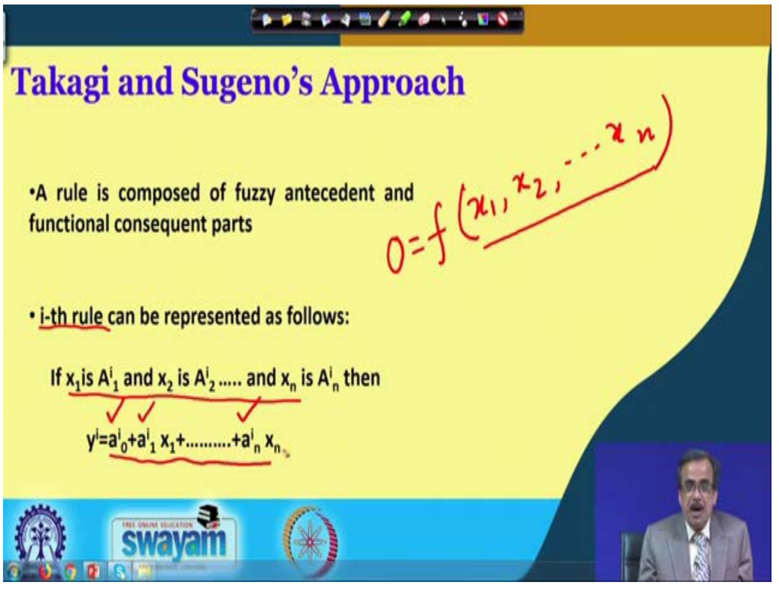
Fuzzy Logic and Neural Networks

Prof. Dilip Kumar Pratihar
Department of Mechanical Engineering
Indian Institute of Technology, Kharagpur

Lecture – 11
Applications of Fuzzy Sets (Contd.)

We are going to discuss the working principle of another very popular fuzzy reasoning tool, that is known as Takagi and Sugeno's approach.

(Refer Slide Time: 00:27)



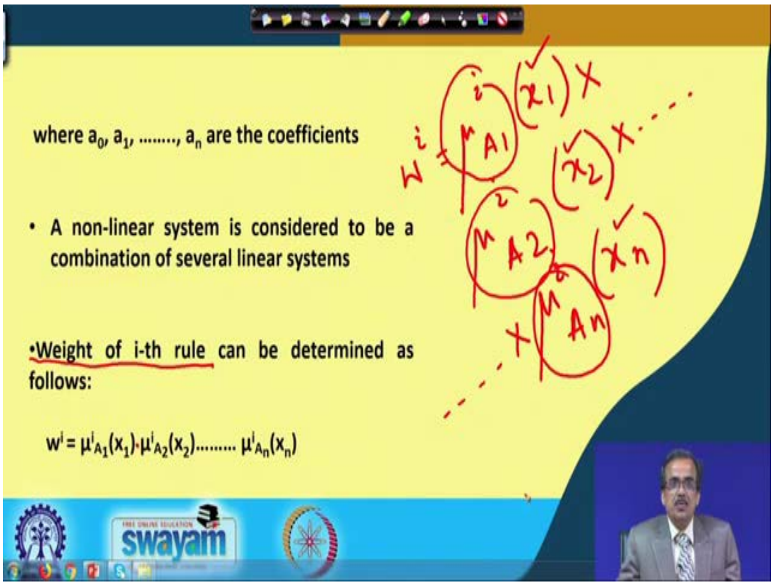
Now, here, in Takagi and Sugeno's approach, what you do is, a particular rule, say i-th rule is represented as follows. Now, here, the rule is written like this if x_1 is A_1^i and x_2 is A_2^i and there are a few terms here and x_n is A_n^i then $y^i = a_0^i + a_1^i x_1 + ... + a_n^i x_n$. Now, here, I have already mentioned a little bit that in this approach for a particular rule, the output of a rule is nothing, but the function of the input parameters or the input variables. Now, here, in this rule I am considering that there are n such variables like your x_1, x_2 up to x_n. So, this output that is y^i that is represented as a function of the input parameters. Now, if you see, we have got a few coefficients for example, say $a_0^i, a_1^i, ..., a_n^i$ these are all coefficients.

Now, these coefficients are to be predetermined. How to determine that? Now, what I do is, we take the help of some optimisation tool and with the help of some available data;

so, we try to find out what should be the numerical values of the coefficients. Now, generally, we use an optimization tool that is known as the least squared technique. So, this least squared technique actually, we generally use, to find out the values for the coefficients.

