Autonomous Vehicle Simulation (AVS) Laboratory, University of Colorado

Basilisk Technical Memorandum

Document ID: Basilisk-Magnetometer

MAGNETIC FIELD MEASUREMENTS

Prepared by	D. Cilden-Guler

Status: Tested
Scope/Contents
The Basilisk Three-Axis Magnetometer (TAM) module is responsible for producing magnetic field measurements in sensor frame from the magnetic field model. The TAM module applies Gauss-Markov process noise to the true magnetic field values. A unit test has been written which validates outputs of the TAM.

Rev	Change Description	By	Date
1.0	First Draft	D. Cilden-Guler	09-26-2019

Contents

---

1 Model Description

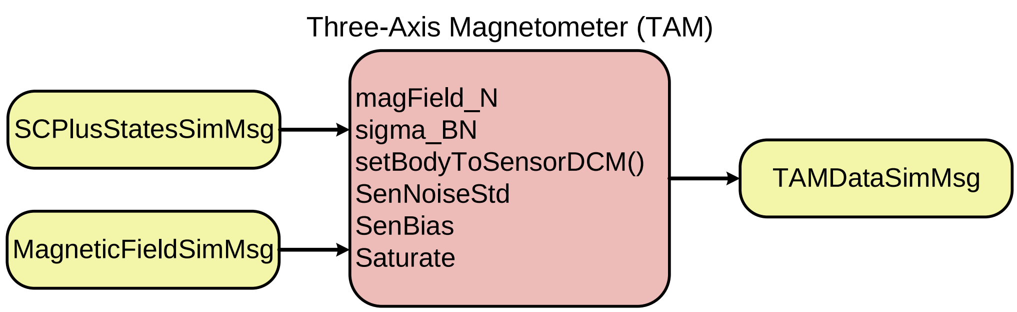

Fig. 1: Illustration of the Magnetometer() module I/O

1.1 General Module Behavior

This document describes how Three-Axis Magnetometer (TAM) devices are modeled in the Basilisk software. The purpose of this module is to implement magnetic field measurements on the sensor frame $\mathcal{S} : \{\hat{s}_1, \hat{s}_2, \hat{s}_3\}$.

There are a multitude of magnetic field models. As all the magnetic field models are children of `MagneticFieldBase` base class, magnetometer can be created based on any of the magnetic field model.

1.2 Planet Centric Spacecraft Position Vector

For the following developments, the spacecraft location relative to the planet frame is required. Let $\boldsymbol{r}_{B/P}$ be the spacecraft position vector relative to the planet center. In the simulation the spacecraft location is given relative to an inertial frame origin O. The planet centric position vector is computed using

$$\boldsymbol{r}_{B/P} = \boldsymbol{r}_{B/O} - \boldsymbol{r}_{P/O} \tag{1}$$

If no planet ephemeris message is specified, then the planet position vector $\boldsymbol{r}_{P/O}$ is set to zero. Let $[PN]$ be the direction cosine matrix[1] that relates the rotating planet-fixed frame relative to an inertial frame $\mathcal{N} : \{\hat{\boldsymbol{n}}_1, \hat{\boldsymbol{n}}_2, \hat{\boldsymbol{n}}_3\}$. The simulation provides the spacecraft position vector in inertial frame components. The planet centric position vector is then written in Earth-fixed frame components using

$$^{\mathcal{P}}\boldsymbol{r}_{B/P} = [PN]\ ^{\mathcal{N}}\boldsymbol{r}_{B/P} \tag{2}$$

1.3 Magnetic Field Models

The truth of the magnetometer measurements in sensor frame coordinates with no errors are output as:

$$^{\mathcal{S}}\boldsymbol{B} = [SN]\ ^{\mathcal{N}}\boldsymbol{B} \tag{3}$$

where $[SN]$ is the direction cosine matrix[1] from $\mathcal{N}$ to $\mathcal{S}$, and $^{\mathcal{N}}\boldsymbol{B}$ is the magnetic field vector of the magnetic field model.

1.4 Error Modeling

The magnetic field vector of the magnetic field models is considered to be "truth" ($\mathbf{B}_{\text{truth}} = {}^{\mathcal{S}}\boldsymbol{B}$). So, to simulate the errors found in real instrumentation, errors are added to the "truth" values:

$$\mathbf{B}_{\text{measured}} = \mathbf{B}_{\text{truth}} + \mathbf{e}_{\text{noise}} + \mathbf{e}_{\text{bias}} \tag{4}$$

where $\mathbf{e}_{\text{noise}}$ is the Gaussian noise, and $\mathbf{e}_{\text{bias}}$ is the bias applied on the magnetic field measurements.

