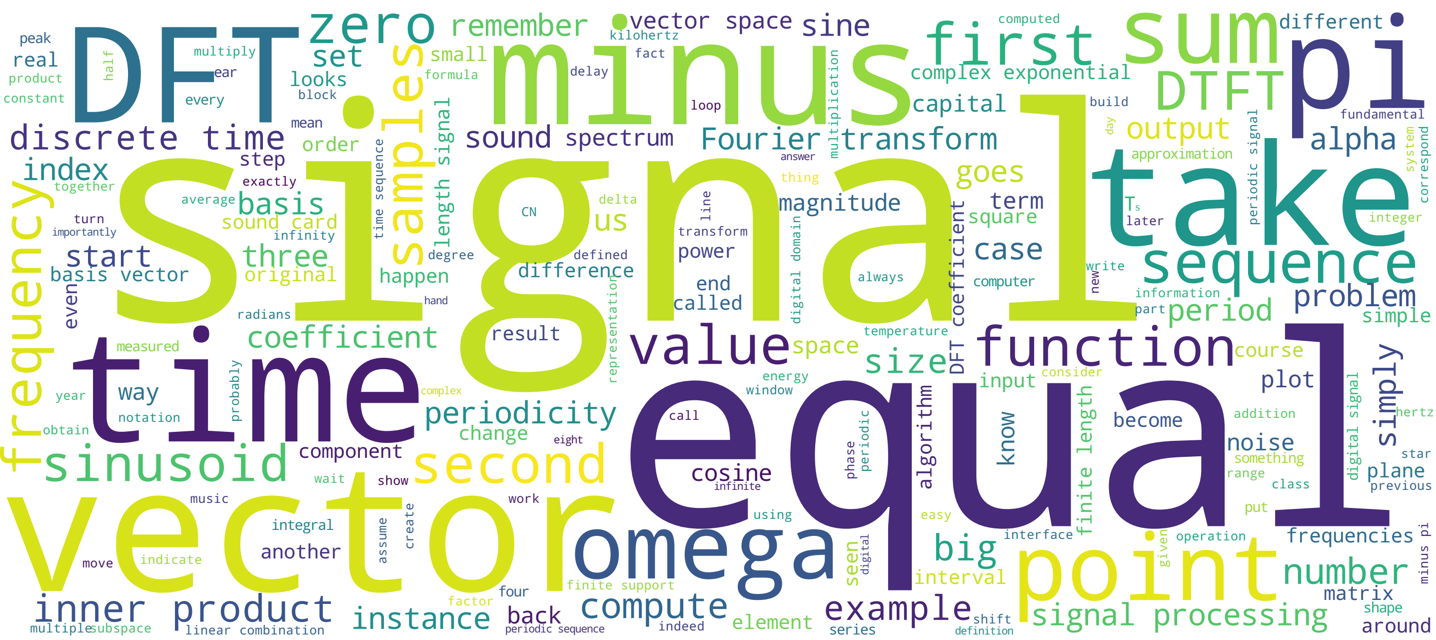
