

MOMAV: A highly symmetrical fully-actuated multirotor drone using SQP based control allocation

Marco Ruggia - Institute of Photonics and Robotics, University of Applied Sciences of the Grison, Switzerland

Abstract—MOMAV (Marco’s Omnidirectional Micro Aerial Vehicle) is a multirotor drone that is fully actuated, meaning it can control its orientation independently of its position. MOMAV is also highly symmetrical, making its flight efficiency largely unaffected by its current orientation. These characteristics are achieved by a novel drone design where six rotor arms align with the vertices of an octahedron, and where each arm can actively rotate along its long axis. Various standout features of MOMAV are presented: The high flight efficiency compared to arm configuration of other fully-actuated drones, the design of the rotating arm assembly including custom slip-rings used to enable continuous arm rotation, and a novel control allocation algorithm based on sequential quadratic programming (SQP) used to calculate throttle and arm-angle setpoints in flight. Flight tests have shown that MOMAV is able to achieve remarkably low mean position/orientation errors of 6.4mm, 2.4° (σ : 2.9mm, 1.1°) when sweeping position setpoints, and 12.0mm, 3.2° (σ : 8.9mm, 2.0°) when sweeping orientation setpoints.

I. INTRODUCTION

Fully-actuated aerial vehicles are able to exert forces and torques independently of each other, enabling them to decouple translation and rotation movements. Unlike an ordinary multicopter, they can for example accelerate forward without having to lean forward, or lean forward without having to accelerate. This characteristic has been found to be particularly useful in manipulation tasks where aerial vehicles need to be in contact with the environment [1, 2]. Such drones could possibly also become useful as an alternative to gimbals for sensors in visual inspections and surveying applications.

Most fully-actuated aerial vehicles are multicopters that fall in one of two categories. Fixed-tilt designs, where the rotor disks are fixedly tilted away from the horizontal plane [3–14], and variable-tilt designs where the rotor disks can be actively tilted [15–28]. In both cases at least six actuators are needed (either propeller motors or tilt servos) to control the six degrees of freedom of the drone (three for position and three for orientation).

The here presented variable-tilt drone named MOMAV (Fig. 1) features six propellers and six arm-angle servos, totaling twelve actuators. This exceeds the minimum of six, making the drone not only fully-actuated, but over-actuated. Thus, the spare actuators can be used to optimize additional objectives. Concretely, the additional objective of MOMAV is to increase flight efficiency independently of its current orientation. This is achieved through the choice of highly symmetric arm directions, and through a sequential quadratic programming (SQP) based control allocation algorithm.

Additionally, a simple low-cost slip-ring assembly is presented on MOMAV, that connects the propeller motors in a



Fig. 1: Prototype drone *MOMAV*

way that enables continuous arm rotation. It serves to simplify the constraints placed on the control allocation. Not doing so would require a more sophisticated solution to prevent the cables powering the propeller motors from winding up around the arms over time.

II. ARM CONFIGURATION

Many arm configurations for fully-actuated drones featuring ingenious mechanical solutions have been proposed. The vast majority of them have strict limitations on the orientations they can achieve [4–6, 8–10, 13, 15–19, 23, 24]. Drones capable of operating across the entire range of orientations have been shown to work only for two types of designs: Variable-tilt planar multicopter configurations where each arm can actively rotate along its long axis [20, 21], and fixed-tilt multicopters with three-dimensionally positioned propellers [3, 7, 11]. This project combines the concepts by implementing a hexacopter with arms aligned to the vertices of an octahedron (3D), where each arm can rotate. Such a configuration will be shown to have a consistently high efficiency across all orientations.

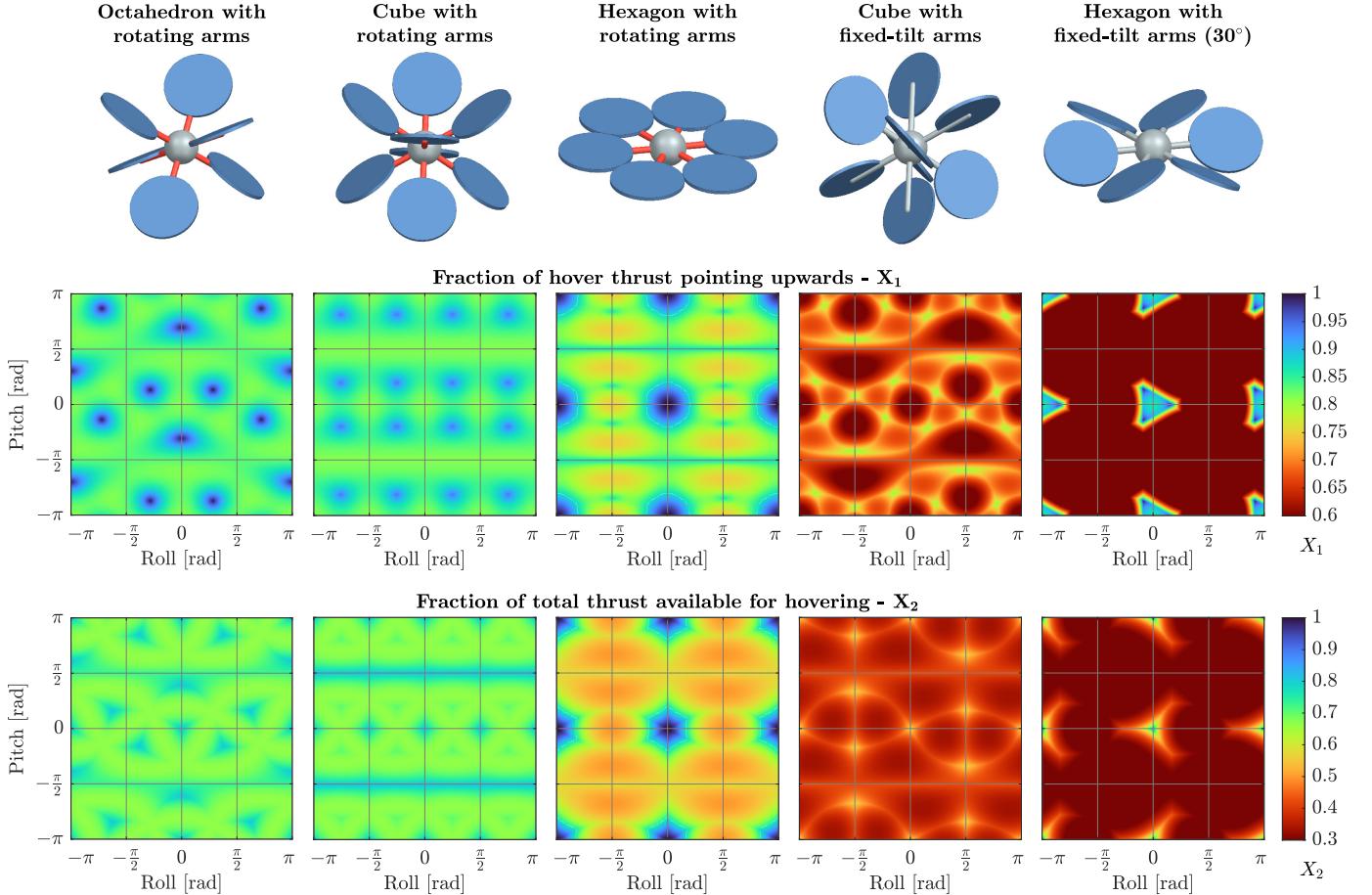


Fig. 2: Flight efficiency X_1 , X_2 depending on orientation for a selection of arm configurations

Two metrics are calculated to determine flight efficiency. X_1 is the fraction of thrust directed upwards during hovering flight. The remainder ($1 - X_1$) is thrust, that is wasted by motors having to push against each other instead of upwards. X_2 on the other hand is the fraction of total thrust (sum of all maximal motor thrusts) that is available for hovering. For example, to be able to hover with $X_2 = 0.5$, the total thrust of the drone needs to be at least high enough to lift twice its weight. Both X_1 and X_2 are calculated by minimizing the sum of squared throttles for arm configurations that exceed the six degrees of freedom needed to impose the hover condition.

