Efficient Active Set Algorithms for Solving Constrained Least Squares Problems in Aircraft Control Allocation

Ola Härkegård[1]

Abstract

In aircraft control, control allocation can be used to distribute the total control effort among the actuators when the number of actuators exceeds the number of controlled variables. The control allocation problem is often posed as a constrained least squares problem to incorporate the actuator position and rate limits. Most proposed methods for real-time implementation, like the redistributed pseudoinverse method, only deliver approximate, and sometimes unreliable solutions. In this paper we investigate the use of classical active set methods for control allocation. We develop active set algorithms that always find the optimal control distribution, and show by simulation that the timing requirements are in the same range as for two previously proposed solvers.

1 Introduction

Aircraft control is one important application where the number of actuators often exceeds the number of variables to be controlled. Such actuator redundancy is motivated by enhanced maneuverability and tolerance towards actuator failures. How to distribute the control effort among the actuators is known as the control allocation problem.

Several different formulations, like direct control allocation [7], daisy chaining [5], and constrained linear programming [4, 12] have been proposed, see [2] for a survey. However, the most frequently encountered approaches are based on constrained quadratic programming [16, 10, 6, 3], and this is the formulation that we will use. Despite its widespread use, most numerical solutions proposed for this formulation of the control allocation problem only deliver approximate solutions of varying accuracy. The underlying, often unspoken, claim is that the general constrained quadratic programming machinery is too complex to be used in real-time aircraft applications.

In this paper we investigate this claim and propose two algorithms, based on classical active set methods [1, 14], for finding the optimal solution of the control allocation problem posed as a constrained sequential least squares problem. Open loop simulations show that the proposed algorithms have timing properties similar to the redistributed pseudoinverse method in [16] and the fixed-point algorithm in [6], and produce solutions with better accuracy.

2 Control Allocation in a Least Squares Framework

The essential task of a control allocator is to compute actuator commands $u \in \mathbb{R}^m$ which provide an overall control effort, a virtual control, $v \in \mathbb{R}^n$ where $m > n$. In the aircraft control case, u represents the commanded positions of the control surfaces and other available controls, while v represents the moments or angular accelerations to be produced in pitch, roll, and yaw ($n = 3$) to achieve the desired aircraft dynamics.

Ignoring the actuator dynamics, which is much faster than the remaining aircraft dynamics, the generated virtual control is assumed to be given by Bu, where B is the control effectiveness matrix. The control vector u is restricted by

$$\underline{u} \leq u \leq \overline{u} \tag{1}$$

where $\underline{u}$ and $\overline{u}$ are lower and upper bounds determined at each sampling instant by the position and rate limits of the actuators.

From a pragmatic point of view the control allocation problem can be stated as follows. Given a virtual control command v, determine u, satisfying (1), such that $Bu = v$. If there are several solutions, pick the best one. If there is no solution, determine u such that Bu approximates v as well as possible.

2.1 Problem Formulations

As stated in the introduction, we will use the 2-norm, $\|u\|^2 = u^T u$, as a measure of how "good" a solution or an approximation is. This leads to the following *sequential least squares* formulation of the control allocation problem:

$$\begin{aligned} u_S &= \arg\min_{u \in \mathcal{K}} \|W_u(u - u_p)\| \\ \mathcal{K} &= \arg\min_{\underline{u} \leq u \leq \overline{u}} \|W_v(Bu - v)\| \end{aligned} \tag{2}$$

[1]Division of Automatic Control, Deptartment of Electrical Engineering, Linköpings universitet, SE-581 83 Linköping, Sweden. `http://www.control.isy.liu.se`, `ola@isy.liu.se`

Here, u_p is the preferred control vector and W_u and W_v are nonsingular weighting matrices. Equation (2) should be interpreted as follows: Given $\mathcal{K}$, the set of feasible controls that minimize $Bu - v$ (weighted by W_v), pick the control that minimizes $u - u_p$ (weighted by W_u).

An approximate way to reformulate a sequential least squares problem like (2) is to merge the two optimization criteria into one by summation and form a *weighted least squares* problem:

$$u_W = \arg \min_{\underline{u} \le u \le \overline{u}} \left\| W_u(u-u_p) \right\|^2 + \gamma \left\| W_v(Bu-v) \right\|^2 \quad (3)$$

To emphasize that primarily, $Bu - v$ should be minimized, a large value of the weighting factor γ is used. In [1] it is shown that $\lim_{\gamma \to \infty} u_W(\gamma) = u_S$.

