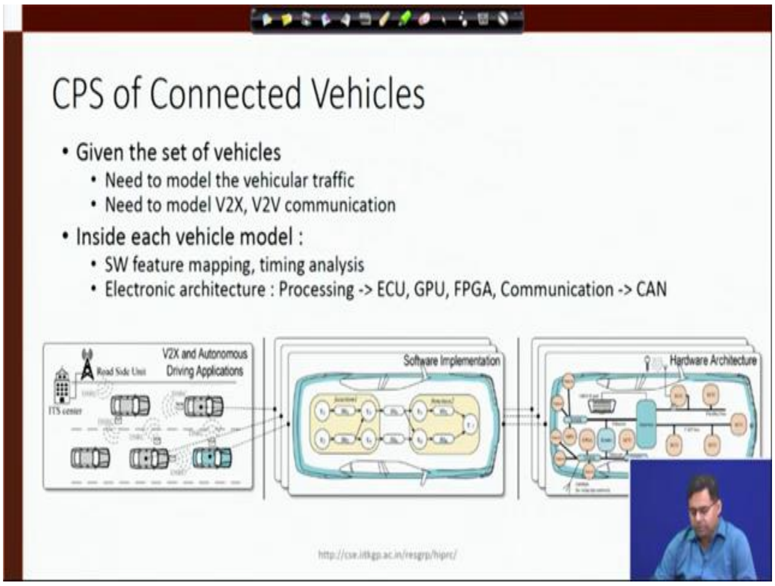
Foundation on Cyber Physical Systems
Prof. Soumyajit Dey
Department of Computer Science and Engineering
Indian Institute of Technology – Kharagpur

Lecture – 02
CPS : Motivational Examples and Compute Platforms (Continued)

Welcome back to this course on Foundations of Cyber Physical Systems. So, if you remember in the previous lecture, we motivated this area of CPS. And we talked about a nice example of building temperature control.

(Refer Slide Time: 00:45)



And today we will be carrying forward from there. And maybe we will be starting with this example of connected vehicles. Like how cyber physical systems have applications in the automotive domain. So, let us let us take up some problems in this area. So, suppose you are given a set of vehicles and you need to model. Ah I mean what are what are the what are the things I can ah control here?

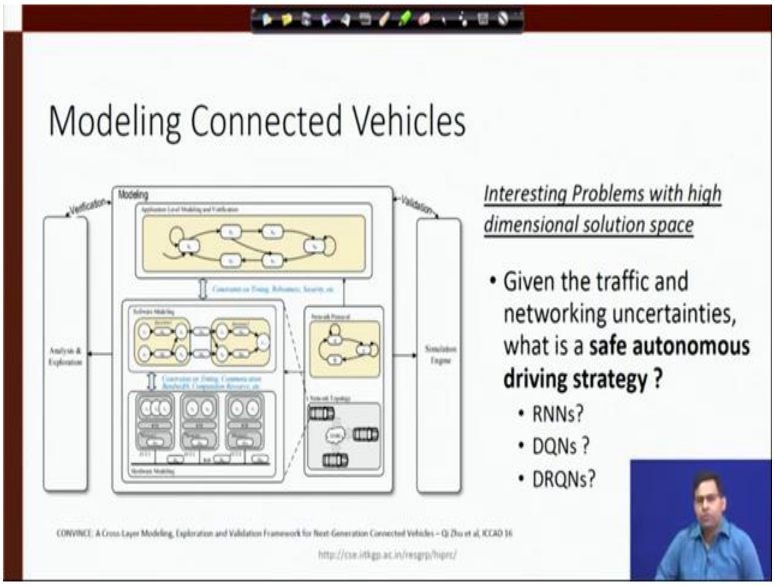
So, well, I can have a set of vehicles. And I will like to make them move autonomously, while being safe and while being aware of the traffic which is already there in place on the tracks. ok So, if I have to design such an autonomous vehicles or control software. Of course, we will have to start with that simulation Modeling modelling simulation control, design and implementation loop. So, the first place again to start with would be to model the vehicle traffic and how the vehicles are moving about, and what is my ego vehicle, how many vehicles are there in my ah platoon? Let us say where the multiple vehicles are communicating among each other. And not only that. We will also need to model the V2X, vehicle to infrastructure and vehicle to vehicle communication because ah well, if if I am doing a simplistic simulation that may not be required.

I can think that well, this communications are happening with some specific amount of delay. But for a realistic simulation, I will need to know that well ah whether my communication is happening following real communication phenomenas. Like packet drops are happening, some real delays are happening and the relay relays are varying with respect to some realistic distribution as per the communication theory standards. So, all those things need to be ensured. right So that is about the vehicular traffic modelling and the requirement of ah having a community having a simulator which is kind of modelling the vehicle to vehicle communication. But there are much more intricacies here. Right What about one vehicle? What about the modelling of one vehicle? Inside it whatever the things do I need to model?

So, inside a vehicle nowadays, you have Ah you have so many complex complex artifacts. right You have multiple electronic com control units or ECUs which are kind of represent the computers which put we put inside cars nowadays. So, they have the software controllers running on them. right And you may have ah In modern vehicles you may have ECUs, GPUs plus graphics processing units, FPGAs. Not only that this multiple ah vehicle computers, we need to communicate among each other in real time and that would require some real time in intra vehicular communication protocol. For example, in this case, the most common protocol is the CAN protocol full form is Control Area Network. So, if you looked at the look at this figure which has been taken from a paper published in this area, some years back in ICCAD. So, you have multiple vehicles if I look at the look at a high level view. So, you have vehicles communicating among each other.

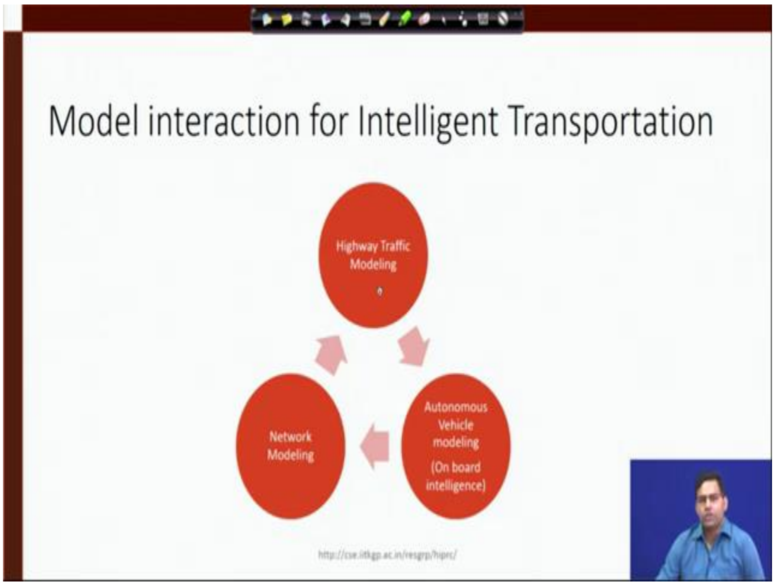
Vehicles communicating with roadside infrastructure. But once I pick up a vehicle and look inside it. There also we have significant amount of software implementations inside the vehicle. That these software implementations of different controllers. They actually, run on this set of heterogeneous vehicular computers like standard electronic control units. And accelerator cards like GPUs and FPGAs and all of them need to communicate using this real time communication buses like CAN, FlexRay etcetera.

(Refer Slide Time: 04:39)



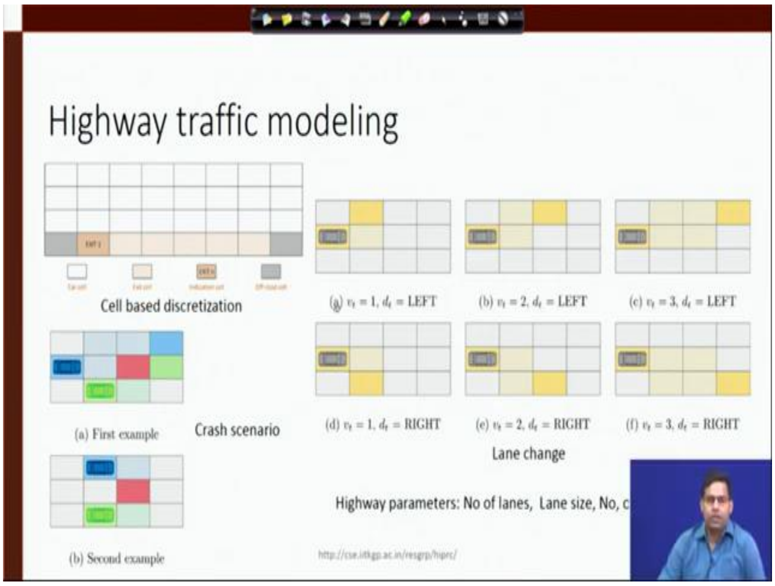
So, there there are a lot of interesting problems to solve there. ok So, if we look at the space of modelling connected vehicles, ah let us consider this simple problem that you have multiple vehicles. And you want to want to control their movement safely. So, the question that comes this given this kind of uncertainties that exist in the vehicle to vehicle communication as well as the uncertainty that exists in the vehicular traffic, what should be my autonomous driving strategy which is also safe? right Am I going to use some neural network strategy which uses things like recorded neural networks or deep queue, neural networks.

(Refer Slide Time: 05:30)



What am I really going to do? right So that is a difficult problem and it involves ah technologies coming from the AI domain, real time control domain, highway traffic modelling, vehicle to vehicle network modelling. So, all these domain specific technologies need to be considered together. Right

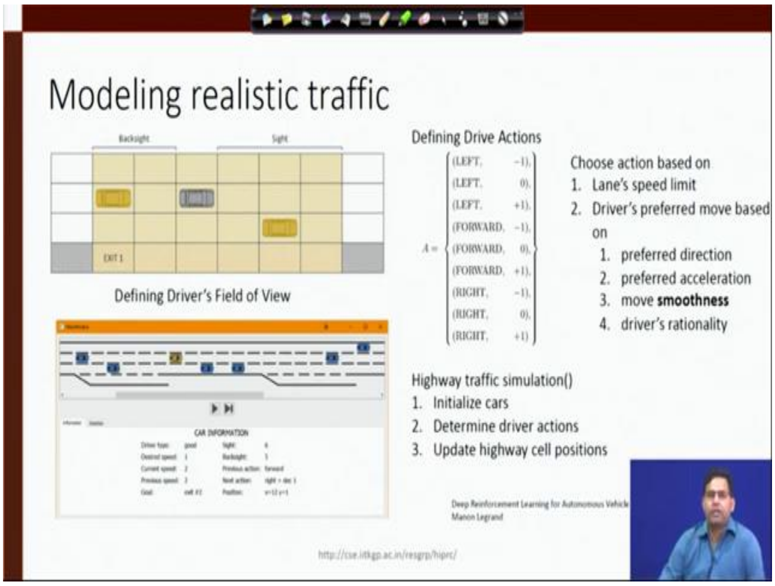
(Refer Slide Time: 05:52)



So that is again a very interdisciplinary system. And here we have a very high level, simplistic example again taken from existing works. So, one can say that well, the vehicles movements dictated by an AI based computer can be discretized that well. A vehicle changing from a central lane to left lane can be one discrete action. ah Vehicle changing central lane to right lane and at can be a discrete action.

