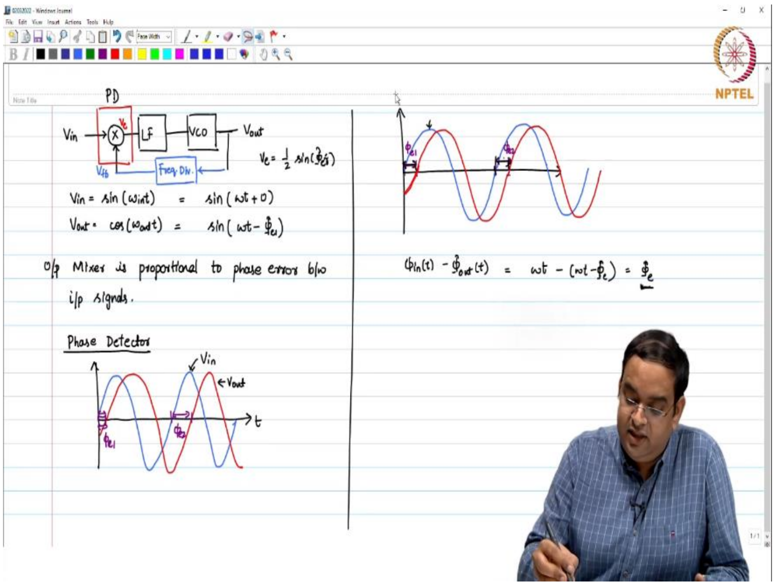
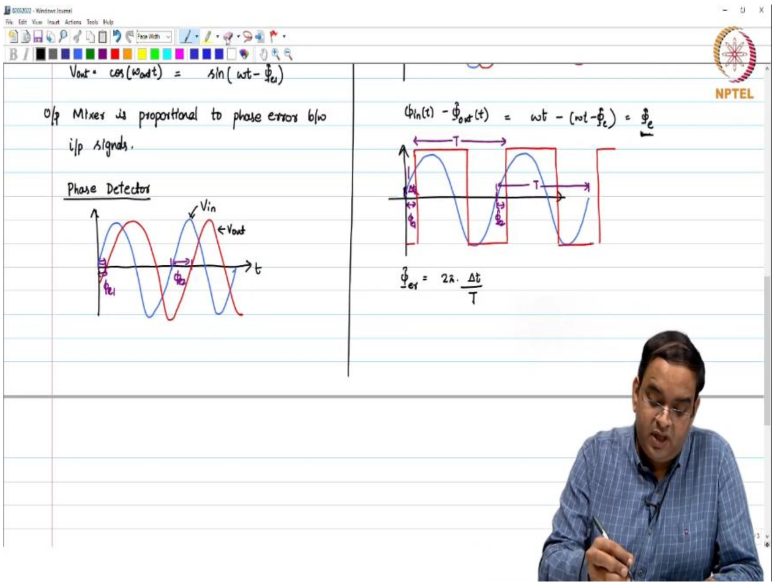
Phase-Locked Loops
Dr. Saurabh Saxena
Department of Electrical Engineering
Indian Institute of Technology, Madras

Lecture – 17
Analog Phase Error Detectors: Part I

(Refer Slide Time: 00:15)





Hello, welcome to this session. In the previous session, we have actually analyzed a simple PLL in transient and frequency domain. So, I will just plot what we have seen so far. We had loop filter, VCO and this was in the feedback path. Here, we confined ourselves to the sinusoidal periodic signals as given by,

$$V_{in} = \sin(\omega_{in}\, t)$$

$$V_{out} = \cos(\omega_{out}\, t)$$

At the end of the last session, we said that we can have a frequency multiplication here, if we just have a frequency divider which you can implement by having some other block which divides the output frequency. So, I will call this as frequency divider. So, this is what we had, but still we kept our input and output signals of this frequency divider as sinusoidal.

Now, all the periodic signals which you are going to use in your system may or may not be sinusoidal signals. As soon as you think about a clock which is used in digital circuits, you think about this clock has levels of 0 and V_{DD}, it is a full swing signal and it appears to be like a square wave signal.

