Delft University of Technology

Fault Tolerant Control in Over-Actuated Hybrid Tilt-Rotor Unmanned Aerial Vehicles

Mancinelli, A.; Voß, N.; Smeur, E.J.J.

DOI
10.2514/6.2025-0317

Publication date
2025

Document Version
Final published version

Published in
Proceedings of the AIAA SCITECH 2025 Forum

Citation (APA)
Mancinelli, A., Voß, N., & Smeur, E. J. J. (2025). Fault Tolerant Control in Over-Actuated Hybrid Tilt-Rotor Unmanned Aerial Vehicles. In *Proceedings of the AIAA SCITECH 2025 Forum* Article AIAA 2025-0317 https://doi.org/10.2514/6.2025-0317

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

AIAA SciTech Forum
6-10 January 2025, Orlando, FL
AIAA SCITECH 2025 Forum

10.2514/6.2025-0317

Fault Tolerant Control in Over-Actuated Hybrid Tilt-Rotor Unmanned Aerial Vehicles

Alessandro Mancinelli[*], Nico Voß[†], and Ewoud J. J. Smeur[‡]
Delft University of Technology, Delft, The Netherlands

Quad-planes combine hovering and Vertical Takeoff and Landing (VTOL) capabilities with efficient forward flight. However, they are often vulnerable to gust disturbances and are not well-equipped to handle actuator faults. Dual-axis Tilt-Rotor quad-planes offer enhanced maneuverability due to their overactuation, which also enables stable hovering even after actuator failures. These vehicles can employ an Incremental Nonlinear Dynamic Inversion (INDI) controller paired with a nonlinear Sequential Quadratic Programming (SQP) Control Allocation (CA) algorithm that can find hover solutions under actuator failure conditions. We explore both a combined allocation of linear and angular accelerations and a cascaded allocation scheme. Due to the large required changes in roll and pitch angles, the cascaded approach is selected for this research. The proposed algorithm was tested on a flying vehicle, demonstrating successful hovering and position control capabilities under a simulated Fault Detection and Identification (FDI) mechanism.

Downloaded by Technische Universiteit Delft on January 7, 2025 | http://arc.aiaa.org | DOI: 10.2514/6.2025-0317

I. Introduction

The development of Vertical Takeoff and Landing (VTOL) aircraft has been driven by the benefit of combining efficient high-speed, long-range flight with the capability of landing on unprepared or size-restricted areas. Before the era of unmanned systems, several projects investigated full-scale concepts to achieve this combination by using separate or multi-purpose propulsion systems. Despite the increased mechanical and control complexity, using a single propulsion system for both modes of flight is desirable for efficiency. Tilt-wing and tilt-rotor systems have established themselves as the current industry standard, enabling a single propulsion system across hover, transition, and forward flight regimes by adjusting the thrust vector accordingly.

For smaller and more agile systems, research has expanded toward utilizing tilt mechanisms for additional functions, such as gust rejection or physical interaction tasks [1, 2]. At the same time, powerful System-on-Chip (SoC) solutions offer affordable computing power to implement complex real-time Control Allocation (CA) algorithms within limited power and weight budgets [3]. The addition of independent rotor tilt axes can increase the degrees of freedom the vehicle can independently control.

By adding only two additional servos and spherical joints to a quadcopter, Zheng et al. [4] were able to control the common motor tilt and demonstrated attitude-independent thrust. This configuration tilts the common thrust vector via a gimbal linkage to control motor rotation parallel to the body's pitch and roll axes.

Several other studies have previously investigated the increased maneuverability resulting from tilt configurations, including Junaid et al. [5], who specifically noted improvements in maneuvering flight and obstacle avoidance. A step further in complexity is the independent tilt of individual rotors: Mousaei et al. [6] developed a quad-plane with independent single-axis tilt along the vehicle's pitch axis on all four rotors and investigated motor failure in hover and forward flight. They successfully simulated recovery from motor failure in both hover and forward flight modes.

In previous research [7], we took this concept one step further and developed a quad-plane with independent dual-axis tilt of each motor, shown in Figure 1. The ability to directly generate forces independent of moments gives the vehicle increased disturbance rejection and maneuverability capabilities. Because of the nonlinearities in tilting each thrust vector, a nonlinear optimization control allocation approach is necessary. Compared to the previously introduced quad-plane [6], the additional tilting of motors parallel to the vehicle's roll axis allows for lateral thrust vectoring and reorientation in roll and pitch under motor failure.

[*]Ph.D. Researcher, Faculty of Aerospace Engineering, MAVLab.
[†]M.Sc. Student, Faculty of Aerospace Engineering, MAVLab.
[‡]Assistant Professor, Faculty of Aerospace Engineering, MAVLab.

Copyright © 2025 by Alessandro Mancinelli, Nico Voß, and Ewoud J. J. Smeur. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission.

Downloaded by Technische Universiteit Delft on January 7, 2025 | http://arc.aiaa.org | DOI: 10.2514/6.2025-0317

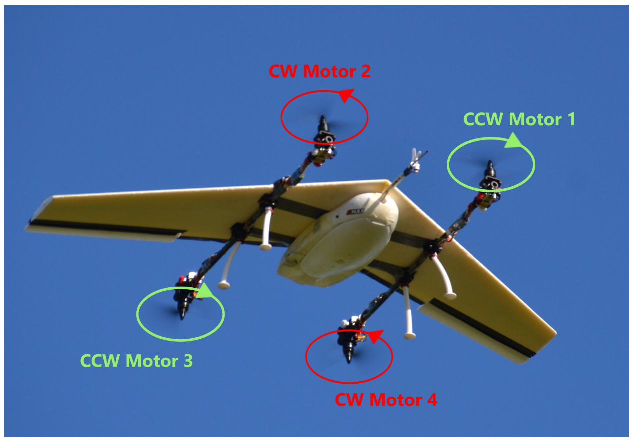

Fig. 1 Dual-axis tilt Tilt Rotor Unmanned Aerial Vehicle (TRUAV) quad-plane developed by Mancinelli et al. [7]

In order to tolerate actuator failures, two major approaches to Fault Tolerant Control (FTC) can be discerned. In active FTC, the system is actively monitored, and faults need to be detected and communicated to the controller. With information on the failure, which relies on a Fault Detection and Identification (FDI) mechanism, the controller takes action to mitigate the failure. In passive FTC, the controller is instead designed to be robust to failures within established limits. This usually results in a more conservative design and reduced overall performance but does not require any active monitoring component [8–10]. As both introduced tilt-rotor quad-planes rely on an onboard dynamics model for their control allocation, a means to inform the controller of the changed configuration is required. For this reason, active fault tolerance is applied to update the model accordingly.

Wang and Sung [11] focused on the effect a lifting body has on more classical types of recovery for quadrotors and proposed a novel Incremental Adaptive Sliding Mode Control (I-ASMC) approach to mitigate uncertainties in the modeled aerodynamic forces and interactions of rotor and wing [11]. The proposed solution simulated flight with only three rotors and had the spinning quad-plane follow a rectangular reference trajectory. This relaxed hover state, allowing yaw rotation, is a proven approach to overcome the inherent under-actuation of quadcopters [12]. Sun et al. previously demonstrated flight with two rotors for a commercial drone [13], and Zhang et al. designed a vehicle capable of tracking position with only a single actuator [14].

In this paper, we demonstrate fault-tolerant control of a dual-axis tilt quadplane, with onboard optimization of a new static hover condition. Maintaining static hover has the advantage of avoiding the need to model and adapt to the complex aerodynamic interactions of a spinning wing. We developed a cascaded nonlinear control allocation strategy that separates attitude control from actuator command optimization. By still including the actuators in the attitude control loop, the obtained pitch and roll angle commands respect the actuator limits, even though the actuator commands are not used and are an output of the inner control loop. Test flights demonstrate the capability of the control framework to deal with actuator faults of the dual-axis tilt quad-plane.

