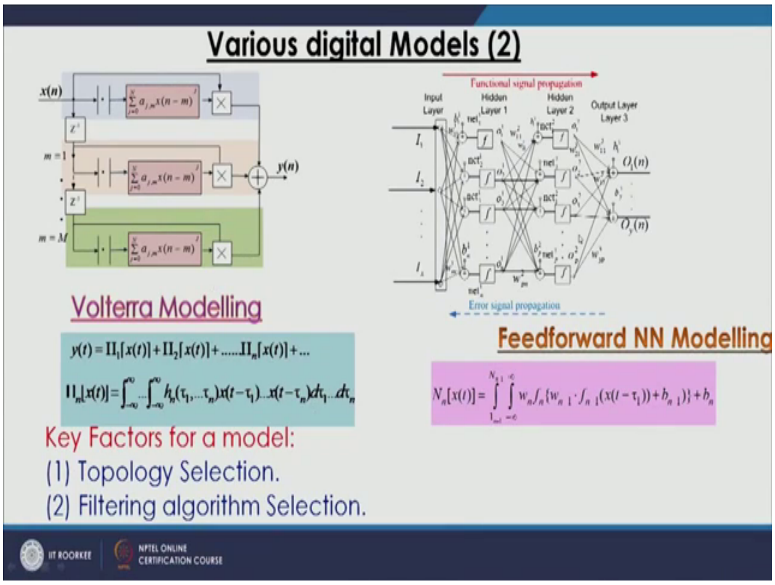
Basics of software-defined radios & practical applications
Dr. Meenakshi Rawat
Department of Electronics & Communication Engineering
Indian Institute of Technology, Roorkee

Lecture – 19
State-of-the-art Digital Predistortion Techniques for Nonlinear Distortion in SDR

Hello everyone. So, in the series of software defined radios and practical applications. We were discussing the digital predistortion techniques and in this lecture we will cover the linear and non-linear both the techniques, the algorithms and the results in the practical system.

So, as we were discussing in the last lecture we have to select one of the topology and once we have selected the topology with which we want to go for the modeling then second concern is the selection of algorithms.

(Refer Slide Time: 00:40)



Various digital Models (2)

Volterra Modelling

$$y(t)=\Pi_1[x(t)]+\Pi_2[x(t)]+\ldots\ldots\Pi_n[x(t)]+\ldots$$

$$\Pi_n[x(t)]=\int_{-\infty}^{\infty}\ldots\int_{-\infty}^{\infty}h_n(\tau_1,\ldots\tau_n)x(t-\tau_1)\ldots x(t-\tau_n)d\tau_1\ldots d\tau_n$$

Feedforward NN Modelling

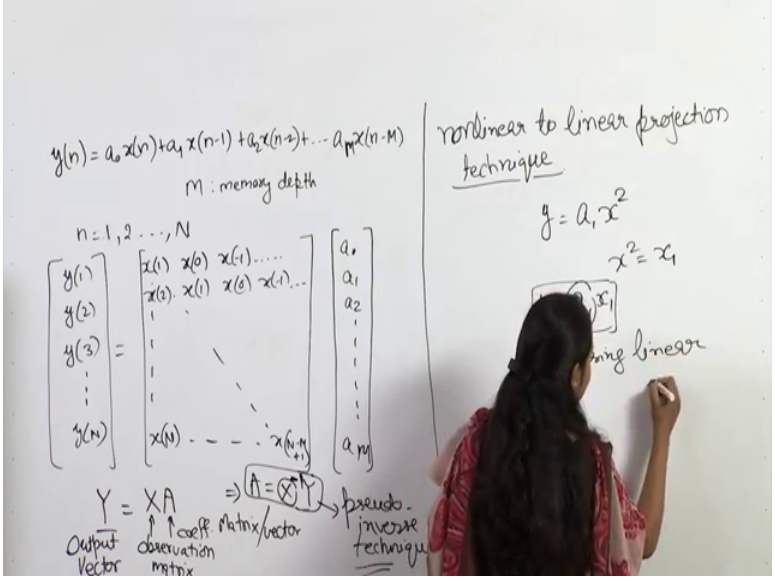
$$N_n[x(t)]=\int_{1_{n-1}}^{N_{n-1}}\int_{-\infty}^{\infty}w_n f_n\{w_{n-1}\cdot f_{n-1}(x(t-\tau_1))+b_{n-1})\}+b_n$$

Key Factors for a model:
(1) Topology Selection.
(2) Filtering algorithm Selection.

