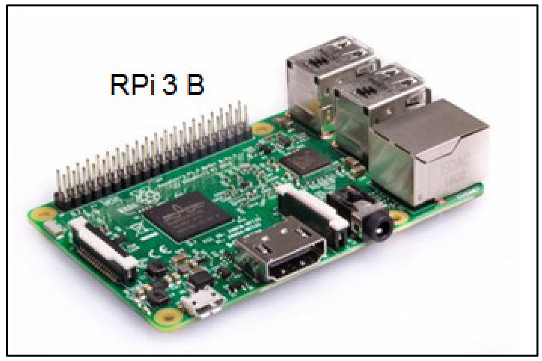
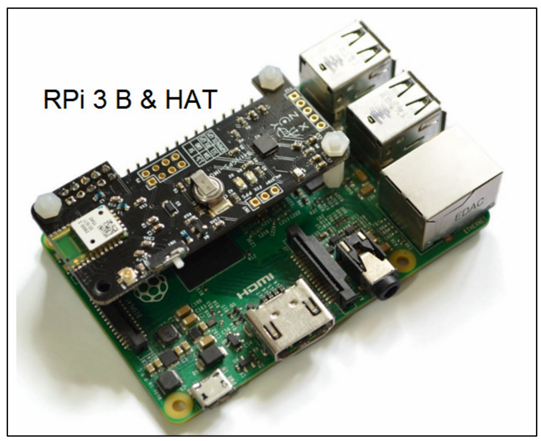
Characterizing an IMU for a Raspberry Pi
Noel Zinn, www.hydrometronics.com, May 2018

The Raspberry Pi (RPi) stands out among single board computers (SBCs) due to its low cost, improving specifications with successive models, large installed base (10+ million), extensive documentation and a thriving market in multi-purpose daughter boards called "HATs", or "hardware attached on top". The RPi used in this study is an RPi 3 B with 1.2GHz 64-bit quad-core ARM Cortex-A53 CPU and 1 GB RAM. This is a thoroughly-capable, $35 RPi running the Raspbian version of Debian Jessie Linux, though not the latest with marginally improved specifications (which is the RPi 3 B+). A Raspberry Pi 3 B is pictured to the right. The surface area of the SBC is about that of a credit card.



The HAT in this study is a BerryGPS+IMUv2 by the Australian, Star-Trek-aficionado company, OzzMaker (www.ozzmaker.com), and it costs $51. OzzMaker produce BerryGPS, BerryIMU and BerryGPS+IMU among other products. A description of the GPS functionality of BerryGPS+IMU is deferred to a later post that will describe a Kalman filter that integrates the GPS and IMU features of BerryGPS+IMU. Today we are only interested in the IMU functionality. That functionality is provided by an onboard STMicroelectronics iNEMO LSM9DS1 MEMS inertial module that has a 3-axis accelerometer, a 3-axis gyro and a 3-axis magnetometer. Additionally, a Bosch BMP280 digital pressure sensor provides altitude thus completing a 10 degree of freedom (DoF) attitude reference system. And ... there's a temperature sensor, too. Today we are specifically interested in the 3-axis gyro and the 3-axis accelerometer.

