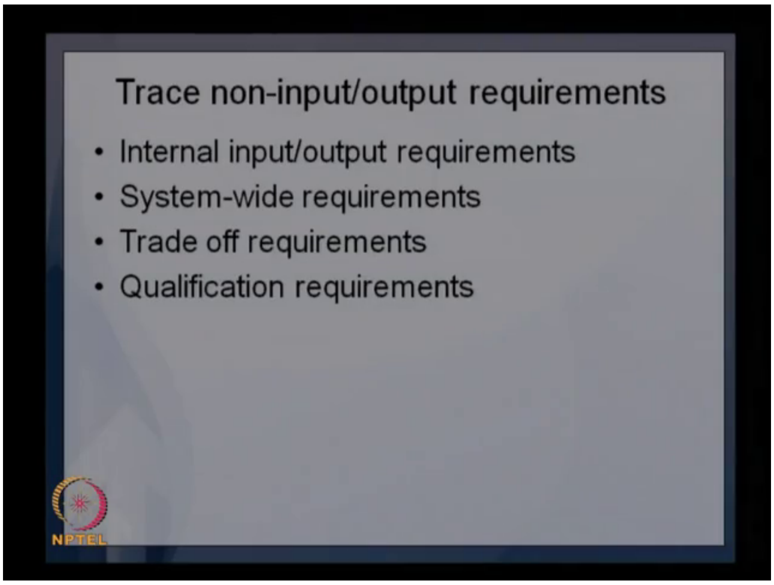
Principles of Engineering System Design
Dr. T. Asokan
Department of Engineering Design
Indian Institute of Technology, Madras

Lecture - 15
Operational Architecture Development Part II

Dear friends a very good morning to all of you and welcome to another session on System Engineering System Design. In the last class we discussed about the operational architecture basically looking at the allocation of a function to components and how do we actually do the allocation architecture, what are the principle of allocation architecture or how do we allocate, what are the basic principle to be used. And then we briefly mentioned that this allocation architecture helps us to identify the tradeoff decisions as well as look at the input and output requirements, which are coming within the system.

In this class we will look at allocation architecture in bit more detail and try to identify how do we get the derived requirements input output requirements coming because of the allocation and how do we do the tradeoff analysis and how do we get tradeoff decisions based on the allocation architecture.

(Refer Slide Time: 01:12)



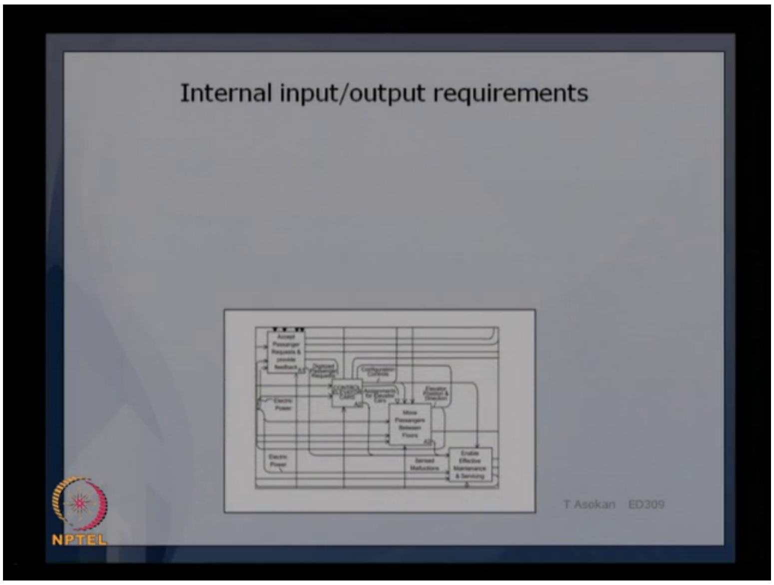
So, to going to the details in the last class we mention that 1 of the major task in allocation architecture is basically to look at the non input output requirements. That is the input output requirements; we know that the from looking at the system, as a hold we have some inputs we have some outputs and based on that we can identify the input output requirements.

But, when we develop the allocation architecture, when we do it from the physical architecture convert that into from the using functional architecture and physical architecture, when we convert that into allocation architecture there will be some other requirements cropping up because of this allocation; and they are basically the internal input output requirements.

Basically we look at what are the inputs and the outputs internal to the system and we need to allocate the components to for these requirements the other one is the system wide requirements there are system wide requirements identified at the design level itself, but then how we actually allocate these requirement system wide requirements of a cost reliability, a availability of the system. So, these things need to be allocated to the component. So, how do we actually do the allocation of system wide requirements or identifying the system, wide requirements, coming out of the allocation architecture and then how do we allocate them. The other one is how do we get the tradeoff requirements and then how do we get the qualification requirements of the system.

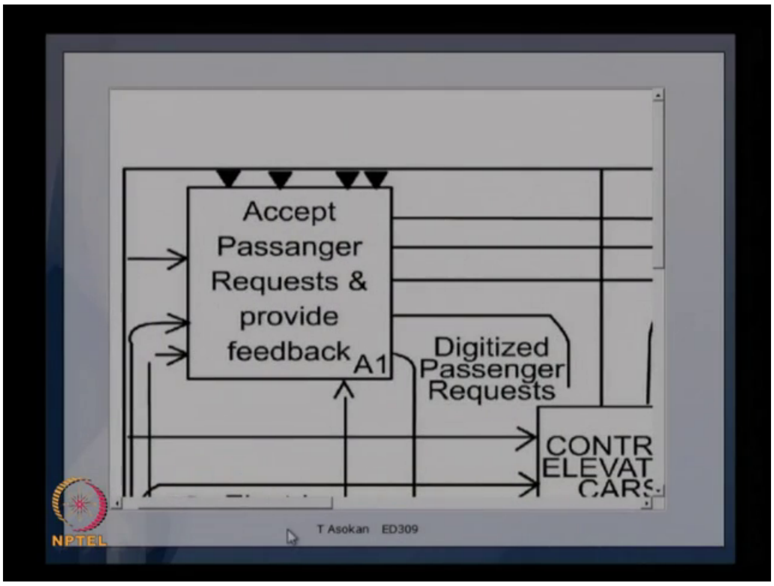
So, all these things need to be analyzed using the allocation architecture. So, we go 1 by 1 we look at the internal input output requirements.

(Refer Slide Time: 02:48)



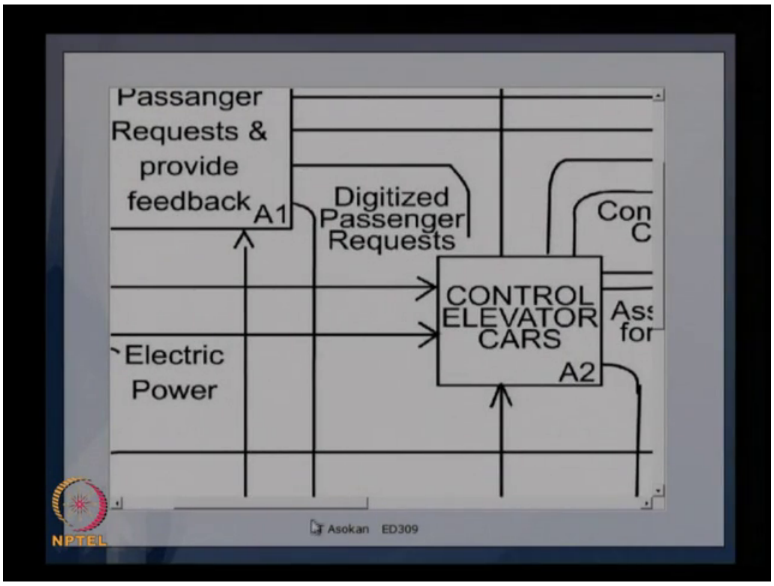
And how do we actually trace these requirements and then get the allocation architecture. So, the internal input output requirements.

