Autonomous Vehicle Simulation (AVS) Laboratory, University of Colorado

Basilisk Technical Memorandum

Document ID: Basilisk-imu_sensor

TESTING IMU SENSOR MODEL

Prepared by	S. Carnahan

Status: Revision 1

Scope/Contents

This unit test validates the internal aspects of the Basilisk IMU module `test_imu_sensor.py` by comparing module output to expected output. The Basilisk IMU module is responsible for producing sensed body rates and acceleration from simulation truth values. The IMU module applies Gauss-Markov process noise to the true body rates and acceleration. The unit test validates MRP switching, static bias, process noise, discretization, saturation, spacecraft center of mass (CoM) offset, sensor misalignment, and bias walk bounds for both the gyroscope and accelerometer.

Rev:	**Change Description**	**By**
Draft	Initial document creation. Test Descriptions.	J. Alcorn
Rev. 1	Added module documentation. Updated with AutoTex implementation	S. Carnahan

Contents

---

1 Introduction

The Basilisk IMU module imu_sensor.cpp is responsible for producing sensed body rates and acceleration from simulation truth values. Each check within test_imu_sensor.py sets initial attitude MRP, body rates, and accumulated Delta V and validates output for a range of time.

2 Mathematical Model

2.1 Data Discretization

Because sensors record data digitally, that data can only be recorded in discrete chunks, rather than the (relatively) continuous values that the computer calculates at each time steps. In order to simulation real IMU behavior in this way, a least significant bit (LSB) value is accepted for both the gyro and the accelerometer. This LSB is applied as follows to any 3-dimensional measurement $\mathbf{m}$:

$$\mathbf{m}_{\text{discretized}} = (\text{LSB})\left\lfloor\left|\frac{\mathbf{m}}{(\text{LSB})}\right|\right\rfloor \tag{1}$$

Where $\lfloor\ \rfloor$ indicate the **floor()** function and LSB can be either the accelerometer or gyro least significant bit.

2.2 Gyro Model

The gyro is modeled by...

2.3 Accelerometer Model

The accelerometer is modeled according to...

2.4 Recursion Formulas

The above formulas for the gyro and accelerometer are broken down into iterative steps in order to be modeled in the simulation. The recursive formulas follow

2.5 Variable Definitions

The variables in Table 2 are available for user input. Variables used by the module but not available to the user are not mentioned here. Variables with default settings do not necessarily need to be changed by the user, but may be.

Table 2: Definition and Explanation of Variables Used.

Variable	LaTeX Equivalent	Variable Type	Notes
InputStateMsg	N/A	string	Default setting: "inertial_state_output". This is the message from which the IMU receives spacecraft inertial data.
OutputDataMsg	N/A	string	Default setting: "imu_meas_data". This message contains the information output by the IMU.
InputMassMsg	N/A	string	Default setting: "spacecraft_mass_props". This is the message from which the IMU received spacecraft mass data.
SensorPos_B	N/A yet	double [3]	[m] Required input - no default. This is the sensor position in the body frame relative to the body frame.
roll, pitch, yaw	N/A yet	double, double, double	Default setting: (0,0,0). To set non-zero initial angles between imu and spacecraft body, call setBodyToPlatformDCM(roll, pitch, yaw)
dcm_PB	N/A yet	double [3][3]	Default setting: Identity. Setting dcm_PB is equivalent to calling setBodyToPlatformDCM(roll, pitch yaw) above. Use one method or the other.
senRotBias	N/A yet	double [3]	[r/s] Default setting: zeros. This is the rotational sensor bias value for each axis.
senTransBias	N/A yet	double [3]	[m/s2] Default setting: zeros. This is the translational sensor bias value for each axis
senRotMax	N/A yet	double	[r/s] Required input - no default. This is the gyro saturation value.
senTransMax	N/A yet	double	[m/s2] Required input - no default. This is the accelerometer saturation value.
PMatrixAccel	N/A yet	double [3][3]	Default: zeros. This is the covariance matrix used to perturb the state.
PMatrixGyro	N/A yet	double [3][3]	Default: zeros. This is the covariance matrix used to perturb the state.
walkBoundsGyro	N/A yet	double [3]	Default: zeros. This is the "3-sigma" errors to permit for gyro states
walkBoundsAccel	N/A yet	double [3]	Default: zeros. This is the "3-sigma errors ot permit for acceleration states.
accelLSB	N/A yet	double	Default: 0.0. This is the discretization value (least significant bit) for acceleration. Zero indicates no discretization.
gyroLSB	N/A yet	double	Default: 0.0. This is the discretization value (least significant bit) for acceleration. Zero indicates no discretization.

3 Library

4 Unit Test

This test is located in `SimCode/sensors/imu_sensor/_UnitTest/test_imu_sensor.py`. In order to get good coverage of all the aspects of the module, the test is broken up into several parts:

1. Gyro/Accelerometer I/O The check verifies basic I/O of body rates and acceleration. Initial attitude MRP, body rates, and Delta V are propagated and corresponding body rates, acceleration, DR, and DV are compared to module output.