So, when you are having a square wave signal, using this mixer based phase error detector becomes difficult. So, at the end of the day, what this mixer is doing is that it is trying to find the phase error between the input and the output phase. The output of the mixer is proportional to the phase error between the input signals. So, one is the reference signal and the other is the feedback signal.

Now, when both the signals are not sinusoidal, how will you define the phase error? How are you going to design that particular block? This is something which we would like to learn now. So, in our case, now, this phase error detector, this particular block which you are seeing in red, this is PD. We will look at different kinds of phase error detectors which will work with sinusoidal signals and square wave signals. Then, depending on the signals which we are having in the system, we will choose the phase detector. So, in this session, we are going to learn about different kinds of phase detectors.

To begin with, let me just first draw the signals which we have been using and what the sinusoidal phase error detector was doing so far. So, if we have a sine wave, assume that it is a sine wave and then you have another sine wave. So, this is just an example, the one in blue is V_{in}, and the one in red is V_{out}, with respect to time. This is what I am plotting. The phase error between these two signals, if it happens that the frequency is going to be the same, the phase error is this, and the frequency of both the input and the output signal is same, the phase error is defined here as the difference between the zero crossing of these two signals. So, here it is like this.

So, if the frequency of the two signals is same, then this particular phase error, φ_{e1}, which is here, and φ_{e2}, they both should be same. It does not appear to be same here. So, let me just try to show you in a much cleaner waveform. So, in this case what you see here is, I will just extend these two sine waves. The difference between the zero crossing which you see here, this particular difference is constant if the two frequencies are same. So, if I call this as φ_{e1} and this one as φ_{e2}, this difference is constant. Do not worry about the maths of it. Let us say, if I have the input signal with phase offset as zero. So, we have,

$$V_{in} = \sin(\omega\, t + 0)$$

Let us say that the output signal is given by,

$$V_{out} = \sin(\omega\, t - \varphi_e)$$

So, the difference between the input and the output phase is given by,

$$\varphi_{in}(t) - \varphi_{out}(t) = \omega t - (\omega t - \varphi_e) = \varphi_e$$

Also, as you see that if the two frequencies are same, this error will be constant, it will not change with respect to time. If the phase error does not change with respect to time, that has been one of the locking conditions for the PLL. So, if you think about it, if I want to measure the phase error between two signals, if somehow I can measure this difference between the zero crossing of these two signals, I actually get the phase error between the two signals.

What we have been doing previously, we were multiplying the two signals where we had the error voltage as $V_e = \frac{1}{2}\sin(\varphi_{er})$. We were not getting the phase error directly, we were getting a signal proportional to the phase error. There is nothing wrong with this kind of phase error detection, it is just that there are other ways also to detect the phase error.

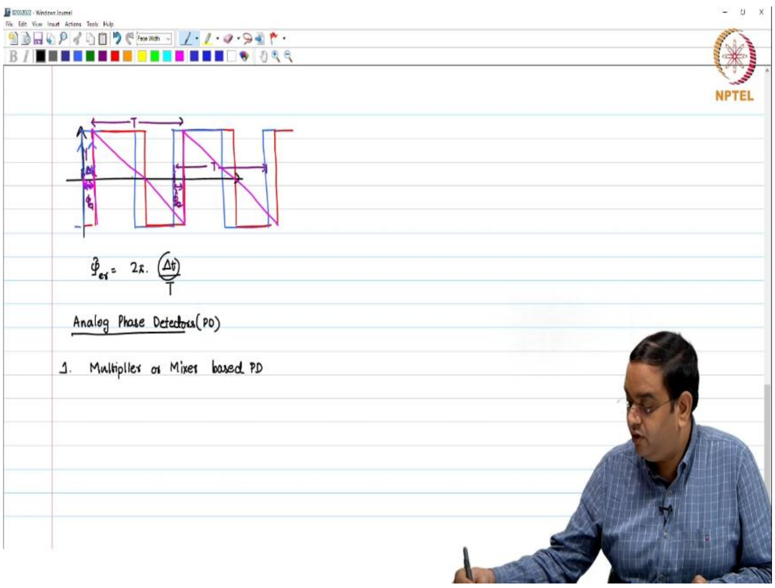
So, right now, we are only considering two sinusoidal signals which have the same frequency. And for the same frequency, the phase error will be constant. So, if somehow I can detect this, then I will be able to get the phase error. So, the phase error detection exploits this. The other case, well, you do not always have sinusoidal signals, if let us say, one signal is sinusoidal and the other signal happens to be a square wave. So, what will you do? So here, I have one of the signals as a sine wave as you see, and let us say the other signal happens to be a square wave, just an example. So, it is a square wave. In place of the sine wave, I am just using a square wave. So, what I am going to do is, I will try to keep the same spacing between the zero crossings. So, this is one part of the square wave, then you have this, it shifts here and then you have the other part going up here. So, these are the two signals. So here, the phase error will be defined as the difference between the zero crossings of the two signals. This is the phase error. So, this is the phase error which you would like to detect. So, this is φ_{e1}, and this is φ_{e2}. By the way, this square wave is a periodic signal and it can very well be the clock which is generated by the PLL and used in your circuits. So, the fundamental period of the square wave and the period of the sine wave are same. If that is the case, the phase error which you are seeing here is the difference between the zero crossing of the two periodic waveforms. So here, the phase error will be defined as,

