Extended incremental non-linear control allocation (XINCA) for quadplanes

Karssies, H. J.; De Wagter, C.

DOI
10.1177/17568293211070825

Publication date
2022

Document Version
Final published version

Published in
International Journal of Micro Air Vehicles

International Journal of Micro Air Vehicles

Original Research

International Journal of Micro Air Vehicles
Volume 14: 1–11

DOI: 10.1177/17568293211070825
journals.sagepub.com/home/mav
SAGE

Extended incremental non-linear control allocation (XINCA) for quadplanes

H.J. Karssies[1] and C. De Wagter[1]

Abstract

Hybrid UAVs have gained a lot of interest for their combined vertical take-off & landing (VTOL) and efficient forward flight capabilities. But their control is facing challenges in over-actuation and conflicting requirements depending on the flight phase which can easily lead to actuator saturation. Incremental Non-linear Control Allocation (INCA) has been proposed to solve the platform's control allocation problem in the case of saturation or over-actuation by minimizing a set of objective functions. This work demonstrates INCA on quadplanes, an in-plane combination between a quadrotor and a conventional fixed-wing, and proposes an extension to control the outer loop. The novel controller is called Extended INCA (XINCA) and adds the wing orientation as a force-generating actuator in the outerloop control optimization. This leads to a single controller for all flight phases that avoids placing the wing at negative angles of attack and minimizes the load on hover motors. XINCA has low dependence on accurate vehicle models and requires only several optimization parameters. Flight simulations and experimental flights are performed to demonstrate the performance.

Keywords

XINCA, unmanned Air Vehicle, control allocation, active set method

Date received 28 November 2021; accepted 16 December 2021

Introduction

Unmanned Aerial Vehicles or UAVs have gained a tremendous amount of popularity. Not only have they proven to be valuable research platforms and entertaining toys, but they have also found many other applications in fields like defense,[1] surveillance,[2] medical assistance,[3,4] transportation of both goods and humans,[5] agriculture,[6] inspection,[7] mapping,[8] and many others.

Some challenges that are often faced in UAV design are endurance, reliability, versatility, and affordability. Existing solutions often perform well on some but not all of these aspects. Fixed-wing aircraft have great endurance thanks to their wing-induced lift.[9–11] Rotorcraft on the other hand, like designs by Luukkonen,[12] Zhiqiang et al.[13] and Smeur et al.,[14] are much more versatile since they can hover, take off and land vertically. They are also inexpensive to produce, mechanically simple and their control has been well solved. The powered generation of lift however severely limits their endurance, and designs like the conventional quadcopter typically have multiple single points of failure. It is therefore that many researchers have proposed hybrid platforms, that aim to combine the best of both.[15]

Some examples of hybrid platforms include tilt rotor/wing UAVs, tail sitters, transformable UAVs, and quadplanes. Tilt-rotor/wing UAVs UAVs[16,17] mechanically change the orientation of their propulsion units to either generate lift during vertical take-off and landing or horizontal thrust while flying horizontally with wing-induced lift. Similarly, tail sitters[18,19] change the orientation of the entire vehicle when transitioning from vertical take-off and landing orientation to horizontal flight. This reduces the mechanical complexity of the system, resulting in a lighter, and cheaper platform, albeit at the cost of sensitivity to wind gusts. A different class of hybrid UAVs consists of transformable UAVs[20] that change the configuration of the entire vehicle.

Lastly, a common class of hybrid UAVs is formed by quadplanes[21–25] which are also referred to as dual-systems (See example in Figure 1). The quadplane has a static

[1]MAVLab, TUDelft, the Netherlands

Corresponding author:
Christophe De Wagter, Micro Air Vehicle Lab, Delft University of Technology, Kluyverweg 1, 2629HS Delft, the Netherlands.
Email: c.dewagter@tudelft.nl

configuration with the hover rotors and the fixed wings in the same plane. A horizontal pusher motor provides propulsion during horizontal flight. Despite the added weight of flight phase-specific actuators, its mechanical simplicity makes this versatile and enduring vehicle a promising research platform.

The control of quadplanes poses several challenges. This includes dealing with large flight envelopes, over-actuation, its non-linear nature, its sensitivity to wind gusts in hover, and the conflicting requirements in hover and forward flight.

Over-actuation and conflicting requirements are often dealt with by using different actuators and switching controllers during specific flight phases, and only briefly combining them during a transition phase between hovering and horizontal flight.[21–24]

Incremental Non-linear Control Allocation (INCA) has been proposed to solve over-actuation. The theoretical study by Stolk[26] applied it to the Lockheed Martin Innovative Control Effector aircraft. Smeur et al.[27] applied it to quadcopters and showed its promising properties in the case of actuator saturation.

This work extends the theory to the control of quadplanes proposes a single formulation through all flight phases that solve the over-actuation, has good saturation handling properties, avoids placing the main wing under undesirable negative angles of attack, and does not require switching between flight phases. The theory is tested in simulation and through actual test flights.

The quadplane used for this research and its control challenges are described in the Section The TU Delft Quadplane. Incremental Non-linear Control Allocation (INCA) is discussed in the Section about INCA, and its optimization methods in the Section INCA Optimization. An extension of this control method, called XINCA, is presented in the Section about XINCA. The implementation of the INCA and XINCA controllers on the TUDelft Quadplane is shown in the Section Implementation, and the Sections about Flight Simulations and Flight Experiments respectively present results from simulations and test flights performed using this novel control method. Lastly, the Section Conclusions and Recommendations summarizes the conclusions.

Figure 1. The TUDelft Quadplane in the CyberZoo.

The TU Delft Quadplane

The quadplane is a hybrid of a fixed-wing aircraft and a quadcopter. A schematic representation of the *TUDelft Quadplane* used in this research, is shown in Figure 2. It shows the quadplane's nine actuators: four upward-facing rotors that could be considered as the *quadcopter actuator set*, and four control surfaces, and a tail rotor that could be considered the *fixed-wing actuator set*. Having actuator sets that can operate simultaneously, quadplanes are considered *over-actuated*.

UAV controllers often use cascaded outer and inner loops shown in Figure 3. The outer loop, also called the position or guidance loop, controls the position error and outputs a reference attitude. The inner loop or attitude loop controls actual attitude and uses that to allocate control to suitable actuators. This allocation is quite straightforward when the vehicle is not over-actuated or when only a single actuator set is used.

Quadplanes could however fly more efficiently when continuously assessing each actuator's suitability to satisfy a certain control demand. This assessment should take into account each actuator's effectiveness based on the system's states, but could also penalize large deviations from preferred actuator positions. Such an optimization problem is known as a control allocation problem. The advantages are that first, it can minimize the control effort of a UAV, potentially resulting in more efficient flight and enhanced flight endurance. The other advantage is that when certain actuators are saturating, it can allocate control to other actuators to still satisfy a given control demand, resulting in safer and more reliable flight. The control allocation method used in this research is called Incremental Non-linear Control Allocation or INCA, which solves the inner loop's control allocation optimization problem and is presented in the Section INCA.

