $L = 2M$:

$$y(k) = (1/(2M)) \cdot \sum_{m=0}^{2M-1} x(Mk - m) \quad (42)$$

Now let's separate (42) into even and odd samples:

$$y(k) = \begin{cases} (1/(2M)) \cdot \sum_{m=0}^{2M-1} x(2Mr - m) & k=2r \\ (1/(2M)) \cdot \sum_{m=0}^{2M-1} x(2Mr - M - m) & k=2r-1 \end{cases} \quad (43)$$

which immediately suggests the implementation in Fig. 17. The division by $2M$ is prohibitive to compute in hardware;

Fig. 16 – Hybrid PLL model, assuming DAC incurs only phase distortion of the fundamental reconstructed signal.

Fig. 17 - Efficient hardware implementation of the decimation process (includes both the decimation filter and the decimator).

however if $2M$ is chosen to be a power of 2, then division by $2M$ can be approximated by discarding the lower $\log_2(2M)$ bits, which is a trivial operation.

A simplified diagram of the implementation of the Integrate and Dump (IAD) module is shown in Fig. 18. Note that the division by $2M$ is done within the IAD module (in order to reduce the number of bits required for the output register). Also, the output rate is $1/(2T_u)$, not $1/T_u$ since the IAD will be used in conjunction with a second IAD module (as in Fig. 17), which will produce an output sample at a rate of $1/(2T_u)$ Hz halfway between each pair of samples of the first IAD module. The “control logic” cloud tells the IAD module when to “integrate” and when to “dump”, and thus must work in tandem with the adjacent IAD module. Hence it is advantageous to implement the structure of Fig. 17 as a single module. The “control logic” cloud is essentially a carefully controlled counter that controls the timing of signals within the IAD modules, and is easy to implement in hardware.

In conclusion, we see that we can implement a nearly ideal decimation filter with only two adders (one per IAD), 4 registers (2 per IAD), and some simple control logic. This is a huge reduction in logic resources as compared to the direct-form implementation.

10. Effects of Implementation Latency

In Section 8 we found that the update rate of the DDS has a measurable impact on the PLL’s phase margin. This is actually a special case of the effect of a delay element within a control loop on the latter’s phase margin.

Assume we have designed the decimation filter as per (41). We can incorporate this filter into the model of Fig. 12 as shown in Fig. 19. The effects of the decimation filter magnitude response on the loop are negligible (see Sec. 9 and [6]) and thus neglected here. Regarding the delay introduced by the decimation filter, with transform $\exp(-j\Omega \cdot 0.5(2M-1))$, it must be taken into account. Taken at a rate of $f_p=1/T_p$ this delay in the Laplace domain is $\exp(-sT_p \cdot 0.5(2M-1))$. We thus have the PLL model of Fig. 20. We also insert into the open loop transfer function a pure delay $\exp(-sT_l)$ that models any implementation latencies totaling T_l seconds (e.g. delays associated with FPGA or ASIC data path, etc.). The revised model is in Fig. 21. The impact of the delays in Fig. 21 is best

Fig. 18 – Implementation of the Integrate and Dump module.

Fig. 19 - Hybrid PLL model with decimation filter response.

Fig. 20 – PLL with decimation filter latency converted to Laplace domain. Magnitude effects of the decimation filter are ignored.

Fig. 21 – Hybrid PLL model with implementation delay added.

Fig. 22 – How a pipeline stage in the datapath from the loop filter to the DDS inserts a delay of T_u .

analyzed as it diminishes the phase margin. Define the total implementation delay as:

$$T_d \triangleq T_l + 0.5(2M-1)T_p \quad (44)$$

Similarly to (31)-(37), it is easily shown that to bound the latency’s impact on the PM by d_L degrees, we must have:

$$T_d = T_l + 0.5(2M-1)T_p < d_L / (180 \cdot 2f_c) \approx d_L / (720 \cdot \zeta f_n) \quad (45)$$

The value of T_d has a strong connection to $T_u=1/f_u$. To see this, consider that $M=f_p/f_u$, and putting that into (44) yields:

$$T_d \triangleq T_l + (f_p / f_u) \cdot T_p - 0.5T_p = T_l + T_u - 0.5T_p \quad (46)$$

T_l is also a strong function of f_u . Consider for example a logic path that is part of the path to the DDS as shown in Fig. 22. Each such stage will add a delay of T_u to the open loop transfer function. Other delays that are not of this type may be present too and are also included in T_l . Hence to bound the total phase margin degradation caused both by the DDS and

the system delay (this total degradation is what determines ζ_{eff}), we must first find f_u that satisfies (37), and then, using this value of f_u , estimate T_l and T_d and check if (45) is satisfied. If not, then either f_u must be increased, d_L increased, or (if possible) f_n decreased. Note that often d_L and d_A are related and one can be determined from the other^{[6],[7]}.

