JOURNAL OF AEROSPACE INFORMATION SYSTEMS
Vol. 15, No. 6, June 2018

Modular Attitude Guidance: Generating Rotational Reference Motions for Distinct Mission Profiles

M. Cols-Margenet,* H. Schaub,† and S. Piggott‡
University of Colorado, Boulder, Colorado 80309

DOI: 10.2514/1.I010554

A modular scheme for generating attitude reference motions is presented. This scheme is a generic strategy to develop the guidance profile for a wide range of mission scenarios through combination of atomic guidance modules that fulfill a well-defined functionality. There are three core functionalities that conform all guidance reference motions: base pointing reference, dynamic reference, and attitude offset. The base pointing functionality encompasses a fixed pointing requirement, for instance inertial pointing, orbit Hill-frame pointing, orbit velocity-frame pointing, or constrained two-body pointing. The dynamic reference functionality defines a dynamic reference motion relative to the base pointing reference, for instance relative fixed spinning or relative scanning. Finally, an attitude offset can be added to the composed reference to account for the requirement of controlling a spacecraft-fixed frame that is not the main body frame. For each one of these functionalities, different software guidance modules are implemented and showcased; the mathematical equations are developed, and numerical simulations illustrate how the individual components are arranged to support different rotational dynamics mission requirements.

I. Introduction

SPACE missions rely highly on the efficiency and reliability of the onboard flight software (FSW), whose core is conformed by the guidance, navigation, and controls (GN&C) tasks. Although this paper only addresses the calculation of satellite rotational guidance maneuvers, it is worthwhile to recall how the three GN&C tasks relate to each other.

Navigation: The current orientation and rates of the spacecraft are estimated using absolute and relative measurements. A range of estimation filtering methods are possible, with many of them being the focus of ongoing research [1–5].

Guidance: The desired attitude reference that the spacecraft should assume is computed and compared to the current state. Typically, a customized set of guidance software is written to meet mission specific needs.

Control: A concrete control law is implemented to drive a spacecraft body frame toward a desired reference motion. Attitude control methods are actively studied, using both linear [6] and nonlinear [7–9] approaches.

Mission GN&C flight algorithms undergo a stringent review and validation process before flight, which can be both costly and time-consuming. Developing flight algorithms through modular designs (i.e., breaking up functionality into a series of simple independent pieces or modules instead of having one large software piece that performs a complex function) has shown to improve efficiency in terms of implementation and testing. Traditionally, talking about modularity of GN&C would mean that there are separate modules used for sensory input, parameter identification, reference trajectory selection/generation, position error determination, control, and output. This kind of split is seen in the GN&C sequences presented in [10,11], for example.

The present work focuses specifically on the attitude reference trajectory generation and aims to bring modularity one step further by dividing the reference generation itself into multiple, exchangeable subcomponents as well. Note that “selecting” a reference implies that the motion is predefined by ground and uploaded onboard. In turn, “generating” a reference implies computing the desired motion autonomously onboard, and this is the concept of operations addressed in the paper. Without loss in generality, this work investigates a methodology to modularize any attitude reference motion into three atomic parts: a base pointing reference, a dynamic reference that is relative to the base, and an attitude offset.

The final, desired reference motion is a result of cascading these three attitude reference parts, as depicted in Fig. 1. Different combinations of pointing and dynamic references yield guidance scenarios of distinct complexity. All proposed guidance schemes are applicable to any type of Keplerian orbit, including elliptic circumnavigation and hyperbolic flyby trajectories.

The mathematics of the paper show the transformations needed to build up complex guidance motions by combination of different attitude/rotational behaviors. Once the desired reference is generated (guidance task), it can be compared to the current state of the spacecraft (estimated in the navigation task), to derive an attitude tracking error and implement a control law to correct the spacecraft state (control task). A closed-loop simulation using the Basilisk software framework that involves navigation, guidance, and controls is used to prove numerically the mathematics of the guidance reference algorithms developed in this paper.

Hence, as core part of the proposed work, different guidance software modules are developed to perform the same function: computing the correct desired rotational motion but for different mission objectives.

Figure 2 illustrates sample mission objective profiles that have associated specific attitude reference motion requirements. These profiles are star tracker recalibration, inertial scanning, nadir pointing, communication antenna pointing, Hill-axis spinning, and planet scanning. With a traditional approach, each final reference would be evaluated through a unique and very specific algorithm. Nevertheless, it is shown in the same figure that these guidance maneuvers can also be achieved as a composition of core guidance functionalities. The advantage of fractionating the reference generation is that, for a given set of core modules, a wide variety of guidance behaviors can be achieved through combination and distinct initialization. In Fig. 2, the boxes in blue tones correspond to sample modules that provide a base pointing reference functionality. The green boxes correspond to sample dynamic reference motions that can be added on top of a base, and the red box is simply the attitude offset that, in this work, is

Presented at the AIAA/AAS Astrodynamics Specialist Conference, Long Beach, CA, 12–15 September 2016; received 28 March 2017; revision received 10 January 2018; accepted for publication 21 March 2018; published online 24 April 2018. Copyright © 2018 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved. All requests for copying and permission to reprint should be submitted to CCC at www.copyright.com; employ the ISSN 2327-3097 (online) to initiate your request. See also AIAA Rights and Permissions www.aiaa.org/randp.

*Graduate Student, Aerospace Engineering Sciences. Member AIAA.

†Associate Chair of Graduate Affairs, Glenn L. Murphy Endowed Chair, Department of Aerospace Engineering Sciences, 431 UCB, Colorado Center for Astrodynamics Research. Associate Fellow AIAA.

‡ADCS Integrated Simulation Software Lead, Laboratory for Atmospheric and Space Physics. Member AIAA.

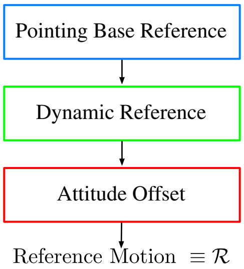


Fig. 1 Steps in the generation of the attitude reference motion.

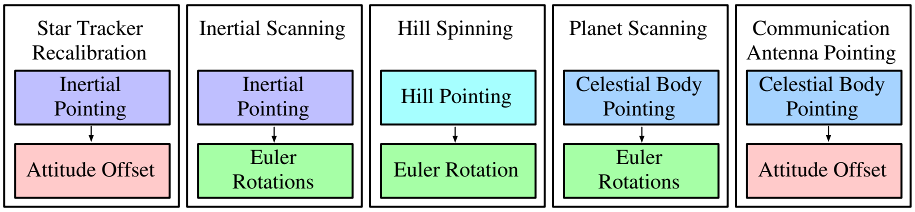


Fig. 2 Sample mission objectives.

implemented within a module that will be labeled later on as tracking error module. Let us take a closer look at the commonalities between the different mission objective profiles in Fig. 2. For example, both a rotation about a Hill frame axis (i.e., "Hill spinning" profile) and a scanning maneuver across a planet (i.e., "planet scanning" profile) can be performed through constant Euler rates defined relative to a specific pointing attitude. Although the pointing nature is in this case distinct (i.e., Hill pointing module versus celestial body pointing module) the rotational functionality is shared (i.e., Euler rotation module). In other instances, it is the pointing reference that is common; pointing a communication antenna toward the Earth (i.e., "communication antenna pointing" profile) and scanning across a planet (i.e., "planet scanning" profile) are both based on a celestial body pointing module. Finally, the difference between driving the principal body frame to a particular reference orientation, or aligning an alternate body-fixed frame (e.g., boresight of a star tracker in "star tracker recalibration" profile or boresight of an antenna in "communication antenna pointing" profile) is simply a constant attitude offset for a rigid-body spacecraft, hence the attitude offset module.

Encapsulating the guidance functionalities in completely independent modules instead of monolithic algorithms is a key aspect in terms of software safety. With the aim of bringing down mission risks, the suggested staging of independent guidance modules allows scaling up the functionality in a safe and systematic manner. Complexity is built through layers of atomic modules and the decoupling between these units simplifies the verification/validation process because they can be individually tested and analyzed. Although the verification of the individual components by itself does not guarantee the combined algorithm is without errors, modularity helps greatly on isolating errors effectively and identifying the root cause of emerging behaviors in complex GN&C algorithms and sequences. As a matter of fact, this work presents several numerical simulations with compound guidance maneuvers that can be regarded as integrated tests. The results section discusses further how modularity and integrated testing dovetail together.

The paper is outlined as follows. First, the guidance generation and control setup used throughout the paper are reviewed (Sec. II). An overview of the Basilisk astrodynamics software is presented upon which the proposed guidance solutions are implemented (Sec. III). Next, the base pointing reference modules are explained, including a detailed mathematic development of the uncommon constrained two-body pointing algorithm (Sec. IV). Then, the dynamic reference modules are discussed, which consider spinning about either a fixed or relative axis, and performing maneuvers defined by constant Euler rates relative to the base reference frame (Sec. V). Finally the attitude tracking error module is presented (Sec. VI). Numerical simulations in the form of integrated tests illustrate the staging strategy to achieve increasingly complex scenarios (Sec. VII). Upon the results of the numerical simulations, which serve as a proof of concept for the proposed architecture, some highlights are drawn regarding the advantages of the layering approach for prototyping and testing (Sec. VIII). Finally, the tradeoffs between a modular guidance sequence and a single custom algorithms are discussed (also Sec. VII).

II. Problem Statement

A. Attitude Guidance Behavior

The goal of the GN&C process is to estimate the current state of the spacecraft body $\mathcal{B}$ (navigation), generate a reference state $\mathcal{R}$ that can be time-varying or not (guidance), derive the attitude tracking error between the current state $\mathcal{B}$ and the desired one $\mathcal{R}$ (also guidance), and apply the required control torque to align $\mathcal{B}$ with $\mathcal{R}$ (control).

Once again, the present work focuses on the guidance process. Both the spacecraft state $\mathcal{B}$ and the final reference state $\mathcal{R}$ are expressed with respect to an inertial frame $\mathcal{N}$ that is common for all the FSW tasks. Note that $\mathcal{N}$ is kept as a generic inertial frame to be chosen (e.g., it could be J2000, Earth fixed-frame, Mars centered frame, etc.) because all that matters is consistency on the picked inertial frame throughout the entire FSW.

The computed reference state $\mathcal{R}$ is, in this paper, composed of three parameters: an inertial attitude measure, denoted through the modified Rodrigues parameters (MRPs) set $\boldsymbol{\sigma}_{R/N}$ [8,12,13], an inertial angular rate vector ${}^{\mathcal{N}}\boldsymbol{\omega}_{R/N}$ expressed in inertial frame $\mathcal{N}$ components, and an inertial angular acceleration vector ${}^{\mathcal{N}}\dot{\boldsymbol{\omega}}_{R/N}$ also in $\mathcal{N}$-frame components. The left superscript denotes the frame with respect to which the vector components are taken. Note that the nomenclature of the rotational state $\mathcal{R} = [\boldsymbol{\sigma}_{R/N}, {}^{\mathcal{N}}\boldsymbol{\omega}_{R/N}, {}^{\mathcal{N}}\dot{\boldsymbol{\omega}}_{R/N}]$ is taken from [8].

On a side note, the use of MRPs in the present work is simply a choice, and any other attitude description like quaternions or direction cosine matrices (DCMs) could be used instead. The use of MRPs is attractive because they are a nonredundant set (composed by three components only). Although MRPs are singular only for rotations multiples of 2π, these singularities can be avoided. It is noted that the MRPs are not unique, but rather there are always two possible MRP sets that can describe a particular orientation. This alternate MRP is known as the shadow MRP set. The shadow MRP set is singular for zero rotations but is nonsingular for rotations of 2π. Therefore, singularity avoidance can be performed by switching between the two sets. Further, both sets obey the same differential equations, making for easy implementation.

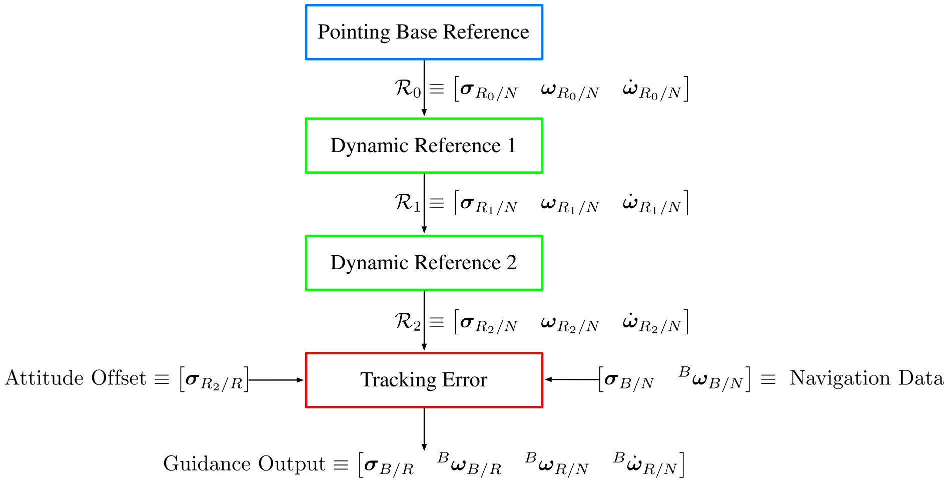

Pointing Base Reference
$\mathcal{R}_0 \equiv [\boldsymbol{\sigma}_{R_0/N} \quad \boldsymbol{\omega}_{R_0/N} \quad \dot{\boldsymbol{\omega}}_{R_0/N}]$
Dynamic Reference 1
$\mathcal{R}_1 \equiv [\boldsymbol{\sigma}_{R_1/N} \quad \boldsymbol{\omega}_{R_1/N} \quad \dot{\boldsymbol{\omega}}_{R_1/N}]$
Dynamic Reference 2
$\mathcal{R}_2 \equiv [\boldsymbol{\sigma}_{R_2/N} \quad \boldsymbol{\omega}_{R_2/N} \quad \dot{\boldsymbol{\omega}}_{R_2/N}]$
Attitude Offset $\equiv [\boldsymbol{\sigma}_{R_2/R}]$
Tracking Error
$[\boldsymbol{\sigma}_{B/N} \quad {}^B\boldsymbol{\omega}_{B/N}] \equiv$ Navigation Data
Guidance Output $\equiv [\boldsymbol{\sigma}_{B/R} \quad {}^B\boldsymbol{\omega}_{B/R} \quad {}^B\boldsymbol{\omega}_{R/N} \quad {}^B\dot{\boldsymbol{\omega}}_{R/N}]$

Fig. 3 Illustration of the inputs and outputs of a sample multireference guidance chain.