1.5 Saturation

Sensors might have specific saturation bounds for their measurements. It also prevents the sensor for giving a value less or higher than the possible hardware output. The saturated values are:

$$\mathbf{B}_{\text{sat}_{\text{min}}} = \min\big(\mathbf{B}_{\text{measured}}, \text{maxOutput}) \tag{5}$$

$$\mathbf{B}_{\text{sat}_{\text{max}}} = \max\big(\mathbf{B}_{\text{measured}}, \text{minOutput}) \tag{6}$$

This is the final output of the sensor module.

2 Module Functions

The magnetometer functions are described below:

- **Noise**: The module can apply Gaussian noise to the measurements.
- **Bias**: The module can apply a bias to the measurements.
- **Saturation**: The module bounds the output signal according to user-specified maximum and minimum saturation values.
- **Scale Factor**: The module can apply a scale factor to the measured value (truth + noise + bias). This is a linear scaling of the output.

- **Interface: Spacecraft State**: The module receives spacecraft state information from the messaging system.

- **Interface: Magnetic Field Vector**: The module receives magnetic field model information through the messaging system

3 Module Assumptions and Limitations

Assumptions made in TAM module and the corresponding limitations are shown below:

- **Magnetic Field Model Inputs**: The magnetometer sensor is limited with the used magnetic field model which are individual magnetic field models complex and having their own assumptions. The reader is referred to the cited literature to learn more about the model limitations and challenges.

- **Error Inputs**: Since the error models rely on user inputs, these inputs are the most likely source of error in TAM output. Instrument bias would have to be measured experimentally or an educated guess would have to be made. The Gauss-Markov noise model has well-known assumptions and is generally accepted to be a good model for this application.

- **External Disturbances**: Currently, the module does not consider the external magnetic field, so it is limited to the cases where this effect is not significant. This can be overcome by using magnetic field models taking into these effects account or adding it as an additional term.

4 Test Description and Success Criteria

This section describes the specific unit tests conducted on this module. The test contains 16 tests and is located at test_magnetometer.py. The simulation of the magnetic field truth vector is set up based on two magnetic field model test scripts: test_magneticFieldCenteredDipole.py and test_magneticFieldWMM.py. The success criteria is to match the outputs with the generated truth.

5 Test Parameters

Pytest runs the following cases in Table 2 when it is called for this test:

Table 2: Parameters for each test. Note that relative tolerance is $\frac{truth-output}{truth}$

Test	noiseStd	bias	saturation	magMode	**errTol**
1	0.0e+00	0.0e+00 ,0.0e+00 ,0.0e+00	-1.0e+03 ,1.0e+03	WMM	1.0e-10
2	0.0e+00	0.0e+00 ,0.0e+00 ,0.0e+00	-1.0e+03 ,1.0e+03	CenteredDipole	1.0e-10
3	0.0e+00	1.0e-06 ,1.0e-06 ,1.0e-05	-1.0e+03 ,1.0e+03	WMM	1.0e-10
4	0.0e+00	1.0e-06 ,1.0e-06 ,1.0e-05	-1.0e+03 ,1.0e+03	CenteredDipole	1.0e-10
5	0.0e+00	0.0e+00 ,0.0e+00 ,0.0e+00	-1.0e-04 ,1.0e-04	WMM	1.0e-10
6	0.0e+00	0.0e+00 ,0.0e+00 ,0.0e+00	-1.0e-04 ,1.0e-04	CenteredDipole	1.0e-10
7	3.0e-07	0.0e+00 ,0.0e+00 ,0.0e+00	-1.0e+03 ,1.0e+03	WMM	1.0e-06
8	3.0e-07	0.0e+00 ,0.0e+00 ,0.0e+00	-1.0e+03 ,1.0e+03	CenteredDipole	1.0e-06
9	0.0e+00	1.0e-06 ,1.0e-06 ,1.0e-05	-1.0e-04 ,1.0e-04	WMM	1.0e-10
10	0.0e+00	1.0e-06 ,1.0e-06 ,1.0e-05	-1.0e-04 ,1.0e-04	CenteredDipole	1.0e-10
11	3.0e-07	0.0e+00 ,0.0e+00 ,0.0e+00	-1.0e-04 ,1.0e-04	WMM	1.0e-06
12	3.0e-07	0.0e+00 ,0.0e+00 ,0.0e+00	-1.0e-04 ,1.0e-04	CenteredDipole	1.0e-06
13	3.0e-07	1.0e-06 ,1.0e-06 ,1.0e-05	-1.0e+03 ,1.0e+03	WMM	1.0e-06
14	3.0e-07	1.0e-06 ,1.0e-06 ,1.0e-05	-1.0e+03 ,1.0e+03	CenteredDipole	1.0e-06
15	3.0e-07	1.0e-06 ,1.0e-06 ,1.0e-05	-1.0e-04 ,1.0e-04	WMM	1.0e-06
16	3.0e-07	1.0e-06 ,1.0e-06 ,1.0e-05	-1.0e-04 ,1.0e-04	CenteredDipole	1.0e-06

