XINCA: Extended Incremental Non-linear Control Allocation on a Quadplane

H.J. Karssies, and C. De Wagter*
Delft University of Technology, Micro Air Vehicle Lab, Kluyverweg 1, 2629HS Delft, the Netherlands

Abstract

Controlling over-actuated Unmanned Aerial Vehicle (UAV) is an important task to achieve reliable fail-safe autonomous flight. Incremental Non-linear Control Allocation or INCA has been proposed to solve the platform's control allocation problem by minimizing a set of objective functions with a method known as the Active Set Method. This work proposes an extension to INCA to control the outer loop of a quadplane UAV, an in-plane combination between a quadrotor and a conventional fixed-wing. The novel controller is called Extended INCA or XINCA and optimizes a mix of physical actuator commands and angular control setpoints fed to the vehicle's inner loop. It does so while adapting to varying flight phases, conditions, and vehicle states, and taking into account the aerodynamic properties of the main wing. XINCA has low dependence on accurate vehicle models and requires only several optimization parameters. Flight simulations and experimental flights are performed to prove the performance of both controllers.

1 Introduction

Unmanned Arial Vehicle or Unmanned Aerial Vehicle (UAV)s have gained a tremendous amount of popularity. Not only have they proven to be valuable research platforms and entertaining toys, they have also found many other applications in fields like defence [1], surveillance [2], medical assistance [3], transportation of both goods and humans [4], agriculture [5], inspection [6], mapping [7], 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 like the ones by [8, 9] and [10] for instance master endurance as a result of the passive wing-induced lift that keeps them airborne. Rotorcraft on the other hand, like designs by [11, 12] and [13], 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. Their powered generation of lift

Figure 1: The TU Delft Quadplane in the Cyberzoo.

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 come up with hybrid platforms, that aim to combine the best of different worlds.

Some examples of hybrid platforms include tilt rotor/wing UAVs, tail sitters, transformable UAVs, and quadplanes. Tiltrotor/wing UAVs like designs by [14] and [15] 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 as discussed by [16] and [17] 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 more reliable, lighter, and cheaper platform, albeit at the cost of sensitivity to wind gusts. A completely different class of hybrid UAVs are the ones that are transformable like the one designed by [18]. By changing the configuration of the entire vehicle, they can transform between very different types of UAVs, like for instance a monocopter and a fixed-wing aircraft.

Lastly, a common class of hybrid UAVs is formed by quadplanes, like the one used as an experimental platform for this research (See Figure 1). Earlier designs include those by [19, 20, 21, 22] and [23]. The quadplane has a static configuration with both upward-facing rotors for vertical take-off and landing and fixed wings with a horizontal propulsion unit for horizontal flight. Despite the added weight of flight phase-specific actuators, its mechanical simplicity makes this versatile and enduring vehicle a promising research platform.

*Email address(es): c.dewagter@tudelft.nl

Making such a Quadplane fly as efficiently and safely as possible poses a number of challenges. These include dealing with large flight envelopes, over-actuation, its non-linear nature, and its sensitivity to wind gusts. The quadplane used for this research and its control challenges are described in Section 2. An existing control method called Incremental Non-linear Control Allocation (INCA) is discussed in Section 3, and its optimization methods in Section 4. A proposed extension of this control method, called Extended Incremental Non-linear Control Allocation (XINCA), is presented in Section 5. The implementation of the INCA and XINCA controllers on the TU Delft Quadplane is shown in Section 6, and Sections 7 and 8 respectively present results from simulations and test flights performed using this novel control method. Lastly, Section 9 gives the conclusions.

2 THE TU DELFT QUADPLANE

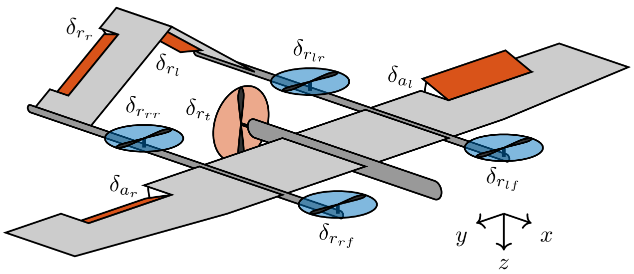


Figure 2: Overview of the nine quadplane actuators
■ = quadcopter actuator set, ■ = fixed wing actuator set

The quadplane is a hybrid of a fixed-wing aircraft and a quadcopter. A conventional example of a quadplane is the one used for this research, the *TU Delft Quadplane*. A schematic representation of this platform 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*. Literature shows that this over-actuation is often dealt with by using only one actuator set during specific flight phases, and only briefly combining them during a transition phase between vertical and horizontal flight [19, 20, 21, 22].

