(Preprint) AAS 25-013

TWO-AXIS GIMBAL SIMULATION OVERVIEW FOR THE EMIRATES MISSION TO THE ASTEROID BELT

Leah Kiner* Cody Allard† and Hanspeter Schaub‡

The Emirates Mission to the Asteroid Belt (EMA) will utilize two Solar Electric Propulsion (SEP) thrusters to traverse the solar system and visit a total of 7 main belt asteroids. Mounted to a two-axis gimballed platform actuated by two stepper motors, each SEP thruster is articulated individually to align with a commanded inertial thrust direction. This paper gives a general overview of the SEP gimbal functionality and outlines the flight software and simulation C++ algorithms developed to command and actuate the two-axis gimbals. Expanding upon previous work in prescribed motion dynamics, this work develops modular flight software and kinematic profiler modules to command and actuate the stepper motors and gimbals. The previously developed prescribed motion dynamics formulation is used to simulate the contribution of the profiled gimbal motion to the spacecraft system dynamics. Simulation results are presented to demonstrate effective implementation of the developed algorithms. The gimbal thrust axis is shown to converge rapidly to the commanded reference direction.

INTRODUCTION

The Emirates Mission to the Asteroid Belt (EMA) is the first main belt multi-asteroid tour mission. Set to launch in 2028, the Mohammad Bin Rashid (MBR) explorer spacecraft will flyby 6 asteroids and rendezvous with the seventh.[1] Depicted in Fig. (1), the explorer will use two Solar Electric Propulsion (SEP) thrusters to navigate through space efficiently. Several other interplanetary space missions have successfully demonstrated the use of SEP technology including NASA's 1998 Deep Space 1[2] mission, NASA's 2001 Dawn[3] mission, and NASA's Psyche[4] mission launched in 2023. JAXA's 2003 Hayabusa 1 and 2014 Hayabusa 2 missions also used SEP technology, along with ESA's 2003 Smart-1 mission.[5–7]

The SEP thrusters are actuated during the EMA SEP Point State,[1] where they are used to thrust along an inertial reference direction while keeping the explorer's solar arrays pointed at the Sun. Further, the SEP thrusters are responsible for autonomously managing the buildup of angular momentum on the spacecraft due to external perturbations while also absorbing the accumulated momentum along the thrust axis due to the thruster swirl torque. An in-depth discussion of the EMA SEP Point state is provided in Reference 1.

*Graduate Research Assistant, Ann and H.J. Smead Department of Aerospace Engineering Sciences, University of Colorado Boulder, Colorado Center For Astrodynamics Research, Boulder, CO, 80303 USA. leah.kiner@colorado.edu

†Guidance, Navigation and Control Engineer, Laboratory for Atmospheric and Space Physics, University of Colorado Boulder, Boulder, CO, 80303 USA

‡Professor and Department Chair, Schaden Leadership Chair, Ann and H.J. Smead Department of Aerospace Engineering Sciences, University of Colorado, Boulder, 431 UCB, Colorado Center for Astrodynamics Research, Boulder, CO, 80309. AAS Fellow, AIAA Fellow.

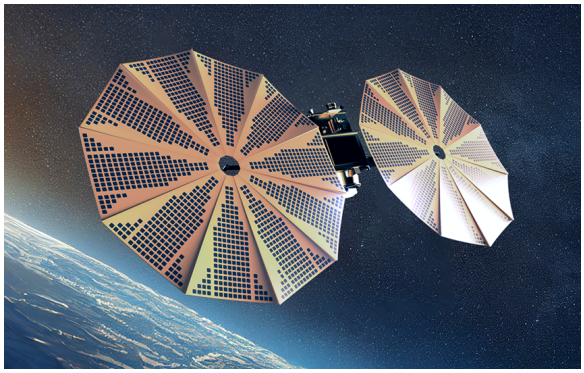


Figure 1: EMA explorer concept.*

A two-axis gimballed platform actuates each SEP thruster on the EMA spacecraft. The tip-tilt motion of each platform is controlled by two stepper motors actuators. A substantial body of literature has been devoted to control and simulation techniques for gimballed platforms. Newtonian/Eularian mechanics,[8–14] Lagrangian mechanics,[9,15–17] and Kane's method[18] have all been used to model the dynamics of gimballed platforms. These previous studies derive the gimbal equations of motion specifically for each actuator application, rather than employing a general set of equations. Existing work has applied specific DC motor torque equations to model gimbals driven by servo motors.[8,11,12,18] Many different control strategies have been applied for gimbal commanding such as sliding mode control[8,15,17] and variations of PID controllers.[14,18–21]

The novelty of this work lies in its use of the previously developed prescribed motion dynamics formulation,[22–24] rather than deriving specific equations of motion for the EMA two-axis gimbal platforms. The formulation requires that the actuated component motion is purely prescribed relative to the spacecraft hub; meaning the hub motion does not impact the actuator motion, however not vice versa. Thus, there is no need to integrate any equations of motion for these components and instead their motion can be modeled independently from the dynamics of the other spacecraft components using a kinematic acceleration profile. This approach preserves the generality of the gimbal equations, allowing them to be applied to simulate any type of articulated spacecraft component. The validity of this assumption must be analyzed based on the bandwidth and performance of the servo controller.

The NASA Cassini spacecraft features several prescribed components, including an articulated main engine, a one-degree-of-freedom (DOF) probe relay antenna, and a high-precision two-DOF scanning platform.[25] Similarly, the Space Station Remote Manipulator System (SSRMS), also known as the "Canadarm" is composed of eight links and seven active joints. Its motion can be controlled either via pre-programmed trajectories or by commanding individual joint angles.[25,26] Reference 23 provides a general overview and simplified simulation development for the two-axis gimbals modeled in this work and serves as the background for the developed C++ algorithms.

The structure of this paper is as follows. The first section gives a general overview of the geometry and functionality of the SEP gimbals and includes a discussion of the parameters and assumptions used for modeling their actuation. The following section outlines the flight software and simulation algorithms developed to command and actuate the gimbals. Next, simulation results for the EMA

*https://space.gov.ae/en/projects-and-initiatives/space-exploration/emirates-mission-to-the-asteroid-belt

SEP Point state are presented demonstrating correct implementation of the C++ algorithms. The concluding remarks are offered in the final section of this paper.

TWO-AXIS SEP GIMBAL ACTUATOR OVERVIEW

This section gives an overview of the two SEP gimbals on the EMA spacecraft. First, the geometry and orientation of the gimbals relative to the spacecraft are discussed. The method by which the gimbals actuate is next described in detail. Note that *general* terms and parameters are used throughout this overview in order to illustrate the applicability of this work to simulate any two-axis gimbal actuator. Parameters specific to the EMA gimbals are given as needed for context throughout this overview.

First, five reference frames are required to fully describe the orientation of the SEP gimbals relative to the EMA spacecraft. The spacecraft body frame is designated by $\mathcal{B} : \{B, \hat{\boldsymbol{b}}_1, \hat{\boldsymbol{b}}_2, \hat{\boldsymbol{b}}_3\}$. This frame is fixed to the rigid spacecraft hub at the origin point B. Next, the actuation of each gimbal is described relative to a hub-fixed mount frame denoted $\mathcal{M}_i : \{M_i, \hat{\boldsymbol{m}}_{i,1}, \hat{\boldsymbol{m}}_{i,2}, \hat{\boldsymbol{m}}_{i,3}\}$ where $i = 1, 2$ for each gimbal. The gimbal mount frames describe the orientation of each gimbal relative to the spacecraft hub in the home configuration where both gimbal angles are zero. Finally, the gimbal body frames are represented by $\mathcal{G}_i : \{G_i, \hat{\boldsymbol{g}}_{i,1}, \hat{\boldsymbol{g}}_{i,2}, \hat{\boldsymbol{g}}_{i,3}\}$. These frames are fixed to each gimbal during actuation and their origin points G_i coincide with both the mount frame origin points M_i and the gimbal center of mass points G_{c_i}.

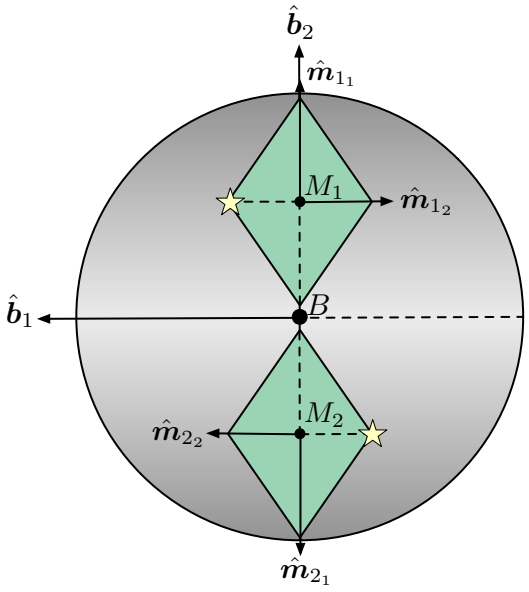


Figure 2: Overhead ($-\hat{\boldsymbol{b}}_3$) view of the SEP gimbal mount frame geometry.