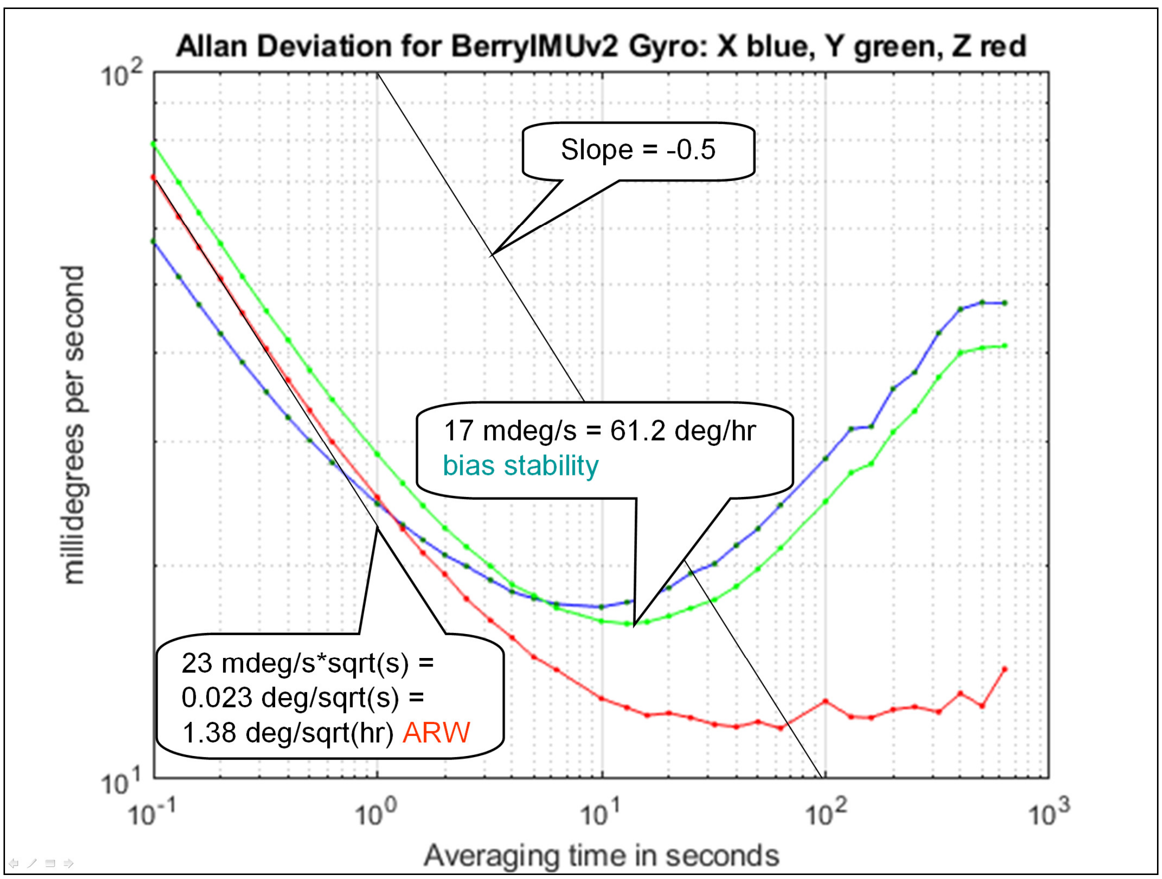


STMicroelectronics provide a datasheet with very basic specifications of the IMU in terms of sensitivity ranges, but not in terms of random walk and in-run bias stability that would be common for higher-end IMUs. This is promised in an overly-concise Design Tip DT0064 on noise analysis that includes Matlab code for the computation of Allan variance from which those specifications can be derived, but they are not delivered. So, in order to place the specifications of the BerryIMU (LSM9DS1) in the context of other MEMS IMUs in my experience, I've computed Allan deviations from an 12-hour data sample collected overnight at my desk. This study reports on that procedure, the results and comparisons with other MEMS IMUs.

The LSM9DS1 IMU produces integer values for GYRx (gyro X), GYRy, GYRz, ACCx (accelerometer X), ACCy, and ACCz. These need to be scaled appropriately as a function of the selected sensitivities: ±245 deg/sec, ±500 deg/sec, and ±2000 deg/sec for the gyro and ±2 G, ±4 G, ±8 G, and ±24 G for the accelerometers. The default configuration in the Python code supplied by OzzMaker is the widest-range, least-sensitive settings. More sensitive, lesser-range settings gave erratic data. Sample data follows:

Time (seconds)	GYRx	GYRy	GYRz	ACCx	ACCy	ACCz
20747.12154	19	74	76	36	-84	1602
20747.13149	17	76	72	45	-77	1596
20747.14139	16	74	72	40	-73	1596
20747.15124	17	69	72	36	-84	1592
20747.16113	16	72	76	37	-81	1593
20747.17099	17	72	71	38	-82	1599
20747.18085	18	75	77	42	-79	1588
20747.19070	13	71	71	38	-89	1591

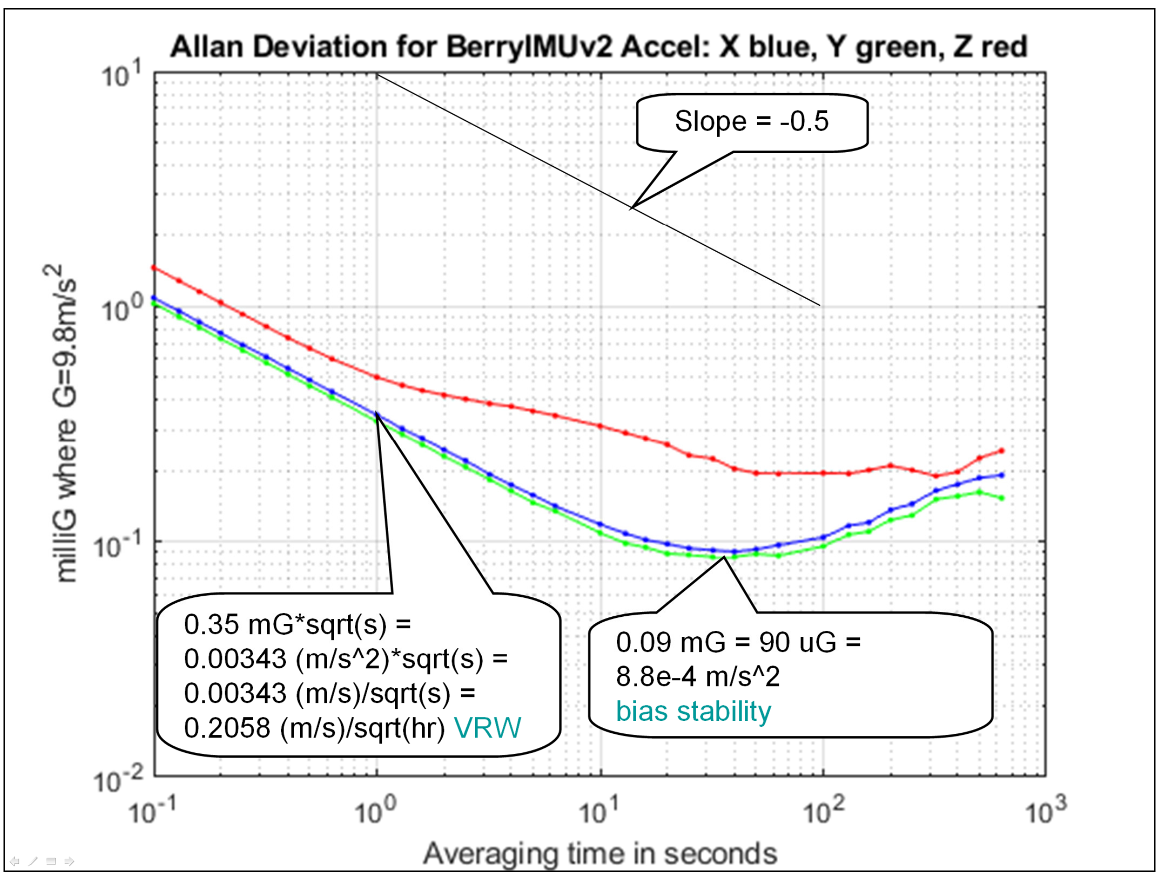
Notice the large readings for gravity in the ACCz channel. The scaling factor for the accelerometer is 0.732433 to nearly produce the expected 1000 mG in the vertical. Some gravity is distributed in the X and Y channels due to imperfect leveling of the IMU. The gyro scaling factor is 70, which produces noisy rotation biases in millidegrees per second in all three gyro channels.



