Ecole Centrale de Nantes **Università degli Studi di Genova**

MASTER ERASMUS MUNDUS

EMARO "EUROPEAN MASTER IN ADVANCED ROBOTICS"

2010 / 2011

Thesis Final Report

Presented by

Dinh-Quan Nguyen

On 15/ 09 / 2011

Title

ANALYSIS OF THE DYNAMIC PERFORMANCES OF SERIAL 3R ORTHOGONAL ROBOTS

JURY

President:	**Wisama Khalil**	**Professor, ECN/IRCCyN**
Evaluators:	**Ina Taralova**	**Associate Professor, ECN/IRCCyN**
	Philippe Wenger	**CNRS Research Director, IRCCyN**
	Sébastien Briot	**Researcher, IRCCyN**
	Stéphane Caro	**Researcher, IRCCyN**

Supervisors: **Philippe Wenger, Sébastien Briot**

Laboratory: Institut de Recherche en Communications et Cybernétique de Nantes

ACKNOWLEDGEMENT

First of all, I want to give my great thanks to my two supervisors at IRCCyN/ECN (Nantes), profs. Philippe Wenger and Sébastien Briot. Without your advice and questioning, I would not come up with good solutions in this thesis work. I have learnt a lot from you.

Secondly, I want to give my thank to prof. Wisama Khalil, head of EMARO program. You always give us, EMARO students, best opportunities during our time under the master course.

I also want to give my thanks to the teaching staffs of EMARO program from University of Genova (UG, Italy), from Ecole Centrale de Nantes (ECN, France) and from Warsaw University of Technology (WUT, Poland). Thank you for your wonderful lectures.

Thank you EMARO guys. You are my friends and also my teachers.

Finally, I want to give my best wishes to all of you in EMARO program. I have had great time during my two years in Genova and Nantes.

Nantes, 02/09/2011

Dinh-Quan NGUYEN

INTRODUCTION

Serial 3R orthogonal manipulators have been studied recently and it has been proved that they can exhibit good performances in term of workspace size and kinematic properties. The aim of this research work is to analyze their dynamic properties and compare them with anthropomorphic robots, which are very popular in industry.

In this thesis, we present four chapters. ***Chapter 1*** shows a summary on the studies that have been done with this kind of robot family, the existing indices for characterizing the dynamic performances of robot manipulators, and introduce a definition for the term "dynamic performance" which will be used to specify our main task. In ***chapter 2***, we will discuss the modeling step in which we construct the dynamic models, geometric models of the manipulators that will be used for the analysis. ***Chapter 3*** shows several results of the dynamic performance analysis mainly focused on the pick-n-place applications, in which we indicate very intuitive aspects of the dynamic properties of the manipulators. And ***chapter 4*** shows an optimization procedure to construct some kinds of exciting trajectories to give the final answer to our main question, "***which type of manipulator is better, orthogonal robots or anthropomorphic robots?***"

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

DCE – Dynamic Capability Equations

IQV – Inertial Quasi-Velocities

EWA – Efficient Working Area(s)

IDM – Inverse Dynamic Model

DDM – Direct Dynamic Model

IGM – Inverse Geometric Model

DGM – Direct Geometric Model

RDW – Regular Dexterous Workspace

ORTH_VER – Orthogonal robot in Vertical mode

ORTH_HOR – Orthogonal robot in Horizontal mode

CHAPTER 1

1.1. Serial 3R orthogonal manipulators

At a first glance, someone may ask: "what are 3R orthogonal robots?". Because nowadays we can see many industrial robots are of anthropomorphic types (i.e. PUMA type) which have a vertical revolute joint followed by two parallel joint and a spherical wrist. The 3R orthogonal manipulators on the other hand are not so popular in industrial field. They have different geometric structure: any two consecutive joint axes are orthogonal. There are few examples of this type such as the IRB 6400C by ABB-Robotics launched in 1998 (not longer commercialized) and the DIESTRO robot built at McGill University.

Recently, researchers showed more and more interests in this kind of manipulators. Because upon changing of their links and joint offsets lengths these robots may have many different kinematic properties. The late studies, i.e. from [1,2,3], exposed deeply the analysis of geometric and kinematic properties of this robot family. For example the determination of the workspace boundary, the size and shape of the workspace, the existence of holes and voids, the accessibility inside the workspace, the feasibility of continuous trajectories in the workspace, singularity curves, and other properties like "cuspidality" (possibility of changing posture without crossing a singularity) were well-discussed.

In this section, we will briefly summary the works that have been done, especially on the classification of the 3R orthogonal robot based on their workspace topologies.

1.1.1. Preliminaries

3R orthogonal manipulators

The orthogonal manipulators are positioning manipulators with three revolute joints in which the two pairs of adjacent joint axes are orthogonal. The length parameters are defined in the DHM notation [32] $d_2,d_3,r_2,r_3 \geq 0$ and $d_4 > 0$ (d_4 cannot be equal to zero because the resulting manipulator would be always singular). The angle parameter α_2 and α_3 are set to -90^0 and 90^0 respectively. The three joint variables are referred to as θ_1, θ_2, θ_3 respectively. These angles can be unbounded due to the geometric of the link & joint offsets (this is also a great advantage of this manipulator). Figure 1.1 shows examples of the architecture of the manipulators. The position of the end-tip is defined by Cartesian coordinates [x y z] of the operation point P with respect to a reference frame **(**O,**X**,**Y**,**Z**) attached to the manipulator base [2,7]:

$$
\begin{aligned}
x &= \left[d_2 + r_3 S2 + \left(d_3 + d_4 C3\right) C2\right] C1 - \left(r_2 + d_4 S3\right) S1 \\
y &= \left[d_2 + r_3 S2 + \left(d_3 + d_4 C3\right) C2\right] S1 - \left(r_2 + d_4 S3\right) C1 \\
z &= r_3 C2 - \left(d_3 + d_4 C3\right) S2
\end{aligned}
\tag{1.1}
$$

where $C1 = \cos\left(\theta_1\right), S1 = \sin\left(\theta_1\right), C2 = \cos\left(\theta_2\right), \ldots$

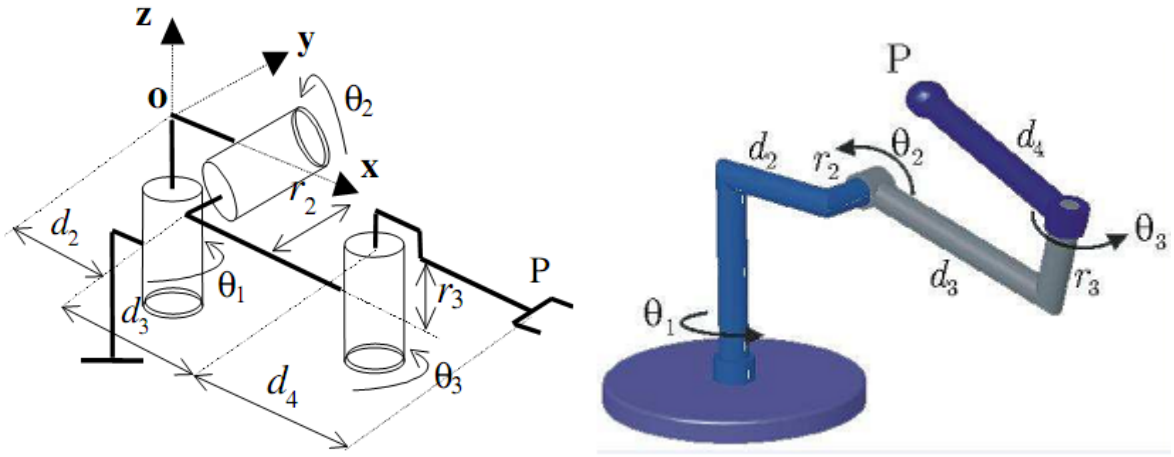


Fig. 1.1. Geometric structure of 3R orthogonal manipulator

Singular curves, cusp points and node points

The singularities of a manipulator play an important role in its kinematic properties. The singularities of a general 3R manipulator can be determined with the determinant of the Jacobian matrix det(**J**) [1,2…]:

$$
\mathbf{J} = \begin{bmatrix}
-S2C2d_4 - C2r_2 & r_3 & -S3d_4 \\
S3S2d_4 + S2r_2 & d_3 + C3d_4 & 0 \\
C2d_3 + C2C3d_4 & 0 & C3d_4
\end{bmatrix}
\tag{1.2}
$$

$$
\det(\mathbf{J}) = d_4\left\{\left(d_3 + d_4 C3\right)\left[\left(d_2 + r_3 S2\right) S3 + \left(d_3 S3 - r_2 C3\right) C2\right] + r_3\left(r_2 + d_4 S3\right) S2C3\right\}
\tag{1.3}
$$

where $C2 = \cos(\theta_2), S2 = \sin(\theta_2), \ldots$

Because there's no term related to θ_1 in the result, the singularity curves (or contour plot of det(**J**) = 0) can be displayed in term of θ_2, θ_3 with $-\pi \le \theta_2 < \pi$, $-\pi \le \theta_3 < \pi$. These curves divide the joint space into singularity-free domains.

Because of its symmetry about the first joint axis, the workspace may be analyzed by its half cross-section defined by $\left(\rho = \sqrt{x^2 + y^2}, z\right)$ in Figure 1.2 (from [33]):

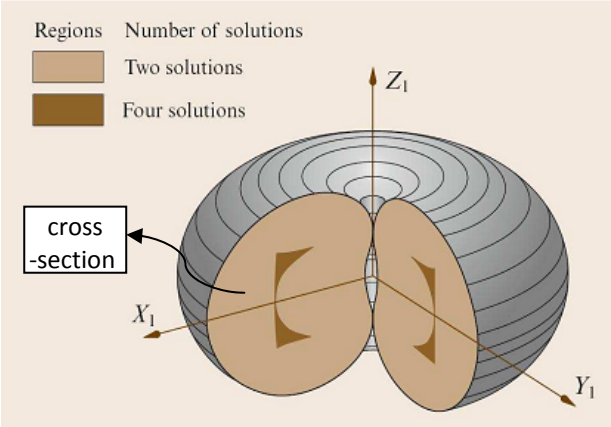


Fig. 1.2. Example of 3R orthogonal robot's workspace

Figure 1.3 shows the singularity curves of a 3R orthogonal manipulator with no offset along its last joint axis ($r_3 = 0$). The term det(**J**) takes on the factored form $\det(\mathbf{J})=(d_3 + d_4C3)[d_2S3 +(S3d_3 - C3r_2)C2]$. The first factor defines two horizontal lines in the joint space (assuming $d_3 \leq d_4$). The singular line defined by $\theta_3 = +\arccos(-d_3/d_4)$ maps onto one singular point in the workspace cross-section, which is located at the self-intersection of the internal singular boundary. The remaining singular line $\theta_3 = -\arccos(-d_3/d_4)$ maps onto an isolated singular point in the workspace. One of the two singular curves defines the external boundary of the workspace and the other one defines the internal boundary.

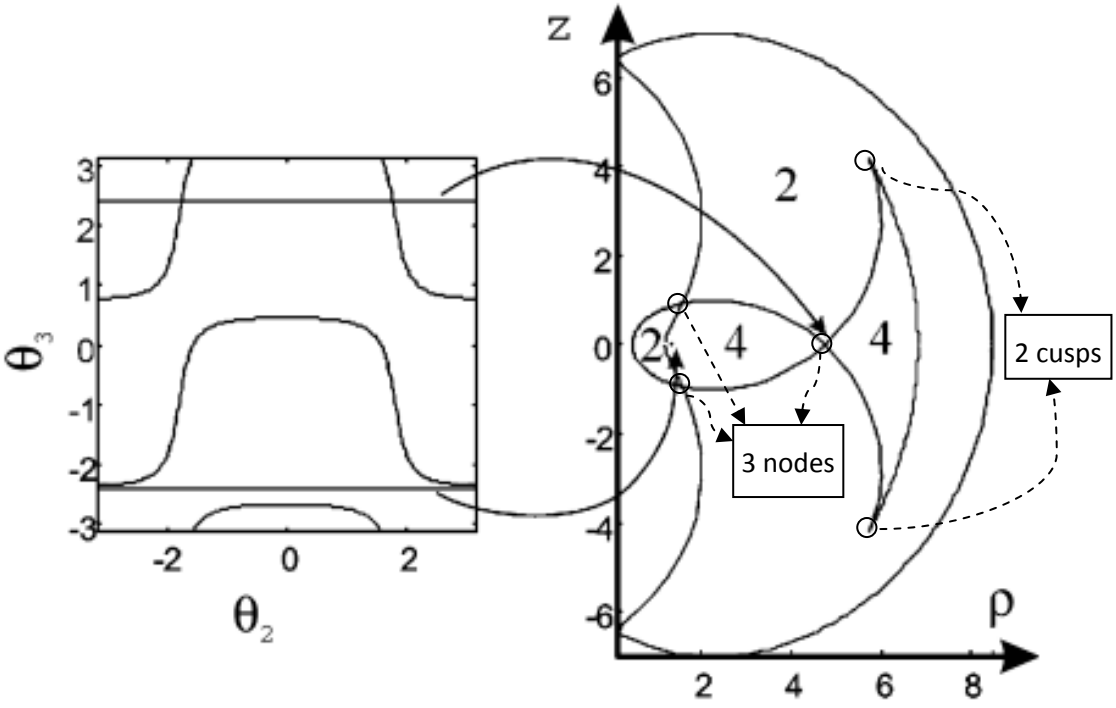


Fig. 1.3. The singularity curves of a 3R orthogonal robot with $d_2=1$, $d_3=3$, $d_4=4$, $r_2=2$, $r_3=0$

The topology of the singularity curves in the cross-section of the workspace can be used as a classification criterion for the manipulator family. A way of defining the topology of these curves is to enumerate their singular points, the cusp points and the node points [4,5].

A cusp point is one where the inverse kinematics admits three equal solutions [6]. A 3-DOF positioning manipulator can change its posture without meeting a singularity if and only if a cusp point is encircled [6].

At **a node point**, the inverse kinematics admits two pairs of coincident inverse kinematic solutions [7]. Two singular curves intersect at a node point.

For the manipulator whose singularities are shown in Fig. 1.3, the workspace features two cusps and three nodes. One of the nodes is at the self-intersection of the internal boundary and the other two nodes are at the intersection of the internal boundary with the external boundary. *These cusps and nodes define two regions with two inverse kinematic solutions and two regions with four inverse kinematic solutions*.

Feasibility of continuous trajectories in the workspace

An important kinematic feature is the feasibility of continuous trajectories within the workspace. This feature is important for process tasks such as welding or cutting. For non-cuspidal manipulators, the regions of feasible continuous trajectories in the workspace are the images of the aspects under the kinematic map [8,9,10]. This is because this map being one-to-one from each aspect onto its image in the workspace, the pre-image of any continuous trajectory of the workspace is a path in the joint space. The workspace of a manipulator is said to be t-connected if any continuous trajectory is feasible (or there'll be no singularity point inside the workspace boundary), which arises when one region of feasible continuous trajectories is found to be coincident with the whole workspace [8].

When the workspace is t-connected and composed of only one 4-solution region, it is said to be well-connected [11]. This is an interesting feature, which arises in Puma manipulators with equal link lengths (providing that all joints are unlimited) [11]. It can be shown that if a given manipulator has a t-connected (resp. well-connected) workspace, then all manipulators belonging to the same domain of the parameter space will have a t-connected (resp. well-connected) workspace as well [3].

1.1. 2 Classification of the ten families of 3R orthogonal manipulators

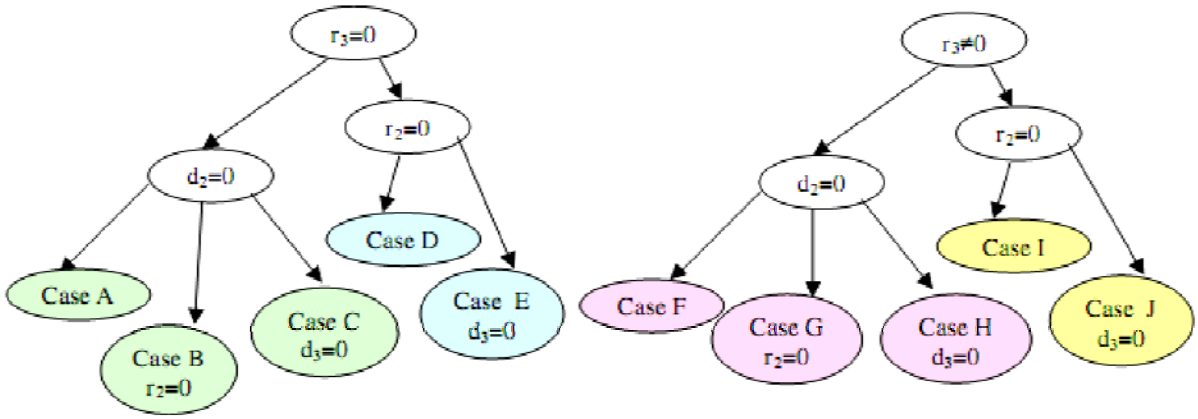


Fig. 1.4 .(from [2]) The ten families of manipulators with at least one parameter equal to zero

<u>Case A</u> ($d_2 = 0, r_2 \neq 0, d_3 \neq 0, r_3 = 0$ – Fig. 1.5)

- Type A1: The manipulators of this type have $d_3 > d_4$. Their workspace admits no voids nor node points, it is composed of two 2-solution regions and one 4-solution region. Their workspace is t-connected.
- Type A2: The manipulators of this type have $d_3 < d_4 < \sqrt{d_3^2 + r_2^2}$. Their workspace admits two node points and no voids. Their workspace is t-connected.
- Type A3: The manipulators of this type have $d_4 > \sqrt{d_3^2 + r_2^2}$. Their workspace admits four node points and no voids. Their workspace is not t-connected.

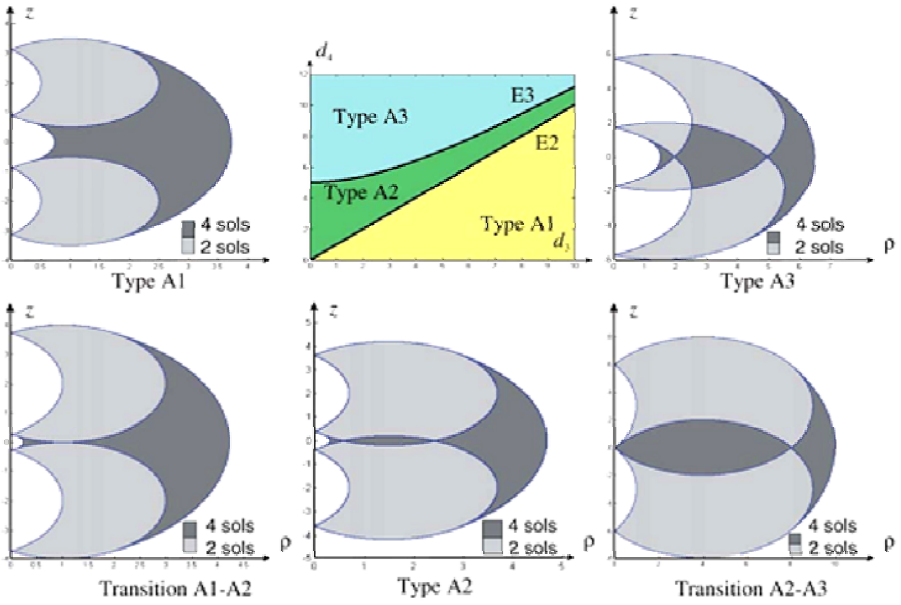


Fig. 1.5. (from [2]) Parameter space of case A and workspaces of the manipulators having the following parameters: Type A1 (d_3=2, d_4=1.5 and r_2=1); Type A2 (d_3=2, d_4=2.2 and r_2=1.5); Type A3 (d_3=2, d_4=3 and r_2=1). Transition A1-A2 (d_3=2, d_4=2 and r_2=1); Transition A2-A3 (d3=3, d4=5 and r2=4)

Case B ($d_2 = 0, r_2 = 0, d_3 \neq 0, r_3 = 0$ – Fig. 1.6)

- Type B1: The manipulators of this type have $d_3 > d_4$. Their workspace has no voids and no node points; it is composed of only one 4-solution region. Their workspace is t-connected.
- Type B2: The manipulators of this type have $d_4 > d_3$. Their workspace admits one node point and no voids. Their workspace is not t-connected.

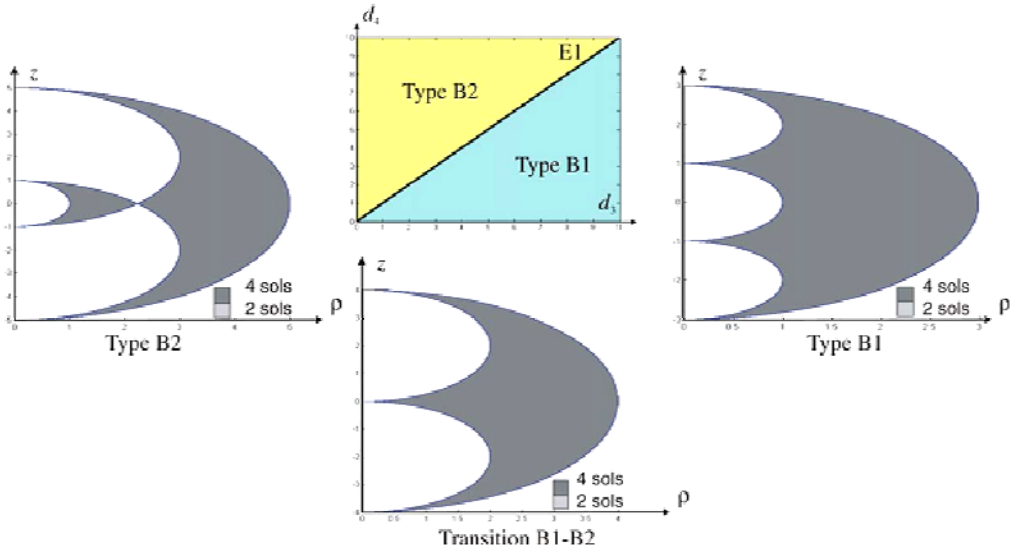


Fig. 1.6.(from [2]) Parameter space of case B and workspaces of the manipulators having the following parameters: Type B1 (d_3=2, d_4=3); Type B2 (d_3=2, d_4=1); Transition B1-B2 (d_3=2, d_4=2).

Case C ($d_2 = 0, r_2 \neq 0, d_3 = 0, r_3 = 0$)

- Type C: The manipulators of this type do not admit voids nor node points in their workspace, which is composed of only one 4-solution region. Their workspace is t-connected. Thus, it is also well-connected.

Case D ($d_2 \neq 0, r_2 = 0, d_3 \neq 0, r_3 = 0$ – Fig. 1.7)

- Type D1: The manipulators of this type have $d_4 < d_2 < d_3$. Their workspace admits one void and 2 node points. Their workspace is not t-connected.
- Type D2: The manipulators of this type have $d_2 < d_4 < d_3$. Their workspace does not admit voids nor node points. Their workspace is not t-connected.
- Type D3: The manipulators of this type have $d_2 < d_3 < d_4$. Their workspace admits one node point and no voids. Their workspace is not t-connected.
- Type D4: The manipulators of this type have $d_3 < d_2 < d_4$. Their workspace admits two node points and no voids. Their workspace is not t-connected.
- Type D5: The manipulators of this type have $d_3 < d_4 < d_2$. Their workspace does not admit voids nor node points. Their workspace is t-connected.
- Type D6: The manipulators of this type have $d_4 < d_3 < d_2$. They are binary, that is, their workspace is composed of only one region, which is reachable with two inverse kinematic solutions. Their workspace admits one void and no node points. Their workspace is t-connected.

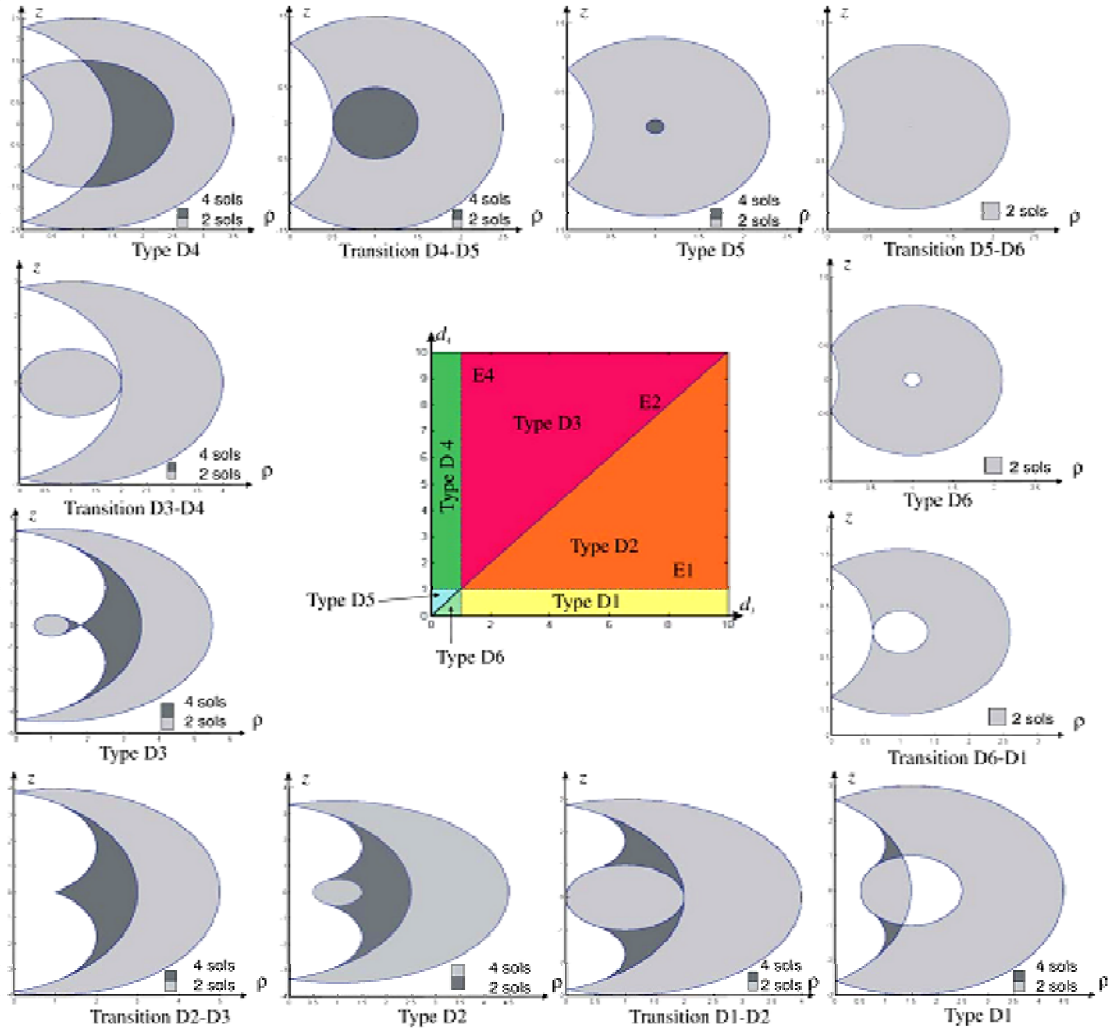


Fig. 1.7.(from [2]) Parameter space of case D and workspaces of the manipulators having the following parameters: Type D1 (d_3=1.4, d_4=0.7 and d_2=1); Type D2 (d_3=2, d_4=1.5 and d_2=1); Type D3 (d_3=2, d_4=2.5 and d_2=1); Type D4 (d_3=0.5, d_4=2 and d_2=1); Type D5 (d_3=0.6, d_4=0.7 and d_2 = 1); Type D6 (d_3=0.7, d_4=0.5 and d_2 = 1); Transition D1-D2 (d_3=2, d_4=1 and d_2 = 1); Transition D2-D3 (d_3=2, d_4=2 and d_2 = 1); Transition D3-D4 (d_3=1, d_4=2 and d_2 = 1); Transition D4-D5 (d_3=0.5, d_4=1 and d_2 = 1); Transition D5-D6 (d_3=0.6, d_4=0.6 and d_2 = 1); Transition D6-D1 (d_3=1, d_4=0.5 and d_2 = 1).

<u>Case E</u> ($d_2 \neq 0, r_2 = 0, d_3 = 0, r_3 = 0$)

- Type E: The manipulators of this type do not admit voids nor node points in their workspace, which is composed of only one 4-solution region. The workspace is t-connected. Thus, these manipulators have a well-connected workspace.

<u>Case F</u> ($d_2 = 0, r_2 \neq 0, d_3 \neq 0, r_3 \neq 0$ – Fig. 1.8)

- Type F1: The manipulators of this type have $d_4 < \sqrt{d_3^2 + r_2^2}$. Their workspace is free of voids and node points. It is composed of two 2-solutions regions and one 4-solution region. Their workspace is t-connected.

- Type F2: The manipulators of this type have $d_4 > \sqrt{d_3^2 + r_2^2}$. Their workspace admits 2 node points and no voids. Their workspace is not t-connected.

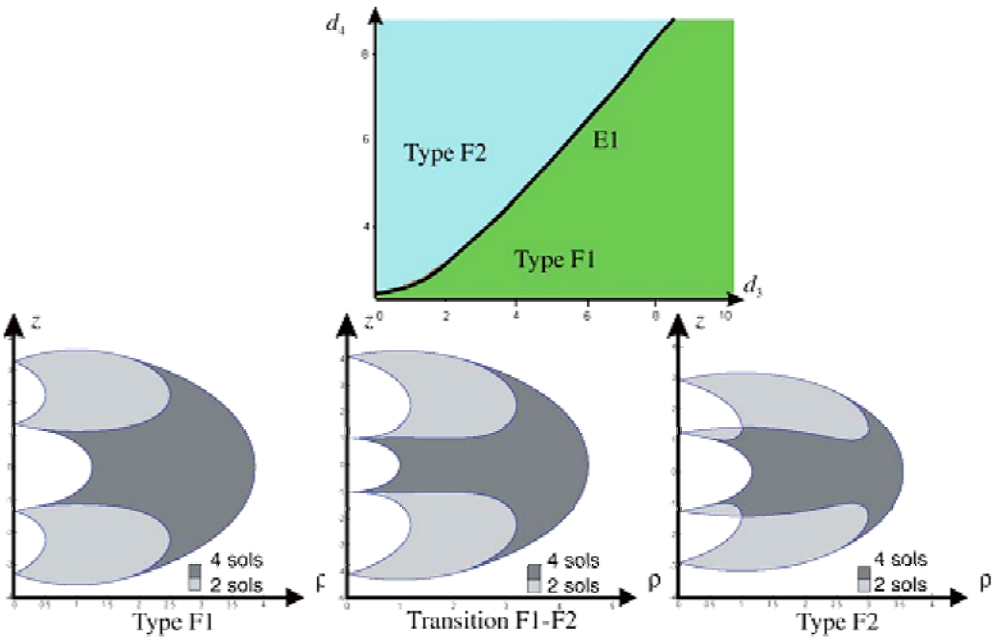


Fig. 9. Parameter space of case F and workspaces of the manipulators having the following parameters: Type F1 ($d_3 = 4.5$, $d_4 = 6.5$,

Fig. 1.8.(from [2]) Parameter space of case F and workspaces of the manipulators having the following parameters: Type F1 (d_3=4.5, d_4=6.5, r_2=2.5 and r_3=2); Type F2 (d_3=3, d_4=3, r_2=2.5 and r_3=2); Transition F1-F2 (d_3=2, d_4=2.24, r_2=1 and r_3=1).

Case G ($d_2 = 0, r_2 = 0, d_3 \neq 0, r_3 \neq 0$)

- Type G: The manipulators of this type do not admit voids nor node points in their workspace, which is composed of only one 4-solution region. The workspace is t-connected. Thus, these manipulators have a well-connected workspace.

Case H ($d_2 = 0, r_2 \neq 0, d_3 = 0, r_3 \neq 0$)

- Type H: The manipulators of this type do not admit voids nor node points in their workspace, which is composed of only one 4-solution region. The workspace is t-connected. Thus, these manipulators have a well-connected workspace.

Case I ($d_2 \neq 0, r_2 = 0, d_3 \neq 0, r_3 \neq 0$ – Fig. 1.9)

- Type I1: The manipulators of this type have $d_3 > d_2$ and $d_4 > \delta$. Their workspace is free of voids and node points. Their workspace is not t-connected.
- Type I2: The manipulators of this type have $d_3 > d_2$ and $d_4 < \delta$. Their workspace admits two node points and one void. Their workspace is not t-connected.
- Type I3: The manipulators of this type have $d_3 < d_2$ and $d_4 > \delta$. Their workspace admits one void and no node points. Their workspace is t-connected.
- Type I4: The manipulators of this type have $d_3 < d_2$ and $d_4 < \delta$. Their workspace admits two node points and one void. Their workspace is not t-connected.

