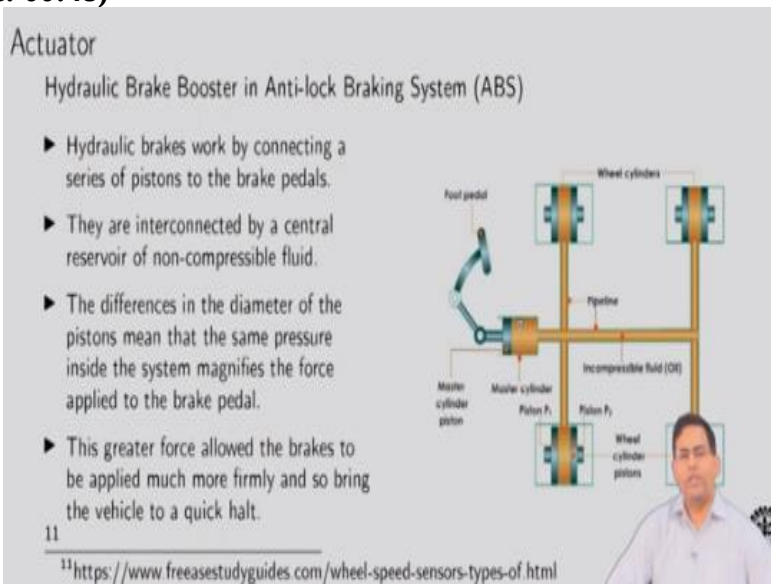
Foundation of Cyber Physical System
Prof. Soumyajit Dey
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Module No # 02
Lecture No # 07
Real Time Sensing and Communication for CPS

Welcome to the lectures of this course on foundations of Cyber Physical Systems. So, I believe in the last lecture we have been discussing about different kind of sensors and the physics behind how those sensors operate of course at a very high level.

(Refer Slide Time: 00:48)



So, continue from there. So, here is just a simple example of an actuator. So, it is basically the hydraulic brake booster which is very much standard in antilock braking systems in automotive. Now all of us are quite aware of the physics principles how I mean in a hydraulic system you have liquids which can amplify the pressure right. So, this breaking principle of the actuator is primarily based on that.

So here of course we are not showing the electronics involved around this kind of a system and this is a very high level picture. Of course, you can understand that this foot pedal I mean the application of the final break will be controlled by anti-lock braking software. So, it is not going to be a direct connection like this. But this is the very simplistic example here. So, we are just trying to have a high level view here, right. So, how a hydraulic brake works?

I think it is now very much evident from this elaborated picture here. So as you can see that all these 4 wheels of a vehicle they would be connected by these pipes and these brake pipes will be containing an incompressible liquid here, right. And that liquid network is having a master cylinder on which the piston coming from the foot pedal will be connecting. Now I will just repeat that. Of course, in the age of software where you have anti-lock braking controller.

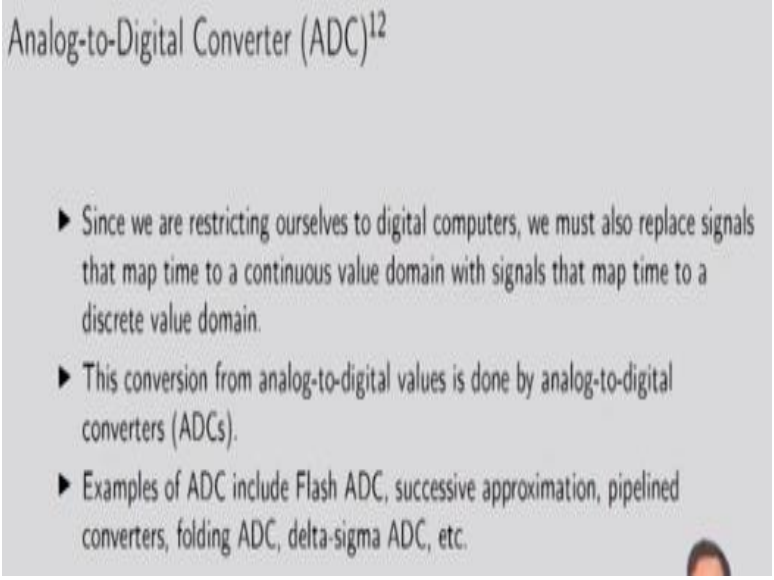
This is not going to be a direct connection like this. So, what happens is in this simplistic example when you put some pressure on the pedal, that pressure will be kind of moving on to the liquid, right. And that same pressure inside the system will be magnified quite a bit when it actually gets applied in the brake pads. So, these amplifications will actually make your car to halt and you can understand that the actual amount of force required for bringing the vehicle to halt is much more than you are applying physically here.

And this amplification is happening due to this hydraulic system that you have. So, essentially that is how the physics behind the actuator works. Now, in a software control system what would happen is this physical actuator, its movement, when it will engage, when it will disengage those things will actually be controlled by suitable software control signals and that is how an anti-lock brake works.

That it decides that how many times to break and then engage and disengage this master cylinder piston so that the vehicle does not sleep over tricky surfaces. So, again this is any high-level view, but it provides you with an accurate enough example of how an actuator works. Because you can understand that this would really mean that the real pressure on which this system can operate that will have a lower bound and upper bound, okay.

And there would also mean that the software signal that should come in a real-life ABS that should also have lower bounds and upper bounds and that will be the variable bounds inside which a practical brake pressure should be available.