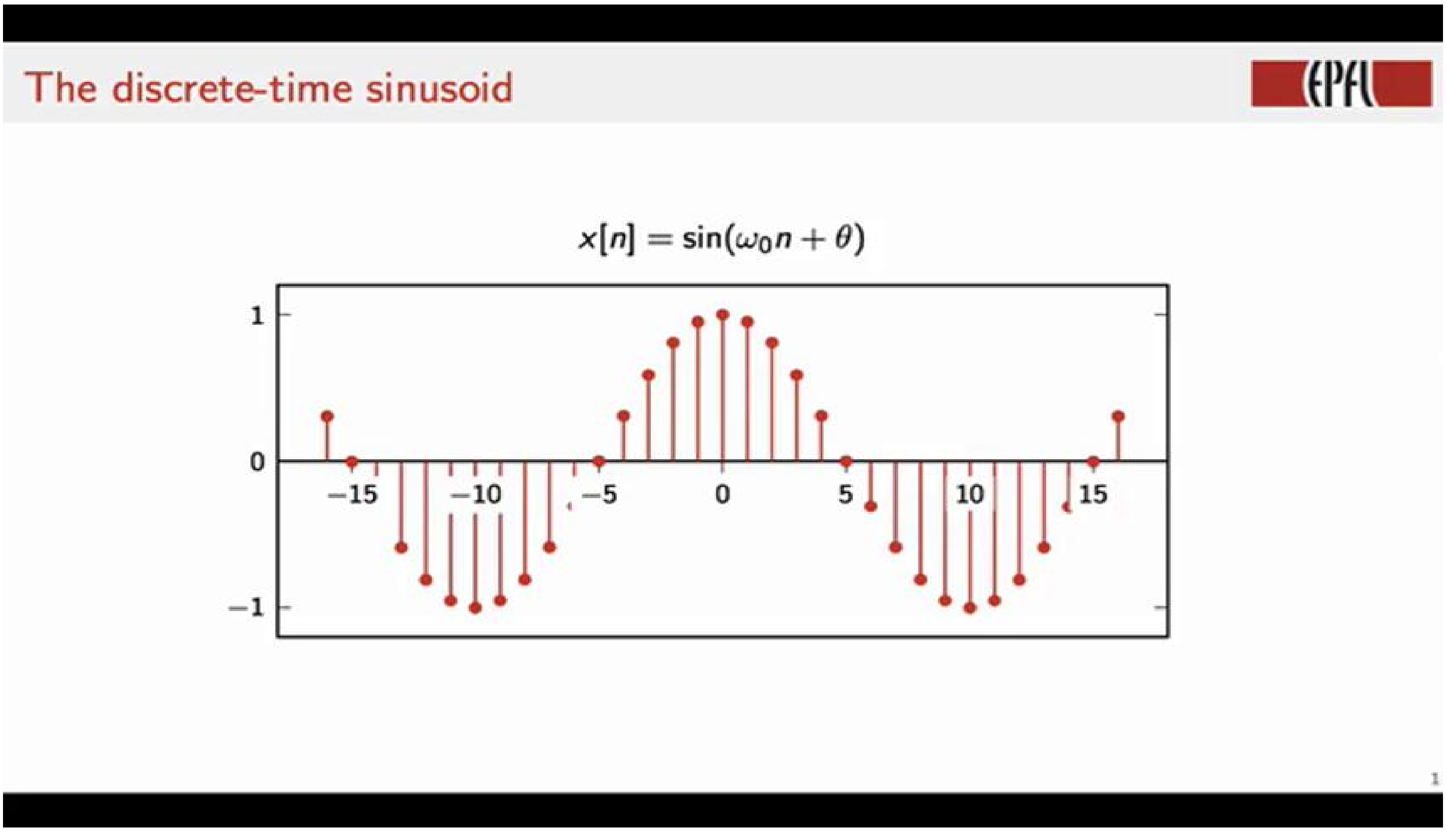
Digital Signal Processing

