Model Predictive Control for an Omnidirectional Multi-rotor UAV for Space Applications

Riley McCarthy and Claus Danielson*
University of New Mexico, Albuquerque, New Mexico

Sean Phillips†
Air Force Research Laboratory, Kirtland AFB, New Mexico

Michal Rittikaidachar and 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. However, to achieve the required high-performance closed-loop control, conventional embedded control techniques are inadequate. In this abstract, we proposed a novel Model Predictive Control (MPC) for omnidirectional UAVs. Numerical simulations validate the benefits of the proposed MPC methodology.

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 needed to be explored to have some performance guarantees. However, pure simulations are often idealized considerations and usually do not 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). In this paper, we consider the case of increasing the fidelity of spacecraft emulation tools through leveraging 6 degree of freedom omnidirectional aerial platforms, [17–19].

In this paper, we develop such a 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

*Mechanical Engineering, Albuquerque, New Mexico
†Research Mechanical Engineer, Space Vehicles Directorate, Kirtland AFB, New Mexico
‡Electrical and Computer Engineering, Albuquerque, New Mexico

the platform both translationally and attitudinally to follow trajectories of a spacecraft dynamics. To acheive this, we leverage a model predictive control (MPC) for multi-rotor vehicles [20] and learning policies [21–23].

1) Physics-Inspired Temporal Learning of Quadrotor Dynamics for Accurate Model Predictive Trajectory Tracking [24]
2) This work is motivated by the recent results presented in [25] where a deep-learning MPC for high-speed agile quadrotors (Neural-MPC) is developed.

This paper is organized as follows. Section II gives preliminaries on the closed proximity spacecraft dynamics in the Hill's frame. Section II.B presents the translational and rotational equations of motion of an omnidirectional mutlti-rotor vehicle. In Section III, we outline a design to mimic motion control of a satellite. A Model Predictive Controller is described in Section III.C, and simulation results are given in Section IV. Section IV ends the paper with a discussion of expected results.

II. Preliminaries

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 1 shows the case of a single agent spacecraft relative to the chief satellite. See [26] 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

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

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.*

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,

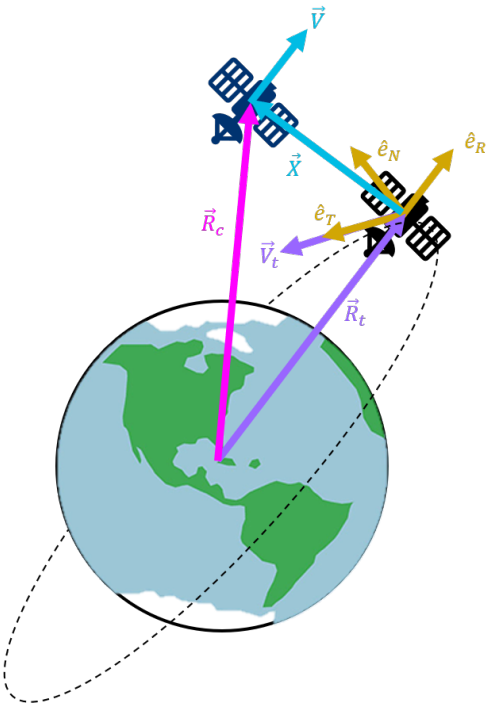

Fig. 1 The Clohessy-Wiltshire (CW) frame.

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.

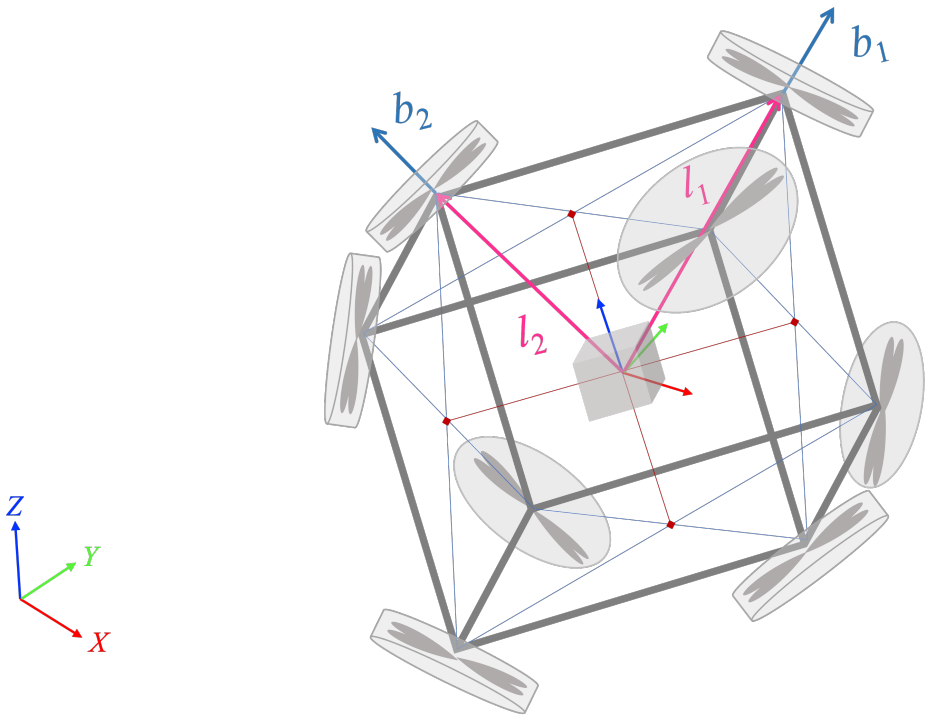

Fig. 2 An omnidirectional multi-rotor vehicle.

