IMAV2017-6
http://www.imavs.org/pdf/imav.2017.6

Prioritized Control Allocation for Quadrotors Subject to Saturation

E.J.J. Smeur*, D.C. Höppener, C. De Wagter
Delft University of Technology, Kluyverweg 1, Netherlands

ABSTRACT

This paper deals with the problem of actuator saturation for INDI (Incremental Nonlinear Dynamic Inversion) controlled flying vehicles. The primary problem that arises from actuator saturation for quadrotors, is that of arbitrary control objective realization. We have integrated the weighted least squares control allocation algorithm into INDI, which allows for prioritization between roll, pitch, yaw and thrust. We propose that for a quadrotor, the highest priority should go to pitch and roll, then thrust, and then yaw. Through an experiment, we show that through this method, and the appropriate prioritization, errors in roll and pitch are greatly reduced when applying large yaw moments. Ultimately, this leads to increased stability and robustness.

1 Introduction

Control allocation is often described as the problem of distributing control effort over more actuators than the number of controlled variables [1, 2, 3]. This is something that occurs in traditional aircraft as well as drones, such as hexarotors and octorotors. What sometimes does not receive a lot of attention, is that the problem of how to deal with actuator saturation is also part of the control allocation topic and, in some cases, can be very important.

Especially for aerial vehicles with coupled control effectors, such as quadrotors, actuator saturation may lead to undesired, or if occuring for longer timespans, even catastrophic behaviour. It may be that the desired thrust, and/or control moments in roll, pitch and yaw, can not be achieved due to actuator saturation. In absence of an adequate control allocation algorithm, it is left to chance which part of the control objective will suffer, it may be the thrust, roll, pitch, or yaw.

However, for the flight stability of multirotor vehicles, it is far more important to apply the right roll and pitch control moments than to apply the right yaw moment, since the thrust vector is indifferent to the yaw in body axis. Therefore, we would like the control allocation algorithm to *prioritize* the control objective of roll and pitch over that of yaw, and to calculate the control inputs accordingly.

In previous research, we have developed an Incremental Nonlinear Dynamic Inversion (INDI) controller for Micro Air Vehicles (MAV) [4, 5]. We have shown that this control method is very good at disturbance rejection and needs little model information. Moreover, we presented a method to include the effects of propeller inertia, yielding faster and more accurate yaw control. This aggressive yaw control can easily lead to saturation of multiple actuators, especially when commanding large yaw changes. These saturations often lead to errors in roll and pitch angles and in the thrust, causing the vehicle to lose control of its position and potentially crash.

But also external moments, such as wind disturbances, or actuator faults can lead to saturation. This is why a control allocation method needs to be added to the INDI control structure. Multiple control allocation algorithms have been proposed, some of which do not adequately address prioritization: ganging, redistributed pseudo-inverse, direct control allocation; and some of which do: linear programming and quadratic programming [6]. In this paper, we will consider a quadratic cost function, and the corresponding quadratic optimization problem. A solution to this problem can be found in a straightforward way using the active set method, as has been shown by Härkegård [7].

In this paper, we integrate the Weighted Least Squares (WLS) control allocation algorithm into the INDI attitude controller. Further, we show through an experiment that prioritization of roll and pitch over yaw leads to stability improvements. The structure of this paper is as follows: first, the INDI control law is introduced in Section 2. Second, Section 3 elaborates on the WLS method and how it integrates with INDI attitude control. Third, the experimental results are presented in Section 4, and we end with conclusions and future work in Section 5.

1.1 Related Work

As opposed to our approach of prioritization, some research has focused on the preservation of control *direction* [8, 3]. This means that in case of saturation, a solution for the actuator inputs is sought that corresponds to a linear scaling of the original control objective. This approach may be useful for systems where all axes are equally important. However, for a quadrotor, if a large yawing moment is needed, the actuators can easily saturate due to the low control effectiveness in this axis. Scaling the desired control moments will make the roll and pitch control suffer, which may lead to instability.

