Modeling, Identification and Control, Vol. 36, No. 2, 2015, pp. 67–79, ISSN 1890–1328

Multicopter Design Optimization and Validation

Øyvind Magnussen   Morten Ottestad   Geir Hovland

University of Agder, Faculty of Engineering and Science, Department of Engineering Sciences, Jon Lilletuns veg 9, 4879 Grimstad, Norway

Abstract

This paper presents a method for optimizing the design of a multicopter unmanned aerial vehicle (UAV, also called multirotor or drone). In practice a set of datasheets is available to the designer for the various components such as battery pack, motor and propellers. The designer can not normally design the parameters of the actuator system freely, but is constrained to pick components based on available datasheets. The mixed-integer programming approach is well suited to design optimization in such cases when only a discrete set of components is available. The paper also includes an experimental section where the simulated dynamic responses of optimized designs are compared against the experimental results. The paper demonstrates that mixed-integer programming is well suited to design optimization of multicopter UAVs and that the modeling assumptions match well with the experimental validation.

Keywords: Multicopter, multirotor, drone, UAV, mathematical modeling, design optimization, experimental validation.

Nomenclature

ω	Speed: [rad/s]
ρ	Density of air
$B_{i,Ah}, B_{i,V}$	Battery i Ah and voltage
C_p	Power coefficient
C_t	Thrust coefficient
D_i	Diameter of propeller i
F, F_{req}	Propeller thrust
g	Gravity constant
i_m, τ_m	Motor current and torque
J	Total inertia
$J_{m,i}$	Inertia of motor i
$J_{p,i}$	Inertia of propeller i
$L_f(D, N_a)$	Length of frame as function of diameter and number of actuators
m_L	Mass of payload
$m_{b,i}$	Mass of battery i
$m_{m,i}$	Mass of motor i
$m_{p,i}$	Mass of propeller i
n	Speed: Revolutions per second
N_a	Number of actuators
$P_{b,i}, P_{m,i}, P_{p,i}$	Price of battery, motor, propeller i
R_a, K_e, K_t	Motor constants
T	Actuator torque
V_i, V_m	Supply and motor voltages
W	Power

doi:10.4173/mic.2015.2.1

1 Introduction

The interest in multicopters (also called multirotors or drones) has increased significantly in recent years, both in academic research and in commercial applications. One example is the Prime Air multicopter by Amazon, see Fig. 1 and Amazon (2015). Other examples are the multicopters designed and developed to demonstrate acrobatic abilities, see for example Hehn and D'Andrea (2014) and the references therein. Despite the large number of published works, very little is published on design optimization of multicopters given an intended application and desired performance specifications.

In Magnussen et al. (2014) a concept for design optimization of a multicopter based on mixed-integer programming was presented. In the current paper the same concept is described in more detail giving all the necessary information to allow a design engineer to repeat the design procedure. In normal practice, a set of datasheets of components such as battery pack, motors and propellers is available to the designer. Variables such as thrust per propeller or motor torque can not be designed freely, but must be chosen from a fixed set of available datasheets. The mixed-integer programming framework solves optimization problems that fall within this category. Certain variables are constrained to be discrete (for example a Boolean variable indicating selection of a particular component represented by a datasheet) while others are continuous variables (for example total weight and flight time). As demonstrated in the paper, the mixed-integer linear programming framework also allows for approximation of nonlinear functions. A nonlinear function is approximated by a set of discrete regions of the y-axis, each represented by a linear function. The accuracy of the approximation can be adjusted by changing the total number of linearized regions.

Numerical methods for nonlinear optimization normally suffer from drawbacks such as long computation times and the fact that the methods can end up in undesirable sub-optimal local minima. Examples are Newton-Raphson gradient search methods and evolutionary-based search methods, such as the Complex method, see for example Whitney (1969) and Tyapin and Hovland (2009) where the optimization algorithm took several hours to converge towards an acceptable but sub-optimal solution. A linear program (LP) on the other hand, using the interior point method, can be solved in polynomial time, see Karmarkar (1984). The solution of an LP with a large number of variables can be found quickly using commercial solvers such as IBM CPLEX, see IBM (2015). The solution of a mixed-integer linear program (MILP) is built on an LP solver and techniques such as branch-and-bound. The optimization solver formulated in this way can handle nonlinear function approximations and discrete design variables and returns a repeatable solution in relatively short time compared to an iterative nonlinear search algorithm where the solution can vary from run to run, depending on the initial conditions. It should be noted, however, that mixed-integer optimization problems are NP-hard, see for example Clausen (1999).

Figure 1: *Example of multirotor UAV design by Amazon for transport and delivery of parcels.*

The paper also contains a section presenting experimental validation of the performance of the multicopter actuators. Both the actuator rise-time constants and the total flight time used in the design optimization procedure are compared with the same parameters estimated from real measurements using the various components available to the designer. The validation compares the performance of four propellers and three motors against the simulation model in three different test cases: I) Longest possible flight time, no payload, II) Longest possible flying time, 1.5kg payload and III) Fastest possible motor response, no payload. For the three test cases, optimized solutions were calculated for multicopters with 4, 6 and 8 actuators, giving a total of 9 potentially different designs. The results demonstrate that the simulation model matches well with the experiments. The experimental validation increases the reliability of the optimization results.

The paper is organized as follows: in Section 2 a dynamic model of a multicopter actuator is presented. Section 3 gives a detailed description of the design optimization procedure. Section 4 presents results from the experimental validation based on three different test cases. Discussion and conclusions are presented in Section 5.

2 Modeling

A Simulink model of the multicopter actuator is shown in Fig. 2. The model generates the simulated thrust response for the 9 different test cases used in this paper (test cases I, II and III with 4, 6 and 8 actuators). The propeller thrusts from the simulation model are later compared with experiments (force measurements using a load cell mounted underneath the actuator). The propeller thrust is calculated from eq. (1):

$$F = C_t \cdot \rho \cdot n^2 \cdot D^4 \tag{1}$$

where C_t is the thrust coefficient, ρ is the density of air, n is the rotor speed (rev/sec) and D is the propeller diameter. The actuator power W and torque T are calculated from eqs. (2)-(3):

$$\begin{aligned} W &= C_p \cdot \rho \cdot n^3 \cdot D^5 && (2) \\ T &= \frac{W}{\omega} && (3) \end{aligned}$$

where C_p is the power coefficient and $w = 2\pi n$ is the rotor speed (rad/sec). The motor controller and actuator dynamics are given by the equations below:

$$\begin{aligned} V_m &= V_i - K_e \omega && (4) \\ i_m &= \frac{V_m}{R_a} && (5) \\ \tau_m &= K_t \cdot i_m && (6) \\ \frac{d\omega}{dt} &= \frac{1}{J}(\tau_m - T) && (7) \end{aligned}$$

where J is the total inertia, V_m, i_m, τ_m are the motor voltage, current and torque, K_e, K_t and R_a are motor parameters and V_i is the supply voltage from the motor controller. Mechanical friction is not included in the dynamic model in eq. (7), since this information is difficult to obtain from the manufacturer's datasheets. Eqs. (1)-(7) are implemented in the Simulink model in Fig. 2.