2.2 Numerical Methods

Let us now survey some of the numerical methods proposed in the literature for solving (2) and (3).

Sequential Least Squares Most existing methods for solving the sequential least squares problem (2) are based on the well known fact that if we disregard the actuator constraints (1), then (2) has a closed form pseudoinverse solution, see, e.g., [1]. In [16, 9] a redistribution scheme is used, in which all actuators that violate their bounds in the pseudoinverse solution are saturated and removed from the optimization. Then, the problem is resolved with the remaining actuators as free variables. The procedure is repeated until a feasible iterate has been reached or all variables have saturated. This *redistributed pseudoinverse* (RPI) scheme does not guarantee that the optimal solution is found or even that $Bu = v$ is met when possible, as illustrated in [2]. A similar method, where only one actuator is saturated in each iteration, is proposed in [10].

Another approximate approach, where the $2m$ box constraints (1) are replaced by one ellipsoidal constraint, is presented in [10]. The new problem is shown to be solved efficiently using a bisection method.

Efforts to compute the exact optimal solution can also be found. In [15], an iterative method, which converges to the optimal solution when the feasible region is given by a general convex set, is presented. In [3], exhaustive search is proposed as a way to determine which actuators should saturate. As noted, the computational requirements of this method grows rapidly with the number of actuators.

In [13], a special case of the sequential least squares problem is considered, namely the *minimal least squares* problem

$$\begin{aligned} u_M &= \arg \min_{u \in \mathcal{K}} \left\| u \right\| \\ \mathcal{K} &= \arg \min_{\underline{u} \le u \le \overline{u}} \left\| W_v(Bu - v) \right\| \end{aligned} \quad (4)$$

where the minimal length solution is picked if $\left\| W_v(Bu - v) \right\|$ does not have a unique feasible minimizer. The author suggests the use of active set methods for solving the problem. The sequential least squares problem (2) can not always be cast into a minimal least squares problem like (4) since in the new variables $\tilde{u} = W_u(u - u_p)$ the constraints (1) will in general no longer be simple box constraints. However, if W_u is a diagonal matrix, corresponding to pure scaling of the variables, the box constraint property is preserved. In these cases, it is favorable to use the minimum least squares approach since it is computationally somewhat cheaper.

Weighted Least Squares For the weighted least squares formulation (3), an iterative fixed-point algorithm is proposed in [6]. The algorithm asymptotically converges to the optimal solution.

3 Active Set Algorithms for Control Allocation

Except for the exhaustive search algorithm, and the active set minimal least squares algorithm, all algorithms in the previous section in general only produce approximate solutions, some of which converge to the optimum as the number iterations goes to infinity. In this section we will further investigate the use of active set methods for control allocation. Active set methods are used in many of today's solvers for constrained quadratic programming, and can be shown to find the optimal solution in a finite number of iterations [14].

A general active set algorithm is outlined in the next section, in which we also motivate why active set methods seem well suited for control allocation. Two active set algorithms, tailored for solving the sequential least squares problem (2) and the weighted least squares problem (3), are presented in Section 3.2 and Section 3.3, respectively.

3.1 The Active Set Algorithm

Consider the bounded and equality constrained least squares problem

$$\min_u \left\| Au - b \right\| \quad (5a)$$

$$Bu = v \quad (5b)$$

$$Cu \ge U \quad (5c)$$

where (5c) is equivalent to (1) if we define $C = \begin{pmatrix} I \\ -I \end{pmatrix}$ and $U = \begin{pmatrix} \underline{u} \\ -\overline{u} \end{pmatrix}$. Active set algorithms (see [1, 14] for an in-depth treatment) solve this problem by solving a sequence of equality constrained problems. In each step some of the inequality constraints are regarded as equality constraints, and form the working set $\mathcal{W}$, while the remaining inequality constraints are disregarded. The working set at the optimum in known as the active set of the solution.

Note that this is much like the RPI method from Section 2.2. The difference is that an active set algorithm is more careful regarding which variables to saturate, and that an active set algorithm has the ability to free a variable that was saturated in a previous iteration.

An active set algorithm for solving (5), adopted from [1], is given below in pseudocode.

