Intuitive Guide to Principles of Communications
www.complextoreal.com

Unlocking the Phase Lock Loop - Part 1

The first Phase Lock Loop (PLL) were proposed by French scientist de Bellescize in 1932 who is also credited with being the inventor of coherent demodulation. Phase-locked loops have many different applications and come to communications systems from the heritage of control and vibration theory where they are used to describe free-body behavior of mechanical systems. In communication systems PLL are used for

1. Carrier synchronization
2. Carrier recovery
3. Frequency division and multiplication
4. Demodulation

The first PLLs were analog but since the 70's integrated circuits have been available to perform the same functions on a chip. These are called digital PLLs. There are basically three classes of PLLs.

1. The linear or analog PLL (LPLL)
2. The digital PLL (DPLL)
3. All-digital PLL (ADPLL)

The conceptual basis behind each of these is the same and they are all specified by the same standard parameters such as loop bandwidth, damping factor etc. and we will look at what all these mean.

A PLL has three core components. They are

1. Phase Detector (PD) or the multiplier
2. The Loop filter (LF)
3. Voltage controlled Oscillator or VCO

The Phase Detector

In simple terms, the phase detector is a multiplier. A gain value is also associated with it which we will call, K_m.

Let's start with two sinusoids, $s_1(t)$ and $s_2(t)$. Both have same frequency but are phase shifted by 90^0. Now multiple these two signals.

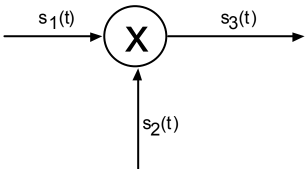


Figure 1 – Detecting the phase by multiplying two sinusoids

We have set the phase of these signals as a variable. Note that s_2 is a cosine hence is 90^0 shifted from s_1.

$$s_3(t) = s_1(t)\,s_2(t)$$

$$s_1(t) = A_1 \sin[wt + \phi_1(t)]$$
$$s_2(t) = A_2 \cos[wt + \phi_2(t)]$$

The output of the multiplier is

$$s_3(t) = K_m A_1 A_2 \sin[wt + \phi_1(t)]\cos[wt + \phi_2(t)]$$

where K_d is the gain of the multiplier. With a little trigonometric manipulation, we can put this equation in a form which is far more illuminating.

$$s_3(t) = \frac{K_d A_1 A_2}{2}\sin[\phi_1(t) - \phi_2(t)] + \frac{K_d A_1 A_2}{2}\underline{\sin[2\omega t + \phi_1(t) + \phi_2(t)]} \qquad (1)$$

In this form, we see that the multiplier signal consists of two parts, the first one (in blue) is function of only the phase difference of the two signals, and the second (underlined) term is at a frequency which is twice the signal frequency (note the 2wt term) plus the sum of the two phases.

We can use this equation to develop the PLL by recognizing that the output signal of the multiplier is *a function of the phase difference of the two input signals*. This is useful information and we can use it to synchronize the two signals. The second part of Eq. (1) at twice the frequency can be discarded by filtering it out since it does not offer anything we need.