By combining eqs. (2)-(7) and using the fact that for DC motors the motor torque and back emf constants are equal, that is, $K_t = K_e$, the following transfer function can be defined:

$$\frac{\omega}{V_i}(s) = \frac{K_t}{\left(\frac{JR_a}{K_t^2 + D_\omega R_a}\right) s + 1} \tag{8}$$

where $D_\omega = \frac{T}{\omega} = \frac{\rho C_p n^3 D^5}{\omega^2}$. Assuming constant parameters in eq. (8), the time-constant would be $t_r = \frac{JR_a}{K_t^2 + D_\omega R_a}$. Since D_ω is a function of the propeller speed, this time-constant is an estimation.

3 Design Optimization

A quadratic program is defined as follows:

$$\begin{aligned} &\min && \mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{g}^T\mathbf{x} && (9) \\ &\text{subject to :} && \mathbf{A}\mathbf{x} \leq \mathbf{b} && (10) \end{aligned}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state vector to be solved, $\mathbf{Q} \in \mathbb{R}^{n \times n}$ is a positive semi-definite penalty matrix, $\mathbf{g} \in \mathbb{R}^n$ is a penalty vector, $\mathbf{A} \in \mathbb{R}^{m \times n}$ is the constraint matrix while $\mathbf{b} \in \mathbb{R}^m$ is the constraint vector. When the penalty matrix $\mathbf{Q} = \mathbf{0}$, eqs. (9)-(10) reduce to a linear program. When some elements x_i where $i \in [1, \cdots, n]$ are constrained to be integer variables, eqs. (9)-(10) are called mixed integer quadratic program (MIQP) or, when $\mathbf{Q} = \mathbf{0}$, mixed integer linear program (MILP).

By constraining the integer variables further to accept only Boolean values (0 or 1), logical constraints can be incorporated and linked with the continuous variables in the optimization problem. The following rules 1 and 4 are taken from Bemporad and Morari (1999) and Mignone (2002). Rules 19a and 19b are modified compared to Mignone (2002) ($\delta = 0$ instead of $\delta = 1$). In the rule definitions below the following notation is used:

$$\begin{aligned} m &= \min_{x \in X} f(x) \\ M &= \max_{x \in X} f(x) \end{aligned}$$

ϵ denotes a small, real, positive constant, typically the machine precision.

It should be noted that equality constraints of the type $ax = b$ must be converted to two inequality constraints $ax \leq b$ and $-ax \leq -b$ to satisfy eq. (10). However, the solver used (CPLEX) allows specification of both equality and inequality constraints.

Rule 1: Product of δ and $f(x)$

Product of Boolean variable and function of continuous variables.

$$z = \delta \cdot f(x)$$

or

$$\begin{aligned} &\text{IF } [\delta == 1] \text{ THEN} && z = f(x) \\ &\text{ELSE} && z = 0 \end{aligned}$$

is equivalent to

$$\begin{aligned} -M\delta + z &\leq 0 \\ m\delta - z &\leq 0 \\ -m\delta + z &\leq f(x) - m \\ M\delta - z &\leq -f(x) + M \end{aligned}$$

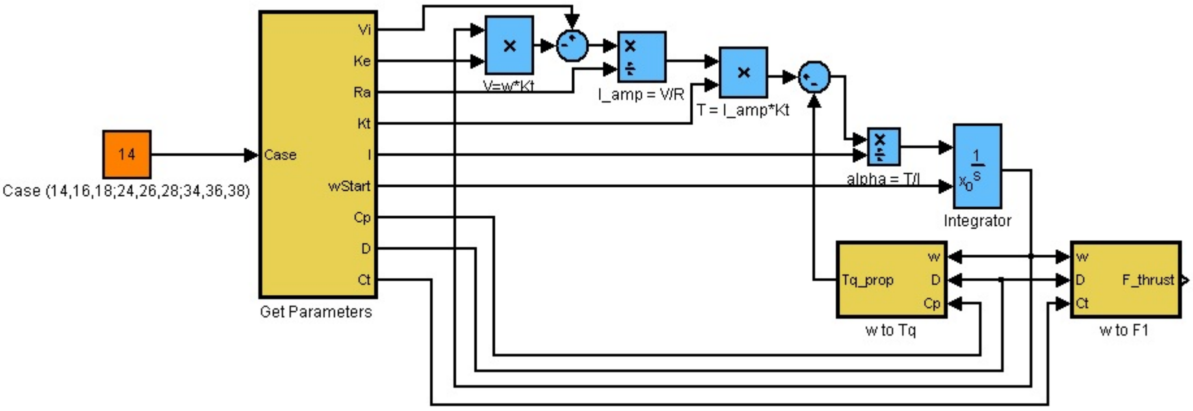

Figure 2: *Simulink implementation of multirotor actuator dynamics.*

Rule 4: Product of several δ's and $f(x)$

Product of several Boolean variables and function of continuous variables.

$$z = \left(\bigwedge_{i=1}^{n} \delta_i\right) \cdot f(x)$$

or

$$\text{IF} \quad \left[\left(\bigwedge_{i=1}^{n} \delta_i\right) == 1\right] \quad \text{THEN} \quad z = f(x)$$

$$\text{ELSE} \quad z = 0$$

is equivalent to

$$\begin{aligned} z + \tilde{M}\left(\sum_{i=1}^{n} \delta_i\right) &\leq f(x) + n\tilde{M} \\ z &\leq f(x) + \tilde{M} \\ -z - \tilde{m}\left(\sum_{i=1}^{n} \delta_i\right) &\leq -f(x) - n\tilde{m} \\ -z &\leq -f(x) - \tilde{m} \\ z + \tilde{m}\delta_1 &\leq 0 \\ &\vdots \\ z + \tilde{m}\delta_n &\leq 0 \\ -z - \tilde{M}\delta_1 &\leq 0 \\ &\vdots \\ -z - \tilde{M}\delta_n &\leq 0 \end{aligned}$$

where $\tilde{M} = \max(0, -m)$, $\tilde{m} = \min(0, -M)$.

