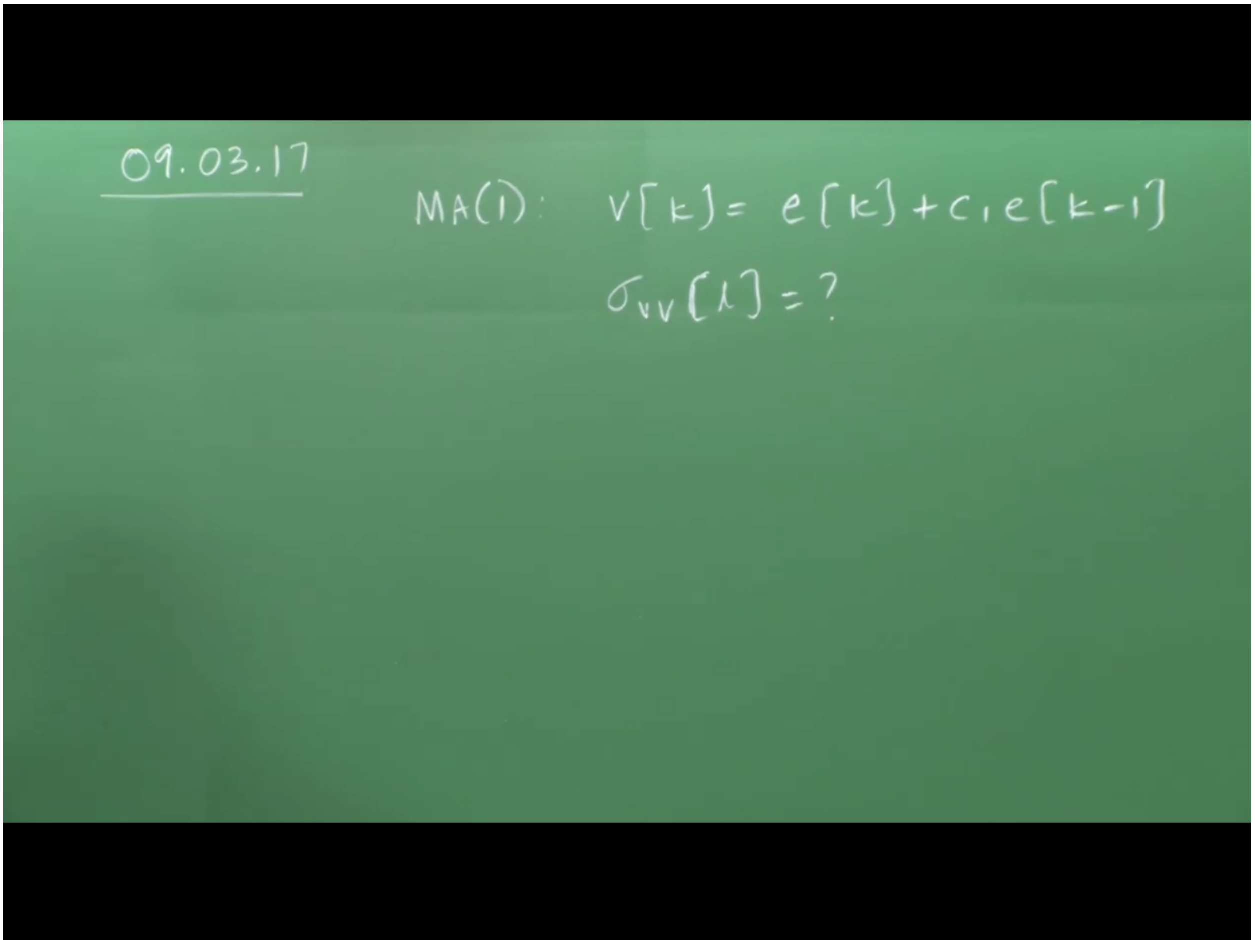
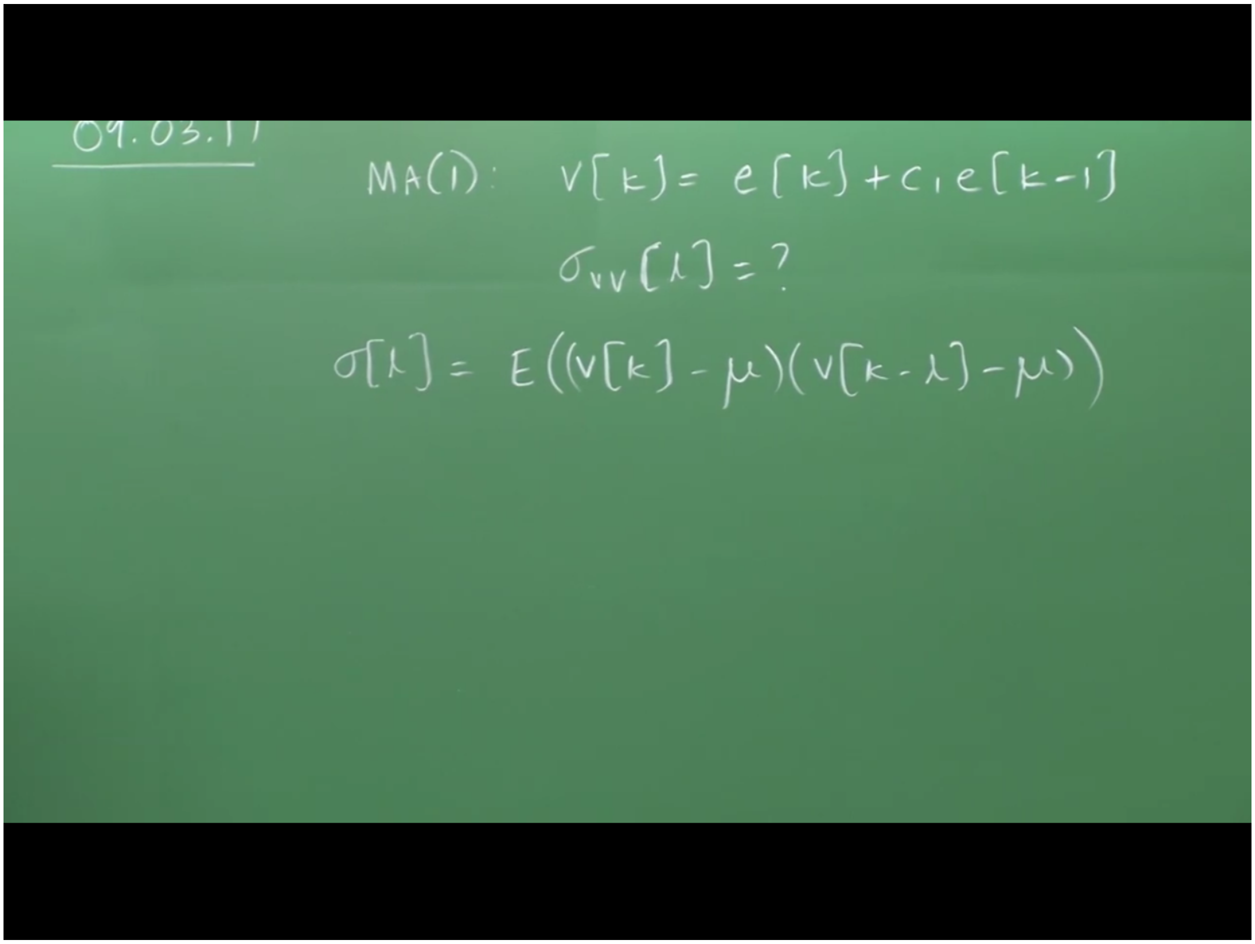
Okay, very good morning. As I said yesterday we will continue our discussion on correlation functions and hopefully close the curtains on it. I will also show you a MATLAB demonstration of how to simulate an AR or an MA process and how to estimate it as well. I will give you the list of MATLAB commands as well towards the end of the lecture. Now one of the things that we learnt yesterday is that these models, that is ARMA models they are… the way they are identified is by first looking at the ACF and PACF signatures. So when you are given data, as you will see in the illustrative example today also, you will notice that we will plot the ACF and PACF. And yesterday I showed you, what are the theoretical signatures for ACF and PACF of AR and MA processes. For ARMA processes, there is no such signature available, that is, all that we know is that both ACF and PACF decay, but when it comes to order determination of the AR and MA components, there is no provision as of now, one has to go by trial and error, of course maybe it's possible to turn to the state space methods, but we will not go in that direction, as of now. Now the general thing that one should be equipped with, although ultimately we are going to deal with data, the reason for discussing this much theory is that you should be an expert in both theory and practice, because after all whatever, tools you are going to use in practice are based on the theory. Now what this also means is that, if I am given the model of a random process, maybe I am given an MA model, let us say, I should be able to derive the theoretical ACF. In practice we don’t use this, in practice we estimate AC. But suppose I want to carry out some theoretical analysis for… for a given MA process and I want to determine how the ACF looks like. Then there are two ways of looking at it, one is just to apply the definition of ACVF to the given model and then evaluate the resulting equations. The… although the final goal is to arrive at the ACF, generally we begin by estimating ACVF, that is auto covariance and then move on to auto correlation.
(Refer Slide Time: 03:49)