Another challenge in controlling quadplanes is caused by the fundamentally different outer loop dynamics of the quadplane during different flight phases. When flying as a quadcopter, for instance, a change in pitch angle causes the quadplane to accelerate in a longitudinal direction. When flying as a fixed-wing aircraft, however, a change in pitch will cause the quadplane to either climb or descent. Furthermore, the quadplane is over-actuated in

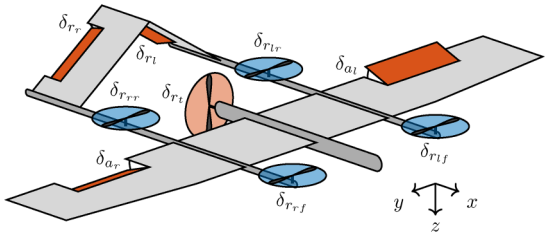


Figure 2. Overview of the nine quadplane actuators

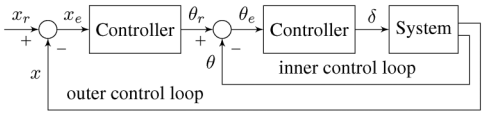


Figure 3. Simplified schematic UAV controller diagram (x = position, θ = attitude, δ = system input)

its outer loop as well as its inner loop, since it can control a positive forward acceleration during hovering with both its pitch angle and pusher rotor. The latter is often preferable since negative pitching maneuvers might introduce an undesirable negative wing-induced lift, which quickly can lead to saturation in the thrust of the hover motors. A positive backward acceleration however is only achievable by pitching the quadplane backward. To address the challenges named above, an extension of the INCA controller is presented in the Section XINCA, which performs an outer loop optimization similar to the INCA inner loop optimization. This method is called Extended Incremental Non-linear Control Allocation, or XINCA.

INCA

The architecture of INCA augments a method called Non-linear Dynamic Inversion, or NDI. NDI measures a vehicle's states and uses an accurate model to predict angular and linear accelerations as a result of these states. Their difference with the vehicle's desired accelerations is then used to calculate appropriate control inputs using reliable actuator models. A successful example of an implementation of NDI is the work by Horn.[28]

However effective, NDI highly relies on detailed and accurate models of the vehicle it controls. A variation on this approach provides a solution to this problem and is called Incremental Non-linear Dynamic Inversion, or INDI, and was demonstrated in flight by Smeur et al.[27,14] While it still relies on an actuator model, instead of using a vehicle model to predict its angular and linear accelerations as a result of its states, it uses inertial measurement data to observe these accelerations. And the control effectiveness model does not need to be as accurate, since the controller will compensate for any unexpected effects of the actuators by incrementing.

Both NDI and INDI invert actuator effectiveness models to calculate appropriate actuator commands. When dealing with over-actuated UAVs however, it is mathematically challenging to derive appropriate actuator commands by simply inverting these actuator effectiveness models, since any calculated actuator command solution is no longer singular, and there exists an infinite number of solutions. INCA deals with this by expressing the control allocation problem as an optimization problem, that needs to be solved by minimizing a cost function.

A schematic representation of INCA is shown in Figure 4. Like an INDI controller, INCA uses the difference between desired accelerations and inertial measurements to determine an incremental control demand, also known as the virtual input to the INCA optimization. The optimization scheme then calculates an optimal actuator increment to satisfy the control demand. Note that while the rotor effectiveness is relatively constant around hover, the effectiveness of the aerodynamic surfaces is proportional to the square of the true airspeed. The optimization method is presented in the next section.

INCA Optimization

Let $\mathbf{H}$ be a matrix containing the linearized effectiveness of all actuators, and τ_c the control demand that will be used as virtual input to the INCA optimization. An unconstrained control command increment $\Delta\delta$ should then always satisfy the following equation:

$$\mathbf{H}\Delta\delta = \tau_c \tag{1}$$

When this increment is constrained by actuator limits, an error between the control demand and the achieved control might occur, but should still be minimized. Also minimizing control effort, i.e., the difference between actual actuator increments $\Delta\delta$ and preferred actuator increments $\Delta\delta_p$, yields:

$$\Delta\delta min \|\gamma \mathbf{W}_\tau(\mathbf{H}\Delta\delta - \tau_c)\|_2 + \|\mathbf{W}_\delta(\Delta\delta_p - \Delta\delta)\|_2 \tag{2a}$$

$$\text{subject to } \Delta\delta_{\min} \le \Delta\delta \le \Delta\delta_{\max} \text{ and } \dot{\delta} \le \dot{\delta}_{\max} \tag{2b}$$

where $\mathbf{W}_\tau$ and $\mathbf{W}_\delta$ are weighing matrices to prioritize selected control demands and actuators over others, and γ is a constant that prioritizes one sub-objective over the other. This type of objective function is called a *Quadratic Program* and can include as many separate sub-objectives as needed. Quadratic Programming is often used for Control Allocation problems. Härkegøard[29] presents proof that it can provide automatic redistribution of control in case of actuator saturation. Stolk[26] and Smeur et al.[14] apply it on a modern fighter jet and a quadcopter UAV respectively. The objective function is often rewritten to a standardized quadratic form, which many solvers can easily work with:

$$\Delta\delta min \Delta\delta^T \mathbf{Q}\Delta\delta + c^T\Delta\delta \tag{3a}$$

$$\text{subject to } \mathbf{A}\Delta\delta \le \mathrm{b} \tag{3b}$$

where $\mathbf{Q} = \mathbf{F}^T\mathbf{F}$, $c = 2\mathbf{F}^T g$,

$$\mathbf{F} = \begin{pmatrix} \gamma\mathbf{W}_\tau\mathbf{H} \\ \mathbf{W}_\delta \end{pmatrix},\ g = \begin{pmatrix} \gamma\mathbf{W}_\tau\tau_c \\ \mathbf{W}_\delta\Delta\delta_p \end{pmatrix},$$

$$\mathbf{A} = \begin{pmatrix} \mathbf{I} \\ -\mathbf{I} \end{pmatrix} \text{and}$$

$$b = \begin{pmatrix} \min(\delta_{\max} - \delta_0,\ \dot{\delta}_{\max}\Delta\mathrm{t}) \\ -\max(\delta_{\min} - \delta_0,\ -\dot{\delta}_{\max}\Delta\mathrm{t}) \end{pmatrix}$$

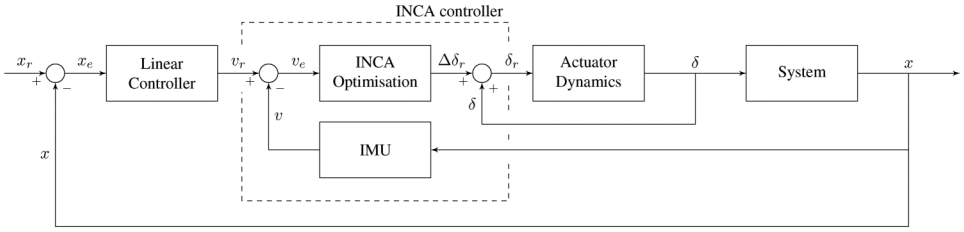


Figure 4. A schematic representation of an INCA controller (x = state vector, v = virtual input, δ = control input vector)

