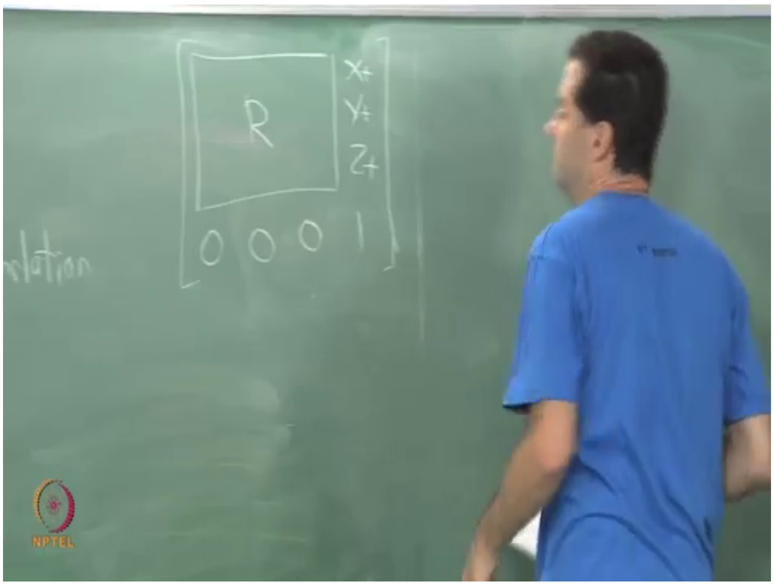
Virtual Reality Engineering

Prof. Steve Lavalle
Department of Multidisciplinary
Indian Institute of Technology, Madras

Lecture – 12-2
Tracking Systems (gyroscope integration)

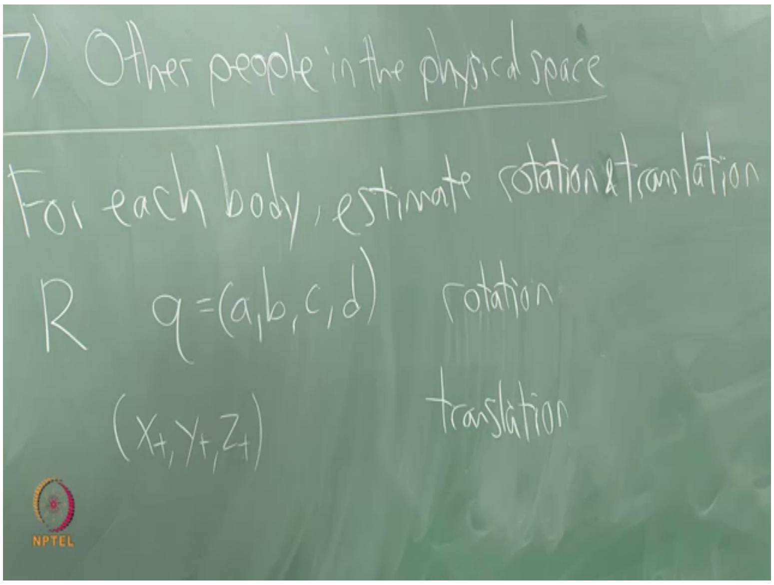
All right.

(Refer Slide Time: 00:17)



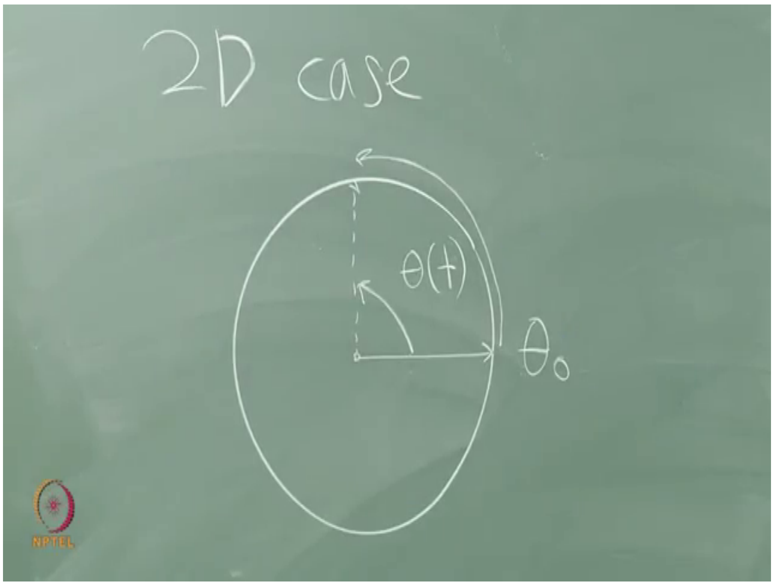
Let me give you a very simple example. So, I am going to I am going to go in two phases I am going to first talk about estimation of orientation and then I will talk about estimation of orientation plus translation.

(Refer Slide Time: 00:32)



So, turns out orientation based on current technology is easier than translation in terms of estimation.