(Refer Slide Time: 04:04)

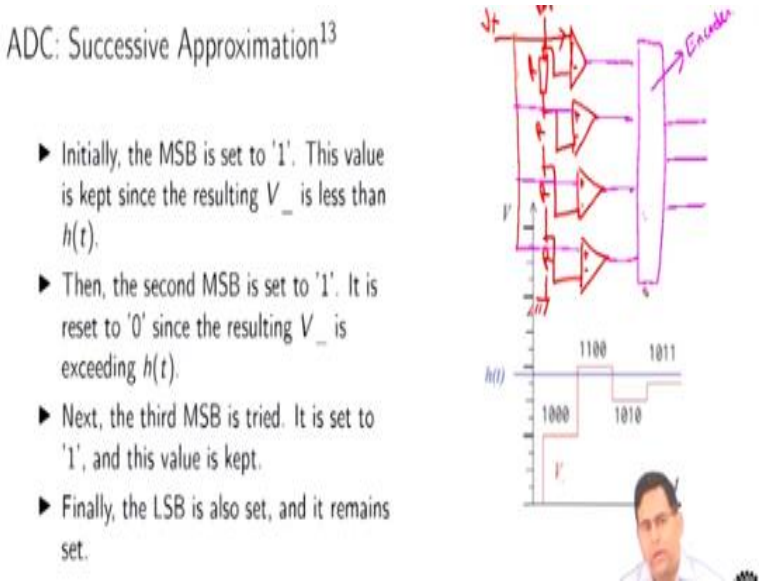


Now the other important thing which is like apart from sensors and actuators and control software's we have been talking about. We need to think about that how I mean they really operate together. Because as you understand that this is a physical system where our control signal that would be coming to do to give the actuation command is a digital command, right. But this physical system we will of course be actuated by some analog values and similarly for most sensors right.

So, any sensor will provide will generate an analog signal right, and that signal needs to be converted to a digital value when that signal needs to be read by a digital computer right. So, these lands up to this requirement of what is an ADC and what the DAC is and how do they work again that parts of an electronics course but again here we will be just having an overview. So, thinking from the sensor side, so sensors are generating analog values and those values needs to be converted to a digital equivalent.

And that has to be done through an analog-to-digital converter and there are very I mean this ADCs have different kinds of algorithms. So of course, the real ones that are used we have very comp I mean will be very sophisticated. We will just touch upon some simple ADC DAC circuits and how they really work.

(Refer Slide Time: 05:34)



So let us take an example here. So here we have an example of ADC algorithm which of course will be implemented in hardware. And let us understand that how I can convert this analog signal to a corresponding sequence of bits which will be transmitted to a digital computer. So, the idea here is of one of successive approximation so what happens is suppose we are trying to reach we are trying to approximately represent this analog value, okay.

So, what we start doing is, we are trying to we are trying to build the corresponding digital equivalent let us say a 4 bit digital equivalent. So, first thing we will do is we will set the MSB to 1 and we will see that well let us let us call this signal V minus. And now let us compare whether this signal is greater than or less than the target signal, which we are trying to approximate here, okay.

Now we see that this is less than the target signal. So, we will now set the next bit to 1, okay. Now once we do that we find that well now the signal the digital equivalent signal is becoming higher right. So, we are comparing and we are finding out that this is higher, right. So, once we find that to be happening then we will unset it and we will just so you see that this 1 has now been changed to a 0.

And we go to the next bit next significant bit and we will set it to 1. So, this is done and now I find that whether it is again lower than the target value then I will again set it to the corresponding then I will just move on to the next bit and I will set it to the higher. So, then this is the best

approximation I can get of this target signal using this 4 bit representation and that will be my output. So of course, you can see that this is going to work on multiple clock cycles when I am going to use this in real life. Not only that, at every point I would need a DAC because I am going to convert this to a digital equivalent and not only that I will need a comparator, right.

So, if I try to create this circuit it should be something like this. Let us try to draw this circuit here. So, you will need a comparator here. So, the logic that I explained of this generating this bit pattern. Let us say it is implemented here in the digital circuit. So, and that output comes here then a D2A comparison happens. Sorry, a D2A conversion happens and then it goes to the to this comparator so you have an operational amplifier which is a kind of working as a comparator here.

And you are taking the decision whether to set or unset the current bit and I mean an accord and moving on to the next decision here in the bit pattern generation logic. The important thing to note here is well we are using a D2A here inside anyway, right. The thing is that is an easy to implement thing, okay. The ADC logic implemented in this way hence becomes one option. The issue with this kind of a circuit is, it can I mean, I can run this circuit at high speed and it can work well. But of course, one thing we observed that the D2, I mean the A2D conversion in this case will take multiple clock cycles.

If I am going for an N bit representation, generation of the final bit pattern for each analog value we will need N number of clock cycles here so there is a problem. Now, is this the only option? No. I can also have what we call as a flash A2D converter. So that means in one cyc, I mean instantaneously without having to go for multiple clock cycles I would like to have this kind of I mean pattern generated for a given signal.

So, let us see how that can be done. So, suppose I employ multiple such comparators. So, as you can see there is a reference voltage. And so, this is the reference voltage and we have this voltage V_X , right. And what we are doing here is, we are having a voltage divider network of this resistance. 4 resistances of the same value R, right, and this is kind of dividing this voltage V_{ref} into 4 possible values, right.

