Design of free-flyer robots for in-space cooperative additive manufacturing

João Luís Bruno Vale

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisors: Prof. Rodrigo Martins de Matos Ventura
Prof. Marco Alexandre de Oliveira Leite

Examination Committee

Chairperson: Prof. João Fernando Cardoso Silva Sequeira
Supervisor: Prof. Rodrigo Martins de Matos Ventura
Member of the Committee: Dr. Plinio Moreno Lopez

November 2021

Declaration

I declare that this document is an original work of my own and that it fulfils all the requirements of the Code of Conduct and Good Practices of the Universidade Lisboa.

Acknowledgments

I would like to thank Rodrigo Ventura, the supervisor of this thesis, for all the readiness and help granted. A big thanks to Alexandre Rocha, for designing ACROBAT's model in CAD, and providing assistance with any CAD related issue. I also would like to show my gratitude to Ana Custodio for kindly providing the code for the Direct Multisearch for Multiobjective Optimization algorithm.

I wish also to show gratitude to my fellow dissertation peers for the help and support through this last step, but also all my colleagues through out the five years of Técnico.

Last but not least, and this goes without saying, a big thanks to all my family and friends.

Thank you all.

Abstract

The growing need for larger assets in Space, ranging from commercial space stations to exploration beyond Earth orbit, will require the capability of manufacturing and assembling large structures in micro-gravity. Considering the use of autonomous space robots to perform these tasks, this thesis studies the optimal design of a free-flyer robot for mobile manipulation with an emphasis on assembly and additive manufacturing. We propose a robot architecture comprising a dexterous six Degrees of Freedom (DoF) parallel manipulator attached to a free-flyer robot body endowed with six DoF propulsion. The design methodology decouples the parallel manipulator from the robot body. For the parallel manipulator, we define performance metrics for workspace volume and accuracy. We employ multi-criteria optimization to determine the geometric parameters which best tradeoff defined metrics. For the robot body propulsion, we find which geometries result in maximum thrust and torque along all directions, in both force and torque space, thus maximizing maneuverability. The construction of a Ground based prototype is studied and efforts towards its realization were conducted.

Keywords

Space robotics; In-orbit manufacturing and assembly; Mobile manipulation; Multi-objective optimization.

Resumo

A crescente necessidade de maiores activos no espaço, desde estações espaciais comerciais a exploração alêm da orbita da Terra, vão necessitar da capacidade de construção e montagem de grandes estrutras em micro-gravidade. Considerando o uso de robôs autonomos para realizar estas tarefas, esta tese propõe o desenho otimo de robôs para a manipulação móvel, com uma enfase em montagem e construção aditiva. Para tal, propomos uma arqutectura composta uma um manipulador paralelo com seis graus de liberdade ligado a um robô capaz de movimentos com seis graus de liberdade. A metodologia de desenho descopula o manipulador paralelo do corpo do robô: Para o manipulador paralelo, defenimos métricas do volume do espaço de trabalho e de exatidão. Usamos otimização multi-objectiva para determinar os parametros geométricos que melhor trocam as métricas definidas. Para o sistema de propulsão do robô, encontramos as geometricas que melhor maximizam o impulso e binário em todas as direcções, de modo a maximizar a manobrabilidade. A construção de um prototipo terreste é estudada e são realizados esforços para a sua realização.

Palavras Chave

Robotica Espacial; Montagem e manufactura em órbita; Manipulação móvel; Otimização multi-objectiva.

Contents

1 Introduction 1

1.1 The ACROBAT's architecture . . . 3

1.2 Objectives . . . 5

1.3 Outline of the approach . . . 5

1.4 Contributions . . . 5

1.5 Document's Organization . . . 6

2 Background 7

2.1 Parallel Manipulators . . . 8

2.1.1 Inverse Kinematics . . . 8

2.1.2 Direct Kinematics . . . 8

2.1.3 Velocity Analysis . . . 9

2.1.4 Workspace . . . 9

2.2 Performance Metrics . . . 10

2.2.1 Workspace Volume . . . 10

2.2.2 Global Conditioning Index . . . 10

2.2.3 Global Sensitivity Index . . . 11

2.3 Free-Flying Robots . . . 12

2.3.1 Propulsion Model . . . 12

2.4 Multi-Objective Optimization . . . 14

2.5 Related Work . . . 15

2.5.1 Additive Manufacturing by Mobile Robots . . . 15

2.5.2 Indoor Space Free-Flying Robots . . . 15

2.5.3 Manipulator Dimensioning and Optimization . . . 16

3 Manipulator 17

3.1 Design Parameters . . . 18

3.2 Inverse Kinematics . . . 19

3.3 Direct Kinematics . . . 21

3.4 Workspace . . . 22
3.4.1 Visualization . . . 22
3.5 Velocity Analysis . . . 24
3.6 Evaluation . . . 24
3.6.1 Visualization . . . 26
3.6.2 Computation . . . 27
3.7 Design Space . . . 28
3.7.1 Bounds . . . 28
3.7.2 Constraints . . . 29
3.8 Optimization . . . 30
3.9 Results . . . 31

4 Robot Body **35**
4.1 Parameterization . . . 36
4.1.1 ACROBAT's parameterization . . . 37
4.2 Evaluation . . . 37
4.3 Design Space . . . 38
4.4 Optimization . . . 39
4.5 Results . . . 40

5 Simulation and Prototype **43**
5.1 Simulator Implementation . . . 44
5.2 Prototype Overview . . . 45
5.3 Manipulator . . . 45
5.3.1 Actuation System . . . 46
5.4 Propulsion System . . . 48
5.4.1 Brush-less DC motors, ESC and propellers . . . 48
5.4.2 Test-bench . . . 49
5.4.3 Ducted fan . . . 50
5.4.4 Calibration . . . 51

6 Conclusion and Future Work **53**
6.1 Future Work . . . 54

Bibliography **55**

List of Figures

1.1 Space technologies. 2
1.2 3D printing technologies. 3
1.3 ACROBAT Computer-Aided Design (CAD) model. 4
1.4 CAD model of ACROBAT's components. 4
1.5 System adapted for additive manufacturing. 5

2.1 Examples of parallel manipulators. Figure 2.1(a) and fig. 2.1(b) taken form [1] and fig. 2.1(c) taken from [2] 8
2.2 Notation of a single propeller, relative to the body's CoM. 13
2.3 Examples of mobile robots capable of additive manufacturing, in multiple applications and environments. 15
2.4 Example of free-flying robots designed for micro-gravity habitats. 16

3.1 Parallel manipulator schematic. 18
3.2 ACROBAT's manipulator constant orientation workspace. The color scheme goes as follows: The red patch is the mobile platform, the green patch the fixed platform, the magenta and blue line-segment are rigid links of size h and d, respectively. The black solid represents the manipulator's workspace. 23
3.3 ACROBAT's manipulator workspace slices with the respective displacement sensitivity, with $\boldsymbol{R} = \boldsymbol{I}$. The first row plot the translational displacement sensitivity and the bottom one the rotational displacement sensitivity. 26
3.4 Plots showing how the algorithm performs with a increasing value of N. 28
3.5 Diagram of link pose when $\alpha_k = 0$ for $k \in \{1, 2, 6\}$, based on fig. 3.1. The black patch represents the fixed base, connecting all anchors $\boldsymbol{B_k}$. The blue lines represent the R-S links, starting at $\boldsymbol{B_k}$ and ending at $\boldsymbol{H_k}$. 30
3.6 Point-wise Pareto Front Approximation. The red diamond represents ACROBAT's first iteration manipulator. 32

3.7 Joint representation of ACROBAT's first iteration manipulator and the optimized manipulator. The color scheme goes as follows: The red patch is the mobile platform, the green patch the fixed platform, the magenta and blue line-segment are rigid links of size h and d, respectively. 33

4.1 ACROBAT's first iteration robot body propeller placement. 36
4.2 $\boldsymbol{r_i}$ and $\boldsymbol{u_i}$ description relative to the body's Center of Mass (CoM). 37
4.3 Point wise pareto-front approximation. 40
4.4 Representative diagram of the robot body. 41

5.1 ACROBAT's first iteration implemented in Webots. 44
5.2 CAD depiction of a proposed experiment. 45
5.3 Six-RSS Manipulator with a mounted extrusion head. 46
5.4 ACROBAT's manipulator actuation components. 47
5.5 ACROBAT's prototype mounted on a stand with two ducted fans installed. 48
5.6 Propulsion system components. 49
5.7 RCbenchmark Series 1580 Test Stand mounted with a ducted fan and propeller. 50
5.8 Ducted fan with the propeller installed. d is the ducted fan diameter. 50
5.9 Bar graph comparing the K_1 and K_2 of different ducted fan diameters. 51
5.10 Cropped screenshot from BLHeli Configurator. 52
5.11 Thrust data taken from each propeller. Each point presents the average of 10 samples, and the curve is a second order polynomial fit. 52

List of Tables

3.1 Design parameters ACROBAT's first iteration manipulator. 19
3.2 Direct Multisearch for Multiobjective stopping criteria settings. 31
3.3 Direct Multisearch for Multiobjective initialization, cache and search step settings. 31
3.4 Direct Multisearch for Multiobjective centre selection, direction and step size settings . . . 31
3.5 Design parameters and performance of ACROBAT's first iteration manipulator and the optimized manipulator taken from fig. 3.6. 33

4.1 ACROBAT's first iteration robot body geometric parameters, with $\phi = \arctan(\sqrt{2})$. 37
4.2 Average and standard deviation of each design parameter relative to propeller position. . 40
4.3 Average and standard deviation of each design parameter relative to thrust direction. . . 40

5.1 Manipulator material. 46

Acronyms

AM	Additive Manufacturing
CW	Clockwise
CAD	Computer-Aided Design
CCW	Counter Clockwise
CoM	Center of Mass
DoF	Degree of Freedom
ESC	Electronic Speed Controller
FFF	Fused Filament Fabrication
GCI	Global Conditioning Index
GTSI	Global Translation Sensitivity Index
GRSI	Global Rotation Sensitivity Index
GPIO	General Purpose Input/Output
ISS	International Space Station
JAXA	Japan Aerospace Exploration Agency
LCI	Local Conditioning Index
MOO	Multi-Objective Optimization
NASA	National Aeronautics and Space Administration
ROS	Robot Operating System
RPM	Rotations Per Minute
UAV	Unmanned Aerial Vehicles
PLA	Polylactic Acid
PETG	Polyethylene Terephthalate Glycol-modified
PWM	Pulse Width Modulation

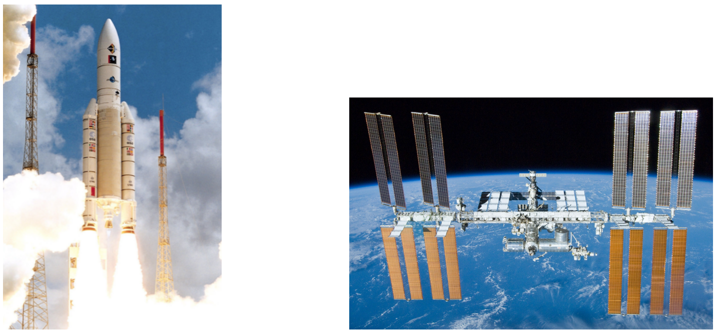
1

Introduction

Contents

1.1 The ACROBAT's architecture 3
1.2 Objectives 5
1.3 Outline of the approach 5
1.4 Contributions 5
1.5 Document's Organization 6

Current space launching systems are rocket-propelled, like the Ariane 5 depicted in fig. 1.1(a), strongly limit space exploration. Indeed, the harsh launch conditions and the limitations in payload volume and weight impose hefty constraints in mission design. The integration of in-orbit robotized manufacturing and assembly of parts, vehicles and structures could be advantageous to mission design, reducing the logistic requirements. Additionally, in-orbit robotized manufacturing and assembly allow missions dependent on structures, vehicles and parts otherwise constrained by current launch systems. The scope of in-orbit robotized manufacturing and assembly is large, enabling the in-orbit construction of large structures, namely, next-generation telescopes, large heat shields to land on Mars, habitats composed of multiple modules similar to the International Space Station (ISS) depicted in fig. 1.1(b), solar arrays to power various missions or solar sails to reach the outer solar system. One of the key technologies for in-orbit robotized manufacturing and assembly is Additive Manufacturing (AM), also known as 3D printing. The integration of AM in mission design also enables robust and adaptive systems that can adapt to unanticipated circumstances, mitigates mission risk and allows material recycling [3].



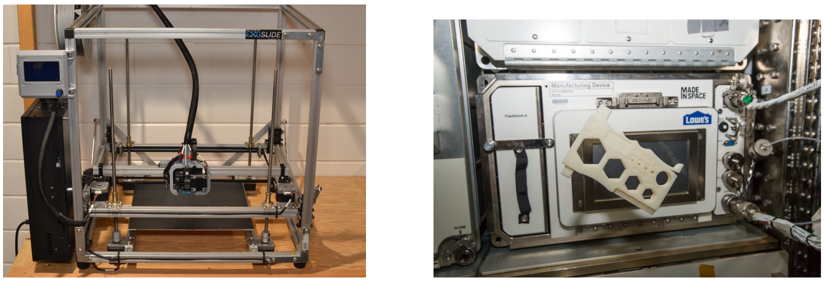
(a) Ariane 5 - Example of a Space Launch Vehicle.

(b) International Space Station.

Figure 1.1: Space technologies.

In recent years AM has found a role as a suitable candidate for part manufacturing in many industries and applications [4]. Existing AM systems are based on a gantry system, depicted in fig. 1.2(a), limiting the size of the printed object. If the needed part does not fit the system's work-space, it must manufactured and then assembled. This can pose a few problems: the part's designer must take work-space constraints into account, limiting the design capabilities. Other problem is that the mechanisms that allow part assembly can weaken the part's structural integrity. In other words, the development of mobile robots capable of AM can increase the effectiveness and range of applications of 3D printing.

In space, one of the first achievements in AM development and integration was made in 2014 by National Aeronautics and Space Administration (NASA), in partnership with Made In Space, Inc., launch-



(a) 3D Printer Gantry System. **(b)** Made In Space, Inc. ISS 3D Printer.

Figure 1.2: 3D printing technologies.

ing the first 3D printer, depicted in fig. 1.2(b), to the ISS. Tests conducted using the fused Fused Filament Fabrication (FFF) process concluded that the empirical data did not suggest an engineering impact of micro-gravity in the material outcome [5]. However, the fully autonomous coordination, control and design of mobile robots capable of AM, without being restricted to the work-space of a gantry system, is still largely a open problem, involving the cooperation of multiple spacecraft. To develop such system, the design of spacecraft dexterous enough for the material deposition process, while maintaining the requirements of autonomy and accuracy is of high importance. With cooperative in-orbit manufacturing and assembly tasks in mind, this thesis proposes the design of ACROBAT: an aerial free-flying robot for pressurized micro-gravity environments. The nature of its propulsion system disallows the use of said system in vacuum. Instead, the intent is for this to be a stepping stone towards the design and development of spacecraft.

1.1 The ACROBAT's architecture

Figure 1.3 shows a Computer-Aided Design (CAD) model of the first iteration of ACROBAT. The first iteration was designed to be a Ground Prototype. It can be decoupled into two main components:

1. A robot body equipped with six propellers, placed such that the kinematics are holonomic, meaning the robot can freely move with six DoF. Considering the target environment, propellers fans were chosen instead of other alternatives, such as cold gas thrusters. This is because they are simpler, faster to prototype, and easier to integrate into the design. The robot body will also house the main computer, batteries and required electronics and sensors (e.g IMU, camera). The CAD model of the robot body is depicted in fig. 1.4(a), and it was conceptualized to have a distance between two adjacent propellers of 19cm.
2. In isolation, the robot body has unbounded motion, but a low actuation bandwidth. In other words,

ACROBAT can have any position within the environment but the propeller's spin-up and spin-down time constrain the robot's fine movement capability which is detrimental for the targeted tasks. To enhance ACROBAT's manipulation capabilities, we attached a six DoF six Revolute-Spherical-Spherical (RSS) parallel manipulator, similar to HEXA [2] to the robot body, with a task dependent tool as the end-effector. Its CAD model can be seen in fig. 1.4(b).

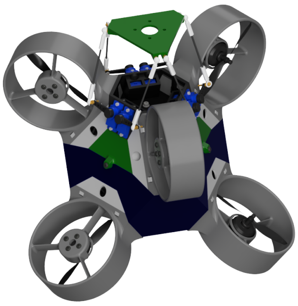
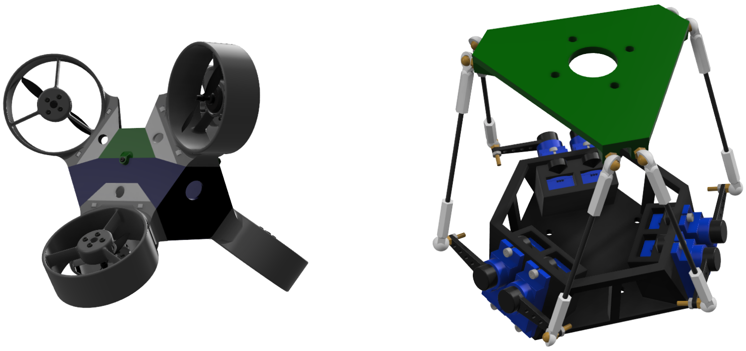


Figure 1.3: ACROBAT CAD model.



(a) Robot Body. **(b)** Six RSS Manipulator.

Figure 1.4: CAD model of ACROBAT's components.

ACROBAT was designed to be modular, simplifying an iterative design and test approach. Figure 1.5 presents an adaptation for additive manufacturing. Two types of robots compose this example: a platform robot, equipped with a deposition platform as end-effector; and an extrusion robot, equipped with an extrusion head for material deposition as end-effector. Another possible adaptation is using ACROBAT for fine manipulation, by equipping the end-effector with a gripper.

