Simple kalman filter example

There are a ton of Kalman filter overviews online, and lots of them give a general overview of what they do, but then they all hit a wall of variables and matrices, and fail to give good simple examples.

The best guide I found is a PDF scan of a much-faxed copy of Roger M. du Plessis' 1967 classic "Poor Man's Explanation of Kalman Filtering".

His paper is great because he starts with a single-variable filter (no matrices!) with no control component and no prediction algorithm. I'm going to try to give an even simpler example with softer terminology and more hand-waving.

Kalman filters are a way to take a bunch of noisy measurements of something, and perhaps also some predictions of how that something is changing, and maybe even some forces we're applying to that something, and to efficiently compute an accurate estimate of that something's true value.

Let's say we want to measure the temperature in a room. We think it's about 72 degrees, plus or minus 2 degrees. And we have a thermometer that gives uniformly random results within a range of +/-5 degrees of the true temperature.

We take a measurement with the thermometer and it reads 75. So what's our best estimate of the true temperature? Kalman filters use a weighted average to pick a point somewhere between our 72 degree guess and the 75 degree measurement. If the weight is large (approaching 1.0), we mostly trust our thermometer. If the weight is small, we mostly trust our guess and ignore the thermometer.

Here's how we choose the optimal weight, given the accuracy of our guess and the accuracy of the thermometer:

```
weight = temperature_variance / (temperature_variance + thermometer_variance)
                        0.29 = 2 / (2+5)
```

If temperature_variance were very large compared to thermometer_variance, weight would approach 1.0. That is, we'd ignore our guess and just use the measured value. Likewise, if thermometer_variance dominated, the weight would approach 0, and we'd put very little trust in our thermometer readings.

29% weight means we'll trust our guess more than the thermometer, which makes sense, because we think our guess is good to +/-2 degrees, whereas the thermometer was only good to +/-5.

Now we do the weighted average:

```
        estimate = guess + weight*(measurement - guess)
               72.87 = 72 + 0.29*(75-72)

                       or equivalently

        estimate = (1-weight)*guess + weight*measurement
               72.87 = 0.71*72 + 0.29*75
```

That is, we went 29% of the way from 72 to 75, or equivalently, we took 71% of the guess plus 29% of the measurement.

There's one last thing to compute: how confident we are in our estimate of 72.87 degrees. That equation is:

```
                               temperature_variance*thermometer_variance
        estimate_variance =    ---------------------------------------
                               (temperature_variance + thermometer_variance)

                       1.43 = 2*5 / (2+5)
```

So we think our estimate is correct to +/-1.43 degrees.

Now we have a guess that the temperature in the room is 72.87 degrees, +/-1.43 degrees. And we still have a thermometer that tells the temperature +/-5 degrees.

That's basically the situation where we started, so we can run the whole algorithm again:

First we compute the weight, using our new, more accurate guess confidence:

```
    weight = temperature_variance / (temperature_variance + thermometer_variance)
                       0.22 = 1.43 / (1.43+5)
```

We take a measurement, and this time let's say it comes up as 71 degrees.

Now we can compute the weighted average of our old guess and our new measurement:

```
            estimate = guess + weight*(measurement - guess)
                  72.46 = 72.87 + 0.22*(71-72.87)
```

And the new confidence level:

```
                               temperature_variance*thermometer_variance
        estimate_variance =    ---------------------------------------
                               (temperature_variance + thermometer_variance)

                       1.11 = 1.43*5 / (1.43+5)
```

So after the second measurement, we estimate that the actual temperature is 72.46 degrees, +/-1.11 degrees.

Kalman filters are nice because we don't have to remember the whole history of measurements and estimates. We just keep track of our most recent estimate and our confidence level in that estimate.

If we decide to turn on the air conditioner, so that the temperature in the room starts decreasing, we can expand our calculations to include that "control" variable, and use it to update our estimates by "predicting" how much colder it'll be at each measurement. Once I figure out how to do that, maybe I'll post a more complicated example :)