Foundation of Cyber Physical Systems

Prof. Soumyajit Dey

Department of Computer Science and Engineering

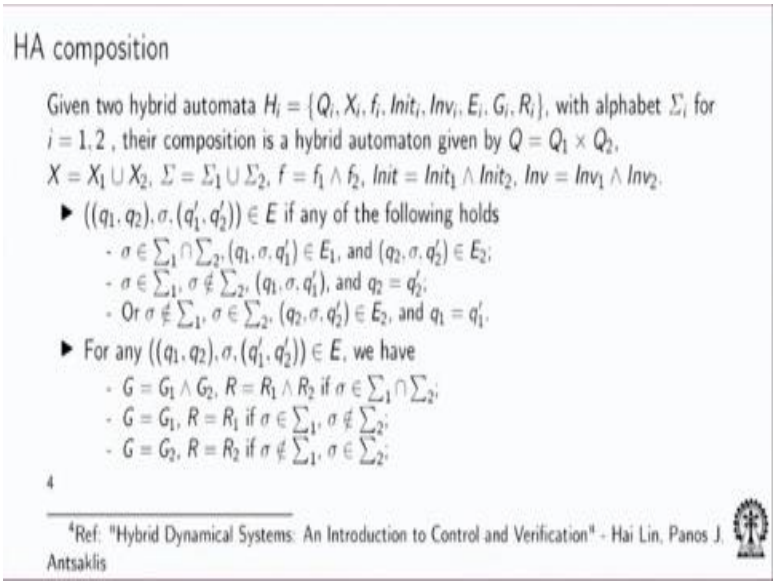
Indian Institute of Technology, Kharagpur

Lecture - 32

Hybrid Automata Based Modeling of CPS (Continued)

Hello and welcome to the course on Foundations of Cyber Physical Systems. So, in the last lecture we have been talking about hybrid automaton and different kinds of models. For example, we took two examples of leaking gas burner, bouncing balls, some of those nice example nice and simple examples.

(Refer Slide Time: 00:45)



So, today we will be starting with a example of I mean first formal definition of how multiple search automatons can be composed. So, let us see what it means. So, suppose you are given this kind of two hybrid automatons and this definition has been taken from this book as you can see. I mean of course it is a standard definition and different books you will find it in different flavours. So, let us say we have hybrid automaton H_1 and H_2 for each one you have Q_1, X_1.

The flow function f 1 the initial state $Init_1$ invariant set in web Inv_1, E_1, G_1, R_1 with the I mean etcetera. And similarly for it H_2. And they have two alphabets I mean σ_1 and σ_2 for their input events. So, now we are considering an automaton where the event transitions will be having guards, resets and also an input event which was not part of our initial definition. But we said that they are they always are there I mean in different classes of automata.

So, when I compose these two automata, we will have a set of states. If you remember our discussion or finite automaton and we compose them for the intersection automaton when we created them. In a very similar way, the composition of is the hybrid automaton which is given like this. It will have a set of locations given by $Q_1 \times Q_2$ where Q_1 and Q_2 are the original set of locations.

And you have this set of real valued valuations which will be just $X_1 \cup X_2$ because the valuations of both of automata will come in your set of symbols will also become a union and then you will have this set of finals this flows, and the flow set will be an intersection we will explain with more why is this is so with some example why the flow has to be an intersection etcetera.

And your initial states will also be I mean an intersection that means you will start with initial state of both of them and your invariance since you are composing the states, the invariant conditions will also get composed. Now what about the transitions? So, this is how we will have the transitions. If you have this, I mean state q_1, q_2 like it is certain a tuple and with an input sigma you are jumping to a state, q_1', q_2', when is this possible?

It is possible under these three conditions. In case this input event is a part of the input alphabet of both the component automatons. And then if in the first automation for the input event sigma and starting from state q_1, I am supposed to jump to some state q_1' and in the second automaton similarly from some q_2 supposed to jump to some state q_2', then here I go to q_1', q_2' but the more

other cases I mean non-obvious cases are suppose the symbol is a part of only one of these input symbol set.

Suppose it is not in the second set that would only mean I will follow simply the transition function of the first automaton. That means I will just follow the q1 σ q_1' this transition. So, from q_1, I will move to q_1' and this q_1 is basically same as q_2' so, I do not I the second element in the tuple will not change. And same thing for the other case that means this sigma is not there in the first alphabet.

