LDPC and Polar codes in 5G Standard

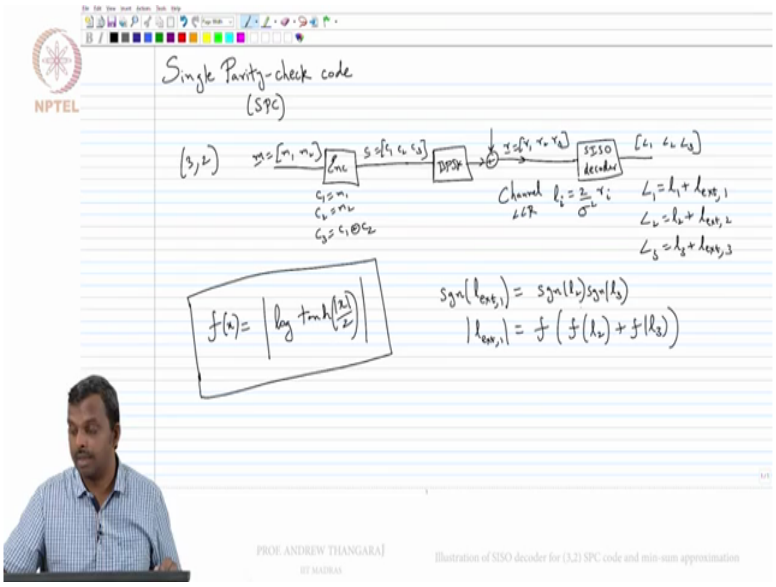
Professor Andrew Thangaraj
Department of Electrical Engineering
Indian Institute of Technology, Madras

Illustration of SISO decoder for (3,2) SPC code and min-sum approximation

Hello, in the previous lecture we saw SISO Decoder Soft In Soft Out Decoders for a (()) (00:23) code and single parity check code. There was some interesting calculation we did maybe some of it was little complicated thing, the most important thing is to understand how the SISO decoder works and how to implement it, that is sort of the focus of this code but once in a while it is also good to see where some of this things come from such intuition is good and debugging etcetera some times, ok.

So let us proceed a little bit more on that, so I want to summarize quickly what this SISO decoder does for the single parity check code, so that is the most important ingredient that we will use again and again in the later part of the course, ok.

(Refer Slide Time: 01:02)


Single Parity-check code (SPC)

$(3,2)$ $m = [m_1\ m_2]$ Enc $s = [c_1\ c_2\ c_3]$ BPSK $r = [r_1\ r_2\ r_3]$ SISO decoder $[L_1\ L_2\ L_3]$

$c_1 = m_1$
$c_2 = m_2$
$c_3 = c_1 \oplus c_2$

Channel LLR $l_i = \frac{2}{\sigma^2} r_i$

$L_1 = l_1 + l_{ext,1}$
$L_2 = l_2 + l_{ext,2}$
$L_3 = l_3 + l_{ext,3}$

$$\boxed{f(x) = \left|\log \tanh\left(\frac{|x|}{2}\right)\right|}$$

$$\text{sgn}(l_{ext,1}) = \text{sgn}(l_2)\,\text{sgn}(l_3)$$
$$|l_{ext,1}| = f\left(f(l_2) + f(l_3)\right)$$

PROF. ANDREW THANGARAJ
IIT MADRAS
Illustration of SISO decoder for (3,2) SPC code and min-sum approximation

