Introduction To Phase-Lock Loop System Modeling

By Wen Li, Senior System Engineer, Advanced Analog Product Group
and Jason Meiners, Design Manager, Mixed-Signal Product Group,
Texas Instruments Incorporated

1. Introduction

Phase-lock loops (PLLs) have been one of the basic building blocks in modern electronic systems. They have been widely used in communications, multimedia and many other applications. The theory and mathematical models used to describe PLLs come in two types: linear and non-linear. Non-linear theory is often complicated and difficult to deal with in real-world designs. Analog PLLs have been well modeled by linear control theory. Starting from a well-defined model in continuous-time domain, this paper introduces a modeling and design method for a digital PLL based on that same linear control theory. It has been shown that a linear model is accurate enough for most electronic applications as long as certain conditions are met. Fig. 1 is a mixed-signal diagram of the Texas Instruments device THS8083, which targets LCD monitor and DTV applications. The task of PLLs inside these devices is to recover the pixel clock based on an input reference signal HS (horizontal sync.) This PLL has been accurately modeled by the method introduced here.

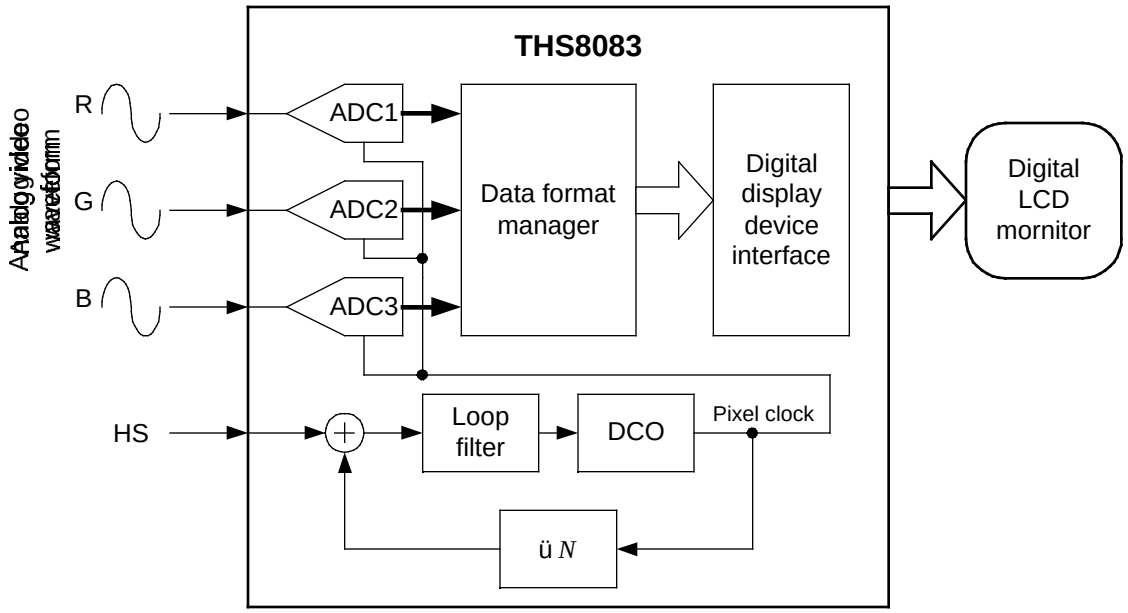


Fig. 1. A typical PLL application

Note: From a PLL system point of view, the DCO has the same function as a VCO but is implemented in the digital domain. The output frequency of the DCO is a function of the input digital value.

2. A linear PLL model in a continuous time domain (S-domain)

2.1. Block diagram of a typical PLL

Fig. 2 is a functional block diagram of a typical PLL system.

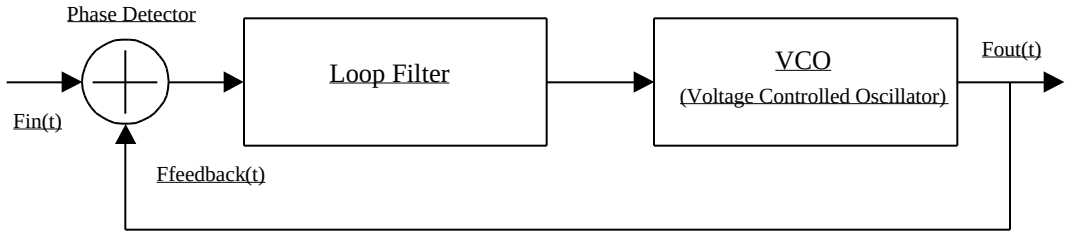


Fig.2. Functional block diagram of a typical PLL

From this diagram, the PLL can be easily recognized as a feedback control system. This system consists of following main components:

- **Phase Detector:** It detects the phase difference between the input signal $F_{in}(t)$ and the feedback signal $F_{feedback}(t)$,
- **Loop filter:** Typically, it is a filter with low-pass characterization,
- **VCO:** Voltage Controlled Oscillator whose output frequency oscillator is a function of the voltage of its input signal,

2.2. A linear model of the PLL in S-domain

Based on the condition that phase error is small, which can be expressed mathematically as $\text{Sin}(\theta) \approx \theta$, a PLL can be accurately described by a linear model. Fig. 2 is the block diagram of linear PLL model.

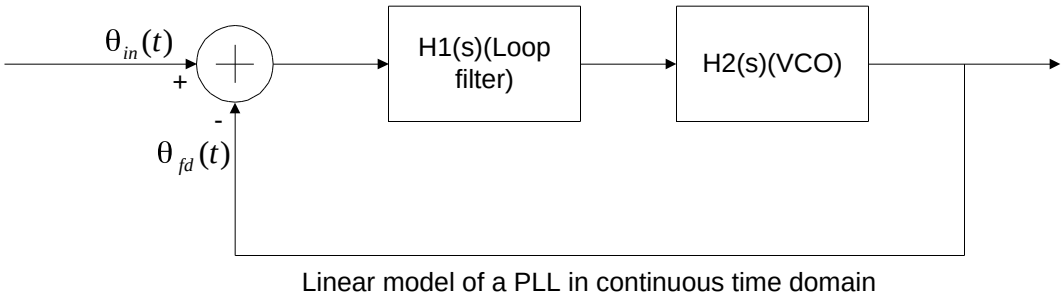


Fig. 3. A block diagram of the linear model of a PLL

where,
θin(t) is the phase of the input signal
θfd(t) is the phase of the feedback signal

Since the system is described in continuous time-domain, the transfer functions of each component are given out in Laplace-transform format.

· Transfer function of the loop filter:

$$H1(s) = \frac{G_{lp}}{G_{lp} + S} \qquad \text{(EQ001)}$$

· Transfer function of the voltage-controlled-oscillator:

$$H2(s) = \frac{G_{vco}}{S} \qquad \text{(EQ002)}$$

Closed-loop transfer function of PLL:

$$H_{cl}(s) = \frac{G_{lp}G_{vco}}{S^2 + G_{lp}S + G_{lp}G_{vco}} \qquad \text{(EQ003)}$$

Based on the close-loop transfer function (EQ003), one can easily tell that is a 2nd-order system. In automatic control system theory, the transfer function of the 2nd-order system can often be written in the following format:

$$H_s(S) = \frac{\omega_n^{\,2}}{S^2 + 2\zeta\omega_n S + \omega_n^{\,2}} \qquad \textbf{(EQ004)}$$

where, ω_n is defined as **natural undamped frequency**
ζ is defined as the **damping ratio,**

and this system is called as a standard prototype 2nd-order system.
Based on the transfer function of 2nd-order prototype system, a characteristic equation of the system is defined as:

$$\Delta(s) = S^2 + 2\zeta\omega_n S + \omega_n^{\,2} \qquad \text{(EQ005)}$$

By solving the roots of the characteristic equation, two poles S_0, S_1 of the system can be derived:

$$S_0 = -\zeta\omega_n + j\omega_n\sqrt{1-\zeta^2} = -\alpha + j\omega \qquad \text{(EQ006)}$$

$$S_1 = -\zeta\omega_n - j\omega_n\sqrt{1-\zeta^2} = -\alpha - j\omega \qquad \text{(EQ007)}$$

where, α is defined as the damping factor

ω is defined as damped frequency

Based on (EQ006) and (EQ007), as soon as ζ, ω_n of the system are given, the poles of a 2nd-order prototype system can be determined. Those two parameters are usually used to specify performance requirements of a system. As a matter of fact, most transient-response performances of a system can be determined based on these two parameters. The following are a list of performance parameters defined based on ζ, ω_n. Derivations of these equations can be found in most of control theory textbooks.[1]

Damping factor α: $\alpha = \zeta\omega_n$ (EQ008)

Damped Frequency ω: $\omega = \omega_n\sqrt{1-\zeta^2}$ (EQ009)

Settling time: $t_s = \frac{4}{\zeta\omega_n}$ (EQ010)

Maximum overshoot time: $t_{\max} = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}}$ (EQ011)

Maximum overshoot: $M = 1 + e^{-\pi\zeta/\sqrt{1-\zeta^2}}$ (EQ012)

Maximum overshoot in percentage: $M_{pct} = 100e^{-\pi\zeta/\sqrt{1-\zeta^2}}$ (EQ013)

Up to this point a 2nd-order system has been defined in S-domain, and this system will meet performance requirements specified by ζ, ω_n.

