



Tuning the Performance of the VN-100

Understanding the Principles of a Kalman Filter

The VN-100 uses a sophisticated sensor fusion algorithm called an Extended Kalman Filter to calculate the orientation of the sensor from the raw measurements made by the onboard MEMS sensors. A Kalman filter is an optimal recursive algorithm that estimates the state of a dynamic system using a collection of noisy sensor measurements. The “optimal” term doesn’t necessarily mean that the filter will produce an absolute best estimate for the desired state. In most cases it does not. What it does mean is that based upon the assumptions made in the dynamics of the problem and the design of the filter itself, the Kalman filter will choose the most likely solution from a statistical perspective.

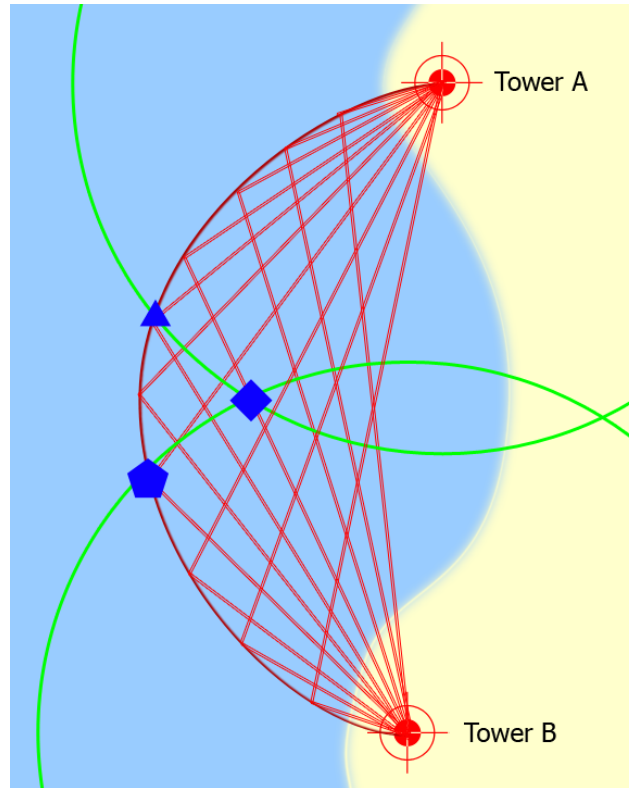
If you are familiar with Kalman filters then you will likely want to just skip this next section and jump straight to the section titled Tuning the Filter. The next section is a simple example to help those who are completely new to Kalman filters understand what we mean when we say “tuning” a filter.

Imagine the following scenario:

You are out for the weekend on your yacht about a mile off the coast and you and your buddy decide to take a nap. When you wake up you notice that it is already dark and you can’t find your GPS receiver. You decide that since you can’t measure your position directly you will try to estimate it using the information that you have available to you. You start to look around your environment for clues. First you notice that off in the distance you can see two red beacons on the coast. You are familiar with the coastline and for the sake of discussion happen to know that one of these red beacons is a Company A cell-phone tower located in town A, and the other is a Company B cell-phone tower in town B. Luckily you happen to have an Company A cell phone and your friend happens to use Company B. So you check your cell phone and it shows that you have one bar of reception which you know from previous experience means that you are about 2 miles away from the tower. Since your phone only has 5 bars this method of estimating distance isn’t very precise you figure that you can only trust your measurement of 2 miles to say ± 0.5 miles. Your friend’s phone shows the same one bar of signal strength so you estimate that you are 2 miles away from tower 2 with a ± 0.5 mile uncertainty. Then you realize that you can get a second measurement by measuring the angle between the two towers. Using a compass you measure the angle between the towers to be 100 degrees. Since you are using a cheap compass you bought at the local hardware store you gather that your angle measurement has an uncertainty of ± 5 degrees. You plot your findings on a sheet of paper and get a diagram similar to the one shown below.