When the inequality constraints are treated as equality constraints ($\mathbf{A} = b$ instead of $\mathbf{A} \leq b$), the solution to the optimization problem is given by the following linear system, as long as $\mathbf{Q}$ is a positive definite matrix[30] and $\mathbf{A}$ has full row rank[31]:

$$\begin{bmatrix} \mathbf{Q} & \mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta\delta \\ \lambda \end{bmatrix} = \begin{bmatrix} -c \\ b \end{bmatrix} \tag{4}$$

where λ is known as the vector containing the Lagrange multipliers. Explicit solutions for both the optimal input increment $\Delta\delta$ and Lagrange multipliers λ can be derived algebraically as:

$$\Delta\delta = -\mathbf{Q}^{-1}(\mathbf{A}^T\lambda + c) \tag{5a}$$

$$\text{where } \lambda = -(\mathbf{A}\mathbf{Q}^{-1}\mathbf{A}^{\mathrm{T}})^{-1}(\mathbf{A}\mathbf{Q}^{-1}\mathrm{c} + \mathrm{b}) \tag{5b}$$

The values of the Lagrange multipliers are used to determine what constraints to release during the optimization process, and whether or not the solution has already reached its optimum.

Since the calculation of UAV control demands typically needs to be performed several hundred times per second, the optimization used in an INCA controller needs to be as efficient as possible. Based on control allocation research performed by Stolk[26] and Smeur et al.,[14] the optimization method selected for this research is the *Active Set Method*. This method requires similar amounts of computing power as e.g. the Redistributed Pseudo-Inverse method and the Fixed-Point algorithm, yet yields more accurate solutions. It also scales efficiently with larger amounts of actuators, which is validated in the Section Flight Experiments.

A detailed description of the Active Set Method[32] is summarized in Figure 6 and illustrated in Figure 5 for a hypothetical optimization problem with a constrained two-dimensional input.

Choosing a suitable starting point for the Active Set Method has a significant effect on the solver's efficiency. In control allocation, each solution is likely to be in the neighborhood of the solution of the previous time step. The Active Set Method has a relatively low computational cost,[32] and since the solution progresses towards the final solution each time step, even when cut off before reaching the optimum to save computational time, the solution will be close to optimal.

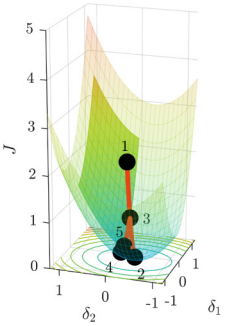
XINCA

To simplify the outer loop control, hybrid UAVs like quadplanes are often controlled in either a vertical, horizontal or short transitional flight mode. Separating these flight modes however often results in sub-optimal flight control, not always using the most effective or efficient actuators nor making use of redundant actuators in case of actuator saturation. Worse even, actuator sets can even counteract each other. In hover, for instance, a downward pitch to move forward with a non-zero wind will cause a negative lift of the wing counteracting and possibly saturating the hover motors.

To solve this, a new control scheme is proposed, called Extended Incremental Non-linear Control Allocation (XINCA). It is an extension of INCA which takes the specific quadplane outer-loop dynamics into account as additional constraints. As shown in Figure 7, a linear controller on the position errors selects the desired linear reference accelerations. The error between these reference accelerations and measured accelerations then enters the XINCA optimization block. Like the INCA optimization, the XINCA optimization possesses several constrained actuators to achieve this control demand. These XINCA actuators do not only include physical actuators of the platform, but also some of its attitude angles and its vertical thrust command. In this research, the XINCA output includes the tail pusher rotor command, the vertical thrust command, and the vehicle's pitch and roll commands. The tail rotor command is directly fed to the tail rotor itself. The thrust command and two attitude angle commands serve as input for the inner loop's INCA optimization. The XINCA optimization is also performed with the Active Set Method. Since the effectiveness of the XINCA actuators is dependent on the aircraft's states, it needs to be re-assessed at every iteration. But the resulting controller does not need to differ anymore for any of the flight regimes or flight modes.

Implementation

XINCA is implemented in the open-source drone hardware and software platform Paparazzi UAV.[33] The quadplane itself makes use of a *Lisa/MX autopilot* board. Since this



Point	Description	Active constraints
1	Starting point	None
2	Unconstrained optimum	None
3	Best feasible solution in direction of optimum	$\delta_2 \geq 0$
4	Optimum with active set as equality constraints	$\delta_2 \geq 0$
5	Best feasible solution in direction of optimum and final solution	$\delta_1 \geq 0, \delta_2 \geq 0$

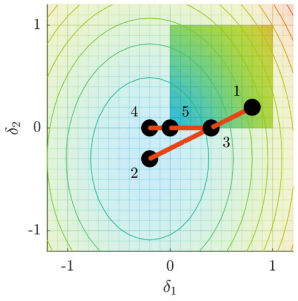


Figure 5. The Active Set Method performed on a hypothetical cost function J with a two-dimensional input space (Constraints: $0 \leq \delta_1 \leq 1$ and $0 \leq \delta_2 \leq 1$, starting point: $(\delta_1, \delta_2) = (0.8, 0.2)$

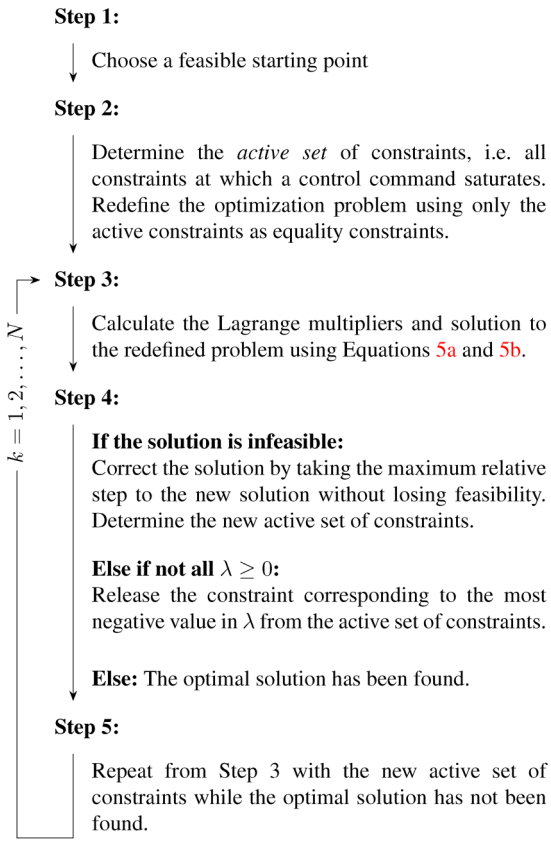


Figure 6. Active set method

flight controller can control a maximum of eight actuators, the two ailerons share one control command, making them respond symmetrically yet in the opposite direction while reducing the computational cost of the INCA optimization.

