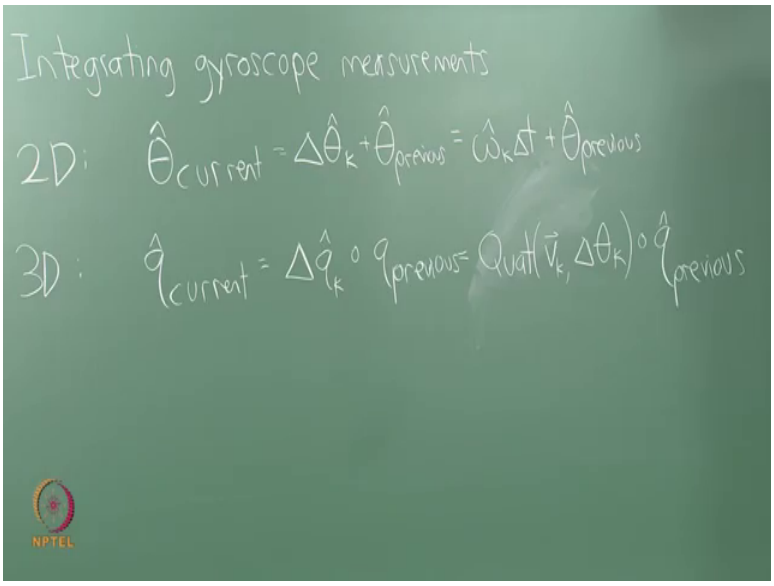
Virtual Reality Engineering
Prof. Steve Lavalle
Department of Multidisciplinary
Indian Institute of Technology, Madras

Lecture – 13
Tracking Systems (tilt drift correction)

(Refer Slide Time: 00:16)



Everyone, let us get started welcome back remember that last time we went over perception of motion. We talked about stroboscopic apparent motion and then we got started on tracking and in particular I am looking at head tracking. And we got into the case of integrating gyroscope measurements which are providing readings in terms of angular velocity.

So, radians per second and in the 2 dimensional case I gave a simple example imagining a merry go round or a top spinning and if you get measurements of that here is the here is the example of the measurements just scalar measurements; essentially it is rotation with respect to the vertical z axis. So, you have omega k hat at each step you have delta t which is 1 millisecond for our examples we have been giving.

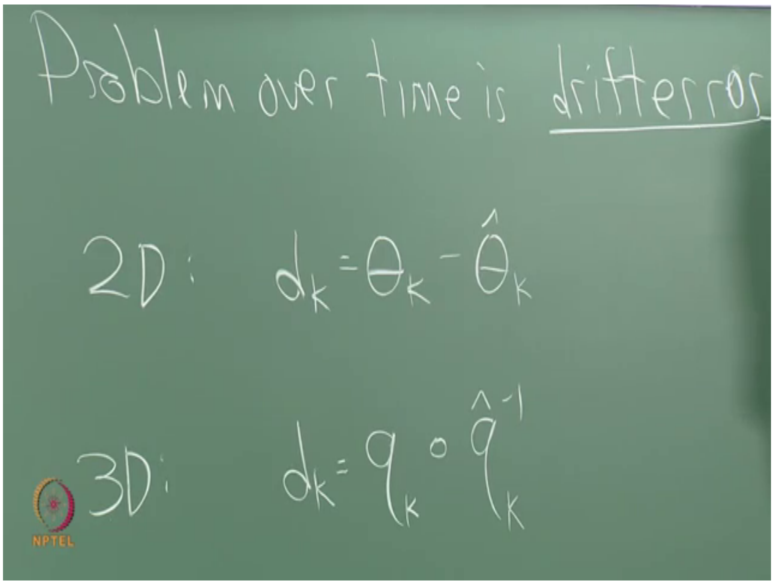
And so, we just have this very simple update; we calculate some change in the theta estimate and just start adding. We add an each step here and just keep repeating this equation over and over again. And then I showed you that the 3 dimensional case is very similar except that we switch everything to quaternions and so, the replacement for this omega k; delta t is constructing a quaternion that will give you the change that occurs during delta t, it is an entire small rotation about some particular axis V k and the total amount of rotation is theta k.

