Acta Astronautica 247 (2026) 782–794

Contents lists available at ScienceDirect

Acta Astronautica

journal homepage: www.elsevier.com/locate/actaastro

Hardware-in-the-loop emulation for 6-DOF free-floating spacecraft using active gravity compensation with a collaborative robot

Hadi Jahanshahi, Mohammad Alizadeh, Zheng H. Zhu *

Department of Mechanical Engineering, York University, 4700 Keele Street, Toronto, Ontario, M3J 1P3, Canada

ARTICLE INFO

Keywords:
Free-floating spacecraft
Hardware-in-the-loop emulation
Ground testbed
Active gravity compensation
Admittance control

ABSTRACT

This technical note presents a hardware-in-the-loop testbed that uses a UR10e collaborative robot to emulate 6-DOF free-floating spacecraft motion through active gravity compensation. Experimental validations across all three Cartesian axes demonstrated high translational fidelity with a mean speed correlation of 0.9992 between the measured robot trajectory and an independently initialized rigid-body simulation, low residual forces below the controller deadband, and minor rotational errors of 2° to 6° attributed to kinematic coupling through the robot joint structure. The platform provides a cost-effective testbed for ground-based emulation of free-floating spacecraft dynamics.

1. Introduction

Robotic on-orbit servicing and active debris removal missions require rigorous ground-based validation of guidance, navigation, and control algorithms prior to space deployment. A critical challenge in such testing is the accurate reproduction of 6-DOF (degrees of freedom) free-floating dynamics on Earth, where a spacecraft subject to an external wrench undergoes translation and rotation without gravitational effect [1,2]. Many ground-based experimental approaches have been developed to address this challenge, each involving trade-offs among dynamic fidelity, cost, workspace, and achievable degrees of freedom.

Air-bearing tables are among the most widely used platforms for free-floating emulation, enabling near-frictionless planar translational and rotational motion on precision-leveled granite surfaces [3–5]. Although providing high fidelity planar motion, they are inherently limited to 3-DOF, require specialized facilities, and offer restricted workspace. Suspension-based systems use cables or springs to partially compensate gravity, but parasitic pendulum effects degrade the fidelity of the emulated dynamics [6]. Neutral buoyancy facilities support full 6-DOF motion emulation, but they are costly and subject to significant hydrodynamic disturbances [7].

Robotic hardware-in-the-loop (HIL) testbeds have emerged as a flexible alternative for emulating full 6-DOF motion within the robot workspace [8,9]. In such systems, a robotic manipulator physically drives spacecraft mock-ups along trajectories that are dynamically updated at each sampling interval. The updates are computed using a rigid-body dynamic model of the mock-up spacecraft together with measured interaction forces and torques between the mock-up and the robot arm's end-effector. Typically, an admittance-control framework is employed: the measured wrench is mapped through a rigid-body dynamics model of the mock-up spacecraft to obtain translational and angular accelerations, which are then integrated to generate joint velocity and position commands for the manipulator. In a typical dual-robot HIL configuration, one manipulator carries the servicer spacecraft equipped with a gripper, and the other carries the target spacecraft or debris mock-up; each robot independently emulates the free-floating dynamics of the body it represents, enabling physical simulation of approach, contact, and capture maneuvers [8,10,11]. Recent work has addressed gravity compensation accuracy in such configurations [12] and has proposed aerial-ground testbeds for compliant on-orbit manipulation simulation [13]. Comprehensive reviews of ground testing methods for space robots are provided in Refs. [14,15].

While the admittance-based emulation concept is well established, existing publications on robotic HIL testbeds leave two gaps. First, they primarily validate the overall mission outcome (e.g., successful docking or capture) rather than rigorously quantifying the fidelity of the free-floating dynamics emulation itself; the accuracy with which the robot reproduces the physics of an unforced coasting body has not been characterized independently of the contact phase. Second, most existing testbeds employ industrial robots (e.g., FANUC [8]) with dedicated

* Corresponding author.
E-mail address: gzhu@yorku.ca (Z.H. Zhu).

https://doi.org/10.1016/j.actaastro.2026.05.058
Received 29 April 2026; Received in revised form 26 May 2026; Accepted 27 May 2026
Available online 28 May 2026

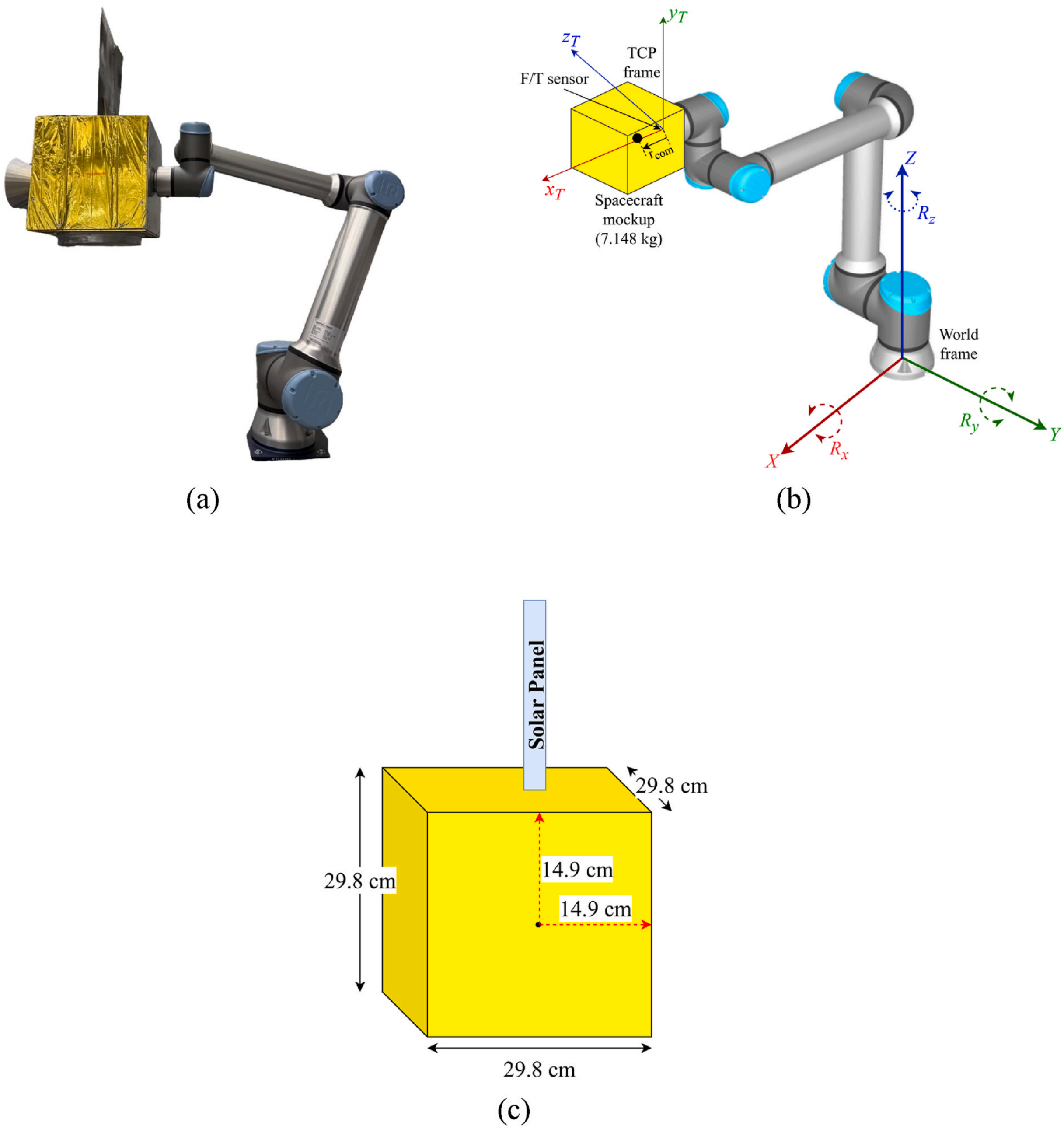

Fig. 1. Experimental setup: (a) UR10e robot arm with spacecraft mock-up mounted at the TCP; (b) coordinate frame definitions showing the world frame {W} at the robot base and TCP frame {T} at the tool flange, with the CoM offset r_{com} indicated; (c) Spacecraft mock-up dimensions and geometric center.

external precision force/torque sensors, whereas collaborative robots with integrated sensors present a lower-cost but noisier alternative that requires additional signal processing.

This technical note addresses both gaps by developing a HIL emulation for 6-DOF free-floating spacecraft using active gravity compensation via a collaborative robot with integrated sensors. A single Universal Robots (UR) 10e collaborative robot arm, using only its integrated force/torque sensor at the end-effector, is configured to emulate the 6-DOF free-floating response of a spacecraft mock-up. A multi-stage signal processing pipeline, comprising deadband filtering, confirmation gating, and adaptive low-pass filtering, is developed to handle the higher noise floor of the built-in sensor compared to external precision sensors. The primary contribution is a coast-phase validation methodology that decouples the assessment of free-floating emulation fidelity from the push-phase controller behavior. An independent rigid-body simulation is initialized from the robot state at the moment external contact ceases, and the resulting velocity decay profiles are quantitatively compared. The validation also characterizes the residual rotational error of the platform and identifies the kinematic coupling between joint-space and Cartesian-space motion, combined with partial payload compensation by the robot's internal force estimation, as the structural sources of this error, informing future improvements. This platform is designed to serve as the target emulator within a broader dual-robot HIL testbed for autonomous proximity operations and grasping research.

2. Platform description

2.1. Hardware configuration

The experimental platform consists of a UR10e collaborative robot arm with a spacecraft mock-up rigidly mounted at the tool center point (TCP) of the end-effector, as shown in Fig. 1(a). The UR10e has six

Table 1
System parameters.

Parameter	Symbol	Value	Unit
Spacecraft mass	m	7.148	kg
CoM offset, $\{T\}$ frame	$\mathbf{r}_{\text{com}}$	[122.8, −14.5, −14.2]	mm
Inertia I_{xx} (body)	$I_{xx,B}$	0.384	kg·m^2
Inertia I_{yy} (body)	$I_{yy,B}$	0.505	kg·m^2
Inertia I_{zz} (body)	$I_{zz,B}$	0.290	kg·m^2
Control frequency	f_c	125	Hz
Translational damping	b_t	3.0	N·s/m
Trans. Velocity decay	λ_t	0.9998	per step
Rot. Velocity decay	λ_r	0.970	per step
Force deadband	F_{db}	2.5	N
Torque deadband	τ_{db}	0.20	N·m
Confirmation steps	N_{conf}	8	steps
ServoL lookahead	t_{la}	0.05	s
ServoL gain	K_s	500	—

revolute joints, a reach of 1300 mm, a payload capacity of 10 kg, and an integrated six-axis force/torque sensor at the tool flange that measures three force components (F_x, F_y, F_z) and three torque components (τ_x, τ_y, τ_z).

