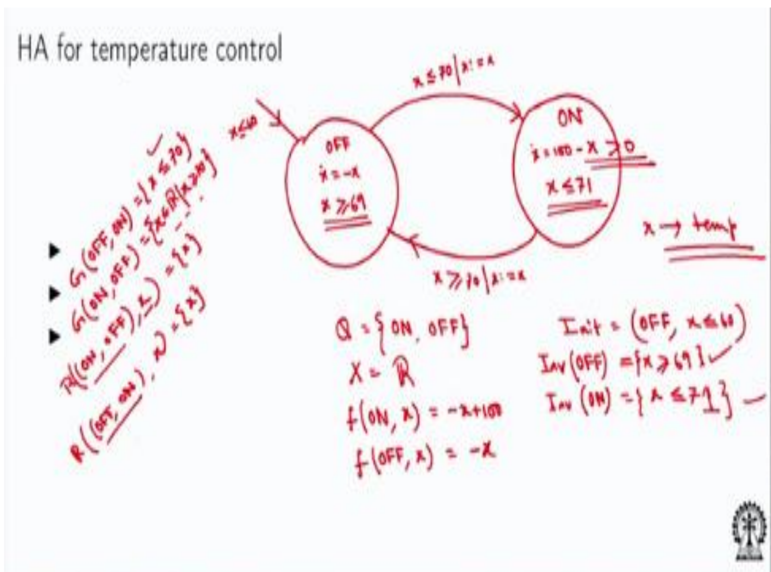
Foundation of Cyber Physical Systems

Prof. Soumyajit Dey
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 31
Hybrid Automata Based Modelling of CPS (Continued)

Hello and welcome back to the lectures on Cyber Physical Systems and their foundations. So, we had been discussing in the last lecture about this temperature controller system.

(Refer Slide Time: 00:39)



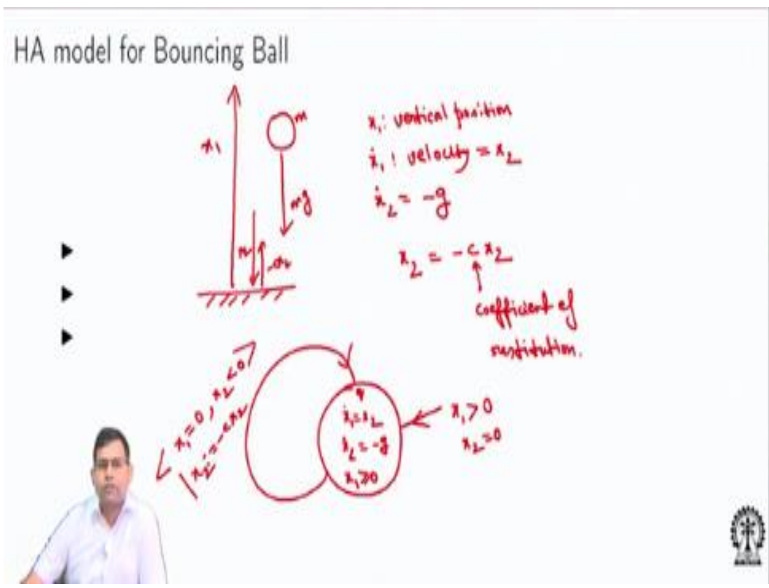
So, I think we left off here we were talking about this automaton. Let us just continue from this point and we try to figure out what are the tuple definitions of this automaton. So, if you can see you have two states here. You have only one variable temp. And that is kind of you I mean modelling the temperature values. So, for X in this case the set of all possible values is a kind of the real set, because temperature reading should be real.

Now your vector fields in each of the locations ON and OFF for this variable X, set of initial states. So, it is off the location and the valuation of the continuous variable is this, you have this set of invariance, invariance for OFF. Now we are the guards and resets. So, the guard while moving from the off state to the on state is given by this x less equals 70 and the guard for ON to OFF state. So, one thing I must say that, this is like a shorthand. Ideally you see these are set.

So, the way you can write is that x is an element from the real set with the constraint that x greater equals 70. So, that is actually how you represent the set and similarly it can be for this like that. And when I am doing the reset. Here, practically this there is no reset but since as part of the definition it has to be a complete one. So, ON to OFF, and OFF to ON in both cases for the variable for this edge for this variable it is just that value only.