Figure 3 illustrates the generation of a compounded attitude reference $\mathcal{R}$. A guidance maneuver with this same architecture is showcased later in Sec. VII.D yielding a spiral-shaped scanning. As a brief general preview, the compounded reference in Fig. 3 is computed through addition of a base pointing reference (i.e., definition of $\mathcal{R}_0$ relative to $\mathcal{N}$), a first dynamic reference motion relative to the base (i.e., definition of $\mathcal{R}_1$ relative to $\mathcal{R}_0$, which results in the output of $\mathcal{R}_1$ relative to $\mathcal{N}$), a second dynamic reference relative to the former one (i.e., definition of $\mathcal{R}_2$ relative to $\mathcal{R}_1$, which results in the output of $\mathcal{R}_2$ relative to $\mathcal{N}$), and an attitude offset (i.e., definition of $\mathcal{R}$ relative to $\mathcal{R}_2$, which comes as an input).

The tracking error module is always the last component in the guidance chain. This module reads the output of the second dynamic reference module (i.e., $\mathcal{R}_2$ relative to $\mathcal{N}$) and adds an attitude offset if applicable (i.e., definition of $\mathcal{R}$ relative to $\mathcal{R}_2$). This happens when the generated dynamic reference is meant for a spacecraft-fixed frame that is not the main one. In this way, the very final desired reference is obtained (i.e., $\mathcal{R}$ relative to $\mathcal{N}$). On the other hand, the tracking error module is also provided with information of the current state of the main spacecraft body frame (spacecraft attitude $\boldsymbol{\sigma}_{B/N}$ and rate $\boldsymbol{\omega}_{B/N}$), through the navigation data message in Fig. 3. With knowledge of the navigation state $\mathcal{B}$ and the guidance state $\mathcal{R}$, derived relative to a common inertial frame $\mathcal{N}$, tracking errors can be readily computed. The output of the tracking error module is also the final output of the whole guidance block, and it is used to feed the control block next.

It is essential that all the reference generation modules (base and dynamic) output the same message structure, as in Fig. 3. This feature allows multiple modules to be readily chained in a sequential manner to achieve complex reference motions. In the overall modular stack, each data set enters the system only once and is then transferred downstream as necessary. For instance, in Fig. 3, the inertial $\mathcal{N}$ frame is an explicit input to the base pointing module, and the dynamic reference modules only have implicit knowledge of it. Having a single point of contact with the individual incoming and outgoing data packets avoids redundancy and the conflicts that may arise from it. Note that this section makes use of generic reference frames $\mathcal{R}_0$, $\mathcal{R}_1$, $\mathcal{R}_2$, $\mathcal{R}$. In the core sections of the paper (Secs. IV–VII), particular definitions for all these frames are illustrated.

B. Attitude Control Modified Rodrigues Parameter Feedback

This section is related to controls and not to guidance. Therefore, it is not a core part of the paper, but it is necessary to allow the reader to replicate the numerical simulations presented. The final control equations used in the simulations are outlined without deepening on the developments behind. For further insight, the reader is pointed to [8]. Presenting the control equations is also useful to realize how the output of the guidance task, which is the focus of this work, is used downstream in the FSW process. Figure 4 depicts the flow from one task to another. The equations outlined next belong to the algorithms that compute a control torque ("control law" module in Fig. 4) and map it to the set of chosen spacecraft actuators ("control torque mapping" module in Fig. 4).

In the numerical simulations of this paper, a rigid spacecraft with $N = 4$ reaction wheels (RWs) is modeled. The associated differential equations of motion (EOM) are the following:

$$[I]\dot{\boldsymbol{\omega}}_{B/N} = -[\tilde{\boldsymbol{\omega}}_{B/N}]\Big([I]\boldsymbol{\omega}_{B/N} + [G_s]\boldsymbol{h}_s\Big) - [G_s]\boldsymbol{u}_s + \boldsymbol{L} \tag{1}$$

where $[I]$ is the spacecraft inertia tensor, $\boldsymbol{L}$ is an external torque, and $\boldsymbol{u}_s$ is the set of RW motor torques. The RW spin axes are defined in the $3 \times N$ projection matrix

$$[G_s] = \begin{bmatrix} \hat{\boldsymbol{g}}_{s_1} & \cdots & \hat{\boldsymbol{g}}_{s_N} \end{bmatrix} \tag{2}$$

where $\hat{\boldsymbol{g}}_{s_i}$ is the ith RW spin axis. The $N \times 1$ RW inertial angular momentum matrix $\boldsymbol{h}_s$ is

$$\boldsymbol{h}_s = \begin{bmatrix} J_{s_1}(\boldsymbol{\omega}_{B/N} \cdot \hat{\boldsymbol{g}}_{s_1} + \Omega_1) \\ \vdots \\ J_{s_N}(\boldsymbol{\omega}_{B/N} \cdot \hat{\boldsymbol{g}}_{s_N} + \Omega_N) \end{bmatrix} \tag{3}$$

where J_{s_i} is the RW spin axis inertia.

Given the EOM in Eq. (1), the control law implemented is an MRP feedback law that is globally asymptotically stabilizing:

$$[G_s]\boldsymbol{u}_s = K\boldsymbol{\sigma}_{B/R} + [P]\boldsymbol{\omega}_{B/R} - \boldsymbol{\omega}_{R/N} \times ([I]\boldsymbol{\omega}_{B/N} + [G_s]\boldsymbol{h}_s) + [I](\boldsymbol{\omega}_{B/N} \times \boldsymbol{\omega}_{R/N} - \dot{\boldsymbol{\omega}}_{R/N}) + \boldsymbol{L} \tag{4}$$

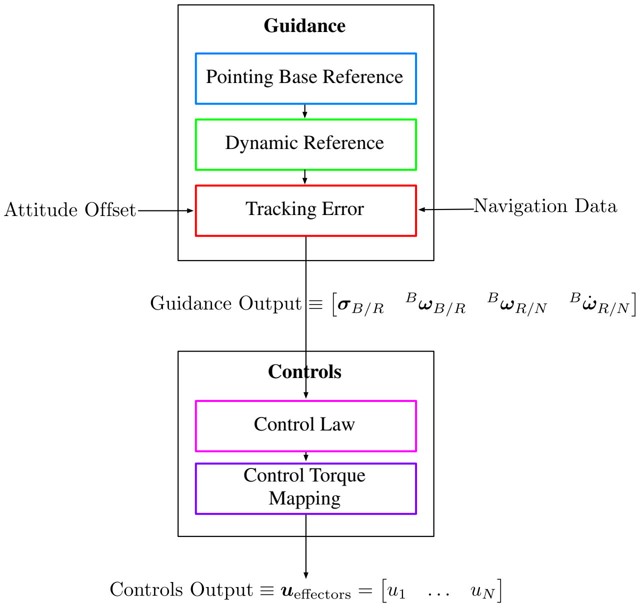

Guidance
Pointing Base Reference
Dynamic Reference
Tracking Error
Attitude Offset
Navigation Data
Guidance Output $\equiv [\boldsymbol{\sigma}_{B/R} \quad {}^{B}\boldsymbol{\omega}_{B/R} \quad {}^{B}\boldsymbol{\omega}_{R/N} \quad {}^{B}\dot{\boldsymbol{\omega}}_{R/N}]$
Controls
Control Law
Control Torque Mapping
Controls Output $\equiv \boldsymbol{u}_{\text{effectors}} = [u_1 \quad \ldots \quad u_N]$

Fig. 4 Illustration of the guidance and control tasks.

Here, $\boldsymbol{u}_s$ is the control torque being computed, K is the attitude error gain, and $[P]$ is the rate error gain matrix. The control block is fed with the guidance output, as illustrated in Fig. 4. This information includes the MRP attitude error $\boldsymbol{\sigma}_{B/R}$, the body rate error ${}^{\mathcal{B}}\boldsymbol{\omega}_{B/N}$, the reference rate ${}^{\mathcal{B}}\boldsymbol{\omega}_{R/N}$, and the reference inertial acceleration ${}^{\mathcal{B}}\dot{\boldsymbol{\omega}}_{R/N}$.

Note that, in Eq. (4), the inertial rates and accelerations as well as all the inertia and other matrix components are expressed in body $\mathcal{B}$-frame components. Now, it becomes evident that controlling a spacecraft frame that is not the main body $\mathcal{B}$-frame (e.g., star tracker component frame $\mathcal{B}_c$) has an impact in the guidance-control sequence. A common approach is to map the vector and tensors of Eq. (4) into the component body $\mathcal{B}_c$-frame aimed to be guided and controlled. In this paper, a different strategy is proposed that is more efficient; the offset from the main body frame $\mathcal{B}$ to the control frame $\mathcal{B}_c$ is added to the generated reference. This function is performed as part of the attitude tracking error module, and it was introduced previously with the concept of the “attitude offset”. Essentially, a difference in spacecraft control frame is treated instead as a difference in the final guidance reference. The advantage of this trick is that the vector and matrix components in Eq. (4), which are usually known in $\mathcal{B}$-frame components, never have to be mapped or recomputed. The mathematics associated to the attitude offset approach are presented later on in Sec. VI.

On a final note, let us recall that the particular feedback law outlined in Eq. (4) is not critical to the presented work. Any other asymptotically stable attitude control law could be substituted without impacting the guidance results.

III. Overview of the Basilisk Astrodynamics Software Framework

The Autonomous Vehicle System Laboratory at the University of Colorado and the Laboratory for Atmospheric and Space Physics are building a software tool for prototyping and developing FSW named Basilisk. The Basilisk framework is conformed by two independent computer processes: a high-fidelity simulation of the physical spacecraft (including dynamic, kinematic, and environment models, i.e., dynamics, kinematics, and environment suite) and a flight algorithm GN&C set (FSW algorithms). These two processes are developed in a modular architecture using C/C++ modules that communicate with each other through a message passing interface, as shown in Fig. 5. Having an FSW suite and a dynamics, kinematics, and environment (DKE) suite in different processes allows the migration of the flight algorithms to different

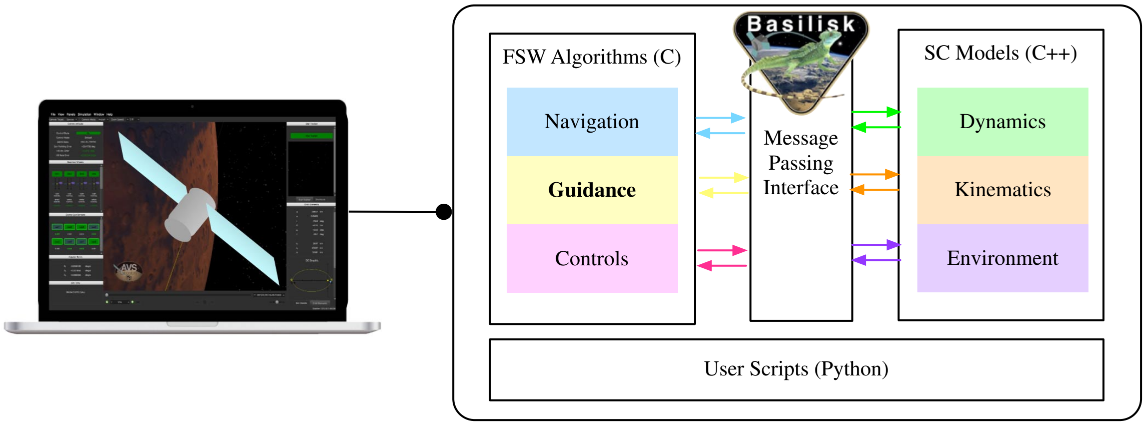


Fig. 5 Internal architecture of Basilisk, where the guidance block is the focus of this work.

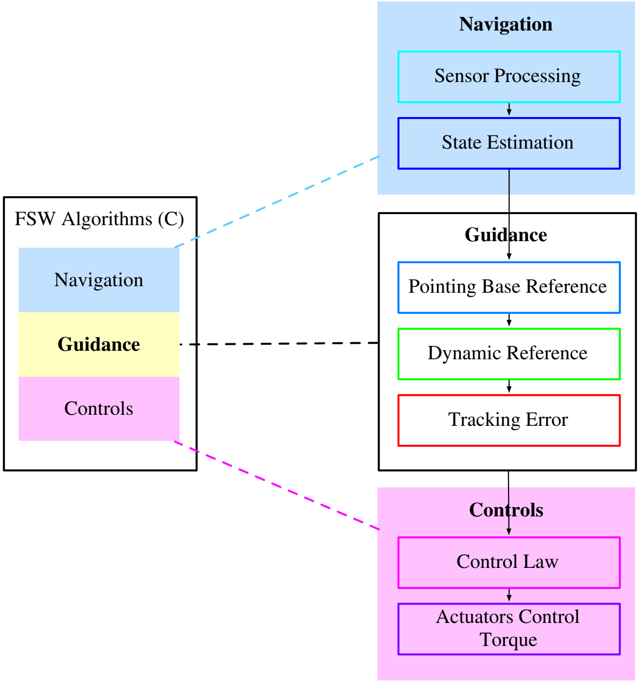


Fig. 6 FSW tasks leveraged through Basilisk: navigation and controls. Flight algorithms under testing: guidance.

testbeds (pure software testbed, hardware-in-the-loop testbed, etc.). Basilisk belongs to the category of pure software testbed, where all the hardware is also simulated.

