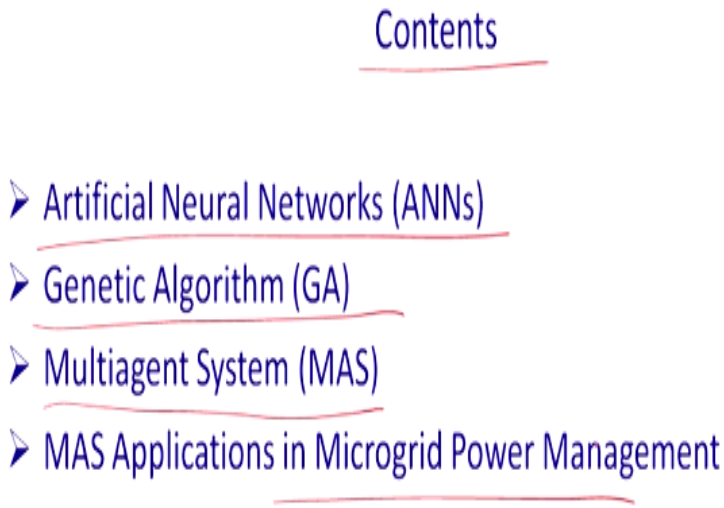
DC Microgrid and Control System
Prof. Avik Bhattacharya
Department of Electrical Engineering
Indian Institute of Technology – Roorkee

Lecture - 23
Intelligent Microgrid Operation and Control (Continued)

Welcome to our lectures on the DC Microgrid and the Control System. Today we shall continue with the intelligent control intel intelligent microgrid operation and control. We have our talked about fuzzy logic controller.

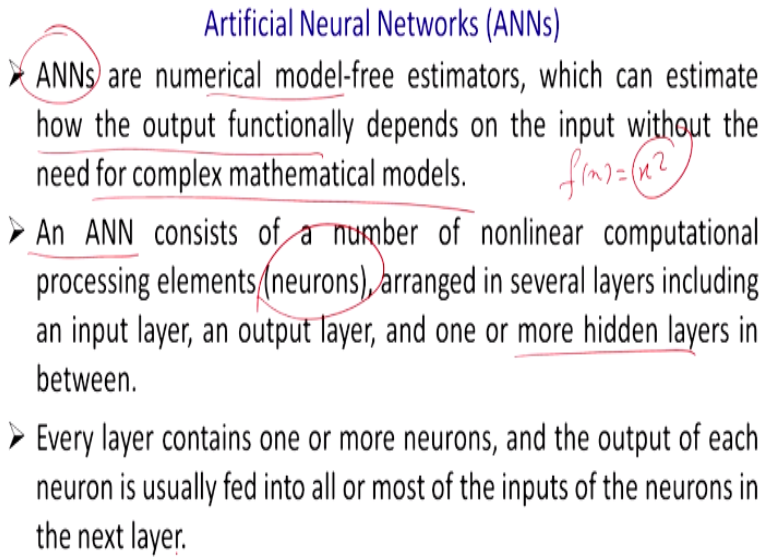
(Refer Slide Time: 00:51)



Today we shall talk about another intelligent control, for namely ANN and GN other entities. So, broadly today this will be our content of our discussions. There is artificial neural networks, artificial neural networks students are requested to brush up their basics because we cannot take all the whole courses on INN here which just can touch, but we can show that how it can be applied in case of that microgrid AC as well as DC.

Now the genetic algorithm, same thing, so it is a heuristic function for any optimizations, genetic algorithm can be used, thereafter multiagent system that is MAS, and the multiagent systems applications in the microgrid because you have a different stakeholder that is they have different kind of load, you got a different kind of source, you got an energy storage element; for this reason that exactly fits to the multiagent system, and for this reason, we shall show it how to deal with it in our applications.

(Refer Slide Time: 02:09)



Now, what is the ANN, ANN is basically inspirited from human brain applications or the animal brain applications. ANNs are of course it is artificial neural network, the full form is that. A numerical model-free estimators, when you know that input and output can be mapped. So, you know how our brain works. It is very difficult to understand because you are trying to understand the brain, but some portion of whatever function it is known to us, we try to emulate the same kind of application artificially and thus it becomes artificial neural network.

