Parallel Thruster-Propeller Control for Emulating Spacecraft Proximity Dynamics on a Free-Flyer

E. Krantz*[1], P. Sanjaya[1], P. Roque[2], G. Tibert[1], H. Mao[1], J. W. Burdick[2], and C. Fuglesang[1]

***Abstract*—Ground-based free-flyer testbeds simulate spacecraft dynamics using air bearings for near-frictionless planar motion, but do not inherently reproduce orbital proximity dynamics. To address this limitation, we present a dual-actuation architecture that physically emulates Clohessy–Wiltshire dynamics on an ATMOS free-flyer. Two model predictive controllers operate in parallel: one controls solenoid thrusters with an orbital dynamics model, representing the spacecraft controller; the other drives bidirectional propellers to physically apply the orbital accelerations, continuously updating its reference to account for thruster-induced velocity changes. Experimental results demonstrate accurate trajectory tracking with concurrent emulation of proximity dynamics by the propellers.**

***Index Terms*—Orbital Robotics, Model Predictive Control, Free-Flying Testbed**

I. Introduction

Ground-based spacecraft simulators are essential tools for developing and validating guidance, navigation, and control algorithms for on-orbit proximity operations [1], [2]. These testbeds typically employ flat surfaces and air bearings to approximate undamped planar motion, enabling the study of microgravity rendezvous, docking, and formation flying scenarios [3]. While such platforms are capable of closely reproducing free-floating rigid-body dynamics, they do not capture the relative dynamics of spacecraft in orbit, commonly described by the linearized Clohessy-Wiltshire (CW) equations [4].

Several approaches have been proposed to address this limitation. In [5], scaled CW dynamics are emulated on an air-bearing testbed by commanding the platform's own compressed-air thrusters to simultaneously apply CW-equivalent accelerations and execute spacecraft maneuvers. However, using a single actuation system for both roles introduces two drawbacks. The binary, duty-cycle-controlled nature of solenoid thrusters limits the accuracy with which continuous orbital forces can be reproduced. Moreover, spacecraft control and dynamics emulation share the same actuation budget, coupling their performance. Hardware-in-the-loop (HITL) approaches using robotic manipulators or dynamically tilted surfaces [1] offer alternatives but sacrifice the free-floating nature of the platform.

In this work, we present a dual-actuation architecture for physically emulating relative orbital dynamics on free-flyer testbeds. We demonstrate the approach on ATMOS [6], an open-source free-flyer platform, which we extend with a second actuation modality, yielding two independent systems operating in parallel: (i) the existing compressed-air solenoid thrusters, controlled by a Model Predictive Controller (MPC) whose dynamics model includes the CW equations, representing the spacecraft actuation system; and (ii) a new propeller-based actuation system with four bidirectional motors providing continuous proportional thrust, controlled by a second MPC that physically applies the CW accelerations.

The contributions of this paper are:

1) A dual-actuation architecture for emulating orbital dynamics on free-flying platforms, in which two closed-loop controllers operate in parallel, with the dynamics reference continuously updated to account for spacecraft thrust.
2) A propeller-based actuation module providing continuous, proportional thrust as an additional actuation modality for the ATMOS platform, enabling injection of orbital dynamics, disturbances, or other environmental forces.

We validate the approach experimentally using time-scaled CW dynamics, demonstrating that the thruster MPC accurately tracks spacecraft trajectories despite concurrent propeller-applied orbital forces.

II. Dual-Actuation Control Architecture

A. Relative Orbital Dynamics

The relative motion of a chaser spacecraft with respect to a target on a circular orbit is described by the CW equations [4] in the local orbital Hill frame $\mathcal{H}$. This rotating reference frame is centered at the target, with $\hat{\boldsymbol{x}}$ along the radial (outward from the central body), $\hat{\boldsymbol{z}}$ along the orbit angular momentum, and $\hat{\boldsymbol{y}}$ completing the right-handed frame. The linearized relative equations of motion are:

$$\begin{aligned} \ddot{p}_x &= 3n^2 p_x + 2n\dot{p}_y + F_x/m, \\ \ddot{p}_y &= -2n\dot{p}_x + F_y/m, \\ \ddot{p}_z &= -n^2 p_z + F_z/m, \end{aligned} \tag{1}$$

where p_x, p_y, p_z is the relative position of the chaser in $\mathcal{H}$, m its mass, $n = 2\pi/T$ the mean orbital motion, T the orbital period, and F_x, F_y, F_z are external forces acting on the chaser. Since the x-y dynamics are coupled while z is independent, only the in-plane Hill x-y dynamics can be faithfully reproduced on a planar testbed. We therefore restrict the analysis to these two dimensions.