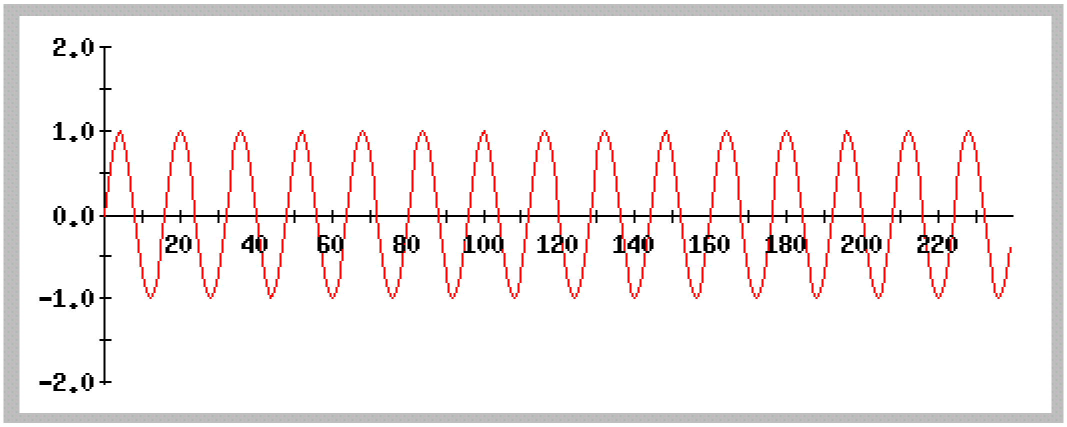


Figure 2 - Signal s_1 or the forcing signal

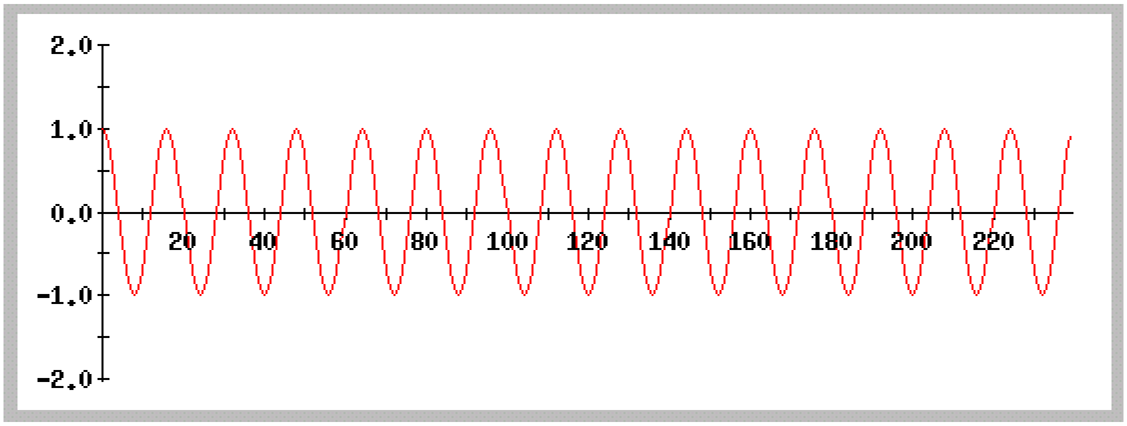


Figure 3 - Signal s2, note 90 degree shift

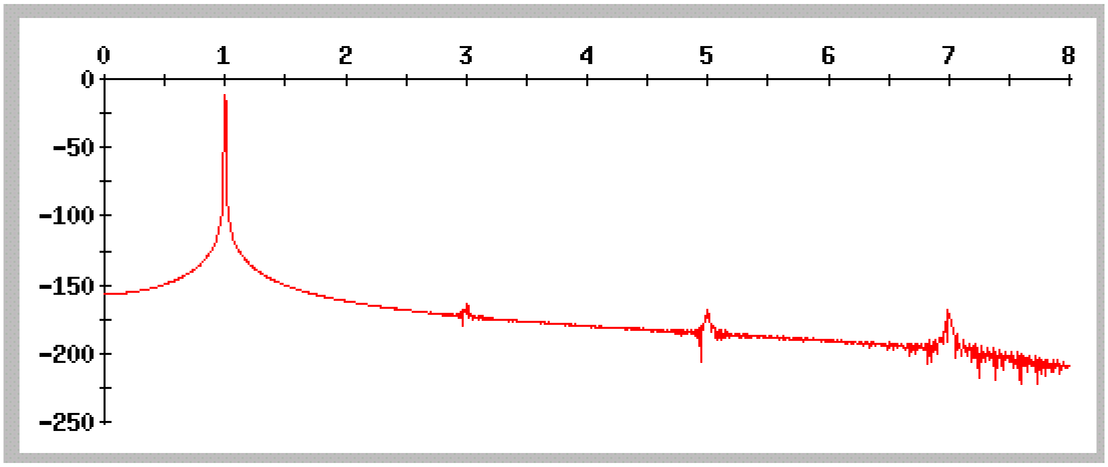


Figure 4 – FFT of input signal. It has just one peak at the signal frequency of 1 Hz.

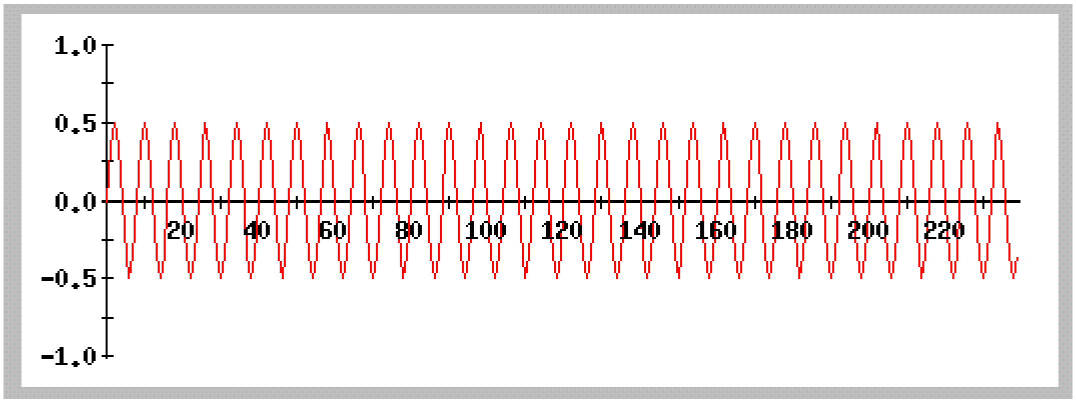


Figure 5 – Signal s_3, the signal out of the multiplier. Note that its average amplitude is 0 and it seems to be of higher frequency than the original signals s_1 and s_2.

The FFT of the multiplier signal s_3 consists of two pulses, one at dc since the phase difference is not a function of the frequency and the second at twice the signal frequency as we can see in Figure 6.

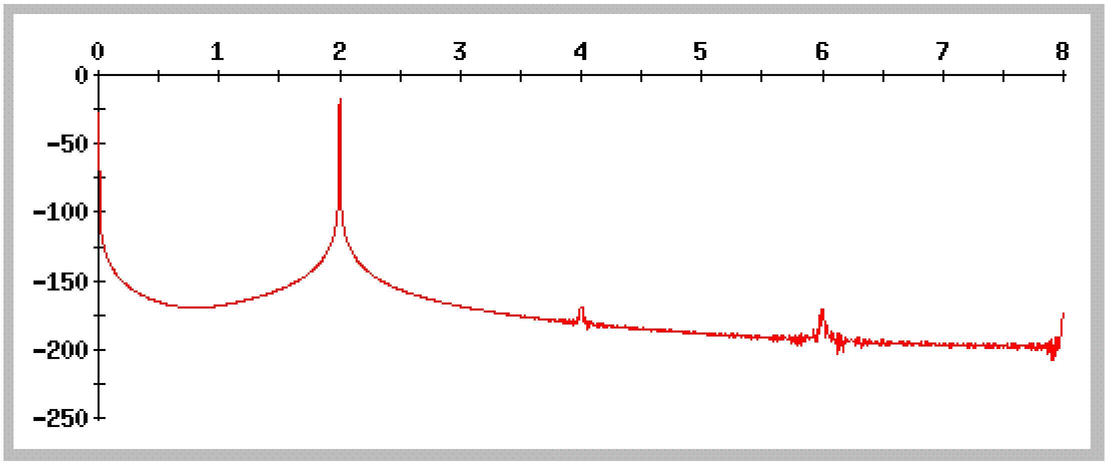


Figure 6 – FFT of the signal out of the multiplier has two peaks, one at dc which is the phase difference between the two signals and the unwanted term at twice frequency.

Loop filter - Getting rid of the unwanted term by filtering

Now add a low pass filter at the output of the multiplier signal. Its bandwidth should be quite small so it both knocks out noise and the unwanted double-frequency term.

