

Control for an Omnidirectional Multi-rotor UAV for Space Applications*

Riley McCarthy[†]

University of New Mexico, Albuquerque, New Mexico

Tristan Thomas[‡]

Clemson University, Clemson, South Carolina

Claus Danielson[§]

University of New Mexico, Albuquerque, New Mexico

Sean Phillips[¶]

Air Force Research Laboratory, Kirtland AFB, New Mexico

Rafael Fierro^{||}

University of New Mexico, Albuquerque, New Mexico

Omnidirectional multi-rotor platforms are well suited to verify, validate, and test multi-satellite collaborative autonomous algorithms in a terrestrial laboratory. Unlike conventional multi-rotors, an omnidirectional configuration enables the vehicle to produce an arbitrary combination of forces and torques to emulate satellite dynamics unattainable with conventional designs. We achieve this by: i) Controlling the omnidirectional multi-rotor to track a reference trajectory, done here using custom controllers programmed within a ROS2 framework. ii) On-the-fly generation of a reference trajectory identical to the one experienced by a real satellite. In this work we demonstrate the capabilities of such an omnidirectional multi-rotor, as well as provide examples for generating these trajectories in accordance with the Clohessy-Wiltshire relative motion equations.

I. Introduction

Due to increased availability of access to the space domain, there has been a significant rise in interest. More specifically, this rise is due to the fact that the cost of satellite launching has been dramatically reduced in recent years as well as the commercial prospect of satellite global internet services and on-orbit manufacturing and servicing [1, 2]. However, even though the cost of launches has been reduced in recent years, the cost is not free. This means that the verification, validation and testing of novel algorithms remains an important consideration in deploying advanced algorithms in this domain. This therefore necessitates a laboratory methodology for testing close proximity satellite interactions in a safe terrestrial environment.

Due to the inherent nonlinear dynamics of satellite systems and uniqueness of the space domain, it is difficult to properly develop a terrestrial based facility to test advanced algorithms. There currently exist several mechanisms for testing advanced satellite algorithms, each with their own trade offs. The most widely used testing platform is high fidelity simulations which allows for dynamic adjustments of granularity and coupled with such methods as falsification methods [3] or other reachability analysis tools mentioned [4], a full state space exploration may not need to be explored to have some performance guarantees. However, pure simulations are often idealized considerations and usually do not

*Approved for public release; distribution is unlimited. Public Affairs approval #AFRL-RV-2043

[†]Mechanical Engineering, Albuquerque, New Mexico

[‡]Mechanical Engineering, Clemson, South Carolina

[§]Mechanical Engineering, Albuquerque, New Mexico

[¶]Research Mechanical Engineer, Space Vehicles Directorate, Kirtland AFB, New Mexico

^{||}Electrical and Computer Engineering, Albuquerque, New Mexico

consider interactions through hardware tools [5]. One difficulty in achieving a useful platform for terrestrial emulation of the space environment is the lack of friction in the space domain. Leveraging omni-directional wheeled robots, in [6], a planar spacecraft multi-agent dynamics were emulated. Linear and spherical air bearings allow for effective removal of surface drag which allows for near frictionless manipulations of platforms to better mimic actuators in the space domain; see [7–16]. Each spherical air bearing platform also has limitations. These limitations are in terms of restrictions of spacial movements either in degrees of freedom (like planar air bearing) or physical space constraints (like spherical air bearing). Some research has been done into using unmanned aerial vehicles as a means of testing satellite autonomy algorithms [17], using the same close proximity spacecraft dynamics as in this paper. In this paper, we consider the case of increasing the fidelity of spacecraft emulation tools through leveraging 6 degree of freedom omnidirectional aerial platforms, [18–20].

In this paper, we develop such an aerial testbed platform that will be well suited for satellite emulation. In particular, we leverage an omnidirectional multi-rotor aerial vehicle design. Then, we design a control methodology to actuate the platform both translationally and attitudinally to follow trajectories mimicking those of a spacecraft.

This paper is organized as follows. Section II.A gives preliminaries on the close proximity spacecraft dynamics in the Hill's frame. Section II.B presents the translational and rotational equations of motion of an omnidirectional multi-rotor vehicle. In Section III we detail the design and control of the omnidirectional multirotor, and provide examples for trajectory tracking. Section V describes the trajectory generation for mimicking the motion of a satellite according to the translational dynamics from Section II.A.

II. Spacecraft Emulation

A. Close Proximity Spacecraft Dynamics

In this paper, we are interested in testing relative motion satellite systems with the aim of developing omnidirectional robotics to follow such trajectories. In the remainder of this section, we will introduce the translational and rotation equations of motions governing the continuous-time evolution of the spacecraft. This will provide the trajectories that the omnidirectional platform will follow.

1. Translational Dynamics

In this paper, we are interested in studying resilient formation control of spacecraft in close proximity. Therefore, we leverage the classical relative motion Hill's frame (also referred as the Clohessy-Wiltshire frame (CW)) to describe the natural motion of each agent in the network. Figure 1a shows the case of a single agent spacecraft relative to the chief satellite. See [21] for more information on the Hill's frame.