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

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

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

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

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 $D_y \in \mathbb{R}^{3\times3}$ and $D_\omega \in \mathbb{R}^{3\times3}$ contain the damping coefficients. The rotation matrix $R(q) \in \mathbb{R}^{3\times3}$ 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

$$T_x = \begin{bmatrix} b_1, b_2, ..., b_n \end{bmatrix} \tag{5a}$$

$$T_\omega = \begin{bmatrix} b_1 \times l_1, b_2 \times l_2, ..., b_n \times l_n \end{bmatrix} \tag{5b}$$

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.

C. Propeller Orientation

To utilize UAVs to verify, validate, and test multi-satellite algorithms in a terrestrial laboratory the UAV must be omnidirectional in order to simulate the desired satellite behaviors. There are multiple examples of omnidirectional vehicles in current literature with each design having its pros and cons. For this work we utilize a design first seen in the work of Brescianini [27] utilizing eight bi-directional propellers fixed on the edges of an octahedron. To ensure the

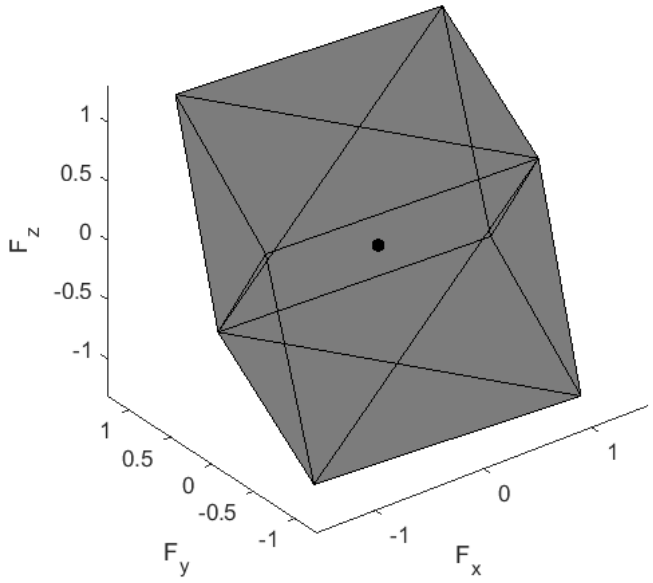

Fig. 3 Visualization of the F1 set for the proposed simulated platform.

system is omnidirectional the propeller thrust and orientations were optimized by minimizing the condition number of the allocation matrix. The allocation matrix is defined to be the matrix that maps propeller inputs into the resultant forces and torques on the body frame as follows:

$$\mathbf{w} = \mathbf{F}u(t) \tag{6}$$

where $\mathbf{w}$ is the applied wrench due to propeller inputs $\mathbf{u}(\mathbf{t})$ and $\mathbf{F}$ is the allocation matrix. The optimization problem can be written as:

$$\min \quad \kappa(\mathbf{F}) \tag{7}$$

$$\text{s.t.} \quad \text{Rank}(\mathbf{F}) = 6 \tag{8}$$

where $\kappa(\mathbf{F})$ is the condition number of $\mathbf{F}$. The constrained nonlinear optimization problem was solve numerically using Matlab's fmincon routine. As discussed in Hamandi [28] a system is omnidirectional if:

$$\text{Rank}(\mathbf{F}) = 6 \text{ and } \begin{bmatrix} 0 \\ 0 \end{bmatrix} \in \ int\{W\} \tag{9}$$

where W is the set of admissible wrenches. Following [28] it is easily seen that a system is omnidirectional if the origin is an interior point of the set of forces, F_1, where F_1 is the set of forces that the system can apply while applying zero-torque. The F_1 set for the proposed system, utilizing unity thrust forces and lengths, is represented in Figure 3.

III. Control Design to Mimic Relative Motion Satellite Dynamics

A. Linear Controller

In this section, we design a linear controller for the omni-directional multi-rotor vehicle. The equations of motion (4) can be rewritten as

$$\dot{q}(t) = \tfrac{1}{2} q(t) \otimes \omega(t) \tag{10a}$$

$$J\dot{\omega}(t) = T_\omega u(t) - \omega(t) \times J\omega(t) \tag{10b}$$

$$m\ddot{y}(t) = R(q(t))T_x u(t) - g \tag{10c}$$

where D_y and D_ω are assumed to be zero for simplicity.

The first step of our control design is to linearizing the dynamics of the system about a nominal attitude trajectory defined by $\bar{q}(t)$ and $\bar{\omega}(t)$. The equations of state for this time-varying linearized system are then

$$\begin{bmatrix} \dot{x}(t) \\ \ddot{x}(t) \end{bmatrix} = \begin{bmatrix} 0_{3\times3} & I_{3\times3} \\ 0_{3\times3} & 0_{3\times3} \end{bmatrix} \begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ R(\bar{q}(t))T_x \end{bmatrix} \begin{bmatrix} u_1(t) \\ \vdots \\ u_n(t) \end{bmatrix} \tag{11a}$$

$$\begin{bmatrix} \dot{\bar{q}}_0(t) \\ \dot{\bar{q}}_v(t) \\ \dot{\bar{\omega}}(t) \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0_{1\times1} & -\bar{\omega}(t)^\intercal & -\bar{q_v}(t)^\intercal \\ \bar{\omega}(t) & -\bar{\omega}^\times(t) & \bar{q_0}(t)I + \bar{q_v}^\times(t) \\ 0_{3\times1} & 0_{3\times3} & -2J^{-1}\bar{\omega}^\times(t)J \end{bmatrix} \begin{bmatrix} \bar{q_0}(t) \\ \bar{q_v}(t) \\ \bar{\omega}(t) \end{bmatrix} + \begin{bmatrix} 0_{1\times n} \\ 0_{3\times n} \\ T_\omega \end{bmatrix} \begin{bmatrix} u_1(t) \\ \vdots \\ u_n(t) \end{bmatrix} \tag{11b}$$

where q_0 and q_v represent the scalar and vector components of the quaternion $q = (q_0, q_v)$ respectively. $\omega^\times$ and $q_v^\times$ denote the matrix representation of the pseudo cross product.