2. MRP Switch The check validates that the module accounts for attitude MRP switching in calculation of body rates. Initial attitude MRP and body rates are propagated for a sufficient amount of time for the MRP to switch to the shadow set. The test verifies that the module sets the MRP switch flag to TRUE.

3. Static Bias The check validates static bias in gyro/accel measurements. Gyro and accelerometer static bias are set to nonzero values. Initial MRP, body rates, and DV are propagated. Module output is verified to contain data with static bias.

4. Process Noise The check verifies that the Gauss-Markov model applies noise of appropriate mean and standard deviation to the attitude coordinate output. This check does not consider bias random walk. Accelerometer and gyro noise standard deviations are set to nonzero values for each axis. Module output is verified by taking the standard deviation of output data and comparing to specified values.

5. Discretization The check verifies that the module correctly discretizes the gyro/accel data according to the specified least significant bit (LSB). LSB of gyro and accelerometer are set to nonzero values. Output is verifed to round input to nearest multiple of LSB.

6. Saturation The check verifies that the module saturates the output according to specified values. Gyro and accelerometer maximum output are set to nonzero values. Output is verified to not exceed specified saturation values for both negative and positive cases.

7. Accelerometer Center of Mass Offset The check validates that the accelerometer will give appropriate output based on an offset in center of mass from accelerometer. Sensed acceleration is given by the equation

$$\ddot{\boldsymbol{r}}_{\text{sensed}} = \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r}_{\text{SC}}) + \dot{\boldsymbol{\omega}} \times \boldsymbol{r}_{\text{SC}} + \ddot{\boldsymbol{r}}_{B/N} \tag{2}$$

   where $\boldsymbol{\omega}$ is the spacecraft angular velocity vector, $\boldsymbol{r}_{\text{SC}}$ is the position vector of the IMU with respect to the spacecraft center of mass, and $\ddot{\boldsymbol{r}}_{B/N}$ is the actual inertial acceleration of the spacecraft.

8. IMU Misalignment The check validates measurements taken when the IMU is not correctly aligned (i.e. the IMU measurements are taken in a frame with constant rotational offset from assumed IMU orientation).

9. Bias Random Walk Bounds The check verifies that the Gauss-Markov model correctly applies bias random walk to the gyro and accelerometer output. Specified walk bounds are validated.

4.1 Test Parameters

This section summarizes the test input/output for each of the checks.

- Error Tolerance

  There are specific error tolerances for each test. Error tolerances are determined based on whether the test results comparison should be exact or approximate due to integration or other reasons. Error tolerances for each test are summarized in table 3.

Table 3: Error tolerance for each test.

Test	Tolerated Error
Gyro/Accelerometer I/O	1.0e-06
MRP Switching	-
Static Bias	1.0e-03
Process Noise	1.0e-03
Discretization	1.0e-05
Saturation	1.0e-03
Accelerometer Center of Mass Offset	1.0e-05
IMU Misalignment	1.0e-04
Bias Walk Bounds	-

4.2 Test Results

All checks within test_imu_sensor.py passed as expected. Table 4 shows the test results. Figures 4.2 and 4.2 show the module output for the process noise and walk bounds checks, respectively.

Table 4: Test results

Test	Pass/Fail	Notes
Gyro/Accelerometer I/O	Passed	
MRP Switching	Passed	
Static Bias	Passed	
Process Noise	Passed	
Discretization	Passed	
Saturation	Passed	
Accelerometer Center of Mass Offset	Passed	
IMU Misalignment	Passed	
Bias Walk Bounds	Passed	