The INCA module in Paparazzi UAV is based on an INDI module from Smeur et al.[27] and used by Smeur et al.[14] It is extended to include seven of the quadplane's eight actuators, and scale the effectiveness of the three actuators that are aerodynamic surfaces, i.e. two separate ruddervators and the combined ailerons. The control is calculated as follows:

$$\begin{bmatrix} \Delta \dot{p} & \Delta \dot{q} & \Delta \dot{r} & \Delta \ddot{z} \end{bmatrix}^T = \mathbf{H} \Delta \delta \tag{6}$$

$$\text{where } \delta = \begin{bmatrix} \delta_{r_{lf}} & \delta_{r_{rf}} & \delta_{r_{rr}} & \delta_{r_{lr}} & \delta_a & \delta_{r_l} & \delta_{r_r} \end{bmatrix}^T$$

The control effectiveness matrix $\mathbf{H}$ is separated into two parts. $\mathbf{H_1}$ accounts for increments in actuator inputs, and $\mathbf{H_2}$ accounts for counter-torque effects during the spin-up of the upwards facing rotors, such that:

$$\mathbf{H} = \mathbf{H_1} + \Delta t \mathbf{H_2} \tag{7}$$

The actuator effectiveness matrix units are either rads^{-2} PPRZ^{-1} or ms^{-2} PPRZ^{-1}, where PPRZ stands for Paparazzi actuator units ranging from -9600 for bi-directional or 0 for mono-directional actuators to 9600. To illustrate INCA's ability to handle inaccurate actuator models because of its incremental nature, only a simple approximation of the actuator effectiveness is used to control the quadplane. This approximation is based on theoretical calculations using estimations of the inertial properties and their actuator positions. The resulting actuator effectiveness matrices are:

$$\mathbf{H_1} = 10^{-3} \cdot \begin{bmatrix} \delta_{r_{lf}} & \delta_{r_{rf}} & \delta_{r_{rr}} & \delta_{r_{lr}} & \delta_a & \delta_{r_l} & \delta_{r_r} \\ 11 & -11 & -11 & 11 & 0.15u^2 & 0 & 0 \\ 9 & 9 & -9 & -9 & 0 & 0.11u^2 & -0.11u^2 \\ -0.6 & 0.6 & -0.6 & 0.6 & 0 & -0.03u^2 & -0.03u^2 \\ -0.8 & -0.8 & -0.8 & -0.8 & 0 & 0 & 0 \end{bmatrix} \begin{matrix} \\ \Delta \dot{p} \\ \Delta \dot{q} \\ \Delta \dot{r} \\ \Delta \ddot{z} \end{matrix} \tag{8a}$$

$$\mathbf{H_2} = 10^{-3} \cdot \begin{bmatrix} \delta_{r_{lf}} & \delta_{r_{rf}} & \delta_{r_{rr}} & \delta_{r_{lr}} & \delta_a & \delta_{r_l} & \delta_{r_r} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -55 & 55 & -55 & 55 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{matrix} \\ \Delta \dot{p} \\ \Delta \dot{q} \\ \Delta \dot{r} \\ \Delta \ddot{z} \end{matrix} \tag{8b}$$

where u represents the true airspeed over the aerodynamic control surfaces, which in this work is simplified by the substitution of the forward body velocity since tests are

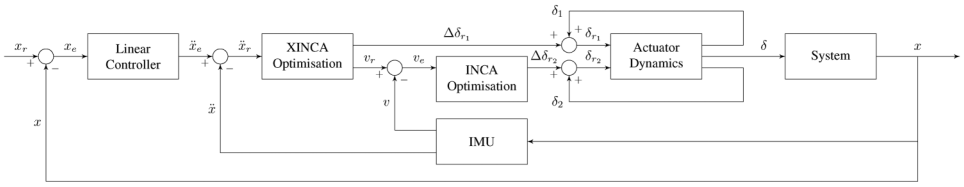


Figure 7. A schematic representation of a XINCA controller (x = state vector, v = virtual input, δ = control input vector)

performed in an indoor environment without wind. Negative values are replaced by zero.

Since the actuators do not provide any form of feedback, an estimation of the current actuator deflection is performed each time step. This is done by a first-order approximation with a time constant τ:

$$H_{act} = K\tau s + 1 \tag{9}$$

Each actuator position is estimated as:

$$\delta_{est} = \delta_{prev} + \alpha(\delta - \delta_{prev}) \tag{10}$$

$$\text{where } \alpha = 1 - \mathrm{e}^{-\tau \Delta \mathrm{t}}$$

The used time constant used for the four upwards facing rotors is 29 s^{-1}. For the control surfaces, an estimation of 100 s^{-1} is used. The optimization parameters in Equations 2a and 2b are chosen as:

$$\mathbf{W}_\tau = diag\begin{bmatrix} 100 & 100 & 1 & 1000 \end{bmatrix}$$

$$\mathbf{W}_\delta = diag\begin{bmatrix} 10 & 10 & 10 & 10 & 1 & 1 & 1 \end{bmatrix}$$

$$\gamma = 10000$$

$$\delta_p = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

These values are selected to prioritize pitch and roll and especially thrust over yaw commands while $\mathbf{W}_\delta$ penalizes the use of rotors over aerodynamic surfaces as the latter use less energy. Finally, γ prioritizes achieving the control demand over minimizing control effort. The actuator limits are set to either 0 and 9600 for rotors or -9600 and 9600 for control surfaces, again expressed in PPRZ units.

The XINCA controller works similarly to the INCA controller and is based on an existing outer loop INDI module by Smeur et al.[34,35] This existing module uses the vertical thrust vector to control the position, by either changing the thrust itself or changing its orientation by pitch or roll increments. It is augmented by including a tail rotor command as the fourth actuator. The control is then calculated as follows:

$$\begin{bmatrix} \Delta\ddot{x} & \Delta\ddot{y} & \Delta\ddot{z} \end{bmatrix}^T = \mathbf{H}\begin{bmatrix} v_r & \delta_{r_t} \end{bmatrix}^T \tag{12}$$

$$\text{where } \mathrm{v_r} = \begin{bmatrix} \Delta\theta & \Delta\phi & \Delta\mathrm{T} \end{bmatrix}^{\mathrm{T}}$$