IIT ROORKEE NPTEL ONLINE CERTIFICATION COURSE

So, they are several filtering algorithm in the literature, but first of all we have to see how we can apply those algorithm. For example in linear filter theory we have some of the examples where we do the fitting and in our Wiener Hammerstein's model we are using a fir. So, we will start our discussion from the fir filter there fitting and then we will take the non-linear to linear mapping for the memory polymer model and we will show how we can apply those in the non-linear terms.

(Refer Slide Time: 01:25)


y(n) = a0 x(n) + a1 x(n-1) + a2 x(n-2) + ... aM x(n-M)
M : memory depth
n = 1, 2 ..., N
y(1) y(2) y(3) ... y(N) = x(1) x(0) x(-1) ..... x(2) x(1) x(0) x(-1) ... x(N) ... x(N-M+1) a0 a1 a2 ... aM
Y = XA
Output Vector
observation matrix
coeff. Matrix/vector
=> A = X⁺Y
pseudo-inverse technique
nonlinear to linear projection technique
y = a1 x²
x² = x1
x1
ing linear

So, for a fir model our y n was a function of input signal and its previous values of the input signal m is memory depth here. So, basically we can represent the same thing as a matrix also. So, if we take different value of n from n, n equal to 0 to 1, not 0, it will have integral value positive integral value 1, 2 up to n.

Then we can basically represent this as y 1, y 2, y 3, up to y n output and x n can be represented as x 1 x 0 x minus 1 and so on till mth value and coefficients can lie here let us put it a m. So, it will be a m here with will be, which will be with respect to this one. Similarly it will have all the value is still x capital N and this value will be x n minus m plus 1. So, by arranging this values it will be x 2 value then x previous value its o previous value x minus 1 and so on.

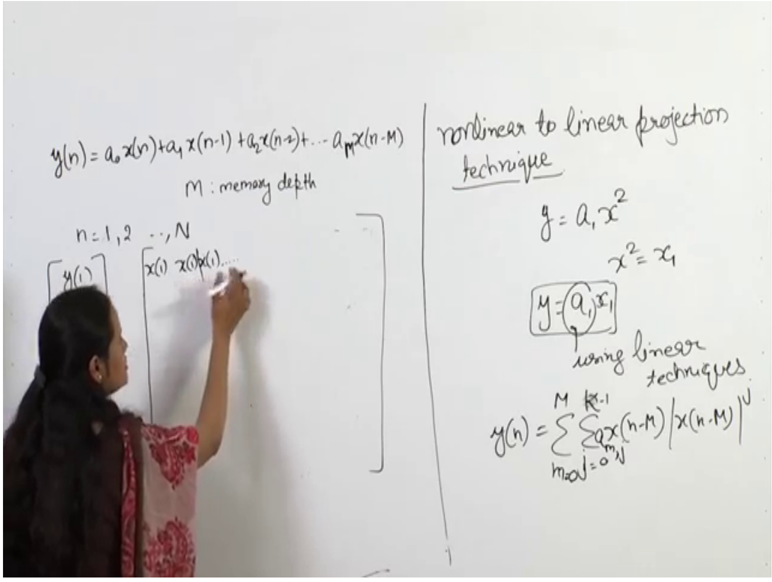
So, we can arrange this term as this one also. So, we can represent this as Y, this as X and this one as A. So, A will be representing the coefficient matrix, this X will be observation matrix which is including all the input data, actually this matrix coefficient matrix can also be called coefficient vector it is simply a vector and this will be the output vector, output signal vector right.

So, if we are able to represent it in this way then easily by using matrix calculations we can calculate our coefficients here. So, what will be coefficient here? It will be X inverse Y. Now, if this matrix x is not square and if it is it may be singular then that inversion it is not, one not possible. So, we have pseudo inverse techniques which takes care of inversion process even if this is not a square when if it is a rectangular kind of matrix.

. So, normally they go with the pseudo inverse techniques such as Penrose Maury inversion technique. So, this is the basic arrangement of the data here. So, observation matrix output vector and the coefficient vector will be there. Now, it was for the fir filter which we use in the Wiener Hammerstein how we will calculate our coefficients in the non-linear filter. So, for the neural networks it is contained a different kind of processing they use step by step techniques, but for the memory volume voltage series when you are using polynomial we can do non-linear to linear projection technique.

What is this technique? Basically here if you look at this it is a linear algebraic condition which we are imposing here and we are solving for that one, when we are talking about non-linear terms if we can do the projection. So, that non-linear terms can be represented as a linear term then we can still use this equation. And how can we do? So, for example, if our function is y is equal to a 1 x square we know that is a non-linear equation , but if we select x square to be some other function let us say x 1 then with respect to x 1 and y it is a linear equation. So, by choosing x 2 equal to x 1 we can still calculate a 1 using linear techniques. And this is what we call non-linear projection.