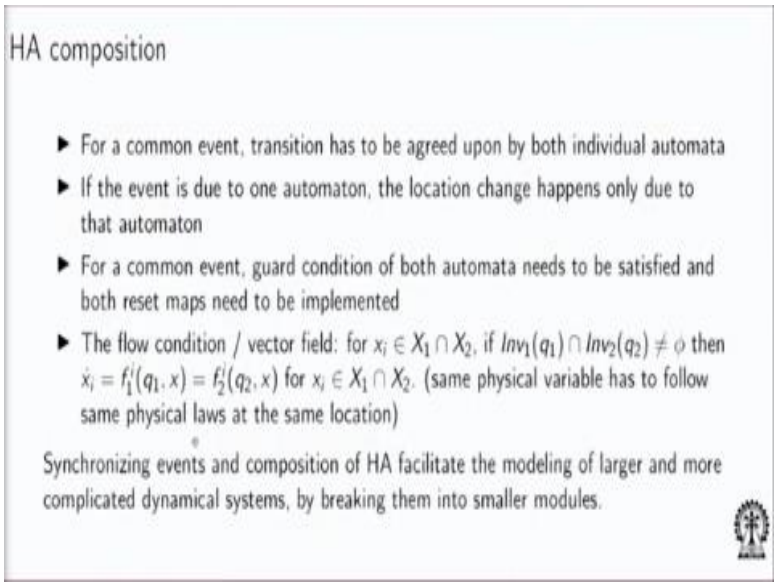
But it is in the alphabet of the second automaton. In this case you will follow essentially the transition function of the second automaton. So, then your q_1 and q_1' are same so you are not changing in the first component of the locations but only the second component of the location is changing from q_2 to q_2' and that is what is being captured here. Now what about the edge, I mean the guards and resets.

So, suppose you have any edge q_1, q_2 to q_1', q_2' you have any edge like this. Now if you have, I mean your guard will be like this so it will just be a combination of G_1 and G_2 where G_1 and G_2 are the individual guards of the transitions from q_1 to q_1' and q_2 to q_2'. Similarly, you have a reset condition as a combination of R_1 and R_2. Now this will happen when your symbol your alpha your element sigma, your input element is a member of both the alphabets σ_1 and σ_2.

If it is the case that sigma is a member of only $\sigma_1$or σ is a member of only σ_2 then it is different. If it is a member of only σ_1 then the guard condition will only be G_1 and the research set will be R_1. That means in principle since its just being recognized by first automaton. So, you are following essentially the transition of the first automation itself and the second component remains same.

Similarly, if it is the component of the first order if it is not a member of the first automaton only the second automaton then you are just following the guard and reset of the second automaton and the first component is remaining the same. That is what it will just mean.

(Refer Slide Time: 06:16)



And now the important thing is that if it is a common event then the transition has to be agreed upon by both the individual automaton, like we just said that if it is sigma y belonging to both the alphabets then the guard and resets must be satisfied by both the automatons. And the other things are what we have just said. Now coming to the flow condition or the vector field. So, if X belongs to these two, I mean you pick up any common variable which is a member of both the automatons.

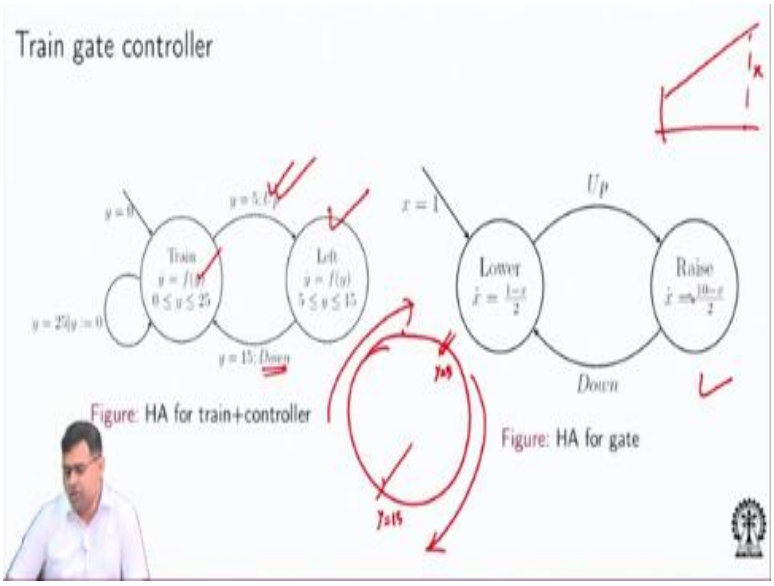
And if the invariant condition, if it satisfies the invariant condition of both the automatons that means you are talking about a variable continuous variable which is a member of both the automatons and it satisfies the invariant condition of the individual locations of both the automaton that means this condition, this intersection is non null. Then what must be happening is that this is the common variable.

So, it cannot at a time obey two physical laws two different physical laws, it has to open only one physical law of evolution. Hence although we have this two flow factor, flow functions f_1 is the

flow function for component automaton one for this variable X for some state q_1. f_2 is the flow function for this automaton two for the same variable for the state q_2. So, when we are having this q_1 q_2 this combined state X_i this variable must have this common flow function.

So, this flow functions must be equal. So, in this way this combination this kind of synchronization events or some such common events and composition of hybrid automaton what is the usefulness. The usefulness is you can combine small automatons or component systems and build a larger system similarly you can model a larger automaton by breaking it into smaller modules and their communication.

