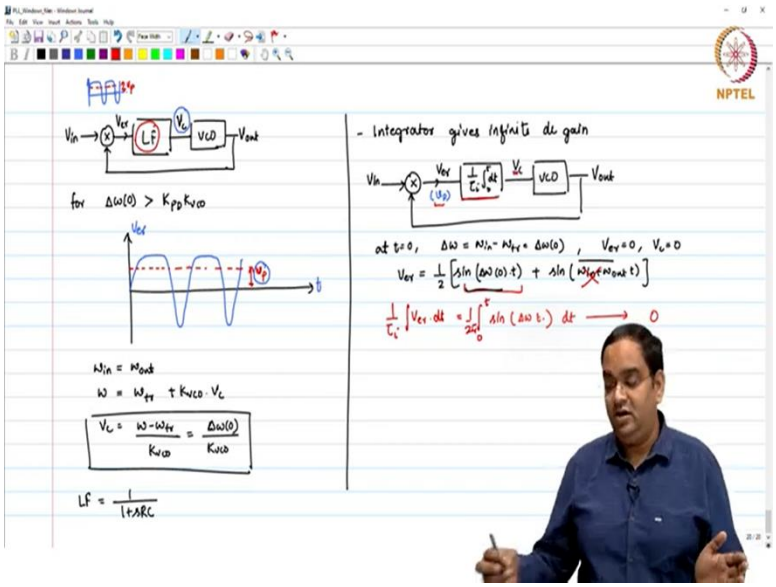
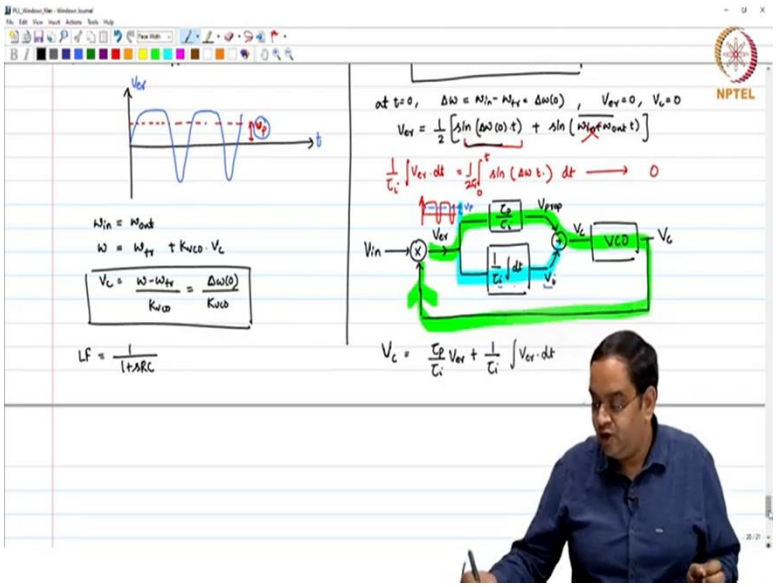
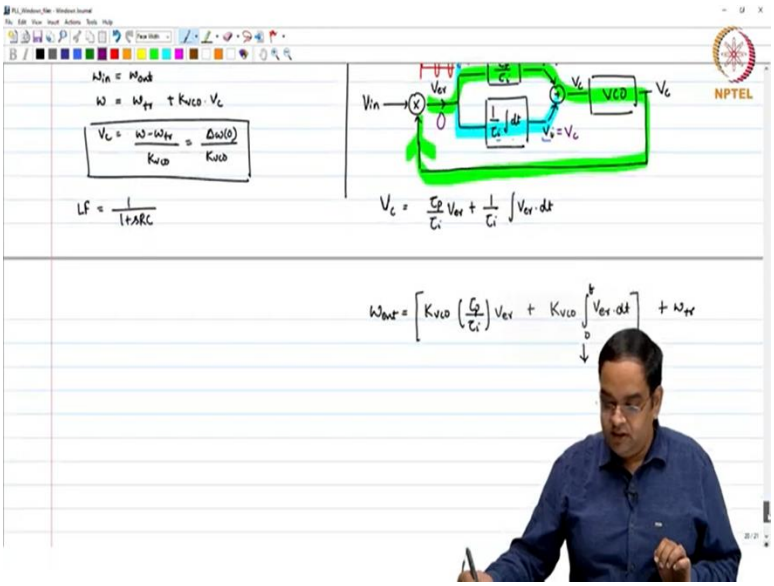
Phase-Locked Loops
Dr. Saurabh Saxena
Department of Electrical Engineering
Indian Institute of Technology, Madras