6 Test Results

The following two tables show the test results. All tests are expected to pass.

Table 3: Test result for `test_unitTestMagneticField.py`

Test	Pass/Fail	Notes
1	PASSED	
2	PASSED	
3	PASSED	
4	PASSED	
5	PASSED	
6	PASSED	
7	PASSED	
8	PASSED	
9	PASSED	
10	PASSED	
11	PASSED	
12	PASSED	
13	PASSED	
14	PASSED	
15	PASSED	
16	PASSED	

7 User Guide

7.1 General Module Setup

This section outlines the steps needed to add a `Magnetometer` module to a sim. First, one of the magnetic field models must be imported:

```
from Basilisk.simulation import magneticFieldCenteredDipole
magModule = magneticFieldCenteredDipole.MagneticFieldCenteredDipole()
magModule.ModelTag = "CenteredDipole"
```

and/or

```
from Basilisk.simulation import magneticFieldWMM
magModule = magneticFieldWMM.MagneticFieldWMM()
magModule.ModelTag = "WMM"
```

Then, the magnetic field measurements must be imported and initialized:

```
from Basilisk.simulation import magnetometer
testModule = magnetometer.Magnetometer()
testModule.ModelTag = "TAM_sensor"
```

The model can be added to a task like other simModels.

```
unitTestSim.AddModelToTask(unitTaskName, testModule)
```

Each `Magnetometer` module calculates the magnetic field based on the magnetic field and output state messages of a spacecraft. To add spacecraft to the magnetic field model the spacecraft state output message name is sent to the `addScToModel` method:

```
scObject = spacecraftPlus.SpacecraftPlus()
scObject.ModelTag = "spacecraftBody"
magModule.addSpacecraftToModel(scObject.scStateOutMsgName)
```

The transformation from $\mathcal{B}$ to $\mathcal{S}$ can be set via `dcm_SB` using the helper function:

$$\texttt{setBodyToPlatformDCM}(\psi, \theta, \phi)$$

where (ψ, θ, ϕ) are classical $3-2-1$ Euler angles that map from the body frame to the sensor frame $\mathcal{P}$.

7.2 Specifying TAM Sensor Noise

Three types of TAM sensor corruptions can be simulated. If not specified, all these corruptions are zeroed.

To add a Gaussian noise component to the output, the variable

`SenNoiseStd`

is set to a non-zero value. This is the standard deviation of Gaussian noise in Tesla.

Next, to simulate a signal bias, the variable

`SenBias`

is set to a non-zero value. This constant bias of the Gaussian noise.

Finally, to set saturation values, the variables

$$\text{maxOutput}$$

$$\text{minOutput}$$

are used.

7.3 Connecting Messages

Two possible input messages to the TAM module, the following message inputs are required for the module to properly operate:

`MagneticFieldSimMsg`

`SCPlusStatesSimMsg`

The first message is used to get the magnetic field vector from one of the models, while the second message provides the spacecraft inertial orientation. The output message of the TAM module is then:

`TAMDataSimMsg`

REFERENCES

[1] Hanspeter Schaub and John L. Junkins. *Analytical Mechanics of Space Systems*. AIAA Education Series, Reston, VA, 4th edition, 2018.

[2] F. Landis Markley and John L. Crassidis. *Fundamentals of Spacecraft Attitude Determination and Control*. Springer, New York, 2014.

[3] Michael D. Grifin and James R. French. *Space Vehicle Design*. AIAA Education Series, Reston, VA, 2005.