(Refer Slide Time: 08:24)



So, let us take an example. So, what we have here is we are considering a train which is kind of moving in a circular track, and from some point y = 0 we are measuring the distance covered. Let this be the distance value is y = 5 and y = 15. So, this is a circular track where the train is moving. When this value of y is between 5 and 15 in this region, we will say that the train is far away from the gate so it has left the gate.

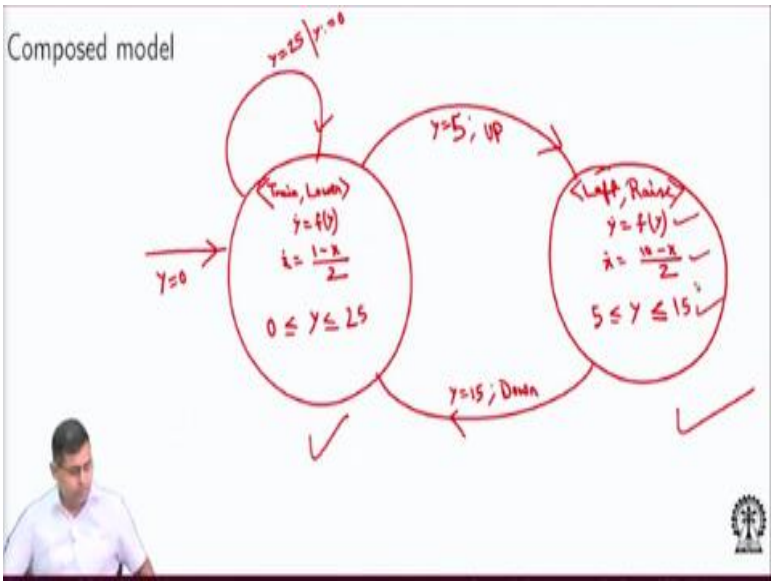
And that is just following the dynamics of the train itself is given by y dot = f y and when it has reached this point y = 15 the gate must be closed. So, this represents I mean the kind of the train

automaton the train and its controller automaton is sending a signal that gate must be closed when y = 15. Well for the train nothing changes it is following the same its own flow function only thing changes is that this state this location in the other location they are differing because this when I am here that means I have sent the signal that you close the gate. And when I am here it means I have sent the signal that you open the gate, the gate is sending an up signal. So, when I am here the only invariant I have is y is between the overall value of 0 to 25 that is the overall track length here and fine, and when the train is I mean kind of out at this point y = 5 so it sends this up signal to the gate.

Now this is my automation for the gate when it gets an up signal the gate has a continuous variable which is kind of denoting let us say this is my gate and it is denoting the height of the gate. So, when it gets a signal down x must have a negative, I mean a negative differential. So x dot is 1 - x by 2 something like that which gives me a negative value and the gate will come down. When I am getting initially so the initial condition is x = 1.

So, when I mean the gate will eventually lower down here and then when it is up, the up signal comes. So this is the flow vector now 10 - x by 2 etcetera which will ensure that this is positive and the gate will in a finally go up. So, that is how these two different automatons are conducting themselves.

(Refer Slide Time: 11:24)

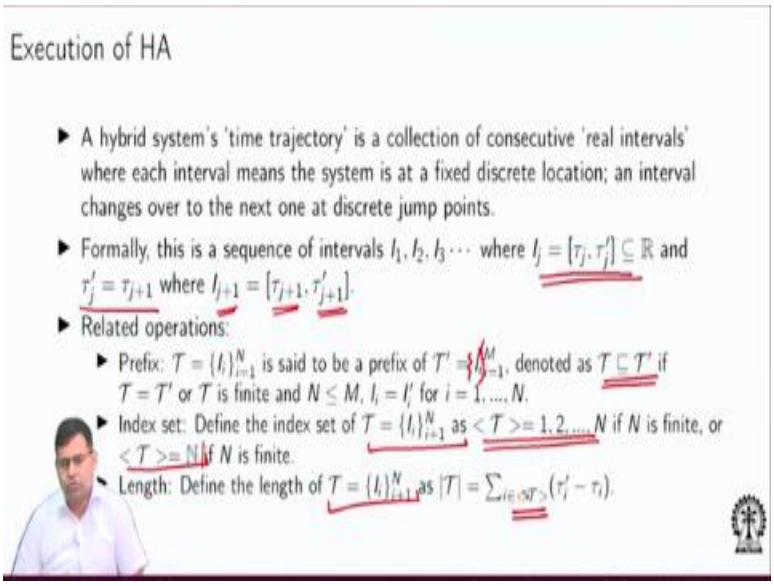


And if I am going to kind of combine them that is the combined location of train plus its controller and the gate. So, that is the dynamics of the train. This is the dynamics of the gate that means you are signal the gate to come lower down. So, this is the composed automaton as you can see. So, in each of the states you have the train I mean approaching the gate and it has I mean this is the synchronization of the states.

Because based on the common events it has been decided which of the states can be together and which of the states cannot be together. So, the train is, when the train is near, I mean it has already signaled down to the gate so this is one admissible state and when the train is far further that means its satisfying y = 5 in between that plus y = 5 to 15 in between it is a kind of in the left state and the gate is rising up.