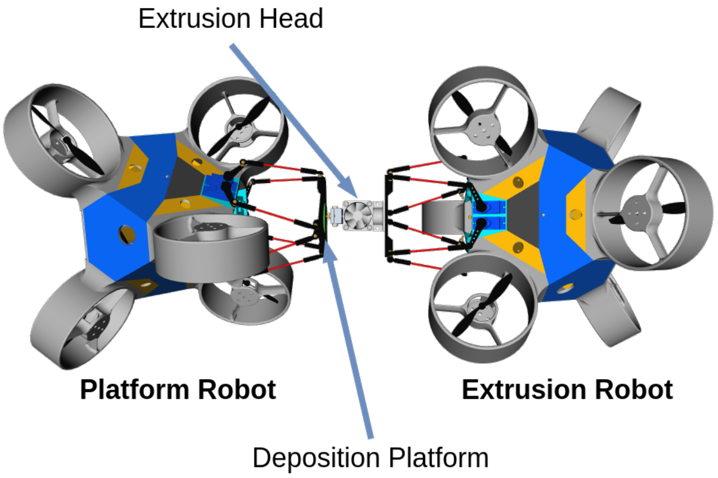


Figure 1.5: System adapted for additive manufacturing.

1.2 Objectives

The goal of this thesis is to optimize the robotic system described in section 1.1 to AM. Specifically, the objective of this thesis is to answer the question ”How should one dimension the proposed robotic system?”. To do so, we will decouple the manipulator from the robot body and treat each as an individual system.

A prototype will be built and tested on Earth, using an air bearing table to simulate micro-gravity. Given that the prototype is being designed to be tested on Earth, space-proofing the prototype, namely, radiation shielding, heat isolation or long distance communication are out of the scope of this thesis.

1.3 Outline of the approach

To achieve the objectives stated in section 1.2, we will decouple the manipulator from the robot body and treat each as an individual system. To design the manipulator, we wish to maximize both workspace volume and accuracy. In the case of the robot body, we will look to maximize the maximum force and torque possible in any direction. We will also compare the optimized designs with ACROBAT’s first iteration presented in section 1.1.

1.4 Contributions

Besides this document, a few contributions were made during the development of this thesis. First, a methodology was presented for the design of parallel manipulators and thrust actuated rigid bodies. An implementation of ACROBAT in a realistic physics simulator was done, to be used as a test-bench for future algorithms. Lastly, efforts were conducted in the construction and development of ACROBAT's prototype.

The developed design methodologies were presented in the AeroBest 2021 conference, which took place in July 21-23, 2021 in Lisbon, Portugal. Aerobest is an ECCOMAS Thematic Conference on Multidisciplinary Design Optimization of Aerospace Systems. Our submission was titled *A multi-objective optimization approach to the design of a free-flyer space robot for in-orbit manufacturing and assembly* [6] and a presentation to the audience was given.

1.5 Document's Organization

This thesis will begin with a motivation and introduction in chapter 1. The fundamentals behind this thesis will be discussed in chapter 2. To understand how this thesis fares in the state-of-art, similar works will also be presented in chapter 2. Given the outline described in section 1.3, we will divide the ACROBAT's design into two parts: the manipulator's design, discussed in chapter 3 and the robot's body design, discussed in chapter 4. Chapter 5 will discuss ACROBAT's implementation in a realistic physics simulator and will also address the efforts of building the Earth prototype. To conclude, chapter 6 will wrap up the main achievements and address the future work.

2

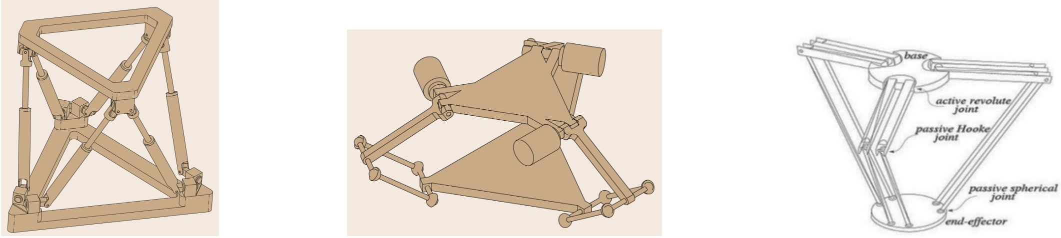
Background

Contents

2.1 Parallel Manipulators 8
2.2 Performance Metrics 10
2.3 Free-Flying Robots 12
2.4 Multi-Objective Optimization 14
2.5 Related Work 15

2.1 Parallel Manipulators

A Parallel Manipulators can be defined as closed-loop mechanism composed of a fixed base and an end effector connected by at least two independent kinematic chains [7]. Examples of successful parallel manipulator designs are the Gough-Stewart Platform [8] depicted in fig. 2.1(a), the Delta Platform [9] depicted in fig. 2.1(b) and the HEXA [2] depicted in fig. 2.1(c). Compared with serial manipulators, parallel manipulators generally have higher load capability and accuracy but reduced operational workspace.



(a) Gough-Stewart Platform [8].

(b) Delta Platform [9].

(c) HEXA [2].

Figure 2.1: Examples of parallel manipulators. Figure 2.1(a) and fig. 2.1(b) taken form [1] and fig. 2.1(c) taken from [2]

2.1.1 Inverse Kinematics

The inverse kinematics problem consists in computing the vector of actuated joints $\boldsymbol{\alpha}$ for a given pose vector $\boldsymbol{p}$ of the end effector

$$\boldsymbol{\alpha} = f(\boldsymbol{p}). \tag{2.1}$$

In closed-loop manipulators, the relationship expressed in eq. (2.1) is usually easy to find and there's a closed-form expression available [7].

2.1.2 Direct Kinematics

The direct kinematics problem consists in finding the vector of end effector pose $\boldsymbol{p}$ for a given vector of actuated joints $\boldsymbol{\alpha}$

$$\boldsymbol{p} = g(\boldsymbol{\alpha}). \tag{2.2}$$

Unlike the inverse kinematics, the direct kinematics problem is usually harder to solve, requiring solving a system of non-linear equations. In the case of 6 Degree of Freedom (DoF) manipulators, it is common for a closed form expression for the general direct kinematic equation to be unavailable, requiring the use of numerical methods [7].

2.1.3 Velocity Analysis

The kinematic Jacobian $\boldsymbol{J}$ of a parallel manipulator is defined as a linear relation between the actuated joint velocities vector $\dot{\boldsymbol{\alpha}}$ and the end effector velocities vector, often called the twist $\boldsymbol{w} = \begin{bmatrix}\boldsymbol{v}^T & \boldsymbol{\omega}^T\end{bmatrix}^T$ [7], where $\boldsymbol{v}$ is the linear velocity and $\boldsymbol{\omega}$ the angular velocity. The kinematic Jacobian is pose variant and is defined as

$$\boldsymbol{w} = \boldsymbol{J}(\boldsymbol{p})\dot{\boldsymbol{\alpha}}. \tag{2.3}$$

For manipulators with 6 DoF, the relationship of the derivatives of the orientation representation vector $\dot{\phi}$ and the manipulator's angular velocity $\boldsymbol{\omega}$ is given by [7]

$$\boldsymbol{\omega} = \boldsymbol{H}(\phi)\dot{\boldsymbol{\phi}}, \tag{2.4}$$

where $\boldsymbol{H} \in \mathbb{R}^{3\times 3}$ is dependent on the rotation convention used in $\boldsymbol{p}$. On the other hand, we can safely establish that the linear velocity $\boldsymbol{v}$ is the derivative of the pose's translational component $\boldsymbol{t}$ ($\boldsymbol{v} = \dot{\boldsymbol{t}}$). Equation (2.3) may be rewritten in terms of dimensionally homogeneous arrays, leading to

$$\boldsymbol{v} = \boldsymbol{J}_t\dot{\boldsymbol{\alpha}} \qquad\qquad \boldsymbol{\omega} = \boldsymbol{J}_r\dot{\boldsymbol{q}} \tag{2.5}$$

Consequently, $\boldsymbol{J} = [\boldsymbol{J}_t^T \boldsymbol{J}_r^T]^T$.

2.1.4 Workspace

The workspace $\mathcal{W}$ of a manipulator can be defined as the set of $\boldsymbol{p}$ the end effector can take while satisfying the constraints. Let's first assume that the manipulator kinematic state can be fully described by $\boldsymbol{p}$, which is usually the case with parallel manipulators. Generally, three types of constraints restrict parallel manipulators:

1. Constraints imposed by the actuators;
2. Constraints imposed by the passive joints;
3. Constraints imposed by the mechanical interference of links.

These constraints can be written as a system of kinematic equations

$$n(\boldsymbol{p}) = 0 \tag{2.6}$$

and kinematic inequalities

$$h(\boldsymbol{p}) \leq 0 \tag{2.7}$$

imposed by the workspace constraints. A manipulator workspace can be defined in multiple ways: The reachable workspace $\mathcal{W}_R \in \mathbb{R}^6$ is the set of $\boldsymbol{p}$ that the end-effector can reach with at least one orientation $\mathcal{W}_R = \{\boldsymbol{p} : g(\boldsymbol{p}) = 0, h(\boldsymbol{p}) \leq 0\}$. The constant orientation works-space $\mathcal{W}_C \in \mathbb{R}^3$ is a subset of $\mathcal{W}_R$ where $\boldsymbol{p}$ has constant orientation.

2.2 Performance Metrics

Performance metrics can be defined as metrics which evaluate the manipulator in a given set of attributes. Performance metrics are of high importance for manipulator design and dimensioning, enabling comparison between designs and configurations. Performance metrics can be classified as Local, if they are dependent on the manipulator's pose or Global, if they evaluate the characteristics of the whole manipulator's workspace. A comprehensive literature survey of performance metrics can be found in [10].

2.2.1 Workspace Volume

Workspace volume is a metric used to measure the size of a workspace. Workspace volume dimension depends on the type of workspace that is being measured. For instance, if the constant orientation workspace is considered, the workspace volume is a 3D volume. Formally, $V_{\mathcal{W}}$ is given by

$$V_{\mathcal{W}} = \int_{\mathcal{W}} d\mathcal{W} \tag{2.8}$$

where $\mathcal{W}$ is the targeted workspace.

2.2.2 Global Conditioning Index

The condition number k is a measure of the degree of independence of the manipulator's Jacobian matrix and used as a measure of the manipulator's dexterity [1, 10], first introduced in [11]. It is also considered to measure how close $\boldsymbol{p}$ is to a singularity [10].

Assuming $\dot{\boldsymbol{p}}$ and $\dot{\boldsymbol{q}}$ have the same physical units, a unit ball in joint space is mapped by $\boldsymbol{J^{-1}}$ into an ellipsoid whose semi-axes are the singular values of $\boldsymbol{J}$ [1]. In other words, $\boldsymbol{J}$ deforms the unit ball in the joint space into an ellipsoid in the end-effector pose space. Given that dexterity is defined as the ability to move and apply forces in every direction with equal ease, the dexterity increases as the Jacobian deformation decreases. A measure of the Jacobian imposed deformation is the condition number. If $\boldsymbol{J}$ is full rank, the condition number is defined as

$$k = \frac{\sigma_{max}}{\sigma_{min}} \tag{2.9}$$

where σ_{max} and σ_{min} are the maximum and minimum singular values of $\boldsymbol{J}$. A computationally simpler way of calculating the condition number using the Frobenius norm [10] is given as

$$k(\boldsymbol{J}) = \frac{1}{n}\sqrt{tr(\boldsymbol{J^T J})}\sqrt{tr[(\boldsymbol{J^T J})^{-1}]}. \tag{2.10}$$

The condition number can become infinity if, for example, σ_{min} reaches zero. To overcome this difficulties, the Local Conditioning Index (LCI) is defined as the reciprocal of k: $\frac{1}{k}$. The LCI is a local metric because it depends on $\boldsymbol{p}$ and it is bounded LCI $\in [0, 1]$. Higher values of LCI represent more dexterous manipulators. If the LCI gets close to zero, $\boldsymbol{J}$ is badly conditioned and the manipulator could fall into a singular configuration.

The LCI evaluates the dexterity in a single pose. To describe the overall dexterity over the entire workspace, the Global Conditioning Index (GCI) is introduced. It is given by

$$\text{GCI} = \frac{\int_{\mathcal{W}} k^{-1}(\boldsymbol{J})\, d\mathcal{W}}{\int_{\mathcal{W}} d\mathcal{W}} \qquad \text{GCI} \in [0, 1], \tag{2.11}$$

being a metric that must be maximized.

2.2.3 Global Sensitivity Index

A measure of accuracy must evaluate how a small displacement of the active joints $\delta\boldsymbol{\alpha}$ translates into a displacement of the end-effector pose $\delta\boldsymbol{p}$. These small active joint displacements may originate from the sensor noise, calibration errors or even thermal expansion and compression. Actuator displacement can be modeled by the kinematic Jacobian [7] such that

$$\delta\boldsymbol{p} = \boldsymbol{J}(\boldsymbol{p})\delta\boldsymbol{\alpha}. \tag{2.12}$$

Considering a actuator displacement bounded by a hyper-cube of side two

$$\|\delta\boldsymbol{\alpha}\|_\infty \leq 1, \tag{2.13}$$

if all actuated joints share a type (e.g. all actuated joints are revolute), the maximum displacement in translation $\sigma_{p,\infty}$ and the maximum displacement in rotation $\sigma_{r,\infty}$ is given by [12]

$$\sigma_{p,\infty} = \|\boldsymbol{J}_t\|_\infty \qquad \sigma_{r,\infty} = \|\boldsymbol{J}_r\|_\infty \tag{2.14}$$

where $\boldsymbol{J}_t$ and $\boldsymbol{J}_r$ are blocks of $\boldsymbol{J}$ such that $\boldsymbol{J} = [\boldsymbol{J}_t^T \boldsymbol{J}_r^T]^T$, obeying

$$\boldsymbol{v} = \boldsymbol{J}_t \dot{\boldsymbol{\alpha}} \qquad\qquad \boldsymbol{\omega} = \boldsymbol{J}_r \dot{\boldsymbol{\alpha}}. \tag{2.15}$$

Design wise, $\sigma_{p,\infty}$ and $\sigma_{r,\infty}$ measure translational and rotational sensitivity to actuator displacement and are representative of manipulator accuracy.

The aforementioned metrics only evaluate the performance in a given pose. To evaluate the sensitivity of the manipulator over the entire workspace, we propose the Global Translation Sensitivity Index (GTSI) and the Global Rotation Sensitivity Index (GRSI), given by

$$\text{GTSI} = \frac{\int_{\mathcal{W}} \sigma_{t,\infty} \, d\mathcal{W}}{\int_{\mathcal{W}} d\mathcal{W}} \qquad\qquad \text{GRSI} = \frac{\int_{\mathcal{W}} \sigma_{r,\infty} \, d\mathcal{W}}{\int_{\mathcal{W}} d\mathcal{W}}. \tag{2.16}$$

2.3 Free-Flying Robots

A Free-Flying robot can be defined as a vehicle capable of navigating and manoeuvring space with six DoF. The proposed propulsion model will assume the existence of an atmosphere, targeting Aerial vehicles actuated by propellers.

2.3.1 Propulsion Model

A single propeller i, rigidly attached to the body frame $\mathcal{B}$ coincident with the Center of Mass (CoM), produces a thrust $\boldsymbol{F_i}$ and torque $\boldsymbol{M_i}$ on the vehicle body while rotating at speed n_i (in revolutions per second) [13]. $\boldsymbol{F_i}$ results directly from the propeller thrust

$$\boldsymbol{F_i} = f_i \boldsymbol{u_i} \qquad\qquad f_i = K_1 \left| n_i \right| n_i \tag{2.17}$$

where $\boldsymbol{u_i}$ is a unit vector aligned with the propeller's axis of rotation and K_1 is a propeller constant. $\boldsymbol{M_i}$ is caused by the propeller non-central thrust $\boldsymbol{F_i}$ and reaction torque τ_i

$$\boldsymbol{M_i} = \boldsymbol{r_i} \times \boldsymbol{F_i} - \tau_i \boldsymbol{u_i} \qquad\qquad \tau_i = w_i \, K_2 \left| n_i \right| n_i \tag{2.18}$$

where $\boldsymbol{r_i}$ is the propeller position relative to $\mathcal{B}$, w_i is, for a positive or forward thrust, -1 if the propeller rotates clockwise or 1 if it rotates anti-clockwise and K_2 is another propeller constant.

Propeller constants K_1 and K_2 are given by

$$K_1 = \rho D^4 C_t \qquad\qquad K_2 = \frac{\rho D^5}{2\pi} C_P \tag{2.19}$$

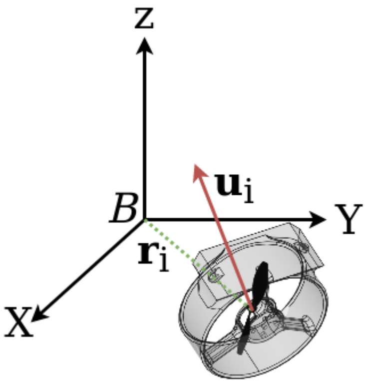


Figure 2.2: Notation of a single propeller, relative to the body's CoM.

where ρ is air density, D is propeller diameter, C_T is the thrust coefficient and C_P the power coefficient. C_T and C_P are blade dependent coefficients [14]. We assume that the propeller has similar K_1 and K_2 for both directions of rotation.