Nonlinear positive function: $y = g(x)$

The scalar $y \geq 0$ equals the nonlinear function $g(\cdot)$ of the scalar $x \in [0, x_{max}]$. The set $\mathbf{z}$ approximates $g(x)$ by using a set of straight line segments $\mathbf{a} \in \{a_1, \cdots, a_N\}$, $\mathbf{k} \in \{k_1, \cdots, k_N\}$. Only one of the Boolean variables in the set $\delta \in \{\delta_1, \cdots, \delta_N\}$ is true and specifies which element in $\mathbf{z} \in \{z_1, \cdots, z_N\}$ is used. $M = max(g(x))$. The nonlinear function approximation builds on rules 1, 19a and 19b in the equations below:

$$\begin{aligned} z_1 &= \delta_1(a_1 x + k_1) \\ &\vdots \\ z_N &= \delta_N(a_N x + k_N) \\ [z_1 \leq 0] &\rightarrow [\delta_1 = 0] \\ \left[z_1 \geq \frac{M \cdot 1}{N}\right] &\rightarrow [\delta_1 = 0] \\ &\vdots \\ \left[z_N \leq \frac{M \cdot (N-1)}{N}\right] &\rightarrow [\delta_N = 0] \\ [z_N \geq M] &\rightarrow [\delta_N = 0] \\ \sum_{i=1}^{N} \delta_i &= 1 \\ y &= \sum_{i=1}^{N} z_i \end{aligned}$$

Rule 19a: Implication of $f(x) \leq f_L$

Implication.

$$[f(x) \leq f_L] \rightarrow [\delta = 0]$$

is equivalent to

$$M\delta \leq f(x) - f_L + M + \epsilon$$

Rule 19b: Implication of $f(x) \geq f_U$

Implication.

$$[f(x) \geq f_U] \rightarrow [\delta = 0]$$

is equivalent to

$$M\delta \leq -f(x) + f_U + M - \epsilon$$

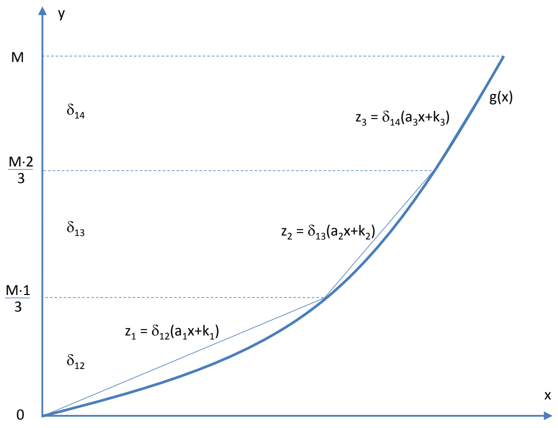

Figure 3: *Illustration of mixed-integer linear approximation of a nonlinear positive function $g(x)$ with three regions (δ_{12}, δ_{13} and δ_{14}).*

Fig. 3 illustrates how a nonlinear function can be approximated using a combination of discrete and continuous variables. In this example three regions δ_{12}, δ_{13} and δ_{14} are chosen to approximate the nonlinear function $g(x)$. The δ's are chosen to represent an equal spacing on the y-axis in the figure. In each region represented by δ_i a linear approximation $z_i = \delta_i(a_i x + k_i)$ is used.

The objective function $\mathbf{g}^T\mathbf{x}$ in eq. (9) can be defined as shown in Table 1 depending on the application's requirement specifications. For payload transfer and package delivery, for example, the longest possible flight time or smallest possible power consumption may be desirable objectives. For acrobatics, on the other hand, the designer may instead want to select the solution giving the smallest possible time-constant or smallest possible roll and pitch inertias.

Criterion	Objective function
Min. Time-constant	$\mathbf{g^T x} = x_{16}$
Min. Power	$\mathbf{g^T x} = x_{17}$
Min. Price	$\mathbf{g^T x} = x_3$
Max. Flight-time	$\mathbf{g^T x} = x_{21}$
Min. Inertia Roll	$\mathbf{g^T x} = x_{13}$
Min. Inertia Pitch	$\mathbf{g^T x} = x_{14}$

Table 1: *Different candidate objective functions to be minimized/maximized depending on the requirement specifications.*

Variables	**Description**
$\delta_1 \cdots \delta_4$	Propeller 1-4 selection
$\delta_5 \cdots \delta_7$	Motor 1-3 selection
$\delta_8 \cdots \delta_{11}$	Battery 1-4 selection
$\delta_{12} \cdots \delta_{i_1}$	Used to calculate n^3
$\delta_{i_1+1} \cdots \delta_{i_2}$	Used to calculate inertia (roll)
$\delta_{i_2+1} \cdots \delta_{i_3}$	Used to calculate inertia (pitch)

Table 2: *Boolean decision variables. If $\delta = 1$, then the corresponding component/region is selected. If $\delta = 0$, the component/region is not selected.*

For the multirotor design optimization, the Boolean decision variables δ_i are defined as shown in Table 2 and the continuous variables x_i are defined as shown in Table 3. The complete mixed-integer state vector $\mathbf{x}$ introduced in eq. (9) is defined as follows:

$$\mathbf{x} \quad = \quad [\delta_1 \cdots \delta_{i_3}, x_1 \cdots x_{j_{12}}]^T \tag{11}$$

The indices i and j of discrete and continuous variables are summarized in Table 4. Three different MILPs are created. One with the number of actuators constrained to $N_a = 4$, one with $N_a = 6$ and one with $N_a = 8$. The total number of discrete and continuous variables varies between the three programs and is equal to 1795, 2444 and 3091, respectively. To avoid multiplication of several free variables in the constraints, it was decided to create three separate MILPs rather than creating one optimization program with N_a as a free variable. To find the optimal solution, the three MILPs are run separately and the solution with the lowest objective function value is chosen.