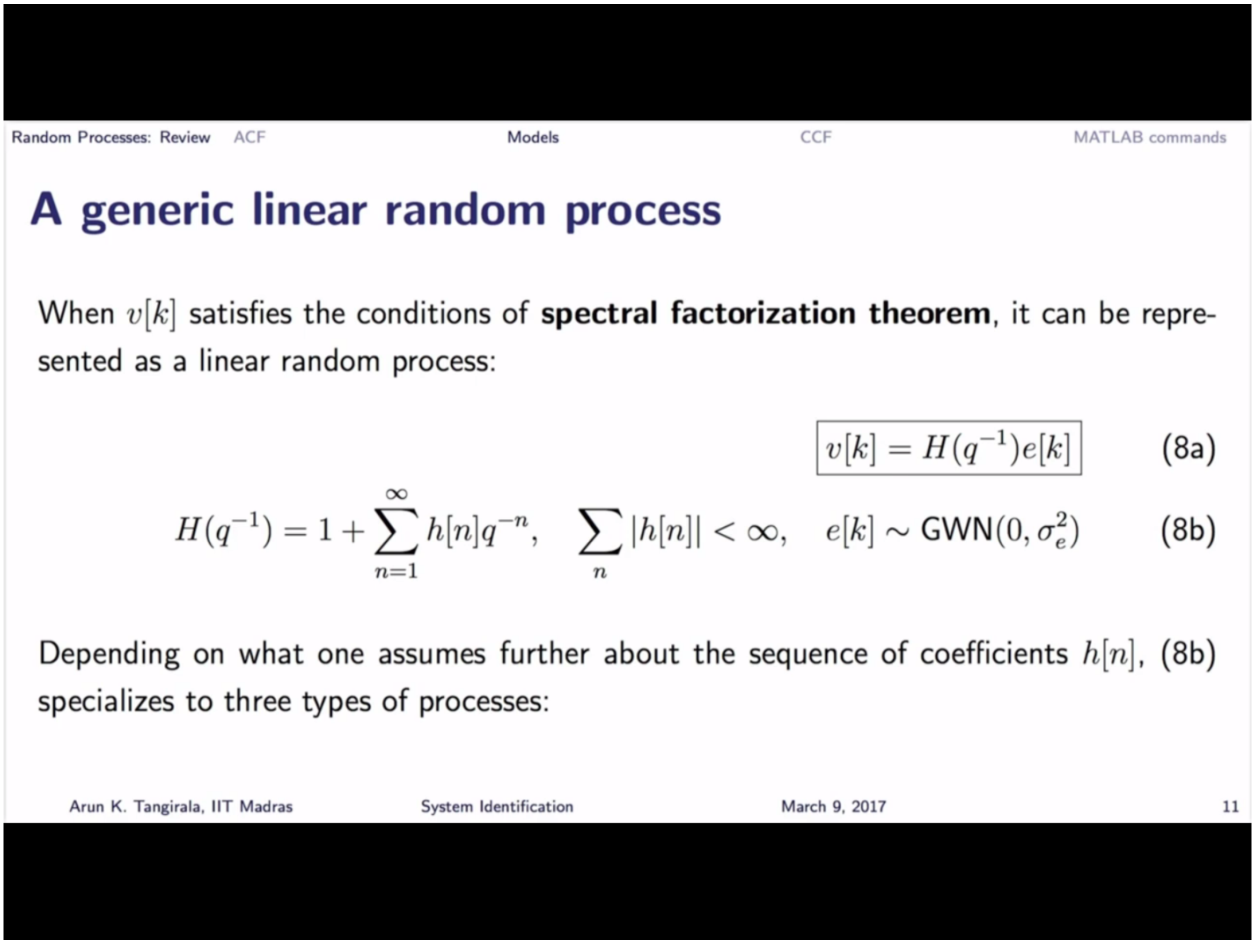
So as an example, suppose I am given MA1 process, where the governing equation is VK = EK + C1 EK-1 and I have to determine the theoretical ACVF of this MA1 process, as I always say, when in doubt go back to the basics, proceed in a step wise manner. Assume now EK is 0 mean, generally the white-noise processes that we deal with, we assume to be 0 mean, if they are not, then you can always include them in your model. So EK is given to be of 0 mean and of variant sigma square E, C1 is some coefficient bounded in magnitude. Now I want to find out, what is the ACVF of this kind of a process. How do I proceed, I start with the definition of ACVF of this kind of a process. How do I proceed. I start with the definition of ACVF. What is the definition of ACVF. I am not going to write the subscripts, just to save time and chalk. So sigma L is expectation of VK-Mu times VK-L-Mu. Now of course we are given is E0 mean, the first thing that we should be assured is that the given model is stationary.

(Refer Slide Time: 04:20)



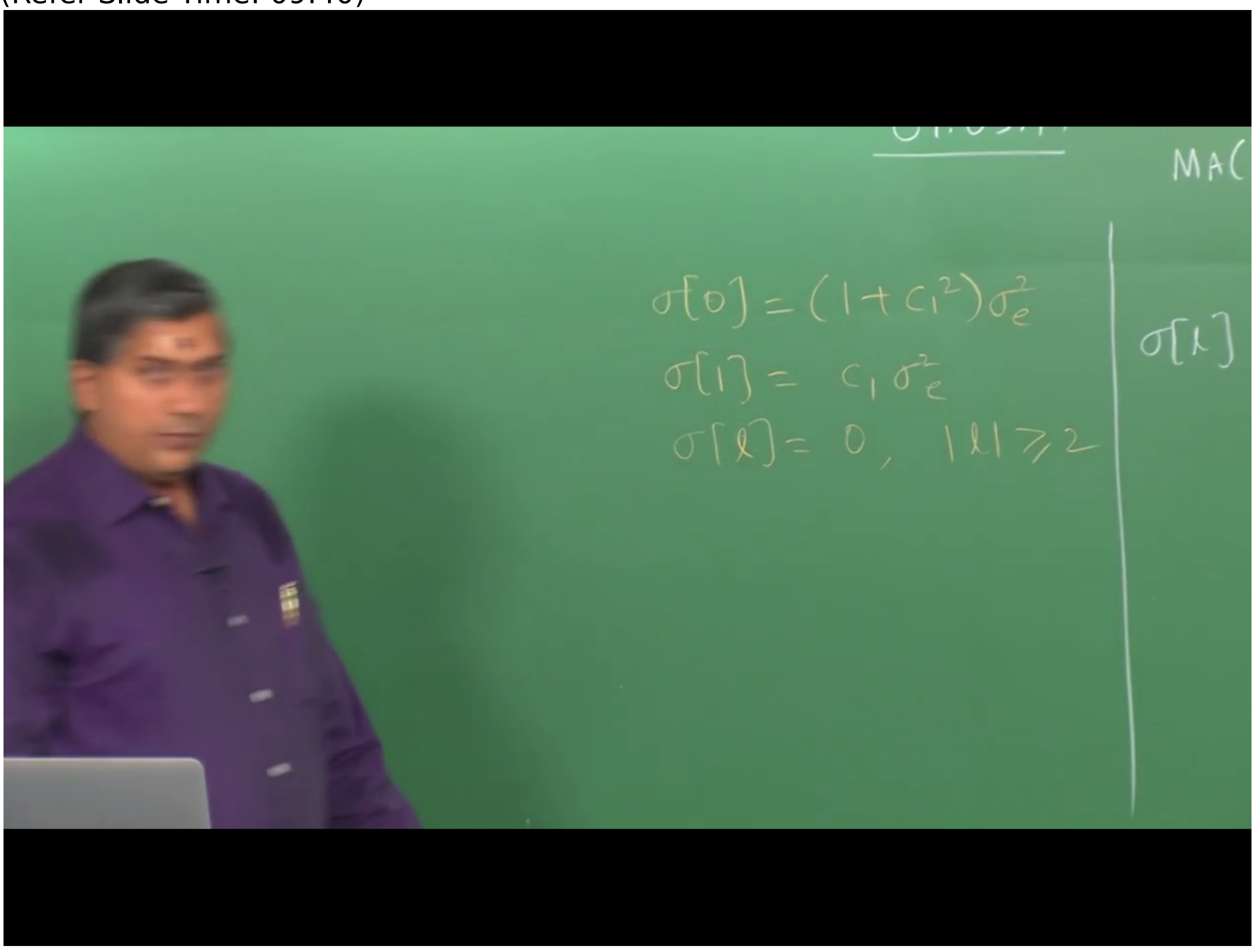
Now what… what is the condition under which his MA model is stationary? Is it stationary for any, suppose I say C1 is some constant, finite value constant, then is VK stationary. What are the requirements. Go back to the linear random process, right. The liner random process says the coefficient should be absolutely convergent, right. Now what are the coefficients that we have, the impulse response coefficients 1 and C1, nothing to worry about. You can straight away see that it is stationary.

