2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

GNC Design and Orbital Performance Evaluation of ISS Onboard Autonomous Free-Flying Robot Int-Ball2

Taisei Nishishita[1], Keisuke Watanabe[1], Daichi Hirano[1] and Shinji Mitani[1]

Abstract— **The International Space Station (ISS) crew has to complete various tasks in a limited time. The Int-Ball project is one of the activities aimed at the efficient utilization of crew time. Int-Ball2 is an autonomous mobile robot that conducts imaging operations in place of crew members by remote control from the ground. This paper first introduces the design of the guidance, navigation and control (GNC) and propulsion systems of Int-Ball2, offering autonomous flight inside the ISS. This robot has unique features compared to other free-flying robots in space, including its small size for use in crowded environments and its high thrust despite its size. In addition, the navigation system uses a visual SLAM algorithm, which does not rely on external markers. The results of orbital performance verification tests for 6-DOF translational and rotational motion are also presented in this paper. Our analysis indicated that the proposed GNC algorithm provided sufficient performance to conduct the required photography. Finally, the results of ground tests simulating the microgravity environment are compared with the results of orbital experiments to evaluate the differences in the robot's mobility performance in these tests. The results confirmed that the ground verification method was valid for achieving stable on-orbit mobility.**

I. INTRODUCTION

Many experiments have been conducted on the International Space Station (ISS), and leading to numerous discoveries. In orbital experiments, the crew conducts the tests instead of researchers on the ground. However, crew resources are limited and precious, and their effective utilization is crucial to maximize the results. Photography is one of the crew tasks that robots can perform readily to significantly improve the crew's efficiency. When performing experiments on the ISS's Japanese Experiment Module (JEM), the crew must set up cameras around the module to send videos of their work to the ground. However, shooting from an arbitrary location is impossible because the camera must be mounted on the JEM wall. In addition, it is not easy to repeatedly instruct the crew to fine-tune the camera's angle of view, which consumes much valuable crew time. To solve this, the Japan Aerospace Exploration Agency (JAXA) developed the Int-Ball, an autonomous mobile camera robot. This robot can move freely in the JEM to an arbitrary location commanded by ground control. The first Int-Ball was launched in 2017 to verify basic functions such as the propeller propulsion system [1]. The next-generation Int-Ball2 was developed using lessons learned from the on-orbit experiments of the first model [2]. Int-Ball2 was launched by Falcon 9 in June 2023 and is currently undergoing an operational checkout of its functions on the JEM.

Various free-flying robots of the same type as Int-Ball have been developed and flight-demonstrated on the ISS. Astrobee, developed by NASA, draws in air from the impellers on the left and right sides of the robot and opens and closes 12 nozzles to generate thrust [3], [4]. DLR also developed and launched a free-flyer, Crew Interactive MObile companioN (CIMON). The robot has 14 propeller fans arranged in a 7-pair cylindrical configuration that can generate thrust in either the positive or negative direction of the fan's normal axis [5], [6]. In common with the two robots, the propulsion system's airflow path inside the body is carefully designed to avoid interference between thrusters. This makes the airflow less turbulent and reduces thrust fluctuations. On the other hand, Int-Ball2 needed to be as small as possible so as not to disturb the crew's work even in a crowded environment. Consequently, Int-Ball2 prioritizes miniaturization without providing isolated air-intake ports and paths, allowing it to draw air inside through slits at various places on the body surface. To achieve stable flight under this constraint, it is necessary to realize a highly robust control system, which is technically challenging.

This paper first proposes a design method for guidance, navigation, and control (GNC) and propulsion systems for a small, lightweight space free-flyer. The proposed design method was implemented in Int-Ball2, and an on-orbit flight performance test was conducted. The results of this orbital test are presented to demonstrate the effectiveness of the proposed method. Finally, this paper compares the ground and orbital test results and evaluates the validity of the ground tests.

II. HARDWARE

A. Propulsion

Int-Ball2 uses a propeller mechanism for propulsion. The mechanism has the advantages of easy thrust variation and compactness. However, it cannot generate negative thrust. Besides, the propeller's drag torque, thrust power efficiency (dependent on the cant angle), and choking must be considered. As a result of determining the entire system, such as mass, dimensions, power resources, and internal usable space, Int-Ball2 was designed with eight propeller modules (PMs) arranged diagonally, as shown in Fig. 1. Clockwise (CW) and counterclockwise (CCW) propeller blades are used in pairs to cancel the angular momentum introduced by the propeller rotation. To avoid choking, the surface slit area is greater than that of Int-Ball.

[1]The authors are with the Research and Development Directorate, Japan Aerospace Exploration Agency (JAXA), 2-1-1 Sengen, Tsukuba, Ibaraki, 305-8505, Japan nishishita.taisei@jaxa.jp

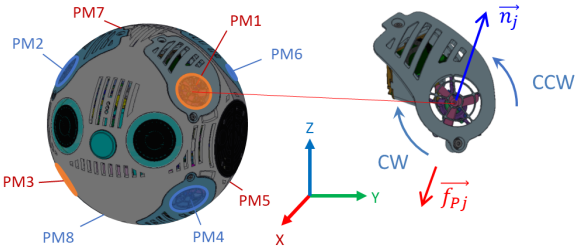


Fig. 1. Int-Ball2 propeller module layout.

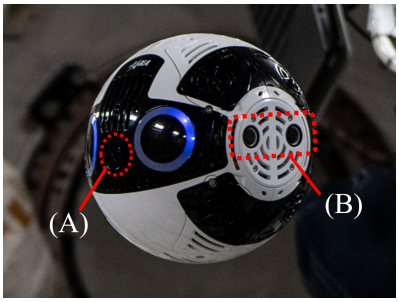


Fig. 2. Appearance of Int-Ball2 flight model flying in the JEM. (A) Main camera. (B) Navigation camera.

