Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Autonomous Systems Lab
Prof. Roland Siegwart

Master Thesis

Morphology Optimization of a Tilt-Rotor MAV

Spring Term 2018

Supervised by:
Karen Bodie
Zachary Taylor

Author:
Luca Rinsoz

Declaration of Originality

I hereby declare that the written work I have submitted entitled

Morphology Optimization of a Tilt-Rotor MAV

is original work which I alone have authored and which is written in my own words.[1]

Author(s)

Luca	Rinsoz

Student supervisor(s)

Karen	Bodie
Zachary	Taylor

Supervising lecturer

Roland	Siegwart

With the signature I declare that I have been informed regarding normal academic citation rules and that I have read and understood the information on 'Citation etiquette' (`https://www.ethz.ch/content/dam/ethz/main/education/rechtliches-abschluesse/leistungskontrollen/plagiarism-citationetiquette.pdf`). The citation conventions usual to the discipline in question here have been respected.

The above written work may be tested electronically for plagiarism.

Place and date	Signature

[1]Co-authored work: The signatures of all authors are required. Each signature attests to the originality of the entire piece of written work in its final form.

Contents

Abstract

Hier kommt der Abstact hin ...

Symbols

Symbols

ϕ, θ, ψ	roll, pitch and yaw angle
$\mathcal{F}_W$	inertial world frame
$\mathcal{F}_B$	inertial body frame
$\mathcal{F}_{P_i}$	i-th propeller frame
p	position of the MAV in $\mathcal{F}_W$
ω_B	angular velocity of the MAV in $\mathcal{F}_B$
${}^W R_B$	rotation matrix from $\mathcal{F}_B$ to $\mathcal{F}_W$
${}^B R_{P_i}$	rotation matrix from $\mathcal{F}_{P_i}$ to $\mathcal{F}_B$
$R_X(\gamma)$	canonical rotation matrix about the X axis of angle γ
$R_Y(\gamma)$	canonical rotation matrix about the Y axis of angle γ
$R_Z(\gamma)$	canonical rotation matrix about the Z axis of angle γ
α_i	i-th propeller tilt angle
w_i	i-th propeller rotation speed
τ_{ext_i}	i-th propeller counter rotation torque
T_i	i-th thrust
m	total mass of the MAV
I_B	body inertia of the MAV
n	MAV's number of propellers
L	MAV's arms length
κ_f	propeller thrust coefficient
κ_m	propeller drag coefficient
g	gravity constant
$c(\gamma)$	cosine of the angle γ
$s(\gamma)$	sine of the angle γ

Acronyms and Abbreviations

cw	clockwise
ccw	counterclockwise
ETH	Eidgenössische Technische Hochschule
GUI	Graphical User Interface
MAV	Micro Aerial Vehicle
ROS	Robotic Operating System

sqp	sequential quadratic programming
UAV	Unmanned Aerial Vehicle

Chapter 1

Introduction

Rotary wing micro aerial vehicles (MAVs) have been well studied in academia and found a lot of applications in the world such as search operations [1], photography [2] or even toys [3]. They encountered such a broad success because of their agility and mechanical simplicity. Nevertheless, traditional multi-rotor vehicles are under-actuated, which means that they cannot control their torque and force independently [4]. They are thus unable to change their position without changing their orientation.

Recently the focus has been on designing MAVs able to perform more complex tasks such as camera motion for the film industry [5] or bridge inspection where huge resources (i.e. cranes and large man-power) are needed. The ultimate goal would be for a drone to be able to interact with its surrounding and apply forces to it, in order to perform maintenance where human can not access, or to do construction work in harsh environments.

To perform these tasks, an MAV has to be able to hover in any orientation, and for a proper disturbance rejection while manipulating, the drone must have the potential to accelerate instantaneously in any direction. Hence, the MAV has to be able to decouple its orientation and position control. A drone that has a decoupled force and torque control is referred to as an omni-directional MAV.