Recently, Faessler et al. implemented a heuristics based algorithm for priority management [9]. They showed that prioritizing roll and pitch over yaw can lead to stability improve-

*Email address: e.j.j.smeur@tudelft.nl

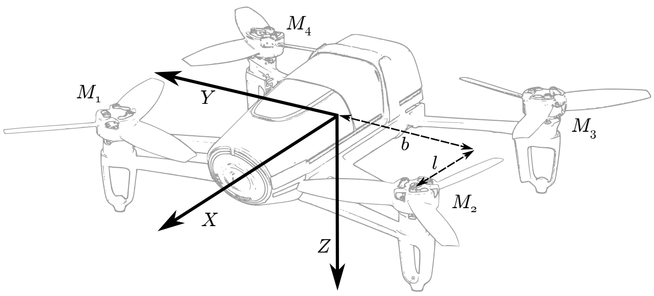

Figure 1: Axis definitions.

ments. However, the suggested algorithm resembles the Redistributed Pseudo Inverse method (RPI), which is known in some cases to not find the control solution even if the control objective is achievable [10]. Furthermore, the scheme is particularly constructed for quadrotors, and does not generalize.

The WLS approach is much more general, as it does not depend on a certain configuration of actuators. The method has been suggested for quadrotors by Monteiro et al. [11], but was only implemented in simulation. Furthermore, the weighting matrix, that determines the priorities in the cost function, is not discussed.

2 Incremental Nonlinear Dynamic Inversion and Actuator Saturation

In previous work [4], we derived INDI control for MAVs. A detailed derivation is beyond the scope of this paper, but the main feature of the controller is its incremental way of controlling angular accelerations. The basic idea is that the current angular acceleration is caused by the combination of inputs and external moments. In order to change the angular acceleration, all that is needed is to take the previous inputs and increment them, based on the error in angular acceleration and the control effectiveness.

A distinction is made between two types of forces and moments: those that are produced by inputs, and those that are produced by changes in inputs. The forces and moments produced due to the propeller aerodynamics fall in the first category, and the torque it takes to spin up a propeller falls in the second category. Both need to be accounted for in different ways, which is why the control effectiveness matrix is split up in two parts: $G = G_1 + G_2$, where G_2 accounts for the propeller spin up torque. Though the algorithms presented here have broad applicability, we will, in order to promote clarity, consider the quadrotor shown in Figure 1, with the illustrated axis definitions. We define the angular rotation vector Ω, its derivative $\dot{\Omega}$ and the angular rate of the propellers ω. Then, if we assume a linear control effectiveness and that gyroscopic effects of the vehicle can be neglected [4], the system equation in incremental form is

$$\dot{\Omega} - \dot{\Omega}_0 + G_2 L(\omega - \omega_0) = (G_1 + G_2)(\omega - \omega_0), \quad (1)$$

subject to

$$\omega_{\min} \leq \omega \leq \omega_{\max}, \quad (2)$$

where L is the lag operator, e.g. $\omega(k-1) = L\omega(k)$. Note that the angular acceleration needs to be obtained by deriving it from gyroscope measurements through finite difference. This signal can be quite noisy, and will need appropriate filtering. In order to synchronize all signals with subscript 0, they all need to be filtered with this same filter.

Equation 1 can be turned into a control law using the matrix inverse or the pseudo-inverse:

$$\omega = \omega_0 + (G_1 + G_2)^{-1}(\nu - \dot{\Omega}_0 + G_2 L(\omega_c - \omega_0)), \quad (3)$$

but calculating the control input like this does not guarantee satisfying Equation 2. If the control inputs exceed the bounds, simply clipping them will result in different control moments than desired.

Instead, Equation 3 is replaced with a method that calculates the control inputs while respecting the limits and prioritization. This can be done with a weighted least squares (WLS) optimization. Since our system description (Equations 1 and 3) is in incremental form, we will first write it as a standard least squares problem through a change of variables:

$$v = Gu \quad (4)$$

subject to

$$u_{\min} \leq u \leq u_{\max}. \quad (5)$$

where the control objective is $v = \dot{\Omega} - \dot{\Omega}_0 + G_2 L(\omega - \omega_0)$, and the input is $u = \omega - \omega_0$. The limits $u_{\min}$ and $u_{\max}$ follow from these definitions and Equation 2.