Similarly for this edge from OFF to ON for this variable it is x only. So, that is how you have it. So, let us move on with some more interesting examples here.

(Refer Slide Time: 04:52)



Let us take an example from a very practical phenomenon which is let us just consider hybrid automata for a bouncing ball. Now why is a ball bouncing on a hard surface an interesting case. It is a real-world scenario. But what is more important is if you observe the dynamics of a ball in this setting. It has continuous dynamics, when the ball is falling down or moving up from the surface it is under the influence of g it is a continuous dynamic.

And there is also a discrete dynamic, when instantaneously the ball changes direction of motion when it is touching the surface. So, that is why it is a nice mix dynamic system having a continuous dynamic as well as a discrete switch. So, this is your ball of some mass m. Let us say this variable x_1 is denoting the distance from the ground. And when you take the derivative of velocity that is the acceleration it is -g.

And when the ball hits the ground, let us say, I mean, it will have a change of direction of motion. And there will be a change in the velocity also I mean the scalar value also and the speed will be reset using this kind of I mean equation here. So, what will happen this velocity component will change like this. So, if it is hitting the ground with x_2, you will have this. So you are hitting the ground with the velocity x_2 you will bounce back with - cx_2.

So, if it is like this it is - cx_2. Now this c represents what we call as the coefficient of restitution. And this minus sign is there because of course there is a change in direction and velocity being a vector we must have a change of sign here. So, now with this background if we; just draw these bouncing balls automaton, the location q. This is the vector field $\dot{x_1} = x_2, \dot{x_2} = -g$.

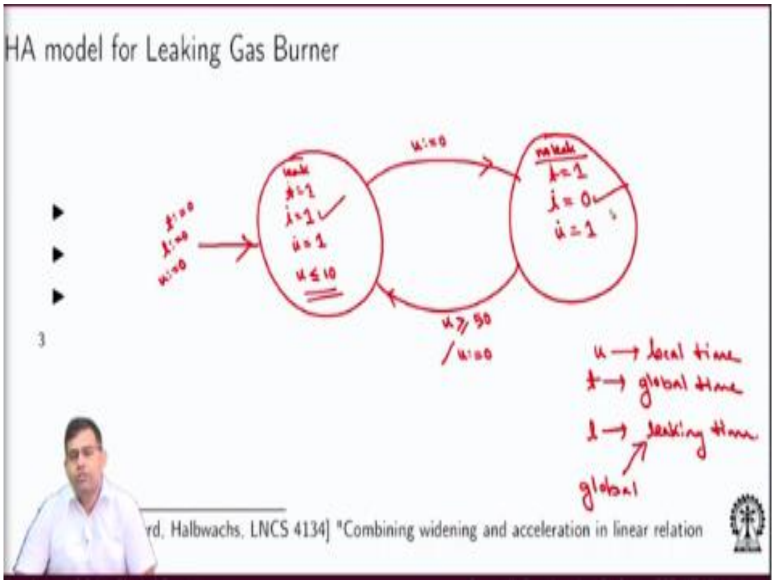
And the invariant is that your height is always this x_1 is always greater than or equal to zero, that is the invariant. Now what about the discrete switch? And of course, we need to also give the initial condition again its same as the invariant that x_1 is something some value greater than zero. And initially of course you do not have a velocity. So, you just drop the ball from some height, so x_1 is greater than zero and $x_2 = 0$ that is the initial condition.

And when this transition is happening, what do you have is that the wall is touching the surface. So, the guard condition must be that x_1 is equal to zero, and x_2 is negative, this must be your guard condition. Let us understand when do I have a change when do I have the discrete switch, I have it when the ball is coming down. So, I mean you have x_2 less than zero. You are coming down and you are just touching the surface that means $x_1 = 0$.

And as a result, what we have is this reset. So, this is the reset condition you can see this is another non-trivial reset condition – cx_2. So, this is the guard, and this is the reset, this is the discrete switch is a single continuous dynamical state. Because whether you go up or down does not matter you are always under the same g with so $\dot{x_2}$ is - g, $\dot{x_1}$is x2 that is your single continuous dynamical step.

So, this is like a hybrid automaton which succinctly represents a bouncing ball. I hope this was a useful example.