Figure 1 – Example Problem



In this example you have three measurements or “sensors” that you are using to determine your position:

Sensor A: Distance to tower A

Sensor B: Distance to tower B

Sensor C: Angle between the two towers

From this example you can see that technically you only need two of the above three sensors to determine your position. If you were to use just sensor A and sensor B then your best estimate for your position would be the blue square. If you were to only use sensor A and sensor C then your best position estimate would be at the blue triangle. And last if you were to use just sensor B and C then your best position estimate would be the blue pentagon. It is entirely possible to use all three sensors in your estimate as long as you can assign some type of relative “trust” to the sensors. Depending upon how much you trust each sensor compared to the other two your position would be somewhere in between the three blue shapes. You could write a mathematical algorithm where the inputs are the uncertainty in your three measurements and the output is the “optimal” solution for your position. In this case the solution is only “optimal” in the sense that for whatever uncertainty parameters that you place into the algorithm the math will produce the most probable solution for the position of your boat.



As in the case given above there are many parameters that affect the accuracy of the mentioned measurements and it is impossible to account for all of these and find the “exact” uncertainty in your measurement. The best you can do is guess as to what the uncertainties of the measurements are so that you get a reasonable estimate output from your algorithm. The mathematics involved in determining the orientation of your VN-100 sensor isn’t near as straight forward as the position estimation problem just discussed but the principals are very similar. In the case of attitude determination we have three sensors available to use, the accelerometer, magnetometer, and angular rate gyro. The goal of tuning a Kalman filter is to find the correct uncertainty parameters to give to the filter so that it knows how much to “trust” each sensor. The Kalman filter used by the VN-100 has 10 parameters that you can tune to determine the trust level for each of the individual sensors.

Parameter 1: Gyro random walk variance

Parameter 2-4: X-axis, Y-axis, Z-axis gyro noise variance

Parameter 5-7: X-axis, Y-axis, Z-axis magnetometer noise variance

Parameter 8-10: X-axis, Y-axis, Z-axis accelerometer noise variance

In an ideal case where your sensors are perfectly calibrated and your calibration model accounts for all factors that affect the sensor output, then the only source of error in your sensor is due to noise. This means that the variance of your sensor measurement is equal to the variance of the noise inherent your sensors. In this ideal world tuning your Kalman filter would be quite easy, as all you would have to do is collect some steady state data from your sensors, calculate the standard deviation of the noise, square it, and place these numbers into the Kalman filter. Unfortunately life isn’t quite so simple. There are other errors in your measurements that you need to consider when tuning your Kalman filter. The errors in a filter’s measurements come typically come from one of the following sources:

1. Incorrect mathematical assumptions
2. Un-modeled sensor dynamics
3. Incorrectly measured or time varying calibration parameters
4. Noise

When you tune a Kalman filter you have to think about what other sources of error other than noise you expect to influence the uncertainty in your measurements.

Incorrect mathematical assumptions

When you calculate your orientation using accelerometers, magnetometers, and gyros you are essentially making several assumptions or “leaps of faith”. In order to determine attitude you need to have two non-parallel vectors. In the case of an AHRS you are taking advantage of two vector fields found in nature, the Earth’s magnetic and gravitation fields. These vector fields are for the most part fixed in inertial space and are non-parallel at most locations on the surface of the Earth. The problem comes from the fact that although we can measure the Earth’s gravitational acceleration with an



accelerometer and the Earth's magnetic field with the magnetometer, we are also measuring other sources of acceleration and magnetic fields. In the case of the accelerometer any relative motion of the sensor will induce inertial accelerations that will be measured by the accelerometer. Without an external means of estimating these external accelerations, they will show up as errors in the determination of the "acceleration" due to gravity. The magnetometer measures the total magnetic field, which is the magnetic field of the Earth plus any local magnetic fields created by surrounding ferromagnetic objects or magnetic fields induced by electric currents. The most predominant ferromagnetic materials are iron, nickel, cobalt, and their derivative alloys. As long as these materials are rigidly attached to the same structure as the sensor, then their effects can be modeled and accounted for. The VN-100 has both hard and soft iron calibration parameters to account for these effects. If the ferromagnetic materials are not rigidly attached to the sensors then their effects cannot be easily accounted, as they will be a function of the distance and direction the object has with respect to the sensor. Sources of error that fall into this category would be objects such as metal furniture in an office space, nearby vehicles, or even rebar in the concrete floor.