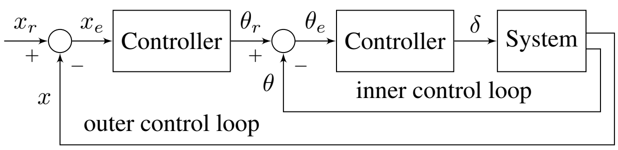


Figure 3: Simplified schematic UAV controller diagram (x = position, θ = attitude, δ = system input)

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 Section 3.

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 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. 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 Section 5, 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.

3 INCA

Incremental Non-linear Control Allocation or INCA, is a very promising control allocation algorithm. It has already theoretically been demonstrated on over-actuated vehicles like the Lockheed Martin Innovative Control Effector aircraft by [24]. [25] have proven the control method to be effective in actual flight on non-over-actuated quadcopters. The architecture of INCA augments a method called Non-linear Dynamic Inversion, or Non-linear Dynamic Inversion (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

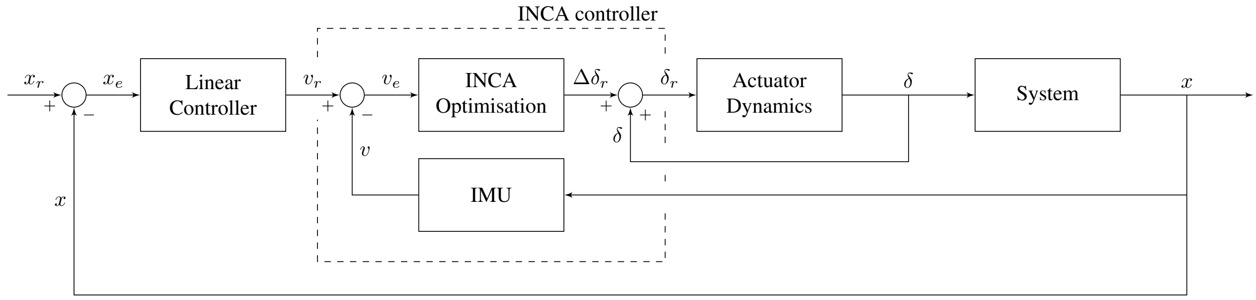


Figure 4: A schematic representation of an INCA controller (x = state vector, v = virtual input, δ = control input vector)

using reliable actuator models. A successful example of an implementation of NDI is the work by [26].

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 Incremental Non-linear Dynamic Inversion (INDI) [25]. 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.

An example where INDI has been proven successfully in quadcopter flight is presented by [27].

Both NDI and INDI invert actuator effectiveness models in order 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 exist infinite solutions. INCA deals with this by expressing this control allocation problem as an optimization problem, that needs to be solved by minimizing a certain 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, the effectiveness of the aerodynamic surfaces is proportional to the square of the true airspeed. The optimization method is presented in Section 4.

4 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:

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

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

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. [28] presents proof that it can provide automatic redistribution of control in case of actuator saturation. [24] and [27] both 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:

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

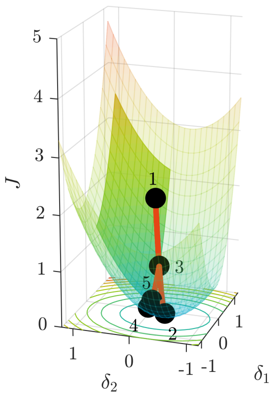
$$\text{subject to } \mathbf{A}\Delta\delta \leq b \tag{3b}$$

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

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

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

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



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	$\delta_1 \geq 0, \delta_2 \geq 0$

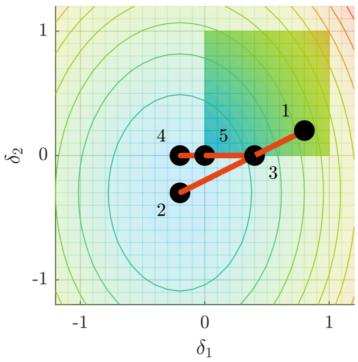


Figure 5: The Active Set Method performed on a hypothetical cost function J with a two-dimensional input space (Constraints: $0 \leq \delta_1 \leq 1$ and $0 \leq \delta_2 \leq 1$, starting point: $(\delta_1, \delta_2) = (0.8, 0.2)$)

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 [29] and $\mathbf{A}$ has full row rank [30]:

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

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:

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

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 [24] and [27], 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 Section 8.