Where $\delta = d_2\sqrt{1+\frac{r_3^2}{d_3^2-d_2^2}}$

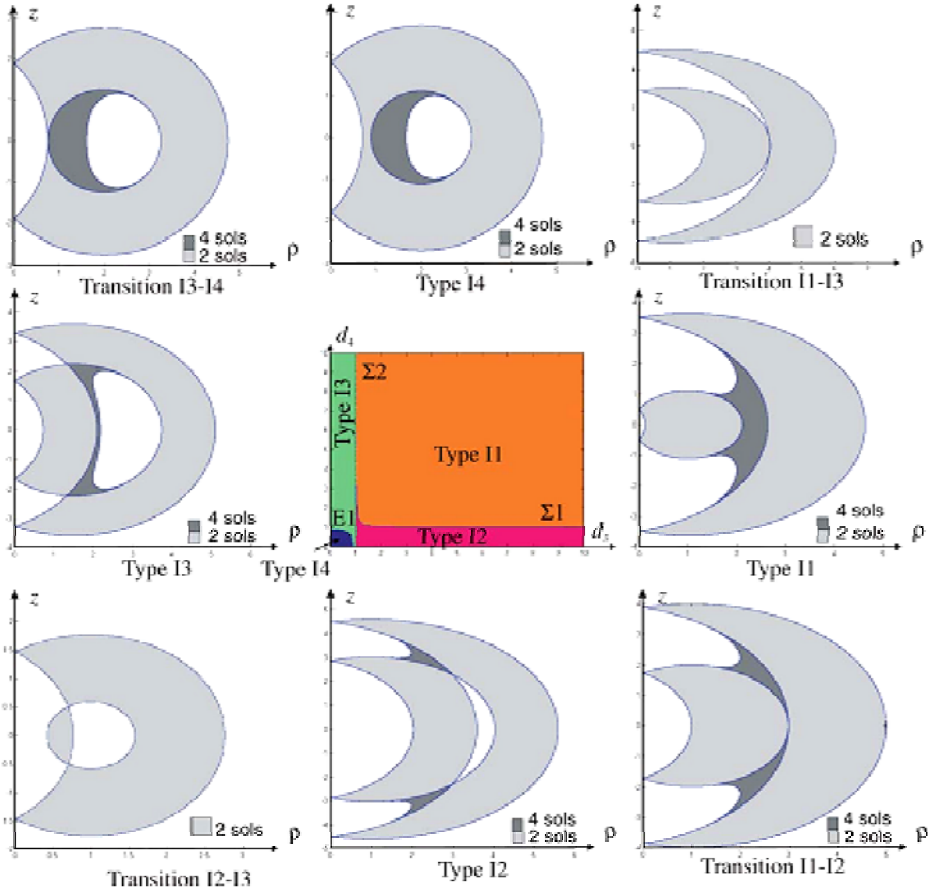


Fig. 1.9.(from [2]) Parameter space of case I with r_3=0.5 and workspaces of the manipulators having the following parameters: Type I1 (d_2=1, d_3=2.5, and d_4=1.5); Type I2 (d_2=1, d_3=3, and d_4=0.7); Type I3 (d_2=1, d_3=0.5, and d_4=0.7); Type I4 (d_2=1, d_3=0.3, and d_4=2); Transition I1-I2 (d_2=1, d_3=3, and d_4=1); Transition I1-I3 (d_2=1, d_3=1, and d_4=4); Transition I2-I3 (d_2=1, d_3=1, and d_4=0.7); Transition I3-I4 (d_2=1, d_3=0.2, and d_4=0.8).

<u>Case J</u> ($d_2 \neq 0, r_2 = 0, d_3 = 0, r_3 \neq 0$)

- Type J: The manipulators of this type admit one void and no node points in their workspace, which is composed of only one 4-solution region. Their workspace is t-connected.

Type	DH conditions	Void	Node points	4-Solution region	Workspace *t*-connected	Workspace well-connected
B1	$d_2 = 0,\ r_2 = 0,\ r_3 = 0\ d_3 > d_4$	0	0	All the workspace	Yes	Yes
C	$d_3 = 0,\ r_3 = 0$	0	0	All the workspace	Yes	Yes
E	$d_3 = 0,\ r_2 = 0,\ r_3 = 0$	0	0	All the workspace	Yes	Yes
G	$d_2 = 0,\ r_2 = 0$	0	0	All the workspace	Yes	Yes
H	$d_2 = 0,\ d_3 = 0$	0	0	All the workspace	Yes	Yes
A1	$d_2 = 0,\ r_3 = 0,\ d_4 < d_3$	0	0		Yes	No
D5	$r_2 = 0,\ r_3 = 0,\ d_3 < d_4 < d_2$	0	0		Yes	No
F1	$d_2 = 0,\ d_4 < \sqrt{d_3^2 + r_2^2}$	0	0		Yes	No
D2	$r_2 = 0,\ r_3 = 0,\ d_2 < d_4 < d_3$	0	0		No	No
I1	$r_2 = 0,\ d_3 > d_2$ and $d_4 > \delta$	0	0		No	No
B2	$d_2 = 0,\ r_2 = 0,\ r_3 = 0,\ d_3 < d_4$	0	1	All the workspace	No	No
D3	$r_2 = 0,\ r_3 = 0,\ d_2 < d_3 < d_4$	0	1		No	No
A2	$d_2 = 0,\ r_3 = 0,\ d_3 < d_4 < \sqrt{d_3^2 + r_2^2}$	0	2		Yes	No
D4	$r_2 = 0,\ r_3 = 0,\ d_3 < d_2 < d_4$	0	2		No	No
F2	$d_2 = 0,\ d_4 > \sqrt{d_3^2 + r_2^2}$	0	2		No	No
A3	$d_2 = 0,\ r_3 = 0,\ d_4 > \sqrt{d_3^2 + r_2^2}$	0	4		Yes	No
D6	$r_2 = 0,\ r_3 = 0,\ d_4 < d_3 < d_2$	1	0	Null	Yes	No
I3	$r_2 = 0,\ d_3 < d_2$ and $d_4 > \delta$	1	0		Yes	No
J	$r_2 = 0$ and $d_3 = 0$	1	0	All the workspace	Yes	No
D1	$r_2 = 0,\ r_3 = 0,\ d_4 > d_2 > d_3$	1	2		No	No
I2	$r_2 = 0,\ d_3 > d_2$ and $d_4 < \delta$	1	2		No	No
I4	$r_2 = 0$	1	2		Yes	No

Table 1.1.(from [2]) All types and kinematic properties of 3R orthogonal manipulator

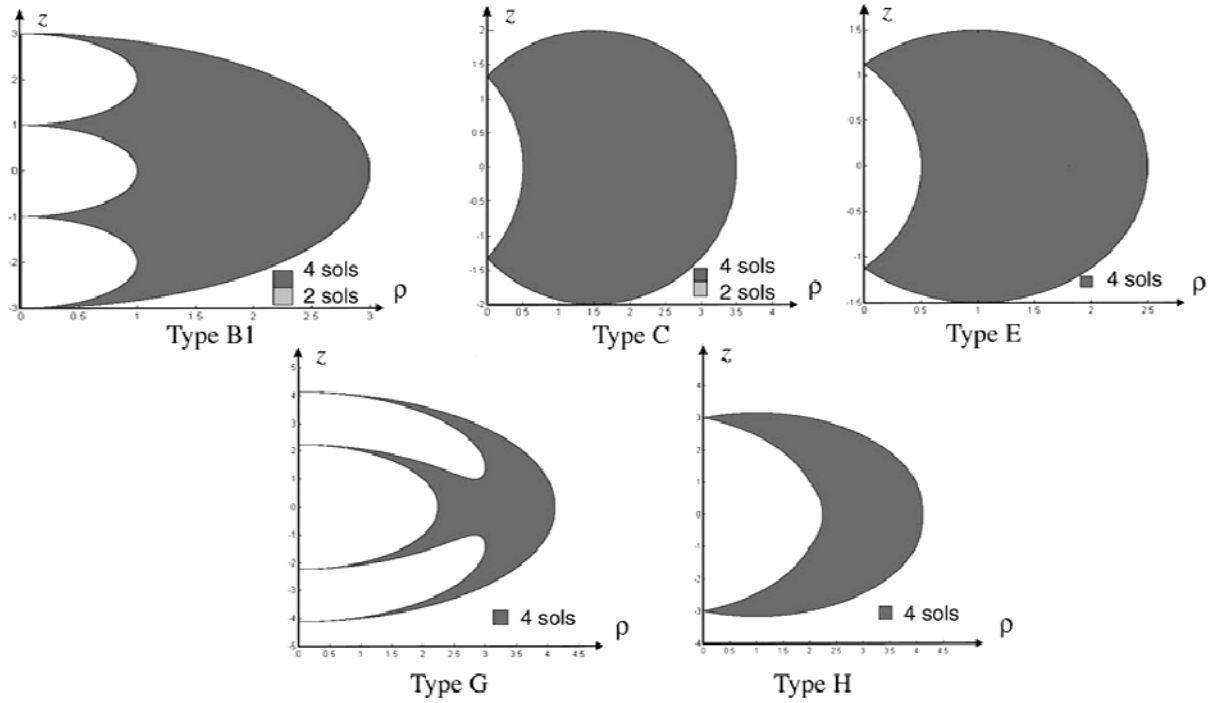


Fig. 1.10. (from [2]) The five manipulator types with well-connected workspace

Classification synthesis and discussion

Ten families of 3R orthogonal manipulators have been classified as function of the topology of their workspace. For all manipulators of one given type, the following global kinematic properties are the same:

- Number of nodes
- Number of voids
- Number of 2-solution and 4-solutions regions
- t-connectivity and well-connectivity of the workspace.

The classification of the ten families is synthesized in Table 1.1. It is apparent that some manipulator types have better properties than others. Besides, several manipulator types have similar properties. From Table 1.1, five manipulator types, namely, Types B1, C, E, G and H, have a well-connected workspace (workspace is fully reachable with four inverse kinematic solutions and fully t-connected) (Fig. 1.10) - which was shown in the past to be specific to Puma-type manipulators. On the other hand, manipulator Types A3, D1, D6, I2, I3 and I4 have poor kinematic performances and should be discarded by the designer. This is an interesting information for the designer, supporting for further optimization design problems in geometric, kinematic as well as dynamic performance.

1.2. Dynamic performance: State of the arts

What do you mean by "dynamic performance"? This question is quite difficult to give a correct answer. The fields related to dynamic performance of a robot are still very opening. Many papers that have discussed this problem tend to apply on particular cases of robotics instead of a general one. We can take a look at the industrial standard tests to evaluate a manipulator's performances like for example the ISO 9283 (since 1998 till now) which shows the performance criteria and related test methods for manipulating industrial robots. We could find some criteria related to dynamic performance of a robot but mainly on path-related characteristics of a manipulator and mainly from the static point of view and of course with real robots.

The motion of a manipulator, a multibody mechanical system, involves several effects caused by inertia, centrifugal, Coriolis, gravity and friction forces/torques. Most of these are nonlinear effects, which makes very difficult to predict and control the dynamic performance of the manipulator. Several measures of dynamic performance of manipulators have been introduced in the literature. Asada [12] introduced the generalized inertia ellipsoid (GIE) as a tool to measure the capability of changing end-effector's velocity in different directions for fixed kinetic energy. Yoshikawa [13] introduced the dynamic manipulability ellipsoid (DME) for measuring the ease of changing the end-effector's configuration by a set of joint torques with ked magnitude (Euclidean norm). Both the GIE and the DME are based on the relationship between the generalized inertia force of the end-effector and the generalized inertia torques of joints. Tourassis and Neuman [14] investigated the inertial characteristics of the dynamics equations of manipulators. They introduced the coefficients of coupling to measure the structural coupling of the dynamics equations. Khatib and Burdick [15] derived a

dynamics model in Operational space, which is, in fact, the space spanned by the vector representing the pose of the end effector. Using the operational space approach, they presented an optimization method for finding the largest and uniform bounds of end-effector's acceleration corresponding to the minimum joint torques at both low and high bounds of joint velocities. Graettinger and Krogh [16] introduced another performance measure termed acceleration radius. For given bounds of joint torques, the corresponding acceleration radius defines the minimum upper bound of the magnitude of end-effector acceleration over the whole workspace. Other measures that attempt to capture robot performance as a function of the dynamics can be cited: Voglewede and Ebert–Uphoff [17] propose performance indices based on joint stiffness and link inertia, with the aim of establishing a distance to singularity for any robot posture, while Bowling and Khatib [18] propose a general framework for capturing the dynamic capability of a general robot manipulator that includes the velocity and acceleration characteristics of the end-effector, taking into account factors such as torque and the velocity limits of the actuators.

In this section, we briefly review several methods to analyze the dynamic performance of a manipulator. These methods give quite interesting results such as presented in [19], [20, 21], [22] and [23, 18] – especially, each of them is derived from a different approach point of view.

On the concept of dynamic isotropy

The concept was introduced by On Ma and Jorge Angeles in 1990 [19], presented a measure of dynamic performance of manipulators. An associated index, namely, the dynamic conditioning index, is defined to quantitatively describe this concept. *This index measures the dynamical coupling and numerical stability of the generalized inertia matrix of a manipulator in its dynamics model. Reduction of this index weakens the dynamic coupling while increasing the numerical stability of the dynamics model, which leads in turn to an improvement of the robot performance for simulation and control under dynamical conditions*. The concept can be used as a criterion for manipulator performance and design optimization.

A probabilistic approach to manipulator dynamics

This approach was first introduced by P.K. Bhatti in his PhD in 1989 [20], then again presented by S.S. Rao in 2001 [21]. They studied the effect of various factors causing uncertainties in the behavior of a robotic manipulator. These factors include manufacturing and assembling tolerances, and errors in the joint actuators and controllers. In order to study the effect of these uncertainties on the robotic end-effector and to obtain a better insight into the manipulator behavior, they proposed a method to construct manipulator kinematic and dynamic model using a probabilistic approach. Based on the probabilistic model, kinematic and dynamic performance criteria are defined to provide measures of the behavior of the robotic end-effector. A new term "dynamic reliabilities" of the manipulator was introduced. All the experiments were carried on in both authors works ([20], [21]) applied only on the simple case with 2R planar manipulator in simulation and with the real Stanford Scara robot arm. This approach till now hasn't been exposed widely, however it's still quite interesting.

Dynamic capability equations (DCE)

The DCE was introduced by Bowling and Khatib in 2005 [18] as a new tool for analyzing robotic manipulator performance.

Dynamic capability equations (DCE) provide a new description of robot acceleration and force capabilities. These refer to a manipulator's ability to accelerate its end-effector and to apply forces to the environment at the end-effector. The key features in the development of these equations are that they combine the analysis of end-effector accelerations, velocities, and forces, while addressing the difference in units between translational and rotational quantities. The equations describe the magnitudes of translational and rotational acceleration and force guaranteed to be achievable in every direction, from a particular configuration, given the limitations on the manipulator's motor torques. They also describe the effect of velocities on these capabilities contributed by the Coriolis and centrifugal forces, as well as the reduction of actuator torque capacity due to motor speed.

The DCE have several uses in the analysis, design, and control of robotic manipulators. A short list of these include the following.

1) Comparing the suitability of different manipulators given the dynamic performance requirements of the task.
2) Characterizing the dynamic performance of a manipulator over its workspace or along a path by discretizing the region into a representative set of configurations, and statistically analyzing the DCE for the configurations in the set.
3) Motion planning which can use the worst-case direction information to avoid low performance configurations in a robot's workspace.
4) Selecting actuators for achieving a desired level of dynamic performance or actuator selection.

Energy-Based indices for manipulator dynamics improvement

Several indices for evaluation of nonlinear effects during the motion of a manipulator were presented by Herman [22]. All of them are based on the kinetic energy and arise from an analysis of equations of motion expressed in term of inertial quasi-velocities (IQV).

The IQV depend on the configuration of the system and take into consideration also its geometrical and inertial parameters. The indices enable to detect deformation of joint velocity as a consequence of manipulator motion, the kinetic energy resulting from dynamical couplings and the energy transferred by each link individually. Because the calculation of these indices can help to determine the effects of couplings on behavior of the manipulator then they can be used in the design phase for analysis and reduction of nonlinear effects in any motion. So far, all applications following these indices have been done but focus mainly on improving the control systems for robot manipulators.

1.3. Dynamic performance: a definition

Until now, as far as we have been looking at the studies related to the dynamic performance of robot manipulators, the indices which have been discussed only deal with a certain type of manipulators and at fixed configurations. In general there have been no global framework that could show us a good quantification of the performance.

In the following section, we will introduce a new definition for the term "dynamic performance" and interpret that concept further with several examples, including our main task in this thesis.

1.3.1. *Definition of dynamic performance*

Definition

"Dynamic performance is all the effects on the basic dynamic elements that are caused by the changes of selective factors in action."

Interpretation

We will now give an explanation in aspect of the dynamic performance of robot manipulators. In the definition, we introduced two keywords: "the basic dynamic elements" and "selective factors". What do they mean ?

At first, we need to clarify *our purposes* to do the analysis. For example, our main task here is that we want to do the analysis ***to compare the performance*** of the orthogonal robot with an anthropomorphic type manipulator. But in which applications these manipulators are used? Because the field of robotic applications in industry is very large, *we limit the application to the Pick-n-Place tasks* which are their most suitable ones.

Now which are "***the basic dynamic elements***"? In this case, they are the ***joint velocities***, ***joint accelerations***, ***actuator torques*** and ***reaction forces***… We call these variables *basic dynamic elements because* they play very important roles in the performance of the robots. For example, the maximum actuator torques will decide the capability of a manipulator, while the joint accelerations may relate to the vibrations or accuracy of the system and the reaction forces may relate to the bending rate of the robot arms…

And what are the "***selective factors***"? To determine this, we have to consider the application in that the manipulator is used. In Pick-n-Place tasks, there are two most important aspects which we pay attention to : ***speed*** of the end-effector and ***payload***. They are the *selective factors* that we will consider. And *the effects* are when we are changing these two terms, what will happen to the basic dynamic elements? By doing the analysis based on these effects, we could find the limit boundaries under which our robots can work well and we could have a very clear view of the dynamic performance of the manipulators.

To expand the concept further: why do we want to perform the analysis of the dynamic performance? Because we want to improve the control quality of the whole robot system (increase the accuracy of the end-effector, or limit the risk and increase the stability…), or compare with another robot, or to prevent the deformation of the links … We want to apply the concept for a parallel robot with flexible links, or to analysis the performance in the Pick-n-Place tasks or in machining tasks or in welding tasks… ***For a certain purpose in a certain task, we could perform the analysis more efficiently if we can specify all important aspects: "the goals in analysis", "the effects", "the basic dynamic elements" and "the selective factors"***.

Example : using the definition to formulate a problem

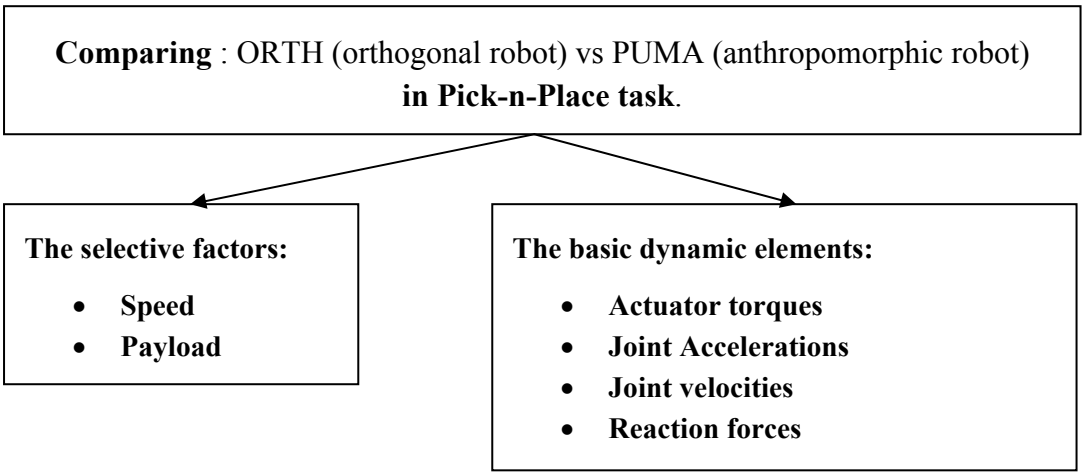


Fig. 1.11. Formulate problem: comparison dynamic performance between ORTH vs PUMA

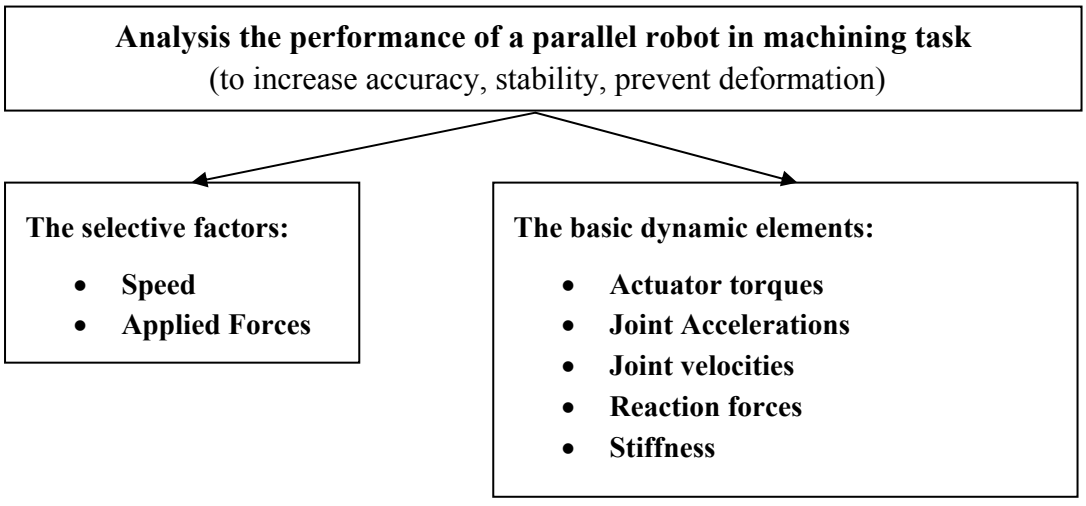


Fig. 1.12. Formulate problem: analysis dynamic performance of parallel robot in machining task

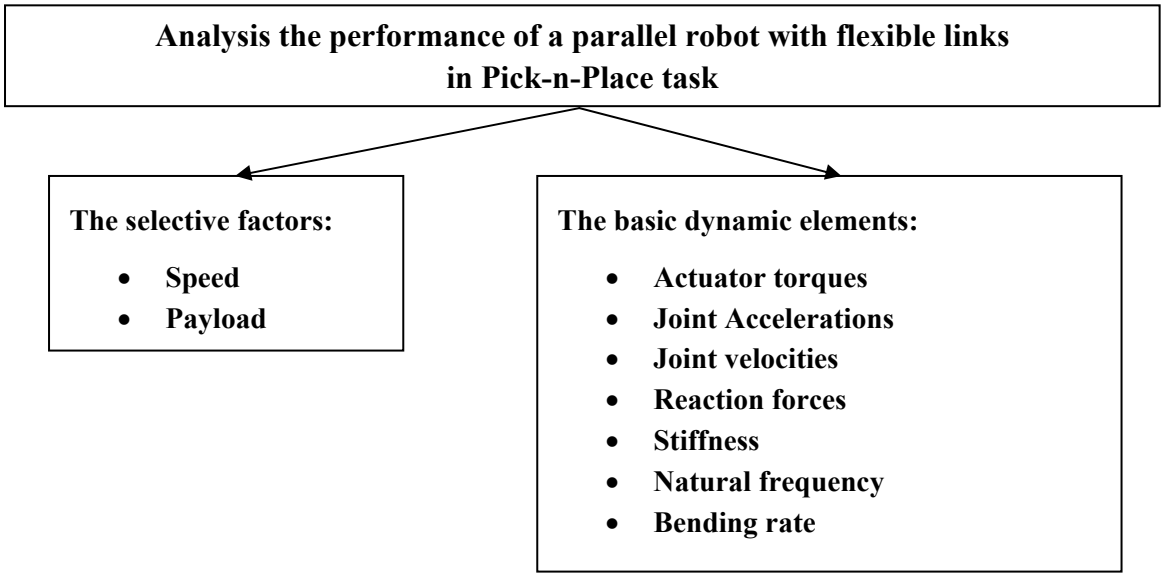


Fig. 1.13. Formulate problem: analysis dynamic performance of parallel robot in Pick-n-Place task

Comment

After specifying all the terms in the definition, depending on our needs, we may eliminate some basic dynamic elements that are not important for the analysis (i.e. in *section 3.2.3*).

We can use this definition not only in robotics, but also in other fields. The advantage of this concept is its simple and quite straightforward view point.

1.3.2. *Efficient working areas (EWA)*

In this section, we introduce an index that could present the performance of a manipulator within the desired working area by trying to find the maximum capability which the manipulators can achieve under the constraints of some effects.

Coming back to our main task: comparing the dynamic performance of the ORTH robot and PUMA robot, from the previous section, we want to perform the analysis in the Pick-n-Place tasks and there are two most important terms that we consider in these tasks: **speed** of the end-effector (or operational velocity) and **payload**. Naturally, one may want to see the effects by changing these two terms on the basic dynamic elements of the manipulators. Hence, we could present the effects directly on these two terms. By doing that, we compute the relations of the maximum values of the related dynamic elements (i.e. maximum accelerations, maximum torques, maximum reaction forces) depend on speed (Vs) and payload (Mass):

$$\ddot{\mathbf{q}}_{\max}, T_{\max}, F_{\max} = g\left(Vs, Mass\right) \tag{1.4}$$

These relations will determine some areas which we call **"Efficient Working Areas" (EWA)** ***in which the manipulators can work more efficiently and the whole robot system is expected not to exceed the limits defined by these EWA areas***.

Computing the EWA

To compute the EWA index, we can achieve from simulation or from experimentation. One way to do in simulation is that, through the Inverse Dynamic Model of the robot manipulator, at fix (Vs, Mass), for certain trajectory of the end-effector, we can compute the maximum values of the joint acceleration, actuator torques and reaction forces… The idea is to find a common area in that it satisfies all the constraint boundaries of the maximum values with respect to each basic dynamic element.

The EWA hence can be measured by the volume of a cube ($\ddot{\mathbf{q}}_{\max} / T_{\max} / F_{\max}$, Vs, Mass) or only by the area defined by the limit boundaries of (Vs, Mass):

- *Acceleration*: $EWA_{acc} = \ddot{q}_{\max} * Vs * Mass$
- *Actuator torques*: $EWA_{torq} = T_{\max} * Vs * Mass$
- *Reaction force*: $EWA_{F_{xy}} = F_{\max} * Vs * Mass$

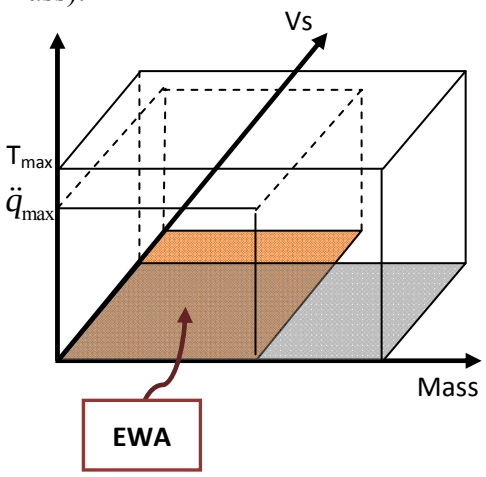


Fig. 1.14. Example of EWA region that is under both constraints of maximum joint torques and joint accelerations

Comment

The advantage of the EWA index in the above case is that, *we can see the tradeoff between the speed and the payload with respects to a certain basic dynamic element*. For example, with a fixed maximum joint torque, if we increase the speed Vs then the payload that the end-effector can carry will be reduced and vice-versa.

In comparison, we expected to bring out the maximum capability of each manipulator (or worst case possible). The EWA then can be the difference between the maximum values of respected basic dynamic elements of two manipulators. The contour plots, for example, of {Tk_{max}(ORHT) – Tk_{max}(PUMA) } will define the working areas that show which robot performs better in term of actuator torques (here, Tk_{max} is the maximum value of each joint's torques with respects to a fixed speed & payload). This will be detailed later in *chapter 3* (dynamic analysis).

The concept of EWA is not limited within the relations of the speed and payload to a basic dynamic element. For a system, for a certain task, all the terms will relate to each other by some ways. For example, if we want to analyze the accuracy of the end-effector, we might want to find the relation of it with the vibration rate or bending rate of the links – which are much related to the accelerations or the reaction forces of each joint, and so on… *The key point to be able to use this concept is that we need to find a way to formulate the relations between any element that may give us necessary information*, for example:

- Inverse dynamic model / Direct dynamic model compute the relation between joint values and Torques or Forces. From them we get the maximum capability in performance.
- Kinetostatic models compute the stiffness of the system is also the link between the elements.
- The probabilistic model introduced by Bhatti [20] is also to determine the accuracy of the robot system.

1.4. Evaluation method in performance comparison

In comparison, we try to find the differences in performance between two manipulators, worse or better. A good evaluation method is expected. We divide our method into some main steps following our concept of dynamic performance:

- **Modeling the two manipulators** to get the relations between basic dynamic elements (in this work, mainly the joint accelerations and actuator torques)
- **Choosing suitable robots** among their families, so that the two manipulators can have similar aspects of geometric, kinematic (workspace) and dynamic parameters.
- **Perform static analysis** to get one important basic dynamic characteristic (static torques profile) of each manipulator.

  What are the differences between the two manipulators because of their changed in geometric structure while they have the same arm links? Do they bring any benefit by changing the structure like that? – The first answer is yes, if we looked at the studies that have been done on the kinematic analysis of the ORTH manipulators, indeed they have nice properties. And now come to the dynamic properties, the static analysis will give us a part of the answer for this. In fact, we use the static torques profiles of the two manipulators as a criteria for further analysis in next steps.
- **Perform dynamic analysis** to get some basic dynamic characteristics while validating with common methods of generating standard Pick-n-Place trajectories within several desired working areas. We show two results in this analysis step. *The first result* provides us some information of the dynamic characteristic of each robot while following basic directions and by changing the speed and payload effects. *In the second result*, we use the EWA index to show the differences in dynamic performance.
- **Perform optimization procedure** to give a persuade answer to the question: "*which manipulator could perform better?*". By doing optimization, we try to find excited trajectories that could give us rich information of the dynamic performance: whether or not we could see clearly the differences between two robots? This step is important since we generate the trajectories which cover all regions in the robot's workspace and with various directions and under different constraints. *And finally, we can compute the EWA areas for the optimized trajectories.*

CHAPTER 2

2.1. Modeling method

Nowadays modeling plays a more and more important role in many research fields. It saves us time and helps us to develop our problems more efficiently and systematically. In this thesis work we are dealing mainly with serial manipulators with rigid bodies. Fortunately, there are several simulation tools that have been developed for these types of robot architecture. We can list some of the methods as following:

- **Symbolic dynamic model (Lagrange/Newton Euler)** (i.e. SYMORO+)
  The only main problem here is to determine the dynamic parameters of each link properly: **[XXj XYj XZj YYj YZj ZZj MXj MYj MZj Mj]**
  Using a CAD software to calculate them is one way, but we can calculate manually for simple structures (see more in the appendix section).
- **Using ADAMS software (or any other CAD system like CATIA, DELMIA)**
  This package is quite nice in simulation, especially for dynamic analysis. It can compute the dynamic parameters of any link with real materials and a lot useful features. Users can also export linear/nonlinear model from ADAMS to MATLAB Simulink and perform co-simulation [35]
- **Using SimMechanic Toolbox in MATLAB Simulink**
  There are a few ways to interact MATLAB with CAD software like CATIA, ProE, Solidwork.. as following: **CAD => Export XML/VRML model => SimMechanic Toolbox**
  In these ways we can build the CAD model freely and also update right away to Simulink model then simulate with 3D Animation toolbox in MATLAB.