The system (11) is uncontrollable since quaternions are an over-parameterization of the attitude space. This can be corrected via the following time-varying state-space transformation 12

$$T = \begin{bmatrix} I_{6\times6} & 0_{6\times3} & 0_{6\times3} \\ 0_{1\times6} & q_v^\intercal & 0_{1\times3} \\ 0_{3\times6} & q_0 I + q_v^\times & 0_{3\times3} \\ 0_{3\times6} & 0_{3\times3} & I_{3\times3} \end{bmatrix} \in \mathbb{R}^{13\times12} \tag{12}$$

which reduces the dimension of the state from 13 to 12. Applying this transformation produces the controllable dynamics

$$\frac{d}{dt} \begin{bmatrix} x'(t) \\ \dot{x}'(t) \\ q_v'(t) \\ \omega'(t) \end{bmatrix} = \underbrace{\begin{bmatrix} 0_{3\times3} & I_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & -\frac{1}{2}\bar{\omega}^\times & \frac{1}{2}I_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \bar{\omega}^\times - J^{-1}\bar{\omega}^\times J \end{bmatrix}}_{A(t)} \begin{bmatrix} x'(t) \\ \dot{x}'(t) \\ q_v'(t) \\ \omega'(t) \end{bmatrix} + \underbrace{\begin{bmatrix} 0_{3\times n} \\ R(t)T_x \\ 0_{3\times n} \\ T_\omega \end{bmatrix}}_{B(t)} \begin{bmatrix} u_1(t) \\ \vdots \\ u_n(t) \end{bmatrix} \tag{13}$$

Now that the system has been linearized about a nominal trajectory an LQR controller and feed-forward controller can be implemented to produce inputs for trajectory tracking according to the following

$$u(t) = u_\infty(t) - F(t)T(t)\big(x(t) - x_\infty(t)\big) \tag{14}$$

where $F(t) \in \mathbb{R}^{n\times12}$ is the LQR gain matrix for the matrices $A(t)$ and $B(t)$ in (13). The transformation $T(t)$ (12) maps the full-state $(x, \dot{x}, q, \omega)$ of the drone (4) to the reduced state $(x', \dot{x}', q_v', \omega')$ of the controllable dynamics (13). The state and input pair $x_\infty \in \mathbb{R}^1 2$ and $u_\infty \in \mathbb{R}^n$ satisfy

$$\begin{bmatrix} A(t) - I_{13\times13} & B(t) \\ C & D \end{bmatrix} \begin{bmatrix} x_\infty(t) \\ u_\infty(t) \end{bmatrix} = \begin{bmatrix} -B_d d(t) \\ r(t) \end{bmatrix} \tag{15}$$

where $d(t)$ are exogeneous disturbances acting on the multi-rotor and the matrix B_d determines their effect on the acceleration of the drone, and $r(t)$ is the reference trajectory.

At every time-step t, new LQR and feedforward controller gains are computed based on the dynamics (13) and linearized about the current state and updated reference trajectory to create the optimal series of control inputs for trajectory tracking.

B. Extended Kalman Filtering

In this section, we present an extended Kalman filter used to estimate the state of the non-linear multi-rotor dynamics. The extended Kalman filter consists of two steps: prediction and update. Prediction is given by

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, u_k) \tag{16a}$$

$$P_{k|k-1} = A_k P_{k-1|k-1} A_k^T + Q_k \tag{16b}$$

where $\hat{x}_{k|k-1}$ and $P_{k|k-1}$ are respectively the mean and covariance of the estimated state at time-instance k based previous measurements up to time-instance $k-1$, and Q_k is the covariance of the process noise. The prediction step (16a) uses the nonlinear multi-rotor dynamics (4) to propagate the mean of the estimated state and the covariance is propagated using the partial derivative $A_k = \frac{\delta f}{\delta x}|_{\hat{x}_{k-1|k-1}, u_k}$ of these dynamics with respect to the state evaluated of the currently mean estimated state $\hat{x}_{k|k-1}$ and control input u_k. The multi-rotor dynamics are augmented with integral dynamics to estimate the disturbances $d(t)$ acting on the multi-rotor. This provides integral-action that produces offset-free tracking [29].

When a new measurement y_k is available, the extended Kalman filter updates the state estimate and covariance estimates as follows

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + P_{k|k-1} C^T (R_k + C P_{k|k-1} C^T)^{-1} (y_k - C\hat{x}_{k|k-1}) \tag{17a}$$

$$P_{k|k} = P_{k|k-1} - P_{k|k-1} C^T (R_k + C P_{k|k-1} C^T)^{-1} C P_{k|k-1} \tag{17b}$$

where R_k is the covariance of the measurement noise. We assume that the position $x \in \mathbb{R}^3$ and orientation $q \in \mathbb{H}$ of the multi-rotor are measured. This simplifies the extended Kalman filter since $C \in \mathbb{R}^7$ is a constant matrix.