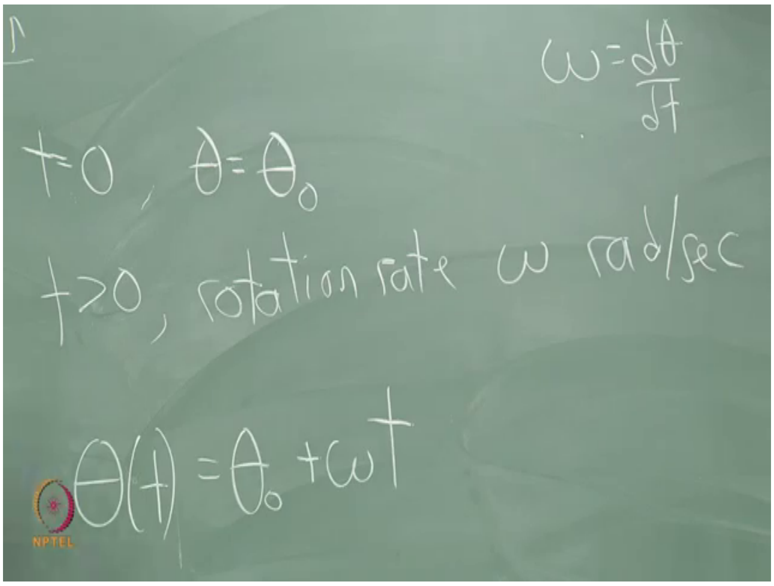
(Refer Slide Time: 00:39)



Its a bit surprising perhaps, but that is the way it works. So, estimating and let us jump down to a very simple case the 2D case which in terms of space of orientation is just one dimensional. So, imagine I just have a spinning wheel .

And maybe have some kind of marker on this wheel that is pointing in some location and I will say that that is at initial orientation theta 0 and over time this wheel starts rotating, and it ends up being at some angle theta maybe I can look make it look like its time parameterized by putting theta of t. So, this wheel is spinning and maybe its rates changing as it goes.

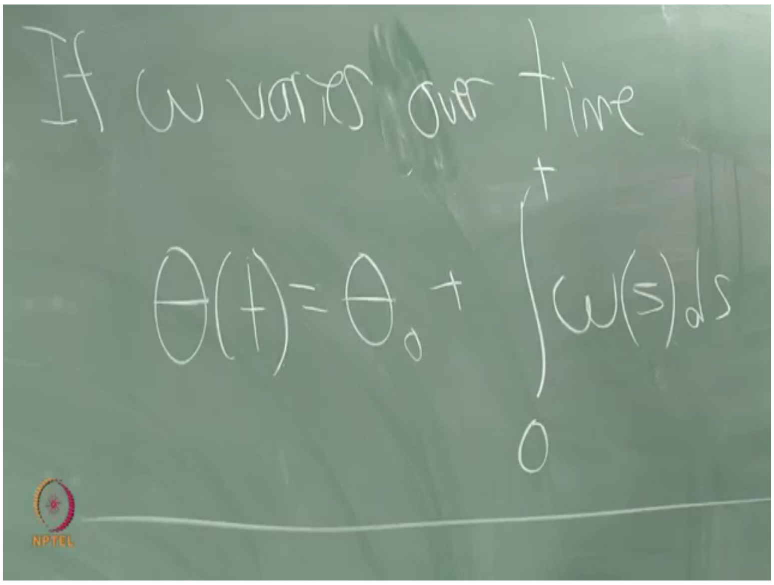
(Refer Slide Time: 01:43)



So, I say time t equals 0 theta equals theta 0 and at t greater than 0 it has a rotation rate .

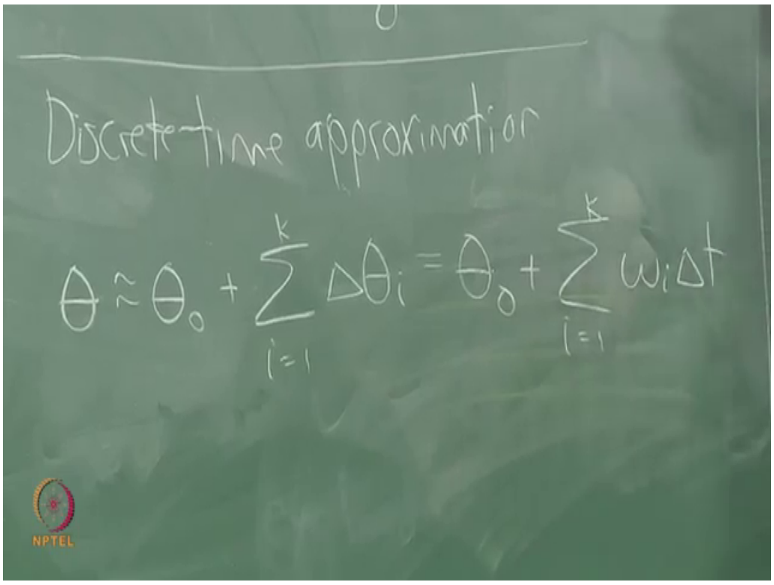
Let us just say its a constant to start. So, rotation rate is omega radians per second. So, omega using calculus is just a d theta dt. So, just the angular velocity. So, if that is the case then theta of t if our theta as a function of time is just whatever my initial theta is plus omega t right. So, that is also how long has it been going for multiply that by the rotation rate and that will give me the total amount of rotation in terms of radians see here all right.