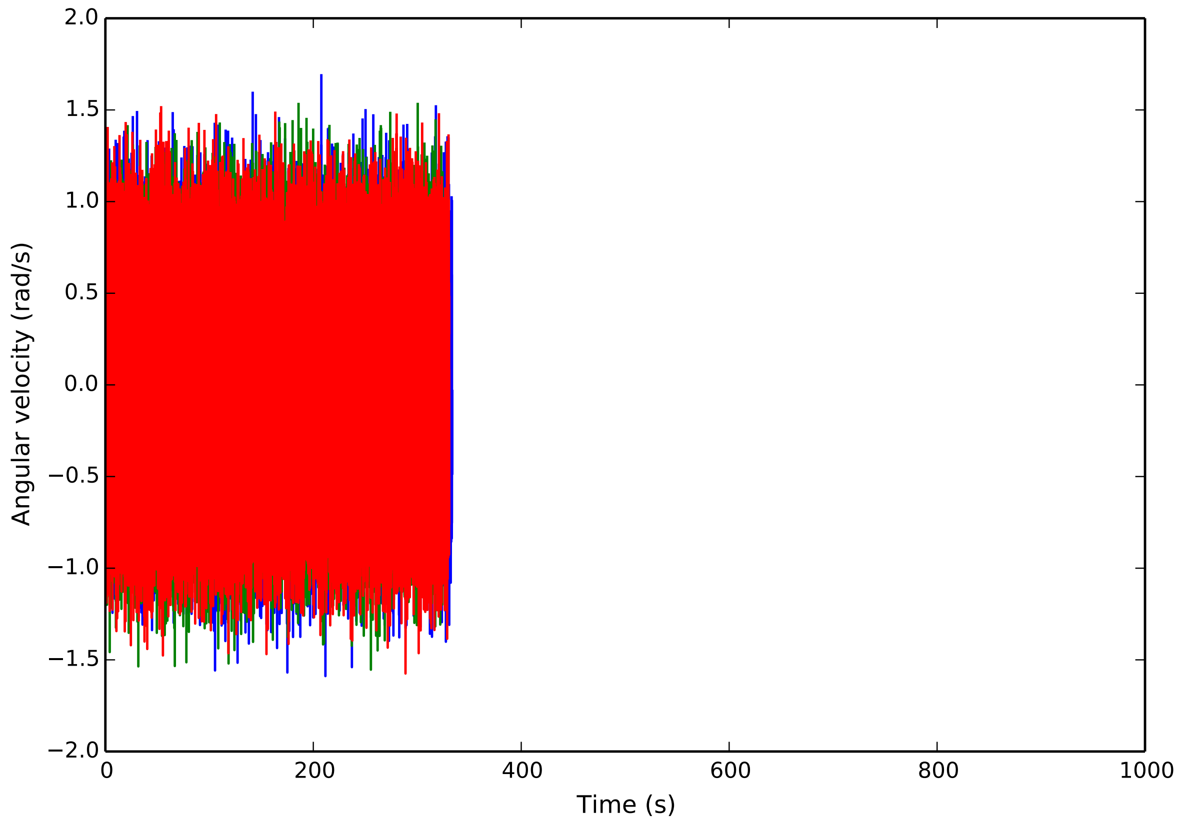

Fig. 1: Module output of noise standard deviation check

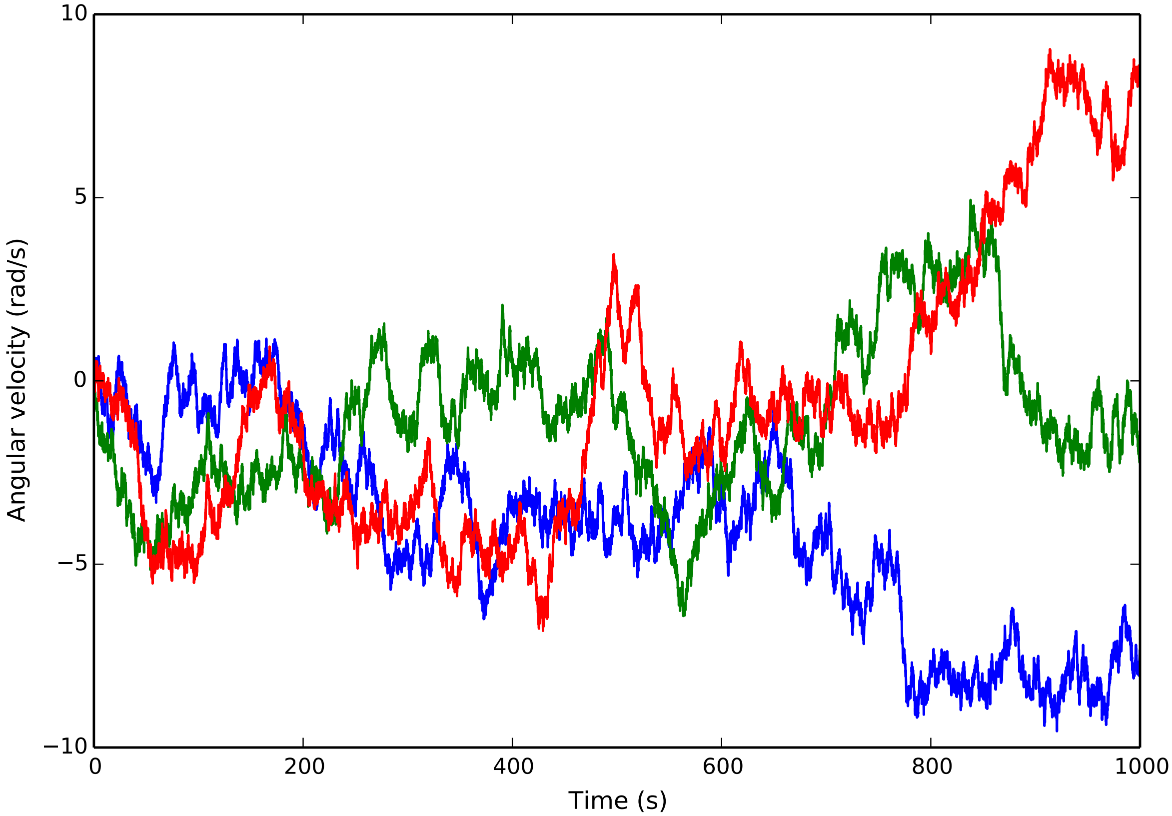

Fig. 2: Module output for random walk bounds check

4.3 Test Coverage

The test covers 100% of the code.

5 Conclusion

What a great piece of code.