École Centrale de Nantes   Warsaw University of Technology

MASTER ERASMUS MUNDUS

EMARO "European master on Advanced Robotics"

2011 / 2012

Master Thesis Final Report

Presented by

Miguel MORALES GONZÁLEZ

On 29/08/2012

Title

Analysis of the performances of serial robots with 4-DOF
based on RPaR orthogonal architecture

Jury

President:	Wisama Khalil	Professor, IRCCyN, ECN, EMARO Coord.
Evaluators:	Wisama Khalil	Professor, IRCCyN, ECN, EMARO Coord.
	Stéphane Caro	Researcher, IRCCyN, ECN
	Philippe Wenger	Director of Research, IRCCyN, CNRS
	Sébastien Briot	Researcher, IRCCyN, ECN

Supervisors:
Philippe Wenger & Sébastien Briot
Laboratory: Institut de Recherche en Communications et Cybernétique de Nantes

Co-supervisor:
Dr. Krzysztof Mianowski, Warsaw University of Technology

Acknowledgments

The EMARO program has been one of the most memorable experiences of my life. Besides the knowledge and academic and professional abilities I have acquired after these two years, I have had the honor of working with excellent professors and world-class researchers, and making long-lasting friendships with people from all over the world.

I would like to thank my supervisors Prof. Philippe Wenger and M. Sébastien Briot for all their help, support and advice throughout my master thesis work. Their expertise, professionalism and guidance were essential to the accomplishment of the objectives of this thesis.

I would like as well to thank Prof. Wisama Khalil because his leadership, experience and above all, his human qualities, have made of EMARO a great program. I will we always be grateful for the opportunity of studying this master program.

Furthermore, I would like to express my gratitude to Prof. Tereza Zielinska, to my co-supervisor Dr. Krzysztof Mianowski, and all the other professors and staff of Warsaw University of Technology and École Centrale de Nantes.

Finally, I would like to thank my mom, Patricia, for her love, unconditional support and great advice throughout all my studies; my dad, Miguel, for his support, good example and encouragement; my grandmother, Trinidad, for her love, smiles and all the good cooking recipes; my brothers, Hugo and Brandon, for being my best friends and for always supporting me in my endeavors; Michel, for his friendship and great conversation and tips; and specially thank to my good friend, Denis Ćehajić, who was a great teammate but above all always an excellent friend and person from whom I learned a lot.

Contents

List of figures

List of tables

Nomenclature

DOF	Degrees of Freedom
mDH	modified Denavit-Hartenberg
3R	3-Revolute joints
RPaR	Revolute Parallelogram Revolute joints
RPaPa	Revolute 2-Parallelogram joints
PUU	Prismatic 2-Universal joints
DGM	Direct Geometric Model
IGM	Inverse Geometric Model
IKM	Inverse Kinematic Model
DDM	Direct Dynamic Model
IDM	Inverse Dynamic Model
SCARA	Selective Compliance Assembly Robot Arm
FEA	Finite Element Analysis
MSA	Matrix Structural Analysis
VJM	Virtual Joint Method
MFE	Manipulating-Force ellipsoid
GIE	Generalized Inertial Ellipsoid
DME	Dynamic Manipulability Ellipsoid
IME	Inertia Matching Ellipsoid
EWA	Efficient Working Areas
KCh	Kinematic Chain

Introduction

Robot manipulators are extensively used in industry in a wide range of applications, from processes involving food preparation to the manufacturing of parts for the automotive and aerospace industries. They have clearly transformed the way in which objects of common use are manufactured and increased the productivity of industrial processes. Despite of the numerous advances in the robotics area and mechanical design, there is a family of robot manipulators that has dominated the industrial applications for many years: the ones based on the anthropomorphic architecture. Its name originates from the fact that the joints of these manipulators are arranged in a way that resembles in some degree the structure of the human shoulder and arm articulations. In the last years, there has been an increasing interest in another kind of architecture called orthogonal (which is called like that because any two consecutive joints of the manipulator are perpendicular with each other). More notably, it has been recently proved that some robots belonging to this family (namely the *3R* orthogonal robot) exhibit good performances in terms of workspace size and kinematic properties, and they present better dynamic properties than their counterparts of the anthropomorphic architecture (*3R* anthropomorphic robot).

Therefore, following this research momentum, the **objective of this thesis** is to study the dynamic performances of another type of robots belonging to the orthogonal family, the 4-DOF *RPaR* orthogonal manipulators, and fairly compare them to their equivalent in the anthropomorphic family, the 4-DOF *RPaPa* anthropomorphic manipulators, to demonstrate that the former ones present better dynamic performances than the later ones, as it is expected.

Most of the previous comparison studies between orthogonal and anthropomorphic manipulators have been focused on kinematic properties. However, as it will be shown later, a novel method compares them in terms of dynamic performance [1], [2]. More importantly, in order to do this comparison it uses the concept of applying exciting trajectories along the workspace of the manipulators to make them experience the highest possible joint torques. The proposed methodology is not found in previous literature for performing this kind of analysis, but this approach offers to perform a better dynamic analysis than other methods; thus, a similar methodology is applied in the current thesis work.

Furthermore, since a comparison between two different families of manipulators will be done, it has to be assured that it is performed as fair as possible, which means to carefully select a manipulator of each family to be compared. In the previous investigation, the manipulators to compare were selected with the help of a kinematic performance index. In the current research work a more robust approach is implemented to complement the previous study. In this thesis it is proposed to select the design parameters of the robots (the mass of the links) via an optimization procedure. A

technology-oriented optimization procedure such as the one presented in [3] turns out especially appropriate for this purpose.

In order to apply the optimization procedure, some design objectives and constrains must be established. A measure or technology index is useful for this purpose. There are several indices that may be used in an optimization procedure. As it will be explained later, due to the structure of the manipulators under study, the stiffness of the manipulator is of particular interest for the kind of robots analyzed [3], thus this parameter is selected as measure. To calculate the stiffness of the manipulator, the method presented in [4] is applied because it is a suitable methodology to be used for an optimization procedure as it will be justified later.

Accordingly, the detailed objectives of this thesis can be summarized as follows:

- To study the general characteristics of the *4-DOF RPaR* orthogonal and *4-DOF RPaPa* anthropomorphic manipulators.

- To develop the elastostatic model of both types of manipulators to obtain their stiffness matrices.

- According to the performance measure (based on the stiffness), to perform a technology-oriented optimization procedure to find some design parameters of the robots (the mass of the links).

- To apply the cited dynamic analysis to compare both manipulators and conclude which one has better performances.

Another goal of this thesis as important as the ones already mentioned, but that it may not be noticeable at first sight, is to *propose a robust methodology to analyze and compare in a fair way the dynamic performances of manipulators*. This will complement the previous research work and probably it will offer a research and discussion topic for interested investigators.

According to the objectives described, this document is divided in five chapters. The first chapter has the aim of summarizing the work of [1] and [2]; meanwhile, the second to the fifth chapters have the aim of discussing the current research work. Of these ones, the third chapter discusses the elastostatic modeling formulation and its application to the manipulators under study. Furthermore, the fourth chapter presents the parameter optimization procedure. Finally, the fifth chapter presents the dynamic performance analysis of both manipulators.

Chapter 1

Recapitulation of previous work

There has been an increasing interest in the study of manipulators belonging to the orthogonal family with the aim, among others, of comparing them with their counterparts in the anthropomorphic family. One of these studies [1] and [2] consisted of the dynamic performance comparison between the *3R* orthogonal robot and the *3R* anthropomorphic robot and it is particularly relevant for the purposes of this thesis. Therefore, the methodology, main ideas, results and conclusions of this research work will be explained in this chapter.

1.1 Methodology of previous research

The goal of the previous research was to make a dynamic performance analysis of the *3R* orthogonal serial manipulator and compare it against the *3R* anthropomorphic serial manipulator. It is important to present the approaches followed and the results obtained by the previous research because a similar methodology is applied in the development of this thesis.

It is possible to summarize the methodology of the previous research in the following flow diagram:
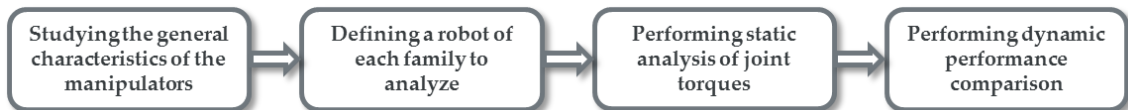
Figure 1.1 Methodology followed by the previous research

The first step in the comparison process was to study the general characteristics of the family of manipulators under investigation; thus **Section 1.2** has the objective of giving a brief description of the two manipulators analyzed by the previous research. After that, **Section 1.3** explains the approach that was followed to select the manipulators that were to be compared. **Section 1.4** presents the static analysis of the joint torques of the robots that was performed. Finally, **Section 1.5** discusses the dynamic performance procedure applied on both manipulators and the conclusions they obtained.

It is important to mention that since some of the concepts studied in this chapter were also used in this research project, they will be explained in detail in the corresponding chapter (**Chapter 3 to 5**). The reader will be referred to these chapters appropriately.

1.2　*3R* anthropomorphic and *3R* orthogonal Manipulators

This section describes some general concepts of anthropomorphic and orthogonal architecture manipulators, with emphasis on the *3R* orthogonal architecture.

1.2.1　*3R* anthropomorphic manipulators

Serial robot manipulators with anthropomorphic architecture (anthropomorphic shoulder) consist of 3 revolute joints in serial configuration in which the last two axes of the shoulder are parallel between each other and orthogonal to the first joint axis, as shown in the diagram of **Figure 1.2**. For 6-DOF robots based on this architecture, the following three joints, also known as the wrist, normally intersect at right angles at a common point, thus defining a spherical joint. This structure constitutes a typical example of a 6-DOF robot [5]. They are also sometimes called articulated, jointed or elbow manipulators [6]. A schematic view of an anthropomorphic manipulator and its corresponding workspace can be seen in **Figure 1.2**. Anthropomorphic manipulators are widely used in industry. As mentioned in Khalil [5], there is a strong dominance of the RRR anthropomorphic shoulder with respect to the other shoulder architectures. Some common industrial examples of anthropomorphic manipulators [5], [6], [7] are: the PUMA robot from Unimation, the IRB 120, 140, 1410, and 2400 series from ABB [8], and the KR 1000 series from KUKA robots [9] (see **Figure 1.3**).
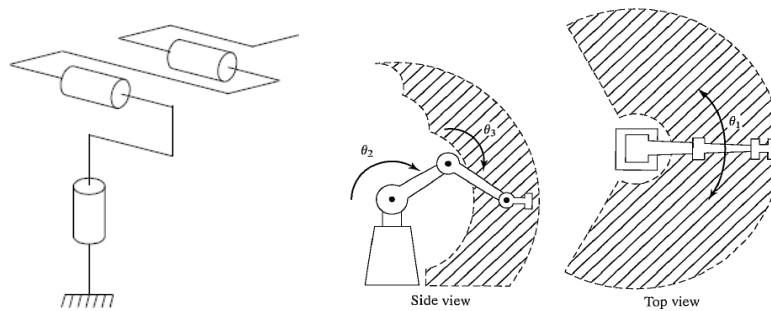
Figure 1.2 Diagram (from [5]) and schematic view (from [6]) of an anthropomorphic robot
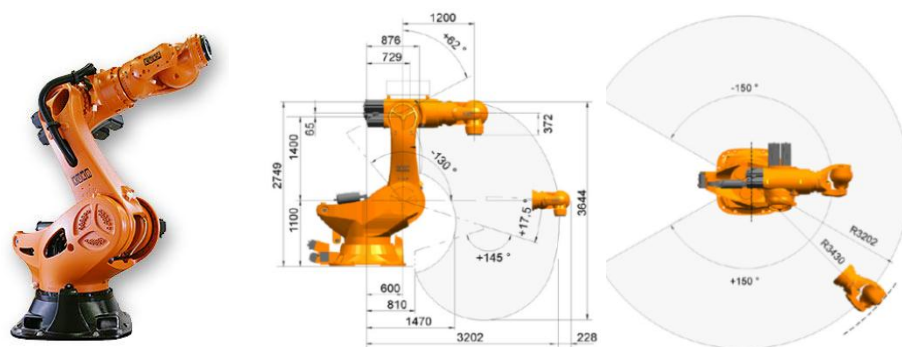
Figure 1.3 KR 1000 series KUKA anthropomorphic robot (from [9])

1.2.2　*3R* orthogonal manipulators

A serial orthogonal manipulator is composed of 3 revolute joints with mutually orthogonal joint axes [10]. A diagram and a schematic view of this kind of manipulators can be seen in **Figure 1.4**; where: d_2, d_3, d_4, r_2, r_3, θ_1, θ_2 and θ_3 are the modified Denavit-Hartenberg (mDH) parameters of the robot (introduced by Khalil and Kleinfinger [5]).
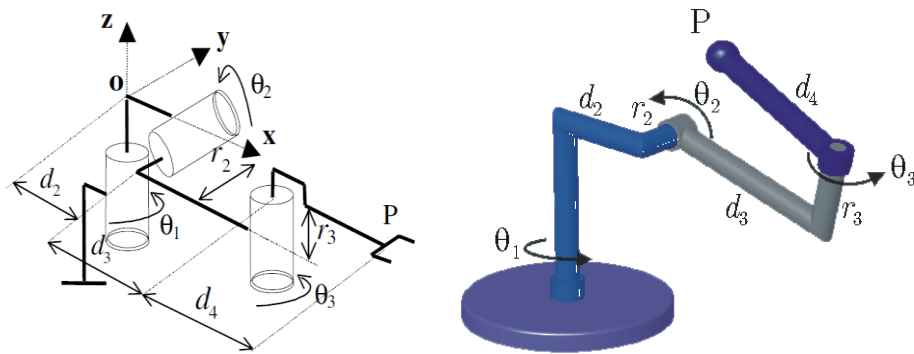
Figure 1.4 Diagram and schematic view of a 3R orthogonal manipulator (courtesy of [1], [10])

As explained in [2], *3R* orthogonal robots are not that common in industry and there are few examples of this type; for instance the IRB 6400C developed by ABB and launched in 1998, but no longer produced [10], and the six-axis isotropic manipulator called DIESTRO [7] developed at McGill University. On the other hand, this family of robots has been studied lately and researchers have found interesting geometric, kinematic and dynamic properties that have increased the attention on this kind of robots. For instance, [10] refers to a list of sources that have studied orthogonal robots with respect to: the determination of the workspace boundary; the size and shape of the workspace; the existence of holes, voids and the number of solutions to the IKM in the workspace; the feasibility of continuous trajectories; singularity curves; and cuspidality among others.

From these works it can be concluded that *3R* orthogonal manipulators present some properties that differ from those of the anthropomorphic robots described in **Section 1.2.1**. For example, orthogonal manipulators may have dissimilar global kinematic properties according to their link lengths and joint offsets unlike anthropomorphic robots [10]. Orthogonal manipulators may be cuspidal (i.e. they can change their posture without meeting a singularity), and they may have two or four solutions of the IKM in their workspace [10]. Instead, *3R* anthropomorphic manipulators are always non-cuspidal (i.e. they always meet a singularity to change their posture), and they always have four solutions to the IKM for all points in their workspace if their joints are unlimited.

Since the kinematic properties of these manipulators change according to their link and joint offset lengths, there has been an increasing interest in analyzing and classifying them. For instance, the work developed by Zein, Wenger and Chablat [10] proposes an exhaustive classification of the 3R orthogonal manipulators based on the topology of their workspace. This classification has some distinct characteristics such as setting to zero at least one of their mDH parameters, except for d_4 (otherwise it would be always singular [13]); and making a careful analysis of the equations defining the curves because these may degenerate when one or more of the parameters are equal to zero. A summary of this analysis is presented next. As mentioned, the classification proposed is based on the topology of the workspace; thus, it is important to define the following concepts [10]:

- The topology of the singular curves is defined by enumerating their cusp and node points.

- A cusp point is one where the IK admits three equal solutions. A 3-DOF manipulator can change its posture without meeting a singularity only through a cusp point.

- A node point is one where the IK admits two pairs of coincident inverse kinematic solutions; two singular curves intersect at this point.

- A void is a region with no solutions to the IKM that is inside the workspace.

- The existence of cusps and nodes can be determined from the IK polynomial.

In addition, it was also considered for the classification the feasibility of continuous trajectories in the workspace. Hence, the next definitions are also included:

- T-connected workspace: is the one in which any continuous trajectory is feasible.

- Well-connected workspace: it is t-connected and composed of only one 4-solution region.

Based on the possible combinations, there are ten possible families to analyze with at least one parameter equal to zero (as seen in **Figure 1.5** and **Table 1.1**). In each family, there may be different subfamilies. For all manipulators of one particular subfamily, the number of nodes, the number of regions with two and four solutions of the IKM, the t-connectivity and the well-connectivity of the workspace are the same. A summary of all the subfamilies found can be seen in the table 1 of [**10**]. More importantly, it is concluded that five types of manipulators (the B1, C, E, G and H types) have a well-connected workspace. The workspace of these orthogonal manipulators can be seen in **Figure 1.6.**

Family	r3	d3	r2	d2
A	0	≠	≠	0
B	0	≠	0	0
C	0	0	≠	0
D	0	≠	0	≠
E	0	0	0	≠
F	≠	≠	≠	0
G	≠	≠	0	0
H	≠	0	≠	0
I	≠	≠	0	≠
J	≠	0	0	≠

Table 1.1 Families of *3R* orth. robots
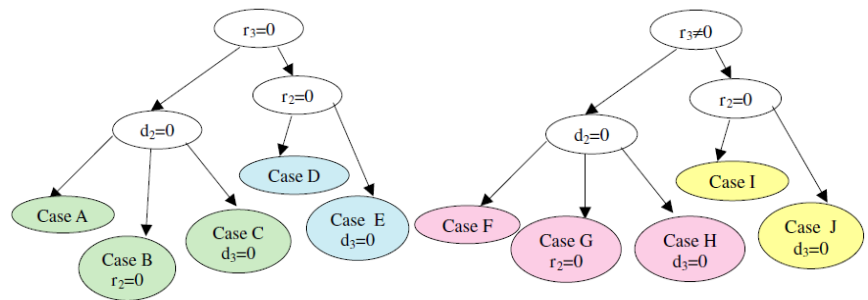
Figure 1.5 Classification of *3R* orth. robots (from [10])
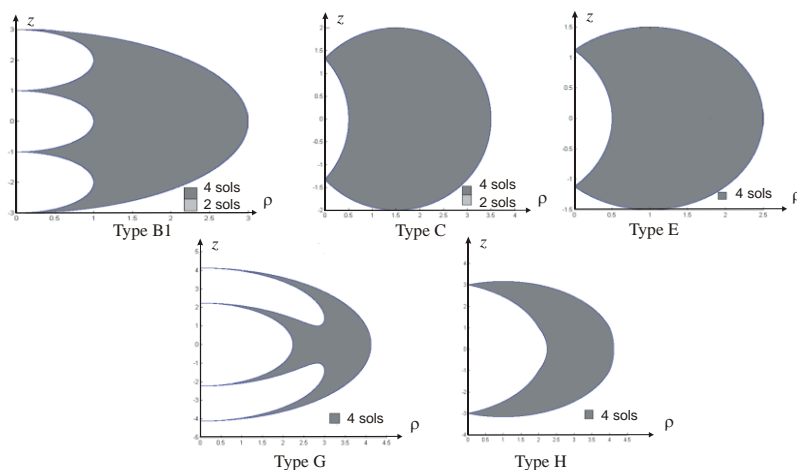
Figure 1.6 Workspace of the 5 types of *3R* orthogonal robots with well-connected workspace (from [10])

This section has presented the main characteristics of the *3R* anthropomorphic and the *3R* orthogonal robots. The next step in the comparison procedure proposed by [**1**] and [**2**] was the selection of a manipulator of each family. The approach followed is explained next.

1.3   Choosing the manipulators for comparison

A specific type of manipulator of each family had to be selected before making the dynamic analysis. This was to define the geometric and dynamic parameters of the robots. In order to do a fair comparison, these parameters had to be defined carefully. This process was implemented in two steps. First, the general geometric characteristics of the manipulators were chosen (that was to select which mDH parameters were equal to or different from zero). The second step involved defining the numerical values of the geometric and dynamic parameters. The two steps will be explained next.

1.3.1   Definition of the general geometric characteristics of the manipulators

The general characteristics of the manipulators were selected with the aid of a performance index. In this case, a kinematic performance index was selected for this purpose. Based on this index the authors were able to select one robot of each family with close kinematic properties; thus ensuring a fairer comparison. This is explained in the following lines. Firstly, it is important to recall the dexterity of a manipulator, which can be defined as the reciprocal of the condition number k of the Jacobian matrix J, as follows:

$$k = \frac{\sigma_{max}}{\sigma_{min}} \qquad k^{-1} = \frac{\sigma_{min}}{\sigma_{max}} \qquad\qquad (1)$$

where: σ_{max} and σ_{min} are the maximum and minimum singular values of J correspondingly.

Moreover, the regular dexterous workspace (RDW) can be defined as a part of the working area of the manipulator which is free of singularities and with a regular shape (for example a square) in which the dexterity is better than a minimal prescribed value λ, which is chosen by the designer. It can be obtained by first finding the largest regular shape inside the workspace of the manipulator and then finding the largest sub-regular shape where $k^{-1} > \lambda$, as seen in **Figure 1.7**. Finally, the kinematic performance index η (Majou quoted in [**1**]) can be defined. It relates the workspace compactness and dexterity as follows:

$$\eta = \frac{a_{RDW}}{\rho_{max}} \qquad\qquad (2)$$

where:

a_{RDW} is the length of the side of the RDW, and

ρ_{max} is the maximum reach of the manipulator (see **Figure 1.7**).
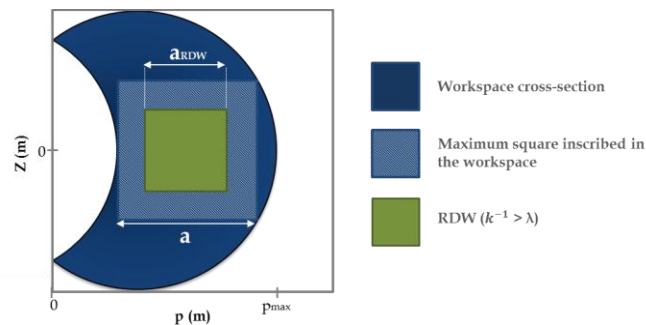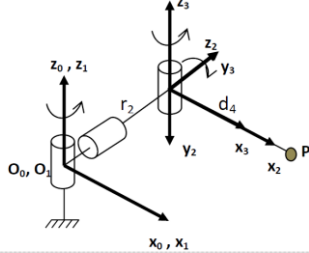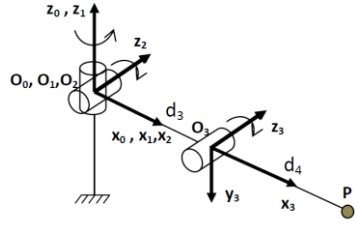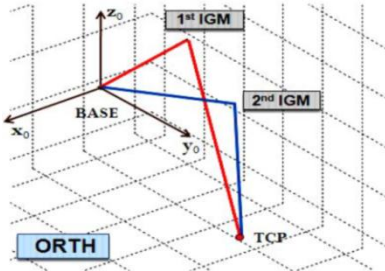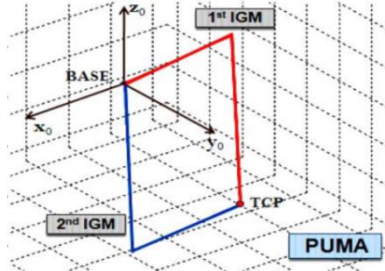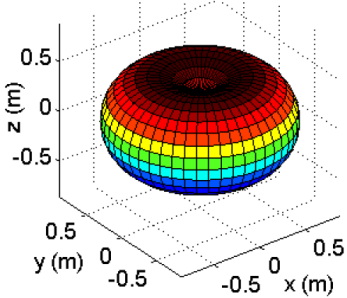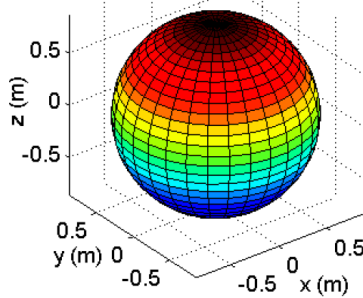
Figure 1.7 Elements for calculation of kinematic performance index η

The index η was then used to select the general geometric characteristics of the manipulators. The methodology was to make an optimization procedure and maximize this index. The robots chosen and their properties are summarized in **Table 1.2.**

Table 1.2 *3R manipulators selected for dynamic performance analysis (info. and fig. obtained from [1], [2])*

	3R orthogonal manipulator	3R anthropomorphic manipulator
Performance index η	$\eta = 0.56$	$\eta = 0.6$
mDH parameters & Manipulators	$d_2 = 0, d_3 = 0, r_3 = 0, r_2 = d_4$ *(type C)*	$d_2 = 0, r_2 = 0, r_3 = 0, d_3 = d_4$
No. solutions to IKM	4	4
Postures of IGM		
No. postures	2 groups with equivalent postures	2 groups with equivalent postures
Workspace		
Workspace type	Torus (sphere of radius d_4 rotated around z)	Sphere with radius = $2 \cdot d_4$

1.3.2   Numerical definition of the geometric and dynamic parameters

The next step in the robots definition was to select the numerical values of its parameters. First, the authors selected the values of the anthropomorphic robot similar to the parameters of a real robot (PUMA-560). Then, the parameters of the orthogonal robot were chosen in such a way that both robots had a similar RDW, mass and equivalent inertia; to assure a comparison as fair as possible. Thus, the geometric parameters were defined as follows: $r_2 = d_4 = 0.433m$ for the orthogonal robot, and $d_3 = d_4 = 0.433m$ for the anthropomorphic one. Finally, the dynamic parameters of the *jth* link of each robot: mass (M_j), first moments (MX_j, MY_j, MZ_j) and inertia tensor ($XX_j, XY_j, XZ_j, YY_j, YZ_j, ZZ_j$) were defined as shown in **Table 1.3** and **Table 1.4**. The presented approach might not be the most straightforward method; thus one of the purposes of this current thesis is to propose a more robust methodology to define the robots parameters in order to make a fair comparison (see **Chapter 4**).

j	XX_j (kgm²)	XY_j (kgm²)	XZ_j (kgm²)	YY_j (kgm²)	YZ_j (kgm²)	ZZ_j (kgm²)	MX_j (kgm)	MY_j (kgm)	MZ_j (kgm)	M_j (kg)
1	0.0166	0	0	0.0166	0	0.0166	0	0	0	4.43
2	0.6438	0	0	0.6438	0	0.0128	0	0	-2.208	10.2
3	0.012	0	0	1.133	0	1.1324	2.8051	0	0	9.8

Table 1.3 Dynamic parameters of *3R* orthogonal manipulator

j	XX$_j$ (kgm²)	XY$_j$ (kgm²)	XZ$_j$ (kgm²)	YY$_j$ (kgm²)	YZ$_j$ (kgm²)	ZZ$_j$ (kgm²)	MX$_j$ (kgm)	MY$_j$ (kgm)	MZ$_j$ (kgm)	M$_j$ (kg)
1	0.0166	0	0	0.0166	0	0.0166	0	0	0	4.43
2	0.0128	0	0	0.4845	0	0.4845	2.2083	0	0	10.2
3	0.012	0	0	1.133	0	1.1324	2.8051	0	0	9.8

Table 1.4 Dynamic parameters of *3R* anthropomorphic manipulator

1.4   Static analysis of joint torques

Before performing the dynamic analysis, a static analysis was performed. The static analysis is a special case of the dynamic analysis in which the torques are calculated using the IDM but with the joint velocities and accelerations equal to zero. Thus, the calculated torques depend only on the robot configuration. The purpose of this analysis was to obtain an insight of the manipulator performance. It assisted the researchers to find which areas of the workspace presented the highest values of torques in the static case; it also helped them to decide which of the postures should be analyzed, and how to set some of the limits for the optimization procedure done in the last part of their analysis (**Chapter 5**).