So, these together become delta q sub k hat ah; I put quat here in front of it to make it look like a constructor from a class let us say suppose you have a quaternion class in c plus plus. So, just to emphasize that there is a quaternion that is constructed here. And remember these two parameters come just directly from the gyro reading; if you just normalize the omega x, omega y and omega z components; you would get the axis of rotation for here.

And then the magnitude of that original omega x, omega y, omega z vector will give you the total amount of rotational change estimated for that 1 millisecond right; you can remember that all right. So, if we do this over time what we mentioned at the end of class is that; we end up with what is called drift error.

So, the problem over time is called drift error.

(Refer Slide Time: 02:28)



So, it is a gradual divergence between the true orientation and our estimate of the orientation. And these will gradually get worse and worse especially because most of the errors that we have in this process are systemic, they are due to calibration difficulties in some way. So, it may be that the gyro along the x axis always gives you a reading that is an over estimate by a certain amount and that will just continue to increase the drift air linearly alright.

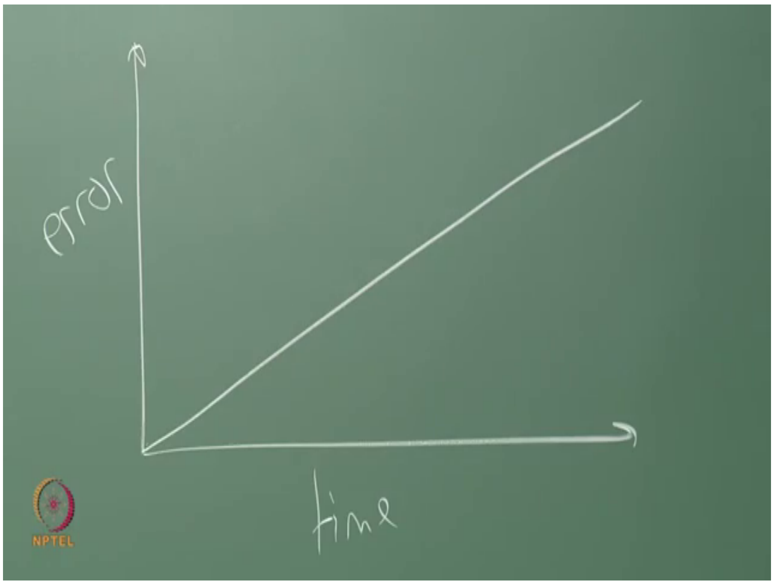
So, if we think about drift errors in 2 D ; I could say that at each step the drift error put d sub k for drift is the true orientation theta k minus theta hat k in this case it is assigned drift error. And in 3 D, we could say that the drift error in this case I want to subtract two orientations, but if I am using quaternions; I have to use the algebra of quaternions and we have to also remember non commutativity. So, I want to take my true quaternion minus the estimated quaternion in defining this drift quantity.

So, let us suppose the true quaternion or true orientation of the head if that is what you are estimating is q sub k. Then I want to essentially subtract off my estimate q hat minus 1 I guess that could go in either direction with this as well I will get two different results, but they are both measurements of the error right. So, depending on which way I want the sign I could have also written q hat minus 1 sub k quaternion product q sub k right there is; it is non commutative it will give me an error in the essentially the other direction is this ok.

So, so the concept here I mean normally for most problems you have dealt with in engineering; you may talk about a scalar error right there is some real valued error it is nonzero um. And therefore, needs to be reduced or eliminated in some kind of way, you may have considered a vector of errors before just keep in mind that the drift error in this 3 dimensional case here it is an entire 3 D rotation; so, it is a quaternion.

And so, if you apply the inverse of that you will be correcting the error and so, this one it is a kind of unusual quantity to have for an error. So, the error is that we get.