Consider an actuation signal vector $\boldsymbol{q}$, with i-th actuation signal following $q_i = |n_i|n_i$, resulting in

$$\begin{bmatrix} \boldsymbol{F_i} \\ \boldsymbol{M_i} \end{bmatrix} = \boldsymbol{a_i} q_i \tag{2.20}$$

where

$$\boldsymbol{a_i} = \begin{bmatrix} K_1 \boldsymbol{u_i} \\ K_1 \boldsymbol{r_i} \times \boldsymbol{u_i} - w_i K_2 \boldsymbol{u_i} \end{bmatrix}. \tag{2.21}$$

To combine the effect of N propellers, resulting force $\boldsymbol{F}$ and torque $\boldsymbol{M}$ are given by the sum of (2.20), which can be written in matrix form as

$$\begin{bmatrix} \boldsymbol{F} \\ \boldsymbol{M} \end{bmatrix} = \begin{bmatrix} \boldsymbol{a_1} & \dots & \boldsymbol{a_6} \end{bmatrix} \begin{bmatrix} q_1 \\ \vdots \\ q_6 \end{bmatrix} = \boldsymbol{A}\boldsymbol{q} \tag{2.22}$$

where $\boldsymbol{A}$ is the actuation matrix. If $\boldsymbol{A}$ is a square matrix, meaning the robot is neither under nor over actuated, and $\boldsymbol{A}$ is full rank, meaning all propellers are non-redundant, $\boldsymbol{A}$ is invertible. As a result, it is possible to establish the following relationship

$$\boldsymbol{q} = \boldsymbol{A}^{-1} \begin{bmatrix} \boldsymbol{F} \\ \boldsymbol{M} \end{bmatrix}. \tag{2.23}$$

Matrix $\boldsymbol{A}^{-1}$ can be rewritten as a composition of matrices b and c

$$\boldsymbol{A}^{-1} = \begin{bmatrix} \boldsymbol{b}_1^T & \boldsymbol{c}_1^T \\ \vdots & \vdots \\ \boldsymbol{b}_6^T & \boldsymbol{c}_6^T \end{bmatrix} \qquad \text{with } \boldsymbol{b_i}, \boldsymbol{c_i} \in \mathbb{R}^3, \tag{2.24}$$

being possible to decouple torque and force and write $\boldsymbol{q}$ as

$$\boldsymbol{q} = \boldsymbol{b^T F} + \boldsymbol{c^T M}. \tag{2.25}$$

2.4 Multi-Objective Optimization

It is common to find problems which require the simultaneous optimization of multiple objective functions. This type of problem is called Multi-Objective Optimization (MOO) and can be generically formulated as

$$\begin{aligned} &\underset{\boldsymbol{x}}{\text{minimize}} \quad \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_n(\boldsymbol{x})) \\ &\text{subject to} \\ &\boldsymbol{x} \in \mathcal{X} \end{aligned} \tag{2.26}$$

where n is the number of objective functions f_i, $\boldsymbol{x}$ the vector of design parameters and $\mathcal{X}$ the feasible design set. Generally, it is impossible to find a global solution which optimizes all objective functions. A solution $\boldsymbol{x}^* \in \mathcal{X}$ is said Pareto Optimal if and only if there does not exist another point $\boldsymbol{x} \in \mathcal{X}$, such that $\boldsymbol{f}(\boldsymbol{x}) \leq \boldsymbol{f}(\boldsymbol{x}^*)$, and $f_i(\boldsymbol{x}) < f_i(\boldsymbol{x}^*)$ for at least one function [15]. The set of Pareto Optimal points form the Pareto frontier. A solution is chosen from the Pareto frontier doing trade-off analysis between the objective functions.

2.5 Related Work

2.5.1 Additive Manufacturing by Mobile Robots

The idea of mobile robots capable of AM is not new. In Building and Construction, a large-scale system capable of AM by a team of mobile robots has been demonstrated in [16]. In aerial repair, an Unmanned Aerial Vehicles (UAV) equipped with a Delta manipulator [9] for aerial repair has been demonstrated in [17]. In orbit, the Archinaut [18, 19] is a proposed robotic system, funded by NASA, capable of autonomously manufacturing and assembling structures in space. The Archinaut is proposed to be equipped with a 3D printer for part manufacturing and a manipulator for part assembly.

(a) Building and Construction [16].

(b) Aerial Repair [17].

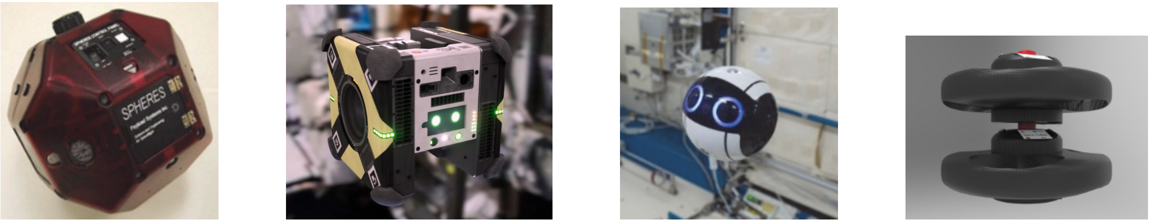
(c) Orbital [18, 19].

Figure 2.3: Examples of mobile robots capable of additive manufacturing, in multiple applications and environments.

2.5.2 Indoor Space Free-Flying Robots

Multiple Free-Flying Robots have been tested in micro-gravity aboard the ISS. SPHERES [20] is a free-flying robot operated by NASA, designed as a testbed for formation flight. For propulsion, SPHERES uses twelve carbon-dioxide cold gas thrusters. Astrobee [21] is in some ways the successor of SPHERES, building on SHEPERE's legacy and lessons learned. For propulsion, the Astrobee uses a relatively more simple system, composed of two large fans inside with twelve duct valves for control. Int-Ball [22] is a free-flying robot operated by Japan Aerospace Exploration Agency (JAXA), with the objective of being an autonomous mobile camera. For propulsion, the Int-Ball uses twelve small encapsulated fans.

Space CoBot [13] is a holonomic aerial robot designed for indoor micro-gravity environments. The propulsion system of this robot is composed of six propellers, similar to the ones employed in traditional Earth multi-rotors. Unlike regular multi-rotors, the propellers are placed so that the kinematics are holonomic.



(a) SPHERES [20]. **(b)** Astrobee [21]. **(c)** Int-Ball [22]. **(d)** Two Space CoBot [13] stacked.

Figure 2.4: Example of free-flying robots designed for micro-gravity habitats.

2.5.3 Manipulator Dimensioning and Optimization

The most used strategy for manipulator optimization consists in defining performance metrics, relevant to the application, and trying to find the design parameters which trade-off these metrics. It is important, in most applications, to consider multiple metrics given that, for example, an increase in work-space volume can lead to a decrease in dexterity [23, 24].

In [25], a 6 DoF 6-RSS parallel manipulator, similar to the manipulator proposed here, is optimized. The researchers define a metric of dexterity, called the GCI, and maximize both the GCI and the dexterous work-space volume $V_{\mathcal{W}}$. To combine both metrics, they use a weighted sum of $V_{\mathcal{W}}$ and GCI to formulate the objective function. Given that evaluating the whole work-space, for a given manipulator configuration, is a computationally demanding task, surrogate model techniques were employed for optimization to minimize the number of objective function evaluations. The approach of using a weighted sum to build a objective function of multiple performance metrics is considered problematic [23] because it only provides one optimal solution, not fully encapsulating the trade-off between the performance metrics. The choice of weights is also not an easy task, given that the objective function would be a composite of functions with different units and orders of magnitude.

Gough-Stewart type mechanisms [8] were successfully dimensioned, using a Multi-Objective Evolutionary Algorithm [24, 26] to build the Pareto frontier. Considering the difficulty in finding close-form solutions to the metrics used in [24–26], discretization based techniques were employed. This technique computes an approximate value of a metric by discretizing the work-space into cells and evaluating each cell. Alternatively, Monte Carlo methods can be used for metric computation [27].

In [23] the authors use interval analysis to compute all the possible design solution which satisfy a set of compulsory requirements. Another similar strategy is to build the objective function with only one performance metric and using the others as optimization constraints. This strategy is adopted in [28, 29] and Controlled Random Search used to solve the optimization problem.

3

Manipulator

Contents

3.1 Design Parameters 18
3.2 Inverse Kinematics 19
3.3 Direct Kinematics 21
3.4 Workspace 22
3.5 Velocity Analysis 24
3.6 Evaluation 24
3.7 Design Space 28
3.8 Optimization 30
3.9 Results 31

In this chapter we will address the problem of the manipulator design. First we will define the design parameters of the parallel manipulator. Secondly, we will address the inverse and direct kinematic problem and deduce the inverse kinematic Jacobian. Then, we will define the manipulator's workspace and it's constraints. We will discuss and define relevant performance metrics. Lastly, we will define manipulator design as a optimization problem and use a numeric solver to approximate the pareto set.

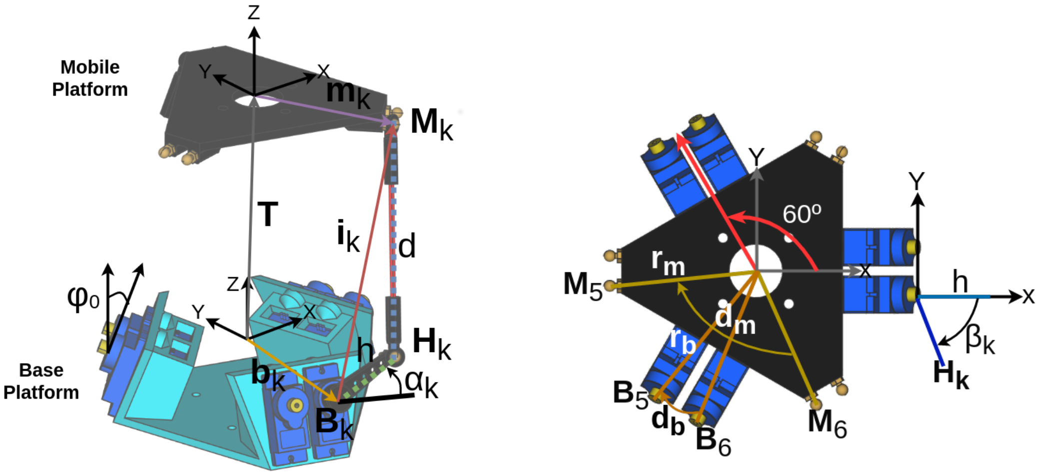
3.1 Design Parameters

The manipulator is composed of a mobile platform, with origin ${}^{p}\mathcal{O}$, where the end-effector is located, and a fixed base, with origin ${}^{b}\mathcal{O}$. The k-th Revolute-Spherical-Spherical arm, with $k \in \{1, \dots, 6\}$, closes the loop between the platforms, connecting base anchor $\boldsymbol{B_k}$ to the platform's anchor $\boldsymbol{M_k}$. As depicted in fig. 3.1, each anchor $\boldsymbol{B_k}$ is connected to each arm by an actuated revolute joint. A rigid link of length h connects $\boldsymbol{B_k}$ to $\boldsymbol{H_k}$, and a rigid link of length d connects $\boldsymbol{H_k}$ to $\boldsymbol{M_k}$. The joints located at $\boldsymbol{H_k}$ and $\boldsymbol{M_k}$ are passive spherical joints. From fig. 3.1(a), we can obtain the following loop closure equation

$$\boldsymbol{i_k} = \boldsymbol{h_k} + \boldsymbol{d_k} = \boldsymbol{T} + \boldsymbol{R}\boldsymbol{m_k} - \boldsymbol{b_k}, \tag{3.1}$$

where $\boldsymbol{T} = \begin{bmatrix} x & y & z \end{bmatrix}^T$ is the translation and $\boldsymbol{R} \in \mathbb{R}^{3\times3}$ is the rotation matrix which define the mobile platform's pose $\boldsymbol{p} = \begin{bmatrix} x & y & z & \gamma & \theta & \psi \end{bmatrix}^T$. $\boldsymbol{R}$ can be described by three Euler angles (γ, θ, ψ), following the ZYZ convention

$$\boldsymbol{R}(\gamma, \theta, \psi) = \boldsymbol{R}(Z, \theta)\boldsymbol{R}(Y, \gamma)\boldsymbol{R}(Z, \psi). \tag{3.2}$$



(a) Platform's dimensions.

(b) Manipulator diagram.

Figure 3.1: Parallel manipulator schematic.

According to fig. 3.1(b), the base anchor arrangement is defined by vector $\boldsymbol{b_k}$

$$\boldsymbol{b_k} = \begin{bmatrix} r_b \cos(\theta_b) & r_b \sin(\theta_b) & 0 \end{bmatrix}^T \qquad \theta_b = \frac{2\pi \lfloor \frac{k+1}{2} \rfloor}{3} + (-1)^k d_b \tag{3.3}$$

where d_b is the angle between the anchor pairing and r_b the base platform radius.

The mobile platform anchor arrangement is defined by vector $\boldsymbol{m_k}$

$$\boldsymbol{m_k} = \begin{bmatrix} r_m \cos(\theta_m) & r_m \sin(\theta_m) & 0 \end{bmatrix}^T \qquad \theta_m = \frac{2\pi \lfloor \frac{k+1}{2} \rfloor}{3} + (-1)^k d_m \tag{3.4}$$

where d_m is the angle between the anchor pairing and r_m the mobile platform radius.

The k-th revolute joint arm orientation is defined by angles ϕ_k and β_k

$$\phi_k = (-1)^{k+1} \phi_0 \qquad \beta_k = \frac{2\pi \lfloor \frac{k+1}{2} \rfloor}{3} + (-1)^k \beta_0 \tag{3.5}$$

where ϕ_0 and β_0 describe the revolute joint arm orientation in space.

In conclusion, to dimension the described manipulator, design parameters r_b, r_m, d_b, d_m, ϕ_0, β_0, d and h must be specified. The resulting vector of design parameters $\boldsymbol{s}$ is given by $\boldsymbol{s} = (r_b, r_m, d_b, d_m, \phi_0, \beta_0, d, h)$.

If we apply the aforementioned parameterization to the manipulator design presented in section 1.1, following the CAD physical dimensions, we arrive at the geometric parameters presented in table 3.1.

Table 3.1: Design parameters ACROBAT's first iteration manipulator.

Parameter	r_b	r_m	d_b	d_m	d	h	ϕ_0	β_0
Value	52.57 mm	48.14 mm	0.148 rad	0.964 rad	117.50 mm	27.00 mm	0.349 rad	1.571 rad

3.2 Inverse Kinematics

As depicted in fig. 3.1, the k-th revolute joint arm $\boldsymbol{H_k}$ is obtained by first rotating the revolute joint arm by α_k along the Y axis, then rotating by ϕ_k along the X axis and finally rotating along the Z axis by β_k, resulting in

$$\begin{aligned} \boldsymbol{H_k} &= \boldsymbol{B_k} + \boldsymbol{R}(Z, \beta_k)\boldsymbol{R}(X, \phi_k)\boldsymbol{R}(Y, -\alpha_k) \begin{bmatrix} h & 0 & 0 \end{bmatrix}^T & (3.6) \\ &= \boldsymbol{B_k} + h \begin{bmatrix} \sin(\beta_k)\sin(\phi_k)\sin(\alpha_k) + \cos(\beta_k)\cos(\alpha_k) \\ -\cos(\beta_k)\sin(\phi_k)\sin(\alpha_K) + \sin(\beta_k)\cos(\alpha_k) \\ \cos(\phi_k)\sin(\alpha_k) \end{bmatrix}. & (3.7) \end{aligned}$$

Using an alternative formulation, the k-th revolute joint arm $\boldsymbol{H_k}$ can be obtained by rotating $\boldsymbol{h_k}$ by $-\alpha_k$ around u_k, so that

$$\boldsymbol{H_k} = \boldsymbol{B_k} + \boldsymbol{R}(\boldsymbol{u_k}, -\alpha_0) \begin{bmatrix} h & 0 & 0 \end{bmatrix}^T \tag{3.8}$$

where u_k represents the revolute's joint axis of rotation given by

$$\boldsymbol{u_k} = \boldsymbol{R}(Z, \beta_k)\boldsymbol{R}(X, \phi_k) \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T. \tag{3.9}$$

Considering the rigid links which make the platform's arms, the manipulator must obey the following constraints

$$d^2 = (\boldsymbol{M_k} - \boldsymbol{H_k})^T (\boldsymbol{M_k} - \boldsymbol{H_k}) \tag{3.10}$$

$$h^2 = (\boldsymbol{H_k} - \boldsymbol{B_k})^T (\boldsymbol{H_k} - \boldsymbol{B_k}) \tag{3.11}$$

Additionally, $\|\boldsymbol{i_k}\|^2$ can be calculated by

$$\|\boldsymbol{i_k}\|^2 = (\boldsymbol{M_k} - \boldsymbol{B_k})^T (\boldsymbol{M_k} - \boldsymbol{B_k}). \tag{3.12}$$

Combining and factorizing eq. (3.10) and eq. (3.12) [30]:

$$\|\boldsymbol{i_k}\|^2 - (d^2 - h^2) = 2\boldsymbol{B_k}^T\boldsymbol{B_k} - 2\boldsymbol{B_k}^T\boldsymbol{H_k} - 2\boldsymbol{B_k}^T\boldsymbol{M_k} + 2\boldsymbol{H_k}^T\boldsymbol{M_k} = 2(\boldsymbol{H_k} - \boldsymbol{B_k})^T(\boldsymbol{M_k} - \boldsymbol{B_k}). \tag{3.13}$$

Substituting $\boldsymbol{i_k}$ and $\boldsymbol{H_k}$ in eq. (3.13) with eq. (3.1) and eq. (3.6), respectively, results in

$$\|\boldsymbol{i_k}\|^2 - (d^2 - h^2) = 2h \begin{bmatrix} \sin(\beta_k)\sin(\phi_k)\sin(\alpha_k) + \cos(\beta_k)\cos(\alpha_k) \\ -\cos(\beta_k)\sin(\phi_k)\sin(\alpha_K) + \sin(\beta_k)\cos(\alpha_k) \\ \cos(\phi_k)\sin(\alpha_k) \end{bmatrix}^T \boldsymbol{i_k}. \tag{3.14}$$

Rearranging eq. (3.15)

$$\|\boldsymbol{i_k}\|^2 - (d^2 - h^2) = 2h(a_k \sin(\alpha_k) + b_k \cos(\alpha_k)) \tag{3.15}$$

$$a_k = \sin(\beta_k)\sin(\phi_k)\boldsymbol{i_k}^{(x)} - \cos(\beta_k)\sin(\phi_k)\boldsymbol{i_k}^{(y)} + \cos(\phi_k)\boldsymbol{i_k}^{(z)} \tag{3.16}$$

$$b_k = \cos(\beta_k)\boldsymbol{i_k}^{(x)} + \sin(\beta_k)\boldsymbol{i_k}^{(y)}. \tag{3.17}$$

Applying the trigonometric identity

$$a\sin(\alpha) + b\cos(\alpha) = \sqrt{a^2 + b^2}\sin(\alpha + \arctan 2(b, a)) \tag{3.18}$$

where

$$\arctan 2(y,x) = \begin{cases} \arctan(\frac{y}{x}) & \text{if } x > 0, \\ \arctan(\frac{y}{x}) + \pi & \text{if } x < 0 \text{ and } y \geq 0, \\ \arctan(\frac{y}{x}) - \pi & \text{if } x < 0 \text{ and } y < 0, \\ +\frac{\pi}{2} & \text{if } x = 0 \text{ and } y > 0, \\ -\frac{\pi}{2} & \text{if } x = 0 \text{ and } y < 0, \\ \text{undefined} & \text{if } x = 0 \text{ and } y = 0 \end{cases} \tag{3.19}$$

to eq. (3.15) results in the following relation

$$\|\boldsymbol{i_k}\|^2 - (d^2 - h^2) = \sqrt{a_k^2 + b_k^2}\sin(\alpha + \arctan 2(b_k, a_k)) \tag{3.20}$$

$$\tag{3.21}$$

therefore,

$$\alpha_k = \arcsin\left(\frac{\|\boldsymbol{i_k}\|^2 - (d^2 - h^2)}{\sqrt{a_k^2 + b_k^2}}\right) - \arctan 2(b_k, a_k). \tag{3.22}$$

The angle α_k of each revolute joint can be calculated using eq. (3.22), solving the inverse kinematics problem. Given that there is a closed-form solution to the inverse kinematics, the manipulator can be fully described with only $\boldsymbol{p}$. In other words, the pose of all the joints can be derived with only knowing the end-effector pose $\boldsymbol{p}$.