The main conclusions of this analysis can be summarized as follows. The static torque profiles for maximum and minimum values of input torques of the orthogonal robot were quite similar, while this was not the case for the anthropomorphic robot. Moreover, the torques of the second and third joints of the orthogonal robot were better distributed than for the anthropomorphic one (compare **Figure 1.8** and **Figure 1.9**). Due to these reasons, it was concluded that the *3R* orthogonal manipulator showed better performance than its anthropomorphic counterpart in static analysis.
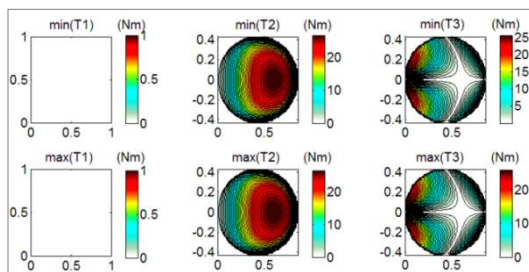
Figure 1.8 Profiles of static torque analysis for *3R* orthogonal robot (from [1])
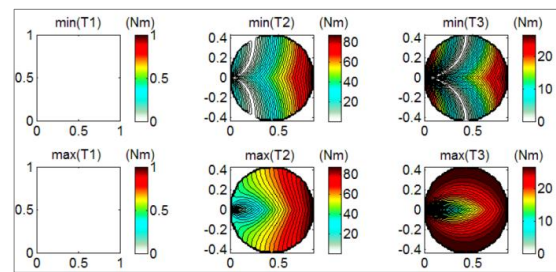
Figure 1.9 Profiles of static torque analysis for *3R* anthropomorphic robot (from [1])

Once all these considerations were studied, the authors proceeded to carry out the performance dynamic analysis which is explained in the next section.

1.5   Dynamic performance analysis of previous work

The dynamic performance of robots is still an open issue of research. A robot manipulator is a complex multi-body mechanical system and the study of its motion includes effects caused by inertial, centrifugal, Coriolis, gravity and dissipative forces, some of them nonlinear [1]. These factors make difficult to forecast and control the dynamic performances of manipulators. There exist several indices of dynamic performance in the literature (see **Section 5.1.2**). These, do not depend on the manipulator trajectory, but they may be hard to understand by engineers [1]. The easiest indices to understand are probably the maximal input torques and the joint accelerations [1], but they depend on the manipulator trajectory, making general conclusions difficult to achieve.

The research work of [1] and [2] proposes a novel method of dynamic performance analysis that uses torques and accelerations but with a new approach to take into account the dependency of these indices on the manipulator trajectory. The same approach is used in the current research work presented, thus it will be explained in detail in **Chapter 5**. For now, it is sufficient to describe a general overview of the methodology (**Section 1.5.1**) and to present the results and conclusions obtained by the researchers (**Section 1.5.2**).

1.5.1   General overview of the method

The methodology proposed a systematic procedure for analyzing the dynamic performances of manipulators (hence, allowing researchers to easily compare two robots). Since the joint torques and accelerations depend on the trajectory it was proposed to move the manipulators on exciting trajectories to make them experience the highest values of the input torques. The trajectories, which are constructed from a set of points joined by a function (polynomial, spline) via two different methods (*point-to-point* or *via-points*), were found with the help of an optimization procedure which had the objective of maximizing the input torques, subject to some constraints such as the maximum joint torques values, and that the trajectory of the end-effector stayed inside the workspace. The use of exciting trajectories is not new in robotics, for instance, they are used for dynamic parameter identification [1], but this concept has not been used for the purposes of dynamic performance analysis; thus, the novelty of this methodology.

1.5.2   Results and conclusions from the dynamic performance analysis

The most representative results of this analysis can be summarized in two test cases in which the authors applied the methodology using different functions and the two methods to construct the trajectory. Moreover, each of the test cases first focused on one of the manipulators to obtain the worst possible case scenario. It is important to add that the selected posture of the two manipulators was the one corresponding to elbow-up because they were similar in both cases (which correspond to the first IGM solution on both robots, see postures in **Table 1.2**).

- In the **first test case**, the optimization procedure was done using a cubic spline function and the *via-point* method to define the trajectory. It was first applied to the orthogonal manipulator to analyze its performance in the worst possible cases (reach constraint boundaries of maximum torques or maximum speed). The trajectory obtained from the optimization was then applied to the anthropomorphic robot. The optimized trajectory can be seen in **Figure 1.10** and the corresponding torques and joint accelerations can be observed in **Figure 1.12**

- In the **second test case** the optimization procedure was done using a 5th degree polynomial and the *point-to-point* method to define the trajectory. It was first applied to the anthropomorphic manipulator to analyze its performance in the worst possible cases. Similarly to the previous test, the trajectory obtained was applied to the orthogonal robot. The optimized trajectory is shown in **Figure 1.11** and the corresponding torques and joint accelerations can be seen in and **Figure 1.13** correspondingly.
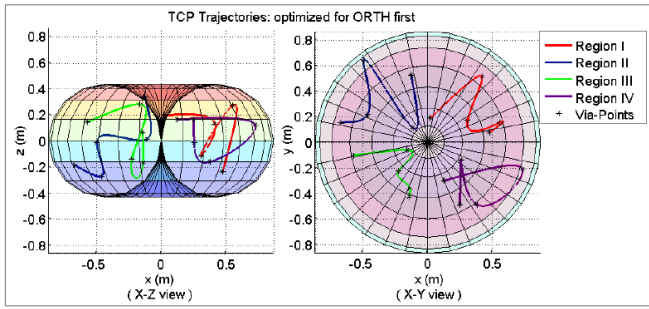
Figure 1.10 Optimized trajectory for test case 1
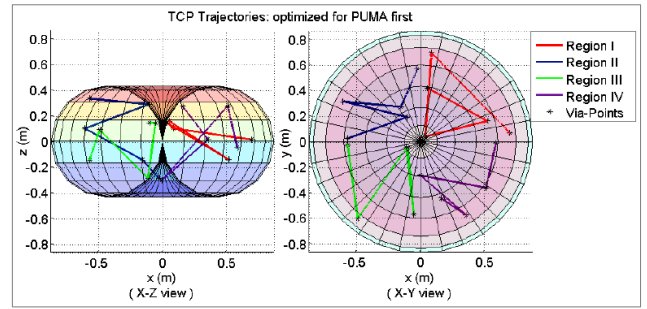
Figure 1.11 Optimized trajectory for test case 2
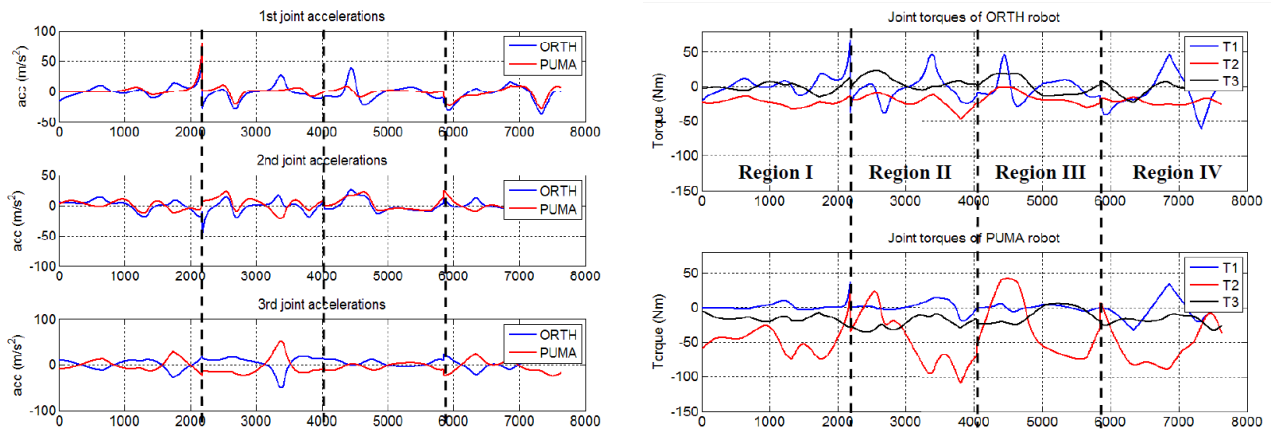
Figure 1.12 Joint accelerations and torques for test case 1 (courtesy of [1])
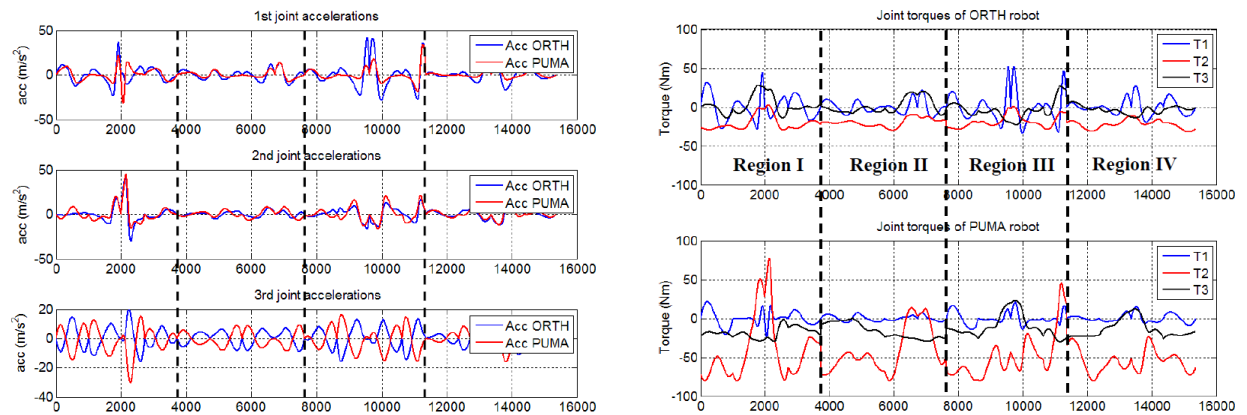
Figure 1.13 Joint accelerations and torques for test case 2 (courtesy of [1])

Analysis of results

From these results it was remarked that in term of joint torques, the orthogonal manipulator had better performance than the anthropomorphic robot. The joint torques of the anthropomorphic robot were in general larger than the ones of the orthogonal robot for both test cases. Regarding the joint accelerations, the authors concluded that both manipulators had fair performance. Moreover, a deeper analysis was done by taking advantage of the EWA (discussed in **Section 5.1.2**). The idea was to obtain the joint torques for the same trajectory but now changing the payload and speed of the end-effector. Then, the differences between the maximum torques were computed. The results for both methods can be seen in **Figure 1.14.**
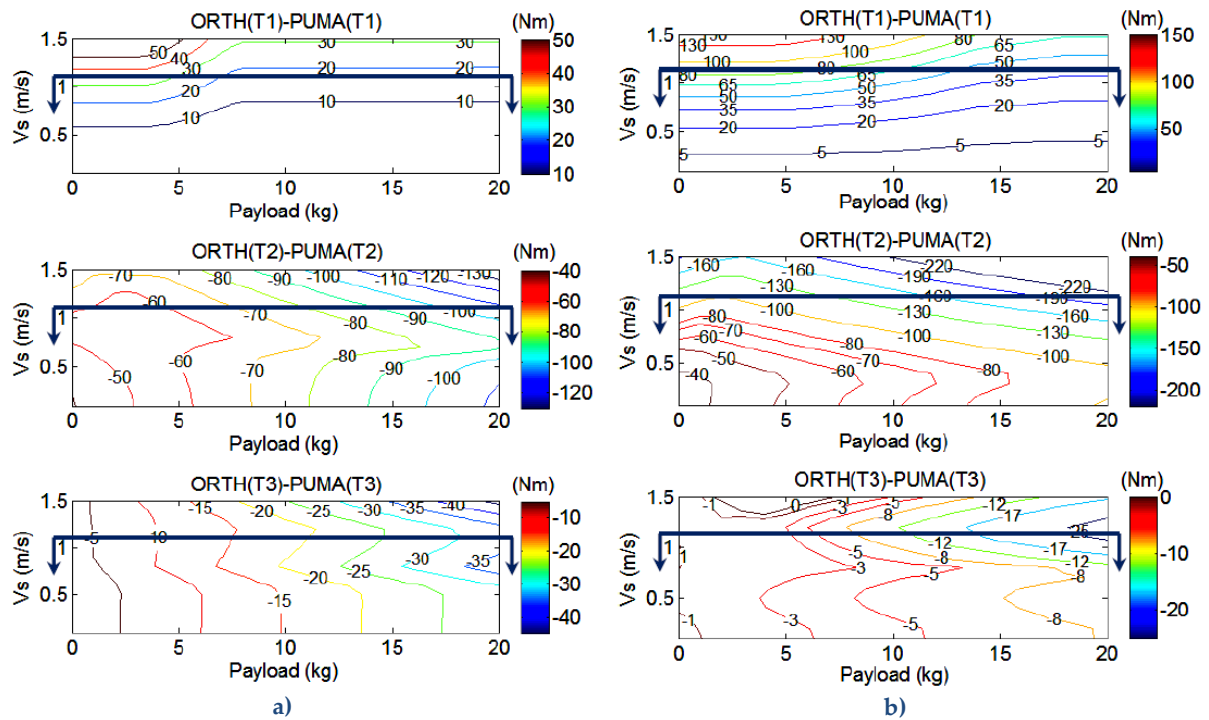
Figure 1.14 Difference of maximum absolute torques between the orthogonal and anthropomorphic manipulators, (a) for test case 1, and (b) for test case 2

In addition, the analysis was limited to the area below the velocity of the end-effector with value 1 m/s because in practice this velocity is sufficient. Since these plots were calculated as the difference between the torques of orthogonal and the torques of the anthropomorphic robot, then the negative values meant that the orthogonal manipulator performed better than its counterpart. Consequently, it was concluded that the performance of joint 2 and 3 was better in the orthogonal case. In the case of joint 1 the anthropomorphic manipulator presented better values but the difference between both robots for this joint was relatively small compared to the differences encountered in the other two joints. As a main conclusion, this previous research determined that:

The dynamic performances of the *3R* orthogonal manipulator seemed better than those of the *3R* anthropomorphic manipulator.

Chapter 2

Manipulators and methodology

This chapter has the objective of presenting the manipulators under study in this thesis, and to summarize the methodology to follow. Thus, **Section 2.1** discusses the general characteristics of the families of the two manipulators. Subsequently, **Section 2.2** presents a description of the specific manipulator of each family that was analyzed. Finally, **Section 2.3** describes the methodology applied.

2.1   *RPaPa* anthropomorphic and *RPaR* orthogonal architectures

In this section the robots based on *RPaPa* anthropomorphic and *RPaR* orthogonal architectures are introduced. Some important definitions and the description of the robots are discussed next.

2.1.1   Important definitions

The families of robots under study have parallelogram joints, and they are both capable of performing Schoenflies motions. Thus, it is important to define both terms briefly.

Parallelogram joints (Pa)

As described in [**5**] a joint connects two successive links of a robot limiting the number of DOF between them (e.g. revolute and prismatic 1-DOF joints). The concept of parallelogram joints also known as Pa or Π joints [**11**] is commonly used in robotics. As explained by Angeles [**12**], a parallelogram is a four-bar linkage with its opposite links of the same length, (see **Figure 2.1**). It is composed of: (1) one fixed, (2) one input, (3) one coupler, and (4) one output link. In a Pa mechanism, an opposite link moves with a relative pure translation with respect to its counterpart, i.e. each point of one link describes a circular trajectory onto the other link. Thus, a Pa joint couples two links, while allowing a relative translation along a circular trajectory. Pa linkages are neither higher nor lower kinematic pairs [**12**] but they can be treated as joints because, these mechanisms appear in many industrial robots, they are well documented in the technical literature, and they have been studied systematically as valuable means in parallel-robot design [**11**]. Pa joints, when combined with other kinematic pairs, can generate translations and Schoenflies motions [**12**] (see next).
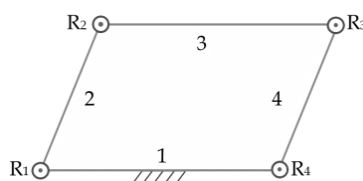
Figure 2.1 Parallelogram joint Pa (based on [12])

Schoenflies-motion 4-DOF robots

Schoenflies motions[1], denoted here as 3T1R motions (3 translations, 1 rotation) or SCARA motions [13], consist of three linearly independent translations and one rotation around an axis of fixed direction [11]. Therefore, a robot performing this kind of motions must have 4 DOF. Many different robot architectures are able to generate Schoenflies motions [13]. For instance, a complete minimum set of serial topologies capable of producing Schoenflies motions based on the combination of revolute (R), prismatic (P), helical (H) and Pa (Π) joints is proposed in [11]. In this study it is also concluded that the minimum number of Schoenflies-motion generators with R and P pairs is 3, with R, P and H pairs is 17, and with R, P, H and Π pairs is 79. Schoenflies-motions robots are usually used for pick-and-place tasks and palletizing operations. The SCARA robot (see **Figure 2.2**) and the Delta robot [13] (i.e. ABB FlexPicker) are examples of robots capable of performing Schoenflies motions.
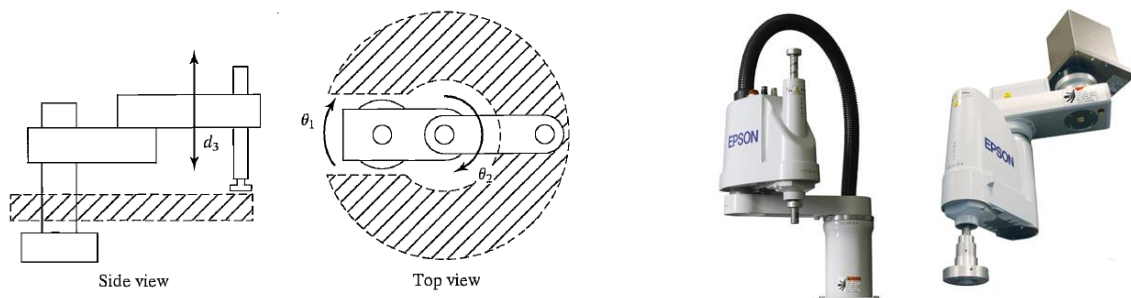
Figure 2.2 Schematic view of SCARA robot [6], and EPSON LS and RS SCARA robots [14]

2.1.2 *RPaPa* anthropomorphic architecture

Serial robot manipulators with *RPaPa* architecture consist of 3 joints in serial configuration in which the first joint is revolute and the last two are *Pa*. Moreover, it is common that this type or robots have a 1-DOF revolute wrist attached to the *RPaPa* chain (thus, *RPaPa-R* kinematic chain). This combination of joints produces Schoenflies motions as studied in [11] and discussed previously. A schematic view of an *RPaPa-R* robot can be seen in **Figure 2.3a.** This kind of robots can be classified along with the *3R* manipulators, studied in **Section 1.2.1**, as of anthropomorphic type because the last two joints of the shoulder have parallel axes. Some examples of these robots are the KR 100/180 (see **Figure 2.3b**) of KUKA robotics [9] and the IRB 660 of ABB [8].
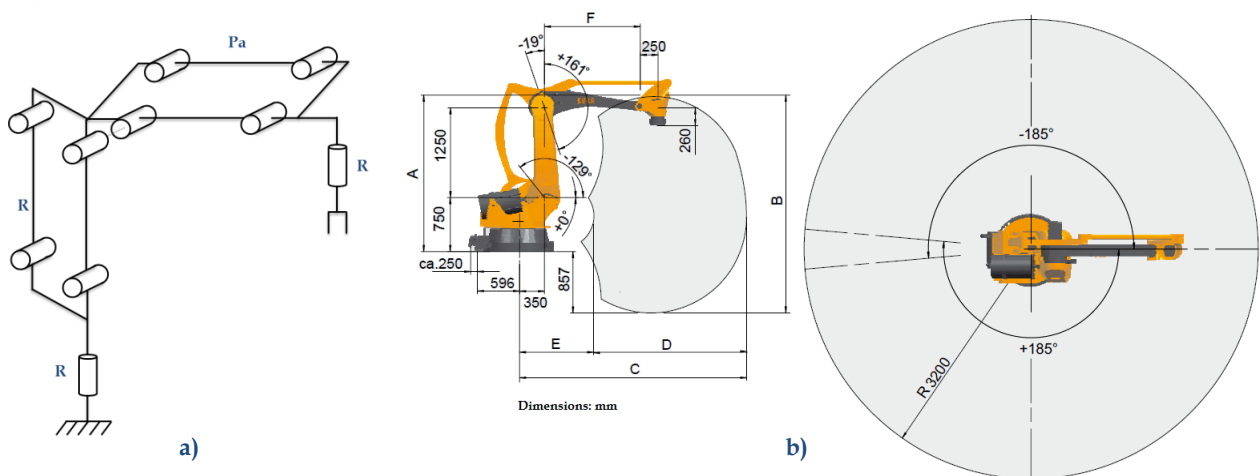
Figure 2.3 (a) Kinematic structure of an *RPaPa-R* manipulator, (b) KUKA KR 100/180 PA robot [9]

---

[1] They are named after the German mathematician Arthur Moritz Schoenflies (1853-1928) who first studied them [5].

2.1.3 *RPaR* orthogonal architecture

RPaR orthogonal robots are 3-joint robots in which the first joint and third joints are of revolute type and the second one is a parallelogram [15]. These robots are of orthogonal type (analogous to the *3R* orthogonal robots studied in **Section 1.2.2**) because any two consecutive joint axes are perpendicular to each other. Correspondingly to the previous case, if a 1-DOF revolute wrist is attached to the *RPaR* chain (thus, *RPaR-R* kinematic chain), the combination of kinematic pairs will produce Schoenflies motions as studied in [11] and discussed previously. The kinematic structure of a manipulator of this type can be observed in **Figure 2.3a.**

Similarly to the case of *3R* orthogonal robots, *RPaR* orthogonal manipulators are uncommon in industrial applications. One of the few examples found of this kind of architecture are some products of the range of industrial manipulators produced by the Italian company DALMEC, namely the manipulators series PM, PS and Maxipartner [16] (**Figure 2.4b** shows the Maxipartner MXS as an example of this family). In fact, these mechanical systems are pneumatic manipulators designed as lifting assistance tools for workers, with applications found from food to car industry.

Figure 2.4 (a) Kinematic structure of an RPaR-R orthogonal manipulator
(b) RPaR orthogonal DALMEC Maxipartner MXS manipulator (from [16])

This family of robots has been recently studied in [15]. In this research work, geometric and kinematic analyses were performed which included the development of the geometric and kinematic models, the analysis of the Jacobian matrix and the singularities, and the analysis of the workspace characteristics. Moreover, they were compared against the *3R* orthogonal manipulators in the mentioned characteristics and in terms of kinematic performance. Finally, an optimization procedure based on kinematic indices and a genetic algorithm was proposed (for more details see [15]).