3. Modeling of a digital PLL in Z-domain

So far all the modeling done in previous sessions are in the continuous time domain. This model can directly be applied to an analog PLL, but the design requirement is to build a digital PLL (DPLL.) Normally the output responses of a discrete-time control system are also functions of continuous-time variable *t*. Therefore, the goal is to map the system that meets the time-response performance requirements specified by ζ, ω_n to a corresponding 2nd-order model in Z-domain.

3.1. A linear model of a PLL in discrete-time domain

First, a block diagram of the model of a DPLL is presented as Fig. 4.

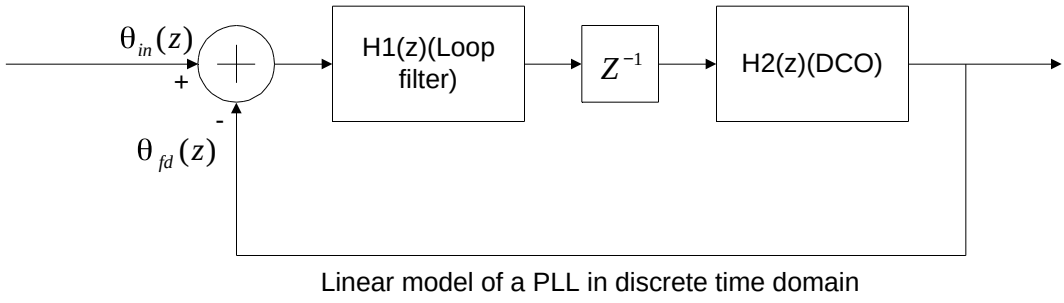


Fig. 4. A DPLL model in Z-domain

Transfer functions of each component in the DPLL are in Z-transfer format as follows:

· Transfer function of the loop filter:

$$H1(Z) = \frac{aZ - 1}{Z - 1} \qquad \text{(EQ014)}$$

· Transfer function of a digitally-controlled oscillator (DCO):

$$H2(Z) = \frac{cZ}{Z - 1} \qquad \text{(EQ015)}$$

· Z^{-1} is a delay unit, usually it is a register or register array

With the block diagram and the transfer functions of components in it, a linear time invariant (LTI) model can be developed to represent the DPLL with the closed-loop transfer derived as:

$$H(Z) = \frac{acZ - c}{Z^2 + (ac - 2)Z + (1 - c)} \qquad \textbf{(EQ016)}$$

3.2. Mapping the poles of a 2nd-order system from S-domain to Z-domain

The transfer function of a 2nd-order PLL in Z-domain can be written in a general format as (EQ017).

$$H(z) = \frac{N(z)}{(Z - Z_1)(Z - Z_0)} \qquad \text{(EQ017)}$$

where, Z_0, Z_1 are two poles of the system in Z-domain

Corresponding to the S-domain analysis, a characteristic equation of a discrete-time system is defined as:

$$\Delta(z) = (Z - Z_1)(Z - Z_0) = Z^2 - (Z_1 + Z_0)Z + Z_1 Z_0 \qquad \text{(EQ018)}$$

Defining C_1 and C_0 to be coefficients of the characteristic equation:

$$C_1 = -(Z_1 + Z_0)$$
$$C_0 = Z_1 Z_0 \qquad \text{(EQ019)}$$

Then, the characteristic equation can be re-written in a simplified format:

$$\Delta(z) = Z^2 + C_1 Z + C_0 \qquad \text{(EQ020)}$$

By definition of discrete-time transformation[2], two poles of this system in Z-domain can be mapped from the poles in S-domain in the following way:

$$Z_0 = e^{S_0 T_s} = e^{(-\zeta\omega_n T_s + j\omega_n T_s\sqrt{1-\zeta^2})}$$
$$Z_1 = e^{S_1 T_s} = e^{(-\zeta\omega_n T_s - j\omega_n T_s\sqrt{1-\zeta^2})} \qquad \text{(EQ021)}$$

where, T_s is the sampling period of the discrete system

With the poles mapped in Z-domain and (EQ019), coefficients C_0, C_1 of the characteristic equation (EQ020) can be derived in a format that is described by parameter ζ, ω_n

$$C_0 = e^{-2\zeta\omega_n T_s}$$
$$C_1 = -2e^{-\zeta\omega_n T_s} COS(\omega_n T_s\sqrt{1-\zeta^2}) \qquad \text{(EQ022)}$$

Now, a characteristic equation is derived by mapping the poles in a continuous-time domain system. Since the characteristic function will largely affect the transient responses of the system, (EQ020) and (EQ017) can determine the transfer function of a DPLL. The numerator of (EQ017) can be a constant scaling factor, or ZEROs can be introduced to tune performances of the system. For example, if the DPLL adopts the architecture-based (EQ016), its transfer function will be determined as soon as the poles are mapped.

