2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

# Hardware-in-the-Loop Simulator with Low-Thrust Actuator for Free-Flying Robot's Omni-Directional Control

Daichi Hirano[1], Shinji Mitani[1], Taisei Nishishita[1], and Tatsuhiko Saito[2]

***Abstract*— Small free-flying robots to assist astronauts and perform experiments need a propulsion system to move freely in microgravity. Hardware-in-the-loop (HIL) simulators can simultaneously verify guidance, navigation, and control (GNC) systems, including flight hardware and software, in three dimensions. However, it is difficult to incorporate a small free-flying robot into the HIL simulator because of the low propulsive force and gravity compensation associated with its attitude changes. This paper proposes a HIL simulator with a propulsion subsystem mounted on a statically fixed force/torque sensor and a GNC subsystem mounted on a dynamically movable robotic arm. This simulator allows us to verify the GNC algorithms comprehensively using actual navigation sensors and propulsive actuators in an emulated flight environment. The actual capabilities of this simulator were successfully demonstrated in motion verifications of a free-flying robot, the Int-Ball2.**

## I. INTRODUCTION

Several free-flying robots have been developed to assist astronauts and perform experiments in the pressured sections of the International Space Station (ISS). The performance of a robot's guidance, navigation, and control (GNC) is initially verified by ground tests. However, existing test facilities have various restrictions. An air-bearing platform on a flat granite table that simulates planar frictionless motion in microgravity ignores any coupling force and torque in three dimensions. Gravity compensation using a counterweight restricts motion, and an accurate simulation is difficult due to mechanical friction. A computer-based simulator can easily verify three-dimensional motion in microgravity, although complex aerodynamics and hardware uncertainties complicate modelling and decrease the accuracy of simulated motion. In contrast, a hardware-in-the-loop (HIL) simulator can emulate three-dimensional motion without such complex modeling using a dynamics calculator with a sensor to measure actual exerting forces. However, a HIL simulator for a space application needs active gravity compensation to calculate free-flying motion in microgravity. Such gravity compensation is hard to perform accurately for small free-flying robots because it requires extremely high precision to account for robot's low control forces.

This paper proposes a HIL simulator with a propulsion subsystem mounted on a statically fixed force/torque sensor and a GNC subsystem including navigation sensors mounted on a dynamically movable robotic arm. The force/torque sensor accurately measures actual physical propulsive force without complex gravity compensation. The robotic arm moves the GNC subsystem to a simulated position and attitude to perform navigation in a simulated environment. This HIL simulator allows us to verify the GNC system, including the flight hardware and software, simultaneously in three dimensions. This has been limited in previous ground facilities. The remaining sections of this paper introduce the details of the HIL simulator system and describe a strategy for ground tests, including this simulator and the classical approach of air-bearing platforms and a computer simulation. Additionally, we present the demonstration results of the proposed simulator. This demonstration was performed in a mockup field of the ISS's Japanese Experimental Module (JEM) to verify the GNC algorithms of a free-flying robot, the Int-Ball2, as shown in Fig. 1, which will be launched to the ISS.

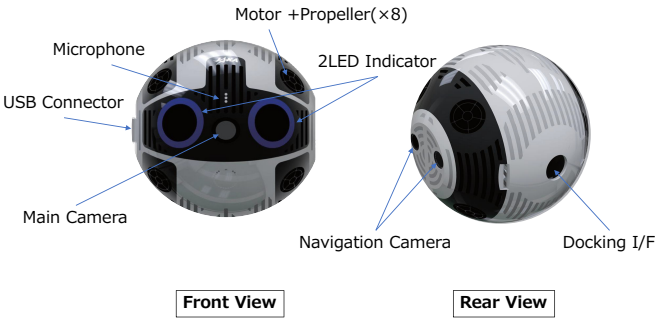



Fig. 1. Int-Ball2's appearance and hardware configuration.

