1

(read slide) I'd like to discuss this in the context of reception and detection. In this presentation I'm assuming a distortionless channel so that the signal received has no ISI (intersymbol interference) and we'll assume additive white guassian noise (AWGN) that is independent of the signal and has a zero mean value.

2

Here is a block model of a general binary signal detector. If we look at the input, we see on the far left our incoming digital form signal, $s_i(t)$.

3

We can refine the symbol of an interval from 0 up to capital T by the two designations that you see on the slide. S1 is logic 1 and S2 is logic 0. So, Si_is composed of those two signals. Also going into the adder that you see on the slide is the noise signal n(t) that we just described, the zero-mean AWGN. Notice that coming out of the adder, then, is essentially the signal we have to detect, r(t).

4

Here is the representation of r(t) from the adder, as you can see it's the sum of the binary modulated signal and the noise. Here's the big picture for you: If you look at this slide, you see the two dotted portions of the digital signal detector. Those two portions highlight the two major steps in signal detection. The first step is represented in the first dotted area labeled "block 1" and it consists of reducing the received signal that we just talked about, r(t), into a single number z(T). Notice z(T) on the right side of the switch. This operation can be performed by a linear filter followed by a sampler, like it shows you on the slide in block 1.

5

If we sample at t = capital T, then we can represent z(T) by what you see here. It the sum of the first term, which is just the signal component part and the second term, which is just the noise component part. We can call the signal component part $a_i(T)$ like you see in the equation and $n_0(T)$ the noise part, also like you see in the equation.

6

Lots of times you'll see this represented in shortened form like you see on this slide, where we just understand that the sampling time is the same for all parts. One thing to notice here is the noise component. Let's look at it for a second. The little zero in the subscript is just to remind us that this noise component is a Gaussian random variable with a zero mean. With that component as part of the equation, it makes the entire sum, z, a Gaussian random variable. And since *n* has a mean of zero, then *z*'s mean will depend solely on a_i.

7

Now remember, there were two cases for our signal, s1 when we sent a logic 1 and s2 when we send a logic 0. This will translate, like it says on the slide, into a1 or a2. Also, *z* is what is known as the "test statistic".

8

The second step in the binary signal detection process consists of comparing the test statistic *z* to the threshold level, called alpha, in the threshold comparator in block 2. The last step is to make the decision. So what is the decision we make?

9

If you're familiar with hypothesis testing in statistics, that is actually what we are doing. But even if you aren't, I think you can understand at least graphically what is going on. Let's start with what is shown here. H_1 is called "hypothesis 1" and H_2 is called "hypothesis 2". All that really means is we either sent a 0 or a 1. Choosing H_1 is the same thing as deciding that signal $s_1(t)$ which was the logic 1 was sent. Choosing $s_2(t)$ is the same thing as deciding that signal $s_2(t)$ was sent, and that was a logic 0. You can see on the slide how we make such a decision. H_1 is picked if *z*, the test statistic is greater than lambda and H_2 is picked if *z*, the test statistic is less than lambda. If $z = \lambda$ the decision is arbitrary, it can go either way. Why is it not totally apparent which one was sent based on the received signal? The answer is that the Gaussian noise increases the probability of error.

10

Here is a good picture of the problem. Remember that in the beginning before the noise was added, a_1 and a_2 were clearly defined values and you can see them market on

the horizontal axis. The problems started when we added noise, the AWGN, or additive white Gaussian noise, and a Gaussian distribution is the same thing as a normal distribution which is the same thing as the bell-shaped curve. The AWGN signal all by itself had a mean of zero. What happened in each case is that a new mean was created when a1 and a2 were both separately translated onto the distribution, so now there are two Gaussian distributions with two means that are close enough together that a significant part of the curves overlap, like you see here. Notice that right in the middle is lambda, which is what we're comparing z with to make our decision, but did we make the right one? Let's look at some situations.

11

Here we see something that I've labeled "case 1". The z is so far below the threshold level that with little probability of error we can conclude that a logic 0 was sent (which corresponds to $s_2(t)$ in the beginning. There is still a minute probability for error, but it is negligibly small. Why? It's because more errors occur the more overlap there is in the Gaussian curves. You can see that even though the flaps in each one extend in both directions to infinity, at some point you just say, well there is such a tiny, tiny part of the curve on the right overlapping where my z value is, we can just say there's no overlap for all practical purposes, and definitively conclude that logic 0 was sent and proclaim no error.

12

What you see here is a 2nd case, which is the corresponding case for what we just looked at when z is significantly greater than lambda. We can pretty much proclaim that a logic 1 was sent in this case, and the error is zero.

13

But what about here? Notice that because of noise, now *z* is close to lambda. There is significant overlap. Even though z is less than lambda, accepting hypotheses H2, which remember is like saying that a 0 was sent, has a high probability of error. It is possible that a1 was actually sent, but the noise caused the signal to extend below lambda leading us to a false conclusion. The greater the overlap, the greater the probability of error.

There are several types of detectors that process this statistical information in various ways to make conclusions. One of them is called the “maximum likelihood detector”. There are also ways of using devices such as a “matched filter” to provide maximum SNR for a transmitted signal and other means to reduce the bit error. Most of them deal with probability distributions and at that point become probability mathematics, but they all rely on and process the information that you have just been shown.