(Refer Slide Time: 02:54)



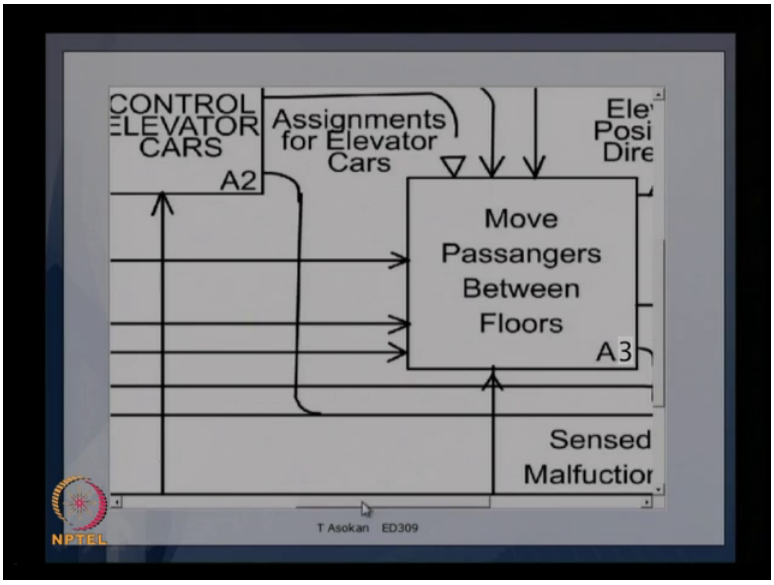
Basically we would look at the example given for the elevator system we can see that when you allocate the component or the system you can see these are the main function A 1, A 2, A 3, and A 4 in terms of IDEF0 diagram.

(Refer Slide Time: 03:11)



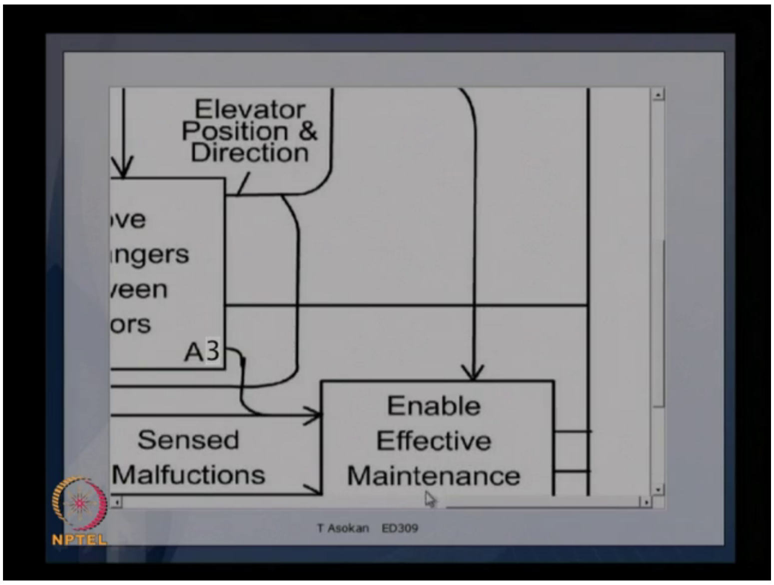
So, this is A1 this A 2.

(Refer Slide Time: 03:13)



This is A 3 and there you have a 4 function.

(Refer Slide Time: 03:15)

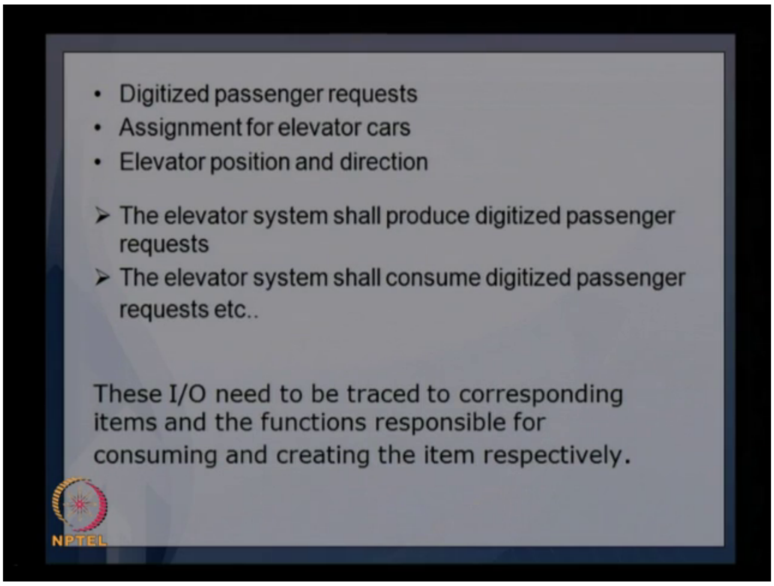


So, if from these function there will be some other requirements coming up, that is the internal input output requirements we look at these you can see digitized passenger requests is 1 of the output from this function, which is going as an input to the control elevator cars. So, this is an internal output from this function as I said internal input to these function and this is the other one assignment of elevator cars is another 1 and then you have another requirement of elevator position and direction. So, this is actually coming as an output from the move passenger between floor function and similarly you will have other one more requirement like the sensed malfunction and for kind malfunction are that need to be sensed within the system.

So, if you like this you have few input and output requirement which is internal to the system. So, for every system whenever you develop and allocate the architecture for these components, we need to find out which is the components which actually satisfy this requirement these input output requirement or to which components these requirement need to be traced. So, 1 thing is basically do trace the requirement and then other one is to allocate these input output requirements.

So, here we can see there are 4 input output requirements.

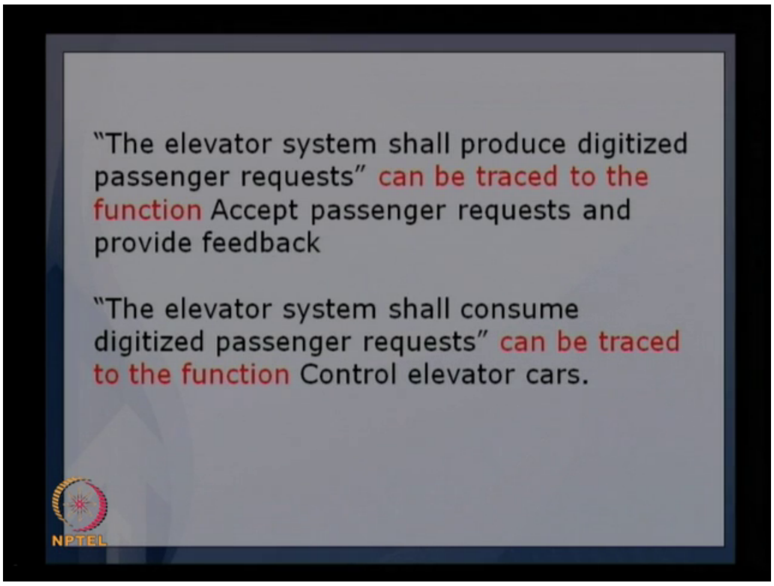
(Refer Slide Time: 04:34)



Like fairly here it is digitized passenger requests assignment for elevator cars and elevator position and direction. So, these are the requirements identified and the requirement is the function which actually needs to be provided the requirement can be written as, the elevator system shall produce digitized passenger requests or the elevator system shall consume digitized passenger requests. So, these are the requirements which can be written using this function decomposition and these are the internal input output requirements, which cannot be identified from the system level. This can be identified only when we have a functional decomposition then try to allocate these requirements.

These input output need to be traced to corresponding items and the function responsible for consuming and creating the item respectively, it is not only identifying the requirements we need to find out to which component or to which system these requirements can be traced. So, that is the tracing of the requirements. So, whatever the requirements we identify. So, this is just a sample only these requirements there will be many such requirements input output requirements, we need to identify the components or we need to trace these requirements to the corresponding items and the functions or responsible for consuming and creating this kind of input and outputs.

(Refer Slide Time: 05:56)

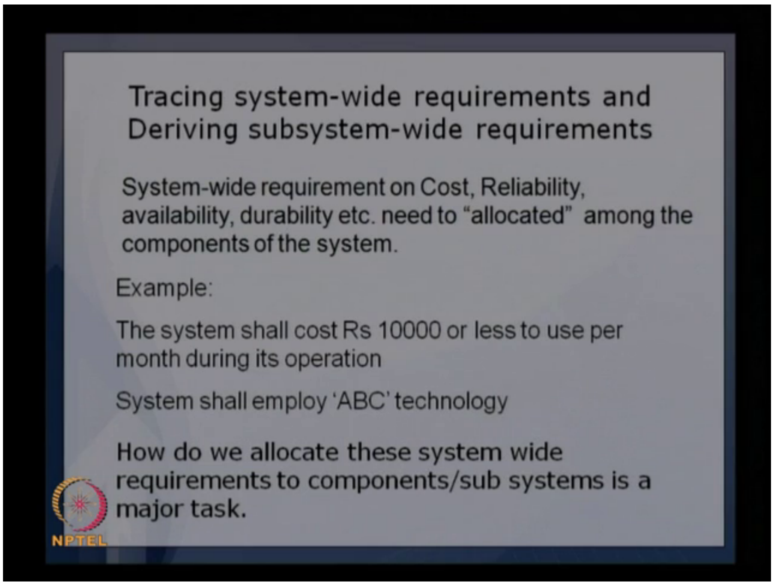


So, here you can see that the elevator system shall produce digitized passenger requests that is 1 of the requirements, this can be traced to the function accept passenger request and provide feedback. So, this is the traceability of the requirements, similarly the elevator system shall consume digitized passenger requests which can be traced to the function control elevator cars.