(Refer Slide Time: 07:09)

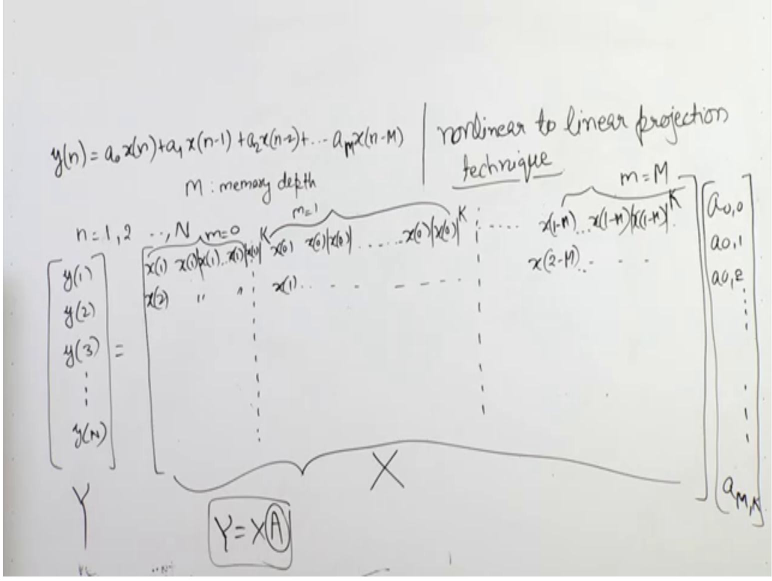

$y(n) = a_0 x(n) + a_1 x(n-1) + a_2 x(n-2) + \dots a_M x(n-M)$
M : memory depth
$n = 1, 2 \dots, N$
$y(1)$
$x(1)\ x(0)\ x(1) \dots$
nonlinear to linear projection technique
$y = a_1 x^2$
$x^2 = x_1$
$\boxed{y = a_1 x_1}$
using linear techniques
$y(n) = \sum_{m=0}^{M} \sum_{j=0}^{K-1} a_{mj} x(n-M) |x(n-M)|^{j}$

So, this is what we can easily use for the non-linear memory polymer. For example, memory polymer equation is given as summation over j from 0 to n minus 1 sorry K minus 1 because K we are using for the nullity order and m from 0 to M which is M is a memory depth which we have discussed in the last lecture and this was the equation for the memory polynomial.

Now, how we can we will do this projection which I which I was talking about? Now instead of this equation we have to map this equation. So, in this equation also we have some coefficients which are dependent on m and j and this coefficient we have to find using this kind of techniques.

So, can we arrange this in terms of matrix and can we apply the linear techniques this is what we have to see. So, y again we can arrange from 0 to n for the n number of data and after that observation matrix has to come into picture.

(Refer Slide Time: 08:37)



y(n) = a₀x(n)+a₁x(n-1)+a₂x(n-2)+... a_M x(n-M)
M : memory depth
nonlinear to linear projection technique
n=1,2 ..,N
m=0
m=1
m=M
K
x(1) x(1)|x(1)|.. x(1)|x(1)|
x(0) x(0)|x(0)| ..... x(0)|x(0)|
x(1-M).. x(1-M)|x(1-M)|
x(2-M)..
x(2)
x(1)..
y(1)
y(2)
y(3)
y(N)
a0,0
a0,1
a0,2
a_M,K
Y
X
Y=XA

No, observation matrix will contain x 1, x 1 at absolute of x 1 and so on for n equal to 0 right.

So, let us make this so that we can understand all these terms together. So, for m equal to 0 we have done the calculation till here x 1 absolute value of x 1 to the power n, not n we are taking it to be K, K is our nonlinearity order right for m equal to 0. After that in the next column we start taking values for m minus 1. So, m equal to 0 is this one and then we take m equal to 1 terms means previous values. So, x 0, x 0 and absolute value of x 0, x 0 until m we can keep do this.

For example, we have done this for m equal to 0 then we had done for m equal to 1 then m equal to 2 and then finally, for m equal to capital M right. So, it will be what? 1 minus M x 1 minus M absolute of x 1 minus M next will be squared and last term will be x 1 minus M to the power K and this will be its observation matrix.

Now, next term x 2 and the same thing x 2 x vector will be here will be x 1 and x 1 x 1 and will be x 2 minus M and so on. So, here what we have done our non-linear kernel we have represented in one, one column, so now, we can again apply our linear equation.

So, after we have defined our observation matrix it is our y this is let us remove this equation now which is the equation some memory polynomial. This is whole equation is our x which is the observation matrix and as you can see if I multiply this whole equation with new vector. So, let us make some space here let us suppose this matrix finished here and this is to the power k, then we can again multiply our a M K all right.

