Autonomous Vehicle Simulation (AVS) Laboratory, University of Colorado

Basilisk Technical Memorandum

Document ID: Basilisk-celestialTwoBodyPoint

GUIDANCE MODULE FOR CELESTIAL TWO BODY POINT

Prepared by	M. Cols

Status: Version 1.2	
Scope/Contents	
This module point one body-fixed axis towards a primary celestial object. The secondary goal is to point a second body-fixed axis towards another celestial object. For example, the goal is to point the sensor towards the center of a planet while doing the best to keep the solar panel normal point at the sun.	

Rev	Change Description	By	Date
1.0	Initial copy	M. Cols	2018-04-23
1.1	Reviewed Version	T. Teil	2019-03-11

Contents

1 Model Description

1.1 Module Goal

This module computes a reference whose aim is to track the center of a primary target, e.g. pointing the communication antenna at the Earth, at the same time of trying to meet a secondary constraint as best as possible, e.g. a solar panel normal axis pointing the closest in the direction of the Sun. It is important to note that two pointing conditions in a three-dimensional space compose an overdetermined problem. Thus, the main constraint is always priorized over the secondary one so the former can always be met.

Figure 1 shows the case where Mars is the primary celestial body and the Sun is the secondary one. Note that the origin of the desired reference frame $\mathcal{R}$ is the position of the spacecraft.

Assuming knowledge of the position of the spacecraft $\boldsymbol{r}_{B/N}$ and the involved celestial bodies, $\boldsymbol{R}_{P1}$ and $\boldsymbol{R}_{P2}$ (all of them relative to the inertial frame $\mathcal{N}$ and expressed in inertial frame components), the relative position of the celestial bodies with respect to the spacecraft is obtained by simple subtraction:

$$\boldsymbol{R}_{P1} = \boldsymbol{R}_{P} - \boldsymbol{r}_{B/N} \tag{1a}$$

$$\boldsymbol{R}_{P2} = \boldsymbol{R}_{S} - \boldsymbol{r}_{B/N} \tag{1b}$$

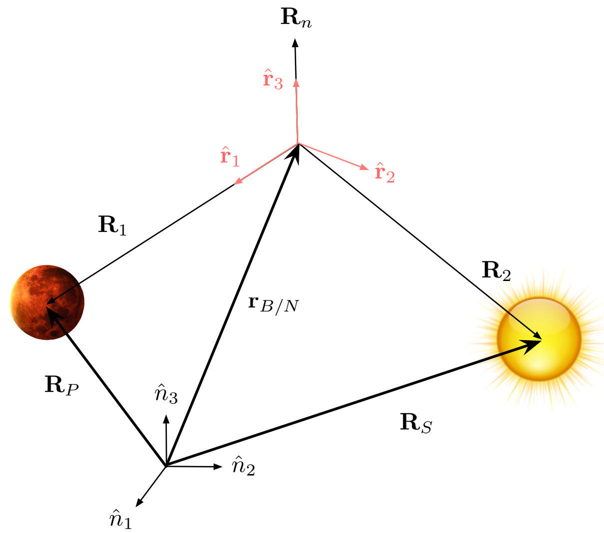

Fig. 1: Illustration of the restricted two-body pointing reference frame $\mathcal{R} : \{\hat{\boldsymbol{r}}_1, \hat{\boldsymbol{r}}_1, \hat{\boldsymbol{r}}_2\}$ and the inertial frame $\mathcal{N} : \{\hat{\boldsymbol{n}}_1, \hat{\boldsymbol{n}}_1, \hat{\boldsymbol{n}}_2\}$.

In analogy, the inertial derivatives of these position vectors are obtained:

$$\boldsymbol{v}_{P1} = \boldsymbol{v}_P - \boldsymbol{v}_{B/N} \tag{2a}$$

$$\boldsymbol{v}_{P2} = \boldsymbol{v}_S - \boldsymbol{v}_{B/N} \tag{2b}$$

$$\boldsymbol{a}_{P1} = \boldsymbol{a}_P - \boldsymbol{a}_{B/N} \tag{2c}$$

$$\boldsymbol{a}_{P2} = \boldsymbol{a}_S - \boldsymbol{a}_{B/N} \tag{2d}$$

The normal vector $\boldsymbol{R}_n$ of the plane defined by $\boldsymbol{R}_{P1}$ and $\boldsymbol{R}_{P2}$ is computed through:

$$\boldsymbol{R}_n = \boldsymbol{R}_{P1} \times \boldsymbol{R}_{P2} \tag{3}$$

The inertial time derivative of $\boldsymbol{R}_n$ is found using the chain differentiation rule:

$$\boldsymbol{v}_n = \boldsymbol{v}_{P1} \times \boldsymbol{R}_{P2} + \boldsymbol{R}_{P1} \times \boldsymbol{v}_{P2} \tag{4}$$

