Autonomous Vehicle Simulation (AVS) Laboratory, University of Colorado

Basilisk Technical Memorandum

Document ID: Basilisk-RadiationPressure

RADIATION PRESSURE MODEL

Prepared by	S. Carnahan

Status: Initial document draft
Scope/Contents
The radiation pressure module is responsible for calculating the dyamic effects of radiation pressure on a spacecraft. Effects can be calculated either by use of a simplified "cannonball" method or table look-up. A unit test has been written which checks both calculation methods against expected output values. The code is working fully and providing accurate results.

Rev:	**Change Description**	**By**
Draft	Initial document creation	S. Carnahan

Contents

Nomenclature

AU	=	1 astronomical unit
SF_{AU}	=	average solar flux at 1 AU from the sun
SF_{eff}	=	effective solar flux at 1 AU given eclipse conditions
F_{SF}	=	solar factor (between 0 and 1) applied to the solar flux due to an eclipse
p_{SR}	=	pressure due to solar radiation
c	=	speed of light
$\mathbf{F}_{\text{radiation}}$	=	Force vector on spacecraft due to solar radiation
c_R	=	coefficient of reflectivity (between 0 and 2)
$A_{\odot}$	=	equivalent cross sectional area of the spacecraft as seen by solar radiation pressure
$\mathbf{r}_{\text{sun}}$	=	position vector from the spacecraft to the sun in the spacecraft body frame

1 Introduction

The Basilisk radiation pressure module (radiation_pressure.cpp) is responsible for calculating the effects of radiation pressure on a spacecraft. A basic flow of inputs and outputs is visualized in 1

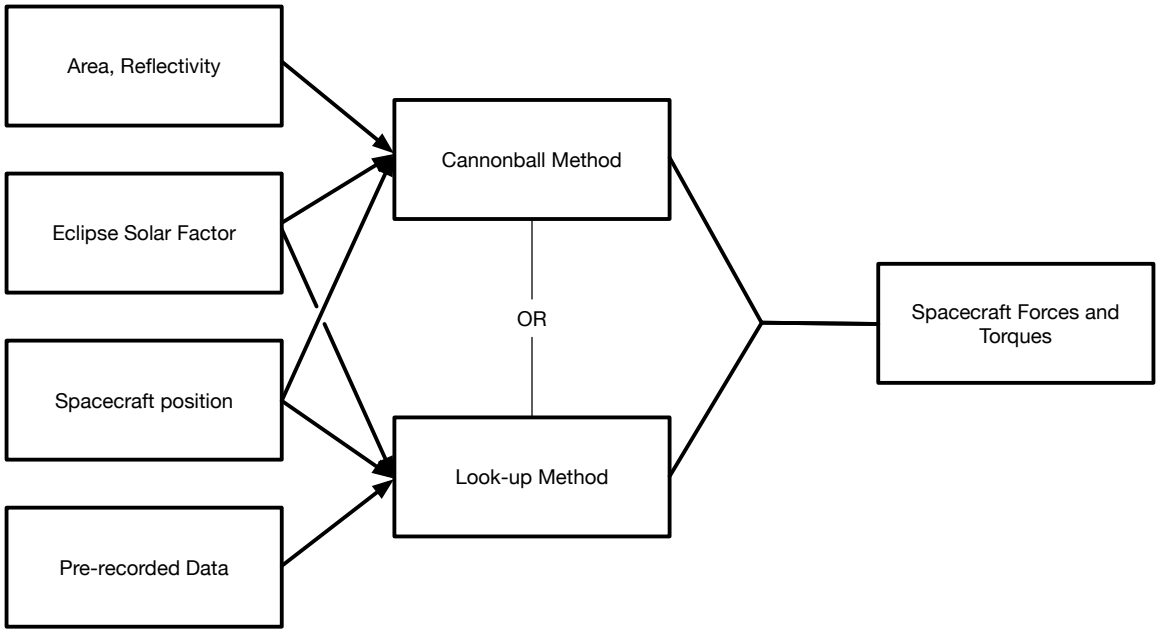

Fig. 1: Inputs necessary for solar radiation pressure calculations flow into either the cannonball method or the look-up method. These methods output their torque and force contributions to the spacecraft.

The mathematical models used are described below. The code unit tests are then presented and discussed.

2 Mathematical Model

Both methods of calculating solar radiation pressure have simple implementations in Basilisk using basic coefficients and assumptions. The methods of arriving at these coefficients can be complex and making the coefficients time-varying to improve accuracy can greatly increase complexity. The cannonball method used here essentially follows the mathematics described by Vallado.