Thrust force $\boldsymbol{F}_j$ and the torque around the center of the body, $\boldsymbol{T}_j$, generated by j-th PM, are as follows:

$$\boldsymbol{F}_j = -f_j \boldsymbol{n}_j, \tag{1}$$

$$\boldsymbol{T}_j = -\boldsymbol{r}_j \times f_j \boldsymbol{n}_j - t_j \boldsymbol{n}_j, \tag{2}$$

where f_j is the propeller thrust of j-th PM, $\boldsymbol{r}_j$ is the position vector from the center of gravity to the j-th PM, $\boldsymbol{n}_j$ is the unit normal vector, and t_j is the drag torque. Positive and negative values of t_j are determined by the direction of rotation of the propeller: negative for CW and positive for CCW. Generated force $\boldsymbol{F}$ and torque $\boldsymbol{T}$ of the entire propulsion system are obtained by

$$\boldsymbol{F} = \sum_{j=1}^{N} \boldsymbol{F}_j = \sum_{j=1}^{N} -f_j \boldsymbol{n}_j, \tag{3}$$

$$\boldsymbol{T} = \sum_{j=1}^{N} \boldsymbol{T}_j = \sum_{j=1}^{N} \{-f_j(\boldsymbol{r}_j \times \boldsymbol{n}_j) - t_j \boldsymbol{n}_j\}. \tag{4}$$

We consider a model in which the generated thrust is proportional to the square of the rotation speed and the drag torque is proportional to the generated thrust. For simplicity, we also assume that the rotation speed is proportional to the duty ratio of the drive signal. Thus, using D_j as the propeller drive duty of j-th PM yields

$$f_j = k_j D_j^2, \tag{5}$$

$$t_j = \kappa_j D_j^2. \tag{6}$$

The scale factor coefficients k_j and κ_j were obtained experimentally by measuring the thrust of each PM while varying D_j and then curve fitting the obtained results.

III. NAVIGATION

Int-Ball2 adopts Visual SLAM (vSLAM) with a stereo camera for its navigation system. This allows the system to extract feature points directly from the surrounding environment without needing external markers on the JEM wall. Int-Ball2 uses SLAM instead of a pure localization method depending on predefined specific features because it will operate in a dynamic environment. In actual operations, the arrangement of objects on board the JEM (cargo transfer bags, experimental equipment, for example) can change often, so we assumed that using a pre-built map would not always work well for localization. The implemented vSLAM algorithm is based on ORB-SLAM2 and optimized for Int-Ball2 hardware. ORB-SLAM2 extracts the Oriented FAST and Rotated BRIEF (ORB) features that are invariant to changes in viewpoint and illumination [7]. The ORB features are used for all SLAM tasks, including tracking, local mapping, and loop closing. These processes can be executed in parallel, enabling real-time estimation. The stereo camera for navigation is mounted on the robot's side, as shown in Fig. 2. Since Int-Ball2 is a camera robot, there is a high possibility of a moving object, such as a crew member, in front of the main camera. The camera localization may suffer if the feature points on the JEM wall are hidden by moving objects. Therefore, we decided to mount the stereo camera not at the front of the robot but at the side where the wall surface can be easily observed. Int-Ball2 also incorporates an IMU. The vSLAM and the IMU measurements are fused using a Kalman filter, and the result is used as the final navigation value.

IV. CONTROL

A. Dynamics Model

First, the Equations of Motion (EOMs) assumed for the control system design are explained. We define a fixed frame $\mathcal{B}$ with the body's center of mass as the origin and a frame $\mathcal{I}$ fixed in the ISS. Since the ISS orbits the Earth, coordinate $\mathcal{I}$ is not strictly an inertial system, but the inertial force acting on the robot is small, so it is treated as a disturbance in this paper. Denoting the attitude of frame $\mathcal{B}$ from frame $\mathcal{I}$ by the quaternion $\boldsymbol{q} = [\boldsymbol{q}_v \; q_s]^T$ ($\boldsymbol{q}_v$: vector part, q_s: scalar part), the attitude kinematics is given by

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{q} \cdot \begin{bmatrix} \boldsymbol{\omega} \\ 0 \end{bmatrix}, \tag{7}$$

where $\boldsymbol{\omega}$ is the angular velocity and the operator $(\cdot)$ is the quaternion product. The robot's dynamics is obtained using the Newton-Euler equation.

$$m\ddot{\boldsymbol{r}} = \boldsymbol{F}_{cmd} + \boldsymbol{F}_{dist}, \tag{8}$$

$$\boldsymbol{I}_s\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times \boldsymbol{I}_s\boldsymbol{\omega} + \boldsymbol{T}_{cmd} + \boldsymbol{T}_{dist}, \tag{9}$$

where m is the mass and $\boldsymbol{I}_s$ is the inertia tensor. $\boldsymbol{F}_{cmd}$ and $\boldsymbol{T}_{cmd}$ are the force and torque control inputs, respectively. $\boldsymbol{F}_{dist}$ and $\boldsymbol{T}_{dist}$ are the force and torque disturbances. $\boldsymbol{r}$ is the position vector of the robot in inertial frame $\mathcal{I}$. Then, f_j is represented as a model that responds to the control

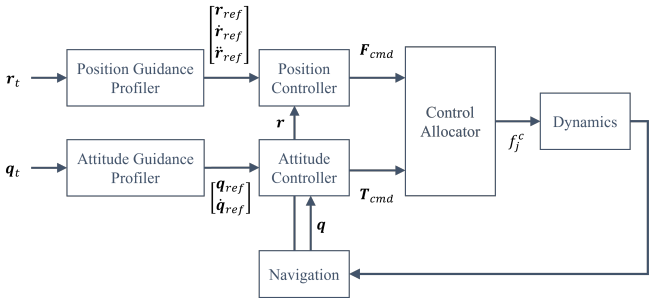
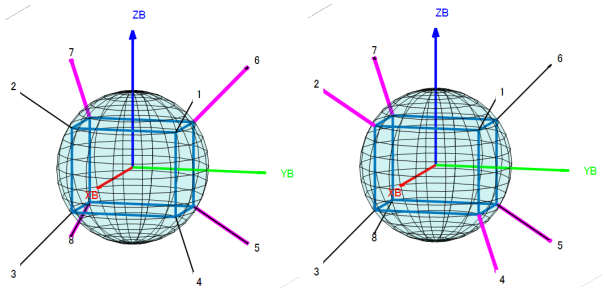


Fig. 3. Block diagram of the GNC architecture.