A detailed description of the Active Set Method [31] is summarized below and illustrated in Figure 5 for a hypothetical optimization problem with a constrained two-dimensional input space:

Step 1:

Choose a feasible starting point

Step 2:

Determine the *active set* of constraints, i.e. all constraints at which a control command saturates. Redefine the optimization problem using only the active constraints as equality constraints.

$k = 1, 2, \ldots, N$

Step 3:

Calculate the Lagrange multipliers and solution to the redefined problem using Equations 5a and 5b.

Step 4:

If the solution is infeasible:
Correct the solution by taking the maximum relative step to the new solution without losing feasibility and determine the new active set of constraints.

Else if not all $\lambda \geq 0$:
Release the constraint corresponding to the most negative value in λ from the active set of constraints.

Else: The optimal solution has been found.

Step 5:

Repeat from Step 3 with the new active set of constraints while the optimal solution has not been found.

Choosing a suitable starting point for the Active Set Method has a significant effect on the solver's efficiency. In

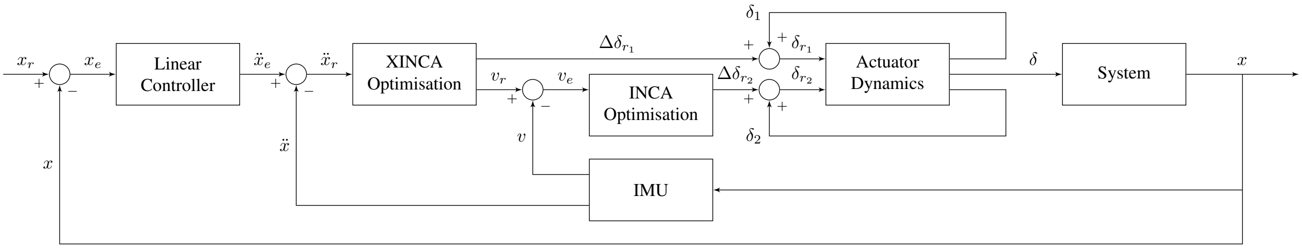


Figure 6: A schematic representation of a XINCA controller (x = state vector, v = virtual input, δ = control input vector)

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 [31], 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.

5 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 6, 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 with, albeit 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 the case of 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 also highly 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.

6 Implementation

XINCA is implemented in the open-source drone hardware and software platform Paparazzi UAV [32]. The quadplane itself makes use of a *Lisa/MX autopilot* board. Since this board 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 [25], and used by [13]. 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 achieved control is calculated as follows:

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

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:

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

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

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 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:

$$H_{act} = \frac{K}{\tau s + 1} \quad (9)$$

Each actuator position is estimated as:

$$\delta_{est} = \delta_{prev} + \alpha(\delta - \delta_{prev}) \quad (10)$$
$$\text{where } \alpha = 1 - e^{-\tau \Delta t}$$

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 in a similar manner as the INCA controller and is based on an existing outer loop INDI module by [33, 34]. This existing module uses the vertical thrust vector to control the position, by either changing this thrust itself or changing its orientation by pitch or roll increments. It is augmented by including a tail rotor command as its fourth actuator. The control is then calculated as follows:

$$\begin{bmatrix}\Delta\ddot{x} & \Delta\ddot{y} & \Delta\ddot{z}\end{bmatrix}^T = \mathbf{H}\begin{bmatrix}v_r & \delta_{r_t}\end{bmatrix}^T \quad (12)$$
$$\text{where } v_r = \begin{bmatrix}\Delta\theta & \Delta\phi & \Delta T\end{bmatrix}^T$$

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

$$\mathbf{H} = \begin{array}{c} \begin{array}{cccc} \Delta\theta & \Delta\phi & \Delta T & \delta_{r_t} \end{array} \\ \begin{bmatrix} -c\theta c\phi T & -s\theta s\phi T & s\theta c\phi & c\theta \\ 0 & -c\phi T & -s\phi & 0 \\ s\theta c\phi T & -c\theta s\phi T & c\theta c\phi & -s\theta \end{bmatrix} \end{array} \begin{array}{c} \Delta\ddot{x} \\ \Delta\ddot{y} \\ \Delta\ddot{z} \end{array} \quad (13)$$