So, and the train dynamics, the gate dynamics and the invariant of this condition. So, that would be the composed model in this case.

(Refer Slide Time: 14:48)



This of the issue of composition the next important thing to decide is what about I mean what is the execution semantics of a hybrid automaton. How do I formally define a run of hybrid automaton? If you remember we talked about run of finite automaton that is a sequence of input events but here we also have this notion of real time. So, for a hybrid automaton we talk about a time trajectory and it is nothing but a collection of consecutive real intervals.

What is happening with a hybrid automaton is that I am spending a continuity of time in different discrete locations. So, that gives rise to a set of real intervals and for each interval I am, the system is in this fixed discrete location. Then there is a location switch and I am going to another location there again a real interval is being spent and then there is another discrete jump I go to another location that is how it is.

So, it is like a sequence of intervals like I_1, I_2, I_3 we have here and for every such interval I_j its basically like this, there are two real time points τ_j τ_j' and this interval is the subset of the real set and the next interval is I_{j+1}. So, it is starting from τ_{j+1} up to some point τ_{j+1}' and what happens is this $\tau_j' = \tau_{j+1}$ which is the so the end point of the I_j it must be equal to the start point of the next time interval, the next real time interval which is I_{j+1}.

So, that is τ_{j+1} and $\tau'_j = \tau_{j+1}$. Now there are these related operations so, there is this prefix operation. So, first of all then what do we have? So, we have a time trajectory to summarize it is a sequence of this intervals. Now once we have a sequence, we can define prefix of a sequence. So, a prefix is defined like this. The tau is one time trajectory given by a sequence of intervals I_1, I_2 up to I_n and let us take.

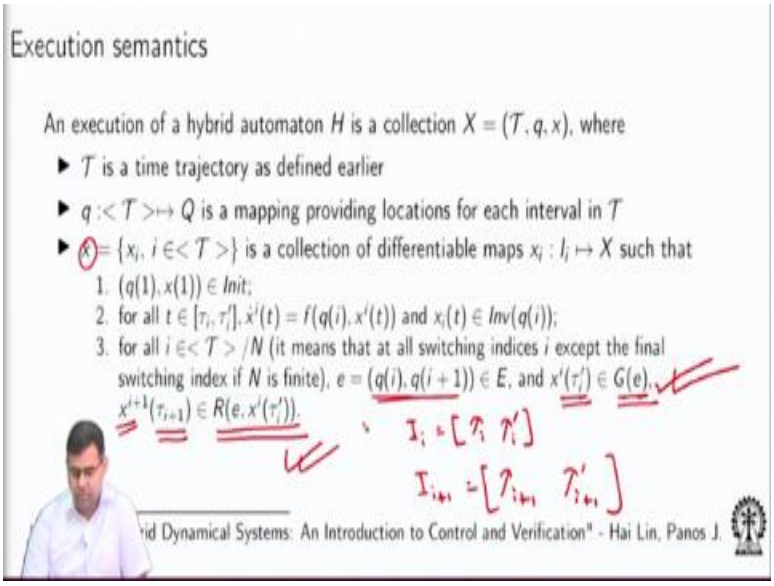
And its prefix as another set another tie set of time trajectory I mean another sequence of intervals tau prime which is up to some M, I = 1 to M. So, this is another set I = 1 to M. And we will say that this tau is a prefix of this set tau prime in two cases either they are equal, tau and tau prime are equal or tau is finite and the number of indices that means the number of intervals in tau is less than equal to the number of intervals in tau prime.

And each of the intervals in tau is I_i, is exactly equal to the intervals you have in tau prime up to this value N. So, the intervals in τ_i τ let them be I_1, I_2 like this the intervals in tau prime let them be I'_i, these are primed here like that. So, we will say that this is exactly a prefix then if whatever are the intervals in tau, they are all present in tau prime in that sequence only and after them there are some more intervals. So, there is the definition.

Now we have another thing called the index set. So, index set is given by this. So this collection of intervals that we have. So, this is my tau and the number of such switches that we have that means the number of in intervals we that we have. It is given by this set the index set. So, if there are N such intervals the index set has values 1 to N. If N is finite and if this run has got infinite number of such consecutive intervals, then the index rate is nothing but the set of naturals because it is up to, I mean it is just continuing infinitely. Now what is the length of time trajectory length we can figure out simply by adding of the consecutive intervals real values. So, if this is the set the consecutive sequence of intervals then you take each of them from the index set. The index set is given by this L-angled tau right angle. So, it is containing this the indexes right or one two three like the indexes of each of these intervals.

Now for each of this inter, indexes you have the finish time and the start time so this subtraction gives you that interval value you just sum it up so that will give you the length of the run of the hybrid automata (20:02) of the time trajectory.