So we say that ANN are numerical mode-free estimators because we don't know the moral of our brain, for this reason, but we know the input and output and we know that it works, and thus same thing has been inspired. So numerical model-free estimators, which can estimate how output functionally depends on the input without the need for that complex mathematical model.

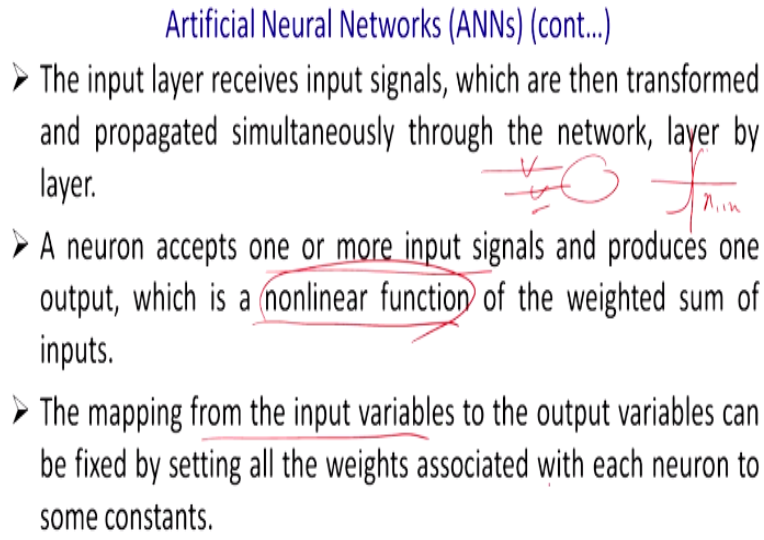
You know if you are putting fx = x square, you can compute any data. But, you know you have details and you want to interpolate some data in between, where a different method of the interpolations that is a mathematical model and thus you got an error, but for example, our populations of India is counted every 10 years, if I wanted to know what is the population today, then we require to interpolate the data.

So for this reason, you know this interpolations can be a mathematical model, but instead of the mathematical model when you have a large number of training site, it can constitute the input output relations without establishing explicitly the mathematical one, that is the

advantage of it. So, you don't want to know what exactly that mathematics and that is it works. An ANN consists of a number of nonlinear computational processing elements, these are inspired from the human brain, for this reason, it is called neurons, arranged in several layer including input layer.

So, there will several layer, output layer and in between there can be some layer, this is called hidden layers and the in between. So, constitute of the ANN can have input, output, if it maps fine, otherwise if it cannot map we required to put it some layer in between. Every layer contains one or more neurons, neuron are the functional block of it, generality it is weight or the gain you can think of the op-amp, then it is a gain, and the output of the each neuron usually fed all for the most of the input of the neurons in the next layer. So, this is constitute of layer, we come to the constructions of the neuron little bit later, maybe the next slides.

(Refer Slide Time: 05:23)



The input layer receives the input signals which are then transformed and propagated simultaneously to the network layer, through the network layer by layer. The neuron accepts one or more inputs, it can op-amp, can take the more. It is something like you visualize this thing as a gain element by op-amp. The op-amp can take multiple input with the scaling and first artificial network was based on (( )) (06:05) and it was based on op-amp.

So, one or more input signal produces one or more output and so you require a little bit of scaling, that is something you don't know because generally you have a single output in case of the op-amp and which is a linear function, which is that is something you have to think about it because you know that gains s are linear, so nonlinear functions of the weighted sum

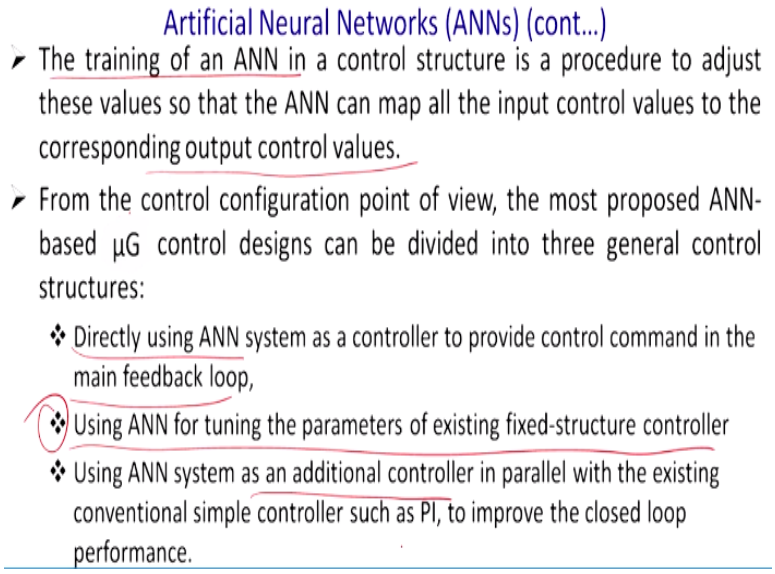
of the inputs. So, it will do some kind of, so you got a, you have inputs x1, x2, x3 and ultimately it will have some kind of numerical functions.