*Corresponding author: eliaskra@kth.se
[1]KTH Royal Institute of Technology, Stockholm, Sweden.
[2]California Institute of Technology, Pasadena, United States.
This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

In the absence of external forces, CW admit closed-form solutions via the state transition matrix $\mathbf{\Phi}(t) \in \mathbb{R}^{4\times4}$ [7]:

$$\begin{bmatrix}\boldsymbol{p}(t)\\ \boldsymbol{v}(t)\end{bmatrix} = \begin{bmatrix}\mathbf{\Phi}_{pp}(t) & \mathbf{\Phi}_{pv}(t)\\ \mathbf{\Phi}_{vp}(t) & \mathbf{\Phi}_{vv}(t)\end{bmatrix}\begin{bmatrix}\boldsymbol{p}_0\\ \boldsymbol{v}_0\end{bmatrix}, \tag{2}$$

with $c = \cos(nt)$, $s = \sin(nt)$, and

$$\mathbf{\Phi}_{pp} = \begin{bmatrix}4-3c & 0\\ 6(s-nt) & 1\end{bmatrix}, \quad \mathbf{\Phi}_{pv} = \tfrac{1}{n}\begin{bmatrix}s & 2(1-c)\\ 2(c-1) & 4s-3nt\end{bmatrix},$$
$$\mathbf{\Phi}_{vp} = n\begin{bmatrix}3s & 0\\ 6(c-1) & 0\end{bmatrix}, \quad \mathbf{\Phi}_{vv} = \begin{bmatrix}c & 2s\\ -2s & 4c-3\end{bmatrix}, \tag{3}$$

where $\boldsymbol{p} = [p_x, p_y]^\top$ is the relative position and $\boldsymbol{v} = \dot{\boldsymbol{p}}$ the relative velocity in $\mathcal{H}$, with initial conditions $\boldsymbol{p}_0 = \boldsymbol{p}(0)$ and $\boldsymbol{v}_0 = \boldsymbol{v}(0)$. The CW trajectory is thus fully determined by n, $\boldsymbol{p}_0$, and $\boldsymbol{v}_0$. This property is exploited by the trajectory generator in section II-D.

B. Nonlinear Model Predictive Control

Both controllers in the proposed architecture are formulated as a nonlinear MPC [8] optimization problem. At each discrete time k, each controller solves the following Optimal Control Problem (OCP) over a horizon of N steps:

$$J^*(\boldsymbol{x}_k) = \min_{\boldsymbol{u}_k} \sum_{n=0}^{N-1} \ell(\boldsymbol{x}(n|k), \boldsymbol{u}(n|k)) + \ell_N(\boldsymbol{x}(N|k)) \tag{4a}$$
$$\text{s.t.} \quad \boldsymbol{x}(0|k) = \boldsymbol{x}_k \tag{4b}$$
$$\boldsymbol{x}(n+1|k) = f\big(\boldsymbol{x}(n|k), \boldsymbol{u}(n|k)\big) \tag{4c}$$
$$\boldsymbol{x}(n|k) \in \mathbb{X}, \quad \boldsymbol{u}(n|k) \in \mathbb{U}. \tag{4d}$$

The stage cost $\ell(\cdot)$ is a weighted quadratic tracking cost penalizing position, velocity, attitude, and angular rate errors, as well as control effort. The terminal cost ℓ_N is defined analogously. Both controllers share this cost structure but differ in their dynamics model $f(\cdot)$, control input $\boldsymbol{u}$, and reference trajectory, as detailed below. In both cases, the state is $\boldsymbol{x} = [\boldsymbol{p},\ \boldsymbol{v},\ \mathbf{q},\ \boldsymbol{\omega}]$, where $\mathbf{q}$ is the unit quaternion from body to Hill frame and $\boldsymbol{\omega}$ is the body-frame angular velocity.

