Foundations of Cyber Physical Systems

Prof. Soumyajit Dey

Department of Computer Science and Engineering

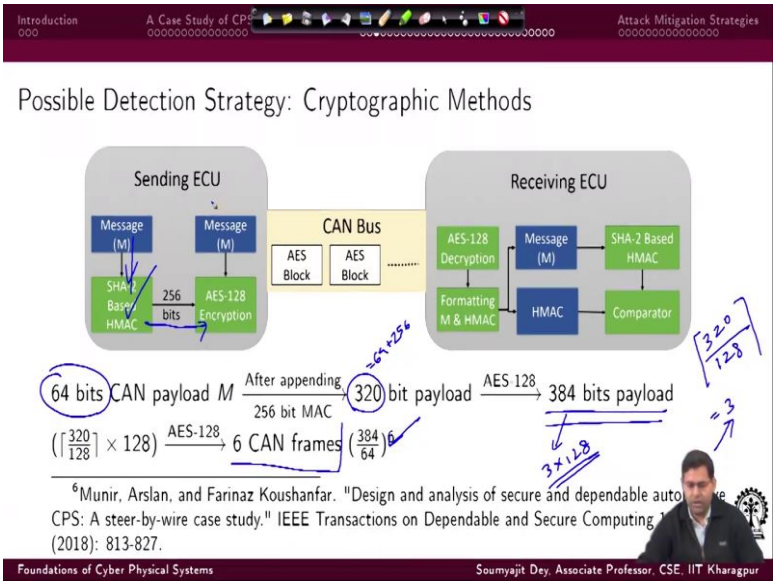
Indian Institute of Technology – Kharagpur

Lecture – 55

Attack Detection and Mitigation in CPS (Continued)

Hello and welcome back to this lecture series on Foundations of Cyber Physical System. So, we have been talking about several attack detection techniques for for a CPS implementation.

(Refer Slide Time: 00:41)



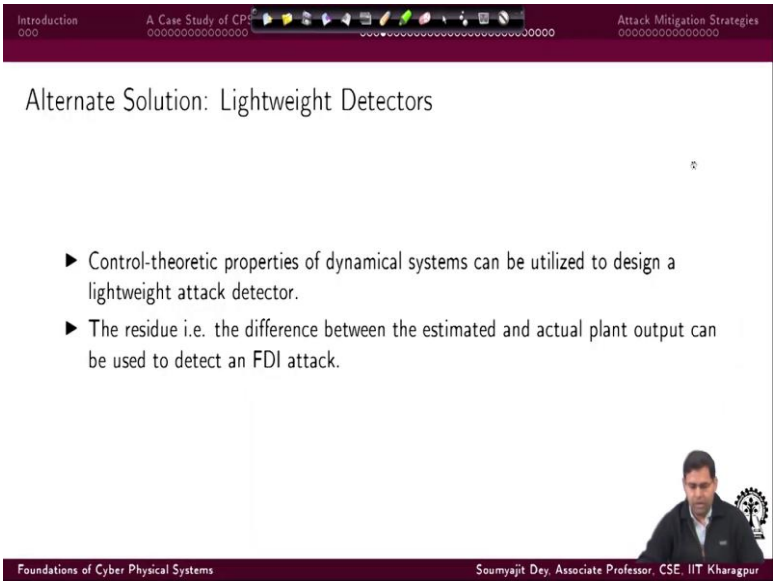
So, let us carry on ah from that point. So, in the previous lecture, ah we we had covered ah this cryptographic method. And we were trying to show that well ah why really you have ah CAN bus overload if you are trying to encrypt packets and followed followed by this Ah message authentication code computation. So, if you just look into this process, so, you have a 64 bit, CAN payload, let us say and when you are trying to send it, you see in this pipeline, you first have a message authentication code which gets appended so that is 64 plus 256. right So, this should give you 320. Now, ah what we are saying is? In the next phase of the pipeline ah you have ah this AES engine. So, typically the way, ah the AES engine is going to work is, ah it it is it is trying to encrypt some data. So, it will do the encryption in blocks of 128. right. So, if you are doing this in blocks of 128.

