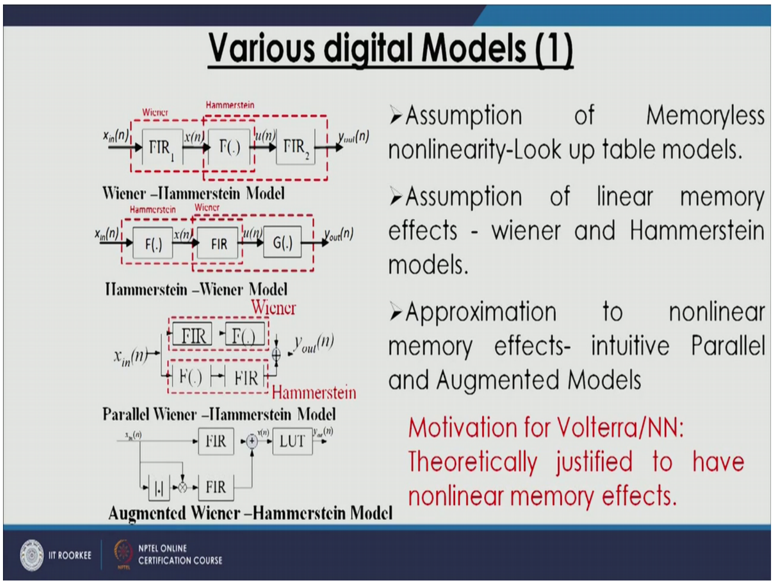
Basics of software-defined radios & practical applications
Dr. Meenakshi Rawat
Department of Electronics & Communication Engineering
Indian Institute of Technology, Roorkee

Lecture – 18
Basic Digital Predistortion Techniques for non-linear distortion in SDR

Hello everyone. In the series of basics of software defined radios and it is practical applications, we were discussing the pre-destruction techniques and among that we today, we will be discussing the topology for the digital predistortion techniques. So, this is these are the models which can be used to represent the inverse or the forward model for the power amplifier.

(Refer Slide Time: 00:46)



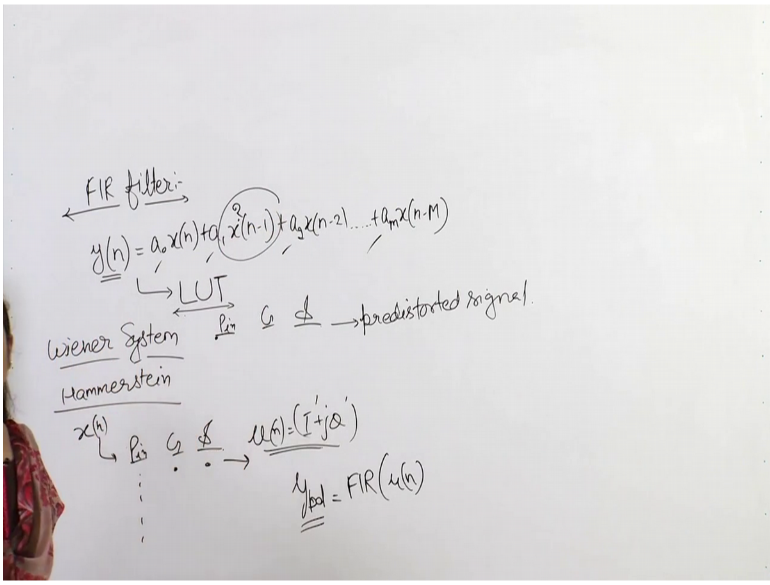
The LUT is the one model, which is representing the nonlinearity which is given by function f here. LUT is basically just the gain and phase response with respect to the input power.

So, if we already know the input power, then we can arrange the inverse gain and phase also. And that information we can achieve from the actual I plus jq complex baseband waveform, which we provided by our self as it any sequence in the MATLAB and the one which we captured from the vector signal analyzer again in the complex form. So, by using that LUT which we described in the last lecture, we can represent this non-linear function. Now, we have earlier discussed the memory effect of power amplifier. Because of the temperature and the effects of elements such as capacitors and inductors, this power amplifier has memory effect.

