Spread Spectrum Communications and Jamming

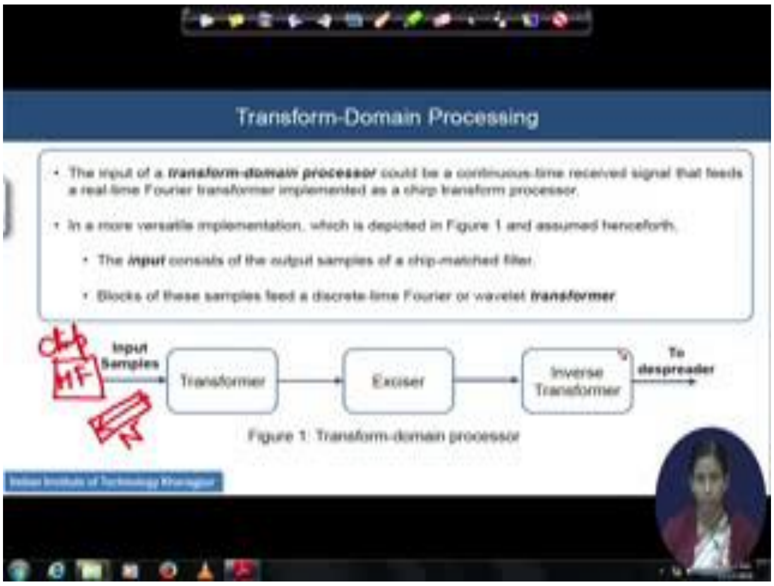
Prof. Debarati Sen
G S Sanyal School of Telecommunications
Indian Institute of Technology, Kharagpur

Lecture - 30
Narrowband Interference Cancellation by Transform-domain Processing

Hello students. We were discussing about the narrowband interference cancellation mechanisms in the spread spectrum receivers. In that context, in the last module, we have already discussed about the time domain adaptive filter architecture by means of which how we can reject it. We also saw that time domain adaptive filters can efficiently reject the narrowband interference, if the adaptive algorithm converges.

And hence there is a possibility that if the interference is dynamic in nature, I mean it is time dependent interference is time dependent, then time domain adaptive filter may not be able to a good approach because they cannot trace the time dependent interference, so is the today's discussion. Let us see actually how we can try track the dynamic or time dependent interference, and the architecture that can do it the methodology that can be applied to trace it is a transform-domain processing.

(Refer Slide Time: 01:35)



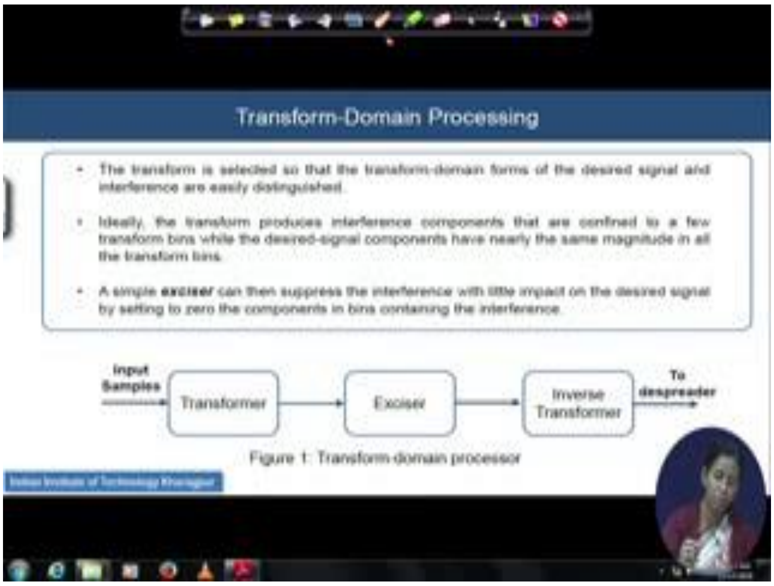
Transform-domain processing is basically they it involve a transformer or a transform-domain processor. I hope you remember that this processor or this processing block in the receiver is coming in between the matched filter and the de-spreading unit. We saw in diagram in a last module that the incoming signal after getting multiplied with the local carrier frequency, it is coming down, and then it is passing through the match filter and the output of the match filter is sampled and that simple that sample signal is now entering into the processing block. But the processing block which is helping us to cancel the narrow band interference and the output of this processing block will be fed into the de-spreading block.