So now when you try to do the comparison you see. So, these comparators are going to generate a bit pattern depending on the value of V_X and how it compares against V_{ref} . How it compares against V_{ref} by 2? How it compares again V_{ref} by 4? And how it compares again 3 V_{ref} by 4, right? So accordingly, this output bit patterns will be sensed and there would be an encoder circuit here, which will be having these decisions that whether V_X is greater than or less than V_{ref} and similarly being compared with 3 V_{ref} by 4, V_{ref} by 2, V_{ref} by 4 and all this.

Accordingly, you will have 4 decisions here, okay. Now it will be the job of this encoder to kind of convert this individual comparison values I mean from this point it becomes a simple digital encoder design problem right, and that would generate the final bit pattern. What is the advantage of this scheme. Definitely it would work in a flash right, because I mean there is no requirement of multiple clock cycles or everything so that is why this has a name it is called flash A2D converter.

What is the disadvantage? Well, it requires more hardware resources. For example, it will require here in this example you are requiring 4 hardware 4 comparators, instead of using one single comparator in a serial manner when you generated that successive approximation signal, right. So that is one alternative here for you.

(Refer Slide Time: 16:07)

Digital-to-Analog Converter (DAC)[14]

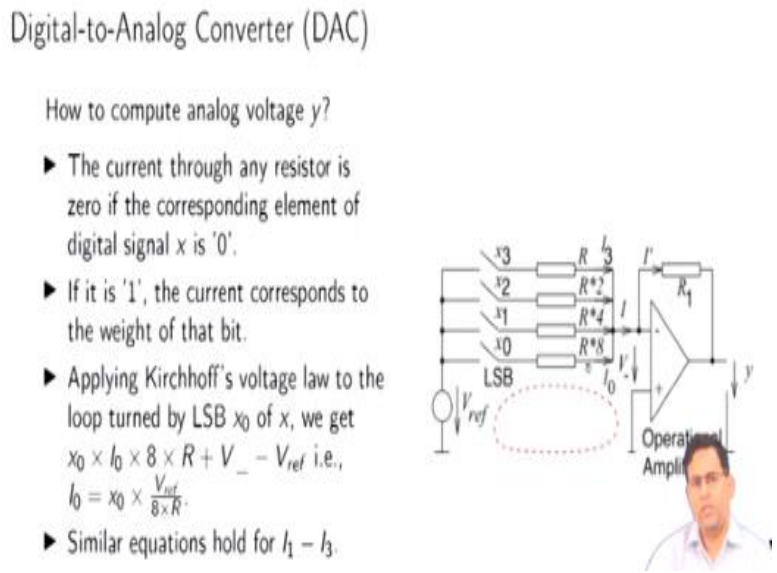
- An actuator is commonly driven by a voltage that may be converted from a number by a digital-to-analog converter(DAC).
- The key idea of the converter is to first generate a current which is proportional to the value represented by a digital signal.
- This current is converted into a proportional voltage.

Now coming to DACs. So, when we talk about these ADCs they are going to take the sensed analog values and they are going to generate signals for the computer where you have the controller

implemented. But then when your controller computes that control output it is again in a digital form as an output of the computer and this needs to actuate some analog action, right.

So, you need to convert to an equivalent analog value. So, typically this actuator would be driven by a voltage and that is converted through this DAC and the voltage should be proportional to the digital signal, okay. So how this can be done? The idea is that if I can have a converter which will first generate a current that is proportional to the value of the digital signal and then I have the equivalent voltage generated.

(Refer Slide Time: 17:03)

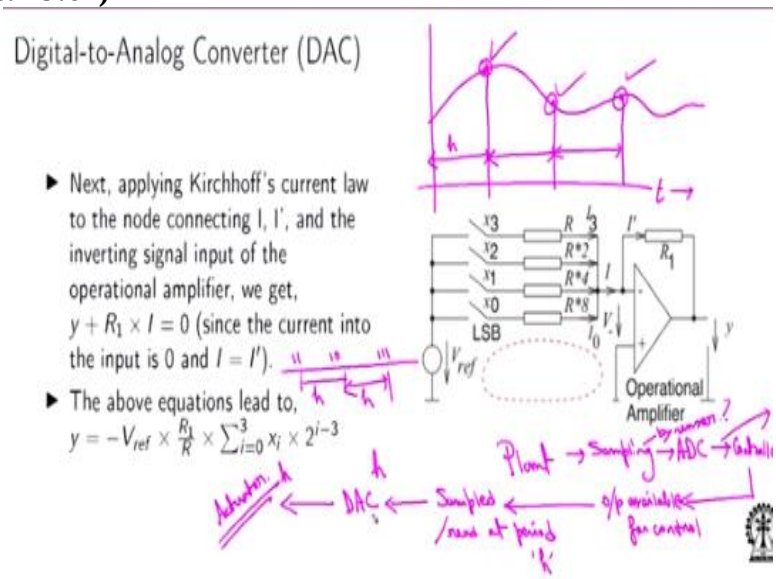


So, this is one possible simplistic example of a DAC circuit and I think it is fairly easy to figure out how it is going to work here, right. So, for example what we are having? We are again having operational amplifier here, okay, with the input the input let us say I mean as we know typically for an Op-Amp, this difference in voltage is kind of approximately I mean here this is connected to the ground. Here we are giving this 0, right. So, what is happening is? This input bit pattern will be used to turn on or turn off the switches.