Depending on the type of applications we can use suitable methods. In our task, we have mainly used the software MATLAB, SYMORO+ and ADAMS to perform the analysis and simulation (fig. 2.1):

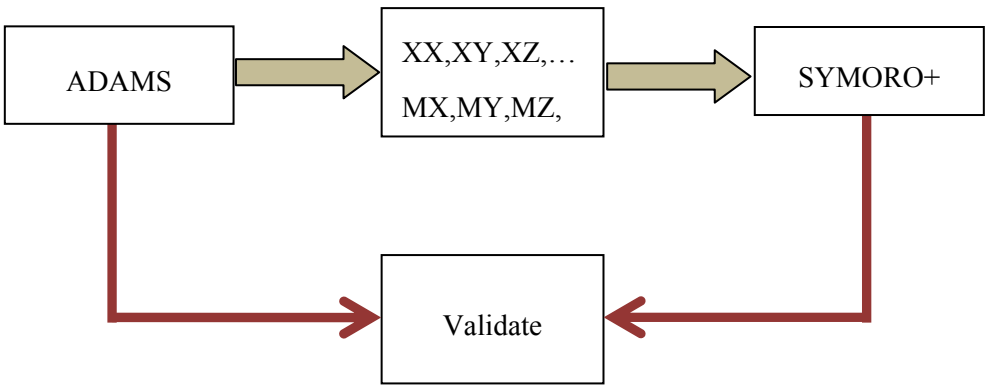
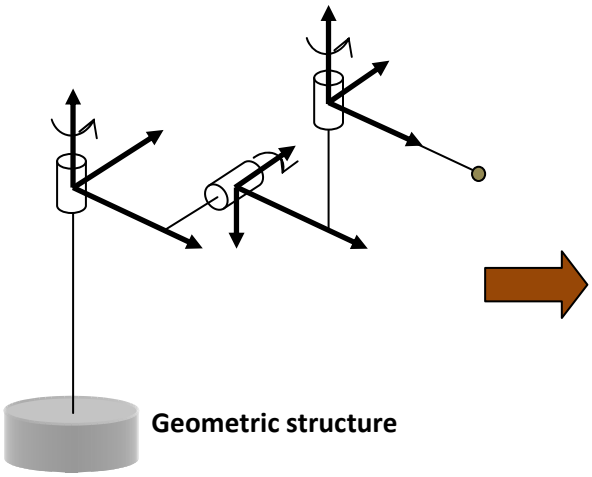
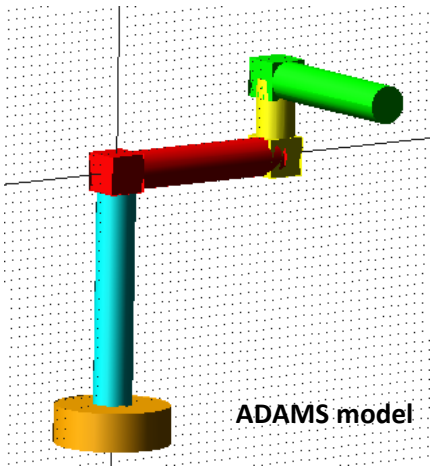


Fig. 2.1. Modeling method

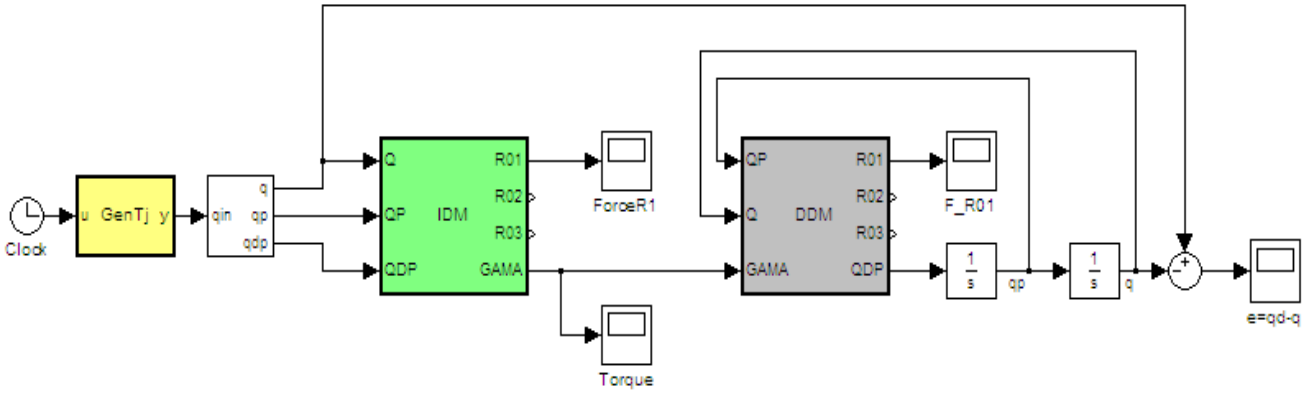
The benefit of using different simulation tools is that we can easily validate if our model is correct or not.

Example of modeling ORTH robot

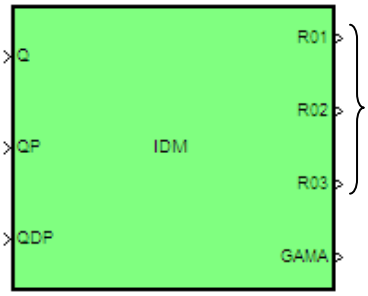




Export: XX,XY,XZ,…
MX,MY,MZ,M



Matlab Simulink scheme (using SYMORO+)



Reaction forces at each frame's origin (R1, R2, R3) with respect to the base frame R0. (**Give the same results as ADAMS**)

Fig. 2.2. Example of modeling steps for ORTH robot

2.2. Choosing manipulators: ORTH and PUMA

As we mentioned before, the main purpose of this thesis work is to compare the Orthogonal manipulators with a common type of anthropomorphic manipulators in industry (PUMA robots) to see which ones could perform better, specifically in the Pick-n-Place tasks. But there are quite a number of sub-families in each type of manipulators based on their workspace topology (as we have seen ORTH robot has ten sub-families). How can we choose the suitable ones for comparison?

The following section will briefly review a part of the work that has been done by Mazen Zein in his PhD thesis in 2007 [27] to select similar sub-families of ORTH robot and PUMA robot based on an index which was introduced by Félix Majou in 2004 [28].

2.2.1. *Selecting criteria: RDW*

Performance index that accounts for workspace compactness and dexterity:

$$\eta = \frac{a_{RDW}}{\rho_{\max}} \tag{2.1}$$

Where, a_{RDW} is side's length of the regular dexterous workspace section and $\rho_{\max}$ is the maximal reach of the manipulator.

The regular dexterous workspace (RDW) is a part of the working area – free of singularities and with a regular shape (cube, square…) – within which, the dexterity is better than a minimal prescribed value λ. The dexterity can be assessed with the reciprocal of the conditioning index k^{-1} of the Jacobian matrix $\mathbf{J}\left(\text{derived from}: \dot{\mathbf{X}} = \mathbf{J}\dot{\mathbf{q}}\right)$:

$$k = \frac{\sigma_{\max}}{\sigma_{\min}}, \quad \sigma_{\max} \text{ and } \sigma_{\min} \text{ are maximum and minimum eigen values of } \mathbf{J} \tag{2.2}$$

Calculate the RDW with square shape:

- Find the maximal square inscribed in the workspace cross section.

  a=2*min(d)

  where d is the Chebyshev distance from the center point M_0

  to a singular point A_s:

  $$d=\max\left(\left|\rho_0-\rho_s\right|,\left|z_0-z_s\right|\right)$$

- Find the maximal sub-square where $k^{-1}>\lambda$ where λ is a prescribed value that is up to the designer

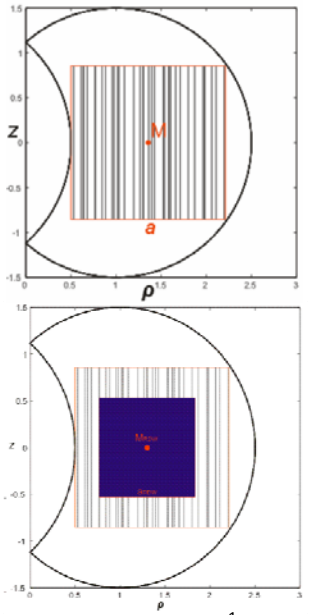


Fig. 2.3. RDW with K^{-1}_{min}=0.25, ORTH type C: d_4 = 1.5 , r_2 = 1

In the following, we shows the optimization results (maximize the index η) of the ORTH and PUMA manipulators which will be used later in the analysis to comparing their performance.

Optimized ORTH robots

The selected optimized ORTH robot belongs to type C with the performance index η=0.56

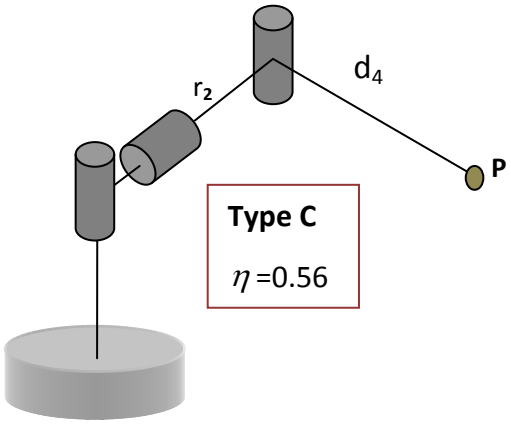


Fig. 2.4. **ORTH robot** - No offset (r_3=0), $r_2 = d_4$ (d_2 = 0)

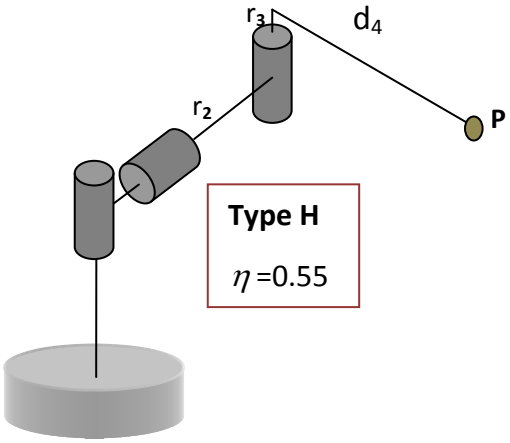


Fig. 2.5. **ORTH robot** - offset ($r_3 \neq 0$), to enable full rotation: d_4=0.95r2, r_3=0.05r_2 (d_2 = 0)

Optimized PUMA robots

The selected optimized PUMA robot has the performance index η=0.6

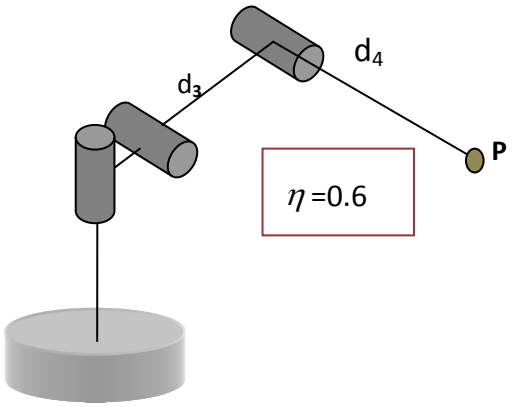


Fig. 2.6. **PUMA robot** - $d_3 = d_4$

2.2.2. *Analytical models*

In this section, all the models of the two selected manipulators (type C - ORTH and PUMA) will be computed.

We can note that, the dynamic models of the two manipulators can be generated from the package SYMORO+ , hence we will mainly focus on their geometric models (inverse geometric model – IGM, and direct geometric model – DGM).

From the structure of these two types, the two robots only differ from the joint axis directions, all their arm links can be the same. We will use this as a common condition for the comparison later. For the ORTH robot, because of the difference in their workspace, we will consider two working modes: Vertical (the workspace is symmetric with respects to gravity) and Horizontal (simply rotate the first joint axis by 90^0 – lose symmetric property).

ORTH robot – Type C (Vertical mode)

J	σ	α	d	θ	r
1	0	0	0	θ_1	0
2	1	-90^0	d_2	θ_2	r_2
3	2	90^0	d_3	θ_3	r_3

Table 2.1. Geometric parameters of ORTH_VER robot

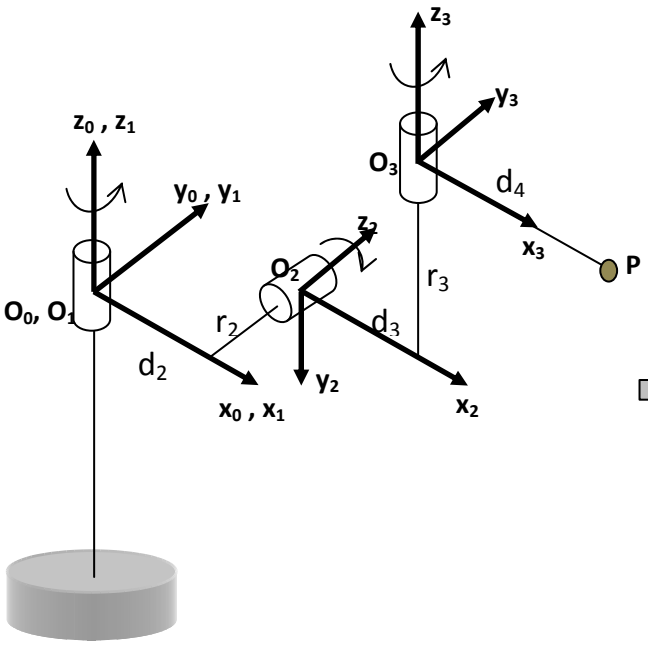


Fig. 2.7. Geometric structure of ORTH robot

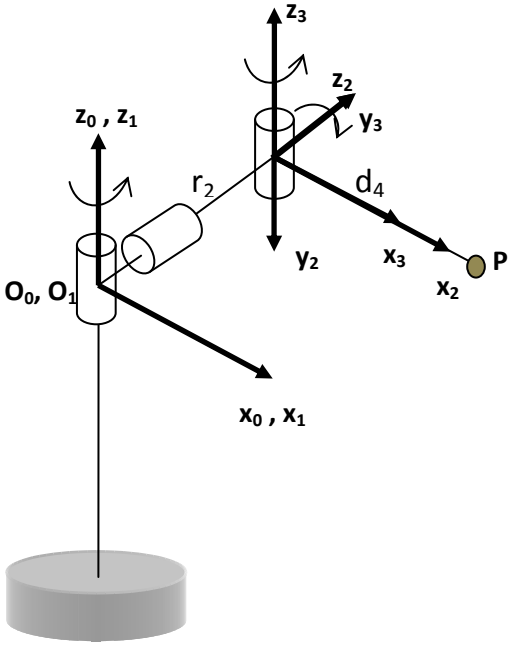


Fig. 2.8. ORTH case: d2=d3=r3=0; r2=d4

Type C : d2=d3=r3=0

r2 = d4 = 0.433 m

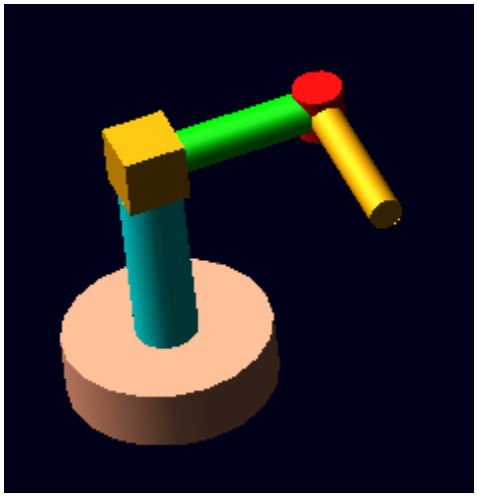
In this case the dexterous workspace index EDR = 0.56

Link mass:

M1 = 4.43 kg

M2 = 10.2 kg

M3 = 4.8 kg



Please note that, we choose the Mass and Link-Length parameters for the selected ORTH robot and PUMA robot similar to the Mass and Link-Lengths of the real PUMA-560 robot [29] just to have a closer analysis with the practical world.

The workspace of the 3R – ORTH robot in this case is a torus, simply generated by rotating a sphere with radius R = d4 around $\mathbf{z}_1$.

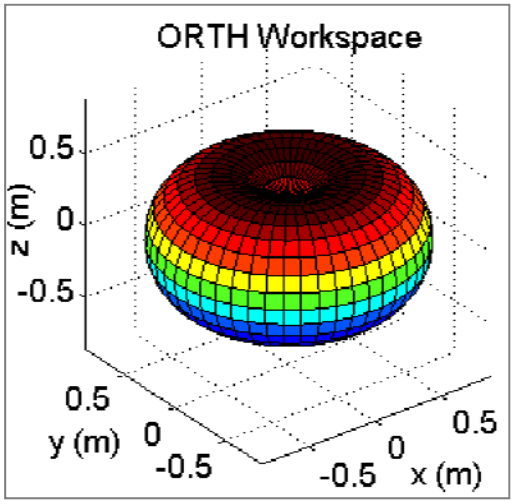


Fig. 2.9. ORTH Workspace (Vertical mode)

We can have exact inverse geometric solutions for this trivial case:

Direct Geometric :

$$(\mathbf{P})\begin{cases} x = \left[d_2 + (d_3 + d_4 C3)C2 + r_3 S2\right]C1 - (r_2 + d_4 S_3)S1 \\ y = \left[d_2 + (d_3 + d_4 C3)C2 + r_3 S2\right]S1 + (r_2 + d_4 S_3)C1 \\ z = r_3 C2 - (d_3 + d_4 C3)S2 \end{cases}$$

$$\rightarrow (\mathbf{O}_2 \equiv \mathbf{O}_3)\begin{cases} x = -r_2 S1 \\ y = r_2 C1 \\ z = 0 \end{cases} ; \qquad (\mathbf{P})\begin{cases} x = d_4 C3C2C1 - d_4(1+S3)S1 \\ y = d_4 C3C2S1 - d_4(1+S3)C1 \\ z = -d_4 C3S2 \end{cases} \tag{2.3}$$

Inverse Geometric (4 *solutions*) :

$$\begin{cases} q_3 = \text{asin}\left(\dfrac{x^2+y^2+z^2}{2d_4^2} - 1\right) \rightarrow 2 \text{ solutions} \\ q_2 = \text{asin}\left(\dfrac{-z}{d_4 C3}\right) \rightarrow 2 \text{ solutions} \\ q_1 = \text{atan2}\left(\dfrac{C3C2y - (1+S3)x}{C3C2x + (1+S3)y}\right) \quad (*) \end{cases} \rightarrow \textbf{4 solutions} \begin{cases} \mathbf{Q}1 = [q_{11} \quad q_{21} \quad q_{31}] \\ \mathbf{Q}2 = [q_{12} \quad q_{22} \quad q_{31}] \\ \mathbf{Q}3 = [q_{13} \quad q_{23} \quad q_{32}] \\ \mathbf{Q}4 = [q_{14} \quad q_{24} \quad q_{32}] \end{cases} \tag{2.4}$$

where :

$$\begin{cases} q_{32} = \pi - q_{31} \\ q_{22} = \pi - q_{21} \\ q_{23} = -q_{21} \\ q_{24} = \pi + q_{21} \end{cases} \rightarrow \text{Substitute in (*) we have} \begin{cases} q_{11} = q_{14} \\ q_{12} = q_{13} \end{cases}$$

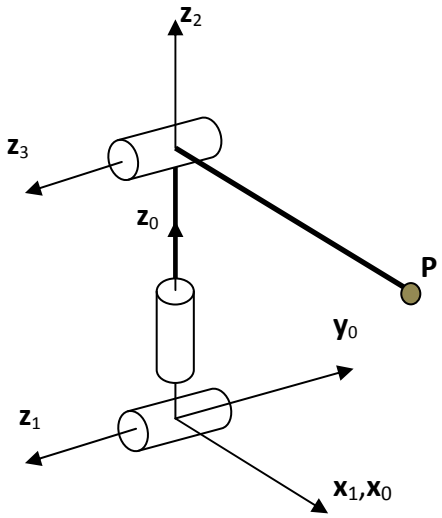
In this case, because of the simplicity in geometric structure, the 4 distinct IGM solutions of ORTH robot result into two groups with equivalent postures : (**Q**1,**Q**4) and (**Q**2,**Q**3).

ORTH robot – Type C (Horizontal mode)

j	σ	α	d	θ	r
1	0	**90^0**	0	θ_1	0
2	1	-90^0	d_2	θ_2	r_2
3	2	90^0	d_3	θ_3	r_3

Table 2.2. Geometric parameters of ORTH_HOR robot

The only parameter changed is $\alpha_1 = 90^0$



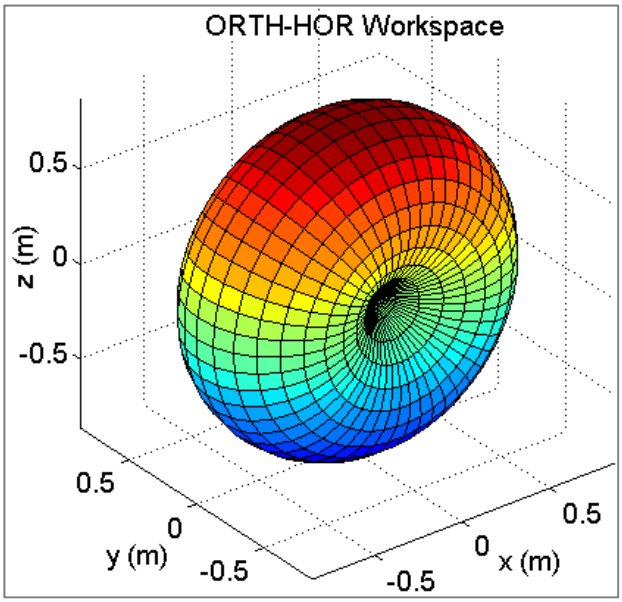


Fig. 2.10. ORTH workspace (Horizontal mode)

Direct Geometric :

$$(\mathbf{P})\begin{cases} x = \left[d_2 + \left(d_3 + d_4 C3\right)C2 + r_3 S2\right]C1 - \left(r_2 + d_4 S_3\right)S1 \\ z = \left[d_2 + \left(d_3 + d_4 C3\right)C2 + r_3 S2\right]S1 + \left(r_2 + d_4 S_3\right)C1 \\ y = -r_3 C2 + \left(d_3 + d_4 C3\right)S2 \end{cases} \tag{2.5}$$

$$\rightarrow \begin{cases} x = d_4 C3C2C1 - d_4\left(1 + S3\right)S1 \\ z = d_4 C3C2S1 - d_4\left(1 + S3\right)C1 \\ y = d_4 C3S2 \end{cases}$$

We can convert all the coordinates in Horizontal mode to Vertical mode:

$$\begin{cases} x_H = x_V \\ y_H = -z_V \\ z_H = y_V \end{cases} \tag{2.6}$$

Hence all the solutions of the inverse geometric can be easy obtained from the Vertical mode solutions!

PUMA robot

j	σ	α	d	θ	r
1	0	0	0	θ_1	0
2	1	-90^0	0	θ_2	r_2
3	2	0	d_3	θ_3	0

Table 2.3. Geometric parameters of PUMA robot

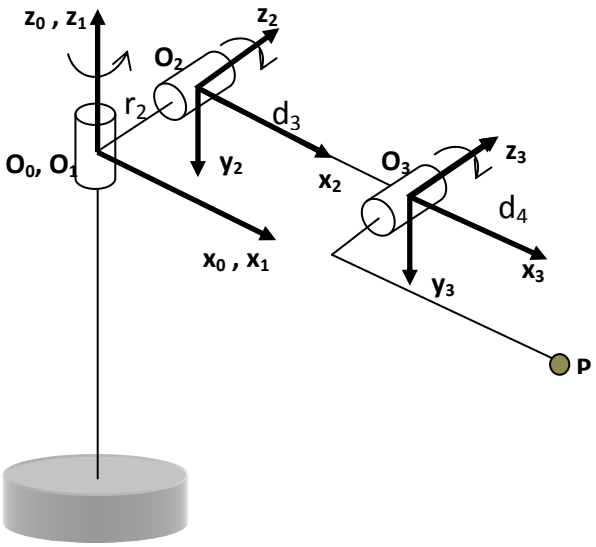


Fig. 2.11. Geometric of PUMA

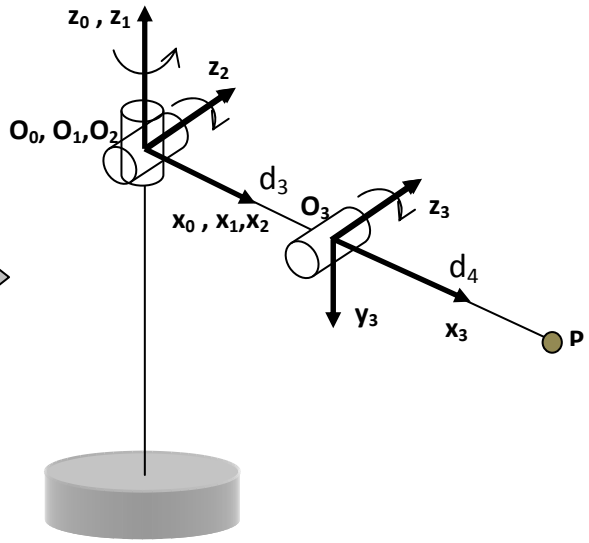


Fig. 2.12. Case d2=d3=r3=r2=0; d3=d4

Params: r2=d2=d3=r3=r4=0

d3 = d4 = 0.433 m

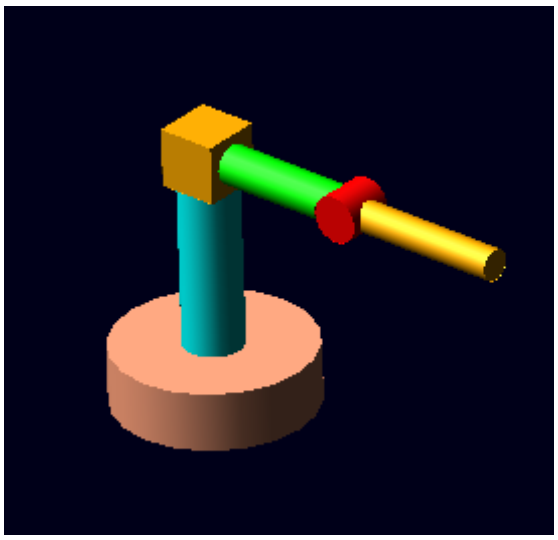
In this case the dexterous workspace index EDR = 0.6

Link mass:

M1 = 4.43 kg

M2 = 10.2 kg

M3 = 4.8 kg



The workspace of the PUMA robot in this case is a sphere with radius R=2*d4 , compare to the orthogonal they have very closed volume along (x,y) plane.

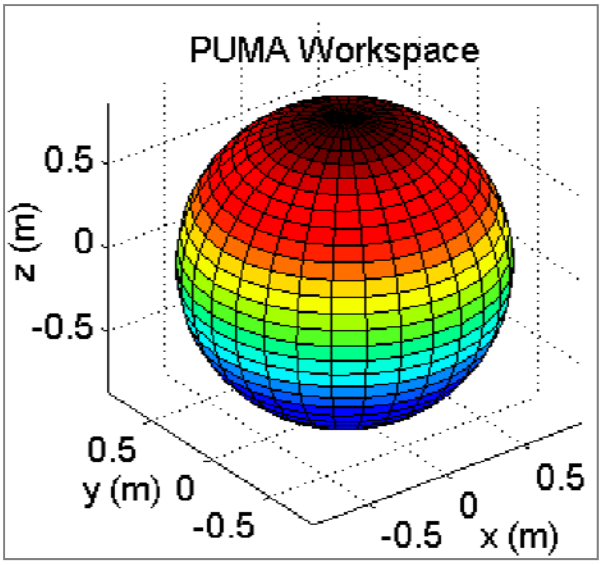


Fig. 2.13. PUMA workspace

Direct Geometric:

$$\begin{cases} x = \left[d_3 C2 + d_4 C23\right] C1 - \left(r_2 + r_4\right) S1 \\ y = \left[d_3 C2 + d_4 C23\right] S1 + \left(r_2 + r_4\right) C1 \\ z = -d_3 S2 - d_4 S23 \end{cases} \tag{2.7}$$

$$S23 = \sin\left(q_2 + q_3\right), \ldots$$

$$\rightarrow \begin{cases} x = d_4 \left[C2 + C23\right] C1 \\ y = d_4 \left[C2 + C23\right] S1 \\ z = -d_4 \left(S2 + S23\right) \end{cases}$$

Inverse Geometric (*2 solutions*):

$$\begin{cases} q_1 = \text{atan2}\left(\dfrac{y}{x}\right) & \rightarrow 2 \text{ solutions} \\ q_3 = \text{asin}\left(\dfrac{x^2 + y^2 + z^2}{2d_4^2} - 1\right) & \rightarrow 2 \text{ solutions} \\ q_2 = \text{atan2}\left(\dfrac{-zC1\left(1 + C3\right) - xS3}{-zC1S3 + x\left(1 + C3\right)}\right) \end{cases} \tag{2.8}$$

In this case, PUMA robot has also 4 inverse geometric solutions and results into two groups of equivalent postures.

For easily notation later, we will indicate the two groups of equivalent postures of each robot as following: **1st + 2nd IGM ORTH** and **1st + 2nd IGM PUMA** (here, 1st group of PUMA or ORTH in Horizontal mode has the elbow-up posture, 2nd group has the elbow-down posture).

Analytical models

In general, we all know that finding inverse solutions for serial robots analytically is not always trivial. However, to be able to perform the analysis well, we need exact models for all the simulation steps. Using numerical methods to compute the joint positions, joint velocities and joint accelerations (also positions, velocities, accelerations in Cartesian coordinates) may give unexpected or bad results. And not only for the modeling part, all the methods which will be used later in *chapter 3* will also give analytical solutions. We can summarize all the models as following:

- **Dynamic models**

$$\begin{aligned} [\boldsymbol{\Gamma}, \mathbf{F}] &= \mathbf{IDM}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \quad \text{(Inverse Dynamic Model)} \\ \ddot{\mathbf{q}} &= \mathbf{DDM}(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\Gamma}) \quad \text{(Direct Dynamic Model)} \end{aligned} \tag{2.9}$$

- **Geometric models**

$$\begin{aligned} [\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}] &= \mathbf{IGM}(\mathbf{P}, \dot{\mathbf{P}}, \ddot{\mathbf{P}}) \quad \text{(Inverse Geometric Model)} \\ [\mathbf{P}, \dot{\mathbf{P}}, \ddot{\mathbf{P}}] &= \mathbf{DGM}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \quad \text{(Direct Geometric Model)} \end{aligned} \tag{2.10}$$

Where

$$\begin{cases} \boldsymbol{\Gamma}: \text{ joint actuator torques} \\ \mathbf{F}: \text{ reaction forces} \quad \text{(see the appendix for how to compute these vectors)} \\ \mathbf{q}: \text{joint positions} \\ \dot{\mathbf{q}}: \text{joint velocities} \\ \ddot{\mathbf{q}}: \text{joint accelerations} \\ \mathbf{P}(x, y, z): \text{Cartesian coordinate of the end-effector} \\ \dot{\mathbf{P}}(\dot{x}, \dot{y}, \dot{z}): \text{velocity of the end-effector} \\ \ddot{\mathbf{P}}(\ddot{x}, \ddot{y}, \ddot{z}): \text{acceleration of the end-effector} \end{cases}$$

CHAPTER 3

3.1. Static analysis

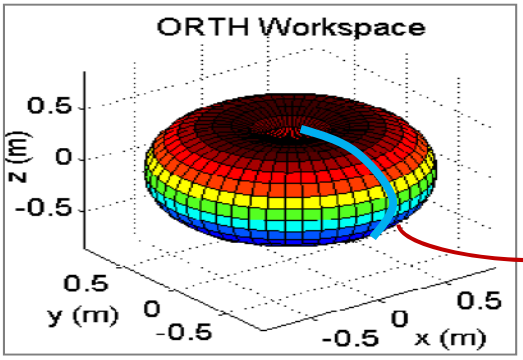
Static analysis is a special case of dynamic analysis where we compute the actuator torques from the Inverse Dynamic Models (IDM) while suppressing all the dynamic terms (joint velocities and joint accelerations). From the computed static torques profiles we will see some basic relations between the joint torques of each manipulator and also the distributing rate of the torques through all the working space. This is one important basic dynamic characteristic. The obtained profiles then can be used for dynamic analysis and optimization procedure in *chapter 4*.

We try to compute the static torques through all the workspace of the two robots. The data can be generated by several ways to cover the whole area of a cross-section:

- Given coordinates of the end-point **P**(x,y,z) >> Inverse solutions >> Results
- Set joint velocities & joint accelerations to zeros ($\dot{\mathbf{q}} = \mathbf{0}, \ddot{\mathbf{q}} = \mathbf{0}$)

3.1.1. *Vertical ORTH vs PUMA*

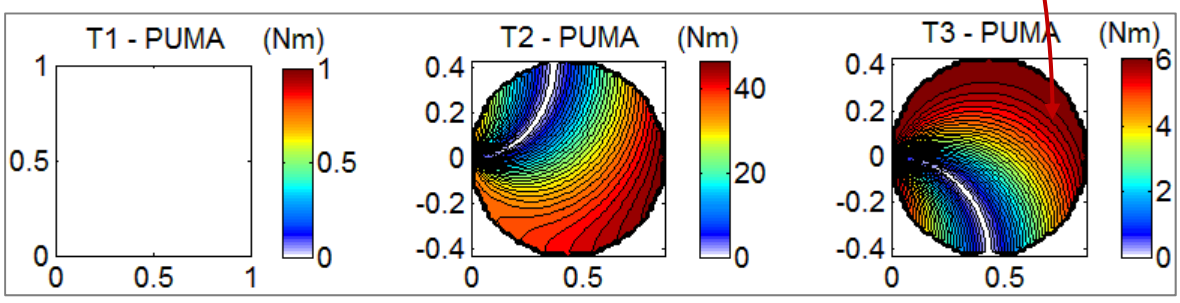
In this case, because of the symmetric properties we only need to consider a cross-section of the ORTH robot workspace (a circle with radius R=D4).