C. Model Predictive Controller

In this section, we extend the LQR controller into a nonlinear MPC that enforces constraints and improves performance by optimizing the control input using the linearized dynamics (11) over a prediction horizon N. The model predictive controller (MPC) obtains the control input $u(t)$ by solving the following constrained finite-time optimal control problem (CFTOC)

$$\min \quad \sum_{k=0}^{N-1} \left\| C(x_{k+1|t-1} + \delta x_{k|t}) - r_k \right\|_Q^2 + \left\| u_{k+1|t-1} + \delta u_{k|t} \right\|_R^2 \tag{18a}$$

$$\text{s.t.} \quad \delta x_{k+1|t} = A_k \delta x_{k|t} + B_k \delta u_{k|t} + B_{d,k} \delta d_k \tag{18b}$$

$$u_{k+1|t-1} + \delta u_{k|t} \in \mathcal{U},\ x_{k+1|t-1} + \delta x_{k|t} \in \mathcal{X} \tag{18c}$$

$$x_{k+1|t-1} + \delta x_{0|t} = Tx(t) \tag{18d}$$

where the incremental input $\delta u_{k|t} = u_{k|t} - u_{k+1|t-1}$ is the difference between the current planned input $u_{k|t}$ at time t and previously planned input $u_{k+1|t-1}$ at time $t-1$. Likewise, the incremental state $\delta x_{k|t} = x_{k|t} - x_{k+1|t-1}$ is the difference between the current $x_{k|t}$ and previous $x_{k+1|t-1}$ state predictions. The prediction model (18b) is the dynamics (11) linearized about the previous input $(u_{1|t-1}, \ldots, u_{N-1|t-1}, Fx_{N|t-1})$ and state $(x_{1|t-1}, \ldots, x_{N|t-1}, (A-BF)x_{N|t-1}$ trajectories shifted in time $t-1 \to t$ with the LQR input $Fx_{N|t-1}$ and state $(A-BF)x_{N|t-1}$ appended to the end of the predicted trajectory. After solving (18), the input trajectory at time t updated $u_{k|t} = u_{k+1|t-1} + \delta u_{k|t}$ by integrating the previous input $u_{k+1|t-1}$ and the optimal change $\delta u_{k|t}$. Likewise, the state trajectory is updated $x_{k|t} = x_{k+1|t-1} + \delta x_{k|t}$. The input and state constraints (18c) are enforced on the integrated inputs $u_{k+1|t-1} + \delta u_{k|t}$ and states $x_{k+1|t-1} + \delta x_{k|t}$, respectively. The MPC implements the control input

$$u(t) = u_{0|t} = u_{1|t-1} + \delta u_{0|t}. \tag{19}$$

Note that the CFTOC can be interpreted as a sequential quadratic program (SQP) where only one linearization is performed at each time instance t [30]. The initial condition (18d) for the CFTOC (18) is the state of the drone $x(t)$ project onto the transformation $T \in \mathbb{R}^{n-1 \times n}$ which eliminate the uncontrollable subspace of the linearized dynamics (18b).

The CFTOC problem (18) was formulated to provide reference tracking and disturbance rejection. The incremental formulation of the CFTOC provides integral-action that improves reference tracking and disturbance rejection [31, 32]. The disturbances δd_k include exogenous forces such as gravity. This disturbance is estimated using the extended Kalman filter.

IV. Numerical Simulation Results

In this section, we demonstrate our model, estimator, and controller through simulations. For the simulation, we consider a multi-rotor with $n = 8$ rotors as shown in Fig. 2. The thrust $T_x \in \mathbb{R}^{3\times 8}$ and torques $T_\omega \in \mathbb{R}^{3\times 8}$ matrix in (4) are given as

$$T_x = \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 1 & 0 & -1 & 1 & -1 & 0 \end{bmatrix} \tag{20a}$$

$$T_\omega = \begin{bmatrix} -0.125 & 0.125 & 0.125 & 0.000 & 0.125 & 0.125 & -0.125 & 0.125 \\ 0.125 & 0.000 & -0.125 & 0.125 & -0.125 & 0.250 & 0.125 & 0.000 \\ 0.000 & -0.125 & 0.000 & -0.125 & 0.000 & -0.125 & 0.000 & -0.125 \end{bmatrix} \tag{20b}$$

The nonlinear dynamics (4) of the multi-rotor are to be simulated in MATLAB using the ODE45 solver. We will consider several simulation scenarios to test the model, estimator, and controller. These will include the multi-rotor moving in a straight line while tumbling Fig. 4, having the multi-rotor orbit in a circle while maintaining a constant orientation Fig. 8, having the multi-rotor tumble while maintaining is position, and a complex trajectory with sinusoidal oscillation in the z direction while the multi-rotor orbits in a circle and tumbles, Fig. 6.

In all cases we anticipate the multi-rotor will converge to the desired trajectory within a finite amount of time, indicating that the controller is performing as expected.

V. Expected Results

In this paper we presented a promising control methodology to emulate the equations of motion of a satellite. In the full paper to be presented and published in the proceedings, we will focus on a high-performance learning-based Model Predictive Control (L-MPC) for an omnidirectional multi-rotor UAV. We will compare the performance of the proposed controller with conventional MPC and LQR algorithms. In addition, we plan to implement the proposed controller on hardware using a modular testbed being developed at the University of New Mexico. Simulation and experimental results will validate the proposed learning based control methodology. Simulations of the eight propeller system are done via MATLAB given varying reference trajectories to track. As traditionally designed drones (i.e. four propellers with parallel thrust vectors) are known to be able to tack reference trajectories in $\mathbb{R}^3$ well, the important observations for these tests are in the crafts ability to make use of all six degrees of freedom, without hindering its ability to track the reference.