Un-modeled sensor dynamics

The sensor dynamics consists of the mechanisms by which the sensor converts its measured stimuli into a raw voltage that is measured by the ADC (Analog to Digital Converter), and then subsequently converted into a digital reading. The purpose of the calibration model is to capture as much of the dynamics that are occurring inside the sensor as possible. It is impossible, not to mention impractical, to attempt to account for every physical mechanism that will affect the sensor output. The art of engineering is to be able to determine what factors are most relevant in a problem and find a simple means of modeling this phenomenon. Low cost MEMS sensors are typically assumed to have a linear relationship with their measured stimuli. This is only an approximation, thus you will inherit some errors from this approximation (typically $< 1\%$ in the case of the sensors used on the VN-100). Other sensor dynamics such as hysteresis and random walk are incredibly difficult (if not impossible) to account for mathematically, and these will also add errors to your measurements. Also MEMS sensors are sensitive to temperature changes as well as their measured stimuli. This is usually given in datasheets as the temperature sensitivity and is typically on the order of 1-5%. These temperature effects can be calibrated for using an improved calibration model, but they require extensive testing in a thermally controlled environment over long periods of time, since they vary from chip to chip, and thus add to the total cost of the product. For users who need the full accuracy of the VN-100 over the full operating range we recommend using the VN-100T which undergoes several hours of extensive testing in a thermally controlled test environment to accurately capture the temperature sensitivity of each sensor.

Incorrectly measured or time varying calibration parameters

No calibration method is perfect and there will be some errors in the estimated value. Typically these errors are small but in the case of the magnetometer hard and soft iron calibration parameters they can have an enormous effect on the accuracy of the estimated measurement if incorrect parameters are used or if the parameters change over time.



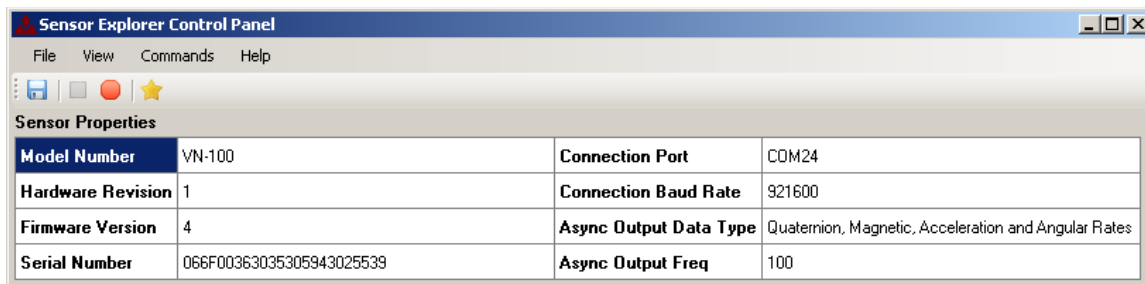
Tuning the filter

The point of explaining the above sources of error was to enlighten you on all of the things that you have to consider when tuning a Kalman filter. Although it is important to consider the noise levels of the sensors, this is only a small piece of the sensor “uncertainty” pie. If noise was the only source of error in your measurements then tuning a Kalman filter would be trivial, as using the sensor steady state measurement variance would suffice as an optimal choice of tuning parameters. When you tune the filter what you are in effect doing is “guessing” at what the effective variance will be for your measured parameter taking into consideration all sources of error for your sensors. Many of these errors such as magnetic hard/soft iron effects and inertial accelerations are a function of the environment the VN-100 is used in. Thus it is important that the designer of the product that uses the VN-100, or any other AHRS for that matter, takes the time to tune the filter for their specific application in order to optimize performance.

To give you a feel for how the parameters affect the performance of the device lets go thru some example cases. As you read along you can open Sensor Explorer and set the tuning parameters to the values given in the example cases.

Here are the settings that I am using in Sensor Explorer. If you are plugged into the USB then you should be able to run the baud rate at the maximum setting of 921600 baud and the data output rate at 100Hz. If you are using a serial cable or have a slow PC then you can use a baud rate of 115200 and a data rate of 50 Hz. Make sure that after you get everything set like you want before you start tuning you use the Commands->Write Settings to save your settings to memory. This will come in handy later during the tuning process.