In the following, the complete set of constraints used in the multicopter design optimization is listed. The constraints make use of the rules 1, 4, 19a, 19b and the

Variables	**Description**
x_1	Total weight
x_2	Weight of actuator (motor+prop.)
x_3	Total price
x_4	Force to lift
x_5	n^2
x_6	n^3
x_7	Length of frame
x_8	Weight of frame
x_9	Actuator roll inertia
x_{10}	Actuator pitch inertia
x_{11}	Frame roll inertia
x_{12}	Frame pitch inertia
x_{13}	Total roll inertia
x_{14}	Total pitch inertia
x_{15}	Actuator rotational inertia $J_m + J_p$
x_{16}	Actuator rise-time constant
x_{17}	Propeller power
x_{18}	Current consumption
x_{19}	Required Ah
x_{20}	Battery Ah
x_{21}	1 / Flight Time
$x_{22} \cdots x_{25}$	n^2 for each propeller type
$x_{26} \cdots x_{28}$	Set of motor inertias J_m
$x_{29} \cdots x_{32}$	Set of propeller inertias J_p
$x_{33} \cdots x_{36}$	Set of battery Ah's
$x_{37} \cdots x_{40}$	Set of power values
$x_{41} \cdots x_{44}$	Set of current values
$x_{45} \cdots x_{48}$	Set of 1 / flight times
$x_{49} \cdots x_{96}$	Set of time-constants
$x_{97} \cdots x_{j_1}$	Distance from cg perp. to roll axis
$x_{j_1+1} \cdots x_{j_2}$	Distance from cg perp. to pitch axis
$x_{j_2+1} \cdots x_{j_3}$	Distance^2 from cg perp. to roll axis
$x_{j_3+1} \cdots x_{j_4}$	Distance^2 from cg perp. to pitch ax.
$x_{j_4+1} \cdots x_{j_5}$	Set of actuator roll inertias
$x_{j_5+1} \cdots x_{j_6}$	Set of actuator pitch inertias
$x_{j_6+1} \cdots x_{j_7}$	Set of frame roll inertia
$x_{j_7+1} \cdots x_{j_8}$	Set of frame pitch inertia
$x_{j_8+1} \cdots x_{j_9}$	Set of frame lengths
$x_{j_9+1} \cdots x_{j_{10}}$	Actuator Distance^2 Roll **z**-set
$x_{j_{10}+1} \cdots x_{j_{11}}$	Actuator Distance^2 Pitch **z**-set
$x_{j_{11}+1} \cdots x_{j_{12}}$	RPM **z**-set for n^3, calc. from n^2

Table 3: *Continuous variables used in the design optimization. "cg perp. to" = "center of gravity perpendicular to".*

Index	$N_a = 4$	$N_a = 6$	$N_a = 8$
i_1	241	241	241
i_2	561	721	881
i_3	881	1201	1521
j_1	98	99	100
j_2	100	102	104
j_3	102	105	108
j_4	104	108	112
j_5	106	111	116
j_6	108	114	120
j_7	110	117	124
j_8	112	120	128
j_9	114	123	130
j_{10}	274	283	290
j_{11}	594	763	930
j_{12}	914	1243	1570

Table 4: *Optimization indices for $N_a = 4$, 6 and 8 actuators.*

nonlinear function approximation defined earlier in this section.

Boolean variable constraints:

$$0 \leq \delta_i \leq 1 \tag{12}$$

$$\sum_{i=1}^{4} \delta_i = 1 \qquad \sum_{i=5}^{7} \delta_i = 1 \tag{13}$$

$$\sum_{i=8}^{11} \delta_i = 1 \qquad \sum_{i=12}^{i_1} \delta_i = 1 \tag{14}$$

$$\sum_{i=i_1+1}^{i_2} \delta_i = 1 \qquad \sum_{i=i_2+1}^{i_3} \delta_i = 1 \tag{15}$$

Total weight and price:

$$x_2 = N_a \left(\sum_{i=1}^{4} (m_{p,i}\delta_i) + \sum_{i=5}^{7} (m_{m,i}\delta_i) \right) \tag{16}$$

$$x_1 = x_2 + x_8 + \sum_{i=8}^{11} (m_{b,i}\delta_i) \tag{17}$$

$$x_3 = N_a \left(\sum_{i=1}^{4} (P_{p,i}\delta_i) + \sum_{i=5}^{7} (P_{m,i}\delta_i) \right) + \sum_{i=8}^{11} (P_{b,i}\delta_i) \tag{18}$$

Individual propeller thrust and n^2:

$$x_4 = (x_1 + m_L)\,g \quad (19)$$

$$x_{22} = \frac{1}{N_a \rho C_{t,1} D_1^4}\delta_1 x_4 \quad (20)$$

$$\vdots$$

$$x_{25} = \frac{1}{N_a \rho C_{t,4} D_4^4}\delta_4 x_4 \quad (21)$$

$$x_5 = \sum_{i=22}^{25} x_i \quad (22)$$

Frame length and weight:

$$x_{j_8+1} = L_f(D, N_a)\delta_1 \quad (23)$$

$$\vdots$$

$$x_{j_9} = L_f(D, N_a)\delta_4 \quad (24)$$

$$x_7 = \sum_{i=j_8+1}^{j_9} x_i \quad (25)$$

$$x_8 = m_F N_a x_7 \quad (26)$$

Rotational inertia of motors and propellers:

$$x_{26} = J_{m,1}\delta_5 \quad (27)$$

$$\vdots$$

$$x_{28} = J_{m,3}\delta_7 \quad (28)$$

$$x_{29} = J_{p,1}\delta_1 \quad (29)$$

$$\vdots$$

$$x_{32} = J_{p,4}\delta_4 \quad (30)$$

$$x_{15} = \sum_{i=26}^{32} x_i \quad (31)$$

Time constants:

$$F_{req} = (N_a(m_{p,1} + m_{m,1}) + m_{b,1} + m_L)\frac{g}{N_a} \quad (32)$$

$$n = \sqrt{\frac{F_{req}}{C_{t,1}\rho D_1^4}} \quad (33)$$

$$D_\omega = \frac{\rho C_{p,1} n^3 D_1^5}{(2\pi n)^2} \quad (34)$$

$$x_{49} = \delta_1\delta_5\delta_8 \left(\frac{R_{a,1}(J_{m,1} + J_{p,1})}{R_{a,1}D_\omega + K_{t,1}^2}\right) \quad (35)$$

$$\vdots$$

$$F_{req} = (N_a(m_{p,4} + m_{m,3}) + m_{b,4} + m_L)\frac{g}{N_a} \quad (36)$$

$$n = \sqrt{\frac{F_{req}}{C_{t,4}\rho D_4^4}} \quad (37)$$

$$D_\omega = \frac{\rho C_{p,4} n^3 D_4^5}{(2\pi n)^2} \quad (38)$$

$$x_{96} = \delta_4\delta_7\delta_{11} \left(\frac{R_{a,3}(J_{m,3} + J_{p,4})}{R_{a,3}D_\omega + K_{t,3}^2}\right) \quad (39)$$

$$x_{16} = \sum_{i=49}^{96} x_i \quad (40)$$

$$x_{16} \leq \tau_{max} \quad (41)$$

It should be noted here that the time-constants calculated in eqs. (35) and (39) are estimates based on the linear transfer function in eq. (8). If real measurements of the time-constants were available for each combination of battery, motor and propeller, these time-constants would be used in $x_{49} \cdots x_{96}$ instead of the estimates based on the datasheet information.