2.2   Specificities of the manipulators under study

In order to perform the dynamic performance analysis, subject of this research work, it is necessary to define two specific manipulators of each family. As it is explained in further chapters, the geometric and dynamic parameters of both robots are selected with the help of some methodologies. Nevertheless, at this moment it is important to present the general kinematic structure of each manipulator since all of the other features of the manipulators and the methodologies applied depend on them. The selection of the kinematic structure of the manipulators is based on two factors: (1) the results and conclusions of the previous investigations studied during the bibliographical report and in

the previous sections of this research work, and (2) commercial examples of each kind of family of manipulators. With this information, a general structure of each robot is defined as shown next.

2.2.1 *RPaPa* anthropomorphic manipulator

The *RPaPa* anthropomorphic manipulator kinematic structure is defined as seen in **Figure 2.5**. It is composed of nine joints and nine moving links. Three of the joints (1, 2 and 4) are active, and the rest are passive. Following the notation proposed by Khalil [**5**], this robot can be classified as a serial closed-loop structure manipulator with two independent closed loops. The mDH parameters (shown in **Table 2.1**) are calculated by opening the closed loops in joints 8 and 9 (thus, modeling it as a tree-structure robot). As it can be seen, it has a similar structure as the Kuka KR 100/180 Pa robot of **Figure 2.3**. Some characteristics to note are that the rotational axes of joints 3 and 4 are collinear, and that the base of each parallelogram has a constant rotation angle of $\pi - \beta_1$ with respect to the $X_0\ axis$ and $\pi - \beta_1 - \beta_3$ with respect to the $X_3\ axis$ correspondingly. In the remaining of this report, the *RPaPa* anthropomorphic manipulator may be referred to by the short form *RPaPa* ANTHRO.
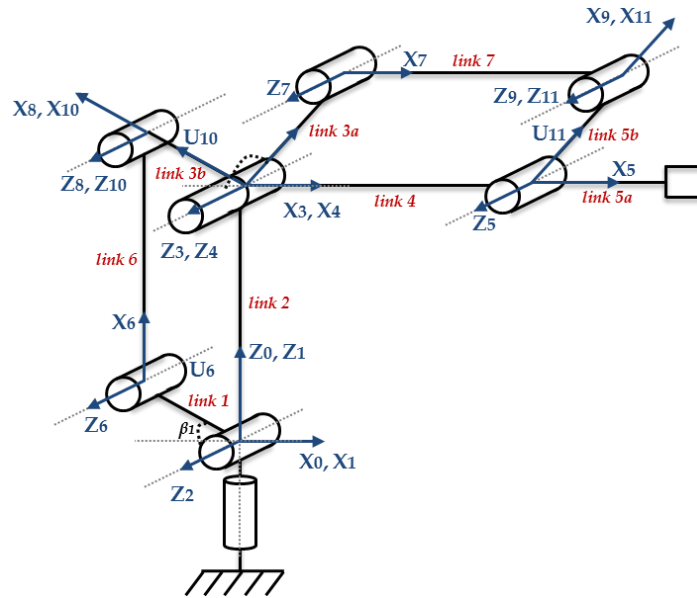
Figure 2.5 Kinematic structure of the *RPaPa* ANTHRO manipulator

j	a(j)	μ_j	σ_j	γ_j	b_j	α_j	d_j	θ_j	r_j
1	0	1	0	0	0	0	0	θ_1	0
2	1	1	0	0	0	$\pi/2$	0	θ_2	0
3	2	0	0	0	0	0	d_3	θ_3	0
4	3	1	0	0	0	0	0	θ_4	0
5	4	0	0	0	0	0	d_5	θ_5	0
6	1	0	0	γ_6	b_6	$-\pi/2$	d_6	θ_6	0
7	3	0	0	γ_7	0	0	d_7	θ_7	0
8	6	0	0	0	0	0	d_8	θ_8	0
9	7	0	0	0	0	0	d_9	θ_9	0
10	3	0	2	γ_{10}	0	0	d_{10}	0	0
11	5	0	2	γ_{11}	0	0	d_{11}	0	0

Table 2.1 mDH parameters of the *RPaPa* ANTHRO manipulator

where: $\gamma_6 = \pi$, $\gamma_7 = \pi - \beta_1 - \beta_3$, $\gamma_{10} = \pi - \beta_1$, and $\gamma_{11} = \pi - \beta_1 - \beta_3$

The active joint variables, denoted by q_a, the passive variables, denoted by q_p, and the cut variables, denoted by q_c, of this manipulator can be expressed as:

$$q_a = \begin{bmatrix} q_1 \\ q_2 \\ q_4 \end{bmatrix} \quad q_p = \begin{bmatrix} q_3 \\ q_5 \\ q_6 \\ q_7 \end{bmatrix} \quad q_c = \begin{bmatrix} q_8 \\ q_9 \end{bmatrix} \tag{3}$$

The direct geometric model (DGM) using homogenous transformation matrices [5] can be expressed as follows:

$$^0T_5 = \ ^0T_1(q_1) \cdot \ ^1T_2(q_2) \cdot \ ^2T_3(q_3) \cdot \ ^3T_4(q_4) \cdot \ ^4T_5(q_5) \tag{4}$$

The passive variables of eq. (4) (q_3 and q_5) must be expressed in terms of the active variables. The passive variables can be found with the help of the geometric and kinematic constraint equations, which are the equations that define the closure of the two loops [5]. The geometric constraint equations of this manipulator can be written as:

- For loop 1: $^1T_2(q_2) \cdot \ ^2T_3(q_3) \cdot \ ^3T_{10} = \ ^1T_6(q_6) \cdot \ ^6T_8(q_8)$ (5)
- For loop 2: $^3T_4(q_4) \cdot \ ^4T_5(q_5) \cdot \ ^5T_{11} = \ ^3T_7(q_7) \cdot \ ^7T_9(q_9)$ (6)

Furthermore, the kinematic constraint equations can be written as shown next:

- For loop 1: $^1J_{10} \cdot \dot{q}_{b1,1} - \ ^1J_8 \cdot \dot{q}_{b2,1} = 0$ (7)
- For loop 2: $^3J_{11} \cdot \dot{q}_{b1,2} - \ ^3J_9 \cdot \dot{q}_{b2,2} = 0$ (8)

where:

$\dot{q}_{b1,1} = [\dot{q}_2 \quad \dot{q}_3]^T, \ \dot{q}_{b2,1} = [\dot{q}_6 \quad \dot{q}_8]^T$, and
$\dot{q}_{b1,2} = [\dot{q}_4 \quad \dot{q}_5]^T, \ \dot{q}_{b2,2} = [\dot{q}_7 \quad \dot{q}_9]^T$.

The active variables are q_2 and q_4 for the first and second closed loops correspondingly. Since the closed loops are relatively simple (both of them are parallelograms), the passive variables can be found in terms of the active variables using a geometric method [5], without using the constraint equations. Thus, from **Figure 2.5** it can be found that:

- For loop 1: $q_3 = -q_2,$ $q_6 = q_2 - \pi,$ $q_8 = -q_2 + \pi - \beta_1$ (9)
- For loop 2: $q_5 = -q_4,$ $q_7 = q_4 - (\pi - \beta_1 - \beta_3),$ $q_9 = -q_4 + (\pi - \beta_1 - \beta_3)$ (10)

2.2.2 *RPaR* **orthogonal manipulator**

Similarly, the *RPaR* orthogonal manipulator kinematic structure is defined as shown in **Figure 2.6**. It is composed of six joints and six moving links (link 3 has longitude of zero). Three of the joints (1, 2 and 4) are active, and the rest are passive. This robot can also be classified as a serial closed-loop structure manipulator with one independent closed loop. The mDH parameters (shown in **Table 2.2**) are calculated by opening the closed loop in joint 6. As it can be seen it has a similar structure as the Dalmec manipulator of **Figure 2.4b**. Being consistent with the notation used in the previous robot, it can be noted that the parallelogram base has a constant rotation angle with respect to the X_0 *axis* of $\pi - \beta_1$ (90 degrees in this case). In the remaining of this report, the *RPaR* orthogonal manipulator may be referred to by the short form *RPaR* ORTH.
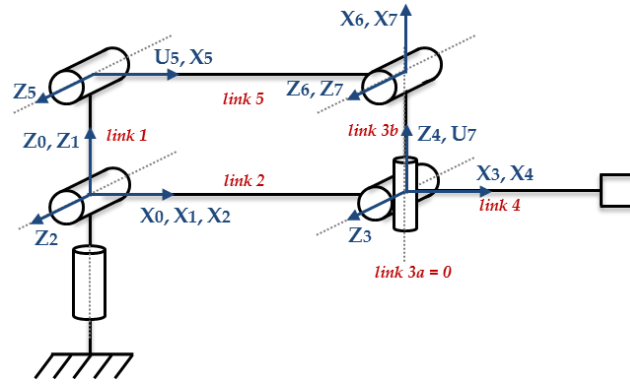
Figure 2.6 Kinematic structure of the *RPaR* ORTH manipulator

j	a(j)	μ_j	σ_j	γ_j	B_j	α_j	d_j	θ_j	r_j
1	0	1	0	0	0	0	0	θ_1	0
2	1	1	0	0	0	$\pi/2$	0	$\theta 2$	0
3	2	0	0	0	0	0	d_3	θ_3	0
4	3	1	0	0	0	$-\pi/2$	0	θ_4	0
5	1	0	0	0	b5	$\pi/2$	0	θ_5	0
6	5	0	0	0	0	0	d_6	θ_6	0
7	3	0	2	γ_7	0	0	d_7	0	0

Table 2.2 mDH parameters of the *RPaR* ORTH manipulator

where: $\gamma_7 = \pi - \beta_1$.

The active joint variables, denoted by q_a, the passive variables, denoted by q_p, and the cut variables, denoted by q_c, of this manipulator are the following ones:

$$q_a = \begin{bmatrix} q_1 \\ q_2 \\ q_4 \end{bmatrix} \quad q_p = \begin{bmatrix} q_3 \\ q_5 \end{bmatrix} \quad q_c = [q_6] \tag{11}$$

Likewise to the previous manipulator, the DGM can be expressed as follows:

$$^0T_4 = {}^0T_1(q_1) \cdot {}^1T_2(q_2) \cdot {}^2T_3(q_3) \cdot {}^3T_4(q_4) \tag{12}$$

The passive variable of eq. (**14**) is q_3 which must be found in terms of the active variables. The geometric constraint equations for this manipulator can be written as follows:

- For loop 1: $^1T_2(q_2) \cdot {}^2T_3(q_3) \cdot {}^3T_7 = {}^1T_5(q_5) \cdot {}^5T_6(q_6)$ (13)

Similarly, the kinematic constraint equations are defined as shown next:

- For loop 1: $^1J_7 \cdot \dot{q}_{b1} - {}^1J_6 \cdot \dot{q}_{b2} = 0$ (14)

where:

$\dot{q}_{b1} = [\dot{q}_2 \quad \dot{q}_3]^T, \quad \dot{q}_{b2} = [\dot{q}_5 \quad \dot{q}_6]^T.$

The active variable of the closed loop is q_2. From **Figure 2.6** it can be found that:

- For loop 1: $q_3 = -q_2, \quad q_5 = q_2, \quad q_6 = -q_2 + \beta_1$ (15)

Once that the two manipulators under study have been introduced, it is important to discuss the methodology to follow in the development of this thesis to accomplish the desired objectives.

2.3   General overview of the methodology

The main goals of this research work are to make a dynamic performance analysis and comparison between the *RPaR* ORTH and the *RPaPa* ANTHRO manipulators, and moreover, to propose a systematic approach to evaluate dynamic performances of manipulators. On the other hand, as described in **Chapter 1**, in the previous research work [**1**], [**2**] a dynamic performance analysis of the *3R* orthogonal and *3R* anthropomorphic manipulators was performed. The relevance of this last work to the current research is that (a) it also provided a systematic approach to perform a dynamic performance analysis of manipulators and (b) the results obtained were also used in a comparison process between a pair of manipulators of the orthogonal and anthropomorphic families. Therefore, it was decided that the research work performed in this thesis would follow some of the approaches and methods used previously. Following all these considerations, the methodology to follow in the current research work can be summarized in the diagram of **Figure 2.7**
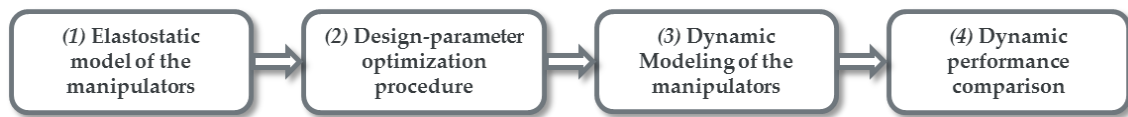
Figure 2.7 Methodology followed throughout this research work

The flow diagram can be justified as follows. The dynamic performance analysis and comparison **(4)**, which is the main objective of this thesis, is implemented following the method proposed by the previous research. The details of this method, its application and conclusions are discussed in **Chapter 5**. Moreover, the dynamic model of the manipulators **(3)** has to be done in order to use it for the comparison. The dynamic modeling is discussed also in **Chapter 5**.

Furthermore, since a comparison between two different families of manipulators is done, it is necessary to assure that it is performed as fairly as possible. This can be accomplished, for instance, by selecting the design parameters of the robots (such as the mass of the links) via an optimization procedure. A technology-oriented optimization procedure **(2)** such as the one presented in [**3**] turns out especially appropriate for this objective. Therefore, the details of this procedure and its application to the manipulators under study are presented in **Chapter 4**.

Finally, in order to apply the mentioned optimization procedure, some design objectives and constrains must be established. As explained before, the stiffness of the manipulator is selected as index. To calculate the stiffness of the manipulator, the elastostatic model of the robots **(1)** has to be done. Therefore, **Chapter 3** has the objective to present the elastostatic model, its application and its validation.

This chapter introduced the manipulators under study, some important definitions and the methodology to follow in the development of this thesis. Once all these important elements were defined, the next chapters will present the work, results and conclusions of this research work.

Chapter 3

Elastostatic Modeling

This chapter presents the main concepts of elastostatic modeling and its application to the manipulators under study. First, **Section 3.1** introduces some general concepts. Then the elastostatic modeling methodology used is described in **Section 3.2**. Finally the modeling of the robots under study is developed in **Section 3.3**.

3.1. General Concepts

In many mechanical systems constituent elements are considered as rigid bodies, meaning that these will not deform under a load. This assumption is valid when deformations are not important for the model; otherwise, the bodies must be studied as elastic objects [17]. One way to model the elasticity of a body is to consider it as linear spring. Thus, it will follow the Hook's law which relates the displacement of a spring (Δl) with the applied external force (F) as follows: $F = K\Delta l$. Where K is the spring proportionality constant also known as stiffness [17]. The extension of this model to three dimensional bodies or mechanisms is called the stiffness model. As defined in [18], the stiffness model describes the resistance of an elastic body or mechanism to deformations caused by external forces/torques. Consequently, elastostatics can be defined as the study of a mechanism under static loads and the deformations caused by these, within the linearly elastic range of the system [7]. For relatively small deformations, the linear relationship between the 3-dimensional translational/rotational displacements and the 3-dimensional translational/rotational static forces/torques causing them is described by the stiffness matrix $\boldsymbol{K}$ as follows:

$$\begin{bmatrix} \boldsymbol{F} \\ \boldsymbol{M} \end{bmatrix} = K \begin{bmatrix} \boldsymbol{t} \\ \boldsymbol{\varphi} \end{bmatrix} \tag{16}$$

where: $\boldsymbol{t} = [\, t_x \quad t_y \quad t_z \,]^T$ is the vector of translational displacements,

$\boldsymbol{\varphi} = [\, \varphi_x \quad \varphi_y \quad \varphi_z \,]^T$ is the vector of rotational displacements,

$\boldsymbol{F} = [\, F_x \quad F_y \quad F_z \,]^T$ is the vector of static forces, and

$\boldsymbol{M} = [M_x \quad M_y \quad M_z]^T$ is the vector of rotational torques.

From mechanics it is known that $\boldsymbol{K}$ is a 6x6 symmetrical semi-definite non-negative matrix, which may include non-diagonal elements to represent the coupling between the translations and rotations [18]; further, as noted in [4] this matrix may be not-symmetrical when the structure is under static load. The inverse of $\boldsymbol{K}$ is called the compliance matrix [17] and is denoted as $\boldsymbol{k}$. Therefore, the elastostatic problem entails finding K. The common methods used to compute the stiffness matrix of a mechanical system are briefly described next, followed by the reasons that lead to the selections of a different approach.

Finite element analysis (FEA)

As stated in [**19**], in FEA each link is modeled as an assembly of a finite number of elements and each element is a continuous member of the link. Furthermore, boundary conditions, changes in geometry and physical properties can be taken easily into account. FEA is considered to have the best reliability and accuracy among the different methods because the links and joints are modeled with their actual dimension and shapes [**4**]. Greater model accuracy can be accomplished by using more elements, modeling with elements of higher order or both [**19**], (of course this increases the complexity of the model). In general, because of its high computational costs, FEA is usually applied at the final design stage for the verification and component dimensioning or for validation and comparative study of other stiffness analysis methods [**4**].

Matrix Structural Analysis (MSA)

Similarly to FEA, the MSA methods are based on the concept of replacing the continuous structure of the body by a mathematical model composed of structural elements of finite size (expressed in matrix form). It incorporates the main ideas of FEA but works with larger flexible elements such as beams, arcs, and cables. This reduces the computational costs [**4**], but also its accuracy; however, it provides a reasonable trade-off between accuracy and computational time [**4**]).

Virtual Joint Method (VJM)

This method, also known as lumped modeling [**4**], rigid finite element models or finite segment method [**19**], discretizes the manipulators as a set of rigid bodies (the links) interconnected by linear springs which introduce the flexible characteristics to the model [**19**]. In other words, it is based on the expansion of the traditional rigid model by adding virtual joints (localized springs) that describe the elastic deformations of the manipulator links, joints and/or actuators [**4**]. Because of these reasons, the VJM is considered to be the simplest one to implement and provides acceptable accuracy in short computational time [**4**]. However, it is very hypothetical and since its standard formulation works with models made of 1-DOF springs, it does not consider the coupling between rotational and translational deflections [**4**] and does not work well with closed-loop mechanisms [**19**].

Why are these methods not selected?

As one of the objectives of this thesis is to perform an optimization procedure based on stiffness analysis in order to find the mass of the studied manipulators, one method must be selected for performing this analysis. None of the three methods explained previously are selected due to the following reasoning: FEA presents high model accuracy but its computational cost does not make it attractive for an optimization process, MSA involves high-dimensional matrix operations which complicates its analytical modeling [**4**], and VJM will not provide the desired accuracy due to the simplifications assumed by the model.

Because of these reasons, a methodology developed by [**4**] is selected for performing the elastostatic modeling. It is targeted to design optimization, it provides high accuracy, almost as FEA, but with lower computational cost, and it is useful for overconstrained mechanisms like parallel robots; thus, making it a perfect match for the purposes of this thesis. This method is discussed in detail the next section.

3.2. 6-DOF Virtual Joint Method Stiffness analysis

The stiffness analysis proposed by Pashkevich, Chablat and Wenger [4] is based on a multidimensional VJM model that replaces the link flexibility by localized 6-DOF virtual springs (instead of a 1-DOF spring as in the traditional VJM) which describe the translational/rotational compliance and the coupling between them. Besides, the spring stiffness parameters are evaluated using FEA to ensure higher accuracy. The method proposed copes with the problem that arises in standard VJM methods in which it is a difficult task to determine the proportionality constants of the springs (which is the reason why the use of standard VJM methods is limited [19]). This is done by employing a new solution strategy of the kinetostatic equations for the unloaded manipulator configuration (in the case of parallel robots it considers simultaneously the kinematic and static relations for each separate kinematic chain and then aggregates the partial solutions in a total one), thus allowing to compute the stiffness matrix for the overconstrained architectures, including singular manipulator postures [4]. Some of the advantages of the proposed methodology are that it is applicable to overconstrained mechanisms and it gives almost the same accuracy as FEA but with lower computational cost. In order to explain this method, let us consider the general structure of a single serial kinematic chain of a parallel manipulator as shown in the left column of **Table 3.1**.

Table 3.1 Six-DOF Virtual Joint Method modeling (based on [3])	
Single serial kinematic chain	6-DOF VJM Flexible model
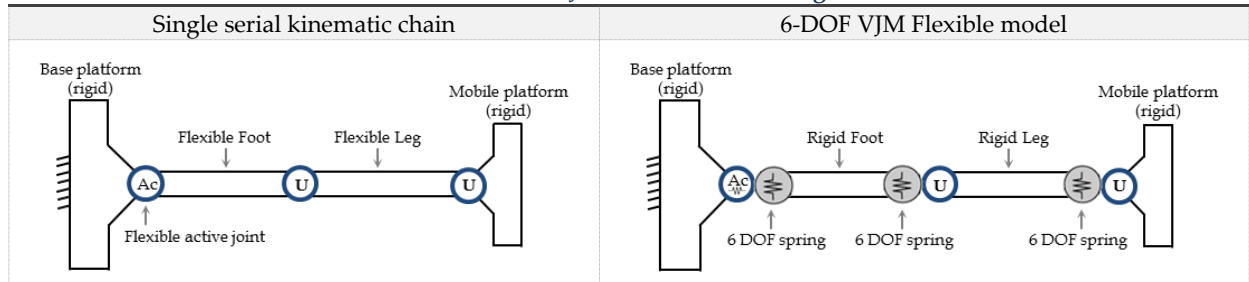
As it can be observed, the kinematic chain is composed of a base and a mobile platforms, which will be consider rigid. These are joined by: an active joint considered as flexible, two links (foot and leg) considered as flexible as well and two rigid universal joints. The elasticity of the non-rigid parts is modeled by replacing them by rigid elements and adding virtual springs that represent their flexibility as follows: for the actuator: a 1-DOF virtual spring for describing the actuator control loop compliance and a 6-DOF virtual spring describing the actuator mechanics compliance; and for the flexible links: 6-DOF virtual spring describing their mechanical compliance. Thus, the new structure of the kinematic chain can be represented as in the right column of **Table 3.1**.

Therefore, the DGM describing the position of the end-effector of this kinematic chain is described by eq.(17) (note: the subscript *i* corresponding to the *ith* kinematic chain has been dropped).

$$\boldsymbol{T} = \boldsymbol{T_B} \cdot \boldsymbol{V_a}(q_0 + \theta_0) \cdot \boldsymbol{V_s}(\theta_{1\dots6}) \cdot \boldsymbol{T_{foot}} \cdot \boldsymbol{V_s}(\theta_{7\dots12}) \cdot \boldsymbol{V_{u1}}(q_1, q_2) \cdot \boldsymbol{T_{leg}} \cdot \boldsymbol{V_s}(\theta_{13\dots18}) \cdot \boldsymbol{V_{u2}}(q_3, q_4) \cdot \boldsymbol{T_E} \qquad (17)$$

where:

(a) $\boldsymbol{T_B}$ is the transformation matrix (TM) that describes the rigid link between the base and the joint.

(b) $\boldsymbol{V_a}(q_0 + \theta_0)$ is the homogeneous matrix that includes the actuated coordinate (q_0), and the virtual spring coordinate (θ_0) which models the control loop stiffness.

(c) $\boldsymbol{V_s}(\theta_{1\dots6})$ is the TM that defines the 6-DOF virtual spring describing the actuator mechanical stiff.

(d) $\boldsymbol{T_{foot}}$ is the TM that describes the rigid foot.

(e) $\boldsymbol{V_s}(\theta_{7...12})$ is the TM that defines the 6-DOF virtual spring describing the foot stiffness.

(f) $\boldsymbol{V_{u1}}(q_1, q_2)$ is the homogenous matrix function representing the first 2-DOF passive U joint (foot-leg).

(g) $\boldsymbol{T_{leg}}$ is the TM that describes the rigid leg

(h) $\boldsymbol{V_s}(\theta_{13...18})$ is the *TM* that defines the 6-DOF virtual spring which describes the leg stiffness.

(i) $\boldsymbol{V_{u2}}(q_3, q_4)$ is the homogenous matrix function for the second 2-DOF passive U joint (leg-platform).

(j) $\boldsymbol{T_E}$ is the TM that describes the rigid link between the leg and the end-effector.

Note: $\boldsymbol{T_B}, \boldsymbol{T_{foot}}, \boldsymbol{T_{leg}}\ \boldsymbol{T_E}$ are constant homogenous matrices, $\boldsymbol{V_a}$ is an elementary translation or rotation, $\boldsymbol{V_{u1}}$ and $\boldsymbol{V_{u2}}$ are two successive rotations, and the matrices $\boldsymbol{V_s}$ are composed of six elementary transformations, the first three correspond to the spring translational deflections and the last three to the spring rotational deflections.

In order to develop the kinetostatic model, the next steps are followed: (a) developing the differential kinematic model, (b) calculating the stiffness models of the virtual springs, (c) applying the virtual work approach to find the static equations of the end-effector motion, and (d) putting together the complete kinetostatic model matrix equations. This process is summarized next.

(a) Differential kinematic model

The differential kinematic equations describe the relations between the end-effector location and small variations of the joint variables as follows:

$$\delta\boldsymbol{t} = \boldsymbol{J_\theta}\,\delta\boldsymbol{\theta} + \boldsymbol{J_q}\,\delta\boldsymbol{q} \qquad (18)$$

where:

$\delta\boldsymbol{t} = [\delta_p \quad \delta_\phi]^T$ is the vector of end-effector trans. $\delta\boldsymbol{p} = \begin{bmatrix}\delta p_x & \delta p_y & \delta p_z\end{bmatrix}^T$ and rot. $\delta\boldsymbol{\phi} = \begin{bmatrix}\delta\phi_x & \delta\phi_y & \delta\phi_z\end{bmatrix}^T$

$\boldsymbol{J_\theta} = \partial\boldsymbol{T}/\partial\boldsymbol{\theta}$ and $\boldsymbol{J_q} = \partial\boldsymbol{T}/\partial\boldsymbol{q}$ are the Jacobian matrices of sizes 6x19 and 6x4 respectively.