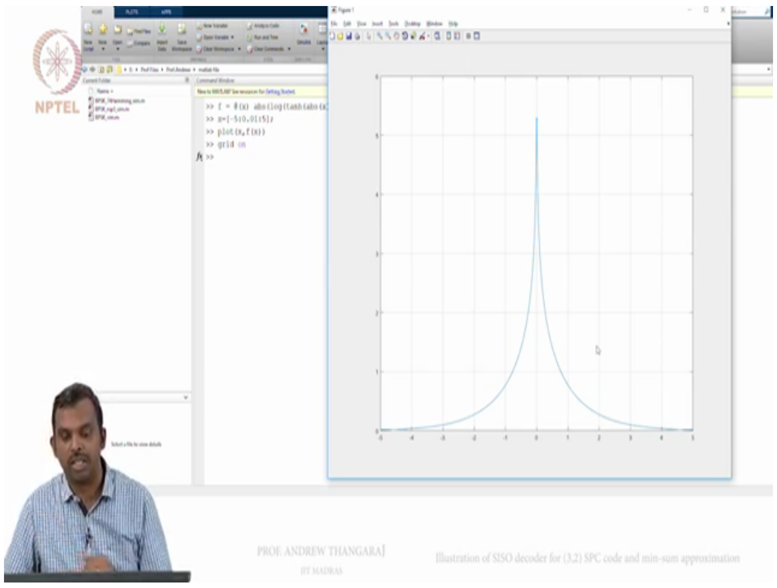
So let be quick summarize, single parity check code or SPC code, ok I took the 3 comma 2 example, so let me first do the 3 comma 2 example, 3 comma 2, so we had two messages m 1 and m 2 and this get us encoded into a code word which is c 1, c 2, c 3 how does the encoding happens c 1 equals m 1, c 2 equals m 2 and c 3 equals c 1 XOR c 2, right so the overall parity is maintain here and after this you do BPSK symbol by symbol transmission over the Gaussian channel and then you get your receive vector r, you remember this r we thought of as a belief itself the channel LLR is like the r and then the SISO decoder produces intrinsic, extrinsic and then the total, right.

So how does that work? So it produces capital L 1, capital L 2, capital L 3 the input small l 1, small l i is 2 times 2 by sigma squared times r i, this is the channel LLR , we saw this derivation in the last class and then there was the this function f of x which was log tan hyperbolic modulus of x by 2, ok and then we saw the this capital L 1 is going to become small l 1 plus l x 1 in fact the same things happen for everything l 2 plus l x 2 and L 3 is l 3 plus l x 3 and then what is each of this l x 1 is? So the l x 1, ok.

We describe the sign separately and the absolute value separately, the sign is the sign of l 2, times sign of l 3, right and then the absolute value is given by f of l 2 plus f of l 3, ok. So in the decoder for the calculation of l extrinsic the function s is involved, so this f of x is log tan hyperbolic of x by 2 and I put absolute value here and one more thing to keep in mind is tan hyperbolic is something that is going to go less than 1, so log of that will be negative and we want f to deal with positive values, so we will put an absolute value around it, ok so this is the definition of f of x.

There is an absolute value around the log tan hyperbolic as well, ok. So once you make this definition, this is fairly straightforward to see, the absolute value of l x 1 is f of l 2 plus f of l 3 and the sign of l x 1 is product of the two sides, ok. So let us see what this function is and plotted and see how it behaves in mat lab, ok. So we going to go there, so that we go to my mat lab screen.

(Refer Slide Time: 04:50)



```
>> f = @(x) abs(log(tanh(abs(x
>> x=[-5:0.01:5];
>> plot(x,f(x))
>> grid on
>>
```

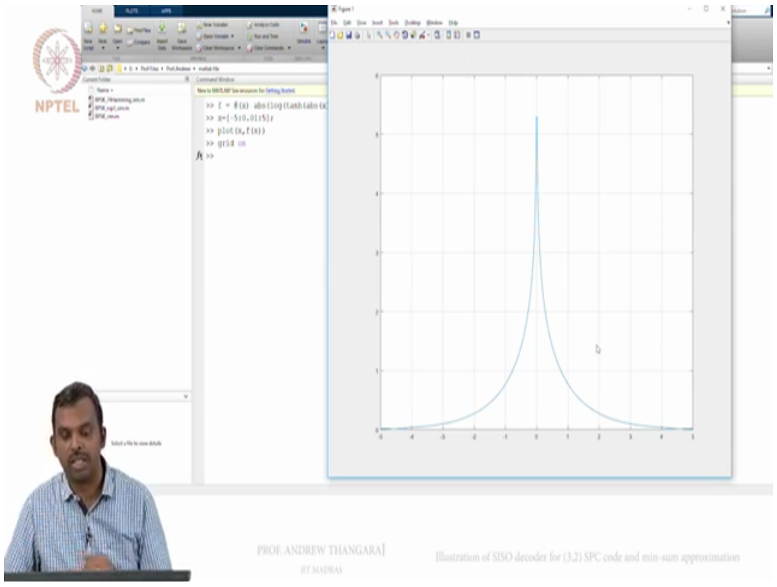
PROF. ANDREW THANGARAJ
IIT MADRAS
Illustration of SISO decoder for (3,2) SPC code and min-sum approximation