Calculate square of roll & pitch lengths: These constraints use the nonlinear function approximation rules. The corresponding Boolean variables are $\{\delta_{i_1+1}, \cdots \delta_{i_2}\}$ and $\{\delta_{i_2+1}, \cdots \delta_{i_3}\}$. The corresponding z-value sets are $\{x_{j_9+1}, \cdots x_{j_{10}}\}$ and $\{x_{j_{10}+1}, \cdots x_{j_{11}}\}$.

$$x_{j_2+1} = x_{97}^2 \quad (42)$$

$$\vdots$$

$$x_{j_3} = x_{j_1}^2 \quad (43)$$

$$x_{j_3+1} = x_{j_1+1}^2 \quad (44)$$

$$\vdots$$

$$x_{j_4} = x_{j_2}^2 \quad (45)$$

Actuator Roll Inertia ($m \cdot R^2$): The factor 2 in eq. (48), (51), (54) and (57) is used because there are two actuators per frame arm.

$$x_{j_4+1} = \delta_1\delta_5(m_{p,1}+m_{m,1})x_{j_2+1} \tag{46}$$

$$\vdots$$

$$x_{j_5} = \delta_4\delta_7(m_{p,4}+m_{m,3})x_{j_3} \tag{47}$$

$$x_9 = 2\sum_{i=j_4+1}^{j_5} x_i \tag{48}$$

Actuator Pitch Inertia ($m \cdot R^2$):

$$x_{j_5+1} = \delta_1\delta_5(m_{p,1}+m_{m,1})x_{j_3+1} \tag{49}$$

$$\vdots$$

$$x_{j_6} = \delta_4\delta_7(m_{p,4}+m_{m,3})x_{j_4} \tag{50}$$

$$x_{10} = 2\sum_{i=j_5+1}^{j_6} x_i \tag{51}$$

Frame Roll Inertia:

$$x_{j_6+1} = \frac{1}{3}L_f(D_1,N_a)m_F\delta_1 x_{j_2+1} \tag{52}$$

$$\vdots$$

$$x_{j_7} = \frac{1}{3}L_f(D_4,N_a)m_F\delta_4 x_{j_3} \tag{53}$$

$$x_{11} = 2\sum_{i=j_6+1}^{j_7} x_i \tag{54}$$

Frame Pitch Inertia:

$$x_{j_7+1} = L_f(D_1,N_a)m_F\delta_1 x_{j_3+1} \tag{55}$$

$$\vdots$$

$$x_{j_8} = L_f(D_4,N_a)m_F\delta_4 x_{j_4} \tag{56}$$

$$x_{12} = 2\sum_{i=j_7+1}^{j_8} x_i \tag{57}$$

Total Roll & Pitch Inertias:

$$x_{13} = x_9 + x_{11} \tag{58}$$

$$x_{14} = x_{10} + x_{12} \tag{59}$$

Calculation of n^3: This constraint uses the nonlinear function approximation rules. The corresponding Boolean variables are $\{\delta_{12}, \cdots \delta_{i_1}\}$. The corresponding z-value set is $\{x_{j_{11}+1}, \cdots x_{j_{12}}\}$.

$$x_6 = x_5^{1.5} \tag{60}$$

Propeller Power:

$$x_{37} = C_p\rho D_1^5\delta_1 x_6 \tag{61}$$

$$\vdots$$

$$x_{40} = C_p\rho D_4^5\delta_4 x_6 \tag{62}$$

$$x_{17} = \sum_{i=37}^{40} x_i \tag{63}$$

Battery Ah:

$$x_{33} = B_{1,Ah}\delta_8 \tag{64}$$

$$\vdots$$

$$x_{36} = B_{4,Ah}\delta_{11} \tag{65}$$

$$x_{20} = \sum_{i=33}^{36} x_i \tag{66}$$

Current Consumption:

$$x_{41} = \frac{N_a}{B_{1,V}}\delta_8 x_{17} \tag{67}$$

$$\vdots$$

$$x_{44} = \frac{N_a}{B_{4,V}}\delta_{11} x_{17} \tag{68}$$

$$x_{18} = \sum_{i=41}^{44} x_i \tag{69}$$

Calculate Ah:

$$x_{19} = \left(\frac{T_F}{60}\right)x_{18} \tag{70}$$

$$x_{20} = x_{19} \tag{71}$$

Calculate the Inverse of Flight Time = A / Ah:

$$x_{45} = \frac{x_{18}}{B_{1,Ah}}\delta_8 \tag{72}$$

$$\vdots$$

$$x_{48} = \frac{x_{18}}{B_{4,Ah}}\delta_{11} \tag{73}$$

$$x_{21} = \sum_{i=45}^{48} x_i \tag{74}$$

4 Experimental Results

The purpose of the experiments presented in this section is to validate the models and assumptions used in the mixed-integer optimization presented in section 3. The experimental setup is shown in Fig. 4. The validation presented in this paper has chosen both the 63% rise time of the actuator thrust as well as the total flight time as the parameters to compare against the simulation model in Fig. 2. The rise times are measured directly from step responses in thrust, while the total flight time is estimated via the measured current when the step responses have reached steady-state.