Analyzing X_1 and X_2 across orientations (Fig. 2) shows that the proposed octahedron with rotating arms achieves $X_1 \in [0.82, 1.00]$, $X_2 \in [0.67, 0.82]$, while a hexagon with rotating arms (like [20, 21]) reaches worse minima at $X_1 \in [0.75, 1.00]$, $X_2 \in [0.50, 1.00]$, and a cube with fixed-tilt arms (like [11]) reaches only $X_1 \in [0.58, 0.81]$, $X_2 \in [0.33, 0.58]$. As expected, designs such as hexagons with fixed-tilt arms (like [6, 13]) are found to have very poor efficiencies outside the vertical orientation. Cubes and tetrahedra with rotating arms were also analyzed and considered for a prototype, but were subsequently disregarded due to manufacturability concerns, and because they showed no significant improvement in either X_1 nor X_2 compared to an octahedron.

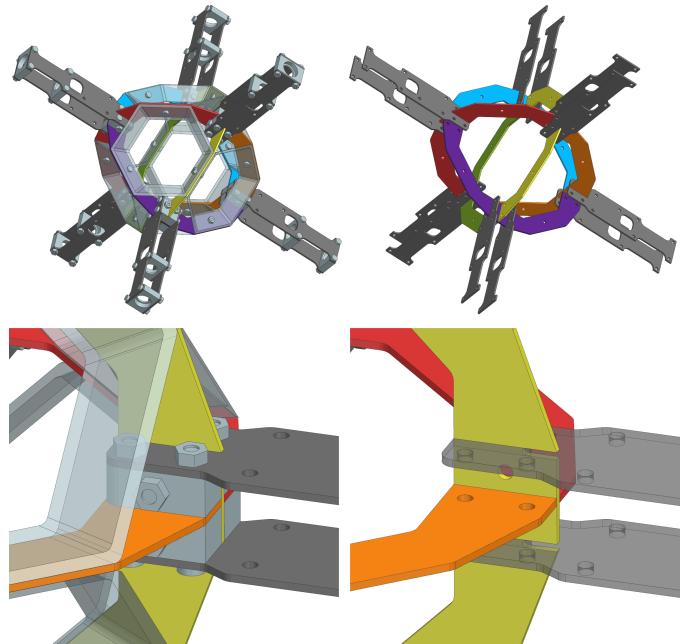


Fig. 3: Views of the octahedral body implementation showing details of the joints that connect the arms to the body

For the chosen octahedron configuration, X_1 is minimal when two opposing faces of the octahedron align with the vertical axis. In that case $(1 - X_{1,\min}) = 18\%$ of thrust is wasted to propellers having to push against each other. Conversely, X_2 is minimal when two opposing vertices align with the vertical axis. In that case only 4 of 6 motors can be used for hovering, so at least $(1 - X_{2,\min}) = 2/6 = 33\%$ of total thrust must be kept as surplus in the design.

In practice, the octahedral body (Fig. 3) is then constructed using carbon fiber sheets interlocking with halved joints. Additional rigidity is gained by connecting the edges of each face with hexagons, creating a truncated octahedron shape. Relying mainly on interlocking joints for connections has resulted in a very rigid construction with no noticeable play.

III. ROTATING ARM ASSEMBLY

A second novel contribution of MOMAV are its arm assemblies (Fig. 4). Their function is to enable control of the arm angles allowing for an arbitrary number of revolutions, and to transmit power to the propeller motors while doing so. These two functions can be challenging to implement. Here, solutions to both are proposed, analyzed, and then implemented on the MOMAV drone.

A. Arm Actuator

While some commercial solutions exist that are potentially suitable as arm actuators, none are found that fulfill all requirements on size, torque, velocity, accuracy, and rotation range demanded by this project. Instead an existing servo motor, the *KST MS325*, is modified to fit the requirements.

The only shortcoming of the *KST MS325* servo is its 200° rotation angle limit. Otherwise it features an outstanding 0.52Nm stall torque and 2.4rev/sec no-load speed for its 23x12x28mm size. This servo motor is also particular in that it

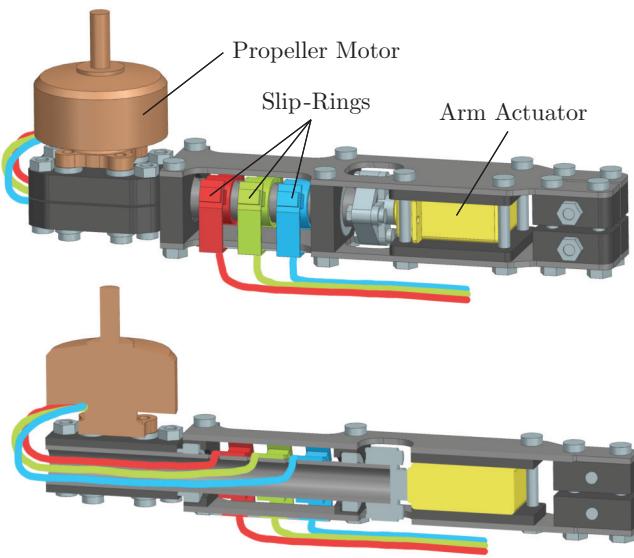


Fig. 4: Rotating arm assembly with a section view showing internal wiring of the three motor phases

features an *AMS AS5600* magnetic encoder to measure angular position instead of the much more prevalent potentiometers. This encoder makes continuous rotation mechanically possible and only prevented by the built-in driver. So, a modification is performed by swapping the built-in driver for an *Atmel SAM D21* microcontroller and a *TI DRV8870* H-bridge. A PID control loop featuring *proportional on measurement* (PoM) [29] is implemented on the microcontroller with the goal of avoiding the overshoot typically present on integrating systems like the DC motor used inside the servo.

To evaluate the arm actuator a series of tests is performed while the propeller motor is running and producing approximately 15N of thrust. A step response test (Fig. 5A) shows that the arm actuator is able to reach a setpoint angle precisely and without overshoot, being mostly only limited by a $\sim 2.4\text{rev/sec}$ velocity saturation.

Under the same conditions, chirp signals of various amplitudes are tested and frequency domain analysis performed by means of bode plots (Fig. 5B). These show a magnitude fall-off with increasing chirp amplitude consistent with the previously noted velocity saturation, and a linear phase fall-off consistent with a time-delay of $\sim 36\text{ms}$.