Consider an Earth centered and fixed inertial reference-frame. For each the deputy spacecraft and the chief spacecraft x_c (also sometimes referred to as the chief), the equations of motion are given by

$$\ddot{R}_j = -\frac{\mu}{|R_j|^3} R_j + \frac{1}{m_j} W_j \quad (1)$$

for each $j \in \{c, d\}$, where the subscripts d and c denote the deputy and chief spacecraft, respectively, μ is the gravitational constant parameter, m_j is the mass of the j -th spacecraft and $W_j \in \mathbb{R}^3$ are the external forces from actuation (perturbations can also be included through this term, but is outside the scope of this paper's discussion). The equations of motion in (1) can yield stable, elliptical orbits about the center of gravity of the frame (approximately the center of the Earth). Now the following assumptions are made to reduce the nonlinear dynamics in (1) into the Hill's frame.

Assumption 1 *The primary force acting on all spacecraft is spherical two-body gravity generated by the central body Earth.*

Assumption 2 *The mass loss of the spacecraft is significantly smaller than the total mass of the spacecraft.*

Assumption 3 *The target spacecraft is in a circular orbit with radius $|R_t| = r_0$.*

Assumption 4 *The distance from the target spacecraft to each other spacecraft is significantly less than that of the distance from the target spacecraft to the center of the Earth.*

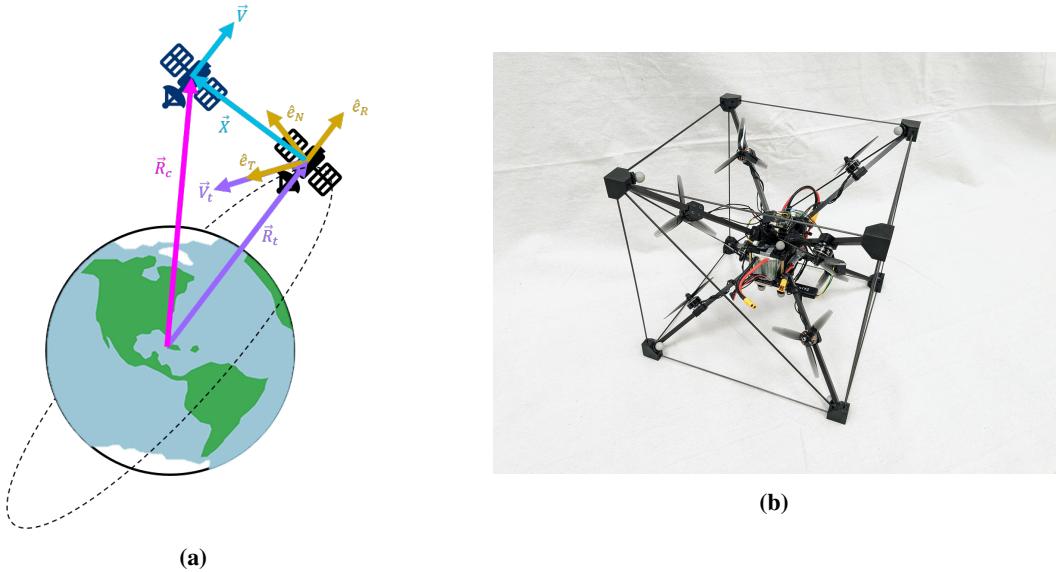


Fig. 1 The Clohessy-Wiltshire (CW) frame (a), the omnidirectional multi-rotor vehicle (b).

Assumptions 2 - 4 are necessary to simplify (1) to yield the classical Clohessy-Wiltshire (CW) relative motion equations.

With the target spacecraft in an equilibrium orbit, we attach a non-inertial frame to it with unit axis vectors e_i, e_j and e_k in the radial track, the in-track and the out-of-plane orbital positions, respectively. The relative position vector x_i , for each $i \in \{c, d\}$, is given by the relation

$$OR_i = \begin{bmatrix} r_0 \\ 0 \\ 0 \end{bmatrix} + x_i$$

where the rotation matrix O transforms the inertial frame to the relative frame. Utilizing Assumptions 3 and 4, the relative equations of motion abide by the following dynamics:

$$\begin{aligned} \dot{x}_i &= v_i \\ \dot{v}_i &= A_x x_i + A_v v_i + B u_i \end{aligned} \tag{2}$$

where $u_i \in \mathbb{R}^3$ is the control signal expressed in the relative frame and the matrices A_x, A_v and B are given by

$$A_x = \begin{bmatrix} 3n^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -n^2 \end{bmatrix}, \quad A_v = \begin{bmatrix} 0 & 2n & 0 \\ -2n & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} \frac{1}{m_i} & 0 & 0 \\ 0 & \frac{1}{m_i} & 0 \\ 0 & 0 & \frac{1}{m_i} \end{bmatrix} \tag{3}$$

where $n = \sqrt{\frac{\mu}{r_0^3}}$ is the mean motion on the target spacecraft's circular orbit (this is also the circular angular velocity magnitude of the target spacecraft in the inertial frame). In the case of the chief satellite satisfying $x_c = v_c = 0$, it is the center of the Hill's frame.