11. Procedure for Designing a Hybrid PLL

We are now in a position to outline a step-by-step procedure for designing a hybrid PLL. The procedure is shown in Fig. 23. The step-by-step procedure for designing a hybrid PLL is shown in Fig. 23. The steps in Fig. 5 and Fig. 23 can be easily performed by a computer program, and the values can be directly incorporated into the design of the ASIC or FPGA. A schematic of the of the digital logic inside the FPGA or ASIC is shown in Fig. 24. Some explanation may be in regard regarding the addition of w_{CEN} to the output of the loop filter. This is because the PLL provides the *correction* to the DDS that is needed in order to maintain lock. The tuning word w_{CEN} is the tuning word that corresponds the center frequency of the DDS (= the center frequency of the VCO in the DDS equivalent model of Fig. 9). For a detailed example of the application of the design procedure to the design of an actual receiver, the reader is encouraged to look at [6 Chap. 9]. Finally, further analysis can be found in [6] and [7], which may be obtained by contacting the author.

References

- [1] J. G. Proakis, *Digital communications*, 4th ed. Boston: McGraw-Hill, 2001.
- [2] H. Meyr, M. Moeneclaey, and S. Fechtel, *Digital communication receivers: synchronization, channel estimation, and signal processing*. NY: Wiley, 1998.
- [3] U. Mengali and A. N. D'Andrea, *Synchronization techniques for digital receivers*. NY: Plenum Press, 1997.
- [4] F. M. Gardner, "Interpolation in digital modems. I. Fundamentals," *IEEE Trans. Commun.*, vol. 41, no. 3, pp. 501-507, Mar. 1993.
- [5] L. Erup, F. M. Gardner, and R. A. Harris, "Interpolation in digital modems. II. Implementation and performance," *IEEE Trans. Commun.*, vol. 41, no. 6, pp. 998-1008, Jun. 1993.
- [6] Y. Linn, "Synchronization and Receiver Structures in Digital Wireless Communications (workshop notes)," in *International Seminar: 15 Years of Electronic Engineering*. Bucaramanga, Colombia, Aug. 15-19, 2006.
- [7] Y. Linn, "A Tutorial on Hybrid PLL Design for Synchronization in Wireless Receivers," in *Proc. International Seminar: 15 Years of Electronic Engineering*, Bucaramanga, Colombia, Aug. 15-19, 2006 (invited paper).
- [8] A. Blanchard, *Phase-locked loops. Application to coherent receiver design*. NY: Wiley, 1976.
- [9] F. M. Gardner, *Phaselock techniques*, 2nd ed. NY: Wiley, 1979.
- [10] J. J. D'Azzo and C. H. Houpis, *Linear control system analysis and design: conventional and modern*, 3rd ed. NY: McGraw-Hill, 1988.
- [11] R. C. Dorf, *Modern control systems*, 5th ed. MA: Addison-Wesley, 1989.
- [12] R. E. Best, *Phase-locked loops: theory, design, and applications*, 2nd ed. NY: McGraw-Hill, 1993.
- [13] H. Meyr and G. Ascheid, *Synchronization in digital communications*. NY: Wiley, 1990.
- [14] R. L. Peterson, R. E. Ziemer, and D. E. Borth, *Introduction to spread-spectrum communications*. NJ: Prentice Hall, 1995.
- [15] W. P. Robins, *Phase noise in signal sources. (Theory and applications)*. London: Peter Peregrinus, 1982.

Fig. 23 – Step-by-step procedure for hybrid PLL design.

Fig. 24 – Implementation of the PLL's digital part

- [16] D. H. Wolaver, *Phase-locked loop circuit design*. NJ: Prentice Hall, 1991.
- [17] B. T. Kopp and W. P. Osborne, "Phase jitter in MPSK carrier tracking loops: analytical, simulation and laboratory results," *IEEE Trans. Commun.*, vol. 45, no. 11, pp. 1385-1388, Nov. 1997.
- [18] W. P. Osborne and B. T. Kopp, "An analysis of carrier phase jitter in an M-PSK receiver utilizing MAP estimation," in *Proc. MILCOM '93*, Boston, MA, USA, 1993, pp. 465-470.
- [19] W. P. Osborne and B. T. Kopp, "Synchronization in M-PSK modems," in *Proc. ICC '92*, Chicago, IL, USA, 1992, pp. 1436-1440.
- [20] Y. Linn, "A Robust Phase Detection Structure for M-PSK: Theoretical Derivations, Simulation Results, and System Identification Analysis," in *Proc. 18th Canadian Conference on Electrical and Computer Engineering (CCECE '05)*, Saskatoon, SK, Canada, May 1-4, 2005, pp. 869-883.
- [21] ETSI (European Telecommunications Standards Institute), "DVB-S2 Technical Report ETSI TR 102 376 V1.1.1," 2005.
- [22] Analog Devices, "AD9851 Datasheet, Rev. C," retrieved from www.analog.com.
- [23] A. V. Oppenheim and R. W. Schaffer, *Discrete-time signal processing*. NJ: Prentice Hall, 1989.
- [24] R. E. Crochiere and L. R. Rabiner, *Multirate digital signal processing*. NJ: Prentice-Hall, 1983.