Even the vehicle doing this maneuver at a low speed or a high speed, as you can see here that it goes just one step ahead, it goes two step ahead. This difference in speed can also be modelled as discrete actions. And accordingly, now if I can model this system of highways, vehicle movements, etcetera. ah In a discretized space and define such discrete actions then well I can have an ah ML based or AI based technology.

(Refer Slide Time: 06:56)

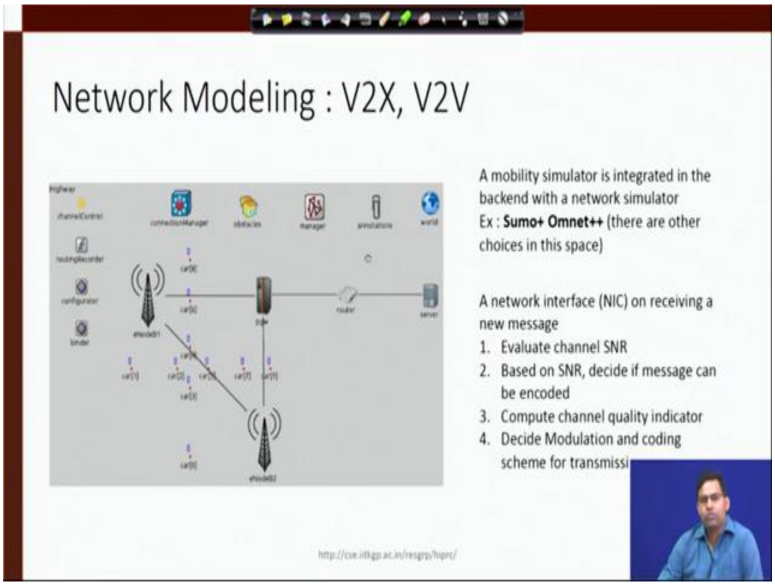


Which will decide that well which action is to be taken when by each vehicle. Who are coordinating among each other? So, this is like a simplistic view of autonomous driving but it is a nice starting point because you can actually, write a program. right You can actually, design an AI agent who can take this kind of discrete actions. You can write a very simplistic simulator, where kind of vehicle movements would be updated over the cells.

The cells which kind of partition the highway and show that well, a highway positions. Different positions of the vehicles can be either here or here but not somewhere here. right So that is the idea of discretizing that continuous flow into this kind of different possibilities. So, there is really an abstraction which we are creating which is simplifying the problem ah for our purpose and creating a toy example here. Right

But of course, ah in real world, if we are trying to implement such an autonomous driving strategy. It will be much more complex but here you what you have is a simplistic scenario which from a beginner's perspective, it may be easy to perceive.

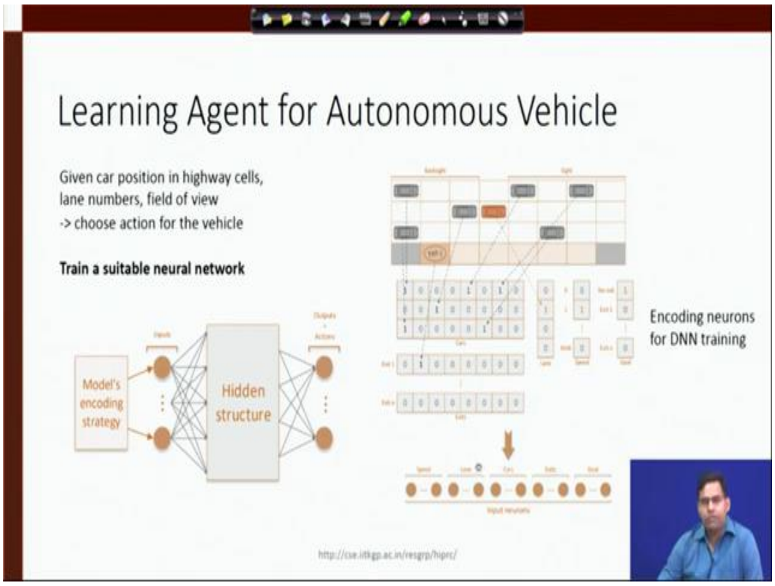
(Refer Slide Time: 08:16)



And again, like I have been saying that well is not only that you need to model the vehicular traffic. But you will also need to model the vehicle to vehicle actions that are there. And for that they are well known simulators like Sumo, Omnet plus plus etcetera. Using which you can do this Ah joint simulation right of the vehicular traffic, as well as the Omnet plus plus will give you the network traffic delay and other things. Right

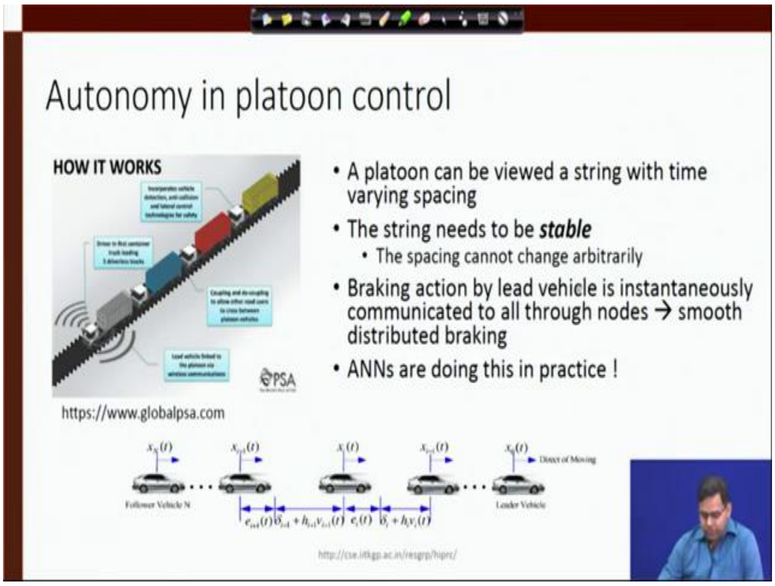
So that is a way in which you can again co-simulate things from different domains together and create a CPS simulation framework. ok You can design, ah the control the control strategies here ah in a simplistic way but of course we will see in our course. How to do that in a more real I mean near to real life involved way later on.

(Refer Slide Time: 09:10)



So, fine you can you can learn an AI agent which will take such discrete actions and do the vehicle driving for your purpose ok such things are possible.

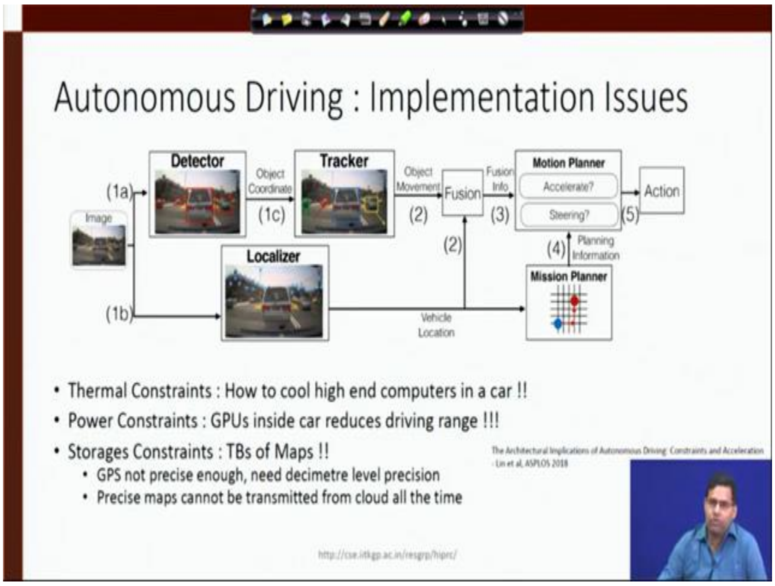
(Refer Slide Time: 09:23)



Now that was about having multiple vehicles coordinating their action. Now, there are some specific use cases, for example vehicle platooning. So, platooning means multiple vehicles are there but in a specific sequence, right in a linear sequence. And the idea of platooning I mean is appealing because if vehicles can move in a sequence like this. And the sequence is very regular. They maintain some desired distance among each other, etcetera, etcetera.

Then due to lack of air drag faced by the succeeding vehicles in the platoon, ah fuel efficiency becomes very high. So that is an economic solution. Right So, people around the world are trying to come up with intelligent, smart, platooning algorithms. I mean which which makes sense and they have been driving such truck platoons on freeways in many countries now by now.

(Refer Slide Time: 10:20)



So now, coming to an example of suppose, I am trying to implement such an autonomous driving system in real life. ah what What constants am I really missing here is? You should I Should I give give it a thought that what more technologies I need to be ah aware of, well a lot of them actually. right. So, if you think of doing an autonomous driving implementation. The first important thing is having a lot of sensors. Right

Because, if it is an autonomous driving vehicle, the perception of the real world has to be keep captured by sensors. And sensors can be a lot of them they can be camera type sensors, they can be radar type sensors, they can be lidar type sensors, etcetera. So, all these sensors would be trying to create a perception of the real world, detect obstacles, detect other things detect points of interest, detect the lead vehicle, detect the distance from the vehicle, etcetera, etcetera.

Detect road signs, detect pedestrians and all these things. right So, this common perception needs to be created. And that in itself is quite a compute intensive problem. right So, once that problem is solved. Then the next problem to solve is the motion plan. That how well, I understand I have a rough idea about ah or maybe an almost accurate idea about what is what is the world settings around me me being the vehicle? Right

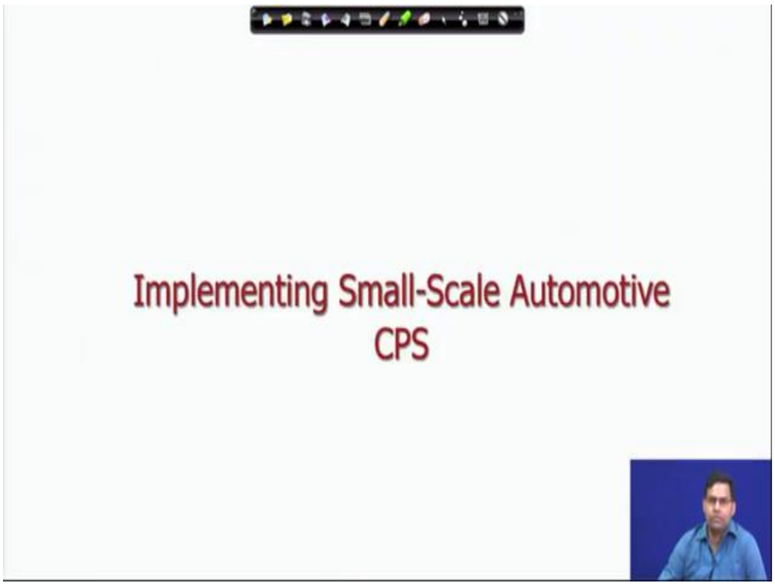
How do I plant my motion? So, there would be there should be a high level plan and a low level plan. The high level plan should have actions ah for for I mean which say that. Well, you move from this waypoint to the next waypoint. And maybe then the through the succeeding waypoint and the low level plant should have control control actions which help you into moving to the to the next waypoint.