Algorithm 1 (Active set algorithm)
Let u^0 be a feasible starting point. A point is feasible if it satisfies (5b) and (5c). Let the working set $\mathcal{W}$ contain (a subset of) the active inequality constraints at u^0.

for $k = 0, 1, 2, \ldots, N-1$
Given u^k, find the optimal perturbation p, considering the constraints in the working set as equality constraints and disregarding the remaining inequality constraints. Solve

$$\begin{aligned} \min_p \; & \|A(u^k + p) - b\| \\ & Bp = 0 \\ & p_i = 0, \; i \in \mathcal{W} \end{aligned} \tag{6}$$

if $u^k + p$ is feasible
Set $u^{k+1} = u^k + p$ and compute the Lagrange multipliers, $\binom{\mu}{\lambda}$, where μ is associated with (5b) and λ with the active constraints in (5c).
if all $\lambda \geq 0$
u^{k+1} is the optimal solution to (5).
else
Remove the constraint associated with the most negative λ from the working set.
else
Determine the maximum step length α such that $u^{k+1} = u^k + \alpha p$ is feasible. Add the primary bounding constraint to the working set.
end

The Lagrange multipliers are determined by

$$A^T(Au - b) = \begin{pmatrix} B^T & C_0^T \end{pmatrix} \begin{pmatrix} \mu \\ \lambda \end{pmatrix} \tag{7}$$

where C_0 contains the rows of C that correspond to constraints in the working set.

So why should an active set algorithm be suited for control allocation?

Active set algorithms are the most efficient when a good estimate of the optimal active set is available. Then the number of changes in the working set, i.e., the number if iterations, will be small. In control allocation, a very good estimate is given by the active set of the solution in the previous sampling instant, since in practice, the optimization problem will not change much between two sampling instants. This means that the optimal solution, and particularly the set of active constraints, will not change much either.

Another appealing property is that in each iteration, a feasible iterate u^{k+1} is produced that yields a lower value of the cost function than the previous iterate, u^k. Thus, the maximum number of iterations, N, can be set to reflect the maximum computation time available.

3.2 Sequential Least Squares

We now present an algorithm for solving the sequential least squares problem (2) in two phases. In phase one, a feasible solution to $Bu = v$ is determined, if possible, which is then used as the starting point in phase two. It is assumed that no three actuators produce coplanar controls. This means that any three columns of B are linearly independent.

Algorithm 2 (Sequential LS)
Phase 1:

1. Let u^0 and $\mathcal{W}$ be the resulting solution and working set from the previous sampling instant.[1]

2. Solve

$$\begin{aligned} u_1 = \arg\min_u \; & \|W_v(Bu - v)\| \\ & \underline{u} \leq u \leq \overline{u} \end{aligned}$$

using Algorithm 1 with the following modification. When the number of free variables exceeds n, the optimal perturbation p is not uniquely determined. In this case, pick the minimum perturbation.

3. If $Bu_1 = v$, move to phase 2. Otherwise, set $u_S = u_1$.

Phase 2:

1. Let u^0 and $\mathcal{W}$ be the resulting solution and working set from phase 1.

2. Solve

$$\begin{aligned} u_S = \arg\min_u \; & \|W_u(u - u_p)\| \\ & Bu = v \\ & \underline{u} \leq u \leq \overline{u} \end{aligned}$$

using Algorithm 1.

---

[1] We make the reasonable assumption that u^0 is feasible w.r.t. the new position limits $\underline{u}$ and $\overline{u}$.

When we have rate constraints affecting $\underline{u}$ and $\overline{u}$, u^0 and $\mathcal{W}$ may not be "compatible" since the fixed components in u^0 are set to the limits specified by $\underline{u}$ and $\overline{u}$ from the *previous* sampling instant. In this case we update the values of the fixed variables in u^0 to reflect the new limits. When the control allocator is initiated, and there is no previous solution available, we can select $u^0 = (\underline{u} + \overline{u})/2$ and $\mathcal{W} = \emptyset$.

3.3 Weighted Least Squares

The weighted least squares problem (3) has a more straightforward solution.

Algorithm 3 (Weighted LS)

1. Let u^0 and $\mathcal{W}$ be the resulting solution and working set from the previous sampling instant.
2. Rewrite the cost function as

$$\left\|W_u(u - u_p)\right\|^2 + \gamma\left\|W_v(Bu - v)\right\|^2 = \left\| \underbrace{\begin{pmatrix} \gamma W_v B \\ W_u \end{pmatrix}}_{A} u - \underbrace{\begin{pmatrix} \gamma W_v v \\ W_u u_p \end{pmatrix}}_{b} \right\|^2$$

and solve

$$u_W = \arg\min_u \left\|Au - b\right\|$$
$$\underline{u} \le u \le \overline{u}$$

using Algorithm 1.

