Autonomous Vehicle Simulation (AVS) Laboratory,
University of Colorado

Basilisk Technical Memorandum

Document ID: Basilisk-spacecraftPointing

SPACECRAFT POINTING MODULE

Prepared by	S. van Overeem

Status: First Public Release
Scope/Contents
The primary purpose of this module is to provide an attitude reference output to make sure that a vector given in the deputy spacecraft's body frame points to the chief spacecraft. The position of the chief- and deputy spacecraft in the inertial frame are used as inputs for this module. The module uses the positions to create a reference vector that points from the deputy to the chief. A coordinate system is built around this vector and the orientation, angular velocity, and angular acceleration of this coordinate system are calculated with respect to the inertial frame. The output consists of these three vectors and can consequently be used as an input for the attitude tracking error module.

Rev	**Change Description**	**By**	**Date**
1.0	First documentation on this module	S. van Overeem	20190116

Contents

1 Model Description 1
1.1 Module Goal . . . 1
1.2 Equations . . . 1
2 Module Functions 4
3 Module Assumptions and Limitations 4
4 Test Description and Success Criteria 6
4.1 Test 1 . . . 6
4.2 Test 2 . . . 6
5 Test Parameters 6
6 Test Results 6
7 User Guide 7
7.1 Input/Output Messages . . . 7
7.2 Module Parameters and States . . . 7

1 Model Description

1.1 Module Goal

This module is intended for controlled formation flying applications where it is necessary to set up at least a chief spacecraft and a deputy spacecraft. The goal of this module is to align a vector defined by the user in the deputy body frame (${}^{\mathcal{B}}\boldsymbol{a}$) with a vector that points from the deputy to the chief spacecraft (${}^{\mathcal{N}}\boldsymbol{\rho}$). Both vectors are displayed in Fig. 1. The ${}^{\mathcal{B}}\boldsymbol{a}$ vector defined by the user can, for instance, be the location of an antenna or a camera within the body frame. Besides the ${}^{\mathcal{B}}\boldsymbol{a}$ vector, the module uses the location of the chief and deputy spacecraft in the inertial frame as an input. The outputs of this module consist of the orientation of a reference frame that is built around the ${}^{\mathcal{N}}\boldsymbol{\rho}$ vector, the angular velocity of the reference frame, and the angular acceleration (all with respect to the inertial frame). Together with the attitude of the deputy spacecraft body frame, the outputs of this module can be fed into the attitude tracking error module. This module consequently calculates the error between the reference frame built around the ${}^{\mathcal{N}}\boldsymbol{\rho}$ vector and the spacecraft body frame to make sure that the body frame eventually aligns with this reference frame.

1.2 Equations

As discussed in Sec. 1.1 the inputs of this module are the position of the chief and the deputy spacecraft with respect to the inertial frame (${}^{\mathcal{N}}\boldsymbol{r}_{\text{chief}}$ and ${}^{\mathcal{N}}\boldsymbol{r}_{\text{deputy}}$ respectively). In order to find the unit vector that points from the deputy to the chief Eq. (1) can be used.

$$ {}^{\mathcal{N}}\hat{\boldsymbol{\rho}} = \frac{{}^{\mathcal{N}}\boldsymbol{r}_{\text{chief}} - {}^{\mathcal{N}}\boldsymbol{r}_{\text{deputy}}}{||{}^{\mathcal{N}}\boldsymbol{r}_{\text{chief}} - {}^{\mathcal{N}}\boldsymbol{r}_{\text{deputy}}||} \tag{1} $$