Lecture – 13
Frequency Acquisition in Type-II PLLs

(Refer Slide Time: 00:18)







In the previous session, we saw the case of the PLL where the PLL did not lock and it generated asymmetric beat note as its error voltage. Let me just write this as the loop filter here followed by VCO, this is V_{in}, and this is V_{out}. We had an initial frequency error which was large, the PLL did not lock and it generated a lot of voltage.

Now, you think about it, for $\Delta\omega(0) > K_{PD}K_{VCO}$, the error voltage generated a waveform which looked like this, this is the error voltage with respect to time. It has some DC voltage which we call as the pull-in voltage. Now, the first question which we should ask is that why the PLL did not lock? Well, the PLL did not lock because the control voltage was not as required. You did not have the control voltage as required to lock the frequency.

Just to repeat this, you can think about it that in order that I lock the VCO to the desired frequency or desired frequency error in the beginning, I should have the following:

$$\omega_{in} = \omega_{out}$$

$$\omega = \omega_{fr} + K_{VCO}.V_c$$

So, the control voltage is given by,

$$V_c = \frac{\omega - \omega_{fr}}{K_{VCO}} = \frac{\Delta\omega(0)}{K_{VCO}}$$

I should have this control voltage in steady state.

In the case which we are discussing, $\Delta\omega(0)$ is pretty large and the control voltage which is generated by the PLL loop is not large enough. But one interesting thing which we found that the error voltage which was generated in such a case has some DC positive value. So, at this node, you can say that you are seeing some voltage waveform like this. It has non-zero DC voltage.

So, if I want to increase the control voltage, this control voltage you need at steady state which means that you need a DC control voltage which is large enough for the PLL to lock. If that is the case, then one can think about it that I am having some non-zero DC value here already, let me amplify this, because this DC value is coming at V_{er}.

So, V_P DC value is going to the loop filter. If I can amplify the DC value through the loop filter, I may be able to lock it. We were not able to do so because the loop filter which we have been using, $LF = \frac{1}{1+sRC}$ has a DC gain equal to 1. So, you are not able to do it. So, if you were not able to do it, we want to give a large DC gain to amplify the voltage as we need. If we want to do that, then we can very well replace our loop filter with an integrator.

We want a large DC gain, an integrator is a block which gives you infinite DC gain. Integrator gives infinite DC gain, this is something which we recall. We have the pull-in voltage, some non-zero DC value, we amplify it with infinite gain, you can say we can surely get as large voltage as we want. So, we change our PLL from the previous one to the one where we replace the loop filter with an integrator. You integrate the error voltage which is coming and pass it to the VCO. This is the error voltage, and this is the control voltage.

So, the assumption is that when we are going to do this, we will have this pull-in voltage, V_P here, and this V_P will actually get amplified or integrated to give you a large control voltage, that is what should happen. But, actually what happens is that when you have a large frequency error in this case, and you start at time instant $t = 0$ with frequency error $\Delta\omega(0)$, we have,

$$\Delta\omega = \omega_{in} - \omega_{fr} = \Delta\omega(0)$$

This is the frequency error, and all the initial values which you have are $V_e = 0$, $V_c = 0$, to begin with. The error voltage is given as,

$$V_{er} = \frac{1}{2}\left[\sin(\Delta\omega(0)t) + \sin\left((\omega_{in} + \omega_{out})t\right)\right]$$

I am neglecting $\sin\big((\omega_{in}+\omega_{out})t\big)$. I have been neglecting this assuming that this is going to be filtered by the loop filter or the integrator. So, $\sin\big((\omega_{in}+\omega_{out})t\big)$ is neglected for now, only $\frac{1}{2}[\sin(\Delta\omega(0)t)]$ remains.