Figure 2 – Sensor Explorer Recommended Settings

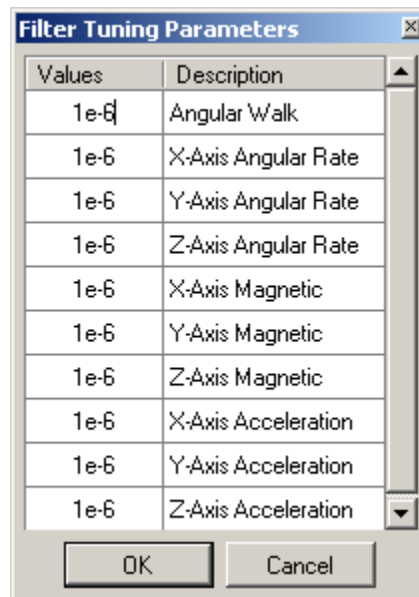


Sensor Explorer Control Panel			
File View Commands Help			
Sensor Properties			
Model Number	VN-100	Connection Port	COM24
Hardware Revision	1	Connection Baud Rate	921600
Firmware Version	4	Async Output Data Type	Quaternion, Magnetic, Acceleration and Angular Rates
Serial Number	066F00363035305943025539	Async Output Freq	100

Let's start with a baseline case that is very similar to the performance of the VN-100 at factory default. Set the filter tuning parameters to the values given in Figure 3 by going to Commands -> Set Filter Tuning.



Figure 3 – Baseline Parameters

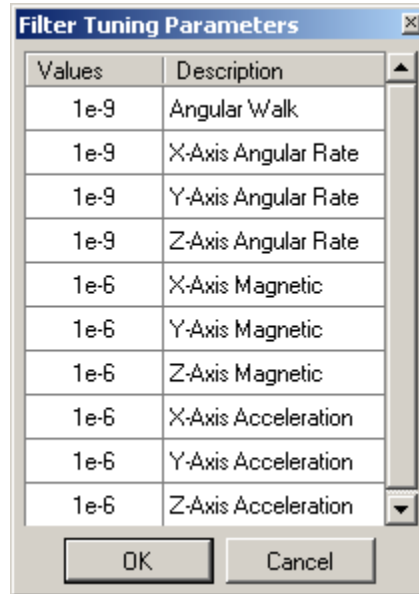


Values	Description
1e-6	Angular Walk
1e-6	X-Axis Angular Rate
1e-6	Y-Axis Angular Rate
1e-6	Z-Axis Angular Rate
1e-6	X-Axis Magnetic
1e-6	Y-Axis Magnetic
1e-6	Z-Axis Magnetic
1e-6	X-Axis Acceleration
1e-6	Y-Axis Acceleration
1e-6	Z-Axis Acceleration

These values will effectively give equal weights to all of the inputs to the Kalman filter. This is a sub-optimal case but it should be close to what the default parameters are like. Try placing the dev board on the table. Send it a tare command, then after waiting a few seconds try to shaking the the board side to side similar to the way you would shake a salt shaker while keeping its attitude constant in order to agitate the accelerometer. Notice how the disturbance in the accelerometer has a visual impact on the attitude solution. Also try to find something that is slightly magnetic such as an AA battery or your car keys. Place the VN-100 down on the table so that it is stationary and try moving the magnetic object around in close proximity to the sensor. You should be able to see a noticeable impact on the attitude solution.

Now try the following parameters listed in Figure 4.