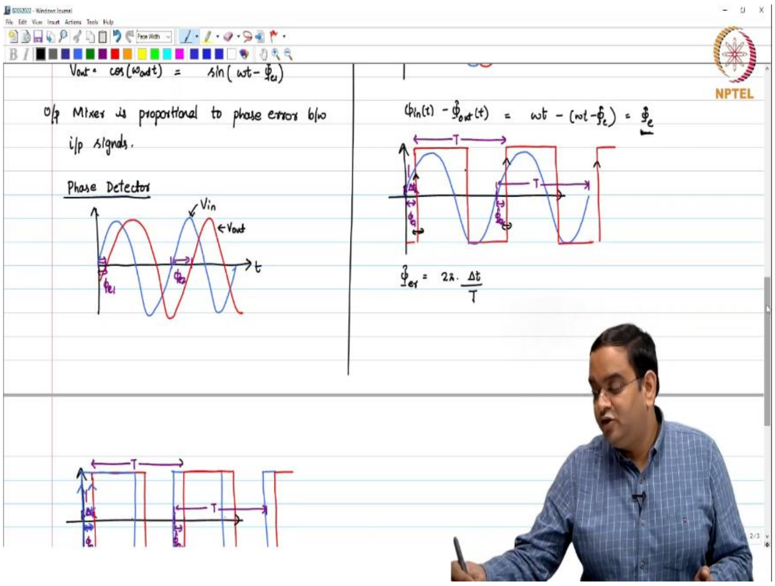
$$\varphi_{er} = 2\pi \frac{\Delta t}{T}$$

where, Δt is the separation between the zero crossings, and T is the time period.

So, when you have such kind of waveforms, just think about it, if you start writing the Fourier series of the square waveform, you have so many signals, but that is not how we define here. We define the phase error here as the time difference between the zero crossings of the two periodic signals. We pick up the point as shown here. If these two frequencies are going to be same, then you know that this $\varphi_{e1} = \varphi_{e2}$. How we are going to detect it, that is something which we will see. Similar to the case we saw where we had one as a sine wave and the other as a square wave, let us see what happens when we have both the signals as square waves.

(Refer Slide Time: 11:47)





So, in case I have both these waveforms as square waves. So, let me just remove this blue one and one of the waveforms which I am going to have is the red one which is already drawn. So, the blue waveform, another square wave, another clock waveform is like this. They both are periodic waveforms. So, how am I going to define the phase error between these two square waveforms? The same definition applies here as well. The time difference between the rising edges of the two periodic waveforms. So, here also, the phase error will be defined as $2\pi\ \frac{\Delta t}{T}$. So here, you see that as Δt increases, the phase error increases. This is an ideal representation of the phase error between two square waves.

How is the phase error detector going to detect and implement this? That is something which we will learn as part of the phase error detectors. So, one thing which you need to understand from here is that the phase error detector which we have used so far works for the sinusoidal signals, but when you have a combination of sinusoidal and square waves or only square waves which are also periodic signals, then the phase error definition which I am using here is the timing error between the zero crossings of the two signals. You can say zero crossing or you can pick up any point at the peak of the signals or so on. Because these are periodic signals, so whatever you are going to choose, that is going to repeat every clock cycle.