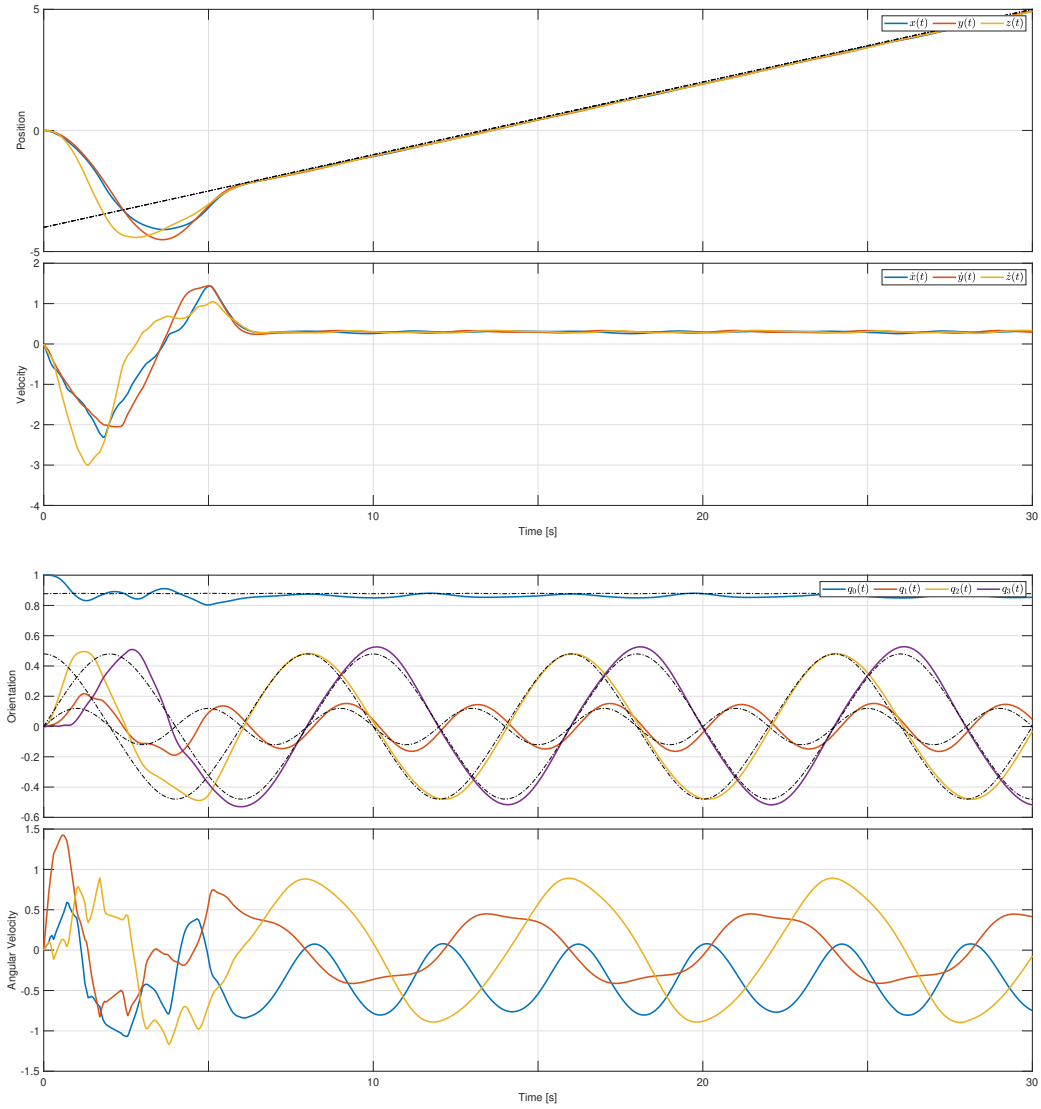

Fig. 4 Linear trajectory with tumble

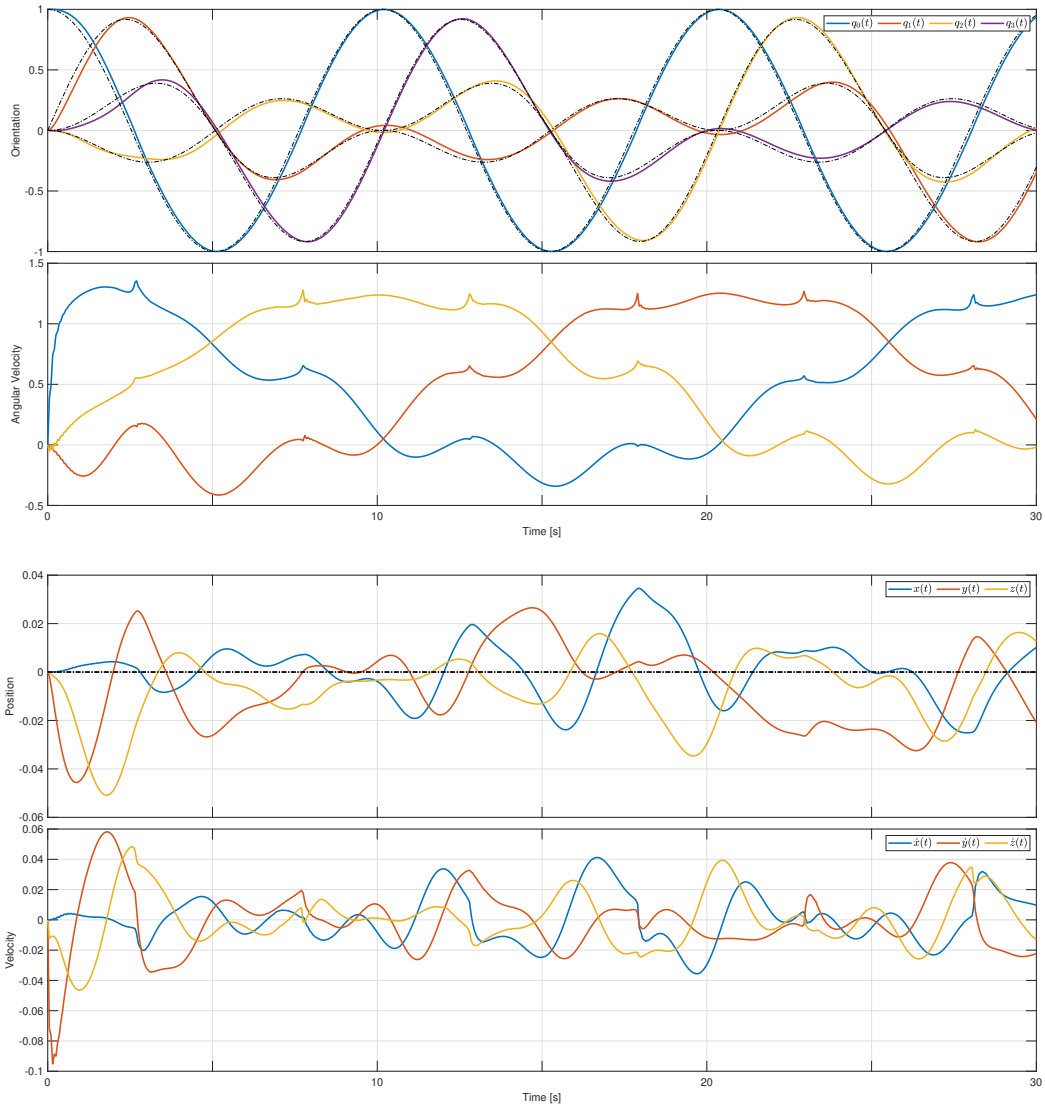

Fig. 5 Hold position and tumble