And that is the control loop at the lower level which is working more frequently. Ok Now, as you can understand, creating that perception itself is quite compute intensive, like we just said. So that would mean that in future we will have accelerators like graphics processing units and FPGAs inside vehicles. right But this this is also at a cost. right We are looking for battery based vehicles.

Now that would mean the the the onboard energy that is there is coming from the vehicles battery also. But again, we are trying to run high-end computers on them. So, the battery drainage will also be happening. So, this also means all these compute I must be doing on the vehicle in real time inside task deadlines but at low power. right So, all these constraints need to be kept in mind while you have such a vehicle implementation.

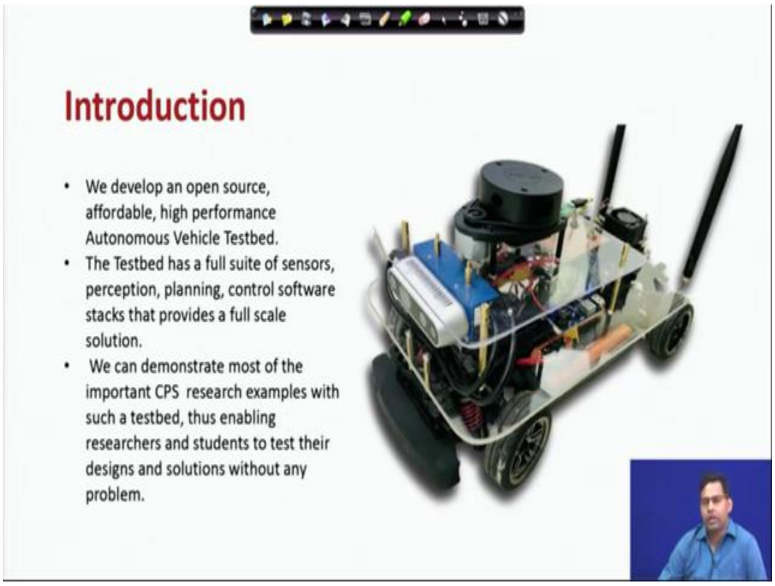
Where you have the sensors creating the perception and the control, ah computer in the vehicle computer. And all this happening that means the control loop in it is entirety. It is executive inside the real time constant, while consuming low power. So, this means there are quite a lot of challenges here. And people are trying to meet those challenges in different possible ways.

(Refer Slide Time: 13:36)



So, we will just talk about small example that suppose you are trying to create a lab scale Ah vehicle Ok with some amount of autonomy.

(Refer Slide Time: 13:49)

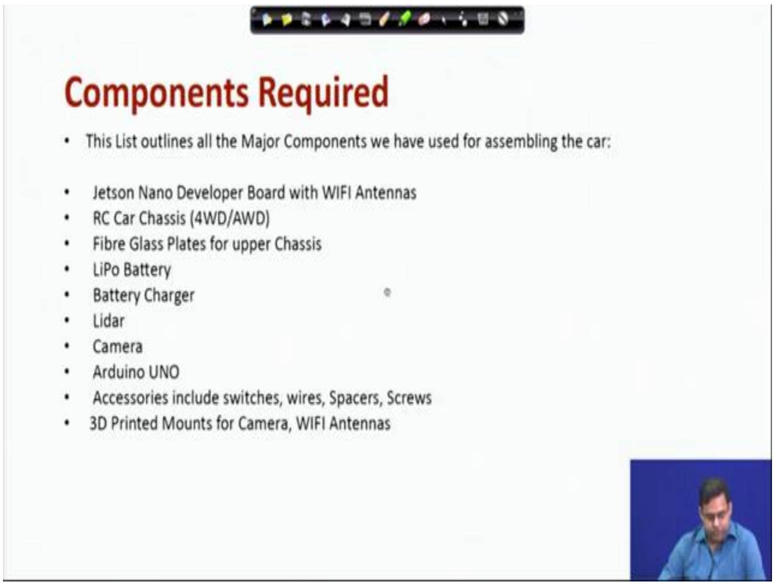


So, so this is an example that we have. This is a lab scale, autonomous vehicle that we created in our lab. So, the the reason I am showing this is this is an introductory lecture. ah You should be kind of enthusiastic You should I hope, you feel enthusiastic about creating such a testbed. And you can see that this is this looks like a toy thing. right But at the same time it it implements lot of real engineering principles.

And these are testbed which can be used for quite an amount of serious research. I mean research concerning development of ah control algorithms, research concerning real time Ah sensing and real time creation of the world perception. That means trying to recognize and understand. what What What is there around the vehicle? Because there are lot of sensors here and also doing the compute in real time using GPUs and CPUs which are on board in the system.

And trying to ah figure out that how the vehicle will move? That means executing the control laws. So, all these things can be done by this lab scale test vehicle that we have. And also there are wireless antennas through which the vehicle can actually, transmit commands, ah to a nearby vehicle which is which has been developed in a similar way. It can also receive commands from a central computer where we can actually, dictate that till the vehicle that you should do this or you should do that. Ok

(Refer Slide Time: 15:22)

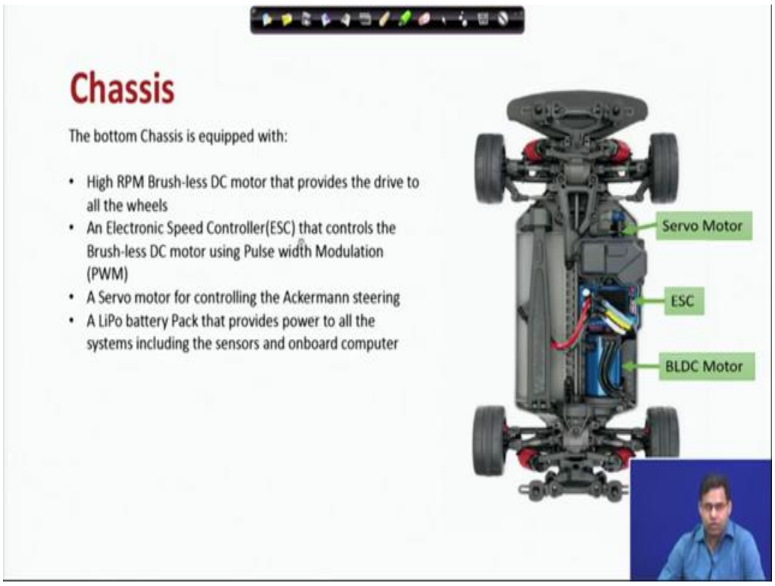


So, this is like a sample component list that we have. So, like you have you need a developer board. ah ah A developer board having both a CPU and a GPU will help. Because the GPU will be useful for accelerated sensor processing at low power. And well you will need some communication interface. And what is important is to get the car chassis. So, this is like one car chassis that we purchased.

And then we took out all the active elements of that. We just have these wheels and the base and the basic frame. And on that we did our development of all the custom components. right So, you have to have the battery battery charger for setting of the vehicle. And then you should be actually, putting in a set of ah set of sensors. For example, this vehicle has two important sensors. There is a depth sense camera and there is a lidar.

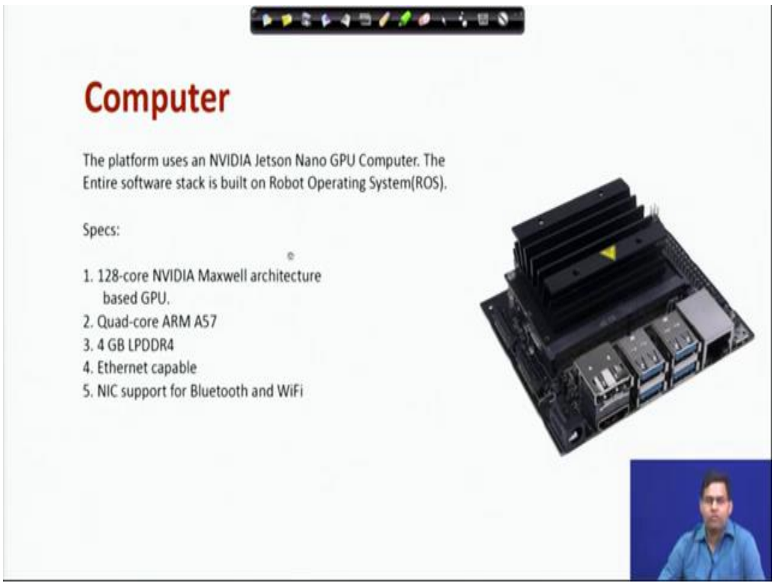
So, the lidar is continuously rotating and accordingly, it is doing a sensing that well. What all things are there around it? At what distance?

(Refer Slide Time: 16:34)



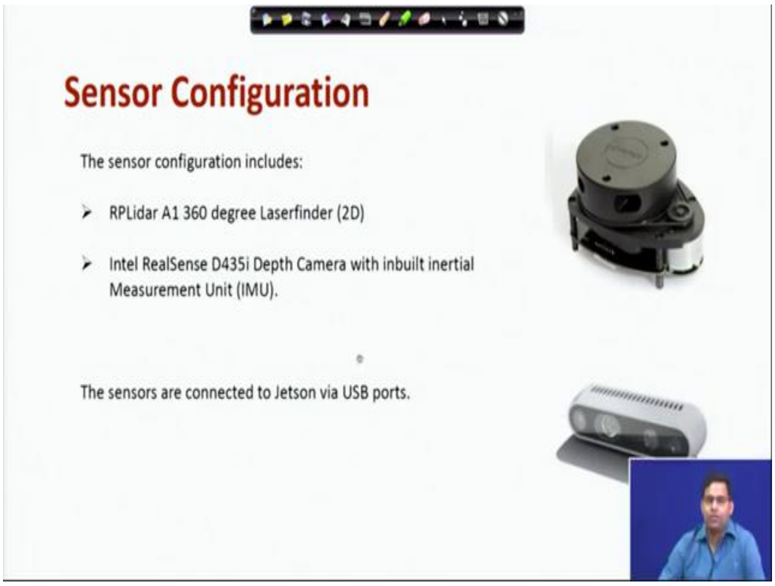
So, this is a view of the chassis. So, this came with a low level controller which is the speed control of the vehicle which you of course kept. And so, we will be designing algorithms which will be giving inputs to this speed controller. Ok And there is another motor which will be the survey motor which will be used to turn the front wheels. Ok And this controller is going to activate this BLDC motor is brushless DC motor which is going to provide the drive uh provide I mean which is finally, going to rotate the wheels. right So, basically, this is going to drive the vehicle. ok So, this mechanical component, we kept, we took out all the other electronic components. And we bro and we and and we created our implementation of them by using this of the shelf cameras, lidars, embedded processors, antennas, stuff etcetera. So, this is the chassis. So, like I have been saying that it involves this high RPM brushless DC motor. Then we have the speed controller. There is a lower level controller of the of this motor. Right