And the second time derivative:

$$\boldsymbol{a}_n = \boldsymbol{a}_{P1} \times \boldsymbol{R}_{P2} + \boldsymbol{R}_{P1} \times \boldsymbol{a}_{P2} + 2\boldsymbol{v}_{P1} \times \boldsymbol{v}_{P2} \tag{5}$$

1.2 Special Case: No Secondary Constraint Applicable

If there is no incoming message with a secondary celestial body pointing condition or if the constrain is not valid, an artificial three-dimensional frame is defined instead. Note that a single condition pointing leaves one degree of freedom, hence standing for an underdetermined attitude problem. A secondary constrain is considered to be invalid if the angle between $\boldsymbol{R}_{P1}$ and $\boldsymbol{R}_{P2}$ is, in absolute value, minor than a set threshold. This could be the case where the primary and secondary celestial bodies are aligned as seen by the spacecraft. In such situation, the primary pointing axis would already satisfy both the primary and the secondary constraints.

Since the main algorithm of this module, which is developed in the following sections, assumes two conditions, the second one is arbitrarily set as following:

$$\boldsymbol{R}_{P2} = \boldsymbol{R}_{P1} \times \boldsymbol{v}_{P1} \equiv \boldsymbol{h}_{P1} \tag{6}$$

By setting the secondary constrain to have the direction of the angular momentum vector $\boldsymbol{h}_{P1}$, it is assured that it will always be valid ($\boldsymbol{R}_{P1}$ and $\boldsymbol{R}_{P2}$ are now perpendicular). The first and second time derivatives are steadily computed:

$$\boldsymbol{v}_{P2} = \boldsymbol{R}_{P1} \times \boldsymbol{a}_{P1} \tag{7}$$

$$\boldsymbol{a}_{P2} = \boldsymbol{v}_{P1} \times \boldsymbol{a}_{P1} \tag{8}$$

1.3 Reference Frame Definition

As illustrated in Figure 1, the base vectors of the desired reference frame $\mathcal{R}$ are defined as following:

$$\hat{\boldsymbol{r}}_1 = \frac{\boldsymbol{R}_{P1}}{|\boldsymbol{R}_{P1}|} \tag{9a}$$

$$\hat{\boldsymbol{r}}_3 = \frac{\boldsymbol{R}_n}{|\boldsymbol{R}_n|} \tag{9b}$$

$$\hat{\boldsymbol{r}}_2 = \hat{\boldsymbol{r}}_3 \times \hat{\boldsymbol{r}}_1 \tag{9c}$$

Since the position vectors are given in terms of inertial $\mathcal{N}$-frame components, the DCM from the inertial frame $\mathcal{N}$ to the desired reference frame $\mathcal{R}$ is:

$$[RN] = \begin{bmatrix} {}^{\mathcal{N}}\hat{\boldsymbol{r}}_1^T \\ {}^{\mathcal{N}}\hat{\boldsymbol{r}}_2^T \\ {}^{\mathcal{N}}\hat{\boldsymbol{r}}_3^T \end{bmatrix} \tag{10}$$

1.4 Base Vectors Time Derivatives

The first and second time derivatives of the base vectors that compound the reference frame $\mathcal{R}$ are needed in the following sections to compute the reference angular velocity and acceleration. Several lines of algebra lead to the following sets:

$$\dot{\hat{\boldsymbol{r}}}_1 = ([I_{3\times3}] - \hat{\boldsymbol{r}}_1\hat{\boldsymbol{r}}_1^T)\frac{\boldsymbol{R}_{P1}}{|\boldsymbol{R}_{P1}|} \tag{11a}$$

$$\dot{\hat{\boldsymbol{r}}}_3 = ([I_{3\times3}] - \hat{\boldsymbol{r}}_3\hat{\boldsymbol{r}}_3^T)\frac{\boldsymbol{R}_n}{|\boldsymbol{R}_n|} \tag{11b}$$