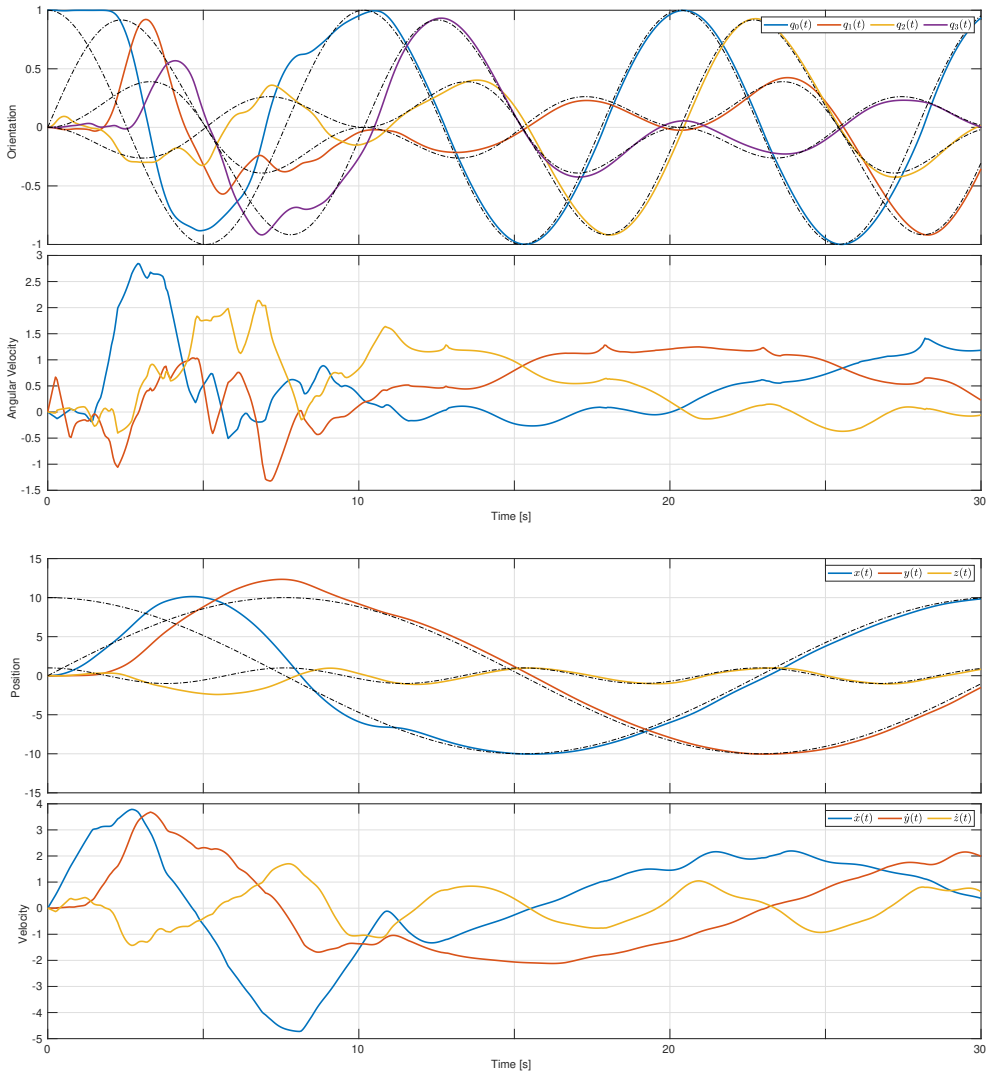

Fig. 6 Complex Tumble Orientation and Position vs Reference Trajectory

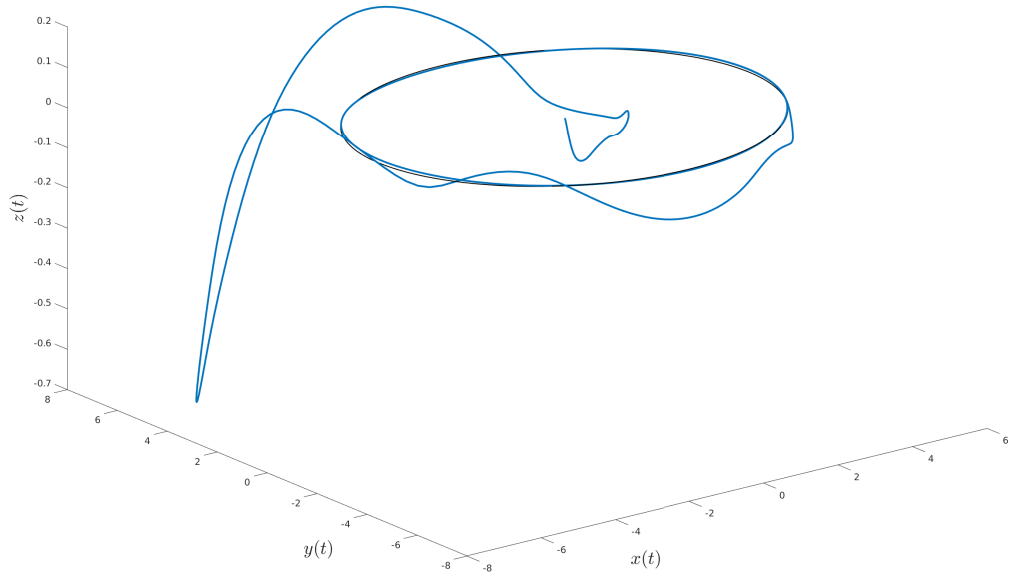