So first thing we will do is, we will define this function f which is a like I said absolute value of log of tan hyperbolic of absolute value of x by 2, ok so that is the function f and then one can define x and this range minus 52 plus 5 and then you can plot of x m of f of x, ok and you can see how it works and let me show you as likely zoomed up washing up this window, zooming I can get it to work , yeah really it is and maybe I should get the axis on, ok and you can see how this works.

So as a function of x this function f falls very steeply, ok for very low values of x this function f is very high, ok and as x increases in value, this function falls quite steeply. So even if you look at the value of f of 2 it is quite low, ok and the other hand values of f close to zero are really high 3, 4 and all that, ok. So this gives you an idea for how to appropriate f, ok in case you want to do a simple calculation it is nonlinear function.

So if you want to appropriate f one sort of simple thing to do is tent to give importance to the dominants values based on the argument, ok. So if f of x if x is very small then f of x is very large, ok if x is slightly larger than f of x becomes very small, ok.

(Refer Slide Time: 06:25)



```
>> f = @(x) abs(log(tanh(abs(x
>> x=[-5:0.01:5];
>> plot(x,f(x))
>> grid on
>>
```

PROF. ANDREW THANGARAJ
IIT MADRAS
Illustration of SISO decoder for (3,2) SPC code and min-sum approximation

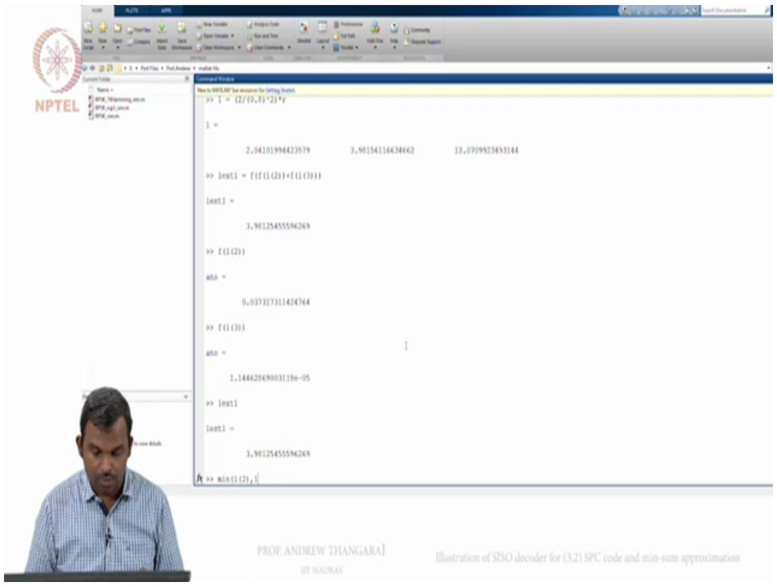
So using this one of the very popular approximation for this kind of calculation is if you have f of l 2 plus f of l 3, this is approximately f of min of absolute value of l 2, absolute value of l 3, ok so because remember what I told you this f function when the argument becomes large f of x, when x is large becomes very small, ok so out of l 2 and l 3 whichever smaller in absolute value is going to dominate the sum f of l 2, f of l 3, ok.

So f of l 2 plus f of l 3 is dominated by the minimum of the two values, so f of min of absolute value of l 2 comma absolute value l 3 is a good approximation, ok so that is this is once again it is approximate is not exact but it is a nice approximation. There is an a s 1 is you make this approximation absolute value of l x 1 has this very simple expression f of min of absolute value of l 2 absolute value of l 3 and we know that f is it is own inverse f of x of x itself and so this will just become min of absolute value of l 2 comma absolute value of l 3.