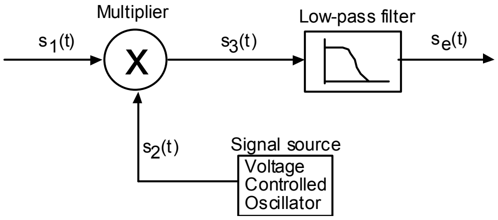


Figure 7 – Filter the multiplier signal to get rid of the double-frequency term. The filter removes these terms which have no useful information.

The following figures show the output of the LPF, as the phase difference is varied. When there is no phase difference (we are starting with 90 phase difference which is required to make this whole thing work.), then the signal out of the LPF is just the first part of Eq. (1). We call this part the error signal (also called the control signal.)

$$s_e(t) = \frac{k_d A_1 A_2}{2} \sin[\phi_1(t) - \phi_2(t)] \qquad (2)$$

If phase difference is 0 degrees (actually 90) then we would expect the signal to be zero, which is the desired and the locked-state of the PLL. If the phase between the two signals (s1 and s2) varies from that, then, we would expect the filtered s_3 signal to change as we see in the figures below.

The steady-state value of the signals below comes from Eq (2). In the examples below, A1, A2 and K_m are all set to 1.0., so at a phase difference of $\pi / 2$, we would expect the signal amplitude to be .5 and this is exactly what you will in Figure 13.