(Refer Slide Time: 20:14)



So, overall, we can define the execution of a hybrid automaton like this. It is a collection first you have the collection is a three tuple. First you have the time trajectory tau, the next you have is a mapping queue from the index set to the set of locations which tells that well inside this time trajectory each of those intervals I have spent in which of the locations, that is what it tells. And then I have x so, what is x?

It is a collection of the differentiable maps. That means it is telling me that inside each of these intervals inside the time trajectory you see i belonging to the index set what were the flow functions that I have followed. So, that is why those are the differentiable maps. So, what are the flow functions that I have followed for each of the variables inside each of the intervals. So, they are given by that so they would be like defined like this.

My initial state is this q_1, x_1, so I mean what is my location and what is my current for this location with work flow function I am starting that is my initial location. And then for each of these intervals τ_i to τ_i', what is the flow function that I am following. So, there would be something of this form that some location I have this $\dot{x_\iota}$ and so for each for that ith interval I am following some flow function f and I am in some state q_i so, you see I mean everything is algebraically defined here quite nicely.

So, let us say I am in the second interval inside this sequence of intervals tau and the second interval so i value is 2. So, i value 2 means what do I really have I am trying to figure out what are the flow functions being followed. So, the flow functions would be very simply figured out like this f is nothing but I will then check with this function q for the value 2 it will give me a location. So, q_2 here will give me a location that where I am in that second interval.

So, that is my q_2 that is my location in that location for each of the variables I will have the flow function defined and that will be my differentiable map $\dot{x_\iota}$ here, that is it. And then some another interesting thing is for all this i's that are in the index set, I mean, minus the final one except of I mean this minus the final means all the switching indexes I except the final one. In case N is finite otherwise if it is of a infinite then there is no final one there is no meaning of – N.

So, in that case we have a set of edges and they are defined by like this the you start from the ith index's location to the i + 1 index's location. And for each of them you have the corresponding so this will give me the edge which I took for going from the ith interval to the i + 1 at interval in the time trajectory set. And this would be my corresponding guard condition and of course this would be my corresponding reset condition.

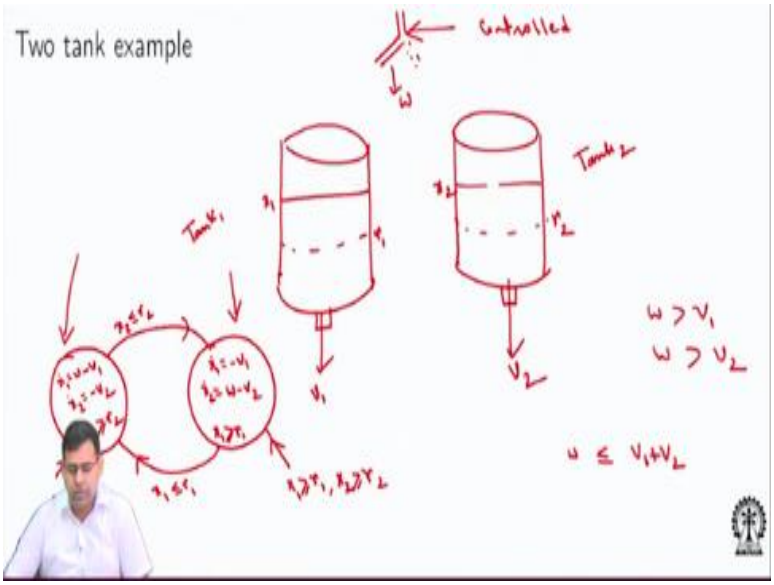
Now how do I get these times? So, it is very simple. So, we are talking about the ith interval so I_i so if you remember I_i starts with τ_i and it ends with τ_i'. So, at the start τ_i I am going to have this

interval this I mean at the end of this I am going to take this transition. When I take this transition, I should be at that time I should have a valuation or my mappings I mean at this time instant tau i prime my vector fields must be giving me suitable valuations of the real values.

So, that this guard condition is satisfied that is the point, and not only that once I take this transition, I get to the second interval set τ_{i+1} τ'_{i+1}. So, now I am at τ_{i+1} I am following the next set of differentiable maps x_{i+1} and they must belonging there must belong that means they must satisfy the reset set of this transition. So, I took this transition e and while I took this transition, I satisfied this set I mean this set of valuations for the map.

So, whatever is the reset condition here that must be satisfied here. So, when I take this map when I go to τ_{i+1} that means the next time instant in that next target location. So, wherever I am in wherever I land up in that location then I would have a differential map which should be belonging to this reset set. That means in that edge set whatever what is the reset set, I should be there.

(Refer Slide Time: 26:10)



So, here let us take a simple example. So, let there be these two tanks. They are filled up with water and this is the reference value of water that means the water level at all possible time must be above this r_1 and r_2 in this one that is tank 1 and this is tank 2. And the actual water level that

that be given by some x_1 and here it some x_2 and they are also having some outlets through which the water is also draining out.

