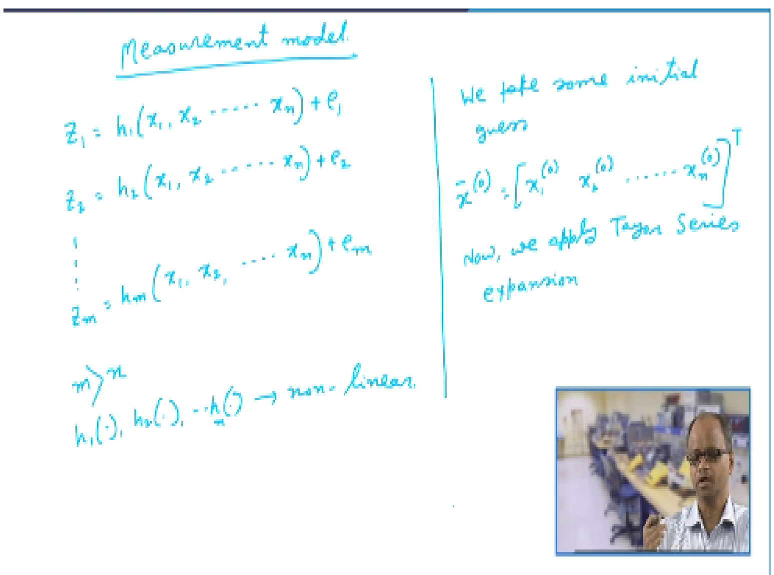
Computer Aided Power System Analysis
Prof. Biswarup Das
Department of Electrical Engineering
Indian Institute of Technology - Roorkee

Lecture – 46
Power System State Estimation (Contd.)

Hello friend, welcome to this lecture on the course on computer aided power system analysis, in the last lecture, we have started discussing on power system state estimation and we have looked into the measurement model for the purpose of power systems state estimation and we have seen that essentially the measurement model for a power system state estimation is basically a nonlinear one.

So, we will look into these details today, again so we are talking about power system state estimation, so we are talking about measurement model.

(Refer Slide Time: 01:12)



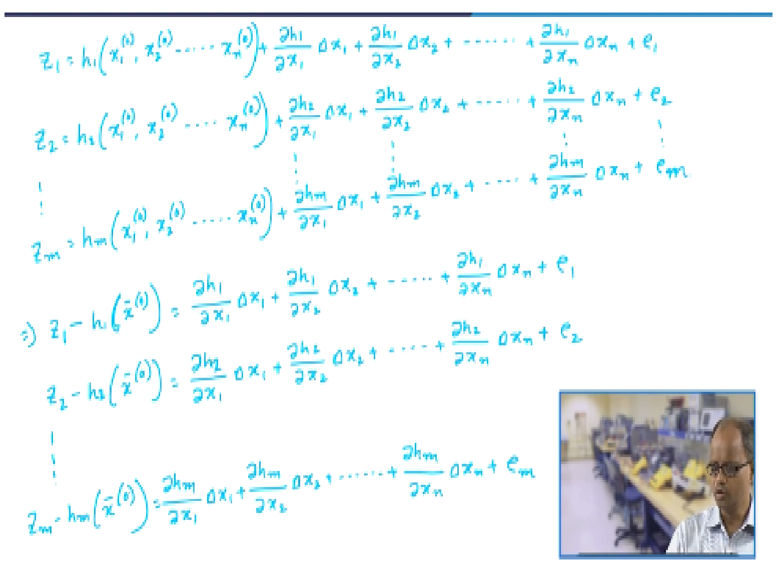
So, let us recapitulate that our measurement model is z1 = h1, so on x2 to xn + e1, z2 h2 x1 x2 to xn + e2 dot dot dot zm = hm x1 + x2 dot dot xn + em and we have seen m is the number of measurements and m is > n and h1 dot h2 dot hm, these are all nonlinear, so now the question is that how do you tackle this nonlinear equations now, if we do recollect our discussion on load flow analysis we have seen that when we try to solve for in nonlinear equation.

So, there is a no closed form solution available, so what do we do actually, we actually, basically do an iterative process starting with an initial guess, so here also we would be doing this iterative process, starting, I mean with initial guess and then what we have done that essentially that at a particular operating point that is at the initial guess, we do take the Taylor's series expansion and then after that because of this Taylor series expansion, this nonlinear equations are converted into linear equation which we can solve utilising any linear equation solution methods.