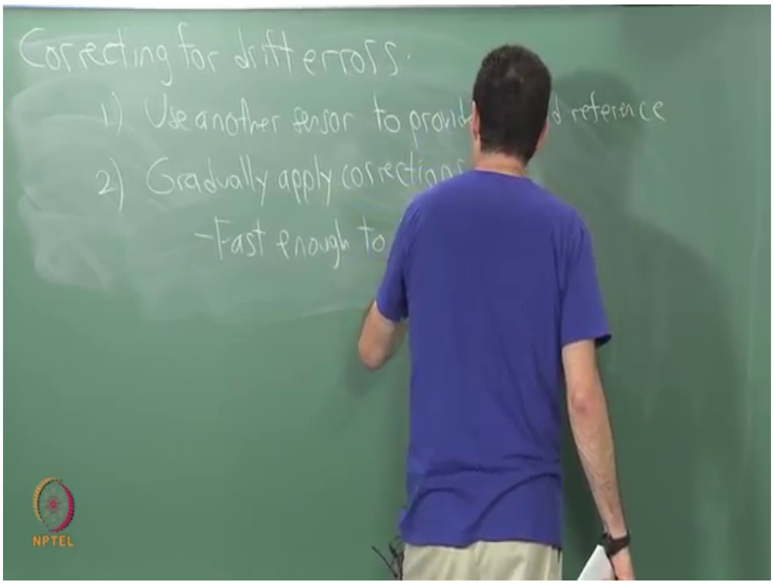
(Refer Slide Time: 05:20)



The drift errors tend to accumulate over time. So, drift error if left unchecked may grow out a linear rate, may grow faster than that, may slower than that it depends on a number of things, but very commonly if you go take plots of the drift error of a gyro after doing integrations um, you will very often see linear behavior especially for a fixed angular velocity.

If you go faster or slower the rate may change and these curves may go up or down and then there is also notice factors and some as well if the gyro is very well calibrated then it may have a different behavior that is mainly dominated by say white or Gaussian noise. So, what to you next is try to use other sensors to try to reduce this drift error back down as close as possible to 0.

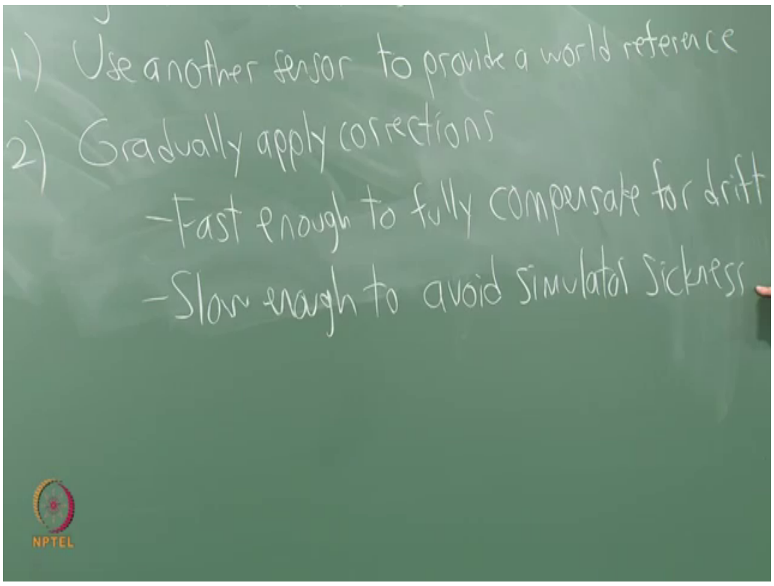
(Refer Slide Time: 06:09)



So, correcting for drift errors. So, two key points for correcting for drift errors one use another sensor or sensors to provide a world reference. So, in other words some kind of reference information with respect to the world frame as the body is rotating taking it is measurements from it is body frame.

And then we want to somehow gradually apply corrections and this is a difficult problem, it has to be fast enough whatever corrections we apply they need to be fast enough to fully compensate for drift.

(Refer Slide Time: 07:30)



But at the same time, they need to be slow enough to avoid simulator sickness or other kinds of disorientation or discomfort.