That some phase v_1 and v_2 and there is an input pipe which can either be here or it can be here, and that pipe has a flow at w and this switch can be controlled. You can control this location. So, this is known as the two tank problem. And so, the thing is you have to have a suitable control law to control the location of that inlet and you have your that is the control and the control problem is that you must have x_1 greater than r_1 and x_2 greater than r_2.

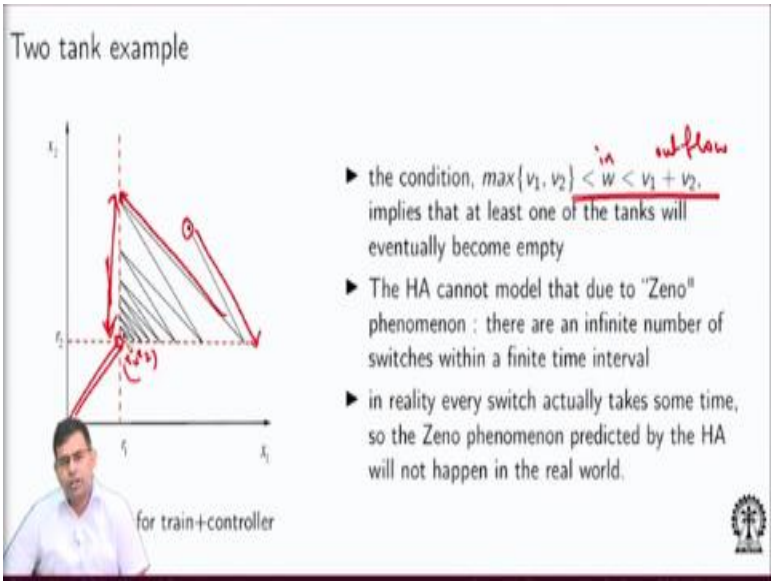
Now what should be your control strategy. So, let us say your control strategy is that you switch the inflow to tank 1 whenever kind of x_1 goes below or $x_1 = r_1$ and you switch the inflow to tank 2 whenever x_2 goes below or $x_2 = r_2$. Let that be the control strategy. Now if we try to identify a hybrid automaton for that let us see how it turns up. So its basically like this. So, this state designates that the water flow is towards I mean tank 1 and this state means that the water flow is towards tank 2.

So, when the water is flowing towards tank 1 you have a net positive water flow assuming that w is greater than both we are assuming that w is greater than both v_1 and I mean w is also greater than v_2. So, you have this that this net increasing the tank 1 value is rising but of course the tank 2 will be going down and you have a location invariant that tank 2 is still having some value greater than the reference.

Whenever it is less than the reference you switch the flow that was by control strategy remember that is what we discussed. So, you switch the flow here so to x_1 now what happens is tank 2 will have a positive gain and tank 1 will have depleting water and similarly we have a condition that I can only be here as long as tank 1 where I have depleting water does not deplete fast enough and go beyond r_1, because whenever that happens I again switch the flow. So, that is the condition.

Now so that is about it. Now let us assume that w is such that let me write it down w is less than $v_1 + v_2$. So, if this is so then what is the expectation. The expectation is eventually, I mean my net inflow is less than the total outflow. That means whatever I do whatever is my control strategy, eventually both the trunks will run out of water that is there is no that is going to happen. So, let us see if the simulation of this situation really leads to that let us see that.

(Refer Slide Time: 32:05)



So, you see suppose I started somewhere here. So, here I am plotting both the values x_1 and x_2. So, initially I am filling up tank 1 so tank 1's value is rising and tank 2's value is going down. So I am I am following this. Somewhere what will happen is tank 2's value will be very small almost touching r_2. So here the control will switch. So, this is the condition we have assumed like I said that the total inflow is less than the total outflow.

So, when the w tank 2 has depleted almost to r_2, then it will switch and now what will happen is tank 2's I mean water level will rise but since tank 1 does not have a supply it is decreasing. Again, I will switch so you see whatever we thought that eventually the water in both the tanks will deplete that is happening. But the problem is I am not going to zero I am going towards this point r_1, r_2 which was which was not supposed to happen.

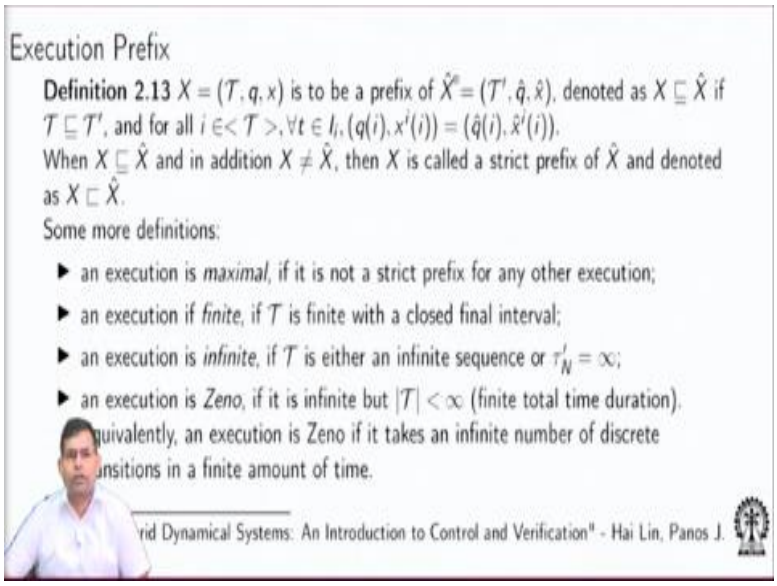
Also, if you see as I approach r_1, r_2, the switching's become very fast and if you think in the limiting case what is happening is I am going to have too many switching's inside this finite amount of time this is a finite amount of time. And here with the switching is becoming extremely fast I mean I am actually going to have infinite number of discrete switches inside this small finite interval which cannot really happen.