3.3. Implementation of a 2nd-order DPLL (Digital Phase-Lock-Loop)

This section presents detailled information for implementing a completed DPLL system based on the previous analysis and model mapping results. First of all, an architecture diagram of a 2nd-order DPLL system is presented in Fig. 5.

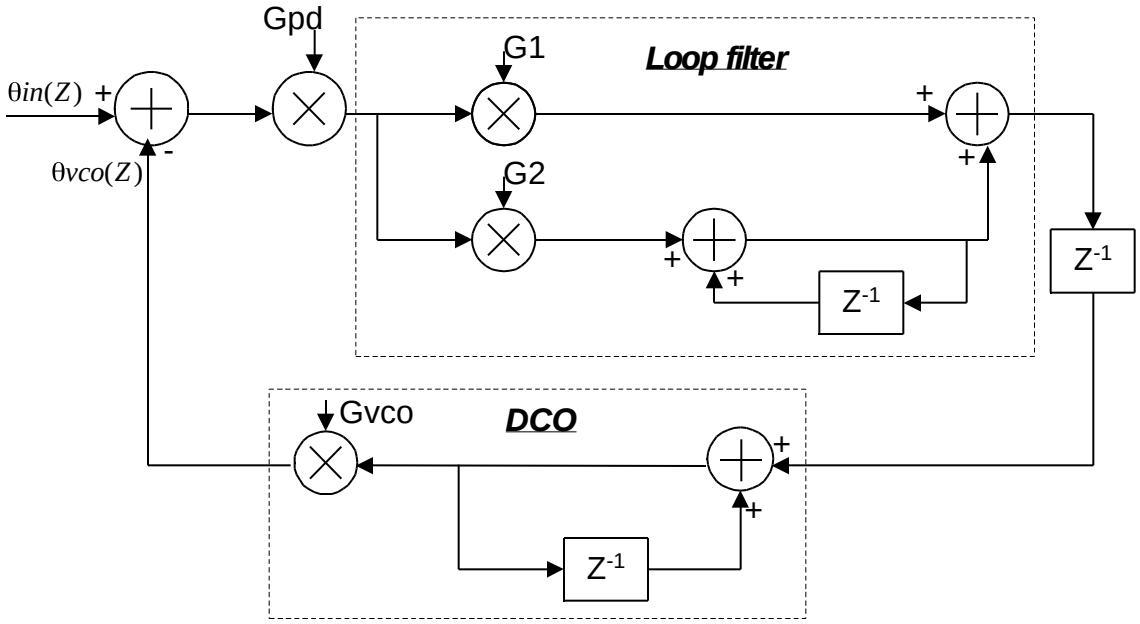


Fig. 5. Block diagram of a completely implemented 2nd-order DPLL system

Based on this architecture, each basic building block is described:

· Loop-filter, an IIR filter has been designed as the loop-filter, H1(z) is its transfer function

$$H1(z) = \frac{G_1 + G_2 - G_1 Z^{-1}}{1 - Z^{-1}} \qquad \text{(EQ023)}$$

where, G_1 and G_2 are the gains of the IIR filter

· A digitally-controlled VCO, or a discrete-time-oscillator, will have H2(z) as its transfer function

$$H2(z) = \frac{G_{vco}}{1 - Z^{-1}} \qquad \text{(EQ024)}$$

where, G_{vco} is the GAIN of the discrete voltage-controlled-oscillator

With these building blocks of the DPLL system, its closed-loop transfer function can be written as:

$$H(z) = \frac{\theta_{VCO}(z)}{\theta_{in}(z)} = \frac{H1(z)H2(z)Z^{-1}G_{pd}}{1 + H1(z)H2(z)Z^{-1}G_{pd}} \qquad \text{(EQ025)}$$

where, G_{pd} is the GAIN of the phase detector

The format of this transfer function can be rewritten as:

$$H(z) = \frac{\theta vco(z)}{\theta_{in}(z)} = \frac{(g1+g2)Z - g1}{Z^2 + (g1+g2-2)Z + (1-g1)} \qquad \textbf{(EQ026)}$$

where, $g1=G_{pd}G_{vco}G_1$
$g2=G_{pd}G_{vco}G_2$

By comparing the characteristic equation $\Delta(z)$ of a DPLL (EQ020) the following equation can be constructed:

$$C_0 = 1 - g1$$
$$C_1 = g1 + g2 - 2 \qquad \text{(EQ027)}$$

And g1 and g2 can be resolved based on (EQ027) and (EQ022):

$$g1 = 1 - e^{-2\zeta\omega_n T_s}$$
$$g2 = 1 + e^{-2\zeta\omega_n T_s} - 2e^{-\zeta\omega_n T_s} COS(\omega_n T_s \sqrt{1-\zeta^2}) \qquad \textbf{(EQ028)}$$

With (EQ026) and (EQ028), the model of a DPLL is completely derived.

4. Stability and steady-state error study of the DPLL system

4.1. Stability of the DPLL system

One mandatory requirement for designing DPLLs is that the DPLL system must be designed to be **stable**. Basically, the stable condition of a discrete-time system is such that the roots of the characteristic equation should be inside the unit circle, |z| = 1, in the z-plane. Normally, after a system is implemented, numerical coefficients can be substituted into the characteristic equation. By solving the characteristic equation numerically, the positions of the poles can be found to determine if the system is stable. However this method is difficult to use to guide the implementation of a DPLL, since numerical coefficients will not be available at the beginning of the process.

One of the most efficient criteria for testing the stability of a discrete-time system is **Jury's stability criterion**[1]. This can guide designs of a DPLL to converge to an optimized stable system quickly, without large amounts of numerical calculation and simulation. It can be applied to the second-order DPLL model directly to determine the stable condition and according to this criterion the necessary and sufficient conditions are that the characteristic equation of a second order system:

$$\Delta(Z) = a_2 Z^2 + a_1 Z + a_0 = 0 \qquad \text{(EQ029)}$$

should meet following conditions in order to have no roots on, or outside, the unit circle:

$\Delta(1) > 0$
$\Delta(-1) > 0$
$|a_0| < a_2$

Applying these conditions to the denominator of (EQ026) the stable condition ranges of this DPLL architecture are:

$0 < g1 < 2$ (EQ030)
$0 < g2 < 4$ (EQ031)

4.2. Steady-state error analysis of the DPLL

A steady-state error analysis of a DPLL is extremely important in the PLL design. The last paragraph describes the stable conditions of DPLL system. Here the steady-state error of phase and frequency of the DPLL will be studied. We will prove that both phase and frequency error of this DPLL system will be zero when the system reaches steady-state.

4.2.1. Phase error analysis

Assuming that the phase of the input signal has a step change this can be described by the step function in the time domain:

$$\Theta in(t) = \Delta\Theta \quad u(t) \qquad \text{(EQ32)}$$

Here $\Delta\Theta$ is the constant that the phase of input signal jumped. Applying the Z transform to EQ032:

$$\Theta in(Z) = \frac{\Delta\Theta \quad Z}{Z-1} \qquad \text{(EQ033)}$$

Based on the linear model presented in 3.1, the output-response function of the DPLL for phase step input can be written as:

$$\Theta fd(z) = H(Z) \quad \Theta in(Z) = \frac{\Delta\Theta \quad Z(acZ-c)}{(Z-1)(Z^2+(ac-2)Z+(1-c))} \qquad \text{(EQ034)}$$

Based on EQ034, by using existing software tool such as MATLAB, a numerical analysis can be carried out. In this way the steady-state error of an implemented DPLL system can be observed. Here we are focusing on the general analytical results.

Assuming E(Z) is the phase-error function, by definition, E(Z) can be written as follows:

$$E(Z) = \Theta in(Z) - \Theta fd(Z) \qquad \text{(EQ035)}$$

Substituting EQ034 into EQ035:

$$E(Z) = [1 - H(Z)]\Theta in(Z) \qquad \text{(EQ036)}$$

Substituting EQ033 and EQ016 into EQ036, the phase-error function can be written as:

$$E(Z) = \frac{\Delta\Theta Z(Z-1)}{Z^2 + (ac-2)Z + (1-c)} \qquad \text{(EQ037)}$$

According to the Final-Value Theorem,

$$\lim_{k \to \infty} e(kT) = \lim_{z \to 1}(1 - Z^{-1})E(Z) \qquad \text{(EQ038)}$$

Based on this theorem, the steady-state error, which is the final value of e(kT) in time domain, can be derived. The condition to use the Final-value Theorem is that the function $(1 - Z^{-1})E(Z)$ has no poles on or outside the unit circle, $|Z| = 1$, in the z-plane. The detail for meeting this condition was shown in 4.1.