So, that numerical function not necessarily linear, this one x1, x2, x3, ultimately x is a function, x is a input, and y will be the, y will be the, y or y1, y2 can be the outputs and that may have some kind of mathematical relation or may not have a some kind of mathematical relation that can be explicitly put it into the terms for mathematical functions. The mapping from the input variables to the output variables can be fixed by setting all a weight associated with the each neuron to some constant.

So, we require to put some gain or some constant and so that is what we go by. Now, one important aspect is that we require to train our ANN. So, someone has to tell I am Dr. Avik Bhattacharya, otherwise it will not work. So, then you not recognize me as Dr. Avik Bhattacharya. So for this reason, we required to have a training set. So, then what happens you know for the given input, for given out, and accordingly this weights of the ANN are being tuned to match input with output.

Once training is complete, then you give that test status where you have not given that data for the training, then also you know that your input and output. Once it is working extremely fine within the range of acceptance of error, then this you put this model for your prediction purpose, that is the way it works.

(Refer Slide Time: 08:45)

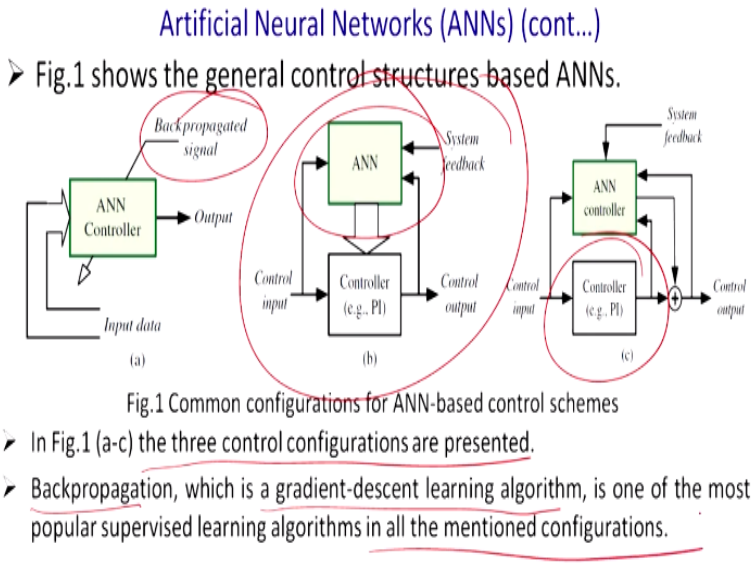


So, the training of an ANN in a control structure is a procedure to adjust these values, so that ANN can map all the input control values to the corresponding output control values, that is the way it start working. From the control configuration point of view, the most proposed in ANN based on microgrid control design can be divided into the 3 general control structures. So, one is one can be rather directly using ANN system as a controller to provide the control command in the main feedback loop, which was we have shown previously, it is basically based on PI controller.

Using ANN for tuning the parameters of existing fixed structure controller, that is also one of the way of employing ANN in micro grid application that is using ANN for tuning parameters for existing fixed structural controller, that is the KPKI of the controller can be tuned very well with the help of ANN, that is the auxiliary application and KP and this PI controller itself can be replaced by the ANN controller, that is the first one directly using ANN

Using ANN system as an additional controller, which we have seen in case of the p in case of the fuzzy logic controller, it is the same thing, additional controller in in parallel with the existing conventional simple controllers such as PI or PID to improve the closed loop performance, here what happens it can reduce overshoot, it can give you a faster settling time, these are the few advantage of this model.

(Refer Slide Time: 10:45)



So, this figure 1 shows the general control structure based on the ANN. Now, you have input data and you have to back propagate signal, that is called the back propagation algorithm. There are many algorithm to tune it nowadays , simplest was the back propagation algorithm

and you got an output. So, you have to train accordingly when input will match the output, the reverse and thus you change the weight like that from output to input.