Two coordinate frames are defined, as illustrated in Fig. 1(b). The world frame $\{W\}$, with axes X, Y, Z, is fixed at the robot base with Z pointing vertically upward. All dynamic computations, velocity commands, and validation analyses are expressed in this frame. The TCP frame $\{T\}$, with axes x_T, y_T, z_T, is attached to the tool flange with x_T aligned along the Wrist 3 axis. The force/torque sensor is co-located with the TCP frame origin at the flange interface. Because the spacecraft mock-up is rigidly mounted at the TCP, a body-fixed frame $\{B\}$ aligned with the mock-up's principal axes of inertia is also defined. In the present configuration the body frame is oriented parallel to the TCP frame and differs from it only by the translation $\mathbf{r}_{\text{com}}$ from the TCP origin to the spacecraft center of mass; the two frames therefore share the same rotation matrix $\mathbf{R}(t)$ with respect to the world frame.

The spacecraft mock-up has a measured mass of 7.148 kg and a center-of-mass (CoM) offset of $\mathbf{r}_{\text{com}} = [122.8, -14.5, -14.2]$ mm from the TCP origin expressed in the tool frame $\{T\}$. The mock-up inertia tensor is defined in the body-fixed frame $\{B\}$ aligned with the principal axes of the spacecraft, $\mathbf{I}_B = \text{diag}\,(0.384, 0.505, 0.290)$ kg·m^2. For dynamic computations performed in the world frame, the body-fixed inertia is transformed to the world frame at each control step using the current orientation of the spacecraft, as detailed in Section 2.3. The identification procedure for these parameters is described in Section 2.2. The robot's force/torque sensor noise floor is approximately 0.9 N (3 σ) for force and 0.06 N m (3 σ) for torque. The UR controller payload is configured using the identified mass and CoM, enabling the gravity compensation to automatically track orientation changes. Table 1 summarizes the identified physical parameters and controller settings.

2.2. Parameter identification

The spacecraft mock-up's mass, CoM offset, and inertia tensor are identified experimentally using the UR10e's integrated force/torque sensor and scripted sinusoidal motion sweeps. During identification, the UR controller payload is set to zero so that the internal gravity compensation does not pre-subtract inertial forces from the sensor readings. All identification is performed with the mock-up mounted at the TCP in the same configuration used for the emulation experiments.

The mass is identified from Newton's second law applied along individual Cartesian axes. The robot executes sinusoidal translation sweeps in the form

$$p(t) = A\,\sin\,(2\pi f t) \tag{1}$$

along the X and Y axes in the world frame $\{W\}$, with amplitude $A = 100$ mm and frequency $f = 0.5$ Hz (period 2 s), performing four sweeps per axis. Due to workspace constraints along the vertical direction, the Z-axis sweeps use a reduced amplitude of $A = 40$ mm. The corresponding acceleration is computed from the second derivative of the commanded trajectory, and the measured reaction force along the sweep axis is recorded from the integrated sensor. The mass is estimated using a weighted least-squares fit of

$$F_{\text{measured}} = -\,m{\cdot}a \tag{2}$$

where the negative sign accounts for the sensor measuring the reaction force (Newton's third law) opposing the inertial force. The weights emphasize samples with high acceleration magnitude where the signal-to-noise ratio is favorable. The identified mass is $m = 7.148$ kg (compared to an initial assumed value of 7.34 kg), with $R^2 = 0.76$. The moderate R^2 value reflects the sensor noise floor (approximately 0.9 N) relative to the inertial forces generated at the sweep amplitude and frequency used.

The CoM offset is identified simultaneously with the mass from the translation sweep data. During translational acceleration, the force applied at the CoM generates a torque about the TCP origin according to

$$\boldsymbol{\tau} = \mathbf{r}_{\text{com}} \times \mathbf{F} \tag{3}$$

A skew-symmetric regression matrix is constructed from the measured force components, and the CoM offset vector is solved via least squares. The identified CoM offset is $\mathbf{r}_{\text{com}} = [122.8, -14.5, -14.2]$ mm in the TCP frame $\{T\}$. The dominant component (122.8 mm along x_T) is consistent with the mock-up geometry shown in Fig. 1(c), where the marked geometric center indicates approximately 149 mm from the face. The approximately 26 mm difference between the marked geometric center and the identified center of mass is attributed to three effects: the offset between the marked reference face and the TCP frame origin at the tool flange, the additional mass of the solar panel mounted on the cube, which shifts the CoM toward the panel side, and non-uniform mass distribution within the cube itself due to internal electronics and structural elements.

The diagonal elements of the body-fixed inertia tensor $\mathbf{I}_B$ are identified from sinusoidal rotation sweeps about each world axis. The robot executes angular oscillations in the form

$$\theta(t) = \Theta\,\sin\,(2\pi f t) \tag{4}$$

about each world axis individually, with angular amplitude $\Theta = 12^\circ$ and frequency $f = 0.5$ Hz over a duration of 12 s. The rotation commands are composed as true world-axis rotations ($\mathbf{R}_{\text{target}} = \mathbf{R}_{\text{world}}(\theta){\cdot}\mathbf{R}_{\text{start}}$) to avoid artifacts from adding angles directly to rotation vector components. The actual angular velocity is read from the robot at each timestep and differentiated using a five-point central difference scheme to obtain angular acceleration. For each axis, the diagonal inertia element is estimated by a single-channel linear regression of Euler's equation:

$$\tau_i = I_{ii}{\cdot}\alpha_i \tag{5}$$

where $i \in \{x, y, z\}$. Single-channel matching is used rather than full tensor regression because cross-channel contamination from uncompensated robot joint dynamics was found to be approximately 51%, rendering cross-coupling terms unreliable. Off-diagonal inertia elements estimated from the full regression violated physical constraints (negative eigenvalues, triangle inequality violation) and are therefore set to zero. Because the identification sweeps are performed about world axes while the spacecraft mock-up is mounted in a near-aligned orientation at the home pose, the identified diagonal elements approximate the principal moments of inertia of the mock-up to within the cross-coupling residual reported above. Four independent identification runs are performed for each axis. The resulting body-fixed diagonal inertia tensor is $\mathbf{I}_B = \text{diag}\,(0.384, 0.505, 0.290)$ kg·m^2, with a coefficient of variation below 1.5% across the four runs, confirming measurement repeatability. These values are included in Table 1.

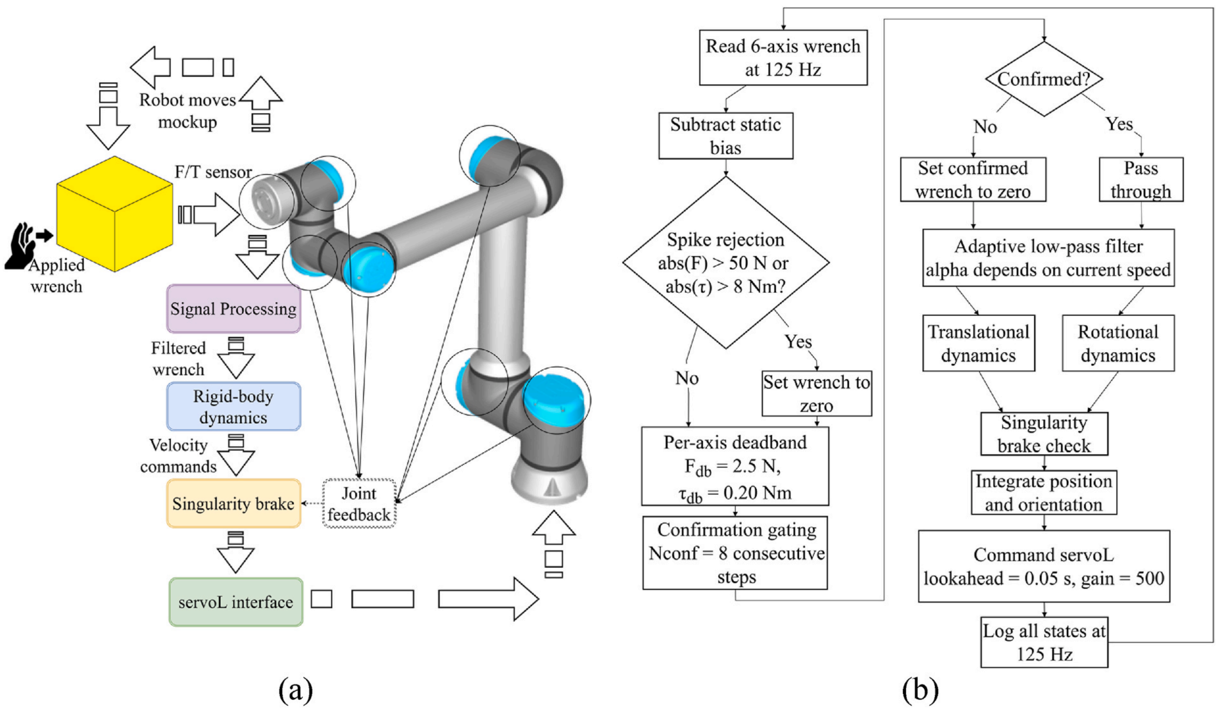

Fig. 2. Controller architecture: (a) system-level block diagram showing the hardware-in-the-loop control loop; (b) detailed controller pipeline.

2.3. Controller architecture

The controller implements an admittance control loop that converts externally applied wrenches into Cartesian velocity commands, which are then integrated to produce pose targets for the UR10e servoL real-time interface. The general admittance relationship is

$$\mathbf{F}_{\text{ext}} = M{\cdot}\mathbf{a} + B{\cdot}\mathbf{v} \tag{6}$$

where $\mathbf{F}_{\text{ext}}$ is the external wrench measured at the TCP, $\mathbf{a}$ is the resulting acceleration, $\mathbf{B}$ is the damping matrix, and $\mathbf{v}$ is the velocity. The mass matrix $\mathbf{M}$ takes different forms for the translational and rotational channels. For translation, $\mathbf{M} = m{\cdot}\mathbf{I}_3$, where $m = 7.148$ kg is the identified spacecraft mass and $\mathbf{I}_3$ is the 3×3 identity matrix. For rotation, $\mathbf{M} = \mathbf{I}_W(t)$, where $\mathbf{I}_W(t)$ is the orientation-dependent inertia tensor expressed in the world frame (defined in Eq. (12) below). This relationship is discretized and augmented with signal conditioning stages as described below. Fig. 2(a) illustrates the system-level architecture and Fig. 2(b) details the controller pipeline. At each 125 Hz control cycle, the following steps are executed.