(a) +X Translation (b) +X Rotation

Fig. 4. Example of PM selection for each axis movement.

command value f_j^c with a first-order lag process with time constant τ_j as follows:

$$\dot{f}_j = \frac{1}{\tau_j}(f_j^c - f_j). \tag{10}$$

B. Guidance and Control Law

The basic architecture of the guidance control system is a 2-DOF control system, as shown in Fig. 3. Feedforward control ensures immediate target tracking, while feedback control suppresses errors between the current and target positions to maintain stability. The translational and rotational dynamics are separated by (8) and (9), allowing independent guidance control systems to be designed. First of all, the position guidance profile ($\boldsymbol{r}_{ref}$, $\dot{\boldsymbol{r}}_{ref}$, $\ddot{\boldsymbol{r}}_{ref}$) to reach target position $\boldsymbol{r}_t$ from the initial position and attitude guidance profile ($\boldsymbol{q}_{ref}$, $\dot{\boldsymbol{q}}_{ref}$) to reach target attitude $\boldsymbol{q}_t$ from the initial attitude are generated, respectively. Feedback control is then performed to follow the generated profiles. The control law is designed based on the method proposed in [8]. For each desired control force and torque obtained in each control loop, the desired thrust of each PM, f_j, is calculated and converted to PWM signals to drive each PM.

1) Position Control: Int-Ball2 uses PID control as a feedback compensator. Defining the error between the position guidance profile and the current position as $\boldsymbol{r}_e \equiv \boldsymbol{r} - \boldsymbol{r}_{ref}$, the control input $\boldsymbol{F}_{cmd}$ is as follows:

$$\boldsymbol{F}_{cmd} = m\left(\ddot{\boldsymbol{r}}_{ref} - k_p\boldsymbol{r}_e - k_i\int \boldsymbol{r}_e \mathrm{d}t - k_d\dot{\boldsymbol{r}}_e\right), \tag{11}$$

where k_p, k_d and k_i are the proportional, derivative and integral gains, respectively. Here further define k_p and k_d as follows:

$$k_p \equiv \omega_n^2 + 2k_i\zeta/\omega_n, \tag{12}$$

$$k_d \equiv 2\zeta\omega_n + k_i/\omega_n^2, \tag{13}$$

where ω_n and ζ are constants. Substituting (11), (12), and (13) into (8) and performing the Laplace transform assuming a one-dimensional system for simplicity, the following open-loop transfer function $G_o(s)$ from the guidance profile to the current position is obtained.

$$G_o(s) = \frac{2\zeta\omega_n s + \omega_n^2}{s^2} + \frac{k_i}{\omega_n^2}\cdot\frac{s^2 + 2\zeta\omega_n s + \omega_n^2}{s^3}. \tag{14}$$

Each parameter was set to satisfy the desired gain and phase margins.

2) Attitude Control: The feedback compensator for attitude is given by

$$\boldsymbol{T}_{cmd} = \frac{1}{\tau_\omega}\boldsymbol{I}_s(\boldsymbol{\omega}_c - \boldsymbol{\omega}) + \boldsymbol{\omega}\times\boldsymbol{I}_s\boldsymbol{\omega}. \tag{15}$$

The second term on the right side removes the nonlinear terms in (9). Substituting (15) into (9), the equation can be written as a first-order lag system with time constant τ_ω as follows:

$$\dot{\boldsymbol{\omega}} = \frac{1}{\tau_\omega}(\boldsymbol{\omega}_c - \boldsymbol{\omega}). \tag{16}$$

Here, the error quaternion is defined by

$$\boldsymbol{q}_e \equiv \begin{bmatrix}\boldsymbol{q}_{ev}\\ q_{es}\end{bmatrix} = \begin{bmatrix}\hat{\boldsymbol{\alpha}}\sin(\epsilon/2)\\ \cos(\epsilon/2)\end{bmatrix} = \boldsymbol{q}_{ref}^{-1}\cdot\boldsymbol{q}, \tag{17}$$

where $\hat{\boldsymbol{\alpha}}$ is the unit vector and ϵ is the rotation angle. The following equation finds $\boldsymbol{\omega}_c$.

$$\boldsymbol{\omega}_c = -\frac{2\tau_\omega}{\tau_{att}^2}\mathrm{sgn}(q_{es})\boldsymbol{q}_{ev} + \boldsymbol{\omega}_{ref}. \tag{18}$$

The first term on the right side is the feedback term; the second is the feedforward one. τ_{att} is a constant, and $\boldsymbol{\omega}_{ref}$ is the angular velocity vector of the attitude guidance profile, which can be derived from (7). Assuming that ϵ is small and can be approximated as $\boldsymbol{q}_{ev} \approx \epsilon\hat{\boldsymbol{\alpha}}/2$, we obtain

$$\ddot{\epsilon} + \frac{1}{\tau_\omega}\dot{\epsilon} + \frac{1}{\tau_{att}^2}\epsilon = 0. \tag{19}$$

Replacing $\omega_n' \equiv 1/\tau_{att}$ and $2\zeta'\omega_n' \equiv 1/\tau_\omega$, the open-loop transfer function of the attitude control system corresponds to the first term of (14) for the position control system.

C. Control Allocation

The propulsion system consists of eight PMs arranged symmetrically, making it easy to identify the propeller selections that can move in each axis. Fig. 4 shows an example of the combination of PMs that can be driven to generate translational force in +X axis and torque around +X axis. The black vectors represent PMs that is not driven, and the magenta vectors represent driven PMs. By driving a predetermined PM, translational force and torque in the desired direction can be output.