So, previously, most of the researchers they believed that, these kind of memory effects are linear in nature. It means, they fade with respect to time. So, if we look at the first structure, which is called wiener structure, wiener is the name of researcher who proposed, that we can use a FIR filter before this non-linearity and it should be able to give us the perfect model. The Hammerstein has proposed that, we put non-linear function which is the LUT before have a FIR filter and that structure was called Hammerstein a structure.

Eventually, as the time progressed many such structures and their variations came into picture. For example, wiener Hammerstein is structure is the one structure, where we are using a FIR filter before the nonlinearity and when a FIR filter after the nonlinearity. And this such a structure was called wiener Hammerstein structure. as you can perceive by yourself, if we put is Hammerstein's structure before the wiener, then it will be called Hammerstein wiener model. So, basically it is just the placing of the FIR element and the non-linear function in the system. Now, assumption was that the memory is linear. What does it mean?.

(Refer Slide Time: 02:58)



Our FIR filter, this a FIR filter actually is given as yn is a function of xn and it is previous values.

So, here, we are assuming that, output is a function of input and it is previous values in a linear fashion, they are just multiplied with a constant and then, there at being added together. So, when we have this assumption, we can use the FIR filter. So, in this case, in the wiener structure, we are putting this a FIR filter before this and after whatever we are getting from the output of the FIR filter, we call it un as we can see from the a structure here.

So, this un is something, when this yn is applied to LUT. So, basically, the LUT, which was created using Pin versus gain and phase [vocalized-noise. Now, that Pin is being read by use a this output from the FIR filter. And accordingly, gainer phase is being applied to your digital predistortion system. So, it was wiener system. First, a FIR and it is output into the LUT and LUT output will be the predistorted signal.

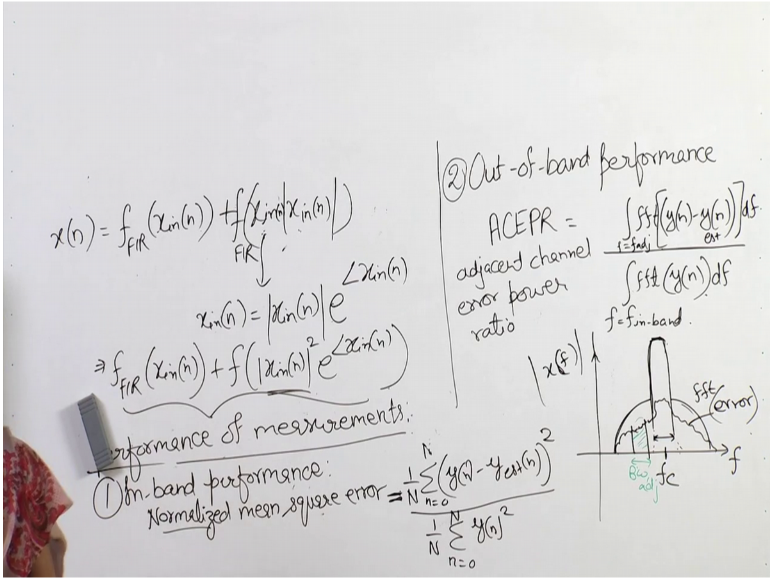
In Hammerstein's system, first of all, you will have your LUT. So, with respect to input x, you can read your value of gain and phase. So, even a pin values are according to xn you can read you again on phase value. With that gain and phase value you can make your u value, which is un, which is again a complex number I dash plus j q dash based on this gain and phase I will applied with this input xn. And once you have this u n, then this u n is going through the FIR filter.

So, your y p d predistorted output is actually a FIR function of u n. Now, this ypd, it is compared, with the required output of the predistorted function and the coefficient of a's in this equation, they are adjusted to fit the model. in time there are some other models some other variations it was proposed that it is possible that, they might not be completely linear kind of effect, but they can have some nonlinearity in the system..

It is possible that xn is not simply related to xn minus 1 in a linear fashion, but x square may be. So, in that case some augmentation of such systems happen and parallel wiener Hammerstein and you can say augmented wiener Hammerstein. Especially in the augmented wiener Hammerstein, you see, xn is being multiplied with the absolute of it is own.