Now, you can see this output has a linear function of the input parameters and that is why so, this approach can be termed as linear approximation of a non-linear system.

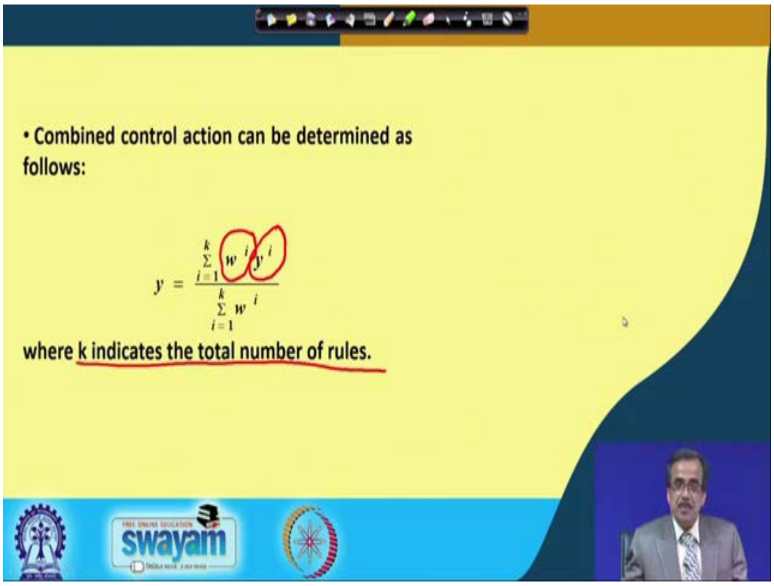
(Refer Slide Time: 02:46)



So, this is nothing, but actually the non-linear system's representation, as a combination of several linear systems. Now, here, if I know the i-th rule; now what you do is, we try to find out the strength of the i-th rule or the weight of the i-th rule. Now, this weight of the i-th rule is represented as $w^i = \mu^i_{A1}(x_1) \times \mu^i_{A2}(x_2) \times \ldots\ldots\ldots\ldots \times \mu^i_{An}(x_n)$. That means, the strength of the i-th rule that is represented as the membership value of A_1 corresponding to the i-th rule, the moment I am passing this x_1 as the input variable multiplied by the μ_{A2} the corresponding to the i-th rule, and the moment I am passing this x_2 as the input variable.

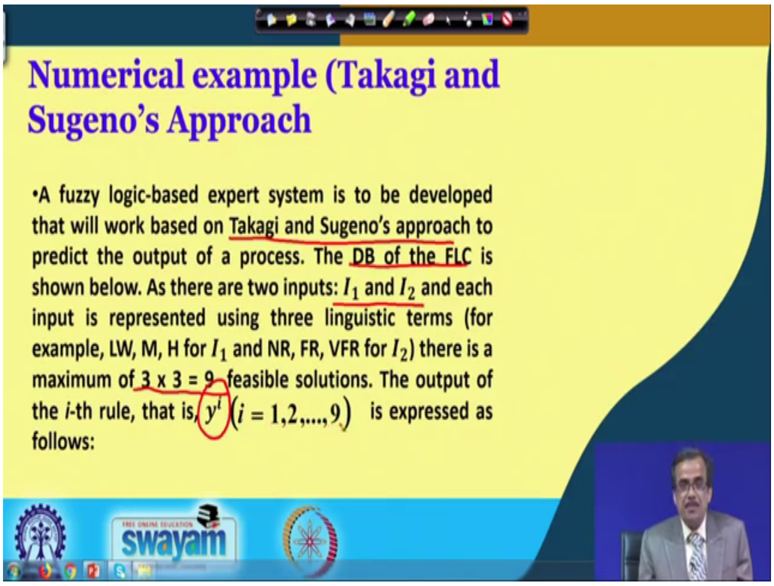
So, all such μ values we try to find out and the last term is nothing, but μ^i_{An} corresponding to your x n. So, here as if we are passing all the input parameters or the input variables like x_1, x_2 and x_n and we try to find out what should be the μ value, what should be the μ value and what should be the μ value and after that we multiply all the μ values and that will be the weight of the i-th rule.