## II. RELATED WORKS

### A. Small Free-Flying Robots

The first free-flying robotic inspection vehicle in space was the Miniature Autonomous Extravehicular Robotic Camera (Mini AERCam), which successfully demonstrated its capabilities of monitoring astronauts during extravehicular activities (EVAs) on a Shuttle flight in 1997 [1]. This robot was updated to the Synchronized Position Hold Engage and Reorient Experimental Satellites (SPHERES) for testing flight technologies during intravehicular activities (IVAs) [2]. Recently, NASA also developed its next-generation robot, Astrobee, which can recharge batteries and autonomously dock and undock [3]. These robots were designed as flexible platforms for various tests of guidance, navigation, and control algorithms, including formation flights, rendezvous, and docking in the ISS and student competitions. In contrast,

[1]D. Hirano, S. Mitani and T. Nishishita are with the Research and Development Directorate, Japan Aerospace Exploration Agency (JAXA), 2-1-1 Sengen, Tsukuba, Ibaraki, 305-8505, Japan. hirano.daichi@jaxa.jp, mitani.shinji@jaxa.jp, nishishita.taisei@jaxa.jp

[2]T. Saito is with the System Development Department, Systems Engineering Consultants Co., Ltd., 4-10-1 Yoga, Setagaya, Tokyo, 158-0097, Japan. tt-saito@sec.co.jp

DLR and Airbus developed a social robot named the Crew Interactive MObile companioN (CIMON), which informs astronauts by showing procedures on a screen and through voice interaction [4].

JAXA also developed the JEM Internal Ball Camera (Int-Ball), which autonomously navigated and took photos and videos under teleoperation control [5]. The aim of this robot was to eliminate the astronauts' time spent taking photos, which could otherwise consume approximately 10 % of total crew resources. The first Int-Ball was launched to the ISS in 2017, and the new updated Int-Ball (Int-Ball2) has been developed as a fully teleoperated system that requires no assistance from the astronauts [6]. This robot has two cameras: a main camera for monitoring and shooting photos and videos, and a navigation camera, as shown in Fig. 1. Eight motors and propellers are also installed in this robot to move freely in pressured sections in the ISS. This study chose Int-Ball2 as a target platform, but our proposed HIL simulator can be applied to all the small free-flying robots described above.

### B. Ground Facilities

An air-bearing platform on a flat granite table is often used in ground tests to simulate planar motion in a microgravity environment [7]. In the Astrobee project, an air-bearing testbed with ISS mockup walls around the table was used to demonstrate the visual localization algorithm in two dimensions [8]. In addition, a gantry system with a gimbal that allows 6-DoF motion was employed in ground verification of computer vision-based control for path planning and obstacle avoidance [9]. However, this test facility cannot simulate dynamic motion precisely using actual thrust force or torque since the motion is based on a propulsion command model.

HIL simulators for space applications have also been studied in several missions. The OOS-SIM in DLR can simulate orbital servicing tasks using industrial robotic arms [10]. DLR also has a facility with two 6-DoF industrial robots called EPOS to simulate rendezvous and docking [11]. GMV has a dynamic test facility platform-art© for an end-to-end on-ground demonstration [12]. Although these facilities can simulate dynamic motion during on-orbit servicing, they do not apply to a small free-flyer simulator that requires accurate measurement of low propulsive forces. In contrast, the HIL simulator proposed in this paper uses an isolated high-resolution force/torque sensor to measure the low propulsive force accurately.

## III. PROPOSED HIL SIMULATOR

### A. System Overview

The proposed HIL simulator's configuration is shown in Fig. 2. One of the main differences from the existing HIL facilities is that the verification target platform is divided into two components in a functional distribution: One is a propulsion subsystem located on a 6-DoF force/torque sensor; the other component is a GNC subsystem with navigation sensors mounted on a robotic arm. The propulsion subsystem is controlled by propulsion commands from the main controller and drives the propulsion actuator based

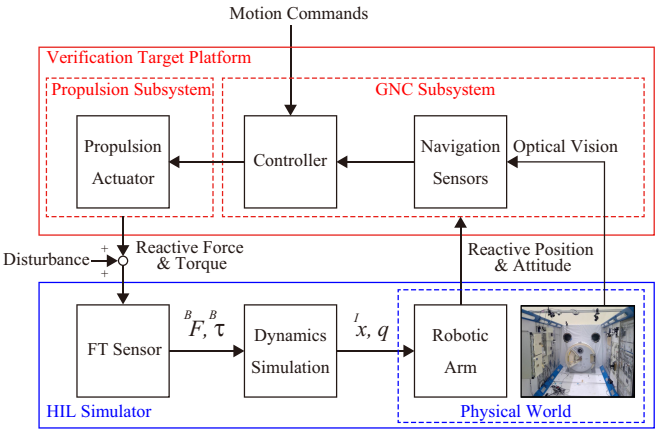



Fig. 2. Configuration of the HIL simulator.

on the flight software. The propulsive force and torque are measured precisely by the force/torque sensor and sent to a dynamics simulator that calculates the target platform's position and attitude. These are updated by solving the equations of motion using the measured force and torque. The calculations are sent in a motion command to the robotic arm so that the GNC subsystem moves according to the calculated position and attitude. The navigation system on the robotic arm estimates its position and attitude using actual images obtained in the flight simulated environment. The navigation results are used for guidance control in the main controller, which generates propulsion commands.