II. Method

A. Reference Frames Definition - Equations of Motion

For the setup of the controller, a number of different right-handed reference frames are of importance to understand the orientations, commands and actuator controls. In Figure 2 we show the following reference frames:

- Γ_e Earth reference frame (NED):
  - Origin fixed to earth surface reference point
  - x_e pointing towards North
  - y_e pointing towards East
  - z_e positive to Earth center
- Γ_b Body reference frame:
  - Origin is fixed to vehicle Center of Gravity:
  - x_b pointing forward along vehicle roll axis
  - y_b pointing out of right wing

Downloaded by Technische Universiteit Delft on January 7, 2025 | http://arc.aiaa.org | DOI: 10.2514/6.2025-0317

  - z_b pointing down
- Γ_c Control reference frame:
  - Origin is fixed to vehicle Center of Gravity:
  - x_c pointing towards nose projected onto earth surface
  - y_c pointing right seen from above, perpendicular to z_c and x_c
  - z_c pointing down to Earth center
- Γ_p^i Propeller reference frame: Origin is fixed at $i-$th gimbal point, the axis directions are aligned with the body frame under zero gimbal controls, which can be seen on the front right motor 2 in Figure 3.

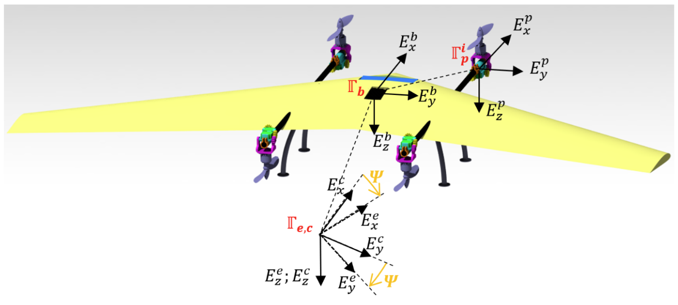

Fig. 2 Definition of reference frames [15]

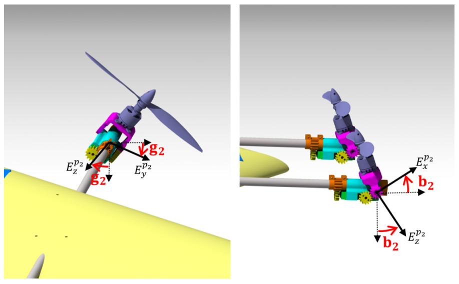

Fig. 3 Definition of gimbal angles as per Mancinelli et al.[15]

Due to the independent actuation of the rotors, a set of four primary rotation matrices is required to establish the equations of motion and control laws. The first rotation matrix describes the transformation from earth to control reference frames Γ_e to Γ_c:

$$R_{ec} = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

This transformation is used primarily within the position error controller to generate the desired velocities and accelerations in the control frame. Additionally, the transform between the body reference Γ_b and control reference frame Γ_c is given by:

$$R_{cb} = \begin{bmatrix} \cos(\theta) & \sin(\phi)\sin(\theta) & \cos(\phi)\sin(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ -\sin(\theta) & \sin(\phi)\cos(\theta) & \cos(\phi)\cos(\theta) \end{bmatrix} \tag{2}$$

Finally, due to the independent orientation of the motors with respect to the body frame, computation of forces and moments requires an additional coordinate transformation for each motor. The transformation of the $i-$th motor to the body frame $\Gamma_p^i \longrightarrow \Gamma_b$ is described by:

$$R_{bp}^i = \begin{bmatrix} \cos(b^i) & 0 & \sin(b^i) \\ \sin(g^i)\sin(b^i) & \cos(g^i) & -\sin(g^i)\cos(b^i) \\ -\cos(g^i)\sin(b^i) & \sin(g^i) & \cos(g^i)\cos(b^i) \end{bmatrix} \tag{3}$$

The definition of angles b_i, g_i is shown in Figure 3 denoting the elevation and azimuth tilt of the considered motor respectively. The dynamics of the vehicle are expressed by the following equations:

$$\begin{cases} \ddot{P}_c = \frac{1}{m}\left(F^p + F^a\right) + g\hat{z}_e \\ \dot{\omega} = I_b^{-1}\left(-\omega \times I_b\omega + M^t + M^d + M^a + M^{\delta_a}\right) \end{cases} \tag{4}$$

where $\ddot{P}_c$ are the linear accelerations within the control reference frame Γ_c and ω is the vector of rotational rates of the vehicle in the body frame. F_p denotes the sum of thrust generated by the individual motors, rotated to the control frame and F_a denotes the aerodynamic forces in the control frame.

Downloaded by Technische Universiteit Delft on January 7, 2025 | http://arc.aiaa.org | DOI: 10.2514/6.2025-0317

The thrust forces and moments generated by the rotors can be computed from the thrust coefficient K_p^T, torque coefficient K_p^M and motor speed Ω_i as follows:

$$F^p = \sum_{i=1}^{N} R_{cb} R_{bp}^i \begin{pmatrix} 0 \\ 0 \\ -K_p^T \Omega_i^2 \end{pmatrix}; \qquad M^d = \sum_{i=1}^{N} -R_{bp}^i \begin{pmatrix} 0 \\ 0 \\ K_p^M \Omega_i^2 \end{pmatrix} (-1)^i. \tag{5}$$

In order to decrease the problem complexity for online computation, certain moment components are simplified from the full model. The following motor-related moment terms are assumed negligible: precession (both from gimbal and body rate), torque due to rotational speed changes, and torque from moving the motor mass about the gimbal. The interaction between the lifting body and the inflow generated by the motors was also excluded from the model. A previous study investigated the roll moment interaction between the wing and the motors [16], but this was not further considered in this project. The remaining components are M^t, the torque generated from motor thrust, and M^a, the aerodynamic moments acting on the vehicle. For a more detailed analysis and definition of the equations of motion, the reader can refer to [15].

B. Controller Layout

The vehicle's controller is based on the Incremental Nonlinear Dynamic Inversion (INDI) implementation of [15] and consists of two main components. The primary component is a single-loop Control Allocation algorithm, which generates commands for 13 physical actuators (8 tilting servos, 4 motor RPM commands, and 1 aileron servo pair) as well as two virtual actuators: the vehicle's roll and pitch angles. The Control Allocation algorithm determines the optimal actuator commands to achieve the desired accelerations by minimizing a cost function that includes a model of the vehicle dynamics, as derived in the previous section. The Control Allocation algorithm receives angular and linear acceleration inputs generated by a linear error controller.

The error controller provides acceleration references for both linear and angular accelerations. For linear acceleration references, the error controller receives a setpoint for the desired vehicle position. It then uses feedback from the vehicle's current position and speed to generate the necessary linear acceleration references. A similar method is applied for angular acceleration generation, where a PD (Proportional-Derivative) error controller, with feedback on body rates and Euler angles, is used to generate the angular acceleration references. A schematic representation of the architecture can be seen in Figure 4.