The problem of overcoming the under-actuation and achieving omni-directionality is not straightforward. To address this problem, several MAV's designs have been presented over the past years. For instance, in [5], Voliro (name of the vehicle) is based on a traditional hexa-copter (see Figure 1.1). The omni-directionality issue is addressed by adding motors to rotate the thrusters around their arm axis, thus allowing a control not only of the thrust produced by each propeller, but also on the orientation of this thrust. This tilting rotor system allows for decoupling the control of position and orientation. By using a control scheme based on an allocation technique, the system provides very good maneuverability.

Figure 1.1: Voliro [5].

In [4], the Omnicopter (name of the vehicle) is described. It is a drone with eight fixed rotors and the drone shape is the result of a mathematical optimization (see Figure 1.2) which maximizes the vehicle's agility with the constraint that its dynamical properties would be as independent as possible on the vehicle orientation.

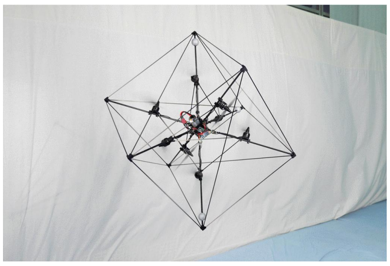


Figure 1.2: Ominicopter [4].

In [6], the MAV is a fixed propeller multi-rotor. The design is also the result of an optimization, which tends to minimize the body volume, maximize the controllability of the system, avoid eventual aerodynamic interactions and maximizes the efficiency in performing manipulation tasks.
The idea presented in [7] is a mix between Voliro and the Omnicopter because the design is a modified hexarotor (see Figure 1.3), which achieves full control over the vehicle's position and orientation using manually tiltable propellers. The paper also provides a methodology to optimize the fixed tilting angles depending on the desired trajectory.

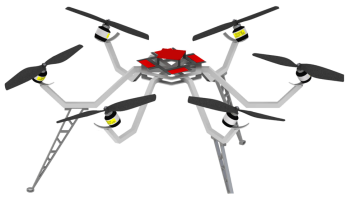


Figure 1.3: Hexacopter with manually tiltable rotors [7].

Yet, nothing in the literature is found about the morphology optimization of MAVs with tilting rotors. Hence the need for the present research project. The aim of this thesis is thus to design a morphology optimization problem for a tilt rotor MAV that accounts for the different factors that influence the morphology such as:

- Omni-directionality
- Flight efficiency
- Controlability

To reach this goal the chosen approach is to build an optimization engine that solves the optimization problem and returns different MAV designs. The most interesting designs are then tested in simulation.
In this report the methods used to build the optimization engine and to simulate the results are discussed. Afterwards, the results returned by the engine are shown and compared based on different criteria. Finally, the results gathered during the simulation phase are also covered.

Chapter 2

Method

As explained in Chapter 1 the aim of this work is to find a drone design that is the result of an optimization problem, which tends to maximize the MAV's omni-directionality, flight efficiency and controllability. To do so it is important to first state what are the parameters that define the design of an MAV. These parameters are defined as:

- β (angles formed by the arms with the horizontal plane see Figure 2.1)
- θ (angles formed by the arms in the horizontal plane see Figure 2.1)
- L (arm length)
- n (number of propeller)

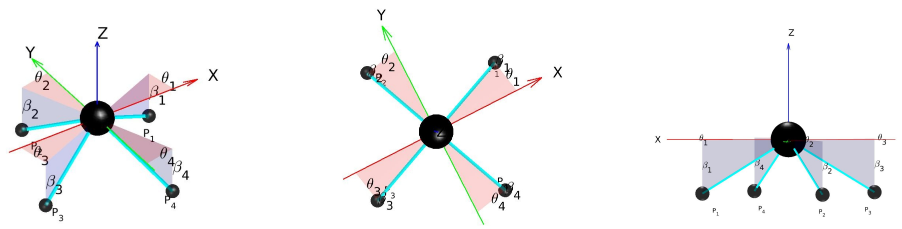


Figure 2.1: Quadcopter to illustrate the parameters that define the morphology of an MAV ($n = 4$, $\beta = [30, 30, 30, 30][°]$, $\theta = [22, 22, 22, 22][°]$, and $L = 0.4[m]$).