The SEP gimbals are located underneath the EMA spacecraft hub and can be viewed from the $-\hat{\boldsymbol{b}}_3$ axis. Figure (2) shows an overhead view of the gimbals in the home configuration. In this configuration, the gimbal body frames $\mathcal{G}_i$ coincide with their mount frames $\mathcal{M}_i$. Viewing Fig. (2), gimbal 1 is seen to be placed along the positive $\hat{\boldsymbol{b}}_2$ axis, while gimbal 2 is along the negative $\hat{\boldsymbol{b}}_2$ axis. The star symbols indicate the gimbal stowed/launch configurations which are discussed in the next section.

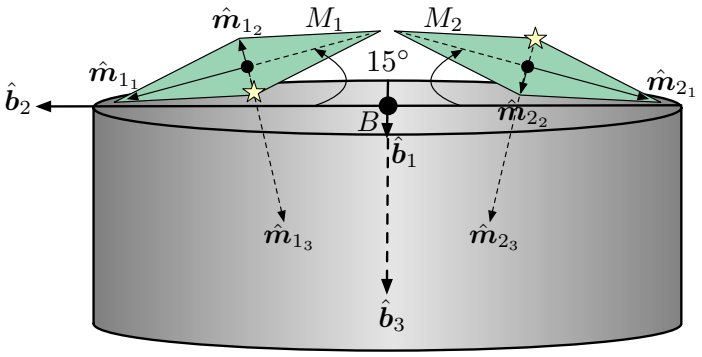


Figure 3: Side view of SEP Gimbal mount frame geometry.

Figure (3) illustrates a side view of the gimbals when viewing from the $\hat{\boldsymbol{b}}_1$ axis. From this viewpoint, it becomes clear that both gimbal mount frames are canted by 15 degrees about the $\hat{\boldsymbol{b}}_1$ axis. The gimbal hub-relative attitudes in their home configurations are represented by the following Euler angle sequence:[27]

$$[\mathcal{M}_i\mathcal{B}] = (3\text{-}1) = (\alpha_i, \beta) = [C_1(\beta)][C_3(\alpha_i)] \tag{1}$$

where $[C_1(\beta)]$ and $[C_3(\alpha_i)]$ are single-axis DCMs corresponding to first and third-axis rotations, respectively:

$$[C_1(\beta)][C_3(\alpha_i)] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\beta & \sin\beta \\ 0 & -\sin\beta & \cos\beta \end{bmatrix} \begin{bmatrix} \cos\alpha_i & \sin\alpha_i & 0 \\ -\sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

The gimbal mount frame attitudes relative to the spacecraft body frame are therefore:

$$[\mathcal{M}_i\mathcal{B}] = \begin{bmatrix} \cos\alpha_i & \sin\alpha_i & 0 \\ -\cos\beta\sin\alpha_i & \cos\beta\cos\alpha_i & \sin\beta \\ \sin\beta\sin\alpha_i & -\sin\beta\cos\alpha_i & \cos\beta \end{bmatrix} \quad i = 1, 2 \tag{3}$$

where $\beta = 15$ degrees for both gimbals, $\alpha_1 = 90$, and $\alpha_2 = -90$.

Next, recall that each gimbal actuates relative to its spacecraft-fixed mount frame $\mathcal{M}_i$. A general depiction of a gimbal body frame $\mathcal{G}$ relative to its mount frame $\mathcal{M}$ is shown in Fig. (4) for reference. Note that the gimbal thrust direction vector $\hat{\boldsymbol{t}}$ is defined as the third gimbal body frame axis $\hat{\boldsymbol{g}}_3$, which points in the direction of the thrust force felt by the spacecraft. Figure (4) depicts the direction of SEP thrusting (blue) which is opposite to the thrust direction vector.

Specifically, the actuation of each gimbal can be represented by a two-axis rotation relative to the mount frame using an Euler angle tip-tilt sequence $(1\text{-}2) = (\psi_i, \phi_i)$, indicating that the gimbal first performs a tip rotation about the spacecraft-fixed mount $\mathcal{M}_i$ frame first axis $\hat{\boldsymbol{m}}_{i_1}$ by ψ_i degrees, followed by a tilt rotation about the intermediate/gimbal $\mathcal{G}_i$ frame second axis $\hat{\boldsymbol{g}}_{i_2}$ by ϕ_i degrees. The Euler angle sequence is mapped to the corresponding direction cosine attitude rotation matrix using the following expression:

$$(1\text{-}2) = (\psi_i, \phi_i) = [C_2(\phi_i)][C_1(\psi_i)] = [\mathcal{G}_i\mathcal{M}_i] \tag{4}$$

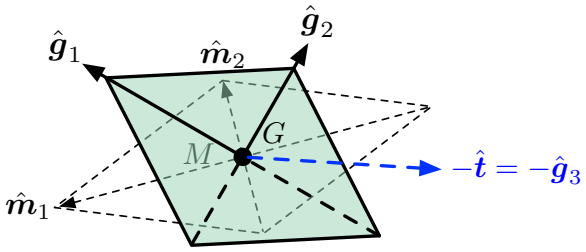


Figure 4: Gimbal and Mount frame definitions.

where $[C_1(\psi_i)]$ and $[C_2(\phi_i)]$ are single-axis DCMs corresponding to the tip and tilt rotations, respectively:

$$[C_2(\phi_i)][C_1(\psi_i)] = \begin{bmatrix} \cos\phi_i & 0 & -\sin\phi_i \\ 0 & 1 & 0 \\ \sin\phi_i & 0 & \cos\phi_i \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi_i & \sin\psi_i \\ 0 & -\sin\psi_i & \cos\psi_i \end{bmatrix} \tag{5}$$

Each gimbal platform attitude relative to the hub-fixed mount frame is therefore:

$$[\mathcal{G}_i\mathcal{M}_i] = \begin{bmatrix} \cos\phi_i & \sin\phi_i\sin\psi_i & -\sin\phi_i\cos\psi_i \\ 0 & \cos\psi_i & \sin\psi_i \\ \sin\phi_i & -\cos\phi_i\sin\psi_i & \cos\phi_i\cos\psi_i \end{bmatrix} \quad i = 1, 2 \tag{6}$$

It is important to note that the motion of each gimbal thrust axis is typically bounded by a cone or diamond, where the boundary of either shape is defined as the actuation limit. In this work, each gimbal's motion is bounded by a diamond with the constraints $\psi_{\text{min}} \leq \psi \leq \psi_{\text{max}}$ and $\phi_{\text{min}} \leq \phi \leq \phi_{\text{max}}$ where $|\psi_{\text{min}}| = |\psi_{\text{max}}| = 15^\circ$ and $|\phi_{\text{min}}| = |\phi_{\text{max}}| = 25^\circ$. Figure (5) illustrates a two-dimensional view of each gimbal's motion boundary. The allowable range is shown in green and the unallowable range is shown in red. The star symbol indicates the gimbal stowed/launch configuration, where $(\psi_{\text{stowed}}, \phi_{\text{stowed}}) = (\psi_{\text{max}}, 0)$.

Next, recall that each two-axis gimbal is actuated via two stepping motors. The rotation angle of each motor is denoted θ, where for gimbal i the associated motor angles are θ_{i_1} and θ_{i_2}. Note that the SEP gimbal motors do not have a continuous 360 degree rotational range. Instead, each motor actuates within the range $0 \leq \theta \leq \theta_{\text{max}}$ where θ_{max} is the hard-stop limit for each motor. The gimbal boundary depicted in Fig. (5) is produced by actuating each stepper motor individually to their hard-stop angles. Starting in the gimbal's stowed configuration where both motor angles are zero, actuating motor 1 to its hard-stop limit traces the north-east edge of the gimbal boundary. If instead motor 2 is actuated independently from the stowed configuration, the south-east edge of the gimbal boundary is traced by the gimbal thrust axis $\hat{\boldsymbol{t}} = \hat{\boldsymbol{g}}_3$. Actuating both motors to their hard-stop limits corresponds to the far left corner of the gimbal boundary. The stepper motors may be actuated simultaneously or individually to actuate the gimbal thrust axes to any location within the boundary.

Both motors rotate with an angular step size of $\Delta\theta$ degrees in the step time Δt. Typically the step angle is a fixed parameter for stepper motors while the motor step time can be selected and varied

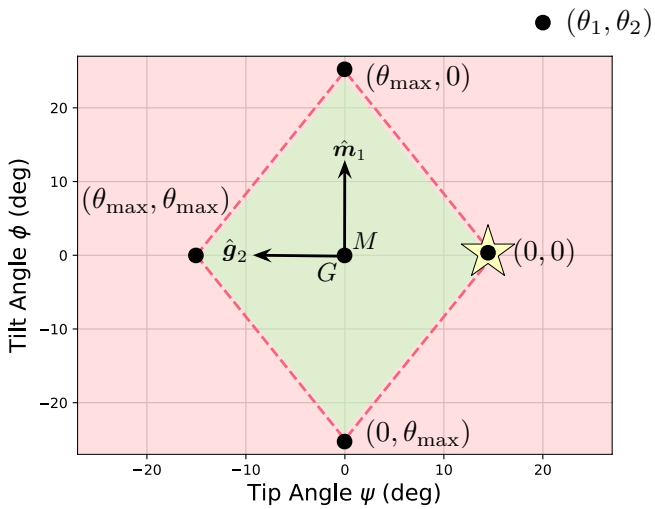


Figure 5: Gimbal tip-tilt motion boundary.

based on mission requirements. Each motor can take a maximum of $n_{s_{\max}} = \frac{\theta_{\max}}{\Delta\theta}$ steps before reaching the hard-stop limit given the fixed motor step angle.