PUMA robot: (here we fix y= 0, vary on (x,z) plane**)**

PUMA - Case 1: Mass(object) = 0kg

At a solution (1st IGM):



Minimum and Maximum static torques:

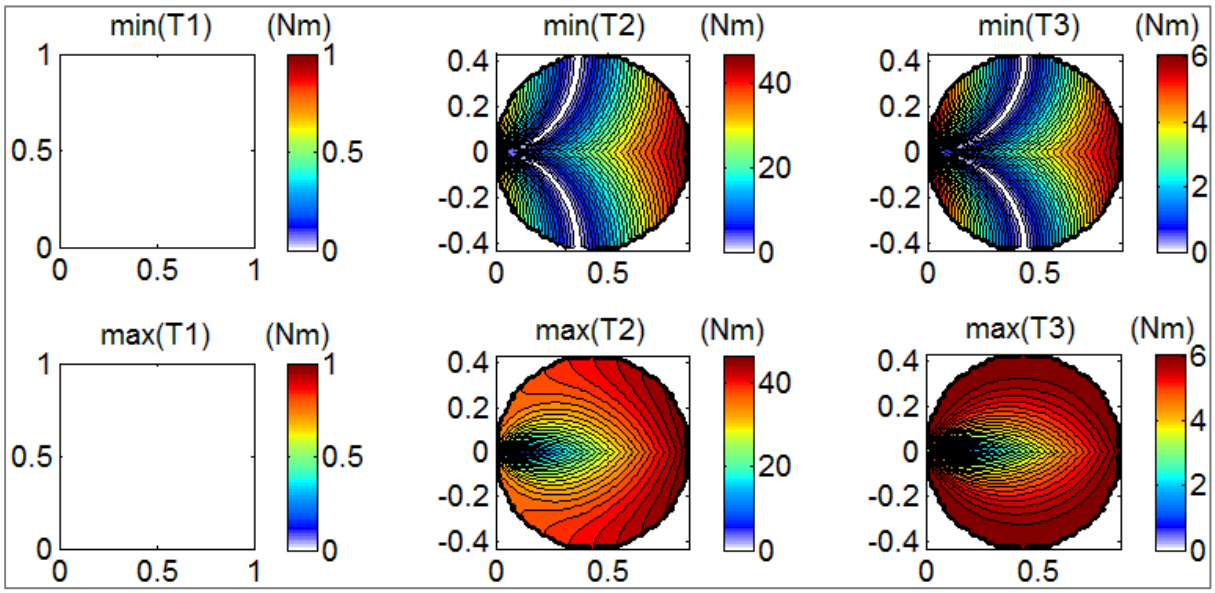


Fig. 3.1. Static torques of PUMA, Mass(object) = 0kg

PUMA - Case 2: Mass(Object) = 5kg

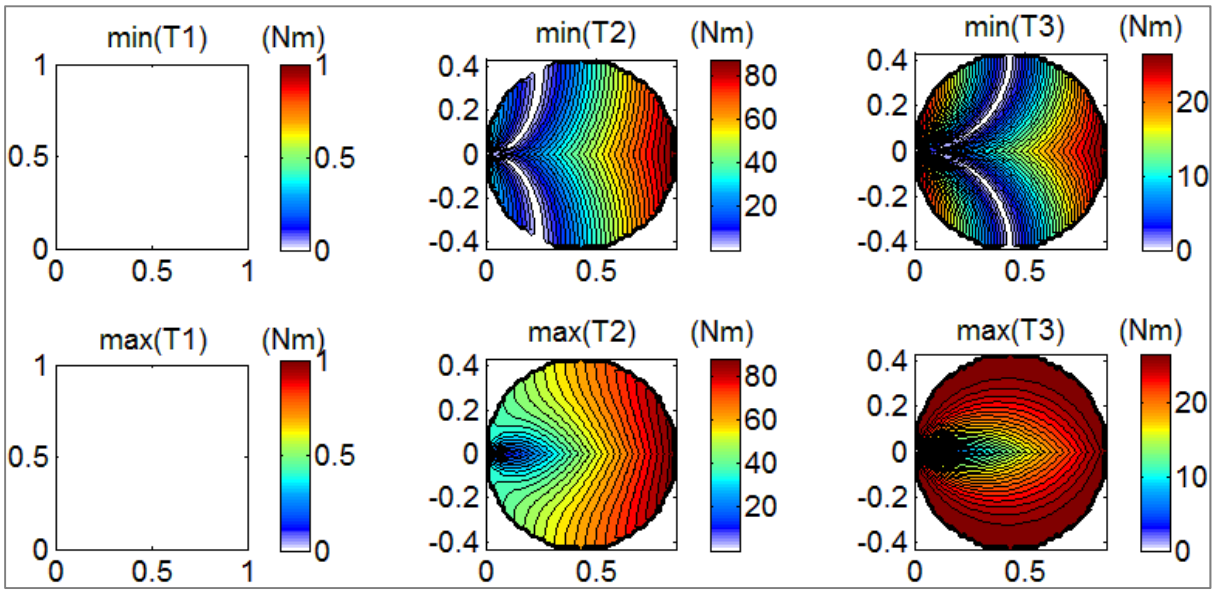
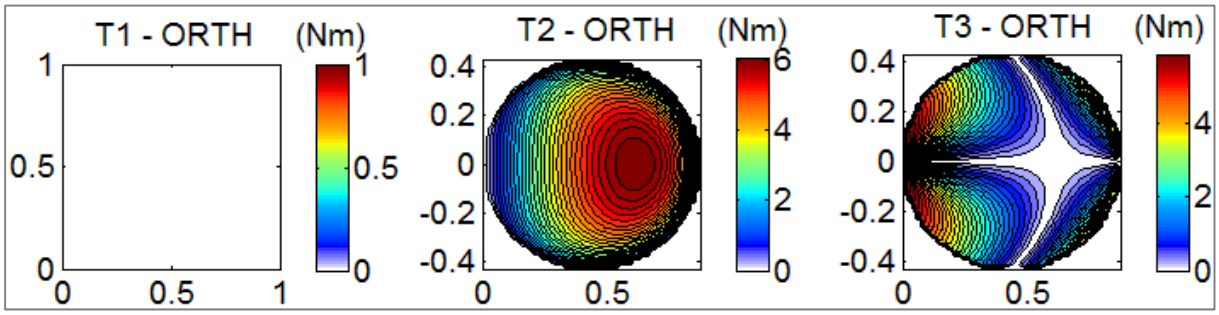


Fig. 3.2. Static torques of PUMA, Mass(object) = 5kg

ORTH_VER robot: (here we fix x= 0, vary on (y,z) plane)

ORTH - Case 1: Mass(object) = 0kg

*At a solution – 1st IGM (all 4 solutions give quite similar results)***:**



Minimum and Maximum static torques:

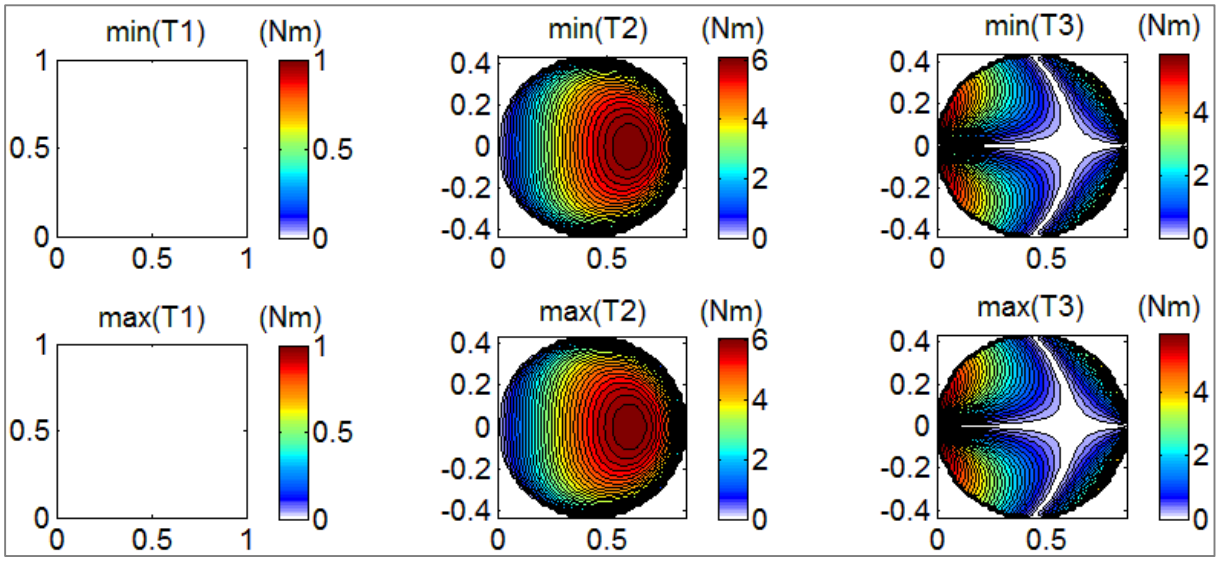


Fig. 3.3. Static torques of ORTH_VER, Mass(object) = 0kg

ORTH - Case 2: Mass(object) = 5kg

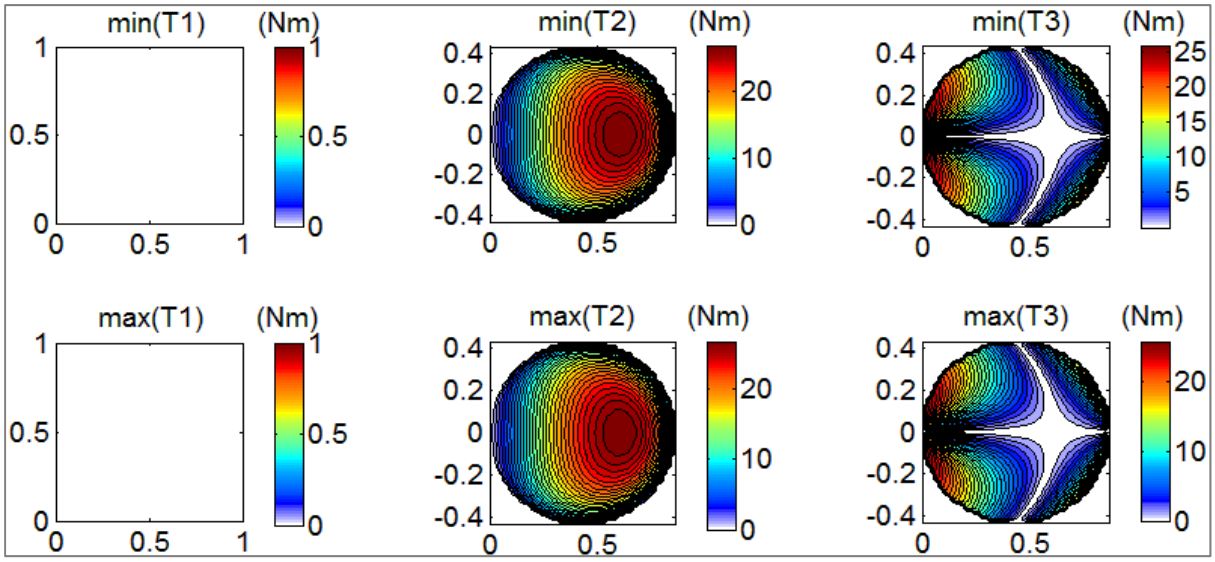


Fig. 3.4. Static torques of ORTH_VER, Mass(object) = 5kg

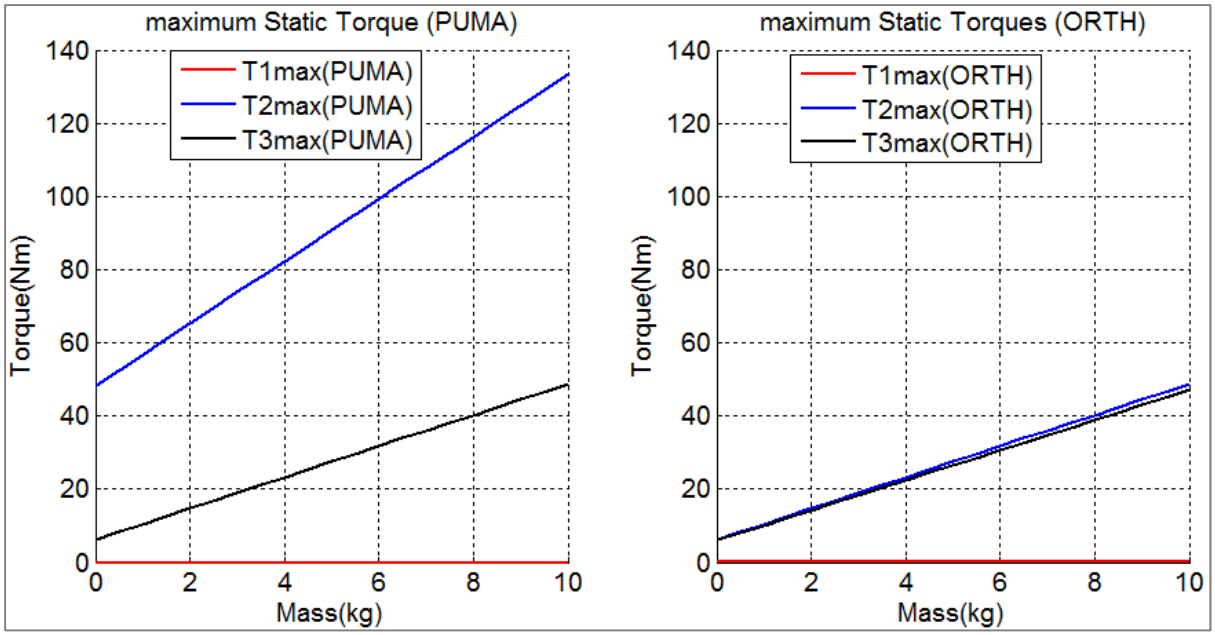


Fig. 3.5. Comparison of static torques, ORTH_VER vs PUMA

From these results, we can tell:

- The Vertical ORTH robot (ORTH_VER) has quite a symmetric torques profile (not so different at maximum, minimum or at a certain set of solution). This is expected if we look at the two postures of the robots as in figure 3.6.
- The static torques in ORTH_VER cases are distributed quite equally for actuators of joint 2 and joint 3. Meanwhile, in PUMA cases, the actuator torques on joint 2 are much larger. It's reasonable because of its geometric properties.
  → ORTH_VER *robot can handle more payloads than PUMA !?*
- Changing the object's mass only affect the magnitude of actuator torques. The proportion in distributing torques on each joint is still maintained.
- In static, the ORTH_VER robot shows better results than the PUMA one.

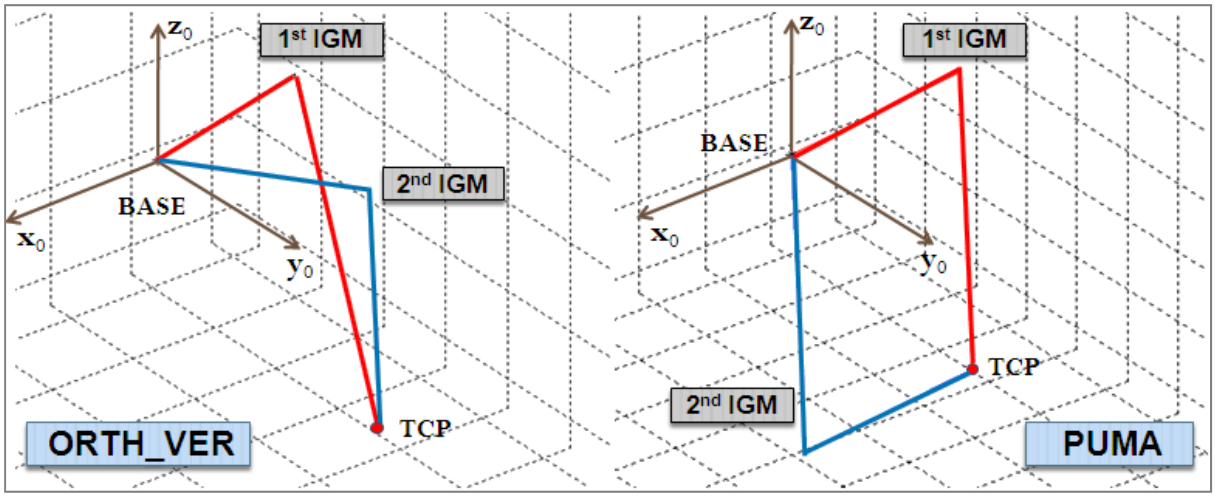


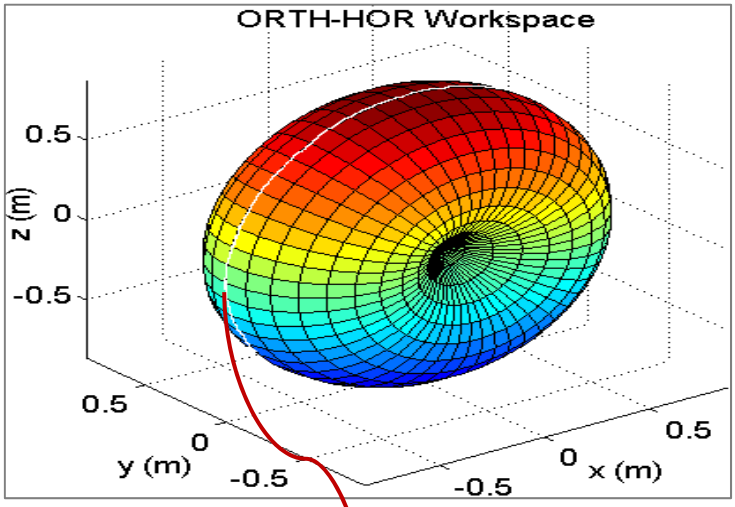
Fig. 3.6. Posture of ORTH_VER and PUMA at an example point in the workspace

3.1.2. *Horizontal ORTH vs PUMA*

We can note that, while changing the mode of the ORTH robot to Horizontal, we lost the symmetric property of the workspace with respects to the gravity. ***The workspace now is a torus but rotating around the y-axis of the world frame.*** For this reason, we cannot analyze at each cross-section anymore. One way to do is take a look at each layer parallel to the ($\mathbf{x}_0$,$\mathbf{z}_0$) plane of the original frame while vary the coordinate y = const.

The second notice is that, now the role of the 1st joint and the 3rd joint of the ORTH robot (at q_2=const) is similar to the second joint and third joint of PUMA, of course with respect to the gravity direction.

We will show some results of static torques at layers with $y_P = 0,\ \frac{2}{5}d_4\ ,\ \frac{4}{5}d_4, \ldots$



Case 1: y_P = *0* | Mass(object) = 5kg

PUMA:

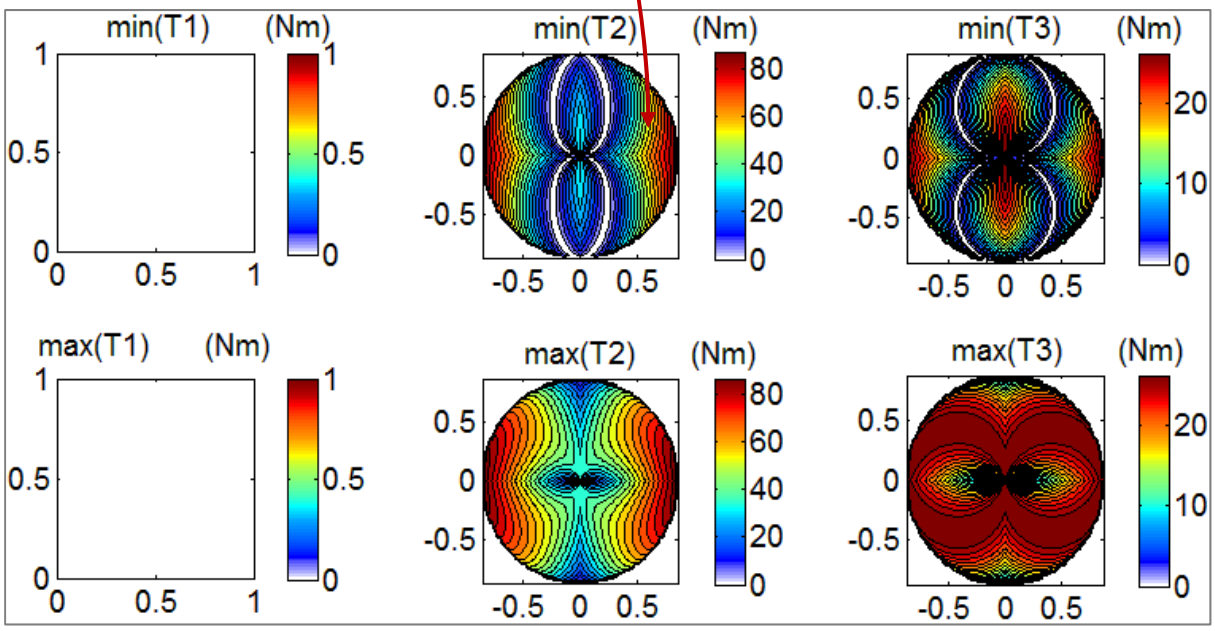


Fig. 3.7. Static torques of PUMA, Mass(object) = 5kg, y_p = 0

ORTH_HOR:

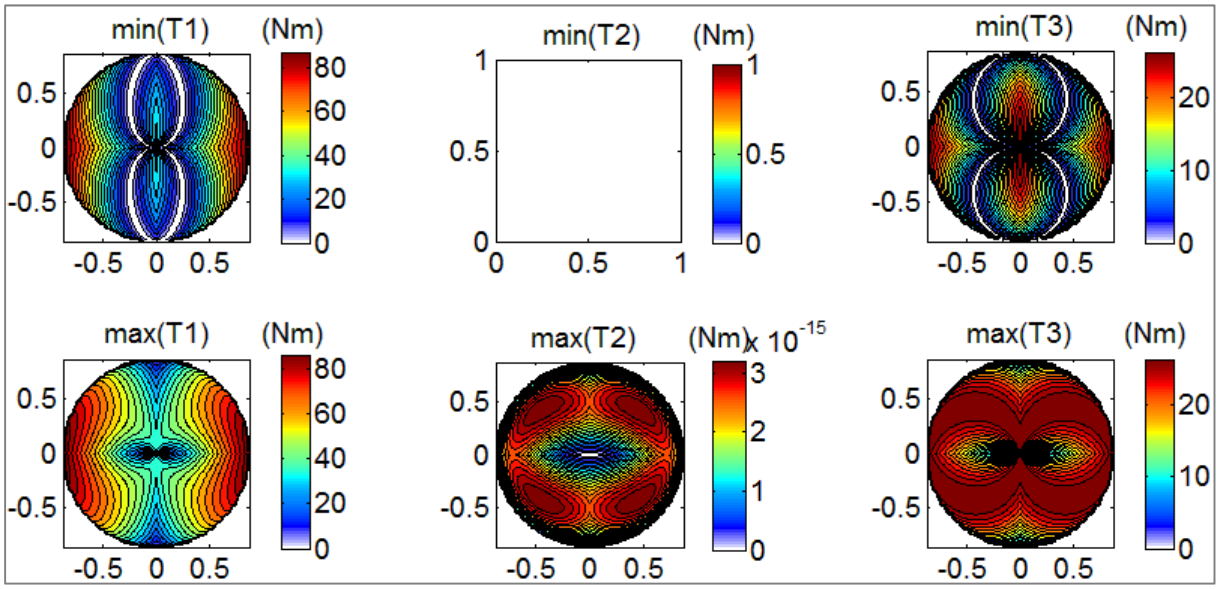


Fig. 3.8. Static torques of ORTH_HOR, Mass(object) = 5kg, $y_p = 0$

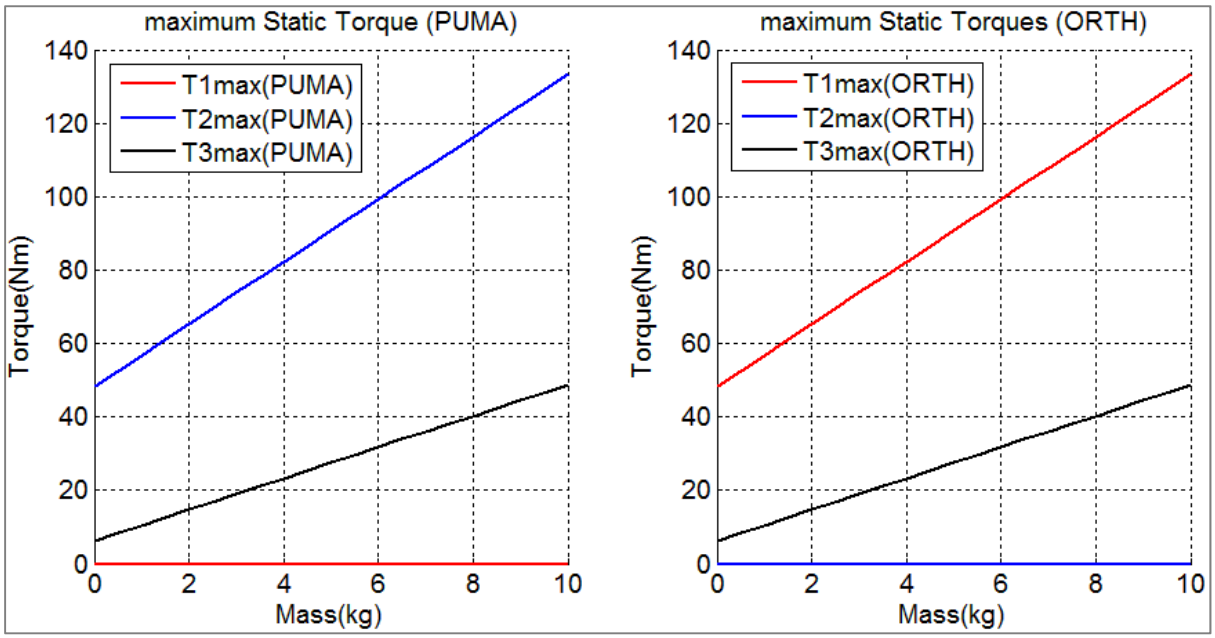


Fig. 3.9. Comparison of maximum static torques, ORTH_HOR vs PUMA, case $y_p = 0$

- In this case, we see that the results are more or less the same for both robots (expected)
- Changing the payload means only changing the magnitude of the joint torques, the same as previous tested (with the Vertical ORTH).

Case 2: $y_P = \frac{7}{8} d_4$

PUMA:

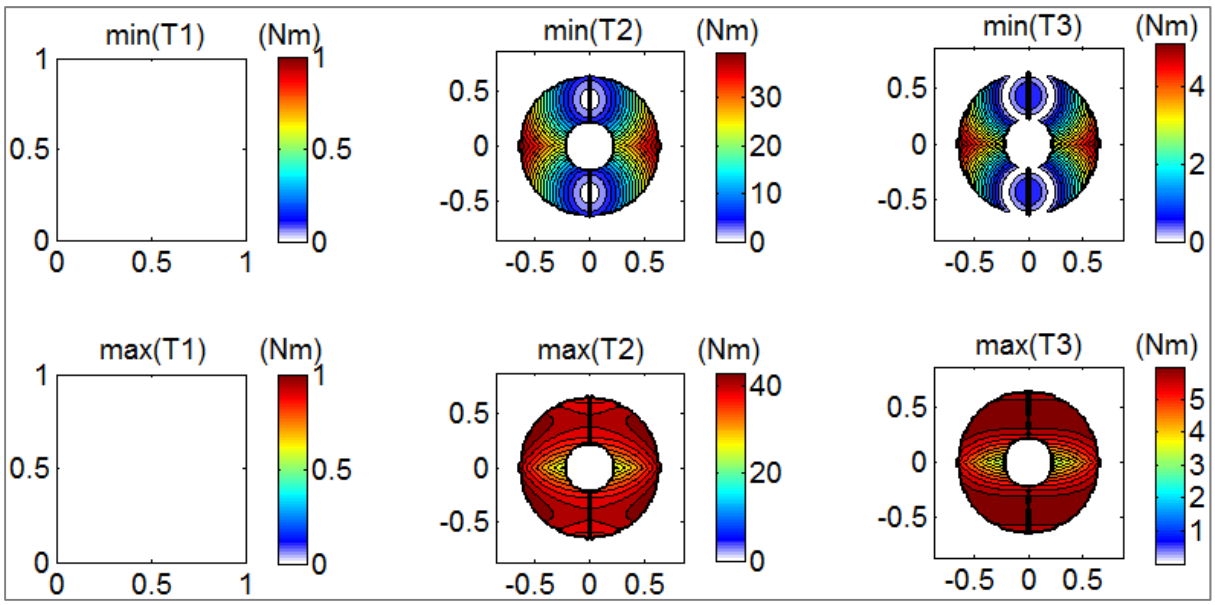


Fig. 3.10. Static torques of PUMA, Mass(object) = 0kg, $y_P = 7/8*d_4$

ORTH_HOR:

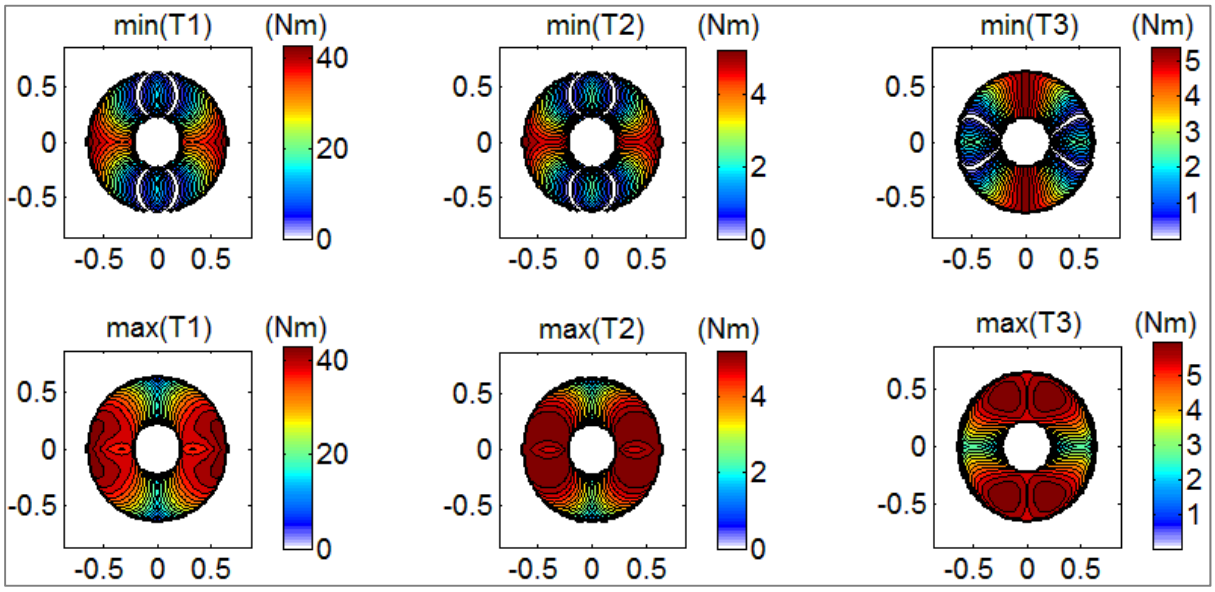


Fig. 3.11. Static torques of ORTH_HOR, Mass(object) = 0kg, $y_P = 7/8*d_4$

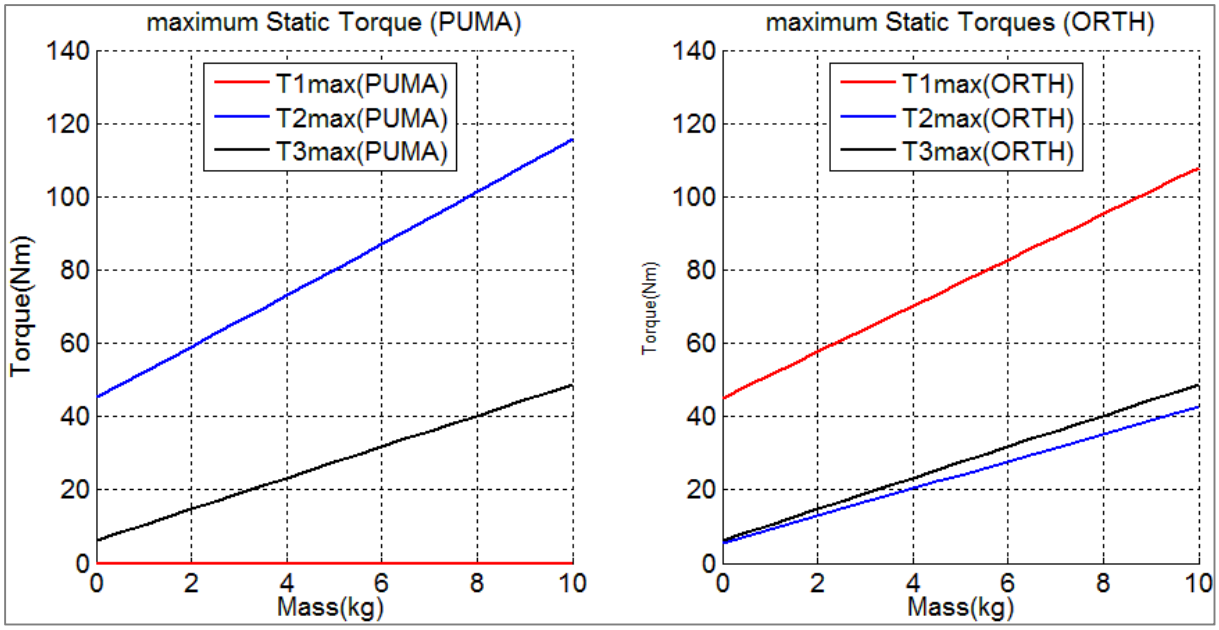


Fig. 3.12. Comparison of maximum static torques, ORTH_HOR vs PUMA, case $y_p = 7/8*d_4$

Case 3: $y_P = \frac{19}{20} d_4$

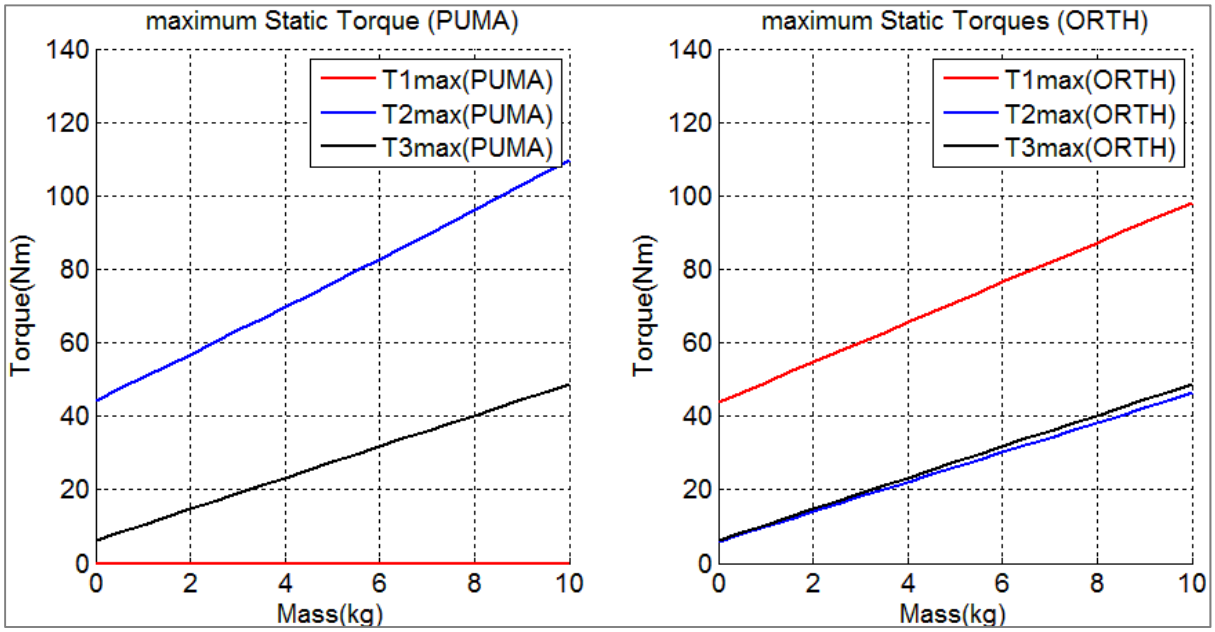


Fig. 3.13. Comparison of maximum static torques, ORTH_HOR vs PUMA, case $y_p = 19/20*d_4$

Comment:

- As y_P is increasing, the layer's boundary is reducing. When y_P is small the profile of ORTH_HOR is close to PUMA, when y_P is large the profile of ORTH_HOR tend to be similar to ORTH_VER.
- In general, both robots show similar results in term of maximum join torques, especially T1max(ORTH) ~ T2max(PUMA) since they have similar role.