(Refer Slide Time: 17:51)



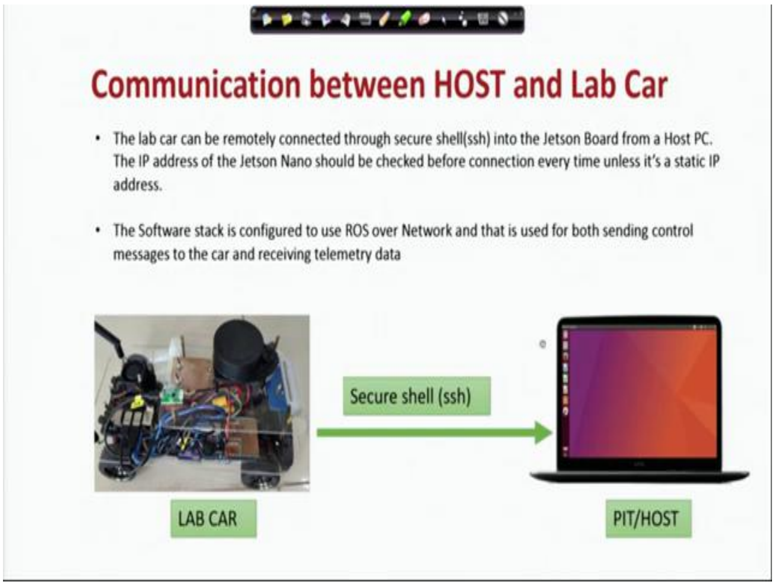
And so, this is the this Is the vehicle computer that we chose. So, we we put atop the vehicle, this Jetson Nano board. And you can actually, have a Robot Operating System that is ROS running on this board. ah Because that that is a very easy to learn operating system which is used in many of these kind of similar embedded platforms. Ok

(Refer Slide Time: 18:15)



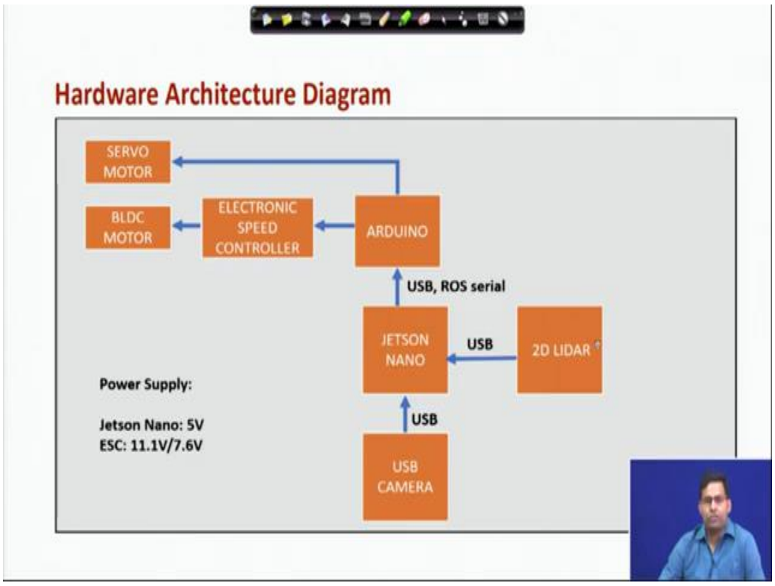
Now, these are the two sensors that we attach to the vehicle. One is the lidar and the other is the depth depth sensor. right It also has an inbuilt IMU. So that is another sensor. So, we will soon learn what these sensors are? How they work etcetera.

(Refer Slide Time: 18:31)



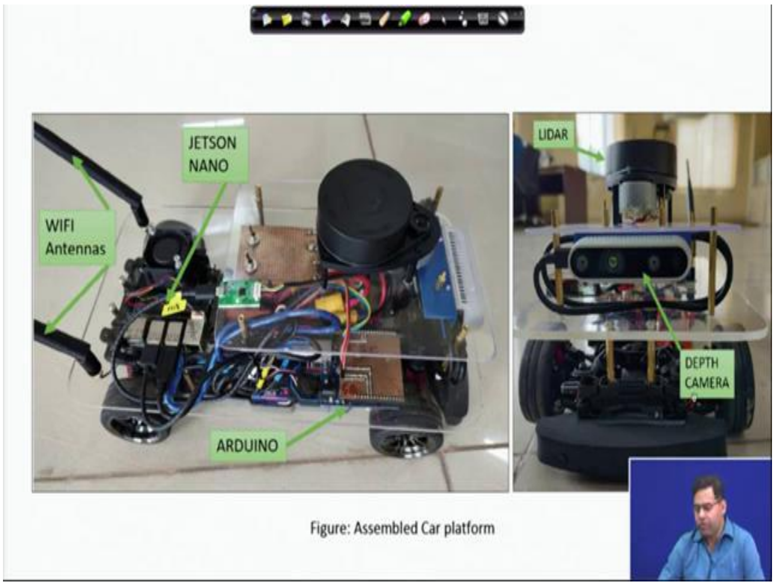
Now, using standard ah TCP based communication, you can establish ah this real time communication between this car and host PC. ok So, we use this secure shell based communication here.

(Refer Slide Time: 18:50)



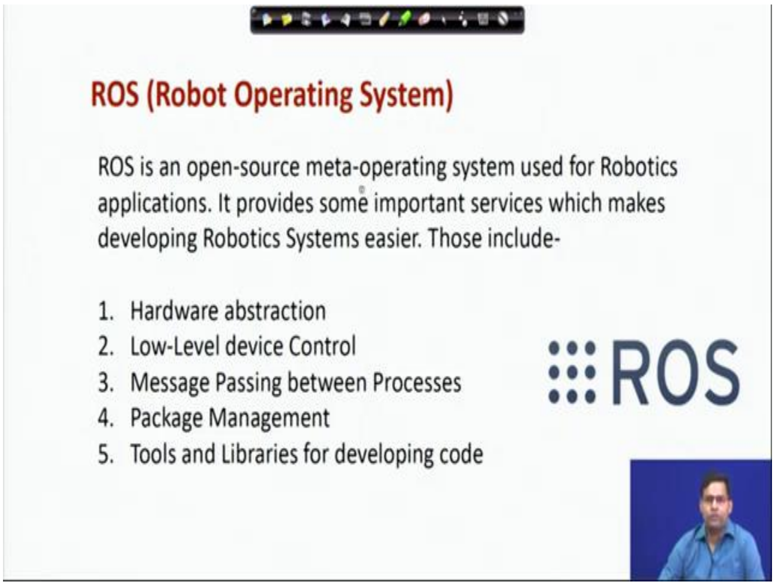
So, this is like the architecture that we are having here. So this part is they are inbuilt with the vehicle on along with this. We attached this arduino board which is having the lower level software which issues commands to the speed controller. And the steering commands are also issued from arduino to the servo motor. Now, this so This is actually, having the controllers. right The Jetson Nano board is running the deep learning pipelines which are processing the lidar and the camera data to create kind of ah real world perception of the vehicle.

(Refer Slide Time: 19:28)



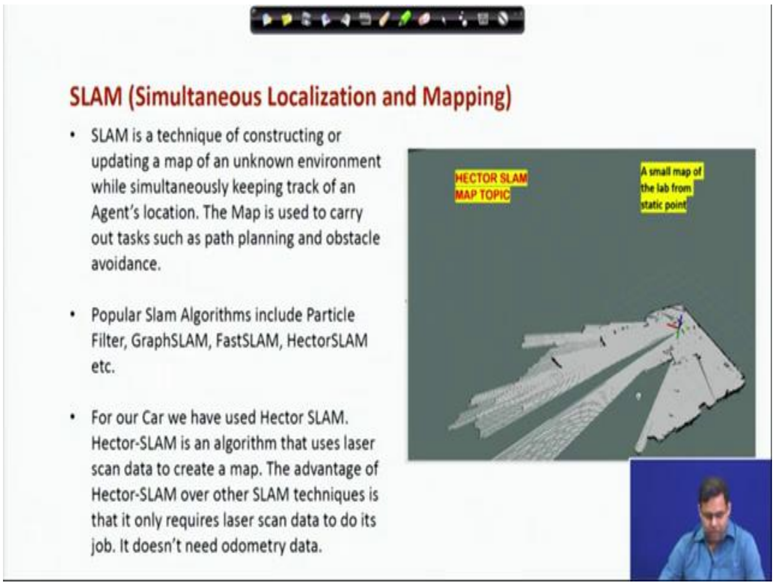
So that is how it is working and here we have again some top views and front views of the vehicle from our lab. Ok

(Refer Slide Time: 19:36)



So, just a few points about the robot operating system. So, because if you are doing such DIY projects this is a very ah standard OS which you may have heard of and if not. It would be a good time to delve into what is ROS and how to use it? Because in many such ah simple simple CPU projects this is a very popular OS to be used. It is a lightweight one and it can be used in embedded processors like this.

(Refer Slide Time: 20:03)

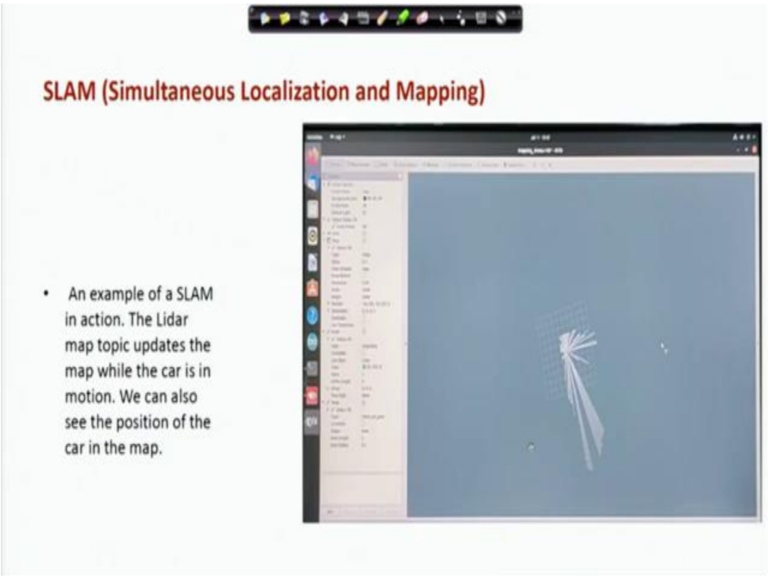


Now, we are using ah this simultaneous localization and mapping techniques. So, this SLAM libraries are also popularly available. And they can be used to create you to create this kind of maps using the lidar data. So, as you can see that as the vehicle is moving around it is it is creating this map. Based on it is obstacle sensing of the lidar. And this is happening using this HectorSLAM algorithm OK which is which is quiet well known.