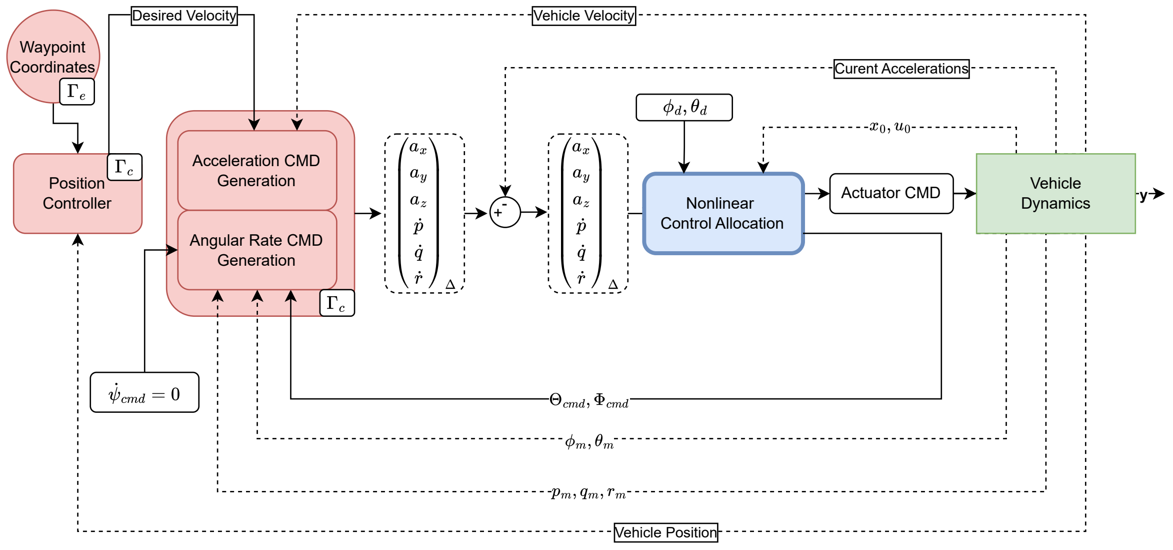

Fig. 4 The original controller layout including the error controllers.

Downloaded by Technische Universiteit Delft on January 7, 2025 | http://arc.aiaa.org | DOI: 10.2514/6.2025-0317

C. The Cost Function

The CA algorithm generates the control input commands by minimizing a cost function. By adjusting the cost function, specific control objectives can be prioritized. This approach integrates with the previously presented dynamics and control scheme by aiming to match the acceleration increments generated by the upstream error controllers, using the equations of motion in Equation 4.

The following equation shows a breakdown of the cost function structure and its individual components:

$$C(u) = ||W_v(f(x_0, u) - \nu_n)||^2 + \gamma_u||W_u(u - u_d)||^2 \tag{6}$$

$$u_s = \text{ arg min } C(u)$$

$$\text{subject to}$$
$$u_{min} < u < u_{max},$$

where the goal is to minimize the cost $C(u)$ by finding a u_s that minimizes the difference between the desired and predicted acceleration increment, $f(x_0, u) - \nu_n$. Here, $f(x_0, u)$ represents the modeled linear and angular acceleration, according to Equation 4 from the control input u and the vehicle states x_0, while ν_n is the vector of linear and angular acceleration increments targeted by the error controller.

As a secondary objective for the optimization process, the cost function also minimizes the difference between commanded and desired actuator settings, $u - u_d$. The control input vector u containing the computed commanded controls and attitudes, is structured as follows:

$$\begin{aligned} u = (&\Omega_1, \Omega_2, \Omega_3, \Omega_4, b_1, b_2, b_3, b_4, \\ &g_1, g_2, g_3, g_4, \delta_a, \theta_{cmd}, \phi_{cmd}), \end{aligned} \tag{7}$$

where Ω_i denote individual motor rotational speed in rad/s, b_i the motor gimbal elevation, g_i the motor gimbal azimuth and δ_a the aileron deflection. The definition of the tilt angles can be seen in Figure 3. The desired control input vector is chosen primarily to minimize motor usage, as follows:

$$\begin{aligned} u_d = (&150, 150, 150, 150, 0, 0, 0, 0, \\ &0, 0, 0, 0, 0, \theta_d = 0, \phi_d = 0). \end{aligned} \tag{8}$$

The primary and secondary objectives of the optimization are differentiated using the scaling factor γ_u, which is set to a very small value. The weighting matrices W_v and W_u are used to prioritize or penalize specific control objectives or actuators respectively. Finally, u_{min} and u_{max} provide the control input constraints for the optimization problem, as specified in Table 1.

D. Cascaded Control Structure

When a motor fails, the required attitude change that allows static hover can be very significant. A limitation of the combined control allocation as in Equation 6, is that a new attitude is found with actuator commands that satisfy the linear and angular acceleration control objective at that new attitude. However, the forces generated by the physical actuators will be valid for the associated computed attitude, which has a much lower bandwidth. Therefore, accelerations initially occur in the wrong frame when big attitude changes are commanded.

To address this limitation, we developed a new control framework that separates the computation of attitude commands from physical actuator commands in a cascaded manner. The controller sequentially solves two distinct optimization problems. The first optimization problem, similar to the control problem in Equation 6, determines the optimal attitude command for the vehicle in alignment with physical actuator limitations, based on linear acceleration target increments. In this initial optimization run, the angular acceleration targets are set to zero, assuming that the vehicle will reach a steady state at the new attitude. Although the first optimization run also produces preliminary physical control input commands, these are disregarded. However, it is important to include them, such that the optimizer can take the required actuator inputs (and limits) at this final state into account.

The attitude commands from the first optimization run are then fed to the attitude error controller, which generates the required angular acceleration commands to achieve the targeted attitude. These angular acceleration commands are subsequently combined with the initial linear acceleration targets and passed to the second optimization run, which computes the final physical actuator commands.

Downloaded by Technische Universiteit Delft on January 7, 2025 | http://arc.aiaa.org | DOI: 10.2514/6.2025-0317

The derivation of the updated control laws used in the first optimization run is as follows:

$$\begin{pmatrix} \nu_n^p & u_d & u_0 \end{pmatrix} \xmapsto[CA-1]{} \begin{pmatrix} \cancel{\Omega_{1-4}} & \cancel{b_{1-4}} & \cancel{g_{1-4}} & \cancel{\delta_a} & \theta_{cmd} & \phi_{cmd}, \end{pmatrix} \tag{9}$$

where the term ν_n^p represents the modified pseudo-control vector, containing only the linear acceleration components of the error controller, defined as follows:

$$\nu_n^p = \Delta \begin{pmatrix} a_x & a_y & a_z & 0 & 0 & 0 \end{pmatrix}. \tag{10}$$

The commanded attitude determined through Equation 9 is then fed to an intermediate error controller. This error controller incorporates a proportional gain on the Euler angle error to produce desired Euler angles derivatives. The Euler angle kinematics are then used to convert the desired Euler angle rates to desired body rates:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix}_d = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \cos\theta\sin\phi \\ 0 & -\sin\phi & \cos\theta\cos\phi \end{bmatrix} \begin{bmatrix} \dot\phi \\ \dot\theta \\ \dot\psi \end{bmatrix}_d = R(\phi,\theta) \begin{bmatrix} \dot\phi \\ \dot\theta \\ \dot\psi \end{bmatrix}_d \tag{11}$$

A scheme of the linear error controller is shown in Figure 5.

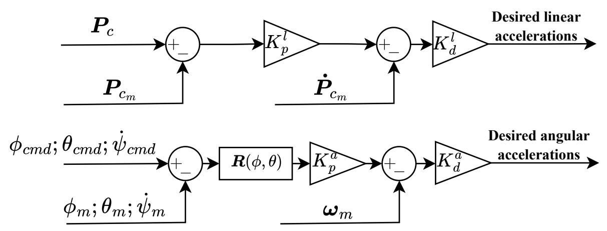
P_c
P_{c_m}
K_p^l
$\dot{P}_{c_m}$
K_d^l
Desired linear accelerations
$\phi_{cmd}; \theta_{cmd}; \dot\psi_{cmd}$
$\phi_m; \theta_m; \dot\psi_m$
$R(\phi,\theta)$
K_p^a
ω_m
K_d^a
Desired angular accelerations

Fig. 5 Linear error controller used to determine the linear and angular acceleration commands.

The angular acceleration increments generated through the error controller, are then added to the initial linear acceleration increments to form the final acceleration increment vector for the second optimization run:

$$\nu_n^s = \Delta \begin{pmatrix} a_x & a_y & a_z & \dot p & \dot q & \dot r \end{pmatrix}. \tag{12}$$

This stage has been modified to only compute outputs for the physical actuators, and the attitude angles are not optimized. Instead, only the attitude generated by the first stage optimizer is used.

$$\begin{pmatrix} \nu_n^s & u_d^s & u_0^s \end{pmatrix} \xmapsto[CA-2]{} \begin{pmatrix} \Omega_{1-4} & b_{1-4} & g_{1-4} & \delta_a \end{pmatrix}_{cmd} \tag{13}$$

Here, u_d^s and u_0^s represent respectively the desired and current control input vectors used in the second optimization run. These vectors are identical to those from the first optimization run, except that they exclude the last two attitude elements.