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:

$$T = \ddot{z} - g \quad (14)$$

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:

$$\mathbf{H} = \begin{array}{c} \begin{array}{cccc} \Delta\theta & \Delta\phi & \Delta T & \delta_{r_t} \end{array} \\ \begin{bmatrix} -c\theta c\phi T & -s\theta s\phi T & s\theta c\phi & c\theta \\ 0 & -c\phi T & -s\phi & 0 \\ c\phi\left(s\theta T - \dfrac{C_{L_\alpha}\rho u^2 S}{2m}\right) & -c\theta s\phi T & c\theta c\phi & -s\theta \end{bmatrix} \end{array} \begin{array}{c} \Delta\ddot{x} \\ \Delta\ddot{y} \\ \Delta\ddot{z} \end{array} \quad (15)$$

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:

$$\mathbf{W}_\tau = diag\begin{bmatrix}10 & 10 & 1\end{bmatrix}$$
$$\mathbf{W}_\delta = diag\begin{bmatrix}10 & 10 & 100 & 1\end{bmatrix}$$
$$\gamma = 10000$$
$$\delta_p = \begin{bmatrix}0 & 0 & 0 & 0\end{bmatrix}^T$$

$\mathbf{W}_\tau$ 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.

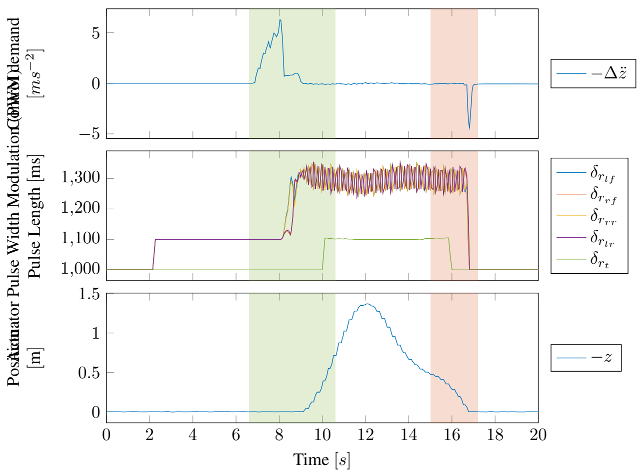


Figure 7: Simulation of vertical quadplane takeoff and landing using XINCA and INCA with five actuators
■ = takeoff, ■ = landing

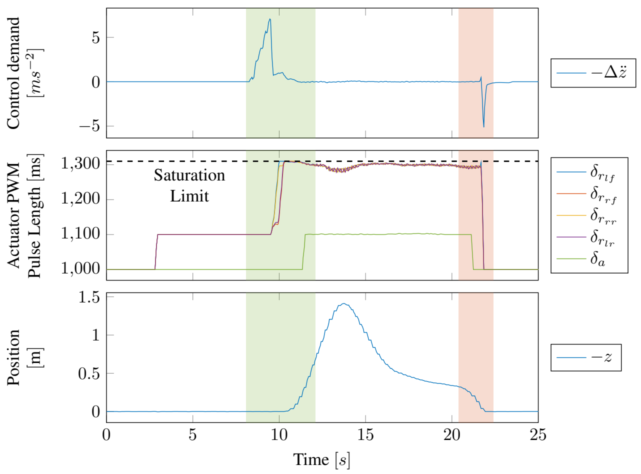


Figure 8: Simulation of a vertical quadplane takeoff and landing with actuator saturation occurring at an actuator PWM pulse length of 1310 ms using XINCA and INCA with five actuators
■ = takeoff, ■ = landing

7 FLIGHT SIMULATIONS

To prove XINCA's performance, several simulations are performed. These are executed within the Paparazzi UAV software to fully assess the performance of the actual code that will also fly onboard the quadplane.