One of this HIL simulator's advantages is that external disturbances exerting on the target platform, such as airflows in ISS, can be measured by the force/torque sensor. This feature allows us to perform motion stability tests for external disturbances using this simulator.

### B. 6-DoF Force/Torque Sensor

The conventional approach to measuring the propulsive force in previous HIL simulators was to attach a force/torque sensor between the robotic arm and the target platform, making it possible to measure the force and torque directly without dividing the target platform. However, this approach requires eliminating the target platform's self-weight effect from the sensor output, which is difficult to do accurately for several reasons. First, the attitude measurement required for gravity compensation is disturbed by the sensor's measurement errors. The errors on the robotic arm's joint angle sensor cause the low-accuracy forward kinematics providing the target platform's attitude. A direct attitude measurement by an acceleration sensor is also involved in the sensor's noise and error. Such attitude measurement errors reduce the accuracy of gravity compensation, having a significant negative effect on the robot's propulsive force measurement because the small free-flying robot's propulsive force is extremely low relative to gravity (e.g., a 3kg robot generates 10 mN gravitational leakage force with only a 0.02° error in inclination). Second, the target platform oscillates due to the mechanical flexibility and the backlash on the arm joints. Such vibrations also disturb the robot's propulsive force measurement.

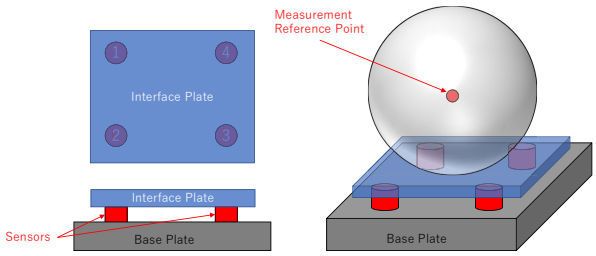



Fig. 3. Overview of the developed multi strain gauges force sensor.

Our approach isolates the force/torque sensor from the robotic arm and attaches it to the ground, allowing us to measure low propulsive force precisely and stably without complex gravity compensation. Note that a general high-resolution force sensor has a flexible structure point with strain gauges to sense deformation. Such flexible structures can induce vibrations by the propulsive force, making it difficult to measure the propulsive force precisely. A high-resolution sensor with a rigid structure is preferred for this purpose. Therefore, we developed a plate-type force sensor with four strain gauges, as shown in Fig. 3. The structure of this sensor is enhanced by the multiple strain gauge system, which avoids vibrations induced by the propulsive force.

The sensing process of the force and torque is described below. The force and torque applied on the target platform's center of mass (CoM) is calculated simply by adding the outputs from the force measured by the multi-strain gauges as follows:

$$ {}^{B}\boldsymbol{F} = \sum_{i=1}^{n} \boldsymbol{F}_i, \tag{1} $$

$$ {}^{B}\boldsymbol{\tau} = \sum_{i=1}^{n} \boldsymbol{d}_i \times \boldsymbol{F}_i, \tag{2} $$

where $\boldsymbol{F}$ and $\boldsymbol{\tau}$ are the force and torque on the CoM of the target platform, respectively. The superscript $B$ of ${}^{B}\boldsymbol{F}$ indicates the body-fixed coordinates of the target platform, and ${}^{B}\boldsymbol{F}$ indicates the $B$-frame component's expression of vector $\boldsymbol{F}$. $\boldsymbol{F}_i$ denotes the force measured on the $i$-th strain gauge. $\boldsymbol{d}_i$ stands for the vector from the measurement point of the $i$-th strain gauge to the CoM of the target platform. Since the force and torque calculated in the above equations include the sensor bias and gravity effects, the active thrust force of the target platform is obtained after eliminating these biases. Additionally, a low-pass filter is applied to the obtained force and torque to remove sensor noise. Furthermore, a dead zone is set to not react to low-level noises.