2.1 Radiation Model

Radiation is modeled by using the solar flux at one astronomical unit and scaling by distance from the sun relative to 1 AU. The solar flux at one AU is taken as :

$$SF_{AU} = 1372.5398[W/m^2] \tag{1}$$

2.1.1 Solar Eclipses

Solar eclipses are are detected by the basilisk eclipse module. The effects of the eclipse are calculated into a solar factor which is applied to the solar flux to create an effective solar flux.

$$SF_{\text{eff}} = F_{SF}(SF_{AU}) \tag{2}$$

Then, calculations are carried on as seen below.

2.2 Radiation Pressure Model

2.2.1 Cannonball Method

The radiation pressure at 1AU, p_{SR}, can be taken as the solar flux divided by the speed of light.

$$p_{SR} = \frac{SF_{\text{eff}}}{c} \tag{3}$$

Then, a "scaling factor" can be determined. This "scaling factor" is equivalent to the magnitude of the solar radiation force divided by the distance between the spacecraft and the sun:

$$\frac{|\mathbf{F}_{\text{radiation}}|}{|\mathbf{r}_{\text{sun}}|} = \frac{-c_R p_{SR} A_{\odot} AU^2}{c|\mathbf{r}_{\text{sun}}|^3} \tag{4}$$

This factor is then multiplied by the position vector from the spacecraft to the sun to get the force on the spacecraft due to solar radiation pressure.

$$\mathbf{F}_{\text{radiation}} = \frac{|\mathbf{F}_{\text{radiation}}|}{|\mathbf{r}_{\text{sun}}|}\mathbf{r}_{\text{sun}} \tag{5}$$

The user must provide the coefficient of reflection and the equivalent area of the spacecraft to use this method.

2.2.2 Table Look-up Method

For the table look-up method, pre-determined values of torque and force acting on the spacecraft due to radiation pressure are given. It is required that these values be given at 1AU from the sun and with a corresponding direction vector form the spacecraft to the sun.

The look-up works by finding the direction vector in the given tables which most closely matches the spacecraft's current position vector. Then, the corresponding force and torque values are taken from the table and scaled according to the magnitude of the spacecraft-sun position vector.

Most important to the user of the table look-up method is the required input and format of data. Data must be recorded in XML format. As an example, see ../cube_lookup.xml (in the radiation pressure folder). Additionally, a utility script called parseSRPLookup.py is provided there to read the XML input into numpy arrays. Experienced users are welcome to store their data in their own format and load it into equivalent numpy arrays as they see fit.

An example of using the provided python script to load data is shown in test_radiationPressure.py . Note that this also requires import of the unitTestSupport library.

2.3 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
stateInMsgName	N/A	string	Default setting: "inertial_state_output". This is the message from which the radiation pressure module receives spacecraft inertial data.
sunEphmInMsgName	N/A	string	Default setting: "sun_planet_data". This is the message through which radiation pressure gets information about the sun and planets.
area	$A_{\odot}$	double	[m2] Default setting: 0.0f. Required input for cannonball method to get any real output. This is the area to use when approximating the surface area of the spacecraft.
coefficientReflection	c_R	double	Default setting: 1.2. This is a factor applied to the radiation pressure based on spacecraft surface properties.
useCannonballModel	N/A	bool	Default setting: True. To switch cannonball mode off, use the call setUseCannonballModel("False")

3 Library

This section means nothing to me.

4 Unit Test

This test is located at `SimCode/dynamics/RadiationPressure/_UnitTest/test_radiationPressure.py` In order to get good coverage of all the aspects of the module, the test is broken up into two sub-tests. In each sub-test, a spacecraft is placed in the solar system and acted upon by the Sun.

4.1 "Cannonball" Method

This test utilizes the "cannonball" method to calculate the effects of radiation pressure on spacecraft dynamics. The cannonball method approximates the spacecraft as a sphere. External forces in the inertial and body frame, as well as external torques in the body frame, are checked against known values.

4.2 Table Look-up Method

This test uses a stored table of known effects of radiation pressure. It looks up values and compares them to the expected result to validate radiation pressure table look-up capabilities.

4.3 Test Parameters

This section summarizes the 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
"Cannonball"	1.0e-12
Look-up	1.0e-12

4.4 Test Results

All checks within test_radiationPressure.py passed as expected. Table 4 shows the test results.

Table 4: Test results.

Test	Pass/Fail	Notes
"Cannonball"	PASSED	
Look-up	PASSED	

4.5 Test Coverage

The test covers 2000% of the code.

5 Conclusion

What a great piece of code.