Figure 7 shows a simple 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

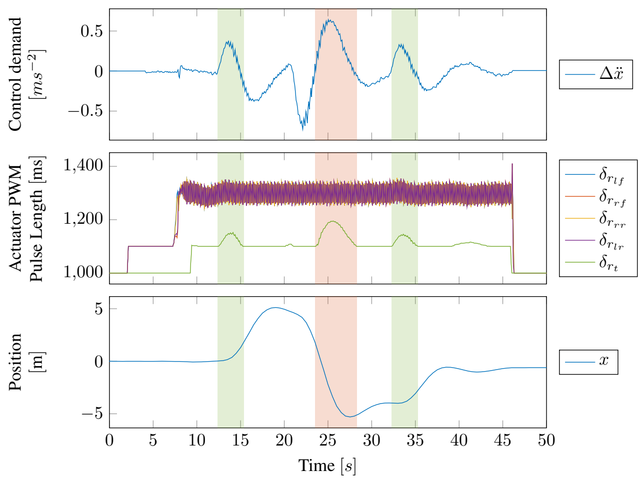


Figure 9: Simulation of forwards and backward flight using XINCA with both pitch increments and tail rotor inputs and INCA with five actuators
■ = forward acceleration by tail rotor, ■ = tail rotor braking

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 8, 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 9 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 [33, 34] 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 10 shows the actuator commands, pitch angle, and lift force for both simulations. The most evident difference can be seen in the pitch angles. Where the INDI controller aggressively pitches forward to achieve forward acceleration, the XINCA controller proves to be able to minimize this negative pitch by using its tail rotor. This difference is reflected in the lift force estimations

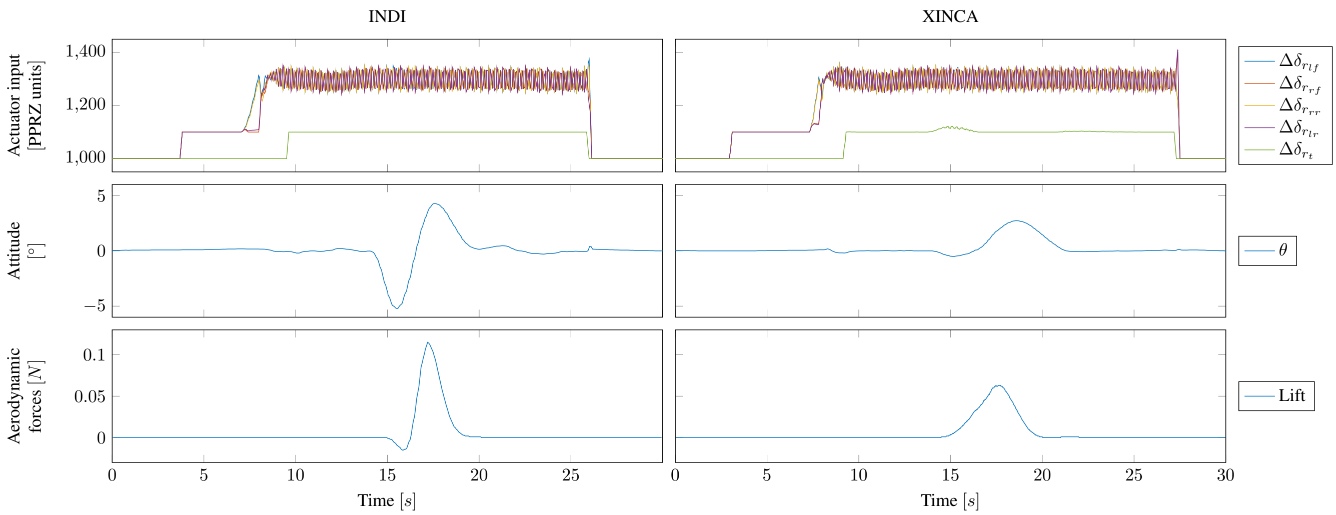


Figure 10: Flight data comparison of forward flight simulation with INDI and XINCA