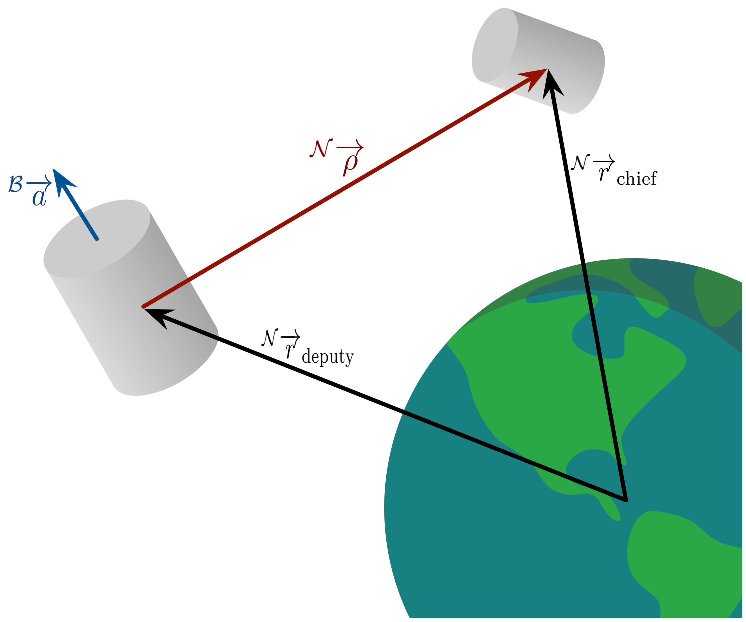

Fig. 1: Vector illustrations

Consequently, a coordinate system is built around the $^{\mathcal{N}}\hat{\boldsymbol{\rho}}$ vector as can be seen in Eq. (2). In this equation, $^{\mathcal{N}}\hat{\boldsymbol{x}}$, $^{\mathcal{N}}\hat{\boldsymbol{y}}$, and $^{\mathcal{N}}\hat{\boldsymbol{z}}$ are the normalized x-, y-, and z-components of the $\mathcal{R}$ frame written in $\mathcal{N}$ frame components. Furthermore, $\hat{\boldsymbol{z}}$ represents the z-component in the $\mathcal{N}$ frame ($[0, 0, 1]^T$). The entries in the direction cosine matrix (dcm) can be observed in Eq. (3).

$$\begin{aligned} ^{\mathcal{N}}\hat{\boldsymbol{x}} &= {}^{\mathcal{N}}\hat{\boldsymbol{\rho}} \\ ^{\mathcal{N}}\hat{\boldsymbol{y}} &= \hat{\boldsymbol{z}} \times {}^{\mathcal{N}}\hat{\boldsymbol{\rho}} \\ ^{\mathcal{N}}\hat{\boldsymbol{z}} &= {}^{\mathcal{N}}\hat{\boldsymbol{x}} \times {}^{\mathcal{N}}\hat{\boldsymbol{y}} \end{aligned} \tag{2}$$

$$[RN] = \begin{bmatrix} ^{\mathcal{N}}\hat{\boldsymbol{x}}^T \\ ^{\mathcal{N}}\hat{\boldsymbol{y}}^T \\ ^{\mathcal{N}}\hat{\boldsymbol{z}}^T \end{bmatrix} \tag{3}$$

Using the "C2MRP" function in C this dcm is converted to the Modified Rodrigues Parameter (MRP) vector $\boldsymbol{\sigma}_{RN}$.

Consequently, it is possible to calculate the angular velocity of the $\mathcal{R}$ frame with respect to the $\mathcal{N}$ frame in $\mathcal{N}$ frame components. In order to find the angular velocity $\boldsymbol{\sigma}_{RN}$ at time t-1 is subtracted from $\boldsymbol{\sigma}_{RN}$ at time t and divided by the timestep (Δt) as can be seen in Eq. (4).

$$\dot{\boldsymbol{\sigma}}_{RN} = \frac{\boldsymbol{\sigma}_{RN}(\text{t}) - \boldsymbol{\sigma}_{RN}(\text{t-1})}{\Delta\text{t}} \tag{4}$$

After that, the angular velocity of the $\mathcal{R}$ frame with respect to the $\mathcal{N}$ frame in $\mathcal{R}$ frame components can be computed using Eq. (5) (equation 3.164 in Analytical Mechanics of Space Systems[1]). The sigma components used in this equation are the average sigma between sigma at t-1 and sigma at t. This is done due to the fact that using the sigma components at solely time t results in incorrect values for omega. In case the chief would orbit the deputy circularly with a known rate it turns out that the results from the numerical method fluctuate around the true angular velocity. Taking the average of sigma results in an angular velocity that is in coherence with values calculated analytically and is constant up to 10^{-9}. Using a dcm ${}^{\mathcal{N}}\boldsymbol{\omega}_{RN}$ is calculated from ${}^{\mathcal{R}}\boldsymbol{\omega}_{RN}$. Finally, the angular acceleration (${}^{\mathcal{N}}\dot{\boldsymbol{\omega}}_{RN}$) can be calculated using Eq. (6).