The revised architecture can be seen in Figure 6. Despite the additional computation step, the sampling frequency for the optimization algorithm consistently remains above 200 Hz.

Downloaded by Technische Universiteit Delft on January 7, 2025 | http://arc.aiaa.org | DOI: 10.2514/6.2025-0317

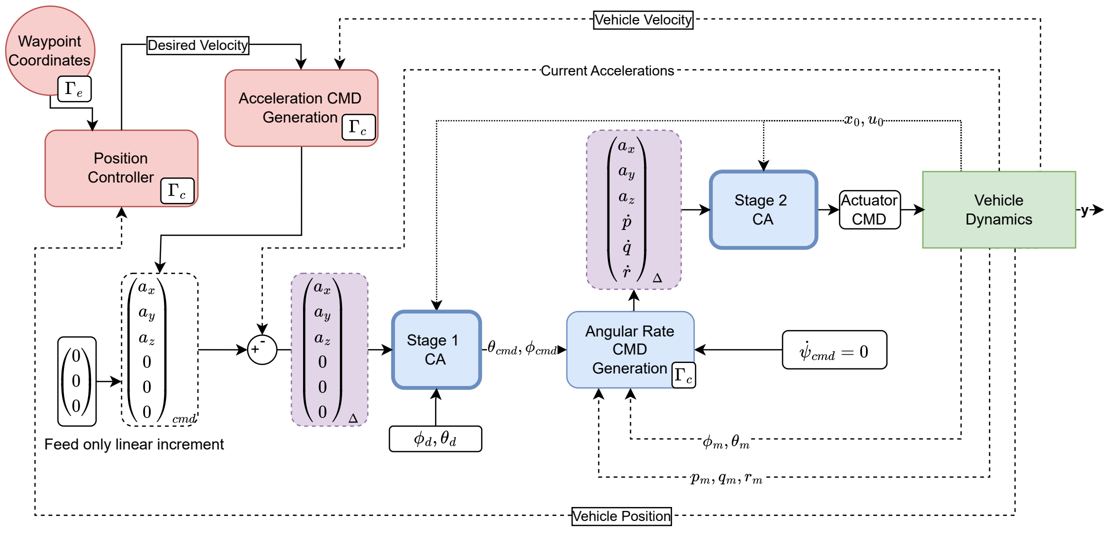

Fig. 6 The cascaded layout optimizing attitude and actuators in different steps.

E. The Control Allocation solver

The presented optimization problem is solved using the Sequential Quadratic Programming (SQP) approach, and makes use of the Matlab *fmincon* function. This implementation includes suggested improvements to K. Schittkowski's well-documented explanation of the algorithm [17] and is built on the work of Nocedal and Wright [18]. The choice of using the Matlab function also allows for the use of the Coder toolbox, accelerating the process of implementing the developed controllers on the drone.

F. Test Scenario and optimal attitude identification

As the most power-intensive flight condition, motor failure during hover was selected as the primary failure case for investigation. While forward flight offers the advantage of lift generated by the wing and effective use of aerodynamic control surfaces, these benefits are absent in hover. Additionally, the failure of a motor renders the respective tilt servos ineffective, as no thrust vector can be generated.

With an identified thrust coefficient of $K_T^P = 1.106465 \times 10^{-5}, \mathrm{N}/(\mathrm{rad/s})^2$ and a maximum motor speed of 1000 rad/s recorded with the current power system, the maximum thrust each motor can provide, calculated from Equation 5, is approximately 11.1 N. For a vehicle mass of 2.5 kg, this results in a hover thrust-to-weight ratio of about 1.8 with all engines operational. The failure of a single motor reduces the maximum thrust and thrust-to-weight ratio to 33.2 N and 1.35, respectively. Further motor failures would leave insufficient thrust, without accounting for the need to generate moments and linear accelerations.

This implies that current actuator fault control strategies, which rely on switching off the opposing motor in the event of a motor failure, would not provide enough thrust for sustained hover flight post-failure. Instead, the vehicle must fully utilize the remaining three operational motors, reorienting its attitude to achieve a new symmetric thrust equilibrium that ensures moment balance.

To analyze the behavior of the first CA optimizer in more detail, we examined the failure of motor 3 (back right). The cost function output from the first optimization run was evaluated across different attitude values. The resulting surface plot is shown in Figure 7. From the plot, we observe that the optimal attitude after the failure of the back right motor corresponds to $\theta = 67°$ and $\phi = 90°$. Another significant observation is that the shape of the cost function may direct the optimization process toward a local minimum located at $\theta = -15°$ and $\phi = -90°$. This aspect is crucial when defining the attitude constraints under actuator failure conditions.

Downloaded by Technische Universiteit Delft on January 7, 2025 | http://arc.aiaa.org | DOI: 10.2514/6.2025-0317

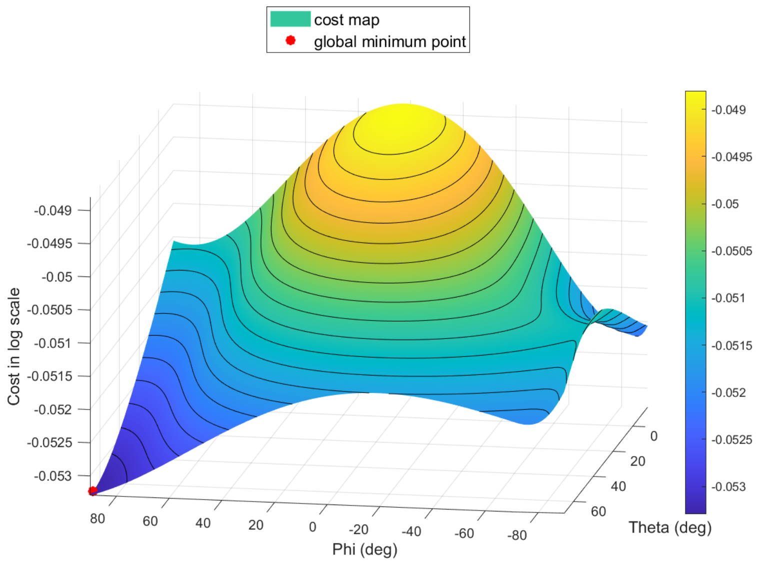

Fig. 7 Cost function value for different attitude angles. The cost function numerical value was obtained using logarithm with base 10.

G. Failure Information: Constraint Sets and Weighting Matrices

It is essential to implement a mechanism that informs the controller of failures and prevents the allocation algorithm from utilizing ineffective actuators. This fault information mechanism is straightforward to implement, as the platform uses very high-speed feedback from all its actuators, as described in [19]. The failure information is communicated to the controller, which subsequently restricts the control input constraints during both the first and second optimization runs. The constraints for the non-faulty actuators can be found in Table 1, which are applied during regular flight. It is important to note that pitch and roll limits were established to prevent significant angular ambiguities that could arise from the ZYX rotation order.

Once the failure is triggered, the controller internally switches to a different set of constraints, causing the selected motor to cut thrust and tilt the faulty rotor into a neutral orientation. This activated set of constraints is shown in Table 2. The motor speed was not fully reduced to zero, as the framework did not allow the motor speed Ω_i to drop below 120 rad/s at idle. Additionally, adjustments were made to the roll angle constraint, which was forced to compute a value in the positive roll region to prevent the optimizer from converging to a local minimum, as discussed in the previous subsection.

Each motor failure case requires a unique constraint. For example, in the event of a back-left motor failure, the roll angle constraint would range between $-\phi_{max}$ and zero, since the shape of the cost function mirrors the one shown in Figure 7.

Table 1 Default Actuator and Attitude Constraints.