So, this idea of localizing the vehicle and it is position inside a bigger space. ok ah This is ah I mean in general known where this common name that is simultaneous localization and mapping or SLAM. So, it is this technique of updating the map right and create creating this perception perceptor sense that will where am I and how exactly things are located around me? And this map will be of course, used for further path planning and vehicle movement. Right

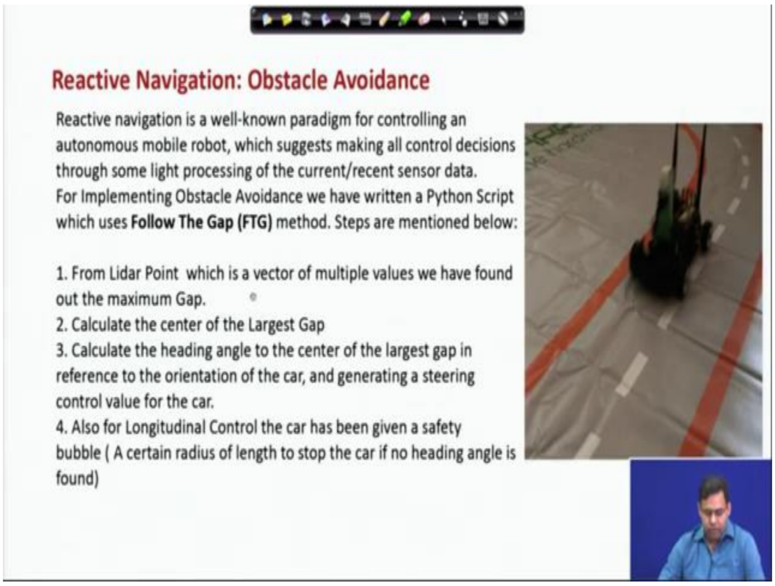
And there are a host of ah host of well-known algorithms in in this domain of SLAM. For example, HectorSLAM, FastSLAM etcetera. And in case of us we used HectorSLAM to create this map here. Ok

(Refer Slide Time: 21:27)



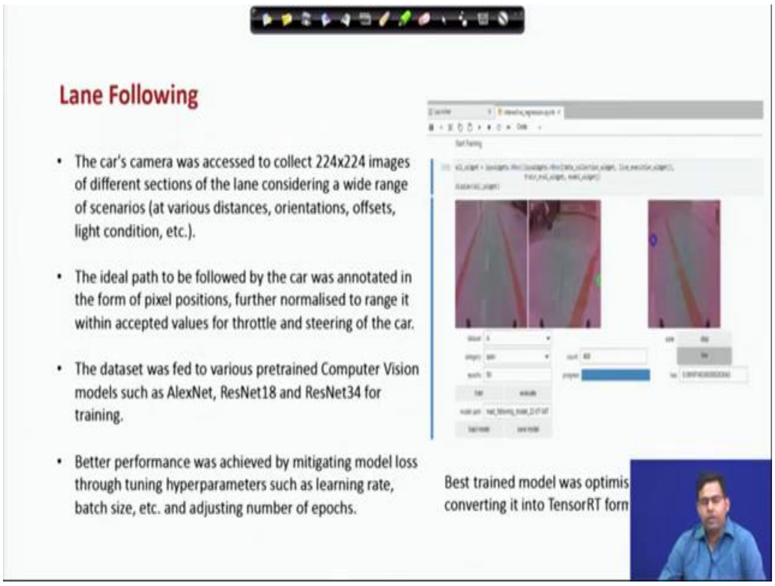
So, a few more examples, like if a vehicle is moving that how this SLAM algorithm is working in real time and creating this map?

(Refer Slide Time: 21:37)



Now, for the vehicle, you can create some obstacle avoidance kind of algorithm. Right

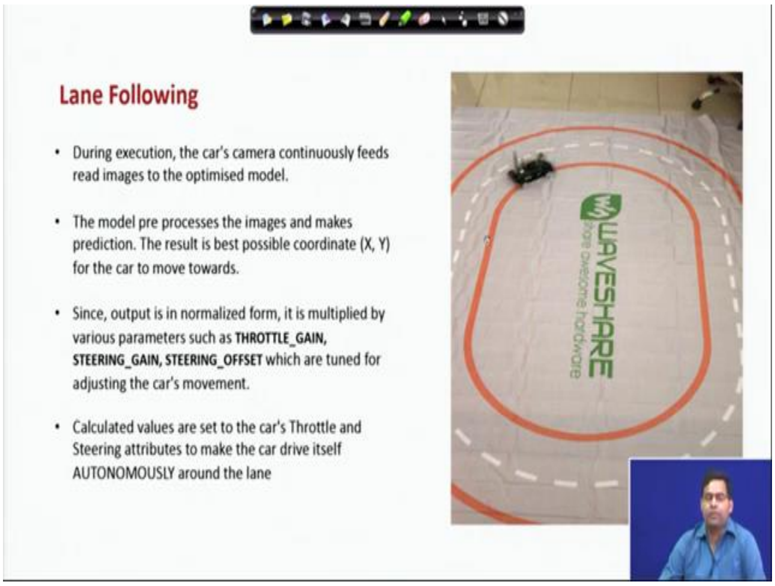
(Refer Slide Time: 21:45)



So, once you have this vehicle hardware ready, you can write some programs for maneuvering the vehicle. So, if your sensory inputs are processed by suitable deep learning pipelines on the GPU boards then based on that you can detect obstacles and accordingly, you can maneuver your vehicles. Right So, you can maneuver your vehicle suitably through waypoints, ah make it avoid obstacles make it follow lane markers.

So, there are lot of opportunities which can be explored. Right I mean and you can do a lot in that way. ok

(Refer Slide Time: 22:22)

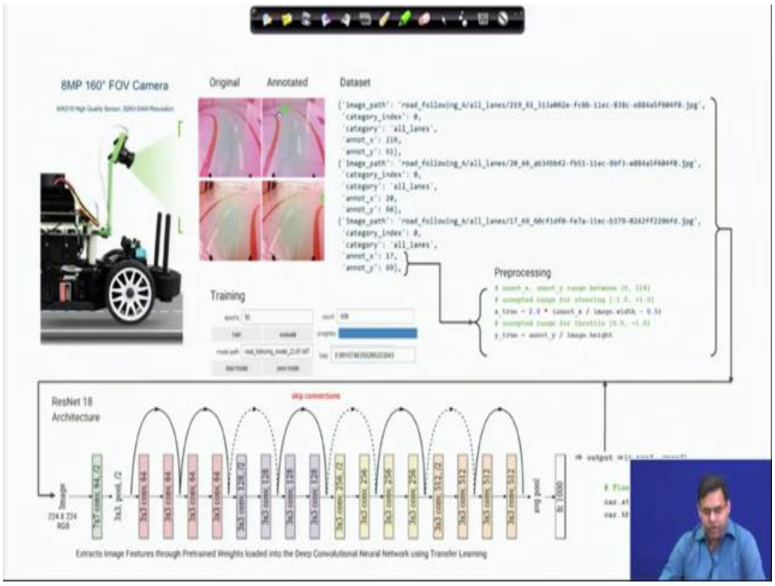


So, this is just another sample vehicle. So, this we we actually, purchase this is not our Ah I mean the one that we are building and we are testing our algorithms on this is another vehicle. So, we are just throwing a screenshot of that here. It is available to use, I mean and it can be used to demonstrate ah basic movements, basic obstacle avoidance and basic lane following control.

So, when the vehicle moves, it can actually, detect this lane markers and and the suitable manipulation commands are ah executed in the vehicle's computer. And the vehicle is instructed to follow the lane. Do not not wander around to some other lane or or to avoid those kind of stuff. ok So, such such algorithms can be implemented and they form very nice use cases of lab scale autonomous driving. ah Labs

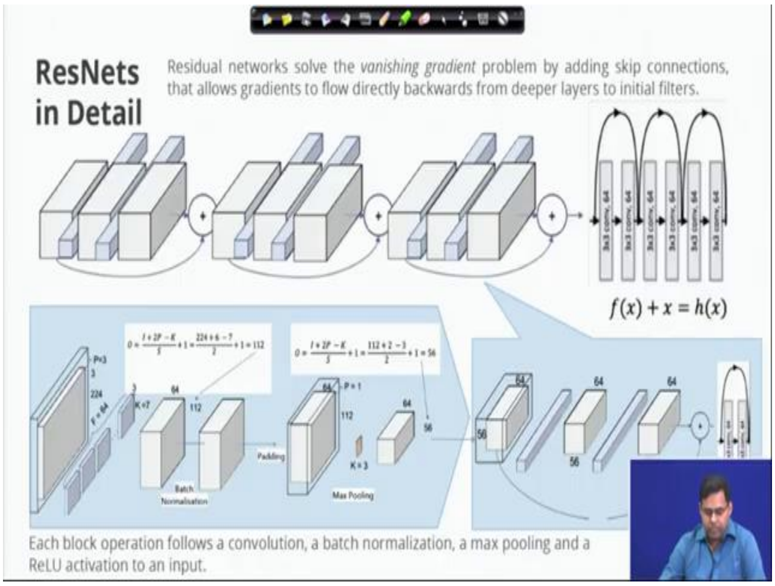
Or you may be also interested in implementing softwares which are ah which are used for not full autonomous movement. But ah kind of aiding the driver for example, a parking software, some so some assisted parking. You can make the vehicle park at a suitable space using some software logic. ah If you build this kind of lab scale, autonomous vehicles or semi autonomous vehicles you can also develop such controls real time in control algorithms and the corresponding softwares and test them. Ok

(Refer Slide Time: 23:50)



So, well this is an example of the popular ResNet architecture and how that was ported in this commercially available vehicle example. ah The architecture of it and how it works? How it can detect the lens? So, as you can see, there are some screenshots where this vehicles camera is actually, able to detect lens. And accordingly, the vehicle maneuvers around the lane markers. Ok

(Refer Slide Time: 24:14)



So, there is some details about this deep learning architecture which is deployed on the vehicle. So, fine, ah with this we will come to the end of this lecture. And in the next lecture we will also talk about some other important aspects of automotive cyber physical systems. For example, how real time simulation of automotive CPS happen. Thank you for your attention.

Foundation on Cyber Physical System
Foundations of Cyber Physical Systems
Prof. Soumyajit Dey
Department of Computer Science and Engineering
Indian Institute of Technology – Kharagpur

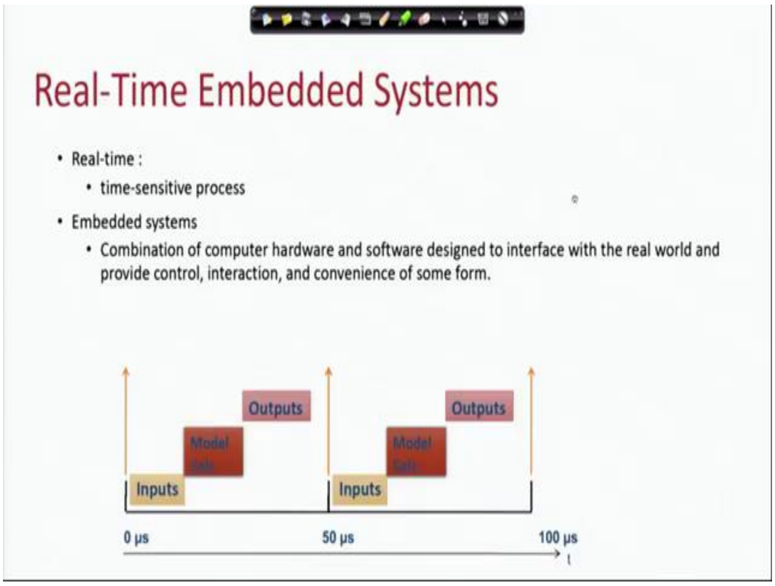
Lecture – 03
CPS : Motivational Examples and Compute Platforms (Continued)

Hello and welcome back to our lectures on foundations of cyber physical system.