$$
{}^{\mathcal{R}}\boldsymbol{\omega}_{RN} = \frac{4}{(1+\sigma^2)^2}[(1-\sigma^2)[I_{3\times3}] - 2[\tilde{\boldsymbol{\sigma}}] + 2\boldsymbol{\sigma}\boldsymbol{\sigma}^T]\dot{\boldsymbol{\sigma}}_{RN} \tag{5}
$$

$$
{}^{\mathcal{N}}\dot{\boldsymbol{\omega}}_{RN} = \frac{{}^{\mathcal{N}}\boldsymbol{\omega}_{RN}(\text{t}) - {}^{\mathcal{N}}\boldsymbol{\omega}_{RN}(\text{t-1})}{\Delta\text{t}} \tag{6}
$$

As stated in Sec. 1.1, the attitude tracking error module calculates the error of the deputy spacecraft's attitude with respect to the attitude of the reference frame (the $\mathcal{R}$ frame). For this reason, using $\boldsymbol{\sigma}_{RN}$ would result in an alignment of the x-axis of the deputy spacecraft's body frame with the ${}^{\mathcal{N}}\hat{\boldsymbol{\rho}}$ vector instead of an alignment of the vector specified by the user (${}^{\mathcal{B}}\boldsymbol{a}$) with the ${}^{\mathcal{N}}\hat{\boldsymbol{\rho}}$ vector. So instead of using the $\mathcal{R}$ frame as one of the outputs of this module, it is necessary to create a different coordinate frame. The name of this frame is the $\mathcal{R}_1$ frame and in case the $\mathcal{B}$ frame aligns with the $\mathcal{R}_1$ frame, ${}^{\mathcal{B}}\boldsymbol{a}$ has to align with ${}^{\mathcal{N}}\boldsymbol{\rho}$. For this reason, an $\mathcal{A}$ frame is built around the ${}^{\mathcal{B}}\boldsymbol{a}$ vector. The creation of this coordinate frame can be found in the Reset function in the code. The vector components of this coordinate system can be seen in Eq. (7). In this equation, $\hat{\boldsymbol{z}}$ represents the z-component of the body frame.

$$
\begin{aligned}
{}^{\mathcal{B}}\hat{\boldsymbol{x}} &= {}^{\mathcal{B}}\hat{\boldsymbol{a}} \\
{}^{\mathcal{B}}\hat{\boldsymbol{y}} &= \hat{\boldsymbol{z}} \times {}^{\mathcal{B}}\hat{\boldsymbol{a}} \\
{}^{\mathcal{B}}\hat{\boldsymbol{z}} &= {}^{\mathcal{B}}\hat{\boldsymbol{x}} \times {}^{\mathcal{B}}\hat{\boldsymbol{y}}
\end{aligned} \tag{7}
$$

The dcm that consequently maps the $\mathcal{B}$ frame to the $\mathcal{A}$ frame can be found in Eq. (8).

$$
[AB] = \begin{bmatrix} {}^{\mathcal{B}}\hat{\boldsymbol{x}}^T \\ {}^{\mathcal{B}}\hat{\boldsymbol{y}}^T \\ {}^{\mathcal{B}}\hat{\boldsymbol{z}}^T \end{bmatrix} \tag{8}
$$

This dcm is transformed in $\boldsymbol{\sigma}_{AB}$ using the "C2MRP" function in C.