The Basilisk framework is important in this paper because it is used to prove, through numerical simulation, the validity of the guidance mathematical developments that conform the core of this work. The FSW process is depicted in Fig. 6. Built-in Basilisk modules for navigation and controls are used; in turn, the guidance modules are the ones under development and integrated testing.

To prove the validity of the guidance behaviors at an integrated test level, reaction wheels are chosen as the attitude control actuators (i.e., continuous control), and all sensor corruptions are set to zero (i.e., navigation provides the true spacecraft states). Thus, if the reference frame, along with the associate reference rates and accelerations, are properly evaluated (i.e., the developed mathematic equations are correct and properly implemented in the algorithms), then the control in Eq. (4) will drive the tracking errors asymptotically to zero.

Once the validity of the guidance algorithms and maneuvers is proved, it still holds out of the Basilisk environment. Their implementation inside the Basilisk architecture has been the authors' choice for convenience. Being open-source and cross-platform in nature, the Basilisk package can be easily deployed and built in a desktop environment. With the generic flight algorithms and DKE suite provided out of the box, the user can run numerical simulations like the integrated scenarios presented in this paper. Cols Margenet et al. [14] explain in detail some of the features that make Basilisk a very attractive tool for flight software development (user-friendly environment, speed of execution, hardware-in-the-loop capability, etc.). In turn, Alcorn et al. [15] provide a review of alternative aerospace commercial-off-the-shelf software packages currently available. Many of these tools, for example Matlab Simulink or LabVIEW, could potentially be used to implement the modular guidance strategy presented in (and the focus of) this paper.

IV. Pointing Base Reference Modules

The first guidance stage consists of modules that generate a base pointing reference $\mathcal{R}_0$. The generated reference adopts the I/O interface defined in the previous sections. The resulting $\mathcal{R}_0$ can be either inertial or noninertial. The common feature of the base modules is that the generated reference does not depend on any prior reference frame. As the name indicates, it conforms the base upon which relative motions can be defined. The base reference modules developed and later integrated to the Basilisk software framework are inertial pointing, Hill orbit pointing, velocity orbit pointing, and constrained two-body pointing. Both the inertial and orbit frame references are widely used and well documented. The novelty lies on the scheme upon which they are architected; through the modular stack and interface definition, base modules can be used in standalone mode and as the base of complex dynamic behaviors. The base reference module described here is a novel kinematic solution to a constrained attitude pointing requirement: two-body pointing. The generated reference adopts the I/O interface defined in the previous sections.

A. Inertial Pointing

The simplest guidance module is inertial pointing. Here, the constant reference frame $\mathcal{R}_0$ is in a fixed general orientation relative to the inertial frame $\mathcal{N}$. The desired inertial orientation is given through an input MRP set $\boldsymbol{\sigma}_{R_0/N}$, whereas the reference frame rates and accelerations are internally set to zero:

$$\boldsymbol{\omega}_{R_0/N} = \dot{\boldsymbol{\omega}}_{R_0/N} = 0 \tag{5}$$

B. Hill and Velocity Pointing

The Hill and velocity base modules are strongly related and therefore presented jointly. Assume that the spacecraft is to align with the orbit Hill frame $\mathcal{R}_0 \equiv \mathcal{H}$: $\{\hat{\boldsymbol{\imath}}_r, \hat{\boldsymbol{\imath}}_\theta, \hat{\boldsymbol{\imath}}_h\}$ or the velocity frame $\mathcal{R}_0 \equiv \mathcal{V}$: $\{\hat{\boldsymbol{\imath}}_n, \hat{\boldsymbol{\imath}}_v, \hat{\boldsymbol{\imath}}_h\}$. Both frames are completely defined by the position and velocity vectors of the spacecraft. The frames $\mathcal{H}$ and $\mathcal{V}$ are each conformed by their own right-handed set of axes where $\hat{\boldsymbol{\imath}}_r$ is the nadir axis pointing radially outward, $\hat{\boldsymbol{\imath}}_v$ is tangent to the orbit and parallel to the velocity vector, $\hat{\boldsymbol{\imath}}_h$ is defined normal to the orbital plane in the direction of the angular momentum, and finally $\hat{\boldsymbol{\imath}}_\theta$ and $\hat{\boldsymbol{\imath}}_n$ complete their respective right-handed triplet.

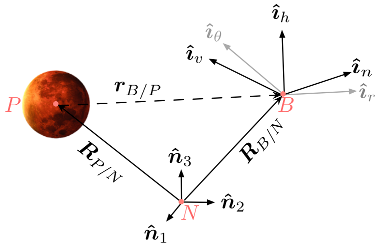


Fig. 7 Illustration of the hill and velocity orbit frames.

Figure 7 illustrates the Hill and velocity frame orientations, each having their origin on the spacecraft location. The inertial frame $\mathcal{N}$: $\{\hat{\boldsymbol{n}}_1, \hat{\boldsymbol{n}}_2, \hat{\boldsymbol{n}}_3\}$ is also depicted. The inertial position and velocity vectors of the spacecraft ($\boldsymbol{R}_{B/N}$, $\boldsymbol{v}_{B/N}$) and the celestial body ($\boldsymbol{R}_{P/N}$, $\boldsymbol{v}_{P/N}$) are the only variables assumed to be known by the module. The relative position of the spacecraft with respect to the planet $\boldsymbol{r}_{B/P}$ and relative velocity $\boldsymbol{v}_{B/P}$ are obtained through

$$\boldsymbol{r}_{B/P} = \boldsymbol{R}_B - \boldsymbol{R}_P \tag{6a}$$

$$\boldsymbol{v}_{B/P} = \boldsymbol{v}_B - \boldsymbol{v}_P \tag{6b}$$

The Hill $\mathcal{H}$ and velocity $\mathcal{V}$ frame orientations with respect to the inertial frame $\mathcal{N}$ are defined through the following direction cosine matrices (DCMs):

$$[HN] = \begin{bmatrix} {}^{\mathcal{H}}\hat{\boldsymbol{\imath}}_r^T \\ {}^{\mathcal{H}}\hat{\boldsymbol{\imath}}_\theta^T \\ {}^{\mathcal{H}}\hat{\boldsymbol{\imath}}_h^T \end{bmatrix} \qquad [VN] = \begin{bmatrix} {}^{\mathcal{V}}\hat{\boldsymbol{\imath}}_n^T \\ {}^{\mathcal{V}}\hat{\boldsymbol{\imath}}_v^T \\ {}^{\mathcal{V}}\hat{\boldsymbol{\imath}}_h^T \end{bmatrix} \tag{7}$$

where the associated unit direction vectors are defined as

$$\hat{\boldsymbol{\imath}}_r = \frac{\boldsymbol{r}_{B/P}}{|\boldsymbol{r}_{B/P}|} \tag{8a}$$

$$\hat{\boldsymbol{\imath}}_v = \frac{\boldsymbol{v}_{B/P}}{|\boldsymbol{v}_{B/P}|} \tag{8b}$$

$$\hat{\boldsymbol{\imath}}_h = \frac{\boldsymbol{r}_{B/P} \times \boldsymbol{v}_{B/P}}{|\boldsymbol{r}_{B/P} \times \boldsymbol{v}_{B/P}|} \tag{8c}$$

$$\hat{\boldsymbol{\imath}}_\theta = \hat{\boldsymbol{\imath}}_h \times \hat{\boldsymbol{\imath}}_r \tag{8d}$$

$$\hat{\boldsymbol{\imath}}_n = \hat{\boldsymbol{\imath}}_h \times \hat{\boldsymbol{\imath}}_v \tag{8e}$$

The corresponding Hill orbit and velocity orbit MRP attitude sets are derived from their corresponding DCM [8]:

$$[HN] \rightarrow \boldsymbol{\sigma}_{H/N} \equiv \boldsymbol{\sigma}_{R_0/N} \quad \text{or} \quad [VN] \rightarrow \boldsymbol{\sigma}_{V/N} \equiv \boldsymbol{\sigma}_{R_0/N} \tag{9}$$

Next, the reference frame rates and accelerations are determined. In the case of the Hill frame $\mathcal{H}$, the angular rate of the reference is that of the orbital motion:

$$\boldsymbol{\omega}_{R_0/N} = \boldsymbol{\omega}_{H/N} = \dot{f}\hat{\boldsymbol{\imath}}_h \tag{10a}$$

$$\dot{\boldsymbol{\omega}}_{R_0/N} = \dot{\boldsymbol{\omega}}_{H/N} = \ddot{f}\hat{\boldsymbol{\imath}}_h \tag{10b}$$

where f is the true anomaly angle, whose variation is expressed through the following general astrodynamics relation:

$$\dot{f} = \frac{\boldsymbol{r}_{B/P} \times \boldsymbol{v}_{B/P}}{\boldsymbol{r}_{B/P} \cdot \boldsymbol{r}_{B/P}} \tag{11a}$$

$$\ddot{f} = -2\frac{\boldsymbol{v}_{B/P} \cdot \hat{\boldsymbol{\imath}}_r}{|\boldsymbol{r}_{B/P}|}\dot{f} \tag{11b}$$

The velocity frame $\mathcal{V}$ orientation differs from the Hill frame orientation by a single-axis rotation of angle $-\beta$ in the orbital plane about $\hat{\boldsymbol{\imath}}_h$. The DCM that maps from $\mathcal{H}$ to $\mathcal{V}$ is expressed in terms of the flight-path angle β or the classical set of orbital elements as [8]

$$[VH] = \begin{bmatrix} \cos\beta & -\sin\beta & 0 \\ \sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{1+e\cos f}{\sqrt{1+e^2+2e\cos f}} & -\frac{e\sin f}{\sqrt{1+e^2+2e\cos f}} & 0 \\ \frac{e\sin f}{\sqrt{1+e^2+2e\cos f}} & \frac{1+e\cos f}{\sqrt{1+e^2+2e\cos f}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{12}$$

The inertial angular rate and acceleration of the velocity frame $\mathcal{V}$ are obtained by

$$\boldsymbol{\omega}_{V/N} = \boldsymbol{\omega}_{V/H} + \boldsymbol{\omega}_{H/N} \tag{13a}$$

$$\dot{\boldsymbol{\omega}}_{V/N} = \dot{\boldsymbol{\omega}}_{V/H} + \dot{\boldsymbol{\omega}}_{H/N} \tag{13b}$$

where

$$\boldsymbol{\omega}_{V/H} = -\dot{\beta}\hat{\boldsymbol{\imath}}_h \tag{14a}$$

$$\dot{\boldsymbol{\omega}}_{V/H} = -\ddot{\beta}\hat{\boldsymbol{\imath}}_h \tag{14b}$$

An analytical expression for β is derived from Eq. (12), whose inertial time derivatives are

$$\dot{\beta} = \frac{e(e+\cos f)}{1+e^2+2e\cos f}\dot{f}$$
$$\ddot{\beta} = \frac{e(e+\cos f)}{1+e^2+2e\cos f}\ddot{f} + \frac{e(e^2-1)\sin f}{(1+e^2+2e\cos f)^2}\dot{f}^2$$

The velocity-frame base pointing module rates and accelerations are thus defined as

$$\boldsymbol{\omega}_{R_0/N} = \boldsymbol{\omega}_{V/N} = (\dot{f} - \dot{\beta})\hat{\boldsymbol{\imath}}_h \tag{16}$$

$$\dot{\boldsymbol{\omega}}_{R_0/N} = \dot{\boldsymbol{\omega}}_{V/N} = (\ddot{f} - \ddot{\beta})\hat{\boldsymbol{\imath}}_h \tag{17}$$

C. Constrained Two-Body Pointing

A base reference frame $\mathcal{R}_0$: $\{\hat{\boldsymbol{r}}_1, \hat{\boldsymbol{r}}_2, \hat{\boldsymbol{r}}_3\}$ is generated that tracks the center of a primary celestial target (e.g., pointing the communication antenna at the Earth) and tries to align the second reference axis toward a second celestial body as best as possible (e.g., pointing a solar panel normal axis to the sun). Two attitude conditions in a three-dimensional space compose an overdetermined problem. Hence, a TRIAD-like approach is used such that the main constraint is always prioritized over the secondary one [16].

Figure 8 depicts the desired reference frame $\mathcal{R}_0$: $\{\hat{\boldsymbol{r}}_1, \hat{\boldsymbol{r}}_1, \hat{\boldsymbol{r}}_2\}$ and the inertial frame $\mathcal{N}$: $\{\hat{\boldsymbol{n}}_1, \hat{\boldsymbol{n}}_1, \hat{\boldsymbol{n}}_2\}$. The $\mathcal{R}$ frame has its origin in the spacecraft body $\mathcal{B}$. The points $\mathcal{P}_1$ and $\mathcal{P}_2$ are the primary and secondary celestial targets, respectively. The initial states known by the module are the position vector $\boldsymbol{R}_{i/N}$, velocity vector $\dot{\boldsymbol{R}}_{i/N}$, and acceleration vector $\ddot{\boldsymbol{R}}_{i/N}$ of the spacecraft (where the subindex $i = \mathcal{B}$) and of the celestial bodies (where the subindex $i = \mathcal{P}_1, \mathcal{P}_2$) with respect to the inertial frame.