(Refer Slide Time: 00:32)

So, in this lecture we will be ah focusing on ah Real-Time Simulation of Automotive Cyber Physical Systems.

(Refer Slide Time: 00:40)



So, what does it really mean first of all? Well, there is simulation but what is real-time simulation? So, let let us understand these fundamental characteristics of embedded systems that they need to execute inside some given deadline. If you are executing a standard software in a desktop system ah there is no real-time deadline for most cases right but you are executing software in an embedded system.

In most cases, it is supposed to execute inside a deadline. So, typically, if I look at a control system, the controller is a software which will get some input from the plant. Then it will execute some control law and then the control actuation commands must be actuating the plant inside the sampling period of the controller. Why? Because in the next period the controller is going to again sample new values and it is going to observe well the control command I gave earlier.

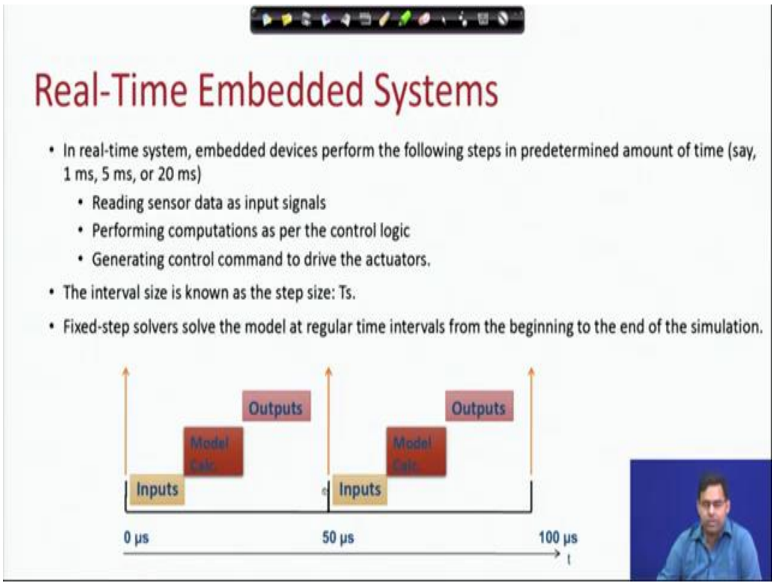
Due to that how did the plant change it is state? ok So, ah whenever I am implementing such a cyber physical system or I I just given control theory example but in most standard CPS examples, we will have this thing that well. There Thus, the the the the software is, what we call as reactive. In nature, reactive means that well, it will do something ah based on the environmental situations it will react to the environment.

Because it has a target objective. Let us say driving the vehicle in such a way that I maintain a distance of 5 meters from the vehicle in front. So that is a control objective and I react to the scenario that right now, this distance is off from 5 meters but by 1 or 2 meter. Then, accordingly, there is a trigger for me and I will work ok so that is one way of implementation. Or the reactiveness can also come from the sampling period that every after every fixed period of let us say 10 milliseconds an interrupt is generated and that is dictating this control software to wake up, execute, take some decision and then again wait for the next event to be generated. And the next event will come on the expiry of the timer which is set for up to the sampling period and after that again the software will wake up and do something. right The other important thing is the control softwares are executed in feedback loop and as is common for most cyber physical softwares.

That means they will execute they will do something based on that the plant will evolve and by observing by how the plant is evolving they will again do something. So, whatever they do now is a function of what they did in many cases earlier in the many instances earlier. right So, coming back to this, this is a example of periodic task execution and the task is doing something based on inputs.

And it is outputting something and this is going on and on in multiple different periods consecutively. right And all these periods have their given deadlines in the system, ok

(Refer Slide Time: 03:55)



So, suppose I am trying to simulate such a real-time system. The question is when I execute my software, how do I really know that the software is really executing inside the actual time bound inside which it is supposed to execute? Well, you see if I am doing this experiment on a desktop computer running, let us say windows or Linux the operating system is not making the processor of the system always available to me. Right

Because there are other processors and I am running my control software as one process. So then, ah I will be getting ah access to the CPU to execute this specific piece of code only in some those time slices when the operating system gives me access. And in those time slices where the operating system is giving the access to some other process I am not really having the CPU right, so that is the problem.

Now, the question is ah even if that is not the case, how do I know that the computer is executing my process exactly in real-time? Because it may have it is own logical notion of time attached to that process. ok So, what we really need for evaluating embedded control systems? Is that we will need a way to measure real-time and we will need the events produced by the software to have an to happen ah inside those real-time boundaries. Ok

Now, this problem is further complicated by the requirement that let us say there is some physical system which I am also modelling. And my control input goes to the physical system and now I am supposed to simulate that physical system up to some time and then observe the output and then accordingly again actuate the control input. right So that means the physical system has a model.

Let us say in case of vehicle, there is a vehicle dynamics model that model has to execute inside that real time. Right And it should, I should, I should then get the output ah which is let us say the current state of the model and I will use it in my some embedded processor to execute the control logic. And then inside my deadline I should give back this control command to the computer, where the model is simulating.

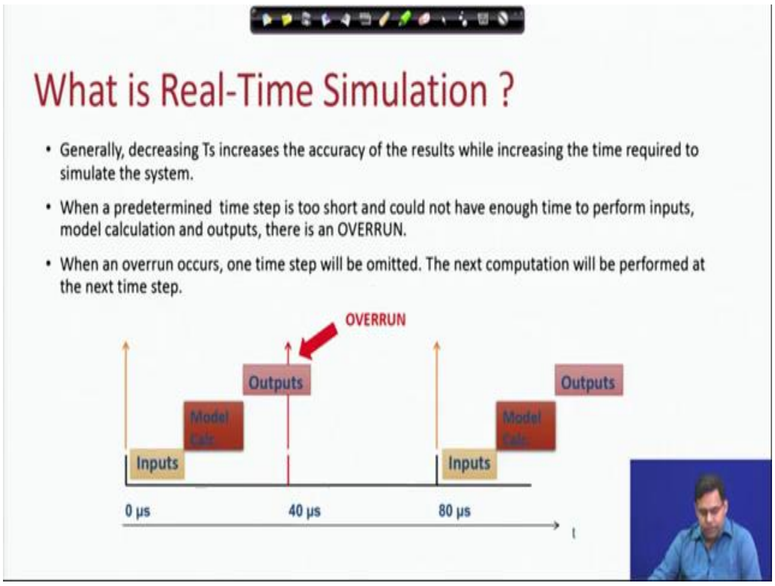
Now, like I said that the model simulation time should be faithful with the real clock. right It should not be like a process like a desktop computer, where this code is running along with other codes and I really do not know that well when the model is producing its output. What is the real-time value? Whether it is as per the real-time requirement here. right So, for this we use what we call as hardware in the loop real-time simulators.

The other important thing that we have in this slide is when I am solving that model so, fundamental is a set of dynamical equations and it is a set of difference, differential equations and they need to be solved. right So, there would be a solver in the back end and it is a continuous time thing. right So but I will have to choose a discrete time step internally, after which those solution values are available.

If that is happening at a very fine granularity then the time taken by the solver to give me the solution of the model at the end of the sampling period. That will be too high and it may be more than the more than the sampling period of the system itself. right So, the solver needs to choose suitable step size so that the number of times the solver has to work inside the sampling period is manageable inside the real-time deadline.

So, this is an involved thing, we will come back to this. We are just trying to enforce upon you that when I am modelling reals, embedded processors, I am running some embedded software and I am trying to see how it works with the plant model. The plant model has to run in real-time and that output has to come to the embedded processor and the processor should run its code. And again, give back the commands to the plant model in real-time.

(Refer Slide Time: 08:08)

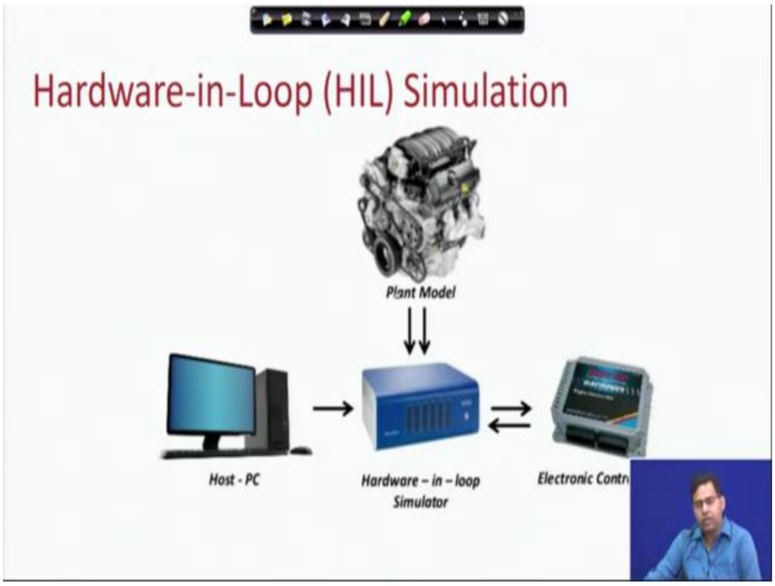


And if that does not happen, this is what it going going to occur. The output computation will be delayed and there will be some time over run.