So, eventually we are saying that that would generate a payload of size 384. Now, let us understand why that happens. So, this is 320 and you are going to do it in blocks of 128. So, you take ceiling of that and that should give you 3. So, the of course the last block will be a bit smaller. right And when you do that. So, you have got these three blocks and each of the blocks have a size of 128. Right

So that is why it is 3 times 128. So, this is the total number of bits which are contained in these three packets. right So, eventually that is why we are saying that well when you start with 64 bits, you will end up with an increased load of this 384. right So, and and that that is essentially I mean I mean when you are trying to transmit it here. This will actually result in 6 CAN frames. So, this has been taken from this reference. Ok

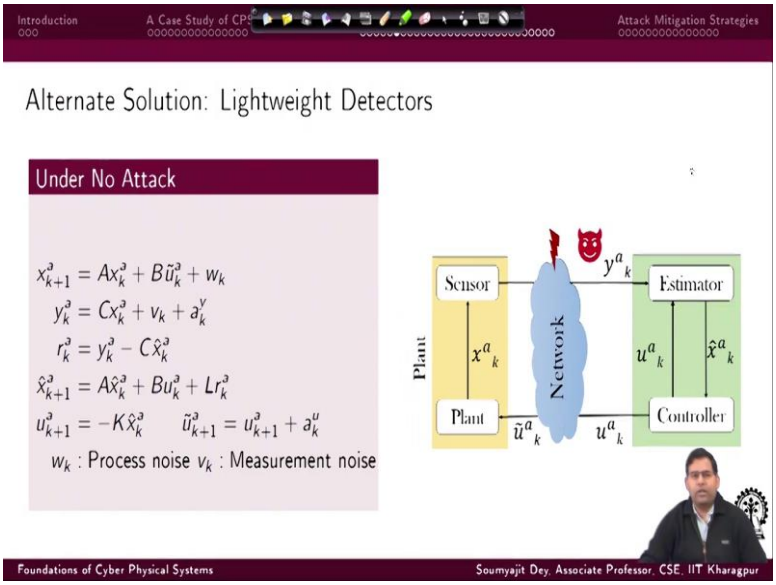
I mean do not think this is a power or something here. So, this calculation and example has been taken from this reference here. So, fine this was about our discussion on this, this cryptographic method.

(Refer Slide Time: 03:18)



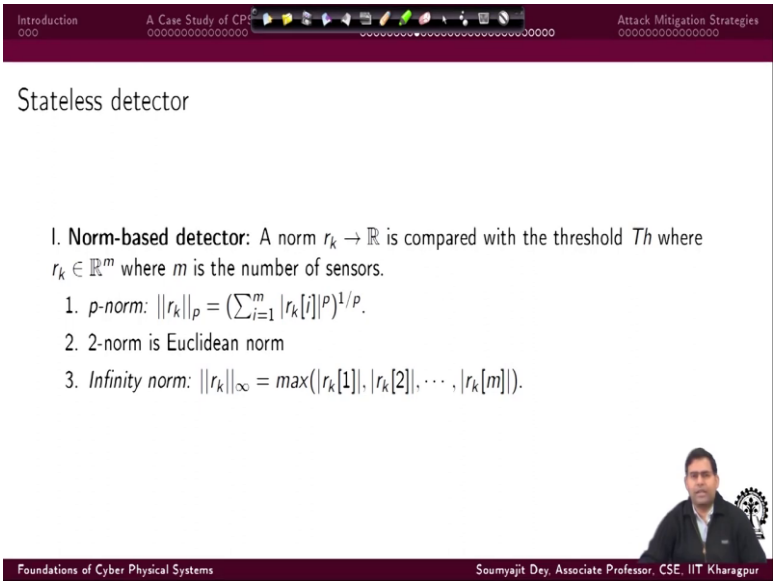
And if you recall after this, we went further into our discussions on lightweight detectors.