So, thereafter you can have a second application, that is, you know you already have a PI controller and that is working fine, but this tuning of the PI controller, you can help take the help of the Nikolsky and all those methods, gain margin, phase margin, from there, you can optimally design, but then also you find that taking also because you are approximated the system as a linear system, performance is not satisfactory.

Then, you require to tune it further and their input output model can help and you can help to tune the PI controller. There is a plenty of papers for tuning PI controller by ANN, that application can be directly applied in case of the DC microgrid because it applies the PI control. Another aspect of course, you have a PI controller and you have designed with the two considerations, still its performance can be improved,

Then what you do here, essentially you put extra ANN controller that will essentially take an input and will add up some extra value of the Y that will be the output of the PI controller and that will give an optimal response of the total system in terms of the overall performance of the system because you know that essentially when you design a PI controller, everything zeroed down by the model retention technique, or even neglect the insignificant pole and zeros and all those things and ultimately you required to bring down the system to the second order system.

Then only your linear control system works fine, and in that approximation process you know you have neglected so many thing and ultimately you came down the value of kp and ki with a due mathematical consideration or if it is PI (( )) (13:24). Then what happened to take those nonlinearity which we have neglected in designing the PI has to be considered to the optimal design of the PI performance and that is this ANN doesn't have any mathematical model and you add with that and this this add on will definitely make the system to performance give you better performance.

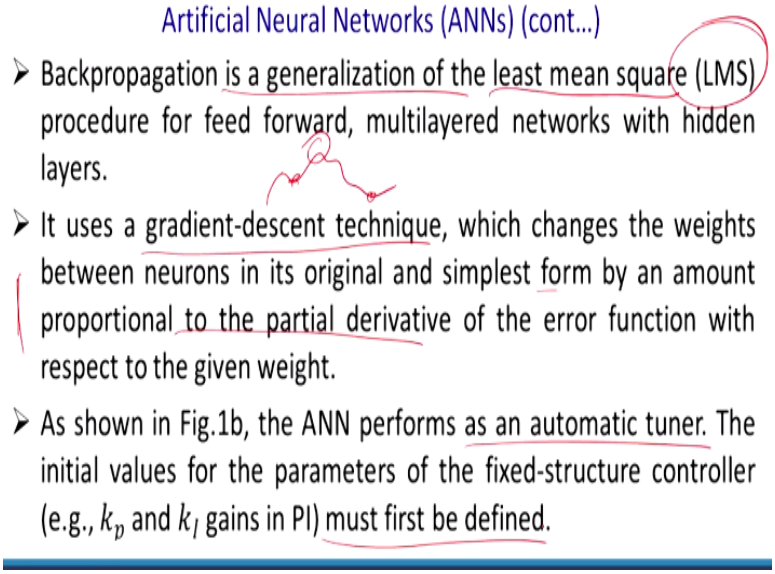
That is the whole idea of using this kind of ANN. So, in the figure 1c, the three controller are significant controller configuration represented and it has been tuned to the backpropagation algorithm, this is simply an algorithm, there are so many algorithm which we are researching

on the ANN, you can find so many papers on different kind of algorithm, there is ILS algorithm, recu mean square algorithm. So, it has own advantage.

So, if generally it involves the inverse of metrics, if you can inverse, if you can avoid the operation of tuning ANN with without help of the inversion of the matrix, your tuning will be faster. So, while you are tuning online, sometime it is required, adaptive, it is called adaptive neuron that is called Adaline. So, they had a different kind of algorithm if you wish to work on the ANN, and so, you are required to read the basic papers and the basic applications of the ANN.

So, backpropagation, which gradient-descent learning algorithm, so it \has a problem that it may converts to the local media minimum, is a gradient-descent learning algorithm in one of the most popular supervised learning algorithm in all the mentioned configuration.

(Refer Slide Time: 15:20)



Backpropagation algorithm with generalized as a generally it will reduce the means square error, so for this reason, it is called the least mean square or LMS procedure for the feed forward, multilayer network with the hidden layers. So that is called the LMS algorithm and it uses the gradient-descent technique. So, you have a different kind of al multi , this is a card let us say, but it has a problem that you know it will start from here.