So, in this case, for this first loop, this xn which is being applied to LUT parallel wiener Hammerstein medal method this xn is what xn is equal to function of a FIR filter x in n plus the parallel branch.

(Refer Slide Time: 07:10)



This branch is basically, x in multiplied with absolute of xn and this is what being applied to function of FIR filter right. So, this is the xn we are getting here which will be applied to LUT. So, then, output will be measured with respect to our ideal output of the inverse model or the forward model and the coefficient of FIR will be tuned. So, that we have the best fitting there. So, in this case, if you see here, it is not simply function of x in and it is delayed version one order has been increased. Now, it is using x square in a way, because, what is the x in n? It is basically, absolute portion and phase of x in n each case.

So, if we I see, this one it becomes actually it is a function of FIR function for sure. But apart from that, it is also a function of x in squared. So, from here, we can see it is the it is using the second order. So, it is becoming more non-linear here. So, it was found that, this kind of complex structure will giving much better fitting performance.

Now, this performance is they are measured in terms of performance of measurement, we have discussed earlier also. In band performance, is measured in terms of normalized mean square error which is given as 1 upon n yn minus y estimated and. So, it is mean square error and when it is divided by, it is actual output at that time. Then, it is called normalized mean square error.

So, what is the benefit here? That suppose, the amplitude the value of yn is very large, then it will lead to very large error and it is also because, if it is of the range 1000 and it is 998, then, we are have a of order 2. But if it is of order one, and if it is 0.9 and 8 then, it is a 0.002. It looks like it has smaller error, but the problem is not that it has smaller error. Thing is that, it is original amplitude was smaller. So, we are getting the smaller error. So, it is not a FIR comparison..

So, that is why, in the mean square error which is given by only this equation, it is not sufficient. When we normalize it with the output value of the signal, extra signal, then it is normally normalized near the one value. And in that way, for different kind of signals different kind of amplitudes, we have merit which we can use to compare different devices and different signals easily.

So, normalized mean square error is the one, which we used which we use to check the in-band performance properly. other performance measurement for the out of band performance out of band performance can be given by ACEPR adjacent channel error power ratio. So, what is this ACEPR? first of all, we do fft oh sorry dft of our error. So, now, this error is actually what it is again yn minus y estimation.

So, let us put it right it in that way yn minus y estimated. So, we take fft of these values and then we choose our frequency limitation. So, we choose our frequency in the adjacent band and we choose our frequency in the in band. So, when we compare these true matrices it is using all the data in the time domain in ACEPR we are choosing our data we are again using error with respect to your original output, which you are trying to model. But the difference is that, we are doing this in the frequency domain by choosing the add adjacent channel, we are showing the error in the adjacent channel with respect to your actual signal fft in the in band.

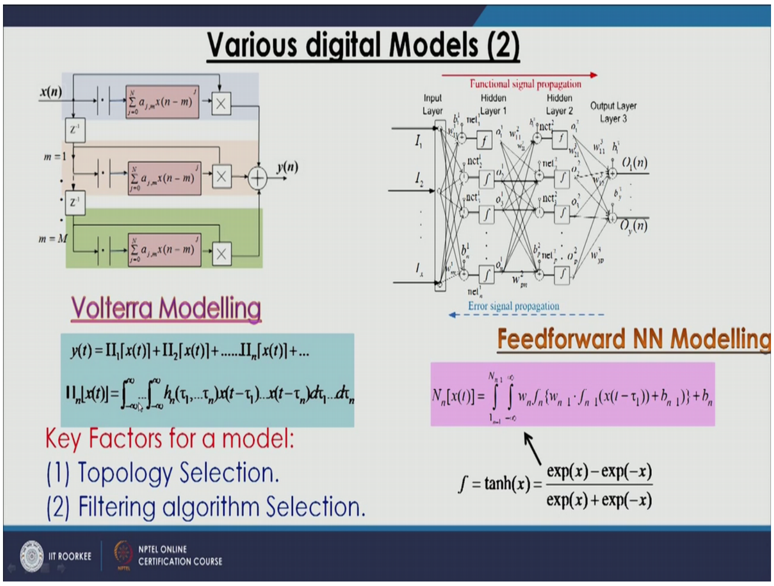
So, for example, if this is the carrier frequency and this is a system response simply and we want to show the signal here. So, let us make it amplitude in. In fact, it will be better observe there xf amplitude value. So, it looks something like that. Because of the power amplifier distortion, it used to be original signal, but now it will be this signal which is distorted and it is leaking into adjacent channels.