So, this is known as the Zeno phenomenon, I mean the switches, because I mean because this is something that it cannot happen in a practical world. I cannot have infinite number of switches inside a finite interval which is happening here. So, this phenomenon is named after Zeno of Elia, as you know Zeno of Elia who was a famous Greek philosopher. So, it represents what a causal system cannot have.

So, this hybrid automaton happens to suffer from Zeno phenomenon and there is a good reason. We are assuming that these switches are not taking any time but for all practical purpose in a real tank even the switching of the controls water in flow to outflow they will take practice if a finite amount of time. Since we are assuming them to be instantaneous actions it is happening that here in this specific case, I am having such as an exhibition of Zeno Phenomenon where we have so many infinite number of switches happening in a finite interval.

So, there are techniques to remove them from automaton. So, that is something you can study in the future in some in some advanced course. But what we only want to do is we want to make you aware that hybrid automaton can exhibit Zeno phenomena in some cases.

(Refer Slide Time: 35:09)



So, if you remember in the previous few slides, we talked about execution prefix. Suppose so let us X be our (35:24) time trajectory of the automaton where you have this sequence of intervals followed by this sequence of locations intervals along with these locations and this switch in the vector field. And let us say this is a prefix of another execution, it is denoted by this, it is I mean X is a prefix of $\hat{X}$. If T is a prefix of T′ that means all the intervals here are covered inside this and in between for all these intervals whatever has been the locations they have remained same in both of the executions and whatever have been these differentiable maps they have also remained same in both of the locations. So, that is it. Now a few more terminologies we will like to introduce is so for such an execution of an automaton I will say that it is minimal it is sorry it is maximal if it is not a strict prefix of any other.

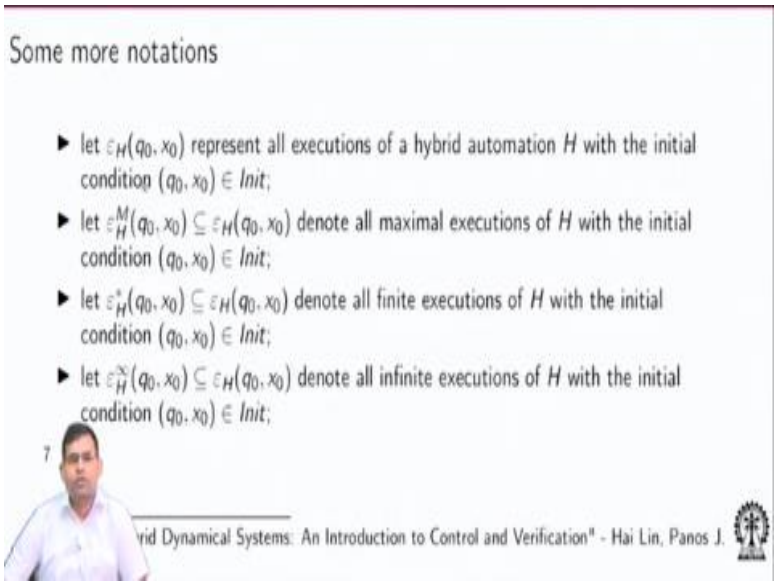
Strict prefix means that there is at least one more interval in the larger sequence. So, if there are no others who have one more interval with a larger sequence then this is the maximal execution. It is finite if its indexed at T that means the number of intervals is finite. It is infinite if the index set is the set of naturals, that means I cannot find a final interval N is its value and the execution is Zeno if the number of number of switches we have is infinite.

That means the number of this intervals that we have is infinite. But if you remember we also had a calculation of the total time this was my calculation of the total time, that is not really so. So, this is a nice way to capture it we are saying that the total time is less than infinity I mean the total time is finite it is a finite time duration but the index set is infinite. So, this is kind of a mathematical way to capture Zeno phenomenon.

So, essentially, we have infinite sequence of intervals but the total time taken by the intervals is in a limiting case is approaching a finite value. So, that is how we can mathematically characterize these different properties of automaton simulation or automaton's execution, whether it is Zeno whether it is infinite that means and all that. So, just to repeat its maximal if there is no larger sequence. It is finite if I have finite number of these intervals where it is staying in different locations. It is infinite if this index set is infinite that means I have an infinite number of this sequence of intervals. It is Zeno if I have infinite number of the sequence of intervals but their total time elapsed is finite. That way that is the Zeno situation. So, it can be this I mean for Zeno it has to be infinite but the total time in elapsed is finite.