Constraint	Minimum	Maximum
Motor speed	120 rad/s	1000 rad/s
Tilt Elevation	-130°	20°
Tilt Azimuth	-95°	95°
Theta cmd	-15°	45°
Phi cmd	-65°	65°

Table 2 Adjustment in Case of M3 Failure.

No Fault	Failed M3
$150 \leq \Omega_{m_3} \leq 1000$ rad/s	$\Omega_{m_3} = 150$ rad/s
$-130 \leq \delta_{el3} \leq 20$ °	$\delta_{el3} = 0$ °
$-95 \leq \delta_{az3} \leq 95$ °	$\delta_{az3} = 0$ °
$-65 \leq$ Phi cmd ≤ 65 °	$0 \leq$ Phi cmd ≤ 65 °

Considering the different weights, it is essential to discuss how the weighting matrices and scaling factors from Equation 6 affect drone behavior in nominal and failure conditions. A secondary objective is minimizing power consumption during failure, making motor cost an important factor alongside primary acceleration tracking. The motor cost influences the computed optimal attitude and motor orientation; for instance, a small motor weight may result in motors not pointing directly upward during hover. A high motor weight minimizes motor use and indirectly promotes vertical alignment, efficiently utilizing the available thrust.

The attitude and servo costs have a similar purpose during hover, both costs stabilize the system back towards pointing the motors and vehicle straight and level. The ailerons were largely unused during the flight test, as the airspeed was set to zero, rendering them ineffective.

Various interactions are involved, most of which change in the event of a failure. During motor failure, the drone should avoid expending unnecessary effort in keeping a stable yaw reference. To facilitate this, the weights are adjusted as shown in Table 3. Lowering the weights on attitude enables the cascaded optimizer to explore orientations far from the initial hover configuration without incurring a large cost from attitude deviation. As a result, the first stage optimizer is less constrained and can find an attitude significantly different from ϕ_d and θ_d.

During the tests, the cost on absolute servo use was set to zero, allowing the motor orientation to be optimized by considering the commanded attitude and minimizing motor power.

Table 3 Default and Failure Weighting Matrices Used in the Stage 1 and Stage 2 Optimization runs.

	Component	**Stage 1 Optimization Run**		**Stage 2 Optimization Run**		**Effect**
		Default	**Motor Failure**	**Default**	**Motor Failure**	
W_u	Motor	$W_\Omega = 10$	$W_\Omega = 70$	$W_\Omega = 20$	$W_\Omega = 20$	Penalize deviation from u_d.
	Servos	$W_{\delta_{az}} = 0$ $W_{\delta_{el}} = 0$ $W_{\delta_{ail}} = 0.5$	$W_{\delta_{az}} = 0$ $W_{\delta_{el}} = 0$ $W_{\delta_{ail}} = 0.5$	$W_{\delta_{az}} = 0$ $W_{\delta_{el}} = 0$ $W_{\delta_{ail}} = 0.5$	$W_{\delta_{az}} = 0$ $W_{\delta_{el}} = 0$ $W_{\delta_{ail}} = 0.5$	
	Attitude	$W_\theta = 100$ $W_\phi = 100$	$W_\theta = 1$ $W_\phi = 1$	—	—	
W_v	Accelerations	$W_{ax} = 0.01$ $W_{ay} = 0.01$ $W_{az} = 0.05$ $W_{\dot{p}} = 0.1$ $W_{\dot{q}} = 0.1$ $W_{\dot{r}} = 0.1$	$W_{ax} = 0.1$ $W_{ay} = 0.1$ $W_{az} = 0.1$ $W_{\dot{p}} = 0.01$ $W_{\dot{q}} = 0.01$ $W_{\dot{r}} = 0.01$	$W_{ax} = 0.01$ $W_{ay} = 0.01$ $W_{az} = 0.05$ $W_{\dot{p}} = 0.1$ $W_{\dot{q}} = 0.1$ $W_{\dot{r}} = 0.1$	$W_{ax} = 0.01$ $W_{ay} = 0.01$ $W_{az} = 0.05$ $W_{\dot{p}} = 0.1$ $W_{\dot{q}} = 0.1$ $W_{\dot{r}} = 0.05$	Penalize acceleration residuals.
γ_u	Cost function	$\gamma_u = 3e^{-7}$	$\gamma_u = 3e^{-7}$	$\gamma_u = 3e^{-7}$	$\gamma_u = 3e^{-7}$	Scales the secondary objective in the cost function.

III. Results and Discussion

This section presents the data gathered from the flight tests. Prior to the flight test campaign, the algorithm was rigorously tested and refined within a simulation environment developed in Simulink. Once the Simulink simulation produced promising results, the code was generated from the MATLAB functions and compiled to run in real time on the UAV's onboard single-board computer. For a more detailed analysis of the hardware used in these tests, refer to [19].

A. Flight test setup

For the test flights, the system was informed of the failure through a switch activated by the pilot. The drone was flown to a pre-selected hover point, and once stable hover at the designated position was achieved, the controller was notified of the failure of motor number 3 (back-left motor). The system response and onboard data were recorded, and the failure was introduced repeatedly.

A representation of the flight test plan is shown in Figure 8, which illustrates the selected failure case of losing motor 3 (back-right). Following this, a waypoint tracking scenario was conducted to evaluate the maneuvering capabilities with the remaining three motors. During the flight test, a safety rope was used to ensure a safe environment and to secure the drone in case of unexpected behavior. The rope was continuously managed by an operator to maintain sufficient slack, ensuring it did not interfere with the flight test results.

Downloaded by Technische Universiteit Delft on January 7, 2025 | http://arc.aiaa.org | DOI: 10.2514/6.2025-0317

Downloaded by Technische Universiteit Delft on January 7, 2025 | http://arc.aiaa.org | DOI: 10.2514/6.2025-0317

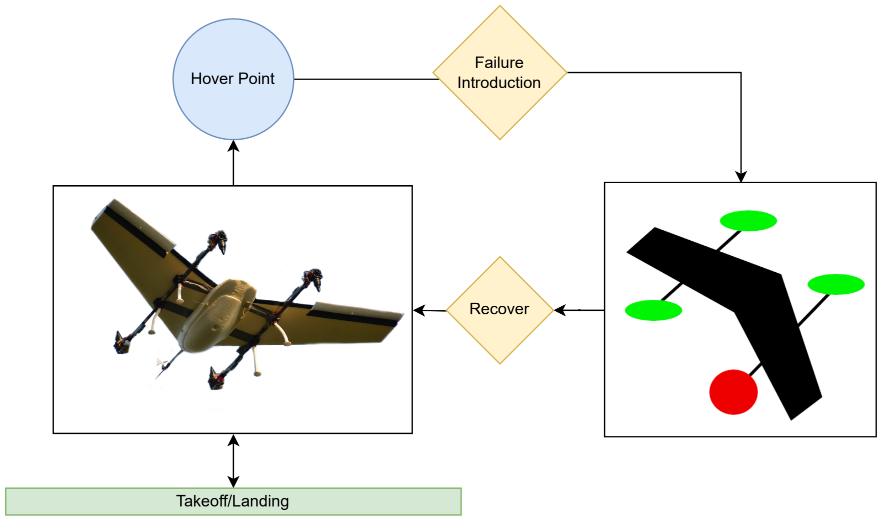

Fig. 8 Test flight representation of maneuver.

B. Flight Test Results

During the test flights, the controller successfully stabilized the vehicle after a failure was introduced on motor 3 and was able to follow waypoint changes. The resulting commands and other metrics logged by the onboard computer were analyzed and are presented in this section. A video showcasing the flight test experiment, including a synthetic 3D vehicle visualization to better analyze the forces generated by the motors, has been uploaded to the MAVLab YouTube page [*].