(Refer Slide Time: 03:23)



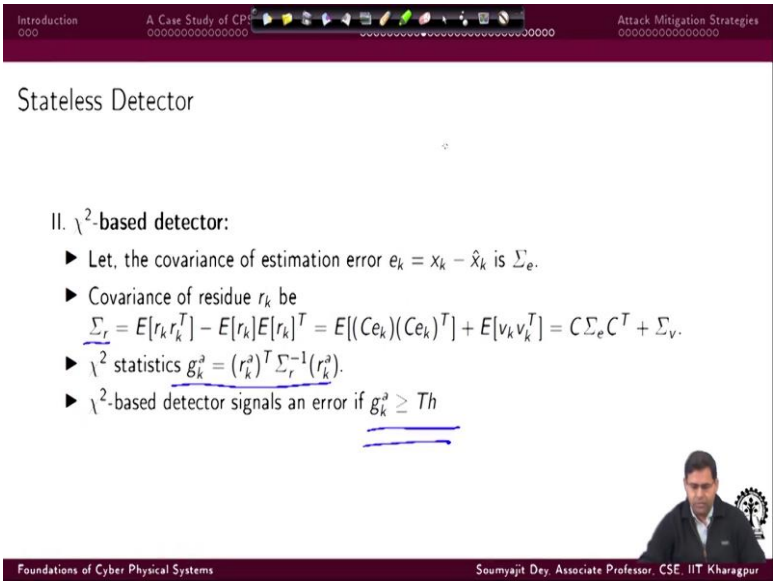
So, with respect to lightweight detectors, we had been talking about this residue-base detector.

(Refer Slide Time: 03:30)



And we talked about well they can be stateless they can be a state full director I mean detector share like that. right And different kind of norms like the Euclidean norm, any general p norm or the infinity norm can be used for computing, a scalar value out of this vectors here. Right

(Refer Slide Time: 03:46)



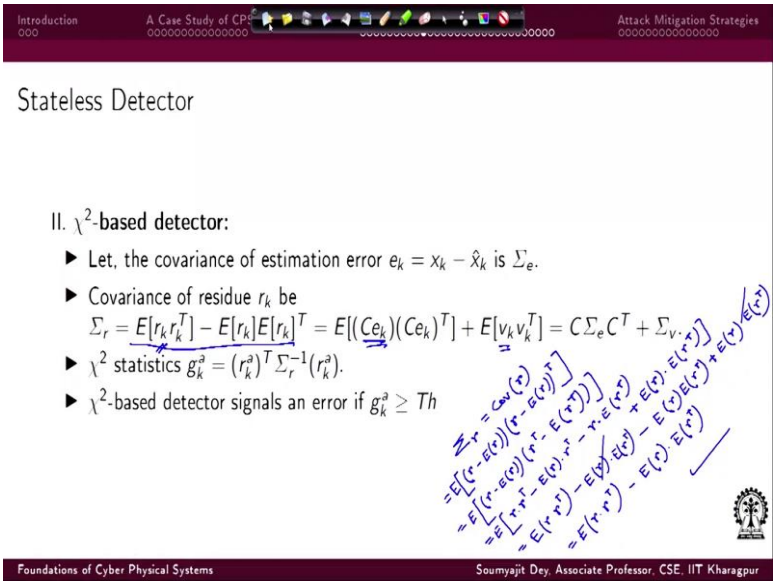
And here we had a direct calculation where we are saying that well there is this example of stateless detector, for example the chi square detector. And for which ah what it does is ah it calculates an estimation error. right And the system would try to figure out that well ah some specific chi square statistics which is this g_k. right ah this stat This chi square statistics if it is beyond the threshold or not.

So that is that has been the working principle of this ah chi square threshold Right ah chi square threshold base detector. Now, what essentially we are doing here is like what is this g_k? So, if you see ah this essentially is ah considering, ah the covariance of the residue. ok So, let us understand what is this covariance of the residue. ah For first of all, let us understand what is C here. So, if you recall from our previous examples. So, we related the residue ah value with ah this y. So ah yeah

So, if I consider about r_k right ah it is like C times the error over k Ah in the over the kth kth instance where the estimation error is given this e_k in that in in that instance, is given by that actual value of the state minus the ah minus the estimate that is x_k hat. right So, this estimation error e_k ah let the covariance of this error ah be given like this. ah By this this value ah Sigma sigma e. right So, let us say this is the covariance of this estimation error. Right

So, the chi square statistic actually that we are interested in ah would be this ah your covariance of the residue. Right So, the covariance of the residue ah let us see how we can really measure it here.