How your PC plays discrete-time sounds



Search MOOC

Video

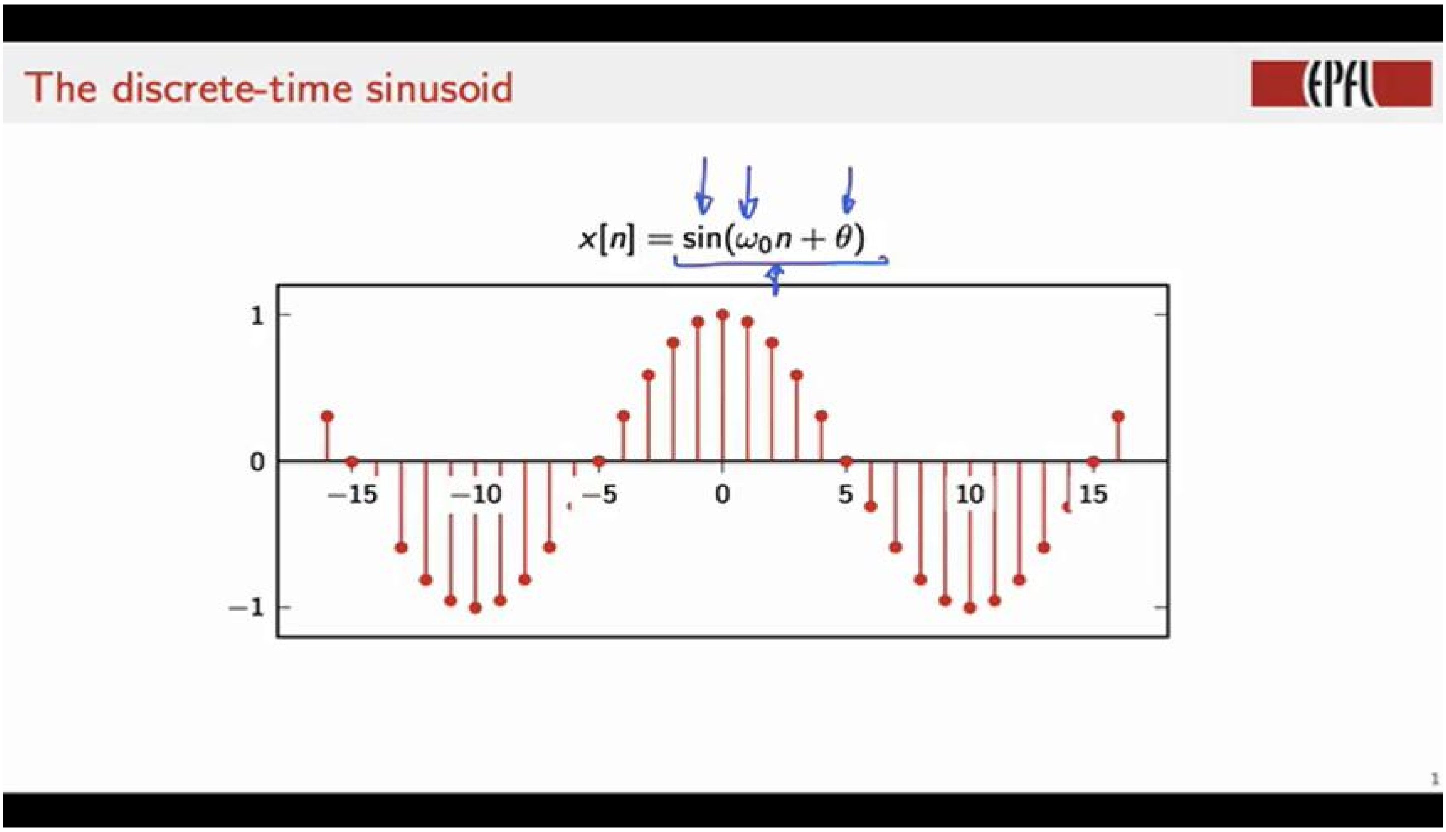


One of the greatest things about digital signal processing is that if you have a computer, and you probably do have one since you're watching these video lectures, if you have a computer, you have a fully functional signal processing lab at your fingertips. More importantly, a computer also has interfaces that will allow you to visualise the signals that you create by writing a few lines of code. You can visualise them as plots on the screen. More importantly— and that's really what got me into signal processing— you can materialise the signals as audio signals that you can hear with your ears. Now, we need an interface because our ears are analog devices, whereas the PC is a digital device. We will see later in this class the precise details of this transformation from digital to analog. Here, let's just go through a handwaving overview to give you the fundamentals of digital-to-analog conversion, so that in future modules, we will be able to use audio examples that you can listen to. Let's start with a simple digital sinusoid. You pick your favorite trigonometric function, say, a sine, a digital frequency, Omega$_0$, measured in radians, and an initial phase, if you want, again, measured in radians.