To solve the problem an optimization engine is developed with MATLAB®. This tool returns the aforementioned parameters along with other information on the corresponding MAV design. The interesting drone designs outputted by the tool are then simulated on Gazebo[1] and the control of the different models is achieved using a Robotic Operating System[2] (ROS) node.
This chapter first covers the theory needed to obtain a generalize mathematical model for a n-rotor MAV with an arbitrary morphology. Then, the optimization problem is defined. Afterwards, the optimization tool is described. In the end, the theoretical background needed to perform the simulations is covered.

[1]An open source robot simulator [8].
[2]An open source collection of software that help developers to create robot applications [9].

2.1 Modelisation of MAVs

In the following part, a dynamical model for a general design of MAV is presented. Such a modelisation is much needed to mathematiacally optimize the morphology of a MAV. This model is inspired from the models presented in [5] and [10].

Assumptions

In this model the first assumption is that the MAV is composed of n+1 rigid bodies: one for each propeller unit P_i and one for the body B. Then, it is considered that the thrust is produced by irreversible fixed-pitch motor-propeller actuators. Finally, only the aerodynamic forces and torques that are responsible for the MAV actuation are considered, all the second order effects and disturbances are neglected and also the airflow interactions between the different rotors are neglected.

Initial Definitions

In order to understand correctly the dynamical model, a few definitions are much needed. First, let us define $\mathcal{F}_W : \{O_W; X_W, Y_W, Z_W\}$ as the world fixed inertial frame and $\mathcal{F}_B : \{O_B, X_B, Y_B, Z_B\}$ as a moving frame attached to the MAV. Also, $\mathcal{F}_{P_i} : \{O_{P_i}; X_{P_i}, Y_{P_i}, Z_{P_i}\}, i = 1...n$ is the frame of the i-th propeller. The propeller rotate around the axis Z_{P_i}, and thus the thrust T_i is produced along this axis. The tilt movement of the rotors is a simple rotation around X_{P_i}. Now let ${}^W R_B$ be the orientation of the body frame with respect to the world frame and ${}^B R_{P_i}$ be the orientation of the i-th propeller with respect to the body frame. From there, it straightforward with the help of Figure 2.2 that

$$ {}^B R_{P_i} \; = \; R_Z\left((i-1)\frac{2\pi}{n}\right) R_Z(\theta_i) R_Y(\beta_i) R_X(\alpha_i), \; i = 1...n\,. \tag{2.1} $$

Equivalently, let

$$ {}^B O_{P_i} \; = \; R_Z\left((i-1)\frac{2\pi}{n}\right) R_Z(\theta_i) R_Y(\beta_i) \begin{bmatrix} L \\ 0 \\ 0 \end{bmatrix}, \; i = 1...n \tag{2.2} $$

be the origin of the i-th propeller frame $\mathcal{F}_{P_i}$. In Equation (2.1) and (2.2), $(i-1)\frac{2\pi}{n}$ is the angle that the i-th arm would form with axis X_B if the arms of the drone are evenly distributed in the horizontal plane, θ_i is the angle that i-th arm forms in the horizontal plane with respect to its evenly distributed position (see Figure 2.1), β_i is the angle that the i-th arm forms with the horizontal plane (see Figure 2.1), α_i is the tilting angles of the i-th propeller about the X_{P_i} axis, L is the arm length and n is the number of propellers.

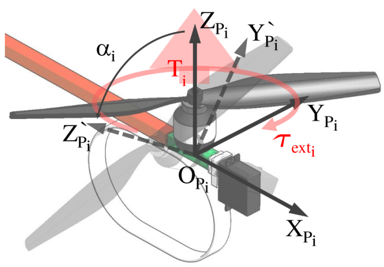


Figure 2.2: Representation of the i-th tilting arm [10].