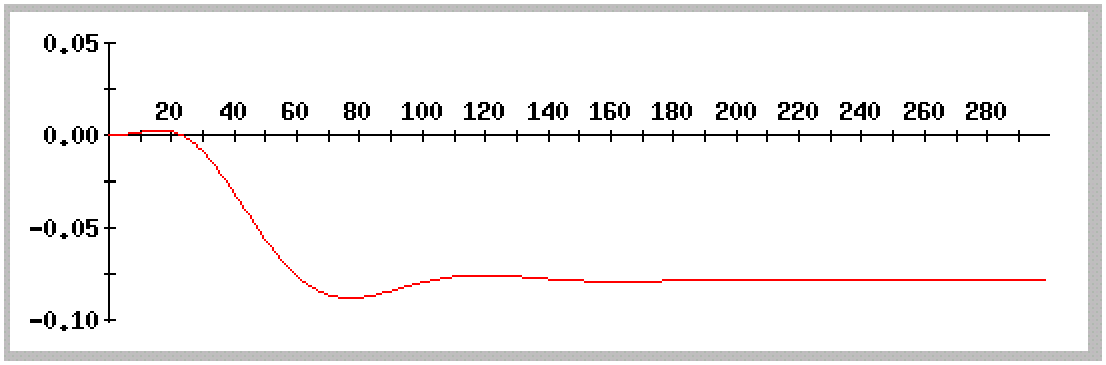


Figure 8 - Delta phase = 10 deg, Steady-state value = -.078

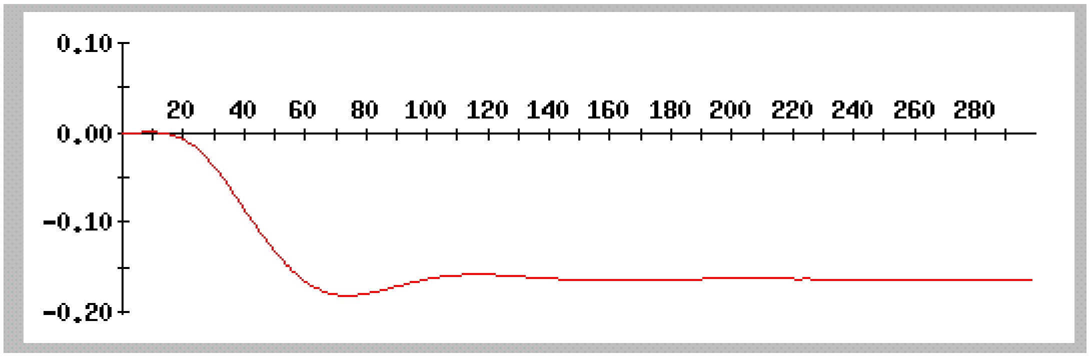


Figure 9 - Delta phase = 20 deg, Steady-state value = -.163

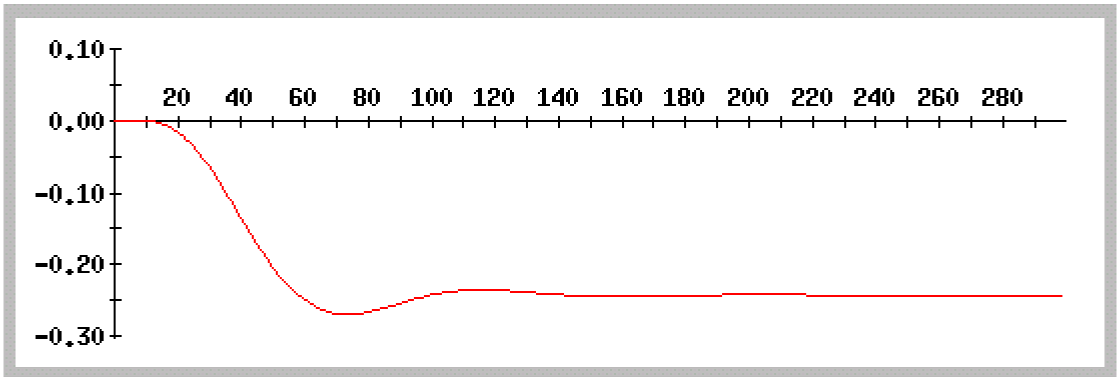


Figure 10 - Delta phase = 30 deg, Steady-state value = -.243

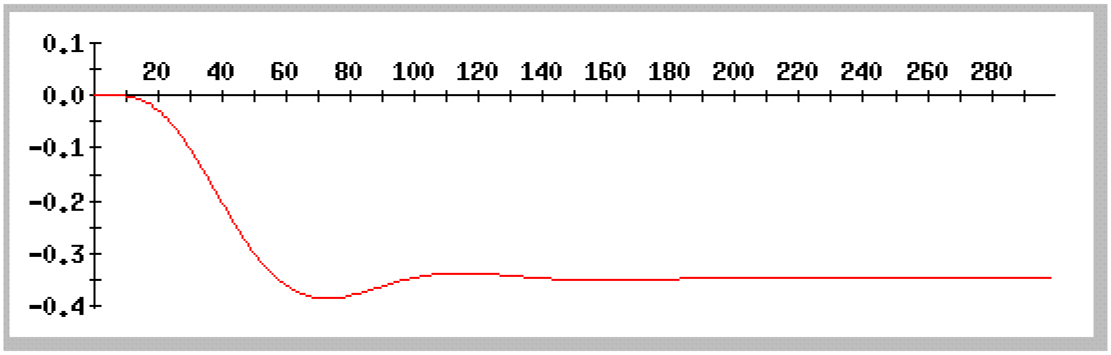


Figure 11 - Delta phase = 45 deg, Steady-state value = -.348

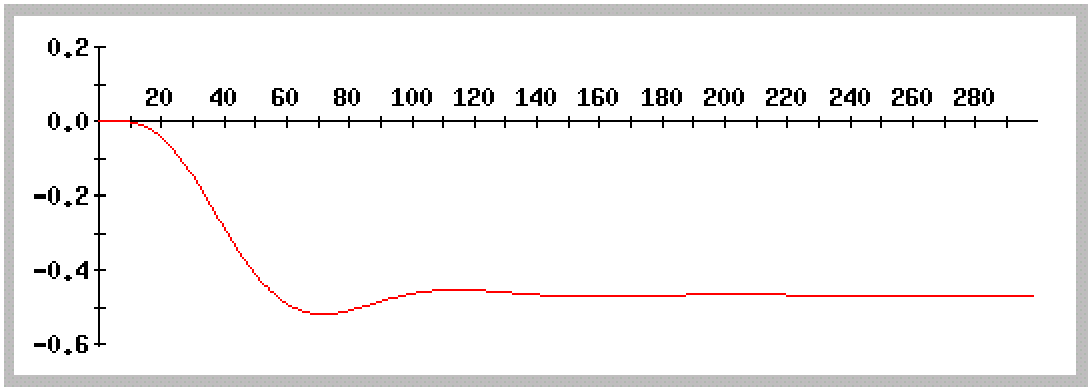


Figure 12 - Delta phase = 70 deg, Steady-state value = -.467

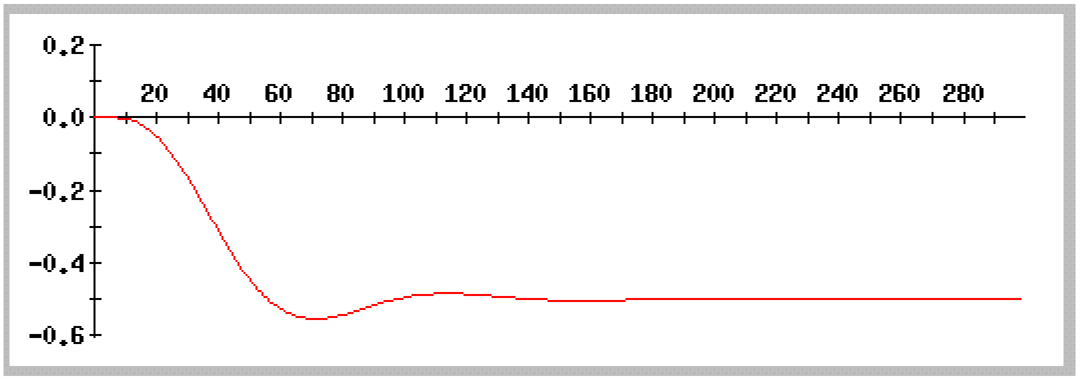


Figure 13 - Delta phase = 90 deg, Steady-state value = -.500

Plotting the steady-state dc value in the figures above against the delta phase difference over the whole 360 degrees, we get the following picture. When the phase difference is 0, 180 and 360 degrees, the error signal amplitude is 0. The amplitude is maximum at phase difference of 90 and 270 degrees. The response is not linear and has a sinusoidal shape. This exactly what was predicted by Eq. (2).

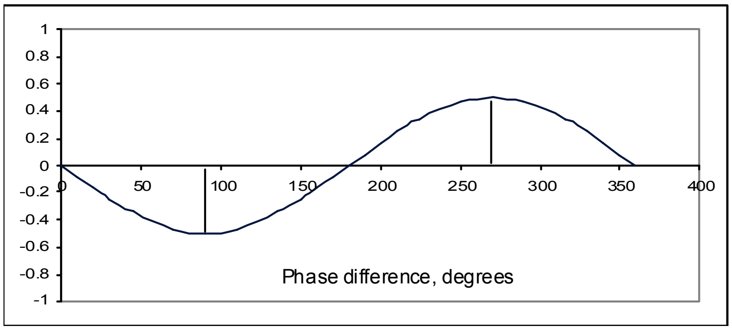


Figure 14 – The error signal amplitude vs. the phase difference detected

As we can see the amplitude of the error signal is directly related the phase error. If there is a sudden phase shift of say 45 degrees at the input then the error signal will go from 0 amplitude to .248 volts over a certain period of time.

Now let's see what the PLL does with this error signal and how it synchs up with the incoming signal.

Voltage-controlled Oscillator – a dynamically changing frequency response

Now we bring in the last player on the stage. The error signal provides us an indication of what is happening to the input phase. We want the error signal to have zero amplitude and we can do that only by changing the phase of signal s_2 to match the phase of signal s_1. VCO which we use to produce the signal allows us to do that.

As its name explains, VCO produces a periodic signal, the frequency of which changes based on a control signal applied externally. If the error signal is zero then, the VCO produces just its quiescent frequency (center frequency). But if the error signal is something other than zero, then it responds by changing its operating frequency.

