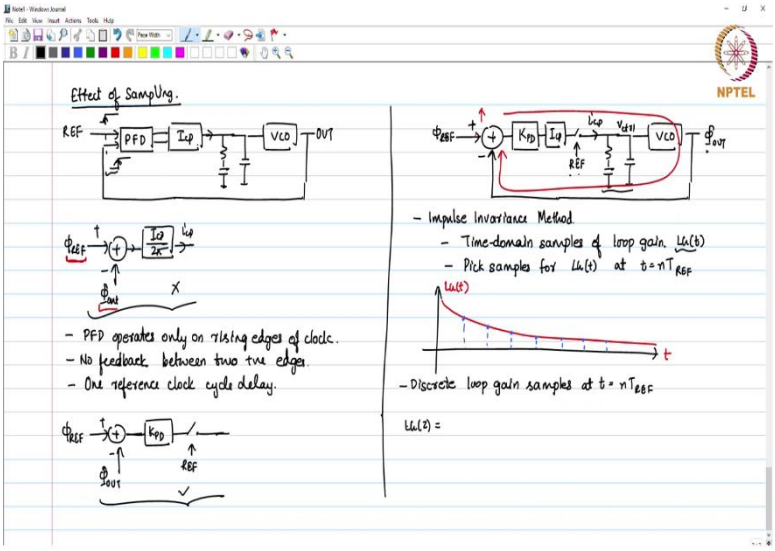
Phase-Locked Loops
Dr. Saurabh Saxena
Department of Electrical Engineering
Indian Institute of Technology Madras

Lecture – 31
Sources of Non-Linearities in CP-PLL: Part II

(Refer Slide Time: 00:16)



Hello, welcome to the session. We have been discussing the non-linearities in the charge-pump and one of the non-linearities in our modeling is that we completely forgot the effect of sampling in our charge-pump. So, let us see what it is. So, what we have is if you look at it, in our PFD based PLLs, we measure the phase error only once in reference frequency, isn't it? So, we give these UP and DN pulses and you have charge-pump, so far so good, and this R C_1 C_2, VCO and so on.

So, the way we modelled our PFD plus charge-pump was that whenever you have phase error, you have φ_{REF} and you have φ_{OUT}, whenever you have phase error, the phase error between reference and output, it gets subtracted by the PFD and converted by the charge-pump and you get current which is proportional to phase error. That is how we modelled. But what happens is actually not this. What happens is when you have phase error between PFD and charge-pump, the current comes at the output only once in a reference frequency. If the phase changes between the input and the output during the reference period, that particular phase error is going to be whatever phase error you accumulate, that phase error is going to be entertained by the PLL loop only during the next reference cycle.

So, if I want to write, it is like PFD operates only on the rising edges of the clock. It does not operate in between, it operates like that. There is no feedback once you get the rising edge on the reference and the rising edge on the VCO, once you get this, this phase error is in, once this phase error is taken into account, after that even if the VCO frequency drifts or anything happens there, there is no feedback between two positive edges.

So, if your V edge has changed here because of this you are changing then it is like until the next V edge comes you are not going to do anything, whatever happens to the VCO. Another thing is there is you can say one reference clock cycle delay. You change anything in the charge-pump, if there is any disturbance or anything in the charge-pump or PFD, if it happens during the current reference cycle or before the rising edge on the reference or the output clock, it will be taken into account otherwise if it happens after that, it is going to be taken into account only in the next reference clock cycle.

So, there is like a one reference clock cycle delay. So, what we are doing in our modeling is the following. That what we are doing so far, we are actually sampling our system in the real PLL, if you look at it, it means I did approximations initially using z-domain transfer function, writing difference equations and so on but those were approximations. When you look at your actual PFD, what we are doing here is the following, we are having our reference phase and our output phase and output or divided phase, if there is a frequency divider and whatever value it gives.

We are sampling this, effectively what we are doing is we are sampling in our system at reference rate. Our actual model is or you can say a model which is closer to our PLL operation is this, rather than this. This is not the exact model, a more accurate model is this one. You can get phase error, but I will sample the phase error only at the reference rate. So, when you have such kind of PLL in the system, so, let me just do it how we will analyze or can we still continue with our modeling which we have been using.

So, let me just remove this and say a more accurate model for our PLL is the following where you can have the phase error between input and output or the reference and the output, it gets multiplied by your PFD gain K_{PD}, it gets multiplied and you can have your charge-pump and this is actually a sampled version.

So, you sample at reference rate by your reference clock, you get from your PFD you have the phase error then that particular whatever input and output phase error you have, φ_{OUT}, by the

way this is OUT here rather than VCO. You have phase error between reference and output, that particular phase error, your PFD gives a proportional value and that particular phase error is sampled and it is sampled at the reference rate.

So, this is something which we have seen. We ignored it at that time, we approximated it but now it is the time to see how far our approximations were valid. So, what happens here is, in this particular case, I_{CP} changes only at fixed intervals and that is how you change your control voltage and so on. So, there are a couple of methods to analyze such kinds of systems.