So, you can think that these switches, I mean what they are really. So, they can be just MOSFET's right, where the bit point pattern once it is high here it will lead to this gate being I mean this this moss which is acting as a pass transistor. So, it will just act as a switch and it will turn on and this connection will be sorted from the drain to source right. So that is how these switches are modeled and accordingly what would happen? The current would flow through this register network. Now observe one thing, the value of the current that will flow here, okay,

that will depend on this resistance network and as you can see in this case this resistance values are chosen like R, 2R, 4R and 8R, right. So, its basically based on the binary weightage of different bits, right. So, if you apply standard Kirchhoff's laws here you will see that the current values are accordingly, I mean dividing this voltage proportionally and taking their corresponding values here right.

(Refer Slide Time: 19:01)



Now what this means is whenever you allow the current to flow it will the amount of current that flows will really depend on the weightage of the bit, right. So, when its LSD you have, I mean the current device I mean V_{ref} divided by R times 8, right. So, that is how the weightage is. So, if I sum up the total current that is coming here what I get it should be an expression like this following the current slope right so you have this kind of an expression (**). So, if you take common this V_{ref} by R, you will have something like X_i and i -3 to the power, right.

Because let us say you take X_3, you are essentially allowing this full current to go, right. Basically, V_{ref} by R to go. If you are taking the LSB, you are allowing the least current that means V_{ref} by 8 R you are trying to you are you are letting it go, right. So, it since it is the least significant bit the amount of current allowed is the least, that is why I mean that way it is modeling the weightage of the bit right.

So, in that way we have a proportional amount of current generated. So, in that way this bit pattern is getting kind of transformed into a proportional amount of current, right. So, when you have a

proportional amount of current moving here, when I apply it through this operational amplifier network, right. So, I will get a proportional voltage generated here, at the output and that would be my analog equivalent, right.

So, you can see that I can just apply Kirchhoff's current law here to this node that is connecting like this right, and what will happen is? I will get this proportional voltage because this will form a standard voltage divider network here. So, this is about a very I mean brief overview of DAC and ADC circuits that we have, right. So, a very standard way to derive these equations like you see, we have all we have done is we have applied this Kirchhoff's voltage and current laws in different points although I try to explain it in a more simpler way just I mean adding the current values and all that, fine. So, we understand how basic DAC and ADC algorithms work. But let us understand one thing also that even for an analog signal, when I am generating the digital equivalent, am I really generating it for all possible analog values? Not really, right. So, let us say, you have an analog signal which is changing its value like this, right. So, what I am doing is I am sampling this signal.

Let us say from the sensor for the sensor itself so we I am actually sampling this signal with a specific periodicity. So, I am sampling this value, then I am sampling this value and I am sampling this value. So, these values represent to me the signal information which is sampled periodically, right. Same and that is what is the analog value which will be given to the analog-to-digital converters to convert and send it to the digital computer, right.

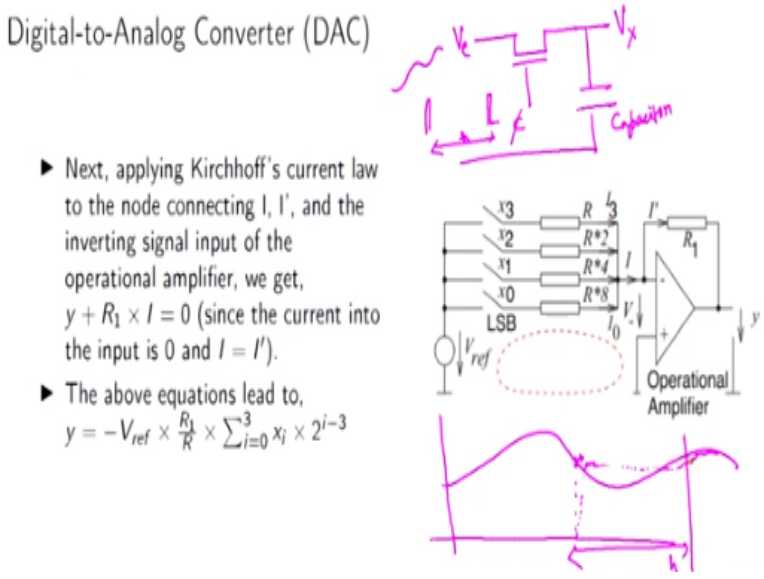
And of course, the digital computers convert value, and the computed value in the digital domain will again be activated by a DAC chip once every period of h and it will be applied to the actuator. That is how we have, I mean, a sample data system which is there acting as an interface between the plant which is analog and the controller which is in the digital domain. So overall, we have this kind of a cycle. You have the plant measurements which may be, I mean, here we have a difference in different cases.

So, either there is a sensor which is attached which is doing continuous sampling, and then, so the sensor is continuously reading the analog value and that value are being sampled periodically. And then those sampled values at periods of h, they are given to the ADC then the rest we already

know, the controller is going to out compute the output available for control. This output will be sampled or read from an output buffer at a periodicity of h and then that will be kind of converted to an analog equivalent, and that will output, that will update the actuator circuits input at the same periodicity. So, the DAC will also work at a period of h. The output of the DAC will update the actuators input value with an analog signal at a periodicity of h. This sampler which will be reading the controller's output will do the sampling at a periodicity of h. So, in an ideal control system that is how it will work. Everything is very regular. Plant is getting sampled, or the plant is, like I said there are 2 options, either the plant is continuously sampled, continuously a measurement is generated and that measurement is sampled to generate these analog samples and then those analog samples are converted to digital samples. So, at this point what you get basically is a stream of digital data. These are all separated by this period of h, right.

And then here this control output is sampled, then the DAC chip will be clocked with a period of h, and that it will update the actuators control input with the period within a period of h, okay. Now the question is how does this kind of sampling happen?