C. Thruster MPC: Spacecraft Controller

The thruster MPC represents the spacecraft's onboard controller. Its dynamics model incorporates CW accelerations, so that the controller can predict and compensate for the orbital environment it believes itself to be operating in. The continuous-time dynamics are:

$$\begin{aligned}\dot{\boldsymbol{p}} &= \boldsymbol{v},\\ \dot{\boldsymbol{v}} &= \boldsymbol{a}^{\mathrm{CW}}(\boldsymbol{p}, \boldsymbol{v}) + \tfrac{1}{m}\,\mathbf{R}(\mathbf{q})\,\boldsymbol{F}^{\mathrm{thr}},\\ \dot{\mathbf{q}} &= \tfrac{1}{2}\,\mathbf{\Omega}(\boldsymbol{\omega})\,\mathbf{q},\\ \dot{\boldsymbol{\omega}} &= \mathbf{J}^{-1}\left(\boldsymbol{\tau}^{\mathrm{thr}} - \boldsymbol{\omega}\times\mathbf{J}\boldsymbol{\omega}\right),\end{aligned} \tag{5}$$

where $\boldsymbol{a}^{\mathrm{CW}}(\boldsymbol{p}, \boldsymbol{v}) = [3n^2 p_x + 2nv_y,\ -2nv_x, -n^2 p_z]^\top$ is the CW acceleration from Eq. (1), $\mathbf{R}(\mathbf{q})$ is the rotation matrix from body to Hill frame, $\boldsymbol{u}^{\mathrm{thr}} = [\boldsymbol{F}^{\mathrm{thr}\top}, \boldsymbol{\tau}^{\mathrm{thr}\top}]^\top$ is the control input body-frame force and torque, $\mathbf{\Omega}(\boldsymbol{\omega})$ is the quaternion kinematic matrix, and $\mathbf{J}$ is the platform inertia matrix. The reference trajectory $\boldsymbol{x}^{\mathrm{ref}}(n|k)$ is a preplanned spacecraft mission trajectory. Note that the thruster controller has no knowledge of the propeller actuation system.

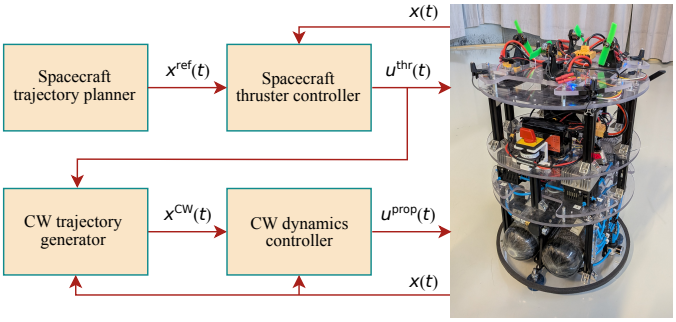

Fig. 1: Dual-actuation control architecture. The spacecraft thruster controller tracks a preplanned trajectory using a dynamics model with CW terms. Its commanded forces update the CW trajectory generator, which defines the reference for the CW dynamics controller. Both controllers are implemented as MPCs. Right: the free-flyer platform equipped with both solenoid thrusters and bidirectional propellers.

D. Propeller MPC: CW Dynamics Controller

The propeller MPC is responsible for physically applying the CW accelerations to the platform. Its dynamics model follows Eq. (5) but *without* the $\boldsymbol{a}^{\mathrm{CW}}$ term, using standard free-floating rigid-body dynamics with body-frame propeller forces and torques as control inputs $\boldsymbol{u}^{\mathrm{prop}}$.

The reference trajectory for this controller is computed analytically from the CW state transition matrix Eq. (2). As the thruster controller maneuvers the spacecraft, the initial conditions for $\mathbf{\Phi}(t)$ must be updated to reflect the applied thrust. At each update step, the trajectory generator evaluates the current trajectory at the present time t to obtain the predicted state $(\boldsymbol{p}(t), \boldsymbol{v}(t))$, applies the thruster-induced velocity change $\Delta\boldsymbol{v}$, and resets the epoch:

$$\boldsymbol{p}_0 \leftarrow \boldsymbol{p}(t), \quad \boldsymbol{v}_0 \leftarrow \boldsymbol{v}(t) + \Delta\boldsymbol{v}, \tag{6a}$$
$$\Delta\boldsymbol{v} = \tfrac{1}{m}\,\mathbf{R}(\mathbf{q})\,\boldsymbol{F}^{\mathrm{thr}}\,\Delta t, \tag{6b}$$

where $\boldsymbol{F}^{\mathrm{thr}}$ is the commanded thruster force during the time period Δt. The updated initial conditions define a new CW trajectory, ensuring continuity. Since the propeller MPC receives a full predicted trajectory rather than a single setpoint, it can anticipate the orbital motion and track it proactively, resulting in smoother force profiles and reduced tracking lag. The overall architecture is illustrated in Fig. 1.