(Refer Slide Time: 04:31)



And, once I got this particular weight of the i-th rule, now next what we can do is, we can find out the output very easily using this particular the expression. Now, for the combined control action, this output $y = \frac{\sum_{i=1}^{k} w^i y^i}{\sum_{i=1}^{k} w^i}$. Now, here, k indicates the number of the fired rules and what is y^i; y^i is nothing, but the output corresponding to your i-th rule and this w^i is nothing, but the weight. Now, how to determine so, this particular output, which is the function of the input parameters.

(Refer Slide Time: 05:21)



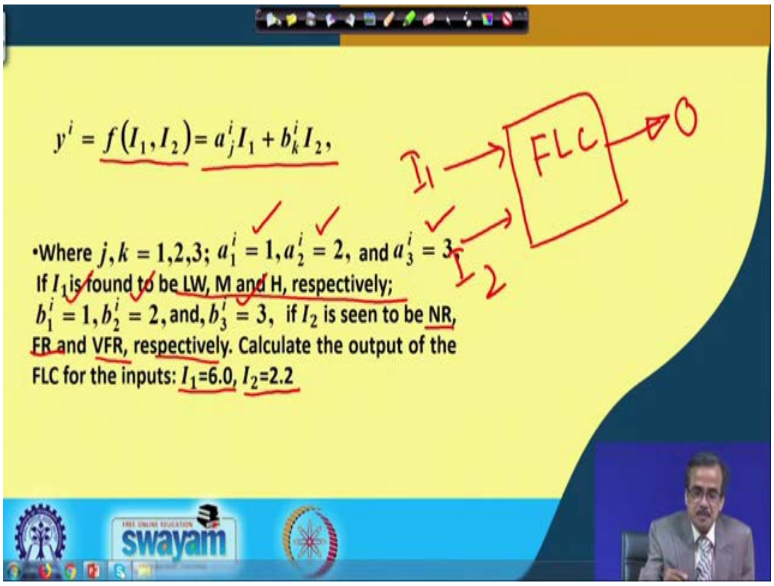
So, that I am going to discuss after sometime. But, before that let me tell you that, here in Takagi and Sugeno's approach, the way we express the output as a function of the input parameters. Now, if I just read one rule so, no control action will be coming to my mind and that is why actually here the interpretability is much less. Although we can go for the better accuracy, because we take the help of some optimizer, some optimisation tool and if you just optimise with the help of some known data. So, there is a possibility, you will be getting very accurate coefficients, the values of the coefficients and ultimately, you will be getting very accurate output.

So, the predication will be a really good and very accurate in this Takagi and Sugeno's approach. Now, to explain the working principle of this approach further, so, we are going to take the help of one numerical example. Now, let me give the statement of this particular the numerical example which we are going to solve. Now, this statement is as follows: like your say a fuzzy logic-based expert system is to be developed, that will work based on Takagi and Sugeno's approach to predict the output of a process. Now, the database of the FLC is shown. So, I am just going to show you the data base, that is the membership function distribution of these particular FLC, particularly for the two inputs because for the output variable, there is no such membership function distribution.

So, as there are 2 inputs I_1 and I_2 and each input is represented using three linguistic terms for example, say low, medium and high for I_1 and near, far and very far for I_2.

So, there is a maximum of 3 multiplied by 3, that is, 9 feasible solutions. The output of the i th rule that is denoted by this y^i, i varies from 1 to up to 9 is expressed as follows.

(Refer Slide Time: 07:43)