Note that (2) has pure complex eigenvalues with zero real component. This results in a stable elliptical orbit in the e_i - e_j plane. These ellipses have their center along the e_i axis, a semi-major axis of length $2a$ along the e_j axis, and a semi-minor axis of length a in the e_i axis, for any given constant $a > 0$. These orbits are important as they do not consume any fuel to achieve which extends the mission life of the spacecraft.

B. Omnidirectional Multi-rotor Vehicle

In this section, we model the dynamics of the omnidirectional multi-rotor vehicle. The translational and rotational motion of a drone with n propellers are modeled by

$$\dot{q}(t) = \frac{1}{2}q(t) \otimes \omega(t) \quad (4a)$$

$$J\dot{\omega}(t) = -D_\omega\omega(t) + T_\omega u(t) - \omega(t) \times J\omega(t) \quad (4b)$$

$$m\ddot{y}(t) = -R(q)D_yR(q)^\top \dot{y}(t) + R(q(t))T_x u(t) - g \quad (4c)$$

where the quaternion $q \in \mathbb{H}$ describes the orientation of the drone, $y \in \mathbb{R}^3$ is the drone position in the inertial-frame, $\omega \in \mathbb{R}^3$ is the angular velocity in the drone-frame. With abuse of notation, $q \otimes \omega$ denotes the quaternion multiplication of q and $(0, \omega)$. The matrices $D_y \in \mathbb{R}^{3 \times 3}$ and $D_\omega \in \mathbb{R}^{3 \times 3}$ contain the damping coefficients. The rotation matrix $R(q) \in \mathbb{R}^{3 \times 3}$ mapping drone-frame to the inertial-frame depends on the orientation q of the drone. The matrices $T_x \in \mathbb{R}^{3 \times n}$ and $T_\omega \in \mathbb{R}^{3 \times n}$ describe the thrust and torque directions, respectively, of each propeller

$$T_x = [b_1, b_2, \dots, b_n] \quad (5a)$$

$$T_\omega = [b_1 \times l_1, b_2 \times l_2, \dots, b_n \times l_n] \quad (5b)$$

where b_i is the thrust direction and l_i is the vector from the center-of-mass of the drone to the i -th propeller.

III. Experimental Setup

A. Hardware

In this section, we outline the hardware components used to assemble the omnidirectional multi-rotor vehicle. The overall design was based on PX4's "omnicopter" [22], which is itself based on the design in [23]. Minor design changes were made to the vehicle to facilitate flying using a Vicon MX optical motion capture system instead of a GPS. This primarily included mounting an Odroid-XU4 as an onboard companion computer linked to the Kakute H7 flight control unit over a serial connection. A bill of materials is shown in Table 1.

Table 1 Omnidirectional multi-rotor bill of materials.

Item	Quantity
Holybro Kakute H7 V1	1
FrSky R-XSR Reciever	1
Tekko32 F4 4in1 ESC	2
CP2102 USB 2.0 to TTL Module	1
Odroid XU4	1
USB WiFi Module	1
Lumenier ZIP 2207.5/2500KV	8
HQProp 3D 5X3.5X3 3-Blade Propeller	8
3300 mAh 6s LiPo Battery	1
8mm × 7mm × 248mm Square Carbon Fiber Tube	8
328mm × 3mm O.D. Carbon Fiber Rod	12
465mm × 3mm O.D. Carbon Fiber Rod	6

B. Reference Trajectory Tracking via PX4's Offboard Mode

Reference tracking of the omnidirectional multi-rotor is achieved using a combination of PX4's control allocation features and custom PID controllers, communicating via ROS2 with the px4_ros_com package [24] and PX4's "Offboard

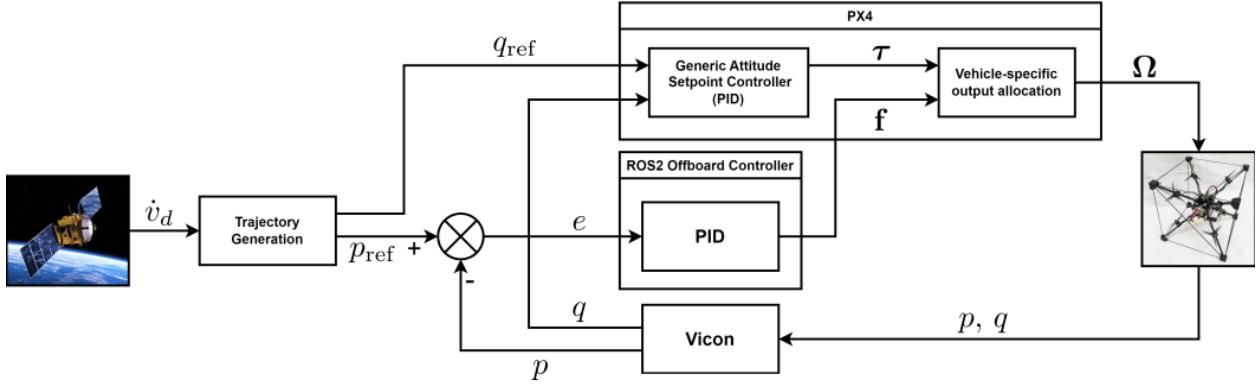


Fig. 2 Omnicopter control block diagram.