### C. Dynamics Simulator

The dynamics simulator calculates the target platform motion in three dimensions from the received force and torque. The dynamics are solved as a rigid body by the following procedure.

First, the translational motion in the inertial frame is computed from the force and torque obtained in the body frame. The force exerted on the CoM of the target in the inertial coordinate is calculated as follows:

$$ {}^{I}\boldsymbol{F} = \boldsymbol{C}_{IB}\, {}^{B}\boldsymbol{F}, \tag{3} $$

where $\boldsymbol{C}_{IB}$ is the direction cosine matrix (DCM) from the body coordinate to the inertial coordinate system. The superscript $I$ indicates the inertial coordinate frame. The linear velocity ${}^{I}\boldsymbol{v}$ and position ${}^{I}\boldsymbol{x}$ are computed simply as:

$$ {}^{I}\boldsymbol{v} = {}^{I}\boldsymbol{v}_0 + \frac{1}{m}\int_0^t {}^{I}\boldsymbol{F}\, dt, \tag{4} $$

$$ {}^{I}\boldsymbol{x} = {}^{I}\boldsymbol{x}_0 + \int_0^t {}^{I}\boldsymbol{v}\, dt, \tag{5} $$

where $m$ is the mass of the target platform.

Second, the rotational motion is calculated. The angular acceleration $\dot{\boldsymbol{\omega}}$ is given from the torque ${}^{B}\boldsymbol{\tau}$ obtained in (2) and the current angular velocity $\boldsymbol{\omega}$ as follows:

$$ \dot{\boldsymbol{\omega}} = \boldsymbol{I}^{-1}({}^{B}\boldsymbol{\tau} - \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega}), \tag{6} $$

where $\boldsymbol{I}$ stands for the inertia matrix of the target platform in the body coordinate frame. The current angular velocity $\boldsymbol{\omega}$ from the inertial frame to the body frame is updated by integrating the above equation. The differential quaternion is calculated from the obtained angular velocity $\boldsymbol{\omega}$ and the current quaternion $\boldsymbol{q}$ as follows:

$$ \dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{q} \otimes \boldsymbol{\omega} = \frac{1}{2}\begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix}, \tag{7} $$

where $\otimes$ represents the quaternion product. The current quaternion is updated by integrating the above differential quaternion. The updated quaternion is used to calculate DCM in (3) at the next step.

The updated position ${}^{I}\boldsymbol{x}$ and attitude $\boldsymbol{q}$ are sent to the robotic arm as a command.

### D. Robotic Arm

The robotic arm is controlled so that the GNC subsystem on the arm moves according to the updated position and attitude. When the robotic arm is controlled in the joint space, the joint angle commands are computed by inverse kinematics with the updated position and attitude. Each joint is controlled to follow the commanded angle.

### E. GNC Subsystem

The navigation of the target platform is performed on the robotic arm using navigation sensors, such as optical cameras, IMU, and LiDAR. To emulate the navigation performance, the robotic arm should be in the simulated environment where the surrounding walls and the lighting condition are similar to the flight condition. The navigation result (i.e., estimated pose) is used in the guidance control to perform path planning and calculation of each propulsion command, which is sent to the separated propulsion subsystem.

### F. Propulsion Subsystem

The propulsion subsystem is operated according to the propulsion command. The propulsion force and torque generated by the propulsion subsystem are measured by the attached 6-DoF force/torque sensor.

TABLE I

COMPARISON OF THE GROUND TEST FACILITIES.

| | Air-bearing platform | Computer simulation | HIL simulator |
|---|---|---|---|
| Dimensions | 2-trans. 1-rot. | 3-trans. 3-rot. | 3-trans. 3-rot. |
| Propulsion | Actual | Model-based | Actual |
| Visual navigation | Possible | Model-based | Possible |
| Hardware | Possible | N/A | Possible |
| Range | Middle | Large | Small |
| Impulsive contact | Possible | N/A | N/A |
| Preparation cost | High | Low | High |

## IV. GROUND VERIFICATION STRATEGY

This section introduces a ground verification strategy, including the air-bearing platform, the computer-based simulator, and the proposed HIL simulator.