3 Using the Active Set Method to Solve the Constrained Allocation Problem

Though in this paper we will apply the algorithm to a quadrotor, for the control allocation we will also consider over-actuated systems. This means that we have to include a cost for actuator usage in the cost function, such that there is only one optimum. This will make the derived methodology easily applicable to other systems, like multirotors with more than four rotors, or some over-actuated hybrid systems like the Quadshot [12].

In most cases, we would like to formulate the control allocation problem as a sequential least squares problem. Primarily, we want to minimize the error between the control objective and the angular acceleration increment produced by the calculated control increment. This can be captured in a first cost function. Secondly, given the inputs that minimize the primary cost function, we would like the actuators to spend the lowest amount of energy possible. If G has full rank, the secondary cost function can be omitted, as the primary cost function will only have one solution. However, when there are more actuators than control objectives, the second cost function will minimize expended energy and avoid actuators steering in opposite directions.

The sequential least squares problem is more difficult to solve than a least squares problem with a single cost function. This is why we adopt the WLS problem formulation from Härkegård [7], where the cost for errors in the control objective is combined with a cost for applying inputs:

$$\begin{aligned} C(u) = & \quad \|W_u\ (u - u_d)\|^2 + \gamma \|W_v\ (Gu - v)\|^2 \\ = & \quad \left\| \begin{pmatrix} \gamma^{\frac{1}{2}} W_v\, G \\ W_u \end{pmatrix} u - \begin{pmatrix} \gamma^{\frac{1}{2}}\, W_v\, v \\ W_u\, u_d \end{pmatrix} \right\|^2, \end{aligned} \tag{6}$$

where W_v is the diagonal weighting matrix for the control objective, and W_u is the diagonal weighting matrix for the inputs. The distinction between the primary and secondary objective is made by the scale factor $\gamma >> 1$. For convenience, we define

$$A = \begin{bmatrix} \gamma^{\frac{1}{2}} W_v\, (G_1 + G_2) \\ W_u \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} \gamma^{\frac{1}{2}} W_v v \\ W_u u_d \end{bmatrix}. \tag{7}$$

Now that the problem is formulated as a regular quadratic programming problem, it can be solved using the well known active set method [7, 13, 14], to find the inputs that minimize the cost function. The algorithm divides the inputs into a free set and an active set, which correspond to the inputs that are not saturated and to the actuators that are saturated respectively. The method disregards the inequality constraints for the free set, and for the active set W treats the constraints as equality constraints. At every iteration, it is evaluated if the division between active and free set is correct. For completeness, we explain our implementation of the active set method in Algorithm 1.

The algorithm stops when the solution is optimal, or a maximum number of iterations is reached. Though the algorithm is guaranteed to find the optimum in a finite number of iterations, we may impose a maximum number of iterations that can be executed in a real time application. If the algorithm stops because the maximum number of iterations is reached, the solution will not be optimum. However, since the value of the cost function decreases at each iteration [14], the result will be better than at the start of the algorithm.

3.1 Particularities for WLS applied to INDI

Since we are applying the WLS control allocation scheme to the INDI controller, the inputs are incremental. This means that the bounds on the input (increment) change every time step, and the solution for the increment at one time may not be feasible the next time step. The initial guess for the input, u^0, can therefore not be the solution of the previous time, as is often done for non-incremental controllers [7, 13, 6]. Instead, we take as initial input the mean of the maximum and minimum input increment:

$$u^0 = \frac{1}{2}(u_{\max} - u_{\min}). \tag{13}$$

Additionally, if we consider an over-actuated system, the choice of the preferred increment u_p becomes important, as