(Refer Slide Time: 04:59)



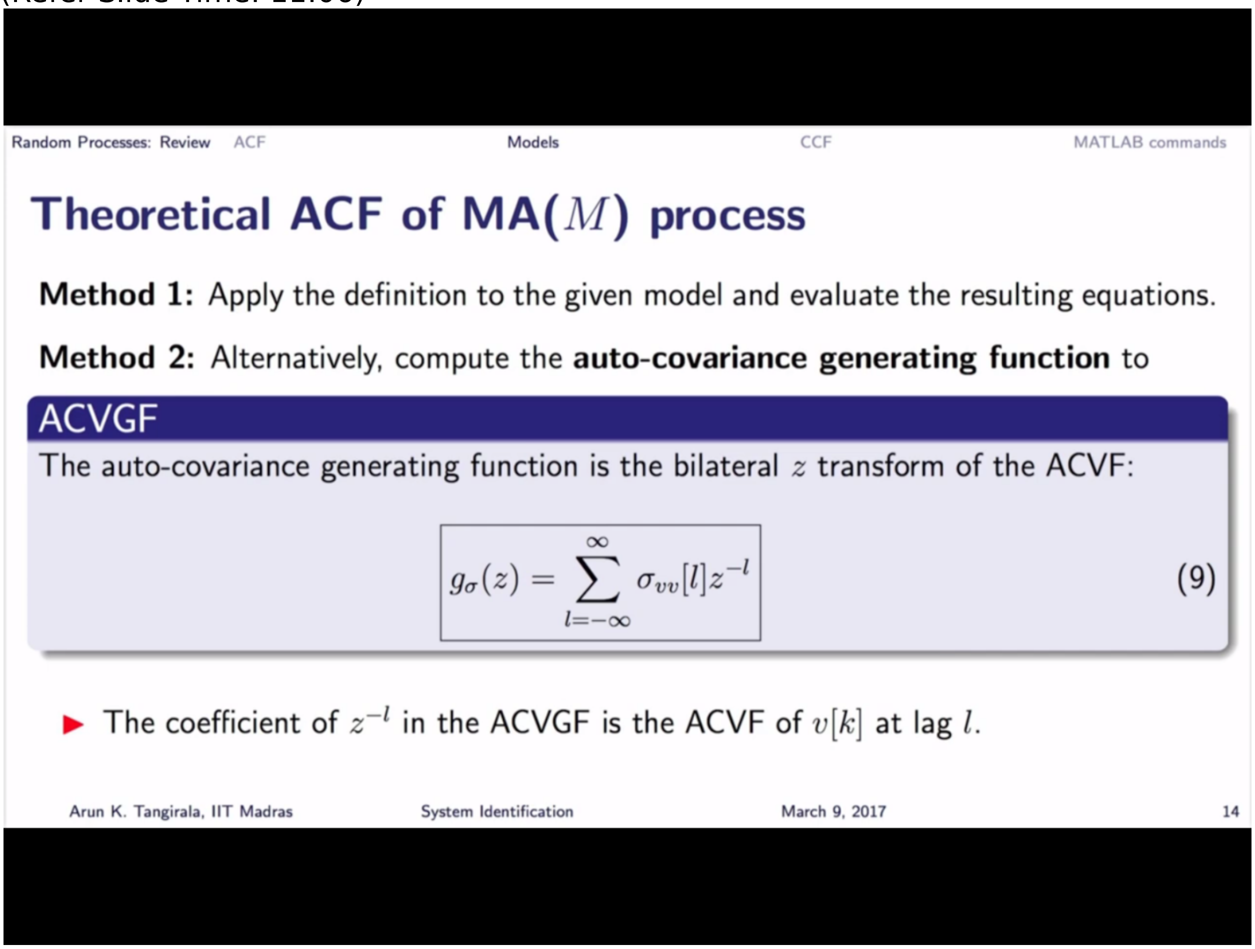
and now you can evaluate the mean, in any case evaluate the mean. What is the mean of VK, 0, because expectation of VK is simply the expectation of the right hand side and we know EK is 0. So Mu E is 0, we know that. Therefore I can now simply write this as expectation of VK times VK-L, as I always say you should be very comfortable with computing these covariances, not only these auto covariances, but in general cross covariances and so on. So now what is the story. There are many ways of doing this, the best way is to just simply substitute. When it comes to MA processes, in this procedure it is best to substitute for the given governing equation. So your given VK is EK+C1 EK-1, plug in here. And then you will have a product of terms. How many terms will you have in the next step, when you evaluate the product of terms, four terms, correct. So you would have, what would be the first term, expectation of, what would you have, expectation of EKE K-L and then + C1 expectation of EK-1 EK-L done. The third term, third term would be... what would be the third term? Well different ways of writing it, so EK... C1 times EKE K-L-1 and what about the fourth term, +C1 square EK-L sorry EK-1 times EK-L L-1, right. Now generally in a time series course, I go by showing, deriving the expression at L=0, L=1 and so on, because... because we are in a review mode, as straight away written the generic expression at lag L. Now when we say we want to compute the ACVF, we want to know the values at lag 0, Lag 1 and so on. Now we can use the property of white-noise sequence to arrive at the answers for sigma 0, sigma 1 and so on. So let's write here, sigma 0, what would it be, plug in L=0, right, what is the first term, what's the first term, when L=0 and I evaluate the expectation... sigma

square E, very good. Second term, why, because by definition white-noise is uncorrelated with its past or future, whatever. So expectation of EK-1 times EK is 0. That is the auto covariance of white-noise is 0 everywhere, except at lag 0 and at lag 0 its value is sigma square E. So the second term vanishes. The third term also vanishes. What about the fourth term C1 square sigma square E. So I can straight away say it's 1+C1 square sigma square E. This is the variance of an MA1 process. In fact now if you stretch your imagination, if you had an MA2. You should expect to see 1+C1 square + C2 square, sigma square A. Essentially the cross terms, all of them will vanish, only the light terms will prevail. Alright, what about sigma 1, the auto covariance at lag 1, again plug in L=1, which of the terms will prevail? Here when I plug in L=1, this term prevails, but this, the rest of the 3 will vanish and therefore I have C1 sigma square E. I don't have to evaluate L= -1, in fact you should see when… at L= -1, you will get the same answer, because auto covariance is symmetric. What about at lag 2, lag 2 all the terms will vanish, as you can see. In fact I can straight away say beyond la1 the auto covariance vanishes.
(Refer Slide Time: 09:40)