Fig. 7 Circle with constant orientation.

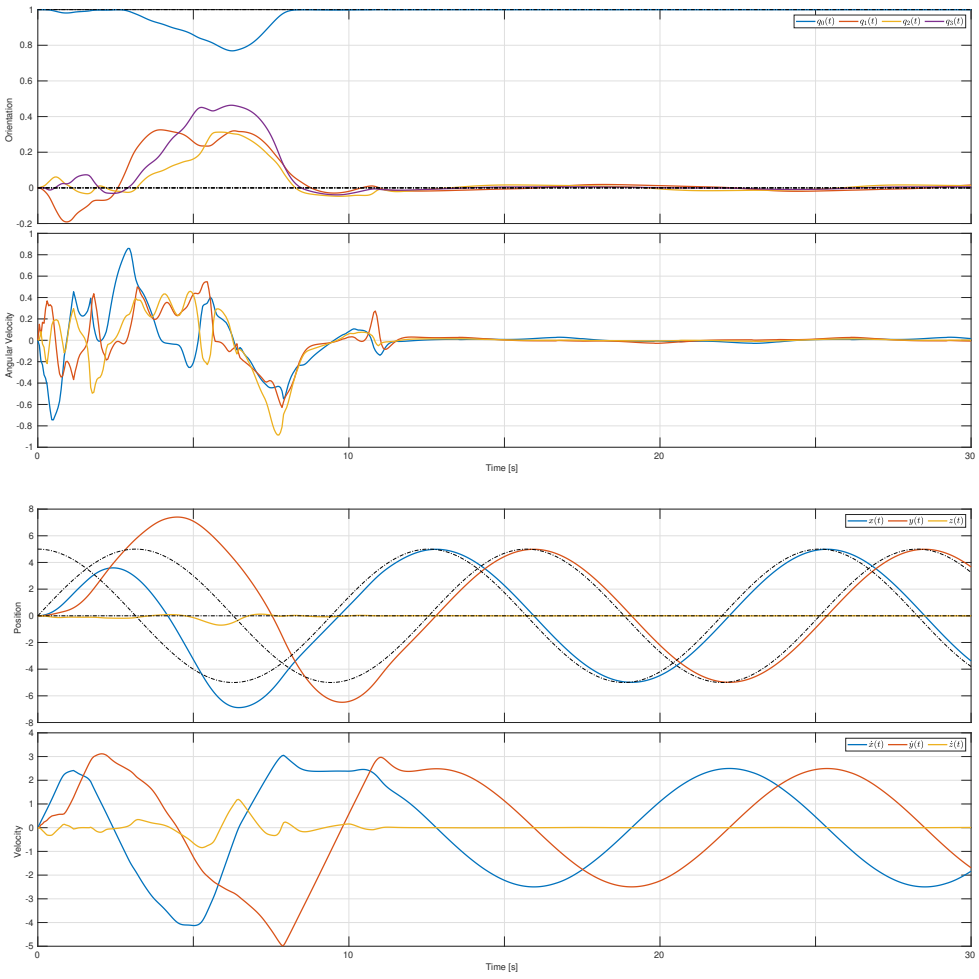

Fig. 8 Circle with constant orientation.