III. Experimental Validation

A. Platform and Setup

The experiment is conducted on a single ATMOS free-flyer [6] equipped with both actuation systems, floating on air bearings over a flat floor for near-frictionless planar motion. An onboard NVIDIA Jetson runs Robot Operating System 2 (ROS 2) and communicates with a PX4-based [9] flight controller for low-level actuation control. The platform with

the extra propeller module has a total mass of approximately 19 kg and yaw inertia of 0.3 kg m^2.

The thruster system consists of eight compressed-air solenoid thrusters providing holonomic 3-degrees-of-freedom (DoF) actuation, each producing approximately 1.5 N at 10 Hz pulse width modulation (PWM). The propeller system consists of four bidirectional brushless motors with field-oriented control (FOC)-capable electronic speed controllers (ESCs), enabling continuous proportional thrust of up to 1.96 N per motor in both directions (3.92 N per body axis, 0.82 N m torque).

Both MPCs are solved asynchronously onboard through ACADOS [10]: the thruster MPC at 10 Hz, matching the thruster PWM switching frequency, and the propeller MPC at 50 Hz, exploiting the continuous actuation capability of the propellers. The thruster MPC outputs body-frame force and torque commands and the propeller MPC outputs individual propeller thrust commands, both allocated to actuators through PX4.

B. Experiment Design

The orbital period for typical Low-Earth-Orbit (LEO) satellites ranges from 90 min to 130 min, yielding mean motions of $n \approx 1\,\text{mrad}\,\text{s}^{-1}$. At this rate, the CW accelerations over the approximately 15 m^2 flat floor would be negligibly small, and the dynamics would evolve far too slowly for practical experiments. We therefore use a scaled period of $T = 2$ min, which produces CW accelerations within the propeller actuation range and allows a full demonstration within a few orbital periods.

The experiment consists of two phases. In the first, the spacecraft moves between three consecutive setpoints which are *unstable* under CW dynamics (the chaser would drift away without active station keeping). Thus, testing the thruster MPC's ability to hold arbitrary setpoints while the propellers apply CW forces of comparable magnitude.

In the second phase, the spacecraft follows two stable relative orbits, defined as 2:1 elliptical trajectories that are natural solutions of the unforced CW equations. These satisfy the drift-free condition $v_{y,0} = -2n\,p_{x,0}$ and take the form:

$$p_x(t) = \alpha \sin(nt + \varphi), \quad p_y(t) = p_{y,c} + 2\alpha \cos(nt + \varphi), \quad (7)$$

where α is the semi-minor axis amplitude, φ the phase, and $p_{y,c}$ is the along-track center offset. The two ellipses use $(\alpha_1, p_{y,c,1}) = (0.5,\ 2.0)$ m and $(\alpha_2, p_{y,c,2}) = (0.375,\ 1.25)$ m. During this phase, the thruster MPC should require minimal control effort since the reference trajectory is dynamically consistent with the CW forces applied by the propellers.

C. Results

The experimental results are shown in Fig. 2. The thruster MPC tracks the preplanned spacecraft trajectory closely throughout both phases, with position errors remaining below 20 mm during the stable ellipse tracking. During the stable ellipse phase, thruster activity is visibly reduced as the reference trajectory becomes dynamically consistent with the applied CW accelerations. The propeller MPC also tracks its reference well, though with a larger offset. The CW trajectory is updated via Eq. (6) from commanded thrust and position, leaving uncertainty in the applied accelerations uncorrected.