Similarly we can actually trace other requirements also to the corresponding functions. So, not only the functions what we need to do is to look at now which function or which component is providing this function. Then the requirement can be traced to that particular component also. Therefore, we can have a traceability of the input output requirements which is internal to the system, as well as we can trace these requirements to the corresponding functions and components.

So, the allocation architecture basically we will look into the details of input output requirements coming from within the system and then how do we trace these requirements to the corresponding functions and components. So, these are the just 2 examples of how do we write down the requirements, input output requirement and then trace this requirement to the corresponding functions and components.

(Refer Slide Time: 07:12)



Other 1 is tracing the system wide requirements and deriving sub system wide requirements. So, the first 1 what we discussed was the input output requirements, the next version is basically the system wide requirements, as I told you the system id requirements can be a cost it can be a reliability of the system or it can be availability or durability, which is identified at the stage of all are system design then we need to actually trace these requirements, as well as allocate these requirements to components basically when we said cost cannot be given to yes particular component. So, this has to be allocated to sub system. So, how do we actually trace the requirement of this particular system wide requirement and then allocate them among the components or if you take the reliability of the system depends on the availability of different components of the system.

So, how do we actually do an allocation of this reliability amongst these components and then ensure that the total reliability of the system is maintained. So, the task of allocation of reliability is basically to look at the various components, which actually contribute to the particular requirement. And then allocate the values accordingly to the components, then the allocation architecture by looking at the architecture we should be able to allocate these values and then trace this value to the corresponding functions and components.

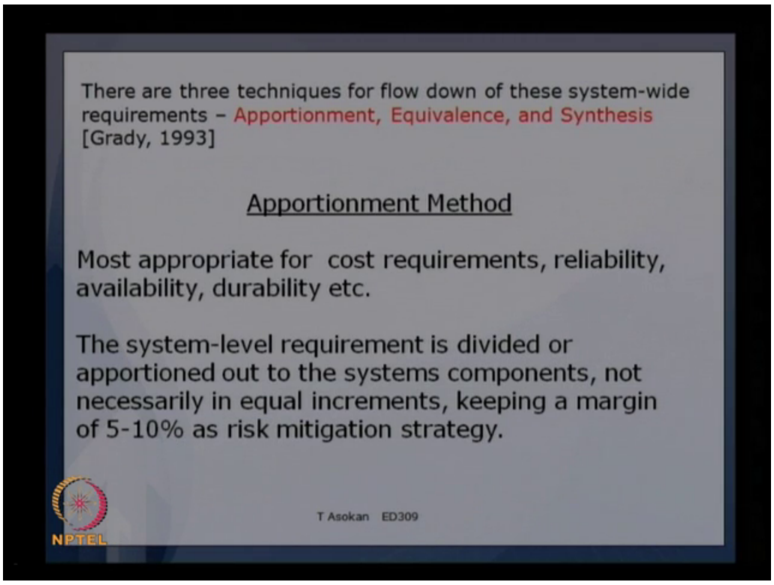
So, that is the system wide requirement on cost reliability, availability, durability etcetera need to be allocated among the components of the system and how do we do this is an example whether the system shall cause rupees 10000 or less to use per month during it is operation or the system shall employ ABC technology. So, these are the typical system wide requirement, like this you can have the system should have a reliability of 0.96 with a design goal of 0.99 that can be another system wide requirement.

So, this is the system wide requirement, when you say the system shall cost 10000 or less to use per month during it is operation, we need to find out what are the things actually will contribute to the cost that is operating cost and then how do we allocate this operating cost among the components. For example, if you take the elevator system itself the operating cost, may include the power charges, it may include the control cost, it may include the input output requirements.

So, all those will be contributing to the total operating cost and this we need to allocate amongst the other component. So, that is the requirement over here similarly the system shall use a particular technology. So, this actually we need to see if there is a particular technology, then how do we actually identify the component is actually require the same technology to be used or a technology which is compatible to the suggested technology. So, that way we need to allocate this system wide requirement.

There are different ways of doing this. So, the question here is how do we allocate the systems wide requirements to component or subsystems, that is a one of the important tasks in allocation architecture, we need to allocate these system wide requirements to corresponding components.

(Refer Slide Time: 10:20)



So, how do we do this? There are a different ways of doing this, one is known as apportionment methods, the other one is known as equivalence method the third one is the a synthesis methods. So, these are all by Grady to propose in 1993.

So, we can use any one of these methods to allocate the system wide requirements to it is components. If you look at the apportionment methods, it is basically providing the actual system wide requirements in a proportional manner to different components. So, the most appropriate for this kind of methods is the cost requirements, reliability, availability, durability etcetera. So, this can actually be apportioned among the components. If there are 5 components which actually contribute a particular requirement, then we find out what should be the contribution from each component and accordingly we divide the requirements, the system wide requirement and then distribute them amongst the member. So, that is allocate the requirement to the members who actually contribute to that particular requirement.

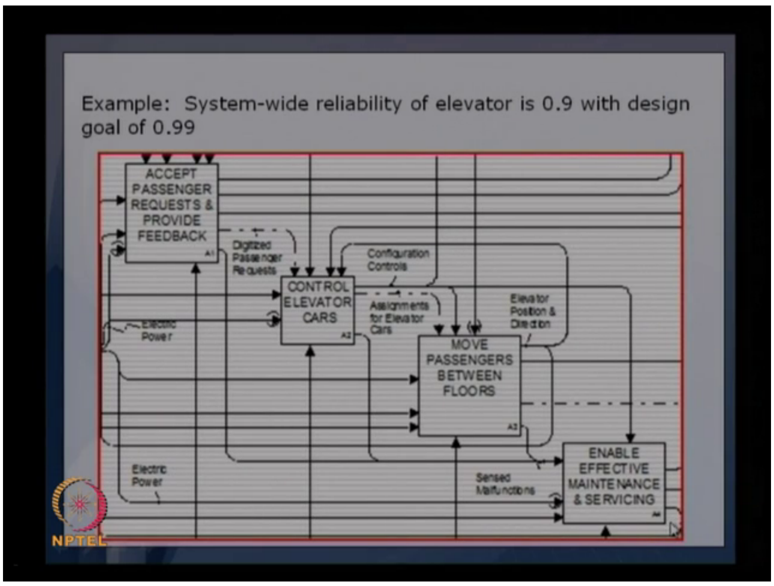
So, here the system level requirement is divided or apportioned out to the systems components not necessarily in equal increments, but keeping a margin of 5 to 10 percent as risk mitigation strategy. So, what we do here is, we look at all the components and then a portion this the main requirement to amongst these components, but I will we do not do it in an equal manner, we will look at the component and then decide what should be the other proportion to which it should be divided.

So, it need not been equal increments, but at the same time we do not give the complete requirement to a all the components, what we do is we keep a 5 to 10 percent margin, over here and then this margin is used for the risk mitigation strategy. So, if you want to avoid the risk at the later stage keep a 5 to 10 percent margin and this margin will be used at later stage to see whether (Refer Time: 12:21) particular risk is coming from any other component. So, that can actually be compensated using this method that is the apportionment method.

One example you could be the cost of operation of the elevator, if you say that it 10,000 rupees per month for operating the system, that is the maximum allowed then we can do 10 percent as a imagined. So, maybe for 9000 we can identify, what should be the maximum cost coming from the components. So, 9000 rupees can be divided amongst these 4 components depending on the importance of each one probably the for the maintenance can actually have 1000 rupees, in the control elevator cars could be for 7000 rupees and then the remaining 1000 rupees can be divided amongst other members.

So, the total 9000 operating cost can be apportion to amongst the members, and 10000 can be kept as a risk mitigation strategy. So, that is the simple way of allocating the system wide requirements in a system. So, here as you can see as I mentioned it is not in equal increment that is 1 of the points you need to note down and again we will need to keep a margin of 5 to 10 percent for risk mitigation.

