Foundation of Cyber Physical Systems

Prof. Soumyajit Dey

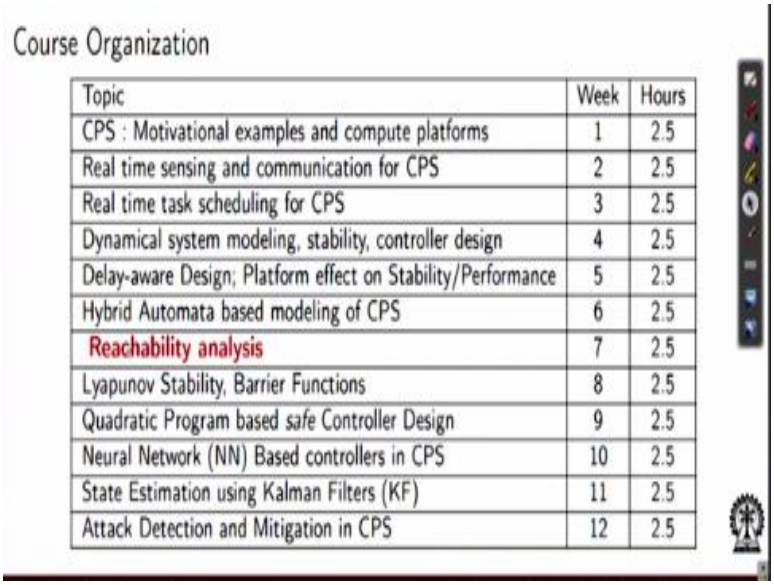
Department of Computer Science and Engineering

Indian Institute of Technology, Kharagpur

Lecture - 33

Reachability Analysis

(Refer Slide Time: 00:39)

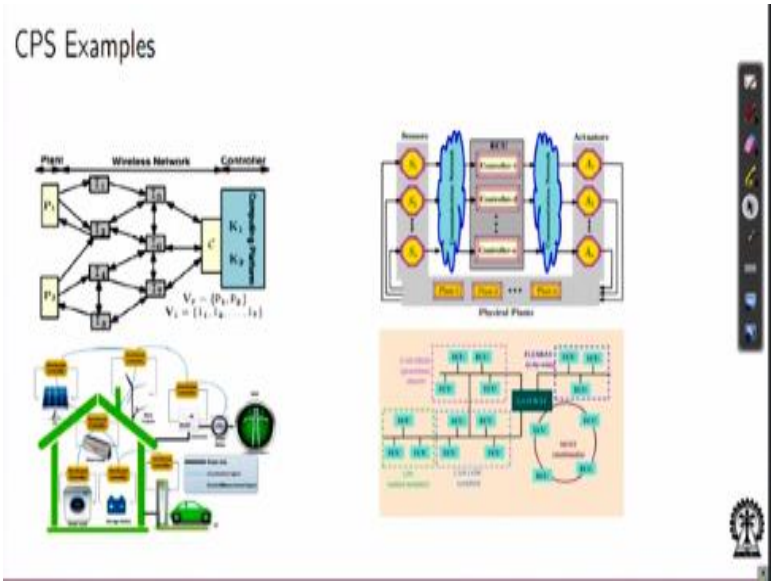


Hello and welcome to the week 7 lecture of the course foundations of cyber physical systems. So, if you remember in the previous lectures, we have been talking about the basics of hybrid automata and how hybrid automata can be used for modelling cyber physical systems. And the next important thing we want to talk about is, I mean well we saw how hybrid automata can be used for modelling CPS.

There are so many tools using which you can actually simulate CPS models of hybrid automata. But what is also important is you will like to reason about such models and figure out whether at the model level these systems can have some specific desirable properties like safety property,

liveness property and others which we will see what those things mean. So, this part we will be covering in this week under the header reachability analysis of hybrid automata.

(Refer Slide Time: 01:27)



So, just if you remember the different examples of CPS that we talked about earlier we can have a CPS which is implemented as a controller scheduled over a compute platform and it is computing control signals for multiple plants. And there are communication between the plants and the controller happening through this kind of networks. Also, we can have a complex cyber physical system in the form of a smart grid where you have electrical appliances inside a home.