### A. Ground Test Facilities

The comparison of three ground test facilities is summarized in Table 1. The air-bearing platform verifies the GNC algorithms using the propulsion system hardware, but the verifiable motion is limited to two dimensions. In contrast, computer simulation can verify three-dimensional motion easily, but the real visual navigation is invalid because an actual visual sensor is not used in the virtual environment. Additionally, the propulsion is model-based, making the precise modeling of the propulsive force difficult due to the complex aerodynamics and hardware uncertainties. The HIL simulator allows us to verify the GNC algorithms using the flight hardware and software simultaneously in three dimensions, which has been difficult in previous ground facilities. However, the range of motion is limited in the robotic arm's workspace. Furthermore, quick response to impulsive contact is difficult due to the time delay imposed by the low-pass filter and the robotic arm's response speed. In addition, the preparation cost and time are high, as well as the air-bearing platform cost due to the hardware.

### B. Verification Strategy

The verification and validation process from the component tests to the system tests for the Int-Ball2 is shown in Fig. 4. Navigation capabilities using cameras and their algorithms are validated in the JEM mockup field. The navigation output errors are evaluated using a motion capture system. Propulsive properties, including generated thrust force and time delay associated with commands, are validated using the force/torque sensor. The validated properties are imported into the controller design as a propellers' assignment matrix and time delay constant, which are used as control parameters for the flight software. GNC software that is written within the ROS framework is certified in ROS/Gazebo simulator as a virtual environment emulator. Based on the mathematical models of navigation and propulsion, the robustness against external disturbances and unpredictable control errors is evaluated by Monte-Carlo simulation, in which variations of disturbances, navigation errors, and propulsive forces are provided randomly in each simulation run.

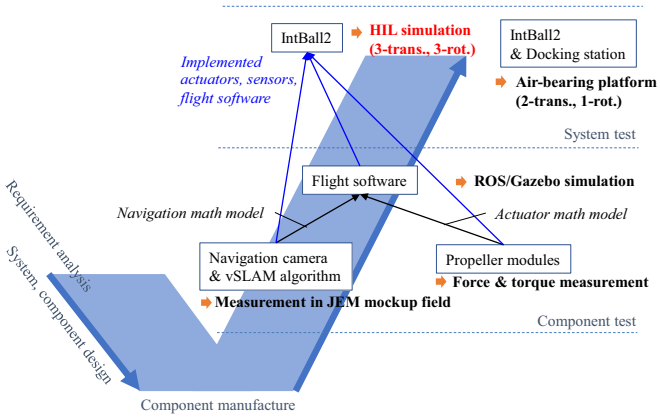



Fig. 4. Verification and validation process on the V-shape curve.

After finding suitable control parameters, the GNC verification integrating the actual hardware is performed on the air-bearing platform. This verification includes docking motion involving impulsive physical contacts, difficult to conduct in the HIL simulator, although the motion DoF is restricted to two translations and one rotation.

Last, we use the HIL simulator for three-dimensional verification. Although the HIL simulator could not perform a broad range of motion as often as the computer simulation, the HIL simulation is the most reliable and effective verification to include all essential functions of the actual propulsive system and real visual navigation in three dimensions. The comprehensive target platform's performance and stability are confirmed in several basic motion tests using the HIL simulator. The navigation results in three dimensions can also be evaluated easily by comparing them with the pose of the robotic arm.