By substituting EQ037 into EQ038:

$$\lim_{k \to \infty} e(kT) = \lim_{z \to 1} \frac{\Delta\Theta Z(Z-1)}{Z^2 + (ac-2)Z + (1-c)} = 0 \qquad \text{(EQ039)}$$

Conclusion: when the phase of the input signal had step-jumping, the phase error of this DPLL will eventually be eliminated by the closed-loop system.

4.2.2. Frequency-error analysis

For an input signal, assuming t = 0, and its frequency jumps from ω0 to ω1, let Δω = ω1 - ω0. Therefore the input phase can be written as follows:

$$\Theta in(t) = \Delta\omega \quad t \quad U(t) \qquad \text{(EQ040)}$$

Apply a Z-transform to EQ040 to transfer it to Z-domain:

$$\Theta in(Z) = \frac{\Delta\omega TZ}{(Z-1)^2} \qquad \text{(EQ040)}$$

Substituting (EQ040) and (EQ016) into EQ036, the frequency-error function is derived as:

$$E(Z) = \frac{\Delta\omega TZ}{Z^2 + (ac-2)Z + (1-c)} \qquad \text{(EQ041)}$$

Applying the Final-Value Theorem to EQ041 to get the steady-error in time domain:

$$\lim_{k\to\infty} e(kT) = \lim_{z\to 1}(1-Z^{-1})E(Z) = \lim_{z\to 1}\frac{\Delta\omega T(Z-1)}{Z^2 + (ac-2)Z + (1-c)} = 0 \qquad \text{(EQ042)}$$

Conclusion: when the frequency of input signal has a step jump, the phase error of this DPLL will eventually be eliminated by the closed-loop system.

5. A design example

Here we give a real design example and the simulation/measuring results of the system.

Design requirements:

- To design a DPLL that can recover the pixel clock of a PC's VGA output graphics signals.
- The frequency of horizontal synchronization signal HS of VGA is f_s = 60023 Hz, T_s = 0.00001666s.
- The relationship between a period of pixel clock T_p and a period of horizontal sync T_s is: $T_s = 1312T_p$.
- PLL locking time to be < 15ms.
- One overshoot permitted during locking process.

Based on these requirements the following performance parameters can be determined:

$\zeta = 0.707$
$\omega_n = 2\pi 100$ rad/s
f_s = 60023 Hz, T_s = 0.00001666s

Based on these parameters, C_0, C_1, g1, and g2 can be calculated by (EQ022) and (EQ028):

$C_0 = 0.9853$
$C_1 = -1.9852$
g1 = 0.0147
g2 = 0.0001

The transfer function of the DPLL that meeting the performance specification:

$$H(z) = \frac{0.0148Z - 0.0147}{Z^2 - 1.9852Z + 0.9853} \qquad \text{(EQ042)}$$

Based on this Z-domain model, the DPLL system performance can be simulated at system level. The following two diagrams are simulation results based on this model:

1. The input response of the model describes the behavior of the system when the input signal phase is a step function. It also proves that this is a stable system.