The actuator's effectiveness depends on the current state. At low speeds aerodynamics do not play a great role yet, so it could be calculated as follows:

$$\mathbf{H} = \begin{bmatrix} \Delta\theta & \Delta\phi & \Delta T & \delta_{r_t} \\ -c\theta c\phi T & -s\theta s\phi T & s\theta c\phi & c\theta \\ 0 & -c\phi T & -s\phi & 0 \\ s\theta c\phi T & -c\theta s\phi T & c\theta c\phi & -s\theta \end{bmatrix} \begin{matrix} \Delta\ddot{x} \\ \Delta\ddot{y} \\ \Delta\ddot{z} \end{matrix} \tag{13}$$

where s and c represent the sine and cosine functions respectively, and T represents the vertical specific-force vector, which is estimated by taking the vertical body acceleration and subtracting the gravitational acceleration:

$$T = \ddot{z} - g \tag{14}$$

When flying at higher velocities, however, the quadplane will start to behave more like a fixed-wing aircraft. The controller should start using the wings to generate lift instead of the hover motors, and as a positive angle of pitch leads to a positive angle of attack on the main wing, one term is added to the actuator effectiveness matrix as follows:

$$\mathbf{H} = \begin{bmatrix} \Delta\theta & \Delta\phi & \Delta T & \delta_{r_t} \\ -c\theta c\phi T & -s\theta s\phi T & s\theta c\phi & c\theta \\ 0 & -c\phi T & -s\phi & 0 \\ c\phi\left(s\theta T - C_{L_\alpha}\rho u^2/S2m\right) & -c\theta s\phi T & c\theta c\phi & -s\theta \end{bmatrix} \begin{matrix} \Delta\ddot{x} \\ \Delta\ddot{y} \\ \Delta\ddot{z} \end{matrix} \tag{15}$$

where C_{L_α} is the change in lift per change in angle of attack, ρ is the air density, u is the true airspeed, S is the wing surface area, and m is the platform's mass. In Equations 2a and 2b the XINCA optimization parameters are chosen as:

$$\mathbf{W}_\tau = diag\begin{bmatrix} 10 & 10 & 1 \end{bmatrix}$$

$$\mathbf{W}_\delta = diag\begin{bmatrix} 10 & 10 & 10 & 1 & 0 & 0 & 1 \end{bmatrix}$$

$$\gamma = 10000$$

$$\delta_p = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T$$

$\mathbf{W}_\tau$ prioritizes pitch and roll over thrust demands since an unstable attitude can be more dangerous than a controlled descent. Moreover, the pitch is used in lift generation. $\mathbf{W}_\delta$ penalizes the use of pitch and roll and especially thrust commands compared to using the tail rotor, and γ prioritizes achieving the control demand over minimizing the control effort. The maximum pitch and roll angles are set to 10°, the vertical thrust limits to -9.0 and 9.0 ms^{-2}, and the tail rotor's limits to 0 and 9600 PPRZ units. To prevent the tail rotor from hitting the ground, it is completely shut off for altitudes below 0.5 m by setting its effectiveness to zero.

Flight Simulations

To demonstrate XINCA's performance, several simulations are performed. These are executed within the Paparazzi UAV software while using the actual code that will also fly onboard the quadplane.

Figure 8 shows a simulation of a quadplane taking off (green) and landing (red). The top plot shows the control demand the INCA controller aims to achieve. The middle plot shows the resulting actuator commands expressed in PWM pulse length. The bottom plot shows the height profile of the flight.

To assess how well the INCA controller handles actuator saturation, a second simulation is performed with an artificial upper actuator limit slightly higher than the nominal throttle level needed for hovering. The result can be seen in Figure 9, which shows that saturation occurs during takeoff. The INCA controller achieves stable flight since its pitch and roll commands are prioritized above its thrust and especially yaw commands.

In the third simulation, the UAV moves forward and backward. Figure 10 shows the control demand and position in the x-direction. The green and red areas show where the tail rotor is being activated by the XINCA controller for acceleration and reducing backward speed respectively. The tail rotor is first activated to accelerate forward. The UAV then uses pitch increments to brake and accelerate backward, after which it activates the tail rotor again twice to brake and move forward again. Finally, it slows down using pitch increments and lands.

Two last simulations are performed to illustrate the benefits of using XINCA over conventional outer loop control methods. Both simulate forward flight of the quadplane and compare a traditional INDI outer-loop controller[34,35] with the novel XINCA controller. Using the main wing's aerodynamic properties in combination with the quadplane's pitch angle and forward velocity, an estimation is made of the wing-induced lift force. Figure 11 shows the actuator commands, pitch angle, and lift force for both simulations. The most important difference can be seen in the pitch angles. Where the INDI controller pitches forward to achieve forward acceleration, the XINCA controller minimizes this negative pitch by using its tail rotor. This difference is reflected in the lift force estimations shown in Figure 12, where the XINCA controller manages to avoid the negative lift caused by pitching forward. At higher wind speeds, the negative lift can become very significant and result in hover motor saturation and possibly dangerous loss of altitude or even loss of attitude control.

Flight Experiments

The XINCA controller was tested in real flight tests of the TUDelft quadplane shown in Figure 1. The flight tests are performed in the *CyberZoo*, which is equipped with an

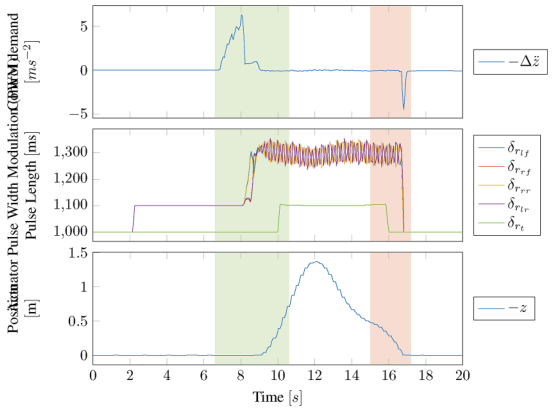


Figure 8. Simulation of vertical quadplane takeoff and landing using XINCA and INCA with five actuators

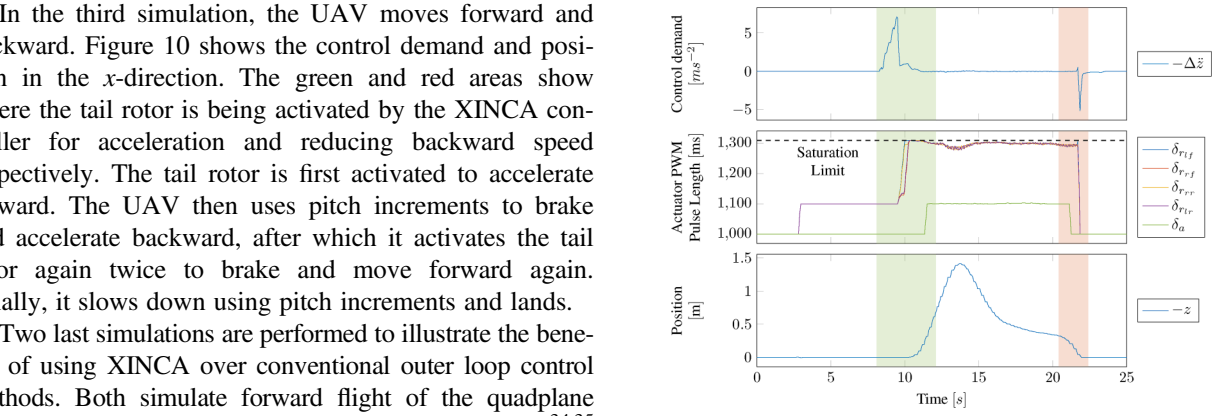


Figure 9. Simulation of a vertical quadplane takeoff and landing with actuator saturation occurring at an actuator Pulse Width Modulation (PWM) pulse length of 1310 ms using XINCA and INCA with five actuators

