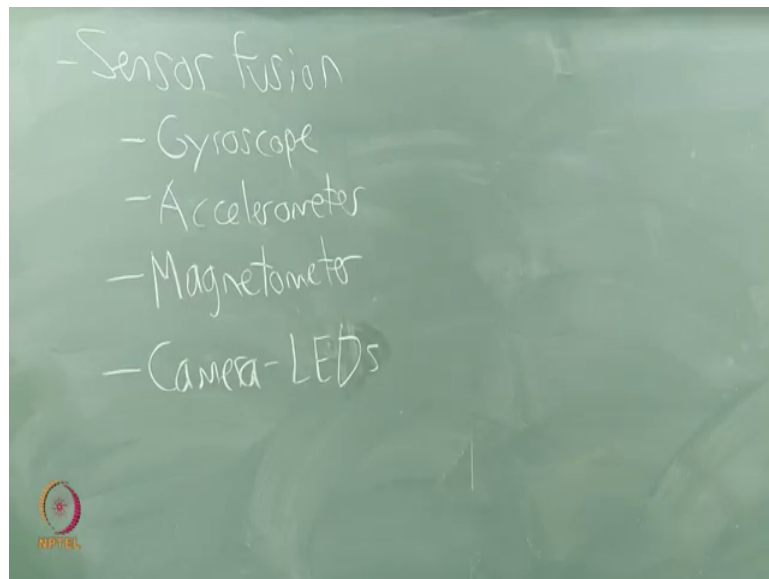


Virtual Reality Engineering
Prof. Steve Lavalle
Department of Multidisciplinary
Indian Institute of Technology, Madras

Lecture 14 - 1
Tracking Systems (filtering)

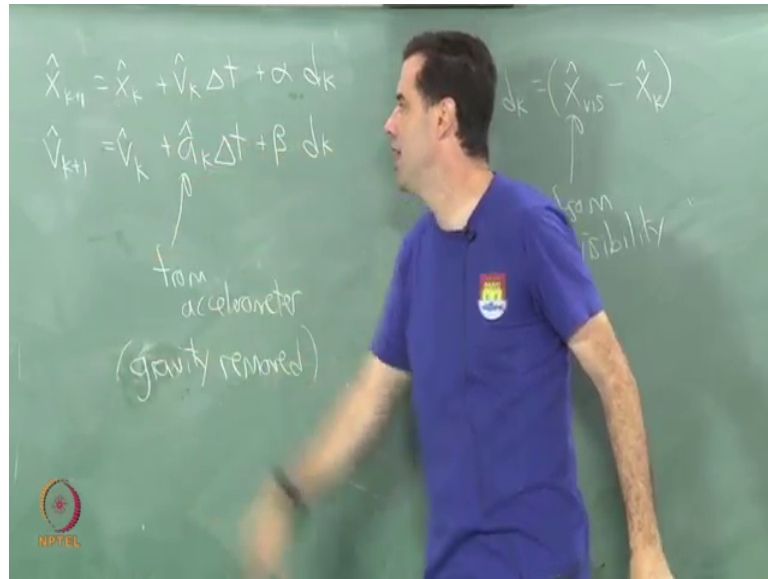
All right. So, when we put it all together.

(Refer Slide Time: 00:16)



We can imagine that we have a sensor fusion going on right, how many different sensors do we have remember we have the Gyro, we have the Accelerometer, we have the sometimes flaky Magnetometer, and now we have a Camera LEDs paired together LEDs paired together right. So, camera with LEDs paired together offering line of sight visibility, when we just had the first 3, we were one thing we said..

(Refer Slide Time: 01:59)



we could do is a very simple complementary filter. So, we get this drift correction signal we get this drift error estimate and, based on that we perform tiny bits of correction in each step.

So, we can do the same kind of corrections for the case of position, but it becomes a little bit more complex because in the original case up here, for orientation only the gyroscope that I am integrating is only one level of integration right, I integrate it once and that provides the orientation, because it is angular velocity that I am getting what I can do in the case of position is I can double integrate the accelerometer. So, that is 2 levels of integration and, then I can correct that drift using this camera LED combination as my drift error signal.