The weighted rows are ordered first in A and b to avoid numerical problems, see [1].

3.4 Computing the Solution

The only information forwarded from one optimization, i.e., from one sampling instant, to the next is the resulting solution and the set of active constraints. All other parameters, such as B, u_p, W_u, and W_v, can be updated to reflect the control effectiveness of the actuators and the control distribution preferences.

The main computational steps in the active set algorithm are to solve (6) for the optimal perturbation and to compute the Lagrange multipliers. In the current implementation, QR decompositions [1] are used for these purposes. Due to space limitations we refer to [11] for details.

The MATLAB m-files used in the simulations can be downloaded from the author's homepage at

`http://www.control.isy.liu.se/~ola/`

4 Simulations

Let us now evaluate the two proposed algorithms and compare them with some of the existing alternative solvers discussed in Section 2.2.

4.1 Algorithms

The algorithms used in the evaluations are:

SLS	Sequential least squares (Algorithm 2)
MLS	Minimal least squares [13]
WLS	Weighted least squares (Algorithm 3 with $\gamma = 1000$)
WLS2	WLS with a maximum of $N = 2$ iterations
RPI	Redistributed pseudoinverse [16]
FXP	Fixed-point iteration algorithm [6] ($\gamma = 1000$, 50 iterations)

In FXP the solution from the previous sample is used as an initial guess to hot start the algorithm.

4.2 Aircraft Simulation Data

Aircraft data, consisting of the control effectiveness matrix B (3×8) and the position and rate limits, are taken from [8] to which we refer for details. The commanded virtual control trajectory, $v(t)$, is shown in Figure 1, and corresponds to the helical path with radius 0.06 used in [8]. $v(t)$ contains 85 samples, each consisting of the commanded aerodynamic moment coefficients C_l, C_m, and C_n. By varying t_f, the final time of the trajectory, different helical rates [8] are achieved. When the rate is too high, some parts of the trajectory become infeasible due to the actuator rate constraints.

4.3 Simulation Results

Three different cases were simulated:

Case 1: Feasible trajectory ($t_f = 25$ s, helical rate $= 0.045$) with W_u and W_v set to identity matrices and u_p set to the zero vector. See Table 1 for simulation results.

Case 2: Same trajectory as in case 1 but with

$$W_u = \text{diag}\begin{pmatrix} 10 & 10 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \qquad (8)$$

This corresponds to penalizing the use of the horizontal tails for control. See Table 2.

Case 3: Partially infeasible trajectory ($t_f = 4$ s, helical rate $= 0.28$) with the same parameter setting as in case 1. See Table 3 and Figure 1.

The simulations were performed in MATLAB 6.1 running on an 800 MHz Pentium III computer. The `tic` and `toc` commands were used to compute the timing properties which were averaged over 1000 runs. The mean and max errors given correspond to the mean and max values of $\left\|Bu(t) - v(t)\right\|$ over time. Furthermore, the average number of saturated controls in the solutions are given and also the average number of iterations performed by each algorithm.