And not only these appliances which are the consumers of the electricity but also there are some small producer systems like solar PVs and others which are present in a modern home and there are other consumers like modern loads like the electric vehicle loads and they are kind of provided the power through a smart grid. And there are several problems in this domain like electricity scheduling, safe operation of the generator figuring out whether the generators of the grid are working inside they are safe operational limits, all these problems can also be modelled as hybrid automata. Now like we said earlier that it is not only the case that I will like to model them using hybrid automata and use a hybrid automata simulation tool for figuring out how given an initial point in the system, initial state in the system how does the system's trajectory evolve over time.

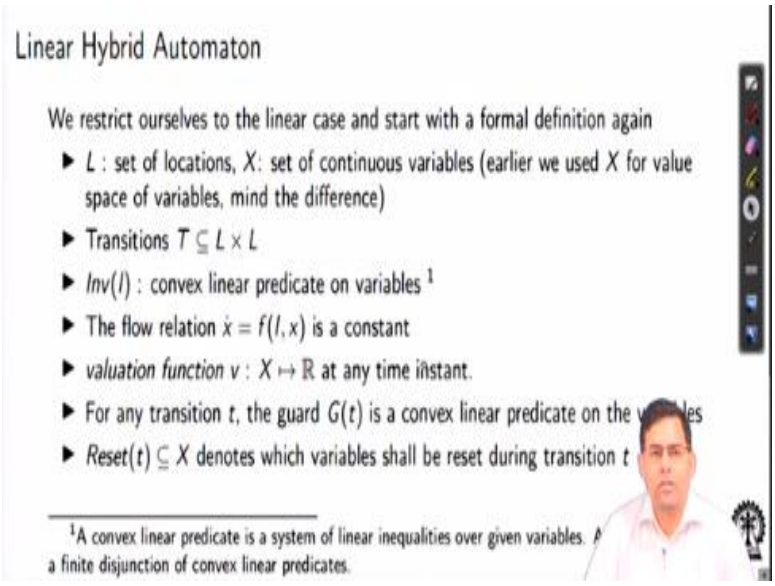
I will also like to see that assuming that the system can start not only from a state but a collection which can possibly have infinite number of initial values, where does the system trajectory go? What are the regions in the state space of the system that the system trajectory will diverge? And whether those regions are safe operational are or not we will like to know all these things. Now of course you can understand that the moment we talk about ah region of the system as an initial region that the system can start from an addition region which is a collection of infinite number of states, I cannot really simulate the system for all those initial states and figure out whether the system is safe. I would rather like to have some formal method through which I can analyze whether the system state having a safe trajectory where it can be, the state can start from any of those initial states which are part of the initial region.

So, we will define what these things mean. There are also several other applications like automotive systems where you can model the different vehicle control loops using hybrid automata and figure out whether the design of the control along with, I mean the delays that can happen on the control side or the sensing side whether with all those uncertainties in the platform the overall system is safe or not.

So, this is the primary reason why we will like to talk about reachability analysis of hybrid automata. We will try to figure out that since it is not possible to simulate the system for all possible starting states because the set of starting sets can be infinite. We will like to have this kind of automated reasoning-based methods through which we can analyze whether the system is safe or not.

We are not claiming that that can be done for any complex system but there are methods through which this can be addressed for some for many practical systems where the complexity is kind of abstracted out and an analyzable model of that system is created using a subclass of some hybrid automata. We will see that.

(Refer Slide Time: 05:26)



So, we will restrict ourselves to this linear case that means we are we are talking about hybrid automata where the dynamics of the continuous variables is always linear. So, let us understand what it means. So, we will go back to the definition of hybrid automata, modify it a bit. So, of course the modifications are not going to happen on all the total elements on some of them. So, we will have our set of locations, and there is one small notational change we are doing.

Earlier we were talking about this set X as the value space of the continuous variables. But now let us use this variable X to be representative of the set of continuous variables in the system. So, suppose you have two or three flow variables in your hybrid automata. Those are the variables that comprise X now. And the discrete switches are modelled by transition relations which will make the automata jump between locations that are in L.

So, in that way we will have a transition relation like this and then what we have is the invariant set. Now the invariant set is if you remember earlier, we told that invariant set again for a location it means that a region of the state space for the continuous variables in which, if the system is in that region then it is allowed to be in that locations. That means if I am in the location, I may satisfy some mathematical condition which captures that region.