So this approximation is called the Min sum approximation and it is very popular in all implementations of this kind of extrinsic calculation, ok instead of looking at a log tan hyperbolic and all of that we can simply look at the minimum of those 2 values, so if you want to do f of l 2 plus f of l 3 you look at the min of l 2, l 3, ok and then simply take as that take that as the l x 1, ok.

So is much simple calculation then doing the nonlinear log tan hyperbolic and all that, ok simply minimum of absolute value of l 2 and absolute value of l 3, ok. So this means some approximation is quite a bit popular, so I want to maybe illustrate that a little bit in mat lab, ok.

(Refer Slide Time: 08:32)



```
>> l = (2/(0.8)^2)*r

l =

        2.04101994423579        3.90156116638662        12.0709923493144

>> lext1 = f(f(l(2))+f(l(3)))

lext1 =

        3.90125455596269

>> f(l(2))

ans =

       0.0373173114247764

>> f(l(3))

ans =

     1.14462869903110e-05

>> lext1

lext1 =

        3.90125455596269

>> min(l(2),l
```

PROF. ANDREW THANGARAJ
IIT MADRAS
Illustration of SISO decoder for (3,2) SPC code and min-sum approximation

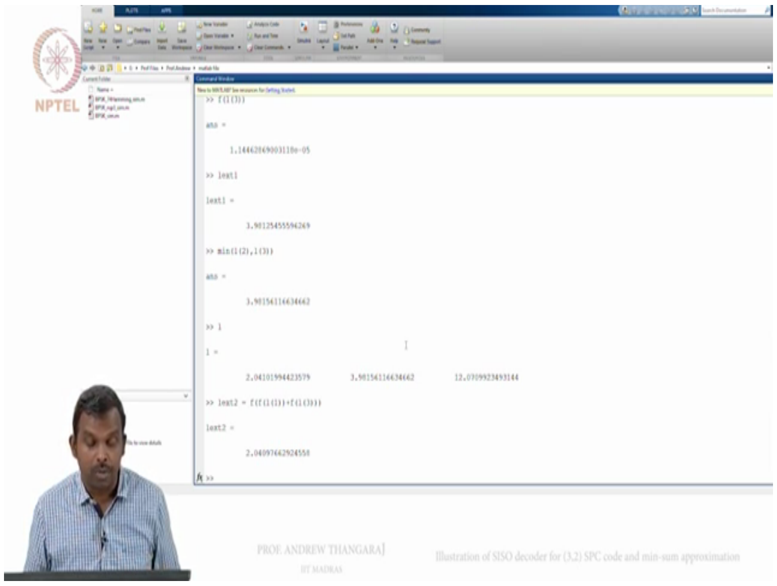
So to illustrate how this approximation works, so let us look at some received values, so I will consider received values of this form 1 plus I will put a noise variance or may be the noise variances point 9 or point 8 something, ok and then random noise 1 comma 3, ok so this gives me 3 received values, you can see the first 2 are greater than 1 and the last one is gone negative, so there is some error etcetera, maybe we will consider one more, ok.

So this is going a little bit high, yeah but his should be fine, ok so this is a good received value I do not want to have some error in the value, so I am picking some other value here. So next first step is to calculate the channel LLR, if you remember the channel LLR is 2 by sigma square, sigma I have picked here as point 8, ok. 2 by sigma square times r this is the channel LLR and you can say it has a slightly larger value, so it goes to 2, 3 and 12, ok.

Now a let us illustrate what happens in l x 1 calculation, so if you look at l x for the first one, ok so this is going to be the sign is going to be the product of the sign is of l 2 and l 3, so in this case both l 2 and l 3 are positive, so you see l 2 and l 3 are positive, so one the sign becomes just positive, so the sign of l x is positive, so l x 1 comes out to be f of l of 2 plus f of l of 3, ok so that is the answer, ok.

So if you want to look at the calculation closely f of l of 2 became point 03 and f of l of 3 became sometime into 10 power minus 5, ok so like I said if you look at of l of 2 is much smaller than l of 3, so the minimum value of these two is dominating the calculation, ok so that is you got and no wonder the l x 1 is so close to the min of l of 2 comma l of 3, ok so you see both of these are really close, ok.