Like let us say the sensor is continuously sampling, that let us have the sensor is continuously, I mean, taking the measurement so it is generating a signal like this. But like I said, I am going to read the signal only at this kind of specific points, right. **(Refer Slide Time: 27:45)**



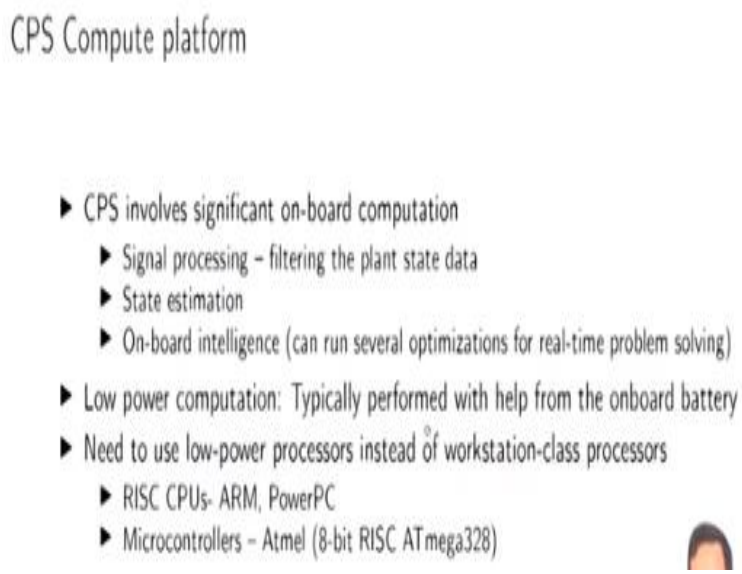
So one idea of doing such sampling would be something like this. It is, you can have a very simple sample and hold circuit. So, let us say here the continuous signal is coming and there is a pass transistor here. The output goes to a capacitor and I am actually reading the voltage across the capacitor. So, now what would happen is, this pass transistor is activated by a clock which is sending a positive edge at a period of h, okay, and h, I mean, so let us say here these positive edges they have a periodicity of h, right.

So, this will be closed or maybe I do a different kind of diagram here. Let us say, a short pulse happening here, then a short pulse happening here, okay. So that would mean when the gate closes, the capacitor will charge very fast, right. So let me again draw the analog signal like this, okay. So, the gate has closed let us say here, right. So, at this point, till now the capacitor's input was kind of, I mean, here there was no charge let us say. At this point when the gate closed, it slowly built up the charge, so the output right now is this.

And then it is kind of almost constantly holding the charge. I mean, very minor decay would be here, right. And then let us say, at this point it is sampled again, okay. So, then here from this point, it will again start and get that get the charge again and then again it will try to hold the charge. So, in this way it is basically sampling almost in a small amount of time depending on the time depending on this capacitance value it will get the charge and it will hold the charge almost I mean with very small decay up till the next point.

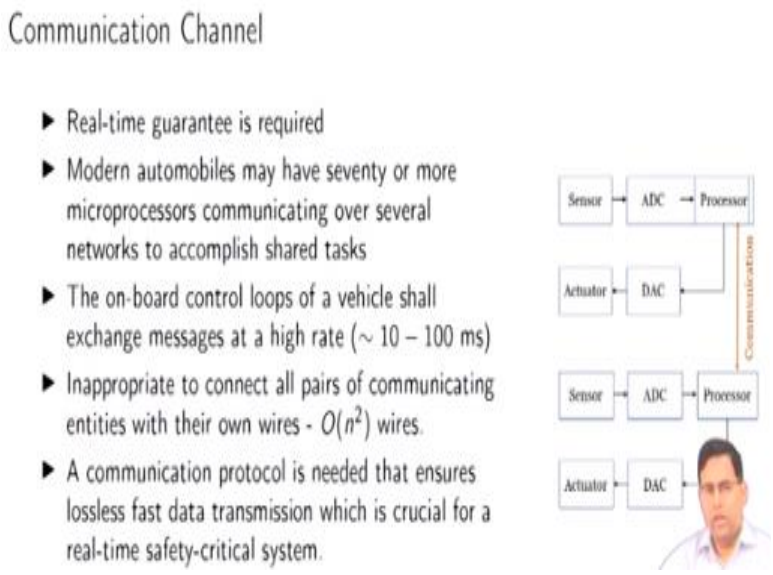
And from the next point again it will depend the voltage there it will again lose or get further charge, right. So that is how a sample and hold circuit will work. Of course, is a very simple example I am showing you here. And it will give you a set of those sampled analog inputs and those sampled analog inputs are going to drive your analog to digital converter. And then at the output I will get a bit stream at a periodicity of h.

(Refer Slide Time: 31:06)



Now of course, in our previous week, we have kind of discussed about different kind of CPU compute platforms and just to summarize its properties like, we want such CPU compute platforms to be very low power and what are the usage of those platforms of course you are going to do the control law computation. But there is a not only that is not the only use. It will be also used for many real-time algorithms. For example, whatever sensor data you are getting that would be noisy.