(Refer Slide Time: 13:27)



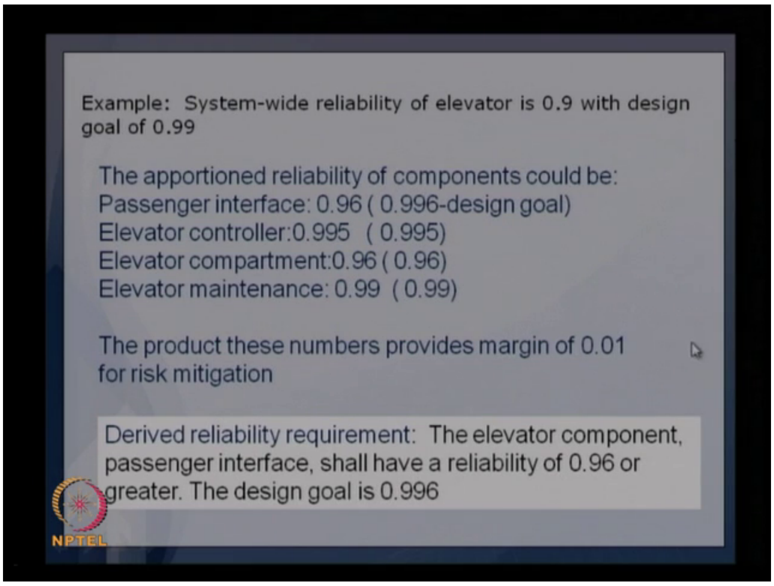
This is another example here the system wide reliability of elevator is 0.9 with the design goal of 0.99.

So, what we are saying here is that, there is a system wide requirement of reliability which is to be maintained at 0.9 and with the design goal of 0.99. So, this is the minimum requirements and the goal is to get a very high reliability of 0.99. Now we have different components in the system and we need to find out what could be the reliability of these components in order to get the; I total reliability of 0.9.

If you take the functions in the elevators in A 1, A 2, A 3 and A 4. So, these are the 4 main functions in the elevator, and we need to have the reliability of these 4 combined together getting a 0.9, that is the requirement. But we do not know how much we should give the value for this and for this, this and this. So, these are the 4 components and we can do it in a the apportionment method we will try to divide these reliability amongst the members, depending on the our knowledge about the system and how it is going to affect the hold overall system, accordingly we will try to provide the reliability values for this.

So, in this case apportionment methods what we do is.

(Refer Slide Time: 14:45)



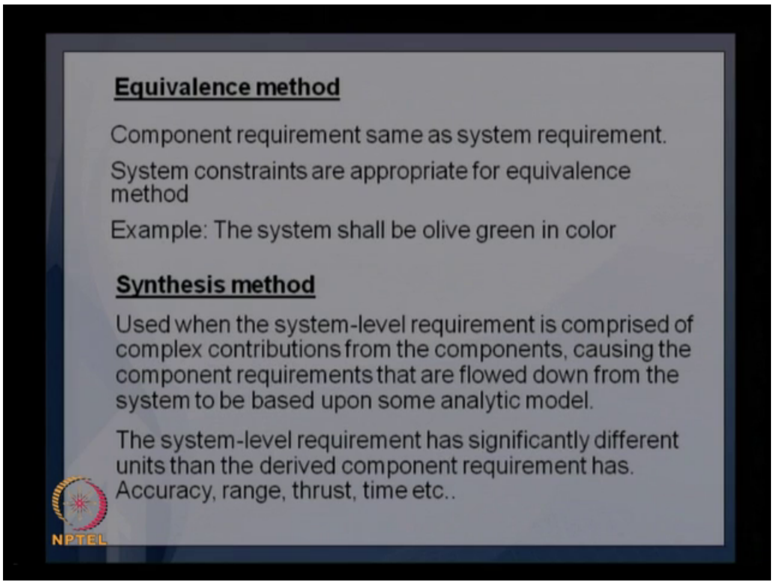
We write down the requirements reliability of components like a passenger interface can have a reliability of 0.96 with the 0.996 as the design goal. Similarly elevator controller can have a reliability of 0.995 and with the design goal of 0.995 and elevator compartment can have a reliability of 0.96 with a design goal of 0.96 and elevator maintenance can have a 0.99 and the 0.99 as the design goal. If you multiply all these reliability we will be getting it as 0.91; that means, there is a risk mitigation value of 0.01 for later use.

So, the total a reliability of components will be 0.91. So, you have a 0.01 or a clearance or the tolerance for risk mitigation. So, this is the way how we do it in apportionment methods and then how do we get this is again it comes from various factors, if you know the particular component you are using, you have to use the reliability value provided by the manufacturer of this particular component or if you know that the corresponding system or there are multiple systems coming in this one. So, we need to look at those systems and it is reliability values and accordingly it is fix a value for this one. Again this is to be done by the design team. So, there is a little bit of subjectivity in this one, but still it is a simple method. So, that is why it is used for the system requirements and allocation of the system wide requirements amongst the components. So, that is the way how the apportionment method works.

So, here once you have this then we can actually identify the derived reliability requirement, the elevator component passenger interface which is this passenger interface shall have a reliability of 0.96 or greater, the design goal is 0.996. So, once you have this one, then you will write down the derived requirement which is coming from the apportionment methods for each component we can write the derived the requirement as the system or the elevator components, I shall have a reliability of 0.96 or greater the design goal is 0.996.

So, this is the way how we give the reliability, how we use the apportionment method to divide the reliability or allocate their system wide requirements to the components.

(Refer Slide Time: 17:07)



Another method is known as equivalence methods, and this method as the name suggests it is basically divided it equally amongst the components. So, the component requirements same as system requirements, there is no difference, but all that system requirement is there the same requirement will be given for the components also. The system constraints are appropriate for equivalents method. So, constraints in terms of when you design the system where will be the system requirements as well as the system needs as well as the constraints.

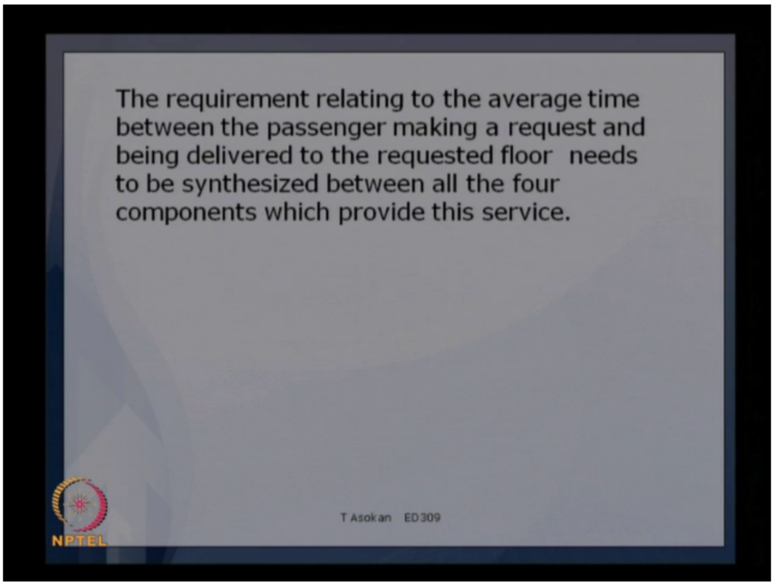
So, what we do here is to use the system constraints basically to we use the equivalents method to allocate constraints to the components. Basically if you look at requirements like the system shall have a olive green in color. So, system shall be olive green color if that is a requirement, then we need to make sure that all the components are of the same color. So, this is known as a simple equivalence method of allocation.

The third one is a synthesis methods, which is used when the system level requirement is comprised of complex contribution from the components, causing the component requirements that are flowed down from the system to be based upon some analytical model. So, this is more of an analytical method of allocation. So, here difficult to identify the component requirements because they are related to different components and each component is having different way of representing it is performance.

So, in this case then we need to have some analytical model to do this. The system level requirement has significantly different units than the derived component requirements has. The system level requirement and the component level requirement may be having different units. So, in that case we will not be able to directly allocate these requirements to the components and therefore, we go for an analytical method. For example, if there is a requirement of system performance time. If you say that the system should provide an output within 20 seconds that is the requirement, but then the processor is one of the component which actually provide this, but the processor speed is different unit and there may be some other motors or something which actually provide the output and it is output will be in a different unit.

So, in these cases we need to ensure that the requirement at the system level is given in a different unit, but the components are in different units. So, we need to have some kind of a mathematical modelling or an analytical method to allocate these system wide requirements to component. So, there we go for the synthesis method. There are different synthesis methods, we will see the methods at later stage, but this is one of the important methods by which we allocate the system wide requirements.