Figure 4 – Increased trust in gyros



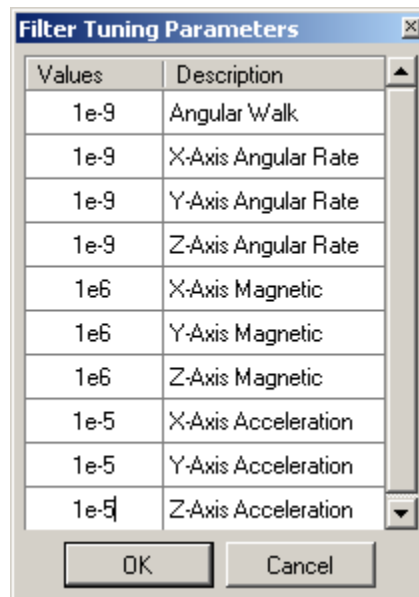
Values	Description
1e-9	Angular Walk
1e-9	X-Axis Angular Rate
1e-9	Y-Axis Angular Rate
1e-9	Z-Axis Angular Rate
1e-6	X-Axis Magnetic
1e-6	Y-Axis Magnetic
1e-6	Z-Axis Magnetic
1e-6	X-Axis Acceleration
1e-6	Y-Axis Acceleration
1e-6	Z-Axis Acceleration

Now that we have increased the confidence in the gyros try the same experiment on the dev board that you did before. Try shaking it and also try to disturb the magnetic field. You will notice that the perturbations in the attitude solution are much weaker, as the filter has a lot more confidence in the gyros. These choices of parameters are fairly good for applications where you can place trust in both the magnetometer and the accelerometer. For most applications you should see a substantial improvement using these parameters as opposed to the factory default. The only real downside to using these parameters is the time that it takes to recover from a loss of the attitude solution. Since the filter places a lot of trust in the gyros, it will take some time (a few seconds) to recover in the event that it loses its attitude solution. Try grabbing the dev board with your fingers, and rotate it around the Z axis at a rate faster than 300 deg/s. Do this in a back and forth motion with your wrist for a few seconds, then immediately place the sensor down on the table. You should notice that the filter has lost its attitude solution since the gyros angular rate went beyond its full scale limit of 300 deg/s. The time it takes to recover is longer in this case than what you will likely see for the case the factory default parameters.

There are a lot of our customers that are interested in tuning the VN-100 so that it places less trust in the magnetometer. This may be necessary in environments where you have an electric motor nearby. With this in mind let's try an example case where we want to tune out the influence of the magnetometer. Try setting the filter parameters to the following values.



Figure 5 – Tune out the magnetometer



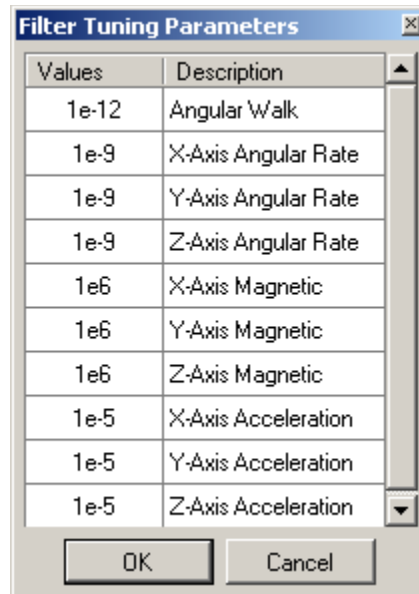
Values	Description
1e-9	Angular Walk
1e-9	X-Axis Angular Rate
1e-9	Y-Axis Angular Rate
1e-9	Z-Axis Angular Rate
1e6	X-Axis Magnetic
1e6	Y-Axis Magnetic
1e6	Z-Axis Magnetic
1e-5	X-Axis Acceleration
1e-5	Y-Axis Acceleration
1e-5	Z-Axis Acceleration