The six-axis wrench (three force components F_x, F_y, F_z and three torque components τ_x, τ_y, τ_z, all expressed in the world frame $\{W\}$) is read from the integrated sensor. A static bias is calibrated at the starting pose (the home configuration illustrated in Fig. 1) by averaging 500 consecutive samples (4 s) at rest. The bias vector is subtracted from all subsequent readings. Individual readings exceeding 50 N (force magnitude) or 8 N m (torque magnitude) are rejected as sensor spikes; these thresholds are set at approximately 7 and 13 times of the respective deadband values, chosen empirically to reject transient electrical interference while passing any physically plausible hand-applied wrench.

Per-axis deadband thresholds of $F_{\text{db}} = 2.5$ N (force) and $\tau_{\text{db}} = 0.20$ N·m (torque) suppress sensor noise. The deadband function is

$$F_{\text{dead}}(t) = \begin{cases} F_{\text{raw}}(t) & \text{if} |F_{\text{raw}}(t)| \geq F_{\text{db}} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

These thresholds correspond to 2.8 and 3.3 times the respective 3 σ sensor noise levels (0.9 N and 0.06 N m), providing a margin that suppresses noise while remaining sensitive to light external (hand-applied) contact.

Force and torque channels are gated independently through a confirmation counter. The counter $c(t)$ evolves as

$$c(t+dt) = \begin{cases} \min\left(c(t)+1, N_{\text{conf}}\right) & \text{if} |F_{\text{dead}}(t)| > 0 \\ \max\left(c(t)-2, 0\right) & \text{otherwise} \end{cases} \tag{8}$$

The confirmed wrench is passed through only when $c(t) = N_{\text{conf}} = 8$, corresponding to 8 consecutive above-threshold readings (approximately 64 ms). The asymmetric increment/decrement ratio (1:2) causes the counter to reject transient spikes (which clear in 4 steps) while responding promptly to sustained contact (which saturates in 8 steps). This gating operates continuously regardless of the robot configuration.

The force and torque confirmation channels operate independently during contact detection. However, the torque channel is additionally gated on the force confirmation state: the confirmed torque is set to zero whenever the force counter has not saturated. This design reflects the physical constraint that external hand contact always produces a measurable force, whereas torque readings in the absence of a confirmed external force arise from internal robot dynamics rather than external interaction. This gating prevents residual sensor torque from driving spurious rotation during the coast phase.

A first-order exponential moving average filter is applied to the confirmed wrench:

$$F_{\text{filt}}(t) = \alpha{\cdot}F_{\text{conf}}(t) + (1-\alpha){\cdot}F_{\text{filt}}(t-dt) \tag{9}$$

where F_{conf} is the confirmed wrench and α is a velocity-dependent smoothing coefficient. At low translational speeds ($\|\mathbf{v}\| < 5$ mm/s), $\alpha = 1.5\alpha_0$ to provide responsive tracking of applied forces during initial contact. At higher speeds, $\alpha = 0.4\alpha_0$ to reduce sensitivity to motion-induced sensor artifacts arising from the UR10e's internal force estimation algorithm. The base coefficients are $\alpha_0 = 0.06$ for force and $\alpha_0 = 0.03$ for torque, yielding effective time constants of approximately 0.13 s (force, low speed) and 0.67 s (force, high speed) at the 125 Hz control rate.

The filtered force vector drives the spacecraft mock-up in accordance with Newton's second law with an explicit linear damping term:

$$\mathbf{a}(t) = \frac{\mathbf{F}_{\text{filt}}(t) - b_t \cdot \mathbf{v}(t)}{m} \tag{10}$$

$$\mathbf{v}(t + dt) = [\mathbf{v}(t) + \mathbf{a}(t) \cdot dt] \cdot \lambda_t \tag{11}$$

where $\mathbf{a}(t)$ is the translational acceleration, $\mathbf{v}(t)$ is the velocity, $m = 7.148$ kg is the identified spacecraft mock-up's mass, and $dt = 1/\,125$ s. The damping coefficient $b_t = 3.0$ N·s/m represents the virtual viscous drag in the simulation; it is a design parameter chosen to produce a physically realistic deceleration profile, yielding a drag-to-mass ratio of $b_t/m = 0.42\ s^{-1}$ and a $1/e$ time constant of 2.4 s. The per-step decay factor $\lambda_t = 0.9998$ provides a supplementary multiplicative velocity reduction that models residual energy dissipation; its value was tuned empirically to match the observed deceleration rate during preliminary coast trials. Together, the continuous damping b_t and the discrete decay λ_t cause the velocity to decay to 37% of its initial value in approximately 2.4 s after external force removal. The values of b_t and λ_t were determined by fitting the analytical speed decay model to speed profiles from preliminary coast trials, minimizing the root mean square (RMS) error between the model prediction and the measured robot speed. The resulting fit is validated in Section 4.1. Velocity magnitude is capped to $\|\mathbf{v}\| \leq 200$ mm/s to ensure safe operation within the robot workspace.

The rotational dynamics are integrated in the world frame using the body-fixed inertia tensor transformed at each control step. The rotation matrix $\mathbf{R}(t)$ corresponding to the current TCP orientation is used to compute the world-frame inertia:

$$\mathbf{I}_W(t) = \mathbf{R}(t) \cdot \mathbf{I}_B \cdot \mathbf{R}(t)^T \tag{12}$$

where $\mathbf{I}_B = \text{diag}\,(0.384, 0.505, 0.290)$ kg·m^2 is the inertia tensor expressed in the body-fixed principal frame. The sensor torque is measured about the TCP origin rather than about the spacecraft center of mass; therefore, the torque vector is transformed to the center of mass before being used in the rotational dynamics:

$$\boldsymbol{\tau}_{\text{CoM}}(t) = \boldsymbol{\tau}_{\text{filt}}(t) - \mathbf{r}_{\text{com}}(t) \times \mathbf{F}_{\text{filt}}(t) \tag{13}$$

where $\mathbf{r}_{\text{com}}(t) = \mathbf{R}(t) \cdot \mathbf{r}_{\text{com},T}$ is the CoM offset vector rotated from the TCP frame $\{T\}$ to the world frame $\{W\}$, and $\mathbf{r}_{\text{com},T} = [122.8, -14.5, -14.2]$ mm is the CoM offset in $\{T\}$. The angular acceleration and angular velocity are then computed by integrating Euler's equation:

$$\boldsymbol{\alpha}(t) = \mathbf{I}_W^{-1}(t)[\boldsymbol{\tau}_{\text{CoM}}(t) - \boldsymbol{\omega}(t) \times (\mathbf{I}_W(t) \cdot \boldsymbol{\omega}(t))] \tag{14}$$

$$\boldsymbol{\omega}(t + dt) = [\boldsymbol{\omega}(t) + \boldsymbol{\alpha}(t) \cdot dt] \cdot \lambda_r \tag{15}$$

where $\boldsymbol{\alpha}$ is the angular acceleration, $\boldsymbol{\omega}$ is the angular velocity, and the gyroscopic term $\boldsymbol{\omega} \times (\mathbf{I}_W \cdot \boldsymbol{\omega})$ accounts for the coupling between rotation axes that arises from asymmetric inertia. The per-step rotational decay $\lambda_r = 0.970$ provides stronger damping than the translational channel; this value was selected to suppress uncontrolled tumbling while maintaining observable rotational response, yielding 2% of the initial angular velocity remaining after 1 s ($0.970^{125} \approx 0.02$). The value of λ_r was determined by observing the angular velocity decay rate during preliminary rotation tests and selecting the decay factor that produced stable, physically plausible tumbling behavior without divergence. Angular velocity is capped to $\|\boldsymbol{\omega}\| \leq 0.50$ rad/s.

All dynamic computations described above are performed in the world frame $\{W\}$, which is the natural inertial frame for Newton-Euler integration. The TCP frame $\{T\}$ rotates with the end-effector and is therefore non-inertial; it is used only for the body-frame definition of the inertia tensor (Eq. (12)) and the CoM offset (Eq. (13)). Any world-frame vector $\mathbf{x}_W$ can be expressed in the TCP frame through the rotation $\mathbf{x}_T(t) = \mathbf{R}(t)^T \cdot \mathbf{x}_W$, with the velocity transformation $\mathbf{v}_T(t) = \mathbf{R}(t)^T \mathbf{v}_W(t)$ and the force transformation $\mathbf{F}_T(t) = \mathbf{R}(t)^T \mathbf{F}_W(t)$. Validation results expressed in the TCP frame are presented in Section 4.1.

The integrated position and orientation are sent to the robot via the servoL interface with a lookahead time of $t_{\text{la}} = 0.05$ s and a gain of $K_s = 500$. A real-time singularity brake monitors three joint-space proximity metrics and scales both translational and angular velocities to prevent singular configurations. The brake factor $\beta \in [0, 1]$ is computed as the minimum of three components:

$$\beta = \min\,(\beta_{\text{vel}}, \beta_{\text{elbow}}, \beta_{\text{wrist}}) \tag{16}$$

The joint velocity brake β_{vel} scales linearly from 1 to 0 as the joint angular velocity magnitude $\|\dot{\mathbf{q}}\|$ increases from 0.6 to 1.5 rad/s, preventing high-speed motion near workspace boundaries. The elbow brake β_{elbow} monitors the third joint angle q_3 and scales linearly from 1 to 0 as q_3 approaches 0 or π (the elbow-straight singularity), with the danger zone defined between 0.05 and 0.12 rad from the singular configuration. The wrist brake β_{wrist} similarly monitors the fifth joint angle q_5 and scales linearly as $|q_5|$ approaches 0 (the wrist singularity), using the same threshold range. Both translational and angular velocities are multiplied by β before integration, providing smooth deceleration as the robot approaches any singular configuration.