ALGORITHMS

This section outlines the C++ algorithms developed to command and simulate the actuation of the two-axis SEP gimbals and each of their two stepper motors. A block diagram illustrating the algorithm flow is shown in Fig. (6). Three flight software modules (yellow) and two simulation modules (blue) are developed in this work to actuate both the gimbals and their motors. The previously developed prescribed motion dynamics simulation module[22,24] (green) is used to simulate the contribution of the profiled gimbal motion on the spacecraft dynamics. Note that because each gimbal is actuated by two stepper motors, two `stepperMotorController` flight software and `stepperMotor` simulation modules are required to simulate a single gimbal. The first section below describes the three modules contained in the gimbal flight software algorithm chain in detail. The first module in the algorithm chain is the `axisToGimbalAngles` module, which computes the gimbal angles (ψ, ϕ) corresponding to a commanded thrust direction vector in the spacecraft body frame ${}^{\mathcal{B}}\hat{\boldsymbol{t}}$. The second module in the algorithm chain is the `gimbalAnglesToMotorAngles` module, which determines the stepper motor angles θ_1 and θ_2 that correspond to the computed gimbal angles. Finally, two instances of the `stepperMotorController` module are used to determine the number of integer steps n_s required for each motor to actuate to the computed reference values.

The second section outlines two unique kinematics modules used for simulation of the two-axis gimbal motion. The first modules in the algorithm chain are two instances of the `stepperMotor` module. Given the computed number of steps n_s for each motor from the `stepperMotorController` modules, each `stepperMotor` module simulates the kinematic motion of each motor using a bang-bang acceleration profile for each individual motor step taken. The second module in the algorithm chain is the `twoAxisGimbal` module. Using the current motor angles and the commanded motor steps from the `stepperMotor` and `stepperMotorController` modules, respectively, this module profiles the two-axis gimbal

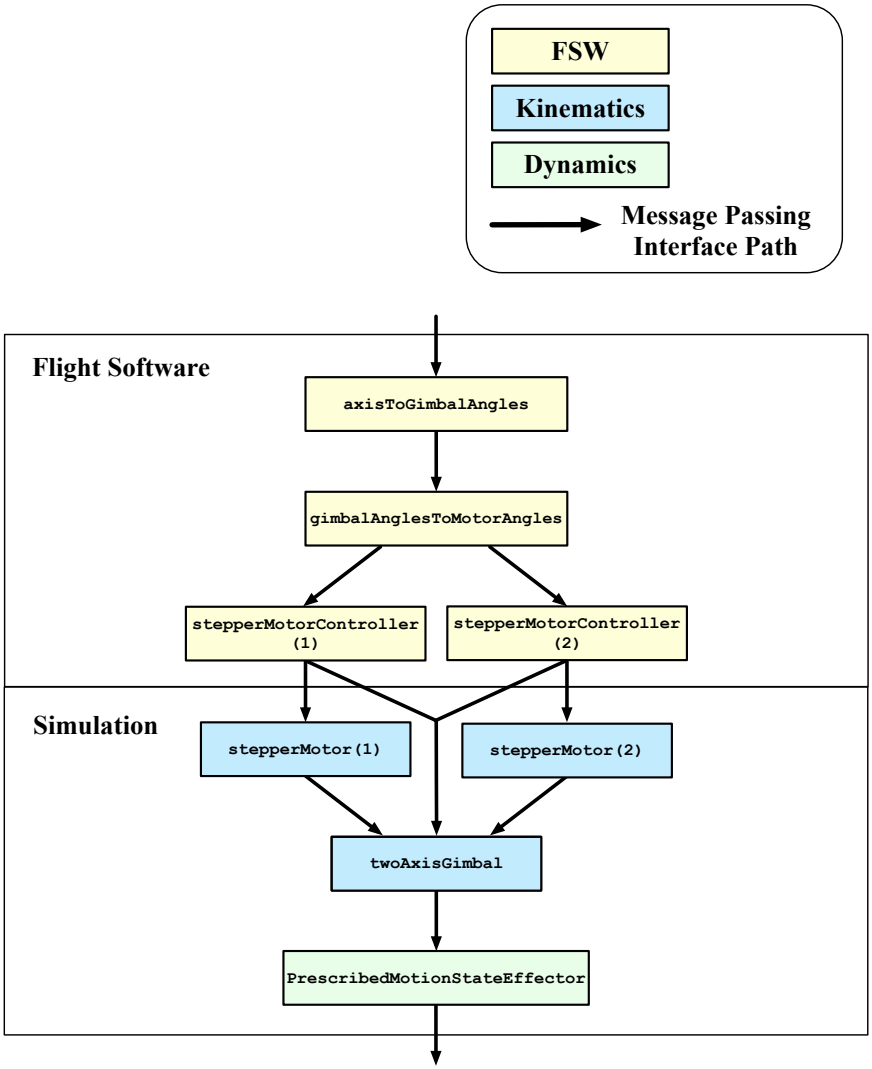


Figure 6: SEP gimbal flight software and simulation algorithms.

hub-relative kinematic motion as a function of time.

The third and final section describes the prescribed motion dynamics module used to simulate the impact of the two-axis gimbal motion on the spacecraft hub motion.

Flight Software Algorithms

`axisToGimbalAngles` Module

The first module in the gimbal flight software algorithm chain is the `axisToGimbalAngles` module; where the incoming thrust direction command ${}^{\mathcal{B}}\hat{\boldsymbol{t}}$ is mapped to the corresponding two-axis gimbal tip and tilt angles ψ and ϕ, respectively.

Given the fixed attitude description of the gimbal platform mount frame $\mathcal{M}$ relative to the spacecraft body frame $\mathcal{B}$ as the Direction Cosine Matrix (DCM) $[\mathcal{MB}]$, the commanded thrust direction can first be mapped to the gimbal mount frame:

$$ {}^{\mathcal{M}}\hat{\boldsymbol{t}} = [\mathcal{MB}]{}^{\mathcal{B}}\hat{\boldsymbol{t}} \tag{7} $$

This vector represents the direction of the thrust force felt by the spacecraft expressed in gimbal mount frame components. This thrust direction vector is identical to the gimbal thrust direction axis, defined as a fixed axis of the gimbal body frame $\mathcal{G}$:

$$ {}^{\mathcal{M}}\hat{\boldsymbol{g}}_i = {}^{\mathcal{M}}\hat{\boldsymbol{t}} \quad i \in \{1,2,3\} \tag{8} $$

Next, the gimbal tip and tilt angles are determined using the Euler angle[27] sequence $(j\text{-}k)$ = (ψ, ϕ), indicating that the gimbal first performs a tip rotation about the gimbal platform mount frame $\hat{\boldsymbol{m}}_j$ axis by ψ degrees, followed by a tilt rotation about the intermediate/gimbal $\mathcal{G}$ frame axis $\hat{\boldsymbol{g}}_k$ by ϕ degrees. The Euler angle sequence is mapped to the corresponding attitude rotation DCM using the following expression:

$$ (j\text{-}k) = (\psi, \phi) = [C_k(\phi)][C_j(\psi)] = [\mathcal{GM}] \tag{9} $$

where $[C_j(\psi)]$ and $[C_k(\phi)]$ are single-axis DCMs corresponding to the tip and tilt rotations, respectively. This matrix can also be expressed using the gimbal body frame axes:

$$ [\mathcal{GM}] = \begin{bmatrix} {}^{\mathcal{M}}\hat{\boldsymbol{g}}_1^T \\ {}^{\mathcal{M}}\hat{\boldsymbol{g}}_2^T \\ {}^{\mathcal{M}}\hat{\boldsymbol{g}}_3^T \end{bmatrix} = \begin{bmatrix} {}^{\mathcal{M}}\hat{\boldsymbol{g}}_{1_1} & {}^{\mathcal{M}}\hat{\boldsymbol{g}}_{1_2} & {}^{\mathcal{M}}\hat{\boldsymbol{g}}_{1_3} \\ {}^{\mathcal{M}}\hat{\boldsymbol{g}}_{2_1} & {}^{\mathcal{M}}\hat{\boldsymbol{g}}_{2_2} & {}^{\mathcal{M}}\hat{\boldsymbol{g}}_{2_3} \\ {}^{\mathcal{M}}\hat{\boldsymbol{g}}_{3_1} & {}^{\mathcal{M}}\hat{\boldsymbol{g}}_{3_2} & {}^{\mathcal{M}}\hat{\boldsymbol{g}}_{3_3} \end{bmatrix} \tag{10} $$