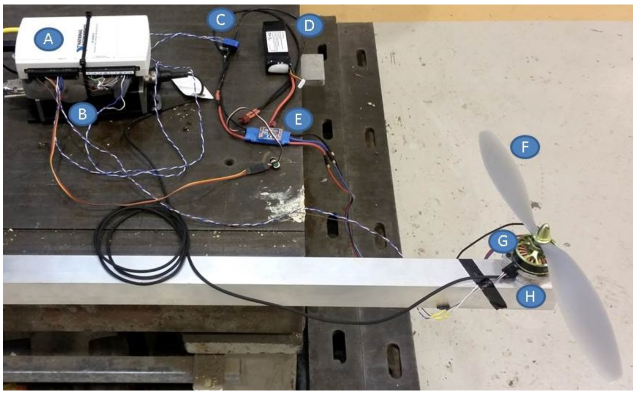

Figure 4: *Experimental setup for validation. A: Digital/analog IO card, B: Load cell amplifier, C: Current sensor, D: Battery pack, E: Motor controller, F: Propeller, G: Motor, H: Load cell.*

Three different optimization test cases were studied, as summarized below:

- I: Longest possible flying time, no payload
- II: Longest possible flying time, 1.5kg payload
- III: Fastest possible motor response, no payload

The design optimization was constrained to use either 4, 6 or 8 actuators in the multicopter. Section 4.1 contains a summary of the different components (datasheets) available for the design optimization. The experimental results for each test case are summarized in sections 4.2, 4.3 and 4.4.

4.1 Summary of Datasheets

The available datasheets are summarized in Table 5, 6 and 7. In total three motors (see Fig. 5), four propellers (see Fig. 6) and four batteries were available (the same as in the design optimization procedure in section 3, Table 2). Since all the batteries have a supply voltage of 11.1V and the experiments focused on validation of the 63% rise-time constant and flight time estimated via the measured current, the same battery could be used in all the tests, see Fig. 7. The selection of battery in the experiments did not have an impact on the estimated time-constants and total flight time.

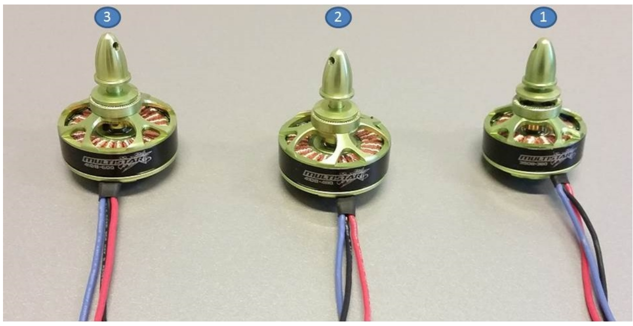

Figure 5: *Three different motors used in the design optimization and validation tests.*

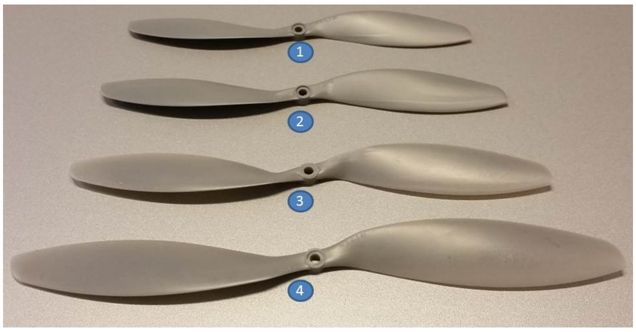

Figure 6: *Four different propellers used in the design optimization and validation tests.*

Figure 7: *Battery used in the design validation tests.*

Motor [#]	**1**	**2**	**3**
Model	4108-380KV Turnigy	4108-480KV Turnigy	4108-600KV Turnigy
$K_{t,i}$	380	480	600
$R_{a,i}$ (Ω)	0.222	0.148	0.123
$m_{m,i}$ (g)	111	111	111
$i_{max,i}$ (A)	17	22	26
$P_{max,i}$ (W)	360	380	400
$D_{m,i}$ (m)	0.047	0.047	0.047
$S_{l,i}$ (m)	0.012	0.012	0.012
$S_{w,i}$ (m)	0.004	0.004	0.004
$\rho_{m,i}$ (kg/m^3)	6800	6800	6800
$P_{m,i}$ ($)	31.36	31.36	31.36

Table 5: *Datasheets: Motors.* $m_{m,i}$ *is the motor weight,* $D_{m,i}$ *is the motor diameter,* $S_{l,i}$,$S_{w,i}$ *are the shaft length and width,* $\rho_{m,i}$ *is the material density of the motor and* $P_{m,i}$ *is the motor price.*

Battery [#]	**1**	**2**	**3**	**4**
$m_{b,i}$ (g)	309	618	1236	308
$B_{i,Ah}$ (mAh)	4000	8000	16000	3300
$B_{i,V}$ (V)	11.1	11.1	11.1	11.1
$B_{i,imax}$ (A)	100	200	400	105
$P_{b,i}$ ($)	25.5	51.0	102.0	26.7

Table 6: *Datasheets: Batteries.* m_b *is the weight of the battery, while* P_b *is the price.*

4.2 Test Case I

In this test case the design was optimized for the longest possible flying time and no payload. As seen in Table 8, propeller 4 was chosen when the design was constrained to use 4 and 8 actuators, while propeller 3 was chosen with 6 actuators. Motor 3 was chosen with 4 and 6 actuators, while motor 2 was chosen with 8 actuators. Battery 3 was chosen regardless of the number of actuators being 4, 6 or 8. Battery 3 has the highest capacity (16000 mAh), but also the highest weight. The selection of the battery with the highest capacity is not obvious. The solution with 4 actuators has both the longest flight time and the lowest price, so it is the prefered design in Test Case I.

Fig. 8 illustrates the experimental results from Test Case I. The black curves show the simulated thrust response when using the Simulink model in Fig. 2 with either 4, 6 or 8 actuators. The red curves show the measured thrust response under acceleration, while the green curves show the measured curves under deceleration. Note that the green curves are inverted about the steady-state value for easier comparison with the acceleration and the simulated response. The circles in the figure represent the values at the 63% rise time. The experimental results confirm that a linear model assumption is appropriate for the multicopter actuator, consisting of battery pack, controller, motor and propeller as shown in Fig. 4. Tables 11 and 12 summarize

Propeller [#]	**1**	**2**	**3**	**4**
$m_{p,i}$ (g)	12	14	18	25
$D_{p,i}$ (m)	0.254	0.2794	0.3048	0.3556
$C_{t,i}$ (m)	4.7	4.7	4.7	4.7
$C_{p,i}$ (m)	0.1222	0.1156	0.1146	0.1027
p_i (deg)	10	11	12	14
$P_{p,i}$($)	4.7	5.0	5.6	7.8

Table 7: *Datasheets: Propellers.* $m_{p,i}$ *is the weight of the propeller,* p_i *is the pitch angle of the blade and* $P_{p,i}$ *is the propeller price.*