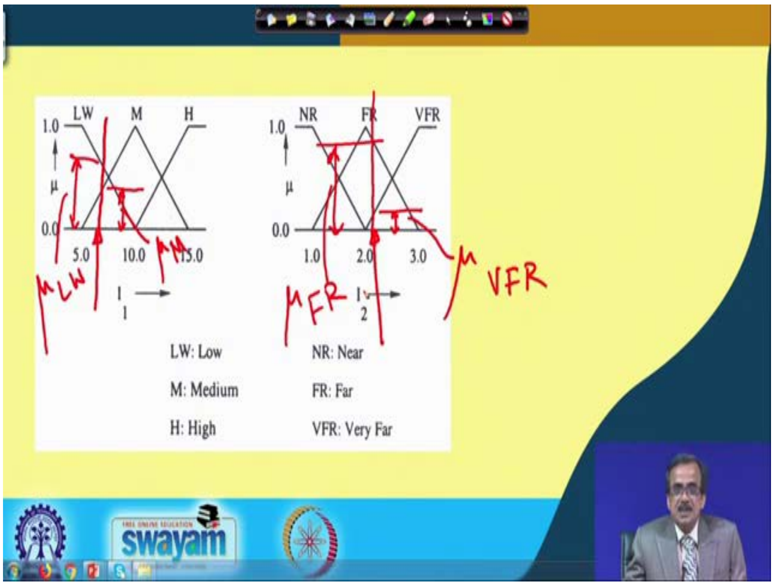


So, I am just going to give you that particular expression for the output of the i-th rule. Now, the output of the i-th rule that is y^i is nothing, but a function of two input variables, that is your life f of I_1, I_2 and that is nothing, but $a_j^i I_1 + b_k^i I_2$. So, what you do is we consider so, this is a linear function of the input variable.

So, output of i-th rule is nothing, but the linear function of the two input variables, your I_1 and I_2, where j k are 1 or 2 or 3. Now, this a_1^i for example, if I put j equals to 1. So, I will be getting $a_1^i = 1, a_2^i = 2, a_3^i = 3$, if I_1 is found to be low, medium and high, respectively. Now, similarly this b_1^i that is I am just going to put k equals to 1. So, $b_1^i = 1, b_2^i = 2, b_3^i = 3$, if I_2 is seem to be near, far and very far, respectively.

Now, we will have to calculate the output of the FLC corresponding to the inputs like I_1 equals to 6.0 and I_2 equals to your like 2.2. So, this is the statement. So, this is a very simple system having 2 inputs and 1 output. So, I have got this particular I_1 and I_2 and I will have to find out this particular output and this is Takagi and Sugeno’s fuzzy logic controller. And, let us see, how to determine the output for a set of inputs.

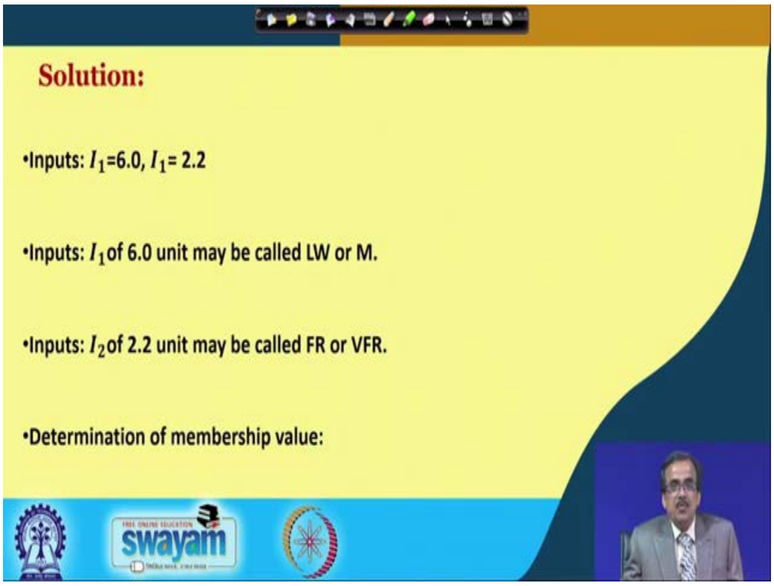
(Refer Slide Time: 09:48)



Now, if you see the membership function distribution for the inputs like this is the membership function distribution for the first input, that is your I_1, the range for I_1 is 5.0 to 15.0. And, this particular range is expressed using 3 linguistic terms like your low, medium and high and as I told previously that for simplicity, we have consider the triangular membership function distribution.

Now, here, we consider one isosceles triangle. Similarly, for this low, we consider some sort of your the right angle triangle and for a high also, we are going to consider some sort of say right angle triangle. Now, similarly for this I_2 the second input variable, what you do is, the whole range starting from 1.0 to 3.0, that is divided into 3 linguistic terms; that means, 3 linguistic terms are used to represent I_2, one is your this NR is the near, FR stands for far and VFR is your very far. So, using actually the three linguistic terms, we can represent the input variables like your I_1 and I_2 and once I got this particular representation for the inputs that is nothing, but the database.