(Refer Slide Time: 10:53)

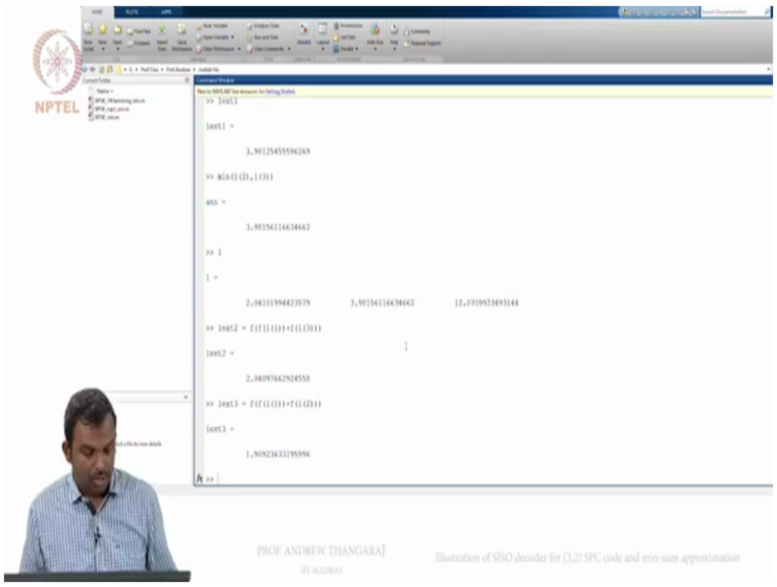


```
>> f(l(3))

ans =

   1.14462869003110e-05

>> lext1

lext1 =

          3.90125455596269

>> min(l(2),l(3))

ans =

          3.90156116634662

>> l

l =

          2.04101994423579          3.90156116634662          12.0709923493144

>> lext2 = f(f(l(1))+f(l(3)))

lext2 =

          2.04097662924558

>>
```

NPTEL
PROF. ANDREW THANGARAJ
IIT MADRAS
Illustration of SISO decoder for (3,2) SPC code and min-sum approximation

So now l x 1 is the (nos) not the only thing he will compute, he will also compute l x 2, is not it l x 2 is once again all three l is are positive, ok if you look at the value l again all of the three l is are positive, so the sign becomes positive throughout sign of l 1 into sign of l 3 is also positive, so if you want a compute l x 2 in a absolute value this is simply f of l of 1 plus f of l of 3 is not it, this is the formula we saw before and once again this you can see is very close to the minimum value it is 2 point 04 and then 12 point 07 really that f of l of 3 does not contribute anything it goes to 10 power minus 3.

(Refer Slide Time: 11:37)



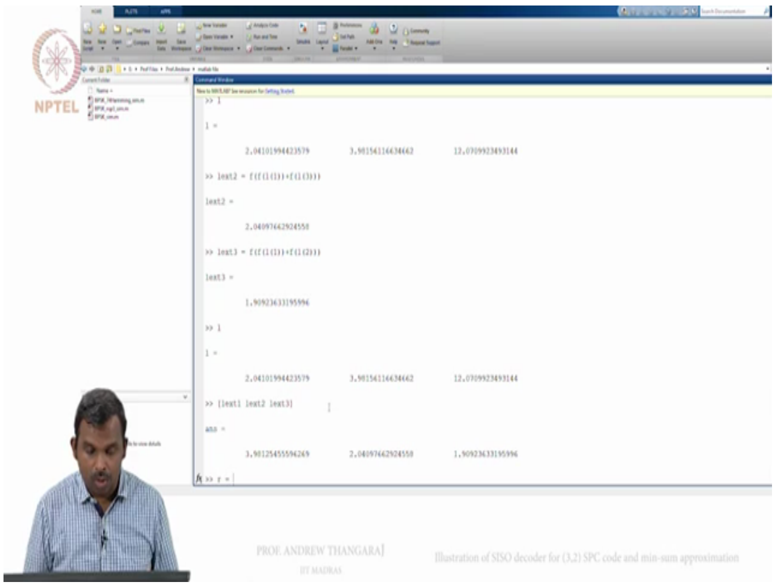
```
>> lext1

lext1 =

          3.90125455596269

>> min(l(2),l(3))

ans =

          3.90156116634662

>> l

l =

          2.04101994423579          3.90156116634662          12.0709923493144

>> lext2 = f(f(l(1))+f(l(3)))

lext2 =

          2.04097662924558

>> lext3 = f(f(l(1))+f(l(2)))

lext3 =

          1.90923633195996

>>
```