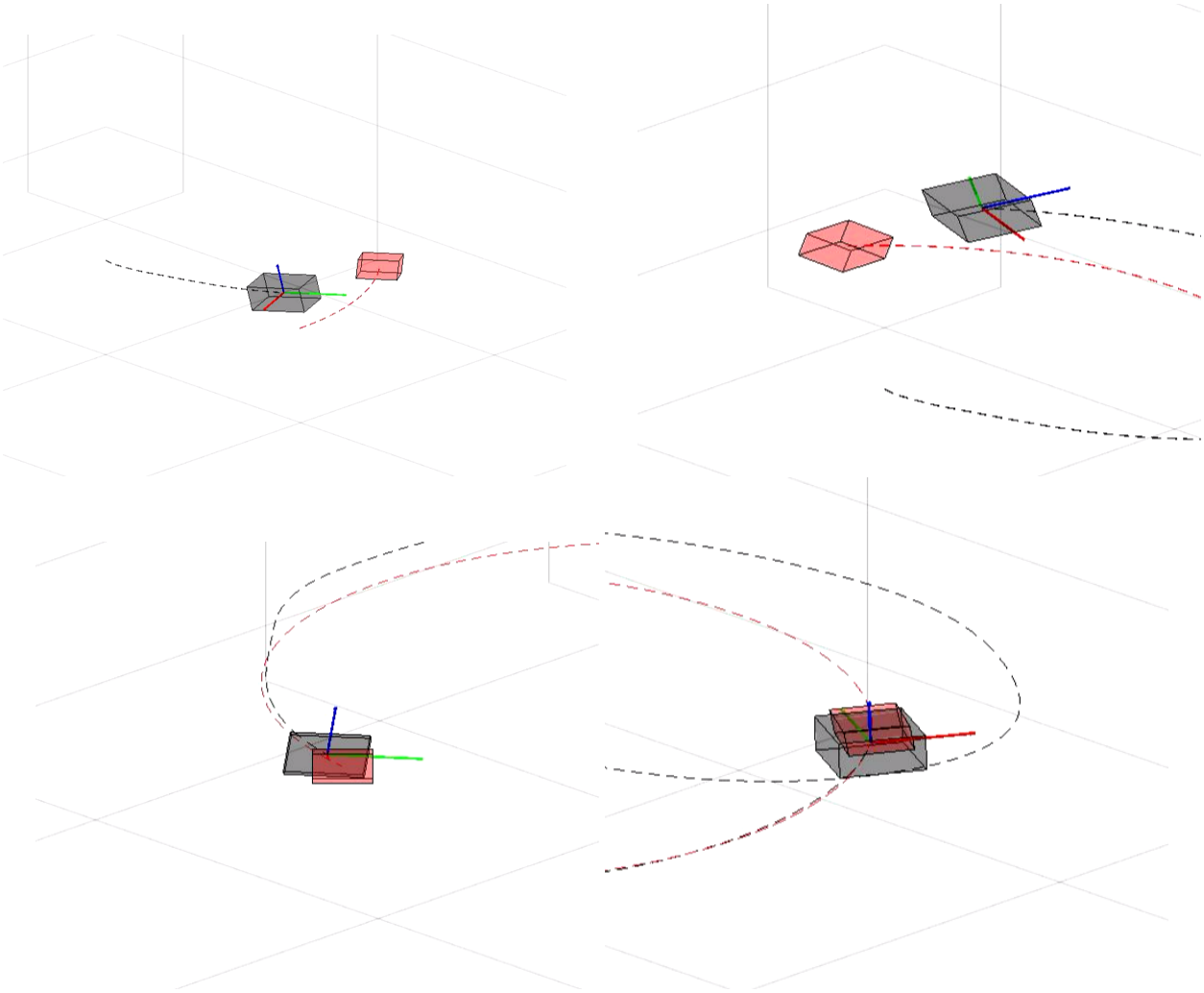

Fig. 9 Omnidirectional multi-rotor.

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., Tsiotras, 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., Tsiotras, P., and Phillips, S., *Formal Verification Applied to Spacecraft Attitude Control*, ????

[5] Cellucci, D., Cramer, N. B., and Frank, J. D., *Distributed Spacecraft Autonomy*, ????

[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 Tsiotras, 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=-gMeOAAACAAJ`.

[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] 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.

[18] 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.

[19] 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.

[20] Dmytruk, A., Silano, G., Bicego, D., Licea, D. B., and Saska, M., "A Perception-Aware NMPC for Vision-Based Target Tracking and Collision Avoidance with a Multi-Rotor UAV," *International Conference on Unmanned Aircraft Systems*, 2022.

[21] Song, Y., and Scaramuzza, D., "Policy Search for Model Predictive Control With Application to Agile Drone Flight," *IEEE Transactions on Robotics*, 2022.

[22] Torrente, G., Kaufmann, E., Föhn, P., and Scaramuzza, D., "Data-driven MPC for quadrotors," *IEEE Robotics and Automation Letters*, Vol. 6, No. 2, 2021, pp. 3769–3776.

[23] Tsukamoto, H., Chung, S.-J., and Slotine, J.-J. E., "Contraction theory for nonlinear stability analysis and learning-based control: A tutorial overview," *Annual Reviews in Control*, Vol. 52, 2021, pp. 135–169.

[24] Saviolo, A., Li, G., and Loianno, G., "Physics-Inspired Temporal Learning of Quadrotor Dynamics for Accurate Model Predictive Trajectory Tracking," *arXiv preprint arXiv:2206.03305*, 2022.

[25] Salzmann, T., Kaufmann, E., Pavone, M., Scaramuzza, D., and Ryll, M., "Neural-MPC: Deep Learning Model Predictive Control for Quadrotors and Agile Robotic Platforms," *arXiv preprint arXiv:2203.07747*, 2022.

[26] de Ruiter, A., Damaren, C., and Forbes, J., *Spacecraft Dynamics and Control: An Introduction*, Wiley, 2012.

[27] 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.

[28] 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. doi:10.1177/02783649211025998, URL https://doi.org/10.1177/02783649211025998.

[29] Borrelli, F., and Morari, M., "Offset free model predictive control," *2007 46th IEEE Conference on Decision and Control*, 2007, pp. 1245–1250. doi:10.1109/CDC.2007.4434770.

[30] Diehl, M., *Optimization Algorithms for Model Predictive Control*, Springer International Publishing, Cham, 2021, pp. 1619–1626. doi:10.1007/978-3-030-44184-5_9, URL https://doi.org/10.1007/978-3-030-44184-5_9.

[31] Pannocchia, G., "Offset-free tracking MPC: A tutorial review and comparison of different formulations," *2015 European Control Conference (ECC)*, 2015, pp. 527–532. doi:10.1109/ECC.2015.7330597.

[32] Betti, G., Farina, M., and Scattolini, R., "A Robust MPC Algorithm for Offset-Free Tracking of Constant Reference Signals," *IEEE Transactions on Automatic Control*, Vol. 58, No. 9, 2013, pp. 2394–2400. doi:10.1109/TAC.2013.2254011.