Algorithm 1: Active set method for WLS problem

Initialization:
$W = \{\emptyset\}, \quad u^0 = (u_{\max} - u_{\min})/2, \quad d = b - Au^0,$
$S = [\emptyset]$
for $i = 0, 1, 2, .., n_{\max}$ **do**
Determine the free columns in A:

$$A_f = A\,(:, h)\,, \quad h \notin W \tag{8}$$

Determine the optimal perturbation by solving the following least squares problem for p_f:

$$d = A_f p_f \tag{9}$$

Now p is constructed from p_f with zeros for the elements that are in W.
if $u^i + p$ is feasible **then**
$u^{i+1} = u^i + p$ and: $d = d - A_f p_f$
The gradient and Lagrange multipliers are computed with:

$$\nabla = A^T d \ \text{ and: } \ \lambda = S\nabla \tag{10}$$

if all $\lambda \geq 0$ **then**
The solution u^{i+1} is optimal $u = u^{i+1}$;
else
The constraint associated with the most negative λ has to be removed from the active set W. Re-iterate with this active set.
else
The current solution violates a constraint which is not in W. Determine the maximum factor α such that αp is a feasible perturbation, with $0 \leq \alpha < 1$. Update the residual d and the solution u^{i+1}:

$$u^{i+1} = u^i + \alpha p \tag{11}$$

$$d = d - A_f \alpha p_f \tag{12}$$

Finally, update the active set and store the sign of the constraint: $S_{jj} = \text{sign}\,(p_j)$ with j the index of the new active constraint.

there is some degree of freedom in choosing the inputs that will produce the required forces and moments. Some of these combinations may require more energy than what is optimal, for instance if two actuators counteract each other in order to produce a net zero output. Clearly, this can be achieved more efficiently by giving zero input to both actuators. For non-incremental controllers, this means that that u_p is a zero vector. For an incremental controller, this means that $u_p = u_{\min}$, assuming that the actuators produce zero force/moment at $u_{\min}$.

3.2 *Choice of Weighting Matrices*

As for any optimization, the result entirely depends on the choice of the cost function. In this case, we have the freedom to choose W_v, W_u and γ.

For W_v, we choose the diagonal elements to be 1000, 1000, 1 and 100 for roll, pitch, yaw and thrust respectively. The reason that we give roll and pitch a higher priority than thrust, is because the thrust can only be applied in the right direction if the vehicle has the right attitude. As an example, suppose that the quadrotor is inverted. With the thrust vector pointing down, it will lose altitude fast. The controller will have to flip the airframe, and increase thrust to climb. However, if priority would be put on the thrust, the vehicle could, in the extreme case, never change the attitude, as all motors would have to give full thrust.

In general, it appears that satisfying (part of) the roll and pitch objectives, will lead to a reduction of said objectives in the short term, as it typically does not take long to rotate to a desired attitude. On the other hand, satisfying (part of) the thrust objective, might not lead to a reduction of this objective in the short term, as the thrust vector may be pointing in the wrong direction or a large continuous thrust may be needed over a long period of time. Therefore to the authors, prioritizing pitch and roll over thrust seems to be the most stable configuration. However, for a specific quadrotor, the best prioritization scheme may depend on the mission profile.

We choose $\gamma^{\frac{1}{2}} = 10000$ and for W_u we take the identity matrix, since all actuators are 'equal'. Do note that the relative scaling of the signals u and v plays a role here. Also note that, even though we give a lower weighting to some signals, they can still become dominant in the cost function if no bounds are applied. As an example, consider a quadrotor that has to climb five kilometers. In case of a simple PD controller without bounds, an enormous thrust will be commanded, leading to more cost in Equation 6.

3.3 *Computational Complexity*

The computational complexity of the active set algorithm scales with the number of actuators in two ways. First, each additional actuator will add a row and a column to the matrix A, and therefore increase the computational complexity of solving the quadratic problem each iteration of the active set algorithm. Additionally, if there are more actuators, more actuators can saturate in different combinations. This may lead to more iterations on average, as well as more iterations in a worst case scenario.

An analysis of the performance of the active set algorithm on a benchmark problem set, with control objectives in $\mathbb{R}^3$ was done by Petersen and Bodson [13]. They report that the method is efficient in case of few actuators, but that it does not scale well with the problem size. Specifically, for 15 actuators or more, an interior point method is more efficient. Since our control objective is in $\mathbb{R}^4$, this point can be somewhere else.

Clearly, it is very beneficial for the computational performance to have few actuators. If computational time is a problem, it might be an option to combine several actuators into single 'virtual' actuators, often referred to as 'ganging'.

However, we are able to run the WLS scheme on an STMF4 microprocessor, which is equipped with a floating point unit, for four actuators at 512 Hz without any problem. Our implementation uses single precision floating point variables.