The normal vector $\boldsymbol{R}_n$ is perpendicular to the plane defined by the two celestial targets and the spacecraft location, and it is expressed as

$$\boldsymbol{R}_n = \boldsymbol{R}_{P_1/B} \times \boldsymbol{R}_{P_2/B} \tag{18}$$

The desired base reference frame $\mathcal{R}$ is computed such that the first unit base vector $\hat{\boldsymbol{r}}_1$ points to the primary target. The third base vector $\hat{\boldsymbol{r}}_3$ is aligned with $\boldsymbol{R}_n$, whereas the last base vector $\hat{\boldsymbol{r}}_2$ completes the right-handed triplet:

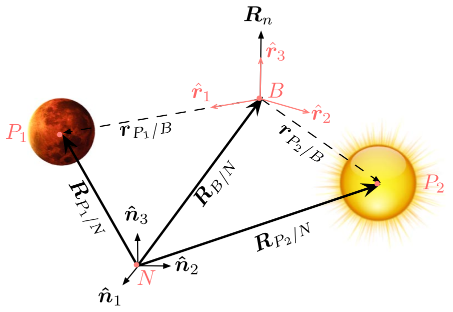


Fig. 8 Illustration of a constrained celestial two-body pointing scenario.

$$\hat{\boldsymbol{r}}_1 = \frac{\boldsymbol{R}_{P_1/B}}{|\boldsymbol{R}_{P_1/B}|} \tag{19a}$$

$$\hat{\boldsymbol{r}}_3 = \frac{\boldsymbol{R}_n}{|\boldsymbol{R}_n|} \tag{19b}$$

$$\hat{\boldsymbol{r}}_2 = \hat{\boldsymbol{r}}_3 \times \hat{\boldsymbol{r}}_1 \tag{19c}$$

The celestial object locations relative to the spacecraft are found through

$$\boldsymbol{R}_{P_1/B} = \boldsymbol{R}_{P_1/N} - \boldsymbol{R}_{B/N} \tag{20a}$$

$$\boldsymbol{R}_{P_2/B} = \boldsymbol{R}_{P_2/N} - \boldsymbol{R}_{B/N} \tag{20b}$$

This setup aligns $\hat{\boldsymbol{r}}_2$ as closely as possible with $\boldsymbol{R}_{P_2/B}$. However, for general configurations, it is not possible to align these vectors perfectly while meeting the primary constraint at the same time.

The DCM that maps from the inertial frame $\mathcal{N}$ to the desired reference frame $\mathcal{R}$ is given by

$$[RN] = \begin{bmatrix} {}^{\mathcal{N}}\hat{\boldsymbol{r}}_1^T \\ {}^{\mathcal{N}}\hat{\boldsymbol{r}}_2^T \\ {}^{\mathcal{N}}\hat{\boldsymbol{r}}_3^T \end{bmatrix} \tag{21}$$

The desired MRP attitude set $\boldsymbol{\sigma}_{R_0/N}$ is derived from $[R_0N]$ [8].

The angular velocity $\boldsymbol{\omega}_{R_0/N}$ and acceleration $\dot{\boldsymbol{\omega}}_{R_0/N}$ still have to be computed. To do so, the time derivatives of the reference base vectors are needed. The following expressions are found for the first inertial time derivatives:

$$\dot{\hat{\boldsymbol{r}}}_1 = ([I_{3\times3}] - \hat{\boldsymbol{r}}_1\hat{\boldsymbol{r}}_1^T)\frac{\dot{\boldsymbol{R}}_{P_1/B}}{|\boldsymbol{R}_{P_1/B}|} \tag{22a}$$

$$\dot{\hat{\boldsymbol{r}}}_3 = ([I_{3\times3}] - \hat{\boldsymbol{r}}_3\hat{\boldsymbol{r}}_3^T)\frac{\dot{\boldsymbol{R}}_n}{|\boldsymbol{R}_n|} \tag{22b}$$

$$\dot{\hat{\boldsymbol{r}}}_2 = \dot{\hat{\boldsymbol{r}}}_3 \times \boldsymbol{r}_1 + \boldsymbol{r}_n \times \dot{\hat{\boldsymbol{r}}}_3 \tag{22c}$$

where the first and second inertial time derivatives of $\boldsymbol{R}_n$ are

$$\dot{\boldsymbol{R}}_n = \dot{\boldsymbol{R}}_{P_1/B} \times \boldsymbol{R}_{P_2/B} + \boldsymbol{R}_{P_1/B} \times \dot{\boldsymbol{R}}_{P_2/B} \tag{23}$$

$$\ddot{\boldsymbol{R}}_n = \ddot{\boldsymbol{R}}_{P_1/B} \times \boldsymbol{R}_{P_2/B} + 2\dot{\boldsymbol{R}}_{P_1/B} \times \dot{\boldsymbol{R}}_{P_2/B} + \boldsymbol{R}_{P_1/B} \times \ddot{\boldsymbol{R}}_{P_2/B} \tag{24}$$

Differentiating the unit vectors in Eqs. (22) yields

$$\ddot{\hat{\boldsymbol{r}}}_1 = \frac{1}{|\boldsymbol{R}_{P_1/B}|}(([I_{3\times3}] - \hat{\boldsymbol{r}}_1\hat{\boldsymbol{r}}_1^T)\ddot{\boldsymbol{R}}_{P_1/B} - 2\dot{\hat{\boldsymbol{r}}}_1(\hat{\boldsymbol{r}}_1 \cdot \dot{\boldsymbol{R}}_{P_1/B}) - \hat{\boldsymbol{r}}_1(\dot{\hat{\boldsymbol{r}}}_1 \cdot \dot{\boldsymbol{R}}_{P_1/B})) \tag{25a}$$

$$\ddot{\hat{\boldsymbol{r}}}_3 = \frac{1}{|\boldsymbol{R}_n|}(([I_{3\times3}] - \hat{\boldsymbol{r}}_3\hat{\boldsymbol{r}}_3^T)\ddot{\boldsymbol{R}}_n - 2\dot{\hat{\boldsymbol{r}}}_3(\hat{\boldsymbol{r}}_3 \cdot \dot{\boldsymbol{R}}_n) - \hat{\boldsymbol{r}}_3(\dot{\hat{\boldsymbol{r}}}_3 \cdot \dot{\boldsymbol{R}}_n)) \tag{25b}$$

$$\ddot{\hat{\boldsymbol{r}}}_2 = \ddot{\hat{\boldsymbol{r}}}_3 \times \boldsymbol{r}_1 + \boldsymbol{r}_n \times \ddot{\hat{\boldsymbol{r}}}_3 + 2\dot{\hat{\boldsymbol{r}}}_3 \cdot \dot{\hat{\boldsymbol{r}}}_1 \tag{25c}$$

The reference angular rate is expressed in reference frame components as

$${}^{R_0}\boldsymbol{\omega}_{R_0/N} = \begin{bmatrix} \boldsymbol{\omega}_{R_0/N} \cdot \hat{\boldsymbol{r}}_1 \\ \boldsymbol{\omega}_{R_0/N} \cdot \hat{\boldsymbol{r}}_2 \\ \boldsymbol{\omega}_{R_0/N} \cdot \hat{\boldsymbol{r}}_3 \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{r}}_3 \cdot \dot{\hat{\boldsymbol{r}}}_2 \\ \hat{\boldsymbol{r}}_1 \cdot \dot{\hat{\boldsymbol{r}}}_3 \\ \hat{\boldsymbol{r}}_2 \cdot \dot{\hat{\boldsymbol{r}}}_1 \end{bmatrix} \tag{26}$$

Taking the inertial derivative of Eq. (26) yields

$$
{}^{R_0}\dot{\boldsymbol{\omega}}_{R_0/N} = \begin{bmatrix} \dot{\boldsymbol{\omega}}_{R_0/N} \cdot \hat{\boldsymbol{r}}_1 \\ \dot{\boldsymbol{\omega}}_{R_0/N} \cdot \hat{\boldsymbol{r}}_2 \\ \dot{\boldsymbol{\omega}}_{R_0/N} \cdot \hat{\boldsymbol{r}}_3 \end{bmatrix} = \begin{bmatrix} \dot{\hat{\boldsymbol{r}}}_3 \cdot \dot{\hat{\boldsymbol{r}}}_2 + \hat{\boldsymbol{r}}_3 \cdot \ddot{\hat{\boldsymbol{r}}}_2 - \boldsymbol{\omega}_{R/N} \cdot \dot{\hat{\boldsymbol{r}}}_1 \\ \dot{\hat{\boldsymbol{r}}}_1 \cdot \dot{\hat{\boldsymbol{r}}}_3 + \hat{\boldsymbol{r}}_1 \cdot \ddot{\hat{\boldsymbol{r}}}_3 - \boldsymbol{\omega}_{R/N} \cdot \dot{\hat{\boldsymbol{r}}}_2 \\ \dot{\hat{\boldsymbol{r}}}_2 \cdot \dot{\hat{\boldsymbol{r}}}_1 + \hat{\boldsymbol{r}}_2 \cdot \ddot{\hat{\boldsymbol{r}}}_1 - \boldsymbol{\omega}_{R/N} \cdot \dot{\hat{\boldsymbol{r}}}_3 \end{bmatrix} \tag{27}
$$

Finally the angular rates and acceleration in Eqs. (26) and (27) are mapped to the inertial frame $\mathcal{N}$

$$
{}^{\mathcal{N}}\boldsymbol{\omega}_{R_0/N} = [R_0N]^{T\,R_0}\boldsymbol{\omega}_{R_0/N} \tag{28a}
$$

$$
{}^{\mathcal{N}}\dot{\boldsymbol{\omega}}_{R_0/N} = [R_0N]^{T\,R_0}\dot{\boldsymbol{\omega}}_{R_0/N} \tag{28b}
$$

Now, all the variables in the output structure of the module, $\mathcal{R}_0 = \{\boldsymbol{\sigma}_{R_0/N}, {}^{\mathcal{N}}\boldsymbol{\omega}_{R_0/N}, {}^{\mathcal{N}}\dot{\boldsymbol{\omega}}_{R_0/N}\}$, have been computed.

V. Dynamic Reference Modules

Next, modules that define a dynamic reference motion are considered. The generated motions are superimposed on top of a base reference frame $\mathcal{R}_0$, which can be any of the aforementioned ones (inertial pointing, Hill-orbit pointing velocity-orbit pointing, celestial two-body pointing, etc.). For instance, the same dynamic scanning pattern could be painted on the inertial frame (i.e., superposition of inertial-pointing module and a scanning module) or, rather, on the orbited planet (i.e., superposition of nadir-pointing module and a scanning module). Another example would be a single-axis spinning case; the spinning motion could be defined about a fixed inertial axis or, also, about a time-varying axis such as that of an orbit. The fixed-axis spinning motion could be achieved either with the inertial three-dimensional (3-D) spinning module [17] or the 3-2-1 Euler angle rotation module. The 3-2-1 Euler angle rotation module is applied to achieve not only spinning motions but also different scanning patterns. It is worth mentioning that other relative dynamic motions can be achieved by implementing mission-specific modules. The only requirement is to maintain the defined I/O module interface.

A. 3-2-1 Euler Angle Rotation Module: General Algorithm

Using the 3-2-1 Euler rotation module, complex dynamic motions relative to the base reference $\mathcal{R}_0$ can be achieved through elegantly simple constant Euler rates. Within the scope of the present work, two different kinds of maneuvers are shown using the same 3-2-1 Euler module: orbit axis spinning and axis scanning. Despite each type of maneuver being completely distinct, both relative spinning and scanning are achieved simply by initializing the 3-2-1 Euler module with different setup parameters. The tuning of the setup leaves even more room to obtain a wide range of raster patterns for the scanning case.

Euler angles are often avoided in guidance algorithms because of their mathematical singularities. Yet, in the presented guidance instances, the advantages of the Euler sets are exploited in a robust and safe manner that is free of numerical issues.

The module shown uses a 3-2-1 sequence, but naturally other Euler angle sequences could be considered. Interestingly, by staging up to three modules of the 3-2-1 sequence, the 12 Euler combinations can be achieved. This strategy is particularly convenient in terms of unit testing; having a single module and using it multiple times implies verifying only one source-code algorithm rather than multiple ones.

Next, the mathematical aspects of the 3-2-1 Euler angle rotation module are developed. An initial 3-2-1 Euler angle orientation $\boldsymbol{\theta}_{R_1/R0}$ and a constant set of rates $\dot{\boldsymbol{\theta}}_{R_1/R0}$ are defined as inputs to this dynamic module:

$$
\boldsymbol{\theta}_{R_1/R_0}\colon \{\psi_0, \theta_0, \phi_0\}
$$

$$
\dot{\boldsymbol{\theta}}_{R_1/R_0}\colon \{\dot{\psi}, \dot{\theta}, \dot{\phi}\}
$$

Because the module considers Euler rates that are constant, the associated differential kinematic equations are integrable. The current Euler angles are thus expressed as

$$
\psi(t) = \psi_0 + \dot{\psi}\delta t \tag{29a}
$$

$$
\theta(t) = \theta_0 + \dot{\theta}\delta t \tag{29b}
$$

$$
\phi(t) = \phi_0 + \dot{\phi}\delta t \tag{29c}
$$

The final time-varying attitude $[R_1N]$ is evaluated by adding the dynamic attitude $[RR_0]$ onto the base reference attitude $[R_0N]$:

$$
[R_1N] = [R_1R_0(\psi(t), \theta(t), \phi(t))][R_0N(\boldsymbol{\sigma}_{R_0/N})] \tag{30}
$$

The output orientation of this dynamic module is obtained by converting the $[RN]$ matrix into the equivalent MRP coordinates: $[R_1N] \to \boldsymbol{\sigma}_{R_1/N}$.

