
# Autonomous Vehicle Simulation (AVS) Laboratory, University of Colorado

## Basilisk Technical Memorandum

**Document ID: Basilisk-imu**

**INERTIAL MEASUREMENT UNIT C++ MODEL**


| | |
|---|---|
| Prepared by | S. Carnahan |


| |
|---|
| **Status:** Tested |
| **Scope/Contents** |
| The Basilisk Inertial Measurement Unit (IMU) module is responsible for producing sensed body rates and acceleration from calculated values. It also provides a change in velocity and change in attitude value for the time between IMU calls. The IMU module applies Gauss-Markov process noise to the true body rates and acceleration. A unit test has been written which validates outputs of the IMU. |

| Rev | Change Description | By | Date |
|---|---|---|---|
| 1.0 | First draft | J. Alcorn | 20170713 |
| 1.1 | Added Math Documentation and User Guide. Implemented AutoTeX | S. Carnahan | 20170713 |
| 1.2 | Rewrote for new IMU implementation and test | S. Carnahan | 20170914 |

# Contents



---

# 1 Model Description

This module is a simulation environment module which simulates the analog voltage interface of a RW cluster. The input is an array of voltages $V_i$. The Reaction Wheel (RW) motor torque $u_{s_i}$ is evaluated using a linear mapping

$$u_{s_i} = V_i \gamma SF + b \tag{1}$$

where $\gamma$ is constant value gain. SF is the scale factor error (i.e. a constant 1% gain error SF $= 1.01$). $b$ is the bias term (i.e. a constant 1 Nm error on torque b $= 1$.) The output of the module is an array of RW motor torques. The deadband and saturation behavior of the RW speed is modeled inside the RW dynamics model.

# 2 Model Functions

The functions of the voltage to torque model are:

- **Voltage to Torque**: Converts given voltage to a torque command.

# 3 Model Assumptions and Limitations

This code makes assumptions which are common to IMU modeling.

- **No noise**: There is no random noise input when making the voltage to torque conversion.
- **All wheels the same**: All reaction wheels utilize the same gain, scale factor, and bias. No wheel can be singled out to have higher scale factor or bias, etc.

# 4 Test Description and Success Criteria

A series of unit tests are performed to check the validity of this module's operation. The expected outcome is calculated independently in python and the results are seen to be correct to machine error.

# 5 Test Parameters

Three base voltages $V_0$ are tested where $V_0 \in (5.0, -7.5, 0.0)$. The input voltages are then setup as

$$\boldsymbol{V} = V_0 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 0.0 \\ 1.0 \\ 1.5 \end{bmatrix} \tag{2}$$

Other inputs to the module are:

```
testModule.voltage2TorqueGain = 1.32        # [Nm/V] conversion gain
testModule.scaleFactor = 1.01 #[unitless] scale factor
testModule.bias = 0.05 # [Nm] bias
```

# 6 Test Results

The unit test results are in the following Tables.

**Table 2:** RW motoor torque output for Base Voltaget = 5.0V.

| time [s] | $u_{s,1}$ (Nm) | Error | $u_{s,2}$ (Nm) | Error | $u_{u,3}$ (Nm) | Error |
|---|---|---|---|---|---|---|
| 0 | 6.676 | 0 | 8.0092 | 0 | 8.6758 | 0 |
| 0.5 | 6.676 | 0 | 8.0092 | 0 | 8.6758 | 0 |
| 1 | 6.676 | 0 | 8.0092 | 0 | 8.6758 | 0 |

**Table 3:** RW motoor torque output for Base Voltaget = -7.5V.

| time [s] | $u_{s,1}$ (Nm) | Error | $u_{s,2}$ (Nm) | Error | $u_{u,3}$ (Nm) | Error |
|---|---|---|---|---|---|---|
| 0 | -9.989 | 0 | -8.6558 | 0 | -7.9892 | 0 |
| 0.5 | -9.989 | 0 | -8.6558 | 0 | -7.9892 | 0 |
| 1 | -9.989 | 0 | -8.6558 | 0 | -7.9892 | 0 |

**Table 4:** RW motoor torque output for Base Voltaget = 0.0V.

| time [s] | $u_{s,1}$ (Nm) | Error | $u_{s,2}$ (Nm) | Error | $u_{u,3}$ (Nm) | Error |
|---|---|---|---|---|---|---|
| 0 | 0.01 | 0 | 1.3432 | 0 | 2.0098 | 0 |
| 0.5 | 0.01 | 0 | 1.3432 | 0 | 2.0098 | 0 |
| 1 | 0.01 | 0 | 1.3432 | 0 | 2.0098 | 0 |

All tests pass:

**Table 5:** Test results

| Voltage Test | Pass/Fail |
|---|---|
| 5.0 | PASSED |
| -7.5 | PASSED |
| 0.0 | PASSED |

# 7 User Guide

This section contains conceptual overviews of the code and clear examples for the prospective user.

## 7.1 Module Setup

The interface module is created in python using:

```
testModule = rwVoltageInterface.RWVoltageInterface()
testModule.ModelTag = "rwVoltageInterface"
```

The only parameter that must be set is the voltage to torque conversion gain $\gamma$. This is done using

```
testModule.voltage2TorqueGain = 1.32        # [Nm/V] conversion gain
testModule.scaleFactor = 1.01 #[unitless] scale factor
testModule.bias = 0.05 # [Nm] bias
```