So, what are you doing when you start your PLL? You are feeding this sine wave to the integrator. You feed this sine wave to the integrator, initially the integrator output was 0, so, the control voltage (V_c) had no information about the phase error. All the information which V_c can have about the phase error will come through the integrator. So, this will come through the integrator. The integrator input itself is a sine wave. So, what will you get if you integrate V_{er} here? You are integrating a sine wave. If you integrate a sine wave with DC value of 0, this integration over an integral number of periods of the frequency error is going to be 0. So, this is the problem, that when you use only integrator in this particular loop, to begin with, you do not have pull-in voltage actually, because the VCO does not have any information about the error.

So, V_c was initially 0. So, it is like running in an open loop while not controlling the VCO frequency and the average of sine wave is going to be 0. So, when we integrate this, we get,

$$\frac{1}{\tau_i}\int V_{er}\,.\,dt = \frac{1}{2\tau_i}\int_0^t \sin(\Delta\omega\, t)\, dt \longrightarrow 0$$

So, V_c does not develop any DC voltage. So, the PLL still does not lock. So, what do we need to do? We need to provide the pull-in voltage for the integrator and at the same time we need to integrate the pull-in voltage to get the frequency lock.

So, what we need to do is the following. You give this as V_{in}, this you get as V_{er}, and V_{er} goes through two paths. The first one is the previous path which has a DC gain of one, you can think that way. So now, I will just use a proportional constant of value $\frac{\tau_p}{\tau_i}$, and the other one with an integrator, $\frac{1}{\tau_i}\int dt$. These two things I am going to add somehow, do not worry about how they will be added, and then this goes to VCO.

Now just try to understand it a little intuitively. You start this PLL with frequency error, $\Delta\omega(0)$, you get V_{er}, this is V_c, I will call this as V_i and this is V_{prop}. Because there is a path from V_{er} to $\frac{\tau_p}{\tau_i}$ going to VCO and coming back, we will be able to generate asymmetric beat note at V_{er} value.

And then, because you are having asymmetric beat note, you have a pull-in voltage, V_P, which is a non-zero DC value, and this V_P gets integrated, you are integrating a non-zero value, so this V_P will keep getting integrated to increase the integral path output voltage. So, I can say that V_P from here gets integrated and increases V_i voltage.

And when you increase V_i voltage, so, in this particular case, what will you write? You will write,

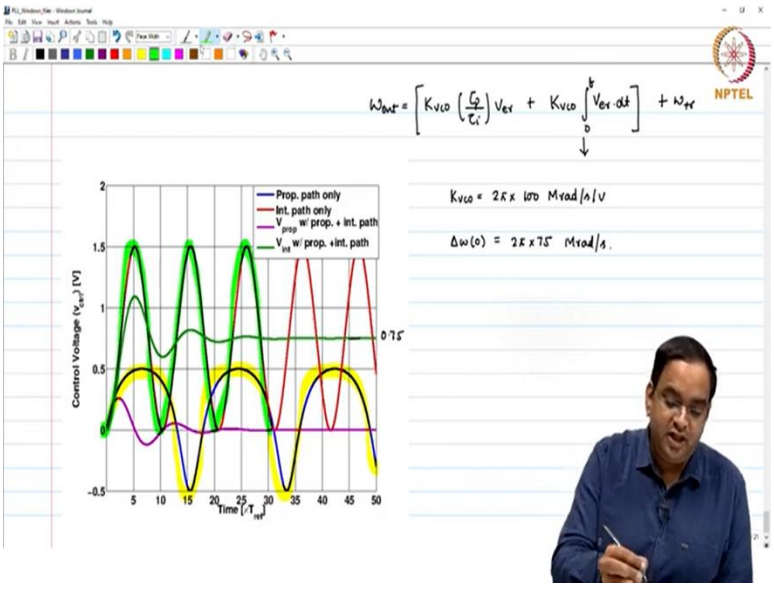
$$V_c = \frac{\tau_p}{\tau_i} V_{er} + \frac{1}{\tau_i} \int V_{er} \, . \, dt$$