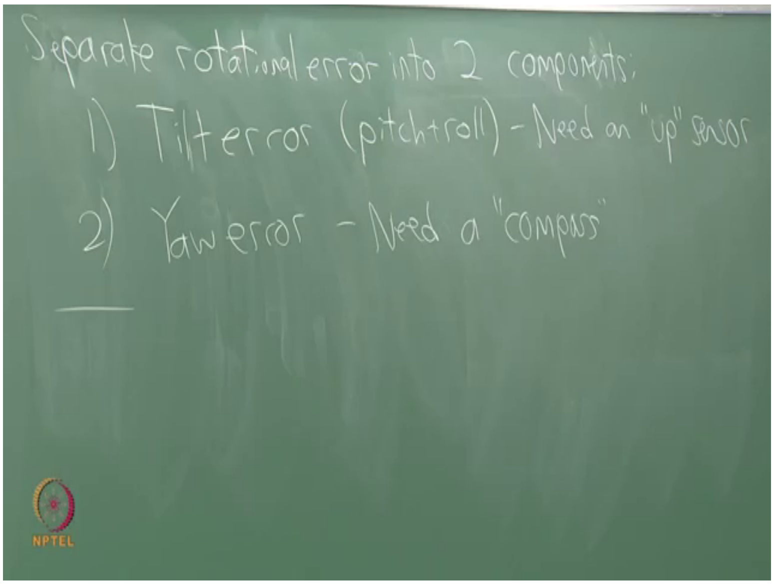
So, suppose for example, you are running along for 30 seconds and then you get a sensor reading from some other sensor that is not a gyro that tells you you are off by 30 degrees. Do you just all of a sudden apply that right or do you gradually apply it within the course of a second; if you do that if you can perceive that the world is all of a sudden tilting the virtual world would be tilting.

So, you have to provide it in a very gradual way it may be the case that your gyro is so, bad and this drift rate it grows so, fast that there is no way to compensate for it without causing some kind of sickness. So, so it could be the case that it is so, sloppy that that you cannot compensate quickly enough and still be within thresholds. It turns out we are very fortunate in today's world of MEMS based gyroscopes that appear on our top of the line smart phones these days.

That the gyro quality especially if the gyroscope has been very well calibrated especially for temperature dependence and other factors; then the growth rate ends up being slow enough. So, that it is possible to compensate for this without as far as we can tell without perceptual side effects causing simulator sickness.

So, we are very lucky to be in this kind of regime today where we can in fact, accomplish this. So, let me give you some of the technical details of how these corrections work. So, we would like to divide the error up into 2 components.

(Refer Slide Time: 09:09)



So, separate rotational drift error into two components. One is called tilt error and this can be viewed as a combination of pitch plus roll right tilt there. So, has the world become tilted; so, it may be the case that my virtual world as I am doing the head tracking appears to have a tilt in it all right. So, this can be any kind of pitch or roll imagine just a few degrees of it um.

So, tilt same kind of thing you might imagine in the real world corresponds to cases in which you are not aligned with gravity right, where the surface that you are walking on it appears to be tilted; a ball will start rolling in any direction and in any one of a number of directions if there is a tilt to it.

This is separate from another error which is the Yaw error it is remaining component. And this has to do with losing track of which direction you are facing alright. So, so the Yaw is the remaining component we had troubles with this early on it at oculus for example, because when people were playing a cockpit game. So, imagine you are sitting inside of a car and you have a dashboard or you are sitting in an aircraft or spacecraft as a cockpit; after 10 or 20 minutes of play the cockpit starts drifting to the side. And you can sit in your chair and if you have a rotating sphere you could rotate with it without; even realizing and then before too long you are tangled up in the cables because of this Yaw drift error; so, then it being a very serious problem.

So, the reason why they are divided into Tilt error and Yaw error is because the kinds of sensors we use to correct for these tend to fit very naturally into these two categories. So, I am going to first talk to you about how tilt error is typically corrected and then and then we will go in a little bit into Yaw error, but it works very similarly all right.

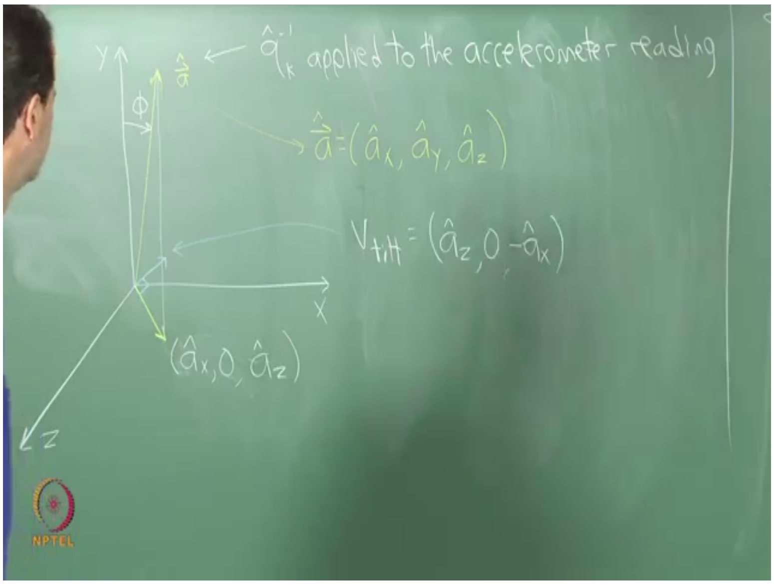
Now to correct tilt error what we would like to have ideally we need I would like to have an up sensor. In other words a sensor that measures which way is up in the global frame that would be nice will not it because that is what the definition of tilt error it is some mismatch with respect to up right some deviation with respect to up.