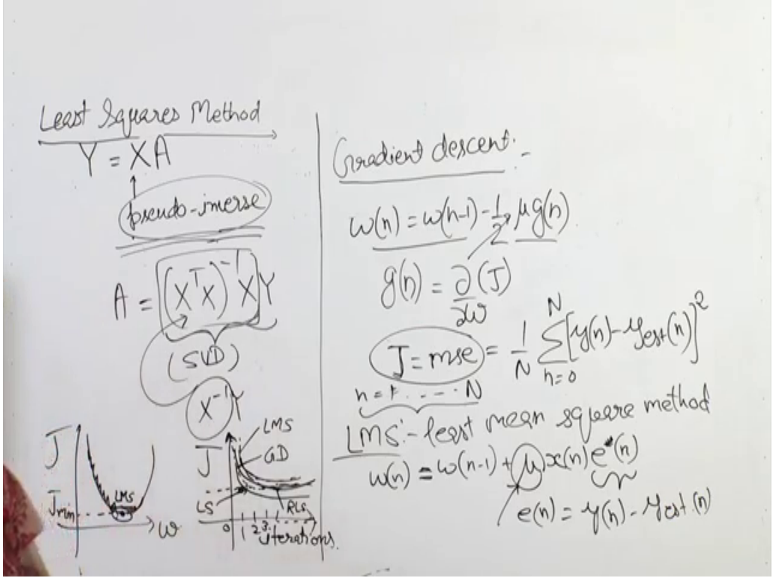
So, we can define our coefficients here according to all the values of M's as well as K and then we multiply we get our output. So, basically again we can represent our equation in a linear first session and if you solve this function because we have done the projection we have put the non-linear terms in one element of this matrix, we have done the mapping of non-linear to linear . So, we can easily calculate our a again using the inversion of this matrix in by using pseudo inverse methods.

So, this is the way we can arrange our matrix and once we know this relation then their different type of algorithms which we can select to make it more easier there. So, there can be two types of processing well one its batch mode. This example which I were showing you here what is, what is for the best mode because you are using from data from 1 to n, nth sample at one time for make making matrix. But sometimes their applications when the system is changing very fast in that case you want to do sample by sample data sampling it means you will be using only first element of this and first vector of this and only this one.

So, we are not using batch mode anymore and it is sample to sample adjustment of the data. So, why is being processed only for single data at a time with total number of coefficient. In both the cases you will notice that number of coefficients are same in one case we are using complete matrix to include n data in one case we are processing one data at a time. So, by based on our applications whether our system is changing fast or do we want fast convergence then we can choose our batch mode or sample to sample processing.

So, in the batch mode processing we have basically least square methods and recursively least square method. So, what is the least square method basically? That you arrange your output matrix, you use your observation matrix and you define your coefficients and based on this batch mode you can apply your least square method.

(Refer Slide Time: 14:07)



Least Squares Method

$Y = XA$

pseudo-inverse

$A = \boxed{(X^T X)^{-1} X} Y$

(SVD)

$X^{-1} Y$

Gradient descent:-

$w(n) = w(n-1) - \frac{1}{2}\mu g(n)$

$g(n) = \frac{\partial (J)}{\partial w}$

$J = mse = \frac{1}{N}\sum_{n=0}^{N}[y(n) - y_{est}(n)]^2$

$n = 1 \ldots N$

LMS:- least mean square method

$w(n) = w(n-1) + \mu x(n) e^*(n)$

$e(n) = y(n) - y_{est}(n)$

J
J_min
LMS
w
J
LMS
GD
LS
RLS
0 1 2 3
iterations

So, by using pseudo inverse of this X you are able to get dual coefficients. So, it is one step process because once you have defined your Y and X observation matrix in one step for n data you will have single solution.

Now, least square method because you are using block base processing and you need to do the pseudo inverse. It will take more resources in digital computation because doing the inversion of the of any system it is a difficult thing and resource consuming thing. So, because of that there are many methods where they try to reduce the complexity of this kind of method, and those kind of method tried to make this pseudo inverse more efficient. So, it is least squares method using pseudo inverse.

So, in this case your A becomes , so , by for calculating this portion which is the pseudo inverse if you see; if I wanted to do X inverse Y it will be inverse of that has been instead of using that we are using this quantity which we use in the least square method . Their methods which is ah singular vector decomposition etcetera, we where you try to minimize the complicity of this portion. So, this is a entirely different research area. So, we are not going deep into it, but we just want to maintain that this is the one step method and it converges very fast.

In sample to simple processing you have gradient descent method and the mean square algorithms quickly going through those methods for the overview. GD method is the best gradient method these are means of this method, and this method we update our coefficients given by this equation, where this g n is actually gradient of this cos vector with respect to weights, and this cos vector is actually mse mean square error of the output.