So, like inside the processing block according to the last module, you are either utilizing the time domain adaptive filter or the today's discussion that is the transform-domain processing. So, as the name suggests this architecture will consist of a transform-domain processor, and where actually inside the processor at input of the processor, you can give the continuous domain signal. And it feeds a real-time some transform-domain processor and it that transform-domain processor may be a real-time Fourier transformer which is implemented as a chip transform processor. And to be practical in sense, the input that is coming the real sample values that are coming as input to the transform-domain processor that samples basically coming from the match filtered, and this match filter is a chip-matched filter.

We understand that in the receiver direct sequence spread spectrum receiver the match filter is matched to the chip waveform. And this inputs hence they are consisting of the output samples of the chip-matched filter. And as a block of these chips from the match filter certain block and the block of the chips will be entering into the transformer. Here we consider that the block of the chips is a size of capital N. And the transformer which is denoted the inside this block, it may be real-time Fourier transformer or it may a wavelet transformer. This Fourier transformer may be a discrete-time Fourier transformer also and or the wavelet transformer can back and come into picture here.

So, this is the fundamental block. After the transformer, we are having a block called exciser; output of the exciser will be fed into inverse transformer. And here actually output of the inverse transformer be fed to the despreader. Remember transformer will transform the domain of the incoming signal the changes the domain of incoming signal. Exciser is fundamentally the block who will help us to reject the interference or suppress the interference in the transformed domain. And once actually the interference suppression is done, inverse transformer is in the place to bring the samples back to the original domain all. So, how the transformer what is the target of designed for the transformer is we will see in the next slide.

(Refer Slide Time: 05:12)

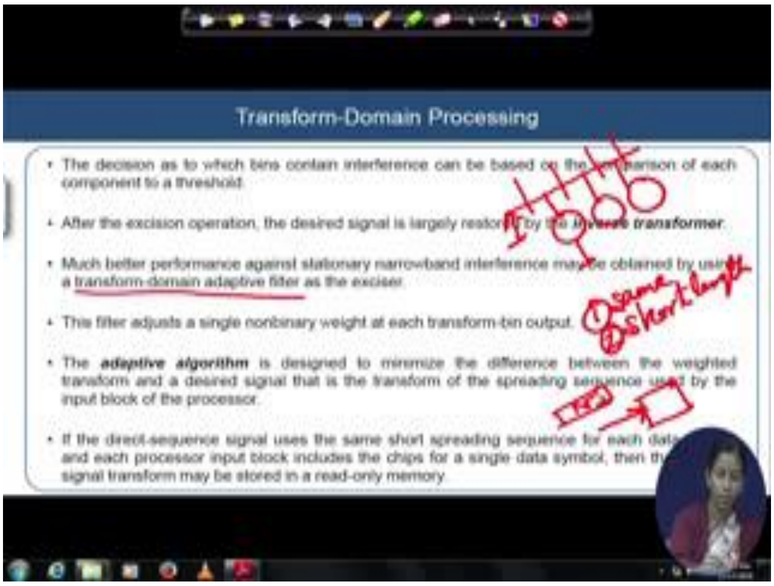


The transformer is selected in such a way that the transformer domain forms the desired signal as well as the interfering signal, such a way that they are distinguishably, they call be distinguished properly. The properties of them we hence we exploit to design this transformer. And how actually the desired signal and the interference will be different in the transform-domain is it depends actually how you are doing it transforming them. And fundamentally when the transformation is there, when the transformed processing is done, you will be able to see that if I see the different bins in the transformed domain, you will be able to find that the interference components will confined into certain bins rather than spreading over all the bins. And if I see that the amplitude of the signal component - desired signal component, and the amplitude of the interfering components, we will see the amplitude of the desired signal is uniformly constant over the multiple bins.