Equations of motion

Using Newton-Euler formalism, the general equations of motion of the MAV are

$$\begin{cases} \dot{\omega}_B = I_B^{-1} \sum_{i=1}^{n} \left({}^{B}R_{P_i} \tau_{ext,i} + \tau_{Bi} \right), \\ \ddot{p} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \frac{1}{m} \, {}^{W}R_B \sum_{i=1}^{n} T_i \,. \end{cases} \tag{2.3}$$

Where

$$\tau_{Bi} = {}^{B}O_{P_i} \times {}^{B}R_{P_i} T_{P,i} \,, \tag{2.4}$$

$$\tau_{ext,i} = \begin{bmatrix} 0 & 0 & -c_i \kappa_m w_i^2 \end{bmatrix}^T \tag{2.5}$$

$$\begin{cases} c_i = 1, & \text{if } i \text{ is odd } (\mathit{cw\ rotation\ to\ produce} + \mathit{thrust}) \\ c_i = -1 & \text{if } i \text{ is even } (\mathit{ccw\ rotation\ to\ produce} + \mathit{thrust}) \end{cases}$$

and

$$T_i = {}^{B}R_{P_i} T_{Pi} \,, \quad T_{Pi} = \begin{bmatrix} 0 & 0 & \kappa_f w_i^2 \end{bmatrix}^T . \tag{2.6}$$

In Equation (2.3) g is the gravity constant, in Equation (2.5), κ_m is the propeller drag coefficient, in Equation (2.6) κ_f is the propeller thrust coefficient and in Equation (2.5) and (2.6) w_i is the i-th propeller rotation speed.
The force and torque that the drone produce in body frame $\mathcal{F}_B$ are

$$\begin{bmatrix} M_B \\ F_B \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} \left({}^{B}R_{P_i} \tau_{ext,i} + \tau_{Bi} \right) \\ \sum_{i=1}^{n} T_i \end{bmatrix} , \tag{2.7}$$

that can be rewritten

$$\begin{bmatrix} M_B \\ F_B \end{bmatrix} = A(\alpha) W \,. \tag{2.8}$$

Where $W = [w_1^2,\ w_2^2,\ ...,\ w_n^2]$ and

$$A(\alpha) = \begin{bmatrix} (-\kappa_f L s(\beta_1) c(\theta_1) + c_1 \kappa_m s(\theta_1)) s(\alpha_1) + (\kappa_f L s(\theta_1) + c_1 \kappa_m c(\theta_1) s(\beta_1)) c(\alpha_1) & ... \\ (-\kappa_f L s(\beta_1) s(\theta_1) - c_1 \kappa_m c(\theta_1)) s(\alpha_1) + (-\kappa_f L c(\theta_1) + c_1 \kappa_m s(\beta_1) s(\theta_1)) c(\alpha_1) & ... \\ (-L \kappa_f c(\beta_1)) s(\alpha_1) + (c_1 \kappa_m c(\beta_1)) c(\alpha_1) & ... \\ s(\theta_1) \kappa_f s(\alpha_1) + s(\beta_1) c(\theta_1) \kappa_f c(\alpha_1) & ... \\ -c(\theta_1) \kappa_f s(\alpha_1) + s(\beta_1) s(\theta_1) \kappa_f c(\alpha_1) & ... \\ c(\beta_1) \kappa_f c(\alpha_1) & ... \end{bmatrix},$$

is the $6 \times n$ allocation matrix and $c(\cdot)$ and $s(\cdot)$ represent the cosine and sine operator respectively.

Static allocation

The optimization engine has to compute the maximal reachable force and torque in a large number of direction. So to compute that in a reasonable times in [5] an approach to transform the non-linear allocation matrix into a static allocation matrix, which renders the problem of inverse kinematic linear. To do so, the system in Equation (2.8) is rewritten as

$$\begin{bmatrix} M_B \\ F_B \end{bmatrix} = A_{static} F_{dec} \,. \tag{2.9}$$

Where F_{dec} is the decomposed force vector defined as follow

$$F_{dec} = \begin{pmatrix} F_{h,1} \\ F_{v,1} \\ ... \\ F_{h,n} \\ F_{v,n} \end{pmatrix}, \tag{2.10}$$

with $F_{v,1} = \kappa_f cos(\alpha_i)$ the vertical force produced by the i-th propeller and $F_{h,1} = \kappa_f sin(\alpha_i)$ the horizontal force produced by the i-th propeller. And the static matrix defined as