In order to make sure that the $\mathcal{A}$ frame aligns with the $\mathcal{R}$ frame (meaning that ${}^{\mathcal{B}}\boldsymbol{a}$ aligns with ${}^{\mathcal{N}}\hat{\boldsymbol{\rho}}$), the following computations are performed and illustrated using a 2D example. Looking at Fig. 2 the orientation of the $\mathcal{N}$ frame can be observed. Besides that, it is possible to see a $\mathcal{B}$ frame and an $\mathcal{A}$ frame relative to this $\mathcal{B}$ frame. Furthermore, Fig. 2 shows the orientation of the $\mathcal{R}$ frame and the $\mathcal{R}_1$ frame. Up until this point $\boldsymbol{\sigma}_{RN}$ and $\boldsymbol{\sigma}_{AB}$ are calculated. The goal is to align the $\mathcal{A}$ frame with the $\mathcal{R}$ frame. This is the same as aligning the $\mathcal{B}$ frame with the $\mathcal{R}_1$ frame (noting that $\boldsymbol{\sigma}_{AB} = \boldsymbol{\sigma}_{RR_1}$). For this reason, the reference MRP that will be used as an output is $\boldsymbol{\sigma}_{R_1N}$ and can be calculated using Eq. (9) (because $\boldsymbol{\sigma}_{BA} = -\boldsymbol{\sigma}_{AB}$).

$$
\boldsymbol{\sigma}_{R_1N} = \boldsymbol{\sigma}_{RN} + \boldsymbol{\sigma}_{BA} \tag{9}
$$

The angular velocity and angular acceleration stay the same because both the angular velocity and the angular acceleration of the $\mathcal{A}$ frame with respect to the $\mathcal{B}$ frame are constant.

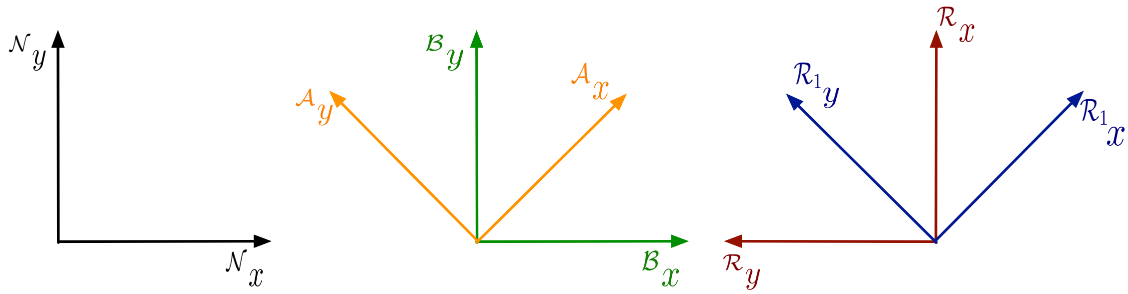

Fig. 2: Aligning coordinate systems example

2 Module Functions

- **Compute vector pointing to chief:** The inputs of the module are the location of the chief- and deputy spacecraft in the inertial frame. From these inputs, it is possible to calculate the vector that points from the deputy to the chief.
- **Compute the orientation of the reference frame with respect to the inertial frame:** After a coordinate system is built around the vector that points to the chief it is possible to determine the orientation of the reference frame with respect to the inertial frame.
- **Compute the angular velocity:** From the change in sigma over the timestep it is possible to compute the angular velocity of the reference frame.
- **Compute the angular acceleration:** Dividing the change in angular velocity of the reference frame over the timestep results in the angular acceleration.

3 Module Assumptions and Limitations

- The user has to manually give a non-zero vector within the body frame as input. If this is not done an error occurs.
- Due to the fact that a numerical approximation is used for the determination of the angular velocity and the angular acceleration, the first two data points of the angular velocity and the first three data points for the angular acceleration are nonsense. For this reason, these are set to zero in the code.
- Due to the numerical method used the accuracy is heavily dependent on the timestep. This can be observed in Fig. 3, Fig. 4, and Fig. 5. As the timestep decreases, the error decreases as well and smoothens out.