Whereas, so the amplitude of the desired signal you will be finding the proximately constant over the multiple bins, whereas the amplitude of the interference signal would not be that. And hence what exciser does is once the identification is over that we in which bin the interferers are present, and or the interfering signal is present, then exciser would the coefficients that bin to be 0. And basically putting those coefficients 0 means it is a complete suppression of that complete suppress of the interference inside that bin. And as we understand that interference is present the interfering signals are present in only one bin or a small portion of a typical bin, hence putting zeros to those small portion of the whole bin architecture, a partially actually that portion is turned on to be 0 by the exciser. It has negligible effect hence over the desired signal.

And now you are actually output of the exciser you are getting transformed bin with the mostly actually with the cleared desired signal, and hence inverse transformer bin these signal cleared signal to the original domain. Instead of the setting 0, purely setting 0, exciser can actually put some minimal value also in some cases, but this is the overall philosophy or the way that transform-domain processor works, and exciser helps us to do the suppression of the interference.

(Refer Slide Time: 08:13)



And remember that instead of in spite of actually deciding that in which bins the things are there, the interference is there the task is not easy to detect. We follow the methodology like this. Suppose, if you are trying to search in which bin the interference is there in the transform-domain, we try to actually measure the value of each and every component of the bin, and we measure compare the amplitude of that typical component with respect to a threshold - predefined threshold.

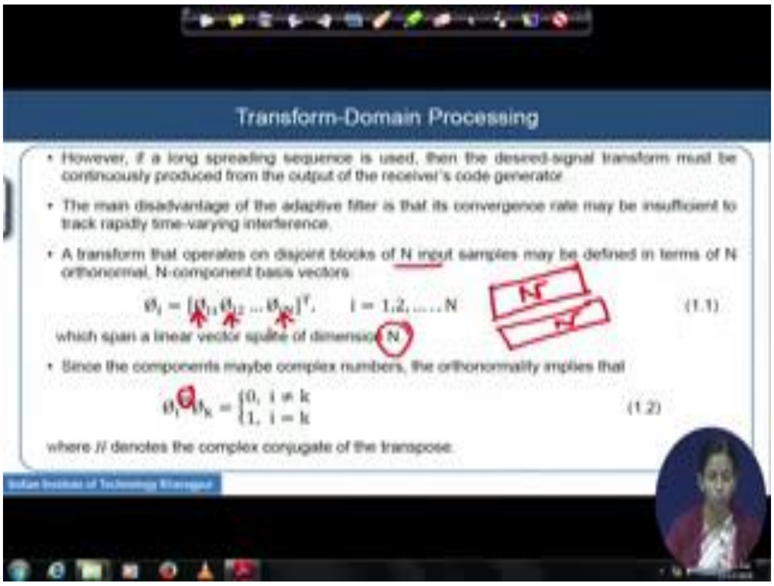
So, whenever actually the threshold is crossed most of that cases may be actually it is a amplitude or may be typical amplitude of the interfering signal because interference signals come with higher power usually. So, we can put the threshold and choice the threshold in such a way that for interfering signal only or the threshold will be crossed most of the time. And hence the detection power processor is over, after detecting the bin, and the component itself it is easy now for the exciser to suppress them and by putting zeroes to those components.

And we can actually deploy another kind of the filter architecture inside the exciser to perform the suppression of interference, if the interference is static. For this stationary narrow band interference, we prefer to deploy a transform-domain at adaptive filter as an exciser. So, what this adaptive filter does is this filter adjust for each and every bin, it adjust a non binary weight vectors. So, it puts actually the weight for each and every weight vector for each and every bin output and the adaptive this is an adaptive filter. So, the adaptive algorithm tries to minimize the difference of this weighted output of the bins and minimize the difference between weighted output of the bins and the desired signal. So, suppose your signal is a PN sequence and you understand actually what will be the value of the desired signal in the transform-domain.