3.2. Dynamic analysis

Let state again the question, ***how can we tell which results will show better performance in the analysis***? In Pick-n-Place, for a certain task, a robot performs better if it can complete the task within expected speed and consume less power (or less torques) than the other. For rigid serial manipulators, one can tell the driving motors are the hearts of the system. It relates directly to the cost to build the robot. In some cases, we will consider the maximum capability of each joint in the worst cases (maximum values) to determine the performance.

Because in case of vertical mode, ORTH robot has two equivalent postures which are both elbow-up (fig. 3.6) and the PUMA robot also has one elbow-up posture (1st IGM) hence we will consider the comparison results between (1st IGM of ORTH_VER) vs (1st IGM of PUMA).

3.2.1. *Assumptions*

In dynamic performance analysis, our idea is to generate some trajectories that could give us as much information as possible of the performance (maximum torques, joint accelerations..).

We will try to test through several methods and several types of trajectories.

Set up the same frame work:

- The analysis will be performed under the same types of trajectory, same payloads and the same operational velocity V = *const* (magnitude of TCP's velocity in the task space). We will vary the payloads at the end-effector and V and compare the results between two manipulators.
- The traveling time between any two points A and B in the task space is known. Means that, we can compute the time by knowing the operational velocity of the TCP in the task space:

$$t = \frac{\mathrm{AB}}{\mathrm{V}}$$

- Assume that, the robot can perform under ideal condition: keep track of the trajectory perfectly. Means that, we don't consider any control system involving with the analysis. Hence, all the terms will be computed with desired joint positions, velocities and accelerations $\left(\mathbf{q}^d, \dot{\mathbf{q}}^d, \ddot{\mathbf{q}}^d\right)$ generated by the certain methods.
- In the dynamic models, we don't consider the effect of frictions (FV, FS)
- The two robots both have the links with same mass, length. The payload will be put at the end-tip point P (TCP). For simply calculating, we assumed it is a steel sphere with density =7801kg/m^3.

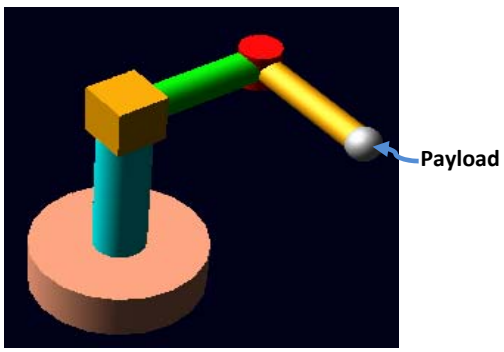


Fig. 3.14. Example of ORTH robot with payload

Trajectory Generation

Following we list some common methods that generate a trajectory for the end-effector.

In the joint-space:

- Using 5th degree polynomial (Point-to-Point) [32]
- Using Trapezoidal velocity model (Point-to-Point) [32]
- Using (Smoothed) trapezoidal velocity model (Via-Points) [32]
- Using Cubic spline function (Via-Points) [32]

In the task-space:

- Using Linear interpolations (Point-to-Point)
- Using 5th degree polynomial (Point-to-Point)
- Using Cubic spline functions (Via-Points) [32]
- Using 2D clothoid arcs (Via-Points) [30]
- Using 3D discrete clothoid arcs (Via-Points) [31]

We have tested all the methods in the analysis with analytical formula, except for the last one (using 3D discrete Clothoid arcs) because this method only gives us numerical solutions and it will take quite amount of time to fully understand and apply the theory.

Type of trajectories:

- **Pick-n-Place task : standard trajectory**
  We test with a standard type of trajectory in industry:

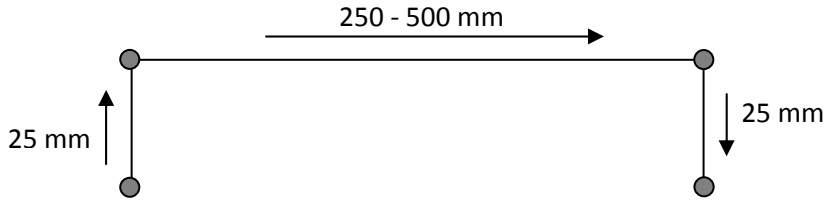


Fig. 3.15. Standard Pick-n-Place trajectory

  Here we chose to test with profile $(25 \times 300 \times 25\ mm)$

  Note that, the fastest Pick-n-Place performance currently is around **(80 – 120) picks/min** or with operational velocity **Vs = (0.9 – 1.4) m/s** (along the linear direction). In simulation, in each case when we choose the Vs, we will select Vs along a straight line Point-to-Point (this is equal to we fix the travelling time between any two via-points along the trajectory). Hence, the actual velocity of the end-effector along one segment of the trajectory may be less or greater than the reference Vs.
- **Optimized trajectory**
  We will verify the performance with some excited trajectories. This will be detailed later in **chapter 4**.

3.2.2. *First result : Basic characteristics*

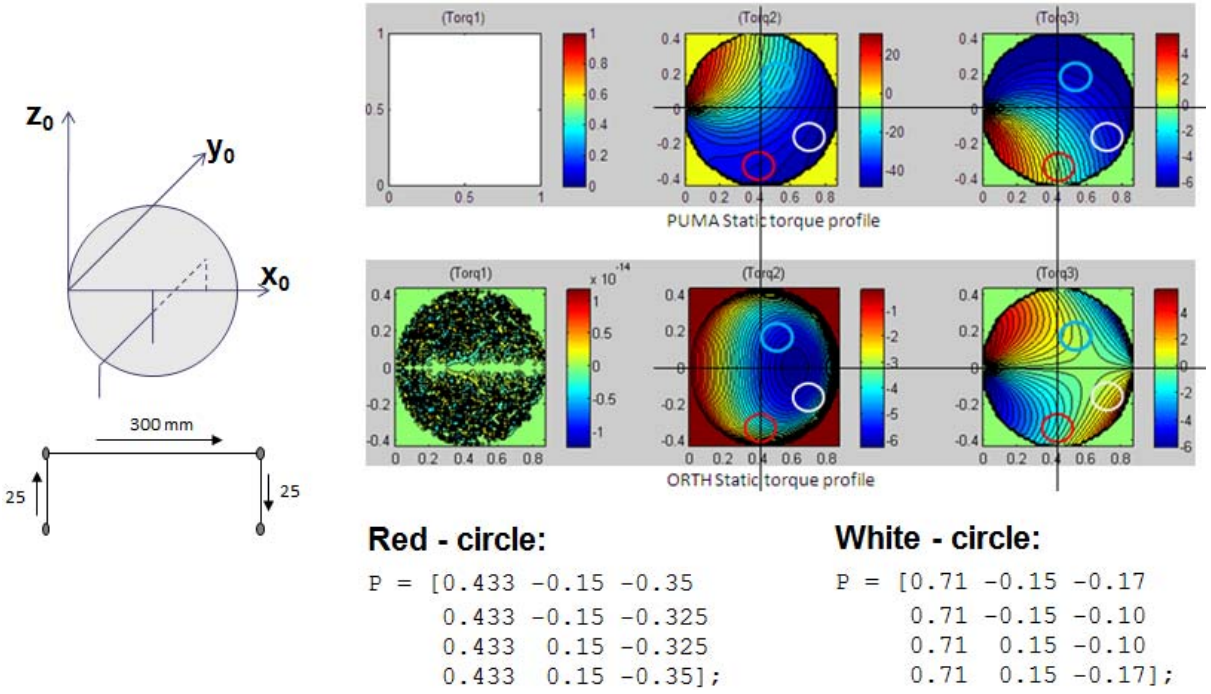


Fig. 3.16. Testing areas in ORTH_VER workspace

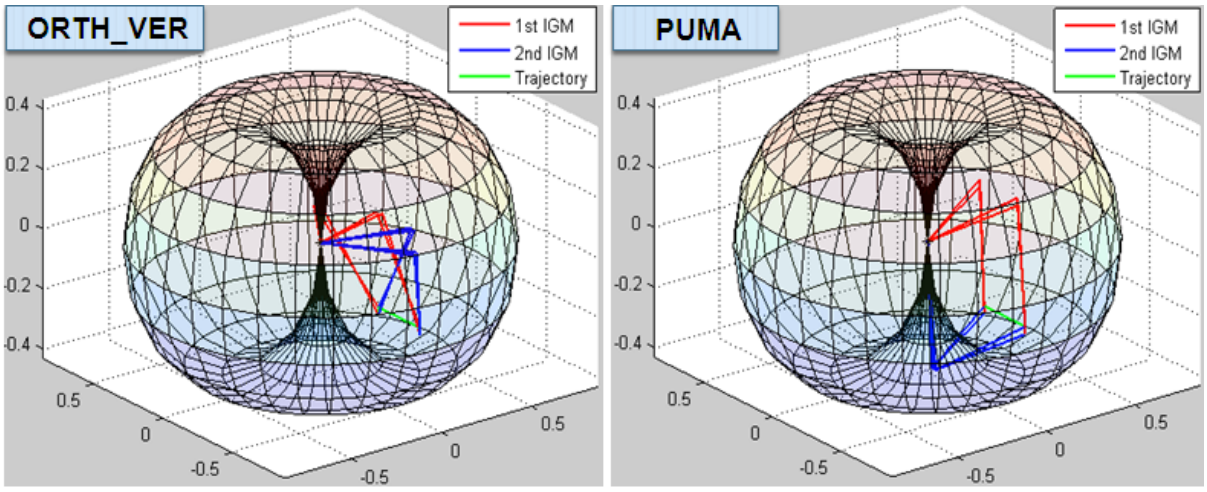


Fig. 3.17. Postures of two manipulators ORTH_VER and PUMA in case of standard Pick-n-Place trajectory

In this test:

- We verify with different Vs and the payload is varied in (0kg, 5kg, 10kg).
- Test with several areas in the cross-section workspace that have exciting properties like high rate of distributing torques (based on the static torques profiles).
- The trajectories are parallel with the plane (y_0, z_0) of the base frame (fig. 3.16).
  →The trajectories are orthogonal to the workspace cross-sections.

The reason we choose the standard type trajectory: assume that we want to analysis the performance of the robots within their regular working areas. In practical, the trajectory for industrial robot in Pick-n-Place task mainly like as the standard type which consists of basic directions and segment: vertical direction, horizontal direction, and blending segment:

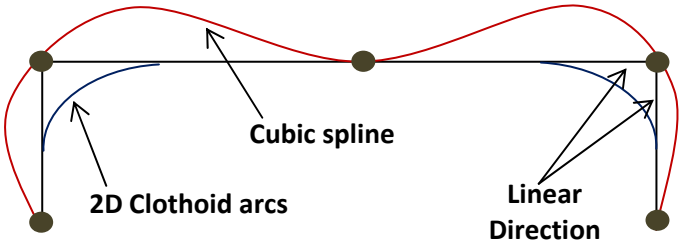


Fig. 3.18. Common Pick-n-Place trajectories

In this analysis step, we exposed the basic dynamic properties of each manipulator and try to find out: **are the basic dynamic characteristics of each robot maintained within different working areas with respect to different effects of (Vs, Mass) or they change?**

These tests may be not sufficient in comparison, but they help us to verify some basic properties of each manipulator (could be useful for the design phase or in control) and also help us to select good trajectory generating methods for later procedures.

Following are our summary on the basic characteristics of the two manipulators (please look at the *appendix A2* for further detail examples).

Comment on ORTH-VER vs PUMA

- The results in case of white-circle area (on the effects by changing Vs or Payload) are similar to the case of red-circle area. ***Different areas give us different profiles, but the basic relations are still maintained*** **:**
  Varying Vs along the z-axis direction (along gravity) will cause great effect on the joint torque 1, *this is much more sensitive in case of ORTH robot.*
  Varying the payload will greatly affect the joint torque 2 of both robots, *but will affect greater and much more sensitive in case of PUMA robot*. This is similar to static analysis.
- ***The dynamic performance of the manipulators depends strongly on the type of trajectory generation methods***. In the joint space, ORTH robot performs better than PUMA. But in the task space, the performance of PUMA robot is more stable than the ORTH robot especially in the view of the effect on joint 1.
- ***The dynamic performance of the manipulators also depends on different segment of the workspace. But the basic relations are still maintained.***
- ***The only property in general that the Vertical ORTH robot is better than PUMA is the proportion of distributing torques between joint 2 and joint 3 with respect to the payload.***

Comment on ORTH-HOR vs PUMA

- *The basic performance of the ORTH robot in horizontal mode is quite similar to the PUMA, since the role of joint 1,3 of ORTH is similar to the role of joint 2,3 of PUMA. The results are very closed at layers with y_P small.*
- *When increasing y_p, the basic performance of the two robots is varied depends on each segment in the workspace. The proportion of distributing torques between joint 2,3 of ORTH still quite balance like the case of vertical mode.*

In basic analysis, it's hard to tell which manipulator performs better. **In term of the effect on joint torque 1, PUMA is better than ORTH; but in term of joint torques 2,3 ORTH is better than PUMA!**

Following we summarize the usage of several trajectory generating methods in all analysis steps. They can be used in a certain analysis step but not always helpful, i.e. some methods are good for indicating basic characteristics but not useful in indicating EWA or in optimization procedure…

Trajectory Generating Method	Type	Usage
Linear interpolations	Point-to-Point (task-space)	Useful for analysis the relations with effects caused by basic motions (vertical or horizontal direction), used in • Basic analysis
5th degree polynomial	Point-to-Point (joint-space / task-space)	Have nice properties both in the joint-space and task-space, most useful in • Basic analysis • Indicating EWA • Optimization
(Smoothed) Trapezoidal velocity model	Point-to-Point (joint-space)	**Not useful**, since the IGM solutions especially at the blending arcs are sensitive (too high accelerations or torques)
Cubic spline function	Via-Points (joint-space / task-space)	Useful for the analysis especially in • Indicating EWA • Optimization
2D clothoid arcs	Via-Points (task-space)	Have nice properties, especially in keeping constant operational velocities along the blending arcs • Basic analysis • Indicating EWA
3D discrete clothoid arcs	Via-Points (task-space)	**Not yet tested**

Table 3.1. Summary on the usage of common trajectory generating methods

3.2.3. *Second result : Indicating EWA*

In this section, we show several results that give us the proof: there exist Efficient Working Areas. Three trajectory generating methods are used (using *5th degree polynomial*, *2D clothoid arcs*, and *cubic spline function*). Verifying the EWA at different areas in the working space (i.e. in the following, we tested at two region red-circle and white-circle) may give us different results but we can take the common areas among them to determine the EWA. We will also see, using certain methods may bring us clear view of EWA while the other methods only give us an estimation.

For rigid body manipulators, the analysis of the reaction forces or joint velocities in this case are not so important (it means we could eliminate the reaction forces and joint velocities from the basic dynamic elements in the analysis formula – *figure 1.11*, *chapter 1, section 1.3.1* – we give a proof for this in **Test 1, figure 3.21**). Thus, the criteria for determining the EWA is the difference of maximum torques of two robots (ORTH_VER vs PUMA) or (ORTH_HOR vs PUMA).

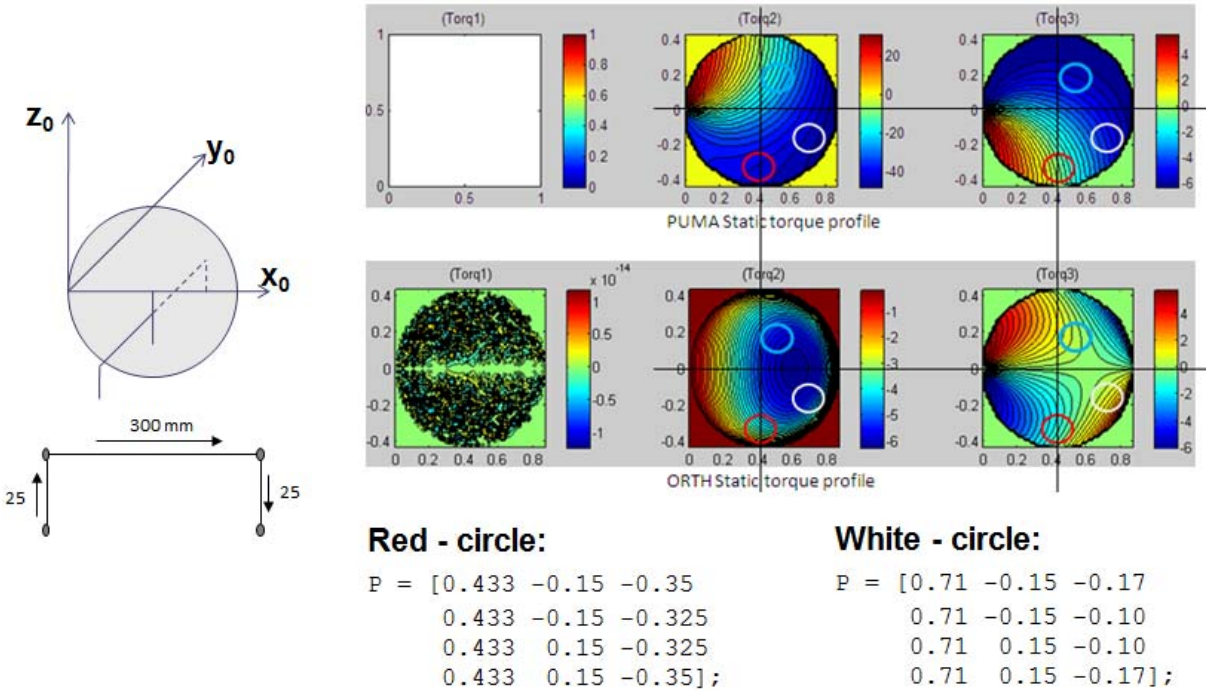


Fig. 3.19. Testing areas

Notice that each segment of a trajectory is composed of three terms: the initial point $\mathbf{P}^{init}$, the final point $\mathbf{P}^{final}$ and the travelling time t_f. One can change the speed of the TCP by varying the value of t_f while keeping the via-points fixed.

Because the torque profiles for each manipulator are complex surfaces (the maximal torque values are varied while changing either the speed Vs or payload), the contours plots of the differences between maximum torque values of the ORTH and the PUMA are presented. In the areas that have the negative values, the ORTH manipulator performs better than the PUMA manipulator.

Test 1: using 5th degree polynomial

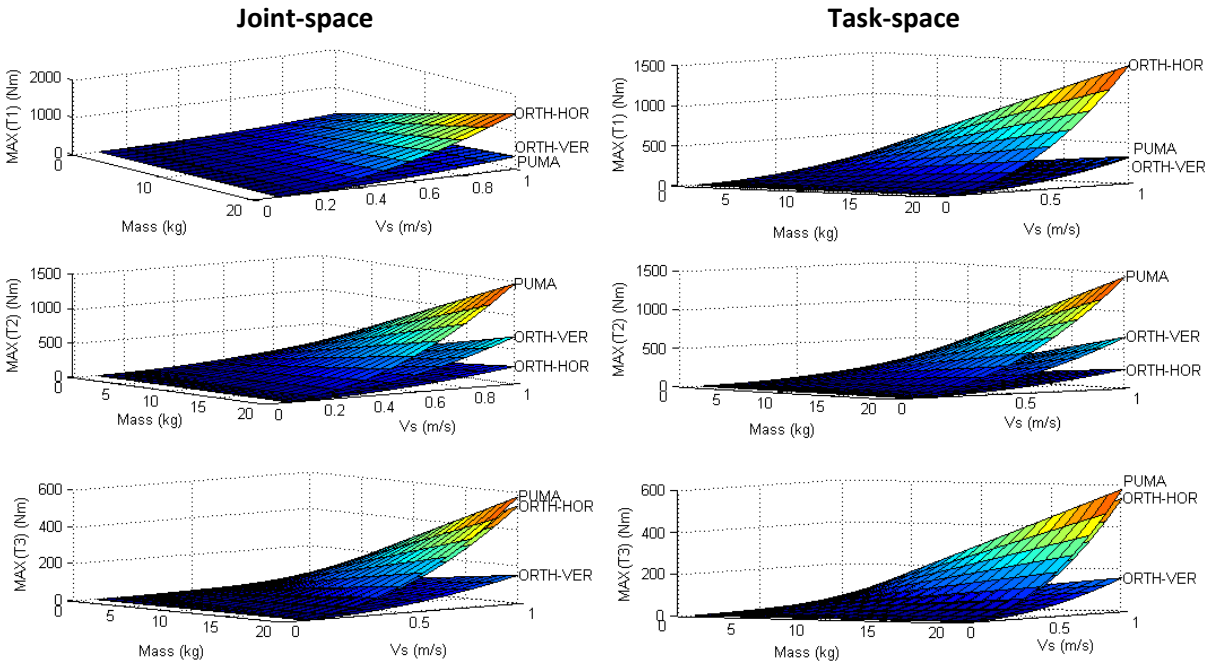


Fig. 3.20. Maximum torques profiles of ORTH vs PUMA in joint-space and task-space using 5th degree polynomial method (in white-circle area)

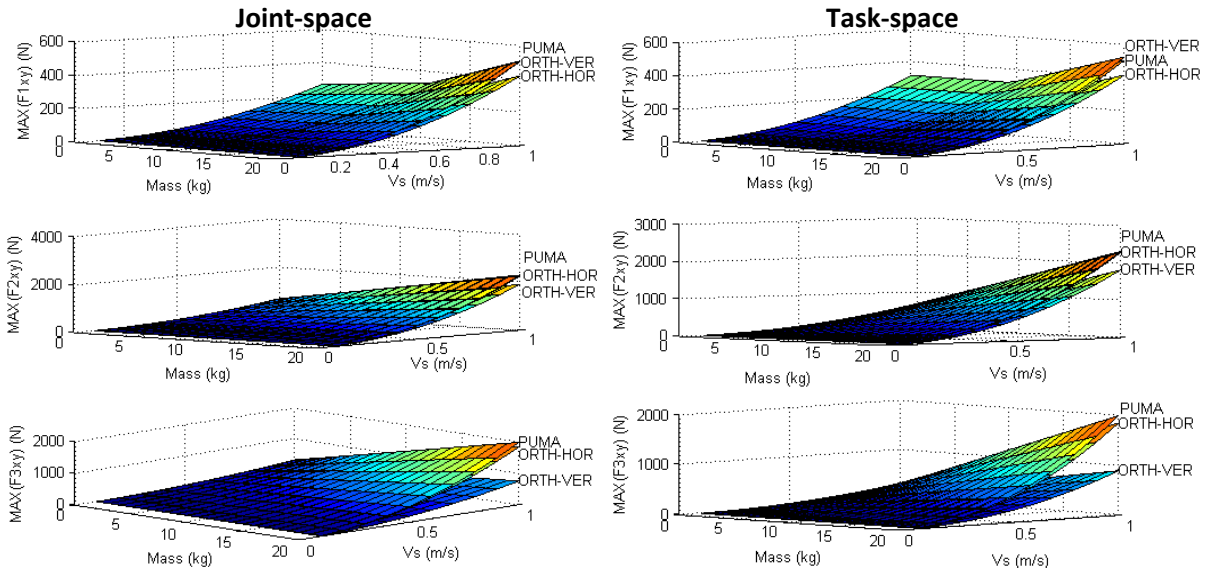


Fig. 3.21. Maximum reaction forces profiles of ORTH vs PUMA in joint-space and task-space using 5th degree polynomial method (in white-circle area)

- Looking at the results (Fig. 3.20), obviously ORTH_VER robot has the best performance if we compare all the joint torques to the maximum values of the **first joint of ORTH_HOR robot** and of the **second joint of PUMA robot**. And ORTH_VER robot has lowest and most balance proportion of distributing torques between 3 joints.
- In figure 3.21, the reaction forces profiles of both robots are quite similar.
  → ***Information on reaction forces could not clearly show the differences, and we can eliminate them from the basic dynamic elements*** (in the formula, figure 1.11).

- In figure 3.22, we can see the trade off in performance of these robots with respect to accelerations.

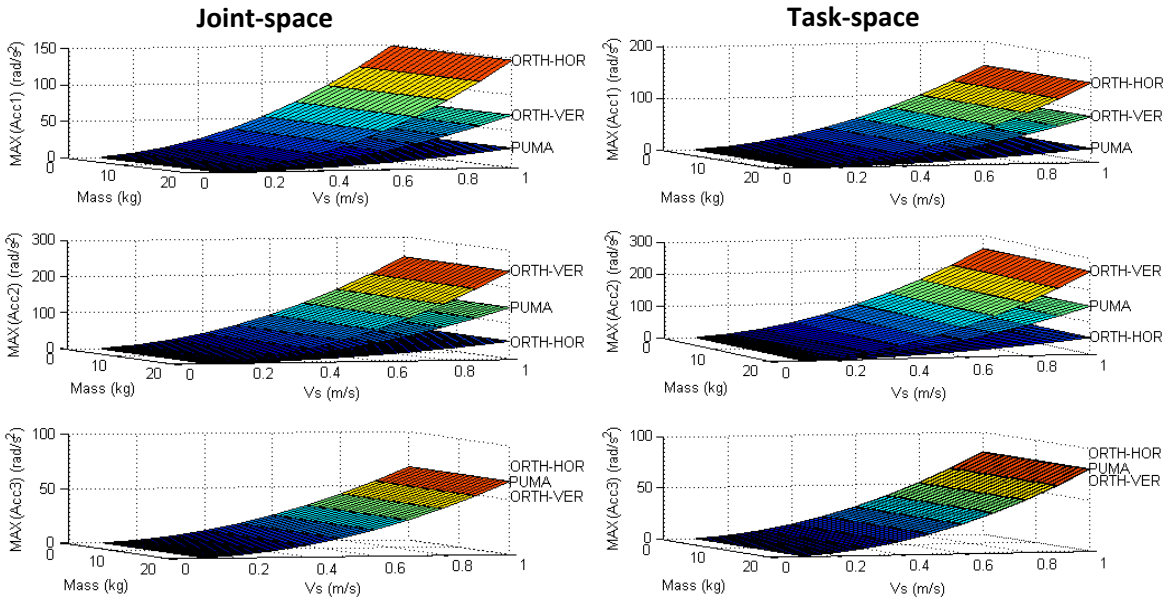


Fig. 3.22. Maximum accelerations profiles of ORTH vs PUMA in joint-space and task-space using 5th degree polynomial method (in white-circle area)

Test 2: using 2D clothoid arcs in the task-space

- Here, we show another results beside the maximum torques that is the proportion between the maximum of joint torques 1,2,3 with respect to the maximum of joint torque 3 of ORTH_VER robot (figure 3.24):

$$\frac{\max\left(T1,T2,T3\right)}{\max\left(T3_{ORTH\_VER}\right)} \tag{3.1}$$

  By doing this, we can have another way to verify the performance. It becomes more clear especially if we want to select the motors for each joint.
- In figure 3.24, we can notice quite a big difference in the maximum accelerations of ORTH_VER robot in the White-Circle area with respect to other robots. This is, in fact, the trade off in performance of this robot in this particular area. (Notice in the area with Vs < 1m/s - The area that has Vs > 1m/s is not practical. For PUMA robot, the fast ones in industrial are allowed to move with speed of the end-effector around 0.5 m/s)

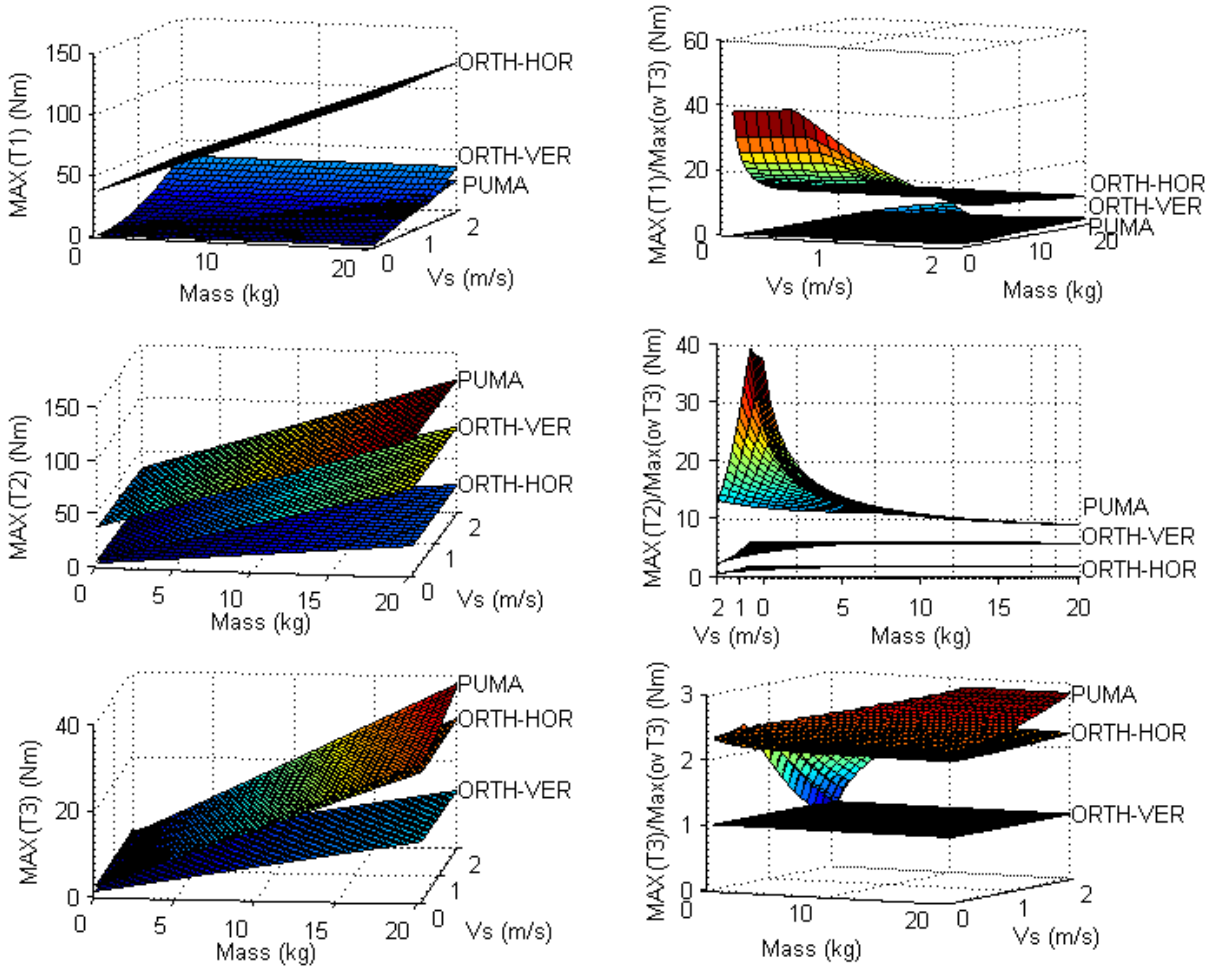


Fig. 3.23. Maximum torques of ORTH vs PUMA in task-space using 2D clothoid arcs (White-Circle area)

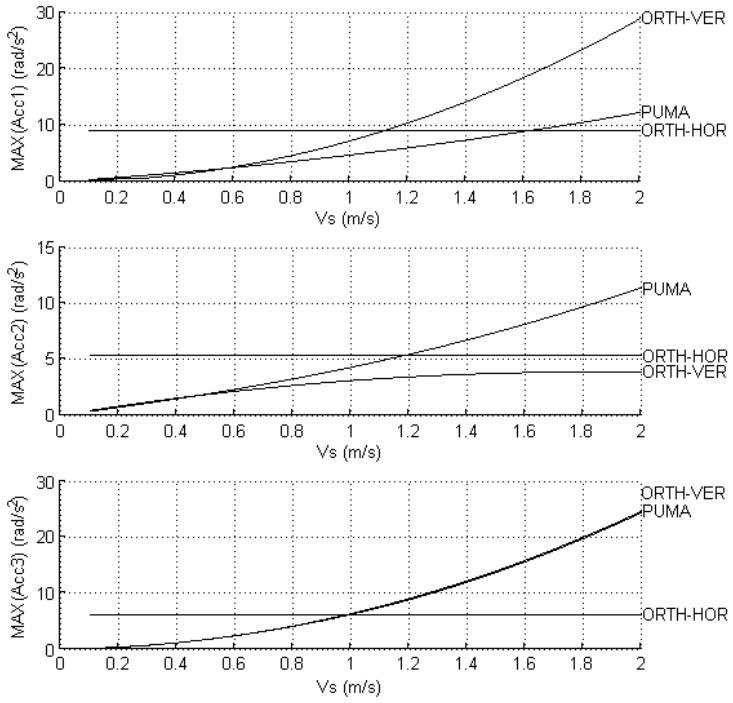


Fig. 3.24. Maximum accelerations of ORTH vs PUMA in task-space using 2D clothoid arcs (White-Circle area)

Test 3: using cubic spline function in the task-space

In this case, we will see the EWA areas more clearly than in the other tests. Notice the areas with Vs < 1 m/s and also the tradeoff of accelerations for ORTH_VER robot.