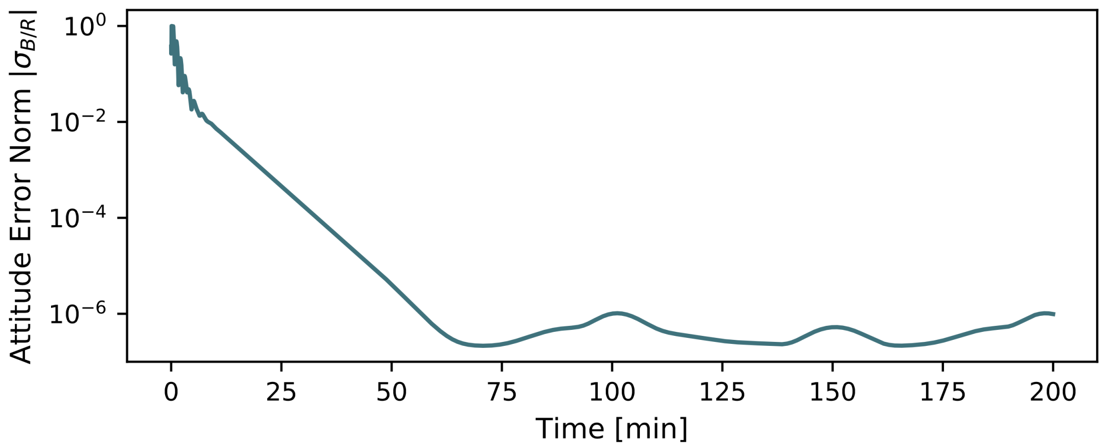

Fig. 3: Timestep equal to 1s

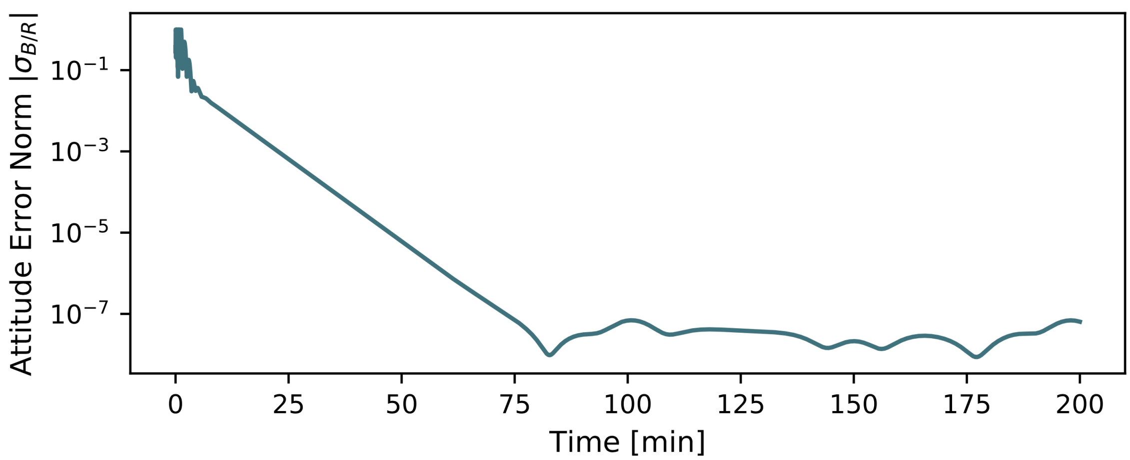

Fig. 4: Timestep equal to 0.1s

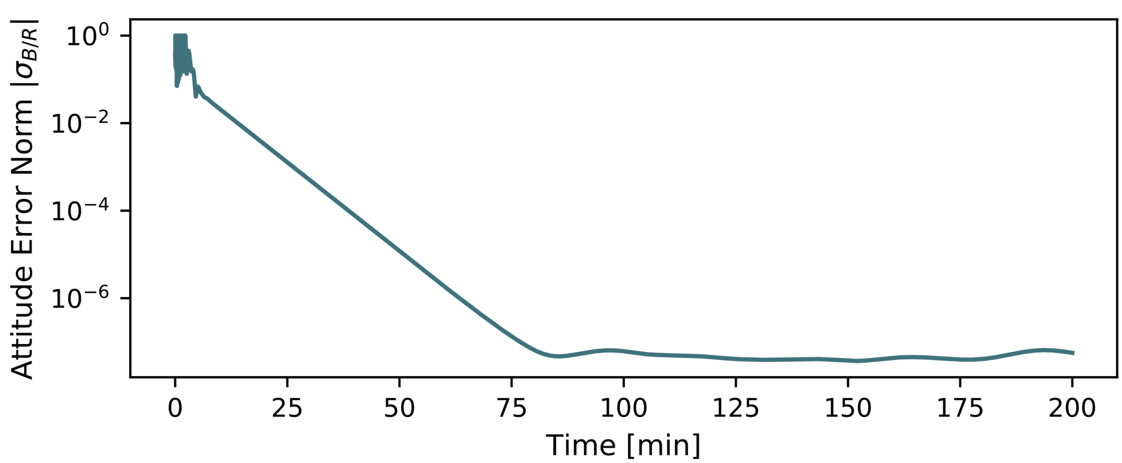

Fig. 5: Timestep equal to 0.01s

4 Test Description and Success Criteria

4.1 Test 1

This test that is performed in order to verify that the module works is a test that inputs a chief position vector and a deputy position vector for each timestep. The chief position input vector looks like the following.

```
[[np.cos(0.0), np.sin(0.0), 0.0],
[np.cos(0.001), np.sin(0.001), 0.0],
[np.cos(0.002), np.sin(0.002), 0.0],
[np.cos(0.003), np.sin(0.003), 0.0],
[np.cos(0.004), np.sin(0.004), 0.0]]
```

The chief position input vector looks like:

```
[[0.0, 0.0, 0.0],
[0.0, 0.0, 0.0],
[0.0, 0.0, 0.0],
[0.0, 0.0, 0.0],
[0.0, 0.0, 0.0]]
```

The chief thus makes a circle around the deputy. A total of three checks are performed. The first check checks whether $\boldsymbol{\sigma}_{R_1N}$ is in coherence with the expected value. The second check checks whether ${}^{\mathcal{N}}\boldsymbol{\omega}_{RN}$ is equal to the expected value and the final check checks for ${}^{\mathcal{N}}\dot{\boldsymbol{\omega}}_{RN}$. In case all three checks are successful, the module is considered working.

4.2 Test 2

In order to verify the robustness of the creation of a coordinate system around the vector defined by the user (${}^{\mathcal{B}}\boldsymbol{a}$) the following input is used: [0,0,1]. This vector aligns with the z-axis of the deputy body frame. This would result in an undefined cross product between the z-axis of the body frame and ${}^{\mathcal{B}}\boldsymbol{a}$. For this reason, in case these two vectors are aligned, the module takes the cross product between ${}^{\mathcal{B}}\boldsymbol{a}$ and the y-axis in the body frame. To test whether this is redundant, $\boldsymbol{\sigma}_{BA}$ is tested for this input.