And then in the case of Yaw error; we would like to have some measurement of which way it was forward initially and we would like to keep maintaining that. So, we need something that behaves like a compass; now it turns out that in the real world we can never get either one of these very precisely, we do not get perfect devices that will give us these.

So, what we were going to use in the case of measuring tilt error instead of our up sensor; we are going to use a linear accelerometer. And in the case of instead of having a compass in case of this, we were going to instead use a magnetometer which measures magnetic fields which might include the earth’s magnetic field and it may include a many other fields superimposed together.

So, let us suppose we have this perfect up sensor and it is accomplished by a three axis three axis accelerometer. So, it is measuring X, Y, Z components as meters per second squared and then I want to take a look at this in a particular coordinate frame.

(Refer Slide Time: 12:54)



So, let me let me draw out a coordinate frame here have X, Y, the Z coming out as usual. And then I am going to draw in this frame and then I will say what the frame represents. I have this vector here I will call it a with a hat on it representing some kind of estimate which is a measured acceleration.

And this what I would like this to represent is my current estimate of my orientation, but the inverse of it applied to the accelerometer readings this should be the current accelerometer reading. So, in other words on the device on the sensor itself which is moving with your head; I have an accelerometer it is measuring meters per second squared acceleration in the X, Y and Z directions. Let us assume just for a moment and I will return back at this assumption; let us assume that your head is actually not undergoing some linear accelerations at present or the sensor is not undergoing linear accelerations; if that is true what would an accelerometer measure?

Student: (Refer Time: 14:27).

Right just gravity right just acceleration due to gravity; so, let us just assume it is going to measure that component alone. If I take that measurement that measurements going to be done in the in the body frame because that sensor is attached to the body, but if I believe that I have constructed a nice estimate q hat sub k of the of the transformation; that will move the body into its frame from the world frame, then I should be able to take the measurement and transform it back into the world frame using the inverse.

So, that is what I do is I take the measurement I bring it back into the world frame and if that is if my if my sensor is correctly reporting up. And if my estimate of this orientation of the body is correct, then this yellow vector should be aligned perfectly with the Y axis instead what typically happens it will be off by some amount.

So, the amount that it is off I will write as phi and this will be some estimate of the drift error. Suppose I should put a hat on that if we keep using hats for estimates alright. So, if that is the case what I would like to figure out is what needs to be done to bring those two into alignment and so, you can look at it either way you could try to bring the yellow arrow up to the white wire or rotate the coordinate frame Y to line up with the yellow arrow I need to take a look at what should be done here.

And so, let us take a look at this a vector and let us; let us write it out as a equals and remember this is already transformed; I have already applied this q inverse to it. I have three components a x hat, a y hat and a z hat all right. So, there is three components to it and now what I would like to do is perform a rotation that is not cause any Yaw to change; I just want to change pitches and rolls in order to do that what I want to do is find an axis in the X Z plane right.

So, if I can find an axis and I always do rotations about axes that go through the origin so, just imagine that I have I need some axis that goes through the origin; it also lies entirely in the X Z plane. So, when I grab on to that axis I can perform a twist that will bring these two together right. I will grab on the axis you can imagine it is attached to maybe the whites let us say and it just performs a twist that lines it up with the yellow.