A constant of K_0 represents the sensitivity of the VCO. It represents the change in the instantaneous frequency of the VCO as a function of the error signal amplitude such that

$$K_O = \frac{d\omega_i}{dv}$$

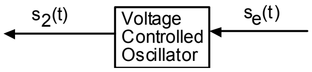
The signal out of the VCO is given by

$$s_2(t) = A_2 \cos(\omega_c t + \phi_2(t))$$

The units of K_0 are Hertz per volts. Given a certain input voltage, it will produce a change in the output signal frequency by the following relationship.

$$\omega_{out} = \omega_c + K_o v(t)$$

where ω_c is its center or operating frequency. So if K = 5000 Hz/volt, then a input of .1 volt would produce a new output frequency of ω_c + 500 Hz.



We know that for a periodic signal p(t), its frequency in Hz is equal to the rate of change of phase in 2π segments, or

$$f_t(t) = \frac{1}{2\pi}\frac{d\phi_i(t)}{dt}$$

and conversely, phase is the integral of the frequency over a certain period of time.

$$\phi_i(t) = 2\pi\int_0^t f_i(t)dt \qquad (3)$$

These relationships apply to all periodic signals, even those that are non-sinusoidal. These two ways of writing the argument of the cosine are equivalent.

$$\begin{aligned} p(t) &= \cos(2\pi ft) \\ &= \cos(2\pi\int_0^t f(t)dt) \end{aligned}$$

We can now write the phase of the feed-back signal as

$$\begin{aligned} \phi_2(t) &= 2\pi K_o\int_0^t s_e(t)dt \\ &= 2\pi K_O s_e(t)t \end{aligned} \qquad (4)$$

So as long as the error signal has a non-zero amplitude, the phase of the VCO signal will keep on increasing until such time as it is decreased to zero. Substitute Eq (4) into Eq (2) to get,

$$\begin{aligned} s_e(t) &= \frac{k_m A_1 A_2}{2}\sin[\phi_1(t) - \phi_2(t)] \\ s_e(t) &= \frac{k_m A_1 A_2}{2}\sin\left[\phi_1(t) - 2\pi K_O\int_0^t s_e(t)\right] \end{aligned} \qquad (5)$$

The equation of s_e can be linearlized by making the following assumption.

$$\begin{aligned} &\sin(\theta) \approx \theta \quad \text{for small } \theta \\ &\sin[\phi_1(t) - \phi_2(t)] \approx [\phi_1(t) - \phi_2(t)] \end{aligned}$$

Now we can rewrite Eq. (5) by removing the sine function.

$$\begin{aligned} s_e(t) &= \frac{k_m A_1 A_2}{2}[\phi_1(t) - \phi_2(t)] \\ s_e(t) &= \frac{k_m A_1 A_2}{2}\left[\phi_1(t) - 2\pi K_O\int_0^t s_e(t)\right] \\ &= \frac{k_m A_1 A_2}{2}[\phi_1(t) - 2\pi K_O S_e t] \end{aligned} \qquad (6)$$

In the above equation Se is the amplitude of $s_e(t)$ at time t.

Let's say that the input signal changes by 10 degrees. This causes the error signal to slowly increase in amplitude from 0 to .1. At time t, the frequency of the signal produced by the VCO increases by K_0Se, where Se is the instantaneous amplitude of the error signal and time T is sampling time.

As long as the error signal is present, the phase keeps changing linearly. However, as the phase of the signal out of the VCO changes, the new difference in phase decreases and the error signal amplitude decreases at the next go-around. This decreases the phase change further until the error signal amplitude has gone to zero. This in a nut-shell is how a PLL works.

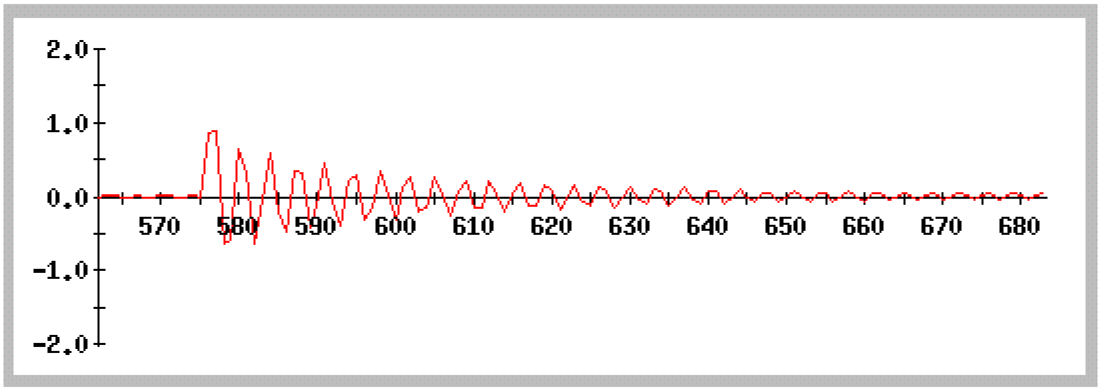


Figure 15 – Error signal

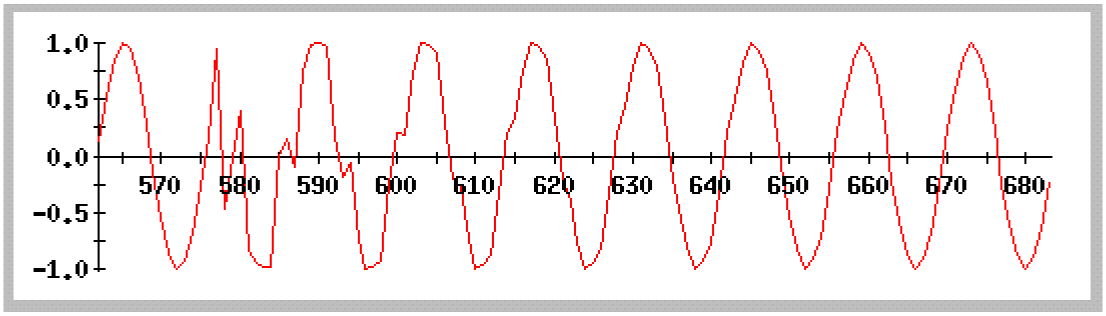


Figure 16 - Resulting signal out of the VCO

Figure 15 shows a signal with fluctuating error signal. At every large error signal amplitude, there is a large change in the phase of the VCO signal, these changes continue to decrease in response to the error signal amplitude until the signal has smoothed out and synchronized with the incoming signal.