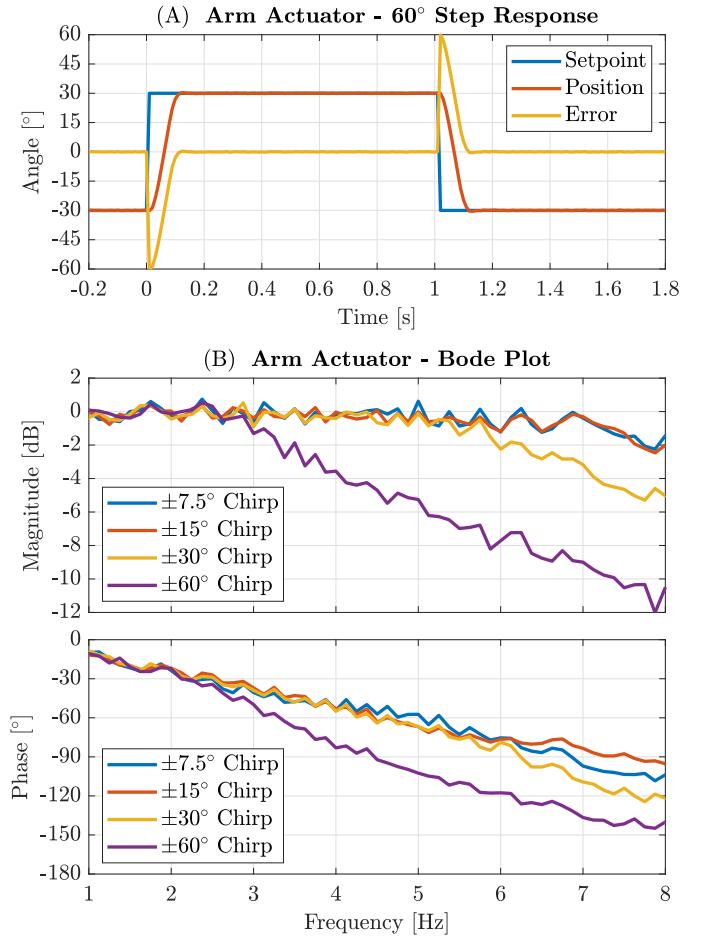


Fig. 5: Arm actuator step responses (A) and frequency response (B) with propeller motor running at $\sim 15\text{N}$ thrust

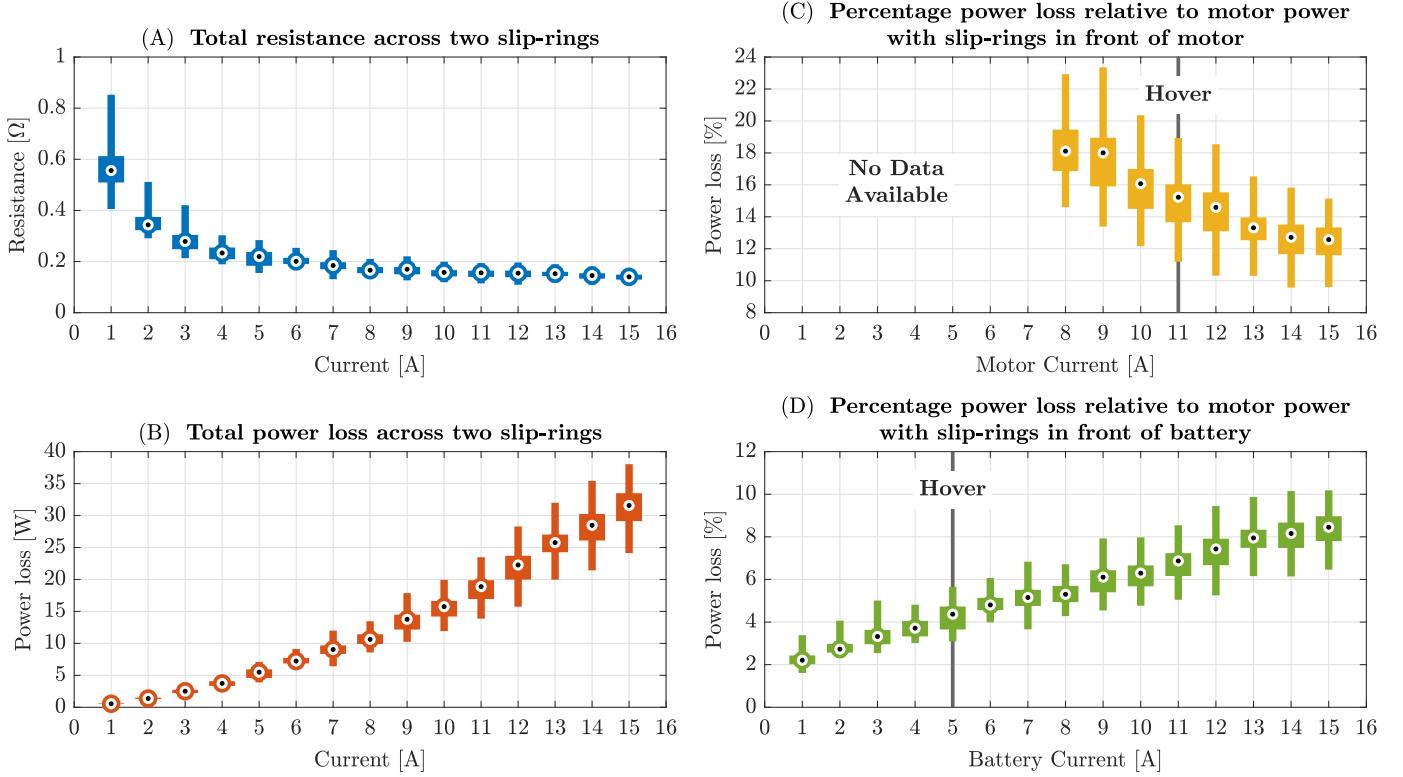


Fig. 6: Resistance (A) and power loss (B) of a pair of two custom slip-rings depending on current draw, as well as power loss as a percentage of motor power for the two possible slip-ring locations (C, D).

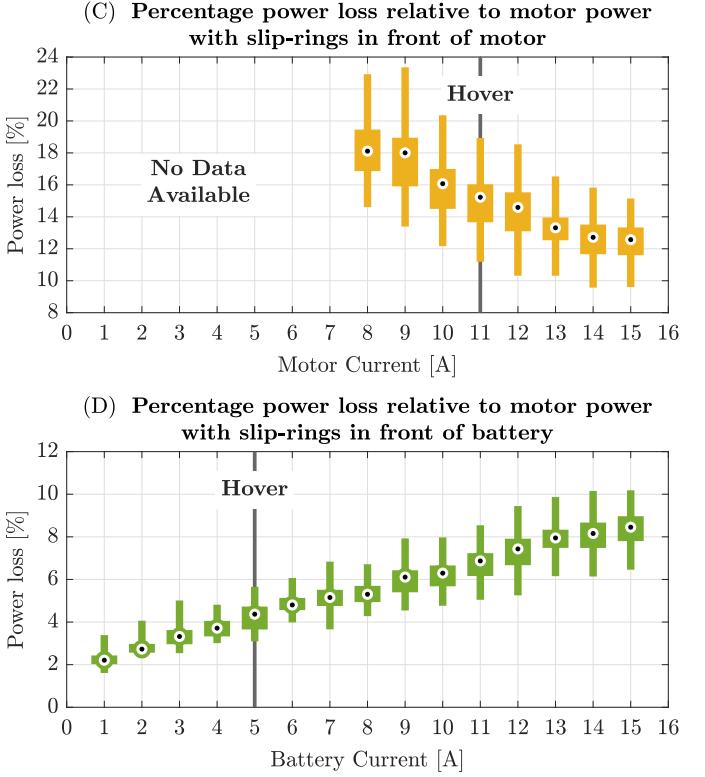
B. Slip-Rings

The second challenge solved by the rotating arm assembly, is to transmit power from the stationary part of the arm to the rotating propeller motor. Slip-rings are proposed as a novel way to avoid cables winding up around the arms over time. Like for the arm actuators, it is exceedingly difficult to buy a slip-ring of suitable size, cost, and current rating for this task. Instead a solution is proposed, that uses brushes harvested from the ubiquitous *RS550* DC motors.