In practice, however, individual differences in the thrust of each PM can cause forces and torques to be generated in

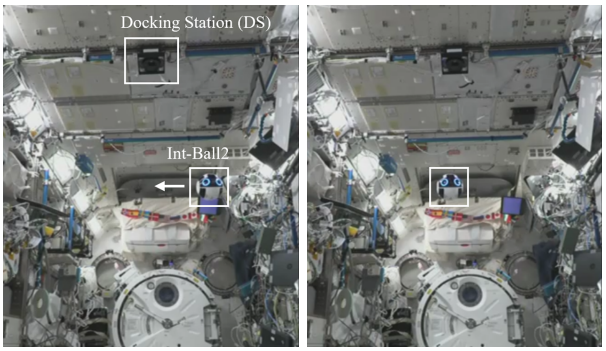


Fig. 5. Int-Ball2 flying inside the JEM during the on-orbit checkout.

directions other than the desired one. Therefore, Int-Ball2 uses a method similar to the sweep-out method to make minor modifications to the propeller assignment matrix. The combination of PMs that should be driven when generating translational force or torque in the direction of each axis is stored in advance as array matrices $\boldsymbol{W}^+ \in R^{8\times6}$ and $\boldsymbol{W}^- \in R^{8\times6}$. Note that the PMs cannot generate thrust in the reverse direction and must have $f_j > 0$, so positive array matrix $\boldsymbol{W}^+$ and the negative array matrix $\boldsymbol{W}^-$ are used independently. Given control translation force and torque $\boldsymbol{u}^T = [\boldsymbol{F}_{cmd}^T \ \boldsymbol{T}_{cmd}^T]$ by the 2-DOF controller, thrust $\boldsymbol{f} = [f_1 \ \cdots \ f_8]^T$ of each PM to output the desired control volume is then calculated as follows:

$$\boldsymbol{f} = \boldsymbol{W}^+(\boldsymbol{p} * \boldsymbol{u}) + \boldsymbol{W}^-\{\bar{\boldsymbol{p}} * (-\boldsymbol{u})\}, \tag{20}$$

where $\boldsymbol{p} \in R^{6\times1}$ is the polarity index (logical array), e.g., $\boldsymbol{p} = [1\ 0\ 0\ 1\ 1\ 0]^T$ means that the thrust in X direction and the torque in X and Y directions are positive. The operator $(*)$ is the multiplication of the components of vectors of the same dimension, and operator $\bar{\boldsymbol{p}}$ is the negation of logical array $\boldsymbol{p}$.

V. ON-ORBIT PERFORMANCE ANALYSIS

A. Checkout Operation Overview

Since Int-Ball2 was launched in 2023, the robot has been tested for basic functions, such as operating the main camera and microphone. Subsequently, in October of the same year, the checkout for the autonomous flight function was performed, and Int-Ball2 made its first flight in the JEM (Fig. 5). This section describes the results of the quantitative evaluation of the mobility performance based on the flight data obtained during the checkout.

The reference pose for evaluating navigation accuracy was obtained using visual markers (EN marker by LEAG Solutions Corp.), as shown in Fig. 6. The EN marker is an extension of the LentiMark [9] developed by the National Institute of Advanced Industrial Science and Technology (AIST) and is planar yet achieves high measurement accuracy by making use of moiré patterns that consist of a micro lens array and stripe pattern, which vary in appearance according to the visual-line angle of observation. Before checking out, EN markers were placed at 12 locations inside the JEM. By downlinking the marker image taken by the onboard main camera and processing it afterward on the ground, we can measure the pose of Int-Ball2 with respect to the marker. Based on this and the marker installation pose information, the pose of Int-Ball2 in the docking station (DS) frame is calculated. The DS is a charging device dedicated for Int-Ball2, which is connected to the robot by magnetic force [10]. Note that the measured value by the marker includes the measurement error specified by the marker specifications (see Table I) and the marker alignment error.

TABLE I
EN MARKER SPECIFICATION.

Item	Value
Marker size	44 mm
Position accuracy (horizontal)	$\leq \pm 0.01\%$
Position accuracy (depth)	$\leq \pm 0.1\%$
Attitude accuracy	$\leq \pm 1\%$
Measurable angular range	± 60 deg

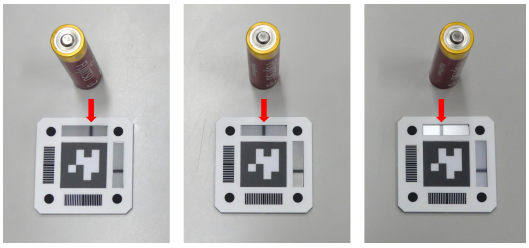


Fig. 6. EN marker appearance (The moiré pattern appears to change depending on the viewing angle).

In the standby state, the robot is installed and docked at the DS, as shown in Fig. 7. Each coordinate axis in the figure represents the fixed coordinate system of the DS. With Int-Ball2 attached to the DS, the +X axis corresponds to the shooting direction of the main camera, the +Y axis to the direction of the navigation camera, and the +Z axis to the zenith. The test scenario executed in checkout is shown in Table II. The attitude is expressed in 3-2-1 Euler angles (Yaw: ψ, Pitch: θ, Roll: ϕ). To evaluate the control performance in all axes, translational movement control of ±200 mm was first applied in each axis, followed by attitude rotation control of ±30 deg about each axis.

Fig. 8 shows the Int-Ball2 on-orbit flight results. The blue

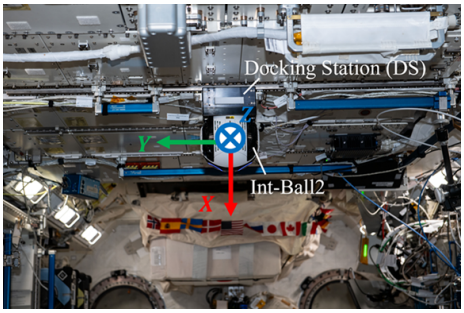


Fig. 7. On-orbit test configuration and definition of the DS coordinate frame.