Now lets' take a look at the total picture.

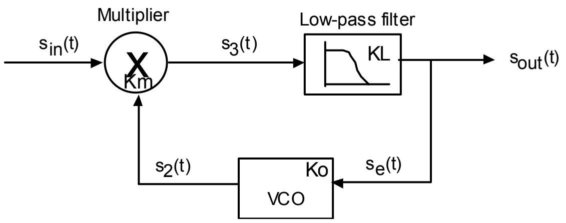


Figure 17 – PLL with its three components

This may not be obvious but the PLL is kind of an adaptive filter. The algorithms and parameters used in PLL analysis are similar to those used for filters, such Bode plot, 3-dB bandwidth and poles and roots. However, there is one big difference; the PLL has a fairly large gain. The gain of the loop is very easily computed. It is just the product of all the component gains.

$$LoopGain = K_d K_{LPF} K_O$$

K_d – the gain of the phase detector (also multiplier)
K_L – the gain of the low pass filter
K_O – the gain of the VCO

Transfer function of a PLL

PLLs are described by transfer functions similar to filters. Before we delve further into transfer functions which are a necessity in describing PLLs, we briefly go over the Laplace transform which is used in the description of filtering systems such as this.

The Fourier transform of a signal f(t) is defined by

$$F(\omega) = \int_{-\infty}^{+\infty} f(t)\, e^{-j\omega t} dt$$

and can be written for shorthand purposes as

$$f(t) \xleftrightarrow{\mathfrak{F}} F(\omega)$$

Fourier transform is a special case of the Laplace transform. We generalize the Fourier transform by setting the jw term in e^{-jwt} equal to a complex variable s.

$$s = j\omega$$

We rewrite the Fourier transform equation as

$$F(s) = \int_{-\infty}^{+\infty} f(t)\, e^{-st} dt$$

This is the Laplace transform equation. The Laplace transform is a more general form of the Fourier transform; here we decompose the signal not into harmonic signals but into family of exponentials of the form e^{-st}.

Laplace representation is used extensively in describing filters since the loop-back function of the filters make them in general non-linear and Fourier transform formulation does not work well. We can write Laplace transform in shorthand as

$$F(s) = \mathcal{L}\{f(t)\}$$

and show the transform pairs as

$$f(t) \xleftrightarrow{\mathcal{L}} F(s)$$

Without going into the mechanics, let me state some transform pairs that we will be using.

1. $\dfrac{d\,f(t)}{dt} \xleftrightarrow{\mathcal{L}} s\,F\left(s\right)$
2. $\dfrac{d^n\,f(t)}{dt} \xleftrightarrow{\mathcal{L}} s^n\,F\left(s\right)$
3. $g(t) = \displaystyle\int_{-\infty}^{t} f(\tau)d\tau \xleftrightarrow{\mathcal{L}} \dfrac{F\left(s\right)}{s}$

Knowing these three relationships, we are now ready to write the transfer function of the PLL but first just the VCO.

The VCO input signal is the error signal $s_e(t)$. The output signal is $V_{out}(t)$. The gain of the VCO is K_O. The frequency of the VCO in response to the error signal is

$$\omega_{out}(t) = \omega_c + K_O s_e(t)$$

since the phase is

$$\begin{aligned}\phi_i(t) &= 2\pi\int_0^t f_i(t)dt \\ &= \omega_c t + K_0 s_e(t)t\end{aligned} \qquad (7)$$

The time domain signal that is output by the VCO is a sinusoid of the from

$$V_{out}(t) = A_o \cos\left(\omega_c t + \underline{K_o \int_{t_1}^{t_2} s_e(t)dt} \right) \qquad (8)$$

The Laplace transform of the phase out is given by Eq. (4)

$$\begin{aligned} \Phi_{out}(t) &= K_o \int_{t_1}^{t_2} s_e(t)dt \\ \Phi_{out}(s) &= \frac{K_0}{s} s_e(s) \end{aligned} \qquad (9)$$

Now we write the Laplace transform of the input/output relationship of the VCO.

$$\frac{\Phi_{out}(s)}{s_e(s)} = \frac{K_o}{s} \qquad (10)$$

Moving on to the complete PLL, we can now write its transfer equation by noting what is happening to the signal and using the appropriate Laplace transform.

The filters used in PLLs are generally called lead-lag filters. (Actually all filters are lead-lag filters.) Here is an elementary structure for a IIR filter. The gains are shown by letter G and are called coefficients of the filter.

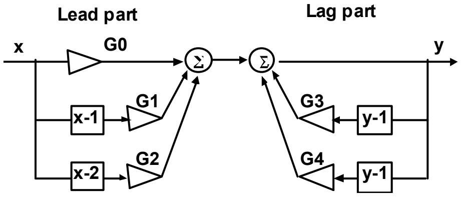


Figure 19 – Generic representation of a filter

We can write the response of this filter by taking a note of what's on the lead side. So the numerator of the filter is equal to

$$G_0 + G_1\, x(n-1) + G_2\, x(n-2)$$

and the denominator is

$$1 - G_3\, y(n-1) - G_4\, y(n-2)$$

we can write the frequency response of this filter by setting each delay as frequency shift.

$$x(n-k) = e^{-jk\omega t}$$

$$H(\omega)=\frac{G_0e^{-j0\omega}+G_1\,e^{-j1\omega}+G_2\,e^{-j2\omega}}{1-G_3e^{-j1\omega}-G_4\,e^{-j2\omega}} \qquad (11)$$

This is kind of a rough looking equation but easy to understand if you know where it is coming from. (For best explanation of how IIR and FIR filters work, see Rick Lyons book "Understanding Digital Signal Processing" best book written on the subject. His down-to-earth approach was an inspiration to me and has helped me to formulate many of my ideas on how DSP should be taught.)