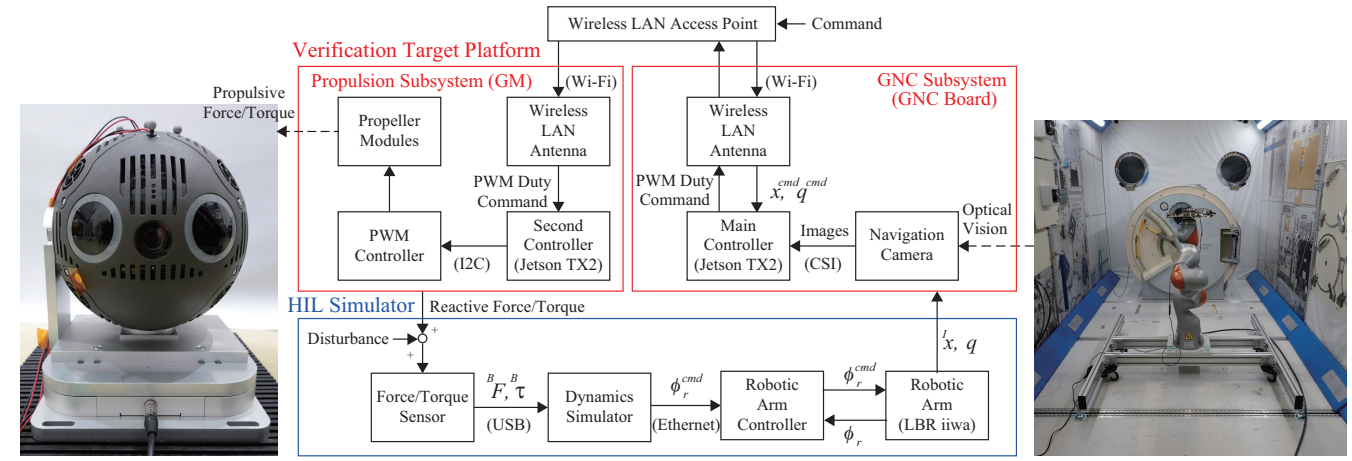
## V. DEMONSTRATION

This section shows the GNC demonstration results of Int-Ball2 in the JEM mockup field.

### A. Experimental Setup

The system configuration of this demonstration is shown in Fig. 5. In this demonstration, we used Int-Ball2's ground verification model (GM) for the propulsion subsystem and the GNC board for the GNC subsystem. The GM receives the PWM duty commands to control eight propeller modules that generate propulsive force and torque. For sensing them, the GM was mounted on the force and torque sensor, which measured force up to 50 N and torque up to 5 Nm with a resolution of 2.5 mN and 0.25 mNm at 1200 Hz. The GM on this force/torque sensor is depicted in Fig. 5(a).

The biquad low-pass filter processed the measured propulsive force and torque with a cut-off frequency of 10 Hz and a dead zone at 10 mN and 1 mNm. The position and attitude were updated at 200 Hz using the measured force and torque in the dynamics simulator. The joint position command was also constructed based on inverse kinematics and then sent to the KUKA's LBR iiwa14, which could be controlled from the external PC using a communication library (i.e., the Fast Robot Interface). The GNC board, including a navigation




(a) GM mounted on the force torque sensor.

(b) Data flow of the Int-Ball2 HIL simulation.

(c) GNC Board mounted on the robotic arm in the JEM mockup field.

Fig. 5. Configuration of the Int-Ball2's HIL simulation.

camera, was mounted on the robotic arm in the JEM's simulated environment, as shown in Fig. 5(c). This field was surrounded by the floor and walls to which equipment and pictures of the JEM were attached for evaluating actual visual navigation performance, as in a real flight environment. Lighting conditions in this field also mimicked the JEM's environment using an illuminometer.

Visual simultaneous localization and navigation (vSLAM) was performed at 15 Hz by the main controller using the images obtained by the navigation camera. The main controller received the target position command from an external PC through a wireless LAN access point. The guidance control used its relative position and attitude from the navigation results. In the guidance control, suitable path planning was performed to connect a smooth path to the commanded target position. The desired force and torque were calculated using a PID control law to track the defined path. Additionally, the main controller generated suitable PWM duty commands based on an assignment matrix that describes the relation between the rotation speeds of each propeller and the desired propulsive force/torque.

Next, the computed PWM duty commands were sent to the GM's controller through a wireless network. The desired propulsive force and torque were generated by controlling the eight propeller modules. The data flow of the procedure above is summarized in Fig. 5(b).

The coordinate definitions used in this demonstration are shown in Fig. 6. The origin of the body coordinate is located at the center of Int-Ball2. The X axis points toward the robot's front, while the Y axis is aligned with the navigation camera. The inertial coordinate is also defined so that the origin is the robot's initial position, and the axes directions are aligned to the body coordinate in the initial pose.

### B. Results and Discussion

Translational motion was verified in the first demonstration. The robot was controlled initially to keep its position and attitude at an initial point. A command to move forward 20 cm was received, then the robot was controlled to follow

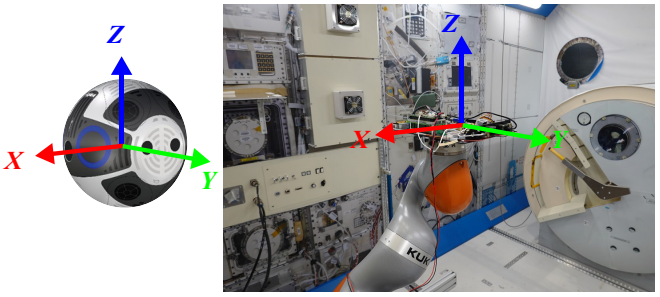



Fig. 6. Definition of body-fixed frame in the demonstration.