So, just to give an idea of how that signal is used let me just write it out in a very simple way. So, so, let us suppose I have \hat{X}_{k+1} I am just taking a single coordinate here, we can just imagine this is just the this actually is just the X axis coordinate, but it will be the same way for y and z and the update goes like this is my previous estimate x_k plus, I have some estimate of my velocity linear velocity along the X axis multiplied by Δt plus α and, then d_k and d_k is going to be the drift error.

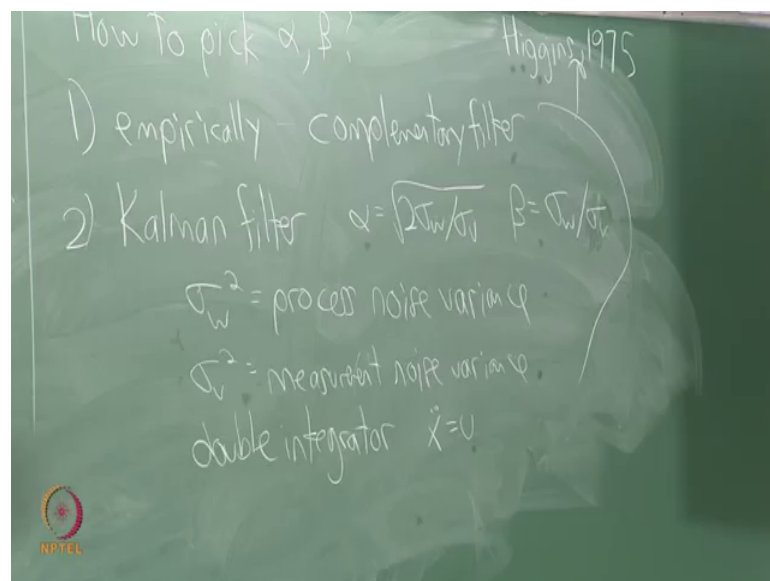
Let me just write that out d_k is equal to let us say \hat{X}_{vis} based on visibility. So, from visibility or in other words from the camera minus this is what I have estimated inside of my filter that I am building here. So, I get some measurement of the error and, I need to

correct for that. So, I am going to correct using this term and, then I also have an update equation here for the velocity. So, my filter now has to keep track of an estimate of both the position and the velocity, and if you remember the filter in the original orientation only case, it was it was directly estimating the orientation, and a sensor was providing the angular velocity. Now I have a filter that is estimating position and velocity, and I have a sensor that is gonna be providing the acceleration directly which is the leftover component of linear acceleration.

As measured by the accelerometer. So, just trying to give a little bit of a feel for what goes on at this stage, but I am not going to be covering all the details. So, we have V_k plus the accelerometer measurement here. So, this is from and it is just a linear acceleration component. So, put up a gravity removed with more detail that would be the acceleration component due to gravity is removed Δt plus and, then another coefficient times this d_k .

So, we just make this very simple filter that is updating an estimating position and velocity based on the accelerations, and then it tends to drift, but the drift is corrected by this signal which is using visibility information and so, there is some amount of correction here using these alpha and beta parameters.

(Refer Slide Time: 04:48)



If you tune these empirically. So, so how to pick alpha and beta again we have the same kinds of issues that we need to pick them large enough. So, that they fully compensate

you need to make sure they are not fully compensate for the drift air, you need to make sure that they don't they are not. So, large that they lead to instability in the tracking which causes some additional simulator sickness. So, that to be careful of that, if you choose them fully empirically just by trial, and error this will very much fall under what's called a complementary filter, but it is second order or 2 stage complementary filter it is not like a usual complementary filter because there are 2 different coefficients here for each axis.