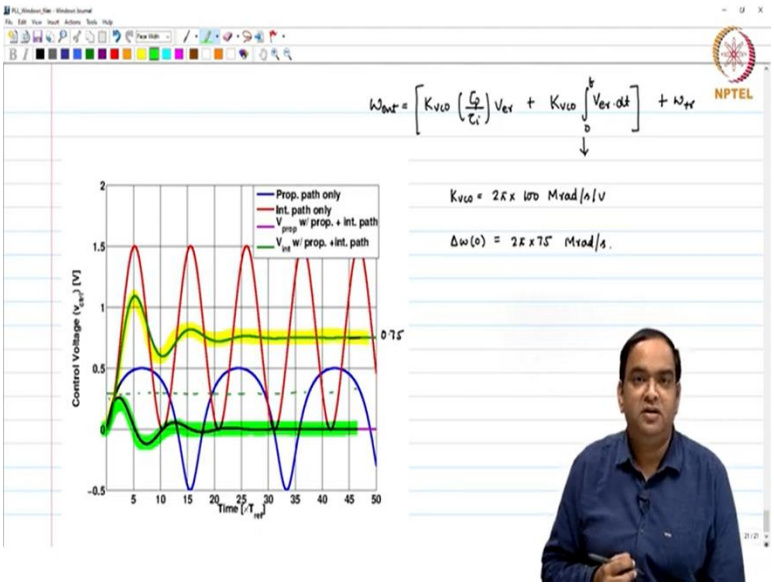
This is what the control voltage will be. So, if I want to write the output frequency, I have,

$$\omega_{out} = \left[K_{VCO} \left(\frac{\tau_p}{\tau_i} V_{er} \right) + K_{VCO} \frac{1}{\tau_i} \int V_{er} \, . \, dt \right] + \omega_{fr}$$

So, you are seeing two values, one is the DC pull-in voltage and the other you can say, if you remove the DC part from your asymmetric beat note, that is something whose average will be 0. So, you have the output frequency like this. This particular part will keep on increasing the frequency in the desired direction such that the overall frequency error reduces. And it is only when the frequency error reduces that the phase error will also keep on changing, and at one point of time, what you will see is that the phase error will be equal to 0. In steady state, what you are going to see is that the phase error will be equal to 0, the error voltage will be 0, V_{prop} will be 0, and V_i will have whatever control voltage is desired to lock the frequency.

(Refer Slide Time: 13:54)





Let me show you the simulation part for this particular analysis. So, in this particular plot, this is the same example which I have been using in which $K_{VCO} = 2\pi \times 100$ Mrad/s/V and I am plotting for a given frequency error here. So, we will figure out what frequency error we have. Well, you see that when we have integral as well as proportional paths, V_i settles to 0.75.

So, if it is 0.75, then, the initial frequency error which I am having is $\Delta\omega(0) = 2\pi \times 75$ Mrad/s. So, this was the initial frequency error which is actually greater than $2\pi \times 50$ Mrad/s, which we have seen earlier where the PLL did not lock. So, let us discuss one by one. So, when I had the proportional path only, which is the path using $\frac{\tau_p}{\tau_i}$, you see that the control voltage just keeps on creating these asymmetric beat notes, there is no change. The PLL did not lock, control voltage just keeps on changing.

In case we had the integral path only, then you see for the case of only integral path, you are integrating a sine wave, you will get cosine and it keeps on integrating and it never settles which means it never reaches a finite value, it keeps on changing. If control voltage keeps on changing, then the frequency of the oscillator will also keep on changing. So, that is what happens.

But, you see that in the case in which we have both the proportional and the integral paths, then the pull-in voltage of this asymmetric beat note is actually utilized to integrate and finally achieve a fixed value on the integral path as shown here. And along the proportional path, what you are seeing is that the phase error actually changes and finally it settles to 0. And the value which you have in the integral path compensates for the initial frequency error.