We can show this equation in the s-domain by making the substitution, s = jw

$$H(s)=\frac{G_0+G_1\,s+G_2\,s^2}{1-G_3s-G_4\,s^2} \qquad (12)$$

Now look at this representation. It also has a leading section and a lagging section. The lagging section is the feedback part.

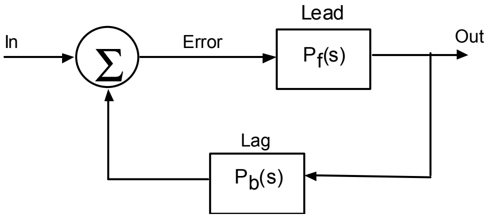


Figure 20 – Generic representation of a feed-back system

The transfer function of this system can be written as

$$H(s)=\frac{Out}{In}=\frac{P_f(s)}{1+P_f(s)P_b(s)} \qquad (13)$$

where index f stands for forward part and b for backward part. We can alternately draw the PLL block diagram as follows.

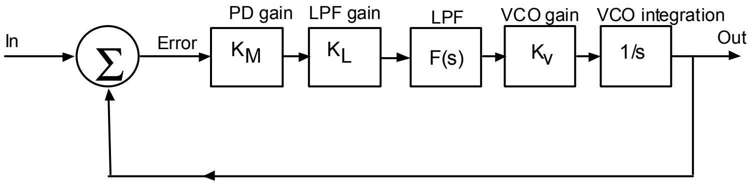


Figure 21 – Linear zed s-domain representation of a PLL

The transfer function of all the components in the top forward of the loop, Pf(s) is just the multiplication all the gains and the filter response and then division by s to represent the integration in the VCO. The gain of the feed back part is one.

$$P_f(s) = \frac{K_d K_L F(s) K_O}{s}$$

set

$$K = K_d K_L K_O$$

Plug this into above equation, we get, the equation of the Closed-loop Transfer function of the PLL.

$$H(s) = \frac{K\,F(s)}{s + K\,F(s)} \qquad 14$$

where the total loop gain is K and F(s) is the Laplace transform of the filter response f(t).

The transfer function of the error can be obtained similarly.

$$\frac{Error}{In} = \frac{1}{1 + P_f(s) P_b(s)} \qquad 15$$

Here again Pb(f) is equal to one and Pf(s) is equal to KF(s)/s.

$$H_e(s) = \frac{\Phi_e(s)}{\Phi_{in}(s)} = \frac{s}{s + K\,F(s)} \qquad 16$$

The actual transfer function depends on the type of loop filter we are using. Three filter types that are used in PLLs are

1. Passive filters lead-lag filters
2. Active lead-lag filters
3. Integrate and lead filters

Most filters are passive in that they do not amplify the signal. An active filter is one that provides amplification in addition to filtering. These types of filters are used in equalizers and of course in PLLs where relative gains are important. The only difference between an active and passive filter is that the gain of a passive filter is 1 or less where the gain of an active filter is K_L. Both filters have the same frequency response except for the linear gain.

The frequency response of the active filter type 2 is given by

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

Now plug this into Eq (), the transfer function of the loop is

$$H(s)=\frac{K\ \ F(s)}{s+K\,F(s)}$$

$$H(s)=\frac{K(1+s\tau_2)}{\tau_2 s^2+s(1+K\tau_1)+K} \qquad 17$$

About the order of PLL – *The order of a PLL is specified by its transfer function. If there is no filter, the PLL is called a first order PLL. The highest power of s in the denominator is used as an indicator of the loop order. The transfer function below is for a 2nd order loop.*

The loop transfer equation can be written in this fashion,

$$H(s)=\frac{\omega_n^2+2s\xi\omega_n}{s^2+2s\xi\omega_n+\omega_n^2} \qquad 18$$

where

$$\omega_n=\sqrt{\frac{K}{\tau_1}} \qquad \xi=\frac{\omega_n\tau_2}{2}$$

The transfer function of the error signal can be written as

$$\begin{aligned}H_e(s)&=1-H(s)\\&=\frac{s^2}{s^2+2s\xi\omega_n+\omega_n^2}\end{aligned}$$

Written this way, the system equation becomes equivalent to what is used in mechanical engineering to describe non-linear behavior of the mass-spring-damper systems. The math is the same and so the terms have been retained and used in the analysis and design of communication system design. Of course, it is nice for once to work with something one can imagine such as a damping factor. There is precious little in communications that is tangible.

The coefficient w_n is called the natural frequency (not to be confused with carrier or center frequency, it has nothing to do with that.). The natural frequency, w_n , is a quality of the response of the PLL. The quantity ξ, which is called the damping factor can be used to examine the transient qualities of the loop. As in mechanical systems, if proper damping factor is not used, the vibrations, error signal in our case, do out damp out and the system becomes unstable. The filter is very important in the design of the PLL since both the natural frequency and the damping factor are a factor of the filter response F(s). In fact we can say that the design of PLL is almost entirely dependent on the design of the loop filter.

Now let's plot these transfer equations and since these are complex equations, we will need to convert these to their magnitude response. And here how we do that.

$$H(s) = \frac{N(s)}{D(s)}$$

$$|H(j\omega)| = \frac{\sqrt{\left(\mathrm{Re}\,al\ part\,of\ N(s)\right)^2 + (imaginary\ part\,of\ N(s))^2}}{\sqrt{\left(\mathrm{Re}\,al\ part\,of\ D(s)\right)^2 + (imaginary\ part\,of\ D(s))^2}}$$

For the closed loop transfer function, the magnitude is given using the above relationship as

$$|H(j\omega)| = \sqrt{\frac{\omega_n^4 + (2\xi\omega_n\omega)^2}{\left(w_n^2 - \omega^2\right)^2 + (2\xi\omega_n\omega)^2}}$$

The error transfer function is given by

$$|H(j\omega)| = \sqrt{\frac{\omega^4}{\left(w_n^2 - \omega^2\right)^2 + (2\xi\omega_n\omega)^2}}$$

Below there two transfer functions are plotted. These plots are called Bode plots. The y-axis is in dBs. The x-axis has been normalized by diving w by wn. You have seen these in books, so what good are these plots? These plots do tell a story. And here is what it says.