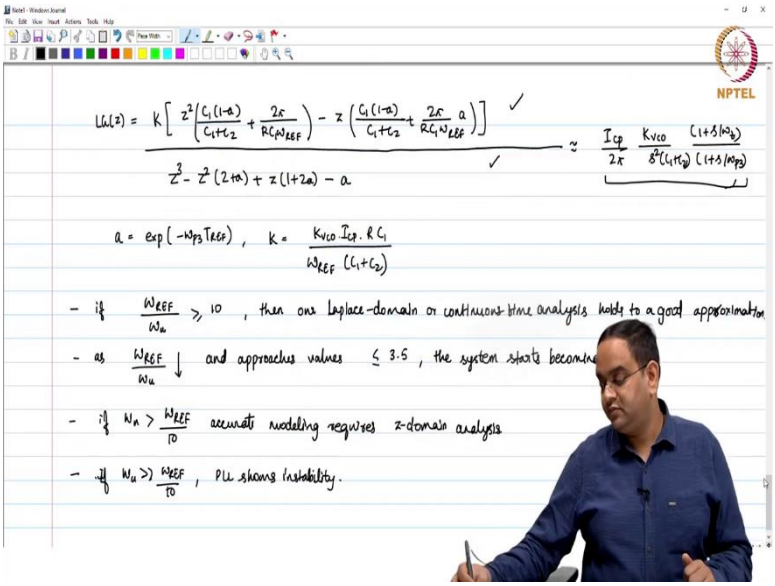
One of the methods is impulse invariance method to analyze such a PLL. There is an important conclusion after our discussion, so, let us look at it how we arrive at that conclusion, impulse invariance method. In this method, you can say, I want to find the loop gain of this system. This is like you have R and C, and you have a sampled value.

So, what you will do is you will apply an impulse and you do all sampling and everything and you see what comes back in time domain. So, when you apply any change at the input and you see what comes back in time domain, that is effectively what you are doing here is you are looking at time-domain samples of the loop gain, and that is something which you are calculating using the exact operation.

So, you calculate time-domain samples of loop gain. I will say that they are $LF(t)$, and you pick up samples at regular intervals from this because when you are going to apply an impulse, you will have a real signal as a function of time. For example, just to tell you, if I have loop gain, this is just a $LF(t)$ in my real system, what I am going to do is I am going to sample that at fixed rate which is T_{REF} because that is what is happening in the system. I am sampling my loop gain signal at fixed intervals t.

So, pick samples for $LF(t)$ at $t = n\,T_{REF}$. So, you get the samples for the loop gain at $n\,T_{REF}$. Now, you have discrete loop gain samples at $t = n\,T_{REF}$. I can use these samples to calculate $LG(z)$. So, you calculate $LG(z)$, so, this is going to be z-domain for the loop gain and then you analyze your closed loop system.

(Refer Slide Time: 11:55)



So, when I go ahead and do all such things, you will be surprised that the actual loop gain expression in our case making sure that the sampled values are from the actual system, the $LG(z)$ is going to be,

$$LG(z) = \frac{K\left[z^2\left(\frac{C_1(1-a)}{C_1+C_2} + \frac{2\pi}{RC_1\omega_{REF}}\right) - z\left(\frac{C_1(1-a)}{C_1+C_2} + \frac{2\pi a}{RC_1\omega_{REF}}\right)\right]}{z^3 - z^2(2+a) + z(1+2a) - a}$$

Well, you can always refer to the papers which first published this. Here, we have,

$$a = \exp\left(-\omega_{p3}T_{REF}\right)$$

$$\omega_{p3} = \frac{1}{\frac{RC_1C_2}{C_1+C_2}}$$

$$K = \frac{K_{VCO}I_{CP}RC_1}{\omega_{REF}(C_1+C_2)}$$

So, while using the sampled values of the PLL and finding the loop gain from there, the loop gain happens to be like this.

Earlier, we were approximating this loop gain as follows:

$$LG(z) \approx \frac{I_{CP}}{2\pi}\frac{K_{VCO}}{s^2(C_1+C_2)}\frac{\left(1+{}^{s}/_{\omega_z}\right)}{\left(1+{}^{s}/_{\omega_{p3}}\right)}$$

This is what we were using but the actual expression is this. Now, you can very well find the bandwidth while using this transfer function. What you are going to see is the following. First, if $\frac{\omega_{REF}}{\omega_u} \geq 10$, then our Laplace-domain or continuous time Laplace-domain or continuous time analysis holds to a good approximation and we have verified this by getting the actual plot or the response comparing the Laplace-domain response with the discrete time modeling.

As $\frac{\omega_{REF}}{\omega_u}$ reduces which can only happen when ω_u increases in comparison to ω_{REF}, when $\frac{\omega_{REF}}{\omega_u}$ reduces and approaches values $\leq$3.5, these are the calculations which you will see, you will find that the system starts becoming unstable and what happens is that you will start seeing the closed loop poles in the z-domain analysis going outside the unity gain circle which is the condition for instability.

So, as you see the condition for instability in case of Laplace-domain analysis is that you have poles in the right half plane. Similarly, in z-domain analysis, if the poles are outside the unity gain circle, then the system will be unstable. So, as this ratio of $\frac{\omega_{REF}}{\omega_u}$ reduces, then what you will have is that your closed loop poles start going outside the unity gain circle and you will lose on the stability.

So, as a rule of thumb now, I will not say that if your unity gain frequency exceeds ω_{REF}, then your PLL will become unstable, no that is not the case. In many implementations we can implement $\omega_u > \frac{\omega_{REF}}{10}$. We have implemented $\frac{\omega_{REF}}{8}$ and even higher, but the thing is the matching between this transfer function and our discrete time transfer function will not be there anymore.

So, if you are implementing your PLL with $\omega_u \leq \frac{\omega_{REF}}{10}$, you can very well be assured that your continuous time analysis or Laplace-domain analysis is quite accurate to model this sampled system. If $\omega_u > \frac{\omega_{REF}}{10}$, accurate modeling requires z-domain analysis. Please do not use Laplace-domain analysis or the continuous time analysis which we have done.

If $\omega_u \gg \frac{\omega_{REF}}{10}$ or it approaches you can say even $\frac{\omega_{REF}}{2}$ or something, the PLL will become unstable. PLL shows instability because your closed loop poles go beyond the unity gain circle. So, keeping this in mind, we will design our PLLs. Thank you.