$$A_{static} = \begin{bmatrix} -\kappa_f Ls(\beta_1)c(\theta_1) + c_1\kappa_m s(\theta_1) & +\kappa_f Ls(\theta_1) + c_1\kappa_m c(\theta_1)s(\beta_1) & ... & ... \\ -\kappa_f Ls(\beta_1)s(\theta_1) - c_1\kappa_m c(\theta_1) & -\kappa_f Lc(\theta_1) + c_1\kappa_m s(\beta_1)s(\theta_1) & ... & ... \\ -L\kappa_f c(\beta_1) & c_1\kappa_m c(\beta_1) & ... & ... \\ s(\theta_1)\kappa_f & s(\beta_1)c(\theta_1)\kappa_f & ... & ... \\ -c(\theta_1)\kappa_f & s(\beta_1)s(\theta_1)\kappa_f & ... & ... \\ 0 & c(\beta_1)\kappa_f & ... & ... \end{bmatrix},$$

a $6 \times 2n$ matrix that is invariant for a drone design. Using the Moore-Penrose pseudo inverse we can easily get the inverse kinematic as follow

$$F_{dec} = A^{\dagger}_{static} \begin{bmatrix} M_{des} \\ F_{des} \end{bmatrix}. \tag{2.11}$$

Which returns the decomposed force vector for a desired force and torque. Finding the tilting angles and propellers rotation speed required to attain this desired force and torque is then pretty straightforward

$$\begin{cases} w_i^2 = \frac{1}{\kappa_f}\sqrt{F_{v,i}^2 + F_{h,i}^2} \\ \alpha_i = atan2(F_{h,i}, F_{v,i}) \end{cases}. \tag{2.12}$$

2.2 Optimization problem

The following section focuses on the optimization problem that the engine has to solve in order to obtain a MAV design that is optimal. The criteria that make this design optimal are also discussed.

Problem statement

The optimization problem is stated as follow

$$\arg\max_x \quad f(x) \quad \text{subject to} \begin{cases} c(x) \leq 0 \\ ceq(x) = 0 \\ A \cdot x \leq 0 \\ Aeq \cdot x = 0 \\ lb \leq x \leq ub\,, \end{cases} \tag{2.13}$$

where $f(x)$ is the cost function, x the argument vector, $c(x)$ the non-linear inequality constraint vector, $ceq(x)$ the non-linear equality constraint vector, A the linear inequality constraint matrix, Aeq the linear equality constraint matrix, lb the lower bound vector of the arguments (x) and ub the upper bound vector.
Once the optimization problem solved, the output is the optimal argument vector x^* that maximizes the cost function $f(x)$. In our case the argument vector x is composed of the MAV's morphology parameters (β, θ, L, n) and cost functions are the subject of next section.

Cost Functions

As stated in Chapter 1 the aim of the project is to obtain a multi-rotor design that is omni-directional. Therefore, it is the heart of the problem to define meaningful cost functions for the optimization problem, which when solved would return parameters for an omni-directional drone. In this section the few cost functions that capture at best the omni-directionality are described.
The first, and also one of the more meaningful cost function consists in maximizing the minimal attainable force and the minimal attainable torque that the MAV can produce in any direction. It makes sense because on of the definition of omni-directionality is defined as the drone capacity to accelerate instantaneously in every directions. In order to do that the MAV has to have high minimal attainable force and torque, hence this cost function. It turns out this cost function is also computationally quite lighter than the other. Indeed, as when the multi-rotor apply a force or torque in the direction parallel to one of its arm, the propeller on this arm is perfectly unable to produce any force or torque in this direction. This is due to the fact that no matter what the tilting angle for this propeller is, the thrust it produces is parallel to the arm direction (see Figure 2.2). Therefore, the minimal attainable forces and torques for the drone are in the direction where it looses a propeller, i.e. the arm directions. So instead of optimizing the force and torque in a large number of direction, it is enough for this cost function to optimize the force and torque in n directions.
The second cost function consist in maximizing the minimal attainable force and the minimal attainable torque that the MAV can produce in any direction and minimizing the MAV's inertia. It is the same as the first cost function, but the last term is added in order to have an easier drone to control and thus put a criterion on the controllability.
The next cost function is designed to maximize the volume of the reachable force and torque space. The force and torque spaces are two polyhedron formed by the drone's attainable forces and torques in every directions (see Figure 2.4a and Figure 2.4b). The idea behind this cost function is to have the biggest task space for the drone and hence increase the MAV's ability to navigate in any orientation and to any position. This cost function is computationally heavy, because in order to have precise polyhedrons for the different spaces, the forces and torques has to be computed in 7490 directions.
After that a cost function that maximize the force, the torque and the hover efficiency in all directions. The aim of this cost function is to maximize the agility of the MAV for good disturbance rejection. Moreover, the term that maximizes the hover efficiency is designed to give the drone design the ability to perform manipulation efficiently in any orientation. Solving the optimization problem for this cost function can also be computationally heavy depending on how many directions you choose to represent "All directions".
The last cost function maximizes the force and the torque in one defined direction d. It is mostly designed to test the optimization engine as it is computationally light and given specific directions the optimal design is pretty straightforward. For instance if you maximize the force in the e_z direction for a 4-rotor MAV, the expected optimal solution would be a standard quadcopter. In the presented cost functions the multi-rotor model described in Section 2.1 is implemented in MATLAB® to compute the different forces and torques in the different directions.