In the following, the angular velocity vector $\boldsymbol{\omega}_{R_1/N}$ and its derivative $\dot{\boldsymbol{\omega}}_{R_1/N}$ are developed. The final angular velocity vector is defined as

$$
\boldsymbol{\omega}_{R_1/N} = \boldsymbol{\omega}_{R_1/R_0} + \boldsymbol{\omega}_{R_0/N} \tag{31}
$$

where $\boldsymbol{\omega}_{R_0/N}(t)$ is the angular velocity of the base reference frame, input to the module. The matrix ${}^{R_1}\boldsymbol{\omega}_{R_1/R_0}$ is obtained from the 3-2-1 Euler angle differential kinematic equations:

$$
{}^{R_1}\boldsymbol{\omega}_{R_1/R_0} = \begin{bmatrix} -\sin\theta & 0 & 1 \\ \sin\phi\cos\theta & \cos\phi & 0 \\ \cos\phi\cos\theta & -\sin\phi & 0 \end{bmatrix} \begin{bmatrix} \dot\psi \\ \dot\theta \\ \dot\phi \end{bmatrix} \tag{32}
$$

Substituting Eq. (32) into Eq. (31) yields the output state $\boldsymbol{\omega}_{R_1/N}$.

At this point, it is important to follow up on the previous statement that the 3-2-1 Euler module is free of singularities. As a matter of fact, each set of Euler angles has a geometric singularity where two angles are not uniquely defined. Such geometric singularities result in mathematical singularities in the differential kinematic equations. These singularities appear always when mapping from angular body rates to Euler angle rates. In the present development, though, the inverse mapping takes place (i.e., the module maps Euler rates to angular velocity), and such a process is completely singularity-free. Any geometric singularity would be inevitably reflected in Eq. (32), which is proved not to be the case.

The angular acceleration of the $\mathcal{R}_1$ reference is computed by taking the inertial derivative of Eq. (31). The short-hand dot notation is used to denote an inertial derivative of a vector:

$$
\dot{\boldsymbol{\omega}}_{R_1/N} = \dot{\boldsymbol{\omega}}_{R_1/R_0} + \dot{\boldsymbol{\omega}}_{R_0/N} \tag{33}
$$

The inertial derivative of the spinning vector $\boldsymbol{\omega}_{R_1/R_0}$ is evaluated using the transport theorem [8] to be

$$
\dot{\boldsymbol{\omega}}_{R_1/R_0} = \frac{{}^{\mathcal{N}}\mathrm{d}}{\mathrm{d}t}(\boldsymbol{\omega}_{R_1/R_0}) = \frac{{}^{R_1}\mathrm{d}}{\mathrm{d}t}(\boldsymbol{\omega}_{R_1/R_0}) + \boldsymbol{\omega}_{R_1/N} \times \boldsymbol{\omega}_{R_1/R_0} \tag{34}
$$

Let us now develop the right-hand side of Eq (34). Making use of the fact that the defined Euler rates are constant, the following expression is obtained for the $\mathcal{R}_1$-frame derivative of $\boldsymbol{\omega}_{R_1/R_0}$:

$$
\frac{{}^{R_1}\mathrm{d}}{\mathrm{d}t}(\boldsymbol{\omega}_{R_1/R_0}) = \begin{bmatrix} -\dot\theta\dot\psi\cos\theta \\ (\dot\phi\cos\phi\cos\theta - \dot\theta\sin\phi\sin\theta)\dot\psi - \dot\phi\dot\theta\sin\phi \\ -(\dot\phi\sin\phi\cos\theta + \dot\theta\cos\phi\cos\theta)\dot\psi - \dot\phi\dot\theta\cos\phi \end{bmatrix} \tag{35}
$$

Simplifying Eq. (34) and joining this equation with Eq. (33), the $\mathcal{N}$-frame angular acceleration is eventually obtained

$$
\dot{\boldsymbol{\omega}}_{R_1/N} = \frac{{}^{R_1}\mathrm{d}}{\mathrm{d}t}(\boldsymbol{\omega}_{R_1/R_0}) + \boldsymbol{\omega}_{R_0/N} \times \boldsymbol{\omega}_{R_1/R_0} + \dot{\boldsymbol{\omega}}_{R_0/N} \tag{36}
$$

B. 3-2-1 Euler Angle Rotation Module: Application to Guidance Maneuvers

1. Principal Orbit Axis Spinning

Spinning about a primary orbit axis is achieved by applying a single axis rotation of the 3-2-1 Euler angle set onto the orbit base reference. Both Hill and velocity frames are considered as base reference orbit frames. The different orbit spinning modes are achieved using the following 3-2-1 Euler angle rates:

$$
\begin{aligned}
\{\dot\psi, 0, 0\} &= \text{Spinning motion about the third axis of the orbit frame (i.e., angular momentum direction } \hat{\boldsymbol{\imath}}_h) \\
\{0, \dot\theta, 0\} &= \text{Spinning motion about the second axis of the orbit frame} \\
\{0, 0, \dot\phi\} &= \text{Spinning motion about the first axis of the orbit frame}
\end{aligned}
$$

Figure 9 illustrates a spinning motion of rate $\dot\phi$ about the nadir axis of the Hill frame $\hat{\boldsymbol{\imath}}_r$. In this case, the base orbit frame is the Hill one:

$$
\mathcal{R}_0\colon \{\hat{\boldsymbol{r}}_{0,1}, \hat{\boldsymbol{r}}_{0,2}, \hat{\boldsymbol{r}}_{0,3}\} \equiv \mathcal{H}\colon \{\hat{\boldsymbol{\imath}}_r, \hat{\boldsymbol{\imath}}_\theta, \hat{\boldsymbol{\imath}}_h\}
$$

and the Euler angle rates of the 3-2-1 set are defined as

$$
\dot{\boldsymbol{\theta}}_{R_1/R0} = \{0, 0, \dot\phi\}
$$

2. Axis Scanning

Scanning rasters are performed through the consecutive request of Euler angle offsets and rates. The use of the 3-2-1 Euler set in particular facilitates a wide range of scanning patterns. Figure 10 illustrates three sample patterns. The scanning is performed relative to the time-varying base reference frame $\mathcal{R}_0\colon \{\hat{\boldsymbol{r}}_{0,1}, \hat{\boldsymbol{r}}_{0,2}, \hat{\boldsymbol{r}}_{0,3}\}$, which keeps track of the orbited celestial body.

The continuous lines in salmon color of Fig. 10 correspond to the nominal raster lines of scientific interest. Figures 10a and 10b are constituted of multiple raster lines. The transition between one raster and the following one is indicated through the dashed salmon lines. Each nominal line is

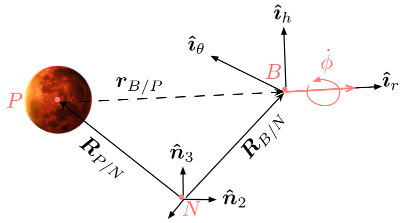
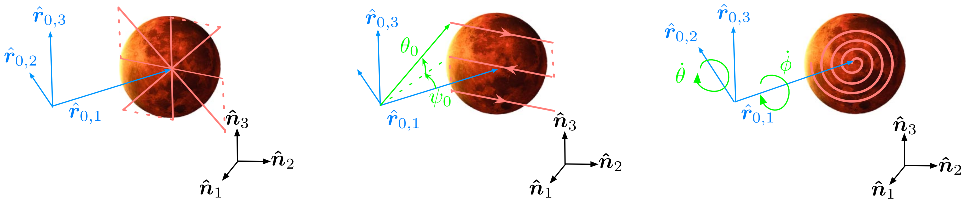


Fig. 9 Hill frame nadir axis spinning.



a) Rectangular Raster Maneuver **b) Asterisk Raster Maneuver** **c) Spiral Maneuver**

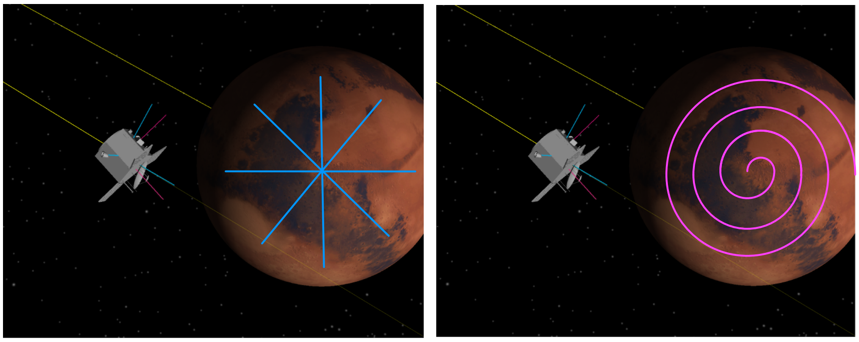
Fig. 10 Scientific scanning patterns.

defined by a specific initial angle set $\boldsymbol{\theta}_{R_1/R0}$: $\{\psi_0, \theta_0, \phi_0\}$ and rate $\dot{\boldsymbol{\theta}}_{R_1/R0}$: $\{\dot{\psi}, \dot{\theta}, \dot{\phi}\}$. The raster lines of the rectangular scanning pattern in Fig. 10a are achieved through constant θ_0 and ψ_0 offset values and a linearly time-varying ψ angle. The raster lines of the asterisk scanning pattern in Fig. 10b are achieved through initial offsets in both ψ and θ and allowing both angles to have constant rates. The spiral scanning pattern in Fig. 10c is accomplished by staging two 3-2-1 Euler modules to simulate a 1-2 Euler sequence. The first Euler module is initialized with a constant $\dot{\phi}$ rate, which determines how quickly the boresight axis progresses through the helix. The second Euler module is initialized with a constant $\dot{\theta}$ rate, which defines how quickly the helix opens.

To complete complex scanning patterns that combine multiple rasters, as in Figs. 10a and 10b, the Euler angle rotation module is complemented with a higher-level module called raster manager. The function of the raster manager is to request the appropriate Euler angle offsets and rates at the configured raster times. The role of the raster manager module is showcased in Sec. VII.

VI. Tracking Error Module

Let $\mathcal{B}$ be the principal body frame of the spacecraft. It might be the case that the spacecraft-fixed frame that we want to control and drive toward the desired reference $\mathcal{R}$ is not the primary $\mathcal{B}$-frame. Instead, we might be interested in controlling an alternate body-fixed frame $\mathcal{B}_c$ such as that of a sensor or a control actuator. Figure 11 shows two guidance scanning maneuvers. In Fig. 11a, one axis of the blue control frame $\mathcal{B}_{c,\text{blue}}$ is to perform the scanning. In Fig. 11b, and one axis of the magenta frame $\mathcal{B}_{c,\text{magenta}}$ acts as the scanning boresight. Imagine that neither the blue control frame $\mathcal{B}_{c,\text{blue}}$ nor the magenta control frame $\mathcal{B}_{c,\text{magenta}}$ coincides with the primary body frame $\mathcal{B}$. Let us recall that the output of the guidance block is designed to be $\{\boldsymbol{\sigma}_{B/R}, {}^{B}\boldsymbol{\omega}_{B/N}, {}^{B}\boldsymbol{\omega}_{R/N}, {}^{B}\dot{\boldsymbol{\omega}}_{R/N}\}$. Such output interface is illustrated in Fig. 12 and makes use of the main body frame $\mathcal{B}$. Theoretically, the guidance output could be expressed in the specific control frame $\mathcal{B}_c$ for each case. Nevertheless, it would then be necessary to keep track of the relationship between the principal body frame and the control body frame $[B_c B]$ in multiple modules inside the guidance-control sequence. Recalling the control law formulation presented in Eq. (1), all the vector components as well as the inertia tensor would then need to be mapped to this new body-fixed control frame. Although Eq. (1) corresponds in particular to an MRP feedback control strategy, any other control law would still require the mapping of both tensor and vectorial variables. However, this mapping involving multiple transformations is not a convenient operation to implement digitally.



a) Blue control frame $\mathcal{B}_{c,\text{blue}}$ performing an asterisk scanning **b) Magenta control frame $\mathcal{B}_{c,\text{magenta}}$ performing a spiral scanning**

Fig. 11 Body-fixed control frames.

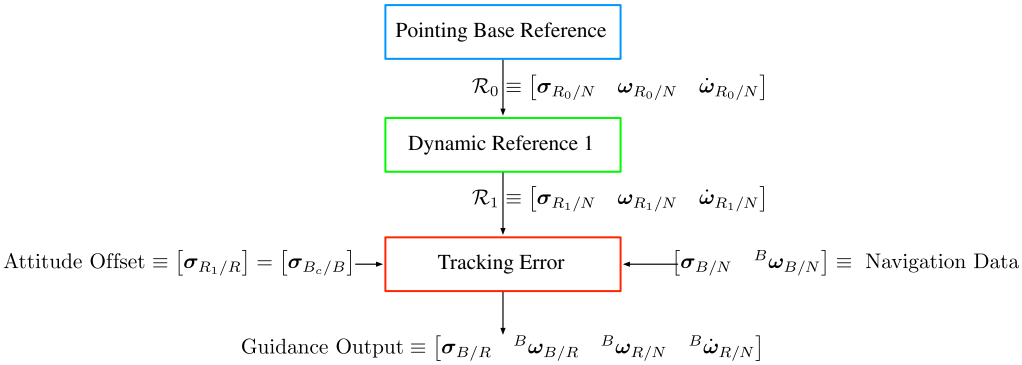

Pointing Base Reference
$\mathcal{R}_0 \equiv [\boldsymbol{\sigma}_{R_0/N} \quad \boldsymbol{\omega}_{R_0/N} \quad \dot{\boldsymbol{\omega}}_{R_0/N}]$
Dynamic Reference 1
$\mathcal{R}_1 \equiv [\boldsymbol{\sigma}_{R_1/N} \quad \boldsymbol{\omega}_{R_1/N} \quad \dot{\boldsymbol{\omega}}_{R_1/N}]$
Attitude Offset $\equiv [\boldsymbol{\sigma}_{R_1/R}] = [\boldsymbol{\sigma}_{B_c/B}]$
Tracking Error
$[\boldsymbol{\sigma}_{B/N} \quad {}^{B}\boldsymbol{\omega}_{B/N}] \equiv$ Navigation Data
Guidance Output $\equiv [\boldsymbol{\sigma}_{B/R} \quad {}^{B}\boldsymbol{\omega}_{B/R} \quad {}^{B}\boldsymbol{\omega}_{R/N} \quad {}^{B}\dot{\boldsymbol{\omega}}_{R/N}]$

Fig. 12 Illustration of the guidance and controls stacks.