By analyzing the flight test data from the repeated failure and moving waypoint scenarios, the following observations can be made:

[*]Video of the experiment: https://youtu.be/7fKJa7_T6L0

Downloaded by Technische Universiteit Delft on January 7, 2025 | http://arc.aiaa.org | DOI: 10.2514/6.2025-0317

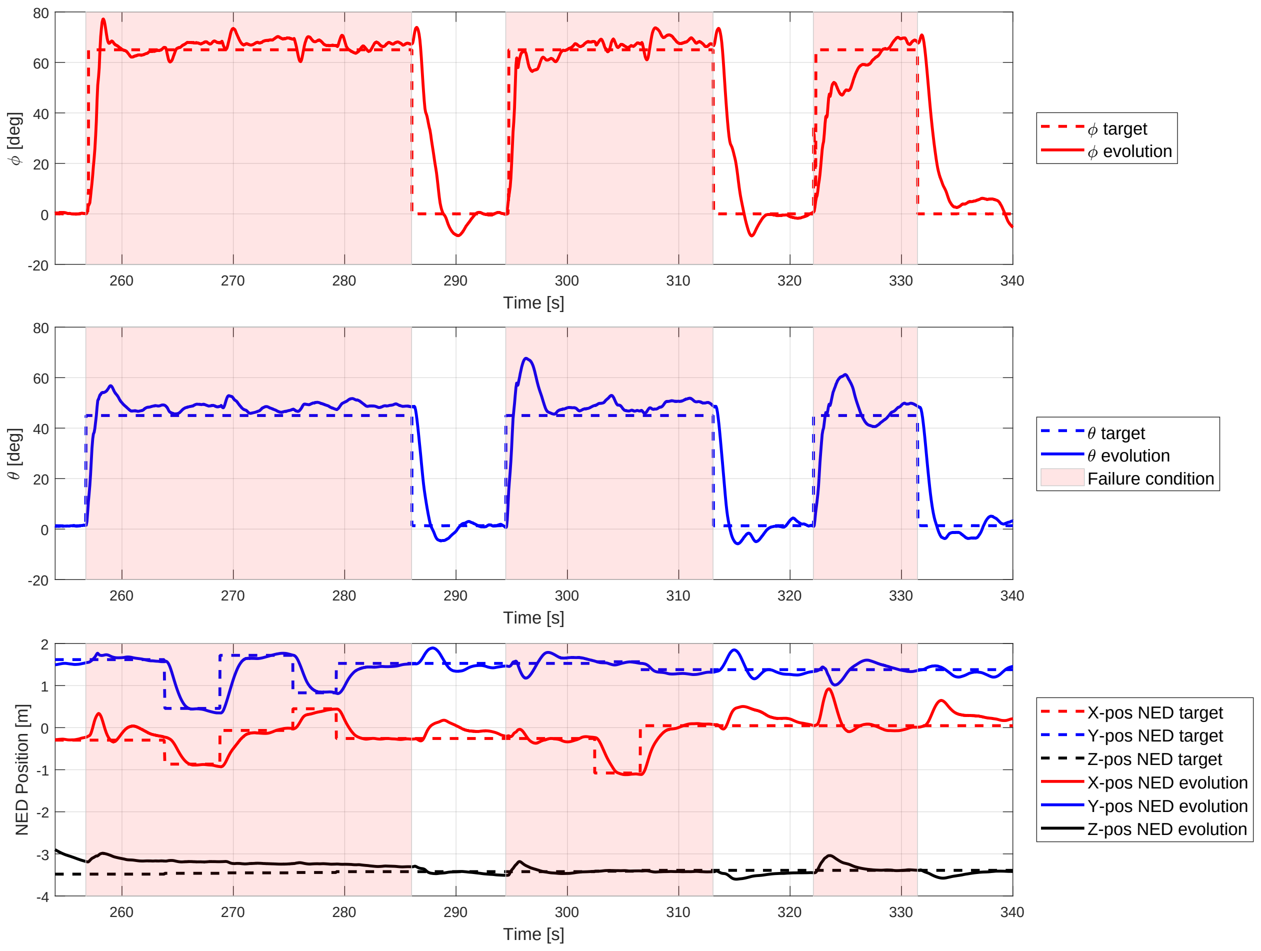

Fig. 9 Attitude and position tracking during the flight test. The red shaded areas indicate when the simulated failure was active.

- Tracking of the position and attitude setpoints remains consistent throughout the flight test, as shown in Figure 9. An initial displacement in position occurs immediately following the fault, but the error controller quickly corrects it. The maximum horizontal position error, recorded during the third failure test, was less than 1 meter. Additionally, there is a slight loss of altitude due to the fault; however, this loss consistently remains below 0.5 meters across all three failure maneuvers. It is worth noting that this loss is more evident during the second and third failure tests, as the first failure was induced before the vehicle reached the initial altitude target.
- Following the failure, the vehicle adjusts its attitude toward the global minimum identified in Figure 7. However, the attitude setpoints are constrained by enforced limits of $\phi = 65°$ and $\theta = 45°$.
- Overall, the transitions to and from the failure state are consistently well-recovered, with the vehicle smoothly switching configurations. The attitude target references generated by the first-stage optimizer remain stable and do not oscillate, indicating a well-conditioned optimization process.

Downloaded by Technische Universiteit Delft on January 7, 2025 | http://arc.aiaa.org | DOI: 10.2514/6.2025-0317

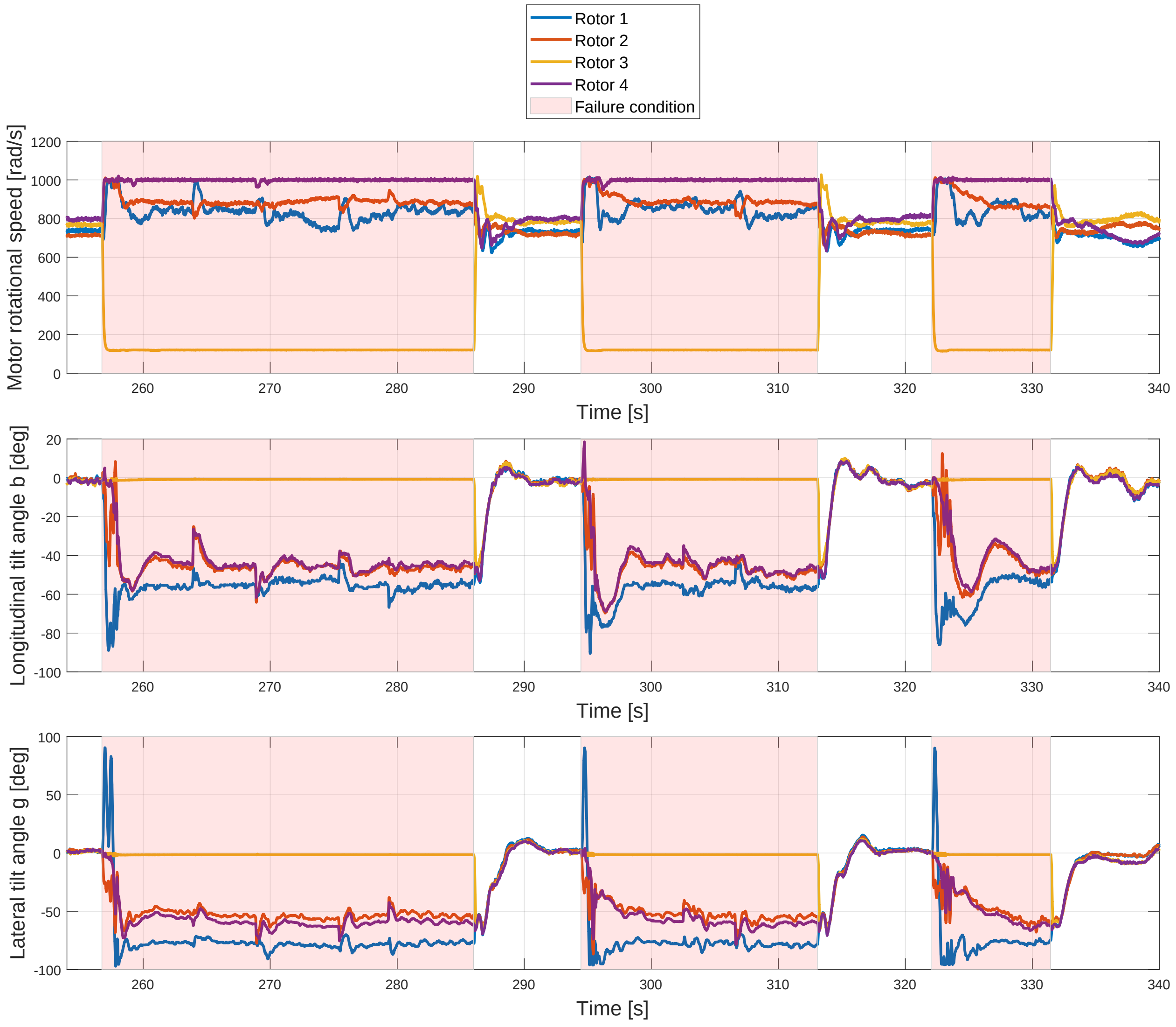

Fig. 10 Actuator evolution during the flight test.