(b) Stiffness models

Since for small deformations a spring can be modeled with eq. (16); then the stiffness model of the virtual springs of the flexible model of a single kinematic chain can be formulated as:

$$[\tau_{\theta 0}] = \boldsymbol{K_{ctr}}\,[\theta_0]; \qquad\qquad [\tau_{\theta 1} \cdots \tau_{\theta 6}]^T = \boldsymbol{K_{act}}[\theta_1 \cdots \theta_6]^T;$$

$$[\tau_{\theta 7} \cdots \tau_{\theta 12}]^T = \boldsymbol{K_{foot}}[\theta_7 \cdots \theta_{12}]^T; \qquad [\tau_{\theta 13} \cdots \tau_{\theta 18}]^T = \boldsymbol{K_{leg}}[\theta_{13} \cdots \theta_{18}]^T; \qquad (19)$$

which can be grouped to the following matrix equation:

$$\boldsymbol{\tau_\theta} = \boldsymbol{K_\theta}\,\delta\boldsymbol{\theta} \qquad (20)$$

where: $\boldsymbol{K_{ctr}}, \boldsymbol{K_{act}}, \boldsymbol{K_{leg}}$ are the stiffness matrices of the actuator, foot and leg respectively,

$\boldsymbol{K_\theta} = diag(\boldsymbol{K_{ctr}}, \boldsymbol{K_{act}}, \boldsymbol{K_{foot}}, \boldsymbol{K_{leg}})$ is the spring stiffness matrix (19x19), and

$\boldsymbol{\tau_\theta} = [\tau_{\theta 0} \cdots \tau_{\theta 18}]^T$ is the vector of joint reactions.

Since there are no passive joint reactions; the next vector of zero values can be defined:

$$\boldsymbol{\tau_q} = \boldsymbol{0}_{nx1} \qquad (21)$$

(c) Virtual work principle

If a virtual external force f is applied to the end-effector (producing virtual displacements $\Delta\theta$ and Δq), then the virtual work of that external force is given by the left hand side of eq. (22). Similarly, for the internal forces the virtual work is given by the right hand side of eq. (22). Since the system is in static equilibrium, the external and internal work should be equal (total virtual work should be zero).

$$External\ Virtual\ Work\ =\ Internal\ Virtual\ Work$$
$$\boldsymbol{f}^T\boldsymbol{J}_\theta\Delta\boldsymbol{\theta} + \boldsymbol{f}^T\boldsymbol{J}_q\Delta\mathbf{q}\ =\ \boldsymbol{\tau}_\theta\Delta\boldsymbol{\theta} + \underset{\boldsymbol{0}}{\boldsymbol{\tau}'_q}\Delta\boldsymbol{q} \tag{22}$$

For eq. (22) to be true, the next must be true as well:

$$\boldsymbol{J}_\theta{}^T\boldsymbol{f} = \boldsymbol{\tau}_\theta \tag{23}$$
$$\boldsymbol{J}_q{}^T\boldsymbol{f} = 0 \tag{24}$$

(d) Complete kinetostatic model:

The complete kinetostatic model is given by equations (18), (20), (21), (23), and (24). Since, the stiffness matrix is non-singular, $\delta\theta$ can be solved from eq. (20), and substituting $\boldsymbol{\tau}_\theta$ from eq. (23) results in $\delta\boldsymbol{\theta} = \boldsymbol{K}_\theta{}^{-1}\boldsymbol{J}_\theta{}^T\boldsymbol{f},$ which then can be replaced in eq. (18). This leads to representing the complete kinetostatic model by only two equations as follows:

$$\boldsymbol{J}_\theta\,\boldsymbol{K}_\theta{}^{-1}\boldsymbol{J}_\theta{}^T\boldsymbol{f} + \boldsymbol{J}_q\,\delta\boldsymbol{q} = \delta\boldsymbol{t} \tag{25}$$
$$\boldsymbol{J}_q{}^T\boldsymbol{f} = 0 \tag{26}$$

or in matrix form as:

$$\begin{bmatrix} \boldsymbol{S}_\theta & \boldsymbol{J}_q \\ \boldsymbol{J}_q{}^T & \boldsymbol{0} \end{bmatrix}\begin{bmatrix} \boldsymbol{f} \\ \delta\boldsymbol{q} \end{bmatrix} = \begin{bmatrix} \delta\boldsymbol{t} \\ \boldsymbol{0} \end{bmatrix} \tag{27}$$

with: $\boldsymbol{S}_\theta = \boldsymbol{J}_\theta\,\boldsymbol{K}_\theta{}^{-1}\boldsymbol{J}_\theta{}^T = \boldsymbol{J}_{\theta ctr}\,\boldsymbol{K}_{ctr}{}^{-1}\boldsymbol{J}_{\theta ctr}^T + \boldsymbol{J}_{\theta act}\,\boldsymbol{K}_{act}{}^{-1}\boldsymbol{J}_{\theta act}^T + \boldsymbol{J}_{\theta foot}\,\boldsymbol{K}_{foot}{}^{-1}\boldsymbol{J}_{\theta foot}^T + \boldsymbol{J}_{leg}\,\boldsymbol{K}_{leg}{}^{-1}\boldsymbol{J}_{\theta leg}^T,$ and
$\boldsymbol{K}_\theta{}^{-1} = diag\big(\boldsymbol{K}_{ctr}^{-1},\boldsymbol{K}_{act}^{-1},\boldsymbol{K}_{foot}^{-1},\boldsymbol{K}_{leg}^{-1}\big)$

Following eq. (16), the motion-to-force mapping of the end-effector for a separate kinematic chain has to be of the form $\boldsymbol{f} = \boldsymbol{K}\delta\mathbf{t}.$ Consequently eq. (27) can be reformulated as:

$$\begin{bmatrix} \boldsymbol{f} \\ \delta\boldsymbol{q} \end{bmatrix} = \begin{bmatrix} \boldsymbol{S}_\theta & \boldsymbol{J}_q \\ \boldsymbol{J}_q{}^T & \boldsymbol{0} \end{bmatrix}^{-1}\begin{bmatrix} \delta\boldsymbol{t} \\ \boldsymbol{0} \end{bmatrix} \quad or \quad \begin{bmatrix} \boldsymbol{f} \\ \delta\boldsymbol{q} \end{bmatrix} = \boldsymbol{K}_{tot}\begin{bmatrix} \delta\boldsymbol{t} \\ \boldsymbol{0} \end{bmatrix} \tag{28}$$

Therefore, the stiffness matrix $\boldsymbol{K}$ for a separate kinematic chain can be calculated by extracting the corresponding 6x6 sub-matrix from $\boldsymbol{K}_{tot}$ corresponding to $\boldsymbol{S}_\theta.$

These equations correspond to a single kinematic chain. After the stiffness matrices $\boldsymbol{K}_i$ for all kinematic chains are computed, the stiffness of the entire robot can be found by adding them all ($\boldsymbol{K}_{ROB} = \sum\boldsymbol{K}_i$). This comes from the fact that the total external force corresponding to the end-effector displacement ($\delta\mathbf{t}$) can be calculated as $\boldsymbol{f}_{tot} = \sum\boldsymbol{f}_i = \sum\boldsymbol{K}_i\delta\mathbf{t}$ since the displacement is the same for all kinematic chains (superposition principle).

In the next section, the stiffness analysis discussed is implemented to obtain the elastostatic models of the *RPaPa* ANTHRO and *RPaR* ORTH manipulators.

3.3. Application of the model

The purpose of this section is to develop the elastostatic models of the *RPaPa* ANTHRO and *RPaR* ORTH manipulators under study. The link characteristics, which apply for both robots, are discussed first. After this, the elastostatic model of both manipulators is presented.

3.3.1 Link characteristics

In order to model the manipulators, the shape of their links has to be defined first. All the links are modeled as beams with circular cross section of radius r. From mechanics, the stiffness (K_{beam}) and compliance matrices (k_{beam}) of a beam with these features can be defined as shown next:

$$K_{beam} = \begin{bmatrix} \dfrac{AE}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & \dfrac{12\,E\,I_{zz}}{L^3} & 0 & 0 & 0 & -\dfrac{6\,E\,I_{zz}}{L^2} \\ 0 & 0 & \dfrac{12\,E\,I_{yy}}{L^3} & 0 & \dfrac{6\,E\,I_{yy}}{L^2} & 0 \\ 0 & 0 & 0 & \dfrac{G\,J_t}{L} & 0 & 0 \\ 0 & 0 & \dfrac{6\,E\,I_{yy}}{L^2} & 0 & \dfrac{4\,E\,I_{yy}}{L} & 0 \\ 0 & -\dfrac{6\,E\,I_{zz}}{L^2} & 0 & 0 & 0 & \dfrac{4\,E\,I_{zz}}{L} \end{bmatrix} \qquad k_{beam} = \begin{bmatrix} \dfrac{L}{AE} & 0 & 0 & 0 & 0 & 0 \\ 0 & \dfrac{L^3}{3\,E\,I_{zz}} & 0 & 0 & 0 & \dfrac{L^2}{2\,E\,I_{zz}} \\ 0 & 0 & \dfrac{L^3}{3\,E\,I_{yy}} & 0 & -\dfrac{L^2}{2\,E\,I_{yy}} & 0 \\ 0 & 0 & 0 & \dfrac{L}{G\,J_t} & 0 & 0 \\ 0 & 0 & -\dfrac{L^2}{2\,E\,I_{yy}} & 0 & \dfrac{L}{E\,I_{yy}} & 0 \\ 0 & \dfrac{L^2}{2\,E\,I_{zz}} & 0 & 0 & 0 & \dfrac{L}{E\,I_{zz}} \end{bmatrix}$$

where: L is the longitude of the beam, A is the area of the circular cross section, E and G are the Young's and shear modulus of the material, $I_{yy} = I_{zz} = \pi r^4/4$ are the circular cross-section second moments of area, and $J_t = \pi r^4/2$ is the circular cross-section torsion constant.

Lastly, it is important to mention that the longitude of the beams is along its x-axis, and the cross section area is on the plane y-z.

3.3.2 Elastostatic modeling of the manipulators

The elastostatic model of both manipulators can be easily developed by analyzing the kinematic chains that compose them; then, obtaining its transformation matrices, and finally calculating the corresponding Jacobian matrices as explained in **Section 3.2**.

Kinematic chains of the manipulators

As it can be deduced from **Figure 2.5**, and **Figure 2.6** both manipulators are composed of two kinematic chains in series as explained next (note: for notation convenience, the term kinematic chain may be abbreviated as *KCh* for the remaining of this report).

- For the *RPaPa* ANTHRO manipulator (see **Table 3.2,** column a):
  1st kinematic chain: Parallelogram with constant base rotation angle of: $\beta_1 = 32°$
  2nd kinematic chain: Parallelogram with constant base rotation angle of: $\beta_1 + \beta_3 = 32° + 110°$

- For the *RPaR* ORTH manipulator (see **Table 3.3**, column a):
  1st kinematic chain: Parallelogram with constant base rotation angle of: $\beta_1 = 90°$
  2nd Kinematic chain: Link parallel to x-y plane rotating around z-axis.

Table 3.2 *RPaPa* ANTHRO elastostatic modeling
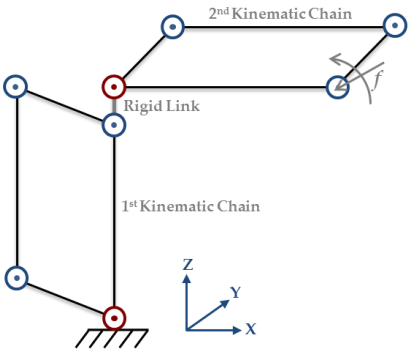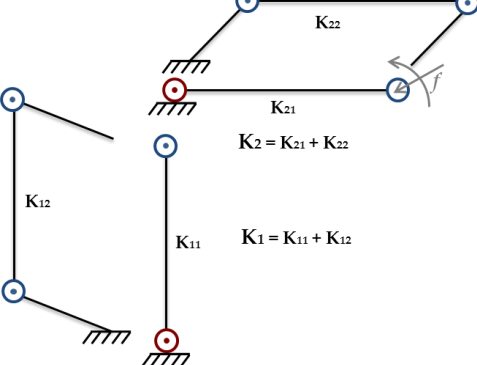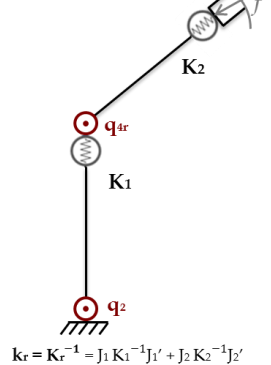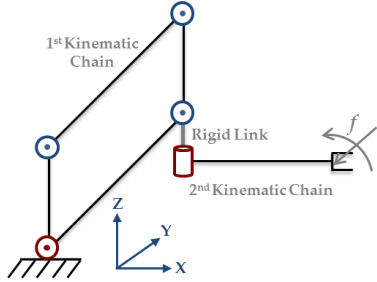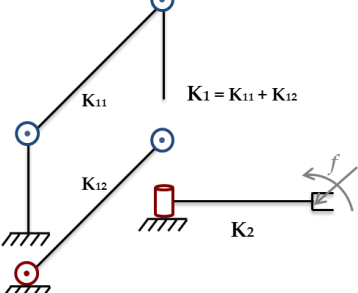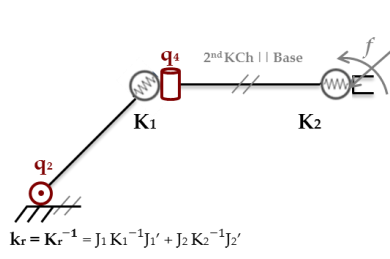
a) *RPaPa* ANTHRO kinematic chains	b) Sub-division of chains	c) Equivalent chains in series

Table 3.3 *RPaR* ORTH elastostatic modeling

a) *RPaPa* ANTHRO kinematic chains	b) Sub-division of chains	c) Equivalent chains in series

The next step consists of finding the stiffness matrix of each kinematic chain (K_1 and K_2) for both robots. It can be seen that each of the three kinematic chains that involve parallelograms can be divided in two sub-chains. The stiffness matrix of the sub-chains can be found following the equations previously explained; then, by adding these together, the stiffness matrix of each parallelogram is obtained, that is K_1 and K_2 for the *RPaPa* ANTHRO robot and K_1 for *RPaR* ORTH robot (see **Table 3.2** and **Table 3.3**, column *b*). Moreover, K_2 of the *RPaR* ORTH robot can be calculated with the described equations as well. Finally, an equivalent model composed of two chains in series (with stiffness K_1 and K_2 respectively) can be developed to find the global stiffness matrix of each robot (see column *c* of **Table 3.2** and **Table 3.3**). Consequently, the stiffness analysis of both manipulators can be simplified by performing the general elastostatic model of a parallelogram with a constant base rotation of β, and the elastostatic model of a link parallel to x-y plane rotating around z-axis. Therefore, what follows is to find the transformation matrices of the chains and sub-chains previously described, as shown next.

Transformation matrix for parallelogram with constant base rotation of β

As discussed, the parallelogram can be divided in two sub kinematic chains as follows:

▪ For the first sub-chain of the parallelogram:

$$\boldsymbol{T}_{para}^{ch1} = \boldsymbol{T}_B \cdot \boldsymbol{T}_{da} \cdot \boldsymbol{V}_1(\theta_{1\dots6}) \cdot \boldsymbol{T}_{Rq_a}(q) \cdot \boldsymbol{T}_L \cdot \boldsymbol{V}_2(\theta_{7\dots12}) \cdot \boldsymbol{T}_{Rq_b}(q) \cdot \boldsymbol{T}_{db} \cdot \boldsymbol{V}_s(\theta_{13\dots18}) \cdot \boldsymbol{V}_{u2}(q_3, q_4) \cdot \boldsymbol{T}_E \quad \textbf{(29)}$$

where:

(a) $\boldsymbol{T}_B = R_y[-(\pi - \beta)]$

(b) $T_{da} = T_x(d)$, with d as the length of the "short" link

(c) $V_1(\theta_{1\dots6}) = T_x(\theta_1) \cdot T_y(\theta_2) \cdot T_z(\theta_3) \cdot R_x(\theta_4) \cdot R_y(\theta_5) \cdot R_z(\theta_6)$

(d) $T_{Rq_a}(q) = R_y[-(q - \pi - \beta)]$, with q as the active joint variable

(e) $T_L = T_x(L)$, with L as the length of the "long" link

(f) $V_2(\theta_{7\dots12}) = T_x(\theta_7) \cdot T_y(\theta_8) \cdot T_z(\theta_9) \cdot R_x(\theta_{10}) \cdot R_y(\theta_{11}) \cdot R_z(\theta_{12})$

(g) $T_{Rq_b}(q) = R_y[-(-q - \beta)]$

(h) $T_{db} = T_x(d)$

(i) $V_3(\theta_{13\dots18}) = T_x(\theta_{13}) \cdot T_y(\theta_{14}) \cdot T_z(\theta_{15}) \cdot R_x(\theta_{16}) \cdot R_y(\theta_{17}) \cdot R_z(\theta_{18})$

(j) $T_E = R_y[-(\beta)]$

- ▪ For the second sub-chain of the parallelogram:

$$T_{parallelogram}^{ch2} = T_B \cdot T_{Rq_a}(q) \cdot T_L \cdot V_1(\theta_{1\dots6}) \cdot T_{Rq_b}(q) \cdot T_E \qquad (30)$$

where:

(a) $T_B = I_{4x4}$

(b) $T_{Rq_a}(q) = R_y[-(q)]$

(c) $T_L = T_x(L)$

(d) $V_1(\theta_{1\dots6}) = T_x(\theta_1) \cdot T_y(\theta_2) \cdot T_z(\theta_3) \cdot R_x(\theta_4) \cdot R_y(\theta_5) \cdot R_z(\theta_6)$

(e) $T_{Rq_b}(q) = R_y[-(-q)]$

(f) $T_E = I_{4x4}$

Transformation matrix for link parallel to x-y plane rotating around z-axis

$$T_{Link\_RotZ} = T_B \cdot T_{Rq}(q) \cdot T_L \cdot V_1(\theta_{1\dots6}) \cdot T_{Rq_b}(q) \cdot T_E(q) \qquad (31)$$

where:

(a) $T_B = I_{4x4}$

(b) $T_{Rq}(q) = R_z(q)$, with q as the joint variable

(c) $T_L = T_x(L)$, with L as the length of the link

(d) $V_1(\theta_{1\dots6}) = T_x(\theta_1) \cdot T_y(\theta_2) \cdot T_z(\theta_3) \cdot R_x(\theta_4) \cdot R_y(\theta_5) \cdot R_z(\theta_6)$

(e) $T_E(q) = R_z(-q)$

Transformation matrix for the equivalent chains in series

Lastly, the global transformation matrix of the robots can be calculated as shown in eq. **(32)** and **(33)**. As it is noticed, this approach requires that the equations describe the position of the end effector by using translation matrices (see T_{KCh1} and T_{KCh2}).

$$T_{RPaPa\_ANTHRO} = T_B \cdot T_{KCh1}(q_{KCh1}) \cdot V_1(\theta_{1\dots6}) \cdot T_{RigidLink} \cdot V_{RL}(\theta_{7\dots12}) \cdot T_{KCh2}^{ANTHRO}(q_{KCh1}, q_{KCh2}) \cdot V_3(\theta_{13\dots18}) \cdot T_E \quad (32)$$

$$T_{RPaR\_ORTH} = T_B \cdot T_{KCh1}(q_{KCh1}) \cdot V_1(\theta_{1\dots6}) \cdot T_{RigidLink} \cdot V_{RL}(\theta_{7\dots12}) \cdot T_{KCh2}^{ORTH}(q_{KC2}) \cdot V_3(\theta_{13\dots18}) \cdot T_E \qquad (33)$$

where:

(a) $T_B = I_{4x4}$

(b) $T_{KCh1} = T_x[L_{KCh1} \cdot \cos(q_{KCh1})] \cdot T_z[L_{KCh1} \cdot \sin(q_{KCh1})]$, with L_{KC1} as the length of the link of the first kinematic chain and q_{KCh1} as the active joint variable of the first kinematic chain.

(c) $\boldsymbol{V_1}(\theta_{1\ldots6}) = T_x(\theta_1) \cdot T_y(\theta_2) \cdot T_z(\theta_3) \cdot R_x(\theta_4) \cdot R_y(\theta_5) \cdot R_z(\theta_6)$

(d) $\boldsymbol{T_{RigidLink}} = T_x(r_{tx}) \cdot T_y(r_{ty}) \cdot T_z(r_{tz})$

(e) $\boldsymbol{T_L} = T_x(L)$

(f) $\boldsymbol{V_{RL}}(\theta_{7\ldots12}) = T_x(\theta_7) \cdot T_y(\theta_8) \cdot T_z(\theta_9) \cdot R_x(\theta_{10}) \cdot R_y(\theta_{11}) \cdot R_z(\theta_{12})$

(g) $\boldsymbol{T_{KCh2}^{ANTHRO}} = T_x[L_{KCh2} \cdot \cos(q_{KC1} + q_{KCh2})] \cdot T_z[L_{KC2} \cdot \sin(q_{KCh1} + q_{KCh2})]$, L_{KCh2} and q_{KCh2} as (b)

(h) $\boldsymbol{T_{KCh2}^{ORTH}} = T_x[L_{KCh2} \cdot \cos(q_{KC2})] \cdot T_y[L_{KCh2} \cdot \sin(q_{KC2})]$

(i) $\boldsymbol{V_3}(\theta_{13\ldots18}) = T_x(\theta_{13}) \cdot T_y(\theta_{14}) \cdot T_z(\theta_{15}) \cdot R_x(\theta_{16}) \cdot R_y(\theta_{17}) \cdot R_z(\theta_{18})$

(j) $\boldsymbol{T_E} = I_{4x4}$

Moreover, a rigid link ($\boldsymbol{T_{RigidLink}}$) is included in the model. Its purpose is to allow joining the two kinematic chains for both robots. This rigid link consists of a constant translation along the x, y or z-axis (in positive or negative direction). The rigid link can be modeled by considering it is made of a high-stiffness material. The transformation matrix of the link can be calculated similarly to eq. **(31)** (depending along which axis is located the rigid link, and it will have a constant rotation of $\pi/2$ or $-\pi/2$ according to its direction). After obtaining the transformation matrices, it is possible to calculate the Jacobian matrices $\boldsymbol{J_\theta}$ and $\boldsymbol{J_q}$ for the entire model. The approach followed is explained next.

Calculation of the Jacobian matrices

From eq. **(17)**, it is noticeable that the Jacobian matrices might be calculated analytically by differentiation of the transformation matrices, but as [**4**] states, this method is not recommended because it may yield to awkward and difficult-to-work expressions. Thus, the semi-analytical method presented in [**4**] and [**20**] is implemented. This approach is explained next. Since all the variables in eq. **(17)** are separated then for a particular virtual joint variable θ_j, this equation can be rewritten as:

$$\boldsymbol{T} = \boldsymbol{T_{Left}} \cdot \boldsymbol{V}_{\theta_j}(\theta_j) \cdot \boldsymbol{T_{Right}} \tag{34}$$

where: $\boldsymbol{T_{Left}}$ and $\boldsymbol{T_{Right}}$ are the homogeneous matrices made of the left-side and right-side elements of the original equation respectively, and $\boldsymbol{V}_{\theta_j}(\theta_j)$ is the transformation matrix corresponding to θ_j.

Therefore, the partial derivative of the transformation matrix $\boldsymbol{T}$ with respect to θ_j at $\theta_j = 0$ can be computed as follows:

$$\boldsymbol{T'} = \boldsymbol{T_{Left}} \cdot \boldsymbol{V'}_{\theta_j}(\theta_j) \cdot \boldsymbol{T_{Right}} \tag{35}$$

where: $\boldsymbol{T_{Left}}$ and $\boldsymbol{T_{Right}}$ remain the same since they are constant for the partial derivative, and $\boldsymbol{V'}_{\theta_j}(\theta_j)$ is the derivative of an elementary translation or rotation (according to θ_j) evaluated at the point $\theta_j = 0$, as seen in eq. **(36).**

$$
T'_x(\theta_j = 0) = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad
T'_y(\theta_j = 0) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad
T'_z(\theta_j = 0) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}
$$

$$
R'_x(\theta_j = 0) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad
R'_y(\theta_j = 0) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad
R'_z(\theta_j = 0) = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}
\tag{36}
$$

In addition, since the derivative T' may be also represented as:

$$V'_{\theta_j} = \begin{bmatrix} 0 & -\phi_z' & \phi_y' & Px' \\ \phi_z' & 0 & -\phi_x' & Py' \\ -\phi_y' & \phi_x' & 0 & Pz' \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{37}$$

Therefore, the *jth* column of J_θ can be obtained from the elements of the matrix T', after applying eq. (35), as follows:

$$J_{\theta_j} = \begin{bmatrix} Px' & Py' & Pz' & \phi_x' & \phi_y' & \phi_z' \end{bmatrix}^T \tag{38}$$

The columns of J_q (when there are passive joints) are obtained in a similar way. It is just necessary to apply a simple transformation to the joint variable: $q_j = q_j^{nom} + \delta q_j$, thus $V_{q_j}(q_j) = V_{q_j}(q_j^{nom}) + V_{q_j}(\delta q_j)$, and follow the procedure just explained.