In the following, a strategic approach is proposed that allows the same body-fixed frame $\mathcal{B}$ to be used in any case, while adjusting only the reference orientation and therefore reducing the number of transformations into one. Let us recall that the base and dynamic modules compute part of the final reference motion. Let us label the output of this stack of modules (i.e., the input to the tacking error module) as $\mathcal{R}_i$, where $i = 0$ if only a base pointing module is used in the stack; $i = 1$ if a base pointing module and one dynamic module are used; $i = 2$ if a base pointing and two relative dynamic modules are used as in Fig. 3. This use of nomenclature is depicted in Fig. 12, with $\mathcal{R}_i = \mathcal{R}_1$ in this example.

In the general case where the body-fixed frame to be controlled is not the spacecraft's primary frame $\mathcal{B}$, the control should drive the control body-fixed frame $\mathcal{B}_c$ to the desired reference $\mathcal{R}_i$. The control body-frame $\mathcal{B}_c$ could correspond to any body-fixed frame that differs from $\mathcal{B}$ by a constant angular offset $[B_cB]$. Thus, as $\mathcal{B}_c \to \mathcal{R}_0$, then $\mathcal{B} \to \mathcal{R}$ if $\mathcal{R}$ is defined such that

$$[B_cB] = [R_iR] \tag{37}$$

The I/O interface of the tracking error module appears in Fig. 12. The attitude tracking error module receives the main body frame orientation $[BN(\boldsymbol{\sigma}_{B/N})]$ from navigation. Further, the constant body to control-body DCM $[B_cB]$ is set within the module as a configurable parameter. Again, the input reference frame DCM $[R_iN(\boldsymbol{\sigma}_{R_i/N})]$ is computed using the base pointing and dynamic modules. Inside the tracking error module, the orientation of the final reference frame $\mathcal{R}$ is defined relative to the inertial frame by subtracting the relative body-frame orientation from the input reference orientation:

$$[RN] = [RR_i][R_iN] = [B_cB]^T[R_iN] \tag{38}$$

With the discussed generalization, it is possible to correct any sensor or component frame orientation using the same classical tracking error algorithms and control laws that deal with the main body frame. Therefore, the need of creating particular code for the guidance and control of different spacecraft-fixed frames is overcome.

Recall the guidance output $\{\boldsymbol{\sigma}_{B/R}, {}^{\mathcal{B}}\boldsymbol{\omega}_{B/N}, {}^{\mathcal{B}}\boldsymbol{\omega}_{R/N}, {}^{\mathcal{B}}\dot{\boldsymbol{\omega}}_{R/N}\}$. Next, the attitude tracking error $\boldsymbol{\sigma}_{B/R}$ and angular velocity error ${}^{\mathcal{B}}\boldsymbol{\omega}_{B/R}$ are derived. The attitude error of $\mathcal{B}$ relative to the $\mathcal{R}$ reference frame is

$$[BR(\boldsymbol{\sigma}_{B/R})] = [BN(\boldsymbol{\sigma}_{B/N})][RN(\boldsymbol{\sigma}_{R/N})]^T \tag{39}$$

The MRP set $\boldsymbol{\sigma}_{B/R}$ is readily obtained from $[BR]$ [8]. The rate tracking error is defined as

$$\boldsymbol{\omega}_{B/R} = \boldsymbol{\omega}_{B/N} - \boldsymbol{\omega}_{R/N} = \boldsymbol{\omega}_{B/N} - \boldsymbol{\omega}_{R_i/N} \tag{40}$$

because $[B_cB]$ is a constant attitude correction, and therefore $\boldsymbol{\omega}_{R/R_i} = 0$.

VII. Numerical Simulations

A. Numerical Scenario Common Setup

The rigid-body equations of motion in Eq. (1) are numerically simulated in Basilisk to validate and illustrate the performance of the presented guidance modules. The orientation is controlled through a set of four RWs. The dynamic states are integrated using a Runge–Kutta 4 scheme running at 10 Hz. The MRP feedback control in Eq. (4) is used to drive the spacecraft orientation toward the desired reference motion using an update rate of 2 Hz. The simulated navigation data $\boldsymbol{\sigma}_{B/N}$ and $\boldsymbol{\omega}_{B/N}$ are without any sensor corruptions to better illustrate that the control law does achieve asymptotic tracking of the reference motion. Each simulation has a maneuver duration of 9600 s ($\approx$2.7 h). The spacecraft is simulated to be orbiting Mars with the orbital parameters in Table 1 and the control-related parameters in Table 2. The spacecraft is flying through the periapsis region of a highly eccentric orbit. Consequently, there is a considerably amount of variability on the spacecraft orbit rates. In the first control scenario, asymptotically tracking this orbital motion emphasizes the challenges of properly evaluating acceleration and rates. With the chosen

Table 1 Initial orbital elements

Parameter	Value
Semimajor axis	7471.618($\approx$2.2R_{Mars}) km
Eccentricity	0.4
Inclination	0.0 deg
Longitude of ascendant node	0.0 deg
Argument of perigee	0.0 deg
True anomaly	270.0 deg

Table 2 Control and spacecraft parameters

Parameter	Value
Attitude error gain K	2.531 $(\text{kg}\cdot\text{m}^2)/\text{s}$
Rate error gain P	45.0 $(\text{kg}\cdot\text{m}^2)/\text{s}$
$\hat{\boldsymbol{g}}_{s_1}$	$[-0.5, 0.5, -(\sqrt{2}/2)]$
$\hat{\boldsymbol{g}}_{s_2}$	$[0.5, 0.5, -(\sqrt{2}/2)]$
$\hat{\boldsymbol{g}}_{s_3}$	$[0.5, -0.5, -(\sqrt{2}/2)]$
$\hat{\boldsymbol{g}}_{s_4}$	$[-0.5, -0.5, -(\sqrt{2}/2)]$
J_{s_i}	0.1591549 $\text{kg}\cdot\text{m}^2$
I_1, I_2	700 $\text{kg}\cdot\text{m}^2$
I_3	800.0 $\text{kg}\cdot\text{m}^2$

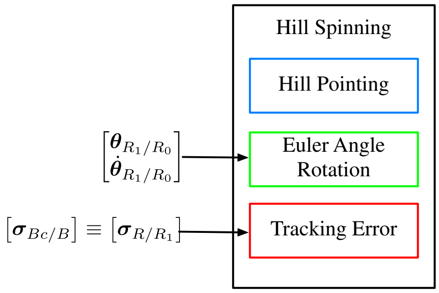

Hill Spinning
Hill Pointing
$\begin{bmatrix}\boldsymbol{\theta}_{R_1/R_0}\\ \dot{\boldsymbol{\theta}}_{R_1/R_0}\end{bmatrix}$
Euler Angle Rotation
$[\boldsymbol{\sigma}_{Bc/B}] \equiv [\boldsymbol{\sigma}_{R/R_1}]$
Tracking Error

Fig. 13 Hill spinning stack.

proportional gain P, the control time decay constant [8] is $\tau \approx 30$ s. The K gain is computed to yield a critically damped system. This choice of gains provides a nonaggressive control response. Although initial requests of large torques are unavoidable (i.e., detumbling phase), operating with nonsaturated reaction wheels is guaranteed for the rest of the simulation time (i.e., tracking phase).

B. Orbit Axis Rotation

The first simulation is nadir spinning. Initially, the spacecraft lines up with the Hill frame. Then, an attitude offset is applied by performing a 180 deg rotation about the third body axis $\hat{\boldsymbol{b}}_3$. The resulting frame points the first principal axis of the spacecraft body $\hat{\boldsymbol{b}}_1$ toward the planet (i.e., radially inward instead of outward). Hence, $\hat{\boldsymbol{b}}_1$ becomes antiparallel to the first axis of the Hill frame $\hat{\boldsymbol{\imath}}_r$. Superimposed on this base reference motion there is a spinning motion about $\hat{\boldsymbol{\imath}}_r$. With a proper tracking, $\hat{\boldsymbol{b}}_1$ remains consequently fixed, whereas $\hat{\boldsymbol{b}}_2$ and $\hat{\boldsymbol{b}}_3$ rotate in the local-horizontal orbit plane. Figure 13 shows the guidance stack to achieve the suggested spinning maneuver about the nadir direction of the orbit. The configuration parameter of the tracking error module $\boldsymbol{\sigma}_{B_c/B}$ is the control body-frame correction

$$\boldsymbol{\sigma}_{Bc/B} = [0.0, 0.0, 1.0] \tag{41}$$

which yields the 180 deg offset of the first and second axes in the orbit plane. The configuration data of the Euler angle rotation module are provided in Table 3. A constant Euler rate about the first axis of the reference $\hat{\boldsymbol{r}}_1$ is defined.

Figure 14 shows plots of the reference attitude sets generated at each stage. In particular, Fig. 14a illustrates the time-varying Hill-frame orientation, denoted $\mathcal{R}_0$ in this scenario. The rapid orientation change is due to the spacecraft flying through periapses on a highly elliptic orbit. The constant Euler angle rates in Fig. 14b show that only $\dot{\phi} \neq 0$. Cascading these two reference frames finally yields the complex reference frame motion shown in Fig. 14c.

Figure 15 displays the final attitude and rate tracking errors as well as the applied control torque. The body frame aligns asymptotically with the desired reference frame, illustrating the proper kinematic addition of the base and dynamic reference frames as well as the correction of the principal body frame to control body frame. The control torque in Fig. 15c is capped to a maximum RW torque of $0.2\ \mathrm{N}\cdot\mathrm{m}$. Note that the variable u_i corresponds to the torque of each reaction wheel. It is shown that saturation only takes place at the very beginning of the maneuver. Performing a nadir axis spinning is an unnatural, nonequilibrium motion, and therefore it is expected that the reference torques in Fig. 15c do not converge to zero.

C. Asterisk Scanning Maneuver

The second simulation performs a challenging scanning maneuver to achieve an asterisk pattern, such as the one illustrated in Fig. 10b, across an inertial reference. The guidance stack for the multiraster scanning motion is shown in Fig. 16. Note that, in this simulation, a raster manager module is attached to the Euler rotation module. The raster manager commands a sequence of Euler angle offsets and rates at the configured raster times. Each raster lasts for 1600 s (≈0.45 h). The complete pattern is achieved with a total of $N = 4$ raster commands. Additionally, the spacecraft is smoothly driven back to the starting scanning point.

The inertial pointing module and attitude tracking error module are initialized with the following parameters:

$$\boldsymbol{\sigma}_{R_0/N} = [0.0, 0.0, 0.0] \tag{42a}$$

$$\boldsymbol{\sigma}_{Bc/B} = [0.0, 0.0, 0.0] \tag{42b}$$

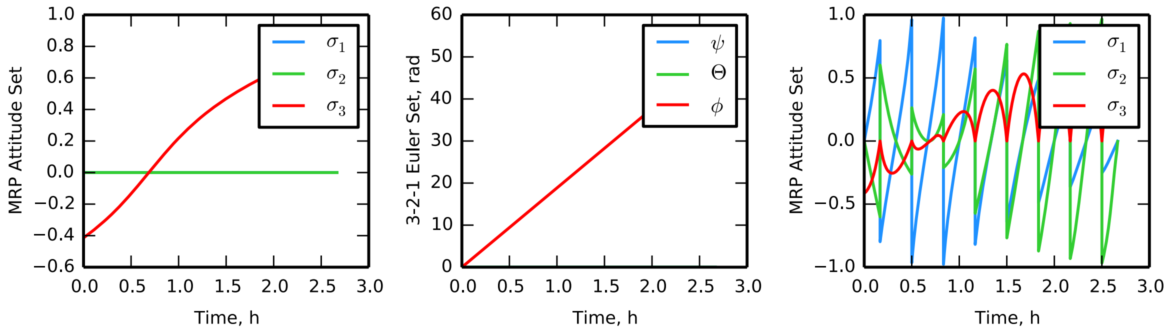
Thus, the requested inertial attitude is that of the global inertial frame with $\mathcal{R}_0 \equiv \mathcal{N}$. The control body frame coincides with the principal body frame through $\mathcal{B}_c \equiv \mathcal{B}$, or equivalently $[B_c B] = [I_{3\times3}]$.