(Refer Slide Time: 05:56)



So, for for simplicity, this is for the kth instance right. So that is why r_k ah, let us just write r for any instance in general. So, this is ah by definition the covariance of the residue in the kth instance. So, we will try to see that well how we can write it directly as this expression. Because if you can write this expression then it becomes simple. Right Because we will just write r_k is C of e_k ah as you can see. Right

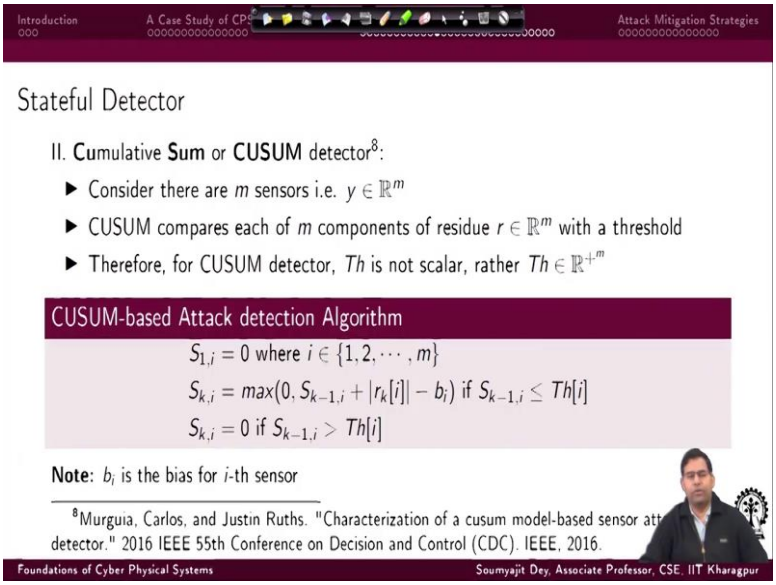
Because ah yeah and then when you talk about this expectation here, ah it is let us, if you just go back. ok First, let us do this derivation and then we can go back to the previous things here. So, sigma r is covariance of r so, if you just apply the covariance formula. Now, if you you just bring in the expectation here, this again expectation over two expectations. So that that it is an invariant, so, it will remain whatever it is.

So, you get something like this. So then you can just strike up any two of this, so, you get. yeah So that is what we have here. So, ah once you are able to compute this sigma r here. ah Then you can just apply this chi square statistics formula ah which uses this sigma r inverse here. Ah So that is basically doing a scalar computation, ah where it is using the residue under attack and it is transpose.

So, this is the r_k^a that is residue under attack in the kth instant. ah It is transpose and the original value here and they are being weighted by sigma r inverse. So, this is just chi standard, chi square statistics formula and you are applying that here to check that whether it will be greater than the threshold or not. So, you can just use the constant threshold base detector which just compares directly the value of the residue with respect to the threshold.

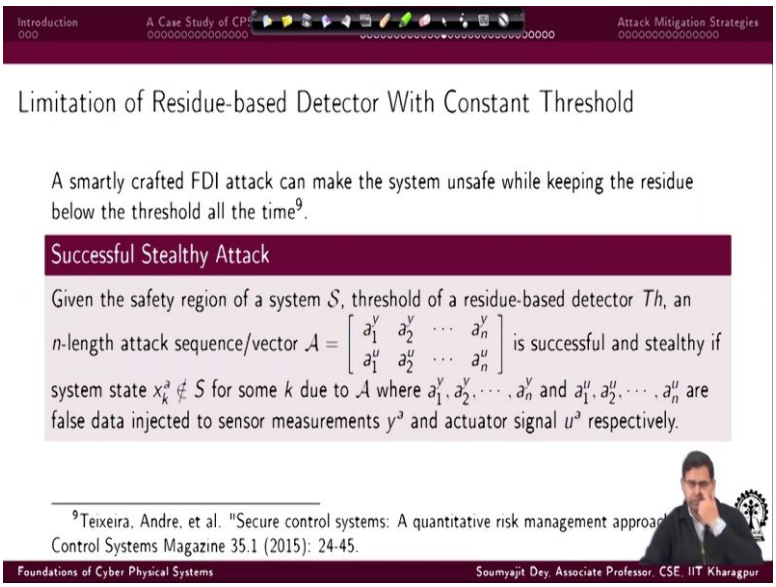
A standard norm base detector or you can use this kind of chi square statistic space detector which is this one. Fine. ok ah