- Looking at Figure 10, which plots the actuator evolution, the continuous failure demonstrates similar stability in the actuator solution during both failure and recovery transitions. However, slight oscillations are observed, particularly in the elevation servos immediately after the failure. A deeper analysis, including simulations, confirms that these oscillations are caused by the redistribution of roll moment generation from the saturated azimuth tilt angle of rotor 1 to the remaining unsaturated actuators.
- Another relevant observation is that rotor 4 operates more intensively than rotors 1 and 2 during the failure condition. This discrepancy can be attributed to the constrained attitude configuration the drone adopts following the failure. Despite motor 4 reaching saturation and a transient saturation of rotor 1's azimuth angle, the other actuators remain well within their operational ranges.
- The recovery from a failure state to a non-failure state shows a smoother actuator response compared to the transition from a non-failure state to a failure state. Nevertheless, the smoothness of the transition could be improved by relaxing the attitude gains, albeit at the cost of increased positioning error.

Downloaded by Technische Universiteit Delft on January 7, 2025 | http://arc.aiaa.org | DOI: 10.2514/6.2025-0317

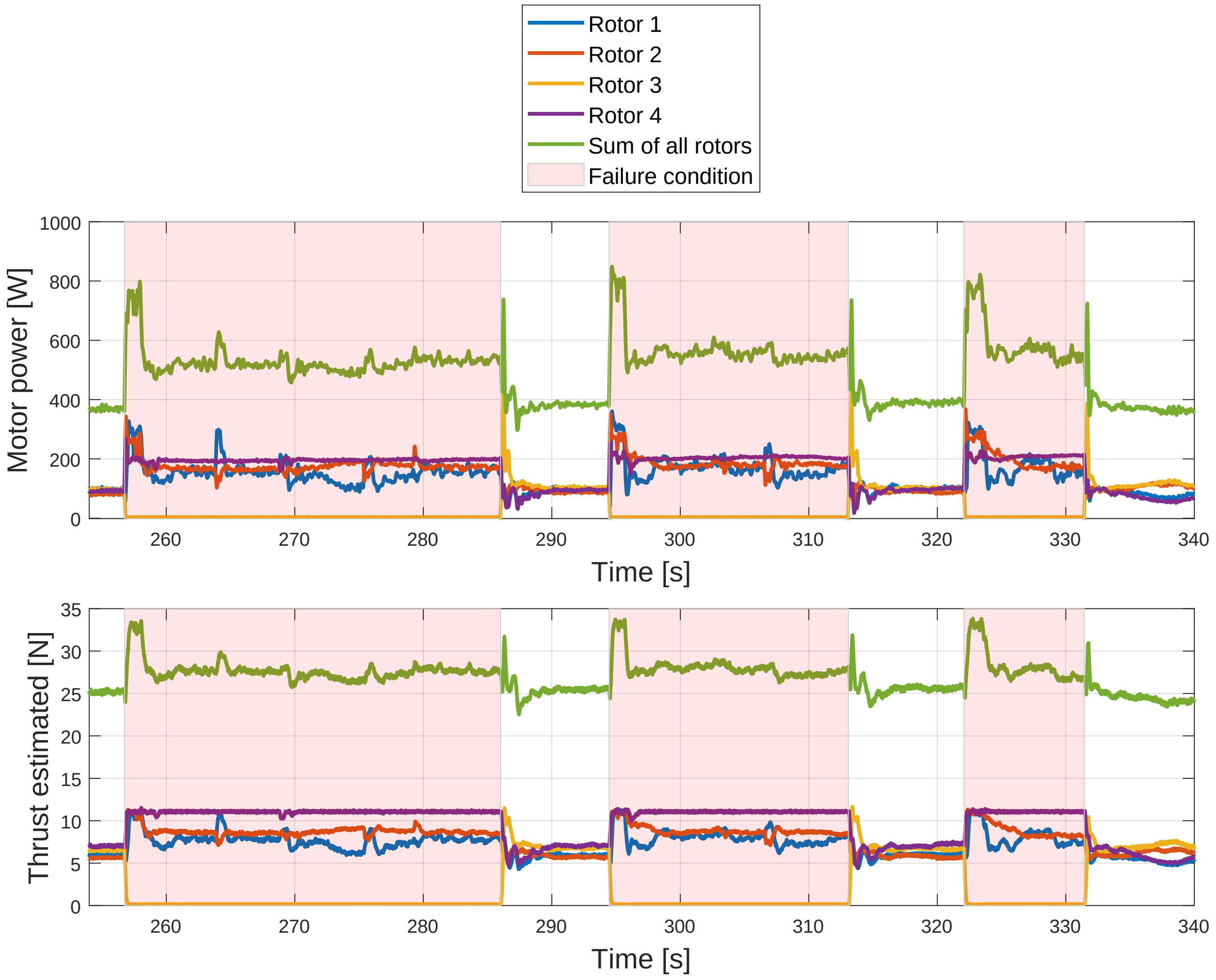

Fig. 11 Measured motor power and estimated thrust during the flight test.

- Figure 11 illustrates the measured motor power and estimated thrust during the flight test. As anticipated, during the induced failure, motor 3's power and thrust drop to near zero, while the remaining motors, particularly motor 4, compensate by increasing their power output and thrust.
- The total thrust generated during failure conditions exceeds the thrust generated during normal hover (non-failure conditions) and is also greater than the vehicle weight. This is essential to counteract the imbalance caused by the inoperative motor and to stabilize the vehicle under the constrained attitude angles. Without these constraints, the total thrust generated would likely be closer to the vehicle's weight, as the vehicle would reach the optimal attitude configuration shown in Figure 7, thereby reducing the additional demand on the functioning motors.
- During failure, the total power consumed by the remaining motors is significantly higher than during non-failure hovering because not all of the rotor thrust was oriented in the earth-vertical direction. This increase is primarily due to the system generating a higher amount of thrust, exceeding the drone's weight, to compensate for the angular acceleration balance. Furthermore, the remaining motors operate closer to their saturation points, where the efficiency of both propellers and motors decreases, further contributing to the higher power consumption.

IV. Conclusion & Recommendations

This project set out to investigate the capabilities of the proposed unified non-linear controller with SQP CA on the dual-axis tilt quad-plane under actuator faults. As the most challenging case for this vehicle, the fault of an engine in hover was chosen to be the prime failure case to be considered. Failure dependent constraints on the faulty actuator were introduced to inform the CA of the changed operating conditions.

Downloaded by Technische Universiteit Delft on January 7, 2025 | http://arc.aiaa.org | DOI: 10.2514/6.2025-0317

With the separation of attitude and actuator command optimization the previously problematic recursion loop between commanded attitude and desired angular acceleration increment was removed. Because of this, the actuator orientation is now always computed with respect to the actual vehicle attitude instead of the desired one. This is especially relevant when big attitude changes are commanded, such as in the case of an actuator failure.