It will be converging here, it may not converge here, and it may converge to the local optimal point whether you are seeking the maximum or minimum. But if it is here, if you are finding time to find the global maximum, it will give you a global maximum since it is closer, so

which changes with the weight between the neurons in its original and the simplest form by an amount of the proportional to the partial derivative of the error function with respect to the given weight.

So, this is the way it will work and it will try to converge the nearest optimal point from the point of starting. So this starting point is very important. If it the starting point, it may not reach the global optimal point. As shown in the figure 1b, please go back, so this is the figure So, here you have an ANN and you have your tuning this parameters of the PI controller by ANN. ANN perform as an automatic tuner, is basically the tuner of the PI controller.

The initial values of the parameters of the fixed strictures of the controller kp and ki of the PI must be defined and it should be close to this optimal point. Otherwise, it will lead to some other point and you get a worse result. So, for this reason, we required to be very cautious because you have to bring very close to this optimal point, then only it can bring it to your destination.

(Refer Slide Time: 17:47)

Artificial Neural Networks (ANNs) (cont...)

- The ANN collects information about the system response and recommends adjustments to be made to the controller gains.
- This is an iterative procedure until the fastest possible critical damping for the controlled system is achieved.
- The main components of the ANN tuner include a response recognition unit to monitor the controlled response and extract knowledge about the performance of the current controller gain setting and an embedded unit to suggest suitable changes to be made to the controller gains.

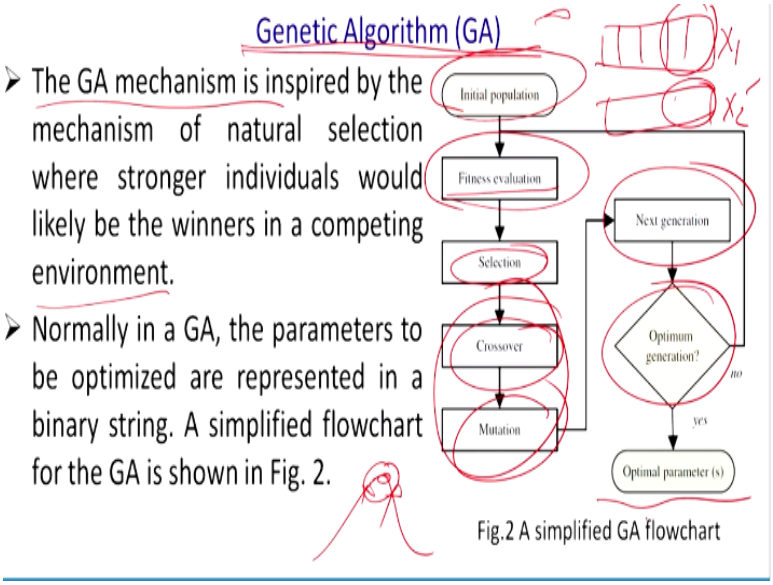
So, the ANN collects information about the system response and rick and recommends adjustment, that is the weight generally, to be made in case of the PI controller gains, that is the kp and kp or if it is a PID controller, then kp and kd. This is iterative procedure until the fastest possible critical damping for the control system is achieved. So you want ultimately this we would have a knowledge of the time response of the second order system and we want that the system should be just below critically damped.

So that kind of response, overshoot is very low, and it has got the faster response. So, all those optimality has to be considered and accordingly PI to be designed and that tuning can give you that optimality. The main component of the PI sorry the ANN tuner includes the responsible response recognition unit to monitor the control response and extract the knowledge about the performance of the current controller gain setting and embedded unit to suggest suitable change to be made into the controller gain.

So, that is something we require to know very well. So, that include the response that is basically the time response of this PI controller, that behavior should be known to you and accordingly you start tuning on online mostly and say please remember that this PI controller if it is implied into the current loop, it is extremely, it has to be extremely fast and for this reason, we have to have a performance of the current controller gain has to be tuned for the optimal purpose.

Now, as we have discussed the ANN, ANN has a problem of this backpropagation algorithm, this is gradient based, and thus it cannot find the global minimum and it may have a tendency to localize or it may depend highly it will be depending on the initial value. If the initial value is close to the global maxima or minima, then it converts to it; if it is not, it will take you for the farther point and to overcome this descent, this difficulty, we have the heuristic method, many heuristic method.