Recall that the gimbal thrust direction axis ${}^{\mathcal{M}}\hat{\boldsymbol{g}}_i$ is known and fixed. Therefore, row i of the known DCM $[\mathcal{GM}]$ can be used to solve for the unknown gimbal tip and tilt angles. For example, assuming $\{i, j, k\} = \{3, 1, 2\}$, the gimbal body frame attitude can be expressed relative to the mount frame using the $(1\text{-}2)$ = (ψ, ϕ) Euler angle sequence:

$$ \begin{aligned} [\mathcal{GM}] = [C_2(\phi)][C_1(\psi)] &= \begin{bmatrix} \cos\phi & 0 & -\sin\phi \\ 0 & 1 & 0 \\ \sin\phi & 0 & \cos\phi \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & \sin\psi \\ 0 & -\sin\psi & \cos\psi \end{bmatrix} \\ &= \begin{bmatrix} \cos\phi & \sin\phi\sin\psi & -\sin\phi\cos\psi \\ 0 & \cos\psi & \sin\psi \\ \sin\phi & -\cos\phi\sin\psi & \cos\phi\cos\psi \end{bmatrix} \end{aligned} \tag{11} $$

where

$$ {}^{\mathcal{M}}\hat{\boldsymbol{g}}_3 = \begin{bmatrix} {}^{\mathcal{M}}\hat{\boldsymbol{g}}_{3_1} \\ {}^{\mathcal{M}}\hat{\boldsymbol{g}}_{3_2} \\ {}^{\mathcal{M}}\hat{\boldsymbol{g}}_{3_3} \end{bmatrix} = \begin{bmatrix} \sin\phi \\ -\cos\phi\sin\psi \\ \cos\phi\cos\psi \end{bmatrix} \tag{12} $$

The required gimbal tip and tilt angles are therefore:

$$ \psi = \tan^{-1}\left(\frac{-\hat{\boldsymbol{g}}_{3_2}}{\hat{\boldsymbol{g}}_{3_3}}\right) \tag{13} $$

$$ \phi = \sin^{-1}\left(\hat{\boldsymbol{g}}_{3_1}\right) \tag{14} $$

`gimbalAnglesToMotorAngles` Module

The `gimbalAnglesToMotorAngles` module is the second flight software module in the algorithm chain. After receiving the computed gimbal angles from the `gimbalAxisToGimbalAngles` module, the corresponding stepper motor 1 and stepper motor 2 angles required to achieve this gimbal attitude are determined and output from the module.

Note that this module requires the gimbal manufacturer to provide two gimbal-to-motor-angle look-up tables (LUT). One LUT is provided for each stepper motor. The structure of these tables is outlined in Fig. (7). Implicitly, each LUT varies the gimbal tip and tilt angles in ascending order along its columns and rows, respectively. Both tables discretize the gimbal angles by the fixed value given by $\delta_{\psi\phi}$. The motor angles corresponding to the discreet combination of gimbal angles are explicitly provided in each LUT. Accordingly, the motor angles corresponding to the minimum gimbal angles are located in the upper-left entry of each table and the angles corresponding to the maximum gimbal angles are located in the bottom-right entry of each table.

A flow diagram for the algorithm developed for this module is provided in Fig. (8. Given the provided LUTs and the desired gimbal tip and tilt angles, this module first determines whether no interpolation, linear interpolation, or bilinear interpolation is required to obtain the corresponding stepper motor angles. If both gimbal angles are exactly divisible by the discretization value $\delta_{\psi\phi}$, the corresponding motor angles can be directly extracted from the LUTs and no interpolation is required. Linear interpolation is required if only one of the gimbal angles is exactly divisible by $\delta_{\psi\phi}$. If neither gimbal angle is divisible by $\delta_{\psi\phi}$, bilinear interpolation is required to obtain each motor angle.

Next, upper and lower bounds for the gimbal angles are determined if interpolation is required. If linear interpolation is required, the upper and lower bounds for the gimbal angle not divisible by $\delta_{\psi\phi}$ are determined as the nearest multiples of $\delta_{\psi\phi}$. If bilinear interpolation is required, the upper and lower bounds for both gimbal angles are computed similarly. Using these boundary values, the corresponding motor angles are directly extracted from the provided LUTs.

A boundary case check must first be completed before interpolation can be performed with the extracted motor angles. This is because in order to generalize this module for any unique two-axis gimbal, there is no constraint placed on the range of each gimbal angle. Therefore, if the range of one gimbal angle is not identical to the other there will be blank entries in the LUTs corresponding to invalid motor positions. If any of the extracted motor angles are returned as blank entries, a boundary case is identified and the type of interpolation required is adjusted.

For example, if linear interpolation is selected for one of the motors and one of the two returned bounding values is empty, no interpolation is performed and the valid bounding value is returned

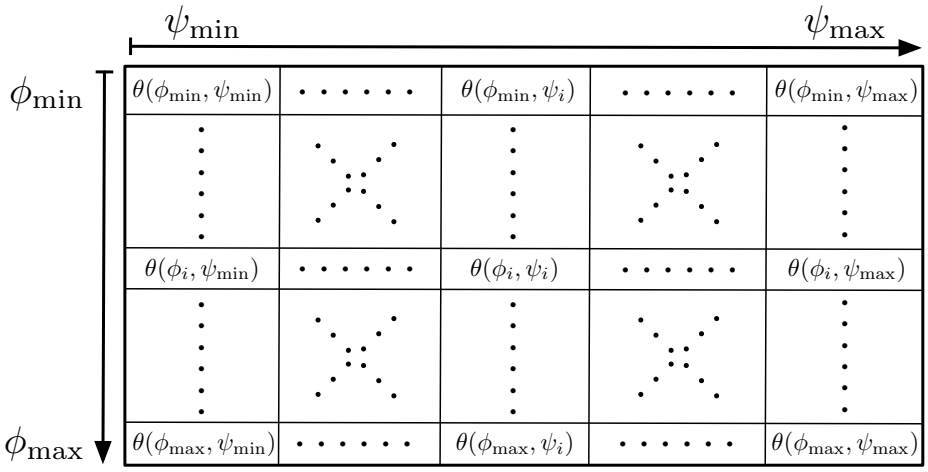


Figure 7: Gimbal-to-motor angle lookup table data format.

as the true motor angle. If neither bounding value returned is valid, an assertion error is thrown. If bilinear interpolation is selected for one of the motors and one of the four returned bounding values is empty, the triangular centroid of the remaining three bounding points is determined as the true motor angle. If two of the four returned bounding values are empty, linear interpolation is instead performed with the two valid values. If only one of the returned values is valid, no interpolation is performed and the valid bounding value is returned as the true motor angle. If all four returned bounding points for each motor are invalid, an assertion error is thrown. With these boundary checks in place, this module is generalized for many different types of two-axis gimbals with varying ranges. The algorithm given in Fig. (8) shows how to compute the corresponding motor angles in any circumstance.

`stepperMotorController` Module

The final modules in the flight software algorithm chain are two `stepperMotorController` modules. For each stepper motor, an instance of this module is invoked and used to determine the number of discreet steps n_s required to actuate the motor from its current position θ to the reference angle θ_{ref} computed in the previous `gimbalAnglesToMotorAngles` module. The required number of integer steps n_s for each motor is computed as:

$$n_s = \begin{cases} \lfloor \frac{\theta_{\text{ref}} - \theta}{\Delta\theta} \rfloor & \text{if } \lceil \frac{\theta_{\text{ref}} - \theta}{\Delta\theta} \rceil - \frac{\theta_{\text{ref}} - \theta}{\Delta\theta} > \frac{\theta_{\text{ref}} - \theta}{\Delta\theta} - \lfloor \frac{\theta_{\text{ref}} - \theta}{\Delta\theta} \rfloor, \\ \lceil \frac{\theta_{\text{ref}} - \theta}{\Delta\theta} \rceil & \text{if } \lceil \frac{\theta_{\text{ref}} - \theta}{\Delta\theta} \rceil - \frac{\theta_{\text{ref}} - \theta}{\Delta\theta} < \frac{\theta_{\text{ref}} - \theta}{\Delta\theta} - \lfloor \frac{\theta_{\text{ref}} - \theta}{\Delta\theta} \rfloor. \end{cases} \tag{15}$$

The above logic is used to ensure an integer number of steps is returned. If the computed number of steps is not an integer, the result is rounded to the nearest integer. Note that while the main function of this module is to output the required number of motor steps n_s, this module also tracks the current motor angle and step count at the flight software control frequency for the purpose of fault protection.