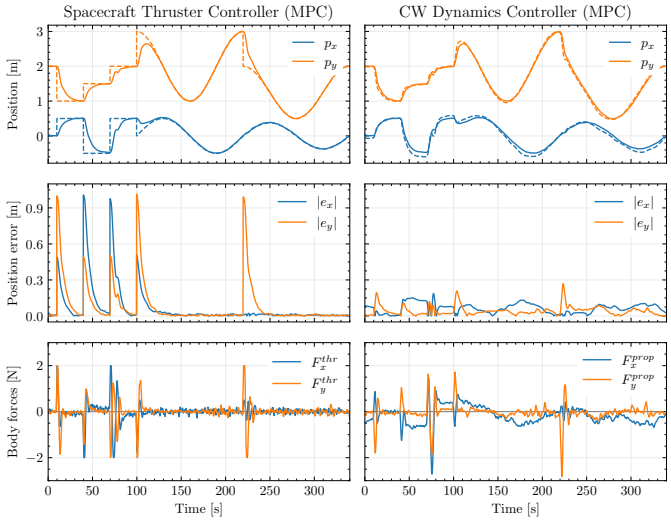

Fig. 2: Experimental results. The chaser first transitions between three static setpoints; for $t \geq 100$ s, it follows two consecutive stable relative-orbit trajectories, during which the propellers apply the majority of the force. Top: measured (solid) and reference (dashed) positions. Middle: absolute position tracking error. Bottom: commanded body forces.

The contrast between the two actuation modalities is evident in the bottom panels: the propellers deliver a sustained, slowly varying force of roughly ±0.6 N that physically realizes the CW accelerations, while the thrusters fire only in short corrective bursts bounded by ±0.15 N, confirming the intended actuation decoupling.

IV. Conclusion

This paper presents a dual-actuation architecture for physically emulating relative orbital dynamics on a planar free-flyer testbed. By separating spacecraft maneuvering (solenoid thrusters) from CW dynamics emulation (bidirectional propellers) into two parallel MPCs with independent dynamics models, the approach decouples controller design from environment simulation at the actuation level. Experimental results with time-scaled CW dynamics demonstrate that the thruster MPC accurately tracks spacecraft trajectories despite concurrent propeller forces of comparable magnitude. The experiment was limited to translational maneuvers; attitude tracking proved challenging due to the finite force rate of the propellers and requires further tuning.

Future work includes extending to simultaneous attitude maneuvers, validating against high-fidelity orbital simulators, using the propellers to actively compensate for known testbed disturbances such as floor unevenness, and multi-agent proximity operations with physically emulated orbital dynamics.

REFERENCES

[1] M. Wilde, C. Clark, and M. Romano, "Historical survey of kinematic and dynamic spacecraft simulators for laboratory experimentation of on-orbit proximity maneuvers," *Progress in Aerospace Sciences*, vol. 110, p. 100552, Oct. 2019.

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

[3] J. L. Schwartz, M. A. Peck, and C. D. Hall, "Historical Review of Air-Bearing Spacecraft Simulators," *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 4, pp. 513–522, 2003.

[4] W. H. Clohessy and R. S. Wiltshire, "Terminal Guidance System for Satellite Rendezvous," *Journal of the Aerospace Sciences*, vol. 27, no. 9, pp. 653–658, 1960.

[5] M. Ciarcià, R. Cristi, and M. M. Romano, "Emulating Scaled Clohessy–Wiltshire Dynamics on an Air-Bearing Spacecraft Simulation Testbed," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 10, pp. 2496–2510, 2017.

[6] P. Roque, S. Phodapol, E. Krantz, J. Lim, J. Verhagen, F. J. Jiang, D. Dörner, H. Mao, G. Tibert, R. Siegwart, I. Stenius, J. Tumova, C. Fuglesang, and D. V. Dimarogonas, "Toward Open-Source and Modular Space Systems With ATMOS," *IEEE Transactions on Field Robotics*, vol. 3, pp. 141–161, 2026.

[7] H. Schaub and J. L. Junkins, *Analytical Mechanics of Space Systems*, 2nd ed. AIAA Education Series, 2009.

[8] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl, *Model Predictive Control: Theory, Computation, and Design*, 2nd ed. Nob Hill Publishing, 2020.

[9] L. Meier, D. Honegger, and M. Pollefeys, "PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2015, pp. 6235–6240.

[10] R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. v. Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, and M. Diehl, "Acados—A modular open-source framework for fast embedded optimal control," *Math. Program. Comput.*, vol. 14, no. 1, pp. 147–183, 2022.