Now in this case we will restrict it to be convex linear predicates on the variables. We will soon describe what it means by a convex linear predicate. Now the next thing is the flow relation. So, for every variable small x inside capital X, we will have a flow relation like we defined earlier that a derivative of x is given by some flow function which is I mean f which is defined based on for each location and for each very continuous variable we will have this flow function.

Now since this is the linear case, we will assume that this is a constant. So, what it means is that for any such variable. So, if this is time and let us pick up some variable x. Let us say initially the value was here, is going to increase with a constant gradient. So, this gradient is basically a constant value which is given here. So, that is how we will now abstract it out I mean it does not increase like this or like this or like this it does not increase like this.

So, that is what we are talking about here. Now the next thing is this is similar we are talking about valuation function. Now what is valuation? It is a function which tells that at any time instant each of the continuous variables who are going to assume some real value like what is the value. So, that means suppose I have continuous variables $X= x_1, x_2$. So at $V(x_1)t$ equals to will be something and $V(x_2)t$ equals to will be something.

Let us say this is 3.9, this is 4.2 like that. So, of course X is an n array set then this is like each variable. We have corresponding real value here is given by this valuation function. The next important thing is for any transition. So, let us just try to create a transition here. So, assume these are my locations. So, this is one transition, let us call it t. Now this transition like earlier they can have a guard and reset.

Now this guard, let this be the transition t and there is a guard G(t) it is a convex linear predicate defined on the variables. So, we will need to define what is a convex linear predicate? We will do that just wait a minute for the time being is just a constraint defined over the variables in x. Now when we say convex linear predicate it is a specific class of constraints. So, like we discussed earlier valuations I mean they give me values for the real variables.

Invariant set guard they represent a region in the state space of those variables and the region is always captured by a mathematical relation. In this case we are focusing on the relation being a convex linear predicate. The other thing is the reset set. So, just like guard we will also have the reset set. For example let us say this reset set is containing x_1. That means when this transition is taken let us say here before the transition was taken the valuation of x_1 was some 4.5, let us say.

The moment I take the transition the valuation of x_1 will be 0 that means it will restart its value it will it will restart its increase of value from 0 the moment I get inside l 2. Now coming back to this issue of convex linear predicate, so what it is? So, a convex linear predicate is a system of linear inequalities over a set of given variables. So let us write a structure for them. Suppose you have variables x_1, x_2 up to x_n and let us say there are some constants a_1, a_2, and a_n.

So, it is a linear combination of these variables and constants. Linear combination that means it can be something like this and they are related with some constant here. I mean in the general case you can push it in this side and have a zero that is also I mean some books will describe it that way. Now what is this? Represents a relation which is any one of this. So, you can see that it is a linear equality or inequality relation defined over the variables and with some constants.

So, that is a convex linear predicate. Had there been only two of these variables it would be a subset of such convex linear predicates you mean or more in general we will I mean, I write this it represents what we call as a convex polyhedron's one surface and there are many subclasses of

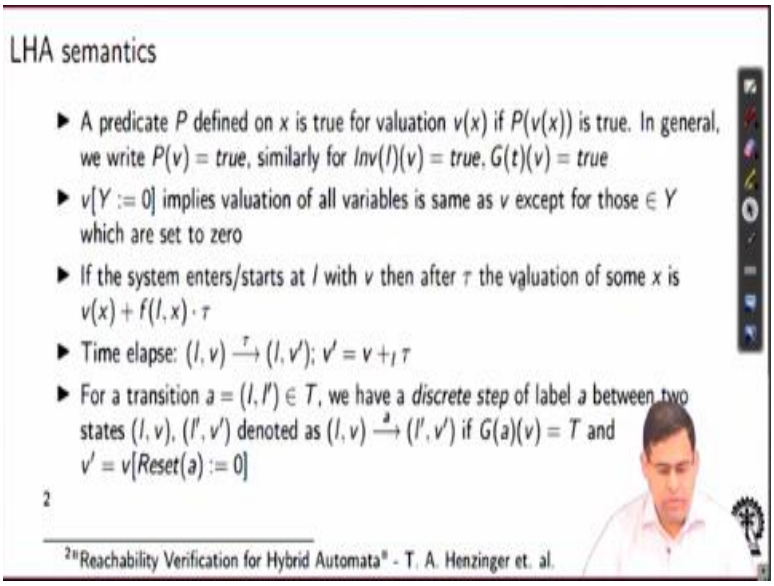
that which we can define later on. Now of course this is one such predicate. Now we can of course keep on combining such predicates.