3. Validation methodology

The validation approach compares the robot trajectory during coast phases (intervals of free drift after the operator releases the mock-up) with an independently conducted rigid-body simulation. Two parallel simulations are evaluated for each detected coast event. Simulation A (Controller replica) replays the recorded filtered force and torque sequences through an offline copy of the full controller pipeline with identical parameters. This simulation must reproduce the logged robot trajectory exactly (RMS = 0.0 mm), serving as a sanity check on the logging and replay infrastructure. Simulation B (Coast-reset rigid body) is initialized from the robot's actual velocity and angular velocity at the instant the coast phase begins, defined as the first timestep where the force confirmation counter drops below the saturation threshold. From that point forward, the simulation propagates rigid-body dynamics using the translational and rotational integration equations (Eqs. (10), (11), (14) and (15)) with the same damping and decay parameters but with zero external wrench ($\mathbf{F}_{\text{filt}} = \mathbf{0}$, $\boldsymbol{\tau}_{\text{filt}} = \mathbf{0}$). If the robot controller faithfully emulates free-floating physics, the measured velocity decay during coast should match Simulation B.

An analytical speed decay profile is derived by noting that in the absence of external force, the scalar speed $s(t) = \|\mathbf{v}(t)\|$ evolves under two independent decay mechanisms: the continuous damping b_t, which produces exponential decay with time constant m/b_t, and the discrete per-step factor λ_t, applied at each control cycle. Combining these yields

$$s(t) = s_0 \cdot \lambda_t^{f_c \cdot t} \cdot \exp\,(-b_t \cdot t\,/\,m) \tag{17}$$

where $s_0 = \|\mathbf{v}_0\|$ is the speed at coast entry and $f_c = 125$ Hz is the control frequency. This closed-form expression provides an independent reference that does not depend on numerical integration and serves as a cross-check on Simulation B.

Coast events are detected automatically by identifying transitions in the force confirmation state from confirmed ($c_f = N_{\text{conf}}$) to unconfirmed ($c_f < N_{\text{conf}}$), requiring a minimum coast duration of 1 s (125 samples) to exclude brief contact interruptions. The torque confirmation state is not used for coast detection because the torque-about-CoM correction (Eq. (13)) converts any residual sensor force into a non-zero torque signal through the cross-product $\mathbf{r}_{\text{com}} \times \mathbf{F}_{\text{filt}}$. With a CoM offset magnitude of approximately 0.124 m and a residual sensor force of order 1 to 2 N during coast, the correction-induced torque (on the order of 0.15 to 0.25 N m) can exceed the torque deadband and keep the torque confirmation counter saturated even when no external contact is present. Detecting coast based on the force confirmation alone ensures clean separation between contact and coast phases.

Table 2
Translational coast-phase validation results.

Test	Axis	v_0 (mm/s)	Coast (s)	Speed RMS (mm/s)	Correlation	$\|\mathbf{F}\|_{res}$ (N)	$\|\boldsymbol{\tau}\|_{res}$ (N·m)
1	Z	118.7	9.73	2.973	0.9995	1.840	0.602
2	X	83.3	12.47	3.205	0.9988	0.814	0.145
3	Y	63.6	9.62	1.628	0.9994	2.502	0.277
Mean				2.602	0.9992	1.719	0.341
± Std				0.831	0.0004	0.846	0.233

4. Results and discussion

4.1. Translational validation

Three experiments are conducted with the operator applying single impulsive pushes along different Cartesian axes of the world frame. Table 2 summarizes the coast-phase validation metrics. All three runs exhibit zero brake activation and zero controller replica RMS error.

The mean Pearson correlation between the robot and the rigid-body model speed profiles is 0.9992 (standard deviation 0.0004), indicating excellent agreement across all tests. Fig. 3 presents the coast-phase speed decay and per-axis velocity comparisons for the three tests. In all cases, the measured robot speed, the numerical rigid-body simulation, and the analytical decay curve (Eq. (17)) closely overlap, with the robot decaying marginally faster than the model due to the residual sensor force discussed in Section 4.2. The analytical decay curve and the numerical rigid-body simulation coincide exactly in every test, confirming that the discrete-time integration in Eqs. (10) and (11) is consistent with the closed-form solution.

The initial coast speeds range from 63.6 to 118.7 mm/s, demonstrating that the model remains valid across nearly a factor of two in velocity magnitude. Test 3 achieves the lowest speed RMS (1.628 mm/s) and the highest correlation (0.9994) despite having the largest residual force magnitude (2.502 N); the residual force is sub-deadband on a per-axis basis and contributes uniformly to the deceleration without distorting the decay profile. Test 2 achieves a 0.9988 correlation with a slightly higher speed RMS of 3.205 mm/s, while Test 1 reaches the highest correlation (0.9995) at the highest entry speed. The per-axis velocity columns in Fig. 3 reveal that Test 1 (row a) excited all three Cartesian axes simultaneously despite being a nominally Z-axis push, indicating kinematic cross-axis coupling through the robot joint structure. Tests 2 (row b) and 3 (row c) produced predominantly single-axis responses, confirming clean impulse application.

Fig. 4 shows the full experimental timeline for the three tests. Each row presents position (left), velocity (center), and force (right) along the dominant axis of motion in the world frame. The vertical dashed line marks the onset of the coast phase used for validation. The robot and controller replica trajectories are indistinguishable in all nine subplots, confirming zero offline replay error and validating the logging infrastructure. The velocity panels show peak speeds during the push phase that exceed the coast-entry speeds v_0 reported in Table 2: peak velocity occurs while the operator's force is still being applied, and the velocity drops by the time the confirmation counter clears and the coast phase begins. The force column demonstrates the multi-stage signal processing pipeline of Section 2.3: the raw sensor signal exhibits noise amplitudes of approximately ±5 N, while the filtered signal cleanly extracts the push impulse and drops to zero once the confirmation counter clears. The Y-axis push (Test 3) shows visible oscillatory noise in the raw force

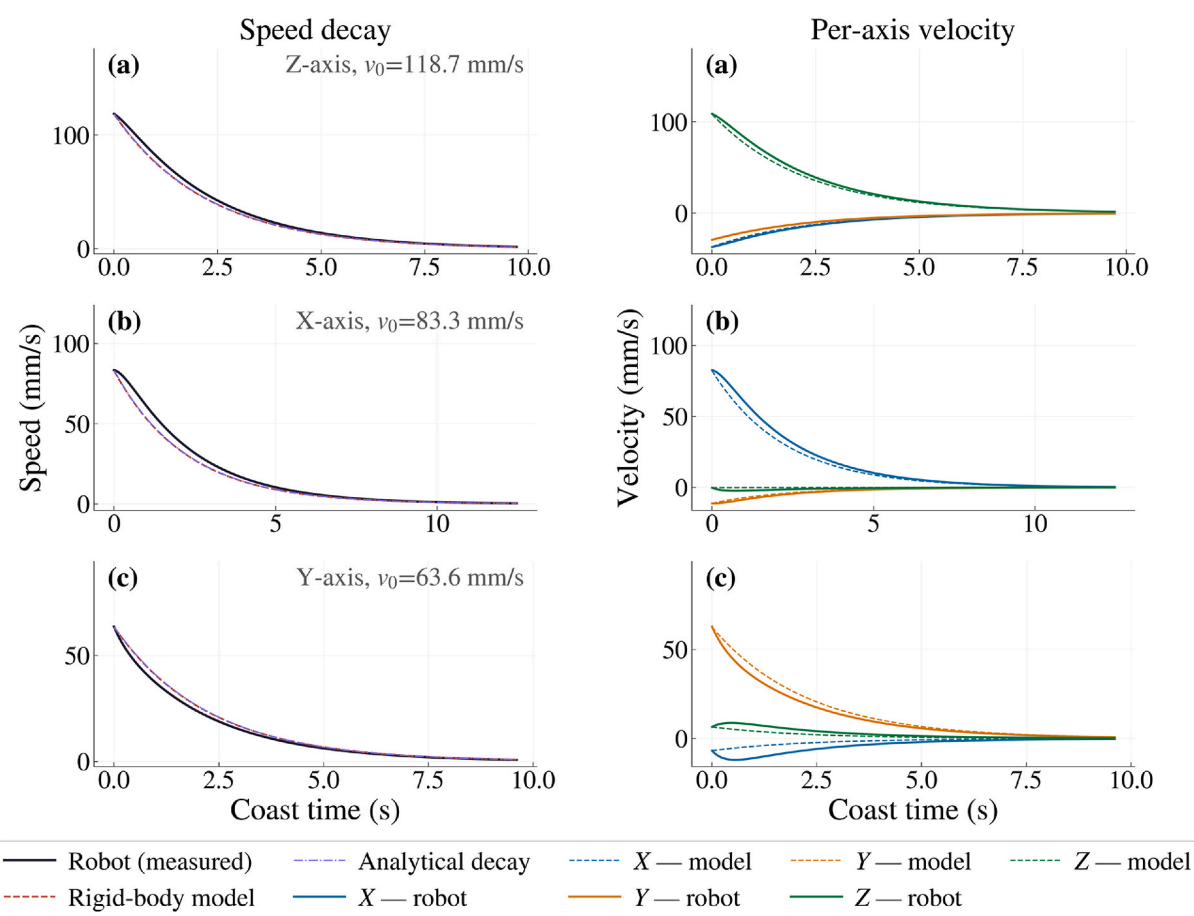

Fig. 3. Coast-phase speed decay (left column) and per-axis velocity comparison (right column).