Now, this our carrier frequency and this is our in band area. So, we first of all plot the fft which will look like that and then we integrate our signal in band the output signal. So, that value will be here all these values and then we will divide this value what we are getting here with respect to our error an our error signal for example, our model error is something like that. This is the error fft.

Then, we take integration of this new area and this is the limitation of this adjacent. So, this is the area, which we are integrating here. So, integration of this area with respect to integration of this area, this is what will give you ACEPR and it is out of band performance criteria. So, whenever you are comparing this model, few plot these 2-ah metric can you see if it is working well in the in band and it is working well in the out of band and by checking that, you can see which model is the best.

So, here, in this case, we were finding that more complex models were giving more better and mse and more better ACEPR. So, from that we got the motivation, that maybe we should go to completely non-linear models which have the non-linear kernels. So, there came the models such as Volterra model and the feed forward neural network model.

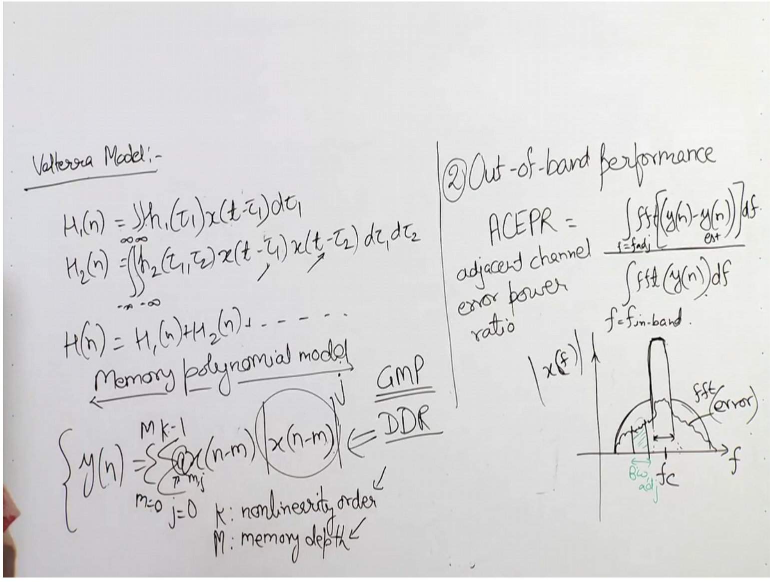
(Refer Slide Time: 16:28)



Now, neural network models are basically non-linear in nature, because, they have neurons and this function f is basically this function f is basically non-linear in nature it is given by hyperbolic tangent function. Similarly, Volterra model, it is a summation of different multi multiplication of different input signal order terms and they are also delayed by a particular amount and it can go from infinite man is infinite to infinite means, basically Volterra model says, they can that we can multiply to any polynomial order and using any memory depth.

Now, we can simplify actually this model..

(Refer Slide Time: 17:18)



Ah if you look at this Volterra model, if you want to open this term, let us put n equal to 1 [vocalized-noise. If that happens, then, what we will get? We will get n, it will just contain your first impulse function first term h 1 and it will contain your 1 delay. So, it is function of tau 1 only 1 delay and x t minus tau.

Similarly, it will be just with respect to theta 1, it has only 1 delay term. So, if you go to h 2, you can expand this expression and you can see that, it will be having 2 delay terms and it will be having 2 delay here also x. So, it is function of 2 delay terms and then it is integrating from minus infinite to infinite for both of them. So, is if you keep increasing this term, it can go to infinite, but of course, infinite cannot be processed.