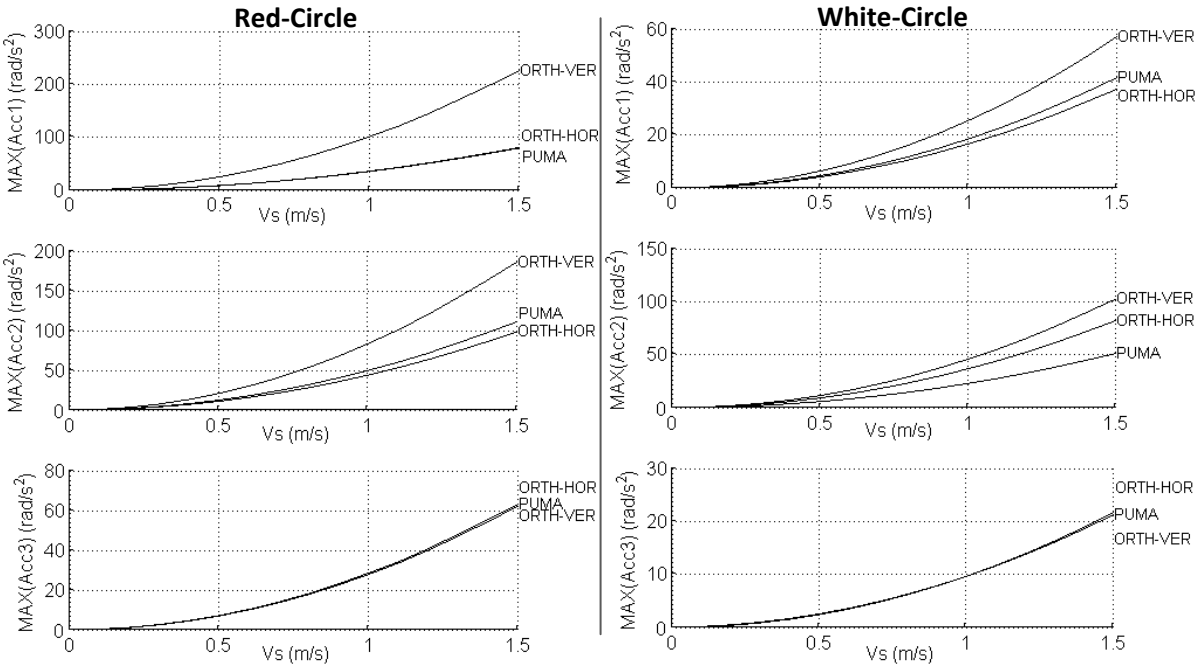


Fig. 3.25. Maximum accelerations of ORTH vs PUMA in task-space using cubic spline function

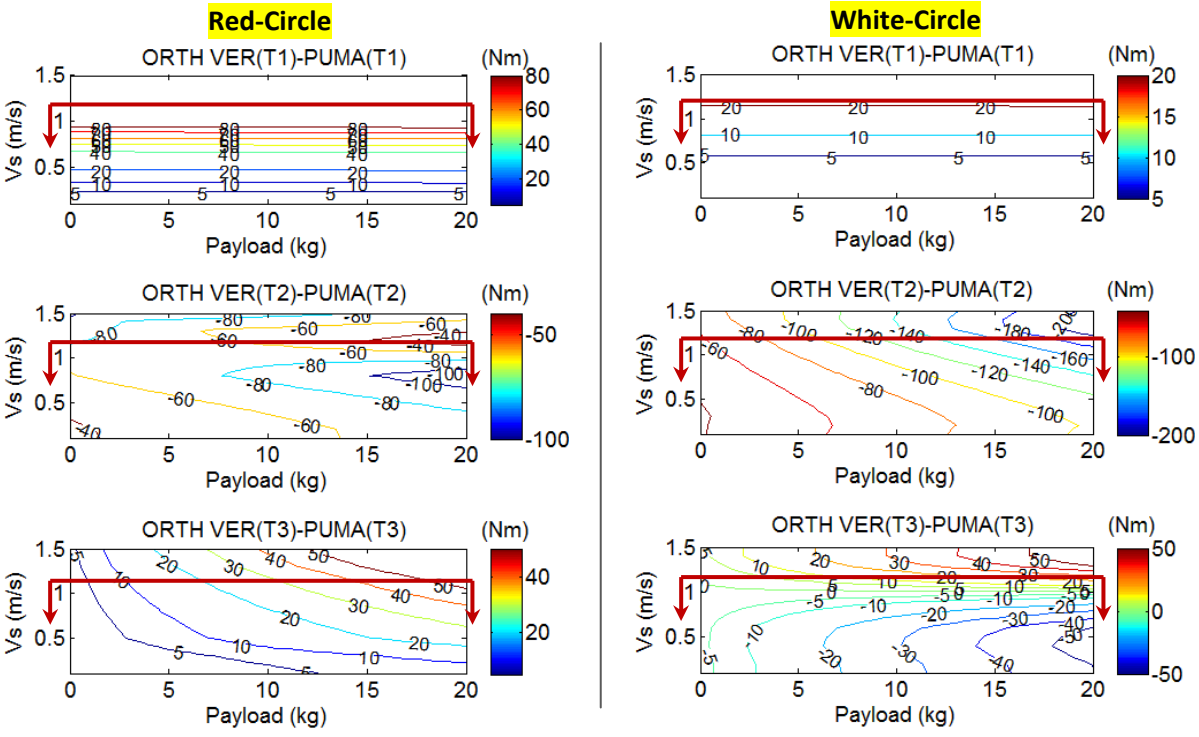


Fig. 3.26. Contours of maximum torques of (ORTH_VER – PUMA) in task-space using cubic spline function

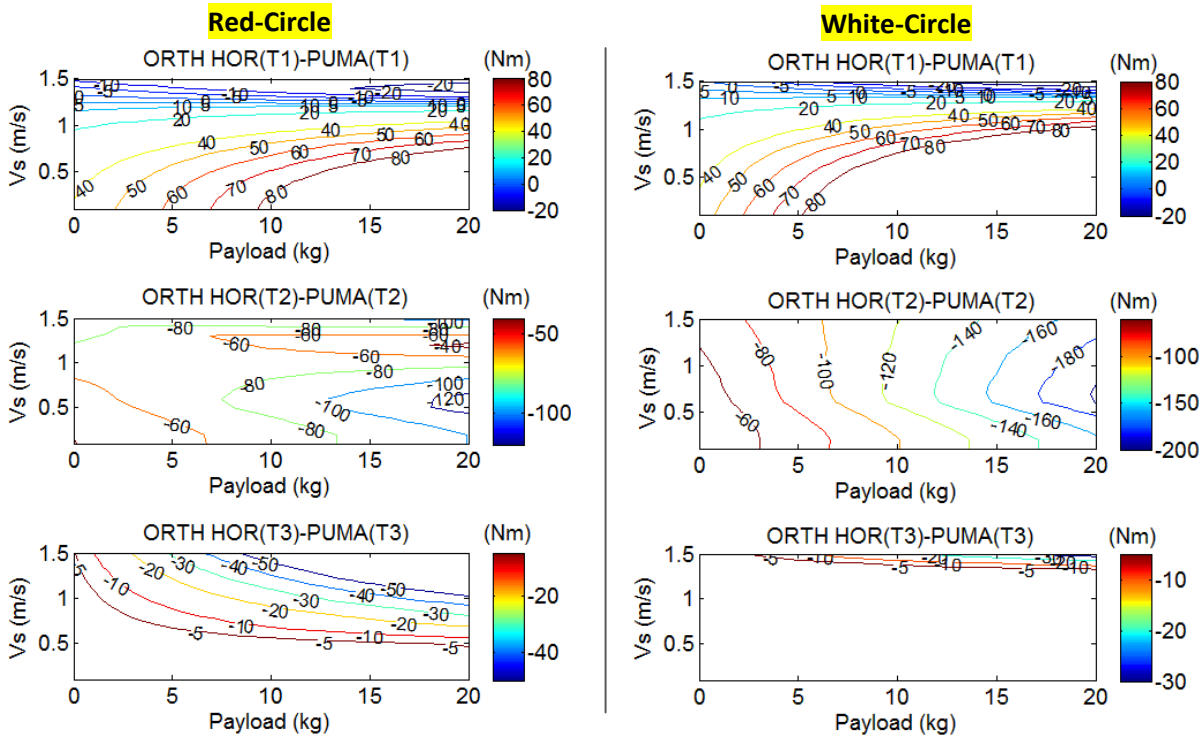


Fig. 3.27. Contours of maximum torques of (ORTH_HOR – PUMA) in task-space using cubic spline function

- In the above results, we computed the difference between maximum torques of ORTH vs PUMA robot, then plot the contour curves. The areas with values < 0 will determine the better performance.

Comment:

- From several results of dynamic analysis, we can assume that in lower operational speed, the ORTH_VER robot has better performance than PUMA robot in term of actuator torques (or economic aspect). However we can see also the tradeoff especially in joint accelerations for ORTH_VER robot.
- In these tests, we chose the two trajectories in the Red-Circle and White-Circle regions (at fix coordinates x_P = const) because they belong to the common workspace of ORTH_VER and ORTH_HOR robot (there is no meaning of comparing the performance between the two modes of ORTH robot). Although we could see some of the dynamic properties but **the main reason for doing these tests is to give a proof of the EWA's existence**.
- Depending on the trajectory generating methods, doing suitable changing in analysis could help us to indicate the EWA areas.

CHAPTER 4

4.1. Stating optimization problem

"Assume that we have now three motors with fixed maximum torques T1, T2, T3. We want to build a 3R serial manipulator for the Pick-n-Place applications. Which type of manipulators we should choose, ORTH or PUMA robot?"

Up until now, we have seen through several examples of the dynamic analysis. Is it enough to stop at that? The answer is not! All the previous steps only show us some intuitive ways to have a grasp of the dynamic performance.

In this section, we introduce the last step : optimization procedure to construct some kinds of exciting trajectory based on common methods (5^{th} degree polynomial, cubic spline…) which could give us rich information of the performance. We applied the procedure for only ORTH robot in vertical mode (ORTH_VER) and PUMA robot since they both have symmetric profiles with respects to the gravity.

We can note that:

- All the trajectory generating methods are **Point-to-Point** or **Via-Points.** We can construct the optimized trajectory consists of several segments in the joint space / task space (P1-P2), (P2-P3), (P3-P4)… or from a subset of via-points (P1-P2-P3-P4)…
- There are several ways to re-construct the trajectory of each segment with each method. For example, with 5^{th} degree freedom method in the joint space, we only need to know the initial position $\boldsymbol{q}^{i}$, final position $\boldsymbol{q}^{f}$ and the total travel time t or we only need to know $\left(\mathbf{q}^{i}, \mathbf{q}^{f}, \dot{\mathbf{q}}^{*}\right)$ where $\dot{\mathbf{q}}^{*}$ is an extreme point of the trajectory (i.e. $\dot{\mathbf{q}}^{*} = \dot{\mathbf{q}}_{max}$) …

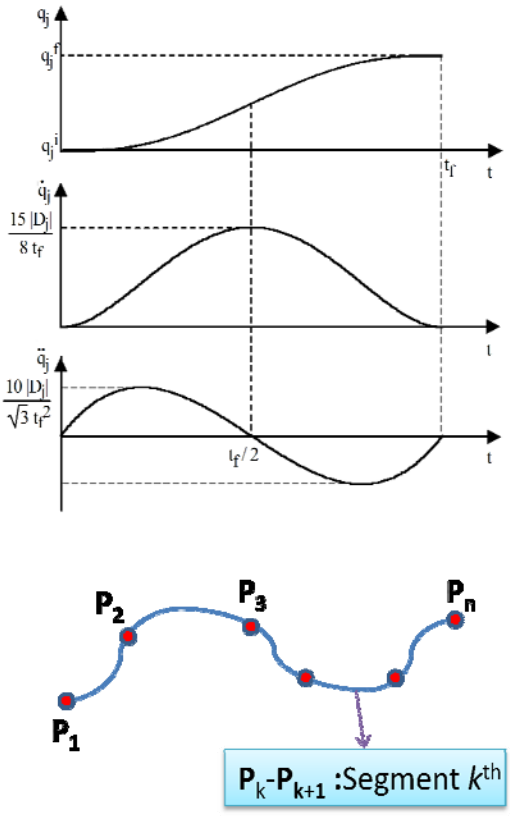


Fig. 4.1. Constructing optimized trajectory profile

Comparison

There are two approaches to do the optimization with the purpose of comparing.

- **From the joint-space**

  $\text{optimize}(\text{ORTH}) \rightarrow \text{recieve}(\mathbf{Qs}, \mathbf{Ts}) \rightarrow \text{project to task - space}(\mathbf{P}, \dot{\mathbf{P}}, \ddot{\mathbf{P}}) \rightarrow \text{inv}(\text{PUMA})$

  $\text{optimize}(\text{PUMA}) \rightarrow \text{recieve}(\mathbf{Qs}, \mathbf{Ts}) \rightarrow \text{project to task - space}(\mathbf{P}, \dot{\mathbf{P}}, \ddot{\mathbf{P}}) \rightarrow \text{inv}(\text{ORTH})$

  First, we can either perform the optimization within the joint-space for ORTH or PUMA robot. After achieving the optimized profile (**Q**s, **T**s), we can project into the task-space and find the corresponding inverse-solution of the other robot, then compare the performance. If in either case, for example, the ORTH shows better results than PUMA then we are happy!

  In this case, if we do the optimization with ORTH robot first, we don't have to worry about exceeding the workspace in task-space – hence can find the inverse solution, because PUMA robot have largest workspace. But doing optimization for PUMA first, we need to **constraint the end-effector's coordinates must be inside the range of ORTH's workspace**.

  And also we have to **check if the trajectory cross the singularity boundary** while solving the inverse geometric equations during the last-step at each way!

- **From the task-space**

  $\text{optimize}(\text{ORTH}) \rightarrow \text{recieve}(\mathbf{Ps}, \mathbf{Ts}) \rightarrow \text{project to joint - space} \rightarrow \text{achieve}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$

  $\text{optimize}(\text{PUMA}) \rightarrow \text{recieve}(\mathbf{Ps}, \mathbf{Ts}) \rightarrow \text{project to joint - space} \rightarrow \text{achieve}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$

  In this case, we can easily set the constraint of the end-effector's position inside the ORTH's workspace, and also we can set the limit to not cross the singularity boundary (especially for PUMA robot), hence all the steps can be done easier than approaching from the joint-space!

  We also have to select which posture of each robot we want to compare.

Questions

- How can we choose the range's limit $\left(\mathbf{q}_{\min}^{i,f}, \mathbf{q}_{\max}^{i,f}\right)$ for each segment (both in the joint-space and task-space)?

  This is difficult. Although we can let this for the users to define the limit themselves but finding a good way to determine the limits is still better.
- How to determine the objective criteria and formulate constraint equations?
- Using the optimization toolbox of MATLAB (*fgoalattain, fmincon*…) : which is the good way to perform the optimization ?

Optimization Setup (in the joint-space): $\mathbf{X^* = fgoalattain(X)}$

Objective :

$$\begin{cases} \sum \text{abs}\left(T_{kmax}\right) \rightarrow \max \quad \left(T_{kmax} = \max\left(\text{Torques of segment } k^{th}\right)\right) \\ \sum \text{abs}\left(\ddot{q}_{kmax}\right) \rightarrow \max \\ \ldots \end{cases} \tag{4.1}$$

Searching parameter : $\mathbf{X} = (\mathbf{q}^i, \mathbf{q}^f, t)$

Constraints :

$$\begin{cases} \left|T_{ikmax}\right| \le T_{imax} \left(i = 1,2,3\right) \\ \mathbf{V}\left(k\right) = \dfrac{\mathbf{L}\left(k\right)}{t_\Sigma} \le \mathbf{V}_{max} \\ \mathbf{0} \le \mathbf{t} \qquad \left(\text{travelling time must be positive}\right) \\ \mathbf{q}^i_{min} \le \mathbf{q}^i \le \mathbf{q}^i_{max} \\ \mathbf{q}^f_{min} \le \mathbf{q}^f \le \mathbf{q}^f_{max} \\ P \in \text{ORTH's workspace} \end{cases} \tag{4.2}$$

where

$$\begin{cases} T_{imax} = \text{maximum actuator torque of joint } i^{th} \ \left(\text{KNOWN}\right) \\ T_{ikmax} = \text{maximum torque of joint } i^{th} \text{ at segment } k^{th} \\ \mathbf{V}_{max} = \text{maximum operational velocity of the end - effetor (average value)}\left(\text{KNOWN}\right) \\ \mathbf{L}\left(k\right) = \sum \left\| P\left(n+1\right) - P\left(n\right) \right\| = \text{total traveling length of the end - point P at each segment} \\ t_\Sigma = \mathbf{t}\left(k\right) = \text{total traveling time at each segment} \\ \left(\mathbf{q}^{i,f}_{min}, \mathbf{q}^{i,f}_{max}\right) = \text{range's limit at each segment}\left(\text{KNOWN}\right) \end{cases}$$

The role of V_{max} is to keep the procedure under the speed assumption, and we can verify at different payload. The aim is to bring out the worst case possible (reach max(Torq/Acc)).

Optimization Setup (in the Task-space): $\mathbf{X^* = fgoalattain(X)}$

Objective :

$$\begin{cases} \sum \text{abs}\left(T_{kmax}\right) \rightarrow \max \quad \left(T_{kmax} = \max\left(\text{Torqs of segment } k^{th}\right)\right) \\ \sum \text{abs}\left(\ddot{q}_{kmax}\right) \rightarrow \max \\ \ldots \end{cases}$$

Searching parameter: $\mathbf{X = (P^i, P^f, t)}$

At each step**:** from X, we can compute $\left(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}\right) = \text{inverse}\left(\mathbf{P}, \dot{\mathbf{P}}, \ddot{\mathbf{P}}\right)$ using analytical solutions.

(here, we cannot use numerical method to compute the velocities and accelerations from just positions, doing this will cause a lot of bad results for the analysis!).

Constraints:

$$\begin{cases} \left|T_{ikmax}\right| \le T_{imax} \left(i = 1,2,3\right) \\ \mathbf{V}\left(k\right) = \dfrac{\mathbf{L}\left(k\right)}{t_{\Sigma}} \le \mathbf{V}_{max} \\ \mathbf{0} \le \mathbf{t} \\ \mathbf{x}_{min}^{i,f} \le \mathbf{x}^{i,f} \le \mathbf{x}_{max}^{i,f} \\ \mathbf{y}_{min}^{i,f} \le \mathbf{y}^{i,f} \le \mathbf{y}_{max}^{i,f} \\ \mathbf{z}_{min}^{i,f} \le \mathbf{z}^{i,f} \le \mathbf{z}_{max}^{i,f} \end{cases} \tag{4.3}$$

where

$$\begin{cases} T_{imax} = \text{maximum actuator torque of joint } i^{th} \ \left(\text{KNOWN}\right) \\ T_{ikmax} = \text{maximum torque of joint } i^{th} \text{ at segment } k^{th} \\ \mathbf{V}_{max} = \text{maximum operational velocity of the end - effetor (average value)} \left(\text{KNOWN}\right) \\ \mathbf{L}\left(k\right) = \sum \left\| P\left(n+1\right) - P\left(n\right) \right\| = \text{total traveling length of the end - point P at each segment} \\ t_{\Sigma} = \mathbf{t}\left(k\right) = \text{total traveling time at each segment} \\ \left(x, y, z\right) = \text{coordinates of the end - point P} \\ \left(\mathbf{x}_{min}^{i,f}, \mathbf{x}_{max}^{i,f}, \mathbf{y}_{min}^{i,f}, \mathbf{y}_{max}^{i,f}, \mathbf{z}_{min}^{i,f}, \mathbf{z}_{max}^{i,f}\right) = \text{range's limit at each segment} \left(\text{KNOWN}\right) \end{cases}$$

Determine the limit ranges

There are several ways to choose the searching limits. In this section, we will use the static torques profiles in case of the ORTH_VER robot to determine $\left(\mathbf{q}_{min}^{i,f}, \mathbf{q}_{max}^{i,f}\right)$.

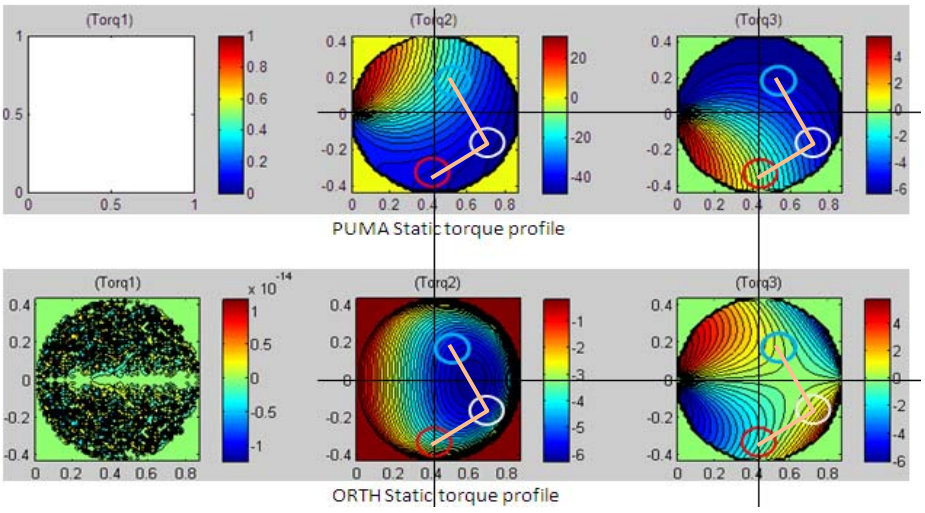


Fig. 4.2. Static torques profiles of ORTH_VER vs PUMA robots

The idea is to choose the limit ranges *that could cover the working space and also in order to construct a trajectory that could pass through the most exciting zones* based on the static torques profiles (i.e. from low distributing torques areas to high distributing torques areas). One way to do is dividing the workspace into small segments like in figure 4.3.

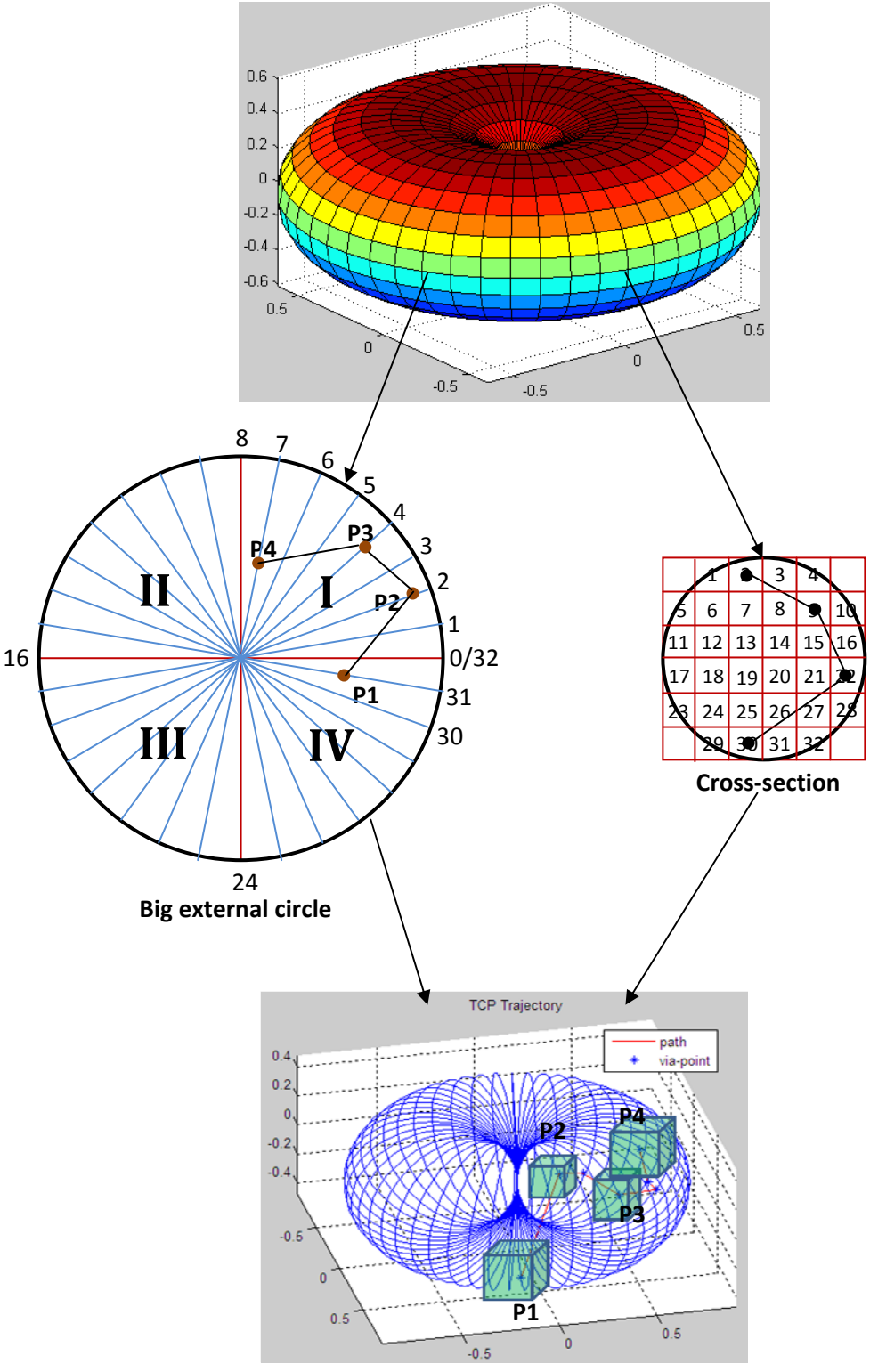


Fig. 4.3. Selecting limit ranges

The workspace can be divided into 4 big regions (**I**,**II**,**III**,**IV**) and 32x32 small segments. The trajectory, for example in this case, can be auto-generated using the indices of each segment

like: **P1**(31,30) – **P2**(2,22) – **P3**(4,9) – **P4**(7,2). If we take the centers point of each segment then expand them to become larger cubes, we will achieve the limit ranges of the searching parameters for optimization. By doing this we could choose a way to cover all the workspace with desired directions more easily.

Note on MATLAB optimization toolbox

Currently, MATLAB provides the users several choices for nonlinear optimization such as following:

Optimization toolbox	Usage
fmincon	Minimize one-objective criteria, useful for small number of searching parameters and less complex objective function.
fgoalattain	Multi-objective optimization – useful for optimization with several objective goals.
ga	Using genetic algorithm for minimize one-objective function, belongs to *global optimization toolbox*
gs	Global search – repeat performing the **fmincon** until find a local minimum for one-objective function problems, belongs to *global optimization toolbox*

Table 4.1. Summary on some common optimization toolbox in MATLAB

- In our case, we used **fgoalattain** function to perform the optimization tasks since it is more efficient if we consider our problem as the multi-objectives one.
- We have also tested optimization procedure using the other functions but they gave very unstable solutions, especially when we verified with large number of searching parameters.
- Later versions of MATLAB (ver. 2010 or 2011) provide global optimization toolbox which is quite nice for finding global solutions. But for now they only support for one-objective problem.
  In general, the optimized results depend on many aspects of formulating the optimization problem itself (initial points, constraints…)

4.2. Optimization results

In the following tests, we try to maximize the average actuator torques of three joints for each manipulator through all the segments of the trajectory (three objective functions):

$$\begin{cases} \sum^{\text{joint 1}} abs\left(T_{k\max}\right) \rightarrow \max \quad \left(T_{k\max} = \max\left(Torques\ of\ \ segment\ k^{th}\right)\right) \\ \sum^{\text{joint 2}} abs\left(T_{k\max}\right) \rightarrow \max \\ \sum^{\text{joint 3}} abs\left(T_{k\max}\right) \rightarrow \max \end{cases} \tag{4.4}$$

We keep the joint accelerations freely to see the trade off in performance of the two manipulators while limiting the actuator torques. This step follows the idea performed before in the dynamic analysis section.

As mentioned, the optimized trajectory is constructed from several segments through *m* via-points. We need only 3 parameters for each segment : the initial and final points $\left(\mathbf{q}^{init}, \mathbf{q}^{final}\right)$ and the travelling time t_f :

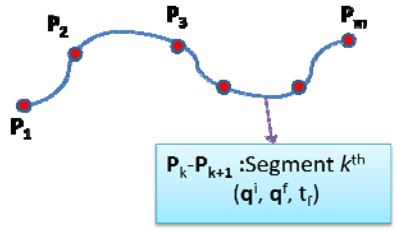


Fig. 4.4. Example of optimized trajectory

With *m* via-points, we have (*m-1*) segments, which means there are total (*4*m-1)* searching parameters (the same number for both in the task-space or in the joint-space because we have only 3 coordinates or 3 joints). Because of this reason, we should reduce the number of via-points to have more stable solutions while performing the optimization tasks.

In our case, *we use the approach from the task-space* since this way is more straight forward.

We construct the optimized trajectory with 5 via-points – 4 segments – (total 19 searching parameters) and repeat in 4 times for four big regions of the workspace (**I,II,III,IV**).

In table 4.2:

- The center via-points (**P**) at each big region is represented by a vector of 2 rows, the first row contains indices for largest external-circle segments – i_k , the second row contains indices for cross-section segments – j_k (similar to figure 4.3). Each column will present a center point $\mathbf{P}_k(i_k,j_k)$.
- The vector **Δ** contains the expanding sizes to form a searching cube with center at $\mathbf{P}_k$ (the size of each cube will be $2\Delta_k \times 2\Delta_k \times 2\Delta_k$), d_4 is the radius of the cross-section.
- The maximum torques (T1, T2, T3) are chose based on the static torque profiles.

A center point $\mathbf{P}(i,j)$ of a searching zone is defined as the center of the corresponding square with index *j* on the layer with index *i*. Coordinates of this point in the task-space can be obtained by: $\mathbf{P}_{i,j} = \mathrm{Rot}\left(z,\theta_i\right)\times\mathbf{P}_{0,j}$ (4.5)

where $\theta_i = i\times(\pi/16)$, $\mathbf{P}_{0,j}(x,0,z)$ is the center point of the j^{th} square on the layer with index 0/32, and $\mathrm{Rot}(z,\theta_i)$ denotes the rotation matrix along the z-axis by an angle θ_i .