TABLE II

ORBITAL CHECKOUT TASK SCENARIO.

No.	Task	Description
1	HP-1	Keep pose at Home Position 1 (HP-1) $(X: 920, Y: 0, Z: 500)$
2	$+X$ Trans.	Move 200 mm in $+X$
3	$-X$ Trans.	Move 200 mm in $-X$
4	$+Y$ Trans.	Move 200 mm in $+Y$
5	$-Y$ Trans.	Move 200 mm in $-Y$
6	$+Z$ Trans.	Move 200 mm in $+Z$
7	$-Z$ Trans.	Move 200 mm in $-Z$
8	$+\phi$ Rot.	Rotate +30 deg in roll
9	$-\phi$ Rot.	Rotate -30 deg in roll
10	$+\psi$ Rot.	Rotate +30 deg in yaw
11	$-\psi$ Rot.	Rotate -30 deg in yaw
12	HP-2	Move to Home Position 2 (HP-2) $(X: 920, Y: 0, Z: 1000)$
13	$+\theta$ Rot.	Rotate +30 deg in pitch
14	$-\theta$ Rot.	Rotate -30 deg in pitch

line in the figure is the navigation value, and the dotted black line is the guidance profile. The robot can move stably along the desired guidance profile without overshooting. The three-dimensional flight trajectory of the robot is illustrated in Fig. 9. The blue line is the estimated trajectory by navigation, the red line is the trajectory measured with the EN markers, and the dotted black line is the guidance profile. Note that the EN markers are only a reference to check the validity of the navigation values, and the measurement errors are not small enough to be treated as ground truth. The EN markers have a small bias error because they are fixed in the JEM but have more noise than the robot's navigation results. As shown in Fig. 9, the navigation values in flight were approximately with a constant bias error. As the accuracy of the visual odometry tends to deteriorate as the distance traveled increases, the generated maps become distorted. The starting point of the test, HP-1, was already more than 1 m away from the DS, and some navigation bias error was already present when the robot was flying at HP-1.

The results of the error evaluation for each flight task are shown in Tables III and IV. Here, control error is the error between the guidance profile and the navigation value when moving control is performed and represents the steady-state characteristics of the guidance and control system. Navigation error is the error between the navigation and EN marker values. The maximum errors in the control of position and attitude were 3.14 mm and 0.36 deg, which indicates that it tracked the target pose accurately. In contrast, the maximum errors in navigation of position and attitude were 91.64 mm and 1.87 deg. Although the navigation error is relatively large compared to the control error, precise localization is not so important to accomplish the shooting task and is sufficient for the mission requirement. On the other hand, SLAM lost is one of the factors that can cause mission failure, so the stability of the navigation system is particularly important. From this perspective, Int-Ball2's navigation system has sufficient performance, as it has always been able to estimate its pose without divergence during the checkout period.

The time history of the Int-Ball2 control input is shown in Fig. 10. Compared with Fig. 8, a large amount of force

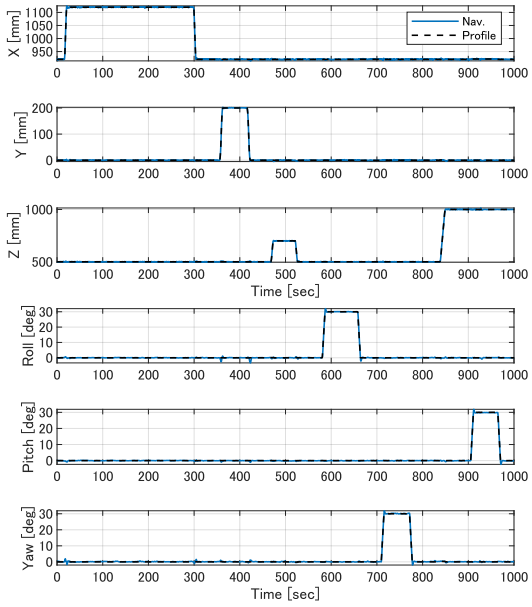


Fig. 8. Time history of the pose of the Int-Ball2 (blue line: navigation value, dotted black line: guidance profile).

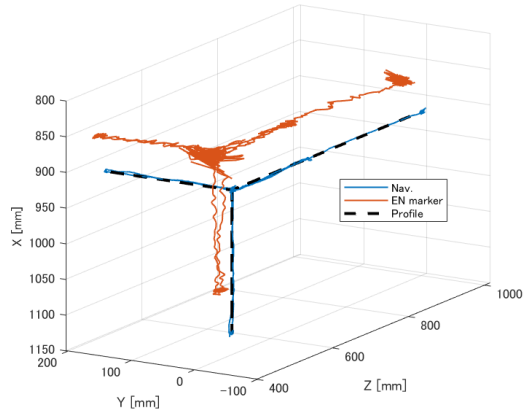


Fig. 9. Flight trajectory during the checkout (blue line: trajectory estimated by the navigation system, dotted black line: guidance profile, red line: trajectory measured with the EN markers)

and torque commands were observed at the start and end of the pose control. Some forces and torques at axes different from the direction of the commanded motion were generated, such as roll axis torque T_x during translational movement in the Y-axis direction. This was probably because the propeller assignment matrix, whose parameters were adjusted on the ground, was imperfect in the JEM environment. There was no significant amount of control during the keeping pose mode although disturbance forces such as air flow inside the JEM would have constantly acted on the robot. The results of evaluating the degree to which the control commands from the control system correspond to the driving force applied to the robot are shown in Fig. 11. The blue line in the figure represents the desired translational command, and the red line is the translational force applied to the robot, calculated simply by multiplying the mass by the acceleration measured by the robot's internal IMU. These values were consistent,

TABLE III

POSE ERROR IN TRANSLATIONAL MOVEMENT IN ON-ORBIT TESTING.