So, you get your estimated output, you get the mean square error J is equal to 1 upon N y n minus y estimated whole square. You define this and then you see its variation with respect to w by differentiating it and in the real life if it not differentiation you did take the difference with respect to the previous step and you apply this formula, and the coefficients can be adjusted by using the previous coefficient value and the new gradient value.

Again you can see it is a sample to sample processing method because you are taking 1 n at a time, you are not taking n equal to 0 to n sorry from 1 to capital N in one shot, . Instead of that you are sampling taking one and then here processing updating your coefficient value and then you are taking second and even updating and processing these values.

So, for a stochastic method where you know there are non-linear system and you are using variance and mean, those kind of system it was proposed basically; to simplify for the deterministic signal like we are using right now where we have the signal in our hand and we want to work on the based on that data least mean square method can be used, the statement of this one is. So, it is u n, let us make it x n because we are calling input signal as x n. So, again it is sample to simple processing.

You need the conjugate of e n which is the error y n minus y estimated and the conjugate of that one the original input data and that this is a step function in the both of this. We can select it by ourselves to see the convergence performance and we keep tuning our w to get our best performance till this y minus y n becomes minimum.

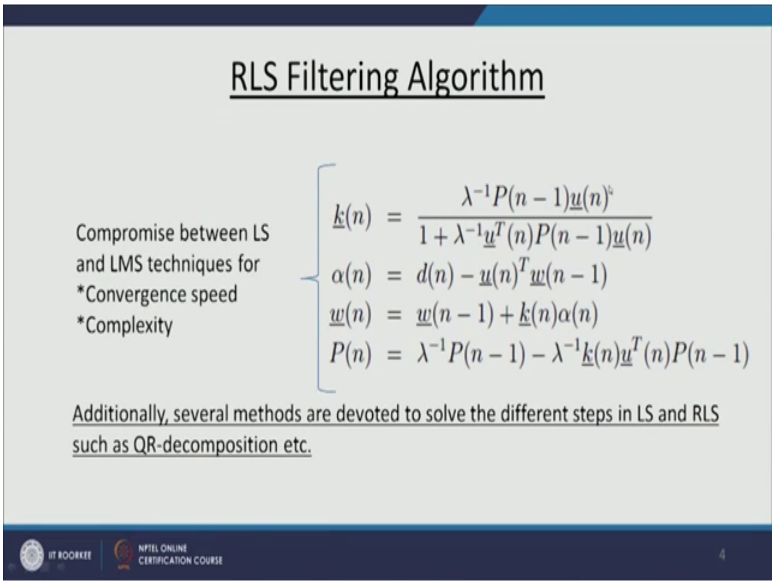
All these methods they are trying to minimize this mean square error, so basically if it is number of iterations , so because it is mean square error if we draw this with respect to w for particular value of w it should be able to get this minimum value of J mean. So, in least square they try to get in one step somewhere near here and gradient descent and LMS method they try to come here slowly stepwise. So, this is what we are getting here. Iteration wise least square is one global solution. So, whatever you will get in one step let us say with iteration you will have 0, 1 and one iteration you will have your least square solution for gradient descent and the LMS they will be reaching here slowly. So, if we I say LMS and gradient descent somewhere they will be working like this with many iterations.

So, we can see here the processing here is very much simpler does the multiplication of two terms and the addition and here little bit complex with respect to this one because you have to calculate the gradient. The benefit here say is that gradient descent method can actually reach near to this point, but in LMS because we are moving the gradient value the stochastic ah property is coming back into picture because of that it just keep rotating here and here in LMS methods. In least square method it is trying to reach here in one step.

So, these are these methods. These methods are simpler to implement less computation least square method you have to do the inversion you have to deal with the matrix. So, of course, it is more complex method. So, while keeping this in mind recursively square is a compromise between these two. So, if least square is getting here and LMS and GD are just reaching here RLS is giving you convergence somewhere in between ok. So, if it is LS, LS it is GDN elements then in between it will be RLS.

And again it is a compromise in terms of complicity also. So, quickly if you look at the equations instead of using simply one weight update coefficient update equation we have set of 3 four equations, but we are not using inversion here. It is simply the multiplication and division of the terms.

(Refer Slide Time: 22:53)



RLS Filtering Algorithm

Compromise between LS and LMS techniques for
*Convergence speed
*Complexity

$$\begin{cases} \underline{k}(n) = \dfrac{\lambda^{-1}P(n-1)\underline{u}(n)}{1+\lambda^{-1}\underline{u}^T(n)P(n-1)\underline{u}(n)} \\ \alpha(n) = d(n) - \underline{u}(n)^T\underline{w}(n-1) \\ \underline{w}(n) = \underline{w}(n-1) + \underline{k}(n)\alpha(n) \\ P(n) = \lambda^{-1}P(n-1) - \lambda^{-1}\underline{k}(n)\underline{u}^T(n)P(n-1) \end{cases}$$

Additionally, several methods are devoted to solve the different steps in LS and RLS such as QR-decomposition etc.