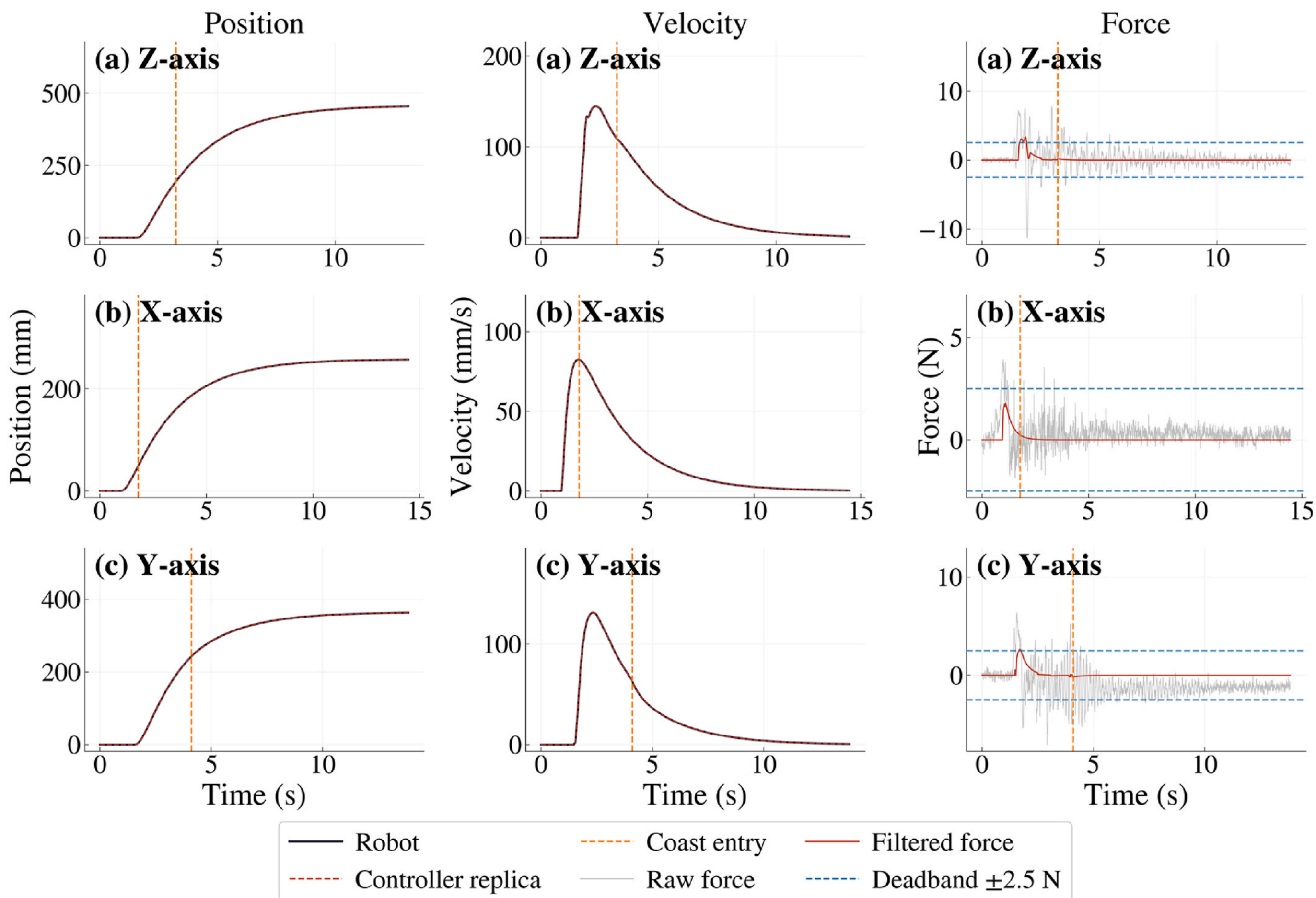

Fig. 4. Full experimental timeline in the world frame: position, velocity, force.

during coast, consistent with motor torque ripple along that joint configuration. Travel distances along the dominant axis range from approximately 250 mm (Test 3) to 450 mm (Test 1), representing a significant fraction of the UR10e workspace.

All dynamic computations and the results presented above are expressed in the world frame $\{W\}$, which is the natural inertial frame for Newton-Euler integration. For comparison, the same trajectories expressed in the TCP frame $\{T\}$ are presented in Fig. 5. Each world-frame quantity is projected onto the instantaneous TCP basis through $\mathbf{x}_T(t) = \mathbf{R}(t)^T\mathbf{x}_W(t)$ as defined in Section 2.3. The total magnitude of position, velocity, and force is frame-independent (a property of orthogonal rotations), but the decomposition across coordinate axes differs depending on the orientation of $\{T\}$ relative to $\{W\}$ at each instant. Table 3 reports the coast-entry velocity vector decomposed in the TCP frame for each test, and Fig. 5 shows that the coast detection, the controller-replica fidelity, and the residual force bounds are all preserved across the two representations.

4.2. Residual force analysis

The mean residual sensor force magnitude during the coast phases is 1.719 ± 0.846 N, and the mean residual torque magnitude is 0.341 ± 0.233 N·m. Fig. 6 presents the residual force and torque profiles for the three tests.

The residual force is below the controller force deadband of 2.5 N in Tests 1 and 2 (1.840 N and 0.814 N respectively). Test 3 exhibits a magnitude of 2.502 N, which is at the boundary of the deadband as a vector norm but remains below the per-axis threshold on each individual component, since the controller deadband (Eq. (7)) operates per-axis rather than on the vector magnitude. As a result, the confirmation counter remains below saturation during coast in all three experiments and no spurious translational inputs reach the rigid-body integrator. The residual force itself is attributed to two effects: uncompensated joint friction within the UR10e arm, and small imperfections in the integrated gravity compensation that depend on the robot's joint configuration. Tests 1 and 3 sit in joint configurations with elevated friction signatures and exhibit higher residuals; Test 2 (X-axis push) operates in a smoother configuration and exhibits the lowest residual.

The residual torque magnitudes are 0.602 N m (Test 1), 0.145 N m (Test 2), and 0.277 N m (Test 3). These values are higher than would be expected from raw sensor noise alone (0.06 N m at 3 σ) because they reflect a combination of two effects. The first effect is the torque-about-CoM correction of Eq. (13): any residual sensor force acting through the CoM offset of approximately 0.124 m produces a torque component of magnitude $\|\mathbf{r}_{\text{com}}\|\cdot\|\mathbf{F}_{\text{res}}\|$. For the measured residual forces, this yields predicted contributions of approximately 0.23 N m (Test 1), 0.10 N m (Test 2), and 0.31 N m (Test 3). The Test 2 and Test 3 measurements are consistent with this prediction within sensor noise, indicating that the residual torque in those tests is essentially the residual force translated to the CoM. The Test 1 measurement exceeds the prediction by approximately 0.37 N m and is attributed to motion-induced sensor artifacts during the high-speed Z-axis trajectory, in which the UR10e's internal force estimation generates a small but persistent torque bias that the static bias calibration (performed at rest) cannot remove. This excess torque has no effect on the coast-phase validation because the

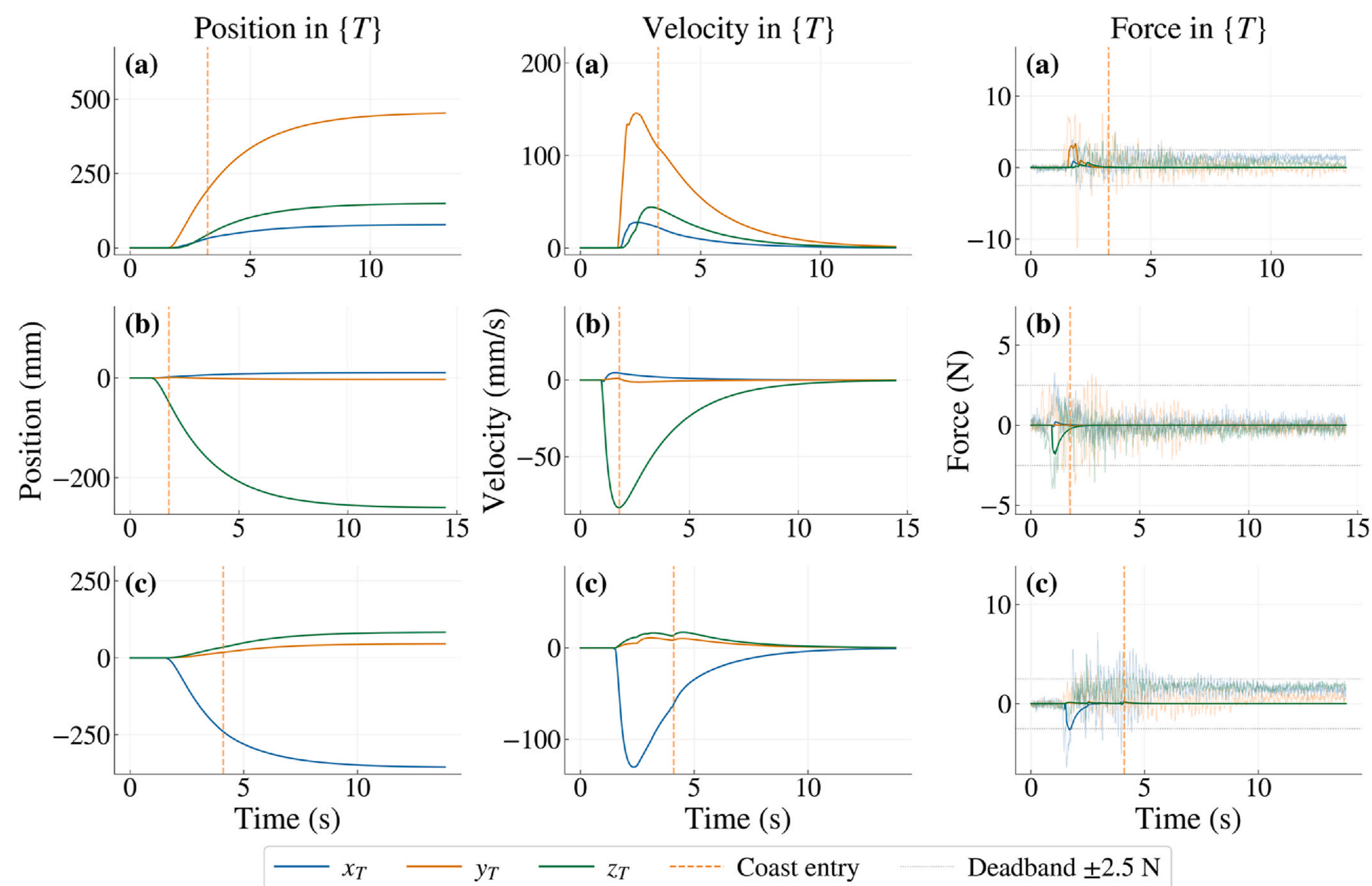

Fig. 5. Full timeline in the TCP frame: position, velocity, force.

Table 3
Coast-entry velocity in the TCP frame $\{T\}$.

Test	World axis	v_{0,x_T} (mm/s)	v_{0,y_T} (mm/s)	v_{0,z_T} (mm/s)	$\|\mathbf{v}_0\|$ (mm/s)
1	Z	+22.3	+108.5	+42.5	118.7
2	X	+4.5	+1.0	−83.2	83.3
3	Y	−61.5	+8.6	+13.9	63.6

torque confirmation channel is gated on the force confirmation state (Section 2.3); with the force counter below saturation during coast, no torque is forwarded to the rigid-body integrator regardless of its magnitude.