When in a normal non-Zeno case what we will have is, the non-Zeno cases whenever its execution is infinite that means I have an infinite number of sequences in that set T the corresponding index set I mean is infinite but the total time taken of the intervals is finite. So, I mean sorry the execution is infinite if both are infinite, that means that the total number of intervals is infinite and the summation of the intervals is also infinite. The time elapsed is also infinite in total**.**

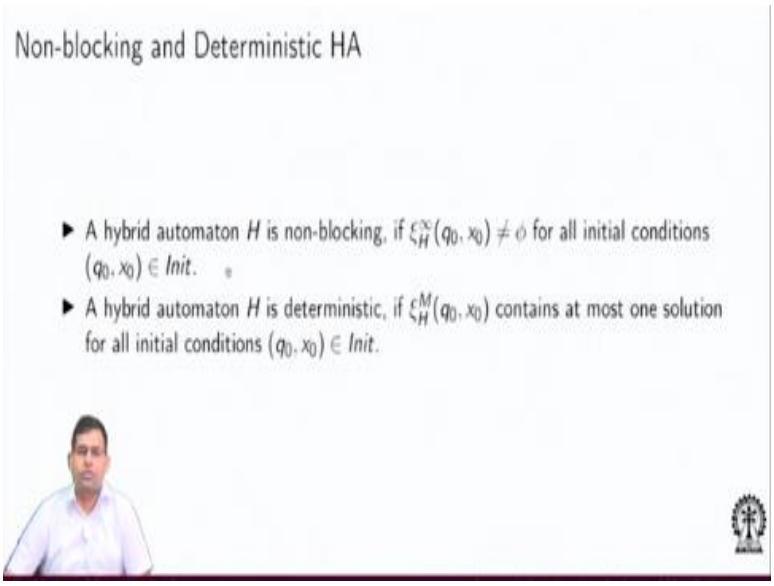
(Refer Slide Time: 38:54)



Now with some more terminology. So, let us say this is telling me, this is a set which gives me all the executions that a hybrid automaton can have starting from an initial step or initial condition. Let us say this is a set of the total number of maximal executions that means the all the executions after which there are no further possible locations. And this is the set which denotes the total number I mean the set of all possible finite executions. That means executions continue and then they end with some final location.

And these are the total possible number of infinite executions that are possible from an initial state. So, just to remember these notations because we will use them for some another definition will end it here. So, this is for all executions of the automaton. And the second one is for all maximal executions. The first one is for all executions among them those who are maximal they are in the set this ε_M. And this ε^* is the setup on those finite executions and these are the infinite executions.

(Refer Slide Time: 40:11)



So, we will say that a hybrid automaton is non-blocking if the set of infinite execution starting from the initial state q_0, x_0 is nontrivial. That means there exists at least one sequence of intervals which is and that sequence is infinite. So, that is an infinite execution sequence. If that is there then if we have such a sequence which is an infinite execution sequence and at least one such sequence is there, then I say that the hybrid automation is non-blocking.

That means it does not I mean there at least exits a trace which is allowed to proceed and maybe I mean it is not happening that whatever stress, whatever execution sequence every automaton we make it follow a different possible locations at different possible times, it eventually blocks due to a guard from condition or an invariant condition. It is not happening. There is at least one possible way for the execution to proceed up to infinity then we call it as non-blocking.

And when do I call it in the deterministic? If I find that the maximal execution set has only one element that means there is only one possible wave starting from every initial condition, I start from every possible q_0, x_0 combination. And for every such initial state if the hybrid automaton has one unique execution sequence. That is what is this says that I mean maximal means what all the possible maximal that means it is not a prefix of anybody else there does not exist anybody else larger than that.

Let me here if that set is unique that means there is only one possible trace one possible unit trace there is no other trace which is larger than that or whatever and that is and it exists. So, this set is a single tone for all initial conditions that makes it the hybrid automation deterministic. That means I give it an initial condition and it and based on that initial condition there is exactly one possible way for the hybrid automaton to proceed.

That means it will be in this location for this much time then it must switch to that location for this much time then it must switch to another location for this much time. That is a unique sequence. If it is so for all possible initial conditions then it is a deterministic automaton and when do I say that it is a non-determined, non-blocking automaton? We say that its non-blocking if there is always at least one possible way for the hybrid automaton to execute up to infinity starting from any initial condition.

Then it is a non-blocking automaton. That means there does not exist a initial condition from where the automaton starts and then whatever way it wants to go forward it is blocked by invariant or guards it is not happening. Then it is a non-blocking automaton it can always progress in some direction up to infinity. Then it is non-blocking. And it is deterministic if its way of progressing is unique.

So, these are the different kind of execution semantics of hybrid automaton and with that we will end our lecture here today. Thank you.