(Refer Slide Time: 02:55)



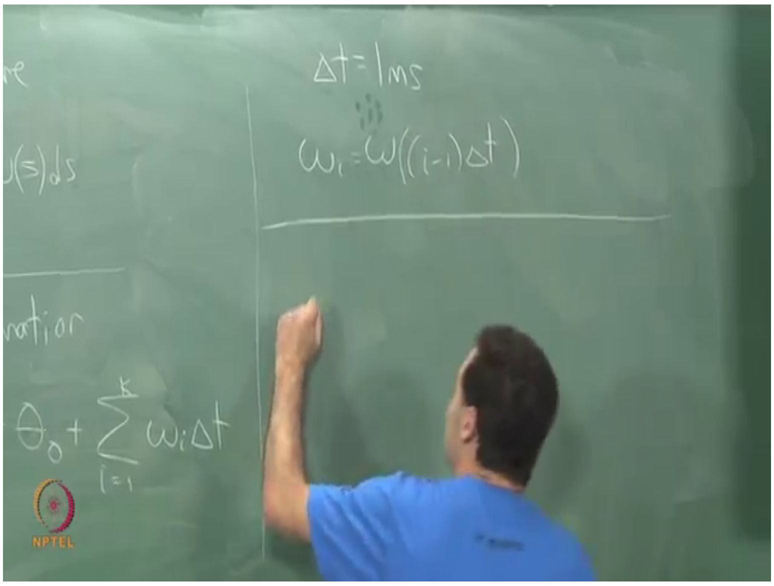
Now, what if omega varies over time, then theta t is just theta 0 plus the integral from 0 to time t omega of I do not want to put a t inside of here because its an integral. So, give some other parameter s ds right. So, we just now write omega as a function of time and we just integrate that right. So, that is all simple calculus and I have not really computed anything here its just you know how it is in calculus its. So, much easier to write things down than to actually do anything with it right as soon as you have to actually do integrals you are in a lot of trouble or if they only show you the ones that work for its them all right.

(Refer Slide Time: 03:57)



So, of course, what we do an actual computation is we make a discrete time approximation and if that is the case then we end up with something like the following we may say that theta is approximately equal to theta 0 plus a sum i goes from 1 to let us say k some step delta theta sub I which I will just do some change in notation say that is equal to again starting a theta 0 i goes from 1 to k of omega i times delta t and i expect delta t to be some fixed sampling rate.

(Refer Slide Time: 05:06)

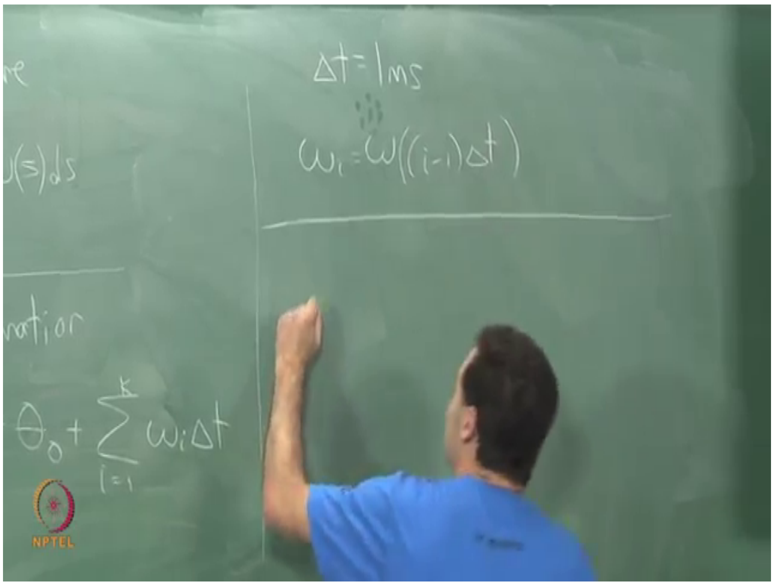


So, maybe delta t is equal to one millisecond this is a common sampling rate for accelerometers and gyroscopes that are used in these kinds of tracking systems. So, that is a very practical and a common number.

And then my omega sub i is just some sampling then let us say that is my omega function that I drew here when I was trying to do calculus at some particular time index maybe i minus one times delta t whatever. So, just some particular time stamp that this omega i corresponds to. So, that is how I estimate my theta of t let me put theta of t over here theta of t right and this k goes this is the highest k, I can get to before reach before advancing beyond time t.

Now, so, that is a scenario I have. So, I must lose something by doing a discretization here right. So, this is going to be a simple Euler integration discrete time approximation to this integral. So, that is one place where I have lost some information, another thing that is going to happen is I am going to assume that I cannot actually accurately measure the angular velocity, but I get a sensor that tells me the value the sensor is going to have limited precision it's going to have noise issues calibration issues. So, I get some sloppy numbers and I am going to do this kind of numerical integration by just trying to estimate the change in theta how much did date a change in each millisecond I just want to add that up it's a pretty easy some reasonable right.

