**Note:**

**About the measurement noise(R) and process noise(Q)**

The *measurement noise* standard deviations can be based on physical characteristics of your sensor and/or measurement process. The process noise is usually treated as a tuning parameter to adjust the gain of the Kalman filter to smooth either more or less the data.  For instance, if your state is [x, dx/dt] then a common process noise model (called almost constant velocity) is Q=[T^3/3, T^2/2; T^2/2, T]q (in Matlab) where q is a positive scaling parameter and T is the sampling time.

In addition,

1. Measurement Noise(R): Represents (electronic, random) noise characteristics of the sensor. It is calculated from the sensor accuracy which is represented using "standard deviation" of measured value from true values (sigma) during the calibration.

sigma\_sq = sigma^2; % MATLAB/Ocatve code  
 R=sigma\_sq\*eye(3,3);  % example for 3 axis position measurements

 2. Process Noise: Decides the accuracy and time lag in the estimated value. Higher Q, the higher gain, more weight to the noisy measurements and the estimation accuracy is compromised; In case of lower Q, the better estimation accuracy is achieved and time lag may be introduced in the estimated value.

Choice of process noise is based on the objective i.e. estimation accuracy Vs time lag (with respect to true value) trade-off.

To clarify further (after a question from Harith), the objective is to estimate the state of the system, i.e. the user location in this case.The R qnd Q matrices in the system model are not unknown quantities. They have to be known to run the KF. As per previous discussion, **R is known from the physics of the sensor/measurement process and Q is best seen as a tuning factor**. Q can be zero and the KF will still run (as long as the system is observable) but it will pretty much ignore the data (gain approx = 0). You need some Q>0 to allow the filter to weight current observations via the Kalman gain. If the filter is not working with Q>0 then you may have an observability problem or a mismatch in your system model (i.e. the model is wrong). Another problem is that you may not have enough data points if your data are very noisy (which they may be for RSSI), or that you have outliers in the data, which requires some validation logic. The assumptions of the model must be respected for the filter to work properly.

Fair enough, but it's important to use the physics of the system as far as possible to adjust the R matrix, which is measurement noise. If you have to inflate it too much just to "tune" the filter, there might also be some modeling errors, e.g. **the noise is not Gaussian and has occasional large values**, etc. In this case, you might need a different type of estimator that can handle the different error distribution. I agree that the Q matrix is in practice just a tuning parameter. It's important to make sure that the structure of Q is right though since it is the state error covariance. Having fixed the structure of Q, it can be inflated or reduced with a single or small number of parameters to control the filter response to the data.

**How to determine measurement noise**

You can try to find a spec sheet for the sensor.   
Usually the model is of AWGN (Added White Gaussian Noise), so you only need the standard deviation.  
Another way would be to compare to a more accurate sensor, or measure something that you know the ground truth for. For example, for an accelerometer on a table, the reading should be 1g in the up direction and 0 in the xy plane.

The usual method is to take measurement is a controlled environment where you know the true value. Then, just use a variance estimate on the error.If you don't know the true value, but the target isn't moving, you can just use a variance estimate on the measurements themselves.If the target is moving, and you don't know where it's true position is, you'll have to take some kind of mean estimate, aka regression. When you have a regression that you are satisfied with, estimate the variance on the residuals.

**Kalman filter is actually not a filter,it’s actually an estimator**

In my opinion, the need of use of Kalman filter depends on your application. Just for “noise” filtering you can simply use low-pass filter. If you want to filter the “gravitational acceleration”, add a sequential high pass filter. It will work well. However, if you want to eliminate uncertainties, like different sensibility for x,y,z axes of accelerometer, take into account biases for every of its axes, use Kalman filter! Kalman filter it’s not a “filter”, it’s a predictor, or model of your accelerometer, with biases, sensibilities and noise, or what you will consider.