3.4. Validation of the model

The elastostatic model of both manipulators can be validated against a finite element analysis software such as the program RDM[2] version 6.17. A summary of this analysis can be seen in **Table 3.4** and **Table 3.5.** The model is evaluated by applying a load to the end-effector (forces and moments columns in the tables) and obtaining the corresponding linear and rotational displacements (δp and δφ columns). Then, the same conditions are implemented in RDM and the results are compared. The elastostatic models are evaluated for different configurations of the robot (Config column) and for different load cases. It can be seen that the results are completely satisfactory. The relative error (expressed in percentage form) between the developed models developed and the FEA software is minimal.

RPaPa ANTHRO Elastostatic Model Validation Results											
Config [deg]		Forces [N]		Moments [Nm]		Results	δpx	δpy	δpz	δφx	δφy	δφz
Test 1												
q1	---	Fx	0	Mx	0	FEA	-6.33E-04	0.00E+00	-6.75E-03	0.00E+00	5.53E-03	0.00E+00
q2	80	Fy	0	My	0	Model	-6.33E-04	0.00E+00	-6.75E-03	0.00E+00	5.53E-03	0.00E+00
q4r	-100	Fz	-2000	MZ	0	error %	0.110%	---	0.006%	---	0.005%	---
Test 2												
q1	---	Fx	-500	Mx	-350	FEA	-9.42E-04	-1.62E-02	-3.55E-03	-8.98E-03	3.55E-03	-1.63E-02
q2	80	Fy	-300	My	250	Model	-9.42E-04	-1.62E-02	-3.55E-03	-8.98E-03	3.55E-03	-1.63E-02
q4r	-100	Fz	-800	MZ	-450	error %	0.001%	0.007%	0.001%	0.002%	0.001%	0.005%
Test 3												
q1	---	Fx	0	Mx	250	FEA	1.92E-03	-2.25E-03	-3.53E-02	8.59E-03	2.25E-02	5.77E-05
q2	90	Fy	0	My	400	Model	1.92E-03	-2.25E-03	-3.53E-02	8.59E-03	2.25E-02	5.76E-05
q4r	-90	Fz	-2060	MZ	0	error %	0.140%	0.040%	0.000%	0.011%	0.001%	0.337%
Test 4												
q1	---	Fx	700	Mx	400	FEA	6.06E-03	8.30E-03	-1.54E-02	4.57E-03	2.07E-03	4.96E-04
q2	90	Fy	1000	My	-500	Model	6.05E-03	8.30E-03	-1.54E-02	4.57E-03	2.07E-03	4.96E-04
q4r	-90	Fz	-1200	MZ	-800	error %	0.102%	0.018%	0.005%	0.032%	0.039%	0.067%
Test 5												
q1	---	Fx	-500	Mx	-350	FEA	-3.80E-03	-2.31E-02	-1.56E-02	-1.33E-02	1.07E-02	-2.19E-02
q2	50	Fy	-300	My	250	Model	-3.81E-03	-2.31E-02	-1.56E-02	-1.33E-02	1.07E-02	-2.19E-02
q4r	-70	Fz	-800	MZ	-450	error %	0.086%	0.030%	0.023%	0.022%	0.007%	0.019%

Table 3.4 *RPaPa* **ANTHRO elastostatic modeling validation results**

---

[2] RDM is a Finite Element Analysis computer software written by Debard of the Institue Universitaire de Technologie du Mans (see http://iut.univ-lemans.fr/ydlogi/rdm_version_6.html)

RPaR ORTH Elastostatic Model Validation Results											
Config [deg]		Forces [N]		Moments [Nm]		Results	δpx	δpy	δpz	δφx	δφy	δφz
Test 1											
q1	---	Fx	0	Mx	0	FEA	-8.06E-04	0.00E+00	-3.89E-02	0.00E+00	3.11E-02	0.00E+00
q2	0	Fy	0	My	0	Model	-8.06E-04	0.00E+00	-3.89E-02	0.00E+00	3.11E-02	0.00E+00
q4	0	Fz	-2000	MZ	0	error %	0.000%	---	0.000%	---	0.000%	---
Test 2											
q1	---	Fx	-500	Mx	-350	FEA	1.27E-03	-2.79E-02	-1.65E-02	-1.73E-02	1.95E-02	-2.83E-02
q2	30	Fy	-300	My	250	Model	1.27E-03	-2.79E-02	-1.65E-02	-1.73E-02	1.95E-02	-2.83E-02
q4	0	Fz	-800	MZ	-450	error %	0.043%	0.043%	0.007%	0.012%	0.001%	0.026%
Test 3											
q1	---	Fx	0	Mx	0	FEA	9.28E-03	-5.08E-03	-4.14E-02	-3.10E-02	2.21E-02	-4.43E-03
q2	30	Fy	0	My	0	Model	9.28E-03	-5.07E-03	-4.14E-02	-3.10E-02	2.21E-02	-4.43E-03
q4	45	Fz	-2000	MZ	0	error %	0.006%	0.072%	0.006%	0.003%	0.003%	0.063%
Test 4											
q1	---	Fx	-500	Mx	-350	FEA	1.09E-02	-1.37E-02	-2.42E-02	-2.59E-02	1.54E-02	-1.82E-02
q2	30	Fy	-300	My	250	Model	1.09E-02	-1.37E-02	-2.42E-02	-2.59E-02	1.54E-02	-1.82E-02
q4	45	Fz	-800	MZ	-450	error %	0.030%	0.051%	0.001%	0.006%	0.001%	0.030%
Test 5											
q1	---	Fx	-500	Mx	-350	FEA	-1.33E-02	5.89E-03	-5.00E-03	6.35E-05	-1.11E-04	-2.46E-02
q2	30	Fy	-300	My	250	Model	-1.33E-02	5.89E-03	-5.00E-03	6.50E-05	-1.11E-04	-2.46E-02
q4	-125	Fz	-800	MZ	-450	error %	0.037%	0.003%	0.031%	2.356%	0.202%	0.023%

Table 3.5 *RPaR* **ORTH elastostatic modeling validation results**

Chapter 4

Design Parameter Optimization

One of the goals of this research work is to propose a more robust methodology to define the robots parameters in order to make the comparison between the two manipulators as fair as possible. One of the possible ways to do this is by performing an optimization procedure of the robot design parameters. Thus, the current chapter has the objective of presenting the methodology for the optimal design of the primary and secondary parameters of a parallel robot proposed in [3] and [21], and how it is applied to the robots under study.

4.1   Presentation of the methodology

The design of a robot is a very complex task because a wide variety of properties (such as mechanical and dynamic properties), diverse models (such as geometric, dynamic, elastostatic and elastodynamic models), architecture family, design constraints and desired characteristics must be taken into account. Thus, the design of a robot involves simultaneous optimization of many types of criteria that may evaluate the kinematic, the kinetostatic or the dynamic properties of the manipulator. If the architecture of the robot is known (like in the case of this research work), the design of a manipulator may be decomposed into two phases as follows:

1) First Phase: *optimization of primary parameters* [21]. It involves finding the values of the primary geometric parameters, such as the length of links, the base radius or the joint limits using the geometric, kinematic and kinetostatic constraints and objectives (workspace, velocity, effort transmission, etc.). This process is based on relatively simple geometric kinematic and kinetostatic models so it is easier to achieve.

2) Second Phase: *optimization of secondary parameters* [3]: It involves finding the values of the secondary geometric parameters, such as the cross-section or the shape of links, using the dynamic, elastostatic and/or elastostatic constraints and objectives (acceleration capabilities, maximal deformations of the tool, natural frequencies of the structure, etc.). This process is more complicated and time-consuming than the first step.

The formulation of the design optimization problem, for both phases, is summarized in the following paragraphs. First, the geometry of the manipulators may be defined by the following mapping:

$$g : \Phi \to W \tag{39}$$

where: $\Phi = \phi_1 \times \cdots \phi_n$ is the configuration space, composed of the joint coordinates,

$W = \mathrm{p}_1 \times \cdots \mathrm{p}_n$ is the workspace, composed of the end-effector coordinates, and

n is the number of degrees of freedom of the robot.

Then, the mechanical properties of the manipulator for each workspace point $p \in W$ and for any given set of design parameters π can be described by the next matrices:

For the first phase:	For the second phase:
$K_v(p,\pi),\;\; K_f(p,\pi),\;\; K_a(p,\pi)$ **(40)**	$K_d(p,\pi),\;\; K_s(p,\pi),\;\; K_v(p,\pi)$ **(41)**
which represent the velocity, force transmission, and accuracy properties, among others.	*which represent the dynamics and stiffness properties among others.*

For each of these matrices, physical consistent scalar measures are defined that may be directly included in the design objectives or constrains, as shown next:

For the first phase:	For the second phase:
$for\;\mathrm{K}_\alpha,\alpha \in \{v,f,a,\ldots\}$ $there\;are\;\sigma_\beta(K_\alpha),\beta \in \{i,t,\ldots\}$ **(42)**	$for\;\mathrm{K}_\alpha,\alpha \in \{d,s,\ldots\},$ $there\;are\;\sigma_\beta(K_\alpha),\beta \in \{i,a,\ldots\}$ **(43)**
such physical measures could be isotropy, transmission factors, etc.	*such physical measures could be input efforts, accuracy of the tool, etc.*

Likewise, the performance measures, which depend on the adopted geometrical structure g and the physical parameters π of the links, for the global evaluations of the robot can be defined as:

For the first phase:	For the second phase:
$n_\gamma(g,\pi)$ $\gamma \in \{m,l,w,\ldots\}$ **(44)**	$n_\gamma(g,\pi)$ $\gamma \in \{m,e,n,\ldots\}$ **(45)**
such performance measures could be the total mass of the manipulators, the length of the links, the workspace size, etc.	*such performance measures could be the total mass, the maximum input effort of the links, the minimum admissible natural frequency, etc.*

Finally, the design optimization problem for both stages can be formulated as follows:

Achieve the best possible value of the performance indices:
$$\boldsymbol{n_\gamma(g,\pi) \to \; min}, \;\; \forall\boldsymbol{\gamma} \qquad\qquad \textbf{(46)}$$
$$\pi$$
Subject to the constraints:
$$\sigma_\beta(\,K_\alpha(p,\pi)\,) \in \; \mathrm{S}_\beta, \;\;\; \forall\alpha,\beta \qquad\qquad \textbf{(47)}$$

which must be satisfied for all points of W, where W includes the manufacturing task.

After having defined the main concepts of the methodology, the next section explains how the method is applied to this research work. Finally, the results obtained are presented at the end of this chapter.

4.2   Application of the methodology

The purpose of applying a design optimization procedure on the two manipulators is to propose a robust methodology to define the parameters of the robots, thus assuring a fair dynamic comparison between them. From the previous section it can be observed that there are two main components that should be selected for the optimization methodology: the optimization criteria based on a performance measure and the design constraints.

Regarding the first component, it is reasonable to select the mass of the robot as a performance measure since the dynamic performance of a manipulator is influenced to a large degree by its mass. Thus the objective of the optimization is set to minimizing the total mass of the manipulators. The latter is defined formally in **Section 4.2.1** and **Section 4.2.2**.

Concerning the second element, it can be divided into two parts: the geometric constraints and the constraints related to the desired mechanical properties. The geometric constraints are straightforward to understand, these are linked to the size of the links and they are explained in detail in **Section 4.2.3**. The second kind of constraints is related with a mechanical property that the designer wants for the manipulator to have, as described in the work of [**3**] and [**21**]. Consequently, according to the objectives of this thesis, the stiffness of the manipulator is selected as the mechanical property to be taken into consideration. In order to consider this mechanical property as a constraint, a physical scalar measure must be defined. Since the stiffness of the manipulator relates how the robot deforms under a static load (as it was explained in the previous section); therefore, the maximum allowable deformation under a static load at the end effector (set by the designer) may be defined as the physical scalar measure and included directly in the design constraints. Moreover, according to each specific design case, this mechanical property may be desired for the whole workspace of the manipulator or for a certain region (which will include the manufacturing task). Since the purpose of the parameter optimization procedure for this research work is to define in a fair way the mass of the manipulator, it is sufficient to define a working area of a certain shape and size that is inside the workspace of the robots and for which the design constraint is fulfilled. The implementation of the mechanical design constraint is explained thoroughly in **Section 4.2.4**.

Lastly, some important details of the implementation of the parameter optimization procedure done for this research work are explained in **Section 4.2.5**.

4.2.1   Definition of the optimization criteria

The optimization procedure has the objective of minimizing the total mass of the manipulator (which depends on the mass of its composing links). Therefore, the performance index $n_\gamma(g, \pi)$ for both manipulators is defined as:

$$n_{massRobot}(g, \pi) \rightarrow \min_{\pi}(massRobot) \tag{48}$$

where:

$\pi = \{link\ parameters\} = \{L_1 \dots L_n, R_1 \dots R_n\}$,

$massRobot = \{m_1 + m_2 + \cdots + m_n\}$ is the total mass of the manipulator, and

L_j, R_j and m_j are the length, radius of circular cross-section area and mass of link j.

4.2.2 Definition of the optimization variables

From the definition of the optimization criteria, it is clear that the variables to be optimized are the length of the links and the radius of the circular cross-section area of each beam since the masses of the links depend on them. As in a parallelogram the opposite links have the same length, the optimization variables for each manipulator can be defined as shown in **Table 4.1**.

Table 4.1 Design-parameter optimization variables	
RPaPa ANTHRO optimization variables	*RPaR* ORTH optimization variables
Link Lengths: **For the 1st *KCh*:** L_{KCh_1} = Length of 1st *KCh* = L_2 = L_6 L_{Base_1} = Length of Base/Top of 1st *KCh* = L_1 = L_{3b} **For the 2nd *KCh*:** L_{KCh_2} = Length of 2nd *KCh* = L_4 = L_7 L_{Base_2} = Length of Base/Top of 2nd *KCh* = L_{3a} = L_{5b}	**Link Lengths:** **For the 1st *KCh*:** L_{KCh_1} = Length of 1st *KCh* = L_2 = L_5 L_{Base_1} = Length of Base/Top of 1st *KCh* = L_1 = L_{3b} **For the 2nd *KCh*:** L_{KCh_2} = Length of 2nd *KCh* = L_4
Link Radii: **For the 1st *KCh*:** Ra_{KCh_1}= Radius of Pa active link of 1st *KCh* = R_{link2} Rp_{KCh_1}= Radius of Pa passive link of 1st *KCh* = R_{link6} Rb_{KCh_1}= Radius of Pa base of 1st *KCh* = R_{link1} Rt_{KCh_1}= Radius of Pa platform (top) of 1st *KCh* = R_{link3b} **For the 2nd *KCh*:** Ra_{KCh_2}= Radius of Pa active link of 2nd *KCh* = R_{link4} Rp_{KCh_2}= Radius of Pa passive link of 2nd *KCh* = R_{link7} Rb_{KCh_2}= Radius of Pa base of 2nd *KCh* = R_{link3a} Rt_{KCh_2} = Radius of Pa platform (top) of 2nd *KCh* =R_{link5b}	**Link Radii:** **For the 1st *KCh*:** Ra_{KCh_1}= Radius of Pa active link of 1st *KCh* = R_{link2} Rp_{KCh_1}= Radius of Pa passive link of 1st *KCh* = R_{link5} Rb_{KCh_1}= Radius of Pa base of 1st *KCh* = R_{link1} Rt_{KCh_1}= Radius of Pa platform (top) of 1st *KCh* = R_{link3b} **For the 2nd *KCh*:** Ra_{KCh_2}= Radius of Pa link of 2nd *KCh* = R_{link4}

4.2.3 Definition of geometric constraints

There are certain geometry constraints that must be defined to make sure that the parameters found by the optimization procedure lead to a robot that can actually be built. For example, a robot with beams of really thin radius and large lengths, or vice versa, is not desired. Moreover, it is also preferred to constraint the ratio between the link lengths of the first kinematic chain and the second kinematic chain (a robot with a large first kinematic chain but small second kinematic chain, or the contrary, is not desired either). Under these and other considerations the next geometry constraints are defined.

Constraint 1: The ratio between the lengths of the active/passive and the base/platform links of the parallelograms should be restrained; thus, avoiding that the manipulators have a really long base in comparison to the parallelogram size, as follows:

$$\kappa \cdot L_{KCh_j} \leq L_{Base_j} \leq \lambda \cdot L_{KCh_j}$$
$$\text{for } j = 1, 2 \text{ for ANTHRO and } j = 1 \text{ for ORTH}$$

(49)

where:
  κ and λ are constant coefficient with: $0 < \kappa < 1$, $0 < \lambda < 1$, and $\lambda > \kappa$,
  with values for the optimization procedure of $\kappa = 0.2 = 20\%$ and $\lambda = 0.7 = 70\%$.

Constraint 2: The sum of the radius of the active and passive links of the parallelograms must be smaller than the longitude of the base/platform (to be feasible to be assembled):

$$\kappa \cdot L_{KCh_j} \leq L_{Base_j} \leq \lambda \cdot L_{KCh_j}$$
$$\text{for } j = 1, 2 \text{ for ANTHRO and } j = 1 \text{ for ORTH}$$

(50)

where:

        v is a constant coefficient with: $0 < v < 1$,

        with a value for the optimization procedure of $v = 0.9 = 90\%$.

Constraint 3: All the optimization variables must have upper and lower bounds, which are up to the designer (respecting the previous 2 constraints).

Constraint 4: The ratio between the link lengths of the first *KCh* (L_{KCh_1}) and the second *KCh* (L_{KCh_2}) for both robots should be selected carefully to obtain the best kinematic properties and workspace shape and area. Since this constrained requires more attention, it is explained in the next sub-section.

**Selection of the ratio between L$_{KCh1}$ and L$_{KCh2}$**

The definition of constraint 4 which establishes the ratio between L_{KCh_1} and L_{KCh_2} is very important since the workspace of the manipulator and its kinematic properties depend on it. The selection for both of them is explained next.

- ▪ *RPaPa* ANTHRO constraint 4 definition:

Regarding manipulators of the anthropomorphic family, it has been extensively studied that the best ratio between the link lengths of the first and second kinematic chains is when the lengths of the two kinematic chains are the same. This relation assures the maximum workspace and the best kinematic properties (see **Table 4.2** column B). If $L_{KCh_1} > L_{KCh_2}$ or $L_{KCh_1} < L_{KCh2}$ holes in the workspace appear, which is not desired (see **Table 4.2** columns A and C). Accordingly, the constraint 4 for the *RPaPa* ANTHRO robot is defined as:

$$L_{KCh_1} = L_{KCh_2}$$

(51)

Table 4.2 Link lengths ratio influence on workspace for a general serial anthropomorphic robot

| (A) $\boldsymbol{L_{KCh_1} > L_{KCh2}}$	(B) $\boldsymbol{L_{KCh_1} = L_{KCh2}}$	(C) $\boldsymbol{L_{KCh_1} < L_{KCh2}}$
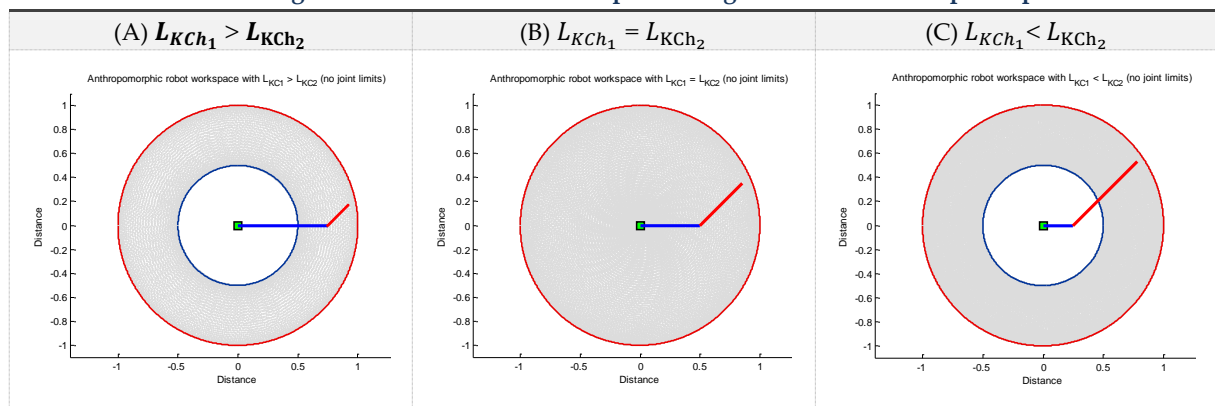
- ▪ *RPaR* ORTH constraint 4 definition*:*

For the *RPaR* ORTH manipulator it is useful to refer to the recent studies of the serial orthogonal family discussed in **Section 1.2.2**. First of all, it can be noticed that the general structure of

the *RPaR* ORTH under study would be equivalent to a *Type E* serial orthogonal manipulator (because L_{KCh_1} and L_{KCh_2} are different from zero, and the other mDH parameters are equal to zero). This kinematic structure is selected because a *Type E* orthogonal manipulator has a well-connected workspace (see **Figure 1.6**). Furthermore, the cross-section of the workspace of this manipulator changes according to the ratio between the length of the links of the first and the second kinematic chains as shown in **Table 4.3** (for $L_{KCh1} > L_{KCh2}$, $L_{KCh1} = L_{KCh2}$, and $L_{KCh1} < L_{KCh2}$).

Table 4.3. Link lengths ratio influence on workspace for the *RPaR* ORTH robot

(A) $L_{KCh1} > L_{KCh2}$	(B) $L_{KCh1} = L_{KCh2}$	(C) $L_{KCh1} < L_{KCh2}$
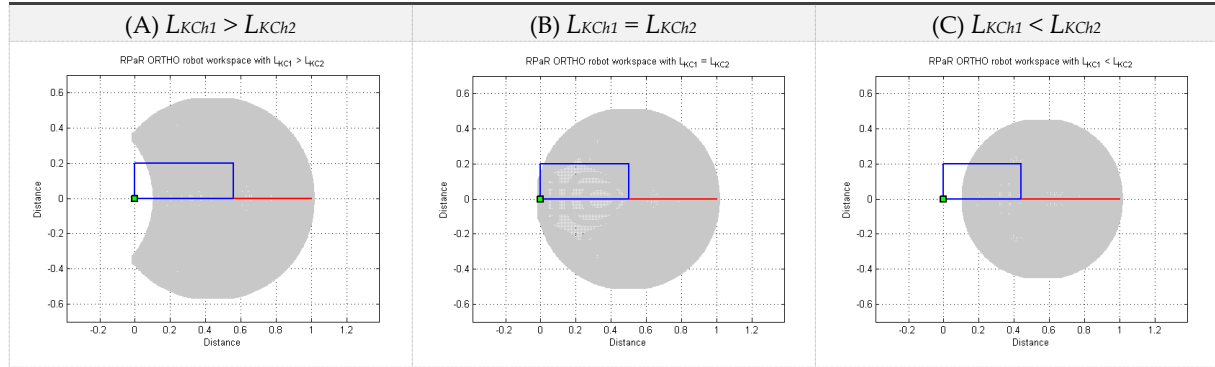
In addition, the kinematic properties of this manipulator also change depending on the relationship between L_{KCh1} and L_{KCh2}. Zein [**22**] provides a systematic way to select the best ratio between the link lengths of the manipulator based on the RDW and the kinematic performance index η, (see **Section 1.3.1**). In this work it is concluded that for orthogonal manipulators of *Type E* the best ratio between L_{KCh1} and L_{KCh2} is when $L_{KCh1} \approx L_{KCh2}$ since this relationship provides the highest values of the index η. Accordingly, the constraint 4 for the *RPaR* ORTH robot is defined as:

$$L_{KCh1} = L_{KCh2} \tag{52}$$

4.2.4   Definition of the elastostatic constraints

The elastostatic constraints are the key component of the design optimization procedure. As it was discussed, the stiffness matrix of the manipulator is selected as the desired mechanical property for the optimization. Moreover, a maximum allowable displacement of the end-effector caused by a static load, for all the points of a desired working area, is selected as the measure to be implemented as the constraint.

The maximum allowable displacement would depend on the application for which the manipulator is intended for. For instance, for certain applications linear and rotational displacements would be critical, meanwhile for other ones only linear displacements in the vertical axis may be important. For this research work, the optimization procedure is applied as a means to find the masses of the manipulators to compare them in terms of dynamic performance. Therefore, the actual value of the maximum allowable displacement is not the most important issue to define as long as it is reasonable and it is the same for both manipulators. For example, if a pick-and-place task is considered (which is a common application of this kind of manipulators); then, a maximum allowable linear displacement of 1 millimeter in each of the three axes can be considered. Consequently, these constraints can be defined as shown next.

Elastostatic Constraints: The linear displacement of the end-effector along the three axes caused by a static load (3D force and torque) should be smaller than the maximum allowable displacement for all the points of the desired working area (WA), as follows:

$$\delta \boldsymbol{p}(f) < \delta p_{max}$$

for all points in *WA*, for ANTHRO and ORTH

(53)

where:

$\delta \boldsymbol{p}(f) = \begin{bmatrix} \delta p_x & \delta p_y & \delta p_z \end{bmatrix}^T$ is the maximum allowable displacement along x, y and z-axis,

$f = \begin{bmatrix} F_x & F_y & F_z & \tau_x & \tau_y & \tau_z \end{bmatrix}^T$ is the vector of forces and torques at the end-effector, and

$\delta p_{max} = 1mm$ is the maximum allowable displacement along the x, y and z-axis.

The desired working area and the static load definition are explained in the following paragraphs.