(Refer Slide Time: 11:23)

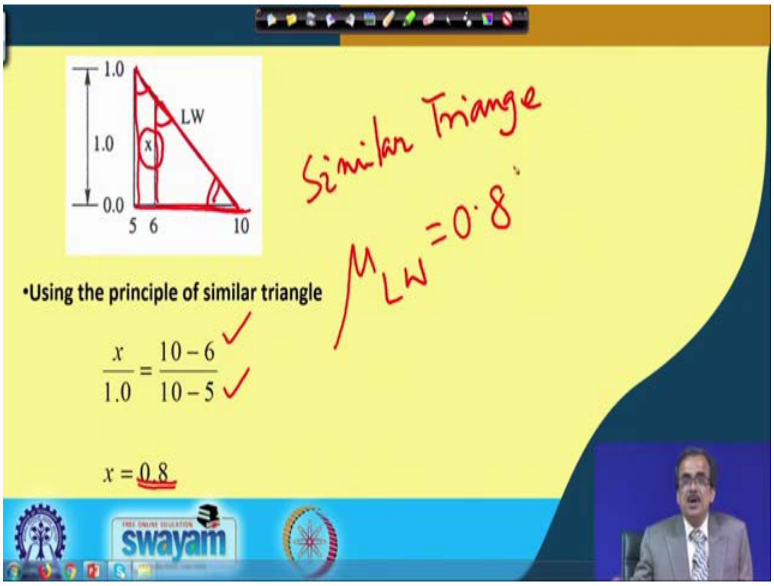


So, now you are in a position to find out what should be the output for the set of inputs. Now, let us try to concentrate. So, here I_1 is 6.0 and this should be in fact, your I_2. So, here there is a small mistake. So, this should be I_2. So, I_2 is equals to 2.2. Now corresponding to your the 6.0 and 2.2. So, let us try to find out like the membership function value. So, 6.0; that means, I am here so, 6.0 I could be here, now 6.0 can be called medium with some membership function value and it can also be called low with another membership function value.

So, it is called medium with this much of membership function value, say μ_{medium} and this can be called low with another membership function value and this is nothing, but is your μ_{low}. Now, similarly like 2.2 so, this value of I_2 is 2.2 and corresponding to this particular 2.2. So, I can find out what should be the membership function value. So, if it is very far so this is nothing, but the membership function value for very far and similarly, this is the membership function value for your the far and once you have calculated so, these membership function value. So, we can proceed further and how to calculate this membership function value that I have already discussed in details.

Now, here so, we have got that this I_1 that is equal to 6.0. So, it may be called either low or medium. Similarly your I_2 that is 2.2 can be called either far or very far. Now, once you have got this particular μ value, now let us see how to determine the μ value, how to determine the μ value and that I am going to discuss once again.

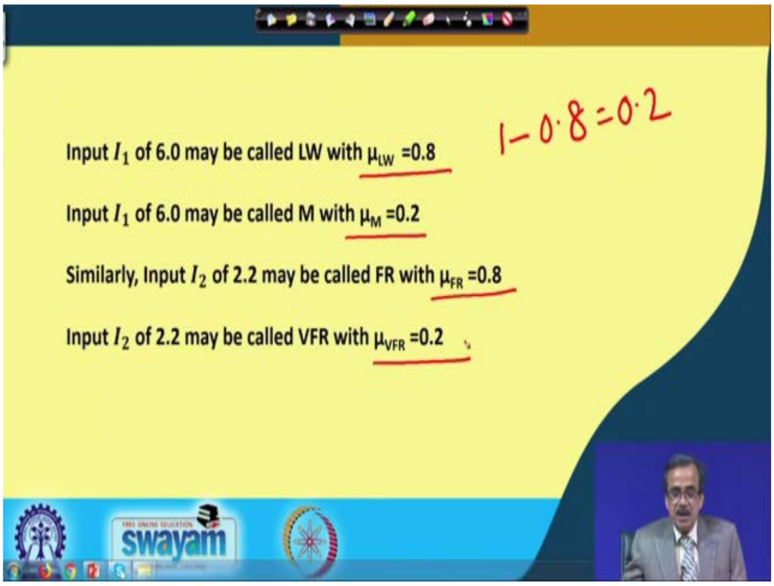
(Refer Slide Time: 13:40)



Now, here you see so, this is your the membership function distribution for low. So, I am just going to show. So, this particular right angle triangle, at it starts from 5 to and it ends at 10 and corresponding to this particular 6. So, what I will have to do is, I will have to calculate this x. Now, as we have already discussed once again we are going to use the principal of the similar triangle. So, if I use the principle of similar triangle, I can find out what should be the value of x for example, say.