(Refer Slide Time: 10:54)



Let us move forward, another simple system this called a leaking gas burner. So, is another classical system it has been taken from this paper where there are much more formal analysis on how leaking gas burner or several other class of hybrid systems can be analyzed using more rigorous techniques which will study sometime later. So, let us first look at the automata. So, what it says is that well that the; I mean the gas burner leaks sometimes.

And whenever the gas burner is leaking that leakage is getting fixed within 10 seconds. And so, whenever the burner leaks the leakage will be fixed within 10 seconds. And once it has been fixed the burner will not leak again for another 50 seconds. That means when the minimum interval between two successive leakages is 50 seconds. In the domain of, I mean just to give you an idea that I mean why do we have this example in the domain of real time systems it is a very standard way to specify the separation of two events.

And they are usually specified like this that between these two events I mean they should happen one after another inside this much time or they should happen at least with this much of separation. You see similar semantics I am giving you. I am saying that if the burner leaks it must be fixed inside some amount of time. And two successive leakages should not happen with an interval less than this.

So, these are standard methods in real time systems of expressing timing constants. And this you are communicating to you using such toy examples of this kind of systems. So, let us just draw this automaton. So, if you look into this automaton, you it is a bit non-trivial in terms of the continuous variable. So, as you can see there are three continuous variables here. And so, what is happening is u is a variable, I mean they are essentially all clock variables.

So, u denotes kind of a local time in a state. t is kind of the global time. Because that is why you see that $\dot{t}$ is always 1. And t, I mean initially all of them when I start that automaton, I am having all of them reset to zero this t, l, and u but after the t is never reset. So, t is kind of the global time. Why do I say u as the local time? Because whenever I change state, I go to low leak or I go to leak state u is reset to zero.

So, that means is hinting towards u is kind of used to figure out, what is the amount of time I am spending in each of these states. That is why when I enter the state, u is always reset to zero. So, u is the local time. And l specifically figures out the amount of time spent in the leakage state. So, that is why you see, when I am in the no leak state there is no derivative, I mean no nontrivial derivative of l.

So, I am just not using the clock. I am using the clock here to figure out what is the time for which there is some leakage. And so, what are my time constants? The time constant is once I move from a leaking state to a no leaking state, I should not have another leakage inside some 50 seconds of time. That is why here I have a guard condition which will not get enabled before u is being 50. So, that is this is how that time constant is being satisfied.

The second one is whenever there is a leakage the leakage must be fixed inside these 10 seconds of time. So, that is captured how that is captured through this invariant you see, that I am saying that well when I reach this state u has been reset. Now here, I can only be up to 10 seconds after that the invariant will be false. So, I cannot be in the leakage state after 10 seconds. The simulation will not proceed.

I must take the jump, so the invariant will be false it will push me out of the state and since there is any no guard to prevent me. So, I will I must make a jump and go to the no leak state. So, this is you see this is one way to kind of the automata is capturing multiple behaviours. I mean this simply means that I can take this since there is no guard here you simply means that I can take this transition anytime.

So, there are various possibilities right, because in different runs the leakage can be fixed in different amounts of time. The constant is it must be fixed inside 10 seconds or 10 units of time. So, before any time before that is a acceptable run of this automaton. So, this is like a non-deterministic behaviour. It is telling that well you can switch any time before this you can switch any time before these 10 units of time. So, that is what is getting captured here.

Now here using this l, we are trying kind of remembering what was the amount of time of the leakage. Because u is kind of, I mean of course l does not have any direct usage in the behaviour definition, but kind if it you need some additional information that if you need to record how much time that leakage happened there, I mean the total leakage time for that l is useful. You may require it if you have a complex automaton if you want to use this information for some kind of property checking you may require it.

So, you see l is never reset. What is reset? The only the local I mean the variable which you are using for local time measurement is reset which is u. l is what we have to I mean l is kind of the global leaking time. That means l is always up counting when the system is in the leaking state. Whenever I am in non-leaking state l is not counting at all. l is not counting at all does not, is not the same as l is starting from 0.

Here l is being stopped at its last value when I come here again l dot becomes one. And l will start from its last recorded value. So, overall l is accumulating the total leakage time, t is accumulating the total time. So, at any point of time these two values together will tell me that how much what is the ratio of leakage time etcetera those kinds of statistics. So, that is the usage of those two clocks and u is the clock which is being used to figure out what is the time that has elapsed. Thanks, with this, we will end this discussion. Thank you.