So, filtering the data using the sensor measurements to do state estimation of the plant variables and maybe you can also run some ML pipelines to do some recognition of, I mean, identifying interesting features from that sensor data, right. And finally, you can also compute the control law. So, all those things have to happen here and it has to happen here. An important following an important characteristic which is that compute platform should be low power. So, like typically there are CPU and microcontrollers which are being used for this purpose like we have already seen in our previous discussions. **(Refer Slide Time: 32:16)**

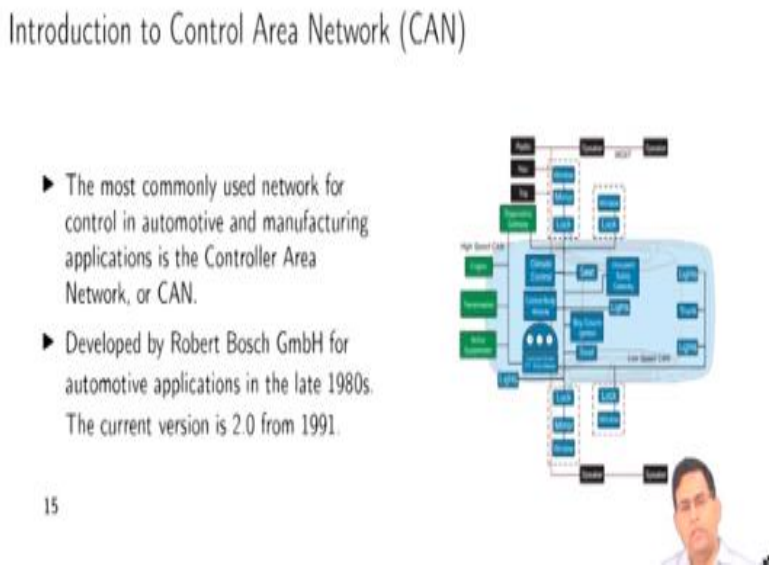


Now what about the communication? So, whenever we are talking about the Cyber Physical Systems, we have said that where communication plays an important role and the communication has to be real-time, right. So let us take an example. Consider that how in a modern vehicle you will have so many control loops and those control loops may be implemented in different ECU's, right. And those ECU's will need to exchange messages among each other, right.

So, and that message exchange can typically happen at a very high rate, okay. So, it can be in the order of, I mean, one message for 10 millisecond or one message for 20 to 30, I mean, up to 100 like that. So, I mean, 10 millisecond period kind of that represents of big, I mean, high rate in the kind of network protocol that is followed inside an automotive. So, the issue is if I have many processors, so any modern automobile can have 50 or 80 ECUs inside them, I mean, of different form factors.

Now the issue is, if I try to connect each of them, right, all possible pairs, right, it would be an $O(n^2)$ number of wires that I would require right and there would be that would be a big mess. So, we understand that there needs to be a standardized bus architecture which needs to be followed whenever you are connecting ECUs, right. And for such bus architectures you need to have a suitable communication protocol that would ensure lossless but fast data transmission.

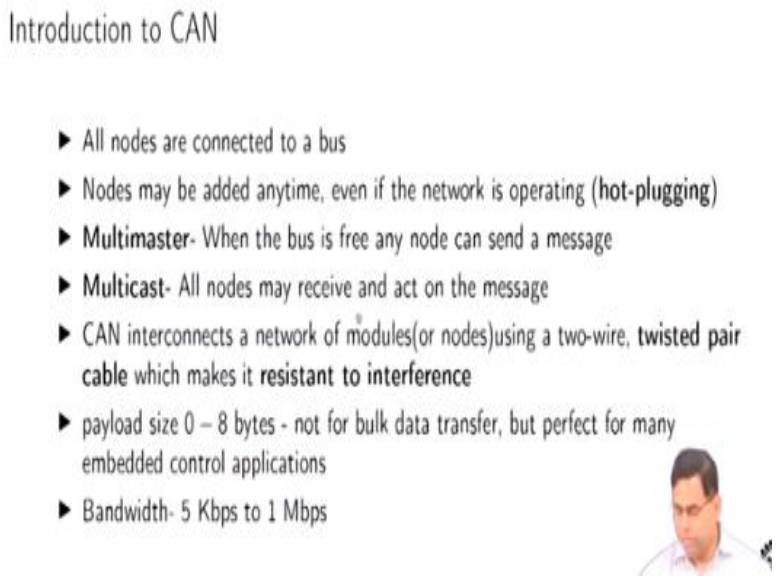
Unlike our standard TCP kind of protocol where it can also be lossy, this has to be, I mean, lossless and even if there is a loss it should be immediately detectable and it should meet the real time guarantees, okay. I mean, even if there is a delay, there should be a good worst-case estimate of that delay, not a probabilistic estimate but a better one actually. **(Refer Slide Time: 34:14)**



So, all of these are properties that would be required to be satisfied in automotive system and also it has to consume very very low bandwidth and it should not take too much power, right. So essentially you need to have a very simple network protocol. Some protocol which does not require too much of a network infrastructure lot of switches nothing like that, because it should be inside a car and it should be as low cost as possible, right.

So, in this way the most commonly used protocol that we have in automotive is called Control Area Network, or CAN, okay. So, this is the detected standard in automotive. So, nowadays automotive Ethernet and other standards are also coming in but this is still this happens to be the de facto standard for all, I mean, intra vehicular communication. This was developed by the company Robert Bosch around nineteen eighties and the current version is 2.0 from 1991.