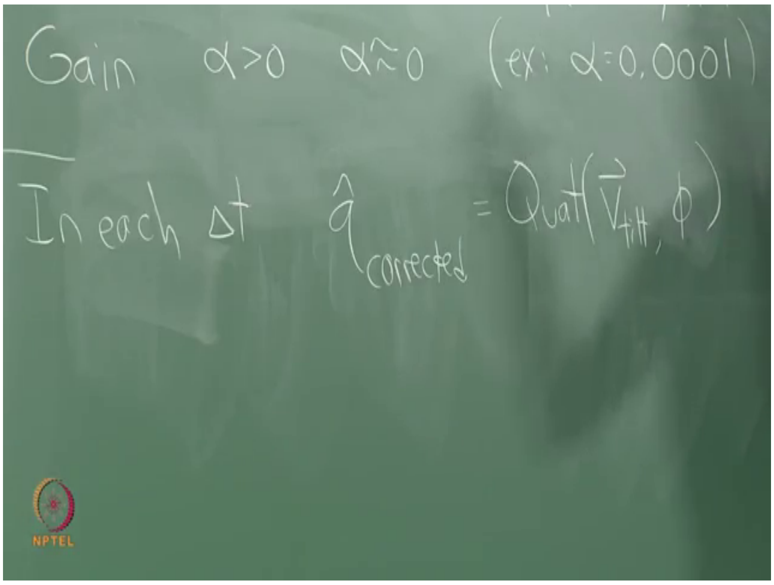
So, that is what I want to do. So, easy way to do that is I just take this a vector and I project it down into the X Z plane; so let us try that. So, I project that into the X Z plane that gives me a vector I will call it V tilt which would be the tilt axis. I am sorry that is just the projection not yet V tilt sorry; I need to do an or operation this axis here is a x hat 0 a z hat correct. So, I just get a 0 component for the Y.

Now, I want to pick a perpendicular vector to that and since I am only really dealing with two dimensions now; I want to pick a perpendicular vector. So, I just you may remember you can just simply swap the two coordinates and negate one of them right which one you negate will tell you which direction the vector points.

So, I want to obtain the following axis was this one what should be perpendicular and I will call that one V tilt and that will be a z hat 0 a hat x minus a hat x I got it negate one of them in order to get an orthogonal vector. So, this will be the tilt axis what I want to do is grab onto this and perform a correction by performing a rotation.

Now, as I said I do not want to perform the entire corrective rotation at once. So, let me give you a very simple way to do the correction and it is equivalent to many alternatives simple way to correct.

(Refer Slide Time: 19:10)



So, you just call this a complementary filter. So, in this filter we will have what is called a gain or gain coefficient alpha greater than 0, but alpha very close to 0.

So, for example, maybe alpha equals 0.000 1 the particular rate of alpha the particular value of alpha is going to depend on the sampling rate which is delta t is 1 millisecond for our continuing example we have been talking about. So, now in order to correct for drift in each time delta t in each step we have now a corrected orientation; which will be some quaternion we generate V tilt and at this point; I could apply the entire theta here and try to correct for it. What is going to happen though is that it is very likely that my theta is a noisy estimate and it is too much to correction to apply all at once for perceptual reasons that I talked about.

So, if you try to apply this and I did some experiments like this in the code myself; I tried applying corrections all at once, you end up with a very unstable view the orientation just changing wildly all the time. So, this is where we use this gain coefficient; so, it is alpha times phi. So, that will perform a very tiny rotation in each step that is making progress in the right direction. And then it just continues like this you end up with a very stable representation of the orientation.

And you end up with a very stable view, but this alpha needs to be large enough so, that it is applying the corrections fast enough to compensate. One of the things we did was perform experiments with a robot moving the sensor through space at different angular velocities to determine upper bounds on the maximum amount of drift error that could occur. So, that we could set the coefficient large enough to overcome that and luckily as I said it was still within thresholds so, that it did not seem perceptible or sickening to the users though. So, we got very fortunate with that.

So, we apply this correction to the to the previous estimate q hat questions about that? Let us see. So, what I want to mention now is a problem with the accelerometer yes.

Student: (Refer Time: 21:58).

Um that is a good question. So, I; I every time I go through this I have to go back and ask myself is it minus or plus based on the way I have set up all the coordinates right. So, one of the possible ways to change it from plus to minus depends on which way I have made this axis right.