(Refer Slide Time: 06:45)



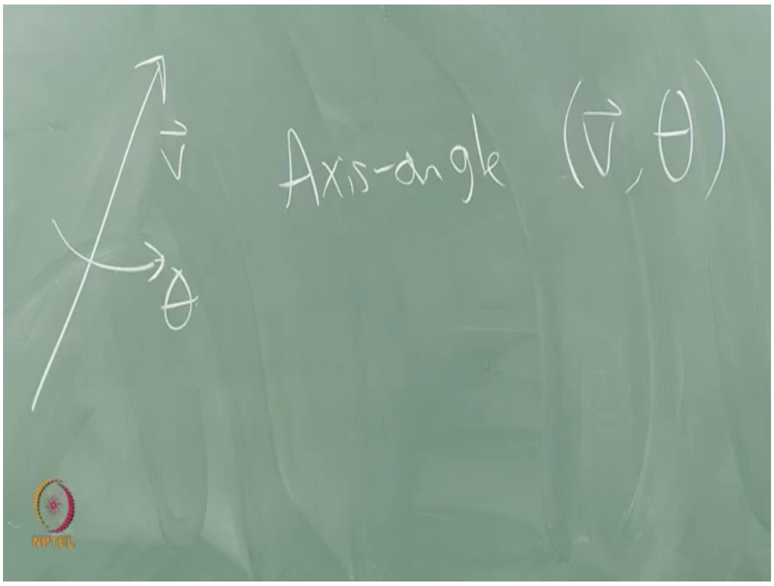
So; that means, that what I am really trying to do is construct some theta hat at some given time I might even start to get confused about what time it is in fact, but I do not want to go get too crazy here and so, that will be theta 0 plus sum i equals 1 to k omega i have put a hat on it to indicate that it is an estimate. So, so measured by a sensor measured by some kind of filter probably not correct, but the best guess I have that is what hat is going to mean delta t, where this is the sensor reading from a one dimensional gyroscope all right.

So, gyroscopes that is just gone to report a scalar quantity that is the yaw in radians per second right questions about that. So, this should make sense right. So, imagine this wheel is turning here like a turntable or a merry go rounds. So, it is turning around its rates varying we hook up some kind of one dimensional gyroscope, we put it on top of this disk and it starts spinning and it takes measurements for us does not matter where we put it on the disk should still report orientation changes the same way, I believe maybe its accuracy might be affected by where it said something to think about.

But depends on how the device is engineered if we place it there we get these measurements out, and then we are just accumulating numerically integrating in the simplest way possible I am not using trapezoidal rule and simpsons rule and other kinds of fourth order runge kutta and other things like that to try to approximate a better, I am just doing simple adding off of the measurements keep it simple here all right just seem fine. Now I would like to do the 3D version of this, and I would like to not get lost in an in an enormous mess of 3D rotations. I would like to not be dragged through a bunch of trig functions and yaw pitch roll and singularities I have seen a lot of mess out there on the internet. In fact, when I searched for different methods for integrating gyro readings it turns out its very simple if you have already committed to using quaternions and its kind of a mess if you have not.

So, let me just show you very very simple way to do things. If you have a quaternion library ready to go it ends up doing the exact same thing here, but just with a like a couple of lines of code just like this would be a couple of lines of code or software the 3 D case ends up being simple like that as well if you understand the principles we have covered already on 3 D rotations.

(Refer Slide Time: 09:32)

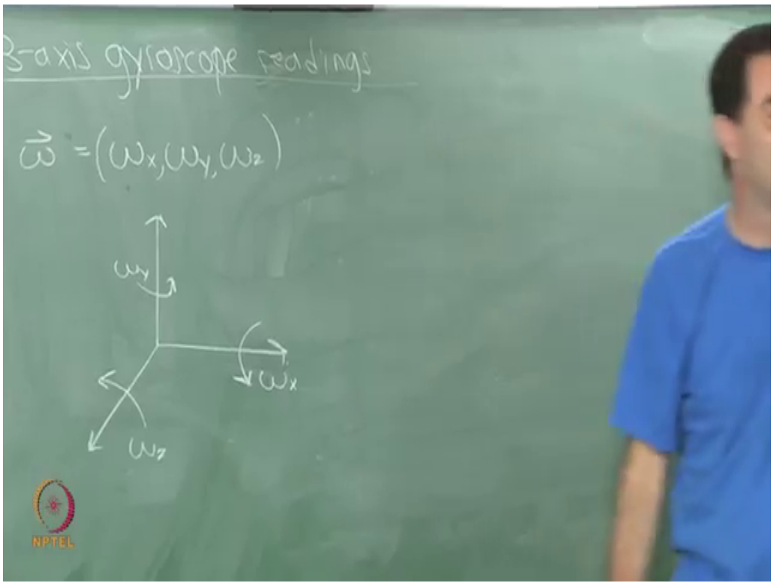


So, now, look at 3 D 3 D orientation case remember Euler's rotation theorem which is every 3 D rotation can be represented as a rotation about some access through the origin and some angle about that axis and there is of course, two different ways it could be done depending on which way the axis is oriented that is we could reverse the axis.