(Refer Slide Time: 20:31)



This is one of the heuristic popular heuristic method that is called the genetic algorithm and that that also inspired form our human evolution system, where it is been assumed that we are

smarter than our parents, because the process of evolution, one may doubt it anyway, that is a different issue, but we have a generating algorithm. So, you got initial populations and that these are called generally the genes, then you will test for this evolution, that is a Darwinian principle, survival of fittest, whomsoever fits best, then tho those genes will be asked to go to the next stage and that is called a fitness evaluations.

So, in in terms of the Darwinian evolution theory, it is survival of fittest, are the process of the natural selections. So, that is called the fitness revolution. So, thus you got selections of the genes and in between, they will have a crossover and mutations. So, they will like you know, you can please go biology which you have studied in our school days, if not studied biology higher in college days, so we have a mitosis.

In mitosis cell divisions, generally that is a different kind of cell division and they have a meiosis cell division, mostly this is for the our reproductive system, and there we will have a crossover and the mutations. So this crossover and mutation, some portions of the of the chromosome will and let us say this one is x1 and this one is x2, so some this portion maybe may come here.

So, it is actually the hybrid x2 and thus you have a crossover and you can give a crossover ratio in the programming, so you can have a 20% gene from where it will come over here and vice versa. So, that is a crossover. Then you give a probabilistic, in program you will give a probabilistic thing, so among these genes, you give some portion of the mutation. So, some of the genes in between the chromosome will appropriately change and thus new value will come, and in that way, you will actually have next generations.

In next generations, again you will go to the same fitness evaluation, selections, crossover, mutation, and this loop will continue till you got an error that is prescribed less than the value you wanted, so that is the optimal generations point and thus you got an optimal power. It has many advantages, though this tuning is slow, because it takes a lot of maturity bloom, and it depends on the crossover and mutations ratio how you are working on it, but it generally you know picks up the global minima, but there is a problem you know.

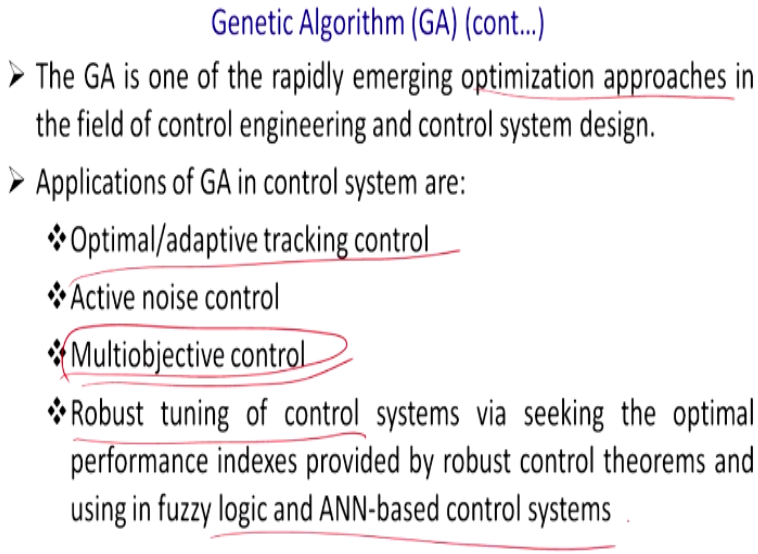
So generally if it is a global minima and if it is here, it may jumps to the crossover at this point. Again, you got a crossover, it may come to this point and it may actually not reach here

and ultimately continue to do the oscillation. So, this is one of the major disadvantages of it. So, but once you are working on it, you will be more familiar to actually damp out these oscillations also. So, let us talk about the GA here. The GA mechanism is inspired by the mechanism of the natural selections, where stronger individual will be likely to likely to be winners of the competing environment.

Please understand that evaluation function to be freed accordingly. If you set a particular fitness function to taking a log from the forest, of course, elephant will win the race, and definitely if you set another milestone where to taking a log into the to the foothills of some mountains, then awkward donkey will win the race. So for the purpose of your requirement, you will feed the evaluations, feeds our evolution, and thus different chromosome will satisfy the different need and ultimately you will have an optimal need.

The normally the GA, in a GA, the parameters to the to be optimized are represented in a binary string. These are called mostly chromosomes, it is inspired from the biological system. A simplified flowchart of the GA has been shown in this figure. Now, so these are the few entities that you require to familiar while writing program in GA.