Mode". This "Offboard Mode" allows for the communication of various desired setpoints to the vehicle, which the vehicle then attempts to achieve. In this case, these setpoints are based on our reference trajectory $r_{ref} \in \mathbb{R}^7$ defined as:

$$r_{ref} = [p_{ref} \ q_{ref}]^T = [x_{ref} \ y_{ref} \ z_{ref} \ w_{ref} \ i_{ref} \ j_{ref} \ k_{ref}]^T. \quad (6)$$

For the omnidirectional multi-rotor in this work, we use the vehicle_attitude_setpoint topic. It contains a thrust setpoint $\mathbf{f} \in \mathbb{R}^3$ and a desired quaternion attitude q_{ref} . The desired thrust setpoint \mathbf{f} has components $\mathbf{f}_x, \mathbf{f}_y$, and \mathbf{f}_z in the body frame x, y , and z directions respectively. They are determined by a custom PID controller using the error e to the desired position p_{ref} . PX4 uses its own PID controller, given the error between the reference attitude q_{ref} and current attitude q , to determine the torque setpoint of the vehicle $\boldsymbol{\tau}$.

These forces and torques are then converted to angular velocities, and eventually PWM signals, for each motor by PX4's dynamic control allocation feature [25] according to the following equation,

$$\mathbf{u} = \boldsymbol{\Gamma} \boldsymbol{\Omega} \implies \boldsymbol{\Omega} = \boldsymbol{\Gamma}^+ \mathbf{u} \quad (7)$$

where $\mathbf{u} = [\mathbf{f} \ \boldsymbol{\tau}]^T$ is the vector of desired forces and torques, for an n -propeller multirotor $\boldsymbol{\Omega} \in \mathbb{R}^n$ is a vector of squared propeller angular velocities, and $\boldsymbol{\Gamma}^+$ is the pseudo-inverse of the vehicle specific control allocation matrix.

C. Control Allocation

The control allocation matrix $\boldsymbol{\Gamma} \in \mathbb{R}^{6 \times n}$ is a matrix of values transforming the squared angular velocities of each propeller into forces and torques effecting the multi-rotor. It is constructed as follows:

$$\boldsymbol{\Gamma} = \begin{bmatrix} k_f b_1 & \dots & k_f b_n \\ k_f \mathbf{S}(l_1)b_1 + s_1 k_\tau b_1 & \dots & k_f \mathbf{S}(l_n)b_n + s_n k_\tau b_n \end{bmatrix} \quad (8)$$

where k_f and k_τ are thrust force and drag torque constants for the vehicles propellers based on physical properties, usually determined empirically. $\mathbf{S}(l_n)$ is the skew symmetric matrix representation of l_n , and s_n is sign of the angular velocity of the n -th propeller.

For over-actuated vehicles such as the omnidirectional multi-rotor described in this work, there are multiple solutions to the control allocation problem described in Equation 7. To obtain a single solution, PX4 utilizes the Moore-Penrose pseudo-inverse method.

D. ROS2 PID Control

The computation of the thrust setpoint \mathbf{f} as described in Section III.B is done using the following equations.

$$e[k] = \mathbf{R}(q[k])e_w[k] = \mathbf{R}(q[k])(p_{ref}[k] - p[k]) \quad (9a)$$

$$\mathbf{f}[k] = K_P e[k] + K_I \sum_{n=0}^k e[n] + K_D (e[k] - e[k-1]) \quad (9b)$$

where k is the discrete time step of the current iteration, and $\mathbf{R}(q[k])$ is a rotation matrix transforming vectors from the world frame to the body frame as a function of the current quaternion orientation of the omnidirectional multi-rotor. e_w is the error between reference position and actual position in the world frame, e is this same error rotated into the body frame.

IV. Capability Demonstration

In the following section we show two experiments done to validate the capabilities of the omnidirectional multi-rotor. Plots of the vehicles quaternion orientation are included here adjacent to plots of the vehicles position in \mathbb{R}^3 relative to the desired path. Experiment IV.A shows the omnicopter pitching, rolling, and then yawing 360 degrees, Experiment IV.B shows the omnicopter translating in a circle while rotating about an axis perpendicular to the circle. Additional experimental results, as well as further descriptions of setup and usage for this system can be found in [26].