So. So, some access through the origin and some amount about that axis. So, I am trying to estimate the orientation of the head, if I want to make a perfectly portable vr system a lot of vr systems right now worked with smart phones, that are just dropped into a case right Google cardboard Samsung gear vr like that many others are out there is a lot of open source ones that are out there so.

So, basically if you just use these gyroscope sensors and additionally hughes accelerometer magnetometer which we will get to, but if you just use these portable sensors on the phone that are based on mems technology, then we are just tracking orientation that is about all we can reasonably get from this. So, if we look at the gyroscope measurements, we get draw here to the side. We have what is called a three axis gyroscope.

(Refer Slide Time: 11:04)



So, we have three-three axis gyroscope measurements or readings, I said readings there let me let me do readings. So, we get some vector omega, I wrote omega as a scalar here I am going to write it as a as a vector here. So, we get these readings, I guess I am not going to write it with a hat yet I am just going to talk about 3 D angular velocity for a bit here. So, before I actually get to the readings, I just want to say that if we look at angular velocity in three dimensions there is omega x omega y and omega z and I will drop my coordinate axes here. Remember that this was the x axis going to the right why is up and z is coming out at us to keep it right handed.

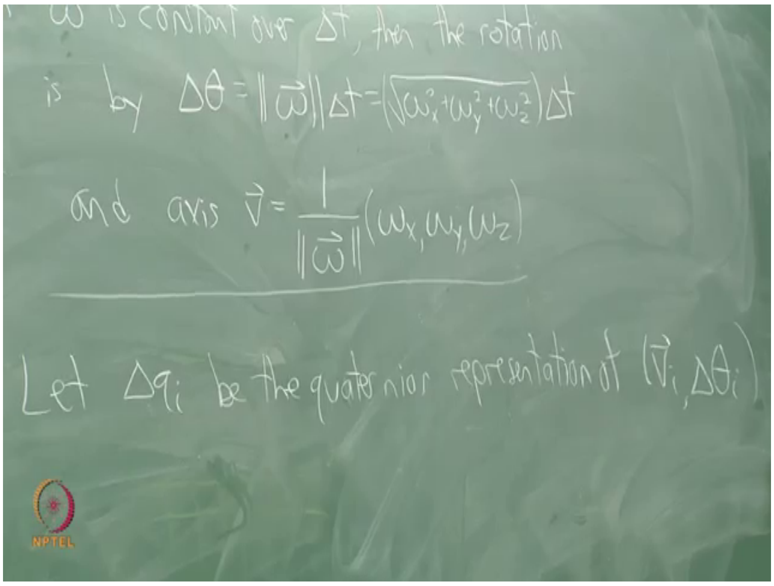
And so, what is interesting about angular velocity is that this is the instantaneous rate of rotation about these three canonical axes. So, if I right we just write it out here and describe it a bit. So, if I write out omega acts as a rotation about the x axis. In fact, rotation rate about the x axis, omega y is rotation rate about the y axis, omega z is rotation rate about the z axis what I mean is that, imagine the 3 D body its tumbling along right its rotating around and then at a given time, I just take a snapshot of it and now I project into the various axis.

So, if I see this object rotating and I take a snapshot of it as projected into the xz plane looking down the x axis. It will look just like 2 D rotation and when I do that projection its rotating at some rate and that rate is exactly what my gyro is going to be measuring is not that nice. Same thing is going to be true for the y part it will be if I look straight down on it. What will it look like in the xz plane how fast is it rotating radians per second single scalar number it will give me that one when I look at it in the xy plain looking down the z axis it will again give me a rate of rotation what is that? That is going to be the omega z component. So, it is giving me these three instantaneous rates of rotation, it feels a lot like yaw pitch and roll rates; however, because its instantaneous we do not have to worry about the weird problems of non commutativity of 3 D rotations this is just an instantaneous measurement of the rates.

So, they are all happening at the same time there is no ordering problems or kinematic singularities we are not putting them all together its part of a sequence of matrices to perform the operation it is just an observation about these rates. So, at that instant I want to know, what is the rate of rotation? So, that if I put these all together, I should be able to extract a single axis three dimensional axis and a rate of rotation about that axis and that if I can get that that would be beautiful that would be just like what I had in the 2D case the axis of rotation of the 2D case is always the z axis right because this is the third axis it stays the same. Here remember this axis should be changing over time and I am looking at these rates in a single snapshot. So, here is what happens is really its really quite beautiful shows up in textbooks mathematicians know this does not show up enough on the internet sadly in terms of answers to peoples questions on forms just the following.

So, let us just assume for a bit that omega is constant over delta t I do not want to allow mega to vary during one millisecond. So, let us just assume that it is constant and so, if of course, this is a an approximation.

(Refer Slide Time: 15:00)



But if omega is constant delta t, then how much rotation has occurred? Then the rotation is by some amount delta theta. So, I am going to tell you, how much rotation there was delta theta is going to be this kind of theta its ax angle representations maybe some amount of rotation the amount is going to be exactly the length of this omega vector times delta t all right.