(Refer Slide Time: 19:47)

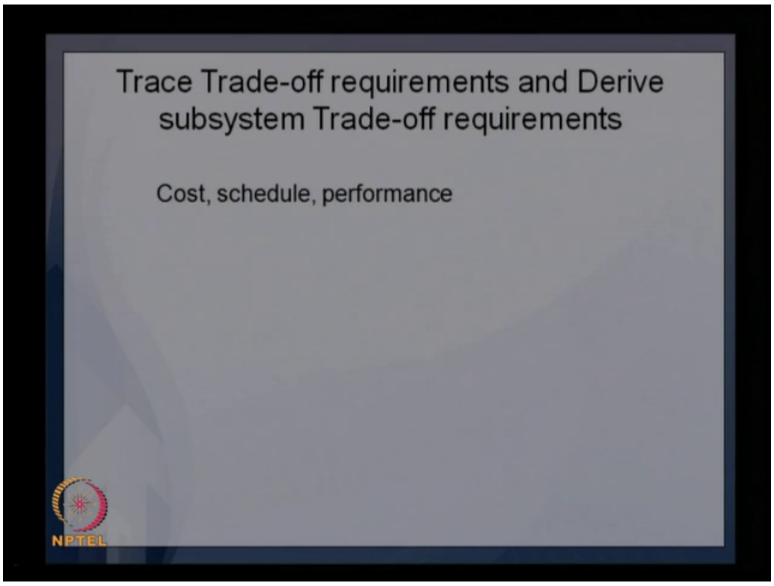


The example is the requirement relating to the average time between the passenger making a request, and being delivered to the requested floor needs synthesized between all the 4 components which provide the service.

Suppose if you have a requirement like this, average time between the passenger making a request and being delivered to the requested floor. So, this needs to be synthesis between the all the 4 components and each component of this system will be having different way of producing an output. Because, it takes a function of the input output interface, the passenger interface there will be a sometime delay in passenger interface and there will be a time delay in the motor output and the acceleration of the elevator car and other elements also.

So, we need to see what kind of a relationship exists between the total time for service as well as other component performance parameters and once we have this mathematical relationship, then we try to identify or try to have a relationship and then see how to allocate the main requirement to this component. So, that is the method of Synthesis methods.

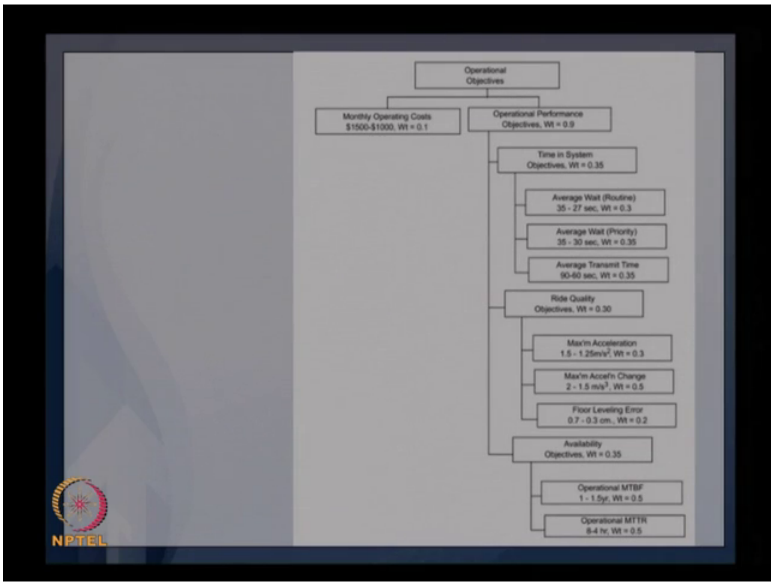
(Refer Slide Time: 20:53)



The next one is basically the trade-off requirements and subsystem trade of requirements. Basically we look at the requirements like cost, schedule, performance etcetera there will be some trade-off in the system because of this. Mainly if you want to reduce the cost or optimize the performance, we need to make some kind of a tradeoff between these parameters the cost the scheduled the performance etcetera.

So, how do we trace this trade-off requirements and then how do we derive the sub system trade-off requirement, because the system will be having some trade-off and sub system will be having at different trade-off. Because there are many other features which actually comes into picture here, the objectives hierarchy what is the system objectives and how the various objectives are given the particular value in the hierarchy, the traders requirement basically depends on those values also. So, we need to look at the hierarchy of the objectives and based on the objectives hierarchy, we need to decide what kind of tradeoffs to be provided in the system.

(Refer Slide Time: 21:57)

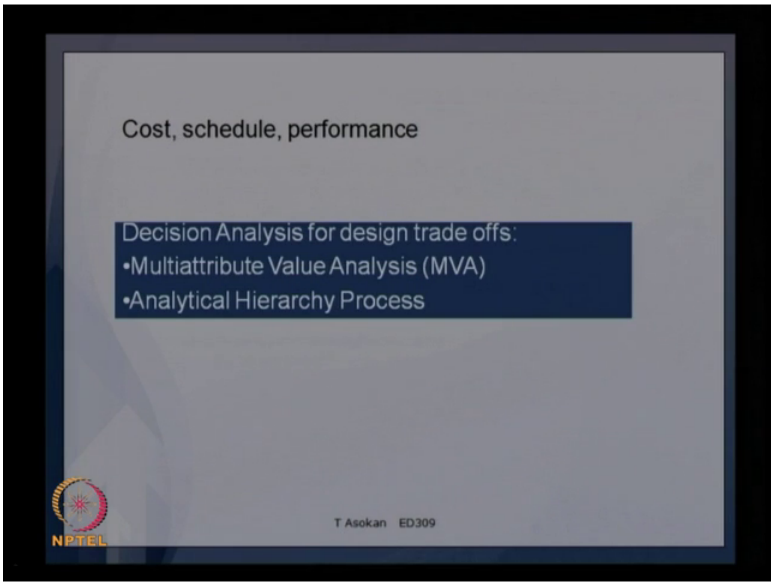


For example if you take the again the elevator case study, you can see that the objectives hierarchy is given here the operational objectives like operational performance and monthly operating cost. So, these objectives are given here you can see that the operational performance objective weight is 0.9, but the monthly operating cost weightage is 0.1. So, here the more weightage is given for operational performance. So, whenever we have a tradeoff, we need to make sure that the performance objective values are met then the cost objectives.

So, we can always have a trade-off with cost to get a better operational performance. And when we look at the operational performance, we will see that a there are many other sub objectives coming from the subsystem like time in systems. So, now, to get the operational performance objective 0.9, we have a timing system objective of 0.35, then we have a right quality objective over here 0.3, and then availability objective of 0.35. So, these objectives to be met when we have the trade-off; so whenever we have a tradeoff analysis, we need to look at the objectives hierarchy and accordingly we need to find out for how we actually allocate these objectives and then decide on the tradeoff of the system wide requirements.

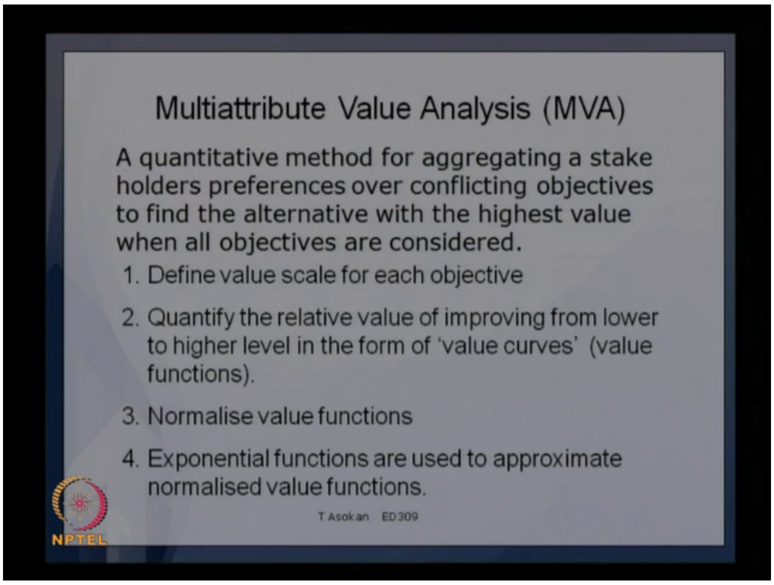
Again different ways of doing this; so we can actually as I told you the cost schedule performance all these can be actually be traded off for various improvement in either performance or the cost or schedule.

(Refer Slide Time: 23:37)



So, this can actually be done by different methods, the decision analysis for design trade-off the standard methods, where we have a multiattribute value analysis which is known as MVA or we can use an analytical hierarchy process. So, any one of these methods can be used for the tradeoff analysis. The multiattribute a value analysis is one of the most common method used to find out how do we actually allocate the requirements and the based on them value functions of the system.