A pair of two brushes is used for each of the three phases of the motor. They push against three copper rings placed on the rotating part of the arm. Wires then connect from the inside of these rings to the propeller motor (Fig. 4).

The current rating for *RS550* motors varies somewhat by manufacturer, but generally lies around 5-8A. The custom slip-rings are tested up to a current of 15A, while being cooled by a small fan. This is similar to the cooling provided by the propeller in the finished drone and the centrifugal fan in the original motor. Overheating is found not to be a problem. Nevertheless, a further in-depth test is performed to characterize the power losses. In particular it is tested how these losses vary between slip-rings and depending on arm-angles. Having high variance in losses can be problematic, as it leads to variance in produced thrust, which in turn makes flight-control more challenging.

The slip-ring losses are measured by short-circuiting all three phases where the propeller motor should be, applying



a regulated current running through two slip-rings, and measuring the voltage needed to push said current. This voltage is then correlated to slip-ring resistance and power loss through Ohm's and Joule's law. Although, the calculated power loss is probably an overestimation, as during motor commutation not all current would be flowing through only two of the three slip-rings at any given time. This test is performed for all three combinations of slip-ring pairs of one arm. Multiple measurements are taken at each arm-angle, by letting the arm actuator slowly spin while measuring. The results are presented as box plots (Fig. 6A, 6B).

The slip-ring losses are furthermore plotted as fractions of the propeller motor power for the used *T-Motor F100 KV1100* motors (Fig. 6C). To do that, the motor power vs. current data is taken from the manufacturer data-sheet, where motor current is inferred from battery current by dividing it by duty cycle (=throttle). At 11A motor current, where the drone would hover, the relative losses are substantial, ranging from 11.2% to 18.9% with a median of 15.2%.

It should be noted that placing the slip-rings at each motor phase was a major design oversight. A better approach would be to place the slip-rings between the battery and the motor driver (Fig. 6D). In such a configuration the current passing through the slip-rings at hover would only be 5A, corresponding to a relative power loss of 3.1% to 5.6% with a median of 4.4%. It would however require the motor drivers to be located on the rotating part of the arms.

IV. CONTROL ALLOCATION

When in flight, the position and orientation control of MOMAV is handled by a very basic, manually tuned PID controller. It takes as input the current position/orientation errors of the drone and outputs desired force/torque setpoints that act to reduce said errors. The way in which these setpoints are then allocated to propeller-throttle and arm-angle setpoints, however is novel. This algorithm for control allocation is presented below.

MOMAV is highly over-actuated, featuring 12 actuators (6 for arm-angles and 6 for propeller-throttles), but only 6 constraints (3 for the force setpoint and 3 for the torque setpoint). It means that the constraints stemming from the desired force (F) and torque (M) are not sufficient to fully define the arm-angles (a) and propeller-throttles (u). For this reason an optimization approach is used (Eq. 1) where the force/torque constraints must be satisfied ($G=0$) and the remaining variables are chosen such that they minimize some objective function (O).

$$\begin{aligned} (u^*, a^*) = \operatorname{argmin} O(u, a) \\ \text{subject to } G(u, a, q, F, M) = 0 \end{aligned} \quad (1)$$

Name	Description	Type
Inputs:		
q	Body orientation quaternion	$\in \mathbb{H}$
F	Desired body force	$\in \mathbb{R}^3, [N]$
M	Desired body torque	$\in \mathbb{R}^3, [Nm]$
Outputs:		
a_i	Angle of arm i	$\in \mathbb{R}, [rad]$
u_i	Motor throttle of arm i	$\in \mathbb{R}, (0, 1)$
Constants:		
r_i	Arm endpoint	$\in \mathbb{R}^3, [m]$
x_i	Arm rotation axis	$\in \mathbb{R}^3, \ \cdot\ = 1$
z_i	Motor thrust direction at zero arm angle	$\in \mathbb{R}^3, \ \cdot\ = 1$
s_i	Motor spin direction	$\in \{-1, 1\}$
μ	Motor thrust constant	$\in \mathbb{R}, [N/1]$
τ	Motor torque constant	$\in \mathbb{R}, [Nm/1]$
Δt	Control loop period	$\in \mathbb{R}, [s]$
Intermediaries:		
n_i	Motor thrust direction	$\in \mathbb{R}^3, \ \cdot\ = 1$
f_i	Motor force	$\in \mathbb{R}^3, [N]$
m_i	Motor torque	$\in \mathbb{R}^3, [Nm]$

TABLE I: Variable names and descriptions

The objective function O is chosen with the intention of minimizing power consumption, while keeping the solution (u^*, a^*) feasible. It is split up in four goals, each one weighted against the others:

- Minimize the squared sum of throttles
- Minimize the squared rotation velocity of the arms

- Constrain the throttles within 0-100%
- Constrain the arm rotation velocities within $\pm 2\text{rev/sec}$

The last two goals are inequality constraints, but since those can be cumbersome to deal with, they are instead implemented as quadratic penalty terms in the objective O . The resulting objective function O (Eq. 2) thus becomes a continuous piecewise polynomial function, summing over penalty functions for throttles $p_u(u)$ and arm velocities $p_{\dot{a}}(\dot{a})$ (Fig. 7).

$$O = \sum_{i=1}^6 p_u(u_i) + \sum_{i=1}^6 p_{\dot{a}} \left(\frac{a_i - a_{i,\text{prev}}}{\Delta t} \right) \quad (2)$$

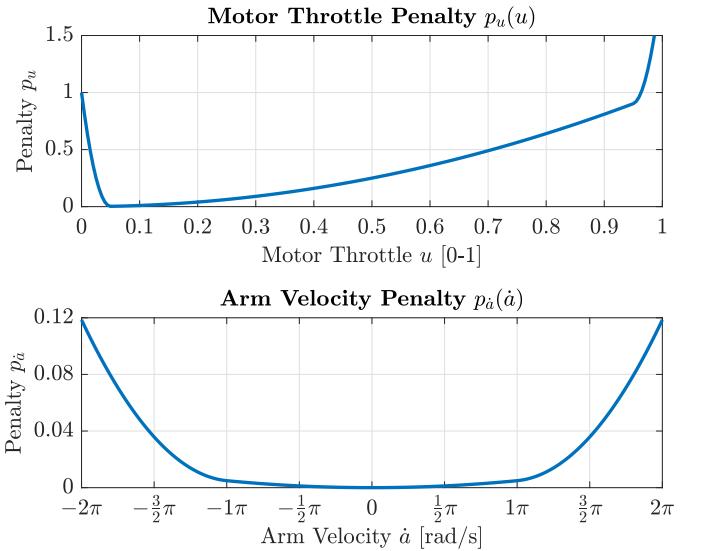


Fig. 7: Penalty functions used in optimization objective O

The constraint function G is defined as the difference between the desired body force/torque F, M and the sum of forces/torques f_i, m_i achieved by the inputs u_i, a_i (Eq. 6). When these are equal to each other, it follows that $G=0$ and the constraint is satisfied. Note that in the following equations quaternions are multiplied with vectors and the result is defined to be the rotated vector.

$$n_i = \begin{bmatrix} \sin(a_i/2) \\ x_i \cos(a_i/2) \end{bmatrix}_{\in \mathbb{H}} z_i \quad (3)$$

$$f_i = \mu u_i n_i \quad (4)$$

$$m_i = \mu u_i (r_i \times n_i) + \tau s_i u_i n_i \quad (5)$$

$$G = \begin{bmatrix} \sum_{i=1}^6 (f_i) - q^{-1} F \\ \sum_{i=1}^6 (m_i) - q^{-1} M \end{bmatrix} \quad (6)$$

Now that the optimization problem is fully defined, a few characteristics can be noted:

- The problem is nonlinear due to the trigonometric functions in the constraint G , suggesting that a nonlinear iterative solver is required.