So, this particular triangle is similar to your this triangle; that means, this angle is equal to that particular angle and this angle is the common angle. So, I can write down x divided by 1.0 is nothing, but 10 minus 6 divided by your 10 minus 5. And, now if I just find out the value of x, x will come out to be equal to 0.8. So, this membership value that is your μ_{low} is nothing, but is your 0.8. So, this is the way actually we can determine the value of the membership.

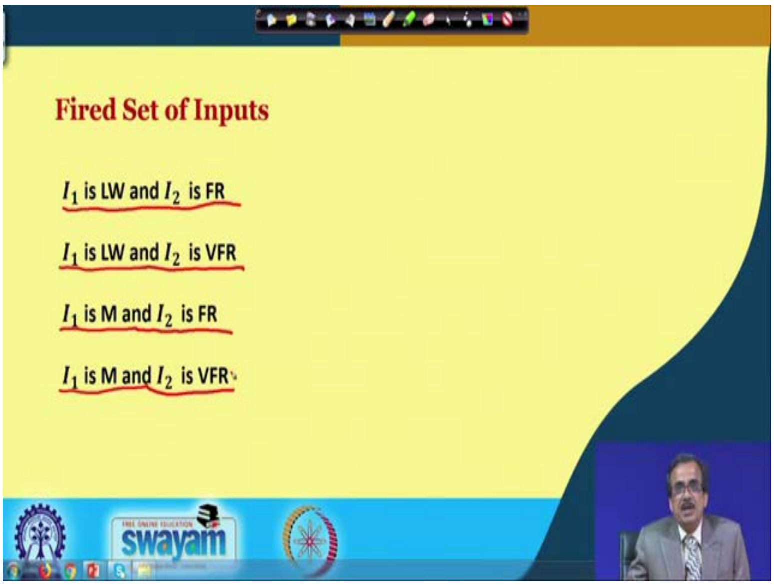
(Refer Slide Time: 15:09)



So, this input I_1 of 6.0 may be called low with membership value 0.8 and the same input I_1 that is equal to 6.0 may be called medium with your the membership value of 0.2 and that is nothing, but 1 minus 0.8 and that is equal to 0.2. Similarly, the input I_2 of 2.2 may be called far with the membership function value. So, μ_{far} is 0.8. So, this can be calculated by following the same procedure.

Now, the input I_2 of 2.2 may be called very far with the membership function value 0.2. So, for each of these inputs I_1 and I_2 and with respect to their linguistic terms, we are able to find out what should be the membership function values. And, once you have got this particular membership function values.

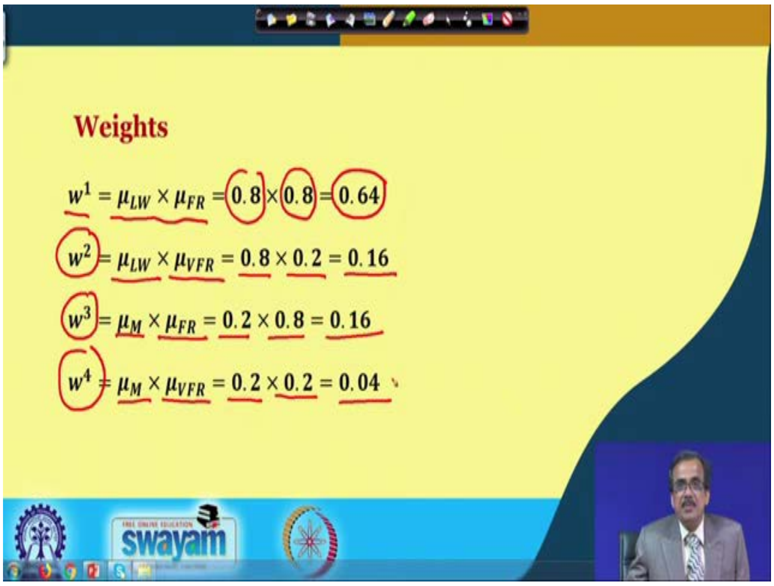
(Refer Slide Time: 16:21)



So, we are in a position to find out what should be the weight of each of these particular the fired rule. Now I before calculate the weight of the fired rule let me try to identify or let me try to mention here the set of fired inputs. Now, the set of fired inputs are as follows. So, if I_1 is low and I_2 is far.