Notes

Summary

0m 00s

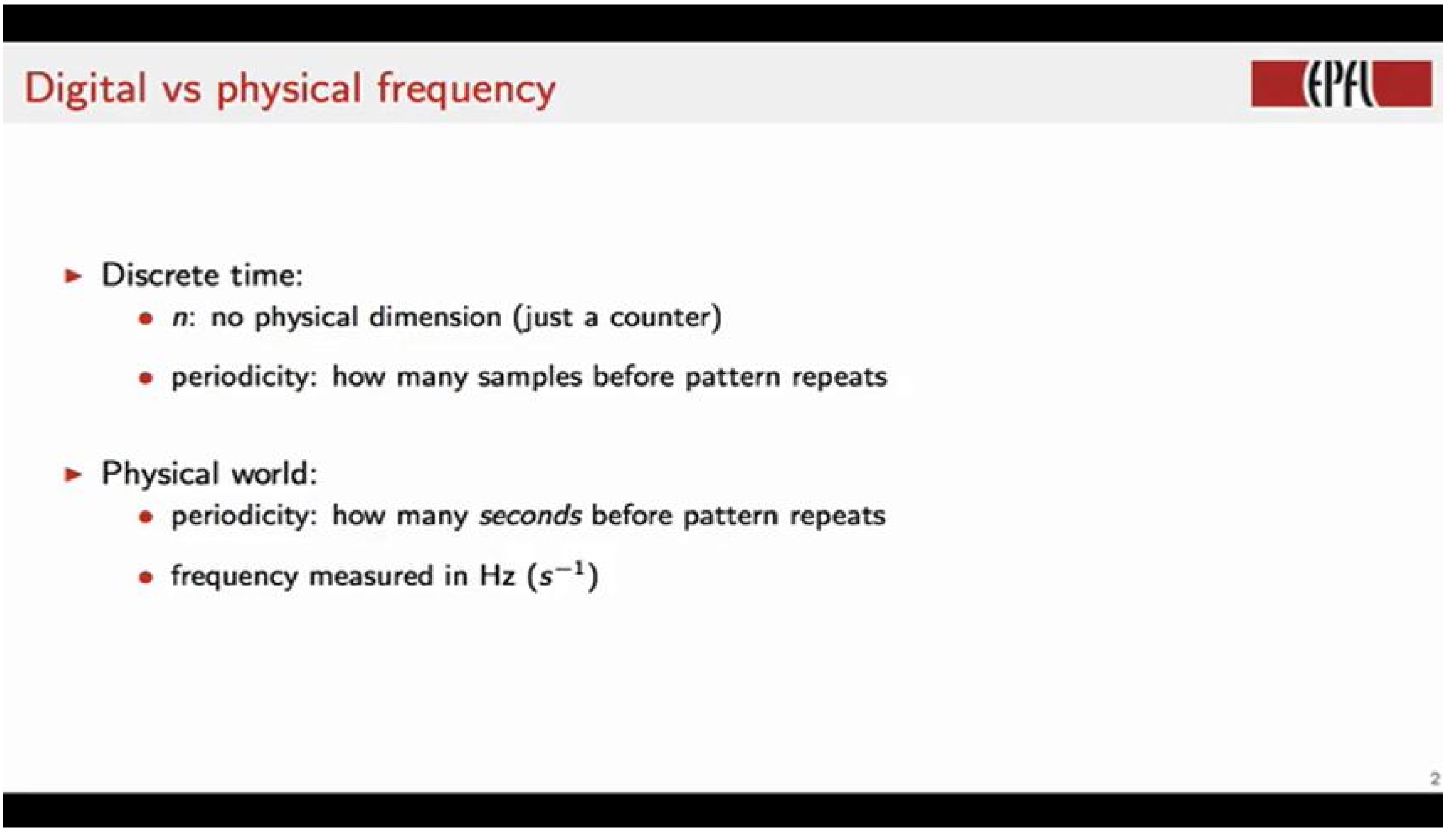


Your discrete time sequence is then computed very easily with a loop. At each step, you increment the index n and you compute the sine of Omega$_0$ n plus Theta. You will get a series of samples that looks like this. How would a sinusoid like this sound? Well, before we can answer this, we have to bridge the gap between the digital representation and the analog representation that we present to our ears.