(Refer Slide Time: 24:07)



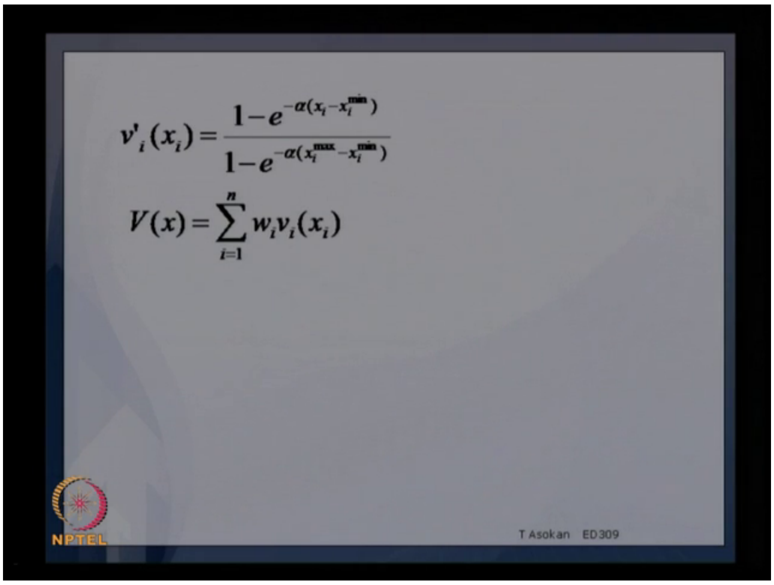
So, here this is a quantitative method for aggregating a stakeholders preference over conflicting objectives to find the alternative with the highest value when all objectives are considered. So, this is analytical method use in order to get a an alternative with the highest value, when all objectives are considered. So, you have multiple objectives and they are little bit conflicting. So, in this situation how do we actually get the alternative, the best alternative with the highest value and all objectives are considered. So, that is the multiattribute value analysis.

So, the various steps involved in multiattribute value analysis are basically we define a value scale for each objective, that is what should be the scale on to which the objectives to maintain. So, what is the minimum value to be and what is the maximum you can have, and then quantify the relative value of improving from lower to higher level in the form of value curves. Now, we have a minimum value and a maximum value, and what kind of a variation we are expecting from this minimum to maximum or it is a linear variation or it is an exponential variation, understand increasing or decreasing variation.

So, what kind of a variation we are expecting that is to be given in terms of value functions? And then we need to normalize this value function, because the values are different scales. So, we need to have an organized value function and once we have the normalized value function, we can actually have a relationship which will actually optimize the total value function and that value function can be used for optimizing the objectives.

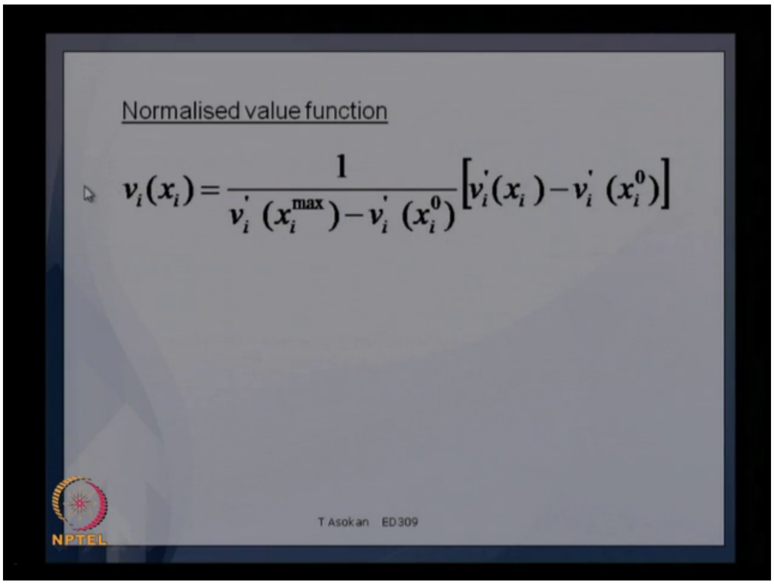
So, normally exponential functions are used to approximate normalized value functions. This value function as I mentioned these value curves, these are the value functions. So, normally exponential functions are used to approximate the normalized value functions and once we have these exponential functions by controlling some of the parameters, we would get various curves depending on the requirements.

(Refer Slide Time: 26:05)



This is the exponential function normally used and the normalization of function is done using this method.

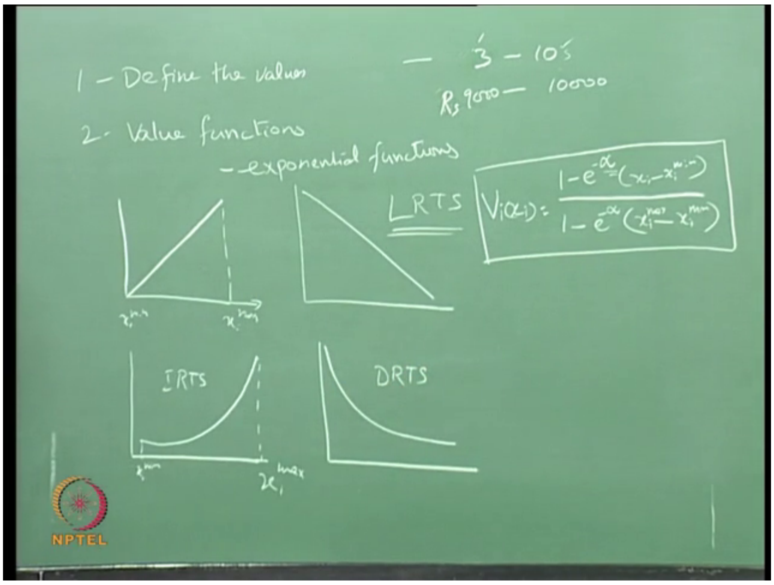
(Refer Slide Time: 26:10)



So, we get the value function v i x i prime based on the exponential curve, and then we normalize it along the different values that is the maximum value and the minimum value, and the value of x i and x i 0. So, this actually will give you a normalization of the function. So, this normalized function will be used for getting the total value function that is v i x i. So, if you have many value functions any value functions. So, we will take the total value function as v x is equal to sigma is equal to 1 to n w i v i x i. So, the v i x i is the value function, w i is the weight of the value function.

We will take a simple example and tell go through these steps and I will tell you how to do this. So, we will go to the board and then explain how do we actually use the value functions and value curves to get the total value function, when we have multiple value functions or multiple objectives in a system. So, as I mentioned the first task is basically do define the values.

(Refer Slide Time: 27:16)



So, define the values basically for each objective. So, we can say that the timing system is between. So, the minimum value is 3 seconds, and the maximum is 10 seconds. So, we can actually define this kind of minimum to maximum values for any function. So, it can be 3 to 10 seconds or you can say the cost which should vary from 9000 minimum cost is 9000 and maximum we can go for is up to 10,000 rupees.

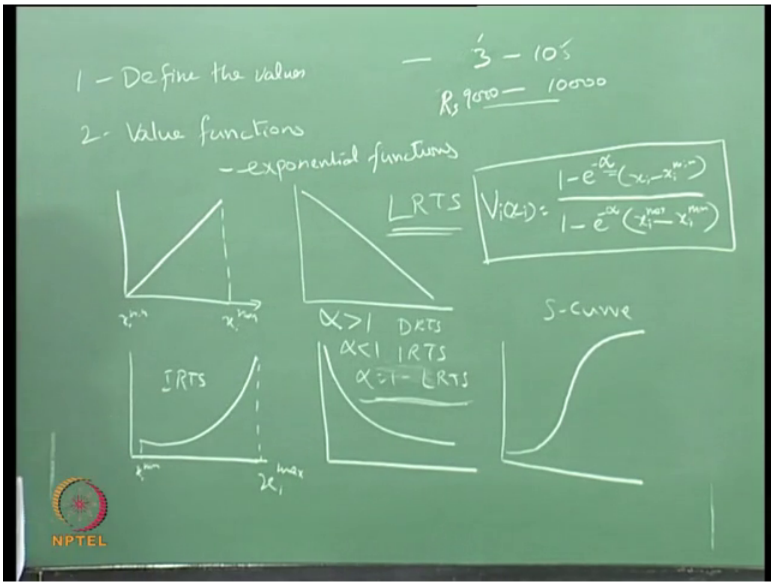
So, this way we can actually define the values for the function. So, that is define the values for various objectives, and then define value curves for value function. So, that is the define value functions. So, again we are using the exponential functions, the exponential functions are used for value functions or value curves. So, when you use exponential curve actually already explained the exponential function, it is 1 minus v i x i is given as 1 minus e to the power of minus alpha x i max, minus x i minimum divided by 1 minus e to the power of minus alpha x i max, minus x i minimum. So, this is the way how we define the value curve.