Solver

In order to solve the previously described optimization problems, the tool uses MATLAB® function fmincon, with different algorithm. The one showing the quickest convergence and the best results being the sequential quadratic programming (sqp).

2.3 Optimization tool

User Guide

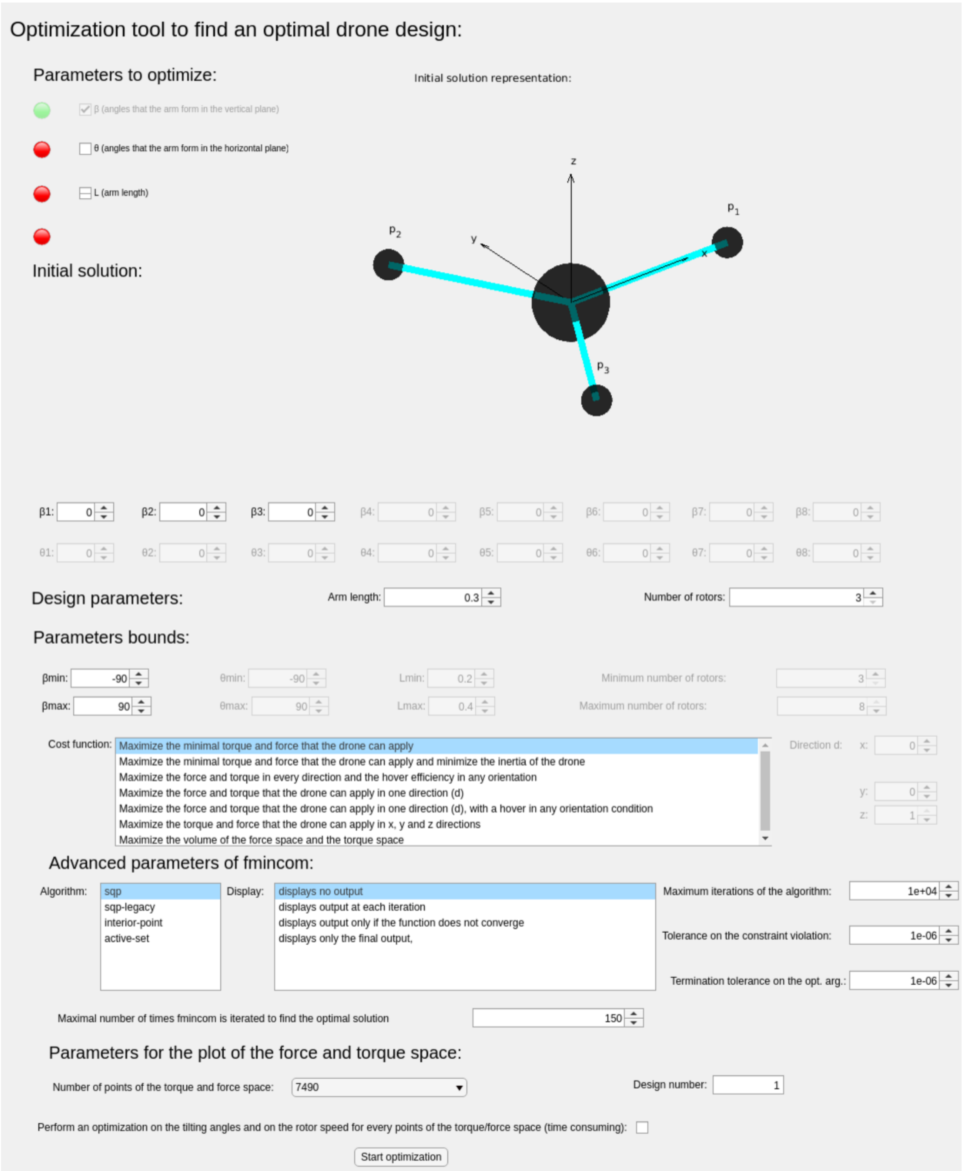
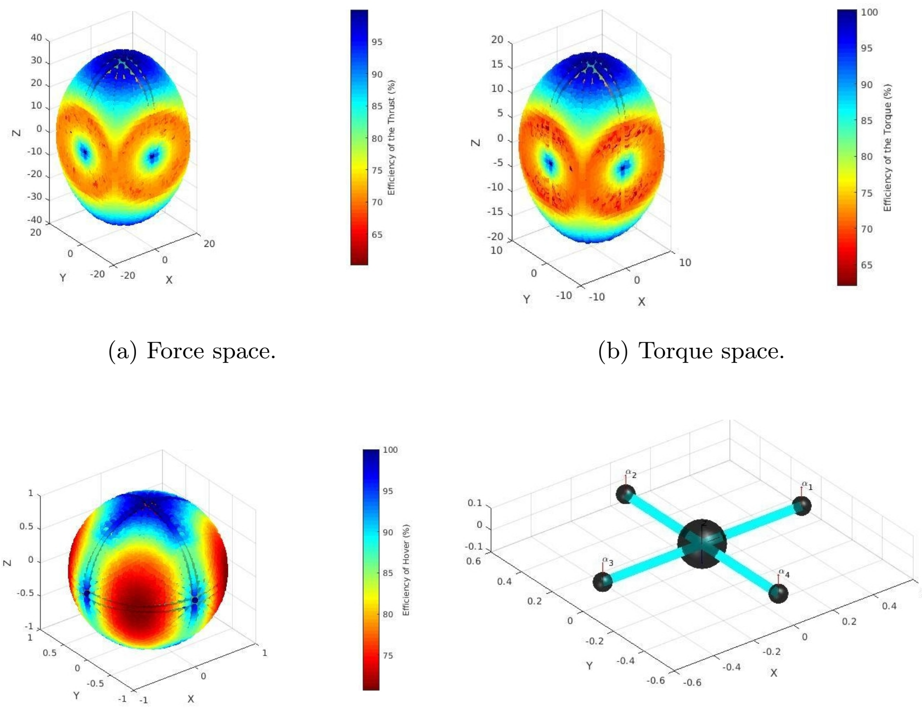


Figure 2.3: MAV morphology optimization tool GUI.

Outcome

Limitations

2.4 Simulation Approach



(a) Force space. (b) Torque space.

(c) Hover efficiency. (d) MAV representation.

Figure 2.4: Example of what is outputted by the optimization engine.

Chapter 3

Optimization Results

Show results produced by the engine.

3.1 Even Designs

3.1.1 Platonic Solids

3.1.2 Quad-copter

3.1.3 Hexa-copter

3.1.4 Octa-copter

3.2 Odd Designs

3.2.1 Tri-copter

Show tricopter.