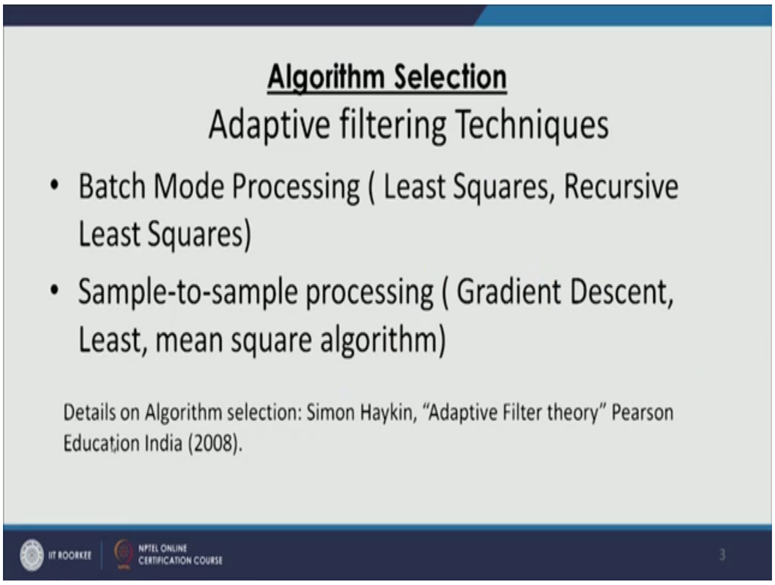
IIT ROORKEE | NPTEL ONLINE CERTIFICATION COURSE

4

So, in this kind of system we define our vector P by ourselves we choose some P and based on that P we define our k n, u n is our input signal here . Once we get our k value we calculate our alpha n value which is basically the error, if you see a desired value minus input signal into weight vector, so just the error. So, this k and error is multiplied with the previous value of k n and it looks very much like this method done apart from the fact that we have to calculate k n and p n here.

So, complicity is lesser than the least square method, but higher than gradient descent and LMS method and convergence speed is faster than these two. So, it is compromise between LS and LMS techniques for this one.

(Refer Slide Time: 23:49)



Algorithm Selection

Adaptive filtering Techniques

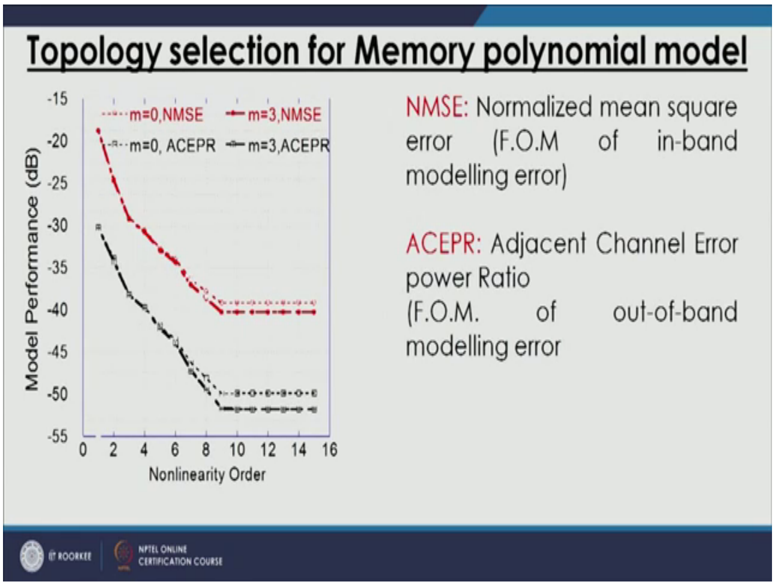
- Batch Mode Processing (Least Squares, Recursive Least Squares)
- Sample-to-sample processing (Gradient Descent, Least, mean square algorithm)

Details on Algorithm selection: Simon Haykin, "Adaptive Filter theory" Pearson Education India (2008).

IIT ROORKEE
NPTEL ONLINE CERTIFICATION COURSE
3

There are several methods in the algorithm wise which you can basically if you go through the Simon Haykin's adaptive filter theory from the Pearson education, India ah. This version is available in India. Then many details on this algorithms you can perceive from there.