The configuration parameters of the raster manager are provided in Table 4, where α and $\dot{\alpha}$ are defined as

$$\alpha = 8.0\ \text{deg} \tag{43}$$

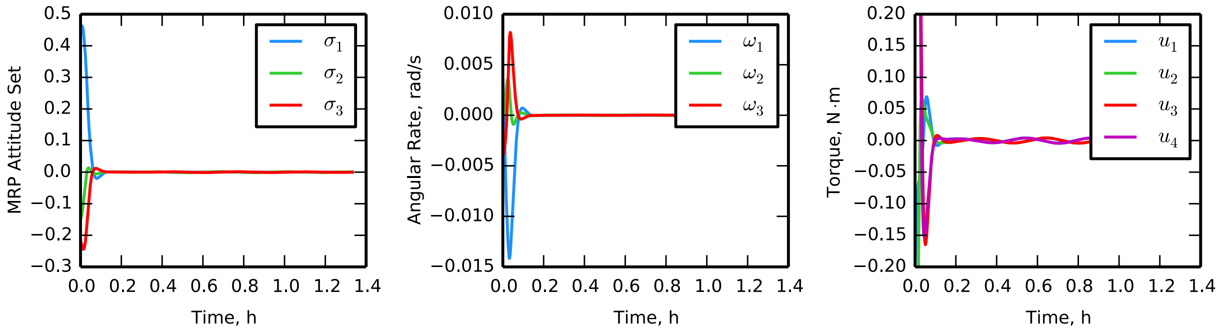
Table 3 Euler angle rotation configuration data

Parameter	Value	Description
$\boldsymbol{\theta}_{R_1/R_0}$: $\{\psi, \theta, \phi\}$	[0.0, 0.0, 0.0] deg / s	Initial 3-2-1 set of Euler angles relative to the base $\mathcal{R}_0$
$\dot{\boldsymbol{\theta}}_{R_1/R_0}$: $\{\dot{\psi}, \dot{\theta}, \dot{\phi}\}$	[0.0, 0.0, 0.3] deg / s	Desired 3-2-1 set of Euler rates relative to the base $\mathcal{R}_0$



a) Base reference MRP $\sigma_{R0/N}$ b) 3-2-1 Euler set $\theta_{R0/R}$ c) Desired reference MRP $\sigma_{R/N}$

Fig. 14 Generated attitude sets.



a) Attitude tracking error $\sigma_{B/R}$ b) Rate tracking error $\omega_{B/R}$ c) Applied control torque u

Fig. 15 Tracking errors and control torque.

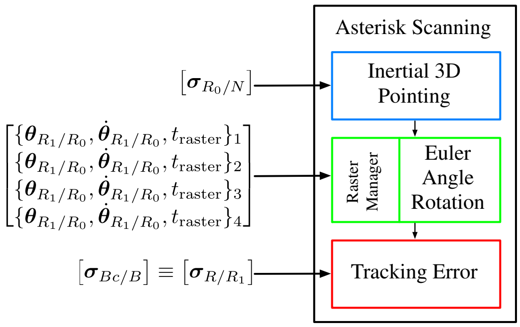

Asterisk Scanning
Inertial 3D Pointing
Raster Manager
Euler Angle Rotation
Tracking Error
$[\sigma_{R_0/N}]$
$\begin{bmatrix}\{\theta_{R_1/R_0}, \dot{\theta}_{R_1/R_0}, t_{\text{raster}}\}_1 \\ \{\theta_{R_1/R_0}, \dot{\theta}_{R_1/R_0}, t_{\text{raster}}\}_2 \\ \{\theta_{R_1/R_0}, \dot{\theta}_{R_1/R_0}, t_{\text{raster}}\}_3 \\ \{\theta_{R_1/R_0}, \dot{\theta}_{R_1/R_0}, t_{\text{raster}}\}_4\end{bmatrix}$
$[\sigma_{Bc/B}] \equiv [\sigma_{R/R_1}]$

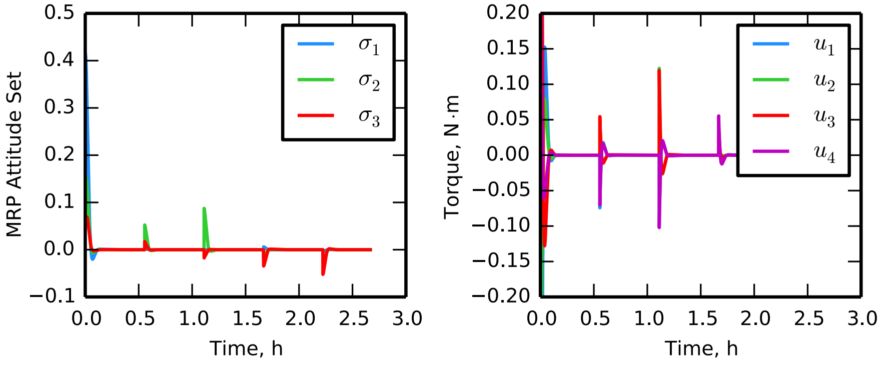
Fig. 16 Inertial asterisk scanning stack.

$$\dot{\alpha} = \frac{2\alpha}{t_{\text{raster}}} \tag{44}$$

Figure 17 shows plots of the attitude tracking error and the reaction wheel control torques during the asterisk scanning. Note that the peaks in the plots correspond to the command of a new raster line and are an expected behavior. At these times, the spacecraft asymptotically transitions from its current raster line to the next.

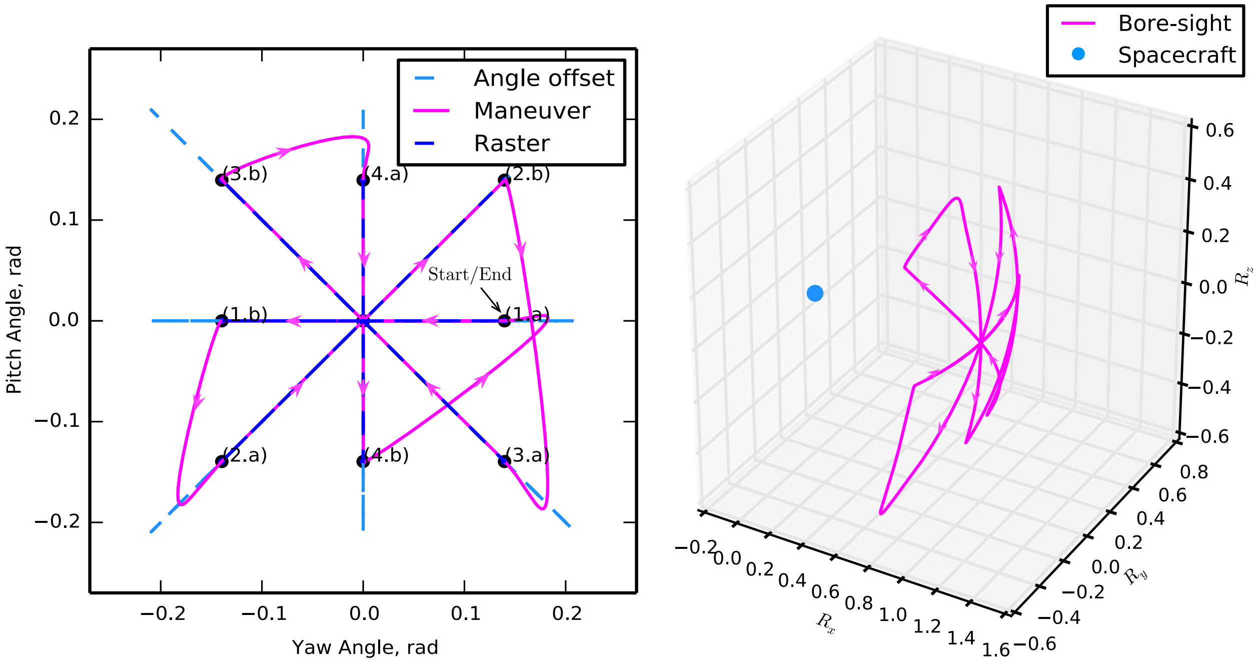
Table 4 Raster manager configuration data

Parameter	Value	Description
$\theta_{R_1/R_0}: \{\psi, \theta, \phi\}_1$	$[\alpha, 0.0, 0.0]$ deg	Initial 3-2-1 Euler angle set of the first raster
$\dot{\theta}_{R_1/R_0}: \{\dot{\psi}, \dot{\theta}, \dot{\phi}\}_1$	$[-\dot{\alpha}, 0.0, 0.0]$ deg/s	3-2-1 Euler angle rates of the first raster
$\theta_{R_1/R_0}: \{\psi, \theta, \phi\}_2$	$[-\alpha, -\alpha, 0.0]$ deg	Initial 3-2-1 Euler angle set of the first raster
$\dot{\theta}_{R_1/R_0}: \{\dot{\psi}, \dot{\theta}, \dot{\phi}\}_2$	$[\dot{\alpha}, \dot{\alpha}, 0.0]$ deg/s	3-2-1 Euler angle rates of the second raster
$\theta_{R_1/R_0}: \{\psi, \theta, \phi\}_3$	$[\alpha, -\alpha, 0.0]$ deg	Initial 3-2-1 Euler angle set of the third raster
$\dot{\theta}_{R_1/R_0}: \{\dot{\psi}, \dot{\theta}, \dot{\phi}\}_3$	$[-\dot{\alpha}, \dot{\alpha}, 0.0]$ deg/s	3-2-1 Euler angle rates of the third raster
$\theta_{R_1/R_0}: \{\psi, \theta, \phi\}_4$	$[0.0, \alpha, 0.0]$ deg	Initial 3-2-1 Euler angle set of the forth raster
$\dot{\theta}_{R_1/R_0}: \{\dot{\psi}, \dot{\theta}, \dot{\phi}\}_4$	$[0.0, -\dot{\alpha}, 0.0]$ deg/s	3-2-1 Euler angle rates of the forth raster
$t_{\text{raster},i}$	1600 s	Time duration of each commanded raster maneuver



a) Attitude tracking error $\sigma_{B/R}$ b) Applied control torque u

Fig. 17 Tracking error and control torque.



a) Nominal Rasters vs. Achieved Maneuver for α = 8 deg b) Achieved 3D Bore-sight Pointing for α = 24 deg

Fig. 18 Achieved asterisk scanning pattern.

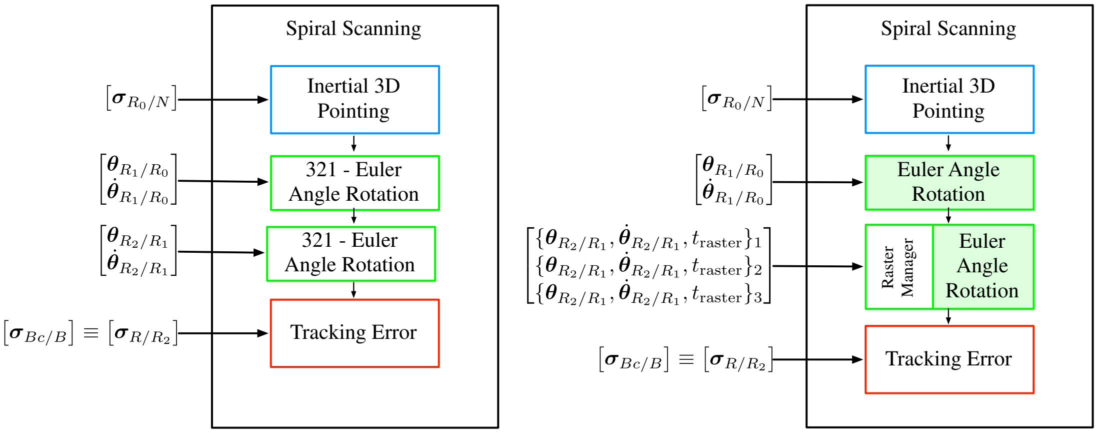
Figure 18a compares the desired nominal raster lines (dark blue lines) with the actual yaw–pitch angles scanned in the maneuver (magenta lines). To get into the desired rasters on time, a small angle offset needs to be commanded, which is also depicted (light blue lines). The choice of the offset α_{offset} is a tradeoff with the picked control gains K, P. In this case, an offset of $\alpha_{\text{offset}} = 0.5\alpha$ is found to work well for any value of nominal scanning angle α. This additional angular offset demands, of course, small adjustments on the raster times. The numbered points at the beginning and end of the scan lines indicate the order of the four rasters. Points labeled with letter a correspond to the beginning of the nominal raster line, whereas points labeled with letter b correspond to the end. Once a raster is completed, a new command takes the spacecraft from its current point b to a new initial attitude and rate corresponding to the next point a (transition). Note that, from point $4(b)$, the spacecraft is driven back to the initial point $1(a)$, where the scanning maneuver started. Figure 18b shows the three-dimensional view of the spacecraft's boresight pointing. Cartesian coordinates of the boresight pointing are represented in a scanning maneuver of nominal angle $\alpha = 24$ deg. The single blue dot corresponds to the position of the spacecraft, from which the scanned pattern is projected on a unit sphere.

D. Spiral Scanning Maneuver

The last numerical simulation performs a scanning maneuver that draws a spiral pattern. This scenario serves to showcase the concatenation of multiple dynamic references on top of a base frame. As a matter of fact, a spiral motion can be easily defined through a 1-2 sequence of constant Euler rates. Such a scanning pattern is represented in Fig. 10c, where the boresight axis is the first one of the base frame $\boldsymbol{r}_1$.