3.3 Direct Kinematics

The direct kinematics problem consists in finding the vector of end effector pose $\boldsymbol{p}$ for a given vector of actuated joints $\boldsymbol{\alpha}$: $\boldsymbol{p} = g(\boldsymbol{\alpha})$. As the best of our knowledge, no general closed form solution for the direct kinematics of this class of manipulators has been established.

However, a special case of the direct kinematics where $\alpha_k = 0$ for $k \in \{1, \ldots, 6\}$ exists. Given the platform geometry and symmetry, we can assume that in this case the end-effector has no rotation ($\boldsymbol{R} = \boldsymbol{I}$) and the translation is purely on the Z axis $\boldsymbol{T} = \begin{bmatrix} 0 & 0 & z \end{bmatrix}^T$. From fig. 3.1(a) we get

$$d^2 = (\boldsymbol{T} + \boldsymbol{C_k})^T(\boldsymbol{T} + \boldsymbol{C_k}) = \boldsymbol{T}^T\boldsymbol{T} + \boldsymbol{C_k}^T\boldsymbol{T} + \boldsymbol{T}^T\boldsymbol{C_k} + \boldsymbol{C_k}^T\boldsymbol{C_k} \tag{3.23}$$

where $\boldsymbol{C_k}$ is given by

$$\boldsymbol{C_k} = \boldsymbol{R}\boldsymbol{m_k} - \boldsymbol{H_k} = \boldsymbol{I}\boldsymbol{m_K} - \boldsymbol{B_k} + h\begin{bmatrix} \cos(\beta_k) \\ \sin(\beta_k) \\ 0 \end{bmatrix} = \begin{bmatrix} r_m\cos(\theta_m) - r_b\cos(\theta_b) - h\cos(\beta_k) \\ r_m\sin(\theta_m) - r_b\sin(\theta_b) - h\sin(\beta_k) \\ 0 \end{bmatrix} \tag{3.24}$$

substituting eq. (3.24) and eq. (3.23), and knowing that $\boldsymbol{C_k}^T\boldsymbol{T} = \boldsymbol{T}^T\boldsymbol{C_k} = 0$, we get

$$z = \pm\sqrt{d^2 - [r_m\cos(\theta_m) - r_b\cos(\theta_b) - h\cos(\beta_k)]^2 - [r_m\sin(\theta_m) - r_b\sin(\theta_b) - h\sin(\beta_k)]^2}. \tag{3.25}$$

Equation (3.25) holds two solutions, one for each working node of the manipulator. However, considering the defined working node, we will only consider $z \geq 0$.

3.4 Workspace

We will only consider constraints of type a) presented in section 2.1.4, but the methodology can be easily expanded to other types of constraints. In the case of the manipulator presented here, for a $\boldsymbol{p}$ to belong to $\mathcal{W}$, it must hold a real solution to eq. (3.22). Meaning that

$$-1 \leq \frac{\|\boldsymbol{i_k}\|^2 - (d^2 - h^2)}{\sqrt{a_k^2 + b_k^2}} \leq 1 \qquad\qquad k \in \{1, \dots, 6\} \tag{3.26}$$

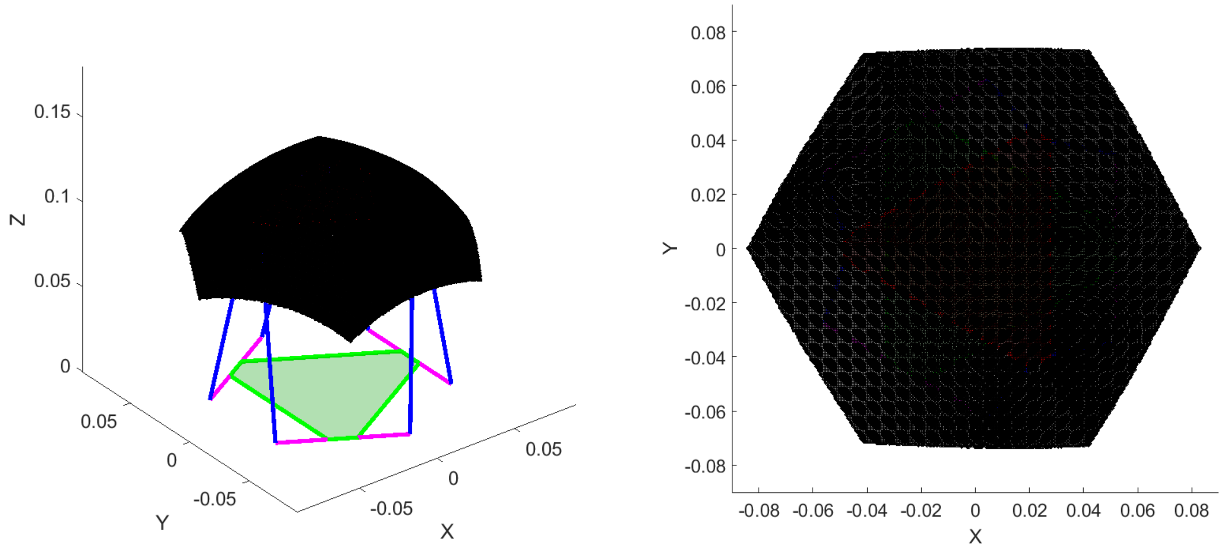
must hold, where each kinematic chain k imposes a constraint expressed in eq. (3.26). Equation (3.26) can be rewritten as

$$h_k(\boldsymbol{p}) = [\|\boldsymbol{i_k}\|^2 - (d^2 - h^2)]^2 - (a_k^2 + b_k^2) \leq 0 \qquad\qquad k \in \{1, \dots, 6\} \tag{3.27}$$

where i_k, a_k and b_k depend on $\boldsymbol{p}$ as shown in eq. (3.1), eq. (3.16) and eq. (3.17) respectively.

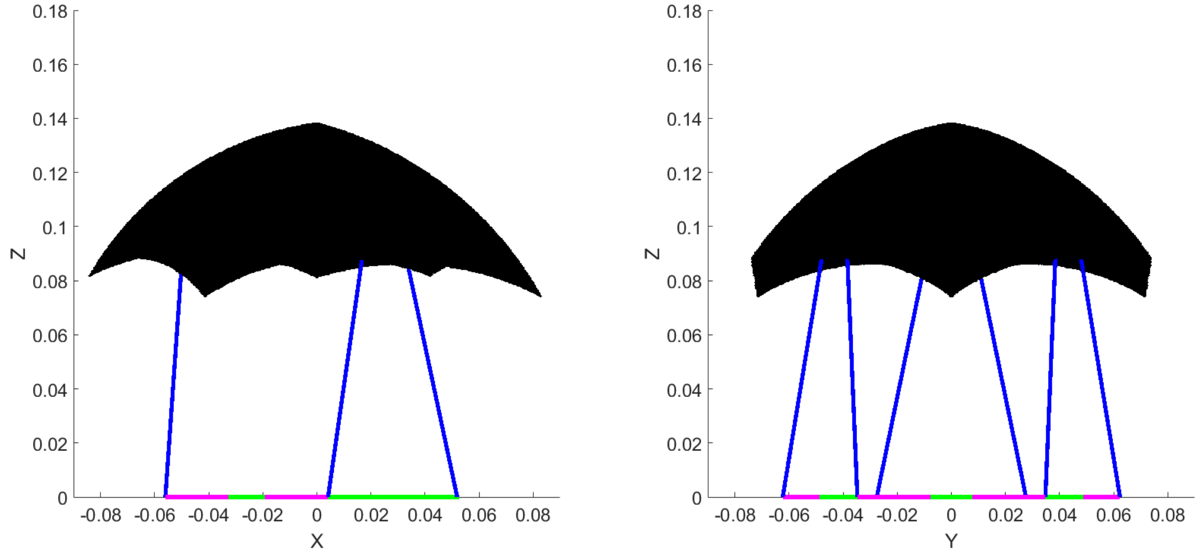
3.4.1 Visualization

To help visualize the workspace geometry, we plotted the manipulator's workspace using cell discretization methods. Given that $\boldsymbol{p} \in \mathbb{R}^6$, the reachable workspace is impossible to intuitively illustrate. So, for representation purposes, we will fix the manipulator's orientation so that the mobile platform's representative rotation matrix, recall eq. (3.1), equals the unity matrix $\boldsymbol{R} = \boldsymbol{I}^{3\times3}$. In doing so, we can represent the manipulator's workspace in a 3D space, often called the constant orientation workspace, with results shown in fig. 3.2.



(a) Perspective.

(b) XY.



(c) XZ.

(d) YZ.

Figure 3.2: ACROBAT's manipulator constant orientation workspace. The color scheme goes as follows: The red patch is the mobile platform, the green patch the fixed platform, the magenta and blue line-segment are rigid links of size h and d, respectively. The black solid represents the manipulator's workspace.

Observing fig. 3.2, it is clear that the manipulator's workspace is not similar to any familiar 3D shape.

This irregular shape is expected, given that the workspace is the interception of six kinematic constraints, expressed in eq. (3.27). Moreover, with some algebraic manipulation, we can observe that each constraint in eq. (3.27) is a torus, meaning that the constant orientation workspace is the interception of six torus, each imposed by one of the six kinematic chains. Another interesting but expected feature of the workspace is that it is symmetric along the Z axis.

3.5 Velocity Analysis

Consider an end-effector with twist $\boldsymbol{w} \in \mathbb{R}^6$. As stated in section 2.1.3, the inverse kinematic Jacobian is given by

$$\dot{\boldsymbol{\alpha}} = \boldsymbol{J^{-1}}(\boldsymbol{p})\boldsymbol{w}. \tag{3.28}$$

where $\dot{\boldsymbol{\alpha}}$ is the velocity vector of the actuated joints and $\boldsymbol{w}$ has a linear velocity $\boldsymbol{v} \in \mathbb{R}^3$ and an angular velocity $\boldsymbol{\omega} \in \mathbb{R}^3$ so that $\boldsymbol{w} = \begin{bmatrix}\boldsymbol{v}^T & \boldsymbol{\omega}^T\end{bmatrix}^T$. Consider the loop-closure equation in eq. (3.1). Differentiating relative to time, we get for each k that

$$\boldsymbol{w}_k^{R-S} \times \boldsymbol{h_k} + \boldsymbol{w}_k^{S-S} \times \boldsymbol{d_k} = \boldsymbol{v} + \boldsymbol{\omega} \times \boldsymbol{Rm_k} \tag{3.29}$$

where $\boldsymbol{w}_k^{R-S}$ is the twist of the R-S link and $\boldsymbol{w}_k^{S-S}$ the twist of the S-S link. To remove the $\boldsymbol{w}_k^{S-S}$ from the equation, we find the inner product of $\boldsymbol{d_k}$ on both sides resulting in

$$(\boldsymbol{w}_k^{R-S} \times \boldsymbol{h_k}) \cdot \boldsymbol{d_k} + (\boldsymbol{w}_k^{S-S} \times \boldsymbol{d_k}) \cdot \boldsymbol{d_k} = \boldsymbol{v} \cdot \boldsymbol{d_k} + (\boldsymbol{\omega} \times \boldsymbol{Rm_k}) \cdot \boldsymbol{d_k} \tag{3.30}$$

rearranging the terms and simplifying, we get

$$(\boldsymbol{h_k} \times \boldsymbol{d_k}) \cdot \boldsymbol{w}_k^{R-S} = \boldsymbol{d_k} \cdot \boldsymbol{v} + (\boldsymbol{Rm_k} \times \boldsymbol{d_k}) \cdot \boldsymbol{w} \tag{3.31}$$

putting in matrix form, and knowing that $\boldsymbol{w}_k^{R-S} = \boldsymbol{u_k}\dot{\alpha}_k$ we get

$$\dot{\alpha}_k = \begin{bmatrix} \frac{\boldsymbol{d_k}}{(\boldsymbol{h_k} \times \boldsymbol{d_k}) \cdot \boldsymbol{u_k}} & \frac{\boldsymbol{Rm_k} \times \boldsymbol{d_k}}{(\boldsymbol{h_k} \times \boldsymbol{d_k}) \cdot \boldsymbol{u_k}} \end{bmatrix} \boldsymbol{w} \tag{3.32}$$

which corresponds to a line of $\boldsymbol{J^{-1}}$.

3.6 Evaluation

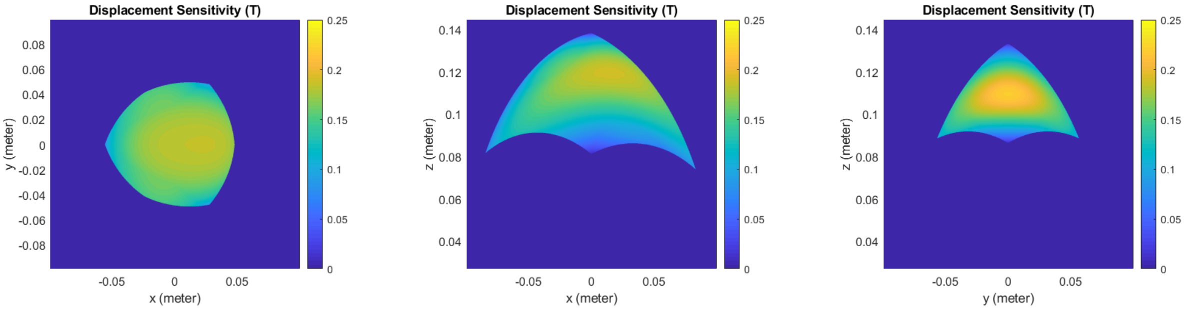
An important step for manipulator design is the choice of performance metrics to evaluate and compare different designs. This choice is heavily dependent on the desired tasks. Given the tasks at hand of assembly and AM, we consider that both workspace volume and accuracy are desirable. The workspace volume bust be high enough to compensate the ACROBAT's main body low actuation bandwidth and have enough accuracy to successfully complete the designated tasks. The use of metrics of accuracy was considered, but given the desired tasks, maximizing accuracy was preferable.

The workspace volume $V_{\mathcal{W}}$ is given by the expression in eq. (2.8).

The accuracy is evaluated using the expressions found in eq. (2.16). We note that this metric can only be used if all actuated joints share a type, which is the case here, given that ACROBAT's manipulator actuated joints are all revolute. As stated in section 2.2.3, the accuracy metric is separated in global translation sensitivity, GTSI, and global rotational sensitivity, GRSI.

3.6.1 Visualization

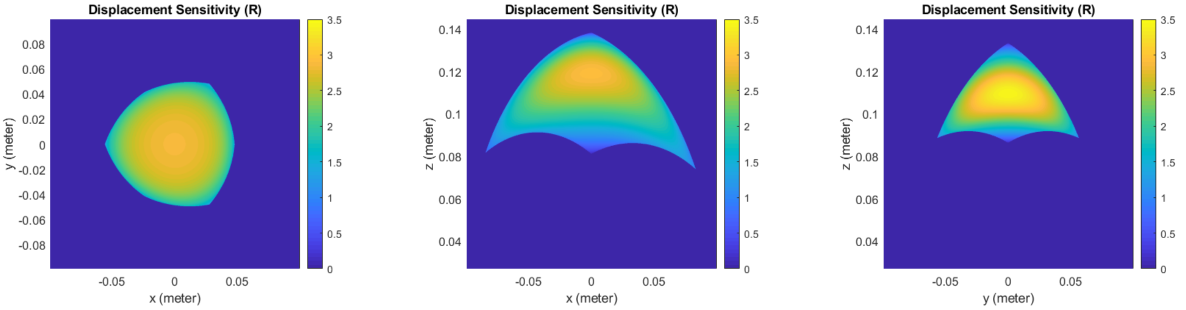
The metrics proposed in the aforementioned section, to evaluate the manipulator's accuracy, evaluate his performance over the whole workspace. To illustrate how the manipulator's accuracy spreads over the workspace, fig. 3.2 presents heatmaps of slices of the ACROBAT's first iteration manipulator translational and rotational displacement sensitivity, presented in eq. (2.14). Given the difficulties in presenting the total workspace, we will consider a constant-orientation workspace (with $\boldsymbol{R} = \boldsymbol{I}$).