Now, here actually the problem statement is you adjust the weight in such a way that the actual or ideal conditioned value of this desired signal in the transform-domain and this weighted value of this transform-domain, the error between these two is minimized. And that is the task of the adaptive algorithm itself to go to head with. And remember it can be done very fast also easily if the spreading sequence that is used to spread each and every symbol - transmitted symbol is same, number one is same and you are using a short length sequence. So, what is a what is a situation is you are using suppose if I am measuring a capital N number of the chips and capital N number of chips are constituting one symbol (Refer Time: 11:25) symbol. And the bins and the spreading sequence that is used for each and every N chips is exactly same by the transmitter use by the transmitter is exactly same.

In such situation, you need not too actually; you can actually speed up the exciser processor by storing transform-domain values of this short sequence inside the memory, but in the read only memory. So, that you can invoke that and whenever the comparison will go on for symbol to symbol basis. And remember, always we are also considering that are the input of the processor, there is a input to the processor is always having a block of capital N chips. So, the number of chips within block is not changing and spreading sequences for block-to-block is exactly used to be exactly same.

(Refer Slide Time: 12:24)



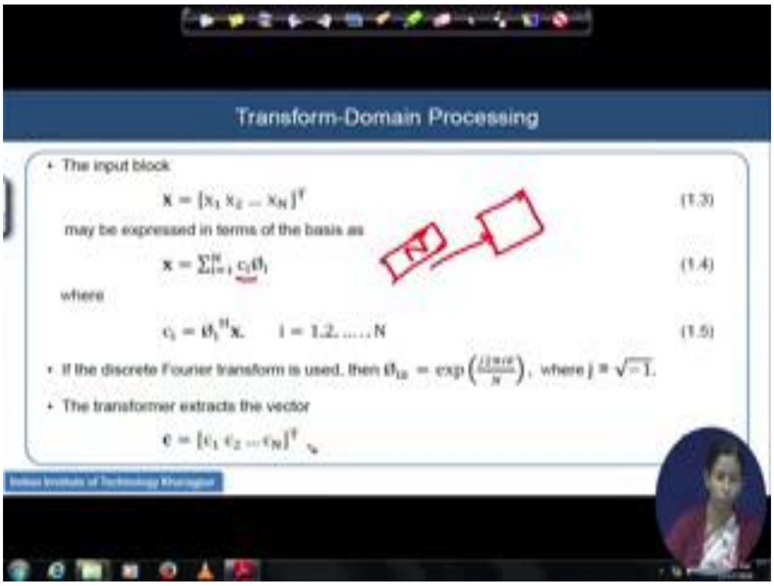
Then by storing the desired signals transformed values in the read only memory, we can speed up the process of the exciser. What if, if there are long spreading sequences used if a long spreading sequence is used then you are helpless and for the every desired signal to give the every desired signal transform output. You have to really take the transformer you have to really put the locally generated PN sequence locally generated code sequence to the input of a transformer and then he will generate values in the transformed domain for the typical receiver code.

The main advantage that we will be getting by this adaptive filters structure putting inside the exciser is that the convergence rate actually it can be actually very nicely or very efficiently suppressing the interference. But with the restriction with the disadvantage or the critically that the adaptive algorithm again like the time domain adaptive filter again the adaptive algorithm needs to converge, and that cannot be done that convergence will never happen very rapidly, if your interference is a time varying interference.

Now, let us assume that transform that operates on a disjoint block of capital N input samples; we define that it in terms of capital N orthonormal and N-component basis vector. So, this is a transform that operates on capital N, we told that the chips are coming at the input of the processor at the block of capital N chips, and we consider that they are disjoints such capital N number of the chips constituting one block. And we consider that the transform is happening by capital N input samples capital N orthonormal and vectors and orthonormal basis vectors, orthonormal basis vectors the components of all those orthonormal basis vectors are given in this equation 1.1. And which this vector spans over the linear space of dimension capital N, and remember each of the component of these vector may be actually a complex quantity also.

