Course Name: INTELLIGENT FEEDBACK AND CONTROL

Professor Name: Leena Vachhani

Department Name: Multi-disciplinary primarily for Mechanical, Electrical, Aerospace and Chemical engineering streams

Institute Name: Indian Institute of Technology Bombay

Week - 04

Lecture - 23

Hello, in this video we will look into control using reinforcement learning. In order to design controller using the reinforcement learning, we'll first understand what is the input-output requirement of the reinforcement learning so that we can make the controller case as a reinforcement learning case. So in order to understand the reinforcement learning framework, it is dynamic programming with rewards and penalties. Rewards and penalties, I'll explain a little later, but let's understand what this particular observation and action space is when this is a reinforcement learning agent. The input to the agent is observations and the output is actions.

So first thing is these are certain key components. The first is to describe what is my environment. The environment, as we had looked into earlier, was that in our case, it is the system or a process under the actual environment. And this describes my reinforcement learning environment. All right.

So environment is an actual environment or simulated. We can consider as of now as a simulated environment which encompasses system completely. Now, in this case, there's an agent, which is more or less like a computing system, which is taking care of the algorithm description of the RL agent. Now, here comes the states of the environment. What is the state?

It is different from the state of the system or a process. The way we used to describe the system of system states as X dot and the system equations at AX plus BU and Y is equal

to CX. In this, our X used to be the state of the system. Correct. Now, this system, this particular state is the state of the environment.

Which encompasses a set of observations, set of actions or something else or some internal states of the system, some internal state of the environment or some interaction between system and the actual environment and so on and so forth. So the environment state, we will have to describe what it is. But at this moment, it is important to understand that it is different from the state of the system that we call upon. Something similar, we have actions. Now, actions, you can call it as the control inputs given to the system as a basic understanding of the system.

So, we have this particular control inputs given to the system, whereas in the RL agent case, this terminology is more or less said as action. Now, action directly mapping to input may or may not be there. All right. Because this particular action is given to the environment. And this environment consists of the system under the actual environment.

So we'll have to figure out what this action space we are talking about, depending upon the control system objective. Something similar is about observations. These observations we can, for the system terminology wise, these observations were the measurements taken from the sensors or the output that we are getting as Y is equal to CX form. But these observations, again, are out of the environment. All right.

Are the output output signals that are being given to the agent. Now, these observations, again, comprises of could be the measurements taken, could be some outputs or the state of the environment. All right. Which may which when we start discussing about certain examples, it will be easier for you to to understand. So what we are talking in terms of the observation here and the actions being taken by the agent.

Once we have the set of observations and based on that certain actions have been taken. So for example, in this particular environment, I had one observation for which I had taken a particular action. I have set of observations based on which we have the set of actions being taken. Now this set of actions that have been taken in sequence, are these taking care of the control objective that we have set? Who will tell us?

That particular thing which is coming up from the policy is taken care by the rewards and the penalties. All right. In all, what we had done in the earlier case, we'll take a look at it and try to put it in the RL framework so that it is easier to understand what was the database approach and what is the RL agent based approach that we are considering here. So more or less to understand in a very, very simplistic way, the policy is something that will decide the mapping between the observations and the actions given a particular state or a given state of the environment or the sequence of the states of the environment. All right.

Let's take certain illustrations to understand what is observation, what is action in the control terminology that we have been looking at. Let's take a simple example of a first order system. The transfer function, we can say it as a two parameter model, which is K by S plus A, where the two parameters are K and A. In this case, we are considering that the model of the system is first order system and we are aware about this particular one. Okay, we'll relax that particular assumption and we'll see what happens.

Now, when I represent this in terms of the state space form, then my system representation becomes something like this. My A matrix is minus A, B matrix is just the unity, C matrix is the gain of the system, and D is zero in this case. So, this particular transfer function when we represent in the state space form looks like this. Fair enough. But at the same time, this particular form gives me the idea of the state of the system.

All right. Mind it. We are talking about state of the system. All right. Not the state that the RL agent considers.

So here's the change in the terminologies. And that's what I am bringing out here. All right, so in the previous case for the data-driven PID control, what we were considering as we were having some system for which the model is unknown. And we were designing controller with the help of PID control. So we were looking into tuning KP, KI and KD, which are the proportional, integral and derivative gains.