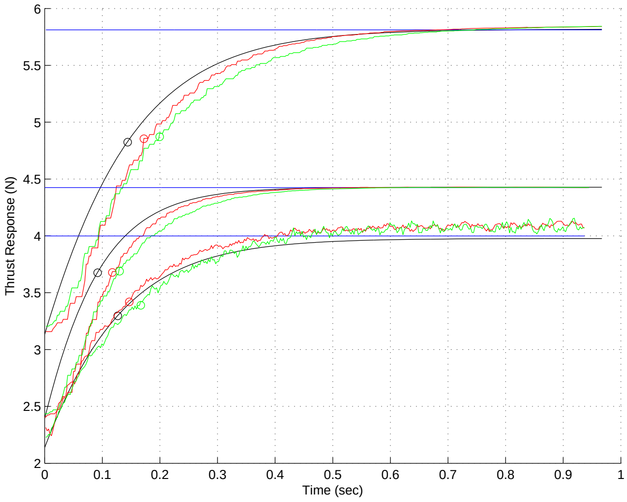

Figure 8: *Thrust response per propeller, Test Case I. Top: 4 Actuators, Middle: 6 Actuators, Bottom: 8 Actuators. Blue: Input, Red: Acceleration, Green: Deceleration, Black: Simulation model. The circles represent the thrust at the* 63% *rise time.*

the three different rise times (simulation, acceleration, deceleration). For Test Case I the simulated rise times are slightly faster than the measured values, ranging from 30 to 42ms faster.

Number of Actuators	**4**	**6**	**8**
Propeller chosen [#]	4	3	4
Motor chosen [#]	3	3	2
Battery chosen [#]	3	3	3
Time constant (ms)	161	165	134
Flight time (min)	62.5	60.0	55.5
Price	227.4	290.2	352.9

Table 8: *Optimization Results: Test Case I.*

4.3 Test Case II

In Test Case II the design was optimized for the longest possible flying time with a 1.5kg payload. As seen in Table 9, propeller 4 and battery 3 were chosen regardless of the number of actuators being 4,6 or 8. Motor 3 was chosen when the design was constrained to 4 and 6 actuators, while motor 1 was chosen with 8 actuators. The solution with 8 actuators has the longest flight time. However, the price with 8 actuators is significantly higher than the price for the solutions with 4 and 6 actuators.

Number of Actuators	**4**	**6**	**8**
Propeller chosen [#]	4	4	4
Motor chosen [#]	3	3	1
Battery chosen [#]	3	3	3
Time constant (ms)	150	156	123
Flight time (min)	25.4	27.1	27.3
Price	227.4	290.2	352.9

Table 9: *Optimization Results: Test Case II.*

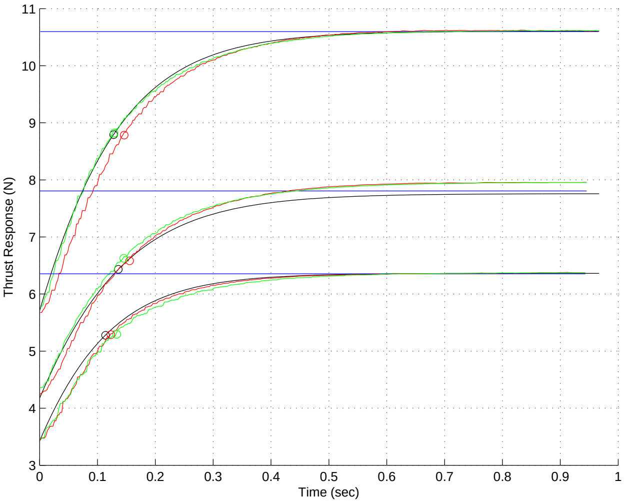

Figure 9: *Thrust response per propeller, Test Case II. Top: 4 Actuators, Middle: 6 Actuators, Bottom: 8 Actuators. Blue: Input, Red: Acceleration, Green: Deceleration, Black: Simulation model. The circles represent the thrust at the* 63% *rise time.*

Fig. 9 illustrates the experimental results from Test Case II. The results for Test Case II are better than for Test Case I. Tables 11 and 12 summarize the three different rise times (simulation, acceleration, deceleration). For Test Case II the simulated rise times are slightly faster than the measured values, ranging from 10 to 15ms faster. Overall, the match between the simulation model and the experiments is good.

4.4 Test Case III

In Test Case III the design was optimized for the fastest possible motor response and no payload. As seen in Table 10 propeller 1, motor 1 and battery 1 were chosen regardless of the number of actuators being 4,6 or 8. Since the time-constant (39ms) is the same for 4, 6 and 8 actuators, the natural choice is to use the solution with 4 actuators since the flight time is the highest

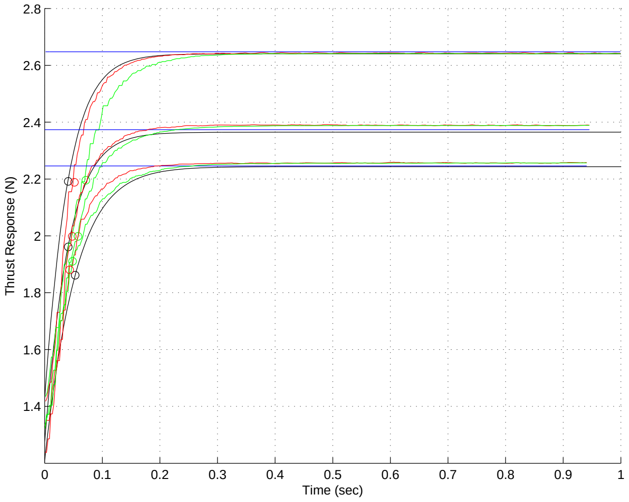

Figure 10: *Thrust response per propeller, Test Case III. Top: 4 Actuators, Middle: 6 Actuators, Bottom: 8 Actuators. Blue: Input, Red: Acceleration, Green: Deceleration, Black: Simulation model. The circles represent the thrust at the 63% rise time.*

Number of Actuators	**4**	**6**	**8**
Propeller chosen [#]	1	1	1
Motor chosen [#]	1	1	1
Battery chosen [#]	1	1	1
Time constant (ms)	39	39	39
Flight time (min)	37.2	29.4	23.8
Price	150.9	213.65	276.4

Table 10: *Optimization Results: Test Case III.*

and the price is the lowest for this choice. Fig. 10 illustrates the experimental results from Test Case III. The results for Test Case III are similar to Test Case II. Tables 11 and 12 summarize the three different rise times (simulation, acceleration, deceleration). For Test Case III the simulated rise times differ from the measured values by -21ms to +8ms.