(Refer Slide Time: 26:22)



The cost function, which determines the optimization problem, represents the main link between the problem at hand and the GA unit. To start optimizations, the GA users the randomly produced initial solution created by the random number generator, that that that entity is there in the MATLAB. So, you can generate the MATLAB, in a MATLAB code that

are random numbers. There are basically the 3 genetic operator used to produce a new generation.

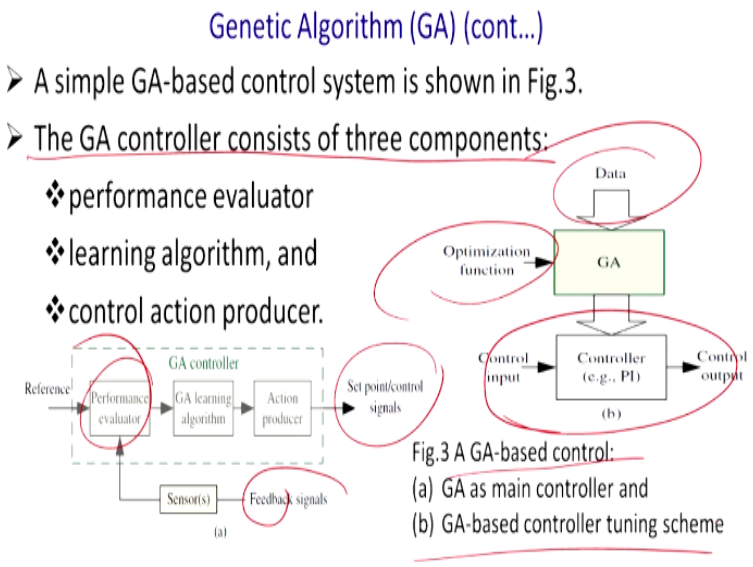
So, you require to some kind of samples and that required to be tested for the fitness and that produces a new generation. These operators these operators are the selections, crossover, and mutations. First to go for the selections, and they come in to the next stage, then you make this then whatever genes are available those who are already fitted you crossover among themselves, thereafter you give a probability of mutations and then again you generate the next generations and test it.

The GA employs operations operator to converge at a global optimal point, that is one of the main features of the GA, and after randomly generating the initial populations, GA uses the genetic operator to achieve the new set of solution in each iteration, and thus, the GA is one of the rapidly emerging optimization approaches in the field of the control generating algorithm system design and application of the GA control systems are definitely the optimal and adaptive control, active noise control, and multiobjective control.

Because you want that multi objectivity, your overshoot should be low, your response should be fast. So, these are conflicting features we require to incorporate in your system and thus you want that some extent optimal solution, like you have money you want to have a liquidity to be high, your return should be high, rates should be low. These are all conflicting. So for this reason, it is called a multiobjective. So, we can have solutions, optimal solution based on our requirement.

Thus robust tuning of the control systems via seeking the optimal performance indexes provided by the robust control theorem and using the fuzzy logic and ANN-based control system. So, this is also where you can apply your GA-based control system.

(Refer Slide Time: 29:29)



So, a simple GA-based control system is shown in the figure 3. You can see that the GA controller consists of the 3 components; the performance evaluator, learning algorithm, and the control procedure. So, you got a reference. So, this is a performance evaluator, since it is a natural selection. Thereafter GA learning algorithm, so you will tune the GA. So, and thereafter, per action producer that is a point of signals and you have feedbacks and accordingly your performance evaluator will change that 5% of the overshoot you apply initially for the fitness.

Gradually you take that actually, it is the 2% of the overshoot, same settling time, first you take some, maybe 100 millisecond, then you come down to the 50 millisecond. So, gradually you may from the sensors, you can actually also fine tune your evaluations, so that is also possible and thus you get a set point control signals. Because if you set a very stringent evaluation and you may not get the very good number of samples to be survived.

For this reason, you have to gradually you have to change the performance evaluator to get it to the right kind of sample, and thus you got a data, your optimization function in GA and you got a control output based on GA. So figure 3 describes the GA-based control, and one is GA main controller, and this is basica, this mechanism is the second one that they have shown in figure 1b, that is turning of the PI controller by GA. So this is a way to you can use to control, this is the way you can tune the parameters of the PI controller by GA.

These are many paper based on that PI controller based on GA for optional system design. Thank you for your attention. We shall continue with our sub portion of GA as well as the multiobjective function in our next class.