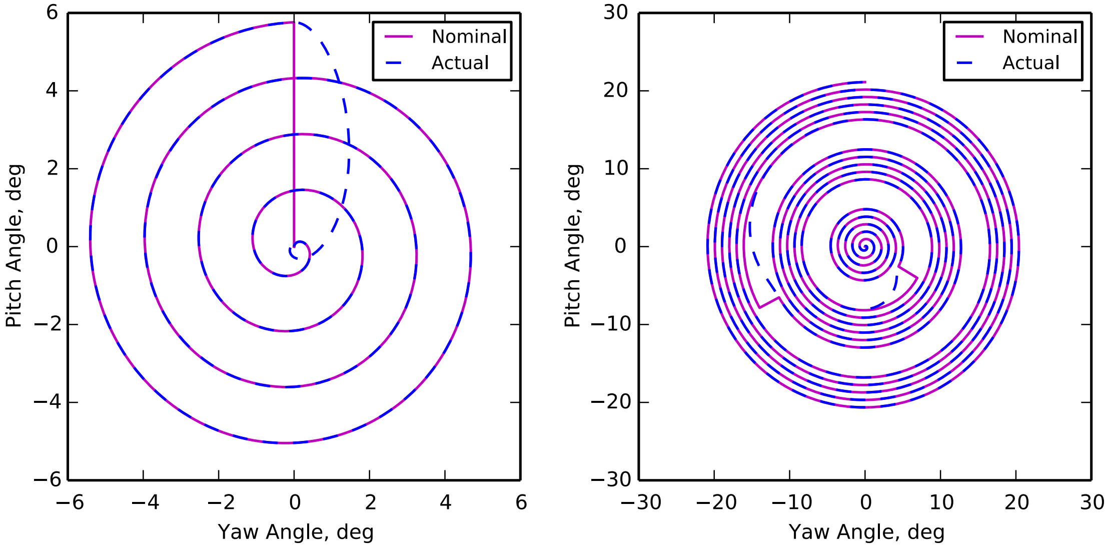
The guidance stack that conforms the spiral maneuver in the numerical simulation is depicted in Fig. 19a. The base reference used is inertial 3-D pointing for the sake of simplicity in analyzing the resulting plots. To simulate a 1-2 Euler sequence, two Euler rotation modules are staged as dynamic references. The inertial pointing module and attitude tracking error module are initialized with the parameter values in Eq. (42).

After the helix pattern is completed, the spacecraft's boresight is smoothly driven back to the starting scanning point. Figure 20a illustrates the nominal helix (magenta line) and the actual angles scanned by the boresight (dark blue line).


Spiral Scanning
Inertial 3D Pointing
321 - Euler Angle Rotation
321 - Euler Angle Rotation
Tracking Error
$[\boldsymbol{\sigma}_{R_0/N}]$
$\begin{bmatrix}\boldsymbol{\theta}_{R_1/R_0}\\ \dot{\boldsymbol{\theta}}_{R_1/R_0}\end{bmatrix}$
$\begin{bmatrix}\boldsymbol{\theta}_{R_2/R_1}\\ \dot{\boldsymbol{\theta}}_{R_2/R_1}\end{bmatrix}$
$[\boldsymbol{\sigma}_{Bc/B}] \equiv [\boldsymbol{\sigma}_{R/R_2}]$
Spiral Scanning
Inertial 3D Pointing
Euler Angle Rotation
Raster Manager
Euler Angle Rotation
Tracking Error
$[\boldsymbol{\sigma}_{R_0/N}]$
$\begin{bmatrix}\boldsymbol{\theta}_{R_1/R_0}\\ \dot{\boldsymbol{\theta}}_{R_1/R_0}\end{bmatrix}$
$\begin{bmatrix}\{\boldsymbol{\theta}_{R_2/R_1}, \dot{\boldsymbol{\theta}}_{R_2/R_1}, t_{\text{raster}}\}_1\\ \{\boldsymbol{\theta}_{R_2/R_1}, \dot{\boldsymbol{\theta}}_{R_2/R_1}, t_{\text{raster}}\}_2\\ \{\boldsymbol{\theta}_{R_2/R_1}, \dot{\boldsymbol{\theta}}_{R_2/R_1}, t_{\text{raster}}\}_3\end{bmatrix}$
$[\boldsymbol{\sigma}_{Bc/B}] \equiv [\boldsymbol{\sigma}_{R/R_2}]$

a) Single Spiral Stack **b) Triple Spiral Stack**

Fig. 19 Spiral scanning stacks.

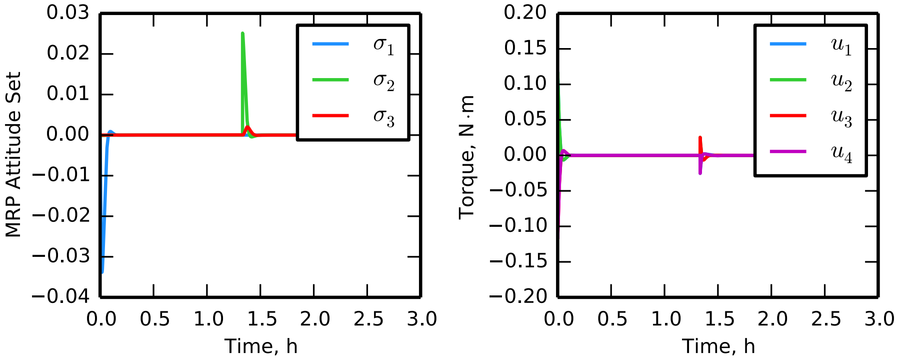


a) Single Spiral Scanning **b) Triple Spiral Scanning**

Fig. 20 Spiral scanning patterns.

Figure 21 shows plots of the attitude tracking error and the reaction wheel control torques during the spiral scanning. The peak in the plots at the middle of the simulation corresponds to the transition from the completion of the spiral pattern to the comeback at the origin, where the boresight remains afterward. It is shown that the spiral reference motion is perfectly tracked with small wheel torques.

An extension of the presented spiral maneuver is displayed in Fig. 20b. Here, the raster manager module is attached to the second Euler rotation module to sequence three spiral motions during the same maneuver, as seen in the module stack of Fig. 19b. Figure 20b compares the nominal helices with the actual scanned angles. This demonstration is a clear example of the high flexibility that a modular architecture provides.



a) Attitude tracking error $\boldsymbol{\sigma}_{B/R}$ **b) Applied control torque $\boldsymbol{u}$**

Fig. 21 Tracking error and control torque.

VIII. Results

The previous numerical simulations showcase guidance maneuvers conformed by superposition of multiple module behaviors. Therefore, it makes sense to consider them as integrated test scenarios. The success criteria for these integrated tests is the asymptotic convergence of the tracking errors to zero. Convergence would not be eventually achieved if there was mathematical conflict between the generated attitude, angular rate, and angular acceleration that come out of every reference generation module, or if there was an error in the superposition approach.

At this point, it is relevant to highlight how modularity has helped in effectively isolating errors and identifying the cause of emerging behaviors during the construction of the integrated guidance algorithms. A key factor during the first stages of debugging is that replacing single modules in the stack is extremely fast and easy to perform, and it does not require modifying the source code. The simple fact of replacing a software component sometimes exposes global changes that are not intuitive in the first place, revealing clues on how to solve the problem early on. In this sense, the Lego-like modular architecture allows even a trial-and-error approach to become systematic. In more involved debugging efforts, having independent modules that have been validated, both individually and in different integrated cases, provides a high degree of confidence on where to look for the roots of emerging problems. In contrast, when there is a resilient problem with a monolithic algorithm, it is very human to turn it upside-down entirely.

In terms of cost reduction, the molecularity of the architecture allows direct reuse of functional algorithm code while avoiding duplication. Because the same module can potentially be used in very different guidance contexts of a mission, there is an overall reduction of lines of code to maintain and to validate. The spiral scanning pattern on a planet provides a great example of module reusability and mission cross-compatibility; the modules Hill pointing and Euler rotation are properly combined to generate the very specific case of scanning a particular planet in a spiral motion. At the same time, when considered individually, these modules are applicable to many other mission profiles. In contrast, its monolithic counterpart would be a very specific algorithm likely inapplicable in another mission context as it is.

In terms of risk reduction, the key factor is that the different profiles are not covered with a single monolithic algorithm full of switch and if statements, which make validation difficult, are extremely fragile, and scale badly. Instead, at every mission stage, only the modules of interest are loaded and reconfigured appropriately, while message connections are redefined as convenient. By building integrated guidance algorithms in a module Lego-like approach, transparency of the integrated algorithm becomes de facto.

Within the scope of the present work, the cost in power and CPU usage of an integrated layered maneuver over its monolithic counterpart have not been compared. The reason is that there are some factors intrinsic to the Basilisk architecture that can impact speed and performance (e.g., message passing instead of single script execution), and it becomes difficult to specifically discern them from the guidance implementation aspects. Nonetheless, some tradeoffs between the proposed layering approach and a custom guidance algorithm can still be discussed.

With the proposed modular scheme, the addition of a base or dynamic module in the guidance stack always involves generating a new reference motion. Therefore, the number of transformations applied unavoidably scales with the superposition of additional reference generation (either base or dynamic) modules. In the context of low-power missions, like those involving CubeSats or small-satellite form factors, the computational load of a module-layering implementation could be at first sight a consideration. That being said, the numerical simulations have shown that complex behaviors can be achieved with the superposition of only two reference modules when staged and initialized strategically. Further, as explained previously, the use of the tracking error module guarantees the capability of driving any generic control body-fixed frame without having to map all the tensors and vectors in the control law equation. Therefore, multiple transformations are avoided at this stage. A final consideration is that the presented guidance algorithms have been implemented directly in C, using optimized libraries for vector and matrix math. That being said, if the requirements of a mission demand an extremely high-efficient guidance algorithm, then a single custom algorithm, handwritten directly in C and optimized, would clearly be more suitable than its layered counterpart.

IX. Conclusions

A novel reference generation architecture is presented where complex guidance is achieved through a set of atomic reference frame behaviors. A fundamental aspect is that the proposed guidance scheme is developed in a general modular manner. Mission-specific needs can then be met throughout a reliable and systematic process that involves 1) the setting of specified input parameters, and 2) the arrangement of particular modules to create a guidance stack. Two types of generated references are distinguished. First, the base reference frame is generated using only the modules' configuration data and, if applicable, the spacecraft's translational state. Next, dynamic reference frame behaviors are superimposed to yield spinning or scanning maneuvers relative to the base reference frame. Alternate body-fixed frame alignments are accounted for during the tracking error computation. A key advantage of this approach is that each guidance module can be tested and verified individually. Because complex guidance functionality is achieved through a combination of tested core modules, the integrated validation of the onboard flight software is simplified.

References

[1] Lefferts, E. J., Markley, F. L., and Shuster, M. D., "Kalman Filtering for Spacecraft Attitude Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 5, No. 5, 1981, pp. 417–429.
doi:10.2514/3.56190

[2] Crassidis, J. L., and Markley, F. L., "Survey of Nonlinear Attitude Estimation Methods," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 1, 2007, pp. 12–28.
doi:10.2514/1.22452

[3] Crassidis, J. L., and Markley, F. L., "Unscented Filtering for Spacecraft Attitude Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 4, July–Aug. 2003, pp. 536–542.
doi:10.2514/2.5102

[4] Karlgaard, C. D., and Schaub, H., "Nonsingular Attitude Filtering Using Modified Rodrigues Parameters," *Journal of the Astronautical Sciences*, Vol. 57, No. 4, Oct.–Dec. 2010, pp. 777–791.
doi:10.1007/BF03321529

[5] O'Keefe, S. A., and Schaub, H., "Shadow Set Considerations for Modified Rodrigues Parameter Attitude Filtering," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 6, 2014, pp. 2030–2035.
doi:10.2514/1.G000405

[6] Bayard, D. S., "High-Precision Three-Axis Pointing and Control," *Encyclopedia of Aerospace Engineering*, edited by R. Blockey, and W. Shyy, Wiley, Chichester, England, U.K., 2010.
doi:10.1002/9780470686652.eae300

[7] Tsiotras, P., Shen, H., and Hall, C. D., "Satellite Attitude Control and Power Tracking with Energy/Momentum Wheels," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 1, 2001, pp. 23–34.
doi:10.2514/2.4705

[8] Schaub, H., and Junkins, J. L., *Analytical Mechanics of Space Systems*, 3rd ed., AIAA Education Series, AIAA, Reston, VA, 2014, Chaps. 3 and 8.

[9] Wie, B., *Space Vehicle Dynamics and Control*, 2nd ed., AIAA Education Series, AIAA, Reston, VA, 2008, Chap. 7.

[10] de Lafontaine, J., Buijs, J., Vuilleumier, P., den Braembussche, P. V., and Mellab, K., "Development of the PROBA Attitude Control and Navigation Software," *Proceedings of the 4th ESA International Conference, Spacecraft Guidance, Navigation and Control Systems*, edited by B. Schürmann, European Space Agency Paper ESA SP-425, Paris, 2000.

[11] Schulte, P. Z., and Spencer, D. A., "Development of an Integrated Spacecraft Guidance, Navigation, and Control Subsystem for Automated Proximity Operations," *Acta Astronautica*, Vol. 118, Oct. 2014, pp. 168–186.

[12] Wiener, T. F., "Theoretical Analysis of Gimballess Inertial Reference Equipment Using Delta-Modulated Instruments," Ph.D. Dissertation, Dept. of Aeronautics and Astronautics, Massachusetts Inst. of Technology, Cambridge, MA, March 1962.

[13] Tsiotras, P., and Longuski, J. M., "A New Parameterization of the Attitude Kinematics," *Journal of the Astronautical Sciences*, Vol. 43, No. 3, 1996, pp. 243–262.

[14] Cols Margenet, M., Schaub, H., and Piggott, S., "Modular Platform for Hardware-in-the-Loop Testing of Autonomous Flight Algorithms," *Proceedings of the International Symposium on Space Flight Dynamics*, Paper ISTS-2017-t-16, Matsuyama-Ehime, Japan, June 2017.

[15] Alcorn, J., Schaub, H., Piggott, S., and Kubitschek, D., "Simulating Attitude Actuation Options Using the Basilisk Astrodynamics Software Architecture," *Proceedings of the 67th International Astronautical Congress*, Paper IAC-16-C1.1.4, Guadalajara, Mexico, Sept. 2016.

[16] Tanygin, S., "The Many TRIAD Algorithms," *Proceedings of the AAS/AIAA Spaceflight Mechanics Meeting*, American Astronautical Soc. Paper 07-104, Springfield, VA, Feb. 2007.

[17] Cols Margenet, M., Schaub, H., and Piggott, S., "Modular Attitude Guidance Development Using the Basilisk Software Framework," *AIAA SPACE 2016*, AIAA Paper 2016-5538, Sept. 2016.

G. P. Brat
Associate Editor