Desired working area definition

The desired working area is a defined space inside the workspace of the manipulator which follows the characteristics and design constrains pursued. In this case, it would mean that the linear deformations due to the static load inside the desired working area are between the limits. Different working areas may be defined according to the application requirements, if these are known. Otherwise, areas of common geometric shape (e.g. square, rectangular, circular, etc.) are normally used. For the purpose of this research work a unitary square working area (size: 1m x 1m) is implemented. Consequently, the optimization procedure must reject all the solutions with a working area smaller than the desired one. This requirement can be modeled as a constraint as shown next.

Working Area Constraint: The size of the working area obtained for the current solution ($WA_{current}$) must be larger than or equal the desired working area (WA) set by the designer, and must cover the area demanded by the application requirement or follow the desired geometric shape: otherwise the solution is neglected:

$$WA_{current} \geq WA$$

where $shape(WA_{current})$ follows the desired application or geometric shape, for ANTHRO and ORTH

(54)

Selection of the static load

The static load f at the end-effector is selected taking again into consideration a pick-and-place application. For instance, if the manipulators were to be used in an automotive assembly plant to move a EURO-2 pallet containing the engine-transmission subassembly for a B-segment car (subcompact car) with the dimensions and weights as shown in **Figure 4.1**, from mechanics it is straightforward to show that making summation of forces along each axis, considering static equilibrium, leads to: $F_x = F_y = 0\,N$, and $F_z = 2060\,N$. Similarly, performing summation of moments around each axis, leads to: $\tau_x = (0.2\,m) \cdot F_1 = 250\,Nm$, $\tau_y = (0.4\,m) \cdot F_1 - (0.15\,m) \cdot F_2 = 400\,Nm$, and $\tau_z = 0\,Nm$. Hence, the static load vector at the end effector can be formulated as follows:

$$f = \begin{bmatrix} F_x & F_y & F_z & \tau_x & \tau_y & \tau_z \end{bmatrix}$$
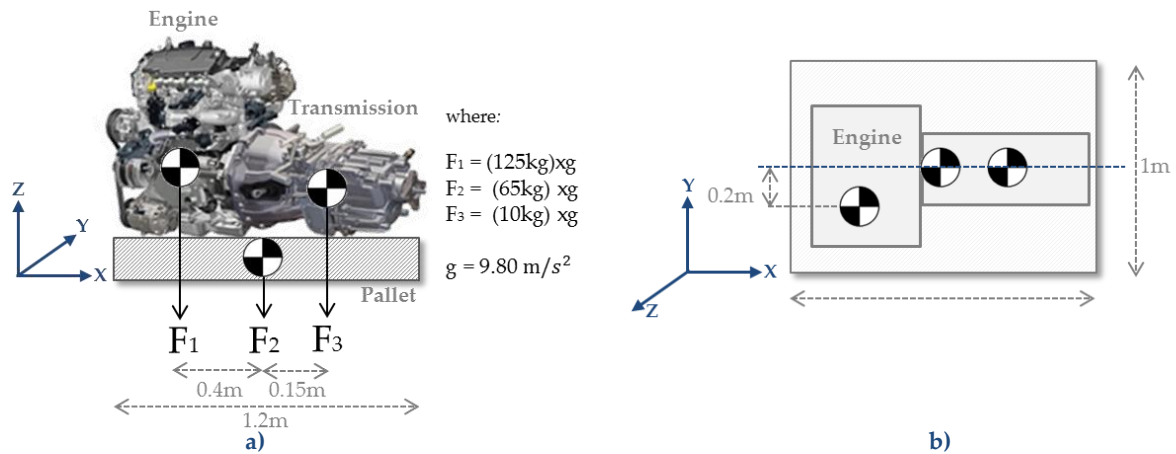
(55)

Figure 4.1 Static load definition (pick-and-place application) a) XZ view, b) XY view

4.2.5 Implementation technicalities

The implementation of the design-parameter optimization is done in MATLAB using the function *fmincon*, which finds the minimum of a constrained nonlinear multivariable function[3], where: the optimization criteria is the function to minimize, constraints 1, 2 and 4 can be considered as linear equalities/inequalities constraints, constraint 3 represents the upper and lower bounds of the optimization variables, and the elastostatic and working area constraints are implemented as non-linear inequalities constraints.

In addition, the testing of the workspace for the design constraints is done by discretizing it in small squares of constant side length (for example 5 cm). Then, the center of each square is checked to see if it fulfills the constraints. After that, the square analyzed is classified in a binary matrix as a square that complied with the design constraints (therefore getting a value of 1) or as a square that did not comply (value 0). From the resulting binary matrix, it is easy to implement an algorithm using dynamic programming to find the largest square matrix inscribed in the manipulator workspace which fulfills the design constraints. If the size of the square matrix found is smaller than the size of the desired working area, then the solution is discarded.

Finally, as stated in **Section 3.3.1**, the links are modeled as beams of circular cross-section. Moreover, the material chosen for modeling the links is aluminum since this material is also used in the industrial applications of similar manipulators. The material properties are shown next:
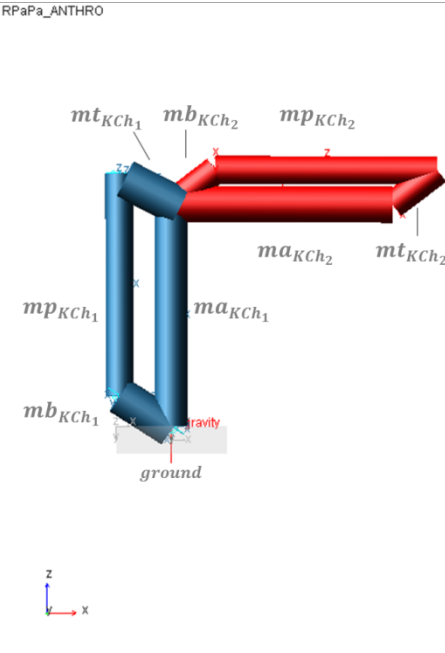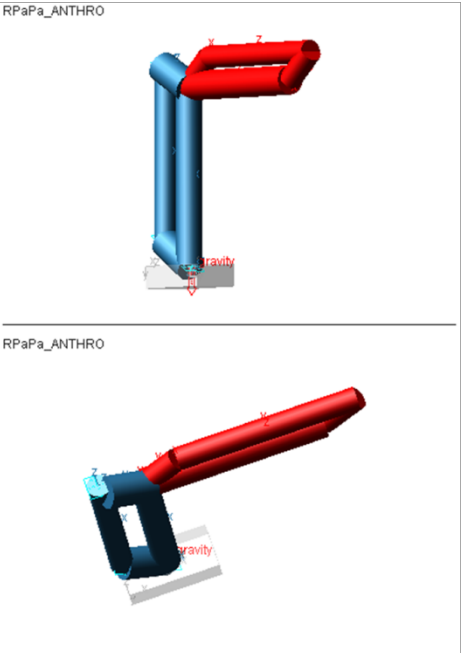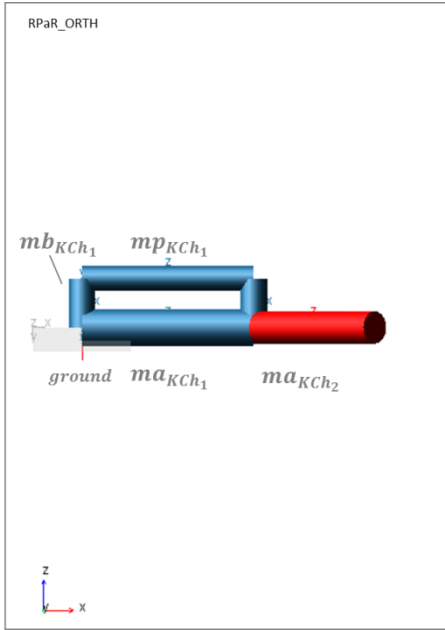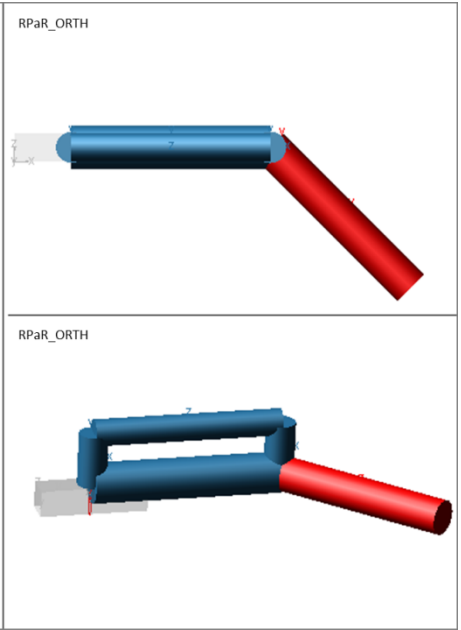
- density: $\rho = 2700 \ kg/m^3$
- Young's Modulus: $E = 67.5 x 10^9 Pa$
- Poisson's ratio: $v = 0.34$
- shear Modulus: $G = \frac{E}{2(1+v)}$

After completely defining the optimization problem, the next section has the objective of presenting the obtained results of the parameter optimization procedure.

---

[3] For more information about Matlab and fmincon, the interested reader may refer to: www.mathworks.com/products/matlab/

4.3   Parameter optimization results and conclusions

The parameter optimization procedure was run successfully for both manipulators. A summary of the results for each robot is presented next. Firstly, a CAD drawing representing the optimized manipulators is shown in **Table 4.4**.

<table>
<tr><td colspan="2" align="center">**Table 4.4 Optimized Robots (CAD drawing)**</td></tr>
<tr><td colspan="2" align="center">*RPaPa* ANTHRO Optimized Manipulator</td></tr>
<tr><td colspan="2"></td></tr>
<tr><td colspan="2" align="center">*RPaR* ORTH Optimized Manipulator</td></tr>
<tr><td colspan="2"></td></tr>
</table>

The mass of the links and total mass found for both optimized manipulators can be observed in **Table 4.5.**

Table 4.5 Mass of the robot

RPaPa ANTHRO		*RPaPa* ORTH	
Link Masses		**Link Masses**	
For the 1ˢᵗ *KCh*:	For the 2ⁿᵈ *KCh*:	For the 1ˢᵗ *KCh*:	For the 2ⁿᵈ *KCh*:
ma_{KCh_1} = 24.0 kg	ma_{KCh_2} = 29.5 kg	ma_{KCh_1} = 34.1 kg	ma_{KCh_2} = 25.6 kg
mp_{KCh_1} = 17.5 kg	mp_{KCh_2} = 17.5 kg	mp_{KCh_1} = 14.8 kg	
mb_{KCh_1} = 8.6 kg	mb_{KCh_2} = 4.4 kg	mb_{KCh_1} = 5.1 kg	
mt_{KCh_1} = 9.4 kg	mt_{KCh_2} = 4.4 kg	mt_{KCh_1} = 5.3 kg	
Total Mass		**Total Mass**	
115 kg		**85 kg**	

Similarly, the optimized variables (the length and radius of the links) that were obtained for both manipulators can be seen in **Table 4.6**.
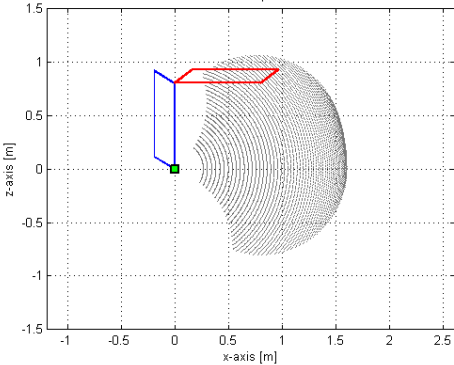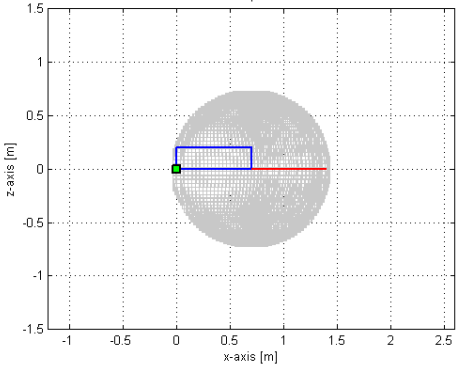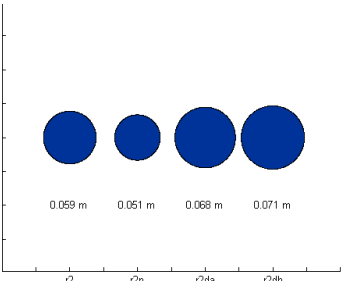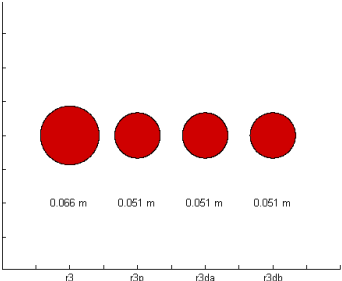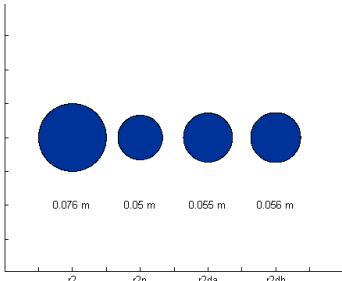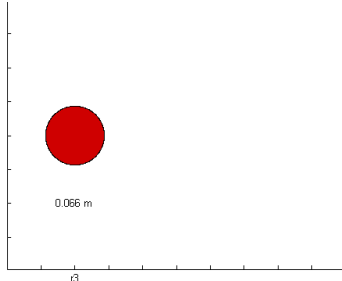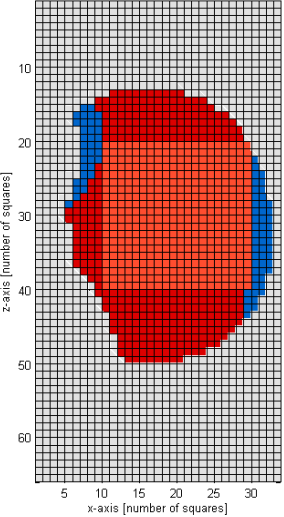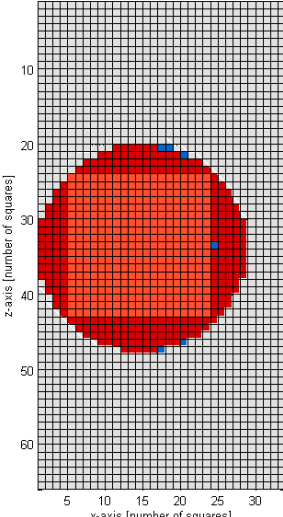
Table 4.6 Optimized variables

RPaPa ANTHRO		*RPaR* ORTH	
Optimized Link Lengths		**Optimized Link Lengths**	
For the 1ˢᵗ *KCh*:	For the 2ⁿᵈ *KCh*:	For the 1ˢᵗ *KCh*:	For the 2ⁿᵈ *KCh*:
L_{KCh_1} = 0.802 m	L_{KCh_2} = 0.802 m	L_{KCh_1} = 0.700 m	L_{KCh_2} = 0.700 m
L_{Base_1} = 0.221 m	L_{Base_2} = 0.203 m	L_{Base_1} = 0.200 m	
Optimized Link Radius		**Optimized Link Radius**	
For the 1ˢᵗ *KCh*:	For the 2ⁿᵈ *KCh*:	For the 1ˢᵗ *KCh*:	For the 2ⁿᵈ *KCh*:
Ra_{KCh_1} = 0.059 m	Ra_{KCh_2} = 0.066 m	Ra_{KCh_1} = 0.075 m	Ra_{KCh_2} = 0.065 m
Rp_{KCh_1} = 0.051 m	Rp_{KCh_2} = 0.051 m	Rp_{KCh_1} = 0.050 m	
Rb_{KCh_1} = 0.068 m	Rb_{KCh_2} = 0.051 m	Rb_{KCh_1} = 0.055 m	
Rt_{KCh_1} = 0.071 m	Rt_{KCh_2} = 0.051 m	Rt_{KCh_1} = 0.056 m	

Lastly a visual summary of the results is presented in **Table 4.7** to easily compare both manipulators. In this table, the workspace of the optimized manipulators is shown first. Then, the cross-section circular area of the beams (all at the same scale) is presented. Finally, the discretized workspace of both manipulators is displayed. In the discretized workspace plots, the points of the grid that belong to the workspace but do not follow the elastostatic constraints are shown in blue color, the points that belong to the workspace and follow the constraints are in red color, and the points of the workspace that follow the constraints and belong to the desired working area in light red color. Please note that all the plots of this table are at the same scale with its corresponding counterpart.

It is important to notice that the total mass of the *RPaR* ORTH manipulator resulted of a smaller value than the one of the *RPaPa* ANTHRO robot. This is expected since the second kinematic chain of the *RPaR* ORTH robot consists of only one link, instead of four links of a parallelogram as for the *RPaPa* ANTHRO manipulator. This is a good preliminary result in the dynamic comparison since the dynamics of a mechanism are highly related to its mass.

Table 4.7 Optimized manipulators summary

RPaPa ANTHRO	*RPaPa* ORTH

Optimized Robots

Optimized Link Radius

For the 1st *KCh*:	For the 1st *KCh*:

RPaPa ANTHRO 1st Kinematic Chain Cross-sectional Areas

0.059 m   0.051 m   0.068 m   0.071 m

r2   r2p   r2da   r2db

RPaPa ORTHO 1st Kinematic Chain Cross-sectional Areas

0.076 m   0.05 m   0.055 m   0.056 m

r2   r2p   r2da   r2db

For the 2nd *KCh*:	For the 2nd *KCh*:

RPaPa ANTHRO 2nd Kinematic Chain Cross-sectional Areas

0.066 m   0.051 m   0.051 m   0.051 m

r3   r3p   r3da   r3db

RPaPa ORTHO 2nd Kinematic Chain Cross-sectional Areas

0.066 m

r3

Workspace with desired working area
(**Blue** – inside workspace, **Red** –follows elastostatic constraints, **Light Red -** desired working area)

RPaPa ANTHRO Discretized Workspace with WA
(1 square = 5 cm)

RPaPa ORTHO Discretized Workspace with WA
(1 square = 5 cm)

43

Chapter 5

Dynamic Performance Analysis

The final chapter of the thesis report has the goal of presenting the main objective searched by this research work: the dynamic performance analysis and comparison between the *RPaR* orthogonal and *RPaPa* anthropomorphic manipulators. As mentioned in **Chapter 2**, the method proposed in [**1**] and [**2**] is followed to perform the dynamic performance analysis of the robot manipulators. Therefore, this chapter is organized as follows. Firstly, some important concepts of dynamic performance of robots are presented in **Section 5.1**. Then, **Section 5.2** presents the dynamic model of the manipulators which has to be has to be developed in order to perform the dynamic performance analysis. Moreover, a static torque analysis, as the one done in [**1**] and [**2**], is done in order to get some insight of the performance of the manipulators. The results and conclusions from this analysis are shown in **Section 5.3**. Furthermore, the dynamic performance analysis methodology and its application to the robots under study is discussed in **Section 5.4**. Finally, the results and general conclusions of the dynamic performance analysis and comparison between the two manipulators are presented in **Section 5.5**.

5.1   Dynamic performance of robot manipulators

In engineering terms, performance is a quantification of the effectiveness and the efficiency of an activity [**23**]. As this research work deals with manipulators, it is necessary to first define the concept of performance in the robotic field; which is done in **Section 5.1.1**. Moreover, since this thesis focuses on dynamic performance, this concept has to be further discussed. In addition, performance has to be measured within a referential; in other words, it is a concept that originates from a difference and cannot be described without establishing a scale [**23**]. Hence, **Section 5.1.2** discusses the concept of dynamic performance in robots and introduces some of the common indices used to evaluate dynamic performance in robots.

5.1.1   Performance in robots

The performance of a robot is of interest for research because of many reasons. For instance, it may be helpful to: set the parameters of the robot (in the design stage) or assist in the selection among various robots for a specific task, among others [**7**]. There exist several performance measures that can be divided in two main groups [**24**]: the measures based on geometric, kinematic and kinetostatic properties; and the measures based on dynamic, elastostatic and elastodynamic properties.

The first ones are based on simple models that evaluate the geometric, kinematic and kinetostatic properties of a mechanism; thus they only use the primary geometric parameters of the

mechanism, such as the length of links. Some examples of these indices are the condition number, the dexterity and the manipulability. The second ones use more complicated models (like the dynamic model) and they use both the primary and the secondary geometric parameters of the mechanisms, such as the cross-section of the links. The most common indices are based on the total mass of the robot, the input efforts, and the maximal deformations. Since this thesis focus on the dynamic performance, the next section discusses briefly this concept and introduces some of the indices to quantify dynamic performance found in the literature.

5.1.2 Dynamic performance in robots

As mentioned in **Section 1.5**, the dynamic performance of robots is an open issue of research because robots are complex mechanical systems. There are several indices of dynamic performance that have been treated in the literature, some of these indices are briefly presented next

Dynamic performance indices: literature overview

Generalized Inertial Ellipsoid (GIE): It is a tool to measure the ability of changing end-effector's velocities in different directions for a fixed value of kinetic energy [2]. It originates from extending the inertia ellipsoid concept (used in mechanics) to a generalized ellipsoid for a series of rigid bodies like a robot.

Dynamic Manipulability Ellipsoid (DME): This index was introduced by Yoshikawa (quoted in [2]) for measuring the ease of changing the end-effector's configuration by a set of joint torques with fixed magnitude. It analyses the ability of the end-effector to perform accelerations along each task-space direction for a given set of joint torques [25].

Acceleration Radius: Defined by Graettinger and Krogh (quoted in [2] and [25]) this index defines a uniform lower bound on the magnitude of the end-effector acceleration (or alternatively at the joints) that can be achieved over the whole workspace of the robot from any state in the operating region, for given bounds of joint torques.

Inertia Matching Ellipsoid (IME): It was proposed as a dynamic performance index by Kurazume and Hagegawa [26]. It has the objective of relating the dynamic torque-force transmission efficiency from joint actuators to a load carried by the end-effector. The IME is a generalization of the DME and the manipulating-force ellipsoid (MFE, which evaluates the static torque-force transmission from the joints to the end-effector). Thus, both the DME and MFE can be derived from the IME.

Efficient Working Areas (EWA): The objective of this index is to calculate the relation of the maximum values of the dynamic elements (maximum accelerations, torques and reaction forces) depending on the speed of the end-effector and the payload [2], as shown in eq. **(56)**. These relations will define some areas in which the robot will work more efficiently and the system is not expected to exceed the limits defined by them. The areas are called EWA.

$$\ddot{q}_{max}, \Gamma_{max}, F_{R_{max}} = g(Vs, Mp) \tag{56}$$

where:

$\ddot{q}_{max}, \Gamma_{max}$ and $F_{R_{max}}$ are the maximum values of the dynamic elements,

Vs is the velocity of the end-effector, and Mp is the payload.

The motivation of this index is to set a measurement to analyze the relationship of the speed of the end-effector and the payload and their influence on the dynamic elements of the robot. An example can be seen in in **Figure 5.1** and eq. (**57**). For details on the calculation of this index, the interested reader may refer to [2].
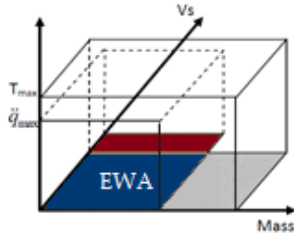

$$EWA_{acce} = \ddot{q}_{max} \cdot Vs \cdot Mp$$
$$EWA_{torq} = \Gamma_{max} \cdot Vs \cdot Mp \qquad (57)$$
$$EWA_{forc} = F_{R_{max}} \cdot Vs \cdot Mp$$

Figure 5.1 Example of EWA region (from [1])

5.2   Dynamic model of the manipulators

The dynamic model of a robot establishes the relationship between the internal and external forces of the system, and the acceleration of its joints; and it is used in the mechanical design, control, simulation, and performance analysis of robots, among others [5]. Furthermore, the inverse dynamic model (IDM) of a robot provides the joint torques and forces in terms of joint positions, velocities and accelerations, as shown in eq. (**58**); whereas, the direct dynamic model (DDM) calculates the joint accelerations in terms of the joint positions, velocities and torques, as shown in eq. (**59**).

$$\mathbf{\Gamma} = f(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, \boldsymbol{f_e}) \qquad (58)$$
$$\ddot{\boldsymbol{q}} = g(\boldsymbol{q}, \dot{\boldsymbol{q}}, \mathbf{\Gamma}, \boldsymbol{f_e}) \qquad (59)$$

where:

$\mathbf{\Gamma}$ is the vector of joint torques (or forces),

$\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}$ are the vector of joint positions, velocities and accelerations respectively, and

$\boldsymbol{f_e}$ is the vector of forces and moments exerted by the robot on the environment.

There are two common formulations of the dynamic model used in robotics; one is the Lagrangian formulation which expresses the model in terms of joint variables and their derivatives, and the other one is the Newton-Euler formulation which expresses the model using linear and rotational Cartesian velocities and accelerations [5].

It is important to recall that the *RPaPa* ANTHRO and *RPaR* ORTH manipulators can be classified as complex, closed-chain robots. Following the methodology of [5], the dynamic model of this kind of robots can be easily calculated as follows. Firstly, it is necessary to obtain the dynamic model of an equivalent tree-structure robot (by opening the close loops of the kinematic chains in one of the passive joints). Then, by multiplying the tree-structure model by a Jacobian matrix (representing the derivative of the tree structure variables with respect to the actuated variables) the closed-chain system is obtained.

Consequently, the general definition of the dynamic model for serial and tree-structure manipulators is discussed first (**Section 5.2.1**), followed by a discussion on obtaining the closed-chain model from the equivalent tree-structure (**Section 5.2.2**). Finally, some details regarding the actual implementation of the dynamic model of both robots are presented (**Section 5.2.3**).