(Refer Slide Time: 10:18)



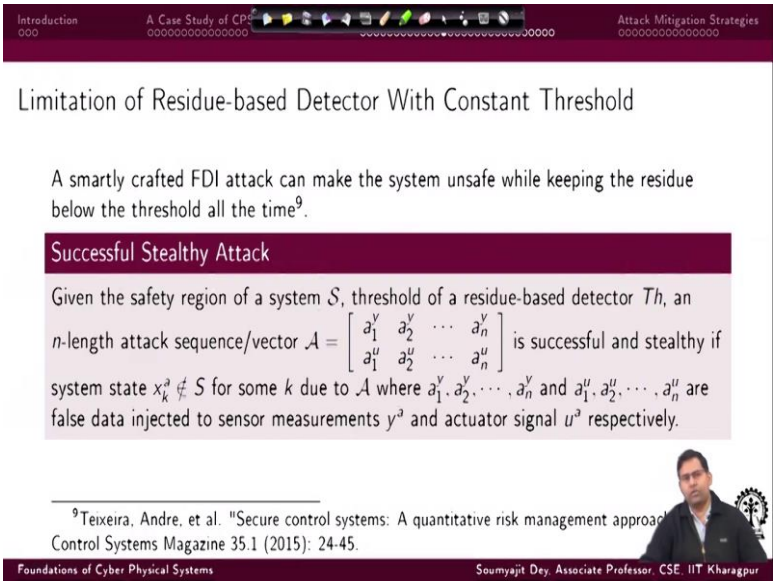
So, maybe we can progress further from here because we also discuss the state full detector.

(Refer Slide Time: 10:23)



And what are the limitations that are going to be there if we have this residue base detectors with Contin constant thresholds. right ah

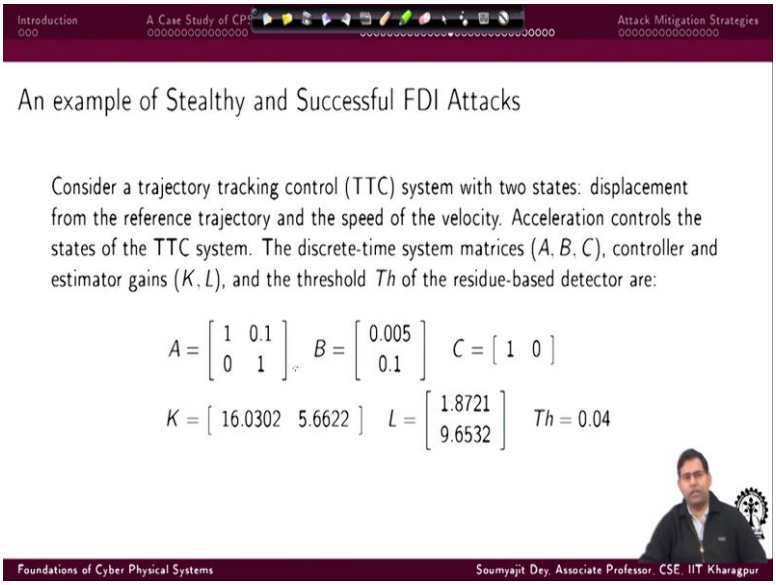
(Refer Slide Time: 10:35)



Because we said that well that there can be a successful stealthy attack because somebody can stay below the attacker. If the attacker has ah knowledge about the systems model, the threshold threshold value that has been set for the detector. Then he can choose to inject the system ah with so, small amount of attacks in both the measurement values and the control values in such a nice way that every time you calculate the residue. Ah