NPTEL
PROF. ANDREW THANGARAJ
IIT MADRAS
Illustration of SISO decoder for (3,2) SPC code and min-sum approximation

On the other hand if you look at l x 3, you see it is a little bit more difficult because l of 1 and l of 2 are close to each other, ok so but let us see what we can do here, if you do this you get something close to 1 point 9, ok so it is not too bad it is sort of close to 2 but not that close it is away by a like point 1 or something like that but something we can live with, ok. So this approximation of f of l 1 plus f of l 2 by the minimum of absolute value of l 1, l 2 it is a very powerful approximation, it is simplifies the decoder in many ways and in fact all practical decoders more or less use this min sum approximation, ok.

So there are some modifications to the algorithm, sometimes the people approximate the min sum using this offset min sum method but let us not worry too much about that for now this approximation is very useful, ok so let us go back to the picture we had, we going to approximate the absolute value in this fashion, ok so I am going to show also maybe one more illustration for what happens when there is an error, ok.

(Refer Slide Time: 12:52)



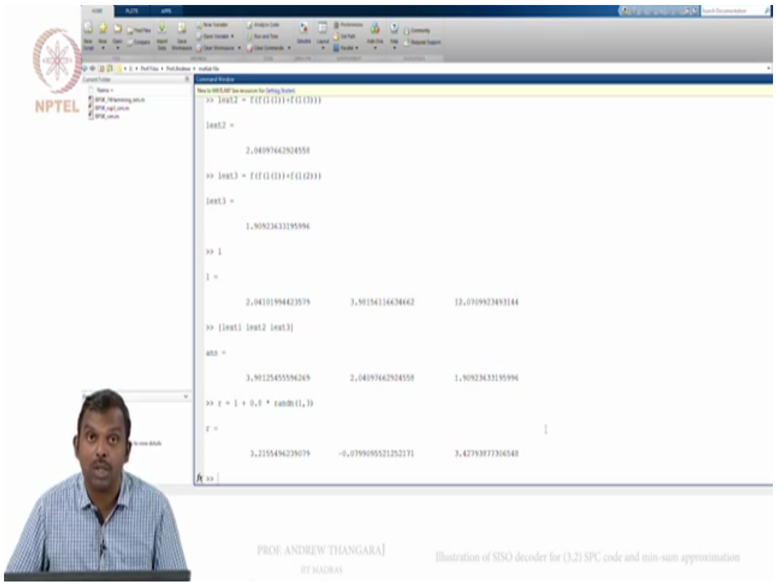
```
>> l

l =

        2.04101994423579        3.90154116634662        12.0709923493144

>> lext2 = f(f(l(1))+f(l(3)))

lext2 =

        2.04097662924558

>> lext3 = f(f(l(1))+f(l(2)))

lext3 =

        1.90923633195996

>> l

l =

        2.04101994423579        3.90154116634662        12.0709923493144

>> [lext1 lext2 lext3]

ans =

        3.90125455596249        2.04097662924558        1.90923633195996

>> r =
```

NPTEL
PROF. ANDREW THANGARAJ
IIT MADRAS
Illustration of SISO decoder for (3,2) SPC code and min-sum approximation

So overall if you look at l x 1, if you look at value of l and l x 1, l x 2 and l x 3, so you can see what is happened here, so you see the first extrinsic LLR is the minimum of the second and third, ok. So first 1 is the minimum of the second and third, the second one 2 point 04 is actually the minimum of first and third and the third one is roughly close to the minimum of first and second, ok.

So it is sort of close to the minimum of first and second not exactly equal but it is roughly close, ok. So finding the minimum is good idea, ok.