(Refer Slide Time: 08:18)

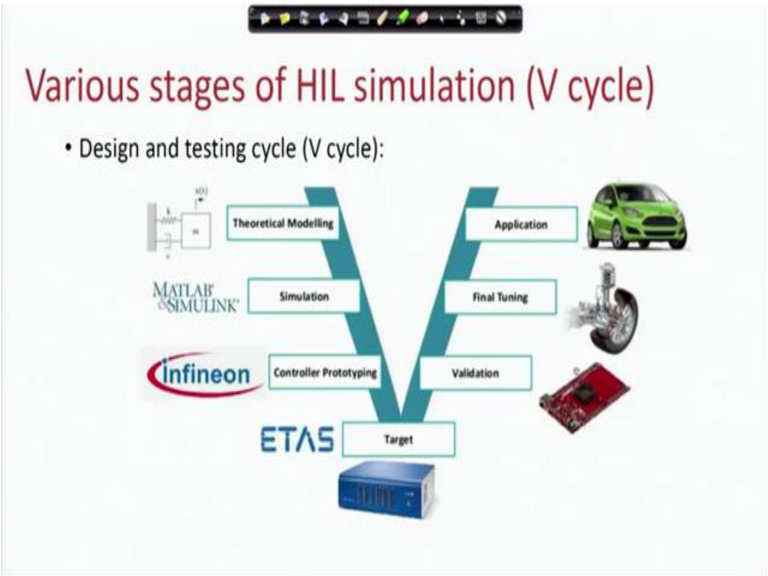
And so, how is all this thing done? So, typically for that companies provide you with things called hardware in the loop real-time simulators. For example, here is one example system that is just shown and there are things coming from various other companies also. right So, you create a model and you can run in this kind of computer. What it will do is, unlike a normal computer, it will show it will show you the model updates happening as per real-time clock. Ok

(Refer Slide Time: 08:48)



So, any kind of physics or any kind of physics based model or that means which replicates a physical scenario can be run in this kind of a hardware in the loop simulator. And then you can see the output in the host PC of the system and it can be connected with the embedded processor, where you have the corresponding software logic running. Ok

(Refer Slide Time: 09:11)

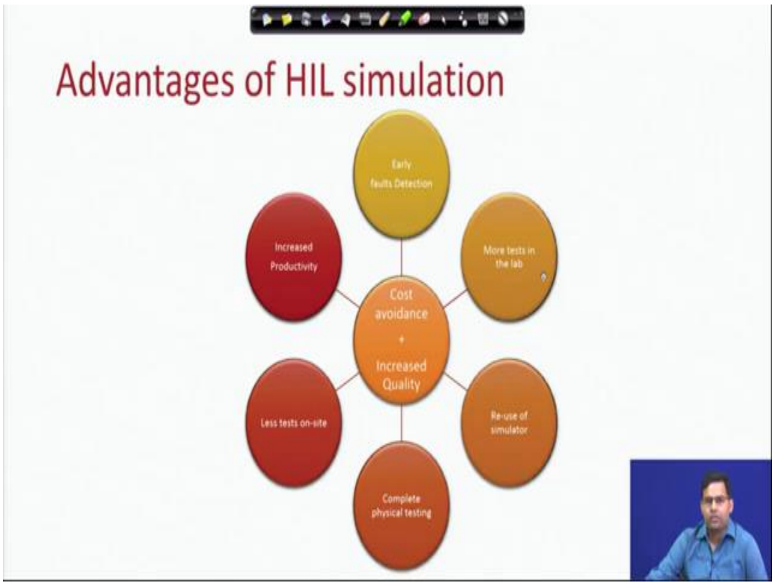


So, this um thing that is ah I mean testing a system can be done at various levels in a in an automotive space. right And this is typically the V cycle of design and testing. So, you can have theoretical modelling then you can have simulation in Matlab and then you can design the controller and prototype it in a embedded software processor board. Then you can attach it with the target system ah which is basically a model of the actual physics physical system and that model is running in a real time computer.

And then you can do validation, then you can do final tuning or maybe you can attach ah this processors and it is output to the sub part of the physical plant. And then you can do some so, you can do validation at this level by attaching this software ah running on the processor, with the model running on the hardware in the loop simulator. And make them simulate together, then you can go to this level where you can attach the sub part of the big physical thing and see that how it is behaving.

And then you can go to the actual application that you have the entire thing may be a car and you bring in the software here you attach that board here and see that well how this is driving around and the required functionalities are working in some test track or test scenario. Ok

(Refer Slide Time: 10:39)

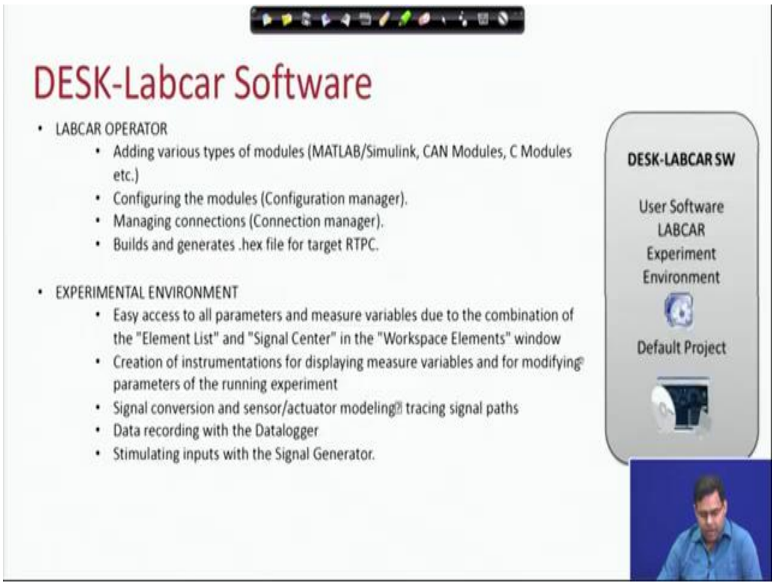


So that is how it works usually so, of course, ah you it may be difficult to get the real system. right So, anyway the preferred mode ah would be to use this kind of a simulator and make your software run in the loop with the simulated model of the plant instead of having the real plant. OK Because of this various advantages that you have with respect, to cost, quality, less number of tests to be done on site, etcetera, etcetera.

(Refer Slide Time: 11:11)

So, we we are just giving some names of some simulators that can be used and there are free like (11:18) I have been saying that there are many other offerings from many companies. So, this is a combination that used to work in our lab. So, you can have ah simulate that HIL simulator, like Labcar and along with that you can have any of your favourite embedded processors ah which can run your control software. Right

(Refer Slide Time: 11:37)

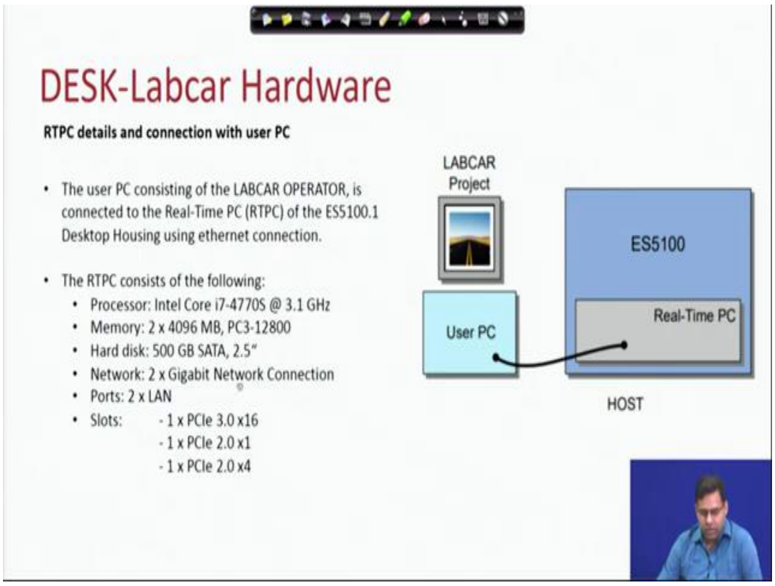


And you can just configure them suitably.

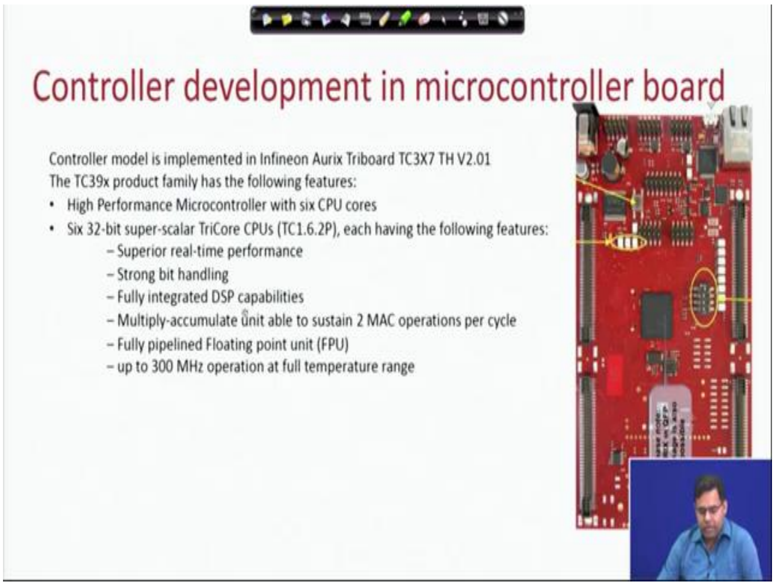
(Refer Slide Time: 11:41)

And you can connect them together. So, this is just like a broader screenshot of how an entire system looks. So, at the back there will be lot of ports just like a computer but there are many additional ports also to interface with your embedded processor.

(Refer Slide Time: 11:57)



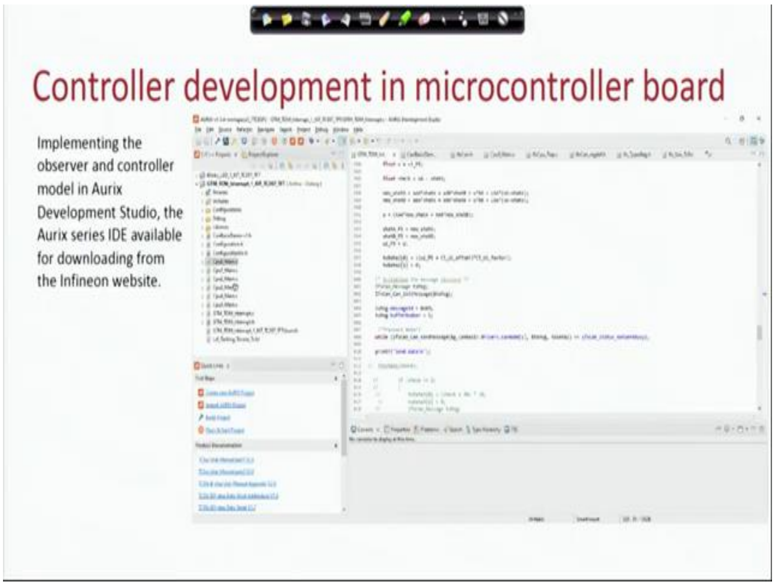
(Refer Slide Time: 12:00)



And then ah well you can you can just interface it with some embedded processor like this. So, this is an example microcontroller board we are showing is infineon boards are typically used in automotive systems along with many other kinds of boards. So, this a typical board will have some multiple one or multiple CPU cores. For example, this specific board has six CPU cores.