First of all, look at the magnitude of the loop transfer function. It looks sort of like the frequency response of a filter. This is what I said earlier, the whole PLL acts like a filter. Where F(s) was the transfer function of just the lowpass loop filter, this response incompasses the other two loop components. The plot shows us how the loop will behave in frequency domain.

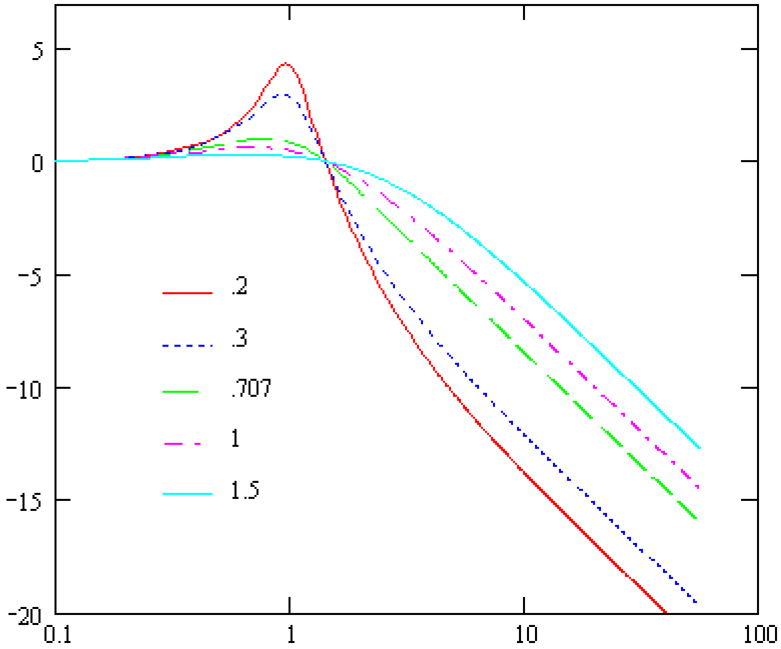


Figure 22- Frequency response of the PLL, x-axis is normalized frequency, the y-axis is Response in dBs. The curves are shown for various damping factors, ξ.

Let me repeat again, the PLL is a filter. So the above graph is its frequency response. The smaller damping factors give better rejection but they also have large transients. At w/wn = 1.0, where the frequency is equal to the natural frequency, remember from your vibrations class that at such point the oscillations become very large, which is just what we are seeing in this graph. Larger factors have better behaved responses but they are sluggish in acquisition and response. The optimum is turns out is the value of .707. Most loops designed use this number as the target value of a good compromise between acquisition time response and frequency response. What do we need this frequency response for? Remember the double frequency term that we need to filter out, this response does by keeping the loop bandwidth narrow. This has secondary benefits because this also limits the noise.

The PLL also has a 3-dB bandwidth just as do all filters. This bandwidth is a function of the damping factor. The 3-db bandwidth for damping factor of .707 is app. 4 times the natural frequency.

We want this cut off frequency to be small but not too small because otherwise it may cut-off baseband information. Generally we want the loop bandwidth to be larger than the largest baseband spectral frequency. If we are transmitting music, which varies from 300 to 30000 Hz, then we would want the loop bandwidth to be larger than 30000 Hz. The most common value of loop bandwidth is 50K Hz.

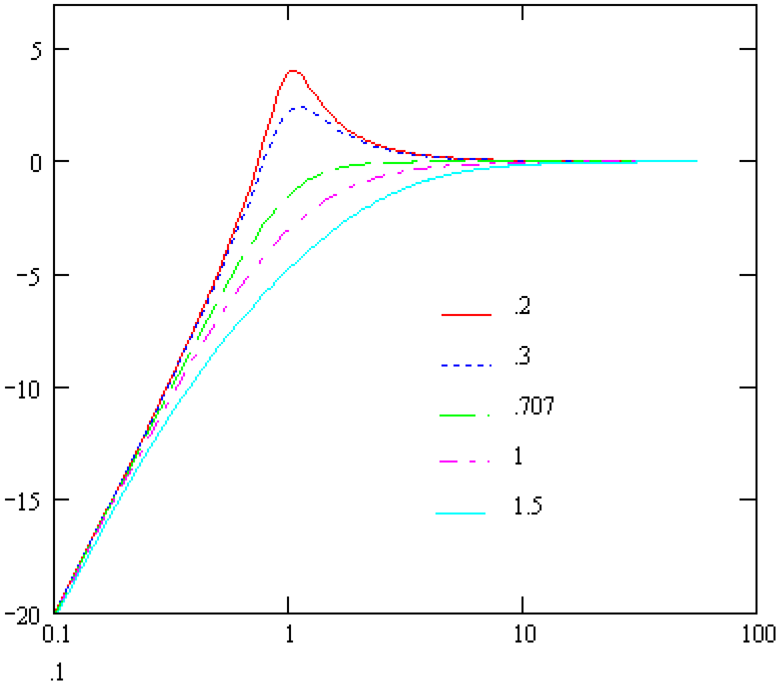


Figure 23- The frequency response of the error function, x-axis is normalized frequency, w/wn,y-axis is in dBs.

This figure tells us what is happening to the error signal as a function of the loop bandwidth. We want the error signal to be able to approach 0. So it is clear that we want the bandwidth to be larger than the natural frequency. Smaller damping factors we can see have larger fluctuations although we can see that they respond faster (the slope is larger for smaller frequencies.). The optimum is again damping factor of .707 which reaches steady state at a loop bandwidth of about 2 times natural frequency.

References:

1. Phaselock Techniques, Floyd Gardener, John Wiley & Sons, 2nd Edition

2. Phase-Locked Loop Circuit Design, Dan H. Wolaver, Prentice Hall, Ist Edition

3. Phase-Locked Loops: Design, Simulation, and Applications, Ronald E. Best, McGraw Hill, 4th Edition

4. Phase-Lock Basics, William Egan, Wiley Inter-science, July 1998

I can be contacted at
mntcastle@earthlink.net

Other tutorials at
www.complextoreal.com