A. Static Hover With Rotation

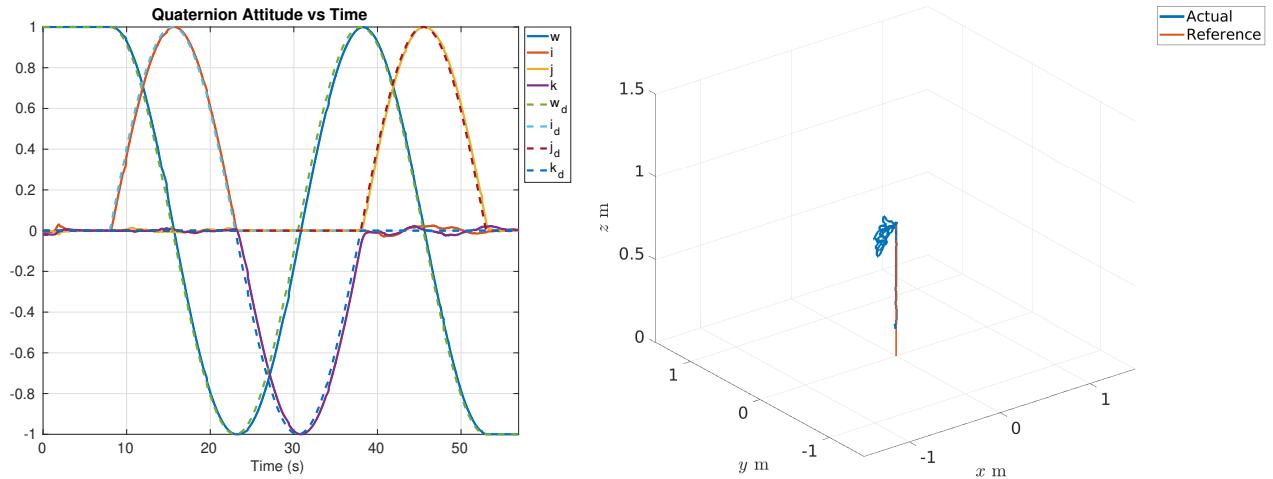


Fig. 3 Omnicopter hovering while rotating 360 degrees about each axis.

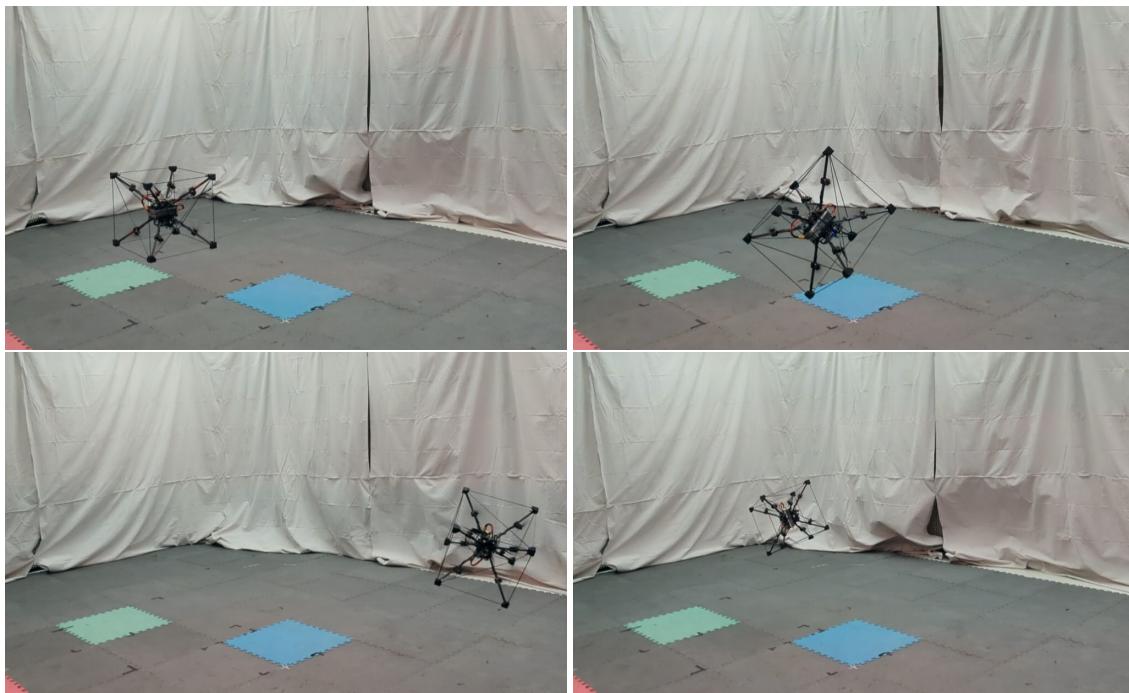


Fig. 5 Omnicopter in flight.