So, if it realize on the orthonormality, then it implies that the theta I hermitian with multiplied with theta k should always give one only if i is equal to k; otherwise, it should give 0. We do hermitian, H stands for the hermitian operator, which denotes the complex conjugate operation over the vector over the component. We took it to be hermitian because we understand this components can may be actually complex.

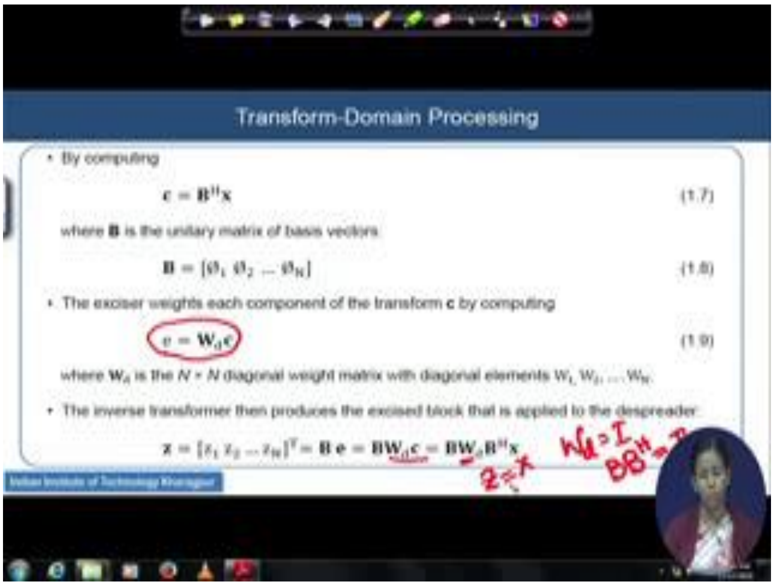
(Refer Slide Time: 15:21)



Now, the equation 1.3 suggests what is the incoming signal vector, X is a incoming signal vector given by X 1 to X N. So, each of this vectors can be now which any component of this vector I can express in terms of the basis function like equation 1.4. Where this new component that is introduced here is c i, and c i is given by the theta u i H into x. Now, if I think that the transformer that where this block capital N blocks of the input chips are entering, this transformer is a Fourier transformer, it is a Fourier transform.

So, you are making a Fourier transformed domain processing. And in this such cases this values of this theta i is a coefficients orthonormal basis function coefficients, it will be given by the e to the power j 2 pi i k by capital N, this is the known values to us. But j is equal to plus minus j is equal to square root of minus 1. And this transformer this transformer then will able to extract the values of all the c i's. So, if I have capital N number of the input chips, I will be only interested in the capital N number of the vectors coming out of that structure of the transformer.

(Refer Slide Time: 16:47)

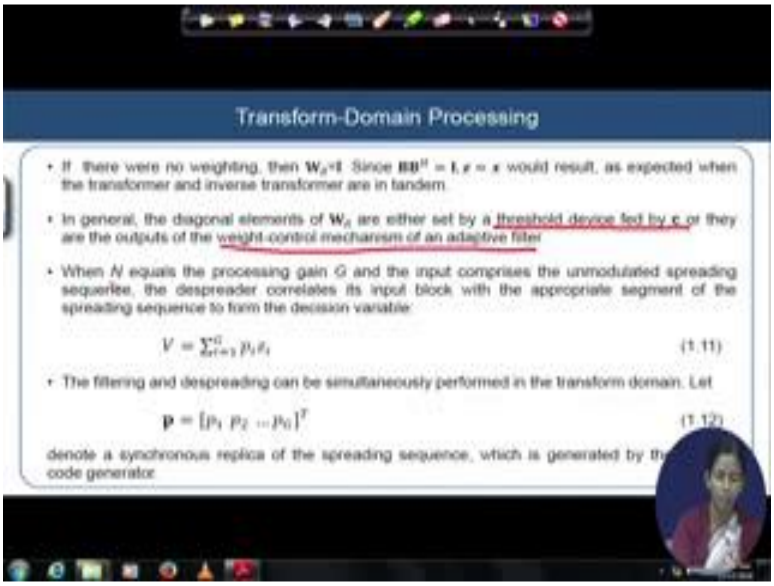


And now once you are having in the value by values of the c, we can compute the value of c from the B hermitian X; and where this B hermitian is nothing but the unitary matrix of all this basis vectors. Now, when exciser is in the path exciser this then weights each of every component of the transform c by computing the exciser weight like this. Where this W d is nothing but a n cross n diagonal matrix and where the diagonal elements will be put as W 1 to W n.