(Refer Slide Time: 13:38)

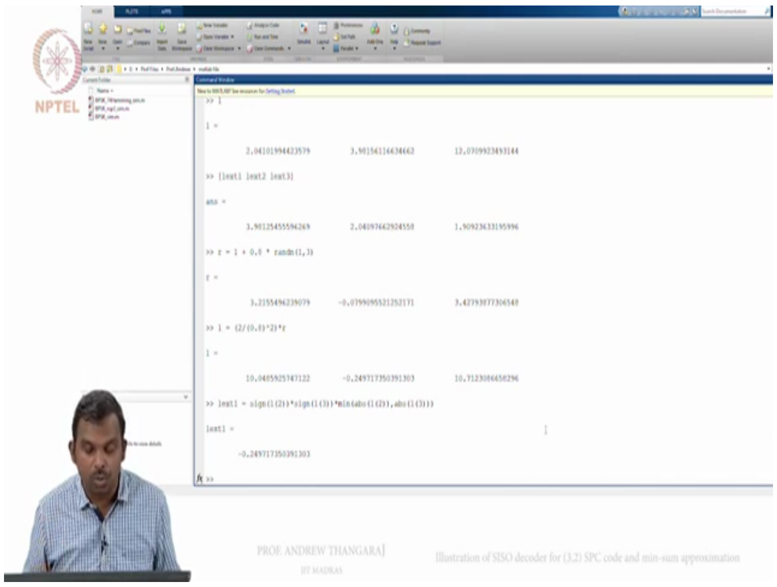


```
>> lext2 = f(f(l(1))+f(l(3)))

lext2 =

        2.04097662924558

>> lext3 = f(f(l(1))+f(l(2)))

lext3 =

        1.90923633195996

>> l

l =

        2.04101994423579        3.90154116634662        12.0709923493144

>> [lext1 lext2 lext3]

ans =

        3.90125455594269        2.04097662924558        1.90923633195996

>> r = 1 + 0.8 * randn(1,3)

r =

         3.2155496239079      -0.0799095521252171        3.42793077306548

fx >>
```

PROF. ANDREW THANGARAJ
IIT MADRAS
Illustration of SISO decoder for (3,2) SPC code and min-sum approximation

So let us try to do this one more case where things may be are little bit more difficult, ok there it is, so this is a received value where there is one error, ok so notice what is happen at transmitted 1 ,1, 1, ok three values 1, 1, 1. The first one had no error, the second one has an error, ok second one has an error, the third one has no error, ok. So now what happens in the decoding is when you try a decode the second bit, ok you will go wrong, right so because this is already gone negative so if you just do a threshold decoding, you going to decode that this is minus 1 on the other hand this is actually plus 1, ok.

(Refer Slide Time: 14:23)

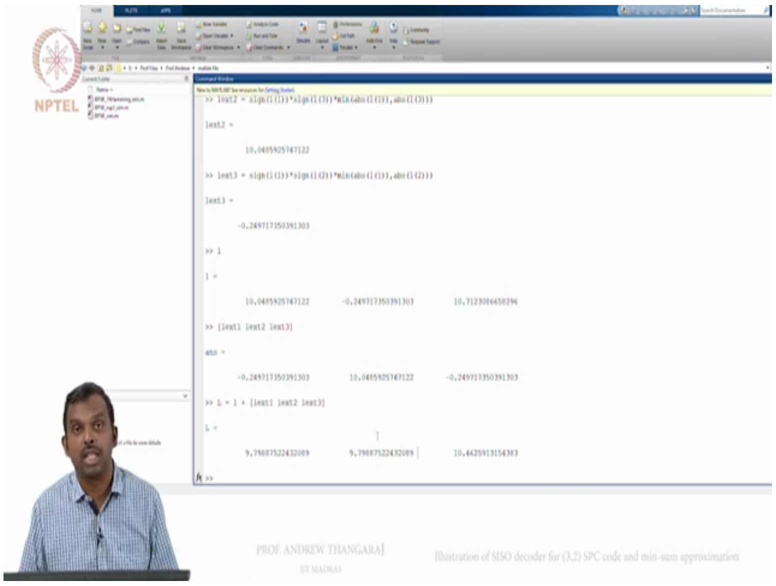


```
>> l

l =

        2.04101994423579        3.90154116634662        12.0709923493144

>> [lext1 lext2 lext3]

ans =

        3.90125455594269        2.04097662924558        1.90923633195996

>> r = 1 + 0.8 * randn(1,3)

r =

         3.2155496239079      -0.0799095521252171        3.42793077306548

>> l = (2/(0.8)^2)*r

l =

        10.0485925747122       -0.249717350391303        10.7123086458296

>> lext1 = sign(l(2))*sign(l(3))*min(abs(l(2)),abs(l(3)))

lext1 =

      -0.249717350391303

fx >>
```