So, this is actually the first set of the fired input parameters or input variables, the second set of fired input parameters are if I_1 is low and I_2 is very far, then the third setup of fired input parameters if I_1 is medium and I_2 is far and the fourth set of your this fired inputs is if I_1 is medium and I_2 is very far. Now, corresponding to these sets of fired input parameters, we should be able to find out the weight. Now, let us see how to find out these weight values.

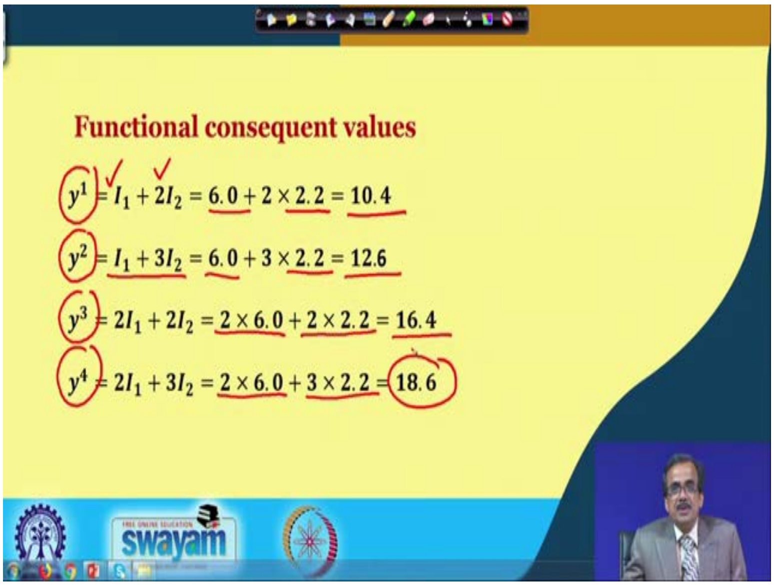
(Refer Slide Time: 17:36)



Now, to determine the weight values, what you do is. So, we try to find out w_1 that is nothing, but the weight of the first fired rule and this w_1 is nothing, but μ_{low} multiplied by μ_{far}. Now, μ_{low} is 0.8 and μ_{far} is once again 0.8 and if you multiply. So, you will be getting your 0.64. Now, similarly corresponding to the second the fired rule, what you can do is, we can find out the weight, that is, w_2 is nothing, but μ_{low} multiplied by $\mu_{veryfar}$, that is 0.8 multiplied by 0.4 and that is nothing, but 0.16

Now, similarly corresponding to the third rule; so, we can find out that is nothing, but μ_M multiplied by μ_{FR} and that is equal to 0.2 multiplied by 0.8. So, we will be getting 0.16 and corresponding to the fourth rule, the weight will be calculated as follows like your μ_M multiplied by μ_{VFR}, that is 0.2 multiplied by 0.2 and here, we will be getting 0.04. So, this is the way actually, we will have to calculate the weights of the different fired rules.

(Refer Slide Time: 19:13)