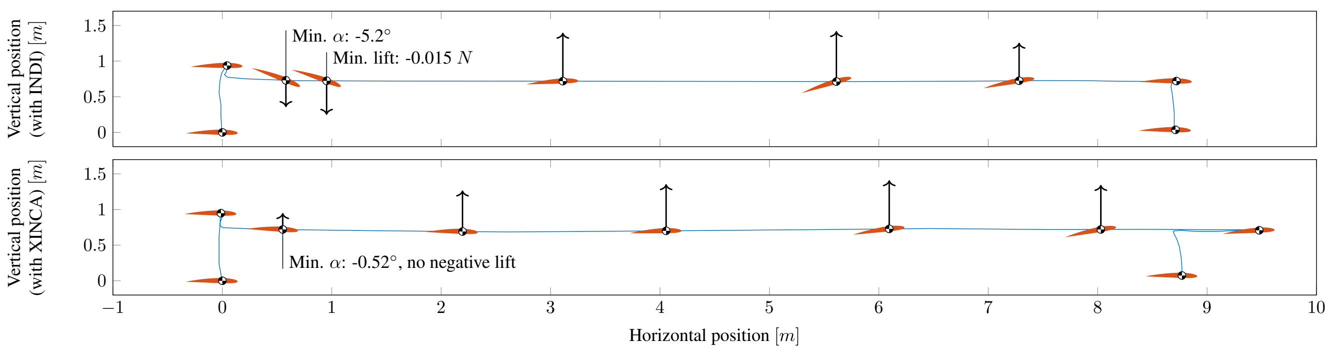


Figure 11: Flight profile comparison of forward flight simulation with INDI and XINCA showing wing-induced lift estimations. Note that illustrated angles of attack are magnified and force vectors are scaled for readability

shown in Figure 11, where the XINCA controller manages to completely avoid the negative lift caused by pitching forward. The INDI controller does inflict some negative lift. At higher wind speeds this negative lift can become very significant and result in an important loss of altitude.

8 FLIGHT EXPERIMENTS

The XINCA controller was tested in real flight tests of the TU Delft quadplane shown in Figure 1. The flight tests are performed in the *Cyberzoo*, which is equipped with an 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 Central Processing Unit (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 fact that the maximum recorded CPU load is still well below 100% can be explained by the fact that the optimization schemes only run once every two cycles, resulting in an average load under 100%. The actual load during one optimization cycle might however re-

	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%

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

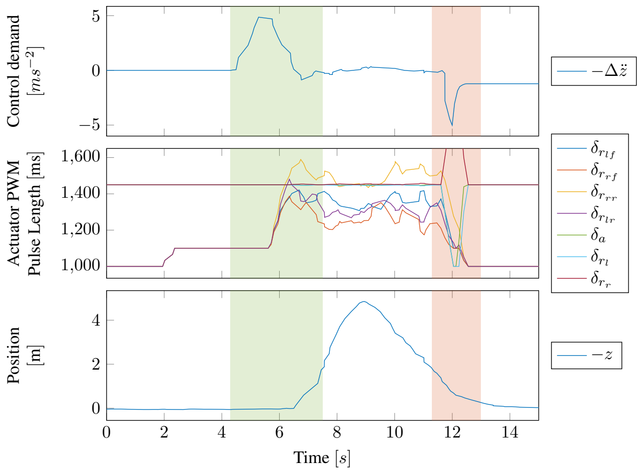


Figure 12: Stable quadplane takeoff and landing using INCA with seven actuators
■ = takeoff, ■ = landing

quire significantly more computing power, resulting in unpredictable behavior of the quadplane. Especially some time-critical processes need to be re-evaluated to perform well 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, so a low-cost outer loop controller is used for this test. 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 12 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 forces result in unreachable control demands. This causes the rotors to saturate at their minimum values, after which the control surfaces are saturated as well in a maximum effort to reach the setpoint.

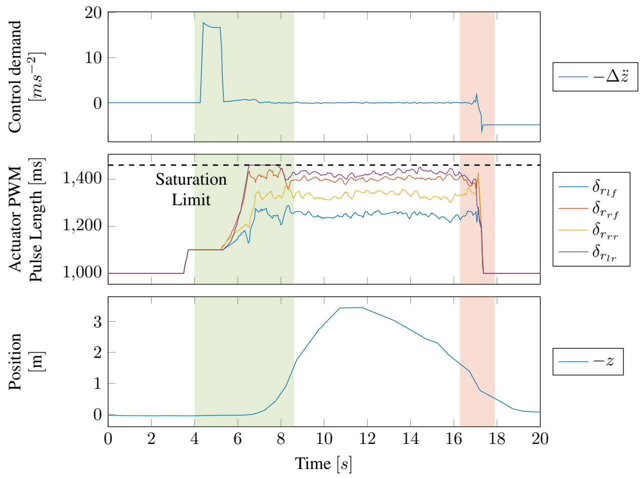


Figure 13: Stable quadplane takeoff and landing with actuator saturation occurring at an actuator PWM pulse length of 1460 ms using INCA with five actuators
■ = takeoff, ■ = landing