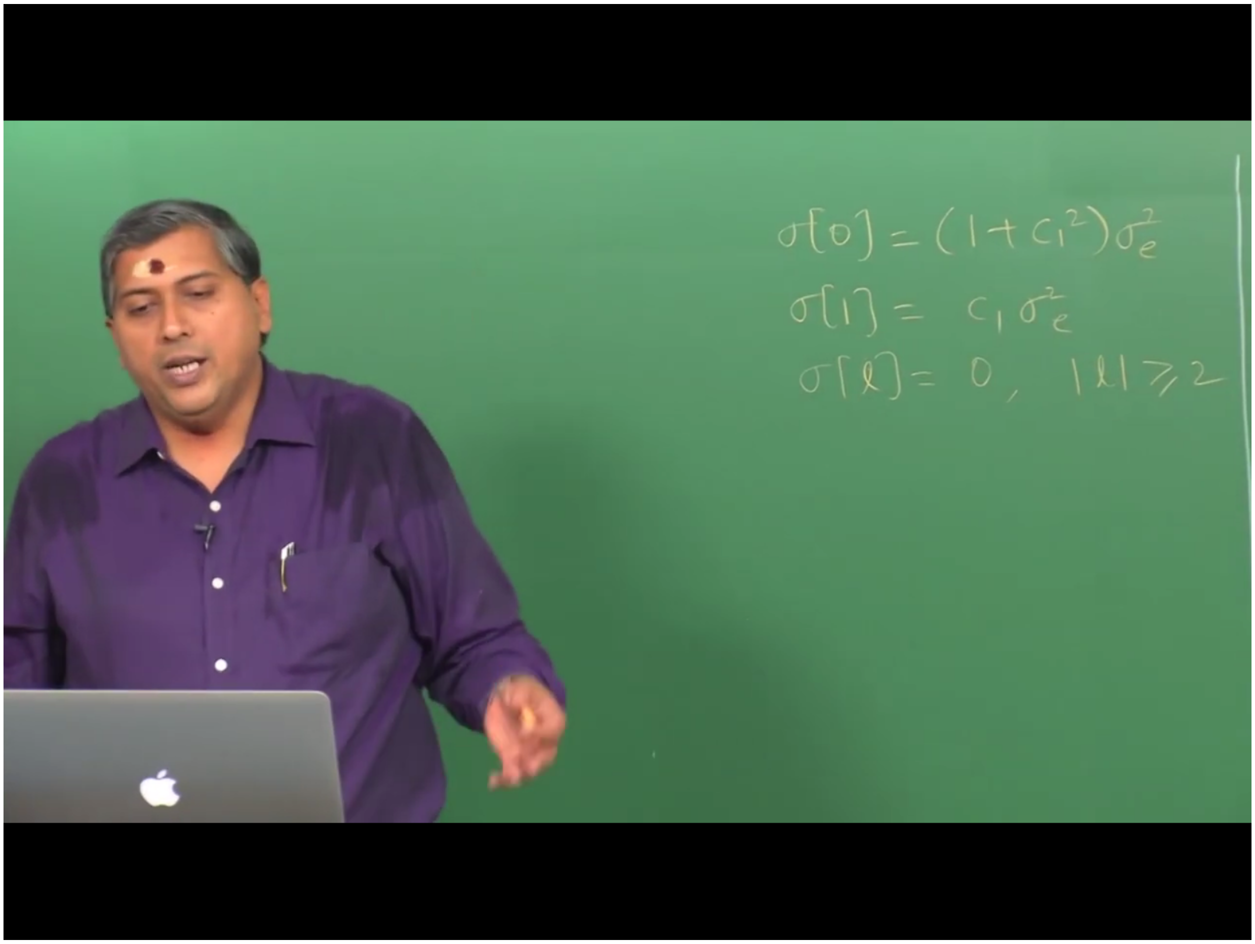
This is exactly what we said yesterday, right. The auto covariance dives down abruptly after which lag the order of the MA process. Now this procedure for computing, sorry, this procedure for computing the ACVF of an MA process is okay in this is in for first order, second order, and so on, but then the number of terms that you have to write down increases as you go to higher order MA processes. So there is a second method of computing ACVF, which is through the auto covariance generating function. Now this auto covariance generating function is in general applicable to any linear random process, but very well suited for MA processes, which means you can apply this method of auto covariance generating functions to AR processes as well. However, it becomes a bit cumbersome to use it for AR process. So what is this auto covariance generating function, it is nothing, but the two sided Z transform of the auto covariance function, right. If you look at the equation here, equation 9, clearly says that the auto covariance generating function is a two sided Z transform of the auto covariance function. What is the purpose of this, now the purpose of this here, given, yeah I think I missed out showing this, that's the most important result, I will add that slide.

(Refer Slide Time: 11:06)



With the auto covariance generating function, you can show that this auto covariance function that you have, D sigma sigma Z is actually nothing but H of Z inverse times H of Z times sigma square E. It is... the equation is missing here in slide. So the definition is given in equation 9. From this definition you can show that the auto covariance generating function is related to the model. So how do you connect these two now, I mean how do you make use of these two, given an MA model or in fact any model, you first find out the auto covariance generating function. So here what is H of Q inverse for MA1 1+C1 Q inverse. Now H of Z inverse is simply obtained by replacing wherever all instances of Q inverse by Z inverse, that is what H of Z inverse stands for. And H of Z, be careful, is the same story, wherever you find Q inverse, you replace that with Z, alright. It's not that you re-write 1+C1 Z inverse in terms of Z, there is a difference. Now when you multiply these two here, what do you get? I mean when you evaluate this, therefore G sigma of Z for MA1, it is for MA1 turns out to be 1+C1 Z inverse times 1+C1 Z times sigma square E. Am I right, and what… so what are the terms that you get. Sorry… So you get, when you multiply these two polynomials, you get 1+C1 square + C1 Z inverse + C1 Z and then of course times here sigma square E. What is the final step now? All you have to do is, if you want ACVF at lag L, you look at the coefficients of Z to the minus L, because that is what the definition says. You should not just remember it as a two sided Z transform. The other way of remember it, this auto covariance generating function is the coefficient of Z to the –L in the auto covariance generating function is nothing, but the auto covariance at lag L. So what is it… how do you read of the auto covariances from the generating function. At lag 0, I have to search, I have to look at the coefficient of Z to the power of 0 and that is 1+E1 square times sigma square E, exactly the answer here. And for the rest of the lags, it is pretty obvious there. If your answer is correct, you should get symmetric results, that’s a quick check. And with this auto covariance generating function, you can straight away prove also that in general for an MA M process, you will not have auto covariances beyond lag M. You just plug in a generic MA M process and evaluate, you will see that you will not have any powers beyond Z to the –M or Z to the M, which straight away goes to prove that ACVF goes to 0 for an MA M process after lag M. This is a more powerful method, but you should also be conversant with this approach, because many a times you may be dealing in some other models, so you should be able to take expectation, this is how you theoretically evaluate ACVF of MA processes.

(Refer Slide Time: 14:58)