And you could choose them according to common filter gains, this would be the optimal gains for a linear system operating under gaussian IID noise and, then this will become the optimal filter, but this in fact, is not a linear system and that would be a bad approximation to this kind of system I think and most of the uncertainty in the system for head tracking is due to the fact that you do not know which way the person is going to look next, that is where most of the uncertainty comes from not from noise, and most of the errors you have are also due to calibration errors in the various sensors that you use rather than gaussian white noise or something like that..

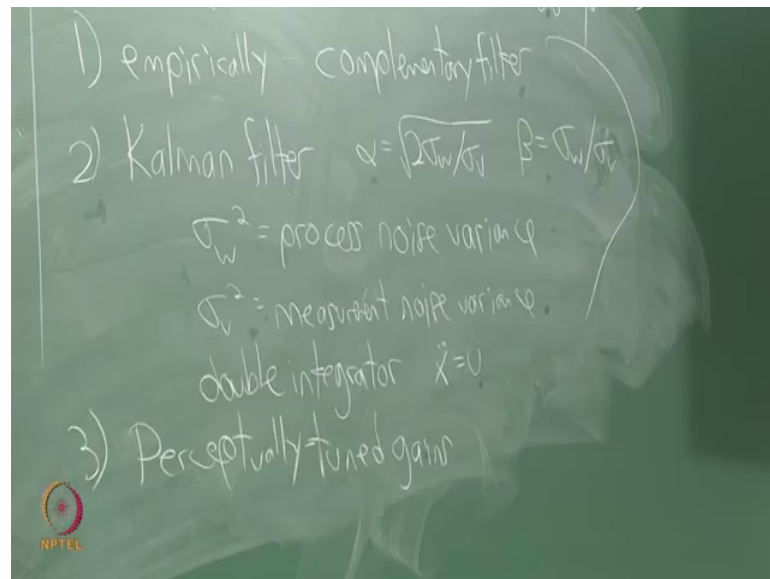
So, but nevertheless you can use a common filter to calculate, these values they would be $\alpha = \frac{\sigma_w}{\sigma_w + \sigma_v}$ and $\beta = \frac{\sigma_v}{\sigma_w + \sigma_v}$, where σ_w squared is the process noise variance, I will explain these in a minute σ_v square is the measurement. So, measurement noise variance means the variance in the estimate of the acceleration that is coming in and process noise variance means in the it is the variance in the update equations. So, in the sum as a linear process. So, if you were to model this as a linear system with noise, it would be a very simple system called the double integrator.

Which in terms of control theory notation is $\ddot{X} = U$ where U is the input to the system, the input to this system under this model would be direct acceleration. So, the acceleration is essentially the input as measured by an accelerometer and, then you formulate these as the process and, measurement error models and, then if you go through the solving the algebraic riccati equation that gives you the common gains you will get values for α and β , if you just tuned them empirically it ends up being a form of a complementary filter..

So, these 2 end up being let us say equivalent processes in this setting, and there's a nice paper by Higgins 1975 that explains how these 2 are equivalent in this context of the common filter and complementary filter and, then in this context end up being very simple and equivalent. So, that avoids doing a lot of extra complicated work, and you can safely avoid use of let us say more powerful numerical, and linear algebra machinery for computing common gain coefficients because, you can just directly set these complementary filter values.

And 3 a third choice here which I should point out is that you may want to what maybe I could fall under number 1, I just want to point out that.

(Refer Slide Time: 08:57).



Perceptually tune gains may be the best in any case. So, rather than applying a standard black box method such as common filter which is for some other problem, which is linear systems with a gaussian noise, you may want to tune these gains alpha and beta, and do experiments with human subjects, and figure out how to adjust them in such a way that gives ultimate performance in terms of people's perception, and comfort in v r which I feel like is the right way to solve the problem over all.

Any questions on that? So, I realize I have not really covered everything fully, but it just gives you some idea there's like a essentially a 2 level complementary filter going on here, where you are trying to correct for the drift. The accelerometer is serving the same

role here as the gyro did for the orientation componen, and the orientation part is still working here, the same way we just added this new part for position.