- The objective O is a convex function, meaning there is no risk of converging to a local maximum instead of a local minimum.
- The constraint G is not convex, meaning a solver could converge to a sub-optimal solution that for example asks an arm to rotate by 360° . Although in practice this can be easily avoided.

The numerical method chosen to find the optimal solution u^* and a^* is sequential quadratic programming (SQP) [30]. In an effort to make the resulting equations more understandable, they are derived by applying Newton's method to the first-order optimality conditions, which are:

- **Primal Feasibility:** $G = 0$, stating that the constraint must be satisfied at the solution.
- **Stationarity:** $\nabla O + \nabla G^T \lambda = 0$, stating that the gradient of the objective must be a linear combination of the gradients of every constraint (linked by some factors λ). This ensures that no direction exists, that is orthogonal to every constraint gradient, and at the same time is not orthogonal to the objective gradient. Moving in such a direction would otherwise allow the objective to be lowered while maintaining constraint satisfaction.

These two equations are stacked into one and the derivatives split into parts with respect to u and a (Eq. 7). Written in this form, the equation is equivalent to stating that the gradient of the Lagrangian $\mathcal{L} = O + G^T \lambda$ must be zero: $\nabla \mathcal{L} = 0$. Here below, all derivatives are written as subscripts, e.g. $O_u = \nabla_u O = \partial O / \partial u$. Their definitions are found in appendix C.

$$\nabla \mathcal{L}(u^*, a^*, \lambda^*) = \begin{bmatrix} O_u + G_u^T \lambda \\ O_a + G_a^T \lambda \\ G \end{bmatrix} \Big|_{\substack{u=u^* \\ a=a^* \\ \lambda=\lambda^*}} = 0 \quad (7)$$

To arrive at the SQP algorithm, the above equation is solved iteratively with Newton's method, by taking a linear approximation at the current best guess $[u, a, \lambda]$ (Eq. 8), and then finding an improvement step $[\delta u, \delta a, \delta \lambda]$ which solves that approximated equation (Eq. 11).

$$\nabla \mathcal{L}(u + \delta u, a + \delta a, \lambda + \delta \lambda) \approx H \begin{bmatrix} \delta u \\ \delta a \\ \delta \lambda \end{bmatrix} + K = 0 \quad (8)$$

$$H = \begin{bmatrix} O_{uu} + G_{uu} : \lambda & O_{ua} + G_{ua} : \lambda & G_u^T \\ O_{au} + G_{au} : \lambda & O_{aa} + G_{aa} : \lambda & G_a^T \\ G_u & G_a & 0 \end{bmatrix} \quad (9)$$

with: $(G_{\star\bullet} : \lambda)_{ij} = (\partial^2 G / \partial \star_i \partial \bullet_j)^T \lambda$

$$K = \begin{bmatrix} O_u + G_u^T \lambda \\ O_a + G_a^T \lambda \\ G \end{bmatrix} \quad (10)$$

$$\begin{bmatrix} \delta u \\ \delta a \\ \delta \lambda \end{bmatrix} = H^{-1} K \quad (11)$$

Iteratively updating the current best guess by the improvement step $[u, a, \lambda] \leftarrow [u, a, \lambda] + [\delta u, \delta a, \delta \lambda]$ should then converge to a solution $[u^*, a^*, \lambda^*]$, where u^*, a^* are such that they produce the desired force/torque setpoints F, M , and where the leftover degrees of freedom are such that they minimize the objective function O .

Although, usually in SQP only a scaled down improvement step $\alpha \cdot [\delta u, \delta a, \delta \lambda]$ is taken, with $\alpha \in [0, 1]$ determined by a line-search algorithm and some merit function. For this particular problem it was found to be sufficient to pick α such that no single arm-angle nor throttle step exceeds a fixed limit (Eq. 12).

$$\alpha = \min(1.0, \frac{\delta a_{lim}}{\max(\delta a)}, \frac{\delta u_{lim}}{\max(\delta u)}) \quad (12)$$

On MOMAV, the complete SQP based control allocation algorithm (Alg. 1) is implemented as a ROS node in C++ and executed on a *Radxa Rock 5A* computer, serving as the flight controller. The *PartialPivLU* solver from the *Eigen* C++ library is used to solve the linear system of equations (Eq. 11). This setup manages to converge to a solution reliably within 1 – 8 iterations, corresponding to 0.4 – 3.2ms in computation time. It manages to do so even during fast orientation changes, when the solution is furthest from the initial guess.

Algorithm 1 SQP algorithm

Input: q, F, M

Output: u, a, λ

$u, a, \lambda \leftarrow$ result from previous run

repeat

$H, K \leftarrow$ calculate with (9), (10)

$(\delta u, \delta a, \delta \lambda) \leftarrow$ solve $H^{-1} K$

$\alpha \leftarrow$ calculate with (12)

$(u, a, \lambda) \leftarrow (u, a, \lambda) + \alpha(\delta u, \delta a, \delta \lambda)$

$O, G \leftarrow$ calculate with (2), (6)

until $|O - O_{prev}| / O < \text{tol}$. **and** $\|G\| < \text{tol}$.

return u, a, λ

The main disadvantages of this algorithm, is that it does not come with any guarantees on computation time, due to its iterative nature. Although this problem can be successfully mitigated by a fast computer, there also exist other algorithms based on the Moore-Penrose pseudo-inverse method [17, 20, 21, 27, 28], which do not suffer from the same issue.

The main advantage on the other hand, is that the behavior of the control allocation can be intuitively tuned by modifying the objective function O . For example, a common challenge in variable-tilt drones is a singularity in arm-angle when an arm happens to point straight upwards during hover. A basic control allocation algorithm could demand the arm to suddenly rotate 180° in such a situation, which is unfeasible. By penalizing arm-angle velocities in the objective O this issue is easily avoided in the here presented SQP based algorithm.