4 Experiments

As mentioned in the introduction, actuator saturation often occurs due to yaw commands, as the yaw moment generation of the actuators is relatively weak. Without proper priority management, this is a case where instability can occur. In order to demonstrate the ability of the WLS control allocator to improve stability of the vehicle through priority management, an experiment is performed.

The hypothesis is that the WLS control allocation scheme, with the prioritization as defined in section 3.2, improves the tracking of pitch and roll when large yaw moments are required, as compared to calculating the inputs with the pseudo-inverse and clipping the result.

To test for this hypothesis, the hovering drone will be given an instant step in its heading reference of 50 degrees. This is enough to cause severe actuator saturation. The drone is controlled by a pilot, who will bring the drone back to the hovering position after each maneuver. During the maneuver, the pilot does not give any commands.

4.1 *Experimental Setup*

The test is performed using a Bebop 1 quadrotor from Parrot, running the Paparazzi open source autopilot software. The Bebop is equipped with an internal RPM controller, which accepts commands between 3000 and 12000 RPM. In practice, we found that in static conditions the motors saturate well before 12000 RPM. To avoid commands above the saturation limit that will not have any effect, the limit in the software is put at 9800 RPM.

Again, for details on the INDI control algorithm employed, we refer to our previous papers [4, 5]. However, we will list the parameters used for the experiment. Prior to the experiment, the following control effectiveness matrices were identified through test flights:

$$G_1 = \begin{bmatrix} 18 & -18 & -18 & 18 \\ 11 & 11 & -11 & -11 \\ -0.7 & 0.7 & -0.7 & 0.7 \\ -0.4 & -0.4 & -0.4 & -0.4 \end{bmatrix} \cdot 10^{-3} \tag{14}$$

$$G_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -65 & 65 & -65 & 65 \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot 10^{-3} \tag{15}$$

The filter that is used for the angular acceleration is a second order Butterworth filter with a cutoff frequency of 5 Hz.

4.2 Results

Figure 2 shows the results of the experiment for the pseudo-inverse on the top and for WLS on the bottom. From the left, the first three figures on each row show the Euler angles for 15 and 12 repetitions of the experiment for the pseudo-inverse and WLS respectively. For WLS, two repetitions were rejected, because the pilot steered during the yaw step. The last figure on each row shows the inputs to the actuators during the first repetition.

First, from the plot of the ψ angle it can be observed that with WLS there is no overshoot, but the rise time is longer. The longer rise time can be explained, because for WLS, the inputs are not saturated the whole time the vehicle is moving towards the reference. Because of this, for WLS, the angular velocity does not become as high and the quadrotor is able to reduce the angular velocity without saturating the actuators. For the pseudo-inverse, the situation can be compared with integrator windup. The quadcopter builds up so much angular velocity while the actuators are saturated, that when it has to reduce this angular velocity, the actuators saturate in the other direction and the vehicle overshoots.

Though now it may seem that WLS solves this problem, this is not the case. The figure merely shows that due to the prioritization, the vehicle can not accelerate as fast in the yaw axis, which is why the overshoot does not occur. For larger heading changes, when the vehicle will accumulate angular velocity in the yaw axis over a longer time, overshoot is also observed.

However, the plots of pitch and roll show the merit of the WLS control allocation (note the different scale). To condense this information, we consider the maximum deviation of the roll and pitch angle from zero as a measure of the performance for each repetition. The mean and standard deviation of this maximum error per repetition is presented in Table 4.2.

Clearly there is a very significant improvement in the tracking of the pitch and roll angles. We therefore conclude the hypothesis, that WLS improves the tracking of pitch and roll when producing large yaw moments, to be true.

Finally, from Figure 2 it does become apparent that there still is some small cross coupling between roll and pitch moments and the yaw moment for WLS. The exact cause is beyond the scope of this paper, and may be a topic of future research, but there are possible explanations. For instance, the controller takes into account a linear control effectiveness, while this can be expected to be a quadratic one. Especially for large input changes, as is the case here, some error may be expected. Furthermore, we may consider the fact that for WLS, everything is combined into one cost function. This means that putting more weight on roll and pitch may reduce the error in tracking these angular accelerations, but will never bring it to zero. To improve this, the sequential formulation may be a solution.