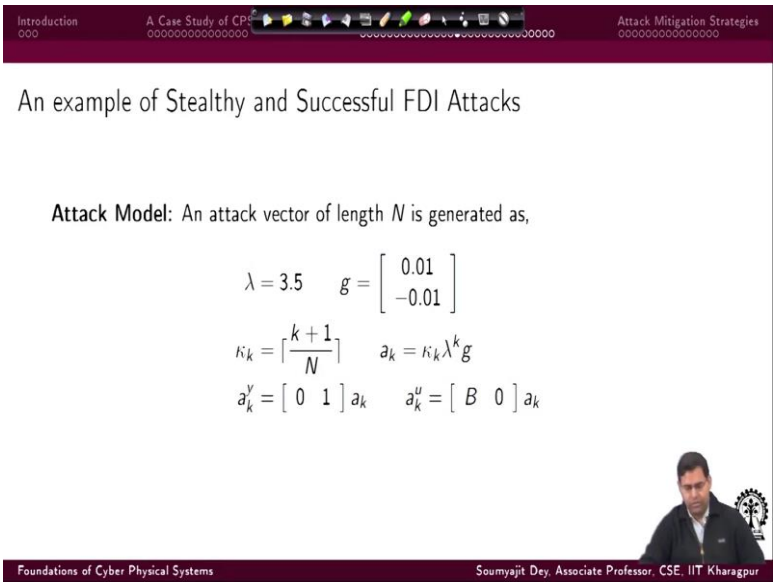
So, every time you will calculate the residue ah here. ah it will be It will remain below the threshold. ah But ah eventually the systems trajectory will we will get out of some safety boundary. And the thing is, you are unable to actually measure it also because the measurements you are getting of the system. ah Those measurements are also being pastured by the a y component of the attack. And due to this, and due to that measurement perturbation, you have an issue there. Right And do to that measurement perturbation you are unable to kind of figure out that well the system is really going out of the boundary. So that can be a situation.

(Refer Slide Time: 11:52)



So, we showed that here with an example like right. ah that let us take this trajectory tracking control system.

(Refer Slide Time: 11:55)

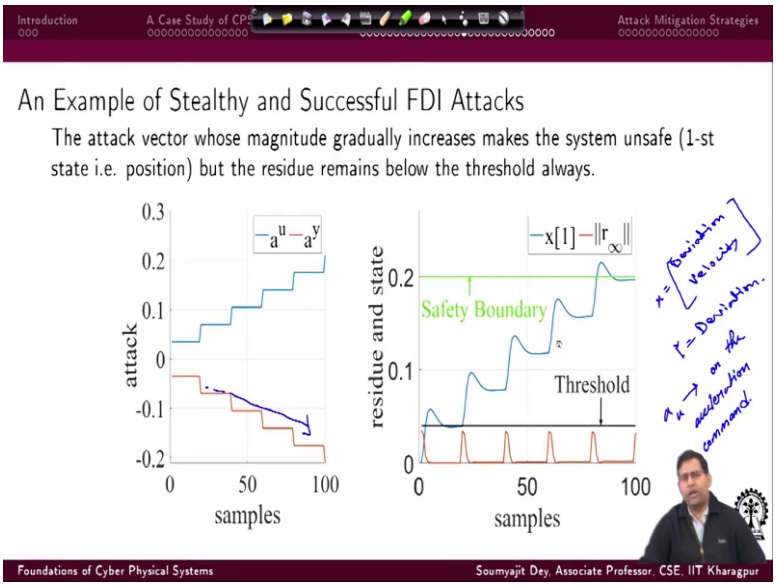


And for this system we considered this an attack generation method that is big followed. So, what is happening is, on the measurements and attack Ah of this value is being injected in the kth instant. OK And there are some generator values which have been said that well in every kth instant this Ah this value ah this scalar value ah is given by k plus 1 by n. right And then you have a weighted value of lambda at every kth instance ah the magnitude of these value is increasing.

And g is a constant value. ok So, you are ensuring that this overall attack value which is slowly increasing ah or with every kth instant. Because lambda is increasing in exponentially and k is also increasing in a linear manner. So, ah over an N length of N may be a significantly large value. ah you are, you are injecting different attack values and the way you will like to do this kind of attack attack is because if we have this kind of a generator ah enclosed in an algebraic expression, then it is very easy to implement in a computer that well these are the attack values which will go and get injected to that variable very fast continuously. right Because the attack values have to go in sync, with the periods of the system, the sampling sampling period of a system. Because you want the attack values to part of the measurements here. So, what you are doing is for this system you have ah your states that is displacement from the reference trajectory and the speed of the velocity and the speed of the vehicle.