How about AR processes. No with AR processes, it’s a bit more involved, but it gives rise to a set of equations that are very nicely positioned, also very… they have… they have other benefits also. Lets quickly talk about it for about three minutes. Now when it comes to ARP process and I want to find out the theoretical ACVF, you again begin with the definition. First of all, how is the… what is the definition of… sorry what is the governing equation for an ARP process. So let me actually show you for an AR1 process, because it's very easy to show, for other processes also it's quite easy, but it saves time. So let's take an AR1 process. What is the governing equation that I have VK+D1 VK-1=PK, fine. Now I… what do I want to find out, the theoretical ACVF, how do I proceed. Again you start with the definition. Before we proceed with the definition, we know that we have to find out the mean, because that’s required in computing the ACVF. Now when I have to compute the mean here, how do I actually compute the mean. I have to assume stationarity. There is no guarantee that this AR model will give me stationary VK. What is the guarantee, I… there are many ways of looking at it. The best way is to recast this AR1 in the standard linear random process framework, that is re-write this difference equation as a convolution form. Once you rewrite this difference equation as a convolution form, you can apply the condition of absolute conversions of the coefficients. So what is the way for… of re-writing this difference equation convolution form, we have learnt that in the deterministic world. So you just doe a long division and write an infinite series. You will see that VK would be written as essentially Ek+sigma-D1 raised to some J EK-J J running from 1 to infinity. Alright, now for this sequence to converge, your… basically 1+D1+D1 square up to this infinite series, geometric series has to converge, because that's what the absolute conversions condition in the coefficients is requiring you to do that. When will that happen? D1 is less than 1 in magnitude, right. This series will converge when D1 is less than 1. So an AR process need not necessarily give rise to stationary VK by default, whereas an MA process will give rise to a stationary VK, so long as a… constants are finite value. So there is no such issue with MA processes. Whereas with AR models, not all auto regressive models give rise to stationary outputs or stationary responses, only when the coefficients of the AR model satisfy some condition, you are guaranteed stationarity. Now here we are able to say, D1 is less than 1 in magnitude, but the general result is not on the coefficients, what is the guess. We have talked about stability in the deterministic world for transfer function models. Poles should be within the circle, exactly the same condition applies here. All AR models are stationary, that is hey give rise to stationary responses or stationary signals, if and only if the poles of AR model are within the unit circle. When we say poles is assumed, you are going to calculate in terms of Z, not Z inverse, clear. So as I told you that's the beauty, the moment we have realized that there is an LTI representation, you can bring in all the theory that you have learnt. There is nothing new. In the deterministic world we use the term stability and in the random world we use the term stationarity. So you can think of those analogues. So any AR model given you to, you should ascertain that it is stationary, if it isn't, you have to assume that it is stationary and state the conditions. Even having said, after having said this many students tend to assume that higher order AR models are stationary, only the rest… individual coefficients are less than one in magnitude, which is wrong. The restriction is not on the coefficients directly, the restriction is on the ports. So for AR2 don't think it is stationary only D1 and D2 are respectively less than 1 magnitude each, that is wrong. AR1 you know it's only single coefficients that happens to be the pole and therefore the condition directly translates to that on the coefficient. So please keep that in mind. So with this assumption, we will assume that this are the only admissible AR models, for this admissible AR models, what is the ACVF, okay. So how do we proceed now, now that we are given this, we are assured that VK is stationary. Since VK is stationary, it's mean must be constant, so what happens to the mean now. How do you evaluate, what are the mean, you are given EK is 0, what is it so difficult? They are two ways of doing this, you are… given it is stationary now. You can directly go to this equation, right. When in doubt, just apply the definition, what is mean expectation of VK. Please make that a habit, when in doubt go back to the definition and start applying the definition. So you apply the definition here, you say expectation of VK+D1 times expectation of VK-1 = expectation of EK. Where given VK is stationary. So what happens to Mu V 0, that's one way of looking at it, the other way of looking at it is by looking at this expression here. Expectation of VK is expectation of the right hand side. We are given the sum will converge and expectation of EK is 0. So both ways confirm that Mu V is 0, but only after taking into account the stationarity requirement, you can do this. So AR process it's a bit more involved as you can see for MA process, it was so easy, but don't worry all the sweat that you break in theoretical analysis of AR models like is made very easy when it comes to estimation of AR models involves only linear Lesqui problem solution , whereas estimation of MA models involves solving a non linear Lesqui's problem. So there you have to break this right, here you are breaking it. So there is a right balance, life is fair, who said life is not fair. Theory, if… if something in the theory is difficult, then practice is made easy. Alright, so now let's move on and look at how to derive the theoretical ACVF quickly. How do… how do I proceed now, same story, but what we will do is, because I know what… what's going to happen, we are going to do this at every lag, sigma 0, sigma 1 and see what happens. So sigma 0, how do I obtain sigma 0? You look at this equation here and rewrite this as VK= -D1 VK-1+EK. For AR process is there is a good way of beginning, it is not necessary that you have to do it, but it’s a good and easy way. Now sigma 0, I know is expectation of VK times VK because I am given, it has… VK at 0 mean. Alright, now what do I do, do not try and plug in the expression for VK. Like you did in MA1, you will be caught in an infinite loop. You will go… keep going and going and you entire quiz, the end semester… your entire semester will go out. The trick here, the simple trick is, simply multiply both sides with EK and take expectations. Don’t think that we are just multiplying and doing something, we are taking… multiplying and then taking averages, expectations. So when I multiply right hand side with VK and take expectation, what is the first term that I get? -D1 very good, simga 1 and then I have a cross covariance between EK and VK-1 am I right. So I have here expectation of EK times VK-1. What can we say about this. Here is where, you just have to apply a very simple logic to evaluate the second term. If you do not do that, again there is a every chance that you will be caught in a loop. Look at this cross covariance carefully. What is it asking, what is covariance after all, dependence between two variables, random variables. So here you have expectation of EK with VK-1 and now the question is whether EK influences VK-1 and if it does by how much, look at VK-1, what does VK-… how is VK-1 generated. What are, you know, what are the terms that contribute to VK-1. If you look at the generating equation, whether you look at this or this, VK-1 contains all the effects of shock waves from K-1 to minus infinity. Am I right, VK-1 contains effects of EK-1, EK-2 up to EK- infinity. You look at this equation here, in place of K you plug in K-1. Now having recognized that, what is the covariance between EK and VK-1? It is 0, b/c there is nothing in VK-1 that is correlated with EK, why, by definition of white-noise. This is by virtue of definition. So we are not doing… we are not learning anything new, we are just… keep applying the definition again and again at appropriate stages. Therefore this term is also 0. Any questions on this, if you have any question, you should get it answered at this stage itself?

(inaudible)

Obviously it's asymmetric.

(inaudible)

Expectation of EK VK+1 is not 0, because we VK+1 will contain VK, effects of VK, which in turn is driven by EK. Therefore the cross covariance, that itself is a very simple way of checking, the cross covariance is not symmetric, right done, so are we done yet, so we get an equation here sigma 0, oh I am so sorry, you should have corrected me, exactly, so you should, it is not VK-1, may be multiplying both sides with VK, you are going to correct this one. Second… the second equation we are going to, any way that argument was anyway necessary for the next lag. So now, now here you have to evaluate the cross covariance between EK and VK, by the same token, now VK contains effects of EK and all the other past EKs. By definition, this EK is only correlated with this term. So when you evaluate the expectation here, only one term will prevail, which is expectation of EK times EK, and what is that sigma square E. Therefore I can replace this with sigma square. Okay so now I have a nice equation sigma 0+, I am going to write it this way +D1 sigma 1=sigma square E, there is a reason why we write it this way, fine. What about sigma 1 now, what do I do about sigma 1, same story. It's expectation of VK times VK-1, what do I get. I multiply both sides of the generating equation with VK-1 and take expectations. What do I get as a first term, Nitya can you tell me?

(inaudible)