And of course, there are new variants of CAN that have come up, like CAN with flexible data rate and other variants. But we what we will do is we will study the basic protocol because that is pretty much used till date. **(Refer Slide Time: 35:30)**

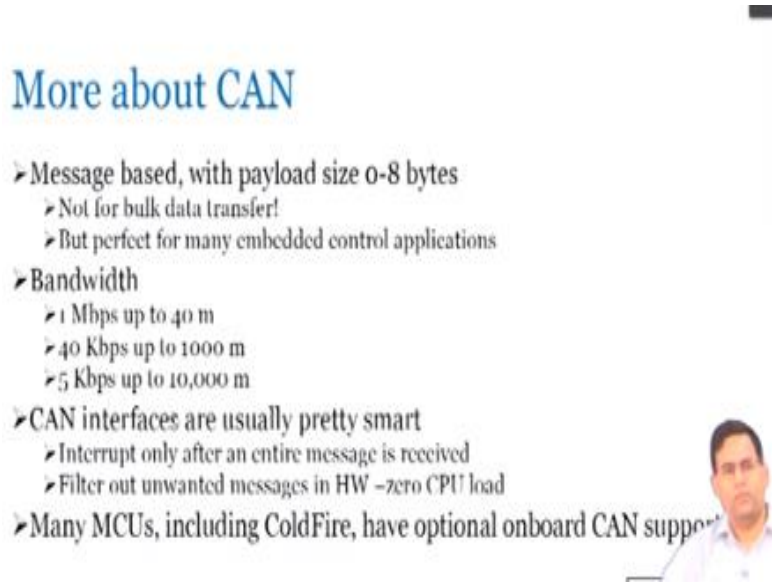


So, how does the CAN network work? So, the basic idea is that all the nodes that you have in a CAN network, they will connect to one common bus, okay. So, they will connect to one common bus, that means all the nodes kind of they are always looking into what data is there inside the bus, okay. And any new node can be added to the network at any point of time even if the network is online so it is hot pluggable, okay.

And whenever the bus is free any node can send the message. That means for sending messages, a node does not need some master nodes permission or stuff like that. So that is not required at all. So, no such thing exists there, okay. And other so this is called a multi master thing. There is no master node or something like that. And the other thing is it is also multicast, that means, all nodes may receive or act on a message that means whatever some node is sending everybody can actually see it can actually read also, right.

Now the way a CAN network is designed is very simple basically it is a twisted-pair cable. So, since it is a twisted-pair wire so that also makes it resistance to electromagnetic interference. Typical payload size that means the actual data that is used is a very small packet. It is a 0 to 8 byte data so this also means that you cannot do some bulk data transfer, the packets are quite lightweight.

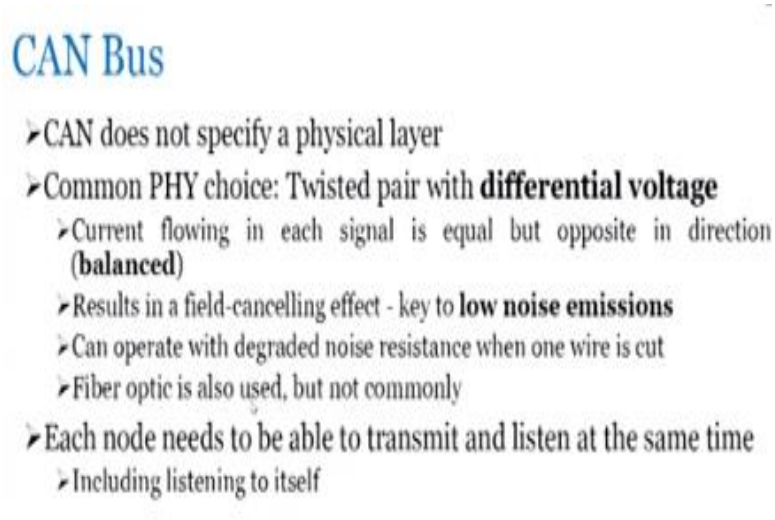
I mean if I compare to standard network packets but at the same time that also makes them perfect for most embedded control applications. The bandwidth offered by this kind of a network is typically 5 kilo bytes per second to 1 Mbps. **(Refer Slide Time: 37:26)**



Now of course, the bandwidth has a limitation on the cable length. We can understand that from simple physics here. So, if you are trying to afford 1 Mbps speed, your cable end to end length should be inside 40 meter. Similarly, for something some limit for 40 kbps or slower speed higher range much, I mean, something like 5 kbps much slower speed you can have a much bigger cable.

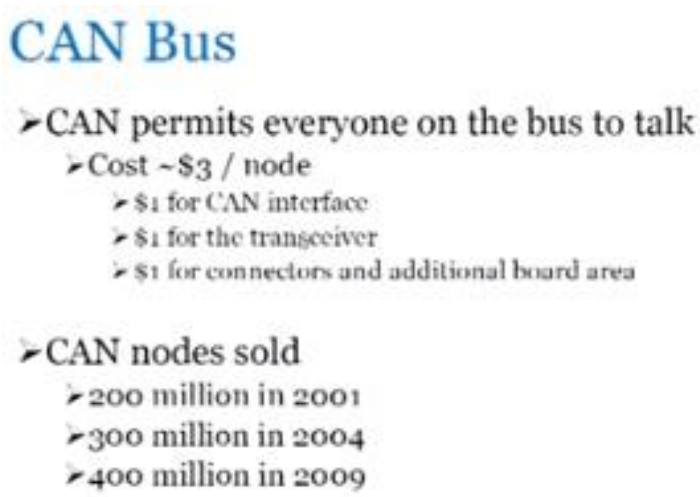
Of course, you do not need this much bigger cable inside an automotive. But you can have a few kilometers of cabling inside an automotive which is kind of which always happens, right. So, these interfaces are usually pretty smart you have can every CAN node will have a CAN controller and these are the CAN interface through which the node will connect to the bus. So, this can be interrupted only after an entire message has been received and it can also filter out unwanted messages so that problem does not go to the CPU.