Task	Control error		Navigation error	
	Pos. [mm]	Att. [deg]	Pos. [mm]	Att. [deg]
$+X$ Trans.	1.65	0.17	78.99	1.29
$-X$ Trans.	1.34	0.15	75.26	1.19
$+Y$ Trans.	1.25	0.14	71.67	0.93
$-Y$ Trans.	1.30	0.16	73.89	1.00
$+Z$ Trans.	1.25	0.20	82.22	1.14
$-Z$ Trans.	1.11	0.15	73.45	1.13

TABLE IV

POSE ERROR IN ROTATIONAL MOVEMENT IN ON-ORBIT TESTING.

Task	Control error		Navigation error	
	Pos. [mm]	Att. [deg]	Pos. [mm]	Att. [deg]
$+\phi$ Rot.	0.97	0.11	76.43	1.58
$-\phi$ Rot.	1.27	0.19	74.15	1.34
$+\theta$ Rot.	3.14	0.36	91.64	1.68
$-\theta$ Rot.	0.72	0.20	84.21	1.87
$+\psi$ Rot.	1.55	0.21	61.05	0.71
$-\psi$ Rot.	1.11	0.15	72.92	0.68

confirming that the robot could output the desired control force along the trajectory.

The time histories of the linear velocity and angular velocity for the movement control along each axis are shown in Figs. 12 and 13, respectively. Int-Ball2 has a maximum translational velocity of 50 mm/s and a maximum angular velocity of 5 deg/s. It is designed to generate a trapezoidal velocity profile that meets these specifications (dotted black line). It was confirmed that the navigation values (blue line) follow the generated guidance profile in all axial directions.

B. Comparison Between Orbital Checkout and Ground Test

This section describes the ground test results of GNC performance using the Int-Ball2 flight model and the air

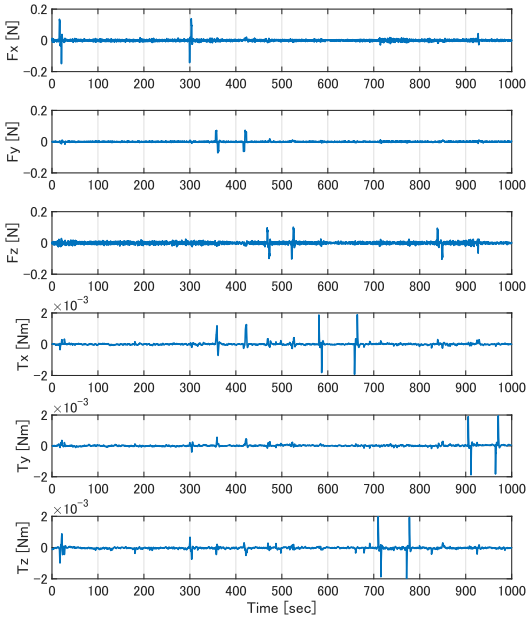


Fig. 10. Time history of the control command force and torque.

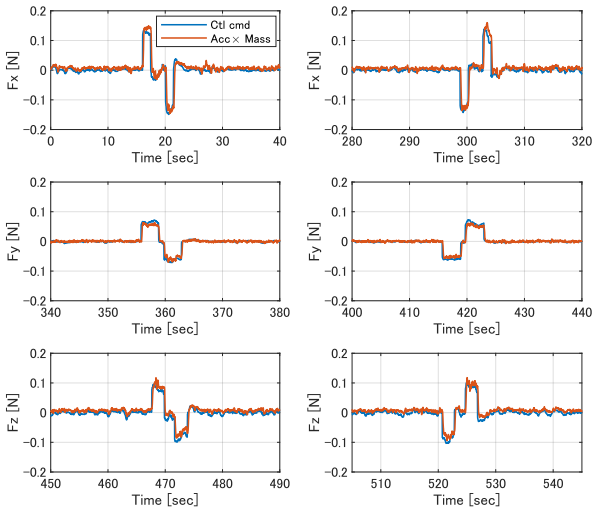


Fig. 11. Comparison of control force commands and actual driving force (blue line: control command, red line: approximate thrust obtained by multiplying IMU acceleration by mass).

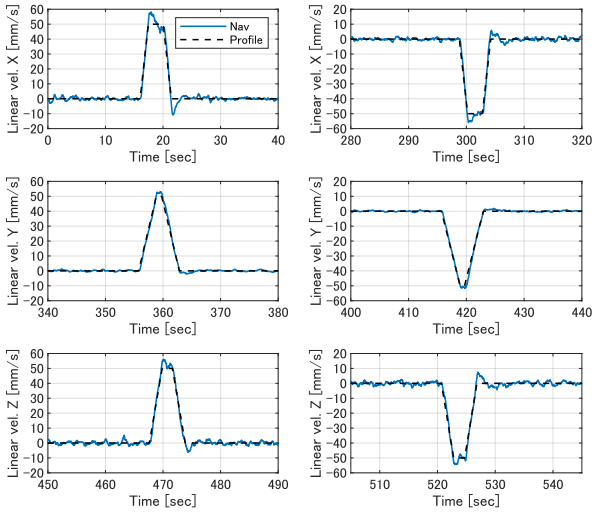


Fig. 12. Time history of the linear velocity (blue line: navigation value, dotted black line: guidance profile).

bearing platform. Since it is difficult to accurately simulate the three-dimensional behavior of a robot in microgravity on the ground, ground tests are conducted by partially simulating the environment. For effective ground testing, it is especially important to carefully consider what to simulate and how accurately. A more comprehensive ground test strategy for Int-Ball2 is described in [11]. In addition to the quantitative evaluation of flight performance, the extent to which the ground test simulated on-orbit behavior and the validity of the test method are also discussed against the on-orbit flight results.