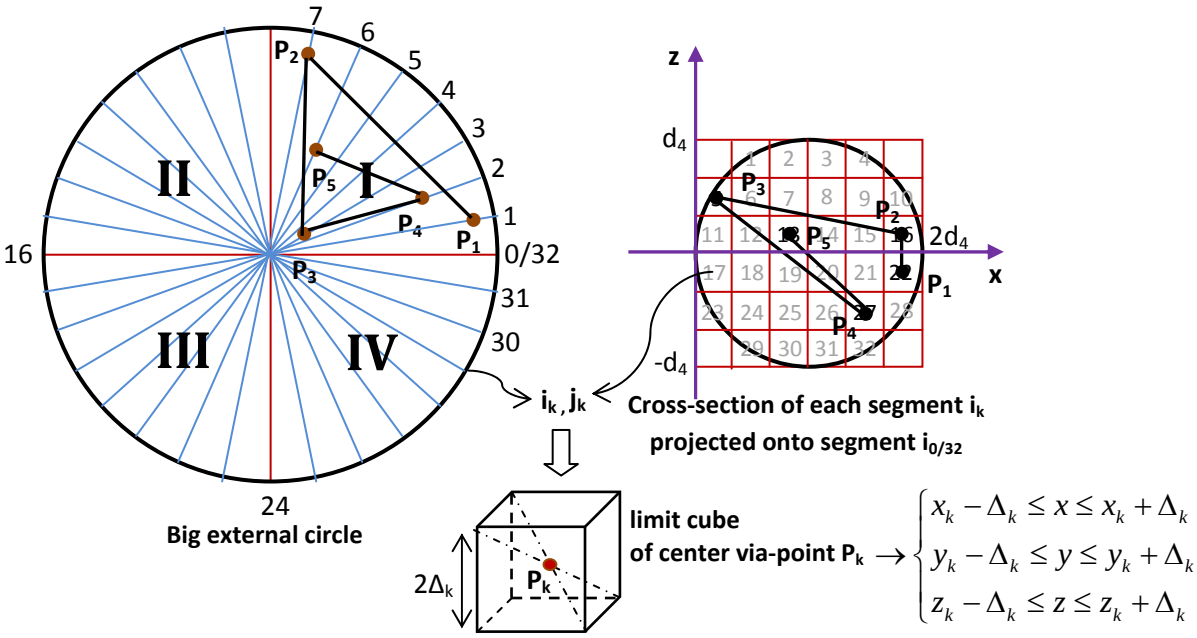
A trajectory with desired passing locations and directions can then be constructed by choosing suitable indices (i_k,j_k) for any via-point $\mathbf{P}_k$.

In optimization, the starting points are initialized as random values within offsets around the center points $\mathbf{P}_k(x_k,y_k,z_k)$. Then the procedure will search for optimized points inside the limit cubes defined by 4.3).

For example, in case of **region I**, **using 5th degree polynomial**, we have:

$$\mathbf{P}:\begin{bmatrix}1 & 7 & 3 & 2 & 6\\ 22 & 16 & 5 & 27 & 13\end{bmatrix},\qquad \boldsymbol{\Delta}:\begin{bmatrix}\frac{d_4}{6} & \frac{d_4}{6} & \frac{d_4}{6} & \frac{d_4}{6} & \frac{d_4}{3}\end{bmatrix}$$

That means the center desired via-points: $\mathbf{P_1}(1,22) - \mathbf{P_2}(7,16) - \mathbf{P_3}(3,5) - \mathbf{P_4}(2,27) - \mathbf{P_5}(6,13)$



$$\rightarrow \begin{cases} x_k - \Delta_k \le x \le x_k + \Delta_k \\ y_k - \Delta_k \le y \le y_k + \Delta_k \\ z_k - \Delta_k \le z \le z_k + \Delta_k \end{cases}$$

Fig. 4.5. Example of constructing limit ranges for desired trajectory

(For further detail in debugging the optimization procedure, please look for *appendix A3*)

Region	Center Via-Points		Max Torques [T1 T2 T3] (Nm)	Max TCP Speed [V_1 V_2 V_3 V_4] (m/s)
	Cubic spline function	5th Degree polynomial		
I	$\mathbf{P}:\begin{bmatrix}1 & 1 & 3 & 5 & 7\\ 26 & 8 & 19 & 15 & 6\end{bmatrix}$ $\boldsymbol{\Delta}:\begin{bmatrix}\frac{d_4}{20} & \frac{d_4}{8} & \frac{d_4}{10} & \frac{d_4}{8} & \frac{d_4}{20}\end{bmatrix}$	$\mathbf{P}:\begin{bmatrix}1 & 7 & 3 & 2 & 6\\ 22 & 16 & 5 & 27 & 13\end{bmatrix}$ $\boldsymbol{\Delta}:\begin{bmatrix}\frac{d_4}{6} & \frac{d_4}{6} & \frac{d_4}{6} & \frac{d_4}{6} & \frac{d_4}{3}\end{bmatrix}$	[80 80 30]	[1.2 1.4 1.4 1.2] (limit of average speed at each segment)
II	$\mathbf{P}:\begin{bmatrix}9 & 12 & 11 & 13 & 15\\ 3 & 11 & 22 & 26 & 27\end{bmatrix}$ $\boldsymbol{\Delta}:\begin{bmatrix}\frac{d_4}{15} & \frac{d_4}{7} & \frac{d_4}{8} & \frac{d_4}{7} & \frac{d_4}{15}\end{bmatrix}$	$\mathbf{P}:\begin{bmatrix}9 & 10 & 12 & 13 & 15\\ 31 & 17 & 15 & 5 & 3\end{bmatrix}$ $\boldsymbol{\Delta}:\begin{bmatrix}\frac{d_4}{6} & \frac{d_4}{6} & \frac{d_4}{3} & \frac{d_4}{6} & \frac{d_4}{6}\end{bmatrix}$	-	[1.3 1.3 1.3 1.3]
III	$\mathbf{P}:\begin{bmatrix}17 & 19 & 20 & 21 & 23\\ 9 & 6 & 24 & 25 & 13\end{bmatrix}$ $\boldsymbol{\Delta}:\begin{bmatrix}\frac{d_4}{6} & \frac{d_4}{7} & \frac{d_4}{6} & \frac{d_4}{7} & \frac{d_4}{6}\end{bmatrix}$	$\mathbf{P}:\begin{bmatrix}17 & 20 & 19 & 23 & 22\\ 20 & 16 & 24 & 9 & 11\end{bmatrix}$ $\boldsymbol{\Delta}:\begin{bmatrix}\frac{d_4}{6} & \frac{d_4}{6} & \frac{d_4}{6} & \frac{d_4}{6} & \frac{d_4}{6}\end{bmatrix}$	-	[1.2 1.2 1.2 1.2]
IV	$\mathbf{P}:\begin{bmatrix}25 & 31 & 27 & 30 & 26\\ 7 & 16 & 27 & 18 & 8\end{bmatrix}$ $\boldsymbol{\Delta}:\begin{bmatrix}\frac{d_4}{8} & \frac{d_4}{8} & \frac{d_4}{8} & \frac{d_4}{8} & \frac{d_4}{8}\end{bmatrix}$	$\mathbf{P}:\begin{bmatrix}25 & 27 & 26 & 29 & 31\\ 2 & 22 & 24 & 3 & 20\end{bmatrix}$ $\boldsymbol{\Delta}:\begin{bmatrix}\frac{d_4}{5} & \frac{d_4}{5} & \frac{d_4}{5} & \frac{d_4}{5} & \frac{d_4}{5}\end{bmatrix}$	-	[1.3 1.2 1.2 1.3]

Table 4.2. Initial parameters for optimization procedure

Test 1: optimize using Cubic Spline function (Via-Points) : ORTH first

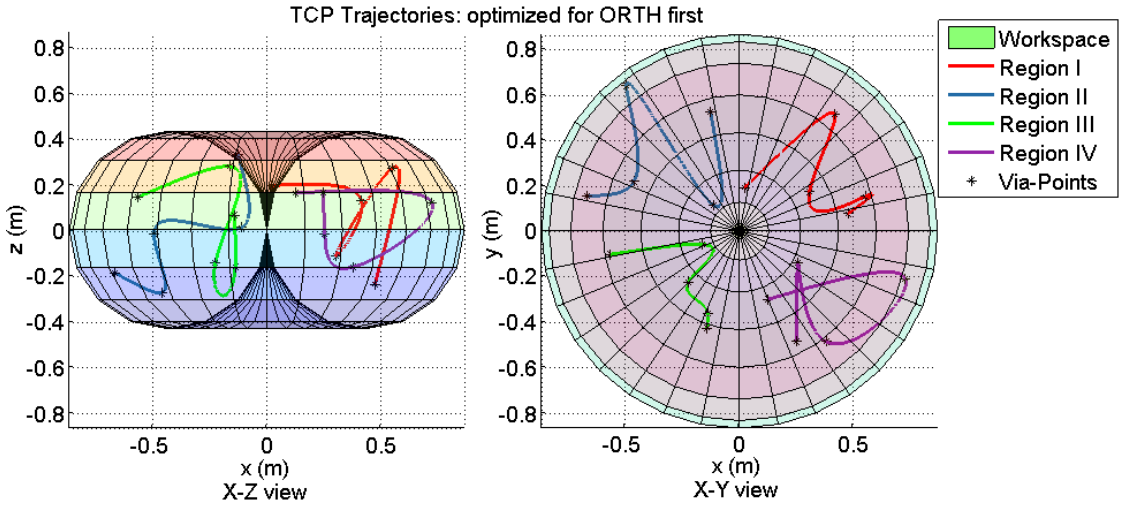


Fig. 4.6. Optimized Trajectories : Cubic Spline, Task-Space (optimize for ORTH first)

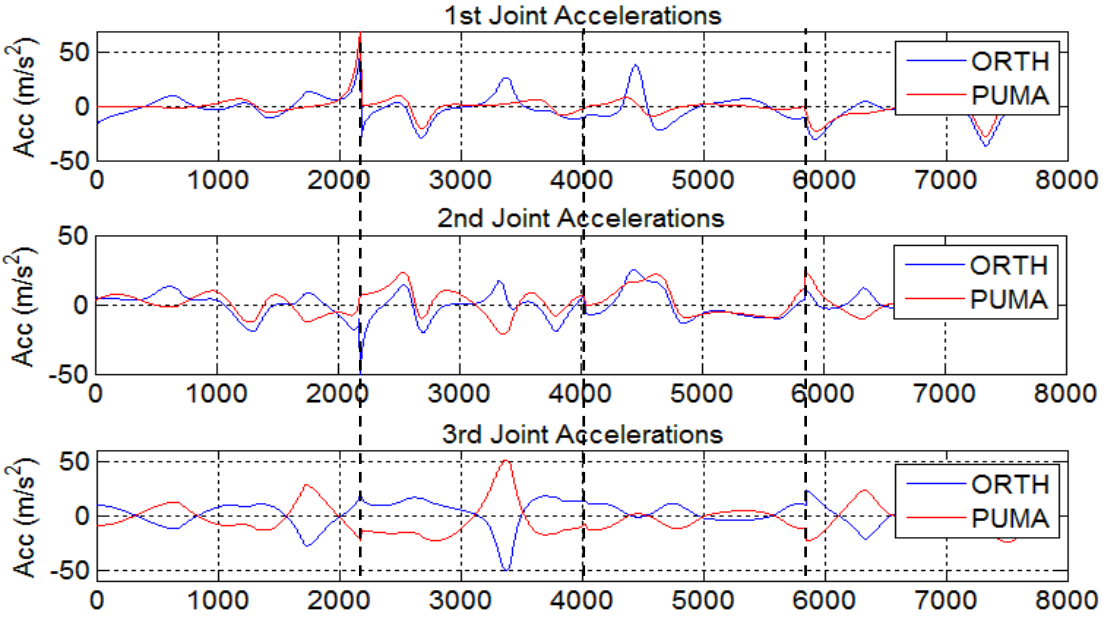


Fig. 4.7. Joint Accelerations : Cubic Spline, Task-Space (optimize for ORTH first)

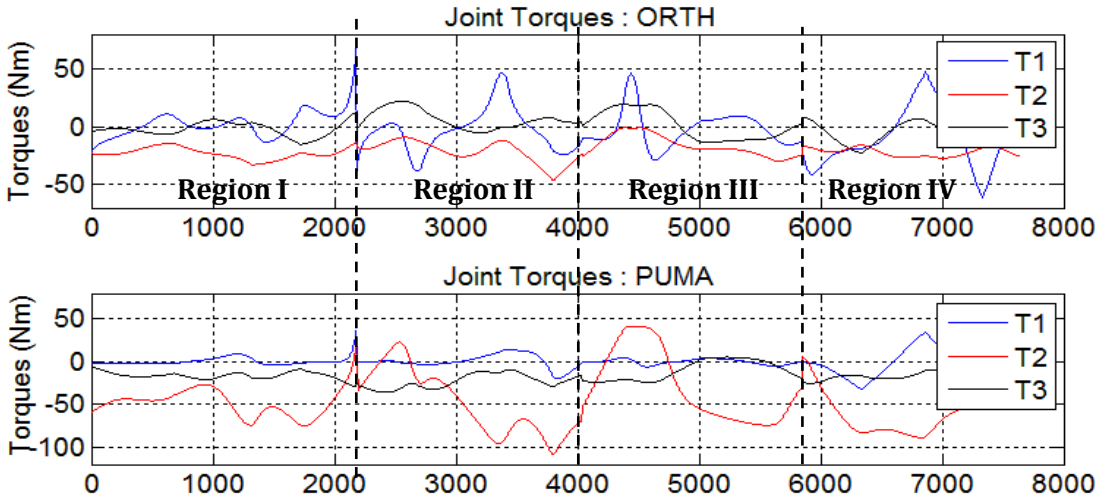


Fig. 4.8. Actuator Torques : Cubic Spline, Task-Space (optimize for ORTH first)

Test 2: optimize using Cubic Spline function (Via-Points) : PUMA first

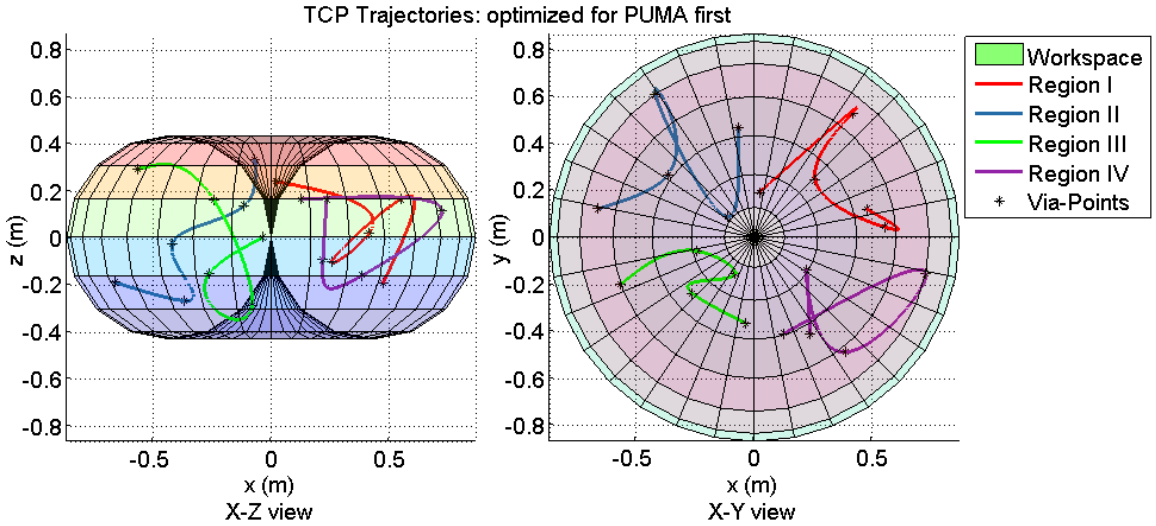


Fig. 4.9. Optimized Trajectories : Cubic Spline, Task-Space (optimize for PUMA first)

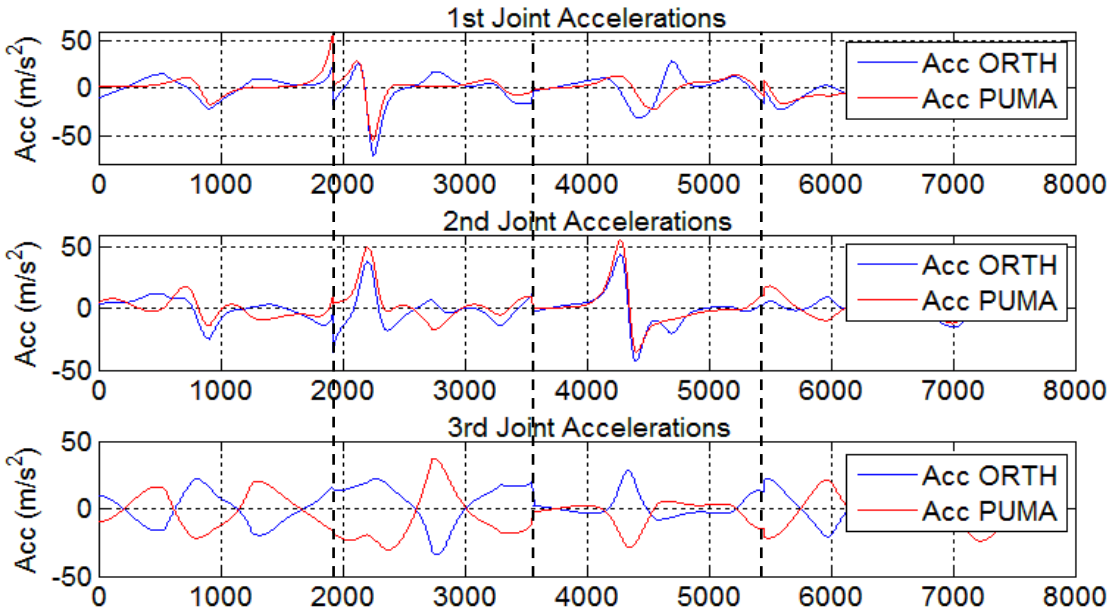


Fig. 4.10. Joint Accelerations : Cubic Spline, Task-Space (optimize for PUMA first)

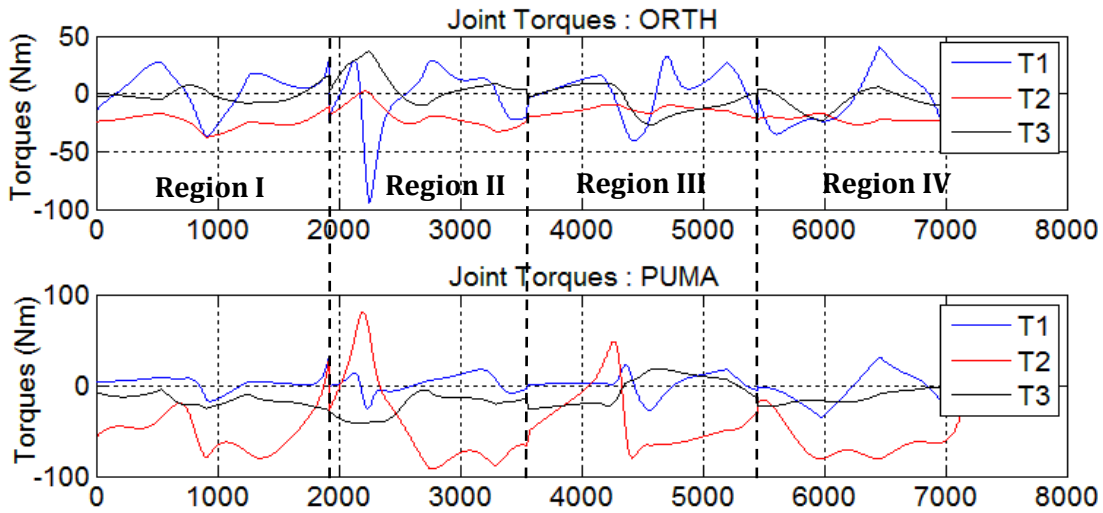


Fig. 4.11. Actuator Torques : Cubic Spline, Task-Space (optimize for PUMA first)

Test 3: optimize using 5th degree polynomial (Point-to-Point): PUMA first

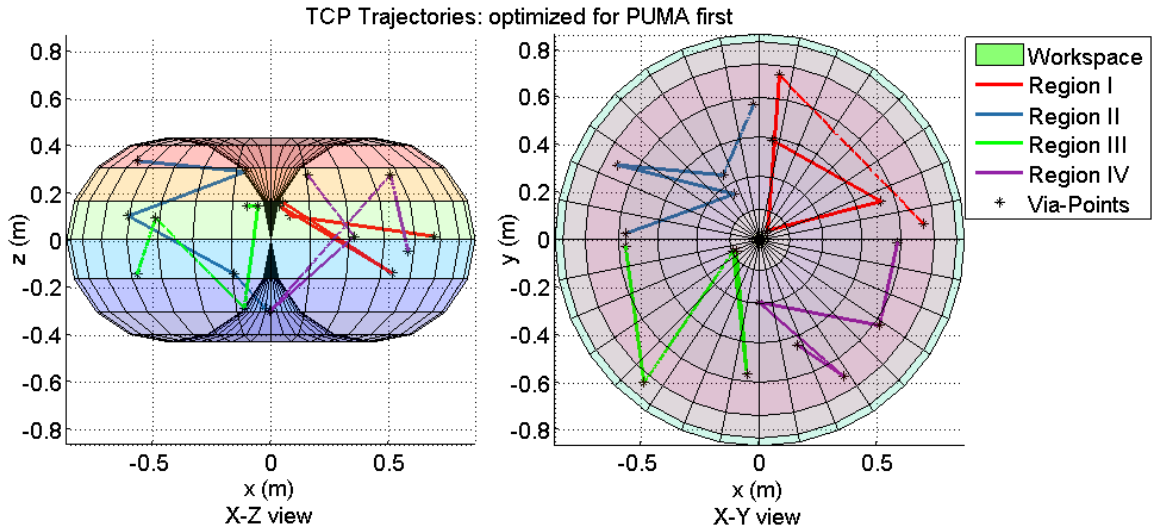


Fig. 4.12. Optimized Trajectories : 5th degree, Task-Space (optimize for PUMA first)

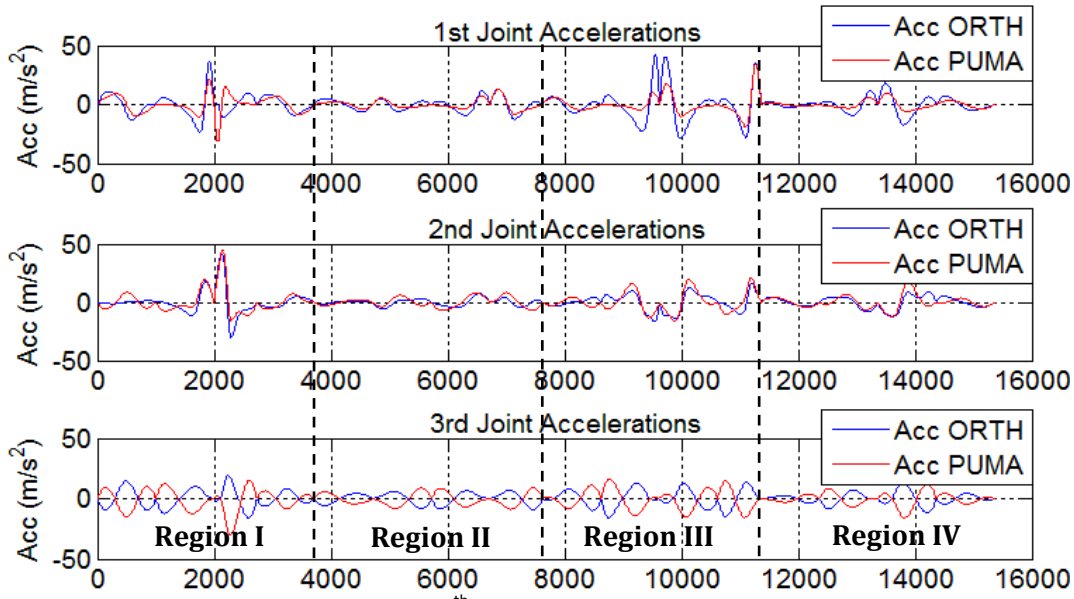


Fig. 4.13. Joint Accelerations : 5th degree, Task-Space (optimize for PUMA first)

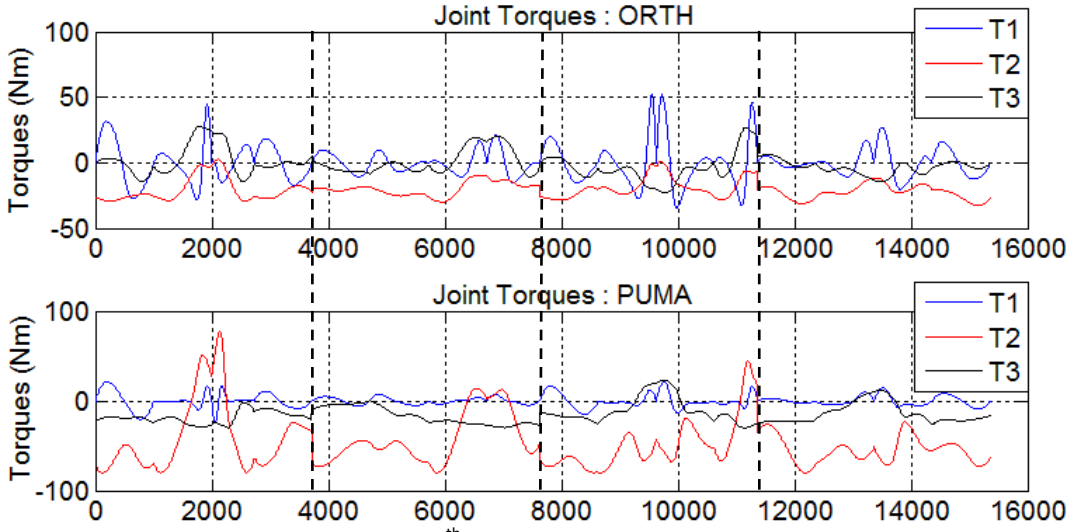


Fig. 4.14. Actuator Torques : 5th degree, Task-Space (optimize for PUMA first)

Test 4: optimize using 5th degree polynomial (Point-to-Point): ORTH first

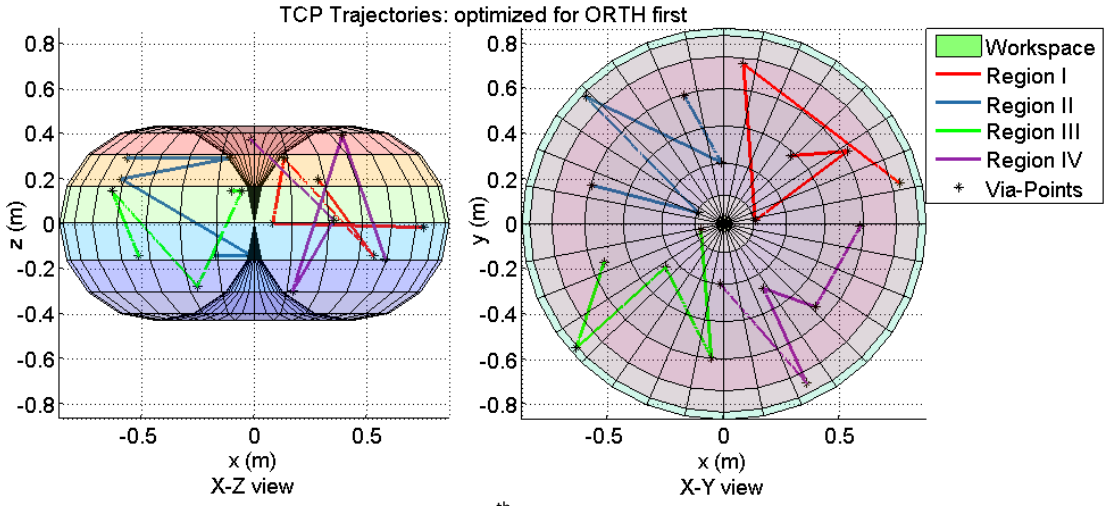


Fig. 4.15. Optimized Trajectories : 5th degree, Task-Space (optimize for ORTH first)

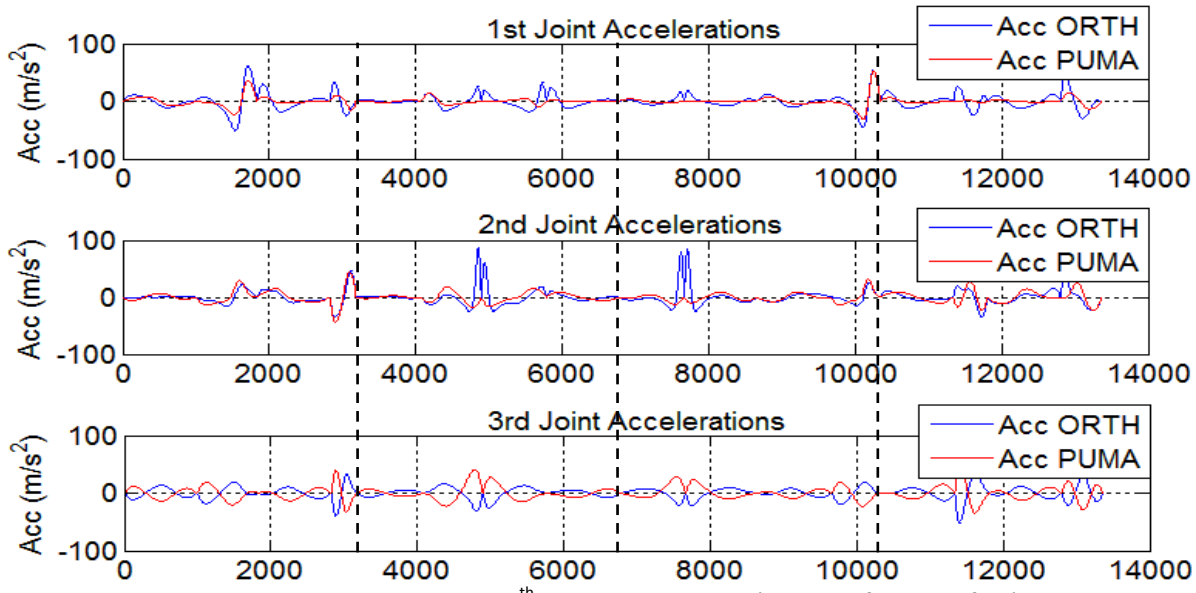


Fig. 4.16. Joint Accelerations : 5th degree, Task-Space (optimize for ORTH first)

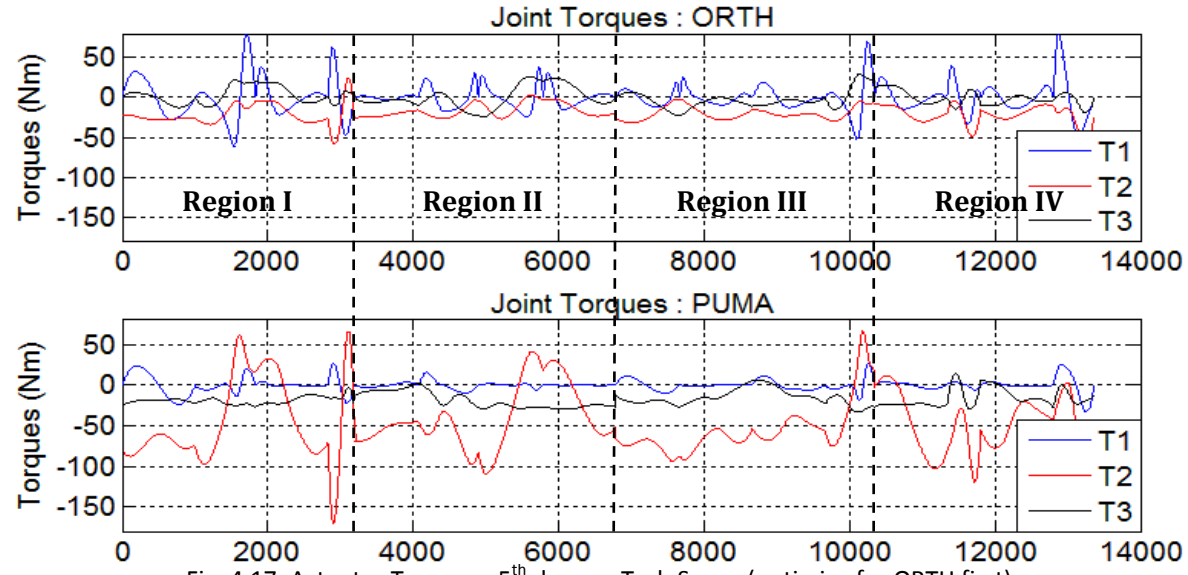


Fig. 4.17. Actuator Torques : 5th degree, Task-Space (optimize for ORTH first)

Indicating EWA areas

In the following, we show several results of the EWA areas for optimized via-points obtained from the tests using cubic spline function (test 1) and using 5th degree polynomial (test 3).