So, there are some simplified terms. one of them is called a memory polynomial and the simplified expression is given here. If you look at this term, it is basically representing the memory polynomial or the polynomial terms with different delay taps here, ok. So, if you go to this one, and try to simplify this then instead of using different delay terms in the multiplication, what they are saying? Because, eventually your hn will be summation of this one, h 1 n plus h 2 n and so on. So, they have simplified this term instead of using this one. This hn is represented as memory polynomial. Only diagonal terms are being taken.

So, no multiplication of these 2 terms will be happening there. So, how it works basically that, yn becomes a function of xn and absolute value of x. So, j is equal to from 0 to n minus 1. let us make it k minus 1 because, n we are using for number of samples. This k is non-linearity order. And this m let us take some 0 to M. This m we can call memory depth. So, the lies model is Volterra model, but volterrra model is very big model it has to be condensed into a smaller model. So, this model came into picture and it is a very popular model which is called memory polynomial model.

Now, how it will work? When M is equal to 0, the output is only function of x n, because, M is 0 here. So, it becomes only a polynomial series right. So, we put a coefficient here, which is a function of M comma j and if we are tearing this coefficient, we are able to get our output easily. From this expression, once M equal to 1 comes into picture, then it will have x n minus 1 term also there, but you see, we are not multiplying n minus 1 and n minus 2 terms together. We are always using that will terms..

We are always using similar terms. So, n minus 1 will be here, then also certain and n minus m will be here and that term also will go from 0 to k minus 1. So, this is how we are explaining this term that for m equal to 0, 1 polynomial branch then, m equal to 1, another polynomial branch m equal to M another polynomial branch and eventually they all are being added together.

This nullity is being applied to only absolute value and the accent which is containing original phase is multiplied after this. So, same here. We are applying nullity only in the absolute portion and the main signal is being applied only once. So, the phase of the signal is being on is not being multiplied non-linear e in this case. So, this was the polymer base model and as you can see many variation of this polynomial model you can do for example, if you are allowing them to mix and match and you make 2 a strings here and you say m it is m 1 and m 2 different values it can take. Then, in that case, it is a allowing it to mix and match.

So, there. So, many models here with such as GMP generalized memory polymer, DDR model dynamic division reduction-based model and they all are taking advantage of this Volterra series and making this memory polymer a little bit more complicated. But to achieve better modeling performance, so, you are not going into all those small models because, they are several and several in the literature. But the basis comes from the Volterra model and it is deviations. So, these are kind of polymer-based models, then recently neural network models they are also a much popular. This kind of models are kind of feed forward type of structures. So, as you can see here, they have non-linear functions here and they are connected with many connections here each connections have some weight.

So, they are those weights are equivalent to this coefficient which we are using here. So, here we can tune this coefficient of it our estimated outputs with the actual output. In neural networks, we can fit these weights in the in this connections, so that, eventually this output is matching perfectly with the output of the our system which is required. So, without going into details, they are subject of their own. But you can easily use them as a toolbox. If you want to apply those to get some results ah, there is available in MATLAB and their toolboxes independently available in the literature also.

So, basically, once you have chosen which kind of structure you want to go with; maybe it kind of polynomial structure or maybe neural network kind of a structure or you can even select some other kind of polymer fittings, maybe spline fitting and we have discussed the parametric model such as raf model saleha model. They also these kind of models which we can use for fitting.

So, there are 2 factors in a model when you want to fit something. One is topology selection, which we have discussed; means, if you are going for the polymer model, what should be the nonlinearity order? and what should be the memory depth? So, once you choose this topology means, choosing the nullity order in the memory depth, then you can go for the tuning of your actual coefficient, so that, you can get your output easily. similarly, neural network models also, you have to choose how many layers you want and how many neurons you want to keep in each layer..

So, this is the part of the topology selections. Once you have chosen the topology, then second part is the filtering algorithm selection, which is the algorithm by which you are actually tuning your coefficient. So, in the next lecture, we will be covering the filtering algorithm selection and we will apply this particular algorithm to a particular topology and we will see how it impacts our PA output linearization.

Thank you.