So, in other words it is just square root of omega x squared plus omega y squared plus omega z squared times delta t which is a one millisecond you know 0.001 in the case of most mems gyros that people are experimenting with right now. So, that is very true. So, that suggests that if I just take the length of this vector that is going to give me the overall rate of rotation this makes good sense because each one of these is in radians per second, and if I take the overall length of this that is the total amount of radians per second if this object stops rotating all three of these components become 0 to make good sense right there is no rotation at all. If its rotating only in the xy plane then the omega z component will have all of the rotation, when I take the length I just get exactly that part. So, this is giving me the overall amount of rotation in radians per second and this will be the amount of rotation that occurred in time delta t the instantaneous amount the instantaneous rate as a scalar is just as magnitude here.

But the question is what is the axis? And what is very interesting about this is that the axis say v is just taking this vector and normalizing it. So, this the direction of this vector is already the instantaneous axis of rotation. So, it is crazy there is like no work to do and its already its already being given I am not deriving the mathematics of this I am just giving you the result with some intuitions along with it.

So, if I just renormalize this vector here. So, I could we said I wanted the access to be a unit vector at some point earlier when I was covering these things. So, if you do not like to normalize it you do not have to as long as you know ah. So, you just divide by the magnitude be careful of dividing by 0, if omega if all three components of omega happened to be 0 then there is no instantaneous rotation happening which you just handle it as a special case, even in the software you handle it as a special case.

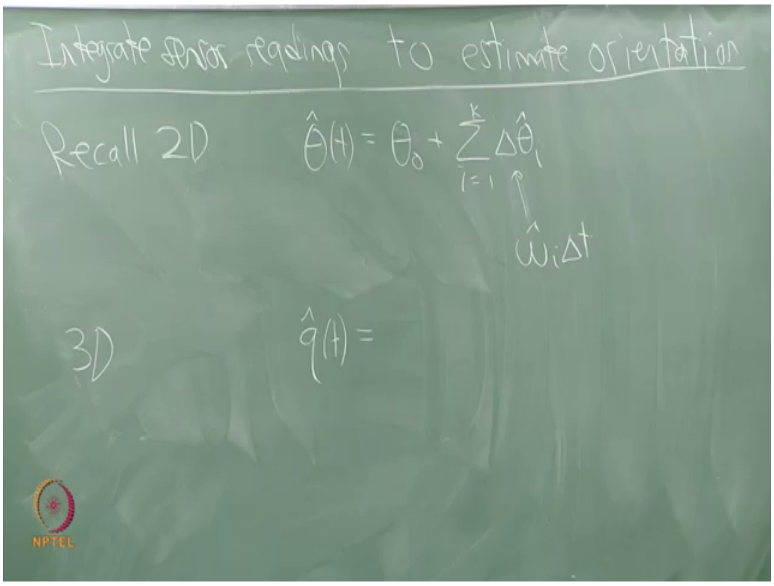
So, this ends up being the axis of rotation during this interval corresponding to this interval delta t. So, I said omega is constant, if it were not constant then the v would be changing and this whole approximation would fall apart of it right if a change is to significantly. So, I am assuming that delta t is so, small that the amount of change in omega is insignificant, and all I am going to do ultimately is get measurements of omega from my three axis gyroscope 3 D gyroscope. So, I can now say let delta qi be the quaternion representation of this axis.

Let us say step I and this delta theta at step I. So, is that all right? So, in the 2D case I was just happy talking about delta theta sub I, now I have to talk about some axis (Refer Time: 19:31) in addition to delta theta does that make sense because in 3 D it is the axis, that we have to recover in addition to just the amount of rotation right. So, the amount of rotation radians per second we are trying to estimate that, but we also have to keep track of this axis. Luckily this three dimensional gyroscope if it provided a perfect reading it would be giving me the three components that I use to directly construct the axis.

So, if I just take for example, in software I may have a quaternion library that I just pass delta theta and V 2 and it will immediately returned back to quaternion that corresponds to. That quaternion in most cases will be fairly close to the identity. Why is that because this is a tiny rotation right how far is it going to rotate in one millisecond? So, just like if your intuition in the 2 D case its only is rotating us on a reasonable rate let us say, how fast is your head turning. So, in one millisecond it is not going to get very far. So, this is going to be a quaternion, where delta theta is small and the axis of rotation. In fact, most of us if we are doing this most of the time, then it is going to be its going to have the largest component corresponding to yaw right.

But if you do some of this and find you are going to pitch component and so forth with. So, now, we want to do is integrate these readings, let me write back the 2 D case and then I want to show you how simple the 3 D case is once you have been tortured by quaternion's right. So, so just I do not think its torture, but some people sometimes react that way all right.