All right. So what we were doing was we were keeping some information vector. We described an information vector and that vector will give me the output vector, which is KP, KI and KD. All right. So this is how depending upon certain state of the system or

state of the environment, we were looking into into describing what should be my KP, KI and KD for that particular state for a particular control objective, which was rise time, lowering the reducing the rise time objective.

All right. So we are also aware about the input-output terminology. So, we have this particular input given by the signal U and the output of the system is given by the variable Y here. All right. So what we have is when the transfer function is your first order transfer function, which we just spoke about.

And we are giving the input the set point as some step input. Then we know that the output looks like. With respect to time, the output may be something, if I have a PID controller sitting there, then the output may be something like the decaying sinusoid or exponential or any other form that we have already studied about, depending upon the different KP, KI and KD values. All right. So we are not talking about this.

All right. For this particular output generation, we have the input U given as the opposite of this. It is easier to draw it in this case. For the decaying sinusoid case, my input given control input given was this. Something similar we have for the exponential rise case, we had the input given, control input given something like this.

We know that if it is proportional, only a proportional controller, then we'll have a steady state error. If it's a KP and KI terms both existing, then steady state error can be reduced. So different values of KP, KI, KD will give you a different output curve. All right, so when we were talking about data-driven PID control, when the cases, when the plant model is unavailable. In these drawings, in these plots, we were considering that the system is first-order system and we were able to do it.

But for example, if I do not know the system model and I'm designing KP, KI and KD for it, then what happens? Then I will have to go with a data driven method where what we were doing was we were considering some samples of U of K, U of K minus 1, U of K minus 2 and some samples of Y of K minus 2. Y of K and Y of K minus one and Y of K and some previous history of Y and some previous history of U we were considering. In order to call it as phi of K, which was my information vector. Now, this is what we are going to call it as the observation in the RL terminology.

This is one example of observation, all right? We can call it as the state of the environment now. Now, how am I describing the state? It depends upon what is the information vector set, the elements of the information vector that I have considered. So we have this controller sitting here, and this was my entire environment.

Of course, the environment may be having also disturbances acting on the plant. So am I modeling the disturbances? Let's not talk about that right now. So my output was my observation would depend upon what is my state and something else depending upon, let's say, what is my YSP as well, all right? And the output vector or the action was, can be given by, okay, I have this KP, KI, and KD.

So given this state of the environment, what is my action? If I'm able to come up with a proper mapping between the observation and action then I have a policy available, now how will I come up with this particular policy which is the mapping from observation to action is what the RL algorithm does, now if this particular policy is to be designed then I need to have a proper way of understanding how am I going to create this policy from the given data available. So, I will have different observations available, observation 1, observation 2, observation 3, which I can call it as, I can map it to the state of the environment S1, S2. I can derive this. What is my state?

I do not know right now because my control objective in this case is not set. We will take a complete example to understand this later. All right. So now this particular state at this moment, of course, what we have done here, our observations or the state are finite. All right.

Need to be finite in order to describe this policy. We have first discretized it. Fair enough. Now, even though we have discretized it, U of K can take any value between, say, minus 10 to plus 10. Unless we describe this particular minus 10 to plus 10 in again quantized levels, we'll not be able to say that my observation space is finite.

So this observation space and the action space, something similar with action space, my KP can say can take values from 0 to 2. KI can take some values from 0 to 2. So, 2.1 minus 0.1 to 0.1 say and sorry usually these gain values are 0. So, we will consider 0 to this and KD can take values from 0 to again some 1 say let us say 1. All right.

So now I have this particular constraints given to the KP, KI, and KD values, because I know that there is no point in considering a very large values of KP and something similar to KI, and KD. But even then, there is there are infinite values of KP that it can consider. So, then my action space is very large. How will I come up with a policy which will describe, whether the action that has been taken given a particular state was a good action or a bad action. So, if I am at state S1 and I have taken action A1, which is what?

Some values of KP1, KPI1 and KD1. So, this way, we'll have to limit our action space or cut down it to saying that, okay, my action space is finite. So, then I will need required to quantize this Kp, Ki, and Kd. Now, with this quantized steps, my action space is limited, is finite, observation space is finite, then I will be able to describe a reward function which will say, okay, when I was at state S1, I took A1, which led me to reach to state S2 and say SK I have reached, all right? So, at K equal to 1, my state was 1 and I have taken this particular sequence of steps, all right?