B. Simultaneous Translation and Rotation

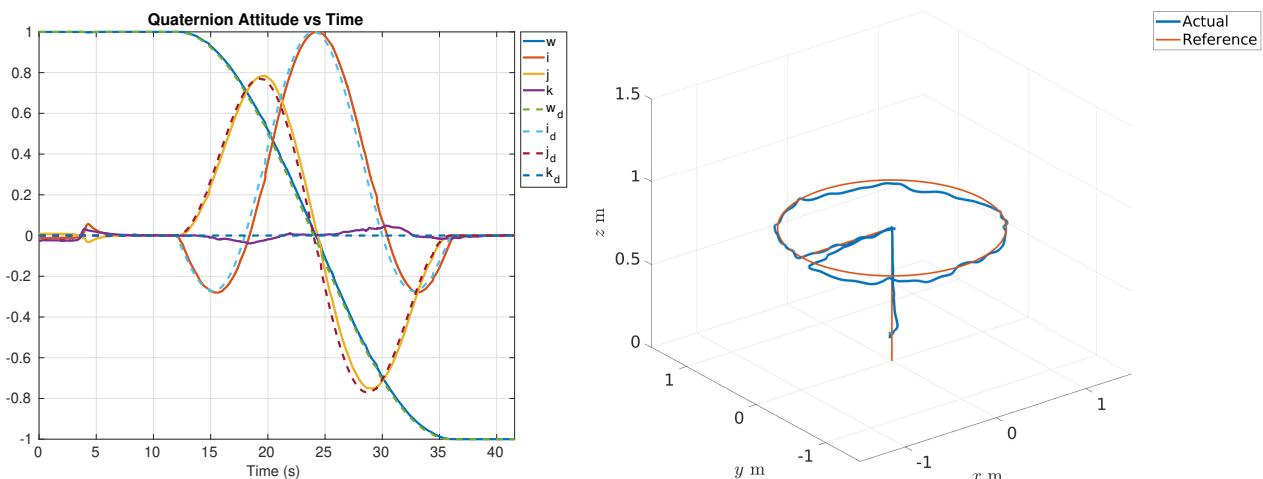


Fig. 4 Omnicopter translating in a circle while rotating.

V. Trajectory Generation to Mimic Relative Motion Satellite Dynamics

In this section we describe the logic and methodology behind the real-time trajectory generation the omnidirectional multi-rotor will attempt to follow. Using the Clohessy-Wiltshire relative motion equations from Section II.A, it is possible to determine the acceleration of a fictitious "deputy" satellite relative to a "chief" satellite. Using kinematics, these accelerations are converted into positions $p_{\text{ref}} \in \mathbb{R}^3$ to be used as the reference position for the omnidirectional multi-rotor.

The acceleration of the deputy in relation to the chief is described by Equation 2, specifically:

$$\dot{v}_d = A_x x_d + A_v v_d + Bu_d . \quad (10)$$

Using kinematics, the relative position and velocity of the deputy satellite at discrete time step $k + 1$ are defined by:

$$p_{\text{ref}}[k + 1] = p_{\text{ref}}[k] + v_d[k]\Delta t + \frac{1}{2}\dot{v}_d[k]\Delta t^2 \quad (11a)$$

$$v_d[k + 1] = v_d[k] + \dot{v}_d[k]\Delta t \quad (11b)$$

where Δt is the amount of time between discrete time steps.

VI. Gazebo Simulation Results

Here we show the results of a Gazebo simulation of the omnidirectional multi-rotor attempting to track a reference trajectory defined by Equation 11a. Consider the following initial conditions: A chief satellite with $R_c = 200$ km orbiting Earth, and a deputy satellite with a mass of 60 kg beginning one meter radially inward along the chiefs \hat{e}_R axis with zero initial relative velocity. Meaning $x_d[0] = p_{\text{ref}}[0] = [-1 \ 0 \ 0]^T$ and $v_d[0] = 0_{3 \times 1}$. The deputy satellite makes no effort to change its acceleration, meaning the control input vector $u_d = 0_{3 \times 1}$ for all time.

Knowing $\mu = 3.986E14 \text{ m}^3\text{s}^{-2}$ for Earth, $n = 0.223 \text{ s}^{-1}$. Meaning A_x , A_v , and B for Equation 2 are then

$$A_x = \begin{bmatrix} 0.149 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -0.049 \end{bmatrix}, \quad A_v = \begin{bmatrix} 0 & 0.446 & 0 \\ 0.446 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} \frac{1}{60} & 0 & 0 \\ 0 & \frac{1}{60} & 0 \\ 0 & 0 & \frac{1}{60} \end{bmatrix} \quad (12)$$

The reference position for the omnidirectional multi-rotor and relative velocity of the deputy satellite at discrete time step $k = 1$ from Equations 11a and 11b are defined as:

$$p_{\text{ref}}[1] = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} + 0_{3 \times 1}\Delta t + \frac{1}{2}\dot{v}_d[0]\Delta t^2 \quad (13a)$$

$$v_d[1] = 0_{3 \times 1} + \dot{v}_d[0]\Delta t \quad (13b)$$

with $\Delta t = 0.01$ s. This trajectory is updated every 0.01 seconds until the simulation is ended by the user. Pesudocode demonstrating how this trajectory generation is implemented is shown in Algorithm 1. This generated trajectory, and the actual path of the omnidirectional multi-rotor, can be seen in Fig. 6.