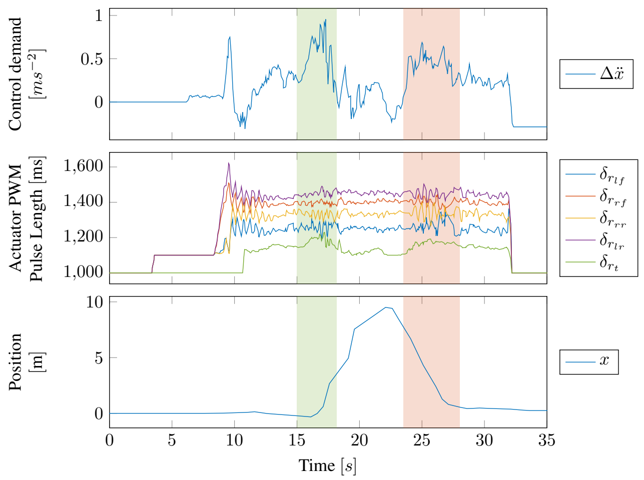


Figure 14: Forwards and backwards quadplane flight using XINCA with both pitch increments and tail rotor inputs and INCA with five actuators
■ = forward acceleration by tail rotor, ■ = tail rotor braking

The difference in actuator inputs between different rotors seen in the first flight can be exploited in the second, where INCA's resilience against actuator saturation is being put to the test. The saturation level is chosen in such a way that one actuator especially saturates, in this case, $\delta_{r_{lr}}$. Like with its corresponding simulation, Figure 13 shows that INCA prioritizes its pitch and roll commands above its thrust and especially yaw commands, resulting in slower but stable takeoff. Saturating 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 14 shows that the quadplane effortlessly manages to perform this longitudinal maneuver. Peaks in the tail rotor command show that this actuator is indeed used for both forward acceleration and backward braking as expected.

9 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 successfully ensures 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, it proves to not require very detailed models of its controlled vehicle, and the Active Set Method makes it suitable for real-time optimization at high frequencies. Recalculation of the actuator's effectiveness at every time step results in high automated adaptability to changing states and conditions to ensure efficient flight control, using the most suitable and efficient actuators available. When optimizing commands for too many actuators, however, this INCA controller is not efficient enough to be used on the TU Delft Quadplane in its current hardware configuration. Allocating control to seven actuators while using a low-cost outer loop controller is at the edge of its computational capacity. Future research on this specific platform, therefore, requires hardware upgrades to achieve more computing power.

Finally, the novel XINCA controller is capable to perform an optimization in the outer control loop by combining attitude angle commands as well as direct actuator commands. This method 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 in general should focus on their performance during level flight, as this has not been sufficiently addressed during 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 forward 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.

REFERENCES

[1] Elisabeth Bumiller and Thom Shanker. War evolves with drones, some tiny as bugs. *The New York Times*, Jun 2019.

[2] Amarjot Singh, Devendra Patil, and SN Omkar. 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*, June 2018.

[3] A. Momont. Drones for good. Master's thesis, Delft University of Technology, 2014.

[4] Lilium, 2020.

[5] U.M. Rao Mogili and B.B.V.L. Deepak. Review on application of drone systems in precision agriculture. *Procedia Computer Science*, 133:502–509, 2018.

[6] Elisabeth Bumiller and Thom Shanker. War evolves with drones, some tiny as bugs. *The New York Times*, Jun 2019.

[7] Sasanka Madawalagama, Niluka Munasinghe, S Dampegama, and L Samarakoon. Low cost aerial mapping with consumer-grade drones. In *37th Asian Conference on Remote Sensing*, pages 1–8, 10 2016.

[8] Zhang Daibing, Wang Xun, and Kong Weiwei. Autonomous control of running takeoff and landing for a fixed-wing unmanned aerial vehicle. *2012 12th International Conference on Control Automation Robotics & Vision (ICARCV)*, 2012.

[9] Marco Palermo and Roelof Vos. Experimental aerodynamic analysis of a 4.6%-scale flying-v subsonic transport. *AIAA Scitech 2020 Forum*, May 2020.

[10] D. Jin Lee, Byoung-Mun Min, Min-Jea Tahk, Hyochoong Bang, and D.h Shim. Autonomous flight control system design for a blended wing body. *2008 International Conference on Control, Automation and Systems*, 2008.

[11] Bai Zhiqiang, Liu Peizhi, Wang Jinhua, and Hu Xiongwen. Simulation system design of a uav helicopter. *2011 International Conference on Electric Information and Control Engineering*, 2011.

[12] Teppo Luukkonen. Modelling and control of quadcopter, Aug 2011.

[13] E.J.J. Smeur, D.C. Höppener, and C. De Wagter. Prioritized control allocation for quadrotors subject to saturation. *International Micro Air Vehicle Conference and Flight Competition (IMAV)*, 2017.