	ϕ		θ	
	μ	σ	μ	σ
Pseudo-inverse	12.2	4.8	22.8	9.7
WLS	0.9	0.2	0.5	0.4

Table 1: Mean and standard deviation of the maximum pitch and roll error in degrees.

5 Conclusion

In this paper we have applied the WLS control allocation scheme to incremental nonlinear dynamic inversion control. We propose the following prioritization of controlled forces an moments: first roll and pitch, then thrust, then yaw. This ensures the capability of the vehicle to come back to a stable situation from any attitude. Through an experiment we show that the WLS control allocation with these priorities improves the stability when applying large yaw moments.

The algorithm is readily applicable to other types of MAVs for which priorities in controlled axes can be defined, such as hexacopters, or even hybrid aircraft such as the *Cyclone* [15]. Future research will focus on how constraints in the guidance loop should be taken into account, and how this is affected by limits in the inner loop. Finally, given the strong disturbance rejection properties of the INDI controller, this control allocation scheme is expected to also increase the robustness against faults.

Acknowledgements

The authors would like to thank Anton Naruta for his help with the implementation. This work was supported by the Delphi Consortium.

References

[1] M. W. Oppenheimer, D. B. Doman, and M. A. Bolender. Control allocation for over-actuated systems. In *Mediterranean Conference on Control and Automation*, June 2006. doi: 10.1109/MED.2006.328750.

[2] Abbas Chamseddine, Iman Sadeghzadeh, Youmin Zhang, Didier Theilliol, and Ahmed Khelassi. Control allocation for a modified quadrotor helicopter based on