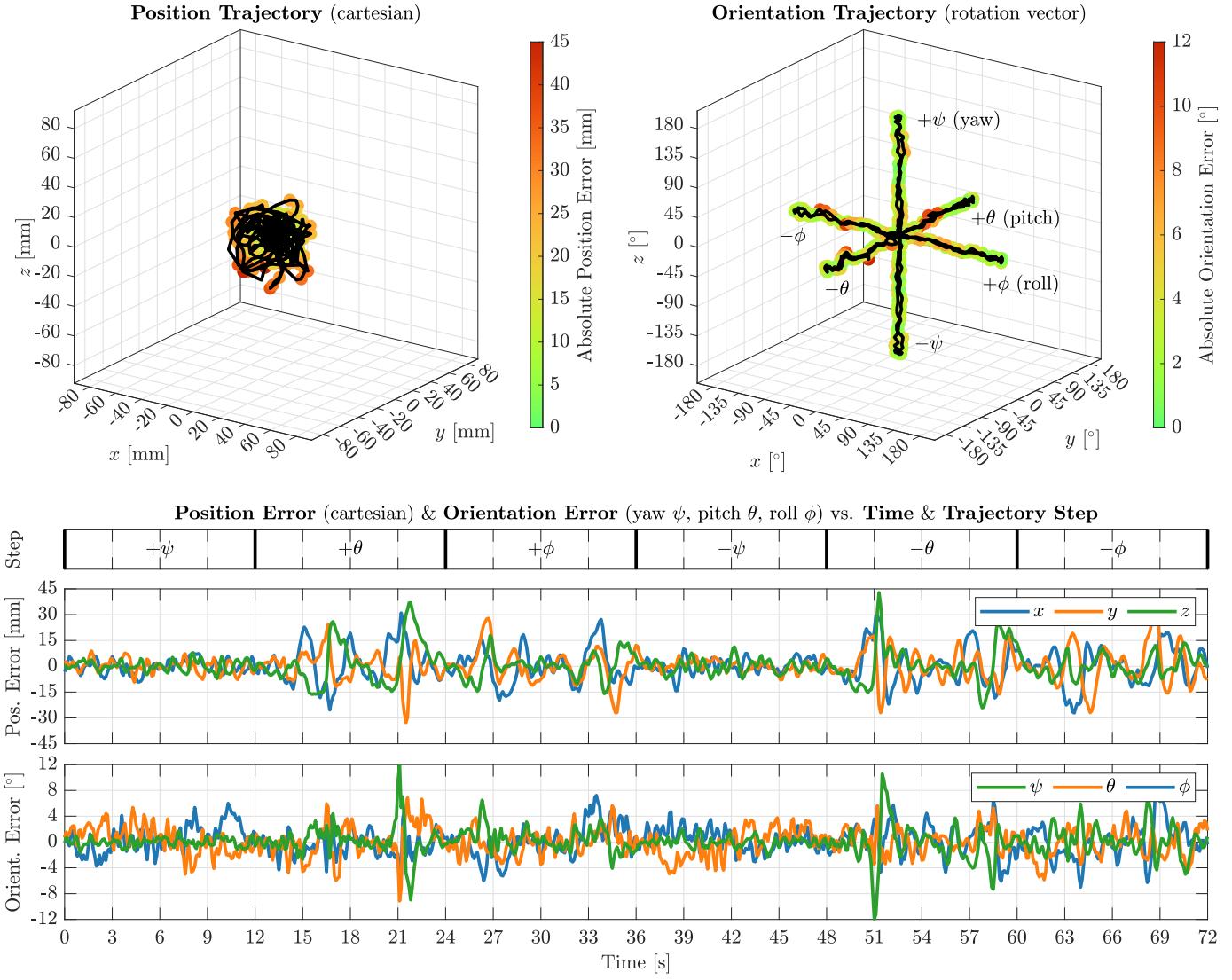


Fig. 8: Testflight data when sweeping through orientations and keeping the position fixed

V. TESTFLIGHTS

Finally, testflights are presented, to show how well MOMAV performs in reality. An *OptiTrack Prime^x 13W* tracking system is used to provide ground-truth position/orientation data for all tests. The same tracking data is also used on-board for control feedback during flight. Although, for stand-alone operation a *RealSense T265* can alternatively be equipped.

Two types of testflights are performed. In one, the orientation setpoint is held fixed, while the position setpoints sweep along the x , y , z axes. In the other, the position setpoint is held fixed, while the orientation setpoints sweep along the yaw (ψ), pitch (θ), roll (ϕ) axes. The sweeps are sequenced with each axis, in positive/negative direction, and always returning to the original pose in between. For example, the orientation-sweep steps are: $[0, +\psi, 0, +\theta, 0, +\phi, 0, -\psi, 0, -\theta, 0, -\phi]$. Orientation steps reach up to $\pm 180^\circ$ and position steps reach up to $\pm 500\text{mm}$, taking 6s per step with smooth accelerations/decelerations equal to half the maximum velocity.

During the orientation-sweep testflight (Fig. 8) a mean orientation error of 3.2° is measured ($\sigma: 2.0^\circ$, $P_{90}: 5.8^\circ$), as well as a mean position error of 12.0mm ($\sigma: 8.9\text{mm}$, $P_{90}: 24.8\text{mm}$). Both errors appear to correlate with angular velocity in pitch and roll, which is maximal around seconds 15, 21, 27, 33, 51, 57, 63, 69. Those are also the times where arm-angle changes are the largest, possibly indicating a connection with the variability in slip-ring losses mentioned previously. Around seconds 16, 20, 50, 58, two of the arms happen to align with the vertical z axis. At two of these times (20, 50) a spike in yaw error can be observed, but otherwise this testflight showcases the ability of the SQP based control allocation to deal with the aforementioned arm-angle singularities.

In comparison, the position-sweep testflight (Fig. 9) show a lower mean orientation error of 2.4° ($\sigma: 1.1^\circ$, $P_{90}: 4.0^\circ$), as well as a much lower mean position error of 6.4mm ($\sigma: 2.9\text{mm}$, $P_{90}: 10.6\text{mm}$). No correlation between position setpoints and errors is apparent.

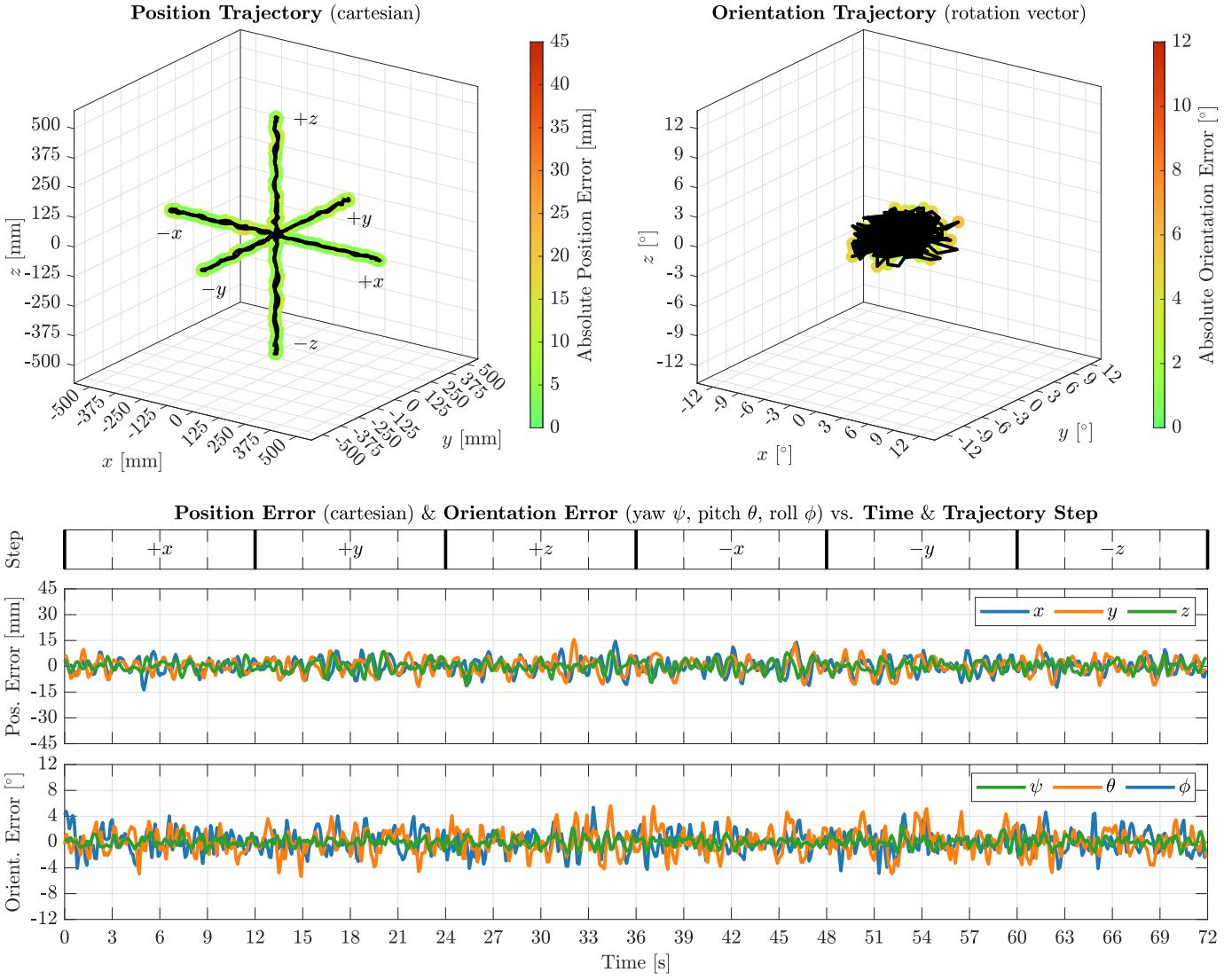


Fig. 9: Testflight data when sweeping through positions and keeping the orientation fixed