Now, you can think about it that this phase error definition is like only the timing error. Why are we only looking at the timing error? Well, PLL is a block which generates a periodic signal whose period is intact, whose period does not change. So, the sine wave can be your reference signal and the square wave can be your output signal or square wave can be your reference signal and square wave can be your output signal, whatever it is, what we need to do in the

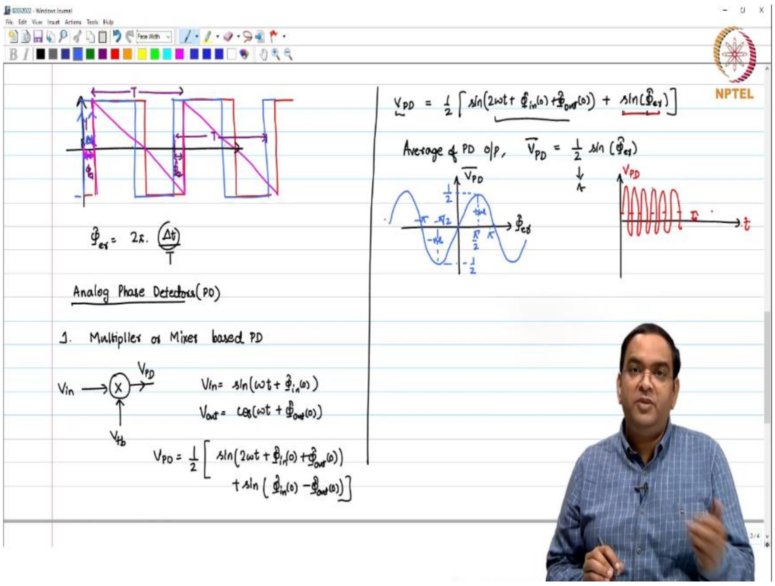
PLL block is that we need to generate an output clock in such a way that the rising edge or the period of this output waveform does not vary. Ideally, it should not vary at all. In real PLLs, it will vary by some amount and you would like to keep that variation limited. So, that is what we need to do.

So, if that is the case, a PLL is a block where you limit the variation in the period of the signal. This means you are limiting the variation in the movement of this rising edge. If the rising edge moves, the period changes. So, we are trying to limit the variation in the time period. So, let us say that here the input is the reference clock, and if we detect only the difference between the zero crossings of the reference clock and the output clock which is a square wave here, then, in the loop, we can try to limit the variation.

So, that is why we are looking at the timing information only. By the way, you can say if that is the case, if I am generating periodic signal, well, a periodic signal can be like this also, isn’t it? So, for example, I am having this, I can have a periodic signal like this also. Just an example. If, let us say, this is the periodic signal. Well, it will be difficult to generate such a kind of signal using a voltage controlled oscillator, but if this is a periodic signal, then it is fine, you can use it as your clock, in some cases. But what we are looking here is the difference between the two periodic points or two periodic voltages in the two signals.

So, this phase error detection implemented by phase error detectors is a part of the PLL. There are different kinds of phase error detectors. So, the first type is analog phase error detectors. One of the analog phase error detectors is something which you have seen. We will go through it quickly. Well, I call this as phase error detectors, but I always write as phase detectors, that is a convention here. So, the first one in analog phase error detectors is multiplier or mixer based. This is the phase error detector which we have been using for long. We will just look at more details of this mixer based PD. How are you going to use it?

(Refer Slide Time: 17:56)



So, in this case, we know that this is a mixer which multiplies the two signals. You are going to use this phase error detector only when you have sinusoidal signals. So, these are V_{in} and V_{fb}. I call the output of the phase error detector as V_{PD}. I am not calling this as phase error right now. This is the output of the phase error detector. So, this is V_{PD}. Given that,

$$V_{in} = \sin\left(\omega t + \varphi_{in}(0)\right)$$

$$V_{out} = \cos\left(\omega t + \varphi_{out}(0)\right)$$

I have included the initial phase offsets for the input and the output signals. I have deliberately made the frequency of the two signals same. So, the output of the phase error detector is given by,

$$V_{PD} = \frac{1}{2}\left[\sin\left(2\omega t + \varphi_{in}(0) + \varphi_{out}(0)\right) + \sin\left(\varphi_{in}(0) - \varphi_{out}(0)\right)\right]$$

Now, as we know that V_{PD} given in the above equation can be just written as,

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

Now, we have this phase error detector in the PLL and the output of this phase error detector has two components, one is depending only on the phase error, and the other has a very high frequency component.