So, this x i minimum is this minimum value and this maximum is the maximum value. So, between this 2 values minimum and maximum we need to define a value curve how the value should change from minimum to maximum. So, we can actually have different values and depending on the value of alpha, you can get different kinds curve. So, the first 1 is linear return to scale. So, you can have a linear return to scale function or we can have a sorry this is the linear return to scale. So, you can actually have a linear return to scale like this. So, x i is the x i max this can be the x i max and this is the x i minimum. So, as it changes this will be varying linearly.

So, this is known as v linear return to scale. Or you can have it be the other way this way also maximal to minimum, you can have a linear return to scale. So, this is known as linear return to scale value functions or you can have an increasing return to scale, increasing return to scale is like this. So, as you go up there you return increases. So, this is an increasing return to scale again this is x i min and x i max. So, you can have increasing returns to scale or you can have decreasing returns to scale. So, this is a decreasing return to scale. So, this is increasing return to scale and this is decreasing return to scale.

So, you can have any this any one of these 4 function, and apart from this you can have other functions like s curve sorry this is again increasing return to scale and you have s curves also you can have an s curve.

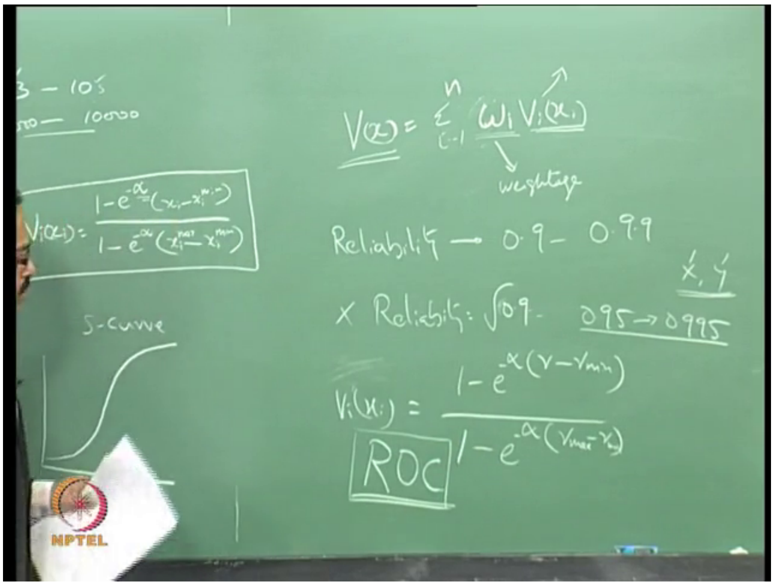
(Refer Slide Time: 31:08)



So, this is an s curve. So, you can actually have an increasing and decreasing return to scale. So, this is known as s curve. So, you can have any one of these outputs, based on the value of alpha as alpha changes will be having various and. So, an alpha is greater than 1 then you have a decreasing return to scale and alpha is less than 1 you will be having a increasing return to scale and then alpha is equal to 1, you have a linear return to scale.

So, these are the various ways you can actually have the curves. So, you can use any one of these curve depending on the system, if you want to have a linear variation of the cost you can go for a linear curve or you want to have an increasing return or a decreasing return you can actually go for accordingly and choose the value of alpha and then defined the function v i x i. And once you have the v i x i we go for the normalization of the function, normalization I already showed you the equation. So, you have the equation for normalization and once you have this normalized curve we can get the final function as v x is equal to.

(Refer Slide Time: 32:28)



So, we can use it as v x is equal to sigma I is equal to 1 to n w i v i x i.

So, you will be having many function you will be having many objectives, because one could be the time, one could be the cost and other could be the reliability or it can be any other objectives. So, you can use any one of these you can use all the objectives. So, you will be getting v i x i for all these objectives and then we will have a weightage factor w i and this w is the weightage for individual objectives. So, this weightage for individual objective and this is the value function and we complete the take the com sigma of this then you will be getting the total value function for are the objectives.

So, using this objective function you can actually define what should be the actual trade-off to be carried out. So, this function will help you to do the tradeoff analysis based on the varying objectives as well as the weightage for given for each individual objective.

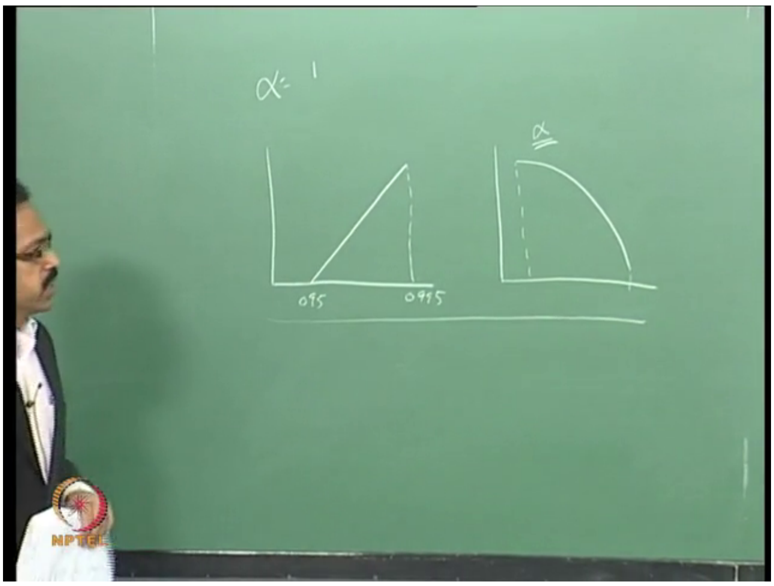
So, that is the method by which we do the trade-off analysis based on the value function for each objective.

So, let us look at the how to get the weightage for each objective or before that we will take a simple example and then they show you how to do this. Let us take the reliability of system. So, take an example of a reliability and how do we do suppose you want the reliability to be between 0.9 to 0.99. So, that is the reliability requirement and then we need to find out what should be the reliability of x and y. Suppose you have 2 this is the reliability total reliability of the system between 0.9 to 0.99, and you have a here 2 components x and y. So, you can have the function for what should be the reliability of x, but should be the reliability of y.

So, in order to get the reliability of x probably you can take it as a simple measure of square root of 0.99 0.9 and then we can get it as 0.95 as the reliability of x, and then for y also we can take it as 0.95 and then we can define a function it is 0.95. So, it varies from 0.95 to 0.995 because you take the square root of 0.99 also. So, the reliability of x varies from 0.95 to 0.995.

Then we can define a function 1 minus e to the power of minus alpha, reliability minus reliability of the minimum reliability divided by 1 minus e to the power of minus alpha r max minus r sorry r max minus r min. So, this function can be defined for the reliability of x and we can plot the reliability under different situations.

(Refer Slide Time: 35:46)



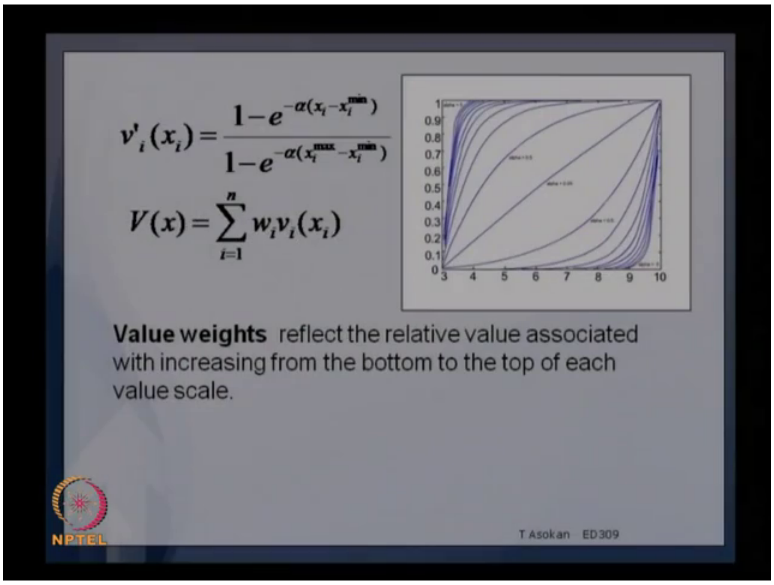
So, you can actually have a reliability of if you take the alpha is equal to a particular value of 1, then you can actually get it as the value function will be point nine to 0.995 like this sorry 0.95 to 0.99 995 or if you want to change this to a different method or it can have it is a returned to scale then you can actually have this way also.