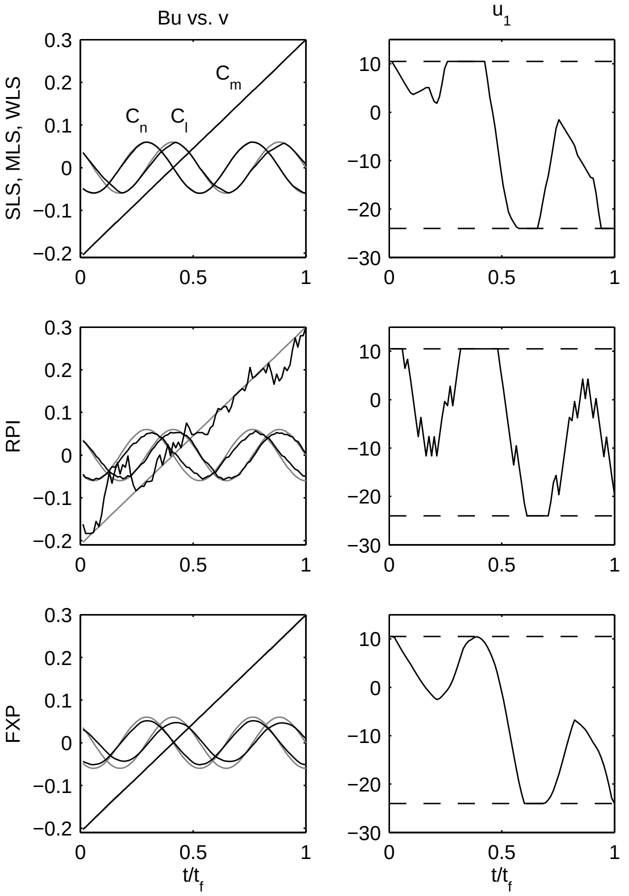

Figure 1: Simulation results for case 3. Left: Commanded (gray) vs. generated (black) virtual controls. Right: Left horizontal tail (u_1) position (solid) and position constraints (dashed).

4.4 Comments

Solution Quality All algorithms produce solutions which satisfy the actuator position and rate constraints.

By construction, SLS and MLS both generate the exact solution to the control allocation problem formulated as the sequential least squares problem (2). WLS only solves an approximation of the original problem, see (3), but with the weight γ set to 1000 as in the simulations, WLS comes very close to recovering the true optimal solution.

WLS2 uses at most two iterations, corresponding to at most two changes per sample in the set of active actuator constraints. For the feasible trajectory, WLS2 almost exactly recovers the optimal solution, while for the infeasible trajectory, the solution quality is somewhat degraded. The reason for WLS2 being so successful can be seen from Figure 2. In most sampling instants, WLS (without any restriction on the number of iterations) finds the optimum in only one or two iterations. In those instants where WLS needs three or more iterations, WLS2 only finds a suboptimal solution, but can be thought of as retrieving the correct active set a few sampling instants later. Similar restrictions on the number of iterations could also be introduced in SLS and MLS.

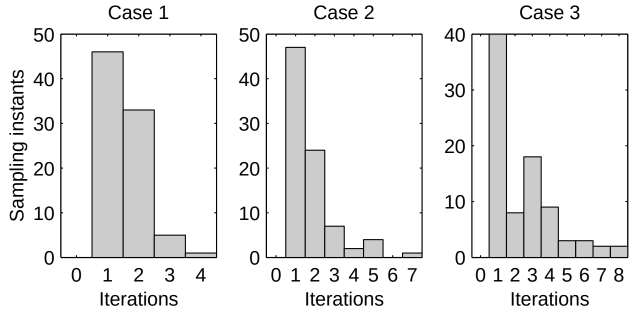

Figure 2: Histograms showing the number of iterations performed by WLS for the three simulated cases. In all cases, it holds that in less than half of the sampling instants more than two iterations are required to find the optimal solution.

The RPI performance seems difficult to predict. In case 1, the RPI solution satisfies $Bu = v$ at each sampling instant although some control surface time histories differ slightly from the optimal ones produced by SLS and MLS. However, in case 2, the RPI solution is rather degraded. The same goes for case 3 where especially the pitching coefficient is poorly reproduced. The general flaw with RPI is the heuristic rule it is based on which claims that it is optimal to saturate all control surfaces which violate their bounds in some iteration. Note that RPI yields the highest average number of actuator saturations in all cases.

In all of the simulated cases, FXP generates rather poor, although continuous, solutions to the control allocation problem. This can be explained by the fact that FXP is a gradient search method and therefore suffers from the scalings introduced in (3) and in (8).

Timing Properties Overall, the timing results for the different methods are within the same order of magnitude. Thus, if RPI is considered a viable alternative for real-time applications, then so should SLS, MLS, and definitely WLS be. For comparison, the typical sampling frequency in modern aircraft is 50-100 Hz which corresponds to a sampling time of 10-20 ms.

WLS2 and FXP both have a maximum computation time which by construction is independent of the trajectory type. This is appealing since it makes the control allocation task easy to schedule in a real-time implementation.

Algo- rithm	Avg. time (ms)	Max time (ms)	Avg. error	Max error	Sat.	Iter.
SLS	1.12	2.44	0	0	1.8	2.5
MLS	1.01	1.83	0	0	1.8	2.5
WLS	0.56	1.17	1.6e-5	3.1e-5	1.8	1.5
WLS2	0.55	0.77	1.6e-5	3.1e-5	1.8	1.5
RPI	0.91	1.50	0	0	2.0	2.4
FXP	1.97	1.99	1.6e-2	2.9e-2	0.5	50.0

Table 1: Simulation results for case 1.