VII. Conclusions

Here we have effectively demonstrated the ability of an omnidirectional multi-rotor to emulate the dynamics of one satellite in close proximity to another according to the relative equations of motion in the Hill's frame. With this system it is possible to test new controllers for satellites in a real world environment before sending them into space. Based on the way the trajectory is generated using Equation 2, any controller that produces inputs that can be formulated as forces acting in the body frame of the satellite can be applied and emulated in the real world.

While the way this trajectory generation is constructed in this paper is good for emulating the motion of the deputy satellite relative to the frame of the chief, this motion is effected only by the vector of force inputs produced by the

Algorithm 1: Trajectory Generation

Input: t : Time after startup.
Output: p_{ref} : Goal position for Omnicopter.
// This function runs every 0.01 seconds

- 1 $\Delta t = 0.01$
- 2 // Set position of chief satellite 5m above ground
- 2 $x_c = [0, 0, -5]$
- 3 // Set initial conditions of Omnicopter/deputy satellite
- 3 $p_{\text{ref}} = [0, 0, 0]$
- 4 $v_d = [0, 0, 0]$
- 5 $\dot{v}_d = [0, 0, 0]$
- 6 $u_d = [0, 0, 0]$
- 7 $m = 60$
- 8 $\mu = 3.986E14$
- 9 $r_0 = 200000$
- 10 $n = \sqrt{\frac{\mu}{r_0^3}}$
- 11 **while** $t < 10$ **do**
 - // Move Omnicopter to starting position over 10 seconds, 5m up and 1m in -x
 - 12 $p_{\text{ref}} = [-1(t/10), 0, -5(t/10)]$
- 13 **while** $t \geq 10$ **do**
 - // Eliminate z offset
 - 14 $p_{\text{ref}} = p_{\text{ref}} - x_c$
 - // Begin updating reference trajectory according to CW equations, using kinematics
 - 15 $\dot{v}_d = A_x p_{\text{ref}} + A_v v_d + B u_d$
 - 16 $p_{\text{ref}} += v_d \Delta t + \frac{1}{2} \dot{v}_d \Delta t^2$
 - 17 $v_d += \dot{v}_d \Delta t$
 - // Add back z offset for Omnicopter
 - 18 $p_{\text{ref}} = p_{\text{ref}} + x_c$

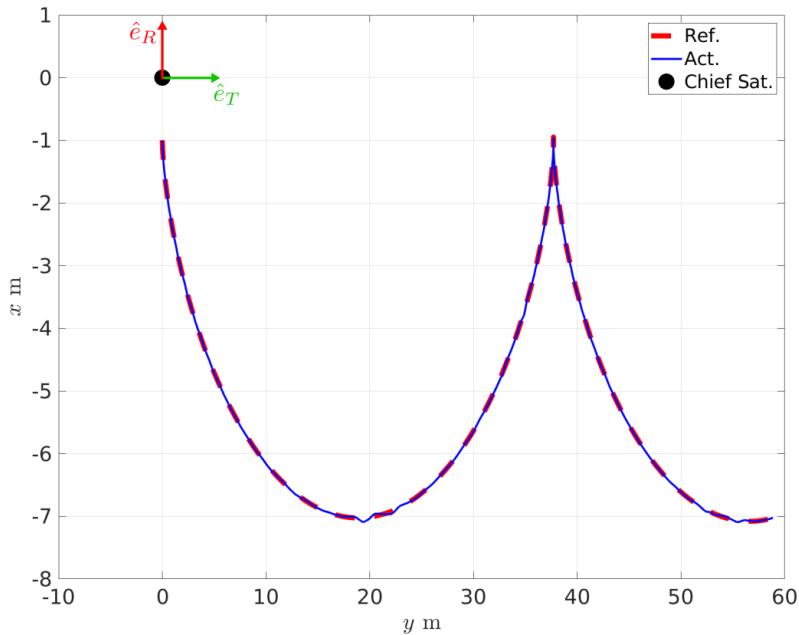


Fig. 6 Omnicopter tracking relative motion trajectory.

deputy. In reality, both satellites may be changing their accelerations at the same time. Emulations of this motion violate the conditions for the Clohessy-Wiltshire equations, and new relative motion equations are required.

Many projects in this field involve simulating interactions with satellites that are unable to produce control inputs at all. For example, an experiment attempting to emulate a satellite equipped with a robotic arm approaching a malfunctioning satellite is possible using our omnidirectional multi-rotor system. With a robotic arm located within the Vicon system and the omnidirectional multi-rotor moving and rotating around it according to the relative motion equations described above.

Acknowledgements

This material is based upon work supported by the Air Force Research Laboratory (AFRL) and Air Force Office of Scientific Research under agreements FA9453-18-2-0022 and FA9550-22-1-0093. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the United States Air Force.