When you employ this as the phase error detector and subsequently you have a loop filter, this high frequency component will be suppressed. Even if you do not have a loop filter, or if the loop filter gain is one, let us say, you just pass it, you have an integrator in the form of a VCO. So, high frequency components will anyways be suppressed. It depends by how much magnitude, but as you start using a first order or second order loop filter, those high frequency components will be suppressed.

So, what we have been doing earlier is that we have been saying that V_{PD} or the output of the phase error detector has $\frac{1}{2}\sin(\varphi_{er})$ component only. This is under the assumption that the high frequency component is going to be filtered out. So, we define the average of the phase detector output as,

$$\overline{V_{PD}} = \frac{1}{2}\sin(\varphi_{er})$$

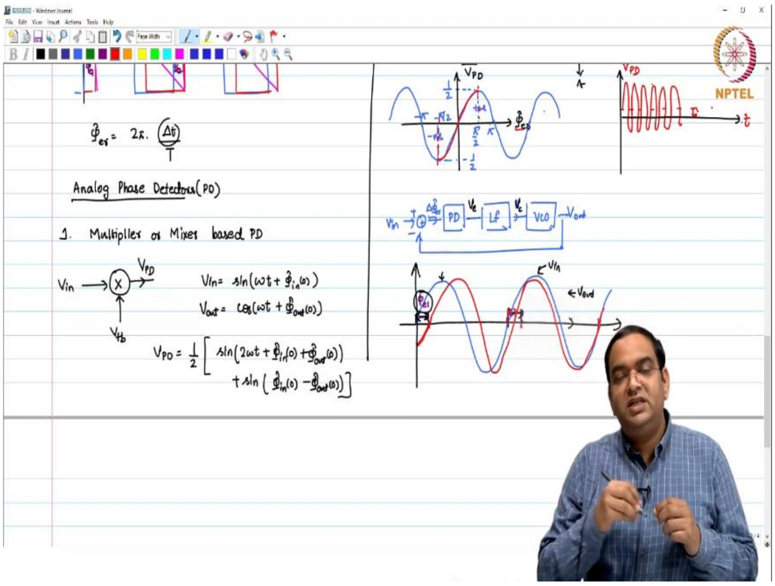
Again, this is something which we have seen, and which we need to understand in a little more detail. As you see here, $\overline{V_{PD}}$ with respect to the phase error is sinusoidal. This is $-\pi$, this is $+\pi$, the peak point is $\frac{\pi}{2}$, and the peak point here is $-\frac{\pi}{2}$. In this case, this magnitude is actually equal to $\frac{1}{2}$ or $-\frac{1}{2}$.

The mixer is going to be implemented by some circuit, and it will have swing limitation. So, $\frac{1}{2}$ will actually change to amplitude A. We are not looking at that right now. So, in this case, the first question which you can ask is that why are we concerned about $\overline{V_{PD}}$? Because that is the quantity which is going to change the control voltage of the VCO, the VCO frequency and the feedback phase. It may happen that if you are changing the phase error, your value will keep on changing. But, for a fixed phase error, you will have a fixed $\overline{V_{PD}}$ voltage.

So, if you just look at V_{PD} signal as such, it is $\frac{1}{2}\sin(\varphi_{er})$. Remember, if the phase error is constant, it is nothing but you can say a very high frequency which is like 2ω kind of sine wave, it is not damping, but it is shifted by $\sin(\varphi_{er})$. That is what you are going to have, in case of actual V_{PD} with respect to time. So, the loop actually processes only this part and changes the frequency. So, that is the part which we are plotting, $\overline{V_{PD}}$.

Now, there are a couple of things to be noted. First, this phase error detector output is for positive phase error for now. For 0 to π, it is positive, and for $-\pi$ to 0, it is negative. Secondly, if the phase error happens to be greater than π, then it will become negative. Also, if it becomes less than $-\pi$, then it will become positive. Now, this is a problem in your system. So, why is it a problem in your system?