VI. CONCLUSION

MOMAV was able to showcase multiple interesting findings. It showed that an octahedral arm configuration with rotating arms is able to hover in any orientation, while wasting little thrust to motors having to fight each other, and requiring little surplus total thrust in the drone design compared to other fully-actuated drones. The rotating arms showed the viability of using modified hobbyist servos as performant actuators, supporting precise angular control across revolutions. They also showed the viability of using cheap DC motor brushes as high current slip-rings. A SQP algorithm was furthermore proposed as a fast and highly tunable method for control allocation, specifically tailored to drones with rotating arms.

All these findings were put to the test during actual flights of the prototype drone. They showed MOMEV's ability to control its orientation independently of its position and vice-versa. Only a minor shortcoming appeared as a slight increase in tracking error during specific phases of flight, which could

possibly be remedied by a proposed redesign of the slip-rings.

For readers interested in further information: An overview of the control architecture, as well as a detailed descriptions of each of its parts is available under https://marcoruggia.ch/home?projects_momav. The entire project source-code, 3D models of the drone, and raw data of all tests are also freely available under <https://github.com/mruggia/momav>.

Future development of MOMEV is not planned. If it were, it would include the implementation of a nonlinear motor thrust model, or even a learned model based on in-flight system identification. It is thought that this change could further reduce position/orientation tracking error. Also of interest would be tests showing how well the tracking accuracy in free-flight translates to accuracy in manipulation tasks involving contact with the environment.

APPENDIX

A. Drone Characteristics

Weight with batteries	2.4 kg
Weight without batteries	1.6 kg
Diagonal with propellers	590 mm
Diagonal without propellers	412 mm
Thrust (max. upwards)	6.4 kg
Hover flight time	8 min
Maiden flight	9 May 2022
Testflight count	274
Development time	8 months
Drone cost	2'200€

B. Drone Components

Core:

Computer	Radxa Rock 5A
Tracking	Intel RealSense T265 or OptiTrack Prime ^x 13W
Battery	Swaytronic 6S 2800mAh (2x)

Arm Assembly:

Servo	KST MS325
Controller	Atmel SAM D21
Driver	TI DRV8870
Brushes	RS550 DC Motor Brushes

Propulsion:

Motor	T-Motor F100
Driver	T-Motor MINI F45A 6S 4In1
Propeller	DJI Mavic Pro 8331 (3-blade-mod)

Structural:

Sheets	Swiss-Composite 1.5mm CFRP Sheet
Arm Tubes	Swiss-Composite 8x10mm CFRP Tube
3D Parts	Dutch Filaments PET-G with Carbon

C. Derivatives used in control allocation

$$\frac{\partial \mathbf{n}_i}{\partial \mathbf{a}_i} = \mathbf{x}_i \times \mathbf{n}_i \quad \frac{\partial^2 \mathbf{n}_i}{\partial \mathbf{a}_i^2} = \mathbf{x}_i \times (\mathbf{x}_i \times \mathbf{n}_i)$$

$$\frac{\partial \mathbf{f}_i}{\partial \mathbf{u}_i} = \mu \mathbf{n}_i \quad \frac{\partial^2 \mathbf{f}_i}{\partial \mathbf{u}_i^2} = 0 \quad \frac{\partial^2 \mathbf{f}_i}{\partial \mathbf{u}_i \partial \mathbf{a}_i} = \mu \frac{\partial \mathbf{n}_i}{\partial \mathbf{a}_i}$$

$$\frac{\partial \mathbf{f}_i}{\partial \mathbf{a}_i} = \mu \mathbf{u}_i \frac{\partial \mathbf{n}_i}{\partial \mathbf{a}_i} \quad \frac{\partial^2 \mathbf{f}_i}{\partial \mathbf{a}_i^2} = \mu \mathbf{u}_i \frac{\partial^2 \mathbf{n}_i}{\partial \mathbf{a}_i^2}$$

$$\frac{\partial \mathbf{m}_i}{\partial \mathbf{u}_i} = \mu(\mathbf{r}_i \times \mathbf{n}_i) + \tau s_i \mathbf{n}_i \quad \frac{\partial^2 \mathbf{m}_i}{\partial \mathbf{u}_i^2} = 0$$

$$\frac{\partial \mathbf{m}_i}{\partial \mathbf{a}_i} = \mu \mathbf{u}_i(\mathbf{r}_i \times \frac{\partial \mathbf{n}_i}{\partial \mathbf{a}_i}) + \tau u_i s_i \frac{\partial \mathbf{n}_i}{\partial \mathbf{a}_i}$$

$$\frac{\partial^2 \mathbf{m}_i}{\partial \mathbf{a}_i^2} = \mu \mathbf{u}_i(\mathbf{r}_i \times \frac{\partial^2 \mathbf{n}_i}{\partial \mathbf{a}_i^2}) + \tau u_i s_i \frac{\partial^2 \mathbf{n}_i}{\partial \mathbf{a}_i^2}$$

$$\frac{\partial^2 \mathbf{m}_i}{\partial \mathbf{u}_i \partial \mathbf{a}_i} = \mu(\mathbf{r}_i \times \frac{\partial \mathbf{n}_i}{\partial \mathbf{a}_i}) + \tau s_i \frac{\partial \mathbf{n}_i}{\partial \mathbf{a}_i}$$

$$\frac{\partial \mathbf{G}}{\partial \mathbf{u}_i} = \begin{bmatrix} \partial f_i / \partial u_i \\ \partial m_i / \partial u_i \end{bmatrix} \quad \frac{\partial^2 \mathbf{G}}{\partial \mathbf{u}_i \partial \mathbf{u}_j} = \begin{cases} i = j : & \begin{bmatrix} \partial^2 f_i / \partial u_i^2 \\ \partial^2 m_i / \partial u_i^2 \end{bmatrix} \\ i \neq j : & 0 \end{cases}$$

$$\frac{\partial \mathbf{G}}{\partial \mathbf{a}_i} = \begin{bmatrix} \partial f_i / \partial a_i \\ \partial m_i / \partial a_i \end{bmatrix} \quad \frac{\partial^2 \mathbf{G}}{\partial \mathbf{a}_i \partial \mathbf{a}_j} = \begin{cases} i = j : & \begin{bmatrix} \partial^2 f_i / \partial a_i^2 \\ \partial^2 m_i / \partial a_i^2 \end{bmatrix} \\ i \neq j : & 0 \end{cases}$$

$$\frac{\partial^2 \mathbf{G}}{\partial \mathbf{u}_i \partial \mathbf{a}_j} = \frac{\partial^2 \mathbf{G}}{\partial \mathbf{a}_j \partial \mathbf{u}_i} = \begin{cases} i = j : & \begin{bmatrix} \partial^2 f_i / \partial u_i \partial a_i \\ \partial^2 m_i / \partial u_i \partial a_i \end{bmatrix} \\ i \neq j : & 0 \end{cases}$$