4.3. Rotation divergence

Fig. 7 presents the per-axis rotation comparison between the robot and the controller model, together with the absolute error for each test. The rotation divergence is heterogeneous across tests and across axes. Test 1 (Z-axis push) exhibits sign disagreement in two of the three axes: the robot rotates by approximately $+4^\circ$ about R_x while the model predicts approximately -2°, and the robot rotates by approximately -2° about R_z while the model predicts approximately $+7^\circ$. The R_y channel shows magnitude underestimation by approximately 1°. The per-axis RMS errors for Test 1 are 5.34° (R_x), 0.85° (R_y), and 8.05° (R_z), with a total RMS of 5.60°. Test 2 (X-axis push) shows the smallest divergence, with errors distributed across all three axes (1.69° in R_x, 1.88° in R_y, 2.43° in R_z) and a total RMS of 2.02°. Test 3 (Y-axis push) exhibits the largest single-axis discrepancy: the robot reaches -2° in R_z while the model over-predicts to approximately -12°, producing a 9.65° RMS error in this channel alone. The total RMS for Test 3 is 5.75°.

The absolute error column in Fig. 7 reveals that the divergence accumulates primarily during the push phase, before the coast-entry line, and then remains frozen during free coast. This confirms that the source of the rotation error is the wrench injected into the rotational dynamics during active contact, not drift in the rotation model itself. The structure of the error, sign disagreements in Test 1 and magnitude over-prediction in Test 3, indicates that the rotational input received by the controller does not faithfully represent the rotation the robot actually undergoes. Two mechanisms contribute. First, the kinematic coupling between joint-space and Cartesian-space motion: translational forces applied at the TCP generate moments through the manipulator's joint kinematics that are physically present in the robot motion but absent from any Cartesian-frame wrench measurement. The Cartesian-space rigid-body model in Eqs. (12)–(15) cannot reconstruct these joint-space contributions from the sensor reading alone. Second, the partial compensation of the payload offset by the UR10e's internal force estimation: because the controller payload is configured with the identified mass and CoM (Section 2.1), the sensor reading already accounts for a portion of the gravitational and inertial moments about the CoM, and the additional correction applied through Eq. 13 ($\boldsymbol{\tau}_{\mathrm{CoM}} = \boldsymbol{\tau}_{\mathrm{filt}} - \mathbf{r}_{\mathrm{com}} \times \mathbf{F}_{\mathrm{filt}}$) can shift or over-shoot the rotational input in some channels. The combined result is a rotational error that is structural rather than parametric: refining the inertia tensor, the damping coefficient, or the moment-arm correction within the present formulation cannot eliminate it.

Fig. 8 presents the angular velocity comparison between the actual measured angular velocity, derived from numerical differentiation of the TCP orientation at 125 Hz with a 200 ms moving-average filter, and the controller model prediction. The figure decomposes the rotation error of Fig. 7 into its rate-level contributions. In Test 1, the model fails in two opposite ways during the push phase. In the ω_x channel, the

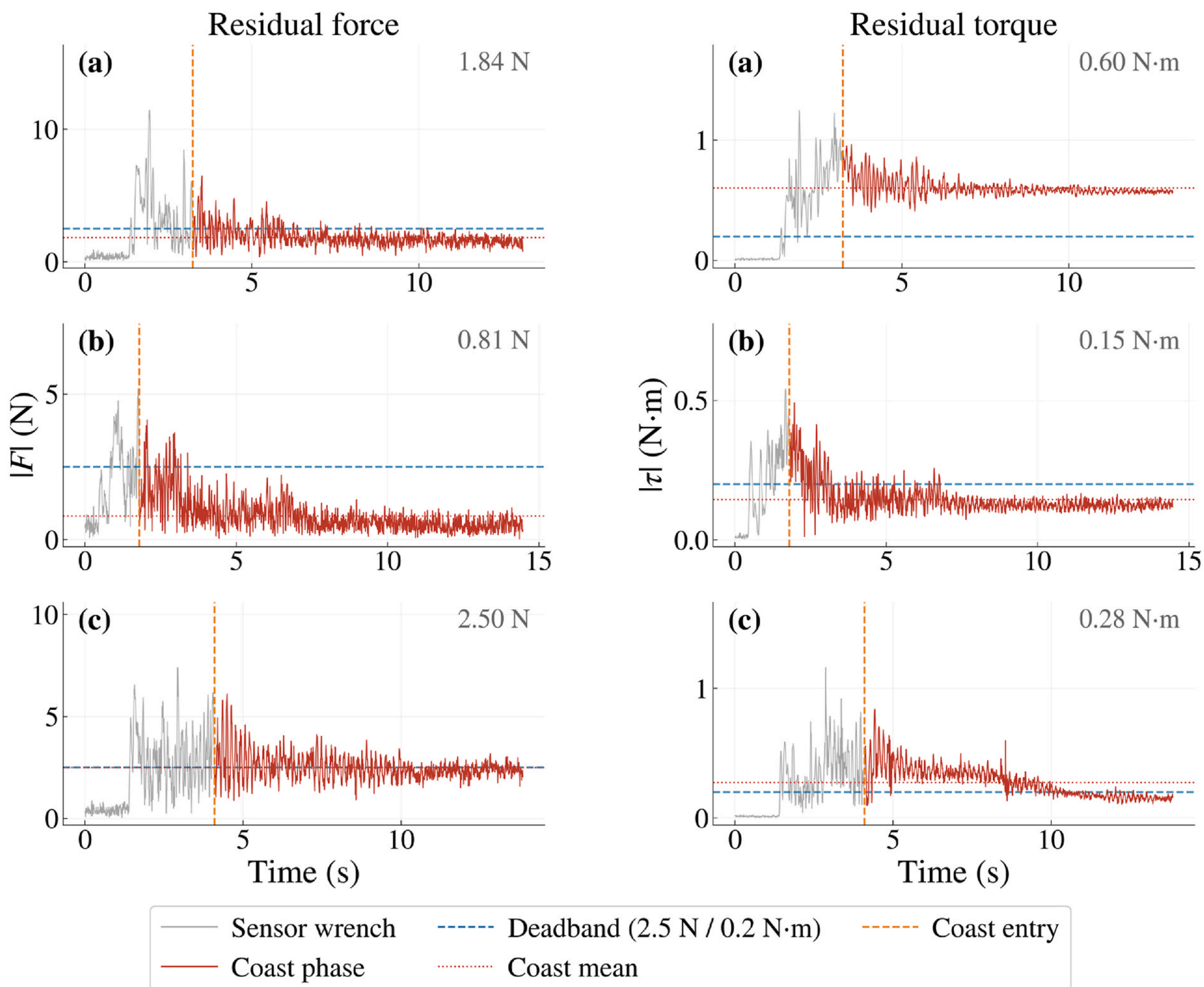

Fig. 6. Residual sensor force and torque during coast phases.

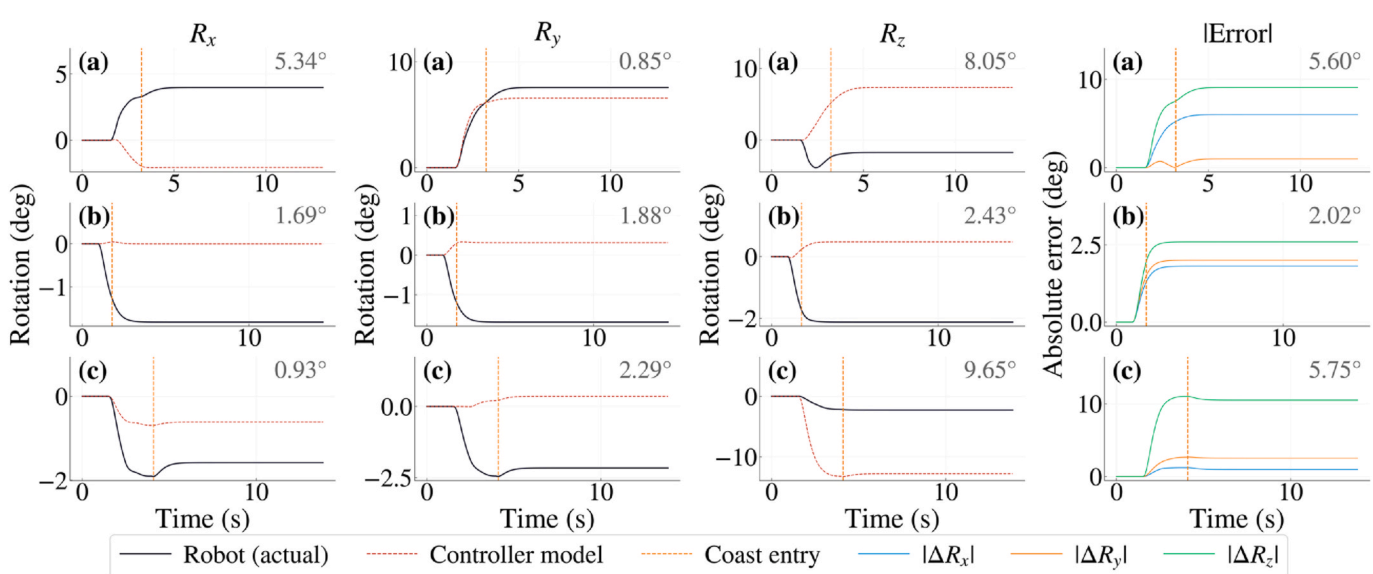

Fig. 7. Per-axis rotation: robot, controller model, and absolute error.

model predicts essentially zero angular velocity while the robot peaks at approximately $+10^{\circ}/s$, with an RMS error of $1.94^{\circ}/s$; this indicates that the torque sensor did not register the rotational component of the push along this axis and the model received no input to drive that rotation. In the ω_y channel, the model peaks at approximately $10^{\circ}/s$ while the robot only reaches approximately $3^{\circ}/s$, with an RMS error of $1.15^{\circ}/s$; here the