(a) XY plane with $z = z_{home}$.

(b) XZ plane with $y = 0$.

(c) YZ plane with $x = 0$.



(d) XY plane with $z = z_{home}$

(e) XZ plane with $y = 0$.

(f) YZ plane with $x = 0$.

Figure 3.3: ACROBAT's manipulator workspace slices with the respective displacement sensitivity, with $\boldsymbol{R} = \boldsymbol{I}$. The first row plot the translational displacement sensitivity and the bottom one the rotational displacement sensitivity.

Looking at fig. 3.3 it is clear that the accuracy is not uniform over the manipulator's workspace. Given that a higher displacement sensitivity is synonym with a lower accuracy, we note that the manipulator's is less accurate in the center of his workspace. This is no surprise, given that the manipulator, for a same actuator speed, travels faster in the workspace's center. The lack of metric uniformity over the workspace validates the necessity to evaluate the performance metrics over the whole workspace, instead of, for instance, only computing their value in a set of pre-determined points.

3.6.2 Computation

As far as we know, the performance metric functions presented in eq. (2.8) and eq. (2.16) lack a closed-form solution, so we will approximate their value by discretizing the workspace. In our implementation, we will use a constant orientation workspace (meaning that $\boldsymbol{R} = \boldsymbol{I}$), but the methodology can be easily expanded to consider the reachable workspace, at the cost of a heavier computational burden. With this in mind, from now on we will consider the manipulator's workspace $\mathcal{W}$ to be 3-dimensional ($\mathcal{W} \in \mathbb{R}^3$).

Even without explicitly computing $\mathcal{W}$, we know that it must be bounded by the box $\mathcal{W}_{max} = \{-d-h \leq x \leq d+h, -d-h \leq y \leq d+h, 0 \leq z \leq d+h | x, y, z \in \mathbb{R}\}$. The discretization approach is divided into three steps:

(1) Dividing $\mathcal{W}_{max}$ into cells of size $\Delta x \Delta y \Delta z$, resulting in the grid $\mathcal{G}_{max}$, with

$$\Delta x = 2\frac{d+h}{N_x - 1} \qquad \Delta y = 2\frac{d+h}{N_y - 1} \qquad \Delta z = \frac{d+h}{N_z - 1} \tag{3.33}$$

where N_x, N_y and N_z is the number cells in each axis;

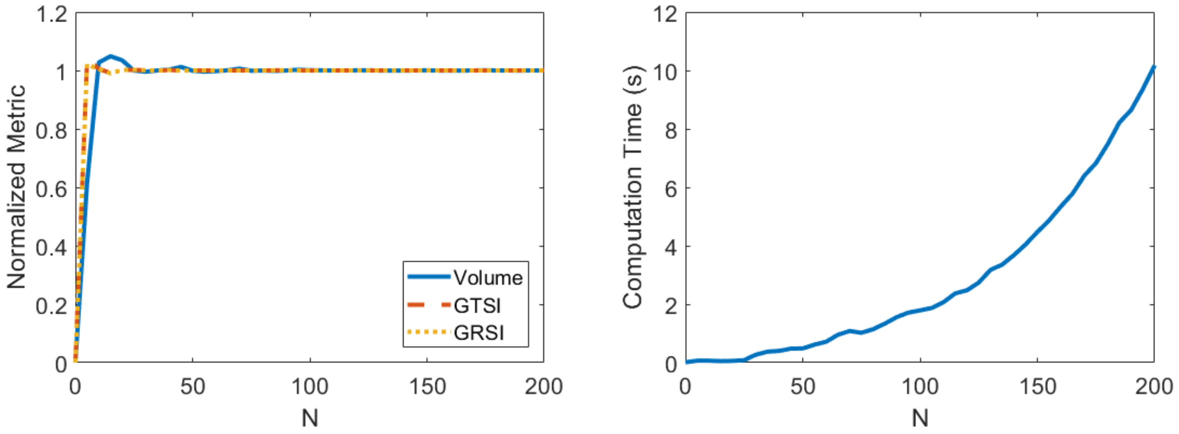
(2) For each cell i in $\mathcal{G}_{max}$ evaluate if i belongs to $\mathcal{W}$: If i obeys eq. (3.27), add i to $\mathcal{G}_{\mathcal{W}}$;

(3) Compute the metrics in $\mathcal{G}_D$:

$$V_{\mathcal{W}} \approx \#\mathcal{G}_{\mathcal{W}} \Delta x \Delta y \Delta z \qquad \text{GTSI} \approx \frac{\sum_{i \in \mathcal{G}_{\mathcal{W}}} \sigma_{t,\infty}}{\#\mathcal{G}_{\mathcal{W}}} \qquad \text{GRSI} \approx \frac{\sum_{i \in \mathcal{G}_{\mathcal{W}}} \sigma_{r,\infty}}{\#\mathcal{G}_{\mathcal{W}}} \tag{3.34}$$

where $\#\mathcal{G}_{\mathcal{W}}$ is the number of cells in $\mathcal{G}_{\mathcal{W}}$. Local indexes $\sigma_{t,\infty}$ and $\sigma_{r,\infty}$ are calculated according to eq. (2.14), with $\boldsymbol{J}$ computed at i.

One question that arises from the aforementioned algorithm is how high must be N_x, N_y and N_z for it to return accurate approximations of the performance metrics. For the sake of simplicity lets consider that $N = N_x = N_y = N_z$: fig. 3.4 presents the performance value and the computation time as a function of N. Given that the absolute value of the performance metrics poses no interest, only how it changes with a increasing N, we normalized their value by dividing $V_{\mathcal{W}}$, GTSI and GRSI by their respective values when $N = 200$. The algorithm was implemented on MATLAB 2019a and ran on an AMD Ryzen 7 3800X 8-Core Processor at 3.90 GHz. The geometric parameters of ACROBAT's first iteration manipulator, consultable in table 3.1, were used in this experiment



(a) Normalized performance metric as a function of N.

(b) Performance metric computation time as a function of N.

Figure 3.4: Plots showing how the algorithm performs with a increasing value of N.

The main takeway from fig. 3.4 is that the discretization algorithm approaches the true value of the performance metrics for a increasing value of N. With this in mind, and taking into account the algorithm's computational overhead, we consider that $N = 50$ produces a good enough approximation.

3.7 Design Space

This section will address the bounds and constraints of the manipulator's design space.

3.7.1 Bounds

Let $\mathcal{S}$ be a bounded box, representing the feasible design parameter space. Given the bounds imposed by the parameterization context and application, we assume that $\mathcal{S}$ is bounded. For instance, consider the base radius r_b. Given that it must be large enough for the actuators to fit and small enough to fit the robot body, we consider that r_b must be larger than 0.01 meters and smaller than 0.06 meters. The same logic applies to the mobile platform radius r_m, which must be large enough for a tool to fit, so we assume that r_m must be larger than 0.01 meters and smaller than 0.06 meters. Considering manipulator geometry, the angle between mobile platform anchor pairings d_m and base platform pairings d_b is defined to be between 0 and $\frac{\pi}{3}$. However, given the platform's symmetry, these bounds would produce redundant designs (The manipulator with $d_m = d_b = 0$ is the same as $d_m = d_b = \frac{\pi}{3}$), so we set the maximum d_b at $\frac{\pi}{6}$. Likewise, we define define ϕ_0 to be between $-\frac{\pi}{3}$ and $\frac{\pi}{3}$ and β_0 to be between 0 and π. A low value of d_m or d_b would lead physical interference between the joints, as they can not be in the same physical space, so we assume that d_b and d_m must at least 0.1 radians.

In conclusion, $\mathcal{S}$ is a bounded box so that $\mathcal{S} = \{0.01 \leq r_b \leq 0.06, 0.01 \leq r_m \leq 0.06, 0.1 \leq d_b \leq \frac{\pi}{3}, 0.1 \leq d_m \leq \frac{\pi}{3}, -\frac{\pi}{3} \leq \phi_0 \leq \frac{\pi}{3}, 0 \leq \beta_0 \leq \pi, 0.01 \leq d \leq 0.2, 0.01 \leq h \leq 0.2\}$.

3.7.2 Constraints

However, not all s inside the bounded box $\mathcal{S}$ produce feasible manipulators. Two types of constraints still apply:

1. Design parameter vector s which lead to kinematically impossible manipulators; and
2. Design parameter vector s which lead to kinematically legal manipulators, but physically impossible given inter-link interference.

Let us start by addressing the first case. A way to verify if s represents a possible manipulator is to ascertain if eq. (3.25) has any real solutions. If it has no real solution, s leads to a manipulator which either has a void workspace or has non-void workspace but there is no pose where $\alpha_k = 0$ for $k \in \{1, \dots, 6\}$. For eq. (3.25) to hold a real solution, the following constraint must hold

$$C_1(\boldsymbol{s}) = [r_p \cos(d_p) - r_b \cos(d_b) - h \cos(\beta_0)]^2 + [r_p \sin(d_p) - r_b \sin(d_b) - h \sin(\beta_k)]^2 - d^2 \leq 0. \tag{3.35}$$

The second case is shown in fig. 3.5. An important class of designs to discard is the s which lead to manipulators with links described by $\boldsymbol{H_k}$ cross paths. Considering the plane in which $\boldsymbol{H_k}$ takes positions, and given that all points $\boldsymbol{B_k}$ share the same plane, we can say that if two links represented by $\boldsymbol{H_j}$ and $\boldsymbol{H_i}$ do not cross paths with $\alpha_k = 0$ for $k \in \{1, \dots, 6\}$, they will not cross for in any other value of α_k, leading us to consider the scenario where $\alpha_k = 0$. Considering points $\boldsymbol{H_1}$ and $\boldsymbol{H_2}$, given that the R joint axis of rotation is located at $\boldsymbol{B_1}$ and $\boldsymbol{B_2}$ respectively, and by construction, we know that $\boldsymbol{B_1^{(y)}} - \boldsymbol{B_2^{(y)}} \leq 0$, we can guarantee that the links represented by points $\boldsymbol{H_1}$ and $\boldsymbol{H_2}$ do not cross paths if

$$C_2(\boldsymbol{s}) = \boldsymbol{H_1^{(y)}} - \boldsymbol{H_2^{(y)}} = [r_b \sin(-d_b) + h \sin(-\beta_0))] - [r_b \sin(d_b) + h \sin(\beta_0))] \leq 0. \tag{3.36}$$

As depicted in fig. 3.5, an analogous relation can be applied to the interlink interference between links represented by points $\boldsymbol{H_1}$ and $\boldsymbol{H_3}$:

$$C_3(\boldsymbol{s}) = \boldsymbol{H_3^{(y)}} - \boldsymbol{H_1^{(y)}} = [r_b \sin(\frac{4\pi}{3} + d_b) + h \sin(\frac{4\pi}{3} + \beta_0)] - [r_b \sin(-d_b) + h \sin(-\beta_0)] \leq 0 \tag{3.37}$$

Given the manipulator symmetry, we do not need to check all $\boldsymbol{H_k}$ pairs for inter-link interference, and the relations presented in eq. (3.36) and eq. (3.37) will suffice.

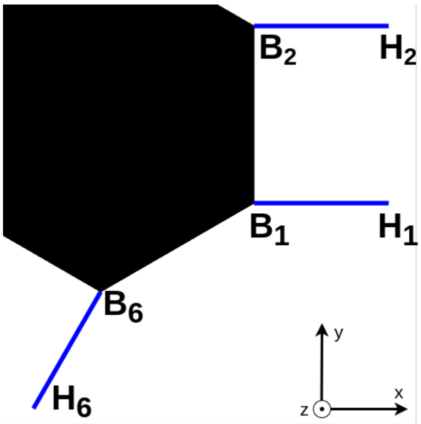


Figure 3.5: Diagram of link pose when $\alpha_k = 0$ for $k \in \{1, 2, 6\}$, based on fig. 3.1. The black patch represents the fixed base, connecting all anchors $\boldsymbol{B_k}$. The blue lines represent the R-S links, starting at $\boldsymbol{B_k}$ and ending at $\boldsymbol{H_k}$.

3.8 Optimization

Given the antagonic nature of workspace volume and accuracy, we wish to find the set of s which best compromise those criteria. Considering this, we consider necessary to employ a multi-objective optimization methodology. Specifically, we wish to maximize the workspace volume ($V_{\mathcal{W}}$) while minimizing the global translational sensitivity GTSI and global rotational sensitivity GRSI. We can easily translate the problem into an all-minimization problem by considering the workspace volume symmetric ($-V_{\mathcal{W}}$).

An approach would be to transform a multi-objective into a single-objective problem, by combining the defined metrics into a single function by means of a weighted sum [15]:

$$f(V_{\mathcal{W}}, \text{GTSI}, \text{GRSI}) = -w_1\, V_{\mathcal{W}} + w_2\, \text{GTSI} + w_3\, GRSI \qquad \text{with } w_1, w_2, w_3 \in \mathbb{R}^+, \tag{3.38}$$

where w_1, w_2 and w_3 are weights associated with each metric. However, this approach has a few problems: First, each metric has a different unit, difficulting the weight choice. Second, to attribute each weight a pondered value, a priori information about the problem is required. Finally, because this approach only produces a single point, it does not allow the trade-off between the metrics to be studied.

With this considerations in mind, we can formulate the following multi-objective optimization problem:

$$
\begin{aligned}
\text{Minimize} \quad & (-V_{\mathcal{W}}, GTSI, GRSI) \\
\text{w.r.t.} \quad & \boldsymbol{s} \in \mathcal{S} \\
\text{subject to} \quad & C_1(\boldsymbol{s}) \leq 0 \\
& C_2(\boldsymbol{s}) \leq 0 \\
& C_3(\boldsymbol{s}) \leq 0\,.
\end{aligned} \tag{3.39}
$$

To compute a point-wise approximation of the Pareto front of the optimization problem formulated in section 3.8, we will use the Direct Multisearch for Multiobjective Optimization [31] algorithm. Given that eq. (2.8) and eq. (2.16) lack a closed-form solution, their derivative also has no close-form solution, so a derivative-free method was chosen. For the sake of reproducibility, tables 3.2 to 3.4 present the used settings.

Table 3.2: Direct Multisearch for Multiobjective stopping criteria settings.

stop_alfa	tol_stop	stop_feval	max_fevals	stop_fparcycles	max_fparCycles
1	10^{-3}	1	200000	1	200000

Table 3.3: Direct Multisearch for Multiobjective initialization, cache and search step settings.

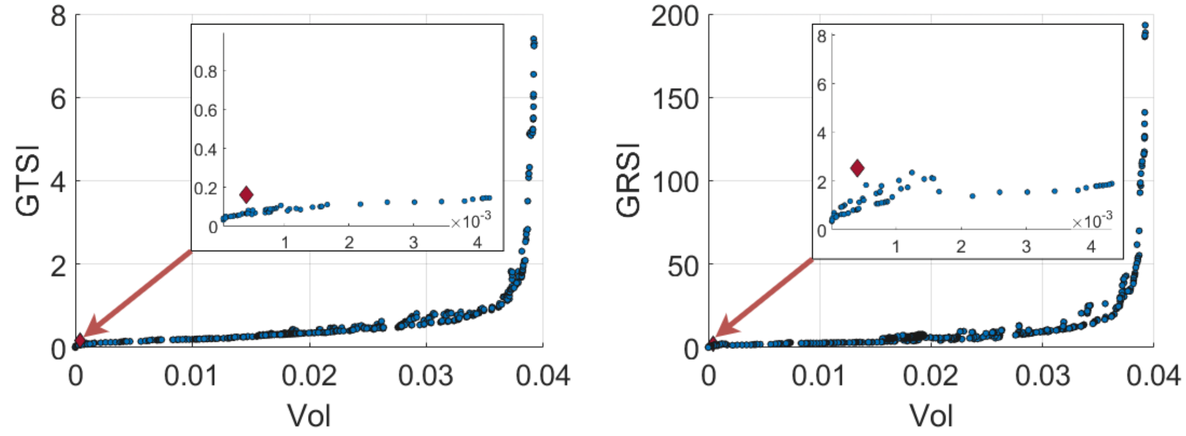
list	user_list_size	nPini	cache	tol_match	search_option	regopt	all_subproblems
3	0	30	1	tol_stop	1	1	0

Table 3.4: Direct Multisearch for Multiobjective centre selection, direction and step size settings

spread_option	selection_strategy	dir_dense	alfa_ini	beta_par	gamma_par
1	1	0	1	0.5	1

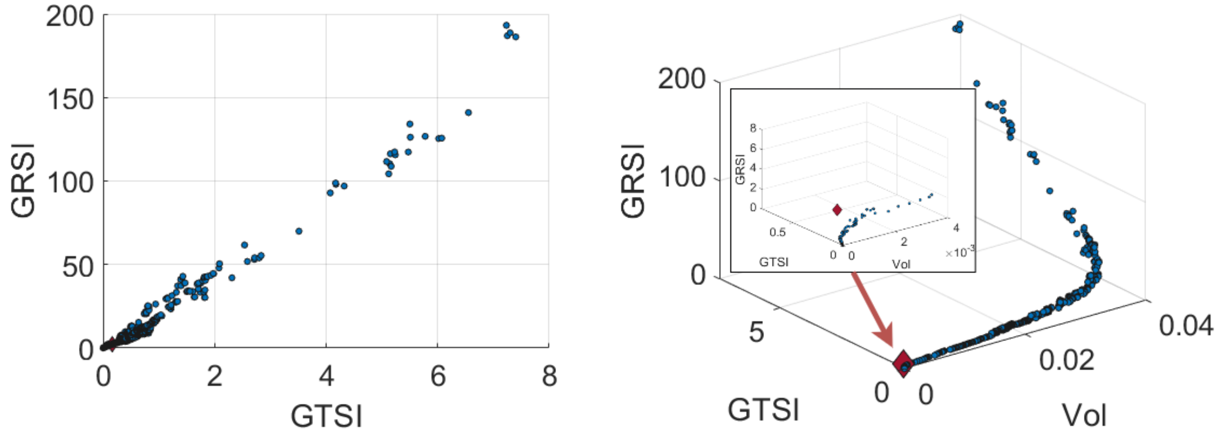
3.9 Results

The point-wise approximation of the Pareto front can be found in fig. 3.6 with a total of 3424 non-dominated points found after the algorithm completed a total of 2×10^5 objective function evaluations.