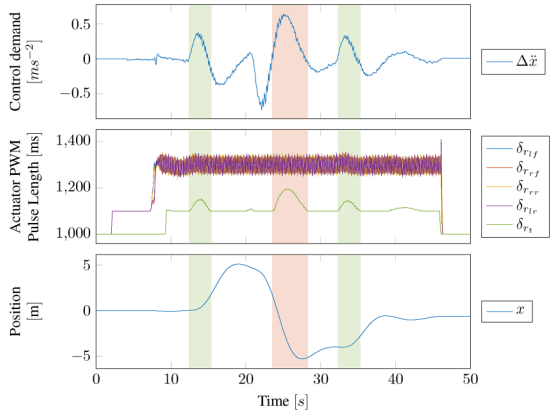


Figure 10. Simulation of forwards and backward flight using XINCA with both pitch increments and tail rotor inputs and INCA with five actuators

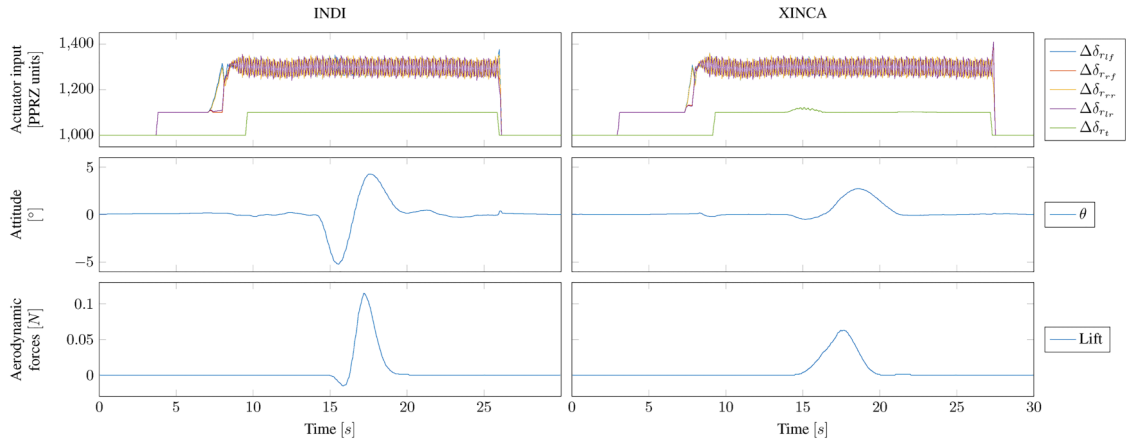


Figure 11. Flight data comparison of forward flight simulation with INDI and XINCA

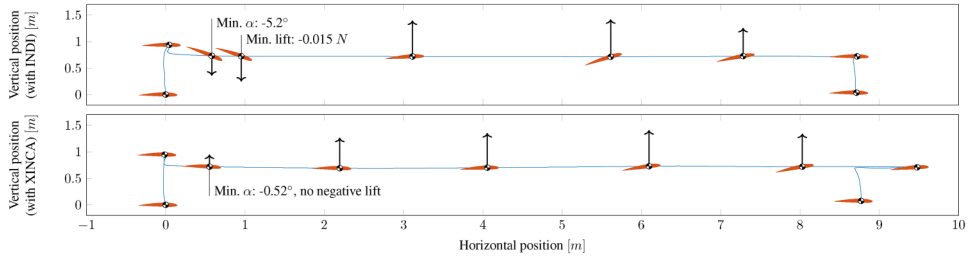


Figure 12. Flight profile comparison of forward flight simulation with INDI and XINCA showing wing-induced lift estimations. Note that illustrated angles of attack are magnified and force vectors are scaled for readability

optical position tracking system for precise vehicle positioning.

During initial attempts to fly the Quadplane with both the INCA and XINCA optimizations, the 32-bit STM32-F4 CPU processor running at 266 MHz could get overloaded. The first measure to reduce the computational cost of the controllers is to run the optimizations of both the inner and outer loops only once every second iteration of the autopilot, which runs at a cycle frequency of 512 Hz.

A system monitoring module in Paparazzi has been used to estimate the autopilot's CPU loads with different configurations using this reduced optimization frequency. These configurations include a combination of the INCA controller with a lower cost outer loop controller, a combination of a lower cost inner loop quadcopter controller with the XINCA controller, and a combination of both the INCA and XINCA controllers. For configurations using the INCA controller, the amount of INCA actuators is varied to determine its effect on computational cost. The results of these measurements can be seen in Table 1. These measurements are obtained on the quadplane itself, yet without flying.

Because of the Active Set Method, the numbers clearly show a quasi-linear correlation between the number of actuators and the CPU load, and that the configuration with both INCA and XINCA does indeed demand a lot of the autopilot's computing power. The actual load during one optimization cycle might however require significantly more computing power. Since the implementation was done on an older code base without an operating system, this could result in unpredictable behavior of the quadplane. Especially some time-critical processes could get in trouble under high CPU load. Ideally, the quadplane's autopilot board is to be replaced by one with sufficient computing power. For this research, however, flight tests will be performed with either both INCA and XINCA without any control surfaces or INCA with all inner loop actuators and a low-cost outer loop controller.

The first test flight aims to confirm that the INCA controller chooses suitable actuators during flight. All inner loop actuators are included in the optimization. Since the CyberZoo's confined space only allows for low-velocity testing, the controller is not expected to allocate a significant amount of control to the control surfaces. The results in Figure 13 confirm this. They show varying inputs for the quadplane's upwards facing rotors, due to a slight asymmetrical configuration, but successfully ensure a stable takeoff and landing. As soon as the Quadplane touches down, the ground reaction forces result in unreachable control demands. This causes the rotors to saturate, after

Table 1. CPU load estimations for different inner and outer loop controllers with different numbers of INCA actuators

Inner loop:	INCA	Other	INCA	
Outer loop:	Other	XINCA	XINCA	INCA Actuators
4	38%	32%	48%	
5	46%			54%
6	54%			62%
7	62%			71%
8	74%			83%

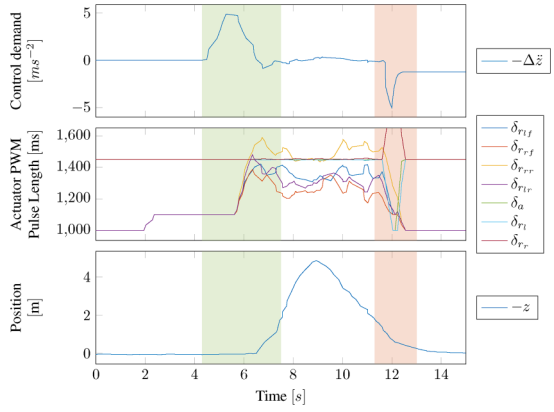


Figure 13. Stable quadplane takeoff and landing using INCA with seven actuators

which the control surfaces are saturated as well in a maximum effort to reach the setpoint.