REFERENCES

- [1] A. Ollero et al. “Past, Present, and Future of Aerial Robotic Manipulators”. In: *IEEE T-RO* (2022), pp. 626–645. DOI: 10.1109/TRO.2021.3084395.
- [2] R. Rashad et al. “Fully Actuated Multirotor UAVs: A Literature Review”. In: *IEEE RA Magazine* (2020), pp. 97–107. DOI: 10.1109/MRA.2019.2955964.
- [3] T. Howard et al. “The Lynchpin—A Novel Geometry for Modular, Tangential, Omnidirectional Flight”. In: *SAE Int. J. Aerosp. 16(3)* (2023), pp. 291–303. DOI: 10.4271/01-16-03-0018.
- [4] A. Flores and G. Flores. “Fully actuated Hexa-rotor UAV: Design, construction, and control. Simulation and experimental validation”. In: *ICUAS* (2022), pp. 1497–1503. DOI: 10.1109/ICUAS54217.2022.9836105.
- [5] L. Ma et al. “ADRC-Backstepping-Compensation Control for A Novel Aerial Manipulator System”. In: (2021). DOI: 10.21203/rs.3.rs-676982/v1.
- [6] R. Rashad, F. Califano, and S. Stramigioli. “Port-Hamiltonian Passivity-Based Control on SE(3) of a Fully Actuated UAV for Aerial Physical Interaction Near-Hovering”. In: *IEEE RA Letters 4.4* (2019), pp. 4378–4385. DOI: 10.1109/LRA.2019.2932864.
- [7] S. Park et al. “ODAR: Aerial Manipulation Platform Enabling Omnidirectional Wrench Generation”. In: *IEEE/ASME 23.4* (2018), pp. 1907–1918. DOI: 10.1109/TMECH.2018.2848255.
- [8] M. Tognon and A. Franchi. “Omnidirectional Aerial Vehicles With Unidirectional Thrusters: Theory, Optimal Design, and Control”. In: *IEEE RA Letters 3.3* (2018), pp. 2277–2282. DOI: 10.1109/LRA.2018.2802544.
- [9] F. von Frankenberg and S. Nokleby. “Disturbance Rejection in Multi-Rotor Unmanned Aerial Vehicles Using a Novel Rotor Geometry”. In: *CDSR* (2017). DOI: 10.11159/cdsr17.130.

- [10] Y. Lei et al. “Aerodynamic design on the non-planar rotor system of a multi-rotor flying robot (MFR)”. In: *IEEE ROMA* (2017), pp. 1–5. DOI: 10.1109/ROMA.2017.8231740.
- [11] D. Brescianini and R. D’Andrea. “Design, modeling and control of an omni-directional aerial vehicle”. In: *IEEE ICRA* (2016), pp. 3261–3266. DOI: 10.1109/ICRA.2016.7487497.
- [12] A. Nikou, G. Gavridis, and K. Kyriakopoulos. “Mechanical design, modelling and control of a novel aerial manipulator”. In: *IEEE ICRA* (2015), pp. 4698–4703. DOI: 10.1109/ICRA.2015.7139851.
- [13] G. Jiang and R. Voyles. “Hexrotor UAV platform enabling dexterous interaction with structures-flight test”. In: *IEEE SSRR* (2013), pp. 1–6. DOI: 10.1109/SSRR.2013.6719377.
- [14] D. Toratani. “Research and development of double tetrahedron hexa-rotorcraft (DOT-HR)”. In: (2012). URL: <https://api.semanticscholar.org/CorpusID:195778790>.
- [15] E. Cuniato et al. “Design and Control of a Micro Overactuated Aerial Robot with an Origami Delta Manipulator”. In: (2023). arXiv: 2305.01961.
- [16] J. Buzzatto et al. “The New Dexterity Omnidirotor Platform: Design, Modeling, and Control of a Modular, Versatile, All-Terrain Vehicle”. In: *IEEE/RSJ IROS* (2021), pp. 6336–6343. DOI: 10.1109/IROS51168.2021.9635995.
- [17] X. Xiongshi, K. Watanabe, and I. Nagai. “Backstepping Control for a Tandem Rotor UAV Robot with Two 2-DOF Tiltable Coaxial Rotors”. In: *Journal of Robotics and Control (JRC)* 2 (2021). DOI: 10.18196/jrc.25116.
- [18] S. J. Lee et al. “Fully Actuated Autonomous Flight of Thruster-Tilting Multirotor”. In: *IEEE/ASME* 26.2 (2021), pp. 765–776. DOI: 10.1109/TMECH.2020.2999586.
- [19] P. Zheng et al. “TiltDrone: A Fully-Actuated Tilting Quadrotor Platform”. In: *IEEE RA Letters* 5.4 (2020), pp. 6845–6852. DOI: 10.1109/LRA.2020.3010460.
- [20] K. Bodie et al. “An Omnidirectional Aerial Manipulation Platform for Contact-Based Inspection”. In: *RSS*. RSS2019 (2019). DOI: 10.15607/rss.2019.xv.019.
- [21] M. S. Kamel et al. “The Voliro Omnidirectional Hexacopter: An Agile and Maneuverable Tiltable-Rotor Aerial Vehicle”. In: *IEEE RA Magazine* PP (2018), pp. 1–1. DOI: 10.1109/MRA.2018.2866758.
- [22] A. Bin Junaid et al. “Design and Implementation of a Dual-Axis Tilting Quadcopter”. In: *Robotics* 7.4 (2018). ISSN: 2218-6581. DOI: 10.3390/robotics7040065.
- [23] M. Ryll, D. Bicego, and A. Franchi. “Modeling and control of FAST-Hex: A fully-actuated by synchronized-tilting hexarotor”. In: *IEEE/RSJ IROS* (2016), pp. 1689–1694. DOI: 10.1109/IROS.2016.7759271.
- [24] M. Odelga, P. Stegagno, and H. Bültlhoff. “A fully actuated quadrotor UAV with a propeller tilting mechanism: Modeling and control”. In: *IEEE AIM* (2016), pp. 306–311. DOI: 10.1109/AIM.2016.7576784.
- [25] D. Kastelan, M. Konz, and J. Rudolph. “Fully Actuated Tricopter with Pilot-Supporting Control”. In: *IFAC-PapersOnLine* 48.9 (2015), pp. 79–84. ISSN: 2405-8963. DOI: 10.1016/j.ifacol.2015.08.063.
- [26] A. Moutinho, E. Mateos, and F. Cunha. “The Tilt-Quadrotor: Concept, Modeling and Identification”. In: *IEEE ICARSC* (2015), pp. 156–161. DOI: 10.1109/ICARSC.2015.38.
- [27] M. Ryll, H. Bültlhoff, and P. Giordano. “A Novel Over-actuated Quadrotor Unmanned Aerial Vehicle: Modeling, Control, and Experimental Validation”. In: *IEEE TCST* 23.2 (2015), pp. 540–556. DOI: 10.1109/TCST.2014.2330999.
- [28] P. Segui-Gasco et al. “A novel actuation concept for a multi rotor UAV”. In: *ICUAS* (2013), pp. 373–382. DOI: 10.1109/ICUAS.2013.6564711.
- [29] B. Beauregard. *Introducing Proportional On Measurement*. <http://brettbeauregard.com/blog/2017/06/introducing-proportional-on-measurement/>. 2017.
- [30] S. Wright, J. Nocedal, et al. “Numerical optimization”. In: *Springer Science* 35.67-68 (1999), p. 7.