Table 13 shows the differences between the time-constants found from the experiments and the estimates based on the transfer function in eq. (8) and also used in the optimization, eqs. (35), (39). The results are satisfactory with differences in the range -41 to +25ms. One error source is the time-constant assumption made in eq. (8) for a linear system.

The actual flight times can be estimated by dividing the battery capacity (Ah) by the measured current (A) when the step-responses have reached steady-state and by the number of actuators (N_a). Table 14 shows the estimated flight times vs. the (inverted) flight times calculated in eqs. (72)-(74). The standard deviation between measured and estimated flight times in Table 14 is 12.8%. Note that the battery voltage as stated in the datasheets is 11.1V, while the voltage when the battery is fully charged is more than 12V. In addition, the battery is not capable of keeping the voltage higher than 11.1V when discharged. Hence, both the estimated flight times and the ones found from experiments in Table 14 probably overstate the actual flight times slightly.

Test	$T_{a,4}$	$T_{d,4}$	$T_{a,6}$	$T_{d,6}$	$T_{a,8}$	$T_{d,8}$
I	172	200	118	130	147	167
II	146	129	156	146	123	133
III	52	72	48	58	43	48

Table 11: *Measured rise times (63%, in ms) for the different test cases and number of propellers. T_a is for acceleration, T_d is for deceleration.*

Test	T_4	ΔT_4	T_6	ΔT_6	T_8	ΔT_8
I	144	-42	92	-32	127	-30
II	128	-10	136	-15	114	-14
III	41	-21	41	-12	53	8

Table 12: *Simulated rise times (63%, in ms) for the different test cases and number of propellers. T_i is the rise time with i propellers. ΔT_i is the time difference between T_i and the average of $T_{a,i}$ and $T_{d,i}$.*

5 Conclusions

This paper has presented an optimization framework for multirotors based on mixed-integer programming. The framework allows for efficient selection of optimal

Test	T_4	ΔT_4	T_6	ΔT_6	T_8	ΔT_8
I	186	25	124	-41	157	23
II	138	-12	151	-5	128	-5
III	62	23	53	14	46	7

Table 13: *Time-constants found from experiments (mean value of $T_{a,i}$ and $T_{d,i}$ in Table 11) for the different test cases and number of propellers. ΔT_i is the time difference (in ms) between T_i and the time-constant estimates based on eq. (8).*

Test	$T_{e,4}$	$T_{o,4}$	$T_{e,6}$	$T_{o,6}$	$T_{e,8}$	$T_{o,8}$
I	66.6	62.5	57.5	60.0	62.0	55.5
II	29.4	25.4	28.5	27.1	33.2	27.3
III	33.9	37.2	25.3	29.4	20.5	23.8

Table 14: *Estimated flight times in minutes from the experiments $T_{e,i}$ vs. flight times calculated in the optimization $T_{o,i}$ for test cases I, II and III and 4, 6 and 8 actuators.*

combinations of components from a set of available datasheets. Nonlinear functions can be approximated by using a combination of discrete and continuous variables. The designs can be optimized towards a set of different criteria, such as flight time, power consumption or dynamic performance, depending on the designer’s preferences. Optimized designs with 4, 6 and 8 propellers are presented using real components available at, for example, HobbyKing (2015).

A simulation model of a multirotor actuator is presented and this model has been validated against experiments in three different test cases (longest possible flying time with or without payload, as well as fastest motor response). The experiments have used a step input in thrust and compared the 63% rise-time against the simulation model. The results are good with step-time differences between simulation and experiments in the range -21 to +42ms. The relatively small differences may be caused by unmodelled effects, such as motor friction and measurement delay. The difference between the estimated time-constants used in the design optimization and the time-constants estimated from the experiments have also been evaluated. The results are good with differences in the range -42 to +25ms. The total flight times have also been validated and have a standard deviation of 12.8% compared to the modeled flight times.

Overall, the results presented in this paper demonstrate that mixed-integer programming provides both a relatively accurate and efficient approach to multirotor design optimization from available datasheets. On an Intel i7-4770S 3.1GHz processor the different optimization problems were solved in typically 5-25 seconds using the IBM CPLEX solver. The experimental validation confirms that the modelling assumptions made in the design optimization formulation are reasonable and hence increase the confidence that the optimized designs actually meet the intended application’s requirement specifications.

References

Amazon. Prime Air. `http://www.amazon.com/b?node=8037720011`, 2015. Accessed: 2015-04-07.

Bemporad, A. and Morari, M. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 1999. 35(3):407–427. doi:10.1016/S0005-1098(98)00178-2.

Clausen, J. Branch and bound algorithms - principles and examples. `http://www.diku.dk/OLD/undervisning/2003e/datV-optimer/JensClausenNoter.pdf`, 1999. Accessed: 2015-04-07.

Hehn, M. and D’Andrea, R. A frequency domain iterative learning algorithm for high-performance, periodic quadrocopter maneuvers. *Mechatronics*, 2014. 24(8):954–965. doi:10.1016/j.mechatronics.2014.09.013.

HobbyKing. Online shop. `http://www.hobbyking.com`, 2015. Accessed: 2015-04-07.

IBM. CPLEX Optimizer. `http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer`, 2015. Accessed: 2015-04-07.

Karmarkar, N. A new polynomial-time algorithm for linear programming. *Combinatorica*, 1984. 4(4):373–395. doi:10.1007/bf02579150.

Magnussen, Ø., Hovland, G., and Ottestad, M. Multicopter UAV Design Optimization. In *Proc. IEEE/ASME Intl. Conf. on Mechatronic and Embedded Systems and Applications*. 2014. doi:10.1109/MESA.2014.6935598.

Mignone, D. The really big collection of logic propositions and linear inequalities. Technical Report AUT01-11, ETH Zurich, 2002.

Tyapin, I. and Hovland, G. Kinematic and Elastostatic Design Optimisation of the 3-DOF Gantry-Tau Parallel Kinematic Manipulator. *Modeling, Identification and Control*, 2009. 30(2):39–56. doi:10.4173/mic.2009.2.1.

Whitney, D. Optimum step size control for Newton-Raphson solution of nonlinear vector equations. *Automatic Control, IEEE Transactions on*, 1969. 14(5):572–574. doi:10.1109/TAC.1969.1099261.