Experiments show that the vehicle can successfully and repeatedly recover from a motor failure. Successful tracking of a changing reference waypoint was demonstrated with one failed motor. This paper highlights the potential of over-actuated tilt-rotor configurations in establishing new ways of recovery by utilizing the available thrust vectoring to re-orient the vehicle into a new hover configuration.

A. Limitations & Recommendations

The flight tests were conducted in a controlled indoor environment, minimizing disturbances but limiting the ability to evaluate robustness under more unpredictable conditions. While the tests demonstrated repeated recovery, a longer and more comprehensive campaign is necessary to assess recovery performance across a wider range of scenarios.

Additionally, the flight tests did not include transitions from low airspeed to high airspeed regimes. In such conditions, aerodynamic forces could be leveraged to balance the vehicle's weight, reducing the impact of motor failure. Future work could explore the algorithm's effectiveness in forward flight mode, validating its capability to handle failures during high-speed operations.

Additional limitations of the project include the following points:

- It is possible that there are still more efficient hover attitudes available under failure. The available angles in pitch and roll were limited due to possibility of Euler angle gimbal lock and due to limitations on the vehicle tilt mechanism. With the ZYX rotation order, pitch angles close to or exceeding 90° cause instability in the controller. A cost function based on quaternions or switching roll and pitch rotation order could open up new possibilities.
- The tests were limited to the failure case of motor 3, more tests should be performed on different (types and combinations of) actuator failures.
- The motor speed cost was chosen to stimulate motor tilt alignment. A better flight efficiency could be achieved if the motor cost would be based on consumed power instead of the motor speed.
- This study focused heavily on the mitigation of a motor failure, but stuck servos and other failure scenarios should be looked at in real test-flights in the future.

Acknowledgment

This work was carried out within the Unmanned Valley Project. The authors would like to thank the European Regional Development Fund who is funding the Unmanned Valley project under grant code KvW00168 for the South-Holland region.

References

[1] Kamel, M., Verling, S., Elkhatib, O., Sprecher, C., Wulkop, P., Taylor, Z., Siegwart, R., and Gilitschenski, I., "The Voliro Omniorientational Hexacopter: An Agile and Maneuverable Tiltable-Rotor Aerial Vehicle," *IEEE Robotics & Automation Magazine*, Vol. 25, No. 4, 2018, p. 34–44. https://doi.org/10.1109/mra.2018.2866758, URL http://dx.doi.org/10.1109/MRA.2018.2866758.

[2] Myeong, W., and Myung, H., "Development of a Wall-Climbing Drone Capable of Vertical Soft Landing Using a Tilt-Rotor Mechanism," *IEEE Access*, Vol. 7, 2019, pp. 4868–4879. https://doi.org/10.1109/ACCESS.2018.2889686.

[3] Papachristos, C., Alexis, K., and Tzes, A., "Model predictive hovering-translation control of an unmanned Tri-TiltRotor," *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 5425–5432. https://doi.org/10.1109/ICRA.2013.6631355.

[4] Zheng, P., Tan, X., Kocer, B. B., Yang, E., and Kovac, M., "TiltDrone: A Fully-Actuated Tilting Quadrotor Platform," *IEEE Robotics and Automation Letters*, Vol. 5, No. 4, 2020, pp. 6845–6852. https://doi.org/10.1109/LRA.2020.3010460.

[5] Bin Junaid, A., Diaz De Cerio Sanchez, A., Betancor Bosch, J., Vitzilaios, N., and Zweiri, Y., "Design and Implementation of a Dual-Axis Tilting Quadcopter," *Robotics*, Vol. 7, No. 4, 2018. https://doi.org/10.3390/robotics7040065, URL https://www.mdpi.com/2218-6581/7/4/65.

[6] Mousaei, M., Geng, J., Keipour, A., Bai, D., and Scherer, S., "Design, Modeling and Control for a Tilt-rotor VTOL UAV in the Presence of Actuator Failure," *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 4310–4317. https://doi.org/10.1109/IROS47612.2022.9981806.

[7] Mancinelli, A., Smeur, E. J., Remes, B., and Croon, G. d., “Dual-axis tilting rotor quad-plane design, simulation, flight and performance comparison with a conventional quad-plane design,” *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2022, pp. 197–206. https://doi.org/10.1109/ICUAS54217.2022.9836063.

[8] Jiang, J., Zhang, Y., and Yu, X., “Fault-tolerant control systems: A comparative study between active and passive approaches,” *Annual Reviews in Control*, Vol. 36, 2012, pp. 60–72. https://doi.org/10.1016/j.arcontrol.2012.03.005.

[9] Jain, T., Yame, J. J., and Sauter, D., “Active Fault-Tolerant Control Systems: A Behavioral System Theoretic Perspective,” *null*, 2017. https://doi.org/10.1007/978-3-319-68829-9.

[10] Abbaspour, A., Mokhtari, S., Sargolzaei, A., and Yen, K. K., “A Survey on Active Fault-Tolerant Control Systems,” *Electronics*, Vol. 9, No. 9, 2020. https://doi.org/10.3390/electronics9091513.

[11] Wang, X., and Sun, S., “Incremental fault-tolerant control for a hybrid quad-plane UAV subjected to a complete rotor loss,” *Aerospace Science and Technology*, Vol. 125, 2021, pp. 1–9. https://doi.org/10.1016/j.ast.2021.107105.

[12] Bouabdallah, S., and Siegwart, R., “Full control of a quadrotor,” *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 153–158. https://doi.org/10.1109/IROS.2007.4399042.

[13] Sun, S., Wang, X., Chu, Q., and de Visser, C., “Incremental Nonlinear Fault-Tolerant Control of a Quadrotor With Complete Loss of Two Opposing Rotors,” *IEEE Transactions on Robotics*, Vol. 37, No. 1, 2021, pp. 116–130. https://doi.org/10.1109/TRO.2020.3010626.

[14] Zhang, W., Mueller, M. W., and D’Andrea, R., “Design, modeling and control of a flying vehicle with a single moving part that can be positioned anywhere in space,” *Mechatronics*, Vol. 61, 2019, pp. 117–130. https://doi.org/10.1016/j.mechatronics.2019.06.004.

[15] Mancinelli, A., Remes, B. D. W., de Croon, G. C. H. E., and Smeur, E. J. J., “Unified incremental nonlinear controller for the transition control of a hybrid dual-axis tilting rotor quad-plane,” , 2023. https://doi.org/10.48550/arXiv.2311.09185.

[16] Wechtler, N., “Implications of Propeller-Wing Interactions on the Control of Aerodynamic-Surface-Free Tilt-Rotor Quad-Planes,” , 2024. URL http://resolver.tudelft.nl/uuid:3193131c-6b68-46a2-afe1-964a044dd6f9.

[17] Schittkowski, K., “On the convergence of a sequential quadratic programming method with an augmented lagrangian line search function,” *Mathematische Operationsforschung und Statistik. Series Optimization*, Vol. 14, No. 2, 1983, pp. 197–216. https://doi.org/10.1080/02331938308842847.

[18] Nocedal, J., and Wright, S. J., *Sequential Quadratic Programming*, Springer, New York, NY, 2006, pp. 529–562. https://doi.org/10.1007/978-0-387-40065-5_18.

[19] Mancinelli, A., van der Horst, E., Remes, B., and Smeur, E., “Autopilot framework with INDI RPM control, real-time actuator feedback, and stability control on companion computer through MATLAB generated functions,” *14th annual international micro air vehicle conference and competition*, edited by D. Moormann, 2023, pp. 109–116. URL https://2023.imavs.org/, 14th anual International Micro Air Vehicle Conference and Competition, IMAV 2023 ; Conference date: 11-09-2023 Through 15-09-2023.

Downloaded by Technische Universiteit Delft on January 7, 2025 | http://arc.aiaa.org | DOI: 10.2514/6.2025-0317