PROF. ANDREW THANGARAJ
IIT MADRAS
Illustration of SISO decoder for (3,2) SPC code and min-sum approximation

Now let us see how this capital L 1, capital L 2, capital L 3 calculation goes and we will use the min sum approximation, so this L x, so you need three values here and remember once again the sign also matters, ok the sign matters as well, so L x 1 is the sign of L of 2, ok so first calculation is L equals 2 by sigma square, so we got the LLR, channel LLR. The next calculation is for L x 1, so now the sign I have to take care of, sign of L of 2 multiplied by sign of L of 3 and then multiplied by the absolute value, for the absolute value I can use the f function or I can use the min function min of abs of l of 2 comma abs of l of 3, so there you go, that is l x 1, so the extrinsic LLR for the first bit is minus 2 point 4, so you can see it is sort of close to this the minimum value that you have, ok.

(Refer Slide Time: 15:38)



```
>> lext2 = sign(l(1))*sign(l(3))*min(abs(l(1)),abs(l(3)))

lext2 =

         10.0405925747122

>> lext3 = sign(l(1))*sign(l(2))*min(abs(l(1)),abs(l(2)))

lext3 =

       -0.249717350391303

>> l

l =

         10.0405925747122       -0.249717350391303         10.7123086650296

>> [lext1 lext2 lext3]

ans =

       -0.249717350391303         10.0405925747122       -0.249717350391303

>> L = l + [lext1 lext2 lext3]

L =

         9.79087522432089         9.79087522432089         10.4625913154383

>>
```

NPTEL
PROF. ANDREW THANGARAJ
IIT MADRAS
Illustration of SISO decoder for (3,2) SPC code and min-sum approximation

So if you want to do L x 2 you can take this first l x 1 expression and sort of modified L x 2 will be sign of L of 1, L of 3 absolute value of L of 1, so again it is 10 point 04 it is sort of close to both the values close to each other but nevertheless we taking min, ok and the L x 3 is sign of L of 1, L of 2 absolute value of L of 1, L of 2, ok so once again minus 2 point 4, ok so we had L and the extrinsic LLR is L x 1, L x 2, L x 3 is this and the total LLR is L plus L x 1, L x 2, L x 3 and that will come out as positive, ok.

So we saw that the initial channel LLR gives you a negative value for the second bit, ok but after you have done the processing you have got positive values throughout and the error is corrected, now only that you got very high values for the output LLR, so you have a lot of belief that these values are correct, ok. So this was a simple illustration of two things, first how the single parity check decoder works, the SISO decoder works, you have received values, you convert them into LLR is and then you use this f function and do some nonlinear calculation, you get one decoder or you do this approximation the min sum approximation and we see that the beliefs get updated very nicely.

The channel LLR get us updated with this extrinsic LLR and you get an output LLR capital L as we showed here and this tent to be you have a very good. Now you can repeat this father received values as well and see how this works to get some more intuitions also showed you how the function f is sort of grows very rapidly, so you have to only look at mid falls very rapidly I am sorry the function f falls very rapidly, so you have to only look at lower values of the argument of the function f, ok.

So as far as this lectures concern I will stopped and in the next one, we will see how to extend the same idea for the single parity check code, SISO decoder to larger number of bits in the code word, so instead of 3 comma 2 supposing I have 10 comma 9 or 8 comma 7, how does the same idea work, ok so we will do that in the next lecture, ok.