$$\dot{\hat{\boldsymbol{r}}}_2 = \dot{\hat{\boldsymbol{r}}}_3 \times \boldsymbol{r}_1 + \boldsymbol{r}_n \times \dot{\hat{\boldsymbol{r}}}_3 \tag{11c}$$

$$\ddot{\hat{\boldsymbol{r}}}_1 = \frac{1}{|\boldsymbol{R}_{P1}|}((([I_{3\times3}] - \hat{\boldsymbol{r}}_1\hat{\boldsymbol{r}}_1^T)\boldsymbol{a}_{P1} - 2\dot{\hat{\boldsymbol{r}}}_1(\hat{\boldsymbol{r}}_1 \cdot \boldsymbol{v}_{P1}) - \hat{\boldsymbol{r}}_1(\dot{\hat{\boldsymbol{r}}}_1 \cdot \boldsymbol{v}_{P1})) \tag{12a}$$

$$\ddot{\hat{\boldsymbol{r}}}_3 = \frac{1}{|\boldsymbol{R}_n|}((([I_{3\times3}] - \hat{\boldsymbol{r}}_3\hat{\boldsymbol{r}}_3^T)\boldsymbol{a}_n - 2\dot{\hat{\boldsymbol{r}}}_3(\hat{\boldsymbol{r}}_3 \cdot \boldsymbol{v}_n) - \hat{\boldsymbol{r}}_3(\dot{\hat{\boldsymbol{r}}}_3 \cdot \boldsymbol{v}_n)) \tag{12b}$$

$$\ddot{\hat{\boldsymbol{r}}}_2 = \ddot{\hat{\boldsymbol{r}}}_3 \times \boldsymbol{r}_1 + \boldsymbol{r}_n \times \ddot{\hat{\boldsymbol{r}}}_3 + 2\dot{\hat{\boldsymbol{r}}}_3 \cdot \dot{\hat{\boldsymbol{r}}}_1 \tag{12c}$$

1.5 Angular Velocity and Acceleration Descriptions

Developing some more mathematics, the following elegant expressions of $\boldsymbol{\omega}_{R/N}$ and $\dot{\boldsymbol{\omega}}_{R/N}$ are found:

$$\boldsymbol{\omega}_{R/N} \cdot \hat{\boldsymbol{r}}_1 = \hat{\boldsymbol{r}}_3 \cdot \dot{\hat{\boldsymbol{r}}}_2 \tag{13a}$$

$$\boldsymbol{\omega}_{R/N} \cdot \hat{\boldsymbol{r}}_2 = \hat{\boldsymbol{r}}_1 \cdot \dot{\hat{\boldsymbol{r}}}_3 \tag{13b}$$

$$\boldsymbol{\omega}_{R/N} \cdot \hat{\boldsymbol{r}}_3 = \hat{\boldsymbol{r}}_2 \cdot \dot{\hat{\boldsymbol{r}}}_1 \tag{13c}$$

$$\dot{\boldsymbol{\omega}}_{R/N} \cdot \hat{\boldsymbol{r}}_1 = \dot{\hat{\boldsymbol{r}}}_3 \cdot \dot{\hat{\boldsymbol{r}}}_2 + \hat{\boldsymbol{r}}_3 \cdot \ddot{\hat{\boldsymbol{r}}}_2 - \boldsymbol{\omega}_{R/N} \cdot \dot{\hat{\boldsymbol{r}}}_1 \tag{14a}$$

$$\dot{\boldsymbol{\omega}}_{R/N} \cdot \hat{\boldsymbol{r}}_2 = \dot{\hat{\boldsymbol{r}}}_1 \cdot \dot{\hat{\boldsymbol{r}}}_3 + \hat{\boldsymbol{r}}_1 \cdot \ddot{\hat{\boldsymbol{r}}}_3 - \boldsymbol{\omega}_{R/N} \cdot \dot{\hat{\boldsymbol{r}}}_2 \tag{14b}$$

$$\dot{\boldsymbol{\omega}}_{R/N} \cdot \hat{\boldsymbol{r}}_3 = \dot{\hat{\boldsymbol{r}}}_2 \cdot \dot{\hat{\boldsymbol{r}}}_1 + \hat{\boldsymbol{r}}_2 \cdot \ddot{\hat{\boldsymbol{r}}}_1 - \boldsymbol{\omega}_{R/N} \cdot \dot{\hat{\boldsymbol{r}}}_3 \tag{14c}$$