Notice how we set the magnetometer to a MUCH higher value of $1e6$. This should completely remove the magnetometer from the solution. Also the confidence was lightly decreased in the accelerometer from $1e-6$ to $1e-5$. Now try slowly rotating the development board around in each of the three axes then lay it down on the table and let it rest for a few seconds. You should see that the solution is locked with the Z-axis pointing down and the yaw angle should be slightly drifting freely. This drift is due to the fact that now that the filter can no longer trust the magnetometer it has no way of capturing an absolute fix on the heading angle. Leave the VN-100 at rest on the table for a few more seconds, and take a look at the angular rate plot. You should notice that the X and Y axes angular rates are being held at zero by the compensation of the accelerometer while the Z-axis (blue plot) is at some non-zero value. Again this is due to the fact that the filter has no way of being able to drive this value to zero. If the filter had a vector with a component in the XY plane then it would be able to bring the rate to zero. Try tilting the board lightly to an angle of around 45 degrees, hold for a few seconds then lay the board back down. After waiting about 5 seconds the angular rate plot should re-adjust its scale and you should see that this movement gave the filter a chance to measure a component of the acceleration vector in the inertial XY plane for a few seconds. This gave the filter enough time to realize that there was a bias on the Z-axis gyro rate and compensate for it. Now try leaving the board at rest for about 15-20 seconds on the table then shake it back and forth on the table for a second, then immediately let it go. If you do it right you should see a violent change in the yaw angle while the pitch and roll hold relatively constant. This is an unfortunate effect of recursive Kalman filters that can at times cause problems. If you saw the violent shake in the angular rate around the Z-axis what likely happened in this case is that because the accelerometer was completely free of noise for around 20 seconds the filter started to trust the sensor more on its own since it saw that it was measuring a very stable vector. In Kalman filtering this is called a collapse of the covariance matrix. When a sensor goes a long time without any noise on it the filter will start to place more and more trust in that sensor, and then when that value changes rapidly it will likely get confused and believe the disturbance much more than it should. There are two ways of



dealing with this, one is to place less trust in the accelerometer, and the other is to tell the filter that the gyro bias shouldn't change as fast. Let's try increasing the confidence in the gyro bias estimate. Try setting the filter setting to the values given in Figure 6.

Figure 6 - Increased trust in gyro bias estimate



Values	Description
1e-12	Angular Walk
1e-9	X-Axis Angular Rate
1e-9	Y-Axis Angular Rate
1e-9	Z-Axis Angular Rate
1e6	X-Axis Magnetic
1e6	Y-Axis Magnetic
1e6	Z-Axis Magnetic
1e-5	X-Axis Acceleration
1e-5	Y-Axis Acceleration
1e-5	Z-Axis Acceleration

Now try to reproduce the same experiment that you did before. Place the development board down on the table, wait for about 30 seconds, then shake it back and forth and let go. You should see a less violent change in the heading angle as opposed to the last case. Also make sure that you try moving a magnet around the sensor and look to see if you notice a change in the attitude. If you look over at the magnetic graph you should be able to see a substantial change in the measured magnetic field without seeing a visible change in the attitude solution.

I hope that these cases have been instructional for you. I recommend that you get yourself a good cup of coffee and just spend a few hours playing around with different parameters and test cases. There is a wealth of knowledge to be learned by just playing around these parameters and looking for their visible influence on the plots and 3D diagram.



Concluding Remarks

Tuning Kalman filters has in the past been considered a “black art” for many engineering applications. Much of this stigma comes from the fact that is very difficult to determine beforehand what the optimal tuning parameters will be for a given problem. It just takes time and an environment where you can play around with the parameters and get a feel for what happens when you turn each of the dials. Even though each of these values do have a mathematical significance, keep in mind that all that really matters is their value relative to each other, so your best bet is to just turn the knobs and watch the output. There may be times where in “tuning” these parameters you end up with a value that causes the filter to go unstable. Although this is something that you definitely want to avoid during end-use, in the testing environment if this happens then there is nothing to fear, just reset the sensor and continue tuning where you left off. You might want to use `Commands->Write Settings` when you get the baud rate and other parameters set just the way you want them so that in the event the filter does go unstable you can just reset it and start where you left off. When tuning the parameters, you will likely want to just adjust one parameter at a time while holding the others constant. It is ok to adjust all three axes noise variances for a given sensor simultaneously; you just don’t want to tune two different sensors such as the accelerometer and angular rate gyro at the same time. Also don’t be afraid to adjust the numbers by large amounts. As mentioned before start tuning by adjusting the numbers by one order of magnitude at a time and then when you start getting close to the range you want then you can adjust by say $\frac{1}{2}$ and order of magnitude. In the early stages of tuning don’t be too obsessed with tuning pasts 2 significant digits, as mentioned earlier, you will likely not be able to see any discernible difference visually. Also I am aware of the fact that the default parameters in the VN-100 have quite a few significant digits listed. This is nothing to really worry about as they are meaningless past the 3rd or 4th digit. It is likely that in future versions of the VN-100 the factory default parameters will be rounded off to the 2nd significant digit for simplicity.