(Refer Slide Time: 24:52)



Well, I am having a phase detector, I will make this as a block rather than a mixer for now. So, you have the phase detector, loop filter and VCO. You give the input and you are looking for the output. Now, think about it, if you have a positive phase error, ideally, then you should increase the control voltage to increase the frequency and recover the phase. But here it actually depends on whether the phase error is greater than π or it is within 0 to π. So, if there is a small phase error here, this phase error is going to be processed not in a linear fashion by the phase error detector. So, if the phase error is greater than π, you see that the sign of V_{PD} changes completely which will not make the PLL to go towards lock.

So, just an illustration, we have used this example here. So, by the way, in this particular case, the phase error is lesser than π. So, let us say, when you measure this phase error here which is less than π, what happens? The phase error is positive. This is your input signal, and this is your output signal. The phase error is positive. For positive phase error, V_{PD} output or V_e is also going to be positive because we know we are in the positive part of the sine waveform. The control voltage from the loop filter will also be positive. If the control voltage is positive, what happens? The VCO frequency increases. Now, what does it mean? If the VCO frequency increases, then, let us say, this happens pretty fast, then you can say that the next rising edge should ideally come closer.

If I increase the VCO frequency, then the time period should reduce. Even though the time period reduces, you still detect the phase error here. Because you still detect the phase error, the frequency may increase further, and at some point of time, may be the very next sine wave may happen to come closer to this, the blue and red align. It should be noted that all these things depend on the other components in the PLL, but this is the thing which is going to happen in an actual PLL.

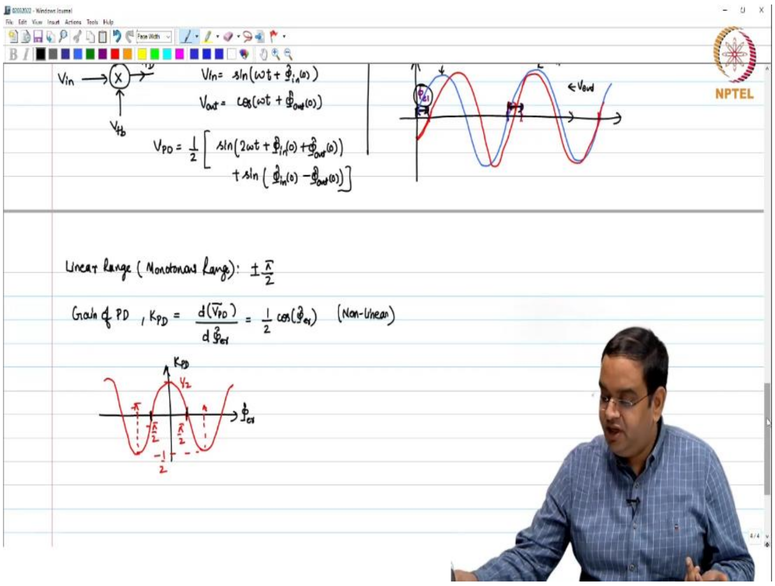
At the time when they align, the phase error is zero. If the phase error is equal to zero, the control voltage will return back to the same value. So, you will have both the input and the output sine waves following each other in this case. This is how tracking happens. Why is this important to understand? Because you had a phase error of one sign, you need to change the frequency of the oscillator in that particular direction to minimize the phase error.

But if the phase error is greater than π here, the output of the phase error detector is negative. So, you will change the frequency of the oscillator in the opposite direction. You will go away from the phase locking. So, this is what will happen in the mixer based phase error detector. It may increase the phase error and then again what will happen? When it comes, it may again go to the positive and it will try to recover the phase error.

So, these things will happen. What we would like in a linear system is that if the phase error is positive, the phase detector should give me a positive output voltage. If the phase error is negative, it should give me a negative output. Because when we have a linear output, then the system will behave in a linear time invariant manner.

Otherwise, see, we cannot say that if it becomes non-linear, it will not lock. Well, the initial locking in the PLL, we have seen a lot in a non-linear fashion. But when any change happens in the PLL, it will be well behaved if we have a linear gain for the system. So, we define the linear range of the phase detector. And by the way, this range is not the linear range. Actually, this is the monotonous range, where it is changing only in one direction.