5 Test Parameters

For an alignmentVector (${}^{\mathcal{B}}\boldsymbol{a}$) that is equal to [1.0, 0.0, 0.0] the unit test verifies that the module output reference message vectors match expected values. Besides that, the second test verifies that $\boldsymbol{\sigma}_{BA}$ is indeed the value that is expected.

Output Value Tested	Tolerated Value
$\boldsymbol{\sigma}_{R_1N}$	1e-12
${}^{\mathcal{N}}\boldsymbol{\omega}_{RN}$	1e-09
${}^{\mathcal{N}}\dot{\boldsymbol{\omega}}_{RN}$	1e-12
$\boldsymbol{\sigma}_{BA}$	1e-12

6 Test Results

Table 2: Test results

Check	Pass/Fail
Test 1	PASSED
Test 2	PASSED

7 User Guide

7.1 Input/Output Messages

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

- `chiefPositionInMsg` - This input message, of type `NavTransMsgPayload`, consists of the location of the chief spacecraft in the inertial frame.
- `deputyPositionInMsg` - This input message, of type `NavTransMsgPayload`, consists of the location of the deputy spacecraft in the inertial frame.
- `attReferenceOutMsg` - This output message, of type `AttRefMsgPayload`, consists of the orientation, angular velocity and angular acceleration of the reference frame with respect to the inertial frame.

7.2 Module Parameters and States

The module has the following parameter that can be configured:

- `alignmentVector` - [REQUIRED] This 3x1 array contains the body-relative vector that points towards e.g. the antenna and needs to be aligned with the vector that points from the deputy spacecraft to the chief spacecraft.

REFERENCES

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