So, depending on the value of alpha you can actually choose what kinds of relationship you want this, to have in while the reliabilities varying from one value to the other value. That is the way how we use the value functions in calculating the total objective value considering multi multiple objectives, and then we need take need to take the weightage wi.

So, we will take this weightage w i. And there are different methods to get the weightage we will use the method called rank order clustering methods. So, that is a method for getting the weightage of various objectives. So, you have multiple objectives, we need to find out which objective should be given the higher weightage that actually comes from the objectives hierarchy. So, we use the objective hierarchy and the method called rank order clustering to get that outputs.

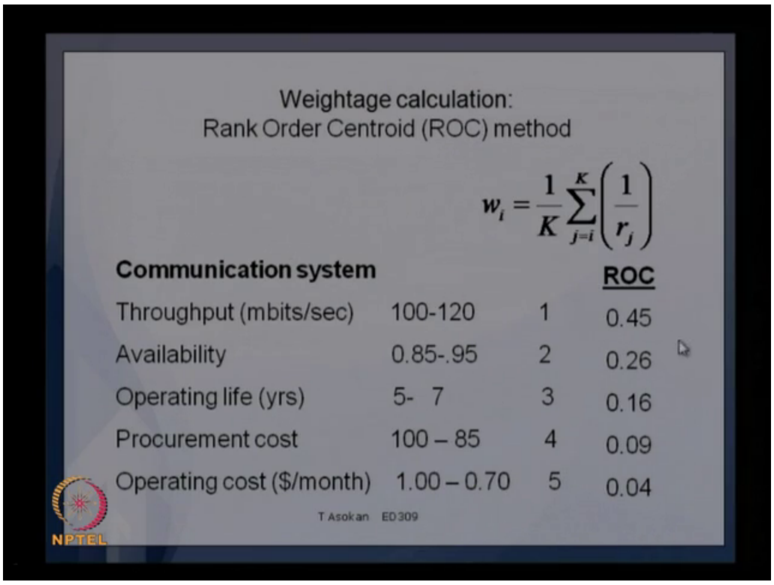
I will just go through the rank order clustering method, but before that I can just show you the curve how it will look like.

(Refer Slide Time: 37:13)



So, when you have various values will be getting the output like this, when you have an alpha 0 value it will be a straight and then otherwise be getting various curves depending on the value of alpha. This is how the objective functions or the value function changes as you change the alpha ok.

(Refer Slide Time: 37:32)



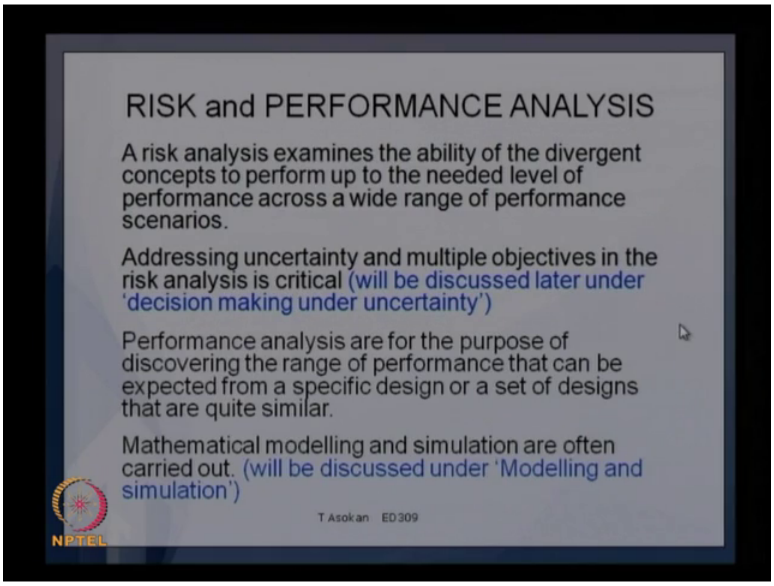
So, I will mentioned about the weightage calculation we are need to find out the w i value for the weightage of each individual objectives. So, this actually is done using this method called rank order clustering or rank order centroid methods it is an rank order centroid. So, the rank order centroid method is given w i is equal to one over K sigma j is equal to I to K 1 over r j, where r j is the value of the objective hierarchy value given by the system and the system design initially it is time of objectives hierarchy development, if give the value of objective hierarchy values for each objective and that value is taken as r j and we take a centroid of this 1 using the rank order centroid methods.

So, for example, you can see this is a communication system requirement, there are requirements like throughputs availability operating life procurement cost and operating cost, these are the various requirement and this you can see these are the values. So, values of these objectives throughput objective 100 to 120, availability is 0.85 to 0.95, operating life is 5 to 7, yes and procurement cost 100 to 85, operating cost is 12.7.

So, these are the values which we will be using in the value curve the minimum and maximum values. So, based on this we can actually have the v i x i that is for the value functions, and then we have to get the w i which is the weights. So, since these are the objectives hierarchy. So, this is the most important 1 1, 2, 3, 4, 5. So, based on this hierarchy we will get the value of w i for each objective.

So, in this actually we can use the same equation, this equation and get the objectives hierarchy. So, when this is the rank order centroid method we will get the weightage w 1 as 0.45, w 2 as 0.26, w 3 as 0.16, w 4 as 0.09 and w 5 as 0.04. So, is there ROC method to get this weightage w i, and then the v i x i can be obtained using the exponential function method and then w i and v i x i you can multiply and then get the total value function for the communication system. So, the communication system function will be v x will be equal to sigma I is equal to 1 to n w i v i xi, where v i x i are the individual value functions ok.

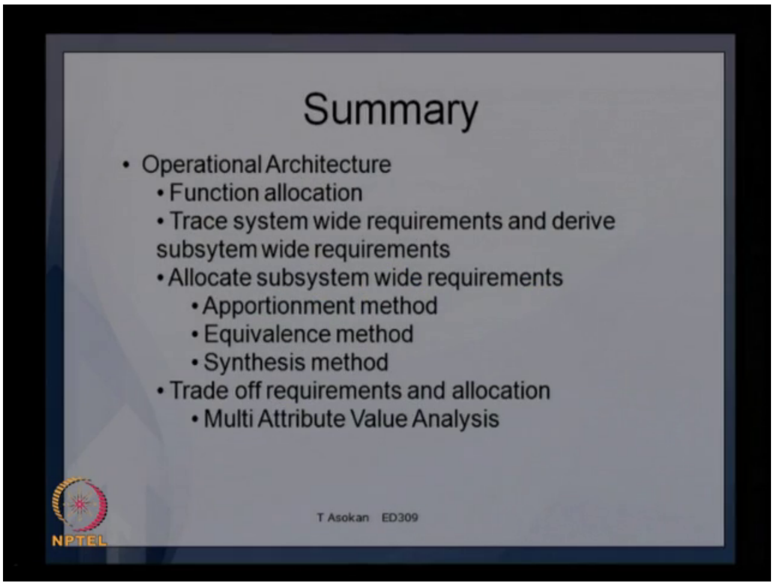
(Refer Slide Time: 39:58)



Then other things to be considered when we do the allocation architecture are the risk and performance analysis. So, the risk analysis examines the ability of the divergent concepts to perform up to the needed level of performance across a wide range of performance scenarios. So, we will be addressing this in a separately. So, we have a separate section for decision making under uncertainty. So, we will be looking at this aspect under that category during that lecture and then another one is the performance analysis for identifying the system performance. So, this also will be looking at we will be discussing this during the modeling and simulation lecture.

So, this both will be discussed at later stage because we have separate session for this. So, the risk and performance analysis not covering in this lecture because we are having a separate session for this ok.

(Refer Slide Time: 40:50)



To summarize what we discussed in during the last 2 lectures where on the operational architecture development or which known as allocation architecture also. We using the functional requirements develop the functional hierarchy and functional architecture, then we developed the physical architecture then the next one was basically how do we allocate this function to the physical elements. So, we discuss different methods of mapping the functions to components, and then we looked at how do we allocate the system wide requirements as well as the internal input output requirements, then system a tradeoff requirements. And we discussed many methods to do this like apportionment equivalent method and synthesis methods and then we discussed about the trade of requirements and allocation of the trade of requirements, and we discussed the method called multi attribute value analysis.

So, these are the topics we discussed during the last 2 lectures, and then the next one is basically looking at the qualification requirement and verification requirements of the system, that is next logical step in the system design. So, we will discuss about the qualification and verification of system, using the standard methods or the standard procedures and really discuss in the next class. So, till then goodbye.