(a) Volume - GTSI.

(b) Volume - GRSI.



(c) GTSI - GRSI.

(d) Prespective.

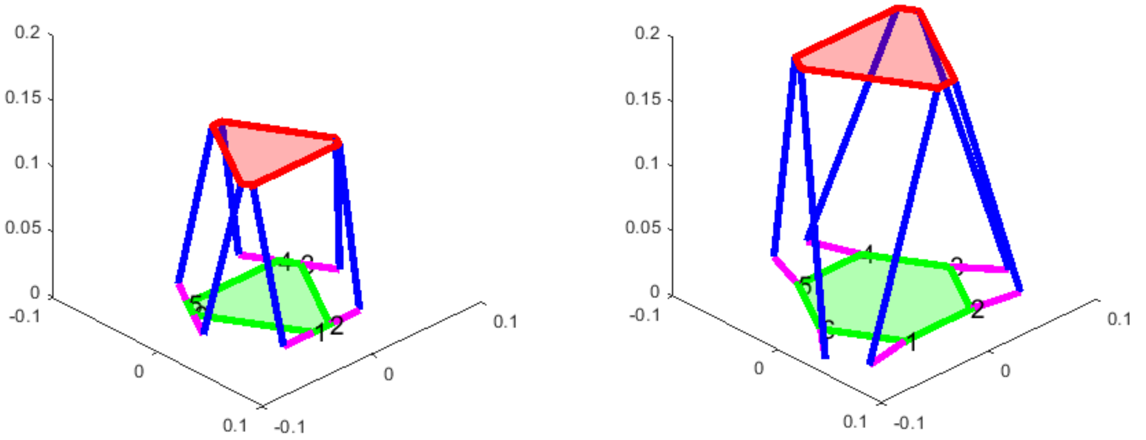
Figure 3.6: Point-wise Pareto Front Approximation. The red diamond represents ACROBAT's first iteration manipulator.

From these results, we found that as the volume increases, both the rotational and translational sensitivity to actuator displacement also increase, resulting in a decrease in accuracy. On the other hand, fig. 3.6(c) suggests that the translational accuracy grows linearly with the rotational accuracy.

But one questions that arises is how should we choose a solution from the Pareto set. One approach to choose a final design is to search the Pareto set to find the solution with GTSI closer to ACROBAT's sub-optimal manipulator. This approach makes sense given ACROBAT's propulsion system lower actuation bandwidth, where the manipulator might need a larger workspace, while maintaining the same tanslational accuracy, to compensate the propulsion's system shortcomings. Table 3.5 compares the design parameters and performance between both designs and fig. 3.7 presents a visual representation of the joint placement. The optimized design has a substantial increase in workspace volume while maintaining the same accuracy.

Table 3.5: Design parameters and performance of ACROBAT's first iteration manipulator and the optimized manipulator taken from fig. 3.6.

Design	r_b	r_m	d_b	d_m	d	h	ϕ_0	β_0	Vol (10^{-3})	GTSI	GRTS
ACROBAT	52.57	48.14	0.148	0.964	117.50	27.00	0.349	1.571	0.408	0.163	2.539
Opt	60.00	60.00	0.5376	0.1302	200.00	40.30	0.000	1.4004	6.110	0.162	2.252



(a) ACROBAT's manipulator.

(b) Optimized design.

Figure 3.7: Joint representation of ACROBAT's first iteration manipulator and the optimized manipulator. The color scheme goes as follows: The red patch is the mobile platform, the green patch the fixed platform, the magenta and blue line-segment are rigid links of size h and d, respectively.

4

Robot Body

Contents

4.1 Parameterization 36

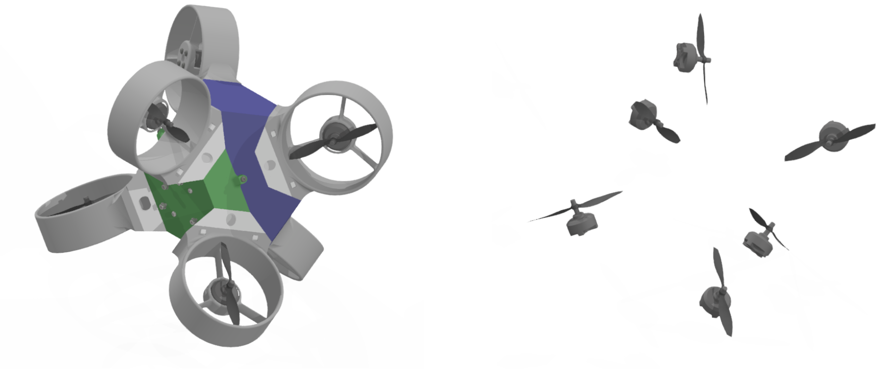
4.2 Evaluation 37

4.3 Design Space 38

4.4 Optimization 39

4.5 Results 40

This chapter will deal with the design of ACROBAT's robot body, in other words how should ACROBAT's propellers be placed. Figure 4.1(a) depicts ACROBAT's robot body and fig. 4.1(b) ACROBAT's propeller placement. First, the relation the propellers will be parametized. After this, a way to evaluate each design must be defined, so a evaluation strategy will be presented. Lastly, the optimization problem will be formulated and numerically solved.



(a) ACROBAT's first iteration robot body.

(b) ACROBAT.

Figure 4.1: ACROBAT's first iteration robot body propeller placement.

4.1 Parameterization

Considering the targeted six propellers, the actuation matrix described in eq. (2.22) is a 6×6 matrix. $\boldsymbol{A}$ depends on propeller constants K_1 and K_2. To bypass this dependency, we divide eq. (2.21) by K_1 resulting in

$$\boldsymbol{a_i'} = \begin{bmatrix} \boldsymbol{u_i} \\ \boldsymbol{r_i} \times \boldsymbol{u_i} - w_i \frac{K_2}{K_1} \boldsymbol{u_i} \end{bmatrix}. \tag{4.1}$$

Considering the size of the propellers that are expected to be used, around 4", the ratio $\frac{K_2}{K_1}$, takes values of magnitude 10^{-2}. Considering this, and for design purposes, we will make the approximation of this term ($\frac{K_2}{K_1} = 0$), resulting in

$$\boldsymbol{a_i'} = \begin{bmatrix} \boldsymbol{u_i} \\ \boldsymbol{r_i} \times \boldsymbol{u_i} \end{bmatrix}. \tag{4.2}$$

Despite this approximation, the methodology can be easily expanded to cover non-zero values of $\frac{K_2}{K_1}$.

The byproduct of eq. (4.2) is a dimensionless actuation matrix. Now, to fully define it, we need to specify each propeller position relative to the Center of Mass (CoM) $\boldsymbol{r_i}$ and thrust direction $\boldsymbol{u_i}$. Without loss of generality, let us assume that each propeller is located on a sphere of unit radius ($\|\boldsymbol{r_i}\| = 1$). Like depicted in fig. 4.2, $\boldsymbol{r_i}$ and $\boldsymbol{u_i}$ can be written as

$$\boldsymbol{r_i} = \begin{bmatrix} \cos(\gamma_i)\sin(\theta_i) \\ \sin(\gamma_i)\sin(\theta_i) \\ \cos(\theta_i) \end{bmatrix} \qquad \text{with } \theta_i \in [0,\pi] \wedge \gamma_i \in [0,2\pi) \tag{4.3}$$

$$\boldsymbol{u_i} = \begin{bmatrix} \cos(\alpha_i)\sin(\beta_i) \\ \sin(\alpha_i)\sin(\beta_i) \\ \cos(\beta_i) \end{bmatrix} \qquad \text{with } \beta_i \in [0,\pi] \wedge \alpha_i \in [0,2\pi) \tag{4.4}$$

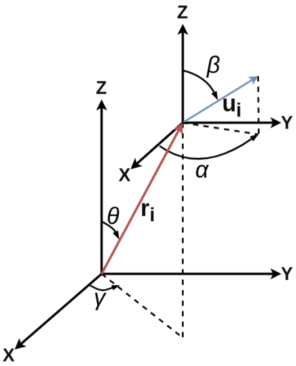


Figure 4.2: $\boldsymbol{r_i}$ and $\boldsymbol{u_i}$ description relative to the body's CoM.

Considering eq. (4.3) and eq. (4.4) expanded for the six propellers, the resulting vector of optimization variables $\boldsymbol{s}$ is given by $\boldsymbol{s} = (\gamma_1, \ldots, \gamma_6, \theta_1, \ldots, \theta_6, \beta_1, \ldots, \beta_6, \alpha_1, \ldots, \alpha_6)$.

4.1.1 ACROBAT's parameterization

Following the parameterization presented in this section, the geometric parameters which describe ACROBAT's first iteration robot body, depicted in fig. 4.1, are presented in table 4.1.

Table 4.1: ACROBAT's first iteration robot body geometric parameters, with $\phi = \arctan(\sqrt{2})$.

i	1	2	3	4	5	6
γ_i	0	$\pi/3$	$2\pi/3$	π	$4\pi/3$	$5\pi/3$
θ_i	$2\pi/3$	$\pi/3$	$2\pi/3$	$\pi/3$	$2\pi/3$	$\pi/3$
α_i	0	$11\pi/6$	$2\pi/3$	$\pi/2$	$4\pi/3$	$7\pi/6$
β_i	$\pi/2 - \phi$	$\pi/2$	$\pi/2 - \phi$	$\pi/2$	$\pi/2 - \phi$	$\pi/2$

4.2 Evaluation

To choose the $\boldsymbol{s}$ which results in the most apt robot body, we must define a performance evaluation criteria. One approach is to use the maximum force and torque possible in any direction [13]: Let us consider that the actuation signal $\boldsymbol{q}$ is bounded between -1 and 1, such that[1]

$$\|\boldsymbol{q}\|_\infty \leq 1. \tag{4.5}$$

Let us also assume that $\boldsymbol{M} = 0$ and that $\boldsymbol{F} = F\boldsymbol{e}$ where $\boldsymbol{e}$ is the force direction. From eq. (2.25) and eq. (4.5) we get that

$$\|\boldsymbol{q}\|_\infty = \|F\boldsymbol{b}^T\boldsymbol{e}\|_\infty \leq 1, \tag{4.6}$$

resulting in the upper bound of F given by

$$F \leq \frac{1}{\|\boldsymbol{b}^T\boldsymbol{e}\|_\infty}. \tag{4.7}$$

in any direction $\boldsymbol{e}$. Likewise, the maximum force attainable along a given direction $\boldsymbol{e}$ is given by

$$F_{\boldsymbol{e}}^{max} = \frac{1}{\|\boldsymbol{b}^T\boldsymbol{e}\|_\infty}. \tag{4.8}$$

To obtain the maximum force attainable in any direction, we must minimize eq. (4.8) in all directions. Considering that $\|\boldsymbol{b}^T\boldsymbol{e}\|_\infty = \max_i |\boldsymbol{b}_i^T\boldsymbol{e}|$ and $|\boldsymbol{b}^T\boldsymbol{e}| \leq \|\boldsymbol{b_i}\|$, maximum force in any direction is given by

$$F^{max} = \frac{1}{\max_i \|\boldsymbol{b}_i\|} = \min_i \frac{1}{\|\boldsymbol{b}_i\|} \qquad i \in \{1, \ldots, 6\}. \tag{4.9}$$

An analogous reasoning can be applied to obtain the maximum torque M^{max}, when $F = 0$, resulting in

$$M^{max} = \min_i \frac{1}{\|\boldsymbol{c}_i\|} \qquad i \in \{1, \ldots, 6\}. \tag{4.10}$$

4.3 Design Space

The parameterization described in section 4.1 has the drawback of leading to many symmetric designs, meaning that a given configuration can be redundantly described by different $\boldsymbol{s}$. For instance, switching a $\boldsymbol{r_i}$, $\boldsymbol{u_i}$ pair with other pair, leads to a different s that represents that same design. To avoid such cases, we shall constraint the propeller position $\boldsymbol{r_i}$ so that γ_i is ordered by increasing values of i:

$$\gamma_1 \leq \gamma_2 \leq \gamma_3 \leq \gamma_4 \leq \gamma_5 \leq \gamma_6 \tag{4.11}$$

Other types of symmetric designs can be obtained by choosing an arbitrary parameter vector $\boldsymbol{s}$,

---

[1] The infinity norm of $\boldsymbol{x}$ takes form $\|\boldsymbol{x}\|_\infty = \max\{|x_1|, \ldots, |x_n|\}$.

and rotating every $\boldsymbol{u_i}$ and $\boldsymbol{r_i}$ by an arbitrary rotation, arriving at a $\boldsymbol{s}$ which represents the same design, but has a different $\boldsymbol{s}$. To try mitigate the symmetric designs caused by rotations, we apply a few constraints to the design space. First, we set a fixed $\boldsymbol{r_1}$ ($\gamma_1 = 0$ and $\theta_1 = \frac{\pi}{2}$). A fixed $\boldsymbol{r_1}$ constrains the possible set of rotations causing symmetric designs to rotations around $\boldsymbol{r_1}$. To further constraint the symmetric designs, we will also set $\boldsymbol{r_2}$ so that $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ always belong to the same plain, no matter the $\boldsymbol{s}$. An easy way to do this, given the current parameterization, is to set $\theta_2 = \frac{\pi}{2}$ and let γ_2 be a free optimization variable. Considering this, we rewrite the vector of optimization variables $\boldsymbol{s}$ as $\boldsymbol{s} = (\gamma_2, \ldots, \gamma_6, \theta_3, \ldots, \theta_6, \beta_1, \ldots, \beta_6, \alpha_1, \ldots, \alpha_6)$.

Finally, given that the propellers are assumed to be bi-directional, we can set $\boldsymbol{u}_i$ to only one hemisphere given by: $\beta_i \in [0, \pi] \wedge \alpha_i \in [0, \pi)$ for $i \in \{1, \ldots, 6\}$. The resulting set of feasible designs $\mathcal{S}$ is a bounded box such that $\mathcal{S} = \{0 \leq \gamma_i \leq \pi, 0 \leq \theta_j \leq 2\pi, 0 \leq \beta_k \leq \pi, 0 \leq \alpha_k \leq \pi\}$ for $i \in \{2, \ldots, 6\}$, for $j \in \{3, \ldots, 6\}$, for $k \in \{1, \ldots, 6\}$.

4.4 Optimization

We wish to maximize both F^{max} and M^{max}, which can be translated into a minimization problem by considering their negative counterpart ($-F^{max}$ and $-M^{max}$). With the discussed bounds and constraints in mind, we can formulate the following multi-objective optimization problem:

$$\begin{aligned} \text{Minimize} \quad & (-F^{max}, -M^{max}) \\ \text{w.r.t.} \quad & \boldsymbol{s} \in \mathcal{S} \\ \text{subject to} \quad & \gamma_2 \leq \gamma_3 \leq \gamma_4 \leq \gamma_5 \leq \gamma_6 \,. \end{aligned} \tag{4.12}$$

Performance metrics F^{max} and M^{max} are calculated from $\boldsymbol{A^{-1}}$, which is numerically computed from $\boldsymbol{A}$, meaning they lack a closed-form solution. Consequently, their derivative is not available, making a derivative-free optimization methodology capable of addressing bounded problems with constraints necessary to estimate the Pareto front of the stated problem. Multiple algorithms have the capability of addressing this type of problem, namely evolutionary multi-objective approaches. However, based on our analysis we decided to use the Direct Multisearch for Multiobjective Optimization [31] algorithm.

4.5 Results

Point-wise approximation of the Pareto front can be found in fig. 4.3. The algorithm ran until a total of 4×10^5 objective function evaluations were generated starting from a random initialization. The Pareto front has a total of 447 non-dominated points.

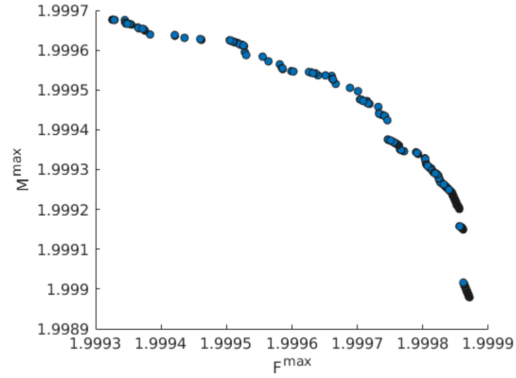


Figure 4.3: Point wise pareto-front approximation.

To check how the s that make the Pareto front in fig. 4.3 distribute over $\mathcal{S}$, we did the average and standard distribution of each optimization variable. The results are shown in table 4.2 and table 4.3. Observing the order of magnitude of the standard deviation presented by the solutions, 10^{-3}, and considering the range of values taken by the performance metrics in the Pareto front, one can assume that the set of solution is well represented by its own average. In fact, the data in fig. 4.3 suggests that the Pareto front is comprised of a single point, being the dispersion justified in the algorithm's numeric nature. If so, it shows that there are solutions to the problem in section 4.4 that are dominant in a way that maximize both F^{max} and M^{max}.

Table 4.2: Average and standard deviation of each design parameter relative to propeller position.