5.2.1 Dynamic model for serial and tree-structure manipulators

According to the Lagrangian formulation, the general form of the dynamic equations of a rigid serial manipulator taking into account dissipative forces and the forces exerted on the environment by its end-effector can be written as [5]:

$$\Gamma = A(q)\ddot{q} + C(q,\dot{q})\dot{q} + Q(q) + \Gamma_f + \Gamma_e \tag{60}$$

where:

$A(q)$ is the symmetric and positive definite inertia matrix of the robot,

$C(q,\dot{q})\dot{q}$ is the vector of Coriolis and centrifugal torques,

$Q(q)$ is the vector of gravity torques,

Γ_e is the vector of torques due to external forces and moment exerted by the manipulator, and

Γ_f is the vector of friction torques.

The vector of external torques (Γ_e) is calculated according to the basic static equation that relates the generalized Jacobian of the robot (J_n^T) and the vector of externals forces and moments exert by the manipulator on the environment (f_e) as shown in [5]:

$$\Gamma_e = J_n^T f_e \tag{61}$$

Additionally, the friction is considered by applying the common model of Coulomb and viscous friction that is discussed in [5]. Thus, leading to the next vector of friction torques:

$$\Gamma_f = diag(\dot{q})\,F_v + diag\big(sign(\dot{q})\big)\,F_c \tag{62}$$

where:

F_v and F_c are the vector of Coulomb and viscous parameters of the joints respectively.

Finally, the dynamic model can be written in the next compact form:

$$\Gamma = A(q)\ddot{q} + H(q,\dot{q}) \tag{63}$$

where:

$H(q,\dot{q})$ groups the vector of Coriolis, centrifugal, gravity, external and friction torques.

On the other hand, the Newton-Euler formulation provides the torques as defined by eq. (58) but without explicitly calculating the matrices A, C and Q of eq. (60). This method represents an efficient and systematic approach for solving the dynamic model of a robot [5] and can be easily programed. The Newton-Euler algorithm presented in [5] is a practical and efficient form of this formulation and it is linear in the inertial parameters. It is based on forward and backward recursive computations.

- The forward recursive computations calculate the link velocities, accelerations and the total wrench (summation of forces and moments) from the first link to the terminal link (hence its name).

- The backward recursive computation calculates the reaction wrenches on the links from the terminal to the first one, which leads to the calculation of the joint torques.

The equations for the forward and backward recursive formulations and more details about this approach can be found in [**5**].

So far, the dynamic model has been discussed for the case of serial manipulators. Since the joint variables are independent in a tree-structure robot, the Lagrange formulation discussed before can be used in a similar way. Likewise, the Newton-Euler recursive equations can be also applied by just considering the proper antecedent of each link as explained in [**5**]. Once the equivalent tree-structure dynamic model has been obtained, it is possible to compute the closed-chain model as shown next.

5.2.2 Obtaining the closed-chain dynamic model

Recalling [5], the total variables of a closed-chain manipulator can be denoted as follows:

$$q = \begin{bmatrix} q_{tr} \\ q_c \end{bmatrix} = \begin{bmatrix} q_a \\ q_p \\ q_c \end{bmatrix} \tag{64}$$

where:

$q_{tr} = \begin{bmatrix} q_a & q_p \end{bmatrix}^T$ is the vector of joint variables of the equivalent tree-structure robot, and

q_c is the vector of cut joint variables.

The joint variables vectors for both manipulators are defined as previously shown in **Section 2.2.1** and **Section 2.2.2**. Moreover, q_p has to be found in terms of q_a. The relationship between them is found by solving the loop closure equations, or analytically if possible. This relationship is shown as well in the mentioned sections.

Similarly, the vector of torques of the equivalent tree-structure robot (Γ_{tr}), which can be computed from the previous definition of the dynamic model, can be written as:

$$\Gamma_{tr} = \begin{bmatrix} \Gamma_a \\ \Gamma_p \end{bmatrix} \tag{65}$$

where:

Γ_a and Γ_p are the vectors of torques of the actuated and passive joint variables of the equivalent tree-structure respectively.

Finally, the vector of torques of the actuated joint variables of the close-chain manipulator, denoted by τ, can be found in terms of the joint torques of the equivalent tree structure as follows:

$$\tau = G^T \Gamma_{tr} = \left(\frac{\delta q_{tr}}{\delta q_a} \right)^T \Gamma_{tr} \tag{66}$$

where:

G is the Jacobian matrix that represents the derivative of the tree structure variables with respect to the actuated variables (shown in the next section for each robot).

For the proof of these equations and more details about this methodology, the interested reader may refer to [3].

5.2.3 Implementation details of the dynamic model

Since the dynamic model is a recurring tool needed in robotics, there are several software programs that have been developed to deal with its calculation. For the purposes of this thesis, the software package SYMORO+[4] is used to compute the dynamic model of the robots under study. SYMORO+ provides the analytical dynamic model of the robot according to the Newton-Euler formulation. It follows the approach and notation used in [3] and throughout this document. Moreover it provides a practical and efficient form of this formulation. Hence, it has a low computational cost making it suitable for using it in time-consuming tasks such as the optimization process of this research work.

To obtain the equivalent tree-structure dynamic model by SYMORO+ it is just needed to enter the mDH parameters for each robot (shown in **Section 2.2.1** and **Section 2.2.2**). Similarly to the previous research work, a point mass is included in the model to simulate a load at the end-effector. To include this point mass it is just necessary to add a new row to the table of the mDH parameters, which represents the constant transformation from the last frame to the frame of the end-effector (with $\sigma = 2$ and the values of the remaining mDH parameters set according to the position of the frame); and considering the inertia parameters of the corresponding link as zero except for the mass.

As explained in the previous section, the closed-chain active torques can be obtained from the equivalent tree structure torques following eq. (**66**). From the relationships between the active and passive variables shown in **Section 2.2.1** and **Section 2.2.2**, the Jacobian matrix $\boldsymbol{G^T}$ for each robot (G_{ANTHRO}^T and G_{ORTH}^T) can be easily obtained as seen in eq. (**67**) and eq. (**68**).

$$G_{ANTHRO}^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 1 \end{bmatrix} \tag{67}$$

$$G_{ORTH}^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{68}$$

Once the analytical model is completely developed, it is then implemented in MATLAB numerically. The values of the mass and the dimensions of the links of the manipulators are obtained from the parameter-design optimization procedure developed in the previous chapter. With these values and basic mechanic equations, the inertia parameters of the links are calculated.

In addition, the model developed is validated as follows: Firstly, the manipulators are modeled in the ADAMS[5] software; then, the joint torques are calculated by this program for certain trajectory functions of the active joints. Lastly, the same trajectory functions are applied to the dynamic model developed in SYMORO+/MATLAB and the torques are computed. It is important to mention that both approaches were compared obtaining the same results; thus validating the model.

With the dynamic model of both manipulators fully developed the next step of the methodology is to perform a static torque analysis, which is discussed in the next section.

---

[4] SYMORO+ is a software package for the automatic symbolic modeling of robots developed under MATHEMATICA and C language by Khalil and Creusot from École Centrale de Nantes (www.irccyn.ec-nantes.fr/spip.php?article600&lang=en)
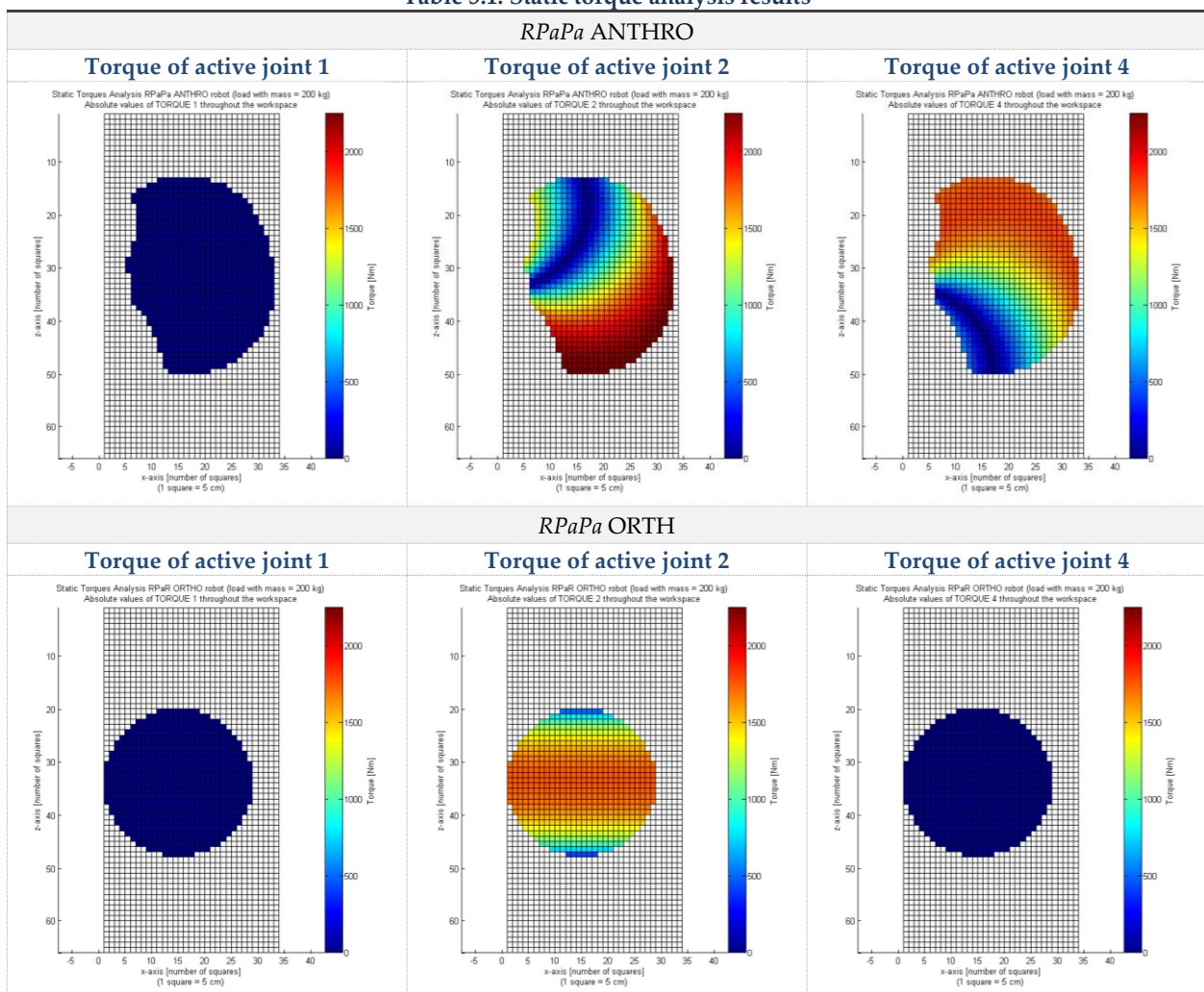
[5] MSC. ADAMS is a known multibody dynamics and motion analysis software developed by MSC Software (see http://www.mscsoftware.com/Products/CAE-Tools/Adams.aspx)

5.3 Static Torque Analysis

As discussed in the introductory chapters, the static torque analysis is a special case of the dynamic analysis in which the torques are computed for the cross-section of the workspace using the IDM but with the joint velocities and accelerations equal to zero. Hence, the joint torques depend only on the robot configuration. The static torque analysis is performed for both robots (for a load of 200 kg and for the same configurations as the ones explained in **Chapter 2**). The results of the analysis are shown in **Table 5.1**, where the absolute value of the torques of the active joints are plotted.

This analysis led to obtaining really interesting and useful conclusions. As it is seen in the presented results, the torque of joint 1 for both manipulators is of $0\ Nm$ for all the workspace cross-section. For the other two active joints several differences are noticeable. In the case of the *RPaPa* ANTHRO manipulator, it is observed that the absolute torque values for the remaining active joints vary significantly throughout the workspace, ranging from $0\ Nm$ to more than $2000\ Nm$. On the other hand, in the case of the *RPaR* ORTH manipulator, it is seen that the absolute torque values of the joint 4 are $0\ Nm$; and for the joint 2, they vary from $0\ Nm$ to approximately $1800\ Nm$. Moreover, the torques of this joint are distributed symmetrically through the workspace which can be considered as a good property.

Table 5.1. Static torque analysis results

RPaPa ANTHRO — Torque of active joint 1 | Torque of active joint 2 | Torque of active joint 4

RPaPa ORTH — Torque of active joint 1 | Torque of active joint 2 | Torque of active joint 4

Therefore, it can be concluded that: **the *RPaR* orthogonal manipulator has a better static torque performance than its counterpart, the *RPaPa* anthropomorphic manipulators** because of the following reasons:

- The joint 2 torque profile is symmetrically distributed for the *RPaR* ORTH robot ranging from $0\ Nm$ to a maximum of $1800\ Nm$, meanwhile for the *RPaPa* ANTHRO robot, it is not equally distributed and it reaches more than $2000\ Nm$.

- The joint 4 torque absolute value for the *RPaR* ORTH robot for all the cross-section of the workspace is of $0\ Nm$, whereas the *RPaPa* ANTHRO robot presents a profile with not equally distributed torques that can reach values of $1800\ Nm$.

Having discussed the static torque analysis it is possible to present the dynamic performance analysis of the two manipulators.

5.4　Dynamic performance analysis:
Exciting trajectory optimization for finding the maximum input torques

The indices commonly used to evaluate the dynamic performance of manipulators found in the literature present the disadvantages that they are difficult to understand for engineers and that straightforward conclusions are difficult to obtain. Thus, to analyze the dynamic performance analysis of the *RPaPa* ANTHRO and *RPaR* ORTH robots a similar methodology as the one of previous research work ([1] and [2]) is applied.

As previously discussed, the methodology is based on the analysis of the robots maximum input torques and accelerations. However, these indices depend on the trajectory of the end-effector. Thus, the main idea is to find some exciting trajectories via an optimization procedure to make the manipulators display the highest values of the input torques. Furthermore, to do the comparison more robust, the optimizing can be done first to one of the manipulators and then applied the trajectory obtained to the other manipulator; and vice versa. Finally, both manipulators can be easily compared, for example, by plotting the resulting joint torques and deciding which one shows a better performance.

In **Section 5.4.1** some important concepts and general assumptions of the methodology are presented. Furthermore, **Section 5.4.2** formally defines the exciting trajectory optimization procedure for finding the maximum input torques. Lastly, several implementation details are explained in **Section 5.4.3**.

5.4.1　Generalities of the trajectory optimization procedure

Before discussing the trajectory optimization procedure, it is important to define the concept of trajectory and to explain some general assumptions that are followed.

Definition of trajectory

A trajectory is composed of several segments which are determined by a subset of m optimized points $(P_1, P_2, \cdots P_m)$ as shown in **Figure 5.2**. There are two ways of constructing a trajectory from this subset of points: *point-to-point* and *via-points* method. In the first one, the trajectory is made from several segments $(P_1 \!-\! P_2, \cdots P_{m-1} \!-\! P_m)$. The manipulator starts at the beginning of each segment and stops at its end. Each segment k is defined by a motion profile curve such as a 5th order polynomial, a trapezoidal profile, etc. In the second one, the trajectory is built by using a continuous spline (for example cubic spline) which is defined by the subset of points $(P_1 \!-\! P_2 \!-\! \cdots \!-\! P_m)$. In this case, the manipulator starts at P_1 and stops at the last point.
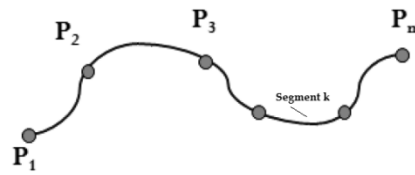

Figure 5.2 Trajectory formed of optimized points (adapted from [1])

General assumptions of the trajectory optimization procedure

Before implementing the optimization procedure some general assumptions must be defined, such as the following ones:

- *The analysis is performed in the task space:* The task space was preferred because the number of variables in the optimization procedure for the trajectory remains invariant regarding the number of DOF of the manipulator (which will not be the case for the joint space). Moreover, if the trajectory is obtained in the task space, it can be later easily applied to the second manipulator.

- *The payload at the end-effector must be defined:* Similarly to the static torque analysis a load at the end-effector is considered. As before, the load is modeled as a mass-point with m = 200kg and the rest of the dynamic parameteres set to zero.

- *The controller of the manipulator is considered as ideal:* This means that it keeps track of the trajectory perfectly; thus, the dynamic models are calculated with the desired joint positions velocities and accelerations.

- *The dynamic model does not consider the friction terms:* No Coulomb neither viscous friction are considered in the calculation of the IDM.

- *The maximum operational velocities in the joint and/or task space must be defined:* As part of the constraints of the optimization procedure, the maximum velocity of the end-effector in the task space and/or the maximum velocity of the joint actuators should be included. For instance, for the manipulators under study, the maximum velocities of the joints of the commercial *RPaPa* Kuka robot (figure 2) were selected as reference. Thus: $\dot{q}_{j_{max}} = 105$ deg/s.

- *The travelling time between two points (h) in the task space can be known or can be considered as an optimization variable*: In the previous research work the time between segments was consider as known. In this research work the traveling time is implemented as another optimization variable to provide more flexibility to the optimization process. Thus, in this case the traveling

time between two points must respect some lower and upper limits (set by the designer) and the total traveling time of the entire trajectory (sum of all the traveling times) can be constrained as well.

▪ *The posture of the manipulators is not allowed to change:* The posture of the manipulator is kept the same for the entire trajectory. As explained before, in the case of the *RPaPa* ANTHRO manipulator there is only one posture available due to the joint limits. For the *RPaPa* ORTH robot, the same posture as the one used for the design parameter optimization and static torque analysis is used.

▪ *The trajectory should be inside the workspace of the robot:* It is obvious that the obtained trajectory should be inside the workspace of the manipulator, which can be verified by directly checking in the task-space or by checking that the joint variables are inside the joint limits for all the points.

With this assumptions explained, it is possible to define the exciting trajectory optimization procedure.

5.4.2 Exciting trajectory optimization procedure definition

The generation of exciting trajectories can be defined as an optimization procedure as follows:

Objective: obtain the maximum values of the torques

$$max \sum_k |\Gamma_{jk}| \quad for\ j = 1, 2, \cdots n \tag{69}$$

Subject to the constraints:

The max. abs. torque of each segment should be smaller than the joint torque limits:
$$abs(\Gamma_{jkmax}) \leq \Gamma_{jmax} \tag{70}$$

The velocity of the end-effector for each segment, $V(k)$, should be smaller than the max vel. allowed:
$$V(k) \leq V_{max} \tag{71}$$

The joint velocities for each segment, $\dot{q}_j(k)$, should be smaller than the max vel. allowed:
$$\dot{q}_j(k) \leq \dot{q}_{max} \tag{72}$$

The end-effector position P should be inside the workspace of the manipulator:
$$P \in W \tag{73}$$

Or: the joint positions for all the trajectory should be inside the joint limits of the robot:
$$q_{j_{min}} \leq q_j \leq q_{j_{max}} \tag{74}$$

The traveling time of each segment (h_k) should be inside the traveling time limits set:
$$h_{k_{min}} \leq h_k \leq h_{k_{max}} \tag{75}$$

The total traveling time for the entire trajectory (t) should be inside the total time limit:
$$t_{min} \leq t \leq t_{max} \tag{76}$$

for: $j = 1, 2, \cdots n\ joints,\ k = 1, 2, \cdots m - 1\ segments, and\ m\ points$.

---

where:

Γ_{jk} is the torque of segment k of joint j,

Γ_{jkmax} is the maximum torque of segment k for joint j,

Γ_{jmax} is the maximum allowed torque of joint j,

V_{max} is the vector of maximal velocities of the end-effector,

$\dot{q}_{max}$ is the vector of maximal joint velocities,

P is the Cartesian position of the end-effector,

W is the workspace of the manipulator,

q_{min} and q_{max} are the vector of minimum and maximum joint limits respectively,

$h_{k\,min}$ and $h_{k\,max}$ are the minimum and maximum traveling time limits of each segment k,

$t = h_1 + h_2 + \cdots h_{m-1}$, is the total traveling time of the trajectory, and

t_{min} and t_{max} are the minimum and maximum total traveling times of the entire trajectory.

Since the joint torque values depend on the trajectory, the searching parameters **X** of this optimization procedure are: the *m* Cartesian points of the trajectory and the travel time *h* for each segment. Thus, the total number of searching parameters is: $N_X = 4_m - 1$; corresponding to *m-1* elements for the travelling times and *3m* elements corresponding to the x, y, and z coordinates of each Cartesian point of the trajectory. Running the optimization procedure leads to finding the trajectory that makes the joints to exhibit their largest values of input torques. Before presenting the results obtained, the next section explains some implementation details of this optimization procedure.

5.4.3 Implementation details

This section has the goal of discussing some of the specificities regarding the implementation of the exciting trajectory position explained before.

Trajectory applied

In this research work the trajectory optimization procedure is implemented for one trajectory type. The trajectory used for this research work is calculated with the *via-points* method using a continuous cubic spline function, which is a common technique to calculate trajectories found in the industrial applications [5]. The calculation of the cubic spline function is performed following the approach explained by Khalil [5] and implemented as well in [1] (with desired initial and final accelerations of the end-effector equal to zero, and iterating to find the proper values of the initial and final velocities). The algorithm details and equations to obtain the trajectory using this method are explained in detail in [5].

Optimization tool used

Similarly to the parameter optimization, the implementation of the exciting trajectory optimization procedure is done in MATLAB using the function *fmincon*, where the optimization criterion is the maximization of the joint torques for the exiting trajectory. Since the function *fmincon* is formulated only for minimization problems (as many optimization programs), then, the maximization problem has to be converted to a minimization one. From optimization theory it is known that this can be done straightforwardly by simply multiplying the maximization objective by minus one. Then, if the value of the optimization criteria is desired, the obtained value is multiplied by minus one again.

On the other hand, the constraints related with the traveling times of each segment can be implemented as linear inequalities constraints, and all the remaining ones can be implemented as non-linear equality/inequalities constraints.

Selection of the via-points for the trajectory

The selection of the via-points for the trajectory is important for the optimization procedure since the value of the joint torques may depend largely on them. The static torque analysis is a useful aid for better selecting the via-points. The idea is to select a number of m points from the workspace areas where the static torques resulted in high values. This is done for each manipulator. For example, for the optimization trajectory of the *RPaR* ORTH manipulator, the via points are selected in such a way the end-effector had to move through the center of the workspace since it is the area that presented the higher input torques for the second joint for the static torque analysis. It is important to mention that the number of via-points selected for the trajectories is of $m = 5$. This number is selected because in practice it provides a good trade-off between the complexity of the trajectory and the computational time needed for the optimization procedures.

Optimizing first for one robot, then applying the results to the other robot

Similar to the approach of [1], the optimization of the exciting trajectory is done for one manipulator to obtain the worst possible case of the joint torques for it (using its corresponding *via-points*). Then, the resulting optimized end-effector trajectory is applied to the other manipulator (adjusting the coordinates of the *via-points* according to the difference in the positions of the working areas with respect to the center of the robots; in this way both trajectories will occur in the same positions inside the WA for each manipulator). The results (joint torques and accelerations) are compared between both robots. Finally, the procedure is repeated but now starting with the opposite robot. This is done in order to make the comparison more robust and fairer. The results for some representative test cases of the dynamic performance analysis are shown in the next section.

5.5   Results and conclusions of dynamic performance analysis

Several test cases (with different via-points) are tested in both robots. The results obtained are consistent among the different test cases. Consequently, **Section 5.5.1** presents some results from two different representative test cases (one where the *RPaR* ORTH robot trajectory is optimized first and one where the *RPaPa* ANTHRO robot trajectory is optimized first). Finally, **Section 5.5.2** discusses the conclusions obtained from this analysis.

5.5.1   Test cases of the dynamic performance analysis

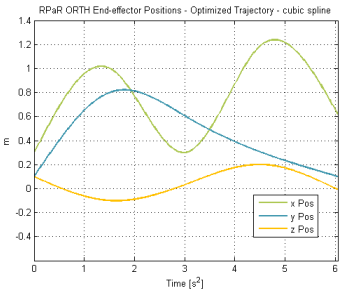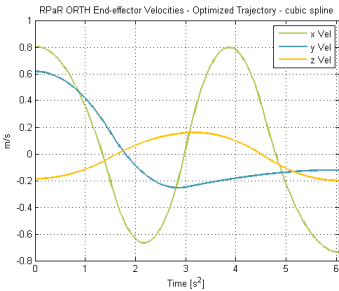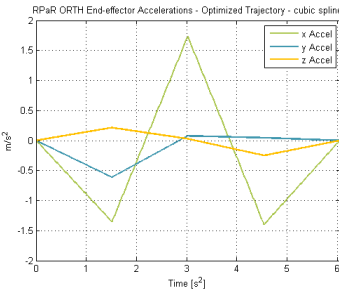**First test case:** *Optimizing for RPaR ORTH first and applying resulting trajectory to RPaPa ANTHRO*

For this test case the 5 *via-points* are selected in such a way to search for the worst possible case for the *RPaR* ORTH manipulator. Then, the obtained end-effector trajectory is applied to the other manipulator. After the procedure is run, the optimized variables (the *via-points* Cartesian coordinates

and the traveling time of each segment) and its corresponding end-effector trajectory are obtained as shown in **Table 5.2**.
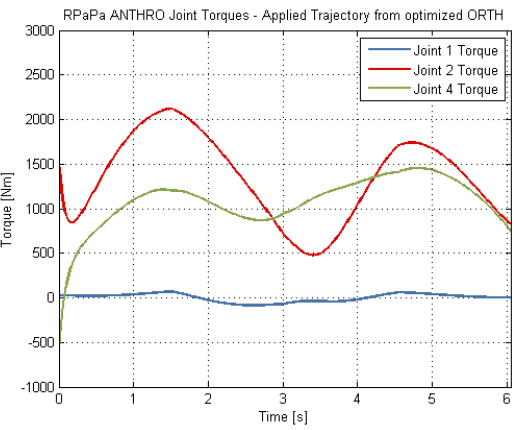
Table 5.2. Dynamic Performance Analysis Results: 1ˢᵗ case (*RPaR* ORTH first)		
Optimized variables (via-points pose and traveling segment times)		

Via-points coordinates:	**Traveling time of segment k:**
$x_{vp} = [\quad 0.30 \quad 1.00 \quad 0.30 \quad 1.20 \quad 0.62 \quad]$ m $\quad$ $y_{vp} = [\quad 0.10 \quad 0.80 \quad 0.60 \quad 0.30 \quad 0.10 \quad]$ m $\quad$ $z_{vp} = [\quad 0.10 \quad -0.10 \quad 0.036 \quad 0.20 \quad -0.008 \quad]$ m	$h_k = [\; 1.5111 \quad 1.5150 \quad 1.5168 \quad 1.5136\;]$ seconds

Corresponding optimized exciting trajectory of the end-effector		
End-effector positions	**End-effector velocities**	**End-effector accelerations**

The joint torque values corresponding to the trajectory of the end-effector for both manipulators are shown in **Table 5.3**.