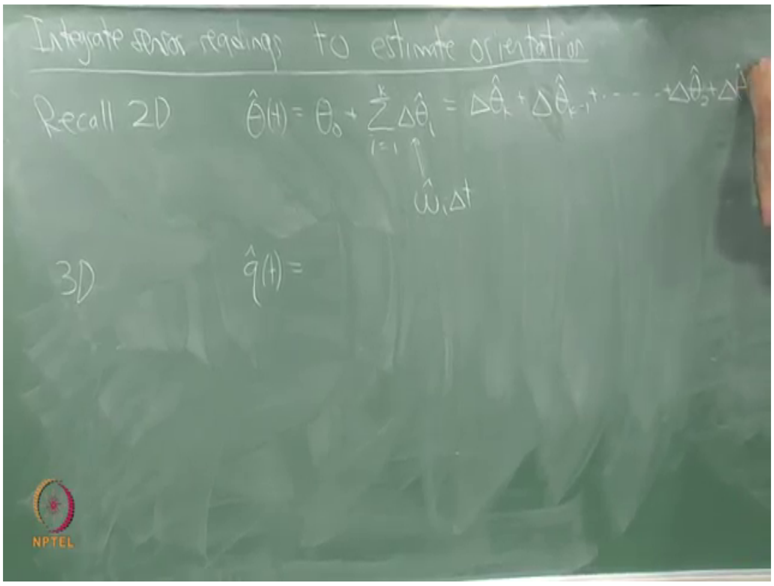
(Refer Slide Time: 21:08)



So, integrating sensor readings to estimate orientation. So, I gave you the 2 D case. So, recall the 2 D case which in terms of orientations was really a 1 D case, I said that our estimate of theta at some particular time t which I guess I have not been too explicit about how that is related to k, but theta 0 although I did say that it is the consider k all the way up to the current time and not beyond it. So, we have I goes from 1 to k I wrote this before and it will be the delta theta hat sub I and you may recall that I said that that is equal to omega i hat delta t correct.

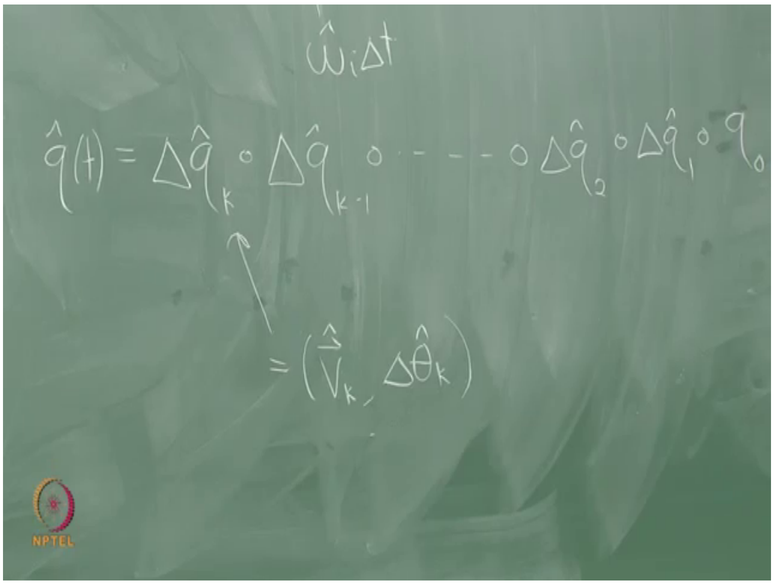
And now for 3 D here is what is going to happen? So, 3 D case we have let us say q hat at some particular time t is equal to we already have one problem I promised you would not get too complicated, but there is one small complication 3 D rotations are not commutative and if you remember things that come later we have to write on the left side right. So, we go from right to left so; that means, that if I write a sum if I were to write this some out here in the usual way up here for the 2 D case, I would go theta 0 plus delta theta 1 hat plus dot dot dot I cannot write it like that I have to write it in the other direction.

(Refer Slide Time: 23:20)



So, to make it match up. So, I guess I could write this one out like this I could write delta theta k hat plus delta theta k minus 1 hat plus dot dot dot dot dot all the way down to let us see delta theta 2 hat plus delta theta 1 hat plus theta 0. So, I am going to write this one in the other direction, because up here I do have commutativity. So, I might as well just write in the other order. So, that my pictures match up in a pretty way. So, that is what I want to do.

(Refer Slide Time: 23:55)

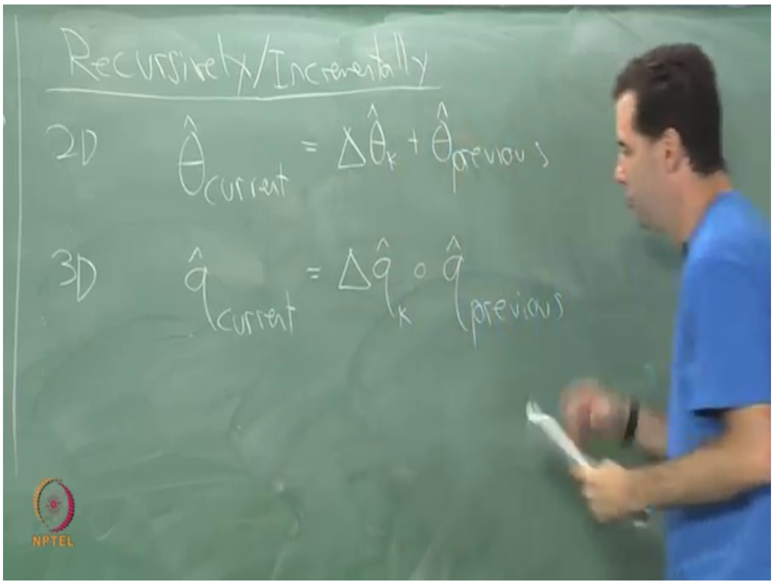


So, I do that I will get down here this will be equal to delta q k hat and putting hats on it because it is coming from sensor readings, quaternion product delta q k minus 1 quaternion product at a dot quaternion product delta q 2 hat, quaternion product delta q 1 hat quaternion product q 0.