The test configuration using the air bearing platform is illustrated in Fig. 14. Photo panels that imitate the inside of the JEM were installed on the wall to enable localization using the navigation camera. Int-Ball2 is fixed on a jig on which the air bearing is mounted and can move smoothly on the granite table. The granite table is 1500 mm wide and

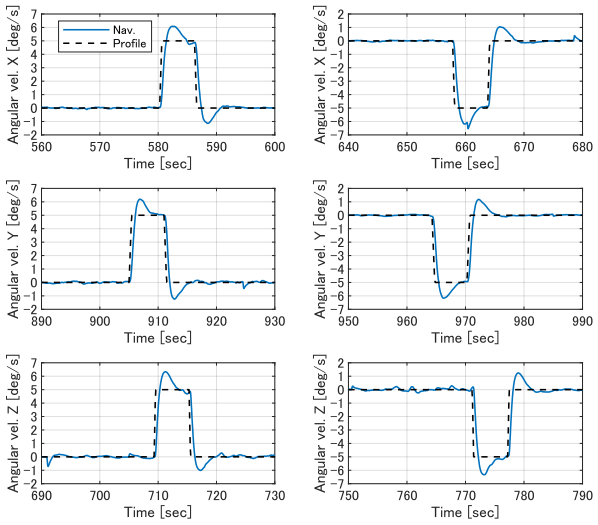


Fig. 13. Time history of the angular velocity (blue line: navigation value, dotted black line: guidance profile).

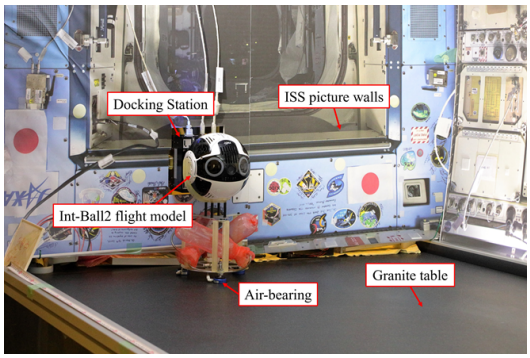


Fig. 14. Ground test configuration with air bearing.

3000 mm long, allowing the robot to move over a wide area. Five motion capture cameras are mounted on the aluminum frame, and reflective markers are attached to the robot to obtain the ground truth of the robot's pose. This motion capture system can obtain accurate reference values within 1 mm depending on the capture environment. The test on the actual device by air bearing is limited because it can only simulate movements in a two-dimensional plane. Therefore, to verify all 6-DOF position and attitude controls, tests were conducted with three patterns of attachment attitudes, as shown in Table V.

Fig. 15 shows the ground test results for each test case. The movement amounts were matched for easy comparison with the results of the on-orbit checkout. 200 mm of translation and 30 deg of rotation in each axis direction were used for movement control. The blue line represents the navigation values, and the dotted black line represents the guidance profile, indicating that Int-Ball2 achieved flight control following the guidance profile. When Int-Ball2 attempted to move in a certain axis direction, some translational and rotational behaviors were observed in other directions. Compared to the on-orbit results shown in Fig. 8, this phenomenon is more pronounced in the ground test. Reasons for the different

TABLE V

INT-BALL2 ORIENTATION SETUP FOR AIR BEARING TEST

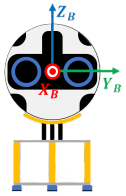
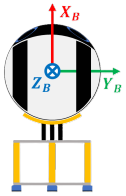
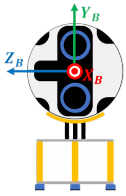
Test case 1	**Test case 2**	**Test case 3**
		
Trans.: X/Y axis Rot.: Yaw axis	Trans.: Y/Z axis Rot.: Roll axis	Trans.: Z/X axis Rot.: Pitch axis

TABLE VI

POSE ERROR IN TRANSLATIONAL MOVEMENT IN GROUND TESTING.

Task	Control error		Navigation error	
	Pos. [mm]	Att. [deg]	Pos. [mm]	Att. [deg]
$+X$ Trans.	7.27	2.45	16.28	0.37
$-X$ Trans.	5.11	1.63	7.44	0.29
$+Y$ Trans.	4.46	1.02	11.47	1.09
$-Y$ Trans.	11.30	1.35	14.18	1.22
$+Z$ Trans.	6.39	0.36	8.04	0.05
$-Z$ Trans.	8.92	0.76	10.77	0.06

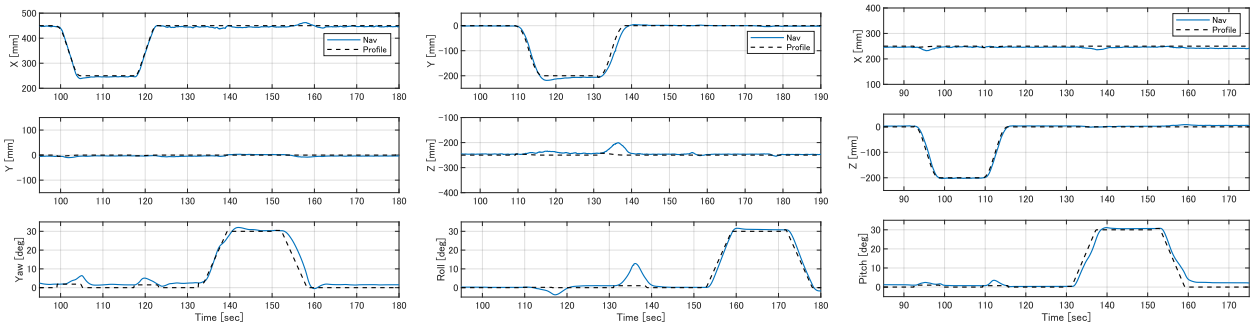
behavior from the on-orbit checkout results include effects such as friction between the granite table and air bearings and changes in the mass characteristics of the entire system due to the mass of the test jig. Int-Ball2's control performance is sensitive to external factors because of its low thrust power. Even if disturbance factors are eliminated as much as possible in the ground test, there will inevitably be a larger disturbance than on orbit. In other words, regarding control robustness, the test conditions on the ground were more conservative than in orbit. Therefore, the robot can fly stably if the control stability is shown in the ground test, indicating that the proposed ground test is effective as a pre-flight verification.