the planned trajectory. Fig. 7 plots the robot's position, attitude, and force profiles. At 3 seconds, the robot generated a large propulsive force along the X axis, moving forward to the updated target point, as seen in the force profiles. Under this propulsive force, the robot approached the target point. A negative propulsive force was generated to reduce the motion speed when it neared the target point. On the approach to the target point, slight deviations in the robot's position in the Z and Y axes were observed. This is attributed mainly to the force sensor's measurement errors and model errors used to calculate the propeller's rotation speed from a desired propulsive force. Since these deviations were small enough, the robot's position and attitude converged successfully to the target point. Also, as seen in the plot, the proposed HIL simulator successfully emulated the coupled three-dimensional motion, which could not be seen in a planar air-bearing platform.

In the second demonstration, the rotational motion was verified. The results of the motion and torque profiles are shown in Fig. 8. The robot was controlled initially to maintain its pose as well. The robot was rotated 90 degrees after the rotation command was received at 3 seconds. Large torques were observed around the Z axis at the beginning and end of rotation. Although slight deviations were induced in position and attitude during rotation, the robot's pose converged correctly to the target attitude, and the position was maintained successfully at the initial point.

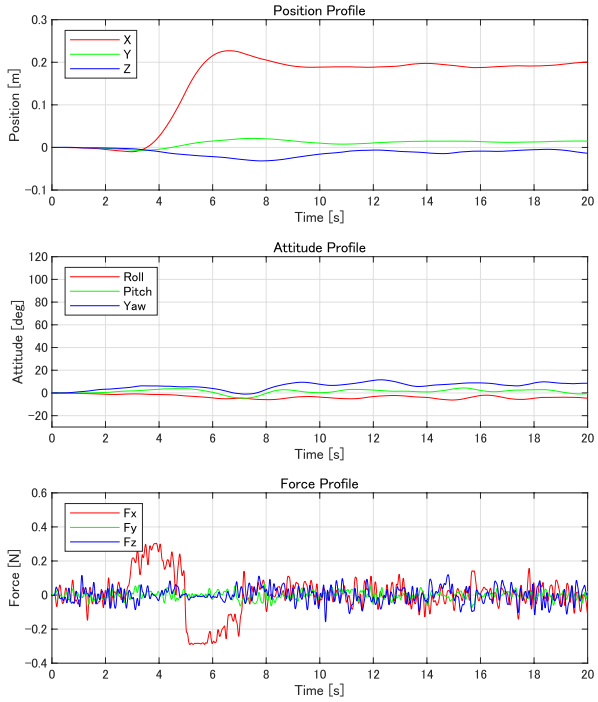



Fig. 7. Demonstration results for translational motion.

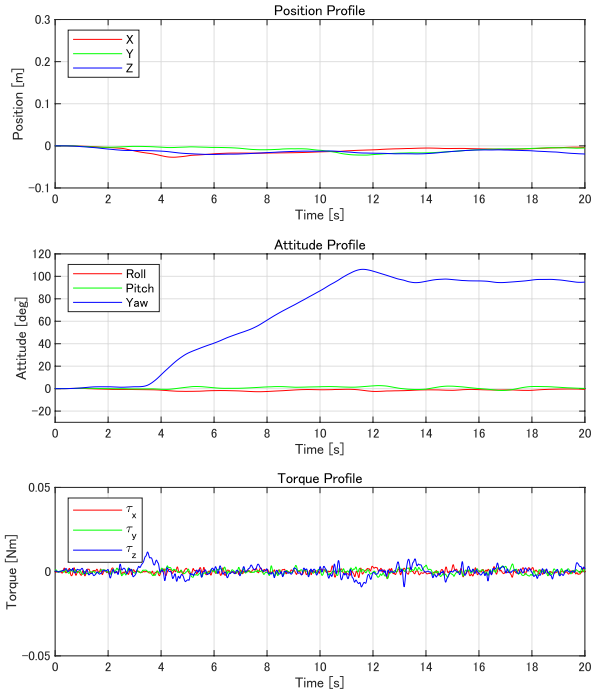



Fig. 8. Demonstration results for rotational motion.