(Refer Slide Time: 30:50)



So here, the linear range of this particular PD, which is more or less the monotonous range, you can say, is equal to $\pm\frac{\pi}{2}$. So, this is something which we want to define here. So, if the phase error is within this range, you will see that the change is in the correct direction and your PLL will lock without going in the opposite direction.

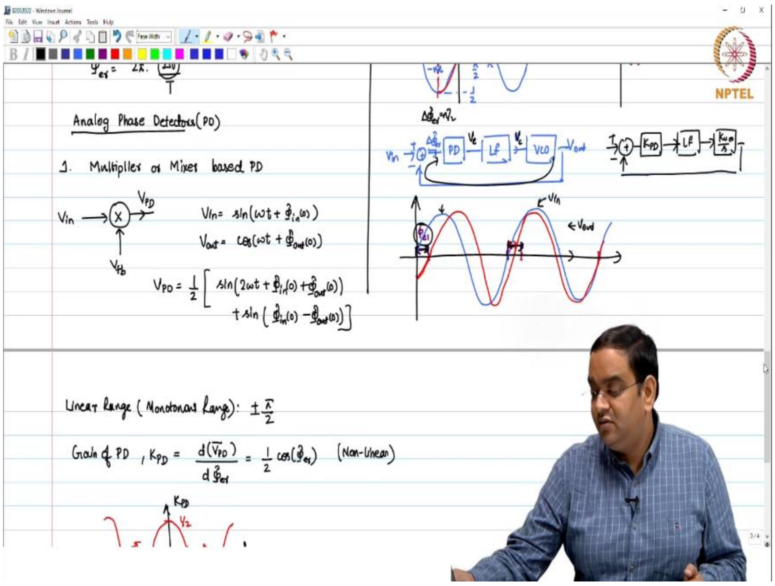
The other thing which we have defined and we will just reiterate it here. The gain of this phase detector which is denoted as K_{PD} is defined as the derivative of the average output voltage with respect to phase error. So, we have,

$$K_{PD} = \frac{d(\overline{V_{PD}})}{d\varphi_{er}} = \frac{1}{2}\cos(\varphi_{er})$$

Now, if I plot K_{PD} with respect to phase error, what you see here is that the gain K_{PD} is a cosine waveform which is like this.

We know that this is the cosine wave. So, this is $-\pi$ to π, this is $-\frac{\pi}{2}$ and this is $\frac{\pi}{2}$. This gain is $\frac{1}{2}$ here and $-\frac{1}{2}$ here. So, within the linear range which is $-\frac{\pi}{2}$ to $\frac{\pi}{2}$, what you see is that the gain of the phase error detector is not linear. This is non-linear which is another problem for the system since you cannot do any linear analysis. So, the gain depends on what phase error you are having. So, the dynamics of the PLL are going to be different. What do I mean by dynamics of the PLL?

(Refer Slide Time: 33:40)



Well, if the phase error is actually very close to $\frac{\pi}{2}$, we have seen here that the phase error detector output is maximum, but in that case, when you are applying the change, the change is going to be very small. So, with respect to $\frac{\pi}{2}$, I get $V_e = \frac{1}{2}$ and all this frequency change. It comes back. It is in feedback. You apply this feedback. You apply feedback here, that particular small change, because the gain is very small. So, you can think about it. When the PD gain is very small, in that case, the feedback will be very slow.

So, for this particular block diagram, you can draw the small signal model like this. PD with gain K_{PD}, so, it actually depends on what is the phase error at a given instant of time. So, let me just write this as $\frac{K_{VCO}}{s}$ because this is a small signal diagram. So, if K_{PD} is very small, then the loop gain is very less and the loop response will be very slow.

So, this is the problem which you see with this kind of phase error detector. Its linear range is limited between $\pm\frac{\pi}{2}$. The gain of the phase error detector is non-linear in this particular case. Well, we have been using these things. I have shown you in simulation also that the PLL locks but when we are considering the linear dynamics of the PLL, all these things are important. Because we should know whether we can apply the linear gain model or not, so, we need to know these parts. Thank you.