That would mean that you can really run programs compiled to these boards in each of this course simultaneously. Right

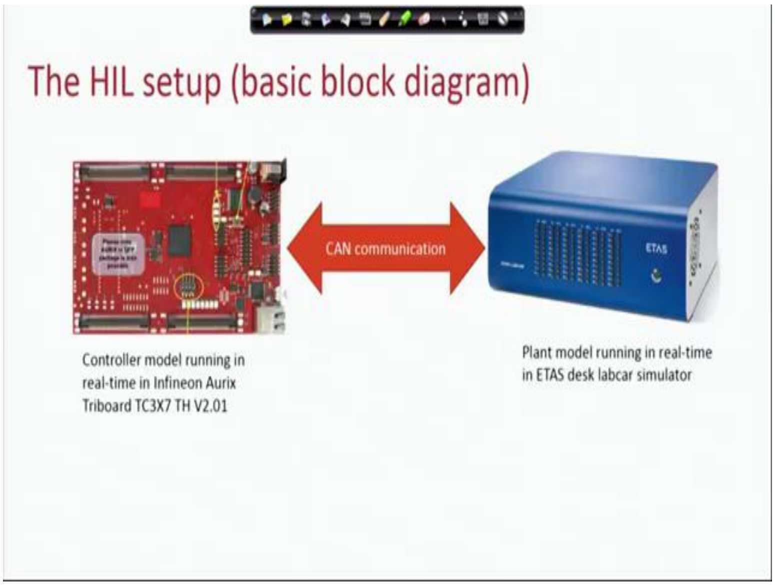
(Refer Slide Time: 12:28)



And this is like a screenshot of the IDE, the Infineon IDE, the Aurix IDE through which you can write your control software, the actual control software which will sends messages which will execute some control logic. And then which will which will send some control actions over any of the interfaces that this boards provide. ok It would provide a typically it would provide a CAN interface through which you can connect these boards output to a CAN cable.

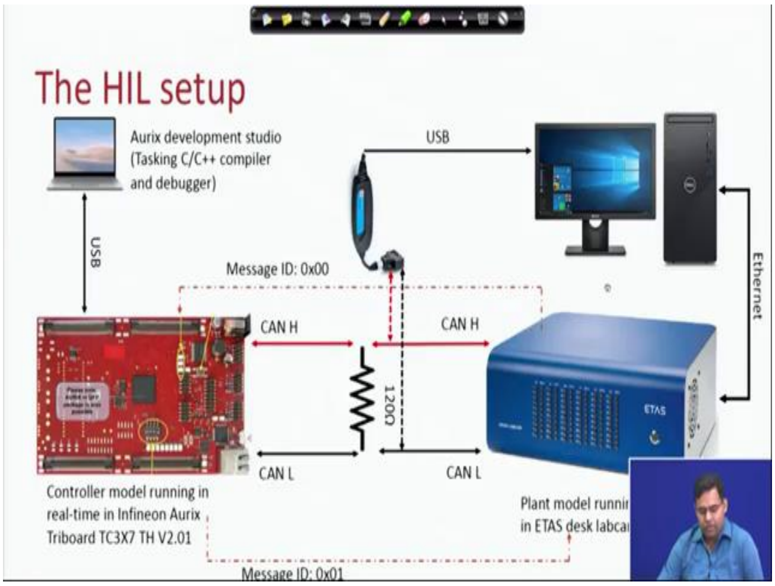
And that CAN cable will you can attach to the HIL system to see that well how the plant reacts to the control actions that are communicated here. Ok

(Refer Slide Time: 13:06)



So, yeah this is the example picture that we are talking about. You can have your control communication you can make the plant model ah evolve and you can watch it is trajectory.

(Refer Slide Time: 13:21)



Now so, this is a typical HIL setup, so, the CAN is basically a set of twisted pair wires with suitable resistance. Ok Now, you can also log the CAN data in another PC and you can see that well what kind of data packets are going because your vehicle sends values from the vehicle.

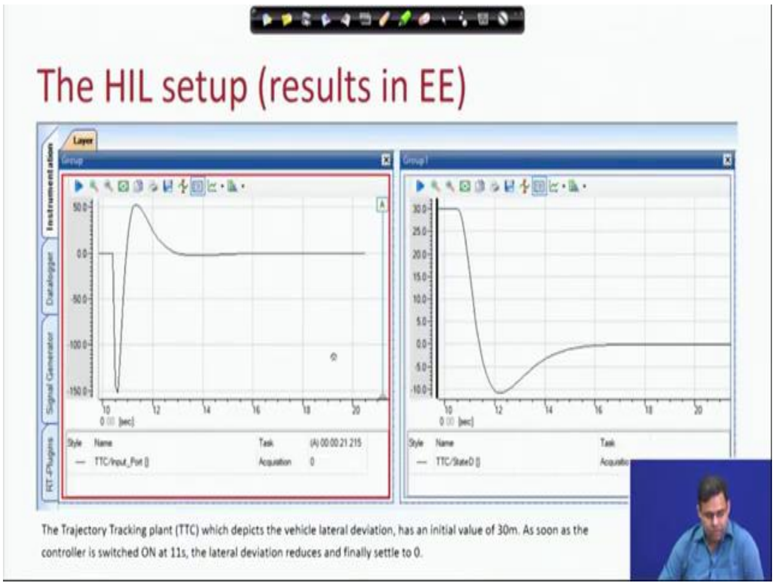
Let us say let us take an example I am modelling a vehicle dynamics. I am having wheel speed and linear acceleration. ok ah

And let us say some more things like the vehicle's steering angle, I am monitoring this signals and I am sensing those signals and sending through the CAN bus to the ECU. The issue is executing some controller. Let us say the adaptive cruise controller, let us say the ABS controller and those control commands go back to the con through the CAN bus through the plant model which is evolving here. right Ah

So, let us say this controller wakes up once every 10 milliseconds. That means I will be computing the dynamics of that plant. ah I want the output of the dynamics once every 10 millisecond and that should be communicated back here. Now, when I evolve those continuous dynamics inside 10 milliseconds this system would choose a suitable step, such of the solver that whether it will solve the dynamics once after the 10 millisecond or multiple rounds in between for increased accuracy. Ok

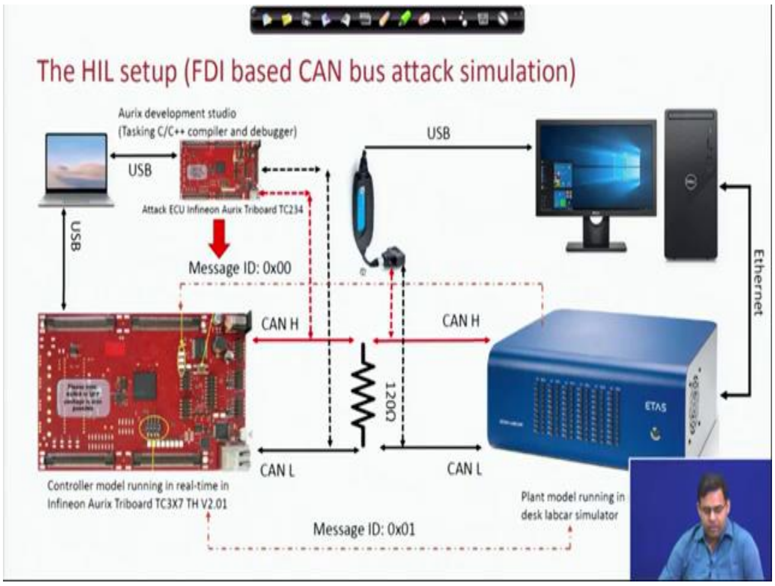
And then once the those equations are solved here right and you get back here. right So that simulation is happening here in real-time because this is a at a real time, computer here. And so that means once that is done, you will get those changed values and then you will be computing ah the control signal. And you will be sending that back through the CAN bus. And then with some modified steering angle or modified breaking value or modified acceleration value those are the discrete actions that your controller to here plant model will again evolve right.

(Refer Slide Time: 15:29)



So that is how a typical HIL setup will work and these are some examples. Let us say you take a trajectory tracking plant. So, essentially, you are following a previous vehicle. And so, there was some lane deviation initially by the previous vehicle and you are slowly collecting the deviation. So, finally, this deviation goes to 0. right So that is a typical example we are trying to show again not going to the details here.

(Refer Slide Time: 15:51)

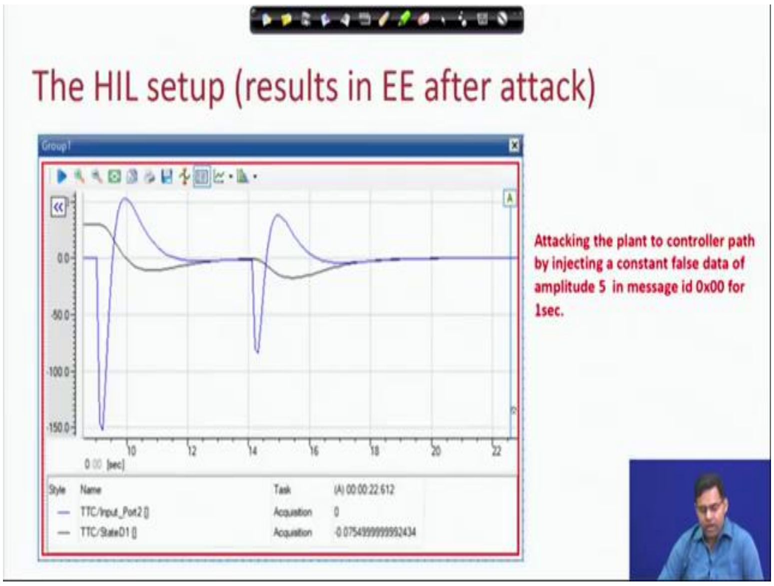


So, this kind of a testbed has more usage. Suppose I am trying to do some experiment on automotive security which is nowadays quite a hot topic. right Suppose I have an idea that the vehicle can be attacked by ah at attacked over the internet or through some other surface. The vehicle can be attacked and communication between the plant and the controller that is the vehicle and let us say one of his ECUs has gone corrupt.

So, the CAN data packets, they are getting transmitted, they may be spiked by the attacker and I want to see that well, how is that going to affect my system? So, let us say I have an attacker ECU whose whole job is to spike these messages with some. ah in some way that means it would, it would do something it would do something to make the vehicle model think that well this is not the ECU which is supposed to send the actual CAN messages.

So, it will send this ECU to an off mode and this attacker would start replicating this ECU's role. And it would start sending spiked messages. ok So that is an example of how you can create a test bed where you are able to replicate automotive security experiments.

(Refer Slide Time: 17:09)



Now, if you do that you may see how things change maybe you can see that well due to such attack deviations start happening. And that would mean the trajectory tracking you are doing that you are following a vehicle and the the set deviation you are supposed to have from the previous vehicle that keeps on altering. ok So, those things can happen and this is the way that you can have a testbed through which you can actually see such experiments carried out.

You can actually, see that let us say you are developing a security scheme and if you prevent that CPU scheme here in this victim ECU. You can actually, figure out that well, this victim ECU able to work or not. And what is the effect of that attack or what is the effect of that counter measure on the plant dynamics? So, this is. This is the important thing here. right We are talking about cyber physical systems.

So, even if you have a security measure, we will like to know what is the effect on the physical dynamics? So that is why you will need the software logic. You will need some safeguard and security but final validation would require the physical model of the system. And you will like to see that how the model evolves and well in case I have taken a security measure if that have some effect on the physical dynamics of the system. ok

So that is how all these concepts get link up and you can get to do some interesting thing on CPS security here. So, with this interesting example, we will be ending our lecture. Thank you for your attention.