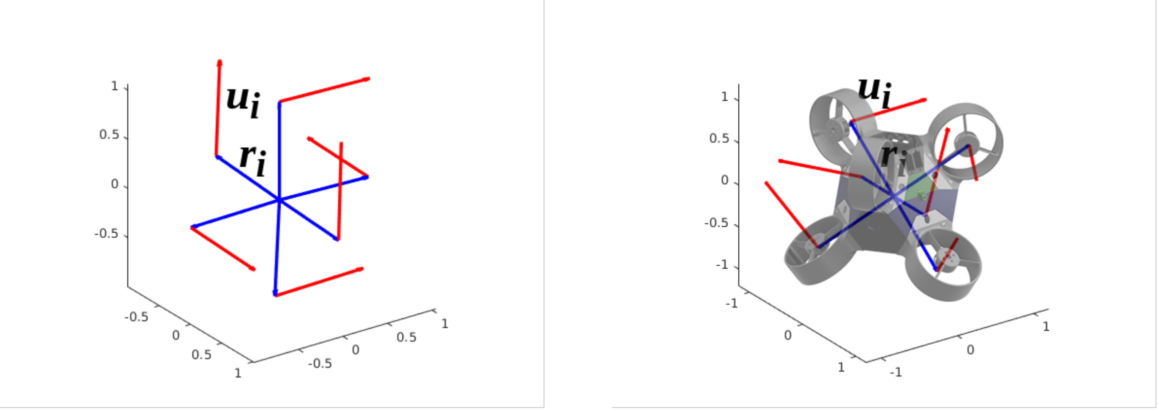
Variable	γ_2	γ_3	γ_4	γ_5	γ_6	θ_3	θ_4	θ_5	θ_6
Avg	1.603	3.202	4.694	4.701	4.785	0.134	3.108	1.577	0.013
Std (10^{-3})	3.252	12.80	5.106	3.642	0.945	2.771	2.717	16.04	11.01

Table 4.3: Average and standard deviation of each design parameter relative to thrust direction.

Variable	α_1	α_2	α_3	α_4	α_5	α_6	β_1	β_2	β_3	β_4	β_5	β_6
Avg	1.593	1.558	1.560	0.025	1.627	0.058	1.549	3.123	1.582	0.872	0.019	1.582
Std (10^{-3})	2.981	3.791	0.607	1.785	1.645	1.159	1.677	1.068	3.370	1.972	1.520	0.682

Figure 4.4 shows a visual representation of propeller placement and thrust direction, between the robot body design presented in table 4.2 and table 4.3, and ACROBAT's first iteration (for comparison purposes, we will consider $\|\boldsymbol{r_i}\| = 1$ in both cases). The design obtained from the Pareto front has a performance vector of $(F^{max}, M^{max}) = (1.9996, 1.9994)$. For comparison, ACROBAT has a performance

vector of $(F^{max}, M^{max}) = (2.0000, 2.0000)$. Performance-wise, both designs had similar results, which is not unexpected given that the problem defined in section 4.4 can hold multiple solutions (different s can have the same performance vector). Considering this, we believe the design presented in section 1.1 to be optimal, given that it has the same performance has an optimal design.



(a) Design obtained from the Pareto front.

(b) ACROBAT.

Figure 4.4: Representative diagram of the robot body.

5

Simulation and Prototype

Contents

5.1 Simulator Implementation 44
5.2 Prototype Overview 45
5.3 Manipulator 45
5.4 Propulsion System 48

5.1 Simulator Implementation

To understand how the system of robot body + manipulator dynamically behaves and to create a platform for testing and validating control algorithms, ACROBAT was implemented in a realistic physics simulator. As a first approach Gazebo[1] [32], an Open Source Physics Simulator for Robotics, was considered but it offered two drawbacks: First it had no out of the box support for propellers. This problem can be solved by using additional packages for multirotor support, like the RotorS library[2]. But the main problem is that it lacks support for closed-loop mechanisms, which is the case of ACROBAT's manipulator. With this limitations in mind, an alternative was searched and we settled on Webots[3] [33], another Open Source Physics Simulator for Robotics.

ACROBAT's first iteration implementation[4] in Webots is depicted in fig. 5.1. The implementation is modular, allowing for an easy change of geometric and/or physical parameters.

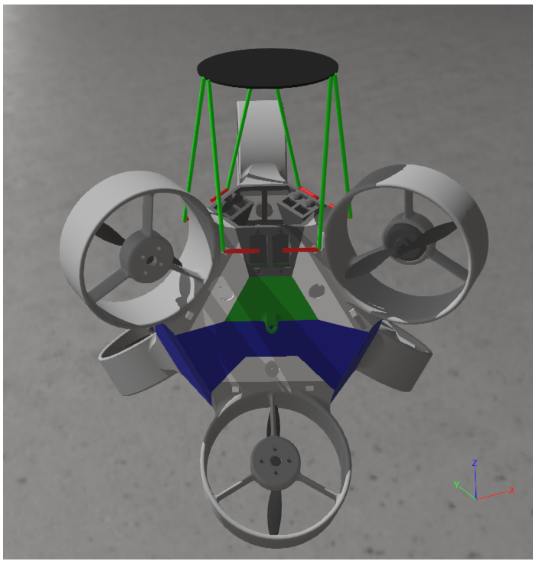


Figure 5.1: ACROBAT's first iteration implemented in Webots.

---

[1] `http://gazebosim.org/`
[2] `https://github.com/ethz-asl/rotors_simulator`
[3] `https://cyberbotics.com/`
[4] `https://github.com/CyberPoliceOfficer/Acrobat/tree/main/Webots`

5.2 Prototype Overview

To validate ACROBAT's design and help build a platform for the development of algorithms, namely for position control and cooperative AM, a prototype was aimed to be built. The prototype is expected to be tested in an air-bearing table, to emulate micro-gravity. Multiple experiments can be conceptualized, with fig. 5.2 showing one where two ACROBATs cooperatively 3D print a part, similar to the setup described in section 1.1, with a robot dedicated to material deposition and other dedicated to extrusion.

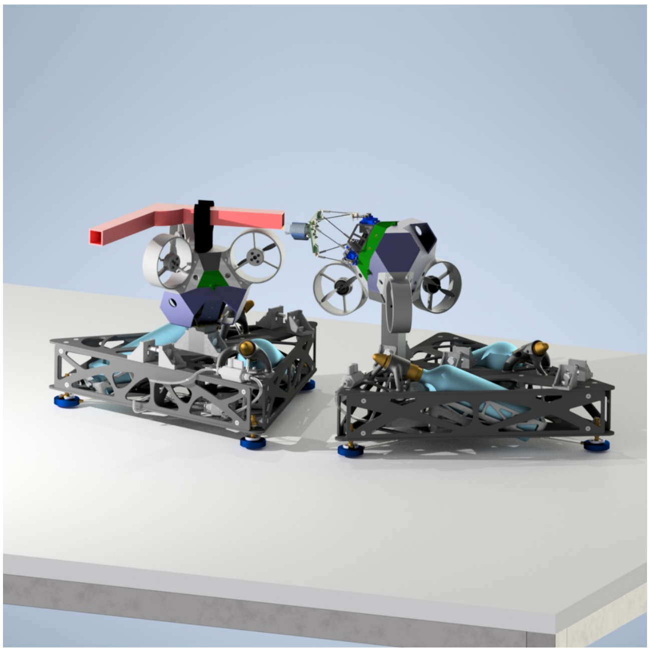


Figure 5.2: CAD depiction of a proposed experiment.

ACROBAT's robot body was designed to be able to move with six DoF, but giving the limitations of the air-bearing table, it will only be able of movements within the table's plane. This means it will only be able to move in two translational directions and rotate in one, being reduced to three DoF. With this limitations in mind, we can safely remove three propellers from ACROBAT and mount it so that all the remaining propellers have their thrust vector parallel with the table's plane, like at depicted in fig. 5.2.

With this considerations in mind, the prototype efforts will be divided into two parts: The construction of the 6-RSS manipulator, discussed in section 5.3, and the propulsion system, discussed in section 5.4.

5.3 Manipulator

To validate the design and study the design options, ACROBAT's first iteration manipulator, with geometric parameters in table 3.1, was built. The prototype with, an extrusion head mounted on the mobile platform, is depicted in fig. 5.3.

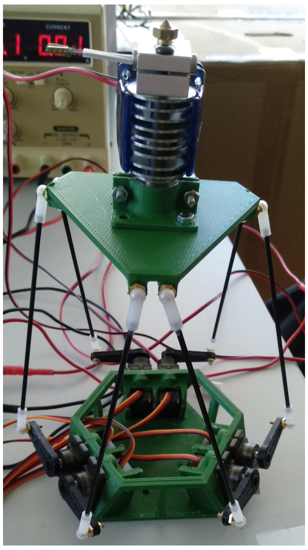


Figure 5.3: Six-RSS Manipulator with a mounted extrusion head.

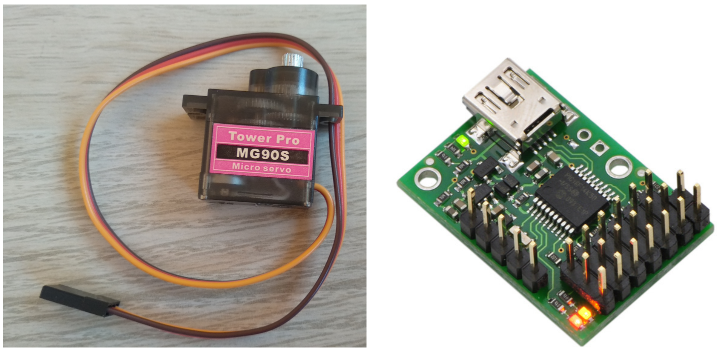
The manipulator's prototype had to rely on commercially available and 3D printable parts. Table 5.1 condensates the used material, and their source. The S-S joint was made from a carbon fiber rod. This rod had to be cut and iteratively filed to reach the specified length.

Table 5.1: Manipulator material.

Part	Source
Mobile Platform	3D printed using Polylactic Acid (PLA)
Fixed Platform	3D printed using PLA
R-S link	3D printed using PLA
S-S link	Carbon Fiber rod
joint at H_k	M2 Ball Socket Joint
joint at M_k	M2 Ball Socket Joint

5.3.1 Actuation System

To actuate the revolute joints, six MG90S 180° servos, with one depicted in fig. 5.4(a) were chosen. The MG90S was selected because of fitting the size requirements, being relatively inexpensive and light (weighing just 13.4 grams). Another option would be use stepper motors instead of servos. This type of motor has better accuracy at the same price range, but they were discarded given their relatively high weight (for reference, a NEMA 17 stepper motor weights 240 grams). The chosen MG90S are capable of movements in a arc of π radians.



(a) MG90S Servo.

(b) Micro Maestro 6-Channel USB Servo Controller from Pololu.

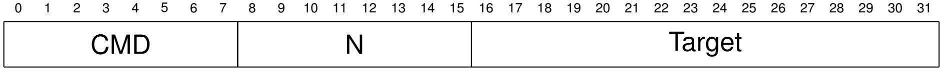
Figure 5.4: ACROBAT's manipulator actuation components.

The communication with each servo is done via Pulse Width Modulation (PWM): Following the MG90S documentation, a PWM of $1000\mu s$ rotates the servo to $-\frac{\pi}{2}$ radians and a PWM of $2000\mu s$ to $\frac{\pi}{2}$ radians. Ideally, one would actuate each servo to 0, by setting the PWM to $1500\mu s$, and install the servo horn. Unfortunately in practice, giving the servo's geared system for installing the horn, this is not possible. In other words, each servo has a q_k^0 PWM value which must be calibrated, close to $1500\mu s$, which corresponds to the k-th horn 0 radian position ($\alpha_k = 0$). Given a desired actuator angle α_k, the corresponding k-th servo PWM input q_k is given by

$$q_k = q_k^0 + (-1)^k \frac{\Delta q}{\Delta \alpha} \alpha_k \qquad k \in \{1, \ldots, 6\} \tag{5.1}$$

where $\frac{\Delta q}{\Delta \alpha}$ is a ratio of PWM μs per radian. In the case of the used MG90S, $\frac{\Delta q}{\Delta \alpha} = \frac{1000}{\pi}$.

Directly controlling the six servos would require a large number of General Purpose Input/Output (GPIO) ports, so we opted to use a intermediary servo driver as a communication layer between the on-board computer and servos. For this task, the Micro Maestro Controller from Pololu, depicted in fig. 5.4(b), was used. The on-board computer communicates with the Maestro via serial port. Following the Maestro documentation[5], to set the k-th servo to a target PWM q_k the following byte array must be written on the serial port:



Where CMD is the desired command, which for position control is $(132)_b$, N the target servo with $N = k - 1$ and Target q_k in quarter microseconds.

---

[5] https://www.pololu.com/docs/0J40

5.4 Propulsion System

ACROBAT's ground prototype was projected to have a propulsion system composed of three propellers. Figure 5.5 depicts ACROBAT mounted with two ducted fans and their respective propellers and brush-less DC motors. As previously stated, the propellers thrust vector is parallel to the plane in which the robot body is mounted.

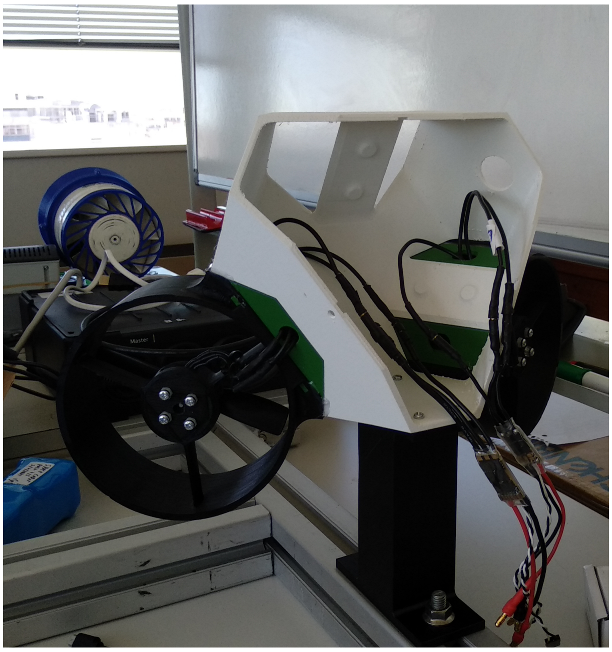


Figure 5.5: ACROBAT's prototype mounted on a stand with two ducted fans installed.

Every structural part was designed to be 3D printed, with most parts being printed in Polyethylene Terephthalate Glycol-modified (PETG) or PLA, given their easiness to work with while maintaining a solid structural integrity. To power the system, a 14.8V 4-Cell 25C LiPo Battery coupled with a power distribution board was used.

5.4.1 Brush-less DC motors, ESC and propellers

To spin the propellers, Cobra CM-2204 2300 kv brush-less DC motors, depicted in fig. 5.6(a), were used. This motors were chosen given their good price-performance, availability and experience in previous projects.

A brush-less DC motors is electronically controlled by a Electronic Speed Controller (ESC). The ESC receives a PWM signal and controls the motor's rotation speed accordingly. The chosen ESC needed two important features: The ability to invert the spinning direction, while the robot is in operation and without rewiring, and having a low dead-band, meaning they must be able to command the motors do spin at relatively low Rotations Per Minute (RPM). With this considerations in mind, the ESC SEFM 30A from Aikon Electronics was chosen.

Ideally, given ACROBAT's configuration, the propellers should have the same thrust coefficient, recall eq. (2.17), no matter the spinning direction. Unfortunately, given the use of commercially available parts, which follow a high demand for drone parts that clearly favor one direction of rotation, a fitting bi-directional propeller is not available. Consequently, the 4 inch propellers from HQProp, depicted in fig. 5.6(b), were used.

(a) Cobra CM-2204 2300kv. **(b)** HQProp 4 inch propeller.

Figure 5.6: Propulsion system components.

5.4.2 Test-bench

To study the performance of the parts discussed in the aforementioned section, the Series 1580 test stand from RCbenchmark[6], depicted in fig. 5.7, was used. This stand allows the user to send PWM signals to the ESC while measuring the propeller's RPM, thrust and torque. The propellers RPM can be calculated with the number of magnetic poles by analysing the ESC output or be optically measured by installing a white tape on the motor. Given the ducted fan design, the optical method can not be installed, so the electrical method was adopted.

---

[6] https://www.tytorobotics.com/products/thrust-stand-series-1580

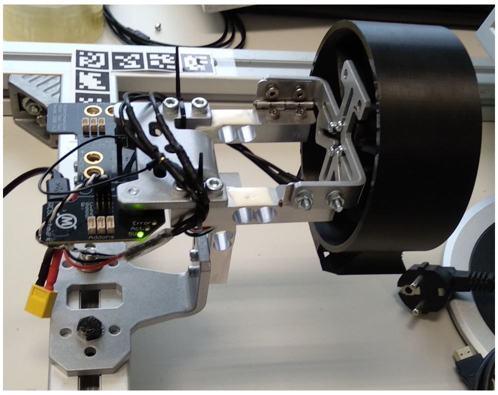
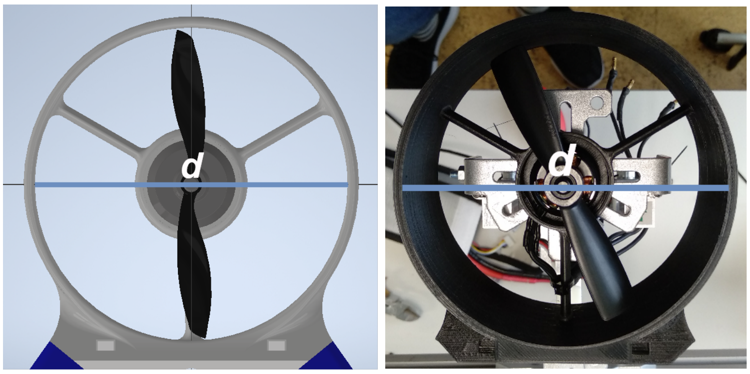


Figure 5.7: RCbenchmark Series 1580 Test Stand mounted with a ducted fan and propeller.