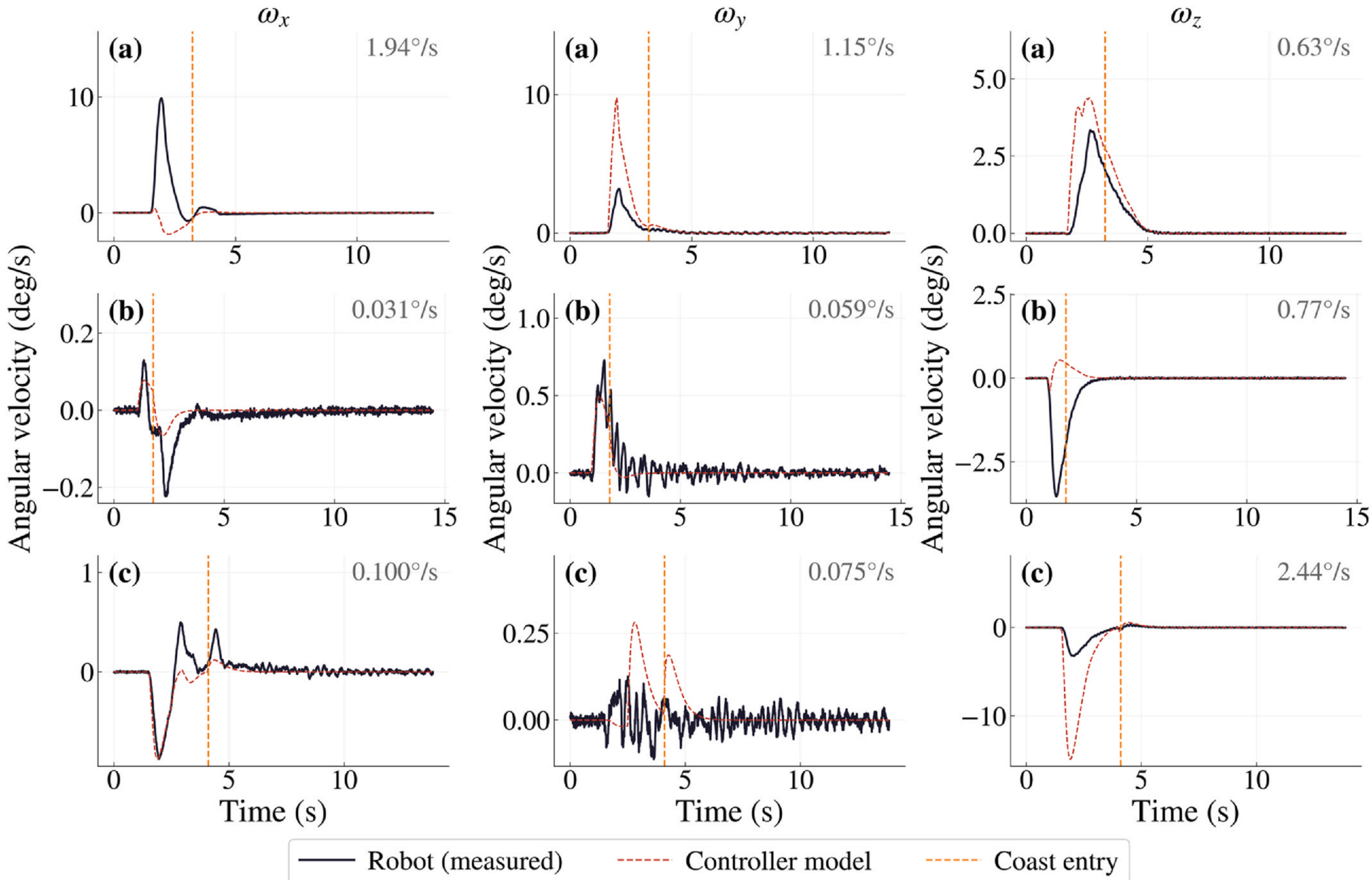

Fig. 8. Per-axis angular velocity: robot and controller model.

corrected torque (Eq. (13)) over-drives the rotational dynamics relative to the actual robot response. In Test 2, all three channels achieve sub-degree-per-second RMS agreement (0.031°/s, 0.059°/s, and 0.77°/s for ω_x, ω_y, and ω_z), confirming that the rotation dynamics model in Eqs. (14) and (15) is accurate when the kinematic coupling is small. In Test 3, the ω_z channel exhibits the most pronounced over-prediction: the model dives to approximately $-15^\circ/s$ while the robot only reaches approximately $-3^\circ/s$, with an RMS error of 2.44°/s. This is the rate-level signature of the 5× position-level over-prediction visible in R_z in Fig. 7.

The peak angular velocities decay to near zero within 1 to 2 s after coast entry, consistent with the per-step rotational decay of 0.970 which reduces the angular velocity to approximately 2% of its initial value within 1 s ($0.970^{125} \approx 0.02$). This rapid decay explains why the rotation divergence in Fig. 7 manifests as a frozen orientation offset during coast rather than a continuing drift: the rotational input generated during the brief push phase is integrated into a permanent orientation error, and the subsequent coast-phase decay returns the angular velocity to zero before the model and the robot can re-converge.

Fig. 9 visualizes the full spacecraft trajectory during Test 1 using four complementary views: a 3D perspective and three orthographic projections onto the $X-Y$, $X-Z$, and $Y-Z$ planes. Oriented satellite cubes are drawn at 1.5-s intervals along the trajectory; the warm-colored face on each cube marks the body $+x$ direction and tracks the rotation of the mock-up over time. The trajectory line is colored by speed, with bright colors indicating higher speed during and immediately after the push and darker colors indicating the slow drift toward the end of the coast. The two-dimensional projections allow direct measurement of the displacement and the rotation of the marker face from any of the three coordinate planes, complementing the 3D view, which shows the overall trajectory shape but is subject to perspective distortion.

4.4. Servo tracking performance

Table 4 reports the servoL tracking error between the commanded and actual TCP poses for the three tests.

The mean position tracking RMS is 1.61 mm and the mean rotation tracking RMS is 0.089°. Peak position errors of 5 to 9 mm occur exclusively during the push phase when the TCP undergoes maximum acceleration; during coast phases, the position tracking error is consistently below 1 mm. Fig. 10 shows the per-axis tracking error over time for the three tests. Test 1 exhibits the largest rotation tracking error (0.192° RMS, 0.625° peak), corresponding to its larger rotation excursion of approximately 8.7° in total; the servo must chase the orientation changes induced by the kinematic coupling, and a small persistent offset of approximately 0.2 to 0.3° remains during coast in all three rotation axes. Tests 2 and 3 produce smaller rotation excursions and achieve rotation tracking RMS values below 0.04°, representing near-noise-floor performance for the servoL interface at 125 Hz.

The largest servo tracking error observed across all three experiments (0.625° rotation, 9.32 mm position) is more than an order of magnitude smaller than the rotation divergence reported in Section 4.3, which reaches 9.65° RMS in the R_z channel of Test 3. This rules out the servo interface as a meaningful contributor to the rotation discrepancy and confirms that the emulation fidelity is limited by the rigid-body physics model receiving an imperfect rotational input, not by the servo execution.

5. Conclusion

This work shows that a single UR10e collaborative robot arm with a built-in force/torque sensor can emulate 6-DOF free-floating

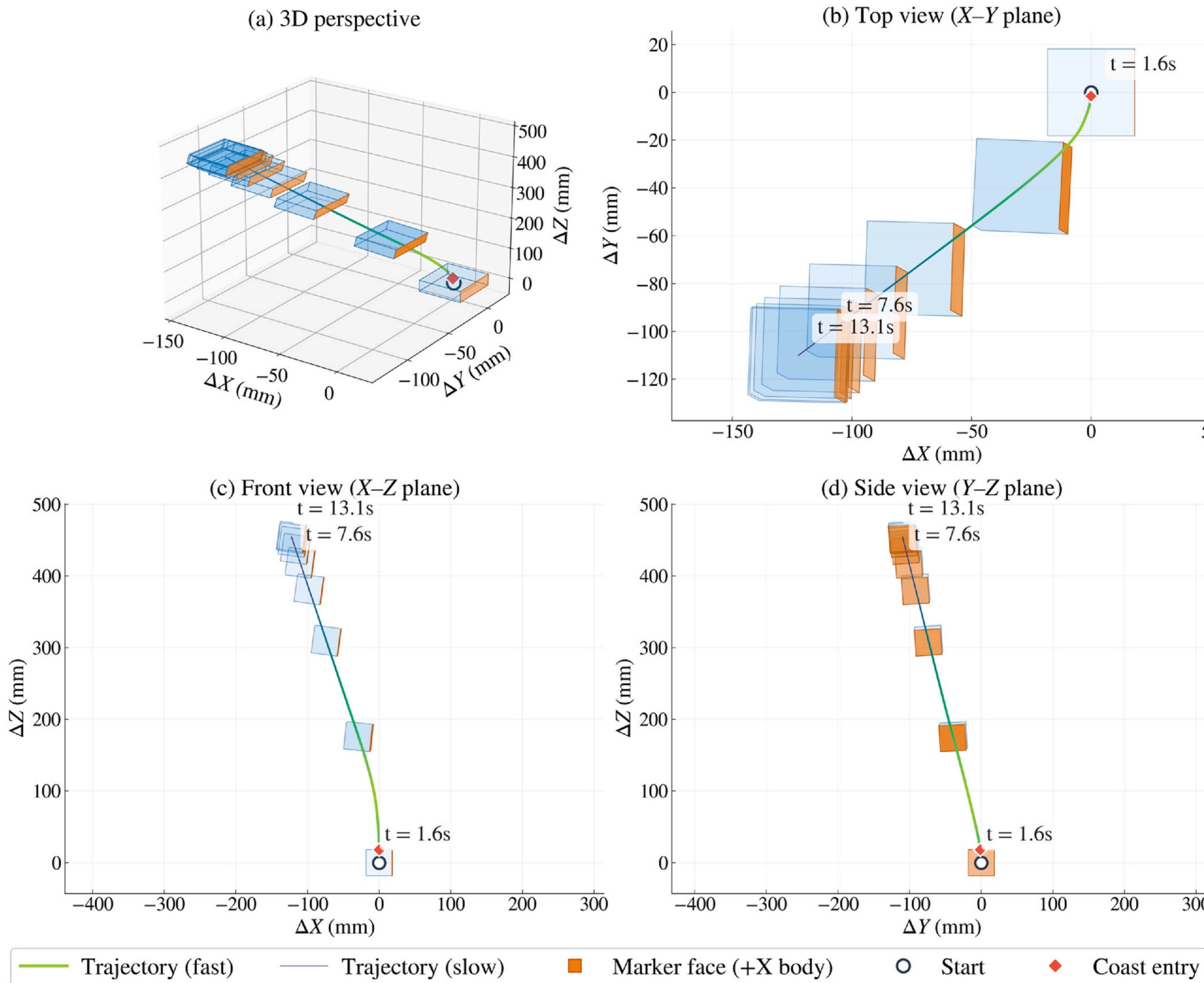

Fig. 9. Four-view trajectory and orientation visualization for Test 1.