Following results show examples of maximum torques profile of the two manipulators for optimized via-points obtained from test 1 and test 3. The maximal torque values of each joint for both manipulators are computed while changing the maximal values of the speed of the end-effector (Vs) and the payload.

In figures 4.18 – 4.19 that indicate the EWA areas, we present two torque profiles: on the left shows the difference of ($T_k$max(ORTH) – $T_k$max(PUMA)), on the right shows these differences with respect to the maximum joint torque 3 of ORTH robot.

In practice, it is enough to consider the areas with a velocity of the end-effector less than 1m/s. The performances of joint torques 2 and 3 of the ORTH robot are better than for the PUMA manipulator. Although joint torque 1 of the ORTH manipulator is larger than for the PUMA, the differences are small with respect to the differences of the second joint torques.

Case 1: EWA profile for via-points obtained from Test 1, validate using cubic spline function method.

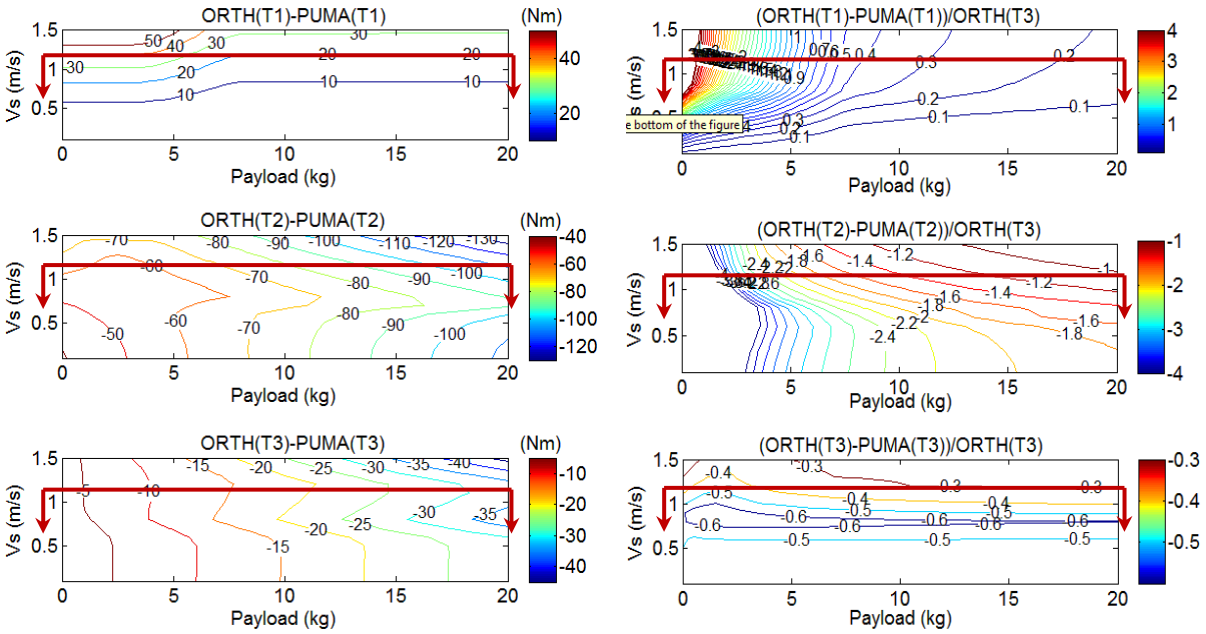


Fig. 4.18. Maximum joint torques of ORTH_VER and PUMA obtained from test 1, verified using cubic spline function

<u>Case 2</u>: EWA profile for via-points obtained from Test 3, validate using 5th degree polynomial method.

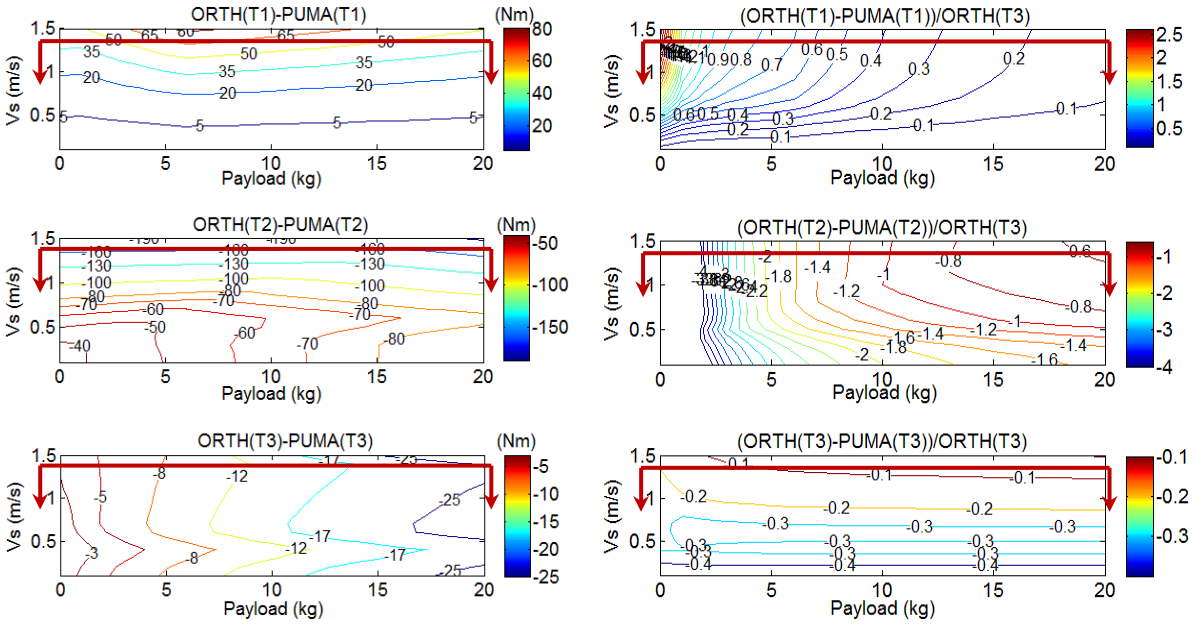


Fig. 4.19. Maximum torques for via-points obtained from Test 3, verified using 5th degree polynomial

Comment

- The results in case of optimized trajectories and indicating EWA areas give us a good answer: *ORTH robot in vertical mode has better performance than PUMA robot in term of joint torques. In term of joint accelerations, we have fair performance between them.*
- Beside doing the optimization following the approach from the task-space, we could also use the approach from the joint space. One important reason we have performed in the task-space is that, we have fixed number of searching parameters for *m* via-points (always *4*m-1*). However, in the joint-space approach, if the number of joints increases ($N > 3$) then the searching parameters will grow larger, totally *(N+1)*m-1* parameters. And the results achieved in either cases are not much different in term of giving rich information.

From optimization results, we have proved again the concept of dynamic performance that has been discussed through all the previous chapters, especially the use of EWA areas.

CONCLUSION

Summary

So far, we have reviewed main aspects of the 3R serial orthogonal manipulator, especially its classification into some interesting types in their kinematic properties. Several methods to evaluate the dynamic performance have been discussed.

By following our concept of dynamic performance, we could have a clear view of our problem as well as to formulate other problems in performance analysis.

From all the results in dynamic analysis and optimization procedure, we can confirm that the introduced EWA index is a good criteria to quantify the dynamic performance of a robot manipulator, in our case is to comparing the performance between different ones.

In general, through our work in this thesis, we have shown that ORTH manipulators have nice properties both in term of their kinematic and dynamic performances.

Future plans

- **Validating the concept of dynamic performance, apply for other problems**
- **Validating the criteria EWA in other analysis**
- **Improving the methodology of evaluation in dynamic performance analysis**

  In this case, especially for one robot only, we can extend our method further by verifying the performance of the manipulator within the boundary of a controller.

  In practical, a robot control system has certain saturations to limit the over-operation of the actuators such as under maximum torques or under maximum accelerations. Because of these reasons, when we generate the desired trajectory, some parts of it will be constrained by the controller and the results will be different than under perfect assumptions. If from the EWA index, we could compute the limit in performance of the robot and feed it to the control system then we may improve the robot performance.

  So the analysis can be divided into two steps: *Analysis using simulation without a controller*, and *verify the performance with the interference of a controller*.

REFERENCE

[1] P. Wenger: “Design of cuspidal and noncuspidal manipulators”, Proceedings of IEEE International Conference on Robotics and Automation, 1997, pp. 2172–2177.

[2] Mazen Zein et al: “An exhaustive study of the workspace topologies of all 3R orthogonal manipulators with geometric simplifications”, Sciendirect, Mechanism and Machine Theory 41, 2006, pp. 971–986

[3] M. Baili: “Classification des manipulateurs 3R orthogonaux”, PhD Thesis, Ecole Centrale de Nantes, 2004.

[4] M. Baili, P.Wenger, D. Chablat: “A classification of 3R orthogonal manipulators by the topology of their workspace”, International Conference on Robotics and Automation, Louisiana, USA 2004.

[5] E. Ottaviano, M. Husty, M. Ceccarelli: “A Cartesian Representation for the Boundary Workspace of 3R Manipulators”, Advances in Robot Kinematics, Ed. Kluwer Academic Publishers, Sestri Levante, 2004, pp. 247–254.

[6] J. El Omri, P. Wenger: “How to recognize simply a non-singular posture changing 3-DOF manipulator”, Proceedings of the Seventh International Conference on Advanced Robotics, 1995, pp. 215–222.

[7] M. Baili, P. Wenger, D. Chablat: “Classification of one family of 3R positioning manipulators”, Proceedings of the Eleventh International Conference on Advanced Robotics, 2003.

[8] P. Wenger, P. Chedmail: “On the connectivity of manipulator free-workspace”, Journal of Robotic Systems 8 (6), 1991, pp. 767–799.

[9] P. Wenger: “Uniqueness domains and regions of feasible paths for cuspidal manipulators”, IEEE Transactions on Robotics 20 (4), 2004, pp. 745–750.

[10] P. Borrel, A. Liegeois: “A study of manipulator inverse kinematic solutions with application to trajectory planning and workspace determination”, Proceedings of IEEE International Conference on Robotics and Automation, 1986, pp. 1180–1185.

[11] B. Paden, S.S. Sastry: “Optimal kinematic design of 6R manipulators”, International Journal of Robotics Research 7 (2), 1988, pp. 43–61.

[12] Asada, H.: “A geometrical representation of manipulator dynamics and its application to arm design”, Trans ASME, J. Dyn. Syst., Meas. and Control, Vol. 105, Sept. 1983, pp. 131-135.

[13] Yoshikawa, T.: “Dynamic manipulability of robot manipulators”, Proc. 24th ZEEE Conf. Decision and Control, 1985, pp. 1033-1038.

[14] Tourassis, V. D. and Neuman, C. P.: “The inertial characteristics of dynamic robot models”, Mechanism and Machine Theory, Vol. 20, No. 1, 1985, pp.41-52.

[15] Khatib, 0. and Burdick, J.: “Optimization of dynamics in manipulator design: the operational space formulation”, Znt. J. Robotics and Auto., Vol.2, May 1987, No.2, pp. 90-98.

[16] Graettinger, T. J. and Krogh, B. H.: ”The acceleration radius: a global performance measure for robotic manipulators”, ZEEE J. Robotics and Auto., Vo1.4, No. l, Feb. 1988, pp. 60-69.

[17] P.A. Voglewede, I. Ebert-Uphoff: “Measuring closeness to singularities for parallel manipulators”, Proc. IEEE Int. Conf. Robot. Autom. (New Orleans), 2004 , pp. 4539–4544.

[18] A. Bowling, O. Khatib: “The dynamic capability equations: a new tool for analyzing robotic manipulator performance”, IEEE Trans. Robot. 21(1), 2005, pp. 115–123.

[19] O.Ma and J. Angeles: “The concept of dynamic isotropy and its applications to inverse kinematics and trajectory planning,” in Proc. IEEE Int.Conf. Robot. Autom., Vol. 1, 1990, pp. 481–486.

[20] P.K. Bhatti: “Probabilistic modeling and optimal design of robotic manipulators”, PhD Dissertation, Purdue University, West Lafayette, Indiana, December 1989.

[21] S. Rao and P.K. Bhatti: “Probabilistic approach to manipulator kinematics and dynamics”, Sciencedirect, Reliability Engineering and System Safety 72 , 2001, pp. 47-58.

[22] P. Herman: “Energy-Based Indices for Manipulator Dynamics Improvement”, Springer, Journal of Intelligent and Robotic Systems, Vol. 44, 2005, pp. 313–325.

[23] A. Bowling and C. Kim: “Velocity Effects on Robotic Manipulator Dynamic Performance”, Journal of Mechanical Design, November 2006, Vol. 128 / 1245

[24] Larson HJ, Shubert BO.: “Probabilistic models in engineering sciences”, vol. 1. New York: Wiley, 1979, pp. 108-13.

[25] Loduha, T. A. and Ravani, B.: “On first-order decoupling of equations of motion for constrained dynamical systems”, Transactions of the ASME J. Appl. Mec. 62, 1995, pp. 216–222.

[26] Box GEP, Jenkins GM.: “Time series analysis: forecasting and control”, Holden-Day, 1976, pp. 56-58.

[27] Mazen Zein: “Analysis cinématique des manipulateurs sériels 3R orthogonaux et des manipulateurs parallèles plans”, PhD Thesis, Ecole Centrale de Nantes, 2007.

[28] Félix Majou: “Analysis cinétostatique des machines parallèles à translations”, PhD Thesis, Université Laval, Québec, 2004.

[29] Peter Corke, Peter I. Corke Brian Armstrong-h'elouvry: “A search for consensus among model parameters reported for the PUMA 560 robot”, Proc. IEEE Int. Conf. Robotics and Automation, 1994, pp. 1608-1613.

[30] D.J. Walton, D.S. Meek: “A controlled clothoid spline”, Sciencedirect, Computers & Graphics 29, 2005, pp. 353–363.

[31] Li Guiqing, Li Xianmin, Li Hua: “3D discrete clothoid splines”, Proc. IEEE in Computer Graphics, 2001, pp. 321-324.

[32] Wisama Khalil, Étienne Dombre: “Modeling, Identification and Control of Robots”, Kogan Page Science

[33] Bruno Siciliano, Oussama Khatib: “Handbook of Robotics”, Springer 2008, pp. 232.

[34] F.P. Beer, E.R. Johnston, D.F. Mazurek, P.J. Cornwell, E.R. Eisenberg: “Vector mechanics for engineers: Statics and Dynamics”, 9th edition, McGraw-Hill, 2010.

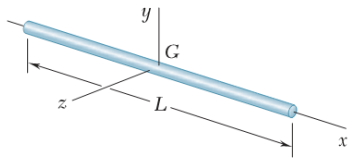
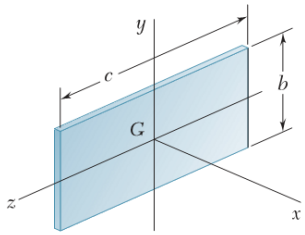
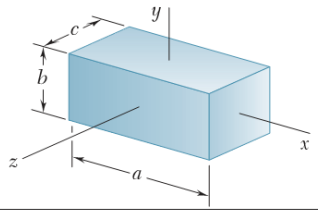
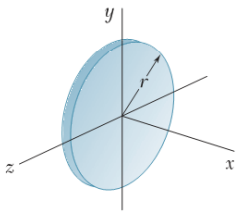
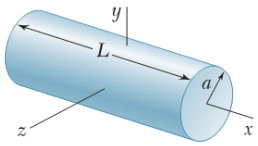
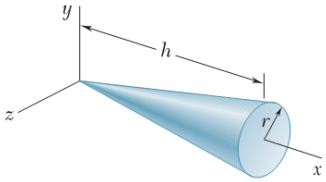
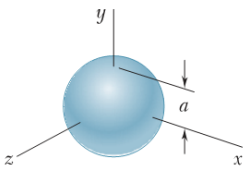
[35] ADAMS guide

[36] MATLAB guide

[37] SYMORO+ guide

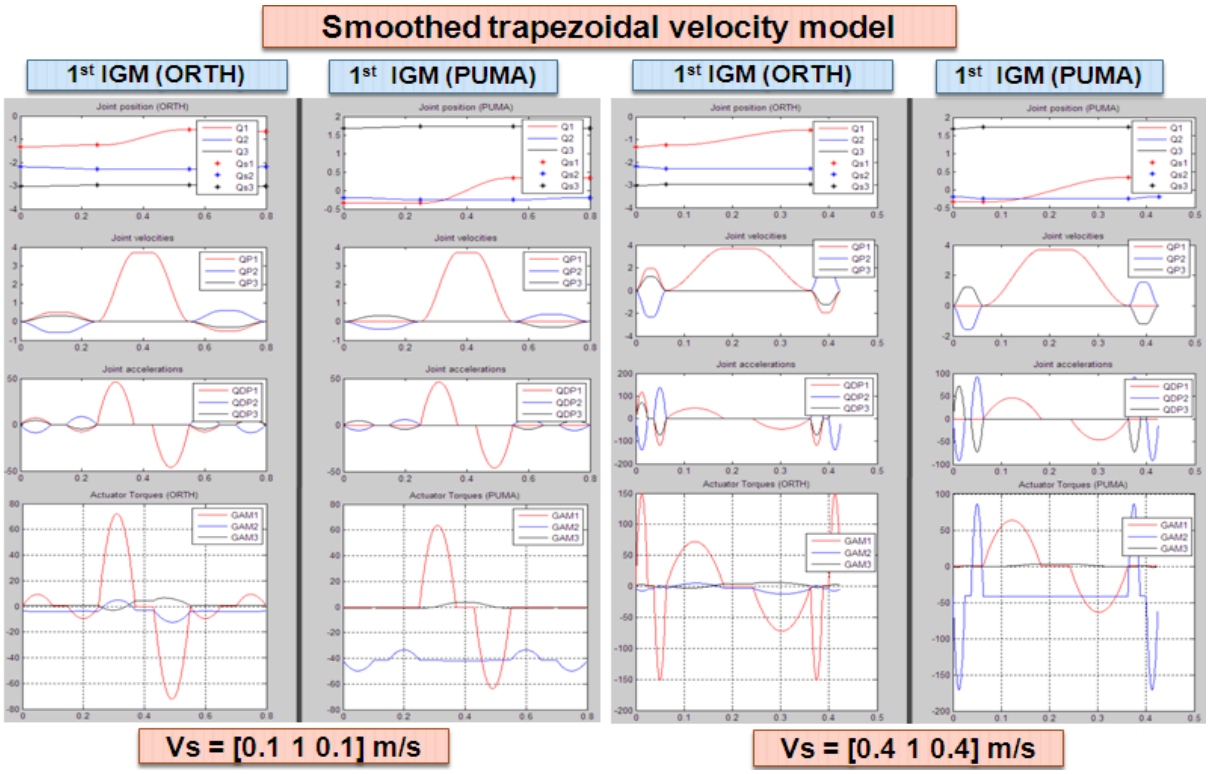
APPENDIX

A1. Mass moments of inertia of common geometric shapes [34]

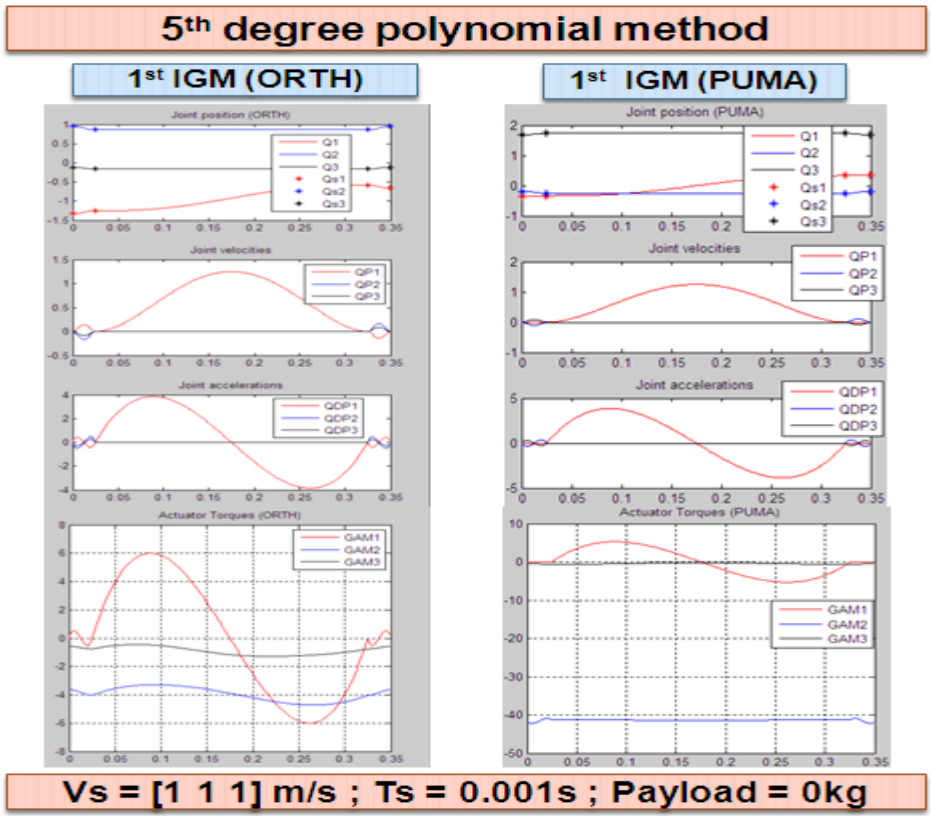
Slender rod		$I_y = I_z = \frac{1}{12}mL^2$
Thin rectangular plate		$I_x = \frac{1}{12}m(b^2 + c^2)$ $I_y = \frac{1}{12}mc^2$ $I_z = \frac{1}{12}mb^2$
Rectangular prism		$I_x = \frac{1}{12}m(b^2 + c^2)$ $I_y = \frac{1}{12}m(c^2 + a^2)$ $I_z = \frac{1}{12}m(a^2 + b^2)$
Thin disk		$I_x = \frac{1}{2}mr^2$ $I_y = I_z = \frac{1}{4}mr^2$
Circular cylinder		$I_x = \frac{1}{2}ma^2$ $I_y = I_z = \frac{1}{12}m(3a^2 + L^2)$
Circular cone		$I_x = \frac{3}{10}ma^2$ $I_y = I_z = \frac{3}{5}m(\frac{1}{4}a^2 + h^2)$
Sphere		$I_x = I_y = I_z = \frac{2}{5}ma^2$

A2. Example results in basic analysis

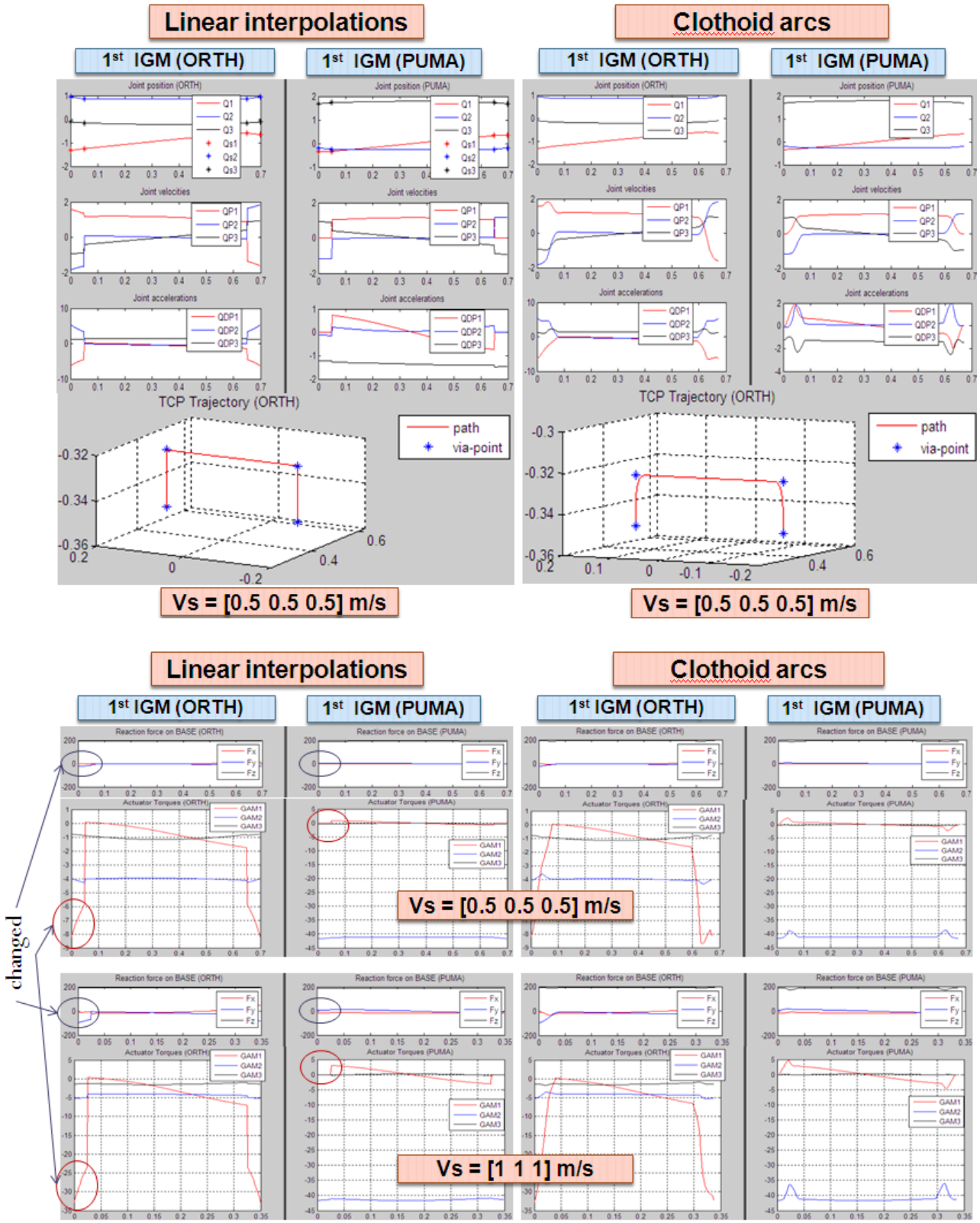
Results in Red-Circle area



This is the reason why we don't use (smoothed) trapezoidal velocity model in testing because the inverse geometric solutions are too sensitive for both robots, even with small Vs.

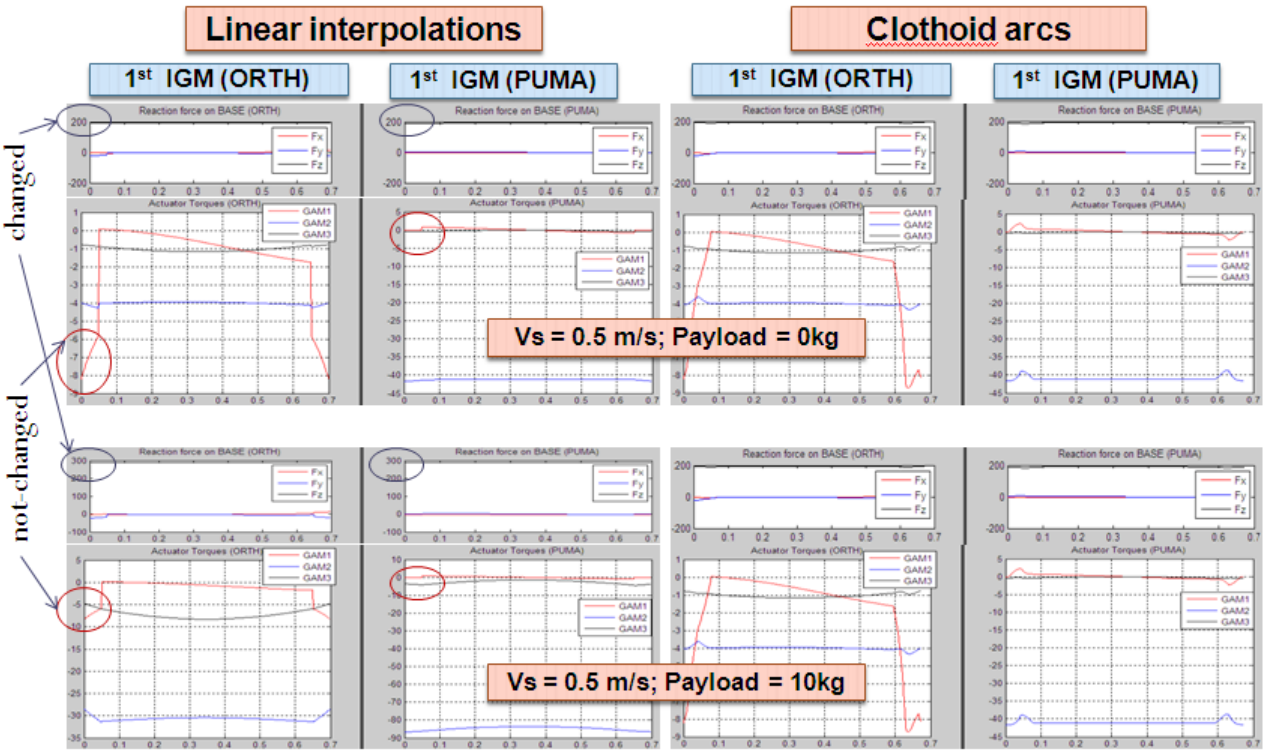


With 5th degree method, the average torque values is quite close to the static profile.



In this test, we keep the Mass = const = 0kg, but changing Vs.

Notice the changes of reaction forces and actuator torques.



In this test, we keep the Vs = const and change the payload. Notice again the changes of reaction forces and torques.

These properties are maintained at different areas in the workspace.

A3. Debugging optimization procedure

Objective function:

Remember our problem is a multi-objective case:

$$\begin{cases} \sum^{\text{joint 1}} abs\left(T_{k\max}\right) \rightarrow \max \quad \left(T_{k\max} = \max\left(Torques\ \ of\ \ segment\ k^{th}\right)\right) \\ \sum^{\text{joint 2}} abs\left(T_{k\max}\right) \rightarrow \max \\ \sum^{\text{joint 3}} abs\left(T_{k\max}\right) \rightarrow \max \end{cases}$$

In our case, we modify the objective as follow:

```
J =  -[sum(Tmax(1,:))/pT(1)/(m-1)-1
       sum(Tmax(2,:))/pT(2)/(m-1)-1
       sum(Tmax(3,:))/pT(3)/(m-1)-1];
```

In this way, insteads of computing the sum of maximum torques, we compute the average of maximum torques of each joint at all segments and normalize these values (pT is the limit torques vector). In this case, the ideal solution will be:

$$J = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

By normalizing the objective function to smaller values, we can obtain more stable solutions. In **fgoalattain** options, we set our goals: try to suppress all elements of J as close to zeros as possible.

Constraints:

To check if the optimization running in good direction (converging or reaching constraint boundaries) we define an error matrix between the current maximum torque values with the limit torques of each joint (pT):

```
errT = Tmax - [ones(1,m-1)*pT(1); ones(1,m-1)*pT(2); ones(1,m-1)*pT(3)]
```

and check also the average operational velocities of the end-effector to see if they are close to the constraints or not. Under our constraints, all elements of `errT` must be less than 0, and any element of `V` must be less than `Vmax.`

The `ck` value is to check if any point of the trajectory is outside the working space or not – if (ck < 0) then the trajectory is inside the workspace.

Normally, when running optimization, we initiate the searching parameters as random values. However, for init-values that give us the `errT` and `V` around or less than the limit values will most likely tend to converge to a local minimum.

In out optimization results, most of the cases result in local minimum solutions, in a few cases, we keep the results even they are not a local minimum, but because they reach many constraint boundaries so could be considered as good solutions.

For example:

```
================================================================
Optimize for ORTH first - 2nd BigRegion - 5th Degree - Task-space

Tmax = [80 80 30]; % (N/m)
Vmax = [1.3 1.3 1.3 1.3]; % (m/s)

================================================================
errT =

  -69.7080  -77.2119  -34.1581  -37.5224
  -56.6824  -54.4820  -50.8997  -56.5363
  -22.4375  -25.2014   -9.5529   -7.0909


V =

    1.2105    1.3000    0.8457    1.3000


ck =

 -3.4578e-004

Local minimum possible. Constraints satisfied.
```

In this case, we found a local minimum solution. With the speeds at segment 2 and 4 reach maximum values (1.3000 m/s).

In the following case, we did not obtain a local minimum solution but still keep it because some of the values reach their limits (in case of `errT`, values close to 0 are reaching maximum limit).

```
================================================================
Optimize for PUMA first - 3rd BigRegion - 5th Degree - Task-space

Tmax = [80 80 30]; % (N/m)
Vmax = [1.2 1.2 1.2 1.2]; % m/s

================================================================
errT =

  -66.9985  -76.2740  -71.1193   -3.5461
   -1.2016   -0.6082  -23.3325   -3.6187
   -5.4920   -5.3876   -7.0974    0.0002


V =

    1.2002    0.7182    1.2010    0.9571


ck =

 -2.4662e-007

Solver stopped prematurely.

fgoalattain stopped because it exceeded the iteration limit,
options.MaxIter = 1000 (the selected value).
```