How is that that is how it looks. And now my delta q I have given you delta qi is the quaternion representation of this v sub i and theta sub i. So, each one of these I guess I did not even write it out with eyes here I only wrote it out with k that is fine I will just say this is equal to k. So, I put an arrow and a hat on that because it is an estimate coming in and delta theta hat k. So, that is what each one of these is this quaternion that I have constructed in each step.

So, that is all I have to do is just keep multiplying my quaternions or if you want you could have converted them into matrices, and just multiply all the matrices together if you like. As well as I said there is a lot of advantages to staying completely inside of quaternion land with the algebra. So, we can just do it all those quaternion multiplications and keep propagating. This forward there is another nice way to look at it because when we are writing code each time we get a new measurement from the sensor all we need to do is update our previous estimate incrementally

(Refer Slide Time: 25:47)



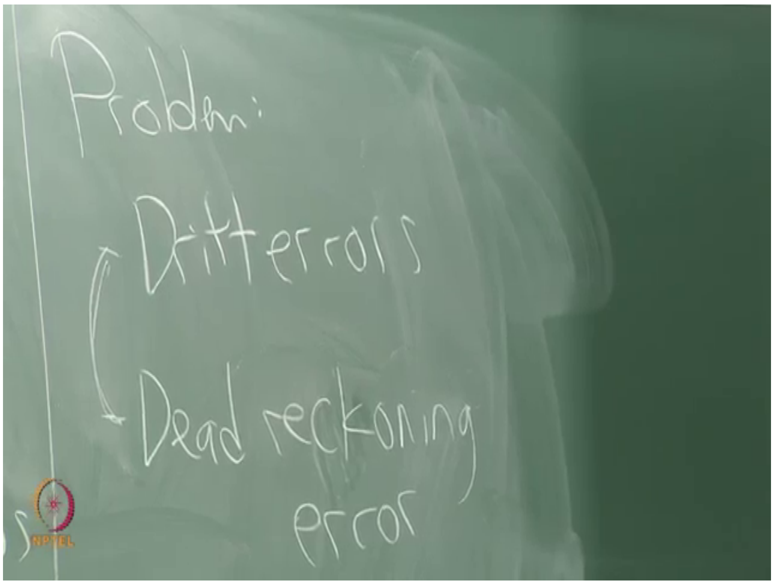
So, if we think of it in that way we can write it like a nice recursion. So, recursively in 2 D we might say our current estimate I will just write word subscripts here our current estimate is equal to the change that is occurred in the last millisecond in terms of our estimates plus whatever our previous estimate was correct.

So, that is just one very simple way to look at in the 2 D case I just get the current change and just add it to the previous estimate that gives my new one. So, I have a very small piece of software just, take how far I think I have rotated in one millisecond and add that on that is all we do each step. So, this is just looking at it in recursive or incrementally if you like does not look like in recursion too much because I have not used the proper kinds of subscripts. In the 3 D case cube current is just equal to the change quaternion product it with my previous estimate.

And, so, there is our nice sort of result for the day you just estimate, what the tiny change was in one millisecond in terms of access and angle, which you can directly read from the gyro you just take the magnitude of the 3 D gyro reading and in radians per second that tells you how far you have rotated and then you look at the direction of that 3 D vector, and that tells you the axis of rotation. You form a simple quaternion just multiply it on and you just do these simple updates when you do that you are integrating the gyro readings questions about that. So, what happens if I keep doing this for I do not know say 10 seconds or an hour, air is tend to accumulate right. It may be the case that my gyro has a little bit of a bias to it, based on the temperature based on the packaging stress, it can be all kinds of reasons for that. So, if there is some kind of bias it will start to accumulate maybe it is just a bunch of noise that is accumulates over time.

So, that we start to get what are called drift errors.

(Refer Slide Time: 28:09)



So, the problem that occurs our drift errors a lot of people may know this has dead reckoning errors this is a term used in robotics and other fields for example, its been around in navigation for many many years dead reckoning error reckoning. So, drift air dead reckoning error same thing essentially. So, these are equivalent concepts.

So, how do we correct for something like that? Well if you have no other sensors at your disposal you are not going to be able to correct for it right. So, we need to get more sensors from somewhere, and I am going to talk about that in the next lecture. So, I will ah open up with going into drift correction techniques, which will help compensate for these drift errors. So, if we have another signal coming in that can tell us errors gives us some kind of estimate of the errors, that are accumulating we need to figure out how to apply some kind of correction. If you are not careful you could accidentally apply the corrections in a way that nauseates people. So, imagine you have air is accumulating and now the world is becoming tilted, you decide to quickly apply some tilt correction and that causes an optical flow, that makes people feel like they are on the ocean on a boat and so, that could be very bad.

So, there is a kind of delicate balance between applying drift corrections and making sure that it is perceptually comfortable. So, we will have to get to that for next time.

Thanks.