Now, the inverse transformer that was actually last following the exciser, he will then produce the excised block output applied to the inverse transformer and then whatever the output coming out from the inverse transformer that will go to the despreader is basically given by again B into e. If I substitute the value of the e, then we are ending up with W d into c. And we if we change the value of this c again we can write on B hermitian into X.

Remember, here one thing if my wd this n cross n diagonal matrix if it is an identity matrix then all the entries will be 1. So, if I consider the W d matrix is an identity matrix, there we end up here with here is with B into B hermitian. Remember B intoBb hermitian will always give you another identity matrix. So, in that typical situation, when W is equal to identity matrix and b hermitian is obviously, will be hermitian matrix then you are ending up with the fact that Z will be equal to X. So, output is basically you are getting as the desired signal only.

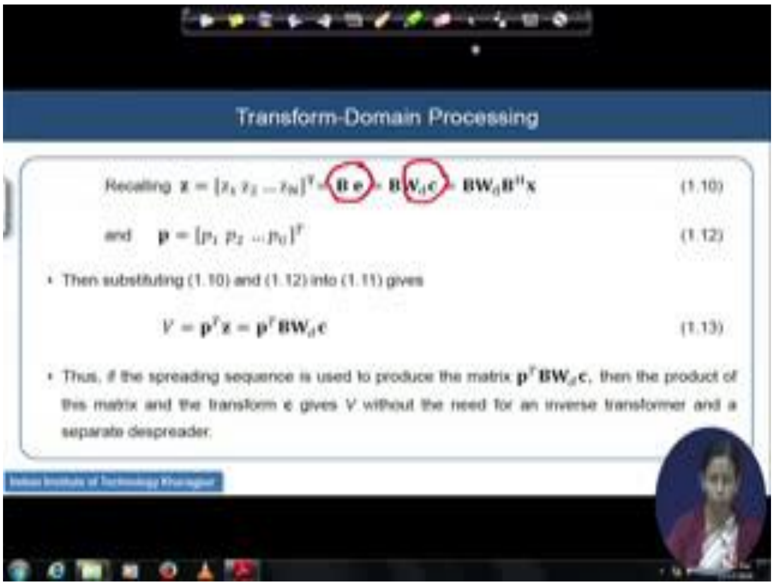
(Refer Slide Time: 18:48)



And in order to achieve that you please remember that the transformer and the inverse transformer both will be in tandem. So, now, the question is in general, this diagonal elements of this of the matrix W, this W d they are either set by the threshold device fed by c or they are the outputs or the weight-controlled mechanism weight-control mechanism of the adaptive filter. So, entries of the W will be either from the threshold device that is fed by c or otherwise it will come by the weight-control mechanism of an adaptive filter.

Now we are talking about a the block of the capital N chips, let us have a situation when this capital N equals to the processing gain of capital G. And then the input comparisons on the unmodulated spreading sequence, and the despreader correlated this input block with the appropriate segment of the spreading sequence to form this decision variable. So, what we are doing we are having the incoming unmodulated signals, and we have locally generated codes and we are multiplying it and we are taking the addition over the processing gain. The filtering and this dispreading both actually so first is they all filter and then is a dispreading operation. So, this filtering and this dispreading can be simultaneously performed also in the transform-domain. How, let us see. Suppose, p is that signal that locally generated code signal that we will be dealing with, and let us assume we are restricting this discussion for a length of capital G number of the chips, where G is also the gain processing gain of the system. And now this capital P vector this small p vector denotes the synchronous replica of the spreading sequence which is denoted by the receiver code generator.