So, let us call this some P_1. So, when I say overall what is the convex linear predicate it is a system of such linear inequalities. So, I wrote one of them, so in general when I define any convex in your predicate it can be a conjunction of such linear inequalities. So, let this be some p. So, this is a convex linear predicate. In general, when I define a linear predicate, it would be a finite disjunction of such convex linear predicates.

So, overall, when I have say some linear predicate, it is basically let this be P along with some P' and P" like that. So, what we did is we defined those linear predicates in terms of linear inequalities over the variables they are conjunction gives me a convex combination that is a convex in your predicate and a finite disjunction of such convex linear predicates gives me this kind of linear predicates in general. So, that is how we define it here.

So, this is how we will restrict ourselves to this general subclass of hybrid automata. Let us call these the linear hybrid automata. So, as you can see, we call this linear because the flow relation is linear and we have all the constants defined in terms of these linear predicates.

(Refer Slide Time: 16:13)



Now this was just a tuple definition. So, it will also have its semantics just like earlier. So, it is like this any predicate P which I define on the variables on x, it is true for some specific valuation. Because, the variables are assigned the real values using this valuation function. So, what it does is given a valuation function I can interpret the predicate P. So, let us say the predicate says x - y less than equal to 2.

So, let us just write it here as an example. Given a valuation function I have v(x) and v(y) and if those values are let us say 3 and let us say 2.5 then this is true. So, in that way I have to interpret the predicates over specific valuation functions. Now similar to this idea since my invariants and guards are all convex linear predicates, I have to interpret them with respect to a location and the valuation of the continuous variables in that location and figure out whether the invariant in that location is true for the given valuation, or similarly the guard of some transition t is true for the given current valuation of the variables. Now this next statement gives me a symbolic encoding of resetting of valuations. So, if you remember when we took a transition here from one location to another, we have an associated reset set.

So, if that is y, this notation v y reset to zero means I am going from the state v of valuations for the new set of valuations for the variables where the variables present in y get initialized to 0 and all the other variables have the same valuation as it was in the original v. So, that is how it is

defined now. If the system starts or enters, I mean a location l with a valuation v then after some time tau the valuation is given using this relation. So, it is v(x) so suppose I am talking about the valuation of some specific variable x the system has just entered a location l.

And after entering the location l tau amount of real time has passed. So, I want to know that at this moment after this time amount of real time has passed what is the valuation of this variable x. So, it should be the original valuation which is v(x) plus the rate of increase of this variable x in the location l which is a constant as per the definition of linear hybrid automata multiplied by the time elapsed which is tau. So, I think this is quite clear.

So, in general such a time elapsing operation is denoted like this. That I am in the state l,v that means I am in location l and presently my valuation of all the variables is captured by the function v after elapsing of real time of amount tau my state is l,v', where v' is nothing but all the original valuations in v for each of the variables increased by an amount that is tau.

So, that is what is denoted by this. I mean you can just write plus here it just means that this is happening in the location nothing different than that this has no hidden meaning etcetera. Now for a transition in l prime, so, whatever we discussed earlier we are just now bringing in the notations again. We have a discrete step, so earlier what we defined was the time relaxing operation now we are talking about the discrete step.

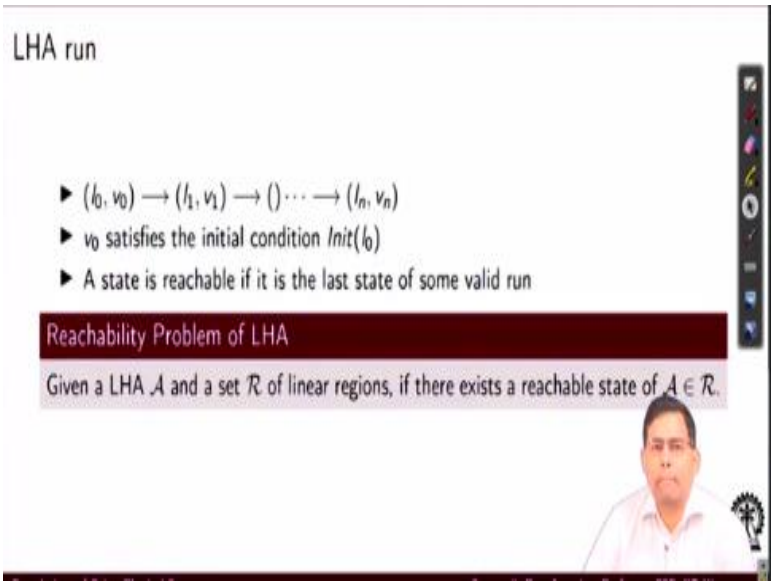
So, suppose here I am taking this transition a, so I have this discrete step with this transition of label a between two of these states l v and let us say that I jump into l prime and the moment I jump into l prime whatever this my valuation is given by v prime. Now this situation that I was at l,v, I made a location switch that means I took a discrete step or discrete jump using this transition with label a.