(Refer Slide Time: 24:01)



Topology selection for Memory polynomial model

NMSE: Normalized mean square error (F.O.M of in-band modelling error)

ACEPR: Adjacent Channel Error power Ratio (F.O.M. of out-of-band modelling error

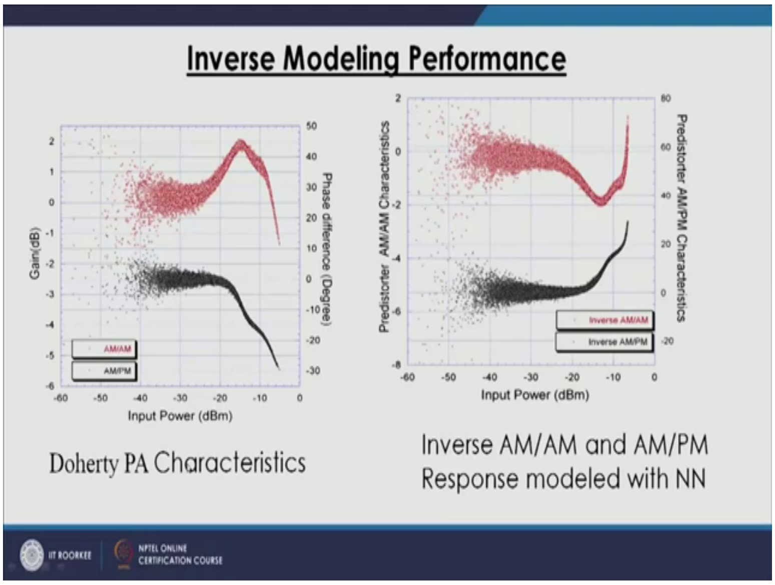
But basically once you select the topology this kind of algorithms you can select easily. So, you have selected the algorithms, let us say we have selected least square method and we have decided we will go with this method and we will select our topology based on this method.

So, for memory polynomial model we have to select memory depth and we have to select non-linearity order. So, how do we do that? We keep changing the nonlinearity order from 0 to some value let us say 20th, and we also keep increasing number of memory depth and we plot them together or we just observe then and whenever it is converging. So, it is a hit and trial method in this case we can see that if we keep increasing number of memory order non-linearity order sorry for m equal to 0 then our system response model performance in terms of ACEPR, we have defined ACEPR earlier adjacent channel error power ratio which is for the auto of band modeling error it is converging at nine ninth order. So, k equal to 9 should be able to give us good order.

Now, if we use m equal to 0 or if you use m equal to 3, we can see there is a difference of 2 3 db, when we use only m equal to 0 and when we use m equal to 3. So, m equal to 3 and n k equal to 9 is giving us good convergence. So, we can choose this value as our model topology, so m equal to 3 and k equal to 9.

Say at the same trend we can see for in terms of NMSE also we are plotting an NMSE for m equal to 0 which is this one and m equal to three which is this one and we can see that it is also converging at n equal to 9 and again m equal to 3 is giving much better than m equal to 0. So, we keep changing from equal to 0 to 1 2 3 and similarly from 0 to any value of non-linearity order or when it is starts to saturate or become even worse then we will stop at that point. Because if you keep increasing here we will get the same performance, but our complexity is increasing right, because we are increasing number of multiplications and additions, so it is better to stop at this point. So, this way we can choose our topology and then by chosen topology and algorithm we can apply our model.

(Refer Slide Time: 26:32)



Inverse Modeling Performance

Doherty PA Characteristics

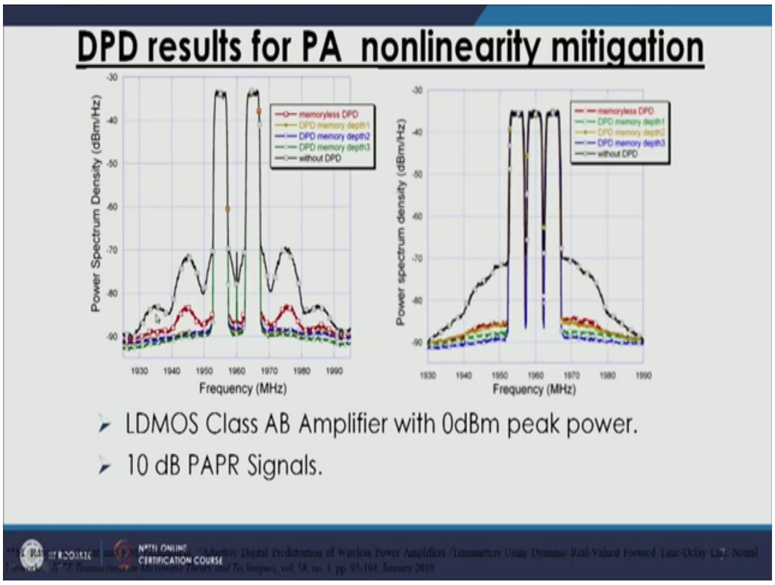
Inverse AM/AM and AM/PM Response modeled with NN

So, here for a practical system Doherty power amplifier we are showing the actual PA characteristic, so in the left hand side figure we can see that our AM, AM is given by this red curve which is the gain, and just normalize gain, so this around 0. We have removed the small signal gain so that we can just see its performance.