Kinematics Algorithms

`stepperMotor` Module

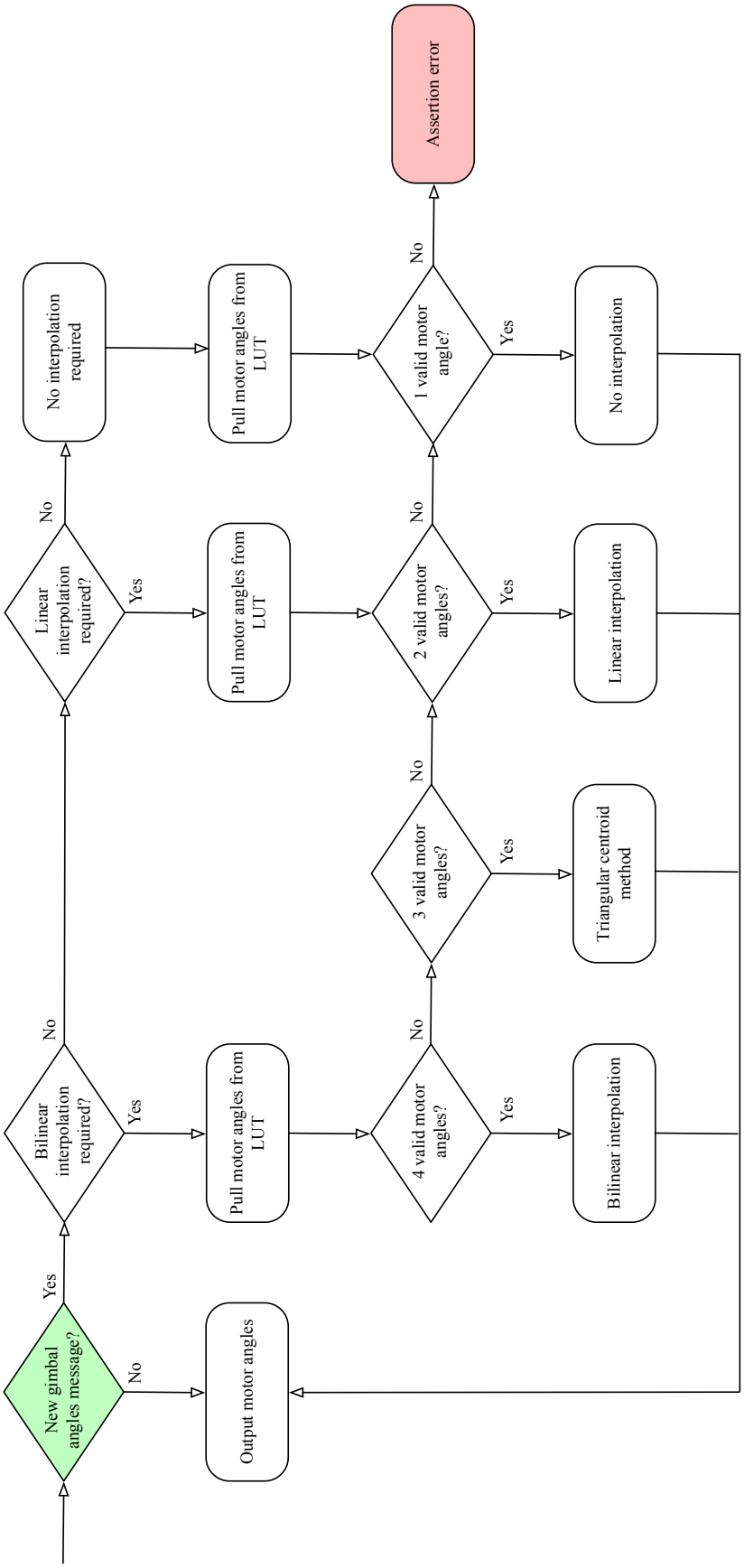


Figure 8: `gimbalAnglesToMotorAngles` algorithm flow.

Given the commanded number of steps to each motor n_s from the flight software `stepperMotorController` modules, the `stepperMotor` kinematics module is next used to simulate the stepping motion of each stepper motor. A bang-bang acceleration profile is used to profile each individual motor step. Given a fixed motor step angle $\Delta\theta$ and a selected motor step time Δt, the maximum angular acceleration $\ddot{\theta}_{\text{max}}$ required to profile each motor step can be obtained using the expression:

$$\ddot{\theta}_{\text{max}} = \frac{4\Delta\theta}{\Delta t^2} \tag{16}$$

Note that each motor can take forward or backward steps. For a forward motor step, the acceleration is applied positively during the first half of the step and negatively during the second half. For a backward step, the acceleration is first applied negatively and second applied positively.

The switch time t_s indicates the time the acceleration is alternated:

$$t_s = t_0 + \frac{\Delta t}{2} \tag{17}$$

The equations used to kinematically profile each motor step are:[23]

$$\ddot{\theta}(t) = \begin{cases} \pm\ddot{\theta}_{\text{max}} & \text{if } t_0 \leq t < t_s \\ \mp\ddot{\theta}_{\text{max}} & \text{if } t_s \leq t < t_f \\ 0 & \text{if } t \geq t_f \end{cases} \tag{18}$$

$$\dot{\theta}(t) = \begin{cases} \pm\ddot{\theta}_{\text{max}}(t - t_0) & \text{if } t_0 \leq t < t_s \\ \mp\ddot{\theta}_{\text{max}}(t - t_f) & \text{if } t_s \leq t < t_f \\ 0 & \text{if } t \geq t_f \end{cases} \tag{19}$$

$$\theta(t) = \begin{cases} \frac{\pm\Delta\theta}{2}\frac{(t-t_0)^2}{(t_s-t_0)^2} + \theta_0 & \text{if } t_0 \leq t < t_s \\ \frac{\mp\Delta\theta}{2}\frac{(t-t_f)^2}{(t_s-t_f)^2} + \theta_{\text{ref}} & \text{if } t_s \leq t < t_f \\ \theta_{\text{ref}} & \text{if } t \geq t_f \end{cases} \tag{20}$$

Note that for each successive motor step, the parameters t_0, t_s, and t_f must be continually updated to reflect the advancement of time. Doing so enables use of the above equations for each motor step.

Additional logic is included in this module to allow for controller interruptions during a stepping sequence. If a new step count command is published before actuation is completed for the previous step command, this module considers the new command as truth and resets the motor step count prior to beginning the new commanded step sequence. If the motor is interrupted during the completion of a step, the current step is completed before the new command sequence is initiated. The algorithm flow for this module is provided in Fig. (9).

***twoAxisGimbal* Module**

The `twoAxisGimbal` kinematics module is the primary module responsible for profiling the gimbal actuation. Given the current motor angles from the `stepperMotor` modules and the motor step commands from the `stepperMotorController` modules, this module profiles the corresponding gimbal motion in time using a second set of LUTs.

Similar to the two gimbal-to-motor angle LUTs described previously, this module requires the gimbal manufacturer to provide two motor-to-gimbal angle LUTs. One LUT is provided for each

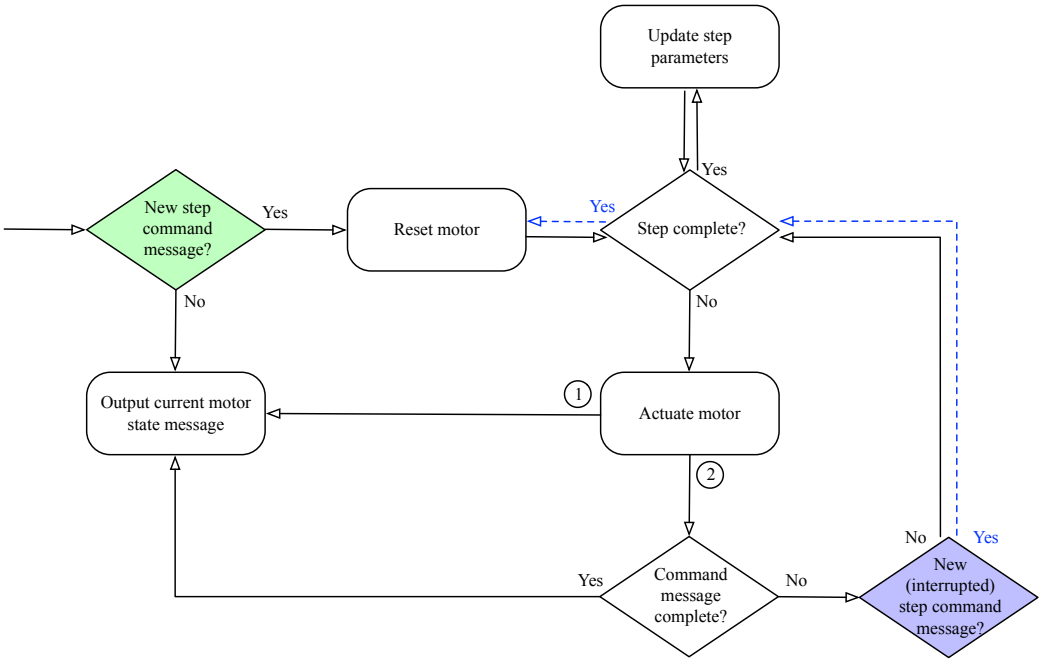


Figure 9: `stepperMotor` algorithm flow.

gimbal angle. The structure of these tables is outlined in Fig. (10). Implicitly, each LUT varies the motor 1 and 2 angles in ascending order along its columns and rows, respectively. Both tables discretize the motor angles by the fixed value given by δ_θ. The gimbal angles corresponding to the discreet combination of motor angles are explicitly provided in each LUT. Accordingly, the gimbal angles corresponding to the minimum motor angles are located in the upper-left entry of each table and the angles corresponding to the maximum motor angles are located in the bottom-right entry of each table. These LUTs are used to interpolate the gimbal motion for any combination of stepper motor motion. Note that these tables are square because the range of both motors are identical. Therefore, no boundary case checks need to be performed prior to gimbal angle interpolation.