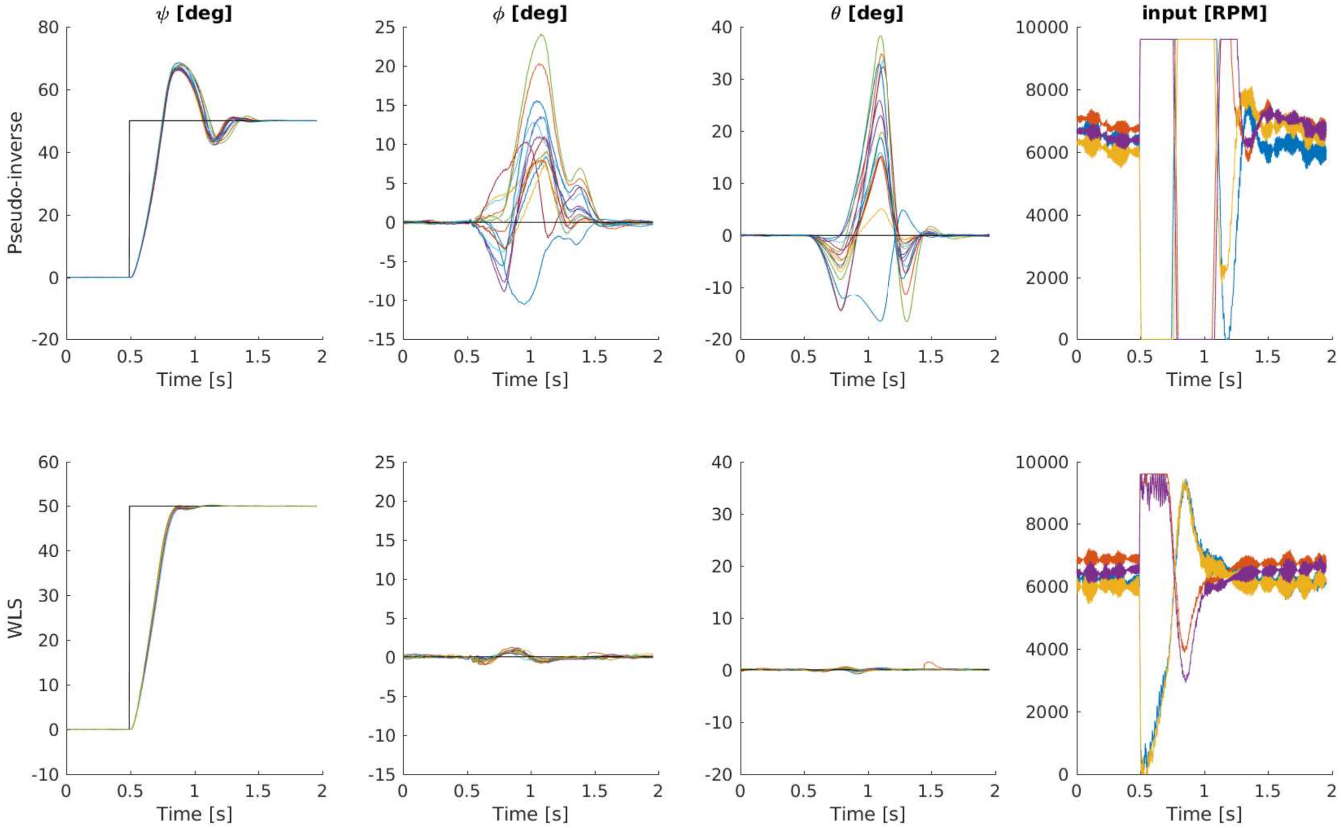

Figure 2: Results of the experiment for the pseudo-inverse (top) and WLS (bottom).

reliability analysis. In *Infotech@Aerospace Conference*, June 2012. doi: 10.2514/6.2012-2511.

[3] Wayne C. Durham. Computationally efficient control allocation. *Journal of Guidance Control and Dynamics*, 24(3):519–524, May 2001. doi: 10.2514/2.4741.

[4] 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. doi: 10.2514/1.G001490.

[5] 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)*. IEEE/RSJ, December 2016. doi: 10.1109/IROS.2016.7759827.

[6] Tor A. Johansen and Thor I. Fossen. Control allocation - a survey. *Automatica*, 49(5):1087–1103, November 2012. doi: 10.1016/j.automatica.2013.01.035.

[7] Ola Härkegård. Efficient Active Set Algorithms for Solving Constrained Least Squares Problems in Aircraft Control Allocation. In *41th IEEE Conference on Decision and Control*, December 2002. doi: 10.1109/CDC.2002.1184694.

[8] P. de Lamberterie, T. Perez, and A. Donaire. A low-complexity flight controller for unmanned aircraft systems with constrained control allocation. In *2011 Australian Control Conference*, pages 284–289, November 2011.

[9] M. Faessler, D. Falanga, and D. Scaramuzza. Thrust mixing, saturation, and body-rate control for accurate aggressive quadrotor flight. *IEEE Robotics and Automation Letters*, 2(2):476–482, April 2017. ISSN 2377-3766. doi: 10.1109/LRA.2016.2640362.

[10] Marc Bodson. Evaluation of optimization methods for control allocation. *Journal of Guidance, Control, and Dynamics*, 25(4):703–711, July 2002. doi: 10.2514/2.4937.

[11] João C. Monteiro, Fernando Lizarralde, and Liu Hsu. Optimal control allocation of quadrotor uavs subject to actuator constraints. In *American Control Conference*, July 2016. doi: 10.1109/ACC.2016.7524963.

[12] Pranay Sinha, Piotr Esden-Tempski, Christopher Forrette, Jeffrey Gibboney, and Gregory Horn. Versatile, modular, extensible vtol aerial platform with autonomous flight mode transitions. In *IEEE Aerospace Conference*, March 2012. doi: 10.1109/AERO.2012.6187313.

[13] John A. M. Petersen and Marc Bodson. Constrained quadratic programming techniques for control allocation. *IEEE Transactions on Control Systems Technology*, 14(1):91–98, January 2006. doi: 10.1109/TCST.2005.860516.

[14] Åke Björck. *Numerical Methods for Least Squares Problems*, pages 187–213. Siam, 1996. ISBN 978-0-89871-360-2. doi: 10.1137/1.9781611971484.

[15] Murat Bronz, Ewoud J J Smeur, Hector Garcia de Marina, and Gautier Hattenberger. Development of a fixed-wing mini uav with transitioning flight capability. In *AIAA Aviation Forum*, June 2017.