Notes

Summary

1m 16s

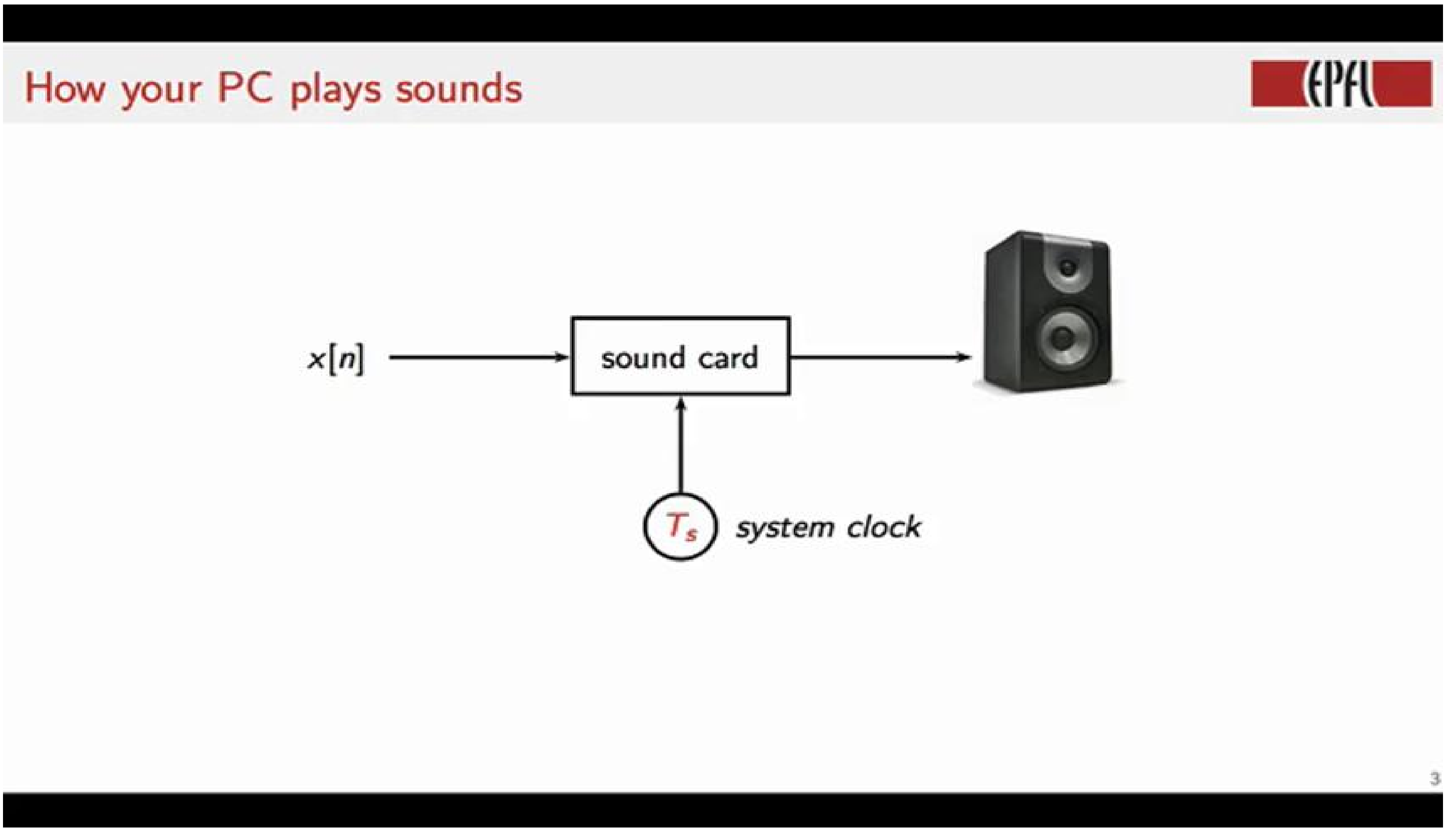


Now remember, in discrete time, n has no physical dimension. The periodicity of a waveform is determined by how many samples you have to wait before the pattern repeats. In the physical world, on the other hand, the periodicity is measured in how many seconds you have to wait before the pattern repeats, and the frequency is measured in hertz, which is a physical unit of 1 over seconds.

Notes

Summary

1m 39s



The PC bridges this gap via a sound card, a device that takes a series of samples and builds an electric signal that we can feed to the loudspeaker. At the heart of the soundcard is a system clock with a period T_s measured in seconds. This T_s is the time that we wait before we take a new sample from the discrete time sequence and feed it in the sound card.

Notes

Summary

2m 01s

Digital vs physical frequency

- set T_s, time in seconds between samples
- periodicity of M samples $\longrightarrow$ periodicity of MT_s seconds
- real world frequency:

$$f = \frac{1}{MT_s}\text{Hz}$$

4

If T_s is the time in seconds between samples, a periodicity of M samples in the digital domain will become a periodicity of MT_s seconds in the physical domain so that the real world frequency of a periodic pattern like a sinusoid in the digital domain will become 1 over MT_s hertz in the physical domain. Let's give an example.

Notes

Summary

2m 24s

Digital vs physical frequency

- usually we choose F_s the number of samples per seconds
- $T_s = 1/F_s$

E.g for a typical value $F_s = 48000$, $T_s \approx 20.8\mu s$. If $M = 110$

$f \approx 440\text{Hz}$

5

Usually, we choose F_s, the number of samples per second that we use in the sound card rather than T_s. But the relationship is very simple. T_s is equal to 1 over F_s, and of course F_s is equal to 1 over T_s. A typical value for F_s is 48 kilohertz. This is probably familiar to you because it's the sampling frequency, say, of a DVD. That T_s is about 20 microseconds. Now, assume we built a sinusoid in the digital domain with a periodicity of 110 samples. If we fed the sinusoid to a sound card at 48 kilohertz, we would get a sinusoid with a perceivable pitch of 440 hertz, which is the standard concert pitch in western music.

Notes

Summary

2m 49s