In the second test, INCA's resilience against actuator saturation is being put to the test. The saturation level is chosen in such a way that one actuator saturates, in this case, $\delta_{r_{lr}}$. Like in its corresponding simulation, Figure 14 shows that INCA prioritizes its pitch and roll commands above its thrust and especially yaw commands, resulting in slower but stable takeoff. Saturating the actuators does result in the INCA optimization having to perform more iterations before it reaches its optimum since the Active Set Method has to explore the edges of the actuator input space in multiple steps. This eventually results in a higher computational load. This test is therefore performed with the INCA controller using only four actuators and a low-cost outer loop controller.

The final flight is the one where the novel XINCA module is being tested. For this flight, the quadplane is controlled by both the INCA and XINCA controllers that together allocate control to a total of five rotors. The flight consists of a takeoff, forward flight, backward flight, and landing. Figure 15 shows that the quadplane effortlessly manages to perform this longitudinal maneuver. The tail rotor command shows that this actuator is indeed used for both forward acceleration phases as expected.

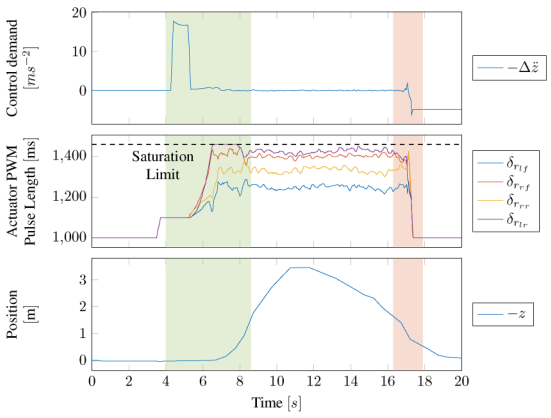


Figure 14. Stable quadplane takeoff and landing with actuator saturation occurring at an actuator PWM pulse length of 1460 ms using INCA with five actuators

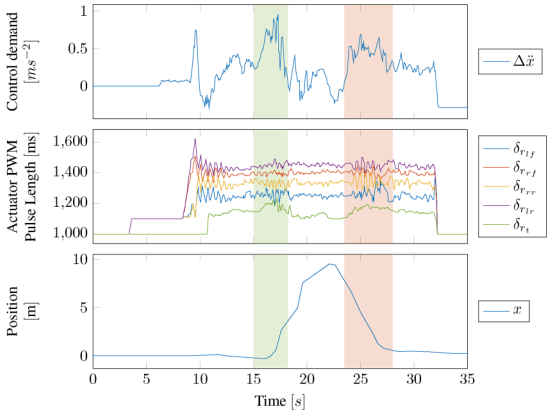


Figure 15. Forwards and backward quadplane flight using XINCA with both pitch increments and tail rotor inputs and INCA with five actuators

Conclusions and Recommendations

During both the simulations and the actual test flights, it was confirmed that the INCA controller chooses suitable actuators and achieves stable flight even in the case of actuator saturation. Prioritizing certain control demands over others plays an important role in maintaining stable flight when saturation occurs. Furthermore, the XINCA controller seamlessly takes the fixed-wing constraints into account in all flight phases without needing to switch modes. Furthermore, relying on the standard formula of lift, it does not require very a very detailed model of the vehicle. The efficient Active Set Method makes it suitable for real-time optimization at high frequencies (200 Hz in this work).

When optimizing commands for too many actuators, this INCA controller is not efficient enough to be used on 32-bit 266 MHz ARM micro-controllers. Allocating control to seven actuators while using a low-cost outer loop controller is at the edge of its computational capacity. Future research,

therefore, requires either a gain in computational efficiency or hardware upgrades.

Finally, the novel XINCA controller is capable of performing optimization in the outer control loop by combining attitude angle commands as well as direct actuator commands. Recalculation of the actuator's effectiveness at every time step allows a single controller to run across the entire flight envelope. This eliminates the inefficient use of separated flight modes while avoiding pitfalls like negative main wing lift in hover. This can contribute to a safer, more efficient, and therefore greener future of human aerial transportation.

Future research on the application of INCA on hybrid vehicles like the quadplane and the application of XINCA should focus on the performance during level flight, as this has not been sufficiently addressed in this work. Outdoor flights should serve two main research objectives. One objective would be to assess how the quadplane allocates more control to its aerodynamic control surfaces as soon as it has an amount of airspeed, making them more effective. The other objective focuses on XINCA, assessing its capabilities to adapt to the different dynamics of a hovering quadplane and one in forward flight.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iD