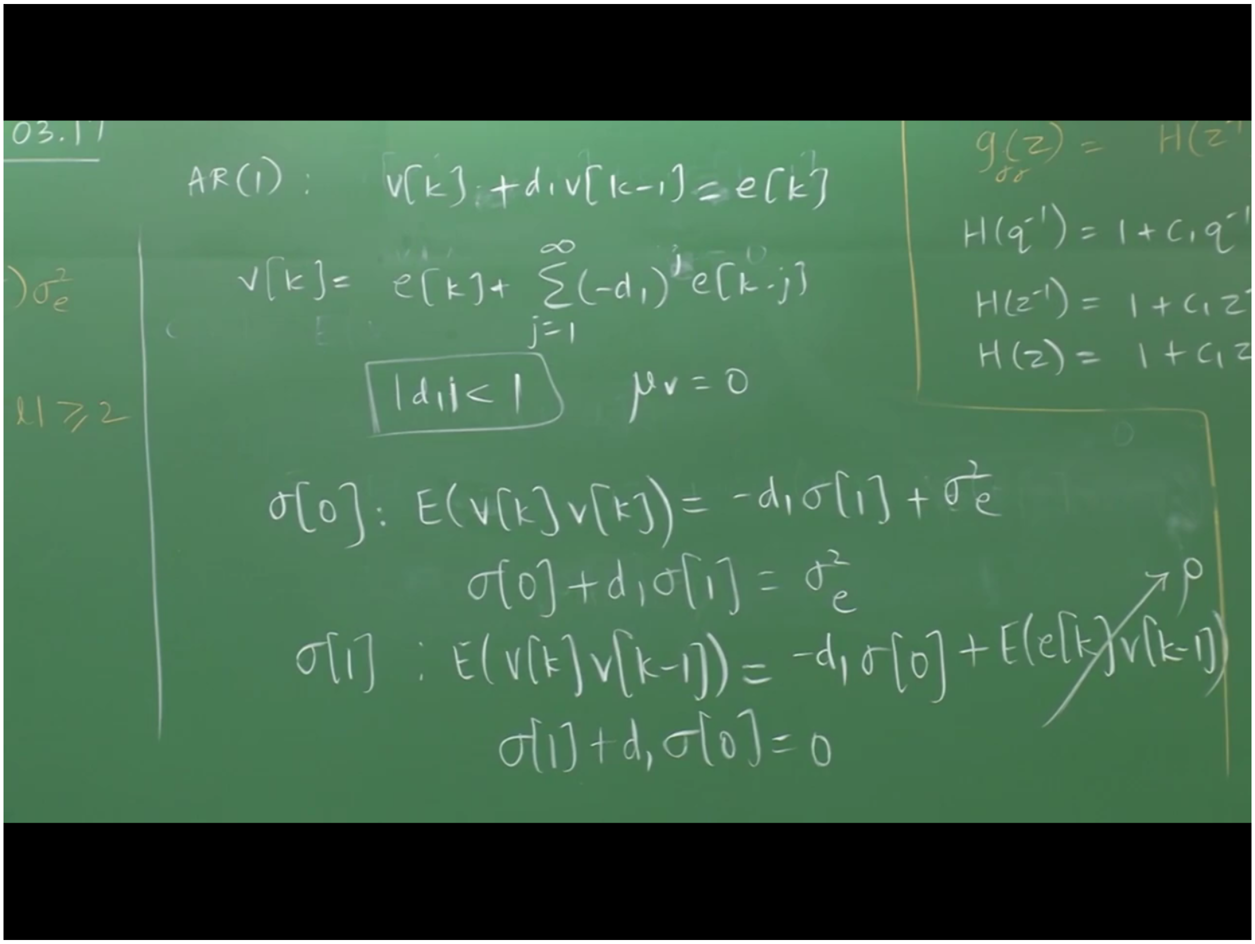
I want people have not taken time series analysis to answer. What do you get, any answer Nitya? What is the difficulty? I am going to multiply both sides of the equation there by VK-1 and take expectations.

(inaudible)

D1, ha… ha good –D1 times… expectation of VK-1 times VK-1, what is that?

(inaudible)

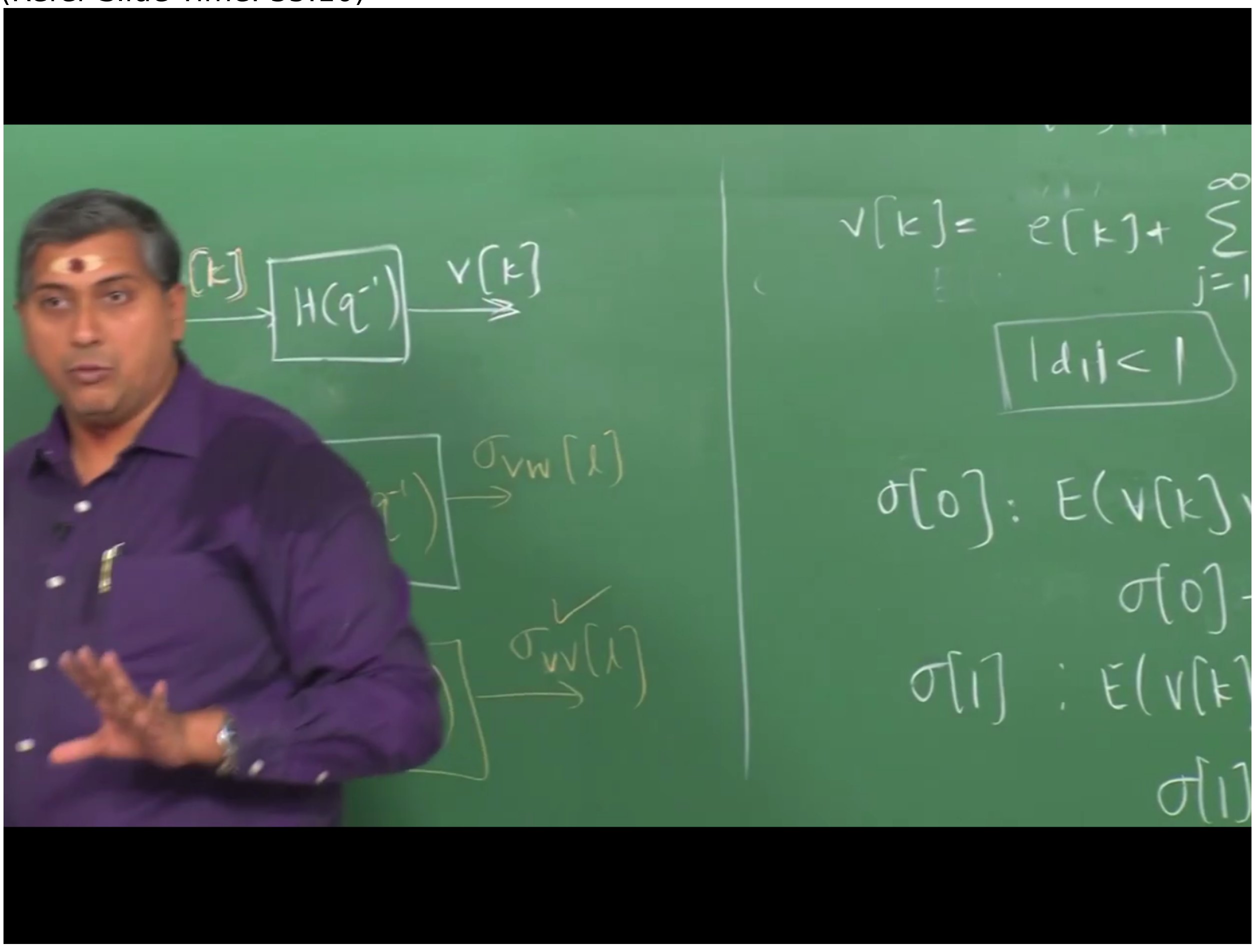
Sigma at, correct, yeah you should be very confident. Zeros are very-very helpful, okay. –D1 sigma 0. What about the second term, we had already discussed that, that's going to be 0, so I am not going to write it, well I am going to write it, but still cancel it out. Now the earlier discussion comes handy and we can say this covariance is 0. So I have a second equation here, you should not tend to think now, oh my god, am I going to solve infinite number of equations forever. Fortunately, there is a cyclicity. I could not just live with this equation alone, because I have two unknowns, I am given D1 and sigma square E, that's what we mean by, I am given a model. So we went on to write a second equation, so that I can hope to see some light and yes so we some light here, we have two equations, two unknowns. Always remember, when you are given a model, it means the model coefficients and sigma square E is given, like wise when you are asked to estimate a model, you are going to estimate the model coefficients and sigma square E, that holds. So now I have two equations, two unknowns. The nice things that you should quickly observe is this difference equation that was governing VK is the same difference equation that is governing the auto covariance. Can you see that? The homogenous part of it, the left hand side, doesn't it look identical to the difference equation that you see for VK, yes or no? This is the hallmark of an auto regressive process. The difference equation that governs VK is the same difference equation that governs the ACVF as well.
(Refer Slide Time: 31:42)



However, the forcing function is different. The forcing function for VK is EK. What about the forcing function for auto covariance. This is a very fundamental result, you should remember in theory of random processes. It doesn't matter, in fact it, whether it's AR or MA. When I have two, let us say some signal WK, when random signal driving VK, W need not be white… need not be white. Some random signal is driving a process and producing another random signal. If H is the one that connects W and V, two random stationary processes, then it is the same H, it is the same H that connects, you can say the… in fact there is another result that we shall run. One result is that, the auto covariance drives the cross covariance. Here the other way round. Here you have the cross covariance driving the auto covariance, either way is okay. You can… here I have written auto covariance driving the cross covariance. I can also say that sigma WV drives sigma VV. Why are these results important? Okay that is correct. So given the coefficients I can estimate the cross covariance or auto covariance. Of course, I should be given the inputs there, I should be given, this result particularly becomes useful, when Week is white, I know it. Here also I have used the fact that W is white. When it comes to estimation, where the series is given and I have to estimate H, I cannot use this, can I use this, I cannot, unlike in the deterministic world, I do not know the input. I am only given V and suppose to estimate H, like we discussed yesterday. I cannot use this model, I mean this kind of a relation equation to estimate H. I have to move to the auto covariance domain, because in practice this W is for at least uni-varied processes, this is EK, whose statistical properties are know and we use one of these to estimate