So, A of K, then A of A of 1, then A of 2, and so on, which led me to this particular sequence for K horizon, K's discrete steps. So within that K's discrete step, is my control objective getting satisfied or not, is what will tell me, okay, my reward is, is it a reward or a penalty? This way, we will be able to describe the control requirement in terms of the reinforcement learning environment. It is similar to what we did in case of the data-driven PID control. In case of the data-driven PID control, we figured out putting the KP, KI, KD values depending upon what my information vector would be.

Now in case of the reinforcement learning, we are coming up with a mapping between the observation and the action and learning this particular policy based on a training data available to us. All right. So putting things together, if I consider in a different setting where I am not having the understanding of the plant, okay? I model of understanding of the plant as in I do not know the model of the plant. I do not have the representation of a plant or a system or a process that we talk about.

At the same time, I do not want to freeze the form of the controller, which is PID. Then also, I will be able to put things in terms of the RL framework. Given a particular control objective, I can consider, okay, my environment is giving me certain observations, which

are going to the RL agent, which is making sure that there's some policy coming up, which is being learned. Of course, we'll have a training data as well as a test data. So with the help of the training data, this policy is coming up, which is giving me output as the actions to be given to the environment.

These actions will then drive the input to the system or a process. And these inputs are my U vector and these are my output vector Y. So then we'll be able to figure out whether my state trajectory that now I'm talking about state of the system. Right. I have the state system state trajectory is optimal in some way or not.

Now, when we are saying trajectory, this is timestamped. Or in case of the discrete time K, this is we are talking about state 1 to going up to state K. So, this particular trajectory is a desired trajectory or not. This is what we would like to optimize it and this forms into the RL learning framework now. Now what we are, if I compare the classical control design methods or the controller or the model-based or what we had here was we, till this particular week, we were looking into model-based control.

Now it is model-free control, then also we look into considering designing the controller through the help of the RL agent such that my system trajectory is the one that I am looking at as a desired trajectory. So it's just not the output, which is the set point value or whatnot, but how am I approaching the output can also be designed and can also be optimized with the help of the learning agents now. All right. Let's try understanding further into what comes under the training phase. We have two things now.

One is actor and a critic. Now. I'm talking about what comes under the RL algorithm typically. So this is actor and a critic way because I have to design a policy, but at the same time, I do not know what the values are going to be. So that is something needs to be decided.

Now, when I'm considering the actor responsibility, I will consider that It is coming up with some kind of a policy, which is modeled using an actor network. So each of this actor and critic is a network, neural network, I can consider. Now, this particular actor network is responsible for giving me a policy. And it will also do the maximizing this

particular expected return by optimizing this policy. Because I will have various different data set, in the data set we have observations and actions being paired up.

Now when I am doing the maximizing these particular returns, now what are these particular returns? This is being decided by the reward policy. So we will have to decide what is my reward. Now, when I am calculating going from a particular state 1 to reaching to the state S of K, I can follow a particular trajectory. But in a second case, I can go from S1 to S of K through some other state.

So, this was my one trajectory. This is my one trajectory 1. This is another trajectory 2. So, starting this initial state is same, final state is same, but the path that I followed to reach or the trajectory that I followed to reach to this particular final state could be different. Now, whether this trajectory is good or that trajectory is good need to be figured out with the help of values associated with it.

Now, this value is evaluated by the critic here. And again, this is going to be done with the help of a network, critic network. And this value function calculation, when I am going from S of 1 to S of 2 in this case, this particular S of 2 would be different from this S of 2 because I have taken certain different actions over there. All right. So that is something will be calculated by the critics.

Now, the total value of this particular trajectory will decide whether it's a good policy that we followed in terms of taking these set of actions A of 1 to A of 2 to A of K minus 1 or this set of actions that I have taken here. So this is how my actor is going to define its policy based on the critic which is giving me the value of a particular trajectory. And we'll explore all kinds of these trajectories depending upon what the data set or the training data that I already have. And this is how then we'll keep up getting a particular policy which is an optimized policy from a reinforcement learning method. We'll take up an example in my next video.

See you then. Thank you.