From 12 hours of overnight data at 100Hz, which is more than 4 million data points, the Allan deviation (AD) plots for the gyro (above) and accelerometer (below) were produced from Matlab code written several years ago for this purpose. Each AD plot (gyro and accelerometer) has all three axes: blue for X, green for Y and red for Z.

There are many ways to specify an IMU and the ways vary by manufacturer. In my experience the two most useful (and universal) specifications are in-run bias stability and angular random walk for gyros and in-run bias stability and velocity random walk for accelerometers. Both of these specifications can be "read" from an AD plot computed from data empirically produced by your own IMU.

Turning first to the gyro AD plot for the LSM9DS1 above, bias stability is the value at the minimum of the plot for the three axes. For the representative green Y axis this is approximately 17 millidegrees per second, more commonly represented as approximately 61 degrees per hour. Interpreting angular random walk is more complicated. First we need a line with a slope of -0.5 on the plot. This is shown. Then the line is moved until it coincides with straight sections of the three axes. I've shown it for the red Z axis. Angular random walk is read where the extended slope line intersects with an averaging time of 1 second (10^0). The reading at that point is multiplied by the square root of a second for reasons beyond the scope of this paper. So the angular random walk of the red Z (vertical) axis is 23 (millideg/sec)*sqrt(sec) = 0.023 deg/sqrt(sec) = 1.38 deg/sqrt(hr). The final representation is most common.



Turning next to the accelerometer AD plot for the LSM9DS1 above, bias stability is the value at the minimum of the plot for the three axes. For the representative green Y axis this is approximately 0.09 milliGs, or 90 microGs, or 8.8e-4 meters per second squared. All three representations are common depending on the quality of the IMU, but we use milliG. Velocity random walk is interpreted similarly to angular random walk. Again we need a line with a slope of -0.5 on the plot. This is shown. Then the line is moved until it coincides with a straight section of the three axes. The straight sections of X (blue) and Y (green) are at the right slope right through the 1-second averaging time. Again we multiply the reading by the square root of a second. So, the velocity random walk of the green Y axis is 0.35 milliG*sqrt(s) = 0.00343 (m/s^2)*sqrt(s) = 0.00343 (m/s)/sqrt(s) = 0.2058 (m/s)/sqrt(hr), the latter being the most common representation.

Using these four values (ARW, VRW and the two in-run bias stabilities), which admittedly are merely representative and not a complete characterization of the IMU, we can compare the LSM9DS1 with the published specifications and prices of some other MEMS IMUs below.

			Gyro		Accelerometer	
			ARW	bias stability	VRW	bias stability
Manufacturer	Model	± US$	deg/sqrt(h)	deg/h	(m/sec)/sqrt(h)	mG
Analog Devices	S16488A	$1,800	0.26	5.1	0.029	0.07
Analog Devices	S16485	$1,500	0.3	6.25	0.023	0.032
Sensenor	STIM300	$8,600	0.15	0.5	0.06	0.05
Epson	G350	$1,700	0.2	6	0.04	0.1
Sparton	IMU-10	$1,800	0.91	10	0.047	0.0305
Systron-Donner *	MMQ50	$4,500	0.3	100	0.5	3
VectorNav **	VN-300	$5,000	0.22	< 10	0.06	<0.04
STMicro *	LSM9DS1	$51	1.38	61.2	0.21	0.09

* includes GPS
** includes dual GPS

So, the LSM9DS1 is among the weakest of the IMUs cited, but, from a price perspective, it is an extraordinary value. Hardware quality is scalable. Software that works with low-end MEMS will also work with a high-end MEMS, fiber-optic or ring-laser instrument mutatis mutandis. Hobby boards nurture professional competence.

A subsequent report will explore a Kalman filter that integrates the IMU with the GPS of the BerryGPS+IMUv2 board.