Note that $\boldsymbol{\omega}_{R/N} \cdot \hat{\boldsymbol{r}}_1$ is the first component of the angular velocity of the reference with respect to the inertial expressed in reference frame components. Hence,

$$\boldsymbol{\omega}_{R/N} = {}^{\mathcal{R}}\begin{bmatrix} \boldsymbol{\omega}_{R/N} \cdot \hat{\boldsymbol{r}}_1 \\ \boldsymbol{\omega}_{R/N} \cdot \hat{\boldsymbol{r}}_2 \\ \boldsymbol{\omega}_{R/N} \cdot \hat{\boldsymbol{r}}_3 \end{bmatrix} \tag{15}$$

Similarly for the angular acceleration:

$$\dot{\boldsymbol{\omega}}_{R/N} = {}^{\mathcal{R}}\begin{bmatrix} \dot{\boldsymbol{\omega}}_{R/N} \cdot \hat{\boldsymbol{r}}_1 \\ \dot{\boldsymbol{\omega}}_{R/N} \cdot \hat{\boldsymbol{r}}_2 \\ \dot{\boldsymbol{\omega}}_{R/N} \cdot \hat{\boldsymbol{r}}_3 \end{bmatrix} \tag{16}$$

Eventually, in inertial frame components:

$${}^{\mathcal{N}}\boldsymbol{\omega}_{R/N} = [RN]\ {}^{\mathcal{R}}\boldsymbol{\omega}_{R/N} \tag{17a}$$

$${}^{\mathcal{N}}\dot{\boldsymbol{\omega}}_{R/N} = [RN]\ {}^{\mathcal{R}}\dot{\boldsymbol{\omega}}_{R/N} \tag{17b}$$

2 Module Functions

- **parseInputMessages**: This method takes the navigation translational info as well as the spice data of the primary celestial body and, if applicable, the second one, and computes the relative state vectors necessary to create the restricted 2-body pointing reference frame.
- **computeCelestialTwoBodyPoint**: This method takes the spacecraft and points a specified axis at a named celestial body specified in the configuration data. It generates the commanded attitude and assumes that the control errors are computed downstream.

3 Module Assumptions and Limitations

The module assumes for now that the planetary acceleration vectors are zero.

4 Test Description and Success Criteria

The mathematics in this module are straight forward and can be tested in a series of input and output evaluation tests.

4.1 Test 1

The first test does a single-body celestial point. It places a spacecraft around Earth on an orbit with parameters given in 2. The earth position and velocity vectors are both set to the zero vectors.

4.2 Test 2

The second test does the 2-body celestial point. It uses the same parameters at above and sets a second planet with zero velocity and position vector:

$${}^{\mathcal{N}}\boldsymbol{r}_2 = \begin{bmatrix} 500 & 500 & 500 \end{bmatrix}^T \text{(km)} \tag{18}$$

Table 2: Spacecraft Orbital Paramters

Orbital Parameter	Value
a	$2.8R_{\text{earth}}$
e	0
i	0°
Ω	0°
ω	0°
f	60°

Table 3: Error tolerance for each test.

Output Value Tested	Tolerated Error
$\boldsymbol{\sigma}_{R/N}$	1e-12
${}^{\mathcal{N}}\boldsymbol{\omega}_{R/N}$	1e-12
${}^{\mathcal{N}}\dot{\boldsymbol{\omega}}_{R/N}$	1e-12

5 Test Parameters

For each of these two tests, the tested parameters are listed Table 3.

6 Test Results

All of the tests passed:

Table 4: Test results

Check	Pass/Fail
1.1	PASSED
1.2	PASSED
1.3	PASSED
2.1	PASSED
2.2	PASSED
2.3	PASSED

7 User Guide

7.1 Input/Output Messages

The module has 2 required input messages, 1 optional input message and 1 output message:

- `inputNavDataName` – This input message, of type `NavTransIntMsg`, provide the translational-navigation states for the spacecraft.
- `inputCelMessName` – This input message, of type `EphemerisIntMsg`, receives the first planet states for pointing
- `inputSecMessName` – (Optional) This input message, of type `EphemerisIntMsg`, receives the second planet states for pointing
- `outputDataName` – This output message, of type `AttRefFswMsg`, writes out the attitude, rate, and inertial derivative of the rate in order to perform control.

7.2 Module Parameters and States

Outside of the message names, this module only has one other parameter:

- `singularityThresh` - This parameter determines the threshold after which two vectors are considered collinear.