Algo- rithm	Avg. time (ms)	Max time (ms)	Avg. error	Max error	Sat.	Iter.
SLS	1.30	3.41	0	0	2.9	2.9
MLS	1.12	2.75	0	0	2.9	2.8
WLS	0.60	1.84	1.3e-4	6.3e-4	2.9	1.8
WLS2	0.56	0.82	1.1e-3	2.1e-2	3.1	1.5
RPI	1.09	1.84	1.1e-2	9.0e-2	4.3	3.0
FXP	1.98	1.98	4.0e-2	7.1e-2	1.0	50.0

Table 2: Simulation results for case 2.

Algo- rithm	Avg. time (ms)	Max time (ms)	Avg. error	Max error	Sat.	Iter.
SLS	0.80	3.28	4.5e-3	1.2e-2	6.1	2.5
MLS	0.89	2.52	4.5e-3	1.2e-2	6.1	2.5
WLS	0.68	2.11	4.4e-3	1.2e-2	6.1	2.5
WLS2	0.51	0.79	1.0e-2	2.7e-2	6.4	1.6
RPI	0.94	1.46	3.0e-2	1.1e-1	7.6	2.6
FXP	1.98	2.01	1.3e-2	2.1e-2	1.0	50.0

Table 3: Simulation results for case 3.

5 Conclusions

The main conclusion that can be drawn from our investigations is that classical active set methods seem well suited for real-time aircraft control allocation. The computational complexity is similar to that of the redistributed pseudoinverse method and the solutions produced are in general of the same or better quality.

Acknowledgment

The author would like to express his gratitude to Emeritus Professor Åke Björck at the Department of Mathematics, Linköpings universitet, for remembering and suggesting the 1984 paper by Lötstedt [13].

References

[1] Å. Björck. *Numerical Methods for Least Squares Problems.* SIAM, 1996.

[2] M. Bodson. Evaluation of optimization methods for control allocation. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Montreal, Canada, Aug. 2001.

[3] M. Bodson and W. A. Pohlchuk. Command limiting in reconfigurable flight control. *Journal of Guidance, Control, and Dynamics*, 21(4):639–646, July–Aug. 1998.

[4] J. Buffington, P. Chandler, and M. Pachter. On-line system identification for aircraft with distributed control effectors. *International Journal of Robust and Nonlinear Control*, 9(14):1033–1049, 1999.

[5] J. M. Buffington and D. F. Enns. Lyapunov stability analysis of daisy chain control allocation. *Journal of Guidance, Control, and Dynamics*, 19(6):1226–1230, Nov.–Dec. 1996.

[6] J. J. Burken, P. Lu, Z. Wu, and C. Bahm. Two reconfigurable flight-control design methods: Robust servomechanism and control allocation. *Journal of Guidance, Control, and Dynamics*, 24(3):482–493, May–June 2001.

[7] W. C. Durham. Constrained control allocation: Three moment problem. *Journal of Guidance, Control, and Dynamics*, 17(2):330–336, Mar.–Apr. 1994.

[8] W. C. Durham and K. A. Bordignon. Multiple control effector rate limiting. *Journal of Guidance, Control, and Dynamics*, 19(1):30–37, Jan.–Feb. 1996.

[9] R. L. Eberhardt and D. G. Ward. Indirect adaptive flight control of a tailless fighter aircraft. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, pages 466–476, Portland, OR, USA, 1999.

[10] D. Enns. Control allocation approaches. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, pages 98–108, Boston, MA, USA, 1998.

[11] O. Härkegård. Efficient active set algorithms for solving constrained least squares problems in aircraft control allocation. Technical Report LiTH-ISY-R-2426, Department of Electrical Engineering, Linköpings universitet, SE-581 83 Linköping, Sweden, May 2002. Available at `http://www.control.isy.liu.se`.

[12] Y. Ikeda and M. Hood. An application of L1 optimization to control allocation. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Denver, CO, USA, Aug. 2000.

[13] P. Lötstedt. Solving the minimum least squares problem subject to bounds on the variables. *BIT*, 24:206–224, 1984.

[14] J. Nocedal and S. J. Wright. *Numerical Optimization.* Springer, 1999.

[15] A. Sabharwal and L. C. Potter. Convexly constrained linear inverse problems: Iterative least-squares and regularization. *IEEE Transactions on Signal Processing*, 46(9):2345–2352, Sept. 1998.

[16] J. C. Virnig and D. S. Bodden. Multivariable control allocation and control law conditioning when control effectors limit. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Scottsdale, AZ, USA, Aug. 1994.