Nowadays what happens is many this microcontroller units that are available in the market, they actually have onboard support of CAN, that means they have the CAN controller and you can just connect the CAN wires with them. **(Refer Slide Time: 38:42)**

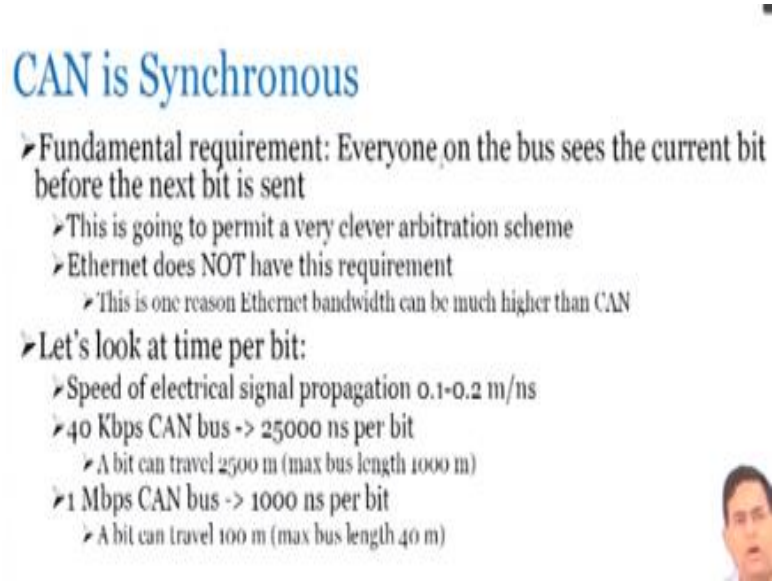


So that that first possibility also is there. Now CAN does not specify a physical layer. So, that means unlike other transceivers the physical layer is not implemented but there is a choice and the standard choice that designers take is a twisted-pair with differential voltage. That means there are 2 cables. They are twisted-pair and the current is flowing in each signal line with equal but they are opposite in direction so that kind of balanced.

And there is a low noise emission and due to the twisted-pair setting its very much resistant to noise. And because think of a vehicle's bonnet there would be lot of noise there will be lot of disturbances and your bus needs to work inside each itself, right. Every node when they transmit, they can also listen at the same time they can see what is getting transmitted, and they cannot they are always able to listen to the channel even when they are kind of transmitting, okay. **(Refer Slide Time: 39:36)**



Now the CAN nodes, the way they are designed, they are very cheap actually. They permit every node on the bus to talk and if a CAN I mean the interfaces the CAN interfaces and the transistors they are very cheap actually. And as you can see the number of can nodes sold in different regions the I mean this this is a pretty old statistics but still the is quite significant I mean because CAN is kind of de facto standard in automotive and CAN is used also in several other industries, okay. **(Refer Slide Time: 40:12)**



So, like we said that everyone on the bus can see that what bit is getting transmitted, okay. So, and that is one important point and this have a significance. We will soon see that how if 2 nodes with different priorities are going to transmit, then how that arbitration, that who will finally be able to

transmit how that can be done, this has a very elegant solution in CAN. We will see that. One important point to note here is, such things are not there in Ethernet.

So, what I mean let me just explain. Suppose there are multiple CAN nodes who are trying to send their individual messages exactly at the same time. Of course, in Ethernet that is pretty much allowed. Only thing that will happen is that this will lead to a packet collision. Now once this Ethernet in the Ethernet type connection this collision is detected, there would be an exponential backoff and in future probabilistically at some time point they will again try to do the data transfer, right.

So, we do not have any guarantee, but due to the high bandwidth that Ethernet, provides the packets will eventually get transmitted with some tolerable delay. In CAN, we have a bandwidth issue. Hence, we need an efficient arbitration scheme and not only that if there should be a real-time guarantee. What we mean is, whenever this case arise that probabilistically multiple nodes are trying to transmit exactly at the same time, there should be a well-formed rule, in place in the protocol which will say exactly who can transmit and exactly who cannot transmit and that should be followed and accordingly the nodes must behave, okay. Now so that is an important thing we will soon see that how this kind of arbitration is done and this also makes CAN synchronous. That means, every node, I mean, the on the bus every node I mean if they are trying to send the bus is operating kind of in a cycle wise manner.

So, I mean if 2 nodes are transmitting, I mean, it is either they are both bits they are trying to send at the same time or in different cycles, okay. Now let us load the time per bit discussion we had. We told that well there is a relation between how much speed I can have and what is the length of the CAN bus something like that, right. So typically, the speed of electrical signal propagation is like this 0.1 to 0.2 meters per nanosecond, okay.

So, if I am having a 40 kilobits per second CAN bus, if that is the data rate, that means, you are you are sending 40 into 1024 number of bits per second, right. So that means for each bit the amount of time you are keeping reserved is 25000 nanosecond. So, inside this 25000 nanosecond,

the distance that this bit may travel will actually limit the physical length of my cables, right. So that is why a bit can travel as per the speed of electrical signal we discussed, this much distance 2500 meter.

So, accordingly, we will keep a bus length which is less than half of that so that this delay does not create any problem, right. Similarly, if I have a 1 Mbps CAN bus, so again, so as a higher bit rate. So, for a higher bit rate we will have 1000 nanoseconds, a smaller amount of time per bit. So, bit will be able to travel less amount of distance and that will again limit the bus length. So, with this we will end this lecture. We will resume from here in the next lecture. Thank you.