(Refer Slide Time: 20:54)



Now, see recall that we have seen these two we have seen already these two equations. The equation number one was on the z, we saw that the expression for the z is was coming from the B into the exciser output, B into the exciser output, and it is a weight; where the exciser is a actually basically the n cross n matrix of W, and multiplied with the weight vectors this c component. And if I further replace this c by B hermitian x and we saw that the expression for the z is finally governed by the B W d B hermitian into x. And we have considered the vector p such a way that the p is actually our code sequence.

Now, if I substitute this one and this one, where into the expression of V that we have seen earlier, come here in the equation 1.11. So, this value of the p and then value of the z are now known to us according to the equation this and this. If I put both these values into that V, then V will, finally obtained as we transpose of z and z is again your B W d into c. So, we are finally, ending up with the decision that is going on. So, V is what, so V was actually the decision variable the final decision variable above the transmitted message. This vernal value of this message the decision is now basically dependent on this parameter c B W and the designed code p.

Now one important thing if the spreading sequence you design in such a way that it produces this equation this matrix p transpose B into W d produce the matrix p transpose B W d into c. If this is the matrix that we can produce by the choice of a proper spreading sequence, then this product then the product of this matrix and the transform matrix c will return me back the original V. So, when this kind of situations happens the spreading sequence is taking care of producing the matrix, which is actually required for the final decision.

And hence in practice once this kind of the transmission is going on then the product of this whole with matrix itself and the transform c gives this V back without how we have got this decision. This decision is now without the inverse transformer and separate despreader requirement, you have you can actually delete completely. So, if the decision is out, spreading sequences taking care of all the part and hence you need not to put any inverse transformer as well as a separate despreader to de-spread the signal and detect the signal. So, automatically despreader and filtering operation is going inside the architecture automatically because of the proper choice of the length of the sequence as well as on the quality of the sequence involved in the processing.

So, this is all about how the transform-domain processing works. So, I repeat that we started with the understanding that the time-domain adaptive filter is not so good in the case of your narrowband interference handling, when the interference is varying over the time. So, we were trying to see is there any other architecture which can nicely handle this, and the answer was yes.

And the possible architecture is the transform-domain processing. Inside a transform-domain processing basically we see a transform-domain processor which takes a inputs from the matched filter, and the match filter is the sampled match filter output it is taking in a block of n chips the inputs are coming into the transformer. And this inputs come from a match filter output, and where match filter output is matched to the chip waveform. And we have also seen that inside the transformer this transformer can be a real-time Fourier transformer or it can be wavelet transformer also.

The transformer is designed in such a way the transformation algorithm is designed in such a way that in the transform-domain easily you can distinguish between a interference and a signals desired signal. And as actually over if I see the output of the transformer over the different transform-domain bins, you will also see that the interference is residing in a short bin, in a small bin compared to the uniform amplitude of the desired signal that is spread over all the bins. So, by detecting for detection of those values in which bins the interference is there, we do a threshold comparison and output of this threshold comparison will identify the location of those bins, where the interference will be maximum, and then the interference will be also out and will be suppressed by the exciser by putting zeros to those bins.

The exciser can be also implemented by adaptive filter architecture. If you feel that the interference is static in nature, so exciser can also do that; and that time the adaptive algorithm selects the non binary weights for the output of each and every bin. And the algorithm should designed in such a way that the weighted output and the desired transformed desired signal transformed output will have will perfectly match with each other with minimal error. And if it satisfied then we by some mathematical derivation we have shown that the output can be easily be a function of easily can be retrieved. And it can be also possible that we need not to utilize inverse transformer and de-spreading operation finally, to give the signal back, rather actually for some typical situation this code design itself can handle the issues.

So, that the absence of inverse transformer and as well as your de-spreading operation the spread signals decision on de-spread signals output can be taken easily. So, exciser has several kinds of the rules not only the suppression; sometimes with proper design of the code, we can also delay the inverse transformation and de-spreading operation also by a proper design of the filter. So, filtration and de-spreading jointly is possible to design inside the exciser in transform-domain.