Autonomous Vehicle Simulation (AVS) Laboratory, University of Colorado

Basilisk Technical Memorandum

Document ID: Basilisk-CSS Sensor Data

CSS COMML

Prepared by	J. Martin

Status: Initial Version
Scope/Contents
The CSS Sensor Data module is responsible for correcting course sun sensor data using a Chebyshev polynomial fit.

Rev	Change Description	By	Date
1.0	Initial Version	J. Martin	20190209

Contents

---

1 Model Description

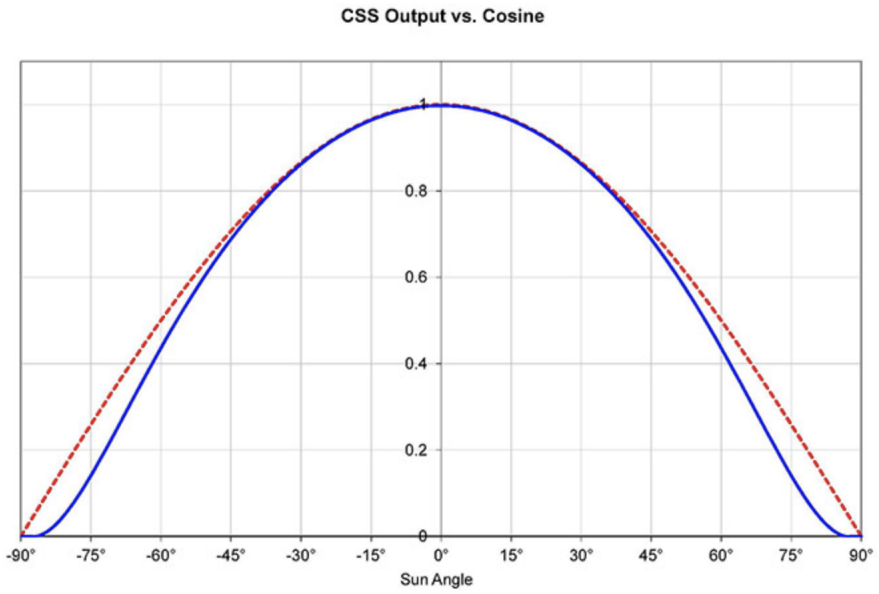

Fig. 1: Example of CSS output (blue) relative to a cosine curve (red).[1]

This module reads in raw sensor data from the `CSSArraySensorIntMsg`, message type, iterates through each raw CSS measurement, normalizes the measurement, checks that the input is within sensible bounds, and then corrects the measurement based on pre-calibrated Chebyshev polynomial. The corrected cosine measurement value is then written out in a `CSSArraySensorIntMsg`.

1.1 Equations

The Chebyshev forumlation follows the standard formulation:

$$T_{i+1}(x) = 2xT_i(x) - T_{i-1}(x) \tag{1}$$

where

$$T_0(x) = 1 \tag{2}$$

$$T_1(x) = x \tag{3}$$

As such, the algorithm to compute the Chebyshev polynomial approximation given by the following exerpt:[2]

1. Suppose we want to evaluate Chebyshev polynomial of order i at x_0, $(T_i(x_0))$
2. The first two order of Chebyshev polynomials can be evaluated using the following form

$$T_0(x) = 1 \tag{4}$$

$$T_1(x) = x \tag{5}$$

3. The Chebyshev polynomial of order $i > 1$ can be computed using the values of Chebyshev polynomials of order $i - 1$ and $i - 2$ and the following recursive formula:

$$T_{i+1}(x) = 2xT_i(x) - T_{i-1}(x) \tag{6}$$

4. Apply this formula up to the order i to evaluate Chebyshev polynomial of order i at x_0.

2 Module Functions

- **Corrects raw CSS data**: This module adjusts the raw CSS data to reflect sensor calibrations.
- **Sends out corrected values to the simulation**: This module outputs a `CSSArrayIntMsg` with to be used for guidance modules.

3 Module Assumptions and Limitations

The chebyshev calibration model and coefficients are calibrated for only one distance. As the spacecraft moves farther away from the distance for which the CSS were calibrated, the model will change. This should be most apparent at high sun angles.

4 Test Description and Success Criteria

The unit test checks for proper functionality of the module for various CSS configurations and input values, both within and outside expected bounds.

5 Test Parameters

Test and simulation parameters and inputs go here. Basically, describe your test in the section above, but put any specific numbers or inputs to the tests in this section.

The unit test verify that the module output guidance message vectors match expected values.

Table 2: Error tolerance for each test.

Output Value Tested	Tolerated Error
outputVector	1e-12

6 Test Results

The results of the unit test should be included in the documentation. The results can be discussed verbally, but also included as tables and figures.

All of the tests passed:

Table 3: Test results

Check	Pass/Fail
1	PASSED
2	PASSED
3	PASSED

6.1 Unit Test Table Results

To automatically create a unit test table to include in the documentation, use the command:

```
unitTestSupport.writeTableLaTeX(
tableName,
tableHeaders,
caption,
dataMatrix,
path)
```

Here are the sample TEX table form the unit tests.

Table 4: Sample output table for param1 = 1 and param2 = 1.

time [s]	Output 1	Error	Output 2	Error	Output 3 $\boldsymbol{r}$	Error
0	2	0	1	0	0.7	0
0.5	3	0	1	0	0.7	0
1	4	0	1	0	0.7	0
1.5	2	0	1	0	0.7	0
2	3	0	1	0	0.7	0

Table 5: Sample output table for param1 = 1 and param2 = 3.

time [s]	Output 1	Error	Output 2	Error	Output 3 $\boldsymbol{r}$	Error
0	2	0	3	0	0.7	0
0.5	3	0	3	0	0.7	0
1	4	0	3	0	0.7	0
1.5	2	0	3	0	0.7	0
2	3	0	3	0	0.7	0

6.2 Unit Test Figure Results

If figures and plots are generated in the python unit tests, these can be also automatically included in the unit test documentation. This is achieved with the command:

Table 6: Sample output table for param1 = 2 and param2 = 2.

time [s]	Output 1	Error	Output 2	Error	Output 3 $\boldsymbol{r}$	Error
0	3	0	2	0	0.7	0
0.5	4	0	2	0	0.7	0
1	5	0	2	0	0.7	0
1.5	3	0	2	0	0.7	0
2	4	0	2	0	0.7	0

```
unitTestSupport.writeFigureLaTeX(
"testPlot",
"Illustration of Sample Plot",
plt,
"width=0.5\\textwidth",
path)
```

7 User Guide

1. Input required: `CSSArrayIntMsg` which contains the raw cosine values of the CSS array.
2. Output required: `CSSArrayIntMsg` contains the corrected cosine values of the CSS array.

REFERENCES

[1] Adcole Corporation. Coarse sun sensor detector (cosine type).

[2] Makoto Nakajima. Note on chebyshev regression. 2006.