References

- [1] Phillips, S., Petersen, C., and Fierro, R., “Robust, Resilient, and Energy-Efficient Satellite Formation Control,” *Intelligent Control and Smart Energy Management*, Springer, 2022, pp. 223–251.
- [2] Soderlund, A. A., and Phillips, S., “Autonomous Rendezvous and Proximity Operations of an Underactuated Spacecraft via Switching Controls,” *Proceedings of the AIAA SCITECH 2022 Forum*, 2022.
- [3] Brewer, J. M., Tsotras, P., Lang, K., and Phillips, S., “Falsification-based Verification for Multi-Mode Spacecraft Attitude Control Systems,” *2021 American Control Conference (ACC)*, 2021, pp. 4296–4301.
- [4] Lang, K., Klett, C., Hawkins, K., Feron, E., Tsotras, P., and Phillips, S., *Formal Verification Applied to Spacecraft Attitude Control*, 2021.
- [5] Cellucci, D., Cramer, N. B., and Frank, J. D., *Distributed Spacecraft Autonomy*, 2020.
- [6] Mercier, M., Phillips, S., Shubert, M., and Dong, W., “Terrestrial Testing of Multi-Agent, Relative Guidance, Navigation, and Control Algorithms,” *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2020, pp. 1488–1497.
- [7] Cho, D.-M., Jung, D., and Tsotras, P., “A 5-dof Experimental Platform for Autonomous Spacecraft Rendezvous and Docking,” *AIAA Infotech at Aerospace Conference and Exhibit and AIAA Unmanned*, 2009. doi:10.2514/6.2009-1869.
- [8] Yang, Y., and Cao, X., “Design and Development of the Small Satellite Attitude Control System Simulator,” 2006. doi: 10.2514/6.2006-6124.
- [9] Schwartz, J. L., Peck, M. A., and Hall, C. D., “Historical Review of Air-Bearing Spacecraft Simulators,” *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 4, 2003, pp. 513–522.
- [10] Schwartz, J., “The Distributed Spacecraft Attitude Control System Simulator: From design concept to decentralized control,” 2004.
- [11] Peck, M., and Cavender, A., “An AirbearingBased Testbed for Momentum-Control Systems and Spacecraft Line of Sight,” 2008.
- [12] Olsen, T., *Design of an Adaptive Balancing Scheme for Small Satellite Attitude Control Simulator (SSACS)*, Utah State University. Department of Mechanical and Aerospace Engineering, 1995. URL <https://books.google.fr/books?id=-gMe0AAACAAJ>.
- [13] Kabganian, M., Nabipour, M., and Fani saberi, F., “Design and implementation of attitude control algorithm of a satellite on a three-axis gimbal simulator,” *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 229, 2014. doi:10.1177/0954410014526380.
- [14] Das, A., Berg, J., Norris, G., Cossey, D., Strange, T., and Schlaegel, W., “ASTREX-a unique test bed for CSI research,” *29th IEEE Conference on Decision and Control*, 1990, pp. 2018–2023 vol.4. doi:10.1109/CDC.1990.203978.
- [15] Crowell, C., “Development and analysis of a small satellite attitude determination and control system testbed,” 2011.
- [16] Boynton, R., “Using a Spherical Air Bearing to Simulate Weightlessness,” 1996.
- [17] Lippay, S. Z., Shubert, M., Soderlund, A. A., Phillips, S., Baker, D., and Li, B., “Emulation of Close-Proximity Spacecraft Dynamics in Terrestrial Environments Using Unmanned Aerial Vehicles,” *SciTech*, 2024.
- [18] Barari, A., and Ferguson, P., “Dynamic Feasibility of Space Environment Emulation using an Omnidirectional Drone,” *2021 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*, IEEE, 2021, pp. 66–71.
- [19] Barari, A., Dion, R., Jeffrey, I., and Ferguson, P., “Testing satellite control systems with drones,” *IEEE Potentials*, Vol. 41, No. 1, 2021, pp. 6–13.
- [20] Hamandi, M., Usai, F., Sablé, Q., Staub, N., Tognon, M., and Franchi, A., “Design of multirotor aerial vehicles: A taxonomy based on input allocation,” *The International Journal of Robotics Research*, Vol. 40, No. 8-9, 2021, pp. 1015–1044.
- [21] de Ruiter, A., Damaren, C., and Forbes, J., *Spacecraft Dynamics and Control: An Introduction*, Wiley, 2012.
- [22] PX4, “Omnicopter,” , 2023. https://docs.px4.io/main/en/frames_multicopter/omnicopter.html.

- [23] Brescianini, D., and D'Andrea, R., "An omni-directional multirotor vehicle," *Mechatronics*, Vol. 55, 2018, pp. 76–93. doi: <https://doi.org/10.1016/j.mechatronics.2018.08.005>, URL <https://www.sciencedirect.com/science/article/pii/S0957415818301314>.
- [24] PX4, "px4_ros_com," https://github.com/PX4/px4_ros_com, 2023.
- [25] Küng, B., and Fuhrer, S., "The New Dynamic Mixing System of PX4," , 2023. URL <https://www.youtube.com/watch?v=NQwepGLMzZ4>, pX4 Developer Summit.
- [26] McCarthy, R., "Control of Fully-Actuated Aerial Manipulators and Omni-Directional Multirotors," Master's thesis, University of New Mexico, 2023.