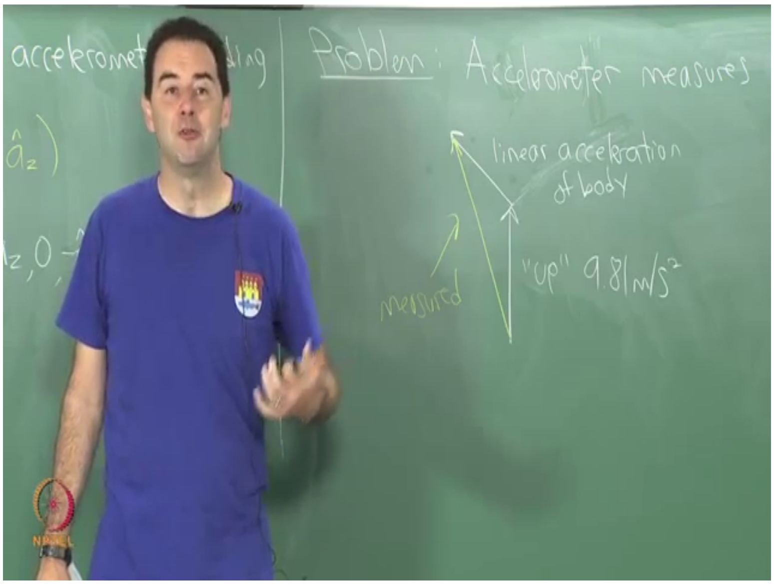
One of the ways it could be plus or minus depends on whether I am rotating a or rotating this and am I applying the inverse of the forward. I checked it this morning I believe it is positive, but maybe we should go back over it again during the break and see if we find an error. So, unless you have a unless you are convinced it is minus. I; I.

Student: (Refer Time: 22:39).

I think it is correct in plus, but um, but again it is quite possible somewhere along the way I have made it may have inverted a step. If you write the code for this you very quickly see that you have made that you have made a mistake and you can go back and check each one of the components. Of course, I never suggest encoding just changing signs until it works because good he will end up with quite a mess after doing this two or three times.

But I believe it is positive and the way I have set it up, but, I admit him I might have made a mistake because there is several steps in a row here; each one of which could flip the sign. I think in this case I am applying an additional rotation to move this frame towards the yellow. So, that is why I think it is positive alright ; that is very good I was anticipating a question you someone usually as an I usually ask it myself and like I have gotten the sign right is very very important to see that to check that all right other questions?.

(Refer Slide Time: 23:45)



So, let us see one big problem with this as I mentioned is that the accelerometer is not a perfect up sensor. Let me put that here I guess; it is a problem the accelerometer measures the following vector sum, there is the the up part which is 9.81 meters per seconds squared.

But there is also the true linear acceleration let me draw off in a different direction here; there is also the true linear acceleration of the body. So, linear of the body and so, it is actually the vector sum of the two that is getting measured by the accelerometer right this is what is measured.

So, what do you do in that case? Well you could invent some simple heuristics I have seen in some published papers for example, someone might say well if the magnitude of this yellow vector the part that you are trying to the measurement that you get; if it is close to 9.8 within some small epsilon of 9.8; then assume that this part must be 0 is that a reasonable assumption? Well it is not bad except I should be able to move downward and that compensates for some of gravity right because if I were free falling; I would completely cancel off this up vector.

So, I should reduce the up vector by moving down in some way, but I can also linear x linearly accelerate while doing them and I should be able to get it so, that I convince the drift correction system that makes this assumption that the world is tilted by a large amount right. So, you end up with this kind of situation of making some heuristics to try to separate out these two vectors; so, that you can get a reliable up reading.

This particular assumption by itself will not work all the time; it is reasonable in many cases, but you have additional information because you also have access to the gyro which is giving you the angular velocity. So, if you all of a sudden get an estimate of up that is extremely far you know it is very very different from what you previously believed up was in the global frame. And it happened in a short amount of time right you can very easily also com compare that with the gyro history because you are integrating the gyro ford and the drift rate is not too bad in a short amount of time.

So, you could give very low confidence or rule out very unusual gyro very unusual accelerometer readings that seem to indicate that up is say 30 degrees off from what you want to believe. Now if the person continues moving their head very wildly and violently for a very long period of time, there may be no good way to compensate and you may be able to kind of fool the system into believing that there is a significant amount of tilt before it is able to go ahead and correct that. So, that is one of the strange outcomes of that I am so.

And it is not really the fault of any engineer in particular it is a problem of the fact that when we measure accelerations in some moving frame, we always get this vector sum and there is no way around that. And we are just trying to separate it out, we do not have the ideal up sensor and it; no one will be able to make one right purely based on inertial measurements.