The algorithm flow for this module is provided in Fig. (11). The logic enables simultaneous or independent motor actuation. The motors can be commanded independently by ensuring one of the motor step command messages is zero. If the number of steps commanded to each motor are identical, both motors will actuate simultaneously for the duration of the gimbal motion. The corresponding gimbal motion will reflect the shortest path to reach the commanded attitude. In this case, the gimbal will take the same number of steps as the motors to reach the commanded attitude. The gimbal motion is considered to be contained within a single actuation segment for this case.

If the motor step commands do not match, the gimbal motion is defined using two actuation segments. During the first segment, both motors will actuate simultaneously until the motor with less commanded steps completes its required step counts. The second actuation segment profiles the gimbal motion for the remaining number of motor steps that must be completed by the motor with the greater number of steps. The number of steps the gimbal actuates n_g in this case is equal

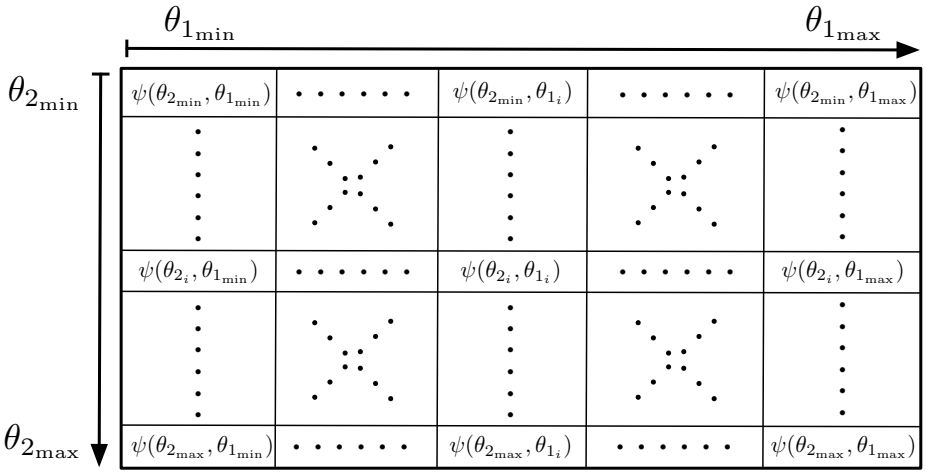


Figure 10: Motor-to-gimbal angle lookup table data format.

to the greater of the two motor step commands.

$$n_g = \begin{cases} n_s & \text{if } n_{s_1} = n_{s_2} = n_s \\ |n_{s_1}| & \text{if } |n_{s_1}| > |n_{s_2}| \\ |n_{s_2}| & \text{if } |n_{s_2}| > |n_{s_1}| \end{cases} \tag{21}$$

For each actuation segment, the gimbal motion is profiled along a Principal Rotation Vector (PRV)[27] path. The required PRV angle is determined by first using the motor-to-gimbal LUTs to obtain the gimbal angles at the start and end of the actuation segment. Next, the corresponding initial ($[\mathcal{G}_0\mathcal{M}]$) and final ($[\mathcal{G}_f\mathcal{M}]$) gimbal attitude DCMs can be determined using Eq. (6). Using these DCMS, the relative attitude ($[\mathcal{F}_f\mathcal{G}_0]$) between the initial and final gimbal attitudes is used to solve for the required PRV $\boldsymbol{\gamma}$:

$$[\tilde{\boldsymbol{\gamma}}] = -\ln([\mathcal{G}_f\mathcal{G}_0]) = \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n} \left(I_{3\times 3} - [\mathcal{G}_f\mathcal{G}_0]\right)^n \tag{22}$$

The tilde matrix operator $[\tilde{\boldsymbol{v}}]$ indicates the matrix cross-product:

$$[\tilde{\boldsymbol{v}}] = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & v_0 \end{bmatrix} \tag{23}$$

Finally, the principal rotation axis and angle information can extracted from the determined PRV:

$$\Phi = |\boldsymbol{\gamma}| \tag{24a}$$

$$\hat{\boldsymbol{e}} = \frac{\boldsymbol{\gamma}}{\Phi} \tag{24b}$$

Given the number of gimbal steps during the actuation segment, the determined PRV angle Φ is divided evenly by the number of gimbal steps to ensure the gimbal and motors finish actuation

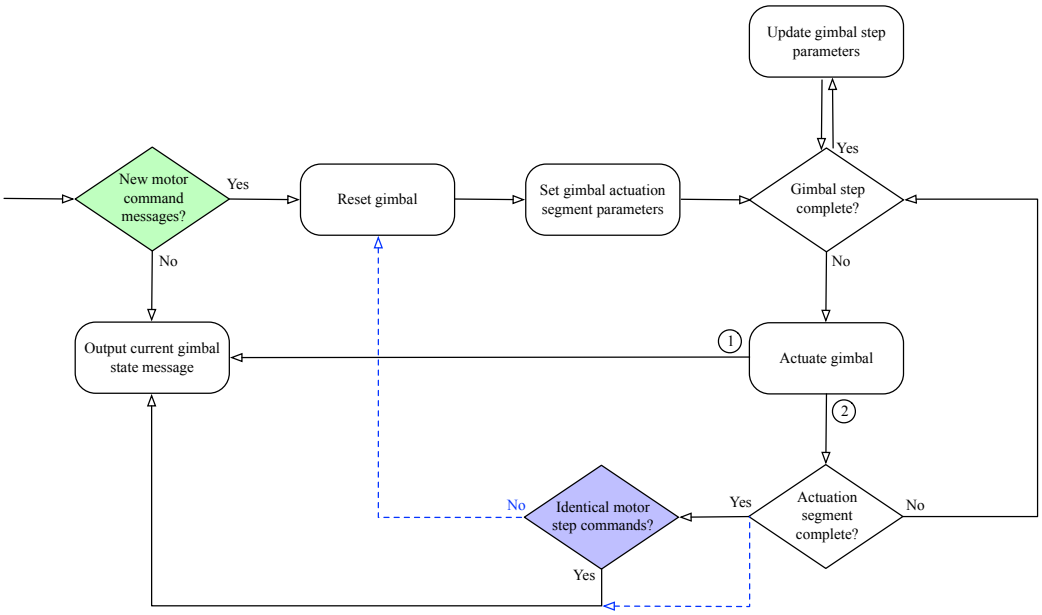


Figure 11: `twoAxisGimbal` algorithm flow.

simultaneously. The gimbal step motion is profile similarly to the motor step motion, where the corresponding gimbal step angle $\Delta\lambda$ is used in Eq. (16) to determine the required gimbal acceleration during each step. Equations (17-20) are used to profile each gimbal step.

At each time step, the gimbal tip and tilt angles are output from this module using Eq. (14) and the inverse of the transformations in Eqs. (22) and (24). This module also outputs the gimbal's hub-relative vector states $\boldsymbol{\sigma}_{\mathcal{G}/\mathcal{M}}, \boldsymbol{\omega}_{\mathcal{G}/\mathcal{M}}$ and $\boldsymbol{\omega}'_{\mathcal{G}/\mathcal{M}}$ using the following expressions:

$$\boldsymbol{\sigma}_{\mathcal{G}/\mathcal{M}} = \tan\left(\frac{\Phi}{4}\right)\hat{\boldsymbol{e}} \tag{25a}$$

$${}^{\mathcal{G}}\boldsymbol{\omega}_{\mathcal{G}/\mathcal{M}} = \dot{\Phi}\hat{\boldsymbol{e}} \tag{25b}$$

$${}^{\mathcal{G}}\boldsymbol{\omega}'_{\mathcal{G}/\mathcal{M}} = \ddot{\Phi}\hat{\boldsymbol{e}} \tag{25c}$$

DYNAMCIS ALGORITHMS

`prescribedMotionStateEffector` Module

The final module in the gimbal algorithm chain is the `prescribedMotionStateEffector` dynamics module. Using the hub-relative gimbal rotational states provided from the `twoAxisGimbal` kinematics module, this module adds the contribution of the gimbal actuation to the spacecraft dynamic using the following equations of motion derived in previous work:[22]

$$m_{\text{sc}}\ddot{\boldsymbol{r}}_{B/N} + m_{\text{sc}}[\dot{\tilde{\boldsymbol{\omega}}}_{\mathcal{B}/\mathcal{N}}]\boldsymbol{c} = \sum \boldsymbol{F}_{\text{ext}} - m_{\text{G}}\boldsymbol{r}''_{G_c/B} - 2m_{\text{sc}}[\tilde{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{N}}]\boldsymbol{c}' - m_{\text{sc}}[\tilde{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{N}}]^2\boldsymbol{c} \tag{26}$$

$$m_{\text{sc}}[\tilde{\boldsymbol{c}}]\ddot{\boldsymbol{r}}_{B/N} + [I_{\text{sc},B}]\dot{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{N}} = \boldsymbol{L}_B - m_{\text{G}}[\tilde{\boldsymbol{r}}_{G_c/B}]\boldsymbol{r}''_{G_c/B} - \left([I'_{\text{sc},B}] + [\tilde{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{N}}][I_{\text{sc},B}]\right)\boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}}$$
$$- \left([I'_{\text{G},G_c}] + [\tilde{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{N}}][I_{\text{G},G_c}]\right)\boldsymbol{\omega}_{\mathcal{G}/\mathcal{B}} - [I_{\text{G},G_c}]\boldsymbol{\omega}'_{\mathcal{G}/\mathcal{B}} - m_{\text{G}}[\tilde{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{N}}][\tilde{\boldsymbol{r}}_{G_c/B}]\boldsymbol{r}'_{G_c/B} \quad (27)$$

Eqs. (26) and (27) are the translational and rotational equations of motion describing a central rigid spacecraft hub with an attached rigid prescribed motion component. Equation (26) is the translational equation of motion for the spacecraft hub point B with respect to the inertial frame. Equation (27) is the rotational equation of motion for the spacecraft system. These vector equations are both frame-independent and general, which enables the simulation of a wide variety of spacecraft configurations.