## VI. CONCLUSIONS

This paper introduced the HIL simulator that can simultaneously verify the small free-flying robot's GNC algorithms using the flight hardware and software in three dimensions. This system is divided into the GNC subsystem on the robotic arm and the propulsion subsystem on the isolated force/torque sensor, which allows the low propulsive force to be measured accurately. We also introduced a verification strategy that uses three different facilities efficiently for the ground verification of the small free-flying robot's GNC algorithms. The effectiveness of the proposed HIL simulator was demonstrated successfully using Int-Ball2's GM and GNC board in the JEM mockup field. The results showed that the proposed HIL simulator reproduced three-dimensional motion in microgravity.

Our planned research includes a flight model's verification using this simulator to confirm the control stability before launch. The actual flying motion in the ISS will be compared to the simulated motion on the ground to evaluate its performance.

## ACKNOWLEDGMENT

The authors thank Mr. Suguru Mukai at Leptrino Inc. for assistance that greatly improved the force plate sensor used in this study.

## REFERENCES

[1] S. E. Fredrickson, L. W. Abbott, S. Duran, J. D. Jochim, J. W. Studak, J. D. Wagenknecht, N. M. Williams, Mini AERCam: Development of a Free Flying Nanosatellite Inspection Robot, in *SPIE Proceedings* Vol. 5088, Space Systems Technology and Operations, 2003.

[2] T. Fong, M. Micire, T. Morse, E. Park, C. Provencher, V. To, D. W. Wheeler, D. Mittman, R. J. Torres, Smart SPHERES: a Telerobotic Free-Flyer for Intravehicular Activities in Space, in *AIAA Space 2013 Conference*, San Diego, CA, USA, 2013.

[3] T. Smith, J. Barlow, M. Bualat, T. Fong, C. Provencher, H. Sanchez, E. Smith, Astrobee: A new platform for free-flying robotics on the international space station, in *International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, Beijing, China, 2016.

[4] V. Schröder, R. Regele, J. Sommer, T. Eisenberg, C. Karrasch, GNC System Design for the Crew Interactive MObile Companion (CIMON), in *69th International Astronautical Congress (IAC)*, Bremen, Germany, 2018, IAC-18-B3,6-A5.3,2.

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

[6] D. Hirano, S. Mitani, K. Watanabe, T. Nishishita, S. Inoue, Y. Kawai, S. Yamaguchi, H. Watanabe, M. Wada, Int-Ball2 for fully-teleoperated JEM onboard camera drone without crew aid, in *71st International Astronautical Congress (IAC)*, The CyberSpace Edition, 2020, IAC-20-D1.6.5.

[7] T. Rybus, K. Seweryn, Planar air-bearing microgravity simulators: Review of applications, existing solutions and design parameters, *Acta Astronautica*, Vol. 120, 2016, pp. 239–259.

[8] P. Kim, B. Coltin, O. Alexandrov, H. J. Kim, Robust Visual Localization in Changing Lighting Conditions, in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, 2017, pp. 5447–5452.

[9] A. Mora, R. G.Ruiz, P. Wofford, V. Kumar, B. V. Ross, A. Katterhagen, J. Barlow, L. Flückiger, J. Benavides, T. Smith, M. Bualat, Astrobee: Current Status and Future Use as an International Research Platform, in *69th International Astronautical Congress (IAC)*, Bremen, Germany, 2018, IAC-18-B3,6-A5.3,1.

[10] J. Artigas, M. De Stefano, W. Rackl, R. Lampariello, B. Brunner, W. Bertleff, R. Burger, O. Porges, A. Giordano, C. Borst, and A. Albu-Schaeffer, The OOS-SIM: An on-ground simulation facility for on-orbit servicing robotic operations, in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, Washington, USA, 2015, pp. 2854–2860.

[11] T. Boge, T. Wimmer, O. Ma, and T. Tzschichholz, EPOS - Using Robotics for RvD Simulation of On-Orbit Servicing Missions, in *AIAA Modeling and Simulation Technologies Conference*, Toronto, Canada, 2010, AIAA 2010-7788.

[12] P. Colmenarejo, M. Graziano, G. Novelli, D. Mora, P. Serra, A. Tomassini, K. Seweryn, G. Prisco, J. Gil Fernandez, On ground validation of debris removal technologies, *Acta Astronautica*, Vol. 158, 2019, pp. 206–219.