3.2.2 Penta-copter

3.2.3 Hepta-copter

3.3 Comparison of Different Designs

$\cos(\beta) = \sqrt{(\frac{2}{3})} => \beta = 35.26°$
$F_{min} = 34.74, F_{max} = 42.55, M_{min} = 17.42, M_{max} = 21.34, H_{eff,min} = 81.65\%, H_{eff,max} = 100\%$
$F_{min} = 26.6, F_{max} = 52.11, M_{min} = 15.1, M_{max} = 26.13, H_{eff,min} = 75\%, H_{eff,max} = 100\%$
Design 1: $F_{min} = 23.18, F_{max} = 28.56, M_{min} = 11.61, M_{max} = 14.3, H_{eff,min} = 81.11\%, H_{eff,max} = 95.2\%$
Design 2: $F_{min} = 23.22, F_{max} = 28.37, M_{min} = 11.65, M_{max} = 14.23, H_{eff,min} = 81.65\%, H_{eff,max} = 94.73\%$
$F_{min} = 44.7, F_{max} = 58.8, M_{min} = 22.4, M_{max} = 29.5, H_{eff,min} = 81.78\%, H_{eff,max} = 96.65\%$
$F_{min} = 46.46, F_{max} = 56.73, M_{min} = 23.3, M_{max} = 28.45, H_{eff,min} = 81.64\%, H_{eff,max} = 94.77\%$

Table 3.1: Comparison between the different number of propellers.

MAV Design	$F_{min}[N]$	$F_{max}[N]$	$F_{mean}[N]$	$M_{min}[Nm]$	$M_{max}[Nm]$	$M_{mean}[Nm]$	$H_{eff,mean}[\%]$
Tri-copter	17.17	21.21	17.95	8.61	10.64	9	85.46
Quad-copter	23.22	28.37	26.87	11.65	14.23	13.47	87.1
Penta-copter	28.95	35.46	29.4	14.52	17.78	14.74	85.35
Hexa-copter	34.74	42.55	39.52	17.42	21.34	19.82	88.9
Hepta-copter	39.96	49.44	47.2	20.04	24.8	23.66	91.1
Octa-copter	44.7	58.8	53.95	22.4	29.48	27.06	91.42

Chapter 4

Simulation Results

Evaluate results in simulation.

4.1 Hexa-copter

4.2 Hepta-copter

4.3 Octa-copter

Chapter 5

Conclusion

5.1 Summary/Achieved

5.2 Improvements

5.3 Further Developement

Bibliography

[1] M. Silvagni, A. Tonoli, E. Zenerino, and M. Chiaberge, “Multipurpose UAV for search and rescue operations in mountain avalanche events,” *Geomatics, Natural Hazards and Risk*, vol. 8, no. 1, pp. 18–33, Jan. 2017.

[2] DJI Mavic Pro & Mavic Pro Platinum – Every Creative Moment – DJI. https://www.dji.com/mavic. Accessed: 2018-09-03.

[3] Aura. https://aura-drone.com/us/. Accessed: 2010-09-30.

[4] D. Brescianini and R. D’Andrea, “Design, modeling and control of an omni-directional aerial vehicle,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 3261–3266.

[5] M. Kamel, S. Verling, O. Elkhatib, C. Sprecher, P. Wulkop, Z. Taylor, R. Siegwart, and I. Gilitschenski, “Voliro: An Omnidirectional Hexacopter With Tiltable Rotors,” *arXiv:1801.04581 [cs]*, Jan. 2018, arXiv: 1801.04581.

[6] A. Nikou, G. C. Gavridis, and K. J. Kyriakopoulos, “Mechanical design, modelling and control of a novel aerial manipulator,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 4698–4703.

[7] S. Rajappa, M. Ryll, H. H. Bülthoff, and A. Franchi, “Modeling, control and design optimization for a fully-actuated hexarotor aerial vehicle with tilted propellers,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 4006–4013.

[8] “Gazebo,” http://wiki.ros.org/ROS/Tutorials, (Online: accessed September 3, 2018).

[9] “ROS/Tutorials - ROS Wiki,” http://wiki.ros.org/ROS/Tutorials, (Online: accessed August 12, 2018).

[10] M. Ryll, H. H. Bülthoff, and P. R. Giordano, “Modeling and control of a quadrotor UAV with tilting propellers,” in *2012 IEEE International Conference on Robotics and Automation*, May 2012, pp. 4606–4613.

Appendix A

UML: Activity Diagram