SIMULATION RESULTS

This section first presents a general gimbal commanding scenario in order to demonstrate the validity of the developed algorithms. This section next provides results simulating the EMA SEP Point scenario using the developed algorithms. This scenario is used to demonstrate successful implementation of the gimbal algorithms into the full EMA Guidance, Navigation and Control (GNC) software stack. Relevant parameters for both simulations are provided in Table (1).

Table 1: Shared simulation parameters.

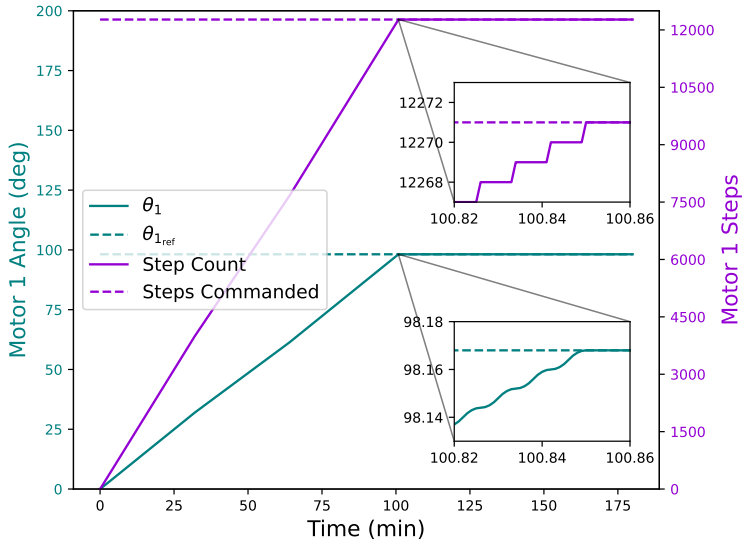
Parameter	**Notation**	**Value**	**Unit**
Gimbal tip angle bound	ψ_{max}	18.5	deg
Gimbal tilt angle bound	ϕ_{max}	27.5	deg
Motor hard stop (upper limit)	θ_{max}	160	deg
Motor step angle	$\Delta\theta$	0.008	deg
Motor step time	Δt	0.008	sec
Motor maximum acceleration	$\ddot{\theta}_{\text{max}}$	2.0	deg/s^2
LUT motor discretization angle	δ_θ	0.5	deg
LUT gimbal discretization angle	$\delta_{\psi\phi}$	0.5	deg

General Gimbal Commanding Scenario

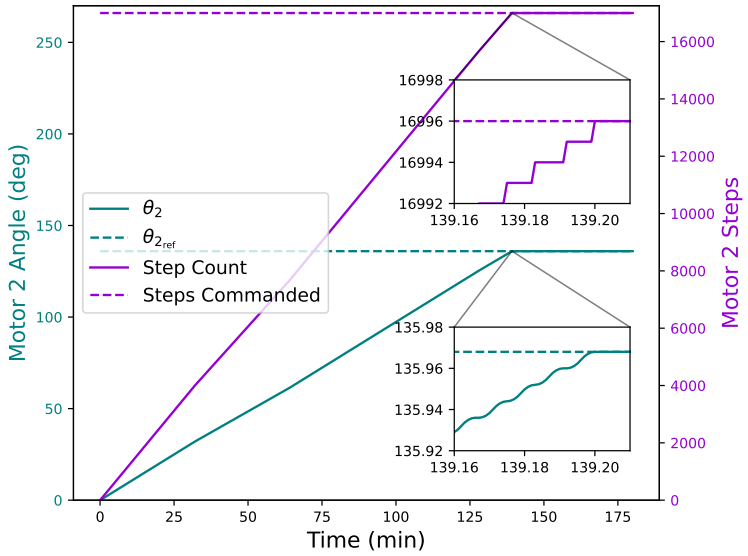
Table (1) provides the parameters used to obtain results for the general gimbal commanding scenario. The gimbal begins in the stowed configuration where both motor angles are zero. The chosen thrust direction vector command corresponds to the gimbal tip and tilt angles $(\psi_{\text{ref}}, \phi_{\text{ref}}) = (-5^\circ, -10^\circ)$. Figure (12) shows the motor angles and steps taken while the gimbal is actuating to the commanded reference. Figure (13) displays the gimbal trajectory and pointing accuracy during the actuation. These results confirm the gimbal algorithms are performing as intended.

EMA SEP Point Scenario

Next, the EMA SEP Point scenario is simulated for seven days. Beginning in the home position, first gimbal 1 is commanded once every hour for 3.5 days to actuate to the required body-relative thrust direction vector. The SEP thrusters are switched after 3.5 days and gimbal 2 is commanded



(a) Motor 1 actuation.

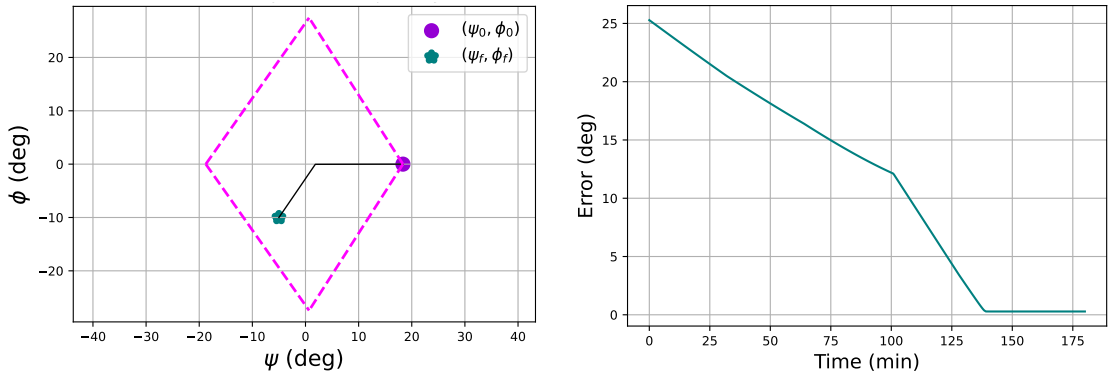


(b) Motor 2 actuation.

Figure 12: Motor actuation.

Table 2: General commanding scenario parameters.

Parameter	Notation	Value	Unit
Initial gimbal tip angle	ψ_0	0.0	deg
Initial gimbal tilt angle	ϕ_0	18.5	deg
Initial motor 1 angle	θ_{0_1}	0.0	deg
Initial motor 1 angle	θ_{0_2}	0.0	deg
Thrust direction reference vector	${}^{\mathcal{B}}\hat{\boldsymbol{t}}$	$[-0.0858 \ -0.4216 \ 0.9027]$	N/A
Reference gimbal tip angle	ψ_{ref}	-5.0	deg
Reference gimbal tilt angle	ϕ_{ref}	-10.0	deg



(a) Gimbal actuation.

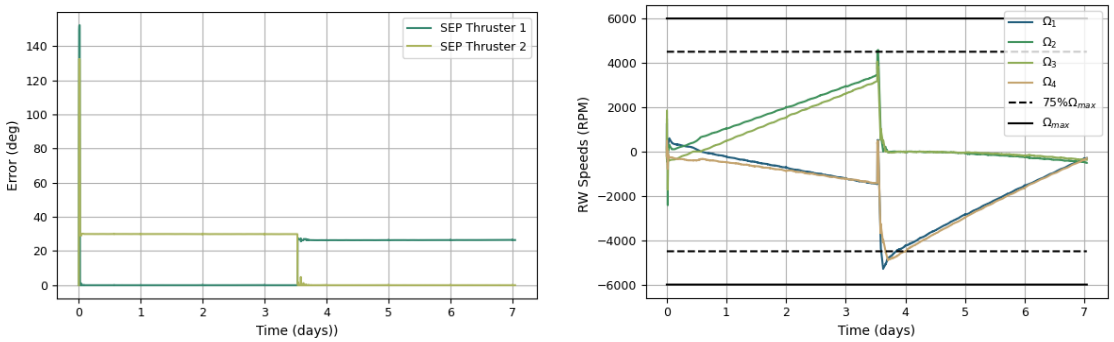
(b) Gimbal pointing accuracy.