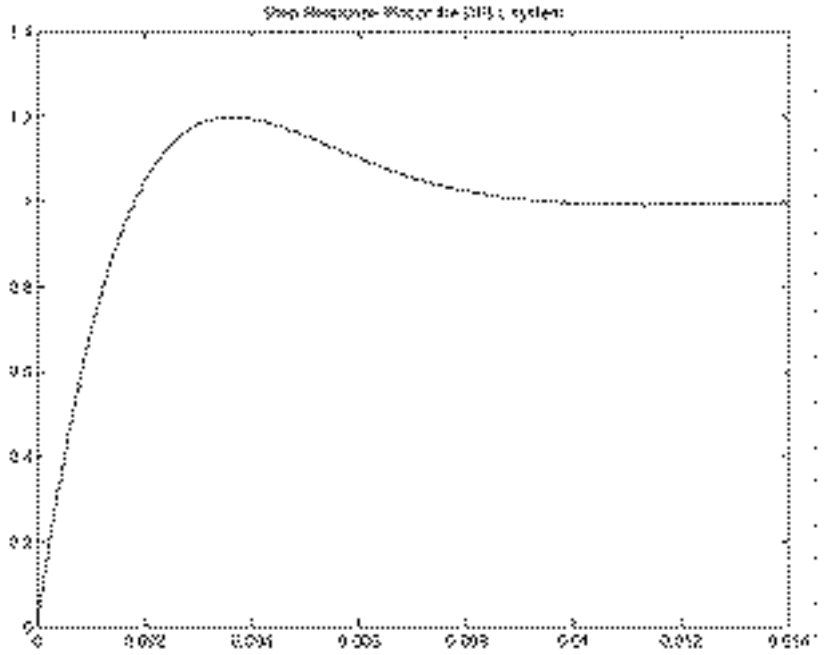


Fig. 6. Step Response of the DPLL system

2. The input response of the model describes the behavior of the system when the input signal has a phase impulse error. It proves that the stable error of the system is zero.

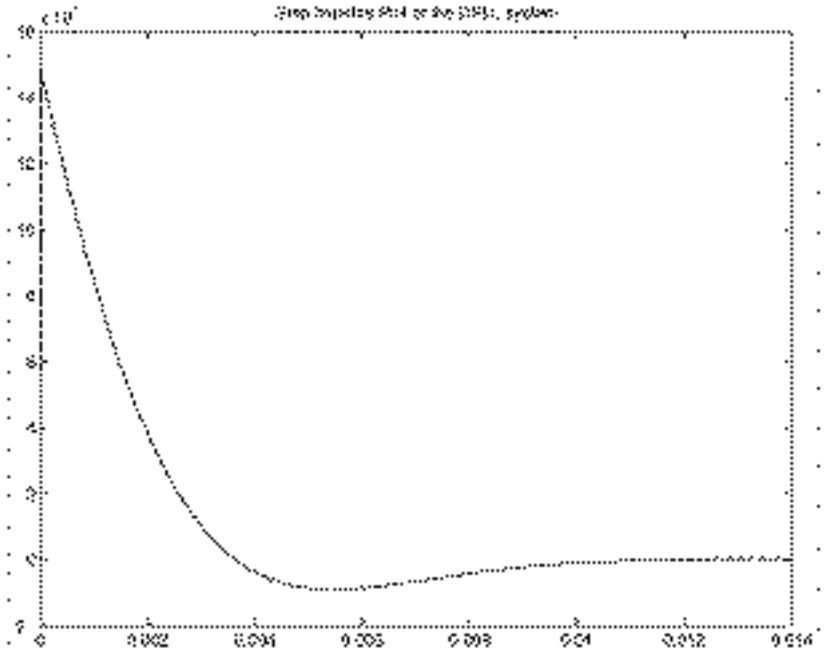


Fig. 7. Impulse input response of the DPLL system

3. Silicon-implemented DPLL based on (EQ032) model. It shows gate-level simulation/measuring results for a phase locking process.