And then you simply repeat this process, so here in this case also we what we will do that we take as some initial guess and that initial guess is given as let us say x0 0, which is nothing but x1 0, x2 0 dot dot dot dot xn 0 transpose, so with this initial guess now, we apply Taylor series expansion, so what do we do; we apply now, we apply Taylor series expansion to each of this equation, we only take as we have done also in the case of a load flow analysis.

In the case of load flow analysis, we have taken only up to first order Taylor series expansion, so here in this case also, we will be taking only up to first order Taylor series expansion, so if we do take the first order Taylor series expansion, we get for every equation something like this.

(Refer Slide Time: 04:59)



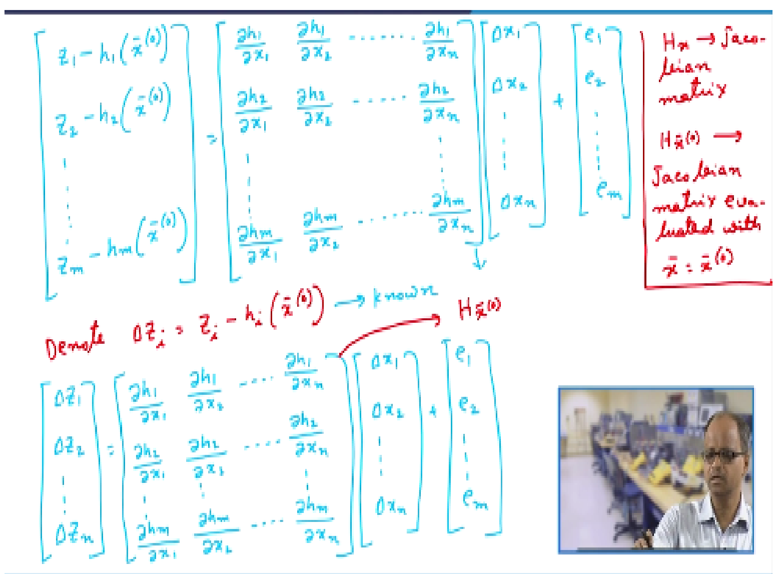
So, z1 is given by h1 x1 0, x2 0 dot dot dot xn 0 + del h1 del x1 * delta x1 + delta x2 + dot dot dot del h1 del xn * delta xn, we only take up to first order Taylor series expansion + e1, then we also do the same thing for z2, so we take h2 x1 0 x2 0 dot dot dot xn 0, that should be on bracket

+ del h2 del x1 to delta x1 + del h2 del x2 delta x2 + dot dot dot del h2 del del xm * delta xn + e2, similarly dot dot dot zm = hm x1 0, x2 0 xn 0 + del hm del x1 * delta x1 + del hm del x2 * delta x2 + del hm del xn * delta xn + en; em sorry, em.

So, then therefore we can write down this equations as something like this, so you can write down this equations as z1 – h1 x0 that is x vector 0 = del h1 del x1 * delta x1 + del h1 del x2 * delta x2 + dot dot dot del h1 del xn + e1, similarly z2 – h1; h2 * x vector 0 = del h1, sorry del h2 del x1 delta x1 + del h2 delta x2 + dot dot dot del h2 del xn delta xn + e2 and similarly, zm – hm, del hm delta x1 del hm + em.

So, now please note that would this quantity is nothing but the function h1 evaluated with the vector x0, similarly this quantity is the function h2 evaluated at the vector with the vector x0, similarly this quantity is the function hm evaluated with the vector x0.

(Refer Slide Time: 09:53)