Table 5.3. Dynamic performance analysis joint torques of 1ˢᵗ case	
Joint Torques (*RPaR* ORTH optimized first and results applied to *RPaPa* ANTHRO)	
RPaPa ANTHRO	**RPaR ORTH**

Finally the joint variables of both robots corresponding to the optimized trajectory are shown in **Table 5.4**.

Table 5.4. Dynamic performance analysis joint variables of 1ˢᵗ case (*RPaR* ORTH first)

RPaPa ANTHRO manipulator		
Joint positions	**Joint velocities**	**Joint accelerations**

RPaR ORTH manipulator		
Joint positions	**Joint velocities**	**Joint accelerations**

Second test case: *Optimizing for RPaPa ANTHRO first and applying resulting trajectory to RPaR ORTH*

Similarly, for this test case the 5 *via-points* are selected in such a way to search for the worst possible case for the *RPaPa* ANTHRO robot. Then, the obtained end-effector trajectory is applied to the other manipulator. After the procedure is run, the optimized variables and its corresponding end-effector trajectory were obtained as shown in **Table 5.5**.
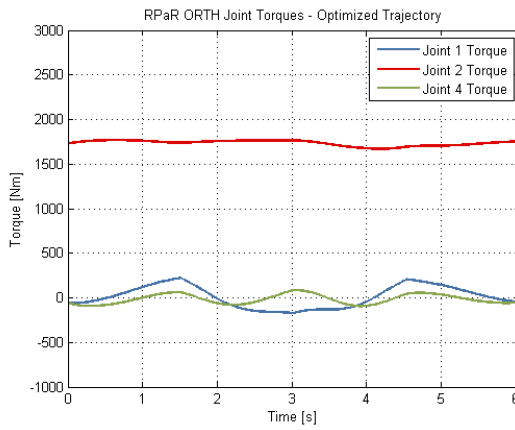
Table 5.5. Dynamic Performance Analysis Results: 2ⁿᵈ case (*RPaPa* ANTHRO first)

Optimized variables (via-points pose and traveling segment times)	
Via-points coordinates: $x_{vp} = [\; 0.60 \quad 0.70 \quad 1.00 \quad 0.60 \quad 1.10 \;]$ m $y_{vp} = [\; 0.10 \quad 0.80 \quad 0.60 \quad 0.50 \quad 0.10 \;]$ m $z_{vp} = [\; 0.60 -0.40 -0.10 -0.40 -0.40 \;]$ m	**Traveling time of segment k:** $h_k = [\; 1.415 \quad 1.415 \quad 1.415 \quad 1.415 \;]$ seconds

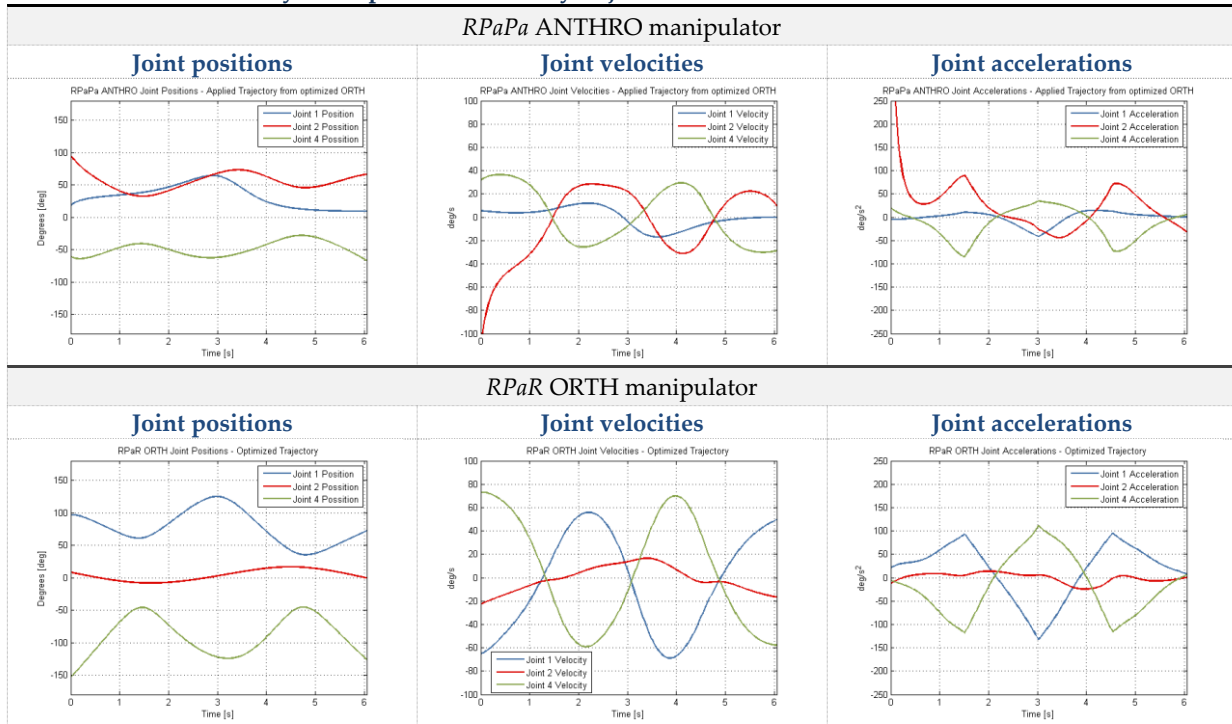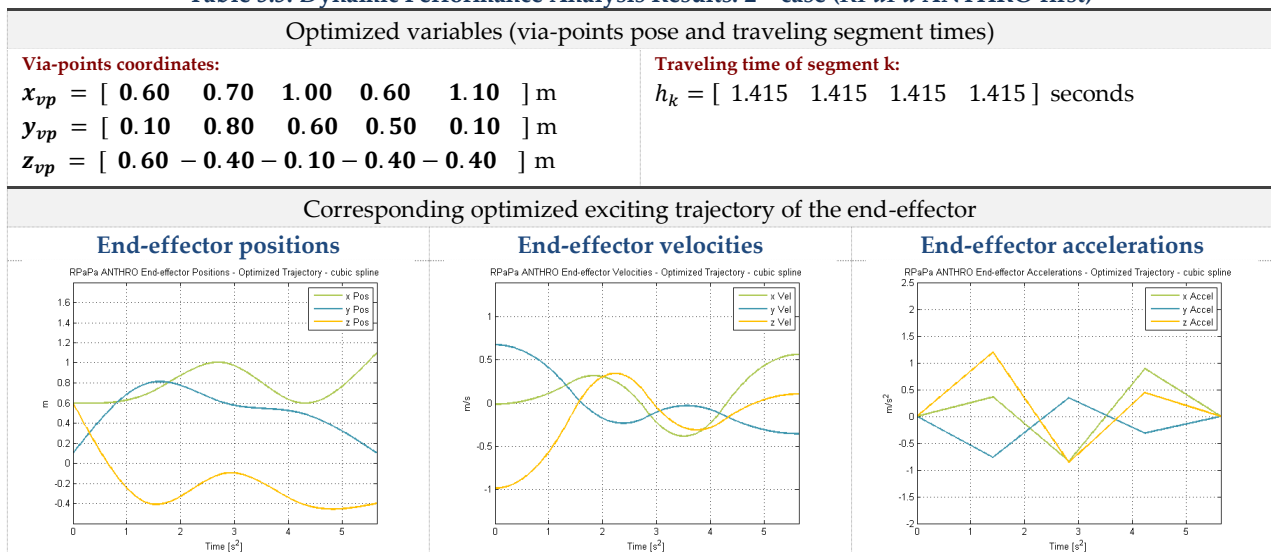| Corresponding optimized exciting trajectory of the end-effector		
End-effector positions	**End-effector velocities**	**End-effector accelerations**

The joint torque values corresponding to the trajectory of the end-effector for both manipulators are shown in **Table 5.6**.

Table 5.6. Dynamic performance analysis joint torques of 2ⁿᵈ case (*RPaPa* ANTHRO first)
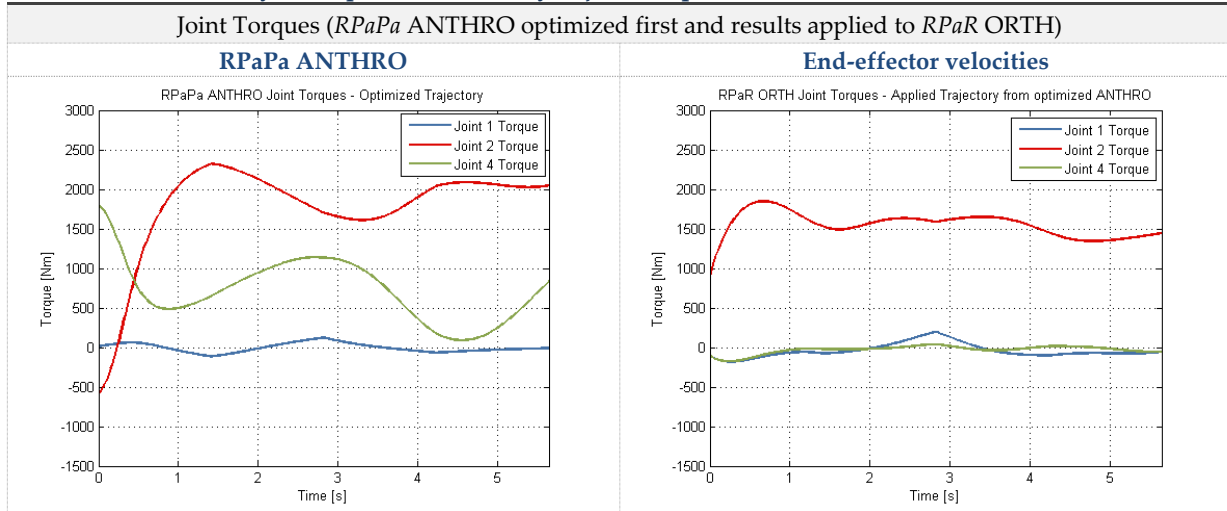
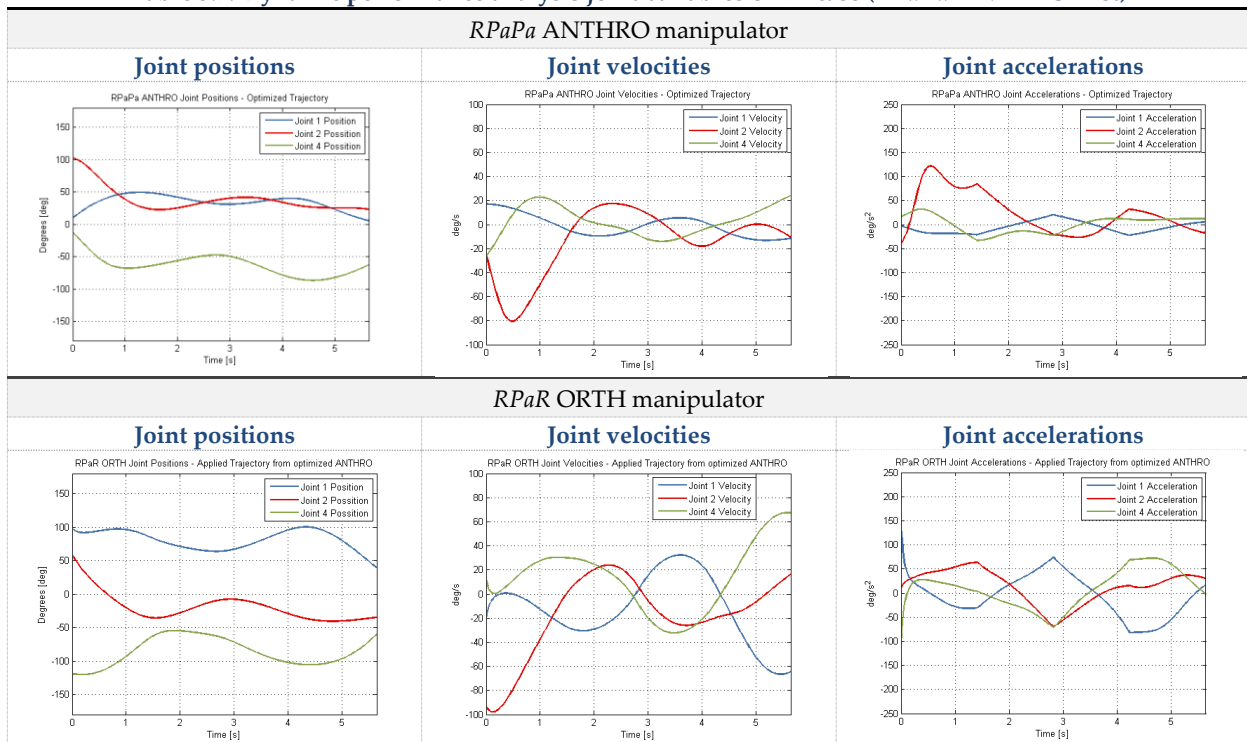| Joint Torques (*RPaPa* ANTHRO optimized first and results applied to *RPaR* ORTH)	
RPaPa ANTHRO	**End-effector velocities**

Finally, the joint variables of both robots corresponding to the optimized trajectory are shown in **Table 5.7**.

Table 5.7. Dynamic performance analysis joint variables of 2ⁿᵈ case (*RPaPa* ANTHRO first)

RPaPa ANTHRO manipulator		
Joint positions	**Joint velocities**	**Joint accelerations**

RPaR ORTH manipulator		
Joint positions	**Joint velocities**	**Joint accelerations**

5.5.2   General conclusions of the dynamic performance analysis

The dynamic analysis performed led to obtain some really interesting conclusions about the performance of both manipulators. In general, for the test cases shown before and others performed, it can be concluded that the *RPaR* orthogonal robot present better values (smaller) of the joint torques than the ones of the anthropomorphic manipulator.

For instance, in the first test case, where the trajectory is optimized to find the worst case for the *RPaR* orthogonal robot, it can be observed that the torques of joint 1 and 3 are considerably smaller than the ones of the *RPaPa* anthropomorphic manipulator. In this test case, the values of the torque of the second joint of the orthogonal manipulator are in average larger than those of the anthropomorphic robot (which is logical since the optimization is aimed to obtain the worst case for the orthogonal robot). Nevertheless, the difference between the torques of the joint 2 of the manipulators is significantly minor in comparison to the difference between the torques of joints 1 and 3, and the anthropomorphic manipulator presents some peak values larger than the average value of the orthogonal robot. In terms of accelerations, the average values for the orthogonal robot seem larger than the ones of the anthropomorphic one, but the latter presents a higher maximum peak value than the former one: and, in general, the values of the accelerations for both manipulators are in the same range. Thus, both manipulators perform similarly in terms of accelerations. As overall conclusion of the first test, the *RPaR* orthogonal manipulator shows better dynamic performances than the *RPaPa* anthropomorphic manipulator because the joint torques of the orthogonal manipulator present smaller values in general, which is desired.

The second test case also shows that the *RPaR* orthogonal manipulator has better dynamic performances than the anthropomorphic manipulator, since the torque values of joint 2 and 4 of the orthogonal robot are evidently smaller than those of the anthropomorphic one (and the values for joint 1 are similar for both robots). This is expected since the optimization of this test case is aimed to generate the worst case for the anthropomorphic robot. In terms of accelerations, similar conclusions as the ones of the previous case can be drawn. Therefore, as overall conclusion of the second test, the *RPaR* orthogonal manipulator shows better dynamic performances as well.

From the results of the dynamic analysis it can be concluded that:

> The dynamic performances of the *RPaR* orthogonal manipulator are better than those of the *RPaPa* anthropomorphic manipulator.

Conclusions

This report had the purpose of presenting the work done for the development of the master thesis called: *Analysis of the performances of serial robots with 4-dof based on RPaR orthogonal architecture.* This master thesis investigated the dynamic performances of a manipulator belonging to the *RPaR* orthogonal family and compared it against its counterpart of the *RPaPa* anthropomorphic architecture. More importantly, it proposed a robust approach for performing dynamic performance analysis of manipulators that may be applied not only to the family of robots presented in this research work, but to other different families and architectures. The work performed can be divided in three parts: the elastostatic modeling of the manipulators, the design parameter optimization procedure, and the dynamic performance analysis.

The elastostatic model was developed, applying a 6-DOF virtual joint method approach, in order to obtain the stiffness matrix of the manipulators and model the linear and angular displacement of the end-effector caused by a static load. The verification of the model developed against finite element analysis software demonstrated that the model done is correct and more important, that it has a high degree of accuracy as expected.

The design parameter optimization procedure was implemented in order to obtain, in a systematic way, the mass of the links of the manipulators, and in this manner assuring a fair comparison between the two robots in the last stage of this research. The procedure had the objective of minimizing the mass of the robots while assuring certain maximum allowable displacements of the end-effector (caused by a static load) for a desired working area. These displacements were calculated with help of the elastostatic model developed previously.

Before performing the dynamic performance analysis, the dynamic model of the manipulators was implemented. Furthermore, a torque analysis was applied to get an insight of the manipulators performance in the static case (which is when the velocity and acceleration of the joints are zero, thus the joint torques depend only on the robot configuration). From this analysis it was concluded that the *RPaR* orthogonal manipulator performed better than the *RPaPa* anthropomorphic manipulator in the static case.

The dynamic performance analysis was performed following the novel method of dynamic analysis based on maximum input torques and exciting trajectories discussed before. This method presented the advantage of using straightforward-to-understand indices (joint torques and accelerations) and it coped with the problem of dependency of these indices on the end-effector trajectory by applying an exciting trajectory optimization procedure. This optimization procedure had the purpose of making the manipulators exhibit the highest torques and accelerations; thus providing reliable information of the dynamic performance of the robots. Moreover, to make the analysis fairer

the following approach was applied: first, the optimization was implemented to find the worst possible case of joint torques for one of the manipulators, then, the trajectory obtained was applied to the other manipulator, and the joint torques and accelerations were plotted to be compared. The same was done starting with the opposite manipulator. Several test cases were performed with different *via-points* and initial conditions obtaining similar conclusions. Two test cases were presented in this document. It was concluded that *RPaR* orthogonal manipulator presents better dynamic performances than the *RPaPa* anthropomorphic manipulator.

Besides of the importance of these results, the contribution made with the work of this thesis, complementing the previous research, is the proposal of a robust methodology for analyzing the dynamic performances of manipulators, which may be an area of interest of study for other researchers. Thus, some envisioned future steps may be:

- To extend the dynamic model of the manipulators to include friction forces and a non-perfect controller, to study the effects of these factors on the performance of the robots.

- To apply the proposed methodology to analyze the performance of other types of orthogonal manipulators based in *RPaR* structure and to manipulators of other serial (single, tree-structure, complex) families and architectures.

- To make the necessary adjustments and apply the proposed methodology to analyze the performance of parallel manipulators.

References

[1] D. Nguyen, S. Briot and P. Wenger, "Analysis of the dynamic performance of serial 3R orthogonal manipulators," in *Proceedings of the ASME 2012 11th Biennial Conference on Engineering Systems Design and Analysis, ASME*, Nantes, 2012.

[2] D. Nguyen, *Analysis of the dynamic performances of serial 3R orthogonal robots, Master Thesis*, Nantes: École Centrale de Nantes / IRCCyN, 2012.

[3] S. Briot, A. Pashkevich and D. Chablat, "Technology-oriented optimization of the secondary design parameters of robots for high-speed machining applications," in *Proceedings of the ASME 2010 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Montreal, 2010.

[4] A. Pashkevich, D. Chablat and P. Wenger, "Stiffness analysis of overconstrained parallel manipulators," *Mechanism and Machine Theory*, vol. 44, no. 5, p. 966–982, 2009.

[5] K. Wisama and É. Dombre, Modeling, Identification and Control of Robots, Norfolk: Hermes Penton Ltd, 2002, p. 491.

[6] J. Craig, Introduction to Robotics: Mechanics and Control, Upper Saddle River, NJ: Pearson Prentice Hall, 2005.

[7] J. Angeles, Fundamentals of Robotic Mechanical Systems: Theory, Methods, and Algorithms, New York: Springer, 2002.

[8] ABB Robotics, "ABB Product Guide Robotics," 2012. [Online]. Available: http://www.abb.com/product/us/9AAC910011.aspx. [Accessed 02 May 2012].

[9] KUKA robots, "KUKA Products Industrial Robots," 2012. [Online]. Available: http://www.kuka-robotics.com/usa/en/products/industrial_robots. [Accessed 02 May 2012].

[10] M. Zein, P. Wenger and D. Chablat, "An exhaustive study of the workspace topologies of all 3R orthogonal manipulators with geometric simplifications," *Mechanism and Machine Theory*, vol. 41, no. 8, p. 971–986, 2006.

[11] S. Caro, W. A. Khan, D. Pasini and J. Angeles, "The rule-based conceptual design of the architecture of serial Schönflies-motion generators," *Mechanism and Machine Theory*, vol. 45, no. 2, p. 251–260, 2010.

[12] J. Angeles, "The Qualitative Synthesis of Parallel Manipulators," in *Proceedings of the WORKSHOP on Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulators*, Quebec City, 2002.

[13] S. Briot, V. Arakelian and S. Guégan, "Design and Prototyping of a Partially Decoupled 4-DOF 3T1R Parallel Manipulator With High-Load Carrying Capacity," *Journal of Mechanical Design ASME,* vol. 130, no. 12, pp. 1-8, 2008.

[14] EPSON Robots, "EPSON SCARA Robots," 1 January 2012. [Online]. Available: http://www.robots.epson.com/. [Accessed 28 January 2012].

[15] B. Kaddar, *Analyse et optimisation des performances cinématiques des manipulateurs orthogonaux de type RPaR,* Nantes: École Centrale de Nantes, 2009.

[16] DALMEC Industrial Manipulators, "Dalmec Products," 2012. [Online]. Available: http://www.dalmec.com/fr/manipulateurs_industriels.html. [Accessed 03 May 2012].

[17] Ruina and P. R. A., Introduction to Statics and Dynamics, Oxford University Press, 2011.

[18] A. Pashkevich, A. Klimchik, D. Chablat and P. Wenger, "Accuracy Improvement for Stiffness Modeling of Parallel Manipulators," in *CIRP Conference on Manufacturing Systems (42)*, Grenoble, 2009.

[19] C. Barnard, *Optimal Trajectory Generation for a High Speed Parallel Robot,* Nantes: École Centrale de Nantes / IRCCyN, 2011.

[20] G. Coralie, *Analyse et Conception d'un Nouveau Manipulateur Parallèle à Deux Degrés de Liberté pour des Applications de Pick-and-Place,* Nantes: Ecole Centrale de Nantes / IRCCyN, 2010.

[21] S. Briot, A. Pashkevich and D. Chablat, "Optimal Technology-Oriented Design of Parallel Robots for HighSpeed Machining Applications," in *Proceedings of the 2010 IEEE International Conference on Robotics and Automation*, Anchorage, 2010.

[22] Zein, *Analyse cinématique des manipulateurs sériels 3R orthogonaux et des manipulateurs parallèles plans*, Nantes: PhD Thesis, École Centrale de Nantes, Université de Nantes, 2007.

[23] A. Poulet, B. Rose and E. Caillaud, "Towards a Quality Referential for Performance in Design," in *Global Product Development. Proceedings of the 20th CIRP Design Conference, ECN*, Nantes, Springer, 2010, pp. 609-618.

[24] A. Pashkevich, A. Klimchik, S. Briot and D. Chablat, "Performance Evaluation of Parallel Manipulators for Milling Application," in *Global Product Development. Proceedings of the 20th CIRP Design Conference, ECN*, Nantes, Springer, 2010, pp. 619-629.

[25] P. Chiacchio, S. Chiaverini, L. Sciavicco and B. Siciliano, "Reformulation of Dynamic Manipulability Ellipsoid for robotic manipulators," *IEEE International Conference on Robotics and Automation,* vol. 3, pp. 2192-2197, 1991.

[26] R. Kurazume and T. Hasegawa, "A new index of serial link manipulator performance combining dynamic manipulability and manipulating force ellisoids," *IEEE TRANSACTIONS ON ROBOTICS,* vol. 1, no. 8, pp. 1022-1028, 2002.