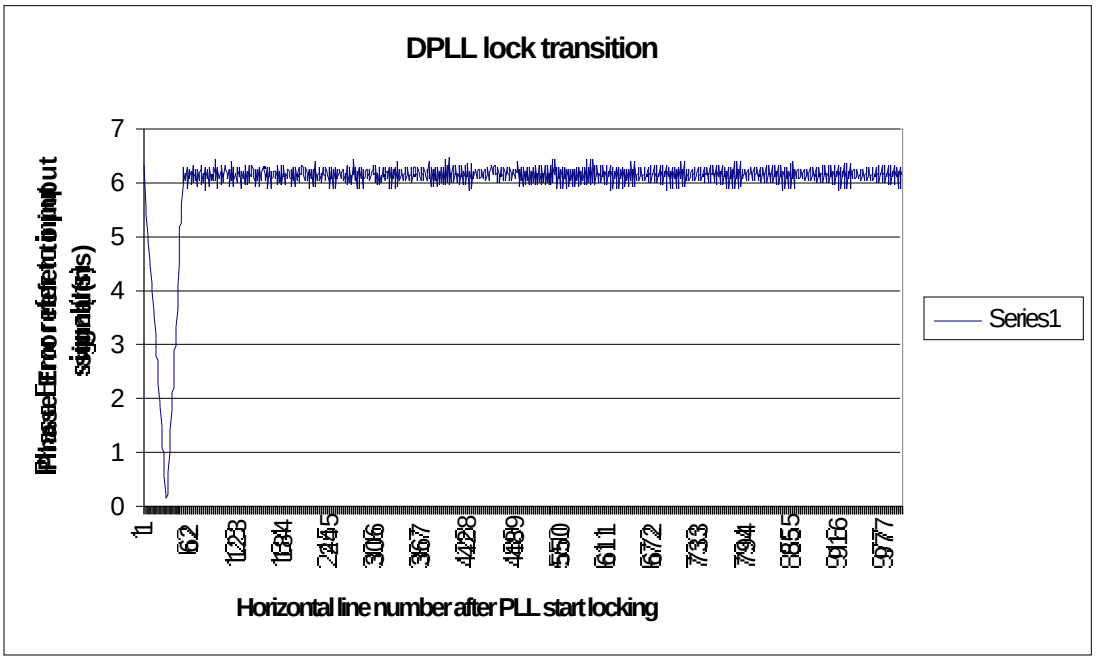


Fig. 8. DPLL lock process based on a silicon-implemented DPLL

Physically, this DPLL is implemented in the following way:

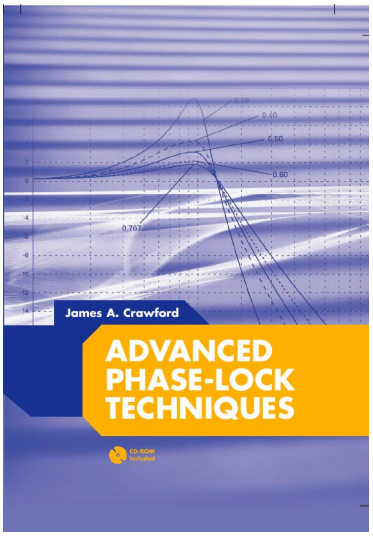
· **Phase detector:** A high-speed counter to sample input signal, calculate phase error

· **Loop-filter:** A digital IIR filter

· **DCO:** A DDS (Direct-Digital-Synthesis) oscillator. From a PLL system point of view, the DCO has the same function as VCO, but it is implemented in the digital domain so that the output frequency of the DCO is a function of the input digital value.

6. References

[1] Automatic control systems; Benjamin C. Kuo
[2] Discrete-time signal processing; Alan V. Oppenheim & Ronald W. Schafer
[3] Phase-Locked Loops, Theory and Applications; John L. Stensby



Advanced Phase-Lock Techniques

James A. Crawford

2008

Artech House

510 pages, 480 figures, 1200 equations
CD-ROM with all MATLAB scripts

ISBN-13: 978-1-59693-140-4
ISBN-10: 1-59693-140-X

Chapter	Brief Description	*Pages*
1	*Phase-Locked Systems—A High-Level Perspective* An expansive, multi-disciplined view of the PLL, its history, and its wide application.	26
2	*Design Notes* A compilation of design notes and formulas that are developed in details separately in the text. Includes an exhaustive list of closed-form results for the classic type-2 PLL, many of which have not been published before.	44
3	*Fundamental Limits* A detailed discussion of the many fundamental limits that PLL designers may have to be attentive to or else never achieve their lofty performance objectives, e.g., Paley-Wiener Criterion, Poisson Sum, Time-Bandwidth Product.	38
4	*Noise in PLL-Based Systems* An extensive look at noise, its sources, and its modeling in PLL systems. Includes special attention to 1/*f* noise, and the creation of custom noise sources that exhibit specific power spectral densities.	66
5	*System Performance* A detailed look at phase noise and clock-jitter, and their effects on system performance. Attention given to transmitters, receivers, and specific signaling waveforms like OFDM, M-QAM, M-PSK. Relationships between EVM and image suppression are presented for the first time. The effect of phase noise on channel capacity and channel cutoff rate are also developed.	48
6	*Fundamental Concepts for Continuous-Time Systems* A thorough examination of the classical continuous-time PLL up through 4^{th}-order. The powerful Haggai constant phase-margin architecture is presented along with the type-3 PLL. Pseudo-continuous PLL systems (the most common PLL type in use today) are examined rigorously. Transient response calculation methods, 9 in total, are discussed in detail.	71
7	*Fundamental Concepts for Sampled-Data Control Systems* A thorough discussion of sampling effects in continuous-time systems is developed in terms of the *z*-transform, and closed-form results given through 4^{th}-order.	32
8	*Fractional-N Frequency Synthesizers* A historic look at the fractional-N frequency synthesis method based on the U.S. patent record is first presented, followed by a thorough treatment of the concept based on Δ-Σ methods.	54
9	*Oscillators* An exhaustive look at oscillator fundamentals, configurations, and their use in PLL systems.	62
10	*Clock and Data Recovery* Bit synchronization and clock recovery are developed in rigorous terms and compared to the theoretical performance attainable as dictated by the Cramer-Rao bound.	52