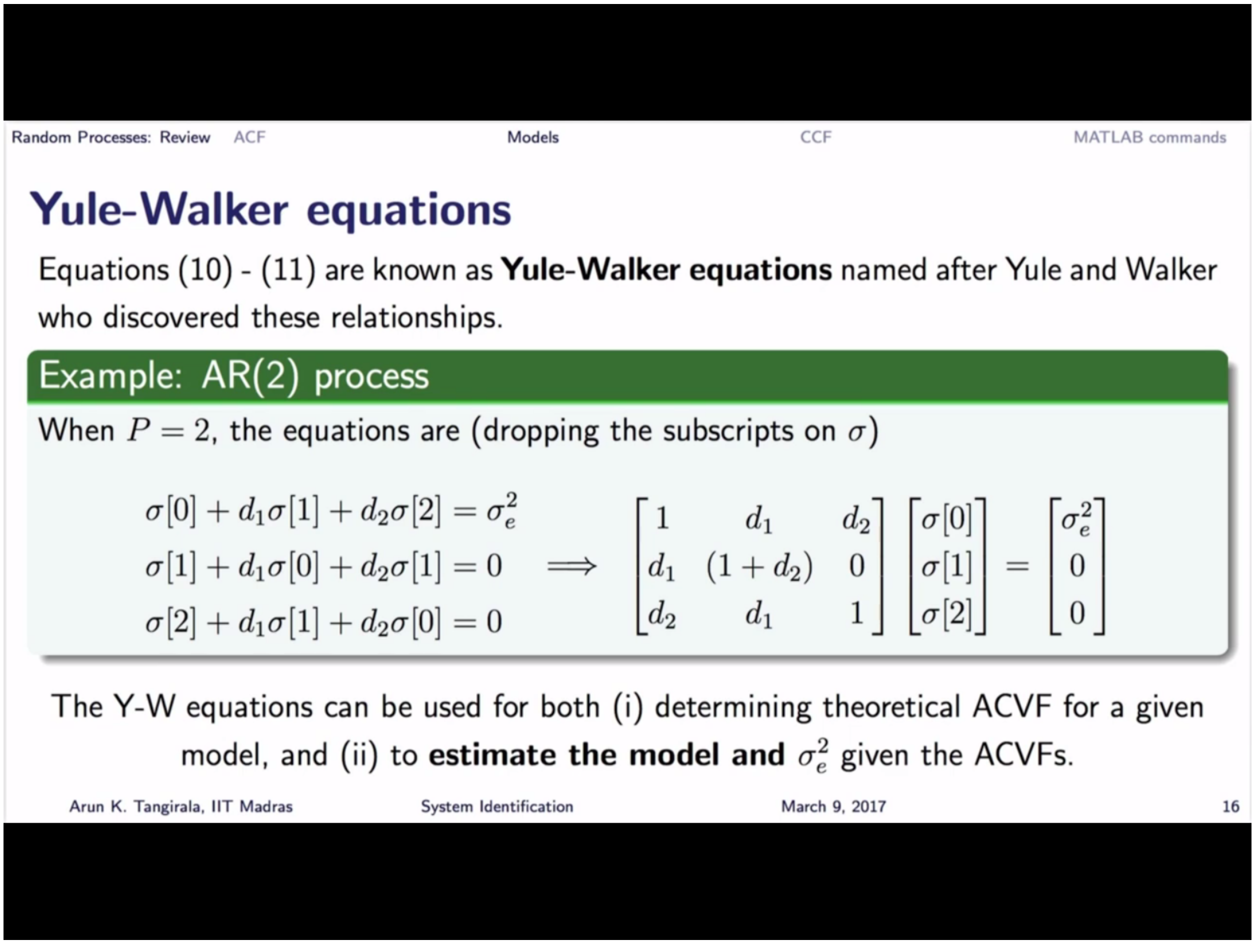
H, because I can always compute for example auto covariance of the given series. So this I can know in practice, this I know the way we have written here, because W is white and the given model, I can then estimate H. So there are many uses to this result. Anyway coming back to the discussion and for an auto regressive process, the governing equation or the generating equation for the ACVF is the same as that for the original process. These set of equations together are known as Yule Walker equations, and that is then…

(Refer Slide Time: 35:10)



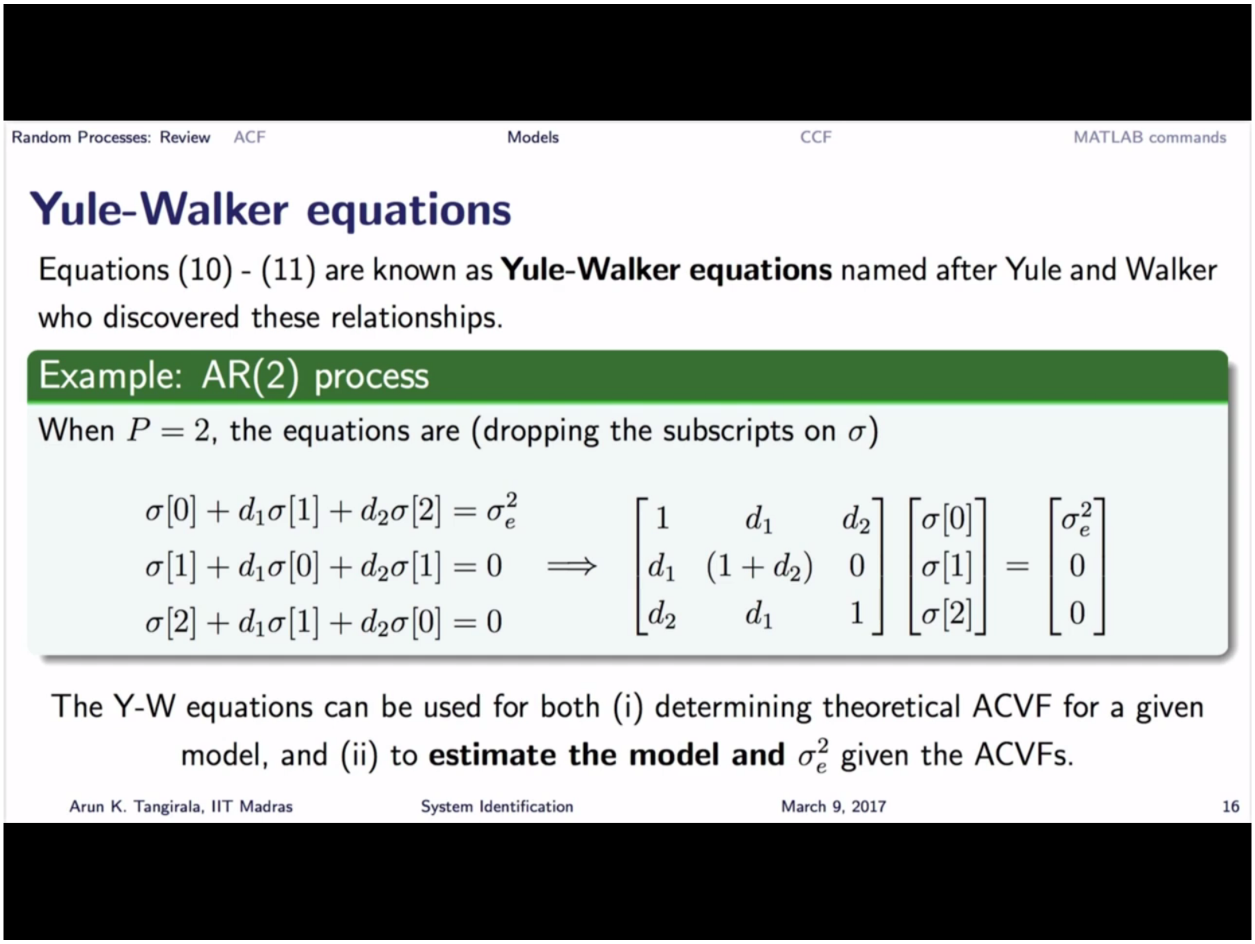
equation 10 and 11, that I have given for a general ARP process and I am just showing you an example on the screen for an AR2. I have derived AR1 on the board, for AR2 the equations would look like this. You can see, when I have an ARP process, I am going to solve P+1 equations, correct. There are two users to this Yule Walker Equation. These were discovered somewhere in the 1920s by Yule and Walker independently, and they are very powerful, they are very popular…