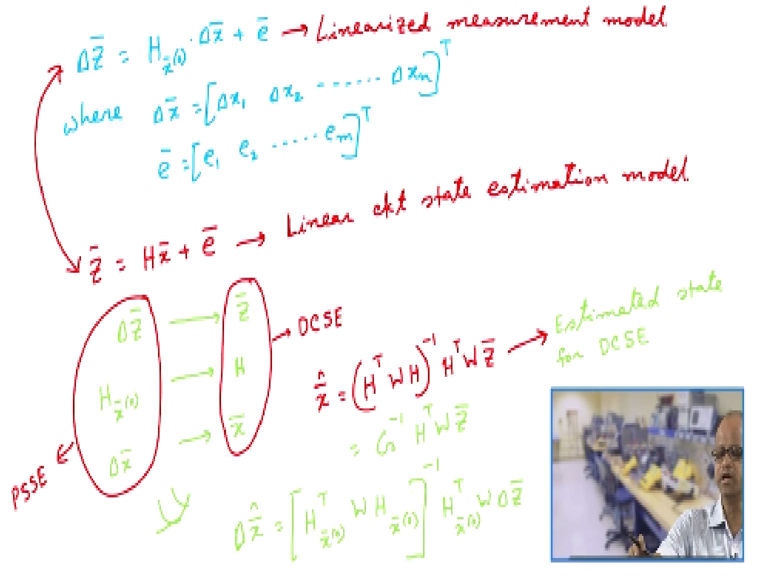
So, then therefore these equation can be written as; so then therefore if I write this equation in a matrix forms, so what I get; I get h1, z2 – h2 dot dot dot, zm – hm, this vector is known, what would be this one; this would be del h1 del x1 del h1 del x2 dot dot dot del h1 del xn, it will be del h2 del x2 del h2 del x2 dot dot dot del h2 del xn dot dot dot del hm del x1 del hm del x2 dot dot dot dot dot del hm del xn, so we are simply delta x1 delta x2 delta xn + e1 e2 em.

Now, suppose denote delta zi = zi – hi * x0, so then therefore this quantity, so this about equation can be written as, delta z1 delta z2 delta zn, please note that this quantity is known, this is a known quantity, it is a known quantity because we have got this measured value and this is nothing but the evaluated value of the function hi with then initial assumption x0, so therefore and if I do evaluate it at x = f0, so, sorry and then if we do evaluate it at x = x0.

So, then therefore this particular matrix is also known, so we again write that del h1 del x1 del h1 del x2 del h1 del xn, similarly del h2 del x1 del h2 del x2 del h2 del xn, similarly del hm del x1 del hm del x2 del hm del xn, delta x1, delta x2 to delta xn + e1 e2 en, now this matrix we let denote as, now let us denote this matrix as hx0, what at this matrix, hx0, that means, why I mean, we are saying that this is an hx, hx denotes that this is the matrix or rather this is basically, Jacobian matrix which calculates the partial derivative of all the measurement functions with respect to the stare vector n.

So, then therefor hx is actually, hx we denote as the Jacobian matrix, it is the Jacobian matrix, right, h is denoted as Jacobian matrix, I mean what does this Jacobian matrix do; it simply calculates the partial derivative of all the measurement functions with respect to the state variables and hx0 is basically the Jacobian matrix evaluated at; evaluated with x= x0, right. So, I should probably use a better notation, I am saying Hx 0, you should probably use a better notation.

(Refer Slide Time: 16:17)



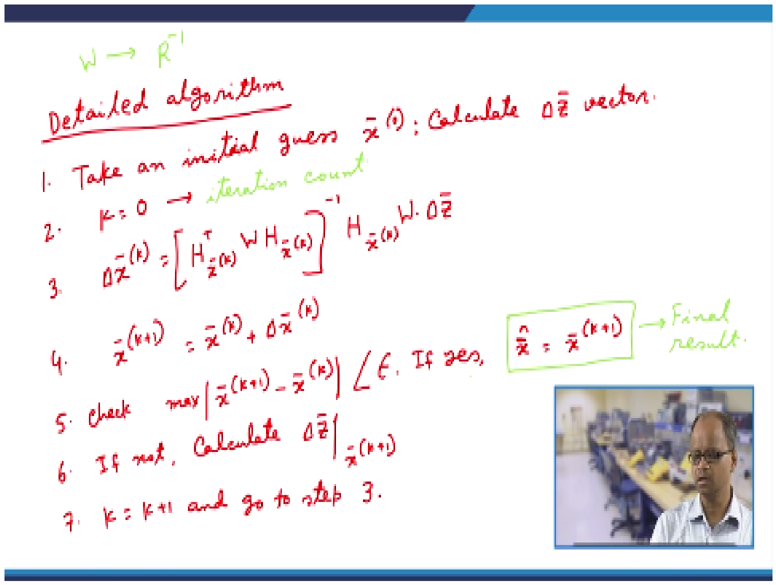
Let us say Hx0 and Hx0, so this is Hx0, so Jacobian matrix is value to x = x0, so then therefore now, this entire equation is can be written as; therefore this entire equation can be written as delta z vector = Hx0 matrix * delta x vector + e vector, where delta x is nothing but delta x1 delta x2 delta xn transpose and vector e is e1, e2, em transpose, so then therefore, this is nothing but the linearized measurement model, so this is the linearized measurement model.