This will not be velocity, of course. And whatever I mean you are also you are you also you are also measuring this deviation from the trajectory. So, x is essentially the deviation and the velocity and your y is basically one of the x components itself which is the deviation component. Ok So that is why, when you are applying the attack, it is basically, you are choosing one of the components because you are trying to attack one quantity here. Ah

(Refer Slide Time: 14:01)

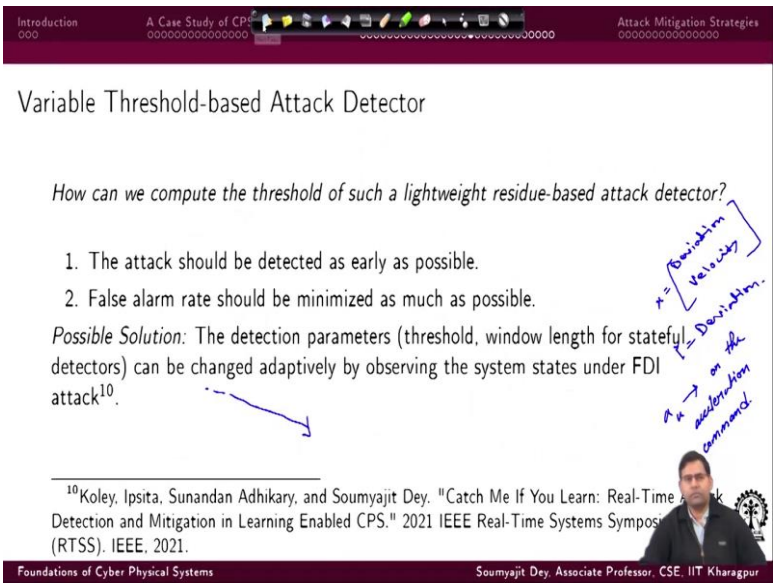


So and based on that this is our plot of what values of attacks we are really doing ah on y and what values of what attack we are really doing on u. So, you can see that this is my ah a k right and based on that I am generating a sequence of values of attack for the x variable. ah sorry the sorry, the y measurement variable which is basically the first state in x. right ah And also we are generating an attack on the ah on the acceleration.

And accordingly, you have a u and a y. So, if I just write down the states here. So, x is deviation from the trajectory intended trajectory. And this is velocity and the measurement y that I am taking is the deviation itself. And that is what I am perturbing. right ah So that is my perturbation here. And I am doing the attack a u on the control input which is ah which this setup on the control input is basically happening on the acceleration command.

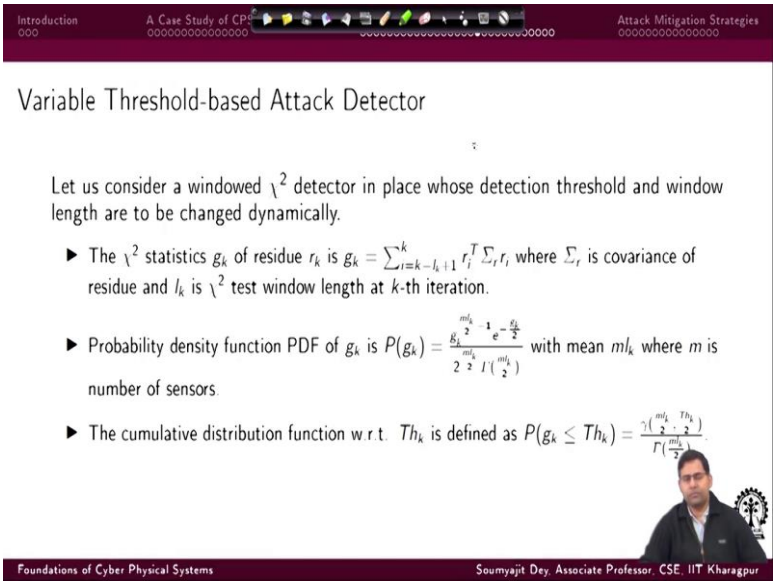
So, based on that as you can see, ah you you have this systems deviation slowly. This is the actual deviations value not the attack value. And that value is slowly wearing out of the safety boundary.

(Refer Slide Time: 15:57)



So, the next thing we talked about is well is it possible to have a variable threshold-based detector and we discussed about an RL based strategy.

(Refer Slide Time: 16:11)



We will talk about that again in the next lecture. Thank you for your attention.