Figure 13: Gimbal performance.

once every hour for the second 3.5 day period. Figure (14) provides the gimbal pointing accuracies and reaction wheel speeds over the seven day thrusting period. After the initial transient periods during the slew to the commanded attitude, the reaction wheel speeds are seen to evolve linearly over time due to the constant swirl torque of each thruster. The reaction wheel speeds are seen to remain within their operational envelopes for the full seven day period. Swapping the SEP thrusters allows the spacecraft to end the 7-day thrust arc with minimal accumulated momentum due to their opposite polarity.

CONCLUSION

The Emirates Mission to the Asteroid Belt GNC SEP Point State requires independent actuation of two two-axis gimballed platforms in order to orient the SEP thrusters to a commanded body-relative thrust direction. This paper outlines the C++ algorithms developed to command and actuate the SEP gimbal platforms. Three flight software modules are developed to command two stepper motors for each gimbal and two kinematics modules are developed to simulate the corresponding stepper motor and gimbal motion. The previously developed prescribed motion dynamics module is used to simulate the contribution of the gimbal motion to the spacecraft dynamics. The presented



(a) Thruster pointing accuracy.

(b) Reaction wheel speeds.

Figure 14: SEP Point scenario results.

simulation results demonstrate successful execution of the EMA GNC SEP Point State. Future work involves rigorous verification and validation of the algorithms presented in this work.

ACKNOWLEDGEMENTS

Support for this effort was provided by the Emirates Mission to the Asteroid Belt which is led by the UAE Space Agency in collaboration with the Laboratory for Atmospheric and Space Physics at the University of Colorado Boulder as its knowledge partner.

REFERENCES

[1] R. Calaon, C. Allard, and H. Schaub, "Solar Electric Propulsion GNC Pointing State Overview For The Emirates Mission To The Asteroid Belt," *AAS Guidance and Control Conference*, Breckenridge, CO, Feb. 2–7 2024. Paper No. AAS 24-056.

[2] M. D. Rayman and D. H. Lehman, "Deep space one: NASA's first Deep-Space technology validation mission," *Acta Astronautica*, Vol. 41, No. 4, 1997, pp. 289–299.

[3] M. D. Rayman, T. C. Fraschetti, C. A. Raymond, and C. T. Russell, "Dawn: A mission in development for exploration of main belt asteroids Vesta and Ceres," *Acta Astronautica*, Vol. 58, No. 11, 2006, pp. 605–616.

[4] T. D. W. H. T. I. e. a. D. Y. Oh, S. Collins, "Development of the Psyche Mission for NASA's Discovery Program," *36th International Electric Propulsion Conference*, University of Vienna, Vienna Austria, Sep. 15–20 2019. Paper No. IEPC-2019-192.

[5] M. Yoshikawa, J. Kawaguchi, A. Fujiwara, and A. Tsuchiyama, "Chapter 6 - The Hayabusa mission," *Sample Return Missions* (A. Longobardo, ed.), pp. 123–146, Elsevier, 2021.

[6] Y. Tsuda, M. Yoshikawa, M. Abe, H. Minamino, and S. Nakazawa, "System design of the Hayabusa 2—Asteroid sample return mission to 1999 JU3," *Acta Astronautica*, Vol. 91, 2013, pp. 356–362.

[7] G. Racca, A. Marini, L. Stagnaro, J. van Dooren, L. di Napoli, B. Foing, R. Lumb, J. Volp, J. Brinkmann, R. Grünagel, D. Estublier, E. Tremolizzo, M. McKay, O. Camino, J. Schoemaekers, M. Hechler, M. Khan, P. Rathsman, G. Andersson, K. Anflo, S. Berge, P. Bodin, A. Edfors, A. Hussain, J. Kugelberg, N. Larsson, B. Ljung, L. Meijer, A. Mörtsell, T. Nordebäck, S. Persson, and F. Sjöberg, "SMART-1 mission description and development status," *Planetary and Space Science*, Vol. 50, No. 14, 2002, pp. 1323–1337.

[8] B. Smith, W. Schrenk, W. Gass, and Y. Shtessel, "Sliding mode control in a two-axis gimbal system," Vol. 5, 02 1999, pp. 457 – 470 vol.5, 10.1109/AERO.1999.790222.

[9] B. Ekstrand, "Equations of motion for a two-axes gimbal system," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 37, No. 3, 2001, pp. 1083–1091, 10.1109/7.953259.

[10] D.-H. Lee, D.-Q. Tran, Y.-B. Kim, and S. Chakir, "A Robust Double Active Control System Design for Disturbance Suppression of a Two-Axis Gimbal System," *Electronics*, Vol. 9, 10 2020, p. 1638.

[11] M. Abdo, A. R. Vali, A. Toloei, and M. R. Arvan, "Research on the Cross-Coupling of a Two Axes Gimbal System with Dynamic Unbalance," *International Journal of Advanced Robotic Systems*, Vol. 10, No. 10, 2013, p. 357.

[12] Y. Wang, X. Sui, T. Zhang, T. Nie, and C. Chen, "Design and Experimental Study on the Torque Balancing Mechanism of a Satellite-Borne Two-Axis Rotary Table," *Machines*, Vol. 11, No. 8, 2023.

[13] S. Niazi, A. Toloei, and R. Ghasemi, "Neuro-predictive controller for stabilization of gimbal mechanism with cross-coupling," *Mechatronic Systems and Control*, Vol. 49, 01 2021.

[14] A. Aly, M. Elhabib, B. Felemban, B. Saleh, and D.-N. Le, "Modeling and Simulation of Two Axes Gimbal Using Fuzzy Control," *Computers, Materials Continua*, Vol. 72, 02 2022, pp. 93–107.

[15] L. Zhang, H. Nan, Z. Zhao, and Y. Yuan, "Adaptive disturbance observer-based dual-loop integral-type fast terminal sliding mode control for micro spacecraft and its gimbal tracking device," *ISA Transactions*, Vol. 130, 2022, pp. 121–135.

[16] K. Tan, T. Lee, A. Mamun, M. Lee, and C. Khoh, "Composite control of a gyro mirror line-of-sight stabilization platform — design and auto-tuning," *ISA Transactions*, Vol. 40, No. 2, 2001, pp. 155–171.

[17] O. Hasturk, A. M. Erkmen, and Erkmen, "Proxy-Based Sliding Mode Stabilization of a Two-Axis Gimbaled Platform," 2011.

[18] Q. Huang, J. Zhou, X. Chen, Y. Yao, Y. Chen, W. Chen, R. Chen, and Z. Lv, "Modeling and Control of a Two-Axis Stabilized Gimbal Based on Kane Method," *Sensors*, Vol. 24, No. 11, 2024.

[19] J. Zhang, Z. Jin, Y. Zhao, Y. Tang, F. Liu, Y. Lu, and P. Liu, "Design and Implementation of Novel Fractional-Order Controllers for Stabilized Platforms," *IEEE Access*, Vol. 8, 2020, pp. 93133–93144, 10.1109/ACCESS.2020.2994105.

[20] S. Senthil Kumar and G. Anitha, "A Novel Self-Tuning Fuzzy Logic-Based PID Controllers for Two-Axis Gimbal Stabilization in a Missile Seeker," *International Journal of Aerospace Engineering*, Vol. 2021, No. 1, 2021, p. 8897556.

[21] M. Şahin, "Stabilization of Two Axis Gimbal System with Self Tuning PID Control," *Politeknik Dergisi*, Vol. 27, No. 4, 2024, pp. 1441–1452.

[22] L. Kiner, J. Vaz Carneiro, and H. Schaub, "Spacecraft Simulation Software Implementation Of General Prescribed Motion Dynamics Of Two Connected Rigid Bodies," *AAS Guidance and Control Conference*, Breckenridge, CO, Feb. 2–8 2023. Paper No. AAS-23-084.

[23] L. Kiner, C. Allard, and H. Schaub, "Multi-Body Prescribed Spacecraft Dynamics Subject To Actuator Inputs," *AAS/AIAA Astrodynamics Specialist Conference*, Big Sky, MO, Aug. 13–17 2023. Paper AAS 23–230.

[24] L. Kiner, C. Allard, and H. Schaub, "Prescribed Motion Dynamics For Spacecraft Solar Array Deployment," *AAS Guidance and Control Conference*, Breckenridge, CO, Feb. 2–7 2024. Paper No. AAS 24-123.

[25] A. Jain and G. Rodriguez, "Recursive Dynamics Algorithm for Multibody Systems with Prescribed Motion," *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 5, 1993, pp. 830–837.

[26] R. McGregor and L. Oshinowo, "Flight 6A: Deployment and Checkout of the Space Station Remote Manipulator System (SSRMS)," 6th International Symposium on Artificial Intelligence and Robotics and Automation in Space: i-SAIRAS 2001, Canadian Space Agency, St-Hubert, Quebec, Canada, June 18-22, 2001.

[27] H. Schaub and J. L. Junkins, *Analytical Mechanics of Space Systems*. Reston, Virginia: American Institute of Aeronautics and Astronautics, Inc., 4 ed., 2018.