Now, if we are recall our earlier measurement model which we have discussed in the earlier lectures for a simple DC circuit, our earlier measurement model was z = H * x + e, so this is the linear circuit state estimation model, right now, if we compared this equation and this equation we find, we simply find that delta z vector plays the role of z vector, here Hx 0 matrix plays the role of H matrix here, delta x vector plays the role of x vector.

So, this side is power system state estimation, this is power system estimation and this side is DC circuit state estimation; power system state estimation, DC circuit state estimation, so there is a very direct correspondence between this equation and this equation and for let us say DC, AC or rather they let us say this a DC circuit state estimation, we know that the estimated vector is H transpose WH inverse * H transpose W * z, so that was the expression.

So, this is the expression of estimated vector, this is the estimated states for DC circuit state estimation, right, so H transpose W, H H transpose * W Z, so this is the inverse of, this is the G inverse H transpose W * z. Now, so if I just, so now because there is a direct correspondence between this and this, so then therefore from here we can simply write that for power system state estimation, my expression of delta x hat would be straight away H transpose * W * H transpose whole of matrix inverse, then H transpose T * W * delta z.

(Refer Slide Time: 21:18)



So that would be the expression for the PSSE, so this would be the expression for PSSE, now in this expression, what is the W; W is the let us say R inverse matrix as usual and we have already explained that this is an R inverse matrix, now in this expression now, but although there is a direct correspondence between this and this but then there is a major difference, in the case of DC circuit state estimation, we have got the estimated states directly that is this particular formula or this particular expression gives us the value of the estimated states.

But here in this case, this is not the estimated state, this is rather the correction in the estimated state, now because it is the correction in the estimated state, so then therefore what we will do; so then therefore, after we calculate this correction in the estimated state, we have to update our estimated state for the next iteration and then we proceed, so then therefore what we will do after we get this delta x vector, we will update our x vector as x0 + delta x vector.

And then we will again repeat this state estimation exercise through an iterative process (()) (22:39) achieve convergence, so then therefore let us look at the detailed algorithm, so now; so we now can talk of a detailed algorithm of course, we will be omitting several details which we have already explained, so we would be only noting down those pertinent points which we have not so far discussed.

So, first thing is that take an initial guess x0, right and also set the value of epsilon and then we set k = 1, so that is the iteration count, this is actually the iteration count, so this k =1 is nothing but the iteration count, so in the third step what we do; we calculate delta xk, in fact we should k = 0, it can be 0, so we calculate delta xk = Hx k * W H transpose * H xk whole of inverse then H xk * W * delta z, you can also write calculate delta z vector, right.

Or then what do we do is; then you update x k + 1 = x0 +; sorry xk k + delta xk, right, then we check whether mod of xk – xk, if it is should be max, check maximum is less than epsilon or not, if yes then x = xk + 1, if not then what we do; if not we calculate delta x with xk + 1, right at x = xk + 1 and then we update 7, we update k = k + 1 and go to step 3, so this is the detail algorithm, so basically what we do that we first take an initial guess, calculate this delta z vector.

Then with this delta z vector and with this x0 vector, we also evaluate this Jacobian matrix and when this and with this Jacobian matrix and this, so then after that we calculate this correction in the estimated values, then we updated this correction, then we check whether my updated values are converged or not, if we will take this epsilon to be very less, let us say 10 to the power -6 or 10 to the power -12.

And so then therefore, if there is no change in this vector between 2 successive iterations, so we can take that this particular state vector has been converged, so if it is yes, so then we simply, so I mean this is our final results and if not, so this is our final result, and then if not, so then again we calculate delta z with actually x, I mean with this updated vector, once you calculate this delta x with this updated vector, then you update this k = k + 1.

And then again, we repeat this process till this convergence is obtained, so this is the basic procedure, this is a simple procedure but now what is remaining that now you have to look into the structure of this Jacobian matrix depending up on the various types of measurements, so that issue would be looking into the subsequent lectures, thank you.