Similarly, phase difference which is AM PM it is again around 0 it is shown by black and we can see it is very non-linear system. So, by doing the inverse modeling means by using input as an output and output divide by small signal gain as an input as we discussed in the last lecture.

We again do the modeling using neural network and I am showing the modeling results here in the right hand panel and you can see that gain curve is this one which we received from the model and the phase term is this one which also we received from the model and if is even by our own eyes we can see that non-linearity profile is inverse of what was for the power amplifier. So, we have a hope that it should be able to correct for the PA non-linear to here. So, this is what we will see here.

(Refer Slide Time: 27:48)



DPD results for PA nonlinearity mitigation

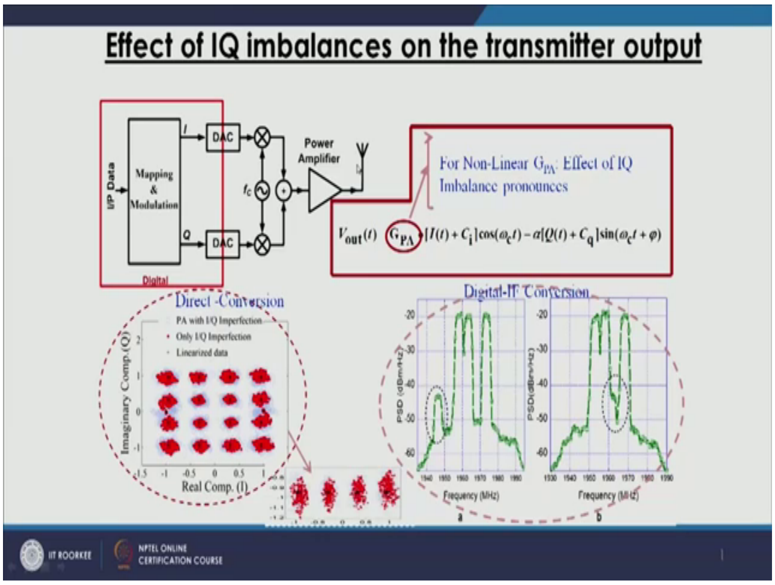
- LDMOS Class AB Amplifier with 0dBm peak power.
- 10 dB PAPR Signals.

**M. Rawat and F. M. Ghannouchi, "Adaptive Digital Predistortion of Wireless Power Amplifiers /Transmitters Using Dynamic Real-Valued Focused Time-Delay Line Neural Networks," IEEE Transactions on Microwave Theory and Techniques, vol. 58, no. 1, pp. 95-101, January 2010

In this case when there is no DPD, no distortion was there this black curve is showing power amplifier output and we can see that auto of band distortion is here it was the original signal. 3 carrier signal which has one missing carrier and third carrier, if 3 carriers are on for this wcdma signal we can see this is the profile and this is the auto of band distortion which can which is comparing because of the power amplifier.

Again we can see when we apply our DPD then for memory less model when I am we are using m equal to 0 in both the cases this red one is the result of the DPD. So, investment is working very perfectly there is ah reduction there, but if we increase m equal to 1, 2 and 3 we can see that it is even becoming better and better, both reduction in the auto of band destruction is happening in both the cases. So, this was an example with the LDMOS class AB amplifier and it was using the signal with the 10 dB PAPR.

(Refer Slide Time: 28:57)



Effect of IQ imbalances on the transmitter output

For Non-Linear G_{PA}: Effect of IQ Imbalance pronounces

$$V_{out}(t)\ G_{PA}[I(t)+C_i]\cos(\omega_c t)-\alpha[Q(t)+C_q]\sin(\omega_c t+\varphi)$$

Now, we can go to the effect of IQ imbalances ah. We had seen the effect of power amplifier non-linearity and how we can use DPD to compromise on that one. Now, apart from the power amplifier non-linearity we have discussed IQ imbalance earlier and because of that IQ imbalance we also see some extra additional distortion such as in the consolation diagram there is a tilting of the phase and the distortion which is more than the original signal constellation at the signal constitution point. So, we want to reduce this vector also.

We have discussed the image appearing because of the IQ imbalance and this is what is appearing here we can remove this effect also using this digital pre-distortion. And in the next class we will be discussing this effect and how we can reduce this by using our digital pre-distortion techniques.

Thank you.