5.4.3 Ducted fan

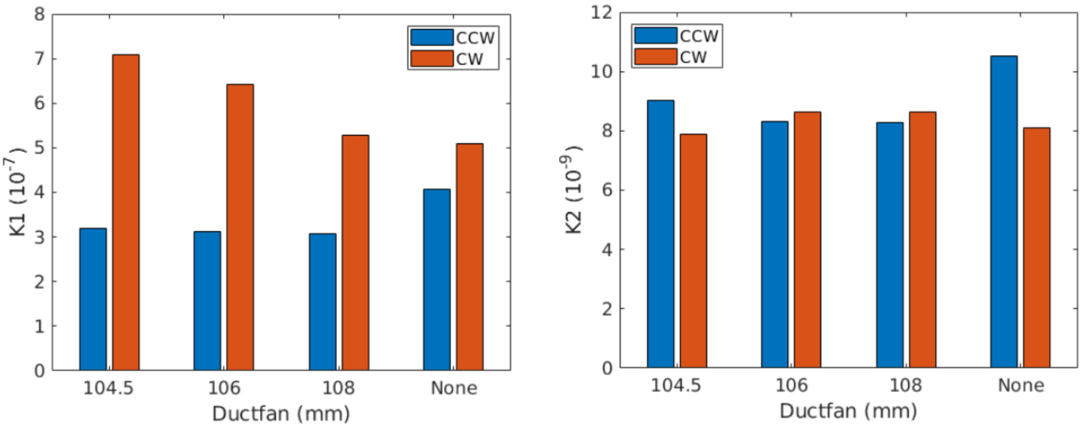
ACROBAT was idealized to have a ducted fan design, meaning that the propeller is cased by a tight duct, as depicted in fig. 5.8. In theory, and if properly built, this design choice would increase the propeller's performance while vastly improving the vehicle's safety. In practice there are a few challenges in the implementation of a ducted fan. It suffers from material compression during the printing, which creates a deviation between the desired diameter and the actual diameter. Other problem is that the propeller expands with a increasing rotating speed, creating a deviations in the propeller diameter. This challenges mean that manufacturing the must fit ducted fan would require a long iterative process of trial and error.



(a) Ducted fan CAD. **(b)** Ducted fan photo.

Figure 5.8: Ducted fan with the propeller installed. d is the ducted fan diameter.

To evaluate the effect of the ducted fan diameter in the propeller's performance, the study shown in fig. 5.9 was conducted. The results were obtained by doing a RPM swipe for each ducted fan, while measuring the thrust and torque. With this data, the must fit K_1 and K_2, following eqs. (2.17) and (2.18) and in the least squares sense, was calculated and plotted in fig. 5.9. The experiment was conducted for a Clockwise (CW) and Counter Clockwise (CCW) direction of rotation and each ducted fan was 3D printed using PLA.



(a) K_1.

(b) K_2.

Figure 5.9: Bar graph comparing the K_1 and K_2 of different ducted fan diameters.

Looking at fig. 5.9, we can clearly the uni-directional nature of the propeller, by analysing the values of K_1 for when there is no ducted fan installed (None in fig. 5.9). One assumption done in section 4.2 was that K_2 is negligible compared with K_1, assumption that is confirmed by the experimental data, given that K_1 is approximately two orders of magnitude bigger than K_2. But the most important take-away is that using a ducted fan improves the thrust coefficient K_1 in the direction of rotation that the propeller was designed to rotate, CW, but deteriorates in the opposite direction. A decreasing d leads to a increasing K_1 while rotating CW, but on the other hand K_1 does not increase with a decreasing d while rotating CCW, with K_1 being unchanged. Looking at the torque coefficient K_2, we conclude that it is not as affected as the thrust coefficient, and remaining roughly the same.

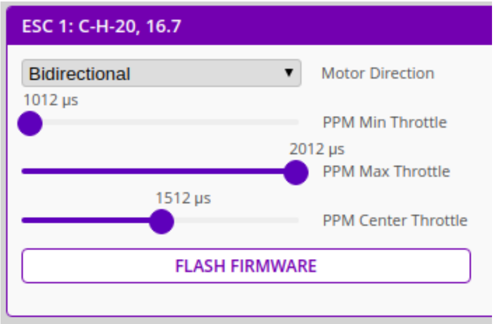
5.4.4 Calibration

One important step in the prototype's realization is the calibration of the propeller-motor-ESC system. Given that the equipment to measure the thrust and RPM is available, as described in section 5.4.2, one possible approach is to find the relationship between the PWM and the propeller's thrust.

First, we must verify that each one of the three used ESC are in the bidirectional mode, and that the PWM bandwidth is properly set. To flash the ESC firmware, the BLHeli Configurator[7] was used, with the relevant settings shown in fig. 5.10.

---

[7] BLHeli Configurator can be found on the Google Chrome store.



ESC 1: C-H-20, 16.7

Motor Direction: Bidirectional

PPM Min Throttle: 1012 µs

PPM Max Throttle: 2012 µs

PPM Center Throttle: 1512 µs

Figure 5.10: Cropped screenshot from BLHeli Configurator.

The procedure for data acquisition was similar to the one described in section 5.4.3, and the setup is depicted in fig. 5.7. Ducted-fans with $d = 108mm$ were used. Even if their performance is inferior, having a ducted-dan is important for safety reasons and given that there were already three of this fans printed, they were used given that the time was constrained. A PWM sweap was conducted, with the average of a sample size of 10 taken in each PWM value. The experiment was repeated for the three propulsors, with results found in fig. 5.11. To minimize deviations, the used battery was fully charged in each experiment.

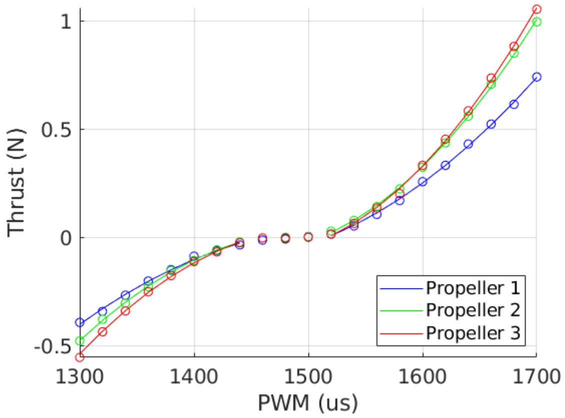


Figure 5.11: Thrust data taken from each propeller. Each point presents the average of 10 samples, and the curve is a second order polynomial fit.

One thing to note, clearly visible in fig. 5.11, is that there's an actuation dead-band. The exact interval is different for each motor/ESC pairing, but empirical evidence suggest that we can safely actuate below $1440\mu s$ and above $1520\mu s$. To fit the acquired data, a second-order polynomial was used. Clearly the model in eq. (2.17) suggests a second order relation-ship. Considering the uni-directional nature of the propeller, a different fit was used for the two directions of rotation, with the corresponding data used accordingly. An interesting observation is the deviations between each propeller. Multiple explanations can be given, but the important lesson is that we must calibrate each propeller individually.

6

Conclusion and Future Work

Contents

6.1 Future Work 54

In conclusion, a methodology for the optimal design of an areal robot for in-orbit construction and assembly was presented, solving the design problem stated in section 1.2.

ACROBAT's parallel manipulator was studied: The inverse kinematic problem was solved, and the inverse kinematic Jacobian found. Both this results will be needed for the manipulator control. A design methodology was also presented, with a point-wise approximation of the pareto-set available. From this pareto-set, the relationship between the performance metrics was understood: (1) A higher workspace volume requires a loss in accuracy (2) The translational accuracy scales linearly with the rotational accuracy. An approach to find an optimal version of ACROBAT's first iteration manipulator from the Pareto set was proposed, and the result illustrated, with the optimal version having a workspace roughly fifteen times larger, while maintaining the same levels of accuracy.

A methodology for the design of ACROBAT's body was presented, with relevant metrics defined. The results revealed a interesting fact, that not only ACROBAT's body had the same performance as a solution found on the Pareto set, but also that it maximizes both trust and torque. Meaning that there are multiple solutions which maximize both trust and torque. With this in mind, we can safely guarantee that the current ACROBAT body is optimal.

In order to create a platform for the validation and testing of algorithm, ACROBAT was implemented on the physics simulator Webots.

Efforts towards the development of the ACROBAT's prototype were conducted. The parallel manipulator was assembled, with the accompanying software for kinematic control developed. The effect of having a ducted fan was studied, for different duct diameters. In the tested propeller, having a duct lead to a increase in the trust coefficient while rotating CW, with the largest gains being seen on the tighter ducts. While rotating CCW, having a duct worsened the trust coefficient. The relationship between the control PWM and the propeller trust and torque was found.

6.1 Future Work

The parallel manipulator design approach can be enhanced in multiple ways. Other types of tasks could require other metrics. For instance, a more general manipulator, for instance for pick and place, might prefer dexterity over accuracy and drop the use of accuracy related metrics. The methodology can also be expanded to deal with constraints imposed by the un-actuated joints. This expansion is not as straightforward as it sounds, giving that the constraints are dependent on the un-actuated joints position, which is a designer choice. What this means in practise is that are ways to build to build the manipulator less constraining than others. This problem can be formulated and solved as an optimization problem, but in the context of design would require the multi-objective optimization solver to solve an optimization problem each time it needs to evaluate a performance metrics. Other enhancements include using a more general workspace, specifically the reachable workspace, at the cost of a considerably heavier computational burden. If the requirements of the manipulator's task are well defined and understood, for instance the accuracy of the manipulator must be higher than a certain specification, the problem can be formulated to find the set in the design parameter space which satisfies said constraints.

In the case of the ACROBAT's body, the results suggest that there is a set of designs which maximize both thrust and torque. One question that can be asked is that if there is a closed form expression to the set of solution in the design parameter space. Or similarly, searching if there is a condition in the design space that guarantees that the design maximizes both thrust and torque. The methodology can also be expanded to deal with other types of actuation system where the actuators only apply thrust in one direction, like in cold gas thrusters.

In the case of the prototype, some can still be done. In the manipulator's case, given the current servos, a calibration procedure must be done. This can be done by either calibrating each servo individually or using a motion capture system. The manipulator performance would also be greatly improved by using better servos, namely servos that be accurately controlled in position in speed, to guarantee a better dynamic response. The un-actuated joints must also be properly lubricated to reduce unwanted friction. Other prototype related tasks imply the configuration and firmware creation to actuate the propellers. The fastest way to do so is to use the GPIO ports of a Single-board computer, and create a Robot Operating System (ROS) topic for actuation. The IMU must also be mounted with an anti-vibration mount, to gurantee better readings.

Bibliography

[1] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*, 2nd ed. Springer Publishing Company, Incorporated, 2016.

[2] F. Pierrot, P. Dauchez, and A. Fournier, "HEXA: a fast six-DOF fully-parallel robot," in *Fifth International Conference on Advanced Robotics 'Robots in Unstructured Environments*. IEEE, 1991.

[3] A. C. Owens and O. L. de Weck, "Systems analysis of in-space manufacturing applications for the international space station and the Evolvable Mars Campaign," *AIAA Space and Astronautics Forum and Exposition, SPACE 2016*, no. September, pp. 1–18, 2016.

[4] I. Campbell, D. Bourell, and I. Gibson, "Additive manufacturing: rapid prototyping comes of age," *Rapid Prototyping Journal*, vol. 18, no. 4, 2012.

[5] T. Prater, N. Werkheiser, F. Ledbetter, D. Timucin, K. Wheeler, and M. Snyder, "3D Printing in Zero G Technology Demonstration Mission: complete experimental results and summary of related material modeling efforts," *International Journal of Advanced Manufacturing Technology*, vol. 101, no. 1-4, pp. 391–417, 2019.

[6] J. Vale, A. Rocha, M. Leite, and R. Ventura, "A multi-objective optimization approach to the design of a free-flyer space robot for in-orbit manufacturing and assembly," *International Conference on Multidisciplinary Design Optimization of Aerospace Systems - AeroBest 2021*, pp. 517–536, 2021.

[7] J. P. Merlet, *Parallel Robots*, 2nd ed. Springer Publishing Company, Incorporated, 2010.

[8] D. Stewart, "A Platform with Six Degrees of Freedom," *Proceedings of the Institution of Mechanical Engineers*, vol. 180, no. 1, 1965.

[9] F. Pierrot, C. Reynaud, and A. Fournier, "Delta: A simple and efficient parallel robot," *Robotica*, vol. 8, no. 2, 1990.

[10] S. Patel and T. Sobh, "Manipulator Performance Measures - A Comprehensive Literature Survey," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 77, no. 3-4, pp. 547–570, 2014.

[11] J. K. Salisbury and J. J. Craig, “Articulated Hands,” *The International Journal of Robotics Research*, vol. 1, no. 1, pp. 4–17, 1982.

[12] P. Cardou, S. Bouchard, and C. Gosselin, “Kinematic-sensitivity indices for dimensionally nonhomogeneous jacobian matrices,” *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 166–173, 2010.

[13] P. Roque and R. Ventura, “Space CoBot: Modular design of an holonomic aerial robot for indoor microgravity environments,” *IEEE International Conference on Intelligent Robots and Systems*, vol. 2016-Novem, pp. 4383–4390, 2016.

[14] B. McCormick, *Aerodynamics, Aeronautics, and Flight Mechanics*. Wiley, 1994.

[15] R. T. Marler and J. S. Arora, “Survey of multi-objective optimization methods for engineering,” *Structural and Multidisciplinary Optimization*, vol. 26, no. 6, pp. 369–395, 2004.

[16] X. Zhang, M. Li, J. H. Lim, Y. Weng, Y. W. D. Tay, H. Pham, and Q. C. Pham, “Large-scale 3D printing by a team of mobile robots,” *Automation in Construction*, vol. 95, no. August, pp. 98–106, 2018. [Online]. Available: https://doi.org/10.1016/j.autcon.2018.08.004

[17] P. Chermprayong, K. Zhang, F. Xiao, and M. Kovac, “An Integrated Delta Manipulator for Aerial Repair: A New Aerial Robotic System,” *IEEE Robotics and Automation Magazine*, vol. 26, no. 1, pp. 54–66, 2019.

[18] S. C. Patané, E. R. Joyce, M. P. Snyder, and P. Shestople, “Archinaut: In-space manufacturing and assembly for next-generation space habitats,” *AIAA SPACE and Astronautics Forum and Exposition, SPACE 2017*, no. 203999, 2017.

[19] S. C. Patané, J. J. Schomer, and M. P. Snyder, “Design reference missions for archinaut: A roadmap for in-space manufacturing and assembly,” *2018 AIAA SPACE and Astronautics Forum and Exposition*, no. September, pp. 1–7, 2018.

[20] D. Miller, A. Saenz-Otero, J. Wertz, A. Chen, G. Berkowski, C. Brodel, S. Carlson, D. Carpenter, S. Chen, S. Cheng, D. Feller, S. Jackson, B. Pitts, F. Perez, and J. Szuminski, “Spheres: A testbed for long duration satellite formation flying in micro-gravity conditions,” *Advances in the Astronautical Sciences*, vol. 105 I, pp. 167–179, 2000.

[21] M. Bualat, J. Barlow, T. Fong, C. Provencher, T. Smith, and A. Zuniga, “Astrobee: Developing a free-flying robot for the international space station,” *AIAA SPACE 2015 Conference and Exposition*, pp. 1–10, 2015.

[22] S. Mitani, M. Goto, R. Konomura, Y. Shoji, K. Hagiwara, S. Shigeto, and N. Tanishima, “Int-Ball: Crew-Supportive Autonomous Mobile Camera Robot on ISS/JEM,” *IEEE Aerospace Conference Proceedings*, vol. 2019-March, pp. 1–15, 2019.

[23] F. Hao and J. P. Merlet, “Multi-criteria optimal design of parallel manipulators based on interval analysis,” *Mechanism and Machine Theory*, vol. 40, no. 2, pp. 157–171, 2005.

[24] R. Badora, “Stability properties of some functional equations,” *Springer Optimization and Its Applications*, vol. 52, pp. 3–13, 2012.

[25] K. Tanakitkorn, “Design Optimisation of a 6-RSS Parallel Manipulator via Surrogate Modelling,” *IOP Conference Series: Materials Science and Engineering*, vol. 501, no. 1, 2019.

[26] Z. Gao, D. Zhang, and Y. Ge, “Design optimization of a spatial six degree-of-freedom parallel manipulator based on artificial intelligence approaches,” *Robotics and Computer-Integrated Manufacturing*, vol. 26, no. 2, pp. 180–189, 2010. [Online]. Available: http://dx.doi.org/10.1016/j.rcim.2009.07.002

[27] R. E. Stamper, L. W. Tsai, and G. C. Walsh, “Optimization of a three DOF translational platform for well-conditioned workspace,” *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 4, no. April, pp. 3250–3255, 1997.

[28] Y. Lou, G. Liu, N. Chen, and Z. Li, “Optimal design of parallel manipulators for maximum effective regular workspace,” *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pp. 795–800, 2005.

[29] Y. Lou, G. Liu, and Z. Li, “Randomized optimal design of parallel manipulators,” *IEEE Transactions on Automation Science and Engineering*, vol. 5, no. 2, pp. 223–233, 2008.

[30] F. Szufnarowski, “Stewart platform with fixed rotary actuators : a low cost design study,” *Faculty of Technology, Bielefeld University, Germany*, 2013.

[31] C. P. Brás and A. L. Custódio, “On the use of polynomial models in multiobjective directional direct search,” *Comput. Optim. Appl.*, vol. 77, pp. 897–918, 2020.

[32] N. Koenig and A. Howard, “Design and use paradigms for gazebo, an open-source multi-robot simulator,” in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, 2004, pp. 2149–2154 vol.3.

[33] O. Michel, “Webots: Professional mobile robot simulation,” *Journal of Advanced Robotics Systems*, vol. 1, no. 1, pp. 39–42, 2004. [Online]. Available: http://www.ars-journal.com/International-Journal-of-Advanced-Robotic-Systems/Volume-1/39-42.pdf