Tables VI and VII show the accuracy evaluation results for the ground test. The control error resulted in a maximum position error of 10.48 mm and a maximum attitude angle error of 2.45 deg. Although the control performance is sufficient, the control accuracy is poorer than actual on-orbit data. On the other hand, the position and attitude errors of the navigation system were smaller than the on-orbit test results, with a maximum error of 20.41 mm and 1.22 deg, respectively. This is thought to be due to the short relative distance from the DS caused by the size

TABLE VII

POSE ERROR IN ROTATIONAL MOVEMENT IN GROUND TESTING.

Task	Control error		Navigation error	
	Pos. [mm]	Att. [deg]	Pos. [mm]	Att. [deg]
$+\phi$ Rot.	3.81	1.43	3.80	1.15
$-\phi$ Rot.	5.20	1.73	12.05	1.06
$+\theta$ Rot.	5.33	0.90	8.33	0.67
$-\theta$ Rot.	10.48	2.25	7.36	0.20
$+\psi$ Rot.	5.56	0.68	13.60	0.18
$-\psi$ Rot.	5.46	1.58	20.41	0.50



(a) Test case 1 (b) Test case 2 (c) Test case 3

Fig. 15. Position and attitude control results in ground tests.

limitation of the granite table, which resulted in a smaller vSLAM drift error than in the on-orbit test. In addition, the air bearing test acquired ground truth using high-precision motion capture, while the on-orbit test used EN markers due to installation costs and other factors. It should be noted that there are differences in the accuracy of the reference values as well. For these reasons, it is assumed that the navigation performance did not deteriorate on orbit, and the differences in evaluation results were simply due to the limitations of the test system. These results confirmed that the orbital flight performance did not deteriorate from that on the ground, indicating that the ground test method was effective as a pre-flight verification.

VI. CONCLUSIONS

This paper introduced the propulsion mechanism and GNC algorithms that realize the autonomous flight function of Int-Ball2. This robot has unique features compared to other space free-flying robots, including its small size for use in crowded environments and its high thrust despite its size. We conducted orbital flight performance tests of Int-Ball2 inside the JEM. The evaluation results showed high control accuracy, with a maximum control error of 3.14 mm for position and 0.36 deg for attitude. The maximum navigation error was 91.64 mm for position and 1.87 deg for attitude, indicating sufficient performance for photographic tasks. The robot was thoroughly tested on the ground in advance to evaluate its mobility performance. The results were compared to orbital flight test data and showed the validity of the ground testing method. We will continue to verify the operation of these off-nominal cases on orbit, eventually making it useful for actual crew work.

ACKNOWLEDGMENT

We would like to thank all our partners involved in the design, development, and operation of Int-Ball2. We would especially like to thank the members of JAXA Human Spaceflight Technology Directorate for their leadership in promoting the project and conducting the on-orbit checkout.

REFERENCES

[1] S. Mitani, M. Goto, R. Konomura, Y. Shoji, K. Hagiwara, S. Shigeto, and N. Tanishima, Int-Ball: Crew-Supportive Autonomous Mobile Camera Robot on ISS/JEM, 2019 IEEE Aerospace Conference, Big Sky, MT, USA, 2019.

[2] D. Hirano, S. Mitani, T. Nishishita, and T. Saito, Int-Ball2 for fully-teleoperated JEM onboard camera drone without crew aid, 71st International Astronautical Congress (IAC), The CyberSpace Edition, 2020.

[3] T. Smith, J. Barlow, M. Bualat, T. Fong, C. Provencher, H. Sanchez, E. Smith, and the Astrobee Team, Astrobee: A New Platform For Free-Flying Robotics on the International Space Station, International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS), Beijing, China, 2016.

[4] A. M. Vargas, R. G. Ruiz, P. Wofford, V. Kumar, B. V. Ross, A. Katterhagen, J. Barlow, L. Flückiger, J. Benavides, T. Smith, and M. Bualat, Astrobee: Current Status and Future Use as an International Research Platform, 69th International Astronautical Congress (IAC), Bremen, Germany, 2018.

[5] R. Regele, T. Stach, J. Sommer, and T. Eisenberg, Cimon: A Visual Navigation System for Flying Through the International Space Station, 69th International Astronautical Congress (IAC), Bremen, Germany, 2018.

[6] V. Schrödera, R. Regeleb, J. Sommera, T. Eisenbergc, C. Karraschd, GNC System Design for the Crew Interactive MObile Companion (CIMON), 69th International Astronautical Congress (IAC), Bremen, Germany, 2018.

[7] R. Mur-Artal and J. D. Tardos, ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras, IEEE Transactions on Robotics (T-RO), vol. 33, no. 5, pp. 1255 – 1262, 2017.

[8] D. Brescianini, and R. D'Andrea, Design, Modeling and Control of an Omni-Directional Aerial Vehicle, 2016 IEEE International Conference on Robotics and Automation (ICRA), pp. 3261-3266, Stockholm, Sweden, 2016.

[9] H. Tanaka, Y. Sumi, and Y. Matsumoto, A visual marker for precise pose estimation based on lenticular lenses, IEEE International Conference on Robotics and Automation (ICRA), pp. 5222-5227, Saint Paul, MN, USA, 2012.

[10] K. Watanabe, Magnetic Docking Mechanism for Free-flying Space Robots with Spherical Surfaces, 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 5994-5999, Paris, France, 2020.

[11] T. Nishishita, D. Hirano, K. Watanabe, and S. Mitani, Complementary Ground Testing Method for Autonomous Flight System of Space Free-Flying Robot, 2024 IEEE Aerospace Conference, Big Sky, MT, USA, 2024.