[14] Jacob Apkarian. Attitude control of pitch-decoupled vtol fixed wing tiltrotor. *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2018.

[15] Ryuta Takeuchi, Keigo Watanabe, and Isaku Nagai. Development and control of tilt-wings for a tilt-type quadrotor. *2017 IEEE International Conference on Mechatronics and Automation (ICMA)*, 2017.

[16] Christophe De Wagter and Ewoud J.J Smeur. Control of a hybrid helicopter with wings. *International Journal of Micro Air Vehicles*, 9(3):209–217, Nov 2017.

[17] Matthew E. Argyle, Jason M. Beach, Randal W. Beard, Timothy W. Mclain, and Stephen Morris. Quaternion based attitude error for a tailsitter in hover flight. *2014 American Control Conference*, 2014.

[18] Danial Sufiyan Bin Shaiful, Luke Thura Soe Win, Jun En Low, Shane Kyi Hla Win, Gim Song Soh, and Shaohui Foong. Optimized transition path of a transformable hovering rotorcraft (thor). *2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 2018.

[19] Janith Kalpa Gunarathna and Rohan Munasinghe. Development of a quad-rotor fixed-wing hybrid unmanned aerial vehicle. *2018 Moratuwa Engineering Research Conference (MERCon)*, 2018.

[20] Jian Zhang, Zhiming Guo, and Liaoni Wu. Research on control scheme of vertical take-off and landing fixed-wing uav. *2017 2nd Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*, 2017.

[21] David Orbea, Jessica Moposita, Wilbert G. Aguilar, Manolo Paredes, Rolando P. Reyes, and Luis Montoya. Vertical take off and landing with fixed rotor. *2017 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*, 2017.

[22] Ma Tielin, Yang Chuanguang, Gan Wenbiao, Xue Zihan, Zhang Qinling, and Zhang Xiaoou. Analysis of technical characteristics of fixed-wing vtol uav. *2017 IEEE International Conference on Unmanned Systems (ICUS)*, 2017.

[23] Gerardo Flores and R. Lozano. Lyapunov-based controller using singular perturbation theory: An application on a mini-uav. *2013 American Control Conference*, 2013.

[24] A.R.J. Stolk. Minimum drag control allocation for the innovative control effector aircraft. Master's thesis, Delft University of Technology, 2017.

[25] Ewoud J. J. Smeur, Qiping P. Chu, and Guido C. H. E. de Croon. Adaptive Incremental Nonlinear Dynamic Inversion for Attitude Control of Micro Aerial Vehicles. *Journal of Guidance, Control, and Dynamics*, 39(3):450–461, March 2016.

[26] Joseph Horn. Non-linear dynamic inversion control design for rotorcraft. *Aerospace*, 6(3):38, 2019.

[27] D.C. Höppener. Actuator saturation handling using weighted optimal control allocation applied to an indi controlled quadcopter. Master's thesis, Delft University of Technology, 2017.

[28] Ola Härkegård. Dynamic control allocation using constrained quadratic programming. *AIAA Guidance, Navigation, and Control Conference and Exhibit*, May 2002.

[29] Henri P. Gavin and Jeffrey T. Scruggs. Constrained optimization using lagrange multipliers. *CEE 201L. Uncertainty, Design, and Optimization*, 2020.

[30] Tor A. Johansen and Thor I. Fossen. Control allocation—a survey. *Automatica*, 49(5):1087–1103, 2013.

[31] O. Harkegard. Efficient active set algorithms for solving constrained least squares problems in aircraft control allocation. *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, 2002.

[32] Pascal Brisset, Antoine Drouin, Michel Gorraz, Pierre-Selim Huard, and Jeremy Tyler. The paparazzi solution. In *MAV 2006, 2nd US-European Competition and Workshop on Micro Air Vehicles*, pages 1–15, Sandestin, United States, October 2006.

[33] E.J.J. Smeur, G.C.H.E. de Croon, and Q. Chu. Cascaded incremental nonlinear dynamic inversion for mav disturbance rejection. *Control Engineering Practice*, 73:79–90, 2018.

[34] Ewoud J. J. Smeur, de G.C.H.E de Croon, and Qiping Chu. Gust disturbance alleviation with incremental nonlinear dynamic inversion. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 5626–5631, Daejeon, South Korea, December 2016. IEEE/RSJ, IEEE.