Table 4
ServoL tracking error.

Test	Pos. RMS (mm)	Pos. Peak (mm)	Rot. RMS (°)	Rot. Peak (°)
1	2.11	9.32	0.192	0.625
2	1.08	5.33	0.038	0.190
3	1.65	8.50	0.037	0.173
Mean	1.61		0.089	

spacecraft motion with high translational fidelity when paired with appropriate wrench detection and filtering. Despite the higher noise floor of the built-in sensor, the selected deadband thresholds of 2.5 N and 0.20 N m reliably distinguished external contact from sensor noise and internal robot forces across all experiments. The coast-phase validation confirms that the platform accurately reproduces free-drift translational dynamics. Across three experiments spanning all three Cartesian axes and initial coast-entry speeds from 63.6 to 118.7 mm/s, the measured translational speed profiles agreed closely with independent rigid-body simulations, with a mean Pearson correlation of 0.9992 and a mean speed RMS error of 2.6 mm/s. Servo tracking errors below 2.2 mm in position and 0.2° in rotation further indicate that the robot's servoL interface is not a limiting factor for emulation accuracy. The main limitation observed is configuration-dependent force-to-torque coupling introduced through the robot joint kinematics, compounded by partial payload compensation by the UR controller. During translational pushes, this coupling produced rotational errors of approximately 2° to 6° in the form of sign disagreements in some channels and magnitude over-prediction in others. Angular velocity measurements indicate that these errors arise from brief rotational impulses during force application, which accumulate into permanent orientation offsets that are then frozen by the rapid rotational decay during the coast phase. The rotational dynamics model itself, including Euler-equation integration with the orientation-dependent inertia tensor and torque transformation to the spacecraft center of mass, was accurate when the kinematic coupling was small. This limitation is structural to the Cartesian-space rigid-body formulation and affects rotational response during contact but does not compromise the validated translational coast-phase fidelity. A secondary limitation is residual joint friction, estimated at approximately 1.7 ± 0.8 N, which acts as an uncompensated drag force and causes the measured velocities to decay marginally faster than the rigid-body model predicts over long coast durations. Future work should address both limitations by incorporating a joint-space dynamics model or a learned correction capable of capturing the configuration-dependent force-to-torque coupling and the residual drag. Such extensions would also support real-time adaptation of

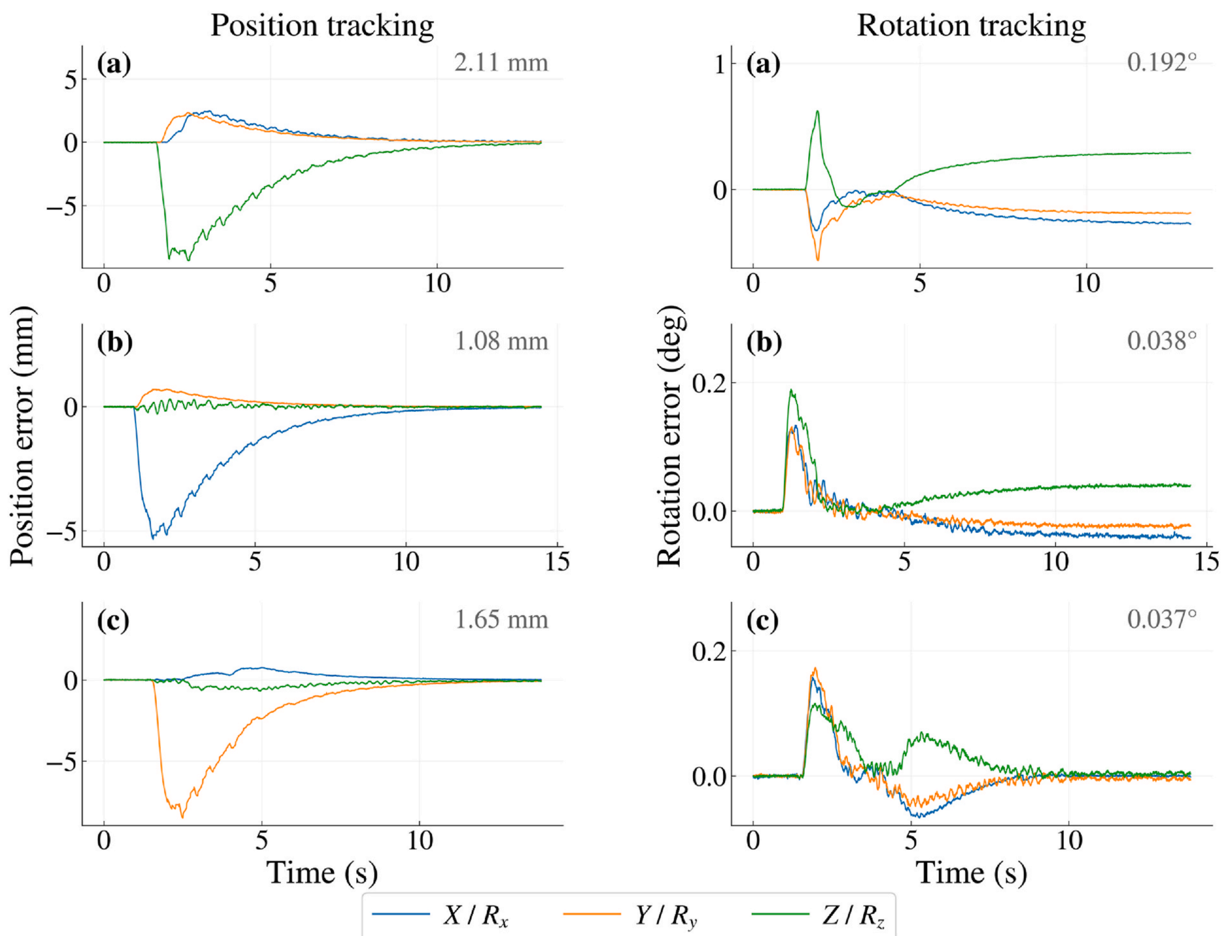

Fig. 10. ServoL tracking error: position and rotation.

inertial parameters, enabling more realistic emulation of spacecraft operations with changing mass and inertia, such as robotic grasping, docking, and refueling.

CRediT authorship contribution statement

Hadi Jahanshahi: Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Validation, Visualization, Writing – original draft. **Mohammad Alizadeh:** Formal analysis, Investigation, Methodology, Validation. **Zheng H. Zhu:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Supervision, Validation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported by the Discovery Grant (RGPIN-2024-06290) and Collaborative Research and Training Experience Program Grant (555425-2021) of the Natural Sciences and Engineering Research Council of Canada. The authors also acknowledge support from the Natural Sciences and Engineering Research Council of Canada (NSERC) through a Postgraduate Scholarship – Doctoral (PGS D). Nous remercions le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG) de son soutien par l'entremise d'une bourse d'études supérieures du doctorat (BES D).

Data availability

The controller software, validation scripts, and sample experimental data are available at https://github.com/HJahanshahi/free-floating-emulation.

References

[1] H. Jahanshahi, Z.H. Zhu, Review of machine learning in robotic grasping control in space application, Acta Astronaut. 220 (2024) 37–61.
[2] H. Jahanshahi, Z.H. Zhu, A comprehensive review of tactile sensing technologies in space robotics, Chin. J. Aeronaut. 38 (2025) 103423, https://doi.org/10.1016/j.cja.2025.103423.
[3] T. Rybus, K. Seweryn, Planar air-bearing microgravity simulators: review of applications, existing solutions and design parameters, Acta Astronaut. 120 (2016) 239–259.
[4] W. Zhu, Z. Pang, Z.H. Zhu, Planar air-bearing microgravity testing for maneuverable space net deployment and retrieval with distributed cooperative control, Acta Astronaut. 236 (2025) 1096–1100, https://doi.org/10.1016/j.actaastro.2025.07.053.
[5] H. Jahanshahi, Z.H. Zhu, Uncertainty propagation networks for neural ordinary differential equations, Neurocomputing 677 (2026) 133134, https://doi.org/10.1016/j.neucom.2026.133134.
[6] K. Yoshida, Engineering test satellite VII flight experiments for space robot dynamics and control: theories on laboratory test beds ten years ago, now in orbit, Int. J. Robot Res. 22 (2003) 321–335.
[7] E. Papadopoulos, F. Aghili, O. Ma, R. Lampariello, Robotic manipulation and capture in space: a survey, Frontiers in Robotics and AI 8 (2021) 686723.
[8] A.A. Ali, B. Beigomi, Z.H. Zhu, Development of 6DOF hardware-in-the-loop ground testbed for autonomous robotic space debris removal, Aerospace 11 (2024) 877.

[9] A. Chhabra, D. Kim, DYNAMOS: design, implementation, and validation of a hardware-in-the-loop simulation testbed for In-Space servicing, IEEE Aero. Electron. Syst. Mag. 41 (2026) 40–53.
[10] O. Ma, J. Wang, S. Misra, M. Liu, On the validation of SPDM task verification facility, J. Rob. Syst. 21 (2004) 219–235.
[11] K. Saulnier, D. Pérez, R. Huang, D. Gallardo, G. Tilton, R. Bevilacqua, A six-degree-of-freedom hardware-in-the-loop simulator for small spacecraft, Acta Astronaut. 105 (2014) 444–462.
[12] L. Guo, Y. Zhao, C. Wei, Gravity compensation for hardware-in-the-loop zero-gravity simulation system of spacecrafts, Aero. Sci. Technol. (2025) 110650.
[13] A. Suarez, A. Gonzalez-Morgado, M. Ruiz, A. Lucena, A.M. Giordano, A. Ollero, Aerial-ground testbed for simulating compliant bimanual on-orbit operations: from aerial to space robotic manipulation, Robot. Autonom. Syst. 187 (2025) 104927.
[14] M. Wilde, C. Clark, M. Romano, Historical survey of kinematic and dynamic spacecraft simulators for laboratory experimentation of on-orbit proximity maneuvers, Prog. Aero. Sci. 110 (2019) 100552.
[15] A. Flores-Abad, O. Ma, K. Pham, S. Ulrich, A review of space robotics technologies for on-orbit servicing, Prog. Aero. Sci. 68 (2014) 1–26, https://doi.org/10.1016/j.paerosci.2014.03.002.