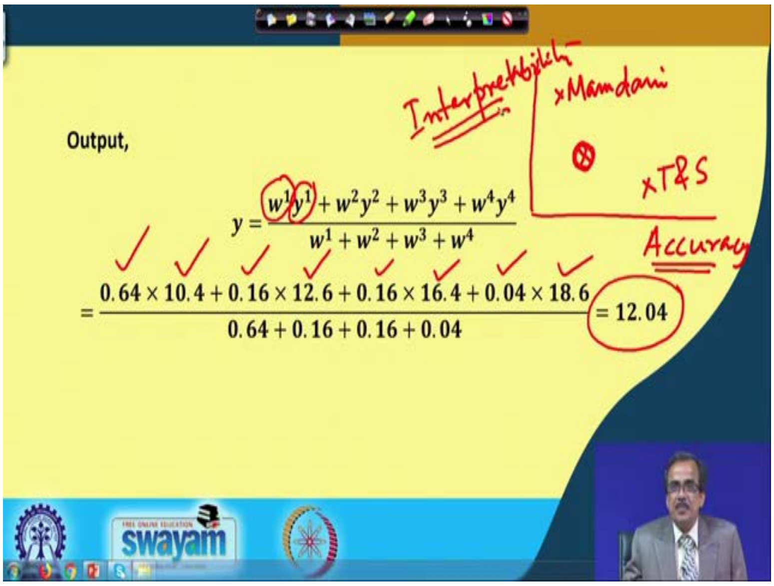


So, once you got this particular the weights now, we are in a position to find out. In fact, the output of each of the fired rule, and then, I will combine. Now, the output of the first fired rule that is denoted by your y_1 that is nothing, but your $I_1 + 2I_2$. Now, how to find out this particular coefficient? The values of the coefficients I have already defined, that this I_1 is represented using three linguistic terms and for each of the linguistic terms, there is a separate value for the coefficient. For example, if it is low, medium and high, if it is low the coefficient is 1, so, if it is medium the coefficient is 2 and if it is high the coefficient of this thing is 3 and something like this; the same is also for I_2, the coefficient of I_2. So, this y_1 is nothing, but $I_1 + 2I_2$ and I_1 is what? I_1 is your 6.0 and I_2 is your 2.2. So, I if I just calculate so, I will be getting the output of the first rule is your 10.4. Now similarly, we can find out what should be output for the second fired rule and that is nothing, but $I_1 + 3I_2$ and that is nothing, but 6.0 plus 3 multiplied by 2.2 and this is your 12.6.

Similarly, your y^3 is nothing, but $2I_1 + 2I_2$ and that is nothing, but 2 multiplied by 6.0 plus 2 multiplied by 2.2 and I will be getting 16.4. Then comes your y^4 is nothing, but $2I_1 + 3I_2$; that means, your 2 multiplied by 6.0 and 3 multiplied by 2.2 and if you just add them up you will be getting 18.6.

So, till now, we have determined the weights of each of the fired rules and the output of each of the fired rules. Now, I am just going to find out like how to determine the output considering the combined control action.

(Refer Slide Time: 21:39)



Now, this output is nothing, but is your w^1 multiplied by y^1. So, this is actually the weight of the first fired rule and this is your the output of the first fired rule; plus w^2 multiplied by y^2 plus w^3 multiplied by y^3 plus w^4 multiplied by y^4 divided by w^1 plus w^2 plus w^3 plus w^4. And if you just substitute all the numerical values here for example in place of w^1 I am just going to put 0.64 and y^1 is 10.4, then w^2 is 0.16 and y^2 is 12.6, then w^3 is 0.16 and y^3 is 16.4, then comes w^4 is 0.04 and your y^4 is 18.6 divided by the sum of all w values then I will be able to find out what should be the output that is nothing, but your 12.04.

So, using this particular the rule, using this particular the technique, that is Takagi and Sugeno’s approach, we are able to find out what should be the control action or the output for a set of inputs. Now, as I told that if I want to find out the output, the first thing will have to know is we have to know the coefficients and as I have already mentioned to determine the coefficient we take the help of some optimization tools, and that is why, this particular approach is able to provide very accurate prediction.

Now, we have already discussed like if we just plot say interpretability versus accuracy. So, supposing that here I am writing interpretability and here I am writing the accuracy.

Now, till now actually we have discussed two very popular approaches of fuzzy reasoning tool, and one is the Mamdani approach and another is actually Takagi and Sugeno’s approach. Now, for the Mamdani approach, the interpretability will be high, but the accuracy is low. So, might be I am here, this could be your the Mamdani approach. So, this is the Mamdani approach, and the Takagi and Sugeno’s approach the accuracy is actually high, but interpretability is not good. So, might be that particular point is here.

So, this is nothing, but Takagi and Sugeno’s approach. So, this is actually the real situation of these 2 algorithms, but what do you want is we want one algorithm, which will be able to do the prediction accurately and at the same time, its interpretability should be good. And, that is why, we try to find out one algorithm and that algorithm may take the position somewhat here, and that is it will give a some sort of your the better prediction, but not at the cost of computational complexity. So, it should be computationally tractable.

It should be interpretable and at the same time the accuracy should be good. So, might be we are going to search for the algorithm which could be here and that will provide a very good combination of these particular accuracy and interpretability. That means, my expected fuzzy reasoning tool should be such that now, in fact, like if I just read the rule, some control action should be understandable and we should be able to understand the output of a particular rule and at the same time, the accuracy should be good. And, a lot of studies have been made how to find out an algorithm for this fuzzy reasoning tool, which will provide both accuracy as well as interpretability.

Thank you.