(Refer Slide Time: 35:50)



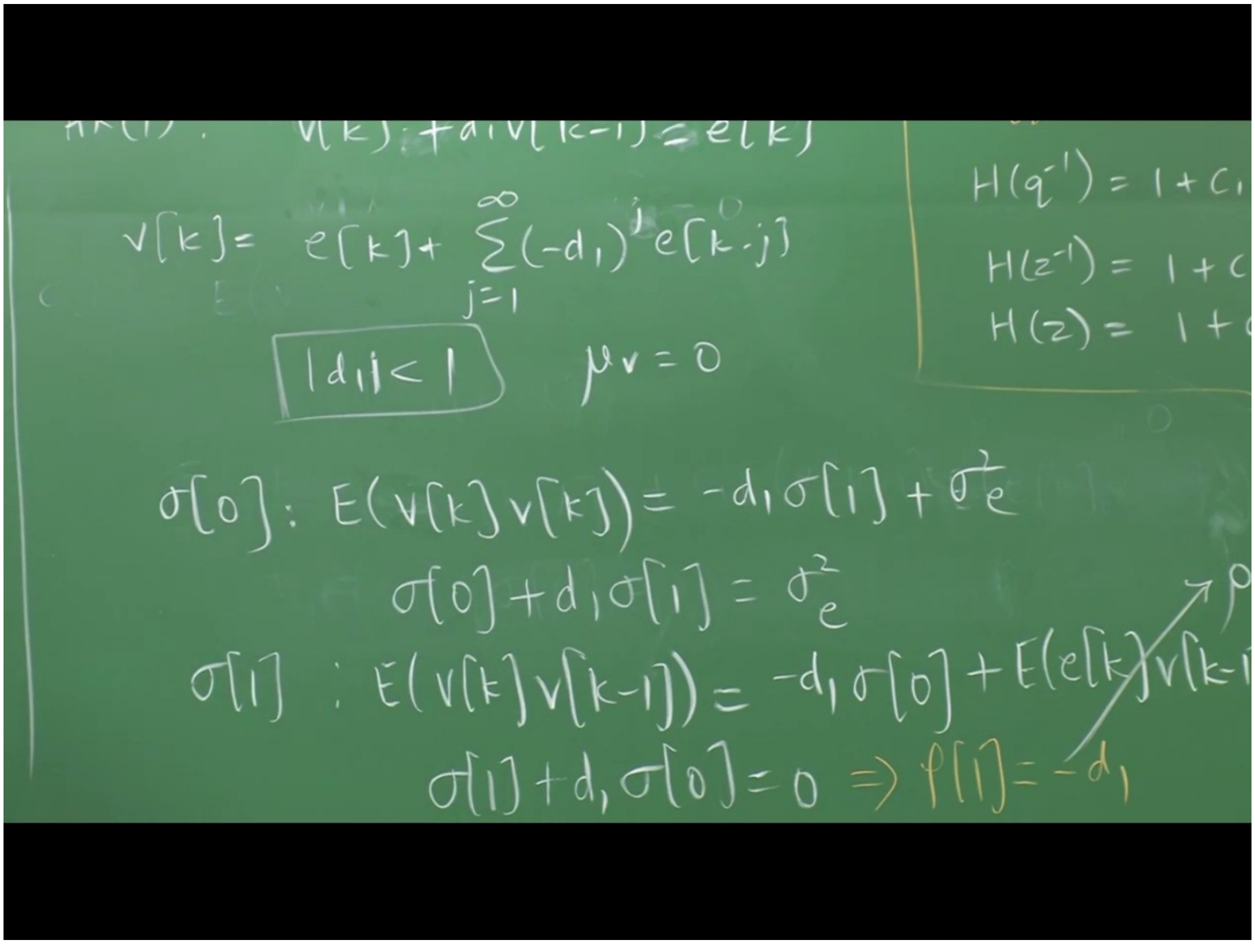
For estimating AR models, for doing theoretical analysis. So one use is, if I am given the model, I can use this equations to determine the theoretical ACVF. The question that you should ask is, these equations are for example, in this… for instance in this example here, by solving this equations, I would get sigma, sigma at 0, sigma at 1, and sigma at 2, what about sigma at 3. How will I get sigma at 3, how will I get sigma at 3, how will I get auto covariance at lag 3-4 and so on. Just recursively apply this, right. This equation is going to recur here as well, you will get sigma 2 + D1 sigma 1 = 0, sigma 3+D1 sigma 2=0 and so on. In fact for AR1 it is straight forward. You can straight away say the auto correlation at lag 1 is –D1, right. I don't even have to evaluate the first equation, if I am only interested in auto correlation. So the other ACS at other lags are generated recursively. So you can see it's a completely different scenario, sorry… when it comes to AR process. I am not being very emotional, don't worry. so when it comes to AR processes, as against an MA process, it was so simple to solve the ACVF of an MA process. Whereas here you have to solve a set of simultaneous equations, which are call the Yule Walker equations. Now consider the reality, the reality is I am given series and I am suppose to fit a model. Suppose I chose to fit an AR2 model, then I solve this Yule walker equations, where the sigmas are replaced by their estimates. This is the basis for so called method of moments. In the method of moments, what… what do you mean by moments here, auto covariances are moments. What is method of moments, it says first derive the theoretical relations between the moments and the parameters. If you look at this equations, the parameters are D1, D2, and sigma square E. Those are the unknowns and the moments are the sigmas at lag 0 1 and 2. We have set of relations between them. Method of moment says assume that these relations hold good for estimates as well, which is not necessarily true, why should estimates satisfies this, because estimates are constructed from time averages and the two finite observations, even if you assume ergodicity, that doesn't apply… applies here, but I have only finite samples. So method of moments forces these equations to be fulfilled even for estimates and that's how the Yule Walker equation or the Yule Walker method for estimating AR models is born. It belongs to this class of method of moments. Alright, so remember this Yule Walker equations, how to setup this, it just comes by practicing, but once you go back and derive this for AR2, you should understand how things work out for AR3 and so on. Generally by hand we do not go beyond AR2, after that we… we use a computer or some kind of a computational tool. Now this Yule Walker equations can be solved recursively and that is what is the Durbin Levinson's algorithm doing for you. If I have an AR1, how many equations I am… am I going to solve? Two equations. If I have AR2, I am going to solve three. The other thing that you should quickly observe is, if I am given the auto covariances, I can first ignore the top equation and only work out the bottom tow equations to get my parameters… model parameter, then I go back and plug in and use that top expression to get an estimate of sigma square E, you understand, you look at the two… three equations here, although sigma square is unknown, it only appears in one equation, which is good news, which means, I can solve for the coefficients first and then use them to estimate sigma square E. That is how it is done in Yule Walker equations. So if it is AR2, sin fact I am going to first solve two equations, get my D1 and D2 estimates, and then plug them in to get an estimate of sigma square, clear.

(Refer Slide Time: 40:05)



Likewise AR1 here, if it is an AR1, you just say that the optimal estimate of at least in a Yule Walker sense or the method of moment sense is nothing, but the correlation itself, of course, negative of the auto correlation. And then you can use this expression estimate to find out... to get an estimate of sigma square E, clear.

(Refer Slide Time: 40:26)



$H(q^{-1}) = 1 + c_1$
$H(z^{-1}) = 1 + c$
$H(z) = 1 + c$
$v[k] = e[k] + \sum_{j=1}^{\infty} (-d_1)^j e[k-j]$
$|d_1| < 1$
$\mu_v = 0$
$\sigma[0]: E(v[k]v[k]) = -d_1\sigma[1] + \sigma_e^2$
$\sigma[0] + d_1\sigma[1] = \sigma_e^2$
$\sigma[1] : E(v[k]v[k-1]) = -d_1\sigma[0] + E(e[k]v[k-1$
$\sigma[1] + d_1\sigma[0] = 0 \Rightarrow \rho[1] = -d_1$