So, exactly this is how it is being described in formal language. So, from this state I jump to l prime v prime it involved a location switch using this discrete transition with label a. Now let us say this transition had some guard G. Now this transition will be possible only if at the time of the transition if my valuation was v, this valuation is satisfying the linear predicate which is the guard here.

So, like we discussed earlier that the guard and invariants are also convex linear predicates and whenever I have transition the guard has to be satisfied. So, this transition is only possible if at the moment of the transition the valuation v is actually making this guard condition which is this linear predicate to be true. And also, how do I calculate v prime? You will calculate v prime exactly using this operation.

So, as you can see that this is why we have defined all these things that when is guard true, when is invariant true what how to write the reset. Because they are in turn used to define this discrete step. Because now v prime is nothing but v with the modification that whatever is the reset set of this transition with label a being set to zero. So, whatever variables are in reset they will be starting from 0 when in v' and all the other variables that are there we will just have the original values as it was in v. So, this treatment of LHS semantics and whatever most of the things that we follow have been taken from this material reachability verification of hybrid automata by T Henzinger and there are many co-authors in this.

(Refer Slide Time: 23:15)



So, in general this is how a run of the LHA would look like. So, let us say your initial location is l_0, v_0 I mean initial state is in l_0, v_0 where the location is l_0 and the valuation is v_0 and you make a discrete switch to l_1, v_1. You may will have some time here then again you make a discrete switch like that. So, typically here we just show the discrete switches and in here we show what is the transition level along with the time of the transition.

So, typically that is how an LHA run is being described. Now one thing we must remember that this initial state l_0, v_0 should have the valuation such that it satisfies the initial locations invariant or the initial states in it that is of l_0, that what is the initial condition of l_0 that Init set. If you remember it was discussed earlier in the definition of hybrid automata, that what that when I am starting from a state what is my possible set of initial valuations.

And then in the initial state I will have a possible invariant. So, ideally both these things have to be satisfied the initial condition as well as the invariant of l_0. You will see that. I think it is also discussed in some of the previous lectures. Now when do I say that a state is really reachable? I would say that a state is reachable if it is the last state of some valid run that means it is possible to start from some initial state comprising, I mean an initial state like we have already defined.

It means that I am in one of the initial locations and my valuation satisfies the initial condition of the automata as well as the invariant of the initial location. So, I start from a valid initial state and I may make suitable time elapsing transitions like this in the different locations and I may make suitable discrete steps to jump to different locations. And in this way, I finally reach a state then I would say that this state is reachable.

So, for every state to be reachable such a run has to exist from some initial state up to that state as you can recall that any state just means it is a collection of location and valuation. So, in general we have this problem which is what we call as the reachability problem of linear hybrid automata. So, it says that suppose you are given this kind of an automata A and a set R of linear regions.

So, it is a collection of linear regions, so that means they would be represented by a linear predicate like we discussed earlier. And the linear predicate as we discussed is it can be a disjunction of the convex linear predicates and the convex linear predicates are nothing but conjunction of elementary context linear predicates or any basically the linear inequalities. So, the question is suppose I am given this A and suppose I am given this set R of such regions, expressed as the set of expressed using linear particles. Is it possible for the system to reach that say, that means can the system start from one of the valid initial states of A and have run like this to reach some point that may point in the state space or alternatively a state which is inside this set R. So, that is the reachability problem and we want to understand how to solve the reachability problem.

So, fine we will stop the lecture here and maybe in the next lecture we will start from this reachability problem and a technique for going about solving. Thank you.