C. De Wagter https://orcid.org/0000-0002-6795-8454

References

1. Sharkey N. The automation and proliferation of military drones and the protection of civilians. *Law, Innovation and Technology* 2011; 3: 229–240. DOI: 10.5235/175799611798204914.
2. Singh A, Patil D and Omkar S. 2018. Eye in the sky: Real-time drone surveillance system (dss) for violent individuals identification using scatternet hybrid deep learning network. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. Salt Lake City, UT, USA: IEEE, pp. 1710–1718. DOI:10.1109/CVPRW.2018.00214.
3. Momont A. 2014. *Drones for Good*. mathesis, Delft University of Technology. http://resolver.tudelft.nl/uuid:36ce77ad-1b06-4149-8da4-a231d.
4. Pulsiri N and Vatananan-Thesenvitz R. Drones in emergency medical services: A systematic literature review with bibliometric analysis. *International Journal of Innovation and Technology Management* 2020; 18: 2097001. DOI: 10.1142/s0219877020970019.
5. MacSween-George S. 2003. Will the public accept UAVs for cargo and passenger transportation? In: *2003 IEEE Aerospace Conference Proceedings (Cat. No.03TH8652)*, volume 1. Big Sky, MT, USA: IEEE, pp. 1–11. DOI:10.1109/aero.2003.1235066.
6. Mogili UR and Deepak B. Review on application of drone systems in precision agriculture. *Procedia Comput Sci* 2018; 133: 502–509. DOI: 10.1016/j.procs.2018.07.063.
7. Rakha T and Gorodetsky A. Review of unmanned aerial system (UAS) applications in the built environment: Towards automated building inspection procedures using drones. *Autom Constr* 2018; 93: 252–264. DOI: 10.1016/j.autcon.2018.05.002.
8. Madawalagama S, Munasinghe N and Dampegama S et al. 2016. Low cost aerial mapping with consumer grade drones. In: *37th Asian Conference on Remote Sensing*. Colombo, Sri Lanka: ACRS, pp. 1–8.
9. Daibing Z, Xun W and Weiwei K. 2012. Autonomous control of running takeoff and landing for a fixed-wing unmanned aerial vehicle. In: *2012 12th International Conference on Control Automation Robotics & Vision (ICARCV)*. Guangzhou, China: IEEE, pp. 990–994. DOI:10.1109/icarcv.2012.6485292.
10. Palermo M and Vos R. 2020. Experimental aerodynamic analysis of a 4.6%-scale flying-v subsonic transport. In: *AIAA Scitech 2020 Forum*. Orlando, FL: AIAA, pp. 1–19. DOI:10.2514/6.2020-2228. AIAA 2020–2228.
11. Lee DJ, Min BM and Tahk MJ et al. 2008. Autonomous flight control system design for a blended wing body. In: *2008 International Conference on Control, Automation and Systems*. Seoul, Korea (South): IEEE, pp. 192–196. DOI:10.1109/iccas.2008.4694548.
12. Zhiqiang B, Peizhi L and Jinhua W et al. 2011. Simulation system design of a uav helicopter. In: *2011 International Conference on Electric Information and Control Engineering*. Wuhan, China: IEEE, pp. 1–4. DOI:10.1109/iceice.2011.5778108.
13. Luukkonen T. 2011. *Modelling and control of quadcopter*. resreport, Aalto University, School of Science. https://sal.aalto.fi/publications/pdf-files/eluu11˙public.pdf.
14. Smeur E, Höppener D and De Wagter C. 2017. Prioritized control allocation for quadrotors subject to saturation. In: *International Micro Air Vehicle Conference and Flight Competition 2017*. Toulouse, France: IMAV, pp. 37–43. http://www.imavs.org/papers/2017/6.pdf.
15. Saeed AS, Younes AB and Cai C et al. A survey of hybrid unmanned aerial vehicles. *Prog Aerosp Sci* 2018; 98: 91–105. DOI: 10.1016/j.paerosci.2018.03.007.
16. Apkarian J. Attitude control of pitch-decoupled VTOL fixed wing tiltrotor. In: *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*. Dallas, TX, USA: IEEE, pp. 195–201. 2018. DOI:10.1109/icuas.2018.8453473.
17. Takeuchi R, Watanabe K and Nagai I. 2017. Development and control of tilt-wings for a tilt-type quadrotor. In: *2017 IEEE International Conference on Mechatronics and Automation (ICMA)*. Takamatsu, Japan: IEEE, pp. 501–506. DOI:10.1109/icma.2017.8015868.
18. De Wagter C and Smeur EJ. Control of a hybrid helicopter with wings. *International Journal of Micro Air Vehicles* 2017; 9: 209–217. DOI: 10.1177/1756829317702674.

19. Argyle ME, Beach JM and Beard RW et al. 2014. Quaternion based attitude error for a tailsitter in hover flight. In: *2014 American Control Conference*. Portland, OR, USA: IEEE, pp. 1396–1401. DOI:10.1109/acc.2014.6859324.
20. Shaiful DSB, Win LTS and Low JE et al. 2018. Optimized transition path of a transformable hovering rotorcraft (thor). In: *2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. Auckland, New Zealand: IEEE, pp. 460–465. DOI:10.1109/aim.2018.8452703.
21. Gunarathna JK and Munasinghe R. 2018. Development of a quad-rotor fixed-wing hybrid unmanned aerial vehicle. In: *2018 Moratuwa Engineering Research Conference (MERCon)*. Moratuwa, Sri Lanka: IEEE, pp. 72–77. DOI:10.1109/mercon.2018.8421941.
22. Zhang J, Guo Z and Wu L. 2017. Research on control scheme of vertical take-off and landing fixed-wing uav. In: *2017 2^{n}d Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*. Wuhan, China: IEEE, pp. 200–204. DOI:10.1109/acirs.2017.7986093.
23. Orbea D, Moposita J and Aguilar WG et al. 2017. Vertical take off and landing with fixed rotor. In: *2017 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*. Pucon, Chile: IEEE, pp. 1–6. DOI:10.1109/chilecon.2017.8229691.
24. Tielin M, Chuanguang Y and Wenbiao G et al. 2017. Analysis of technical characteristics of fixed-wing VTOL UAV. In: *2017 IEEE International Conference on Unmanned Systems (ICUS)*. Beijing, China: IEEE, pp. 293–297. DOI:10.1109/icus.2017.8278357.
25. Flores G and Lozano R. 2013. Lyapunov-based controller using singular perturbation theory: An application on a mini-uav. In: *2013 American Control Conference*. Washington, DC, USA: IEEE, pp. 1596–1601. DOI:10.1109/acc.2013.6580063.
26. Stolk A. 2017. *Minimum drag control allocation for the Innovative Control Effector aircraft*. mathesis, Delft University of Technology. http://resolver.tudelft.nl/uuid:43ce35bc-eb94-4c7b-a826-80a80.
27. Smeur EJJ, Chu QP and de Croon GCHE. Adaptive incremental nonlinear dynamic inversion for attitude control of micro aerial vehicles. *J Guid Control Dyn* 2016; 39: 450–461. DOI: 10.2514/1.G001490.
28. Horn J. Non-linear dynamic inversion control design for rotorcraft. *Aerospace* 2019; 6: 38. DOI: 10.3390/aerospace6030038.
29. Häkegøard O. 2002. Dynamic control allocation using constrained quadratic programming. In: *AIAA Guidance, Navigation, and Control Conference and Exhibit*. Monterey, California: AIAA, pp. 1–10. DOI:10.2514/6.2002-4761. AIAA 2002–4761.
30. Gavin HP and Scruggs JT. 2020. Constrained optimization using lagrange multipliers. https://people.duke.edu/ hpgavin/cee201/LagrangeMultipliers.p.
31. Johansen TA and Fossen TI. Control allocation–a survey. *Automatica* 2013; 49: 1087–1103. DOI: 10.1016/j.automatica.2013.01.035.
32. Harkegard O. 2002. Efficient active set algorithms for solving constrained least squares problems in aircraft control allocation. In: *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, volume 2. Las Vegas, NV, USA: IEEE, pp. 1295–1300. DOI:10.1109/cdc.2002.1184694.
33. Brisset P, Drouin A and Gorraz M et al. 2006. The paparazzi solution. In: *MAV 2006, 2nd US-European Competition and Workshop on Micro Air Vehicles*. Sandestin, United States: EMAV, pp. 1–15. https://hal-enac.archives-ouvertes.fr/hal-01004157.
34. Smeur E, de Croon G and Chu Q. Cascaded incremental nonlinear dynamic inversion for mav disturbance rejection. *Control Eng Pract* 2018; 73: 79–90. DOI: 10.1016/j.conengprac.2018.01.003.
35. Smeur EJJ, de Croon GCHE and Chu Q. 2016. Gust disturbance alleviation with incremental nonlinear dynamic inversion. In: *International Conference on Intelligent Robots and Systems (IROS)*. IEEE/RSJ, Daejeon, South Korea: IEEE, pp. 5626–5631. DOI:10.1109/IROS.2016.7759827.